



**epsc**

**Escola Politècnica Superior  
de Castelldefels**

UNIVERSITAT POLITÈCNICA DE CATALUNYA

# TREBALL DE FI DE CARRERA

**TÍTOL DEL TFC: Xarxa cel·lular de 4G basada en IPv6: desenvolupament d'un demostrador (II)**

**TITULACIÓ: Enginyeria Tècnica de Telecomunicació, especialitat Telemàtica**

**AUTORS: Jaume Giribert Peraire  
Miguel Madrid Villar**

**DIRECTOR: Rafael Vidal Ferré**

**DATA: 8 de maig de 2009**



**Títol:** Xarxa cel·lular de 4G basada en IPv6: desenvolupament d'un demostrador (II)

**Autors:** Jaume Giribert Peraire  
Miguel Madrid Villar

**Director:** Rafael Vidal Ferré

**Data:** 8 de maig de 2009

## **Resum**

En l'escenari actual de mobilitat i deslocalització del treball, la tecnologia mòbil és cada vegada més freqüent i sol·licitada. En aquest context, es fa necessària la recerca de mètodes per a oferir la ubiqüitat dels serveis, i alhora prolongar la vida de dispositius que permetin combinar aquesta necessitat de mobilitat amb l'eficiència energètica i el respecte pel medi ambient.

En aquest TFC s'ha pres com a punt de partida la combinació d'IEEE802.11 amb el protocol Mobile IPv6 per a aconseguir superar les seves limitacions a l'hora d'admetre la mobilitat entre xarxes. A partir d'aquí, l'objectiu és completar aquesta solució amb un seguit de propostes per a reduir el consum de bateria.

La primera és l'ús del que es coneix com a IP Paging, una extensió del protocol Mobile IPv6 encaminada a reduir la seva senyalització. La segona consisteix a combinar l'IP Paging amb els mecanismes d'estalvi d'energia propis d'IEEE802.11, l'original PSM i la seva extensió posterior, anomenada U-APSD (*Unscheduled Automatic Power Save Delivery*), amb la finalitat de reduir encara més el consum derivat de la mobilitat.



**Title:** 4G cellular networking based on IPv6: development of a testbed (II)

**Authors:** Jaume Giribert Peraire  
Miguel Madrid Villar

**Director:** Rafael Vidal Ferré

**Date:** May, 8th 2009

## Overview

In the current scenario of delocalization and mobility of work, mobile technology is increasingly required. In this context, it is necessary to search for ways to provide service ubiquity, while prolonging the life of devices in a way that combines mobility with energy efficiency and protection of the environment.

In this project it has been taken as a starting point the combination of IEEE802.11 with the Mobile IPv6 protocol to achieve their limitations when supporting mobility between networks. From there, the goal is to achieve a solution with a series of proposals to reduce power consumption.

The first one is the use of IP Paging, an extension of Mobile IPv6 protocol to reduce the signaling. The second one is the combination of IP Paging with energy saving mechanisms of the standard IEEE802.11, the original PSM and its subsequent extension, called U-APSD (Unscheduled Automatic Power Save Delivery), in order to reduce the consumption of mobility protocols.

En primer lloc, voldríem agrair al nostre director, Rafael Vidal, el seu suport i confiança a l'hora de dur a terme aquest projecte, així com l'esforç que hi ha dedicat.

En segon lloc, una especial dedicatòria a les nostres xicotes, Judith i Mireia, per la seva inestimable ajuda i comprensió. Sense elles no hauríem pogut fer-ho mai. També un fort agraïment per els nostres pares, gràcies pel suport del dia a dia durant tot aquest temps.

I per últim, voldríem agrair tant als nostres amics de sempre com als de la feina i la universitat les bones estones que han fet més amena la realització d'aquest treball.



# ÍNDIX

<b>INTRODUCCIÓ</b> .....	<b>3</b>
<b>CAPÍTOL 1. SOLUCIONS PER A L'ESTALVI D'ENERGIA</b> .....	<b>7</b>
<b>1.1. Mobile IP versió 6</b> .....	<b>7</b>
I.3.1. Entitats implicades a MIPv6 .....	8
I.3.2. Missatges del protocol.....	9
I.3.3. Neighbor Discovery .....	10
I.3.4. Dynamic Home Agent Address Discovery (DHHAD) .....	11
I.3.5. Funcionament del protocol .....	11
I.3.6. Altres funcionalitats .....	12
<b>1.2. IP Paging</b> .....	<b>13</b>
I.3.1. Estructura de les àrees de paging.....	13
I.3.2. Algoritme de paging basat en distància .....	15
I.3.3. Aplicació d'IP Paging a Mobile IP versió 6 .....	15
<b>1.3. Mecanisme d'estalvi d'energia per a xarxes IEEE802.11</b> .....	<b>16</b>
I.3.1. Mecanisme d'estalvi d'energia: PSM .....	16
I.3.2. Mecanisme d'estalvi d'energia: U-APSD.....	18
<b>1.4. Combinació dels mecanismes d'estalvi d'energia PSM i U-APSD i IP Paging per a MIPv6</b> .....	<b>21</b>
<b>CAPÍTOL 2. MODIFICACIONS A LA IMPLEMENTACIÓ DEL DIMONI UMIP</b> <b>23</b>	
<b>2.1. Modificacions requerides</b> .....	<b>23</b>
2.1.1. Modificacions i implementacions existents.....	23
2.1.1.1. Modificacions i implementacions generals .....	23
2.1.1.2. Modificacions a l'MN.....	24
2.1.1.3. Implementacions a l'MN .....	26
2.1.1.4. Modificacions a l'HA .....	27
2.1.1.5. Implementacions a l'HA.....	27
2.1.2. Modificacions i implementacions creades .....	28
2.1.2.1. Reducció de trànsit a l'MN.....	28
2.1.2.2. Supressió de Router Advertisements .....	29
2.1.2.3. Definició i lectura del nou SSID .....	30
2.1.2.4. Supressió de Neighbor Discovery .....	32
2.1.2.5. L'MN no forma CoA .....	33
<b>2.2. Implementacions realitzades</b> .....	<b>34</b>
<b>CAPÍTOL 3. PROVES DE VALIDACIÓ DE MIPV6</b> .....	<b>37</b>
<b>3.1. Pla de proves</b> .....	<b>37</b>
3.1.1. Prova 1: l'MN canvia a l'estat dormant .....	38
3.1.1.1. Resultats esperats.....	39
3.1.2. Prova 2: l'MN canvia a l'estat actiu .....	39
3.1.2.1. Resultats esperats.....	39
3.1.3. Prova 3: l'MN avisa correctament l'HA per canvi d'àrea de paging .....	40
3.1.3.1. Resultats esperats.....	40
3.1.4. Prova 4: comportament de l'HA per a localitzar l'MN.....	40
3.1.4.1. Resultats esperats.....	41
3.1.5. Prova 5: retard.....	41



3.1.5.1. Resultats esperats .....	41
<b>3.2. Resultats obtinguts .....</b>	<b>41</b>
3.2.1. Prova 1: l'MN canvia a l'estat dormant .....	43
3.2.2. Prova 2: l'MN canvia a l'estat actiu .....	45
3.2.3. Prova 3: l'MN avisa correctament l'HA per canvi d'àrea de paging .....	48
3.2.4. Prova 4: comportament de l'HA per a localitzar l'MN.....	51
3.2.5. Prova 5: retard.....	51
<b>3.3. Conclusions .....</b>	<b>53</b>
<b>CAPÍTOL 4. ESTUDI DE L'IMPACTE DE L'HABILITACIÓ D'U-APSD I PSM SOBRE LA BATERIA D'UN DISPOSITIU MÒBIL .....</b>	<b>55</b>
<b>4.1. Escenari de proves.....</b>	<b>55</b>
<b>4.2. Metodologia emprada .....</b>	<b>59</b>
4.2.1. Connexió dels dispositius .....	59
4.2.2. Configuració dels paràmetres dels dispositius .....	60
4.2.3. Estat dels dispositius durant les proves .....	61
4.2.4. Tipificació de les proves .....	61
<b>4.3. Relació entre consum i durada de la bateria .....</b>	<b>62</b>
<b>4.4. Proves i resultats.....</b>	<b>63</b>
4.4.1. Impacte de l'activació de la interfície WLAN .....	64
4.4.2. Efecte del trànsit multicast/broadcast a la xarxa .....	69
4.4.3. Impacte de l'aplicació de trànsit cap a / des de l'estació de proves.....	81
4.4.4. Avaluació del consum d'un terminal mòbil (STA 4).....	87
4.4.5. Proves de consum sobre el dimoni UMIP modificat.....	89
<b>CAPÍTOL 5. CONCLUSIONS .....</b>	<b>109</b>
<b>5.1. Fases del projecte i resultats obtinguts.....</b>	<b>109</b>
<b>5.2. Futurs desenvolupaments.....</b>	<b>111</b>
<b>5.3. Impacte mediambiental del projecte .....</b>	<b>112</b>
<b>CAPÍTOL 6. ABREVIACIONS .....</b>	<b>113</b>

## INTRODUCCIÓ

En l'escenari actual de mobilitat i deslocalització del treball, la tecnologia mòbil és cada vegada més freqüent i sol·licitada. En aquest context, es fa necessària la recerca de mètodes per a oferir la ubiqüitat dels serveis, i alhora prolongar la vida de dispositius que permetin combinar aquesta necessitat de mobilitat amb l'eficiència energètica i el respecte pel medi ambient.

En aquest panorama destaca amb força la tecnologia sense fils coneguda popularment com a WiFi i estandarditzada per l'IEEE com a 802.11. Aquesta tecnologia és dominant absoluta del mercat del PC portàtils, les agendes personals i les consoles de videojocs, i darrerament està entrant amb força en el món dels telèfons mòbils.

Entre els punts forts d'aquesta tecnologia destaquen les seves elevades velocitats d'accés i la seva facilitat de desplegament, ja que utilitza bandes d'espectre per a les quals no cal pagar. Per contra, dos dels seus punts febles són el seu reduït suport a la mobilitat i, inicialment, un mecanisme d'estalvi d'energia conegut com a PSM (*Power Save Mode*) força limitat i incompatible amb l'ús d'aplicacions amb temps real.

En aquest TFC s'ha pres com a punt de partida la combinació d'IEEE802.11 amb el protocol Mobile IPv6 per a aconseguir superar les seves limitacions a l'hora d'admetre la mobilitat entre xarxes. A partir d'aquí, l'objectiu és completar aquesta solució amb un seguit de propostes per a reduir el consum de bateria.

La primera és l'ús del que es coneix com a IP Paging, una extensió del protocol Mobile IPv6 encaminada a reduir la seva senyalització. La segona consisteix a combinar l'IP Paging amb els mecanismes d'estalvi d'energia propis d'IEEE802.11, l'original PSM i la seva extensió posterior, anomenada U-APSD (*Unscheduled Automatic Power Save Delivery*), amb la finalitat de reduir encara més el consum derivat de la mobilitat.

Finalment, també és objectiu d'aquest projecte que la utilització combinada de Mobile IPv6 amb IP Paging i PSM o U-APSD permetin que un terminal mòbil que no estigui actiu ignori completament tot el trànsit multicast i broadcast sense que això suposi la pèrdua de la seva connectivitat. L'acompliment d'aquest objectiu en porta un altre d'associat, que és mesurar quin estalvi de bateria suposa ignorar aquest trànsit.

Per a desenvolupar aquest estudi s'ha consultat altra documentació relacionada amb aquest camp i s'ha pres com a punt de partida el treball Xarxa Cel·lular de 4G basada en IPv6: desenvolupament d'un demostrador de Víctor Rubio Albadalejo (a partir d'ara, citat com a [11]) i el treball Estudio del consumo de baterías en dispositivos móviles IEEE802.11: técnicas de ahorro, caracterización y evaluación de Joshua Sanz de la Rica (a partir d'ara, citat com a [17]).

Atenent als objectius generals esmentats anteriorment i al punt de partida que suposen els treballs citats, en aquest projecte s'han marcat com a objectius:

- Actualitzar el demostrador ja existent a [11] amb una nova implementació.
- Crear un nou manual d'instal·lació amb la nova implementació del dimoni per al suport de MIPv6.
- Afegir a la nova implementació triada les modificacions creades a [11] retocant el codi font del nou dimoni, i aconseguir les mateixes funcionalitats.
- Afegir al nou codi creat les noves millores d'IP Paging:
  - Suprimir el trànsit multicast.
  - Aconseguir que el node mòbil no formi una nova adreça quan canvia de xarxa sense canviar d'àrea de paging.
  - Enviar els missatges del protocol MIPv6 directament a la MAC del node mòbil.
  
- Estudiar els projectes de codi lliure existent per a tenir suport per a les targetes amb el protocol IEEE 802.11e.
- Crear un manual per a realitzar mesures amb el dispositiu analitzador de potències Agilent N6705A.
- Dissenyar un escenari de proves per als mecanismes d'estalvi d'energia PSM i U-APSD, aprofitant la versatilitat del dispositiu Agilent N6705A.
- Quantificar els mecanismes d'estalvi d'energia PSM i U-APSD en un entorn de trànsit VoIP amb el dispositiu Agilent N6705A.
- Realitzar proves empíriques per a comprovar l'estalvi d'energia en el demostrador MIPv6 modificat, en conjunt amb els mecanismes d'estalvi d'energia PSM i U-APSD amb el dispositiu Agilent N6705A.

Així, doncs, l'estructura del treball és la següent:

En el Capítol 1 s'introdueix la base teòrica sobre la qual es construeixen els diferents escenaris que componen aquest estudi.

Es distingeixen dues vies: d'una banda, la continuació de [11], que consisteix en la implementació d'un demostrador MIPv6 mitjançant un dimoni de Linux, així com la teoria que permet treballar posteriorment amb el model d'IP Paging.

D'una altra banda, es presenten dos mecanismes d'estalvi d'energia: PSM, introduït per l'IEEE 802.11, i U-APSD, que neix per la creixent demanda de millorar la capacitat de QoS, necessària per a les comunicacions de VoIP i que ha estat introduïda a l'estàndard 802.11e. En base a [17], s'ha exposat com aquests mecanismes, juntament amb el demostrador MIPv6 modificat, suposen una mesura d'estalvi que compleix els objectius d'aquest treball.

Una vegada descrits els mecanismes que permeten dur a terme l'estudi, en el Capítol 2 s'analitza la implementació anterior de MIPv6 i la necessitat d'actualitzar-la (ja que ha deixat de tenir suport), així com el procés per a dur a terme aquesta actualització. Aquesta migració a una nova versió de MIPv6 ha fet necessària la modificació de codi existent, i en el Capítol 2 també es documenta i es detalla tot aquest procés. Finalment, en aquest capítol es

descriu com s'ha modificat l'IP Paging per a eliminar el trànsit multicast a l'escenari de proves, amb l'objectiu d'augmentar la durada de la bateria dels dispositius mòbils de prova.

A continuació, el Capítol 3 ofereix una bateria de proves en què es comprova que la maqueta modificada al Capítol 2 funcioni correctament.

Seguidament, en el Capítol 4 s'analitzen, a través d'una bateria de proves, els diferents mecanismes d'estalvi d'energia, els quals s'han aplicat en primer lloc a una sèrie d'escenaris que simulen diverses situacions de trànsit a la xarxa, per a finalment comprovar-ne l'impacte en conjunt amb les modificacions realitzades al Capítol 2.

Finalment, en el Capítol 5 s'exposen les conclusions i les possibles línies futures de treball d'aquest projecte.

Per últim, s'inclouen una sèrie d'annexos que complementen la informació proporcionada i ofereixen abundants dades addicionals sobre els dispositius utilitzats en aquest estudi, tant pel que fa a la seva configuració com al programari implicat.



## CAPÍTOL 1. Solucions per a l'estalvi d'energia

En aquest primer capítol es fa una introducció a les tecnologies emprades en aquest projecte per a estalviar energia en els dispositius mòbils que utilitzen el protocol IEEE802.11 com a capa d'enllaç i el protocol Mobile IP versió 6, que permet el desplaçament de nodes mòbils entre diferents xarxes sense necessitat de reconfiguració i amb independència de la seva capa d'enllaç.

En primer lloc s'expliquen les bases de funcionament del protocol Mobile IP versió 6. A banda del suport a la mobilitat, aquest protocol permet mantenir vàlida una adreça de xarxa durant un període definit per l'usuari i aplicar durant aquest temps tècniques d'estalvi d'energia en cas de no rebre o enviar trànsit.

Concretament, es detalla la integració d'una d'elles: IP Paging en Mobile IP versió 6. IP Paging permet cercar terminals en un grup de xarxes organitzades de manera jerarquitzada i, d'aquesta manera, estalviar l'enviament de trànsit per part dels nodes mòbils cap a l'HA per tal de ser localitzats.

Finalment, a l'últim apartat s'exposa el funcionament de dos mecanismes d'estalvi d'energia vinculats a l'estàndard IEEE802.11 que responen a la necessitat cada vegada major de dur a terme una gestió intel·ligent del consum dels diferents dispositius mòbils de comunicacions, sobretot a partir de la implantació cada vegada més estesa de les comunicacions VoIP.

### 1.1. Mobile IP versió 6

El Mobile IP versió 6 (d'aquí en endavant, MIPv6) és un protocol de suport a la mobilitat que permet que un terminal mòbil es desplaci entre xarxes diferents conservant sempre una mateixa adreça de xarxa.

Sense l'MIPv6, quan un node canvia de xarxa i vol obtenir connectivitat, reconfigura la seva interfície de manera que en primer lloc esborra la configuració anterior i tot seguit en crea una de nova en base als paquets que hagi rebut a la nova xarxa (configuració DHCP, *Dynamic Host Configuration Protocol*).

El fet d'esborrar la configuració anterior i crear-ne una de nova provoca la pèrdua immediata de qualsevol connexió establerta anteriorment pel terminal mòbil, ja que la nova adreça de xarxa és diferent.

Atès que el protocol IP determina la destinació dels paquets per la informació de xarxa (adreça de destinació IP), els paquets enrutats cap a la destinació anterior es descarten automàticament, ja que l'adreça anterior deixa de ser vàlida.

Per tal de resoldre aquest problema, la IETF va proposar l'estàndard Mobile IP (RFC 2002) ([1]), que permet que un mateix terminal tingui dues adreces en una mateixa interfície: una d'única i la creada dins la nova xarxa. També es van definir els mecanismes per a relacionar-les (*binding*). Més endavant, aquest estàndard es va estendre per a donar suport a IPv6 ([3]), creant així el protocol MIPv6 (RFC 3775) ([2]).

Així doncs, amb un terminal amb suport MIPv6 es fa possible moure's entre xarxes (sempre que tinguin suport IPv6) sense perdre les connexions establertes i amb independència de la capa d'enllaç, ja que els paquets enrutats cap a la xarxa anterior es poden enviar correctament cap a la nova.

La intenció d'aquest protocol és aconseguir que la mobilitat entre xarxes sigui transparent per als protocols de les capes superiors amb un mínim de canvis a la infraestructura existent.

### I.3.1. Entitats implicades a MIPv6

Per tal de facilitar la lectura d'aquest document, a continuació es desglossen alguns termes relacionats amb el protocol MIPv6. Aquests termes també es mostren a la figura 1.1:

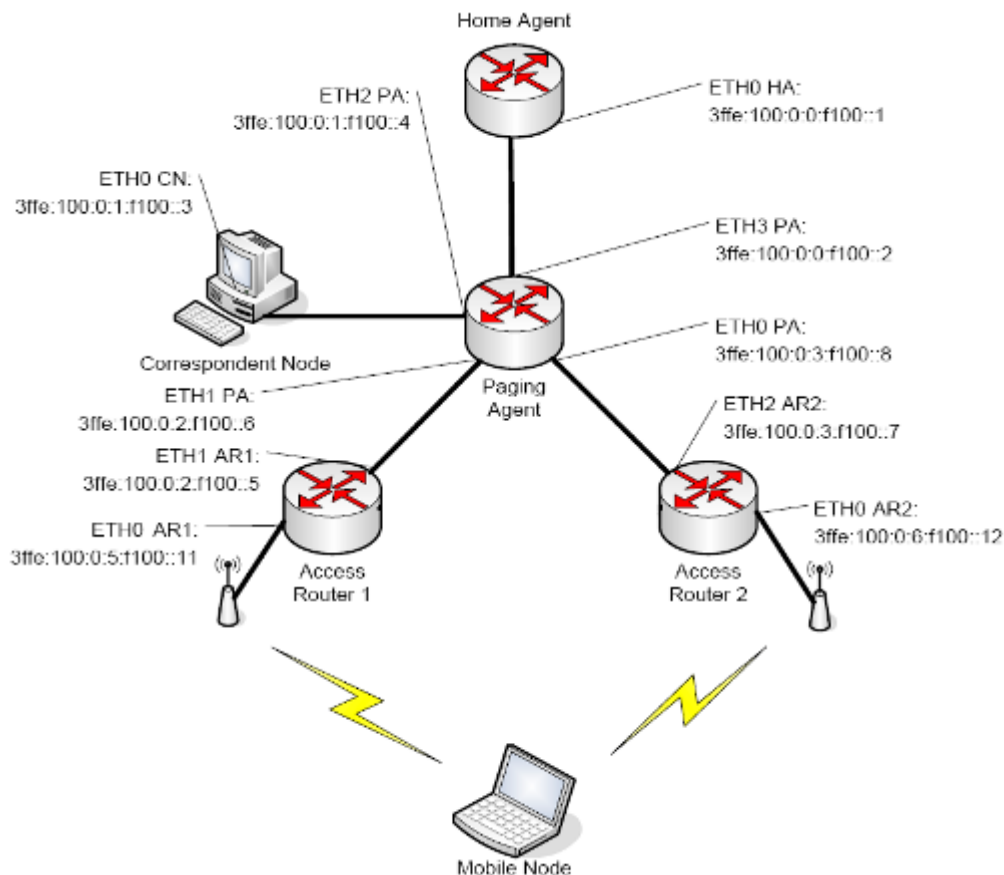


Fig. 1.1. Topologia d'una xarxa amb suport de MIPv6

- **Mobile Node (MN):** terminal mòbil que canvia de xarxa.
- **Access Router (AR):** router de la xarxa visitada pel Mobile Node.
- **Home Agent (HA):** agent (per norma general un router) que redirecciona el trànsit dirigit cap al Mobile Node quan no es troben a la mateixa xarxa. Per a enviar-li el trànsit, es crea un túnel IPv6-IPv6 entre ambdós.
- **Correspondent Node (CN):** terminal que es comunica amb el Mobile Node i és diferent del Home Agent.
- **Home Address (HoA):** adreça fixa del Mobile Node a totes les xarxes assignada per norma general pel Home Agent.
- **Home Link (HL):** enllaç on es troba el Home Agent i al qual pertany la Home Address.
- **Foreign Link (FL):** enllaç on es troba el Mobile Node i diferent del Home Link.
- **Care-of-Address (CoA):** adreça de xarxa assignada al Mobile Node per norma general per el router d'un Foreign Link.
- **Binding:** relació temporal entre una Home Address i una Care of Address del Mobile Node que s'envia al Home Agent i/o al Correspondent Node.
- **Binding Lifetime (BL):** temps durant el qual un binding és vàlid.
- **Binding Cache (BC):** cache del binding que emmagatzema el Home Agent i/o el Correspondent Node.
- **Binding Update List (BUL):** cache emmagatzemada pel Mobile Node que controla el temps durant el qual és vàlid un binding i quines Care-of-Address utilitza amb els Correspondent Nodes.

### I.3.2. Missatges del protocol

Els missatges que defineix el protocol MIPv6 són els següents:

- **Binding Update (BU):** paquet enviat pel Mobile Node cap al Home Agent i/o Correspondent Node per a definir el temps vàlid d'un binding. S'enviarà a cada Lifetime de l'MN o quan canviï de xarxa.
- **Binding Acknowledge (BAck):** missatge de resposta al Binding Update que indica una recepció correcta del paquet esmentat.



- **Binding Refresh Request (BRR):** paquet enviat pel Home Agent i/o el Correspondent Node cap al Mobile Node per a obligar-lo a enviar un paquet Binding Update.

Tots tres paquets s'envien amb l'extensió de la capçalera d'IPv6 de mobilitat (tipus de capçalera 59) definit a [2].

### I.3.3. Neighbor Discovery

Els termes següents no són propis del protocol MIPv6, sinó que formen part del mecanisme Neighbor Discovery emprat a IPv6 i que utilitza missatges ICMPv6.

- **Router Solicitation (RtrSol):** missatge generat per un terminal per a identificar el router de l'enllaç al qual es troba connectat.
- **Router Advertisement (RtrAdv):** paquet generat pel router amb informació sobre ell mateix (temps de vida, interval dels missatges, prefix de xarxa que anuncia). Aquest missatge és el que utilitza el Mobile Node per a crear la seva Care-of-Address quan es troba en un Foreign Link. S'envia a tots els nodes automàticament o com a resposta a una Router Solicitation.
- **Neighbor Solicitation (NghSol):** missatge enviat per un terminal per a determinar l'adreça d'enllaç dels seus veïns.
- **Neighbor Advertisement (NghAdv):** paquet generat com a resposta a una Neighbor Solicitation. Una interfície també pot generar-lo automàticament si canvia d'adreça d'enllaç.

Tots aquests missatges s'expliquen més detalladament a [5].

Durant el treball, s'utilitzaran indistintament les abreviacions i les seves formes esteses corresponents.

A més, al Capítol 6 es pot consultar una llista d'abreviacions (sigles i acrònims) i les seves formes esteses corresponents.

### **I.3.4. Dynamic Home Agent Address Discovery (DHAAD)**

El DHAAD és el mecanisme que permet a l'MN descobrir els HA dins del Home Link mitjançant els missatges DHAAD request (de l'MN cap a tots els HA) i DHAAD request (de l'HA cap a l'MN amb una llista dels HA).

Aquest mateix mecanisme és el que permet conèixer el prefix del Home Link per mitjà dels missatges ICMP Mobile Prefix Solicitation (de l'MN a l'HA) i Mobile Prefix Advertisement (de l'HA cap a l'MN). D'aquesta manera és possible actualitzar el prefix de manera dinàmica.

### **I.3.5. Funcionament del protocol**

Tal com s'ha explicat, el protocol MIPv6 es basa en l'ús de dues adreces de xarxa per tal d'identificar una interfície. La CoA és l'adreça creada a l'MN pel fet de visitar una nova xarxa, mentre que l'HoA és l'adreça creada per l'HA i que servirà per a localitzar l'MN fora de les Home Link.

Una vegada s'executa el protocol, l'MN llegeix els RtrAdv enviats pel router de l'enllaç al qual es troba connectat. Amb el prefix enviat pel RtrAdv, crea la seva CoA tal com es defineix a [4] i comprova que estigui en un Foreign Link o al Home Link (comprovant si el prefix de l'adreça de xarxa és igual al del seu HA).

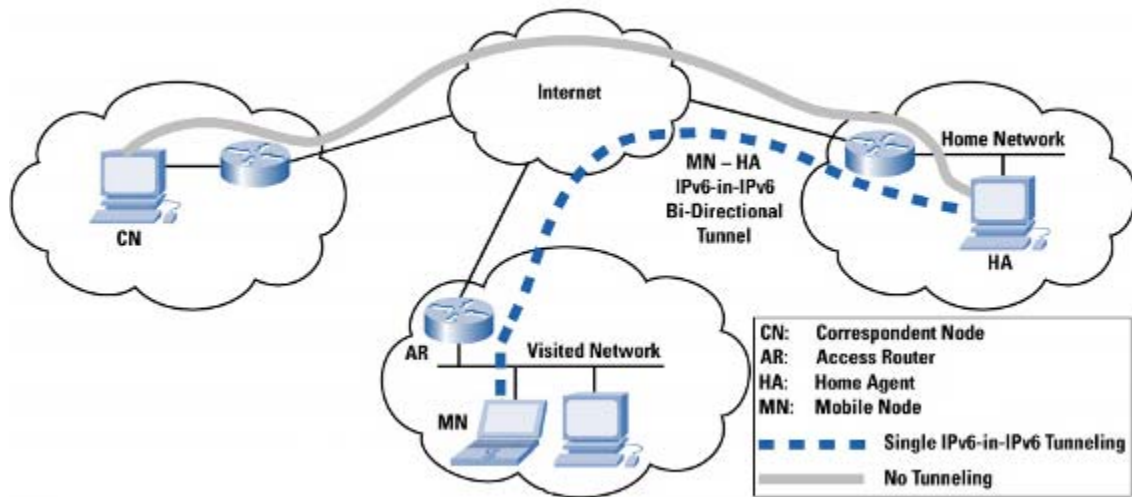
En cas de trobar-se en un Home Link, el funcionament és el que defineix IPv6 [3]. Si en canvi es troba en un Foreign Link, assigna automàticament a la seva interfície una CoA (vegeu l'Annex III) i envia un paquet BU cap al seu HA indicant quina és la seva CoA i el lifetime de l'MN.

Quan l'HA rep el BU, escriu a la seva Binding Cache un nou registre d'MN amb els paràmetres següents:

- Lifetime: el temps de vida de l'MN en segons. Si passat aquest temps l'HA no ha rebut un nou BU, l'MN s'esborrarà.
- La CoA de l'MN: l'adreça que utilitzarà l'HA per a enrutar els paquets dirigits/provinents de l'MN a nivell IP.
- L'HoA de l'MN: l'adreça vinculada a la CoA, vàlida durant un lifetime.

Si tot és correcte, l'HA envia un paquet BACk cap a l'MN, que al seu torn escriu un nou registre a la seva BUL vinculant la CoA, l'HoA i l'adreça de l'HA amb aquest lifetime (a efectes pràctics, l'MN considera l'HA com si fos un CN).

Una vegada acabat aquest procediment, s'aplica el mecanisme de DHAAD i es crea un túnel IPv6-IPv6 entre l'MN i l'HA (que serà una nova interfície virtual per a tots dos), tal com s'especifica a [9] i com mostra la figura 1.2:



**Fig. 1.2.** Esquema de comunicació MIPv6

Quan un terminal vol comunicar-se amb l'MN (mitjançant la seva HoA), contacta amb l'HA. L'HA agafa el paquet amb destinació l'HoA de l'MN i l'encapsula dins d'un altre paquet IPv6 amb adreça origen l'HA i destí la CoA de l'MN i l'envia per la nova interfície del túnel IPv6-IPv6.

Quan el paquet arriba a l'MN, el desencapsula i n'obté el paquet original. D'aquesta manera, el protocol es converteix en transparent per a les capes superiors. Se segueix el mateix procés a la inversa per al cas dels paquets enviats per l'MN cap a un altre terminal.

Els paquets del protocol (BU, BAcK i BRR) no s'envien per aquesta nova interfície túnel.

Mentre l'MN es troba fora del Home Link, l'HA és l'encarregat de prendre el rol de l'MN per a evitar que qualsevol altre node pugui utilitzar l'HoA.

### 1.3.6. Altres funcionalitats

El protocol MIPv6 permet altres funcionalitats que estan fora de l'abast d'aquest projecte, però que s'inclouen per a informar el lector:

- **Route Optimization:** es tracta d'optimitzar l'enviament de paquets entre un CN i l'MN per tal de no haver de passar per l'HA. Per tal de fer transparent aquest procés a les capes superiors, s'utilitza la capçalera IPv6 *destination option header* per als paquets enviats de l'MN al CN. Per als missatges enviats del CN a l'MN s'utilitza l'extensió de capçalera

d'IPv6: *routing header type 2*. Aquest mecanisme s'explica amb detall a [2] i [11].

- **IPSec a MIPv6:** es defineix un mecanisme d'autenticació dels missatges Binding Update anomenat *return routability*. Aquest procés genera claus d'autenticació que només l'MN pot generar. Se'n detalla una explicació més extensa a [8] i [11].

## 1.2. IP Paging

IP Paging introdueix un nou estat al MN anomenat <sup>1</sup>dormant. En aquest estat el MN redueix l'intercanvi de missatges amb el seu HA sense que això suposi perdre la connectivitat amb el MN.

Per a determinar la posició en què es troba un MN dormant s'apliquen algoritmes d'IP Paging basats en distància (vegeu [6] i [7]) sobre una estructura jeràrquica de cel·les.

Es defineix un nou paràmetre numèric anomenat distància de threshold ( $D_t$ ) que determina quan un canvi entre cel·les (traspàs) es considera un canvi d'àrea de paging. Són aquests canvis els únics que motivaran un intercanvi de missatges entre el HA i el MN

Finalment, notar que en aquest projecte s'assumeix que cada cel·la està servida per un sol AR i que, per tant, cada AR correspon a una cel·la.

### 1.3.1. Estructura de les àrees de paging

Per a calcular la distància dels salts durant un traspàs, s'agrupen les diferents cel·les en grups de set, anomenats clústers. Un mecanisme assigna a cada clúster un identificador segons la seva posició i en repeteix l'esquema fins a definir diversos nivells de clústers.

Els identificadors de clúster (de 0 a 6) en defineixen el nivells, mentre que l'identificador 7 indica que un clúster no té subdivisions.

Els AR que formen les cel·les utilitzen aquests identificadors per a formar la seva adreça de xarxa. Amb els darrers 6 bits de l'identificador de xarxa dels routers es pot identificar el clúster i la cel·la en què es troba.

---

<sup>1</sup> Al llarg de tot el projecte s'utilitzarà l'anglicisme *dormant* per a descriure aquest estat, ja que és l'utilitzat en tots els documents que tracten sobre l'estalvi d'energia en dispositius mòbils.

Per exemple, si es pren l'adreça de l'AR següent:

*3ffe:100:0:6:f100:0:0:13*

*12 = 0000 0000 0001 0011 (binari)*

*Els darrers 6 bits seran:*

*010 = 2 (decimal) que indica el clúster*

*011 = 3 (decimal) que indica la cel·la*

*Per tant, l'adreça d'aquest AR indica que es troba a la xarxa 3ffe:100:0:6, al clúster 2 i a la cel·la 3.*

La figura 1.3 mostra com es realitza aquest adreçament:

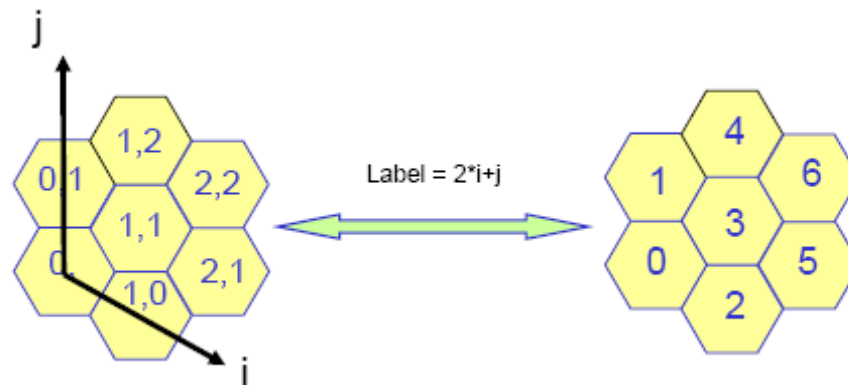


**Fig. 1.3.** Adreçament d'IP Paging

L'assignació d'aquests identificadors es detalla a [6], [7] i [11].

### I.3.2. Algoritme de paging basat en distància

El càlcul de la distància de Paging entre cel·les presenta la problemàtica que els eixos de coordenades  $i$ - $j$  (vegeu la figura 1.4) que representen l'assignació dels identificadors realitzen girs progressius a mesura que es puja de nivell de clúster.



**Fig. 1.4.** Assignació d'identificadors  $i$ - $j$  de les cel·les

Mitjançant un càlculs detallats a [6], [7] i [11], es pot trobar la solució matemàtica a aquest problema i calcular a cada moment quina distància hi ha entre dues cel·les.

### I.3.3. Aplicació d'IP Paging a Mobile IP versió 6

L'MN és capaç de diferenciar a quina cel·la i a quina subxarxa es troba a partir de l'adreça del seu AR.

Quan el terminal es troba en estat actiu, sempre que l'MN canvia de cel·la avisa l'HA mitjançant un Binding Update.

Quan un terminal es troba en estat dormant, realitza un traspàs i canvia de cel·la, i es calcula la distància de paging entre la cel·la nova i l'antiga. El valor calculat es compara amb el valor de  $Dt$ .

- En cas que la distància calculada sigui inferior o igual a  $Dt$ , l'MN no actualitza la seva CoA i no avisa l'HA del canvi (aquest procés s'explica en detall als Capítols 2 i 3).
- En cas que la distància calculada sigui superior a  $Dt$ , l'MN actualitza la seva CoA i avisa l'HA del canvi d'àrea de Paging mitjançant un missatge

Binding Update. Per a aquest escenari, l'HA considera que l'MN segueix en l'estat dormant, ja que no rep ni envia trànsit.

Quan l'HA vol contactar amb el seu MN, envia un paquet BRR cap a aquest últim en funció de la distància de  $Dt$  que s'ha determinat. D'aquesta manera, s'obliga a l'MN a passar a l'estat actiu.

### 1.3. Mecanisme d'estalvi d'energia per a xarxes IEEE802.11

La resposta que ofereix en principi l'estàndard 802.11 al problema de l'estalvi d'energia és el mecanisme PSM (*Power Save Mode*), que redueix el temps d'escolta en el canal de l'estació. No obstant això, tot i que aconsegueix reduir el consum d'energia, aquest mecanisme no respon a les necessitats de QoS d'aplicacions tipus VoIP. Per a respondre a aquesta necessitat, actualment es disposa de diverses solucions basades en el mecanisme d'estalvi APSD (*Automatic Power Save Delivery*) que redueixen el temps que la targeta ha d'estar activada (tal com s'exposa més endavant en aquest capítol).

Per tal de complir els requisits d'aquest estudi, s'ha utilitzat el mecanisme d'estalvi d'energia U-APSD (*Unscheduled Automatic Power Save Delivery*), introduït a l'estàndard 802.11e, que ha estat adoptat per la Wi-Fi Alliance (vegeu [15]), i que s'introdueix amb la denominació WMM-PS (*WiFi MultiMedia Power Save*).

L'estàndard 802.11 reconeix diversos mecanismes d'estalvi d'energia segons la topologia, ja sigui ad-hoc o d'infraestructura. Els requisits per a l'escenari d'aquest estudi fan que s'hagi treballat en mode infraestructura. A continuació es descriuen breument ambdues topologies.

En funció de la topologia es distingeixen els següents models de xarxes:

- Ad hoc: formades per un grup d'estacions que formen una xarxa local i aïllada, connectades entre si en una mateixa àrea de cobertura.
- Infraestructura: són xarxes gestionades des d'un punt central, en el cas d'aquest estudi l'Access Point, a través del qual passa tota la informació de la xarxa. Com que es tracta d'un escenari 802.11, s'utilitza el mecanisme d'accés al mitjà CSMA/CA (*Carrier Sense Multiple Access with Collision Avoidance*).

#### 1.3.1. Mecanisme d'estalvi d'energia: PSM

El mecanisme d'estalvi d'energia PSM neix amb la intenció de reduir el consum d'energia de l'estació de treball (d'aquí en endavant, citada com a STA) per tal de reduir el temps que aquesta roman activa, ja que el mecanisme d'accés al

mitjà CSMA/CA requereix que les estacions estiguin escoltant contínuament per a saber quan els correspon transmetre o rebre informació.

Els AP que treballen en una topologia de mode infraestructura guarden una relació de l'estat en què es troben les estacions a les quals donen servei, de manera que si reben paquets per a una estació amb PSM habilitat que es trobi "adormida", els emmagatzemen en un buffer per a transmetre'ls a aquesta estació quan estigui "desperta".

L'AP utilitza els beacon que envia regularment per a indicar mitjançant el TIM (*Traffic Indication Map*) si té paquets acumulats per a alguna de les estacions que gestiona. Cada estació compta amb un identificador mitjançant el qual l'AP, en el TIM, avisa de l'estat del buffer. Aquest identificador s'anomena AID (*Association ID code*).

Una estació pot passar a estar "desperta" per tres motius:

- LI (*Listen Interval*): una estació en estat PSM desperta cada  $n$  beacons, on  $n$  és un nombre enter més gran que 1. Aquest període de temps s'anomena Listen Interval.
- DTIM (*Delivery Traffic Indication Map*),: independentment del seu LI, una estació pot tenir habilitat el camp *ReceiveDTIM*. L'habilitació d'aquest camp permet a l'estació despertar cada DTIM (aquest temps s'ha indica als beacon enviats per l'AP) per a avisar de la transmissió de trànsit broadcast o multicast.
- En cas de que l'estació necessiti transmetre: cal recordar que el mode PSM no limita les estacions en cap aspecte quan és l'estació que vol enviar informació cap a l'AP.

Ja sigui en un LI o mitjançant un DTIM, quan una estació determina que té dades en cua a l'AP dirigides cap a ella, envia un missatge PS-Poll. L'AP respon a aquest missatge amb un ACK i amb l'enviament de trames acumulades. Aquestes trames es marquen mitjançant un camp anomenat *More Data* mentre queda informació per a transmetre a l'AP. L'estació envia un PS-Poll cada vegada que rep una trama, fins que en rep una amb el camp *More Data* deshabilitat.

La figura 1.5 mostra un exemple d'enviament de dades en PSM.



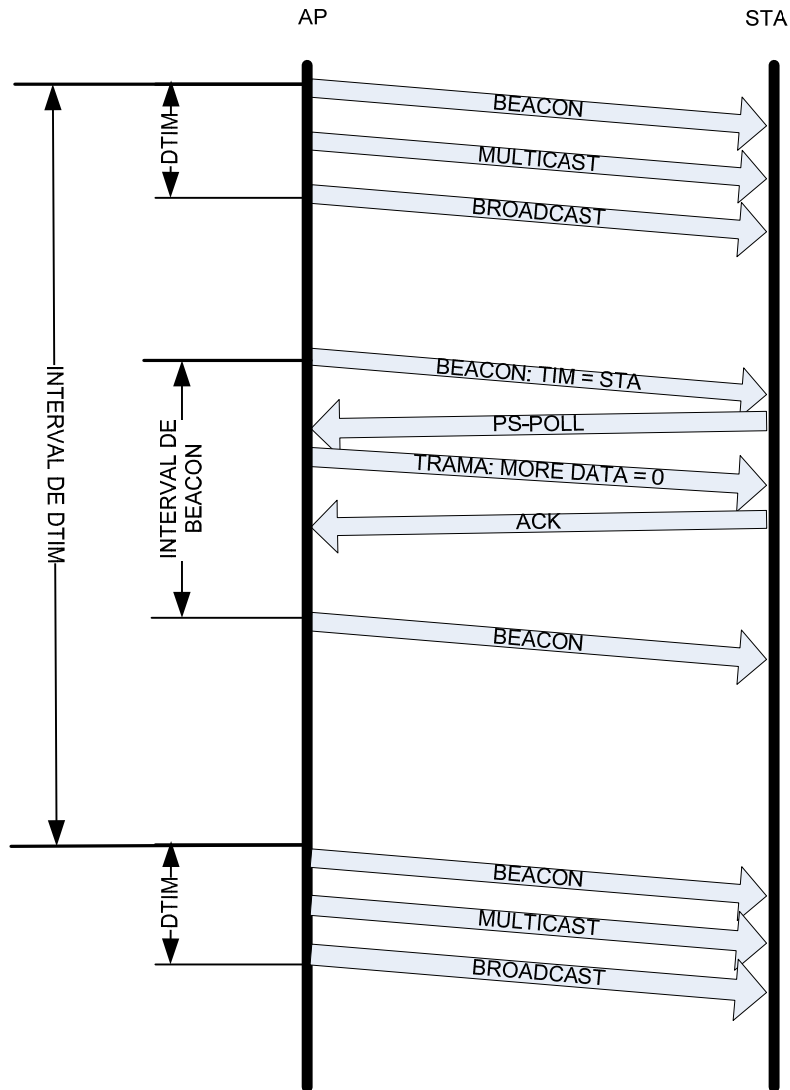


Fig. 1.5. Exemple d'operació PSM

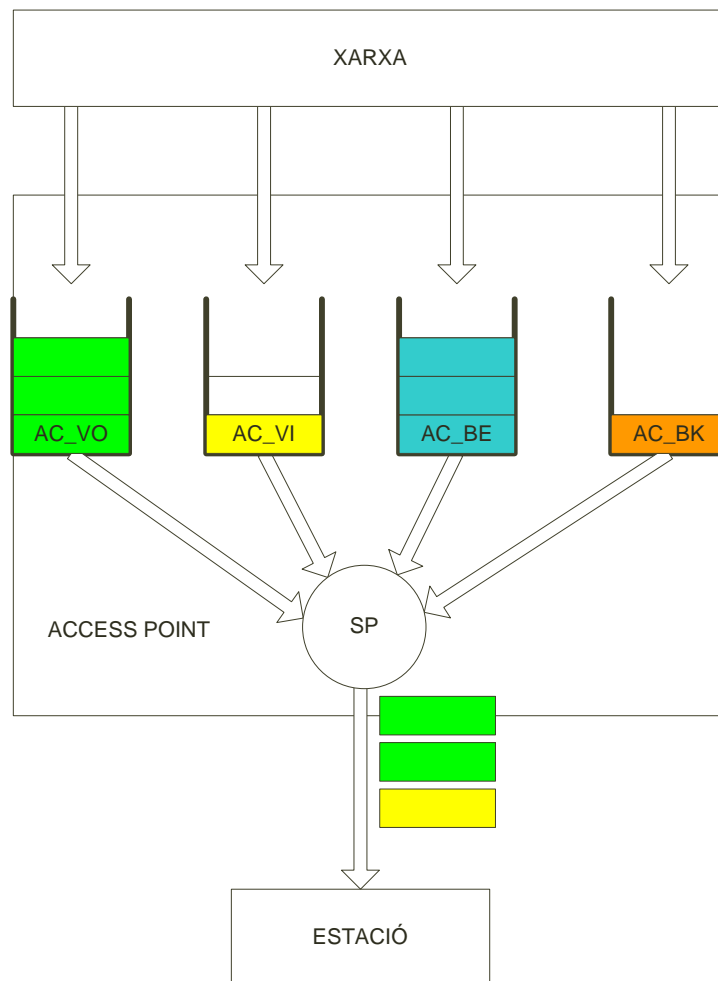
### I.3.2. Mecanisme d'estalvi d'energia: U-APSD

L'U-APSD és el mètode adoptat per la Wi-Fi Alliance, a l'estàndard 802.11e, per a proporcionar QoS.

En U-APSD, l'estació utilitza les trames que envia en sentit ascendent cap a l'AP, (com ara el trigger) que l'AP llegeix per a saber que l'estació està desperta i buidar així les cues dedicades a aquest terminal.

El mètode d'accés al mitjà associat a l'U-APSD és EDCA (*Enhanced Distributed Channel Access*), el qual s'utilitza durant els CP (*Contention Period*) (vegeu [17]). L'EDCA consta de quatre categories d'accés AC (*Access Class*),

que corresponen a veu, vídeo, best effort i background (vegeu la figura 1.6). Gràcies a aquestes cues es prioritza un trànsit o un altre per a descarregar les cues durant el període d'activitat de l'estació.



**Fig. 1.6.** Mecanisme EDCA

Cal destacar que per a mantenir la compatibilitat amb sistemes que no admeten U-APSD, l'AP no informa en els TIM de l'estat de les cues, tret que tots els AC de l'estació es trobin en l'estat *delivery-enabled*. Això es deu al fet que d'aquesta manera els beacon poden ser llegits per estacions amb o sense compatibilitat amb U-APSD i que, tal com s'ha dit anteriorment, són les estacions les encarregades d'informar l'AP sobre el seu estat mitjançant els *trigger*.

L'EDCA opera quan una o diverses de les seves cues es troben en estat *delivery-enabled* i passen per un *unscheduled SP*.

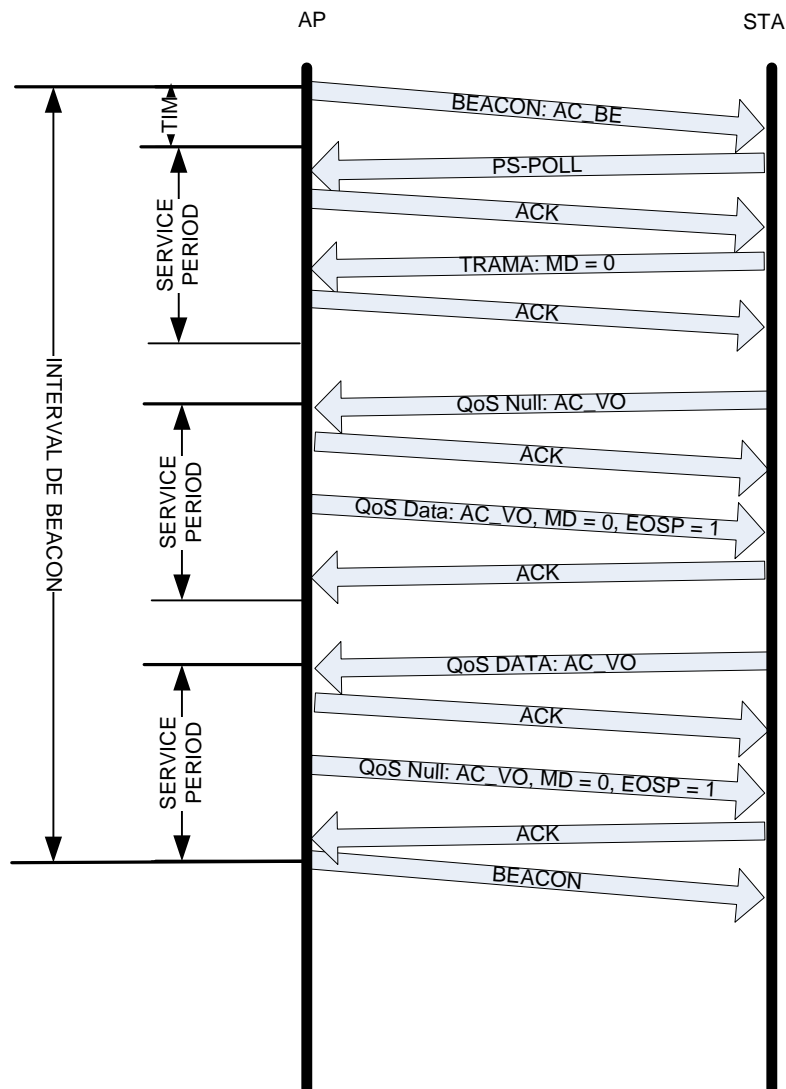
Un SP (*Service Period*) és un període de temps en el qual l'estació està "desperta", en cas que només operi amb mode no programat, *unscheduled*.

El període d'activitat SP s'inicia quan l'AP rep una trama trigger des de l'estació. N'existeixen dos tipus:

- QoS Data: contenen informació.
- QoS Null: substitueixen les trames PS-Poll de PSM, no contenen informació.

Tots dos tipus de trama es poden emprar per a activar un SP i per a indicar el final de l'SP mitjançant un EOSP (*End Of Service Period*).

Durant l'SP, l'AP envia cap a l'estació (des de les cues que es troben en estat *delivery-enabled*) un nombre de trames que no pot superar el definit pel camp *Max\_SP\_Length* (contingut en el camp *QoS info Field*) durant el procés de negociació d'associació entre l'estació i l'AP. La figura 1.7 mostra un exemple d'activació d'un SP:



**Fig. 1.7.** Activació d'un SP en U-APSD

## 1.4. Combinació dels mecanismes d'estalvi d'energia PSM i U-APSD i IP Paging per a MIPv6

Una vegada detallat el funcionament d'ambdós mecanismes d'estalvi d'energia de IEEE802.11, cal tenir en compte que l'U-APSD ha estat dissenyat per a poder treballar en conjunt amb PSM.

Aquest treball parteix de la base que un escenari que treballi amb U-APSD també tindrà activat el PSM, de manera que el trànsit que no correspongui a un AC que no estigui actiu com a *delivery-enabled* se sotmeti per defecte al mecanisme PSM.

El PSM provoca una certa quantitat d'overhead. Cal recordar que per a garantir la QoS en VoIP es recomana reduir el trànsit de pujada que, per exemple, generen els PS-Polls. Aquest fet, però, queda compensat per l'U-APSD, que tal com s'ha vist, aprofita les trames amb dades per a les seves gestions.

La reducció del trànsit entrant i sortint en el terminal durant els períodes SP activats per les trames QoS Data i QoS Null permet estalviar energia.

L'usuari pot configurar el Binding Lifetime durant el qual és vàlid l'enllaç entre l'HA i l'MN com a manera per a reduir l'intercanvi de missatges amb el HA i per tant estalviar bateria.

Amb l'ús d'IP Paging, després d'un cert temps d'inactivitat, l'MN pot passar automàticament a un estat dormant i únicament en surt si rep o envia trànsit o bé per a actualitzar el binding amb l'HA, per esgotament del seu temps de vida o per canvia de l'àrea de paging. Això suposa una reducció del nombre de missatges intercanviats respecte el funcionament estàndard de MIPv6 que indica que cal enviar un missatge de Binding Update (i la posterior recepció del BAck) des de l'MN cap a l'HA cada vegada que es realitza un canvi de xarxa. Aquesta reducció s'ha de traduir en un estalvi de bateria que pot incrementar-se si l'estat dormant es combina amb l'ús de PSM i/o U-APSD.

Però es pot anar més enllà en l'estalvi de bateria si durant l'estat dormant el MN no rep cap tipus de trama/paquet multicast o broadcast. Això en teoria suposaria la pèrdua de connectivitat del terminal ja que no podria completar el procés de Neighbor Discovery necessari per a rebre un paquet. Com s'explicarà en el següent capítol, amb l'ús d'IP Paging això no és així, i un MN en mode dormant utilitzant PSM, amb o sense U-APSD pot rebre un paquet sense necessitat de completar abans el procés de Neighbor Discovery.

Així, doncs, es poden sumar fins a tres estalvis d'energia: un primer estalvi derivat de l'ús de PSM i U-APSD, un segon derivat de tenir suport per al mode dormant amb IP Paging, i un tercer derivat de no rebre el trànsit multicast i broadcast quan s'està en estat dormant.



## CAPÍTOL 2. Modificacions a la implementació del dimoni UMIP

Un dels objectius d'aquest projecte és actualitzar la maqueta MIPv6 (vegeu [11]) amb una nova versió del dimoni per tal d'executar el protocol MIPv6, atès que la implementació anterior ja no disposa de suport.

En primer lloc, s'ha dut a terme una recerca de les implementacions existents del protocol MIPv6 en totes les plataformes per a poder fer una tria en base als requeriments del projecte. Se'n poden consultar els detalls a l'Annex I.

Tot seguit, s'ha actualitzat la maqueta amb la implementació triada a l'Annex I i s'ha creat un manual d'instal·lació per a usuaris. Aquest manual correspon a l'Annex III.

A continuació, s'ha portat el codi font creat a [11] a la nova plataforma, tot intentant conservar i validar cadascuna de les seves funcionalitats.

Finalment, s'han implementat al dimoni modificat amb el codi font de [11] una sèrie de millores per tal d'augmentar l'eficiència energètica d'MN. A l'Annex VII es descriuen amb més detall els canvis realitzats sobre el codi font del dimoni UMIP d'aquest projecte, i també els de [11].

### 2.1. Modificacions requerides

A continuació es descriuen les modificacions requerides per al dimoni UMIP, dividides entre les ja existents a [11] i les implementades per a aquest projecte.

Per tal d'importar i validar totes les funcionalitats del codi de [11] a la nova maqueta ha calgut seguir un procés que ha resultat complex i laboriós.

#### 2.1.1. Modificacions i implementacions existents

Per tal de facilitar la lectura del projecte, en aquest apartat es descriuen les modificacions ja realitzades en el treball anterior.

A [11] es va implementar el suport per a dues funcionalitats:

- **Node dormant:** aplicat a l'MN, es tracta d'un terminal que canvia a estat inactiu quan deixa d'enviar o de rebre trànsit durant un període de temps determinat. Consulteu [11] per a conèixer-ne més detalls.
- **IP Paging:** aquesta funcionalitat es descriu al Capítol 1.

##### 2.1.1.1. Modificacions i implementacions generals

Per tal de donar suport a les funcionalitats requerides, es disposa d'una sèrie d'implementacions que són comunes per a tots dos nodes:

- **Suport per al flag dormant als paquets BU:** cal redefinir el paquet BU a tots dos nodes per tal d'obtenir suport per al flag dormant. Cal afegir-hi la definició i indicar on es pot trobar aquest flag.
- **Càlcul d'àrees de paging:** es defineixen funcions per a fer el càlcul de la distància de paging segons la teoria del Capítol 1.
- **Paràmetre comú de distància de paging:** es defineix un nou fitxer (*paging.conf*) que conté un paràmetre (*Dt*) amb la distància de paging que es vol implementar. Aquest paràmetre ha de ser comú per a tots els nodes.

A la taula següent es descriuen aquests canvis sobre el codi font:

**Taula 2.1.** Implementacions ja realitzades

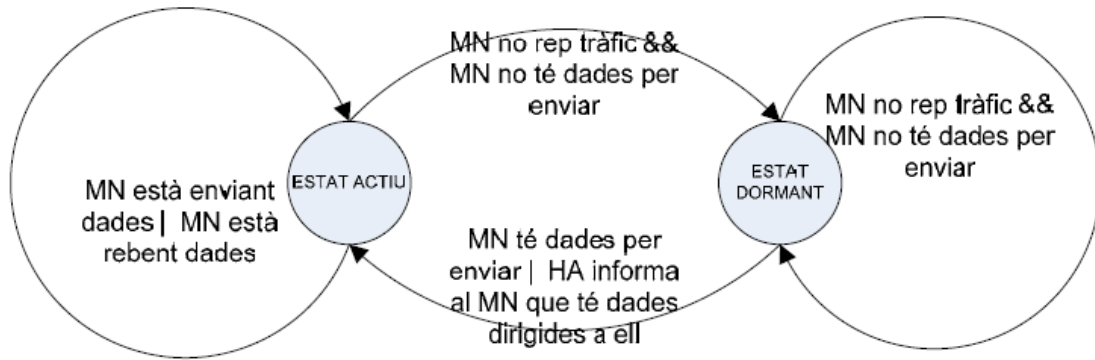
Fitxers modificats	Modificació
<i>/include/ip6mh.h</i>	Definició del flag dormant
<i>/src/proc_sys</i>	Definició de la funció <i>get_distance</i> per a recollir el paràmetre <i>Dt</i> definit.
<i>/src/movement.c</i>	Definició de les funcions <i>id_cell</i> , <i>get_coord</i> i <i>calculate_distance</i> per a realitzar els càlculs de la distància de paging.
<i>/usr/local/etc/paging.conf</i>	Variable <i>DistancePagingArea</i> "number", on <i>number</i> és la distància de paging definida.

Els detalls d'aquestes implementacions es poden consultar al Capítol 3 de [11].

### 2.1.1.2. Modificacions a l'MN

A continuació s'inclou una llista detallada amb els canvis realitzats a l'MN per a donar suport a les dues funcionalitats descrites a l'apartat 2.1.1:

- **Suport a l'MN del estats actiu i dormant:** el terminal mòbil ha de ser capaç de passar a aquests dos estats segons la figura 2.1. L'MN informarà l'HA dels diferents estats en què es troba.



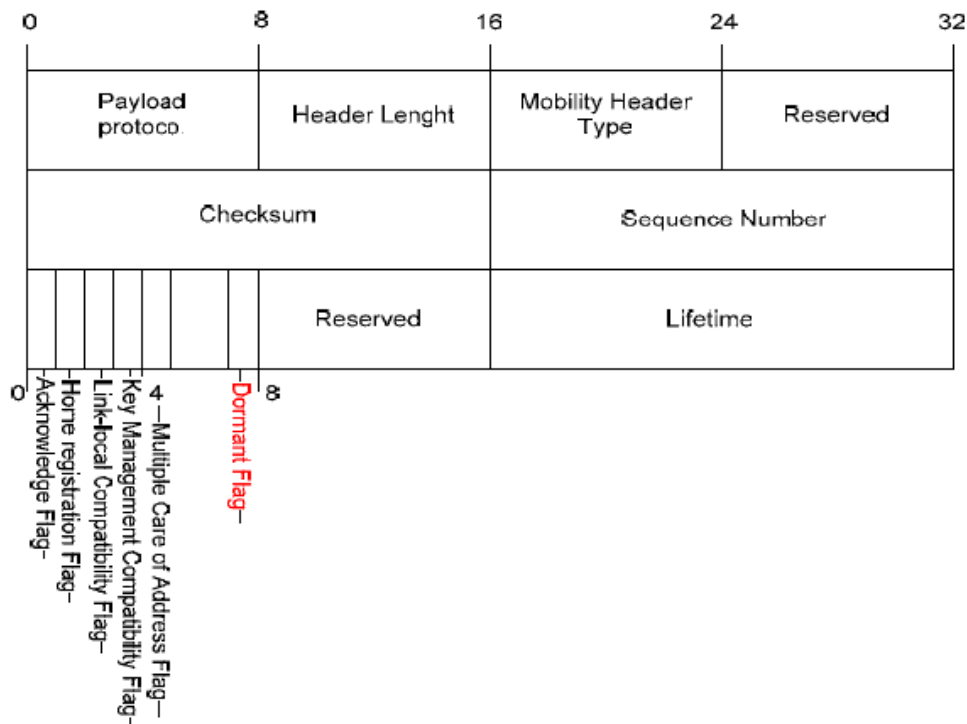
**Fig. 2.1.** Diagrama d'estat de l'MN

El node actiu passa a dormant quan al cap de tot un període d'activitat (un BU Lifetime) el terminal mòbil no rep ni envia trànsit.

En l'estat dormant, l'interval de temps de BU augmenta al màxim permès per l'HA o al temps definit pel paràmetre *HaMaxBindingLife* (vegeu l'Annex III).

El node dormant passa a actiu si rep o envia trànsit o si rep un BRR de l'HA.

Per tal de fer saber a l'HA que l'MN ha passat a estat dormant, s'introdueix un nou flag dormant al camp reservat del paquet BU. Quan l'HA rep un missatge BU amb aquest flag posat a 1 (per defecte és 0), l'HA afegeix l'MN que ha enviat el BU a una llista de nodes dormants.



**Fig. 2.2.** Missatge Binding Update amb flag dormant



- **L'MN només informa l'HA per canvi d'àrea de paging:** quan el terminal mòbil canvia de xarxa, decideix si avisa o no l'HA d'aquest canvi (per defecte, l'MN sempre avisa l'HA d'un canvi de xarxa) segons la distància de paging que tinguin assignada l'HA i l'MN. El càlcul de la distància de paging es pot consultar al Capítol 1 d'aquest treball, i a [6], [7] i [11].

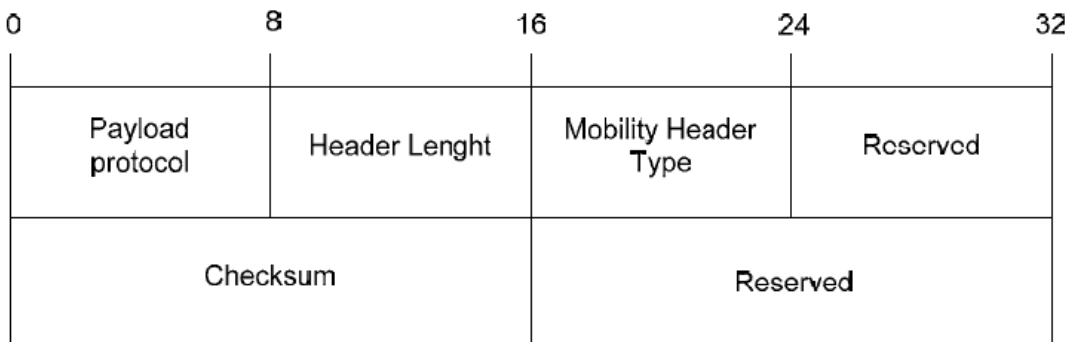
Quan l'MN canvia de xarxa, calcula la seva distància de paging prenent com a paràmetres la nova adreça de xarxa que ha creat (la CoA) i l'antiga CoA que tenia assignada:

- Si la distància calculada és menor que la distància de paging definida, l'MN no informarà l'HA.
- Al contrari, si supera la distància de paging definida, l'MN envia un missatge BU cap a l'HA per a actualitzar la seva posició.

En cas que l'MN vulgui enviar dades cap a l'HA, ha d'actualitzar la seva posició amb un BU cap a l'HA.

En cas que el CN vulgui comunicar-se amb l'MN, l'HA envia un paquet Binding Refresh Request cap a totes les possibles adreces de l'MN (segons la distància de paging definida i la darrera CoA que tingui l'HA).

Una vegada l'MN ha rebut el BRR envia automàticament un BU cap a l'HA per a actualitzar la seva posició, i d'aquesta manera la comunicació entre el CN i l'MN es desenvolupa correctament.



**Fig. 2.3.** Missatge Binding Refresh Request

### 2.1.1.3. Implementacions a l'MN

A la taula següent es descriuen els canvis realitzats sobre el codi font de l'MN per a donar suport a les modificacions de l'apartat anterior:

**Taula 2.2.** Implementacions ja realitzades a l'MN

<b>Fitxers modificats</b>	<b>Modificació</b>
<i>/src/proc_sys.h</i>	Definició de la ruta estadístiques.
<i>/src/proc_sys.c</i>	Definició de <i>go_to_dormant</i> per a indicar quan l'MN ha de canviar a dormant.
<i>/src/movement.h</i>	Variable que guarda l'adreça de l'AR de la nova cel·la visitada.
<i>/src/mn.c</i>	Variables d'estat de l'MN.
	Variables per a guardar l'antiga adreça de la cel·la visitada.
	Modificació de la funció <i>mn_send_bu_msg</i> per a enviar BU amb flag dormant.
	Modificació de la funció <i>do_handoff</i> per a detectar canvis d'àrea de paging.
	Funció <i>wake_up</i> per a forçar el canvi a l'estat actiu quan es detecta trànsit sortint des de l'MN.
	Modificació de la funció <i>mn_rcv_brr</i> per a forçar el canvi a l'estat actiu quan es rep un BRR.

Els detalls d'aquestes implementacions es poden consultar al Capítol 3 de [11].

#### 2.1.1.4. Modificacions a l'HA

Pel que fa a l'HA, cal aplicar-hi les modificacions següents per a donar suport a les funcionalitats:

- **Flag dormant:** l'HA interpreta correctament el missatge BU amb flag dormant i actualitza o crea una llista d'MN dormants.
- **Canvi d'àrea de paging:** l'HA avisa correctament un MN dormant que s'ha mogut de xarxa però no hi ha àrea de paging.
- **Retransmissió del primer paquet:** l'HA retransmet el primer paquet rebut cap a l'MN per a evitar pèrdues importants (com el SYN d'una connexió TCP).

#### 2.1.1.5. Implementacions a l'HA

A la taula següent es descriuen els canvis realitzats sobre el codi font de l'HA per a donar suport a les modificacions de l'apartat anterior:

**Taula 2.3.** Implementacions ja realitzades a l'HA

Fitxers modificats	Modificació
/src/ha.c	Definició de la llista de nodes MN dormants.
	Definició de les funcions <i>insert_mn</i> i <i>delete_mn</i> per a gestionar la llista de nodes dormants.
	Modificació de les funcions <i>ha_rcv_bu_main</i> i <i>ha_rcv_bu_worker</i> per a llegir correctament el flag dormant d'un BU.
	Definició de la funció <i>send_to_areapaging</i> per a enviar missatges BRR a tota la PA.
	Definició de les funcions <i>first_msg_to_mn_dormant</i> , <i>copy_buffer</i> i <i>send_msg</i> per a detectar i retransmetre el primer paquet cap a l'MN.

Els detalls d'aquestes implementacions es poden consultar al Capítol 3 de [11].

### 2.1.2. Modificacions i implementacions creades

Els requeriments per a aquest projecte estan enfocats a fer que els mecanismes d'estalvi d'energia puguin ser més eficaços.

El dimoni UMIP permet mantenir una adreça vàlida durant un cert període de temps quan es troba en estat dormant. L'única condició per a mantenir aquest vincle és renovar periòdicament el binding entre l'HA i l'MN amb un BU.

Aquest mecanisme es pot aprofitar per a fer més eficaç l'estalvi energètic:

- **Activació de mecanismes d'estalvi d'energia:** utilitzar PSM i U-APSD en els períodes en què l'MN es troba en estat dormant. Els detalls de l'activació de PSM s'expliquen amb més detall a l'Annex V.
- **Reducció del trànsit a l'MN:** en relació amb el punt anterior, es pot ajudar aquests algorismes reduint al mínim el nombre de trames que rep i processa l'MN.

#### 2.1.2.1. Reducció de trànsit a l'MN

En estat actiu i tal com s'explica al Capítol 1, l'MN llegeix els RtrAdv enviats pel router al qual es troba associat (aquests missatges s'envien periòdicament) per tal de validar que es troba realment connectat a la xarxa.

Si l'MN deixa de rebre RtrAdv durant un cert temps (valor router Lifetime), dedueix que ha perdut el vincle amb el router i reinicia el procés de recerca d'un nou router.

Tal com es mostra a la figura 2.3, el router envia de manera periòdica els RtrAdv en multicast:

253	7.244677	::	ff02::2	ICMPv6	Router solicitation
254	8.012484	fe80::213:46ff:fe2f:da60	ff02::1	ICMPv6	Router advertisement
255	8.086148	3ffe:100::223:6cff:fe86:c027	3ffe:100::f100:0:0:1	MIPv6	Binding Update
256	8.804127	Cisco_ee:a9:58	Cisco_ee:a9:58	LOOP	Reply
257	9.120323	fe80::213:46ff:fe2f:da60	ff02::1:ff86:c027	ICMPv6	Neighbor solicitation
258	9.292584	3ffe:100:0:6:223:6cff:fe86:c027	fe80::213:46ff:fe2f:da60	ICMPv6	Neighbor advertisement
259	9.292602	3ffe:100::f100:0:0:1	3ffe:100::223:6cff:fe86:c027	MIPv6	Binding Acknowledgement
260	11.295774	3ffe:100::223:6cff:fe86:c027	3ffe:100::f100:0:0:1	ICMPv6	Mobile Prefix Solicitation
261	11.296696	3ffe:100::f100:0:0:1	3ffe:100::223:6cff:fe86:c027	ICMPv6	Mobile Prefix Advertisement
262	13.086696	fe80::223:6cff:fe86:c027	fe80::213:46ff:fe2f:da60	ICMPv6	Neighbor solicitation
263	13.086715	fe80::213:46ff:fe2f:da60	fe80::223:6cff:fe86:c027	ICMPv6	Neighbor advertisement
264	13.684813	fe80::213:46ff:fe2f:da60	ff02::1	ICMPv6	Router advertisement

**Fig. 2.3.** Enviament de Router Advertisements per part del router

També és interessant reduir al màxim el nombre de Neighbor Solicitation/Advertisement que el mecanisme de Neighbor Discovery implementa i que rep o envia l'MN.

### 2.1.2.2. Supressió de Router Advertisements

Per tal d'intentar eliminar aquest trànsit periòdic que augmenta el consum i impedeix que els mecanismes de PSM i U-APSD actuïn de manera més efectiva, s'ha modificat el codi font del dimoni UMIP per a treballar sense RtrAdv quan es trobi en estat dormant.

Durant l'estat actiu, s'emmagatzema sempre el darrer paquet RtrAdv enviat per el router de la xarxa. Cal recordar que sempre que s'activa el dimoni, aquest es troba en estat actiu.

Una vegada entra en estat dormant, l'MN desactiva la lectura i el processament de Router Advertisements i passa a llegir el camp SSID (*Service Set Identifier*) de l'AP Cisco 1130 al qual es troba connectat per a deduir a quina xarxa es troba. Els detalls del format d'aquest camp es defineixen a l'apartat següent.

La lectura d'aquest camp queda definida pel valor de Router Lifetime que envia els RtrAdv (vegeu la figura 2.4). Per simplicitat, s'ha mantingut aquest valor a 30 segons (el valor per defecte).

```

# Frame 254 (110 bytes on wire, 110 bytes captured)
# Ethernet II, Src: D-Link_2f:da:60 (00:13:46:2f:da:60),
# Internet Protocol Version 6
# Internet Control Message Protocol v6
  Type: 134 (Router advertisement)
  Code: 0
  Checksum: 0xac09 [correct]
  Cur hop limit: 64
  # Flags: 0x00
  Router lifetime: 30
  Reachable time: 0
  Retrans timer: 0
# ICMPv6 Option (Prefix information)
  Type: Prefix information (3)
  Length: 32
  Prefix length: 64
  # Flags: 0xe0
  Valid lifetime: 2592000
  Preferred lifetime: 604800
  Prefix: 3ffe:100:0:6::
# ICMPv6 Option (Source link-layer address)
  Type: Source link-layer address (1)
  Length: 8
  Link-layer address: 00:13:46:2f:da:60

```

**Fig. 2.4.** Camp Router Lifetime

Una vegada traduït el camp SSID a una adreça de xarxa, es reescriu el paquet RtrAdv emmagatzemat durant l'estat actiu. L'antiga adreça d'origen del router es substitueix per la nova adreça de xarxa *Local-Link* llegida mitjançant SSID i aquest nou paquet s'autoenvia aquest nou paquet a si mateix.

D'aquesta manera, el dimoni creu que ha rebut un nou RtrAdv i permet no perdre la connectivitat.

Una vegada l'MN passa a estat actiu, disposa d'una adreça *Local-Link* vàlida a la qual pot enviar un Neighbor Discovery per a conèixer la MAC de l'AR. Una vegada rep el paquet Neighbor Advertisement, ja pot enviar el paquet Binding Update cap a l'HA.

### 2.1.2.3. Definició i lectura del nou SSID

El camp SSID ha de tenir un format específic per tal que l'MN pugui deduir en quina xarxa es troba connectat i aplicar-hi (si convé) el procés de canvi d'àrea de paging.

El format és el següent:

```
XXXXF10000000000YYYY
```

*On XX i YY són els valors que poden variar segons la distribució d'adreces que s'han donat a la maqueta (vegeu l'Annex III).*

Així, si l'adreça del router és `3ffe:100:0:6:f100:0:0:12`, el camp SSID anunciat per l'AP connectat a aquest router és:

```
0006f100000000000012
```

Aquest format pren com a premissa que els tres primers dígit de l'adreça (3ffe i 100 i 0) sempre són fixos.

Atès que l'enllaç entre l'MN i el router no és global (*Global-Link*), l'adreça d'origen dels paquets RtrAdv és local (*Local-Link*) i, per tant, cal llegir dues adreces amb l'SSID:

- **Global-Link:** aquesta adreça s'utilitza per a deduir la distància de paging (`3ffe:100:0:X:f100:0:0:Y`).
- **Local-Link:** aquesta adreça s'utilitza per a deduir l'adreça local que empra l'MN per a enviar els paquets a l'AR una vegada surt de l'estat dormant (`fe80:0:0:X:f100:0:0:Y`).

L'adreça *Local-Link* s'ha d'introduir prèviament a la interfície corresponent de l'AR just després d'executar el dimoni Radvd, tal com s'explica a l'Annex III.

Cal definir aquesta nova adreça local perquè durant l'estat dormant es necessita deduir una adreça *Local-Link* vàlida per a enviar directament el paquet Binding Update quan l'MN surti de dormant.

Una vegada configurada, les interfícies dels AR són les següents:

```
[root@AR1 ~]# ifconfig
eth0  Link encap:Ethernet HWaddr 00:13:46:2F:E2:D9
      inet addr:147.83.118.169 Bcast:147.83.255.255 Mask:255.255.0.0
      inet6 addr: fe80::213:46ff:fe2f:e2d9/64 Scope:Link
      inet6 addr: fe80::5:f100:0:0:11/64 Scope:Link
      inet6 addr: 3ffe:100:0:5:f100::11/64 Scope:Global
      UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
      RX packets:910 errors:0 dropped:0 overruns:0 frame:0
      TX packets:989 errors:0 dropped:0 overruns:0 carrier:0
      collisions:0 txqueuelen:1000
      RX bytes:79806 (77.9 KiB) TX bytes:88597 (86.5 KiB)
      Interrupt:58 Base address:0x6c00

eth1  Link encap:Ethernet HWaddr 00:15:F2:49:D2:E0
      inet addr:147.83.118.229 Bcast:147.83.118.255 Mask:255.255.255.0
      inet6 addr: fe80::215:f2ff:fe49:d2e0/64 Scope:Link
      inet6 addr: 3ffe:100:0:2:f100::5/64 Scope:Global
      UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
      RX packets:586 errors:0 dropped:0 overruns:0 frame:0
      TX packets:155 errors:0 dropped:0 overruns:0 carrier:0
      collisions:0 txqueuelen:1000
      RX bytes:65348 (63.8 KiB) TX bytes:14934 (14.5 KiB)
      Base address:0xa000 Memory:cffc0000-cffe0000
```

**Fig. 2.5.** Interfície AR amb nova *Local-Link*

### 2.1.2.4. Supressió de Neighbor Discovery

El protocol Neighbor Discovery a IPv6 substitueix les taules ARP d'IPv4 per a emmagatzemar la llista d'adreces MAC dels terminals connectats a un dispositiu i poder fer l'enllaç al nivell 2. Es tracta d'un protocol que utilitza paquets ICMPv6.

Cada vegada que un terminal vol saber l'adreça MAC d'un altre terminal, envia una Neighbor Solicitation en Multicast i els altres terminals responen amb un Neighbor Advertisement en Unicast dirigit cap al primer. D'aquesta manera, cada terminal pot conèixer la MAC dels seus veïns.

Aquest procés es mostra a la figura 2.6:

5	11.612500	fe80::5:f100:0:0:11	ff02::1:ff86:c027	ICMPv6	Neighbor solicitation
6	11.651227	3ffe:100:0:5:223:6cff:fe86:c027	fe80::5:f100:0:0:11	ICMPv6	Neighbor advertisement
7	11.651243	3ffe:100::f100:0:0:1	3ffe:100:0:5:223:6cff	MIPv6	Binding Refresh Request
8	11.652805	3ffe:100::223:6cff:fe86:c027	3ffe:100::f100:0:0:1	MIPv6	Binding Update
9	11.696233	3ffe:100::f100:0:0:1	3ffe:100::223:6cff:fe	MIPv6	Binding Acknowledgement

**Fig. 2.6.** Mecanisme Neighbor Discovery

La lectura del RtrAdv emmagatzemat quan l'MN es troba en estat dormant fa que no es formi cap nova CoA mentre no es canvia d'àrea de paging.

Atès que el prefix del RtrAdv amb què es forma seguirà sent el mateix que quan estava actiu, l'única diferència entre el paquet original i el modificat és l'adreça d'origen del router. A la figura 2.7 es mostra com el RtrAdv envia el seu prefix:

```

⊕ Frame 1 (110 bytes on wire, 110 bytes captured)
⊕ Ethernet II, Src: D-Link_2f:da:60 (00:13:46:2f:da:60)
⊕ Internet Protocol Version 6
⊖ Internet Control Message Protocol v6
  Type: 134 (Router advertisement)
  Code: 0
  Checksum: 0xac09 [correct]
  Cur hop limit: 64
⊕ Flags: 0x00
  Router lifetime: 30
  Reachable time: 0
  Retrans timer: 0
⊖ ICMPv6 option (Prefix information)
  Type: Prefix information (3)
  Length: 32
  Prefix length: 64
⊕ Flags: 0xe0
  valid lifetime: 2592000
  Preferred lifetime: 604800
  Prefix: 3ffe:100:0:6::
⊕ ICMPv6 option (Source link-layer address)

```

**Fig. 2.7.** Enviament del prefix Router Advertisement

Si la CoA de l'MN està fixada en tot moment mentre està en dormant, es pot introduir una entrada estàtica als AR amb la MAC del dispositiu per a evitar l'enviament del Neighbor Discovery (vegeu la figura 2.8).

```
[root@AR2 ~]# ip neigh add to 3ffe:100:0:0:223:6cff:fe86:c027 dev eth0 lladdr
00:23:6c:86:c0:27 nud permanent
[root@AR2 ~]# ip neigh show 3ffe:100:0:3:f100::8 dev eth2 lladdr 00:04:75:c7:ae:6e router
STALE
3ffe:100::223:6cff:fe86:c027 dev eth0 lladdr 00:23:6c:86:c0:27 PERMANENT
3ffe:100::21e:e5ff:fedf:9dd7 dev eth0 lladdr 00:1e:e5:df:9d:d7 PERMANENT
```

**Fig. 2.8.** Assignació de MAC permanent

D'aquesta manera, quan l'AR disposi del paquet cap a l'MN, no haurà d'activar el protocol de Neighbor Discovery, sinó que enviarà directament el paquet cap a la CoA de l'MN, cosa que estalvia temps i bateria al dispositiu mòbil. La figura 2.9 mostra el nou procés:

1	0.000000	fe80::213:46ff:fe2f:da60	ff02::1	ICMPv6	Router advertisement
2	4.117882	cisco_ee:a9:58	cisco_ee:a9:58	LOOP	Reply
3	4.805030	3ffe:100::f100:0:0:1	3ffe:100:0:6:223:6cff	MIPv6	Binding Refresh Request
4	4.941360	3ffe:100::223:6cff:fe86:c027	3ffe:100::f100:0:0:1	MIPv6	Binding Update
5	4.958301	3ffe:100::f100:0:0:1	3ffe:100::223:6cff:fe	MIPv6	Binding Acknowledgement

**Fig. 2.9.** Mecanisme Neighbor Discovery amb MAC permanent

Aquesta adreça MAC no és coneguda directament per l'AR, sinó que ha estat introduïda manualment per comprovar que és factible l'enviament d'un paquet a la CoA de l'MN sense utilitzar el mecanisme Neighbor Discovery.

Una futura modificació del dimoni podria implementar que l'HA enviés la MAC de l'MN per part de l'HA cap a l'AR en cas de paging. Una vegada rebuda la MAC de l'MN, l'AR la introduiria automàticament a la seva llista de neighbors.

### 2.1.2.5. L'MN no forma CoA

Un dels requisits del projecte és suprimir la formació de una nova CoA a l'MN quan aquest canviï de xarxa sense canviar d'àrea de paging i estalviar consum de bateria en aquest procés.

Aquesta funcionalitat invalida l'enviament dels paquets Binding Refresh Request cap a tots els MN possibles d'una àrea de paging implementat a [11]. Per tant, no és possible despertar un node quan canvia de xarxa però no canvia d'àrea de paging.

En un futur projecte, l'enviament del BRR serà realitzat per l'AR, ja que l'HA enviarà la MAC de l'MN com un paràmetre (tal com s'explica a l'Annex III,



l'adreça CoA es forma a partir de la MAC de l'MN). El nou paquet BRR des de l'AR tindrà com a IP de destí la HoA de l'MN.

## 2.2. Implementacions realitzades

En aquest apartat es detallen les implementacions per a donar suport a les modificacions requerides per a aquest projecte. A l'Annex VII es pot consultar el codi d'aquesta implementació.

Després d'implementar l'antic codi realitzat a [11] a la nova maqueta s'han creat i modificat algunes de les funcions ja existents i s'ha creat una nova funció, anomenada *ssid\_handler*, que permet llegir l'SSID de l'AP al qual es troba connectat l'MN i deduir l'adreça de l'AR, tal com s'explica en aquest capítol mateix.

Quan l'MN s'activa, per defecte es troba en estat actiu. Durant aquest període de temps i fins que no passa a l'estat dormant, les funcions *process\_addr* i *md\_rcv\_ra* del fitxer *movement.c* emmagatzemen el darrer paquet RtrAdv que arriba a l'MN.

Quan un node passa de l'estat actiu a dormant o viceversa, s'executen les funcions *md\_stop* i *md\_start* del fitxer *movement.c*.

La taula següent enumera els paràmetres que cal indicar per a la transició d'estats:

**Taula 2.4.** Funcions a estat actiu i dormant

Funció	D'actiu a dormant	De dormant a actiu
<i>md_start</i> (int)	<b>1</b>	<b>0</b>
<i>md_stop</i> (int)	<b>0</b>	<b>1</b>

Aquestes dues funcions aturen/inicien els serveis de lectura dels RtrAdv, tant pel que fa al dimoni com al kernel de Linux.

Una vegada en l'estat dormant, la funció *ssid\_handler* del fitxer *movement.c* llegeix l'adreça de l'AR cada temps *router Lifetime*.

L'MN segueix indefinidament en l'estat dormant si no envia/rep trànsit.

La taula 2.5 recull de manera resumida tots els canvis realitzats:

**Taula 2.5.** Resum de les implementacions realitzades al dimoni UMIP

O	Modificació	Propòsit
/src/movement.c	Modificació de les funcions <i>md_start</i> i <i>md_stop</i> per a activar i desactivar els RtrAdv i passar a executar la funció de lectura de l'SSID en dormant.	Transició correcta entre estats   Supressió RtrAdv
	Creació de la funció <i>ssid_handler</i> per a la lectura i procés de l'SSID en mode dormant.	Lectura SSID
	Modificació de la funció <i>process_addr</i> per a emmagatzemar el paquet RtrAdv quan el node es troba actiu.	Supressió RtrAdv   node no forma CoA quan passa a dormant
	Modificació de la funció <i>md_rcv_ra</i> per a llegir correctament l'adreça global de l'AR a partir de l'SSID i emmagatzemar el paquet RtrAdv.	Lectura SSID   Supressió RtrAdv
	Variables globals per a emmagatzemar el paquet RtrAdv quan el node es troba actiu.	Supressió RtrAdv   node no forma CoA quan passa a dormant
	Modificació de la funció <i>md_router_timeout_probe</i> per a llegir l'SSID cada temps <i>Router Lifetime</i> .	Lectura SSID
	Modificació de la funció <i>md_update_router_stats</i> per a llegir correctament l'adreça global de l'AR.	Funcionalitat afegida a l'hora d'importar l'antic codi
	Variable global per a no passar a actiu un node que ja es troba actiu.	Transició correcta entre estats
/src/mn.h	Variable global <i>rtr_addr_old</i> ; per a permetre a <i>movement.c</i> tenir accés a l'antiga adreça de la cel·la.	Ajuda al traspàs
/src/mn.c	Modificació de la funció <i>mn_send_bu_msg</i> per a activar/desactivar els estats.	Transició correcta entre estats



## CAPÍTOL 3. Proves de validació de MIPv6

### 3.1. Pla de proves

En el capítol anterior s'han vist les modificacions implementades sobre el codi font del dimoni UMIP per tal de d'admetre nodes dormants i eliminar el trànsit cap a l'MN.

Per tal de validar les modificacions realitzades, s'ha dut a terme un pla de proves sobre el demostrador MIPv6 (vegeu la figura 3.1).

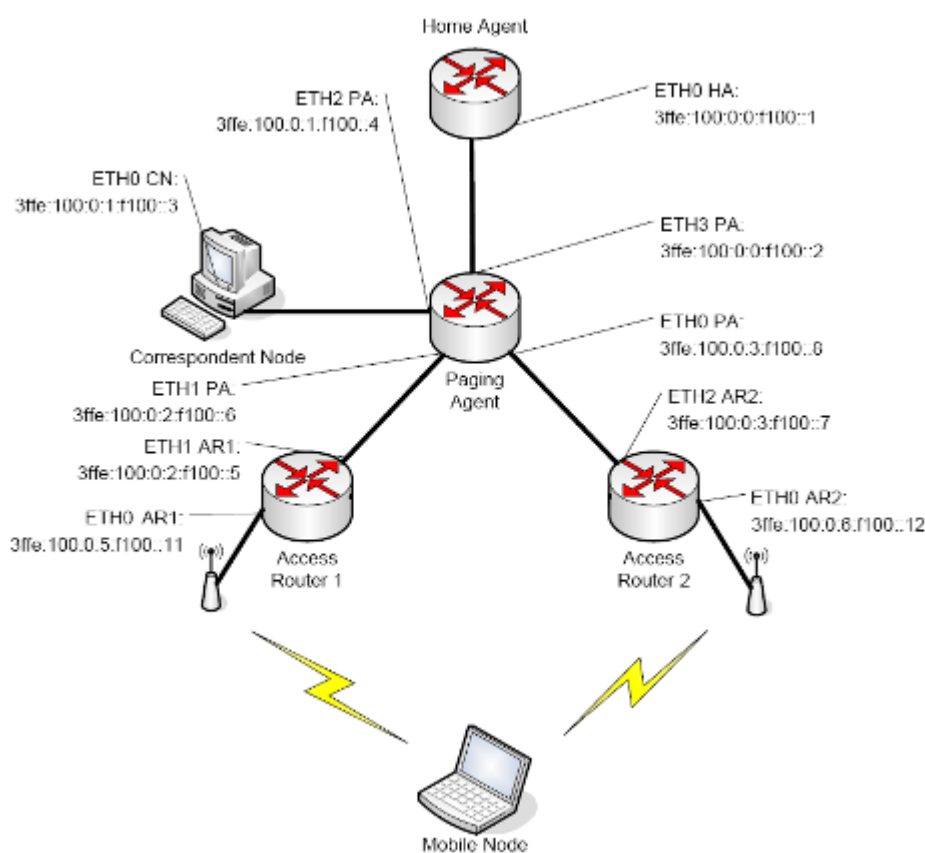


Fig. 3.1. Escenari de proves

Per a facilitar la lectura de les proves, a continuació es resumeix el comportament del MN i l'HA atenent a les modificacions descrites al capítol anterior. En l'estat actiu, l'MN es comporta de la mateixa manera que el dimoni UMIP original, mentre que en l'estat dormant apareixen els següents canvis:

- L'MN només envia els missatges Binding Update cap a l'HA cada temps *HaMaXBindingLife* (paràmetre definit a l'HA). Si aquest paràmetre no es

defineix, es pren per defecte el paràmetre *MnMaxHaBindingLife* de l'MN (vegeu l'Annex III).

- L'MN no processa els Router Advertisement que provenen del router. En lloc d'això, llegeix cada temps *Router Expired* l'SSID de l'AP a què es troba connectat i dedueix l'adreça de xarxa del router. Amb l'adreça llegida, pot emular la recepció d'un paquet RtrAdv i seguir en estat dormant.
- Quan l'MN fa la transició de l'estat dormant a actiu, torna a processar els RtrAdv i es comporta de la mateixa manera que el dimoni original.
- Quan l'MN canvia de xarxa però no d'àrea de paging, fa el traspàs de manera silenciosa i no informa l'HA del canvi. L'MN no passa a l'estat actiu ni canvia de CoA.
- Quan l'MN canvia de xarxa i d'àrea de paging, fa el traspàs però no canvia a l'estat actiu. Tot seguit envia un Binding Update cap a l'HA per a informar del canvi.
- En cas que un node vulgui enviar informació cap a l'MN, l'HA envia un missatge Binding Refresh Request cap a l'MN per a obligar-lo a passar a l'estat actiu. Aquesta implementació només és funcional en cas que l'MN no canviï de xarxa, ja que en estat dormant sempre es conserva la CoA i, per tant, no es pot enviar directament el paquet BRR cap a l'MN tal com s'explica al Capítol 2. En un futur caldrà implementar aquesta funcionalitat.
- En cas que l'MN vulgui enviar trànsit, passa automàticament a l'estat actiu, avisa l'HA i envia la informació.

Al llarg dels apartats següents es descriuen en primer lloc les proves realitzades i tot seguit es validen per tal de deixar constància del funcionament correcte del projecte.

### 3.1.1. Prova 1: l'MN canvia a l'estat dormant

En aquesta prova es vol comprovar que l'MN decideix correctament quan ha de passar a l'estat dormant.

Inicialment, l'MN es registra amb l'HA i passa a l'estat actiu. Tot seguit, es força una comunicació amb el CN (mitjançant l'eina ping6) de manera que es troben els escenaris següents:

- **P1.1 L'MN rep i transmet trànsit**
- **P1.2 L'MN rep trànsit**
- **P1.3 L'MN transmet trànsit**
- **P1.4 L'MN no rep ni transmet trànsit**

### 3.1.1.1. Resultats esperats

Per a cadascuna de les proves realitzades a l'apartat 3.1.1, s'espera que l'MN es comporti de la manera següent:

- **P1.1 L'MN rep i transmet trànsit:** l'MN sempre es troba en l'estat actiu i envia correctament els missatges BU cap a l'HA.
- **P1.2 L'MN rep trànsit:** l'MN sempre es troba en l'estat actiu i envia correctament els missatges BU cap a l'HA.
- **P1.3 L'MN transmet trànsit:** l'MN sempre es troba en l'estat actiu i envia correctament els missatges BU cap a l'HA.
- **P1.4 L'MN no rep ni transmet trànsit:** l'MN fa una transició a l'estat dormant i envia correctament el missatge BU cap a l'HA amb el flag dormant. L'HA inclou correctament el node a la llista de nodes dormants.

### 3.1.2. Prova 2: l'MN canvia a l'estat actiu

En aquesta prova es vol comprovar que l'MN decideix correctament quan ha de passar a l'estat actiu.

Inicialment, l'MN es registra amb l'HA i passa a l'estat actiu. Una vegada en l'estat actiu, no envia ni rep trànsit durant tot un període de Binding Lifetime i, per tant, passa a l'estat dormant. Tot seguit, es força una comunicació amb el CN (mitjançant l'eina ping6), de manera que es troben els escenaris següents:

- **P2.1 L'MN no té trànsit per a enviar ni per a rebre**
- **P2.2 L'MN no té trànsit per a enviar**
- **P2.3 L'MN no té trànsit per a rebre**

#### 3.1.2.1. Resultats esperats

Per a cadascuna de les proves realitzades a l'apartat 3.1.2, s'espera que l'MN es comporti de la manera següent:

- **P2.1 L'MN no té trànsit per a enviar ni per a rebre:** l'MN sempre es troba en l'estat dormant i envia correctament els missatges BU cap l'HA amb el flag dormant activat cada Binding Lifetime (valor que imposa l'HA).
- **P2.2 L'MN no té trànsit per a enviar:** l'MN sempre es troba en l'estat dormant i envia correctament els missatges BU cap l'HA amb el flag dormant activat cada Binding Lifetime (valor que imposa l'HA).
- **P2.3 L'MN no té trànsit per a rebre:** l'MN sempre es troba en l'estat dormant i envia correctament els missatges BU cap l'HA amb el flag dormant activat cada Binding Lifetime (valor que imposa l'HA).

### 3.1.3. Prova 3: l'MN avisa correctament l'HA per canvi d'àrea de paging

En aquesta prova es vol comprovar que l'MN decideix correctament quan ha d'avisar l'HA en cas que l'MN canviï de xarxa.

Inicialment, l'MN es registra amb l'HA i passa a l'estat actiu, i es força un canvi de xarxa.

- **P3.1 L'MN canvia de xarxa**

Una vegada superada la prova, l'MN no envia ni rep trànsit durant tot un període de Binding Lifetime i, per tant, passa a l'estat dormant. Tot seguit, es força un canvi de xarxa.

- **P3.2 L'MN canvia de xarxa dins la mateixa àrea de paging**
- **P3.3 L'MN canvia de xarxa i d'àrea de paging**

#### 3.1.3.1. Resultats esperats

Per a cadascuna de les proves realitzades a l'apartat 3.1.3, s'espera que l'MN es comporti de la manera següent:

- **P3.1 L'MN canvia de xarxa:** l'MN fa el canvi de xarxa i informa l'HA del canvi.
- **P3.2 L'MN canvia de xarxa dins la mateixa àrea de paging:** l'MN fa el canvi de xarxa però no passa a l'estat actiu ni canvia de CoA. Tampoc no informa l'HA del canvi.
- **P3.3 L'MN canvia de xarxa i d'àrea de paging:** l'MN fa el canvi de xarxa, canvia de CoA i informa l'HA del canvi.

### 3.1.4. Prova 4: comportament de l'HA per a localitzar l'MN

En aquesta prova es vol comprovar que l'HA localitza correctament l'MN quan aquest es troba en l'estat dormant.

Inicialment, l'MN es registra amb l'HA i passa a l'estat actiu. Una vegada en l'estat actiu, no envia ni rep trànsit durant tot un període de Binding Lifetime i, per tant, passa a l'estat dormant. Tot seguit, es força una comunicació amb el CN (mitjançant l'eina ping6), de manera que es troba l'escenari següent:

- **P4.1 El CN envia trànsit a l'MN que no ha canviat de xarxa**

### 3.1.4.1. Resultats esperats

Per a la prova realitzada a l'apartat 3.1.4, s'espera que l'HA i l'MN es comportin de la manera següent:

- **P4.1 El CN envia trànsit a l'MN que no ha canviat de xarxa:** l'HA envia un paquet BRR cap a l'MN. L'MN respon a aquest paquet enviant automàticament un Binding Update i passant a l'estat actiu.

### 3.1.5. Prova 5: retard

En aquesta prova es vol comprovar el retard l'eficiència de l'enviament de paquets ICMPv6 del CN cap a l'MN i més concretament l'impacte d'IP Paging sobre el mateix.

Per als dos escenaris següents s'ha realitzat una prova empírica dels valors d'RTT (*Round Trip Time*):

- **P5.1 El CN envia trànsit a l'MN en l'estat actiu**
- **P5.2 El CN envia trànsit a l'MN en l'estat dormant**

#### 3.1.5.1. Resultats esperats

Per a les proves realitzades a l'apartat 3.1.5, s'esperen els resultats següents:

- **P5.1 El CN envia trànsit a l'MN en l'estat actiu:** s'espera un RTT major en l'enviament del primer paquet ICMPv6 cap a l'MN. L'RTT de la resta de paquets ICMPv6 també és superior a la mitjana en cas de fer un ping6 directament a la CoA de l'MN.
- **P5.2 El CN envia trànsit a l'MN en l'estat dormant:** s'espera un RTT major que el de la prova P5.1 en l'enviament del primer paquet ICMPv6 cap a l'MN.

## 3.2. Resultats obtinguts

A continuació es detallen els resultats obtinguts per al pla de proves especificat a l'apartat 3.1. Els resultats es comparen amb els esperats.

En primer lloc, es mostren les figures que validen que l'MN i l'HA s'intercanvien correctament els corresponents missatges BU i BAcK:



La figura 3.2 mostra l'intercanvi entre l'HA i l'MN dels missatges de binding:

No. -	Time	Source	Destination	Protocol	Info
256	16.406944	3ffe:100::223:6cff:fe86:c027	3ffe:100::f100:0:0:1	MIPv6	Binding Update
259	17.614026	3ffe:100::f100:0:0:1	3ffe:100::223:6cff:fe86:c027	MIPv6	Binding Acknowledgement
275	73.407293	3ffe:100::223:6cff:fe86:c027	3ffe:100::f100:0:0:1	MIPv6	Binding Update
276	73.418428	3ffe:100::f100:0:0:1	3ffe:100::223:6cff:fe86:c027	MIPv6	Binding Acknowledgement
300	105.426448	3ffe:100::223:6cff:fe86:c027	3ffe:100::f100:0:0:1	MIPv6	Binding Update
304	105.443317	3ffe:100::f100:0:0:1	3ffe:100::223:6cff:fe86:c027	MIPv6	Binding Acknowledgement
567	114.348127	3ffe:100::223:6cff:fe86:c027	3ffe:100::f100:0:0:1	MIPv6	Binding Update
570	114.474684	3ffe:100::f100:0:0:1	3ffe:100::223:6cff:fe86:c027	MIPv6	Binding Acknowledgement

Frame 256 (110 bytes on wire, 110 bytes captured)  
 Ethernet II, Src: Apple\_86:c0:27 (00:23:6c:86:c0:27), Dst: D-Link\_2f:da:60 (00:13:46:2f:da:60)  
 Internet Protocol Version 6  
 Mobile IPv6 / Network Mobility  
 Payload protocol: IPv6 no next header (0x3b)  
 Header length: 3 (32 bytes)  
 Mobility Header Type: Binding update (5)  
 Reserved: 0x00  
 Checksum: 0x83cf  
 Binding Update  
 Mobility Options

**Fig. 3.2.** Intercanvi de missatges Binding Update i Binding Acknowledge

Una vegada ha transcorregut un període Binding Lifetime sense trànsit enviat o rebut o sense canvi d'àrea de paging, l'MN passa a l'estat dormant i llegeix l'SSID, tal com es mostra a la figura 3.3:

```

Sun Apr 19 20:31:02 md_update_router_stats: rtr_addr_new: 3ffe:100:0:5:f100:0:0:11
Sun Apr 19 20:31:02 mn_chk_ho_verdict: rtr_addr_old 3ffe:100:0:5:f100:0:0:11
Sun Apr 19 20:31:07 bu_refresh: Bul refresh type: 0
Sun Apr 19 20:31:07 mn_get_home_lifetime: CoA lifetime 2591995 s, HoA lifetime 2591887 s,
BU lifetime 60 s
Sun Apr 19 20:31:07 mn_send_bu_msg: valor del wakeup 0
new_val = 1560
last_val = 1560
Sun Apr 19 20:31:07 mn_send_bu_msg: Node Dormant
Sun Apr 19 20:31:07 mh_send: sending MH type 5
from 3ffe:100:0:0:223:6cff:fe86:c027
to 3ffe:100:0:0:f100:0:0:1
Sun Apr 19 20:31:07 mh_send: local CoA 3ffe:100:0:5:223:6cff:fe86:c027
Sun Apr 19 20:31:07 mn_send_bu_msg: Valor de i_dormant 1
Sun Apr 19 20:31:07 mn_send_bu_msg: Valor de tid 48
Sun Apr 19 20:31:07 mn_send_bu_msg: Passem a estat dormant
Sun Apr 19 20:31:07 md_stop: Parem serveis de node MN
Sun Apr 19 20:31:07 md_stop: Serveis de node MN aturats
SSID llegit fe80:000:0000:0005:f100:0000:0000:0011
Sun Apr 19 20:31:07 ssid_handler: llegint SSID des de 3ffe:100:0:5:f100:0:0:11 on iface 4
Sun Apr 19 20:31:07 ssid_handler: llegint SSID des de fe80:0:0:5:f100:0:0:11 on iface 4
Sun Apr 19 20:31:07 ssid_handler: rtr_addr_new: 3ffe:100:0:5:f100:0:0:11
Sun Apr 19 20:31:07 ssid_handler: SSID router: 3ffe:100:0:5:f100:0:0:11
Sun Apr 19 20:31:07 ssid_handler: Direccions iguals, node segueix en dormant
Sun Apr 19 20:31:07 md_add_backup_router: add new backup router fe80:0:0:5:f100:0:0:11 on
interface eth1 (4)
Sun Apr 19 20:31:07 mn_send_bu_msg: Node en dormant
Sun Apr 19 20:31:07 bu_refresh: Entro al else de bu_refresh
Sun Apr 19 20:31:07 bul_update_timer: Updating timer
== BUL_ENTRY ==
Home address 3ffe:100:0:0:223:6cff:fe86:c027
Care-of address 3ffe:100:0:5:223:6cff:fe86:c027
CN address 3ffe:100:0:0:f100:0:0:1
  
```

```
lifetime = 262140, delay = 1000
```

**Fig. 3.3.** Debug de l'MN en passar a l'estat dormant

La figura 3.4 mostra l'enviament del Binding Update amb el flag dormant activat:

No. -	Time	Source	Destination	Protocol	Info
256	16.406944	3ffe:100::223:6cff:fe86:c027	3ffe:100::f100:0:0:1	MIPv6	Binding Update
259	17.614026	3ffe:100::f100:0:0:1	3ffe:100::223:6cff:fe86:c027	MIPv6	Binding Acknowledgement
275	73.407293	3ffe:100::223:6cff:fe86:c027	3ffe:100::f100:0:0:1	MIPv6	Binding Update
276	73.418428	3ffe:100::f100:0:0:1	3ffe:100::223:6cff:fe86:c027	MIPv6	Binding Acknowledgement
300	105.426448	3ffe:100::223:6cff:fe86:c027	3ffe:100::f100:0:0:1	MIPv6	Binding Update

Checksum: 0x82dd

Binding Update

sequence number: 18370

1... .. = Acknowledge (A) flag: Binding Acknowledgement requested

.1.. .... = Home Registration (H) flag: Home Registration

..1. .... = Link-Local Compatibility (L) flag: Link-Local Address Compatibility

```

0000 00 13 46 2f da 60 00 23 6c 86 c0 27 86 dd 60 00  ..F/.# 1...
0010 00 00 00 38 3c 40 3f fe 01 00 00 00 00 06 02 23  ..8<@?  ....#
0020 6c ff fe 86 c0 27 3f fe 01 00 00 00 00 00 f1 00  1...?  ....?
0030 00 00 00 00 00 01 87 02 01 02 00 00 c9 10 3f fe  ..# 1...?
0040 01 00 00 00 00 00 02 23 6c ff fe 86 c0 27 3b 03  ..# 1...?
0050 05 00 82 dd 47 c2 00 00 ff ff 01 00 03 10 3f fe  ..G.# 1...?
0060 01 00 00 00 00 06 02 23 6c ff fe 86 c0 27  ....# 1...

```

**Fig. 3.4.** Missatges Binding Update amb flag dormant

La figura 3.5 mostra com l'HA reconeix el missatge BU amb el flag dormant:

```

Sun Apr 19 20:30:11 mh_send: remote CoA 3ffe:100:0:5:223:6cff:fe86:c027
Sun Apr 19 20:31:08 mh_bu_parse: Binding Update Received
Sun Apr 19 20:31:08 insert_mn: ip6 mn 3ffe:100:0:0:223:6cff:fe86:c027
Sun Apr 19 20:31:08 insert_mn: MN dormant insertado
Sun Apr 19 20:31:08 ha_rcv_bu_main: I received a BU with the dormant flag set
Sun Apr 19 20:31:08 ha_rcv_bu_main: Home address 3ffe:100:0:0:223:6cff:fe86:c027
Sun Apr 19 20:31:08 ha_rcv_bu_main: Care-of address 3ffe:100:0:5:223:6cff:fe86:c027
Sun Apr 19 20:31:08 ha_rcv_bu_main: Lifetime 64767 s
Sun Apr 19 20:31:08 tunnel_mod: modifying tunnel 54 end points with from
3ffe:100:0:0:f100:0:0:1 to 3ffe:100:0:0:f100:0:0:1
Sun Apr 19 20:31:08 __tunnel_mod: modified tunnel iface ip6tnl1 (54)from
3ffe:100:0:0:f100:0:0:1 to 3ffe:100:0:0:f100:0:0:1
Sun Apr 19 20:31:08 mh_send_ba: status 0
Sun Apr 19 20:31:08 mh_send: sending MH type 6
from 3ffe:100:0:0:f100:0:0:1
to 3ffe:100:0:0:223:6cff:fe86:c027
Sun Apr 19 20:31:08 mh_send: remote CoA 3ffe:100:0:5:223:6cff:fe86:c027

```

**Fig. 3.5.** Debug de l'HA en rebre el BU amb flag dormant

### 3.2.1. Prova 1: l'MN canvia a l'estat dormant

L'MN es comporta tal com s'espera, ja que mentre envia o rep trànsit no canvia a l'estat dormant.

Per tal de decidir si el node ha rebut o no trànsit, es llegeixen les estadístiques de la interfície IPv6-IPv6 per on s'envien les dades cap a l'MN, tal com es mostra a la figura 3.6:

```
Sun Apr 19 20:30:06 md_update_router_stats: rtr_addr_new: 3ffe:100:0:5:f100:0:0:11
Sun Apr 19 20:30:06 mn_chk_ho_verdict: rtr_addr_old 3ffe:100:0:5:f100:0:0:11
Sun Apr 19 20:30:10 bu_refresh: Bul refresh type: 0
Sun Apr 19 20:30:10 mn_get_home_lifetime: CoA lifetime 2591996 s, HoA lifetime 2591944 s,
BU lifetime 60 s
Sun Apr 19 20:30:10 mn_send_bu_msg: valor del wakeup 0
new_val = 1560
last_val = 0
Sun Apr 19 20:30:10 mn_send_bu_msg: Node Actiu
Sun Apr 19 20:30:10 mh_send: sending MH type 5
```

**Fig. 3.6.** L'MN rep trànsit

Una vegada ha transcorregut un període Binding Lifetime sense trànsit (és a dir, que les estadístiques no han variat), el node passa automàticament a dormint, tal com es mostra a la figura 3.7:

```
Sun Apr 19 20:33:00 md_rcv_ra: rtr_addr_new: 3ffe:100:0:5:f100:0:0:11
Sun Apr 19 20:33:00 md_update_router_stats: add coa 3ffe:100:0:5:223:6cff:fe86:c027 on
interface (4)
Sun Apr 19 20:33:00 md_update_router_stats: rtr_addr_new: 3ffe:100:0:5:f100:0:0:11
Sun Apr 19 20:33:00 mn_chk_ho_verdict: rtr_addr_old 3ffe:100:0:5:f100:0:0:11
Sun Apr 19 20:33:04 bu_refresh: Bul refresh type: 0
Sun Apr 19 20:33:04 mn_get_home_lifetime: CoA lifetime 2591995 s, HoA lifetime 2591770 s,
BU lifetime 60 s
Sun Apr 19 20:33:04 mn_send_bu_msg: valor del wakeup 0
new_val = 2600
last_val = 2600
Sun Apr 19 20:33:04 mn_send_bu_msg: Node Dormant
Sun Apr 19 20:33:04 mh_send: sending MH type 5
from 3ffe:100:0:0:223:6cff:fe86:c027
to 3ffe:100:0:0:f100:0:0:1
Sun Apr 19 20:33:04 mh_send: local CoA 3ffe:100:0:5:223:6cff:fe86:c027
Sun Apr 19 20:33:04 mn_send_bu_msg: Valor de i_dormant 1
Sun Apr 19 20:33:04 mn_send_bu_msg: Valor de tid 48
Sun Apr 19 20:33:04 mn_send_bu_msg: Passem a estat dormant
Sun Apr 19 20:33:04 md_stop: Parem serveis de node MN
Sun Apr 19 20:33:04 md_stop: Serveis de node MN aturats
SSID llegit fe80:000:0000:0005:f100:0000:0000:0011
Sun Apr 19 20:33:04 ssid_handler: llegint SSID des de 3ffe:100:0:5:f100:0:0:11 on iface 4
Sun Apr 19 20:33:04 ssid_handler: llegint SSID des de fe80:0:0:5:f100:0:0:11 on iface 4
Sun Apr 19 20:33:04 ssid_handler: rtr_addr_new: 3ffe:100:0:5:f100:0:0:11
Sun Apr 19 20:33:04 ssid_handler: SSID router: 3ffe:100:0:5:f100:0:0:11
Sun Apr 19 20:33:04 ssid_handler: Direccions iguals, node segueix en dormant
Sun Apr 19 20:33:04 md_add_backup_router: add new backup router fe80:0:0:5:f100:0:0:11 on
interface eth1 (4)
Sun Apr 19 20:33:04 mn_send_bu_msg: Node en dormant
Sun Apr 19 20:33:04 bu_refresh: Entro al else de bu_refresh
Sun Apr 19 20:33:04 bul_update_timer: Updating timer
== BUL_ENTRY ==
Home address 3ffe:100:0:0:223:6cff:fe86:c027
```

```
Care-of address 3ffe:100:0:5:223:6cff:fe86:c027
CN address      3ffe:100:0:0:f100:0:0:1
lifetime = 262140, delay = 1000
```

**Fig. 3.7.** L'MN passa a dormant després de rebre trànsit

### 3.2.2. Prova 2: l'MN canvia a l'estat actiu

El resultat d'aquesta prova és coherent amb els resultats esperats. Mentre l'MN no rep o envia trànsit, es manté en l'estat dormant.

La figura 3.8 mostra com després de rebre trànsit, l'MN passa a l'estat actiu i modifica el túnel per a rebre dades:

```
Sun Apr 19 20:31:08 ssid_handler: llegint SSID des de 3ffe:100:0:5:f100:0:0:11 on iface 4
Sun Apr 19 20:31:08 ssid_handler: llegint SSID des de fe80:0:0:5:f100:0:0:11 on iface 4
Sun Apr 19 20:31:08 ssid_handler: rtr_addr_new: 3ffe:100:0:5:f100:0:0:11
Sun Apr 19 20:31:08 ssid_handler: SSID router: 3ffe:100:0:5:f100:0:0:11
Sun Apr 19 20:31:08 ssid_handler: Direccions iguals, node segueix en dormant
Sun Apr 19 20:31:08 md_change_default_router: add new router fe80:0:0:5:f100:0:0:11 on
interface eth1 (4)
Sun Apr 19 20:31:08 md_update_router_stats: add coa 3ffe:100:0:5:223:6cff:fe86:c027 on
interface (4)
Sun Apr 19 20:31:08 md_update_router_stats: rtr_addr_new: 3ffe:100:0:5:f100:0:0:11
Sun Apr 19 20:31:10 mn_move: 1844
Sun Apr 19 20:31:10 mn_move: in foreign net
Sun Apr 19 20:31:10 mn_block_rule_add: blackhole is already set.
Sun Apr 19 20:31:10 do_handoff: rtr_addr_old : 3ffe:100:0:5:f100:0:0:11
Sun Apr 19 20:31:10 do_handoff: rtr_addr_new : 3ffe:100:0:5:f100:0:0:11
Sun Apr 19 20:31:10 calculate_distance: coord_res = 0,0
Sun Apr 19 20:31:10 do_handoff: Distance = 0
Sun Apr 19 20:31:10 mn_send_home_bu: 923
Sun Apr 19 20:31:10 mn_get_home_lifetime: CoA lifetime 2591998 s, HoA lifetime 2591884 s,
BU lifetime 60 s
Sun Apr 19 20:31:10 mn_send_home_bu: Moved to foreign network
Sun Apr 19 20:31:10 mn_send_home_bu: Bule for HA exists. Updating it.
Sun Apr 19 20:31:10 tunnel_mod: modifying tunnel 9 end points with from
3ffe:100:0:5:223:6cff:fe86:c027 to 3ffe:100:0:5:223:6cff:fe86:c027
Sun Apr 19 20:31:22 mn_send_bu_msg: valor del wakeup 1
new_val = 1560
last_val = 1560
Sun Apr 19 20:31:22 mn_send_bu_msg: Node Actiu
Sun Apr 19 20:31:22 mh_send: sending MH type 5
from 3ffe:100:0:0:223:6cff:fe86:c027
to 3ffe:100:0:0:f100:0:0:1
Sun Apr 19 20:31:22 mh_send: local CoA 3ffe:100:0:5:223:6cff:fe86:c027
Sun Apr 19 20:31:22 mn_send_bu_msg: Valor de i_dormant 0
Sun Apr 19 20:31:22 mn_send_bu_msg: Valor de tid -1268073584
Sun Apr 19 20:31:22 mn_send_bu_msg: Passem a estat actiu
Sun Apr 19 20:31:22 mn_send_bu_msg: Node activat
Sun Apr 19 20:31:22 tunnel_mod: modifying tunnel 9 end points with from
3ffe:100:0:5:223:6cff:fe86:c027 to 3ffe:100:0:0:f100:0:0:1
Sun Apr 19 20:31:22 __tunnel_mod: modified tunnel iface ip6tnl1 (9)from
3ffe:100:0:5:223:6cff:fe86:c027 to 3ffe:100:0:0:f100:0:0:1
```

**Fig. 3.8.** L'MN passa a actiu després de rebre trànsit

La figura 3.9 mostra que l'MN ha despertat, ja que l'HA ha rebut un paquet dirigit cap a l'MN. Per a localitzar l'MN, l'HA envia un Binding Refresh Request cap a l'MN. Una vegada rep el Binding Update com a resposta del BRR, l'HA també modifica el túnel IPv6-IPv6 al seu torn:

```
Sun Apr 19 20:31:08 ha_recv_bu_main: I received a BU with the dormant flag set
Sun Apr 19 20:31:08 ha_recv_bu_main: Home address 3ffe:100:0:0:223:6cff:fe86:c027
Sun Apr 19 20:31:08 ha_recv_bu_main: Care-of address 3ffe:100:0:5:223:6cff:fe86:c027
Sun Apr 19 20:31:08 ha_recv_bu_main: Lifetime 64767 s
Sun Apr 19 20:31:08 tunnel_mod: modifying tunnel 54 end points with from
3ffe:100:0:0:f100:0:0:1 to 3ffe:100:0:0:f100:0:0:1
Sun Apr 19 20:31:08 __tunnel_mod: modified tunnel iface ip6tn1 (54)from
3ffe:100:0:0:f100:0:0:1 to 3ffe:100:0:0:f100:0:0:1
Sun Apr 19 20:31:08 mh_send_ba: status 0
Sun Apr 19 20:31:08 mh_send: sending MH type 6
from 3ffe:100:0:0:f100:0:0:1
to 3ffe:100:0:0:223:6cff:fe86:c027
Sun Apr 19 20:31:08 mh_send: remote CoA 3ffe:100:0:5:223:6cff:fe86:c027
Sun Apr 19 20:31:23 first_msg_to_mn_dormant: Estoy en el for buscando el mn!
Sun Apr 19 20:31:23 first_msg_to_mn_dormant: aux 3ffe:100:0:0:223:6cff:fe86:c027
Sun Apr 19 20:31:23 first_msg_to_mn_dormant: He encontrado el mn!
Sun Apr 19 20:31:23 copy_buffer: Detectado paquete IPv6 dirigido a mn
Sun Apr 19 20:31:23 copy_buffer: Inserto el mensaje a enviar al mn
Sun Apr 19 20:31:23 copy_buffer: Mensaje insertado
Sun Apr 19 20:31:23 first_msg_to_mn_dormant: len_msg 104
Sun Apr 19 20:31:23 mh_send: sending MH type 0
from 3ffe:100:0:0:f100:0:0:1
to 3ffe:100:0:5:223:6cff:fe86:c027
Sun Apr 19 20:31:23 send_to_areapaging: ip6 bind_coa 3ffe:100:0:5:223:6cff:fe86:c027
Sun Apr 19 20:31:23 send_to_areapaging: Reference: 0:0:0:0:3ffe:100:0:5
Sun Apr 19 20:31:23 mh_bu_parse: Binding Update Received
Sun Apr 19 20:31:23 ha_recv_bu_main: Home address 3ffe:100:0:0:223:6cff:fe86:c027
Sun Apr 19 20:31:23 ha_recv_bu_main: Estoy en ha_recv_bu dentro del if
Sun Apr 19 20:31:23 ha_recv_bu_main: aux 3ffe:100:0:0:223:6cff:fe86:c027
Sun Apr 19 20:31:23 ha_recv_bu_main: He recibido el bu!!
Sun Apr 19 20:31:23 tunnel_mod: modifying tunnel 54 end points with from
3ffe:100:0:0:f100:0:0:1 to 3ffe:100:0:5:223:6cff:fe86:c027
```

Fig. 3.9. L'HA desperta el node MN

La figura 3.10 mostra com l'HA envia el missatge BRR cap a l'MN:

No. -	Time	Source	Destination	Protocol	Info
256	8.472919	3ffe:100::223:6cff:fe86:c027	3ffe:100::f100:0:0:1	MIPv6	Binding Update
259	9.493318	3ffe:100::f100:0:0:1	3ffe:100::223:6cff:fe86:c027	MIPv6	Binding Acknowledgement
312	65.474022	3ffe:100::223:6cff:fe86:c027	3ffe:100::f100:0:0:1	MIPv6	Binding Update
313	65.483803	3ffe:100::f100:0:0:1	3ffe:100::223:6cff:fe86:c027	MIPv6	Binding Acknowledgement
340	122.472889	3ffe:100::223:6cff:fe86:c027	3ffe:100::f100:0:0:1	MIPv6	Binding Update
343	122.493100	3ffe:100::f100:0:0:1	3ffe:100::223:6cff:fe86:c027	MIPv6	Binding Acknowledgement
361	137.800003	3ffe:100::f100:0:0:1	3ffe:100:0:5:223:6cff:fe86:c02	MIPv6	Binding Refresh Request
362	137.908231	3ffe:100::223:6cff:fe86:c027	3ffe:100::f100:0:0:1	MIPv6	Binding Update
363	137.922165	3ffe:100::f100:0:0:1	3ffe:100::223:6cff:fe86:c027	MIPv6	Binding Acknowledgement
416	182.462946	3ffe:100::223:6cff:fe86:c027	3ffe:100::f100:0:0:1	MIPv6	Binding Update
417	182.476032	3ffe:100::f100:0:0:1	3ffe:100::223:6cff:fe86:c027	MIPv6	Binding Acknowledgement
437	239.461911	3ffe:100::223:6cff:fe86:c027	3ffe:100::f100:0:0:1	MIPv6	Binding Update
438	239.468310	3ffe:100::f100:0:0:1	3ffe:100::223:6cff:fe86:c027	MIPv6	Binding Acknowledgement
460	274.712722	3ffe:100::223:6cff:fe86:c027	3ffe:100::f100:0:0:1	MIPv6	Binding Update
461	274.801091	3ffe:100::f100:0:0:1	3ffe:100::223:6cff:fe86:c027	MIPv6	Binding Acknowledgement

Frame 361 (62 bytes on wire, 62 bytes captured)  
 Ethernet II, Src: D-Link\_2f:e2:d9 (00:13:46:2f:e2:d9), Dst: Apple\_86:c0:27 (00:23:6c:86:c0:27)  
 Internet Protocol version 6  
 Mobile IPv6 / Network Mobility  
 Payload protocol: IPv6 no next header (0x3b)  
 Header length: 0 (8 bytes)  
 Mobility Header Type: Binding Refresh Request (0)  
 Reserved: 0x00  
 Checksum: 0x239c  
 Binding Refresh Request

**Fig. 3.10.** L'HA envia un missatge Binding Refresh Request cap a l'MN

La figura 3.11 mostra com el missatge Binding Update cap a l'HA (abans de fer el BRR) conté el flag dormant posat a 1, per tal d'indicar que el node es troba en l'estat dormant:

No. -	Time	Source	Destination	Protocol	Info
313	65.483803	3ffe:100::f100:0:0:1	3ffe:100::223:6cff:fe86:c027	MIPv6	Binding Acknowledgement
340	122.472889	3ffe:100::223:6cff:fe86:c027	3ffe:100::f100:0:0:1	MIPv6	Binding Update
343	122.493100	3ffe:100::f100:0:0:1	3ffe:100::223:6cff:fe86:c027	MIPv6	Binding Acknowledgement
361	137.800003	3ffe:100::f100:0:0:1	3ffe:100:0:5:223:6cff:fe86:c02	MIPv6	Binding Refresh Request
362	137.908231	3ffe:100::223:6cff:fe86:c027	3ffe:100::f100:0:0:1	MIPv6	Binding Update
363	137.922165	3ffe:100::f100:0:0:1	3ffe:100::223:6cff:fe86:c027	MIPv6	Binding Acknowledgement

..1.. .... = Acknowledge (A) flag: Binding Acknowledgement Requested  
 .1.. .... = Home Registration (H) flag: Home Registration  
 .1.1. .... = Link-Local Compatibility (L) flag: Link-Local Address Compatibility  
 ...0 .... = Key Management Compatibility (K) flag: No Key Management Mobility Compatibility  
 ....0... = MAP Registration Compatibility (M) flag: NO MAP Registration Compatibility  
 ....0.. = Mobile Router (R) flag: No Mobile Router Compatibility  
 ....0.0. = Proxy Registration (P) flag: No Proxy Registration  
 Lifetime: 65535 (262140 seconds)  
 Mobility options  
 PadN: 2 bytes  
 Alternate care-of address: 3ffe:100:0:5:223:6cff:fe86:c027 (3ffe:100:0:5:223:6cff:fe86:c027)

```

0000 00 13 46 2f e2 d9 00 23 6c 86 c0 27 86 dd 60 00  ..F/..# 1.....
0010 00 00 00 38 3c 40 3f fe 01 00 00 00 00 05 02 23  ...8<@? .....#
0020 6c ff fe 86 c0 27 3f fe 01 00 00 00 00 00 f1 00  |.....? .....?
0030 00 00 00 00 00 01 87 02 01 02 00 00 c9 10 3f fe  |.....# 1.....?
0040 01 00 00 00 00 00 02 23 6c ff fe 86 c0 27 3b 03  |.....# 1.....?
0050 05 00 6f 84 5b 1c 00 00 ff ff 01 00 03 10 3f fe  |..o.[. 1.....?
0060 01 00 00 00 00 05 02 23 6c ff fe 86 c0 27  |.....# 1.....
  
```

**Fig. 3.11.** L'MN envia un Binding Update amb el flag dormant activat

La figura 3.12 mostra com el missatge Binding Update cap a l'HA (després de fer el BRR) conté el flag dormant posat a 0, per tal d'indicar que el node ha passat a l'estat actiu:

No. -	Time	Source	Destination	Protocol	Info
313	65.483803	3ffe:100::f100:0:0:1	3ffe:100::223:6cff:fe86:c027	MIPv6	Binding Acknowledgement
340	122.472889	3ffe:100::223:6cff:fe86:c027	3ffe:100::f100:0:0:1	MIPv6	Binding Update
343	122.493100	3ffe:100::f100:0:0:1	3ffe:100::223:6cff:fe86:c027	MIPv6	Binding Acknowledgement
361	137.800003	3ffe:100::f100:0:0:1	3ffe:100:0:5:223:6cff:fe86:c027	MIPv6	Binding Refresh Request
362	137.908291	3ffe:100::223:6cff:fe86:c027	3ffe:100::f100:0:0:1	MIPv6	Binding Update
363	137.922165	3ffe:100::f100:0:0:1	3ffe:100::223:6cff:fe86:c027	MIPv6	Binding Acknowledgement

```

1.. .... = Acknowledge (A) flag: Binding Acknowledgement Requested
..1.. .... = Home Registration (H) flag: Home Registration
...1. .... = Link-Local Compatibility (L) flag: Link-Local Address Compatibility
...0. .... = Key Management Compatibility (K) flag: No Key Management Mobility Compatibility
.... 0... = MAP Registration Compatibility (M) flag: No MAP Registration Compatibility
.... .0.. = Mobile Router (R) flag: No Mobile Router Compatibility
.... ..0. = Proxy Registration (P) flag: No Proxy Registration
Lifetime: 15 (60 seconds)
☐ Mobility Options
  PadN: 2 bytes
  Alternate care-of address: 3ffe:100:0:5:223:6cff:fe86:c027 (3ffe:100:0:5:223:6cff:fe86:c027)
0000 00 13 46 2f e2 d9 00 23 6c 86 c0 27 86 dd 60 00  .F/...# |.....
0010 00 00 00 38 3c 40 3f fe 01 00 00 00 00 05 02 23  ..8<@?. .....#
0020 6c ff fe 86 c0 27 3f fe 01 00 00 00 00 00 f1 00  |.....?.....?
0030 00 00 00 00 00 01 87 02 01 02 00 00 c9 10 3f fe  .....# |.....?
0040 01 00 00 00 00 00 02 23 6c ff fe 86 c0 27 3b 03  .....# |.....?
0050 05 00 70 73 5b 1e 80 00 00 0f 01 00 03 10 3f fe  ..ps[. |.....?
0060 01 00 00 00 00 05 02 23 6c ff fe 86 c0 27  .....# |.....

```

**Fig. 3.12.** L'MN envia un Binding Update amb el flag dormant desactivat

### 3.2.3. Prova 3: l'MN avisa correctament l'HA per canvi d'àrea de paging

El resultat d'aquesta prova és coherent amb els resultats esperats, ja que l'MN avisa correctament l'HA en cas de canvi d'àrea de paging.

A la primera prova (figura 3.13) s'ha configurat una distància de threshold igual a 1 i, per tant, el canvi de xarxa entre 3ffe:100:0:6:f100:0:0:11 i 3ffe:100:0:5:f100:0:0:11 suposa un canvi d'àrea de paging.

A la segona prova (figura 3.14), la distància de threshold és igual a 3 i, per tant, no es canvia d'àrea de paging.

La figura 3.13 mostra com l'MN en estat dormant detecta el canvi d'àrea de paging, canvia de CoA i passa a l'estat actiu:

```

Sun Apr 19 21:03:19 ssid_handler: llegint SSID des de 3ffe:100:0:6:f100:0:0:12 on iface 4
Sun Apr 19 21:03:19 ssid_handler: llegint SSID des de fe80:0:0:6:f100:0:0:12 on iface 4
Sun Apr 19 21:03:19 ssid_handler: rtr_addr_new: 3ffe:100:0:6:f100:0:0:12
Sun Apr 19 21:03:19 ssid_handler: SSID router: 3ffe:100:0:6:f100:0:0:12
Sun Apr 19 21:03:19 ssid_handler: Direccions iguals, node segueix en dormant
Sun Apr 19 21:03:19 md_change_default_router: add new router fe80:0:0:6:f100:0:0:12 on
interface eth1 (4)
Sun Apr 19 21:03:19 md_update_router_stats: add coa 3ffe:100:0:6:223:6cff:fe86:c027 on
interface (4)
Sun Apr 19 21:03:19 md_update_router_stats: rtr_addr_new: 3ffe:100:0:6:f100:0:0:12
Sun Apr 19 21:03:21 mn_move: 1844
Sun Apr 19 21:03:21 mn_move: in foreign net
Sun Apr 19 21:03:21 mn_block_rule_add: blackhole is already set.
Sun Apr 19 21:03:21 mn_send_home_bu: 923

```

```

Sun Apr 19 21:03:21 mn_get_home_lifetime: CoA lifetime 2591999 s, HoA lifetime 2591941 s,
BU lifetime 60 s
Sun Apr 19 21:03:21 mn_send_home_bu: Moved to foreign network
Sun Apr 19 21:03:21 mn_send_home_bu: Bule for HA exists. Updating it.
Sun Apr 19 21:03:21 tunnel_mod: modifying tunnel 6 end points with from
3ffe:100:0:6:223:6cff:fe86:c027 to 3ffe:100:0:6:223:6cff:fe86:c027
SSID lilegit fe80:000:0000:0005:f100:0000:0000:0011
Sun Apr 19 21:03:49 ssid_handler: llegint SSID des de 3ffe:100:0:5:f100:0:0:11 on iface 4
Sun Apr 19 21:03:49 ssid_handler: lilegit SSID des de fe80:0:0:5:f100:0:0:11 on iface 4
Sun Apr 19 21:03:49 ssid_handler: rtr_addr_new: 3ffe:100:0:6:f100:0:0:12
Sun Apr 19 21:03:49 ssid_handler: SSID router: 3ffe:100:0:5:f100:0:0:11
Sun Apr 19 21:03:49 ssid_handler: Direccions diferents, comprovant distancia de paging
Sun Apr 19 21:03:49 calculate_distance: coord_res = -1,1
Sun Apr 19 21:03:49 ssid_handler: Distance = 2
Sun Apr 19 21:03:49 ssid_handler: DISTANCE_PAG_AREA = 1
Sun Apr 19 21:03:49 md_add_backup_router: add new backup router fe80:0:0:5:f100:0:0:11 on
interface eth1 (4)
Sun Apr 19 21:03:49 ssid_handler: Salt superior a distancia paging, node activat
Sun Apr 19 21:03:50 md_change_default_router: add new router fe80:0:0:5:f100:0:0:11 on
interface eth1 (4)
Sun Apr 19 21:03:50 md_expire_router: expiring router fe80:0:0:6:f100:0:0:12 on iface eth1 (4)
Sun Apr 19 21:03:50 md_update_router_stats: add coa 3ffe:100:0:5:223:6cff:fe86:c027 on
interface (4)
Sun Apr 19 21:03:50 md_update_router_stats: rtr_addr_new: 3ffe:100:0:5:f100:0:0:11
Sun Apr 19 21:03:50 mn_move: 1844
Sun Apr 19 21:03:50 mn_move: in foreign net
Sun Apr 19 21:03:50 mn_block_rule_add: blackhole is already set.
Sun Apr 19 21:03:50 do_handoff: rtr_addr_old : 3ffe:100:0:6:f100:0:0:12
Sun Apr 19 21:03:50 do_handoff: rtr_addr_new : 3ffe:100:0:5:f100:0:0:11
Sun Apr 19 21:03:50 calculate_distance: coord_res = -1,1
Sun Apr 19 21:03:50 do_handoff: New rtr_addr_old : 3ffe:100:0:5:f100:0:0:11
Sun Apr 19 21:03:50 do_handoff: Distance = 2
Sun Apr 19 21:03:50 mn_send_home_bu: 923
Sun Apr 19 21:03:50 mn_get_home_lifetime: CoA lifetime 2591997 s, HoA lifetime 2591912 s,
BU lifetime 60 s
Sun Apr 19 21:03:50 mn_send_home_bu: Moved to foreign network
Sun Apr 19 21:03:50 mn_send_home_bu: Bule for HA exists. Updating it.
Sun Apr 19 21:03:50 mn_send_bu_msg: valor del wakeup 2
new_val = 0
last_val = 0
Sun Apr 19 21:03:50 mh_send: sending MH type 5
from 3ffe:100:0:0:223:6cff:fe86:c027
to 3ffe:100:0:0:f100:0:0:1
Sun Apr 19 21:03:50 mh_send: local CoA 3ffe:100:0:5:223:6cff:fe86:c027
Sun Apr 19 21:03:50 mn_send_bu_msg: Valor de i_dormant 1
Sun Apr 19 21:03:50 mn_send_bu_msg: Valor de tid -1269359728
Sun Apr 19 21:03:50 bul_update_timer: Updating timer
== BUL_ENTRY ==
Home address 3ffe:100:0:0:223:6cff:fe86:c027
Care-of address 3ffe:100:0:5:223:6cff:fe86:c027
CN address 3ffe:100:0:0:f100:0:0:1
lifetime = 262140, delay = 1000
flags: IP6_MH_BU_HOME IP6_MH_BU_ACK IP6_MH_BU_LLOCAL

```

**Fig. 3.13.** L'MN canvia d'àrea de paging



Per part de l'HA, s'observa com l'MN avisa del canvi d'àrea de paging. Tot i el canvi d'àrea de paging, es considera que el node segueix en dormant, tal com s'indica a la figura 3.14:

```
Sun Apr 19 21:02:23 mh_send: remote CoA 3ffe:100:0:6:223:6cff:fe86:c027
Sun Apr 19 21:03:19 mh_bu_parse: Binding Update Received
Sun Apr 19 21:03:19 insert_mn: ip6 mn 3ffe:100:0:0:223:6cff:fe86:c027
Sun Apr 19 21:03:19 insert_mn: MN dormant insertado
Sun Apr 19 21:03:19 ha_rcv_bu_main: I received a BU with the dormant flag set
Sun Apr 19 21:03:19 ha_rcv_bu_main: Home address 3ffe:100:0:0:223:6cff:fe86:c027
Sun Apr 19 21:03:19 ha_rcv_bu_main: Care-of address 3ffe:100:0:6:223:6cff:fe86:c027
Sun Apr 19 21:03:19 ha_rcv_bu_main: Lifetime 64767 s
Sun Apr 19 21:03:19 tunnel_mod: modifying tunnel 55 end points with from
3ffe:100:0:0:f100:0:0:1 to 3ffe:100:0:0:f100:0:0:1
Sun Apr 19 21:03:19 mh_send_ba: status 0
Sun Apr 19 21:03:19 mh_send: sending MH type 6
from 3ffe:100:0:0:f100:0:0:1
to 3ffe:100:0:0:223:6cff:fe86:c027
Sun Apr 19 21:03:19 mh_send: remote CoA 3ffe:100:0:6:223:6cff:fe86:c027
Sun Apr 19 21:03:51 mh_bu_parse: Binding Update Received
Sun Apr 19 21:03:51 insert_mn: MN ya está en el listado...actualizando la información
Sun Apr 19 21:03:51 ha_rcv_bu_main: I received a BU with the dormant flag set
Sun Apr 19 21:03:51 ha_rcv_bu_main: Home address 3ffe:100:0:0:223:6cff:fe86:c027
Sun Apr 19 21:03:51 ha_rcv_bu_main: Care-of address 3ffe:100:0:5:223:6cff:fe86:c027
Sun Apr 19 21:03:51 ha_rcv_bu_main: Lifetime 64767 s
Sun Apr 19 21:03:51 tunnel_mod: modifying tunnel 55 end points with from
3ffe:100:0:0:f100:0:0:1 to 3ffe:100:0:0:f100:0:0:1
```

**Fig. 3.14.** L'HA rep la notificació de l'MN del canvi d'àrea de paging

El fet que canviï a l'estat actiu per a notificar el canvi d'àrea de paging (però el node segueixi en l'estat dormant) es basa en raons de disseny, ja que no es considera que el node hagi rebut trànsit com per passar a l'estat actiu.

La figura 3.15 mostra com l'MN en estat dormant detecta que el canvi de xarxa no és superior a la distància de threshold i, per tant, no passa a l'estat actiu. En aquesta figura també es pot veure com tampoc no canvia de CoA en aquest procés (és a dir, que conserva la CoA assignada a la xarxa anterior):

```
Sat Apr 25 20:05:43 ssid_handler: llegint SSID des de 3ffe:100:0:6:f100:0:0:12 on iface 5
Sat Apr 25 20:05:43 ssid_handler: llegint SSID des de fe80:0:0:6:f100:0:0:12 on iface 5
Sat Apr 25 20:05:43 ssid_handler: rtr_addr_new: 3ffe:100:0:6:f100:0:0:12
Sat Apr 25 20:05:43 ssid_handler: SSID router: 3ffe:100:0:6:f100:0:0:12
Sat Apr 25 20:05:43 ssid_handler: Direccions iguals, node segueix en dormant
Sat Apr 25 20:05:43 md_update_router_stats: add coa 3ffe:100:0:6:21e:e5ff:fedf:9dd7 on interface (5)
Sat Apr 25 20:05:43 md_update_router_stats: rtr_addr_new: 3ffe:100:0:6:f100:0:0:12
SSID llegit fe80:000:0000:0005:f100:0000:0000:0011
Sat Apr 25 20:06:13 ssid_handler: llegint SSID des de 3ffe:100:0:5:f100:0:0:11 on iface 5
Sat Apr 25 20:06:13 ssid_handler: llegint SSID des de fe80:0:0:5:f100:0:0:11 on iface 5
Sat Apr 25 20:06:13 ssid_handler: rtr_addr_new: 3ffe:100:0:6:f100:0:0:12
Sat Apr 25 20:06:13 ssid_handler: SSID router: 3ffe:100:0:5:f100:0:0:11
Sat Apr 25 20:06:13 ssid_handler: Direccions diferents, comprovant distancia de paging
Sat Apr 25 20:06:13 calculate_distance: coord_res = -1,1
Sat Apr 25 20:06:13 ssid_handler: Distance = 2
Sat Apr 25 20:06:13 ssid_handler: DISTANCE_PAG_AREA = 3
```

```

Sat Apr 25 20:06:13 ssid_handler: Salt inferior a distancia paging, rtr_addr_new:
3ffe:100:0:5:f100:0:0:11
Sat Apr 25 20:06:13 md_add_backup_router: add new backup router fe80:0:0:5:f100:0:0:11 on
interface ra0 (5)
Sat Apr 25 20:06:14 md_change_default_router: add new router fe80:0:0:5:f100:0:0:11 on
interface ra0 (5)
Sat Apr 25 20:06:14 md_expire_router: expiring router fe80:0:0:6:f100:0:0:12 on iface ra0 (5)
Sat Apr 25 20:06:14 md_update_router_stats: add coa 3ffe:100:0:6:21e:e5ff:fedf:9dd7 on
interface (5)
Sat Apr 25 20:06:14 md_update_router_stats: rtr_addr_new: 3ffe:100:0:5:f100:0:0:11

```

**Fig. 3.15.** L'MN no canvia d'àrea de paging i conserva la CoA

A la figura s'observa com el dimoni UMIP afegeix correctament l'adreça *Local-Link* del nou AR, tal com es detalla al Capítol 2.

### 3.2.4. Prova 4: comportament de l'HA per a localitzar l'MN

Els resultats d'aquesta prova són coherents amb els esperats.

Ja que la nova implementació de paging no permet de moment localitzar un node quan no s'ha mogut d'àrea de paging, l'HA envia un missatge BRR cap a l'antiga CoA que té guardada de l'MN.

Es pot comprovar com aquesta funcionalitat és operativa, ja que els resultats són els obtinguts a la prova 2 (vegeu l'apartat 3.2.2).

### 3.2.5. Prova 5: retard

Els resultats d'aquestes proves són coherents amb els esperats, ja que s'observa un retard apreciable en el primer paquet que s'envia cap a l'MN.

Tot i això, se sol perdre el primer paquet cap a l'MN en estat dormant a raó de les modificacions a l'HA.

La figura 3.16 mostra l'enviament de paquets ICMPv6 del CN cap a l'MN en estat actiu:

```
[root@CN ~]# ping6 3ffe:100:0:0:223:6cff:fe86:c027
PING 3ffe:100:0:0:223:6cff:fe86:c027(3ffe:100::223:6cff:fe86:c027) 56 data bytes
64 bytes from 3ffe:100::223:6cff:fe86:c027: icmp_seq=0 ttl=62 time=871 ms
64 bytes from 3ffe:100::223:6cff:fe86:c027: icmp_seq=1 ttl=62 time=5.63 ms
64 bytes from 3ffe:100::223:6cff:fe86:c027: icmp_seq=2 ttl=62 time=97.7 ms
64 bytes from 3ffe:100::223:6cff:fe86:c027: icmp_seq=3 ttl=62 time=118 ms
64 bytes from 3ffe:100::223:6cff:fe86:c027: icmp_seq=4 ttl=62 time=134 ms
64 bytes from 3ffe:100::223:6cff:fe86:c027: icmp_seq=5 ttl=62 time=162 ms
64 bytes from 3ffe:100::223:6cff:fe86:c027: icmp_seq=6 ttl=62 time=178 ms
64 bytes from 3ffe:100::223:6cff:fe86:c027: icmp_seq=7 ttl=62 time=199 ms
64 bytes from 3ffe:100::223:6cff:fe86:c027: icmp_seq=8 ttl=62 time=11.1 ms
64 bytes from 3ffe:100::223:6cff:fe86:c027: icmp_seq=9 ttl=62 time=31.4 ms
64 bytes from 3ffe:100::223:6cff:fe86:c027: icmp_seq=10 ttl=62 time=51.6 ms
```

**Fig. 3.16.** El CN envia trànsit cap a l'MN actiu

La figura 3.17 mostra l'enviament de paquets ICMPv6 del CN cap a l'MN en estat dormant i com apareix el segon paquet duplicat, per la funció *first\_msg\_to\_mn\_dormant* implementada a [11] que reenvia el primer paquet rebut per l'HA cap a l'MN. Es descarta que el primer paquet (icmp = 0) es perdi per causa d'un excés de retard sinó per la pròpia implementació de la funció de captura i reenviament del primer paquet:

```
PING 3ffe:100:0:0:223:6cff:fe86:c027(3ffe:100::223:6cff:fe86:c027) 56 data bytes
64 bytes from 3ffe:100::223:6cff:fe86:c027: icmp_seq=1 ttl=62 time=39.9 ms
64 bytes from 3ffe:100::223:6cff:fe86:c027: icmp_seq=2 ttl=62 time=61.1 ms
64 bytes from 3ffe:100::223:6cff:fe86:c027: icmp_seq=3 ttl=62 time=71.8 ms
64 bytes from 3ffe:100::223:6cff:fe86:c027: icmp_seq=4 ttl=62 time=109 ms
64 bytes from 3ffe:100::223:6cff:fe86:c027: icmp_seq=5 ttl=62 time=133 ms
64 bytes from 3ffe:100::223:6cff:fe86:c027: icmp_seq=6 ttl=62 time=149 ms
64 bytes from 3ffe:100::223:6cff:fe86:c027: icmp_seq=7 ttl=62 time=153 ms
64 bytes from 3ffe:100::223:6cff:fe86:c027: icmp_seq=8 ttl=62 time=174 ms
64 bytes from 3ffe:100::223:6cff:fe86:c027: icmp_seq=9 ttl=62 time=217 ms
64 bytes from 3ffe:100::223:6cff:fe86:c027: icmp_seq=10 ttl=62 time=237 ms
64 bytes from 3ffe:100::223:6cff:fe86:c027: icmp_seq=11 ttl=62 time=38.1 ms
64 bytes from 3ffe:100::223:6cff:fe86:c027: icmp_seq=12 ttl=62 time=62.3 ms
64 bytes from 3ffe:100::223:6cff:fe86:c027: icmp_seq=1 ttl=62 time=11304 ms (DUP!)
```

**Fig. 3.17.** El CN envia trànsit cap a l'MN dormant

### 3.3. Conclusions

Totes les proves realitzades s'han superat amb èxit. Se'n detallen els resultats a la taula següent:

**Taula 3.1.** Resultats de les proves del dimoni UMIP modificat

Prova	Descripció	Resultat
P1.1	L'MN rep i transmet trànsit	Correcte
P1.2	L'MN rep trànsit	Correcte
P1.3	L'MN transmet trànsit	Correcte
P1.4	L'MN no rep ni transmet trànsit	Correcte
P2.1	L'MN no té trànsit per a enviar ni per a rebre	Correcte
P2.2	L'MN no té trànsit per a enviar	Correcte
P2.3	L'MN no té trànsit per a rebre	Correcte
P3.1	L'MN canvia de xarxa	Correcte
P3.2	L'MN canvia de xarxa dins la mateixa àrea de paging	Correcte
P3.3	L'MN canvia de xarxa i d'àrea de paging	Correcte
P4.1	El CN envia trànsit a l'MN que no ha canviat de xarxa	Correcte
P5.1	El CN envia trànsit a l'MN en l'estat actiu	Correcte
P5.2	El CN envia trànsit a l'MN en l'estat dormant	Correcte, però es perd el primer paquet

Una futura implementació d'aquest dimoni possibilitarà la localització de l'MN quan canviï de xarxa però no d'àrea de paging.

Per a aconseguir aquesta funcionalitat, l'HA hauria d'enviar directament el missatge Binding Request Refresh cap als possibles routers on es trobi l'MN, passant com a referència la darrera CoA de l'MN emmagatzemada per l'HA.

Els routers enviaran el paquet directament a l'MN gràcies al fet que coneixen la seva MAC (vegeu el Capítol 2). Només caldria, doncs, col·locar el paquet dins una trama amb la MAC de l'MN com a destí. Com a adreça IP de destí es podria utilitzar la seva HoA.

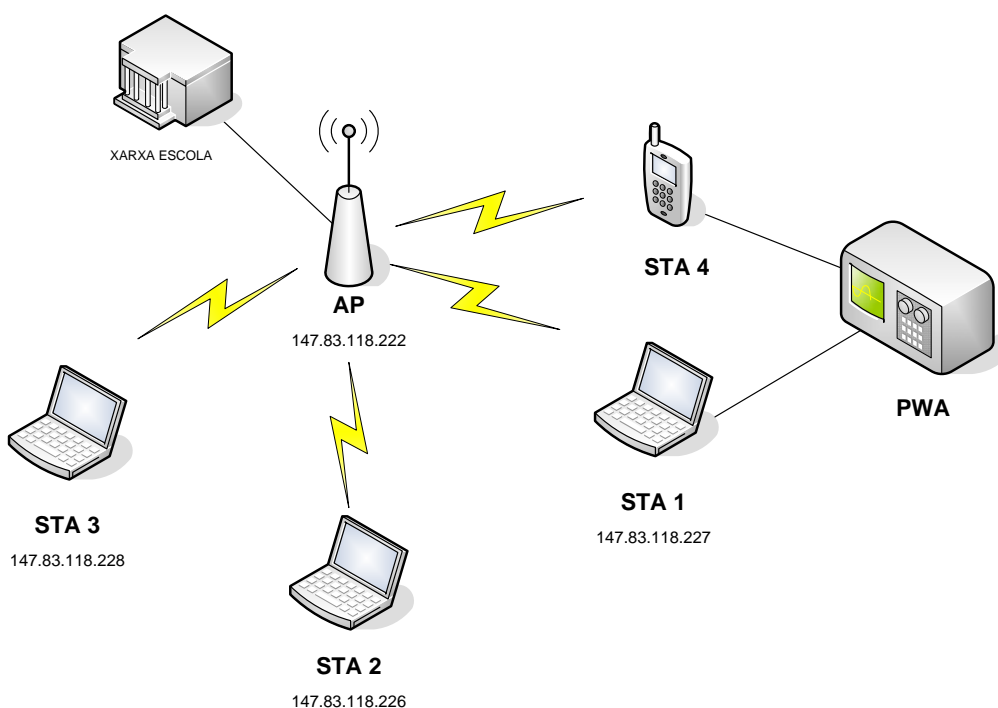
En el cas de l'enviament de dades del CN cap a l'MN, el primer paquet se sol perdre, ja que la funció *first\_msg\_to\_mn\_dormant* només arriba a capturar el segon paquet entrant (*icmp = 1*). En futures implementacions, seria interessant disposar d'una nova opció per a despertar amb un paquet de prova abans de començar una transmissió de dades.



## Capítol 4. Estudi de l'impacte de l'habilitació d'U-APSD i PSM sobre la bateria d'un dispositiu mòbil

En aquest capítol s'avalua la influència de l'habilitació de diversos sistemes d'estalvi d'energia en dispositius mòbils. Aquests sistemes d'estalvi s'han estudiat sobre diversos escenaris, tant amb trànsit a la xarxa com amb la xarxa aïllada.

Concretament, s'ha estimat l'impacte dels modes d'estalvi d'energia PSM i U-APSD. La figura 4.1 mostra la topologia de l'escenari de proves en la seva totalitat.



**Fig. 4.1.** Escenari general de proves

### 4.1. Escenari de proves

Durant les proves s'han plantejat diversos escenaris per a avaluar el comportament de la bateria.

Per a la realització de les proves s'han utilitzat diversos dispositius i programes per a avaluar els resultats obtinguts. Concretament, la mesura del consum dels dispositius mòbils s'ha obtingut amb l'analitzador de potència Agilent N6705, per al qual s'ha inclòs en l'Annex IV un manual que en detalla les especificacions i la configuració bàsica.

S'han distingit dos tipus d'eines: d'una banda, les relacionades amb el maquinari (vegeu la taula 4.1) i, de l'altra, les relacionades amb el programari (vegeu la taula 4.2).

**Taula 4.1.** Característiques tècniques del maquinari utilitzat

<b>STA 1</b>	
DESCRIPCIÓ	Ordinador portàtil Sony VAIO
MODEL	S5M
BATERIA	VGP-BPS2B 11.1V 4800mAh
CPU	Intel Pentium M 740 1,73 GHz
SISTEMA OPERATIU	Ubuntu Linux 7.10 Nautilus 6
TARGETA DE XARXA	Linksys WUSB600N
CHIPSET	Ralink RT2870 USB
DRIVER DE LA TARGETA DE XARXA	2008_0925_RT2870_Linux_STA_v1.4.0.0
CERTIFICACIONS DE LA TARGETA DE XARXA	802.11 A/B/G/N/H, WMM, WMM-PS, WPA/WPA2
<b>STA 2</b>	
DESCRIPCIÓ	Ordinador portàtil BENQ Joybook
MODEL	S72
BATERIA	I304RJ 10.8V 4800 mAh
CPU	Intel Pentium M 730 1,60 GHz
SISTEMA OPERATIU	Ubuntu 7.10 Nautilus 6
TARGETA DE XARXA	Intel PRO 2200 BG
CHIPSET	Intel 2200
DRIVER DE LA TARGETA DE XARXA	Intel 12.2.0.11
CERTIFICACIONS DE LA TARGETA DE XARXA	802.11 B/G, WMM, WPA/WPA2
<b>STA 3</b>	
DESCRIPCIÓ	Ordinador portàtil Sony VAIO
MODEL	FW11M
BATERIA	BPS13A/S 11,1V 4800mAh
CPU	Intel Core2 Duo P8200 2,26 GHz
SISTEMA OPERATIU	Windows Vista SP1
TARGETA DE XARXA	Intel WiFi Link 5100
CHIPSET	Intel 5100
DRIVER DE LA TARGETA DE XARXA	Intel 12.2.0.11
CERTIFICACIONS DE LA TARGETA DE XARXA	802.11 A/B/G/N, WMM, WPA/WPA2
<b>STA 4</b>	
DESCRIPCIÓ	Telèfon mòbil Nokia
MODEL	N-95
BATERIA	BL-5F 950 mAh 3,7 V

ACCESS POINT	
DESCRIPCIÓ	Access Point Cisco Aironet
MODEL	1131 AG
SISTEMA OPERATIU	IOS 12.4(10b)JA
CERTIFICACIONS	802.11 A/G/H/D/I WMM WPA/WPA2
Agilent N6705A	
DESCRIPCIÓ	Analitzador de potència Agilent
MODEL	N6705A
MÒDUL DE POTÈNCIA 1	N6761A 0-50V 0-1,5A 50W
MÒDUL DE POTÈNCIA 2	N6774A 0-35V 0-8,5A 300W
Home Agent	
CPU	Intel ® Pentium ® 4 CPU 3.00 GHz
Targeta ETH0	Realtek Semiconductor Co., Ltd. RT8139
Versió del SO	Ubuntu Linux 7.10 kernel 2.6.22
Driver ETH0	Realtek Semiconductor Co., Ltd. RTL-8139/8139C/8139C+ (rev 10)
Correspondent Node	
CPU	Intel ® Pentium ® 4 CPU 1.60 GHz
Targeta ETH0	Broadcom Corporation: device 8401
Versió del SO	Fedora Core 4 kernel 2.6.15
Driver ETH0	Broadcom Corporation BCM4401 100Base-T (rev 01)
Paging Agent	
CPU	Intel ® Pentium ® 4 CPU 1.80 GHz
Targeta ETH0	3Com Corporation 3C905C-TX Fast Etherlink for PC Management NIC
Targeta ETH1	3Com Corporation 3C905C-TX Fast Etherlink for PC Management NIC
Targeta ETH2	3Com Corporation 3C905C-TX Fast Etherlink for PC Management NIC
Targeta ETH3	3Com Corporation 3C905C-TX Fast Etherlink for PC Management NIC
Versió del SO	Fedora Core 4 kernel 2.6.15
Driver ETH0	3Com Corporation 3c905C-TX/TX-M [Tornado] (rev 78)
Driver ETH1	3Com Corporation 3c905C-TX/TX-M [Tornado] (rev 78)
Driver ETH2	3Com Corporation 3c905C-TX/TX-M [Tornado] (rev 78)
Driver ETH3	3Com Corporation 3c905C-TX/TX-M [Tornado] (rev 78)



<b>AR1</b>	
CPU	Intel® Pentium® 4 CPU 3.00 GHz
Targeta ETH0	D-Link System Inc DFE-538TX 10/100 Ethernet Adapter
Targeta ETH1	ASUSTeK Computer Inc.: device 811d
Versió del SO	Fedora Core 4 kernel 2.6.15
Driver ETH0	D-Link System Inc RTL8139 Ethernet (rev 10)
Driver ETH1	Intel Corporation 82541GI/PI Gigabit Ethernet Controller (rev 05)
<b>AR2</b>	
CPU	Intel® Pentium® 4 CPU 3.00 GHz
Targeta ETH0	D-Link System Inc DFE-538TX 10/100 Ethernet Adapter
Targeta ETH2	ASUSTeK Computer Inc.: device 811d
Versió del SO	Fedora Core 4 kernel 2.6.15
Driver ETH0	D-Link System Inc RTL8139 Ethernet (rev 10)
Driver ETH2	Intel Corporation 82541GI/PI Gigabit Ethernet Controller (rev 05)

L'Annex II inclou una relació de les diverses targetes certificades com a WMM-PS que s'han estudiat per a determinar el maquinari a utilitzar per a l'estudi, així com les justificacions que han portat a seleccionar el maquinari utilitzat.

**Taula 4.2.** Programari utilitzat a l'escenari de proves

Nom	Sistema operatiu	Versió	Funció	Web de descàrrega
Wireshark	Ubuntu 7.10	1.0.7	Captura de trànsit a la xarxa	<a href="http://www.wireshark.org/">http://www.wireshark.org/</a>
Kismet	Ubuntu 7.10	Kismet-2008-05-R1	Captura de trànsit a la xarxa	<a href="http://www.kismetwireless.net/">http://www.kismetwireless.net/</a>
Iperf	Ubuntu 7.10	1.7.0	Simulació de trànsit VoIP	<a href="http://www.noc.ucf.edu/Tools/iperf/">http://www.noc.ucf.edu/Tools/iperf/</a>
Simple Network Tester	Windows Vista SP1	3.0	Simulació de trànsit VoIP	
Cron	Ubuntu 7.10		Administrador regular de processos	Dimoni integrat al kernel
Filezilla	Windows Vista SP1	3.2.3.1	Client FTP	<a href="http://filezilla-project.org/download.php">http://filezilla-project.org/download.php</a>

Nom	Sistema operatiu	Versió	Funció	Web de descàrrega
Vsftpd	Ubuntu 7.10		Servidor FTP	Repositoris Ubuntu

La configuració del maquinari i el programari s'explica amb detall als Annexos V i VI, respectivament.

## 4.2. Metodologia emprada

Per a mesurar la potència consumida pels dispositius estudiats, s'ha seguit una metodologia que comprèn la connexió dels dispositius, les condicions de l'escenari pel que fa a la topologia, la configuració del maquinari i el seu estat durant les proves, i la durada i repetició d'aquestes proves.

### 4.2.1. Connexió dels dispositius

Per al desplegament de l'escenari es parteix de la topologia que es mostra a la figura 4.1, adaptant-la a cada prova en concret tal com es descriu més endavant per a cadascuna d'elles.

Els dispositius connectats a l'analitzador de potència Agilent N6705A segueixen el procés de configuració que es descriu a l'Annex IV.

Per a connectar les STA a l'analitzador de potència ha calgut confeccionar un cable adaptador. La figura 4.2 mostra el cable emprat per a connectar l'STA 1 a l'analitzador de potència:



**Fig. 4.2.** Cable de connexió STA – analitzador multipunta

## 4.2.2. Configuració dels paràmetres dels dispositius

Per a dur a terme les proves ha calgut configurar una sèrie de paràmetres addicionals en els dispositius (a l'Annex V se'n detalla la configuració).

Addicionalment, s'enumeren certs paràmetres que afegeixen informació a l'escenari.

### 4.2.2.1. Access Point (AP)

Les proves s'han dut a terme amb la configuració express per defecte, tret dels paràmetres següents:

Beacon Period	100ms
Tipus de preàmbul	Curt
Encriptació	No
Velocitat de transmissió de beacon:	1Mbps
Interfície ràdio	802.11g
Paràmetres EDCA	Habilitat AC_VO

Addicionalment, en els escenaris en què s'utilitza PSM ha calgut modificar:

DTIM	2
Listen Interval	200ms

En els escenaris amb U-APSD habilitat, ha calgut configurar QoS tal com s'indica a l'Annex V.

### 4.2.2.2. Equips portàtils (STA 1, STA 2, STA 3)

Les proves s'han dut a terme deshabilitant la targeta de xarxa integrada a la placa de l'STA 1.

En cap cas no s'ha modificat la intensitat de la brillantor de la pantalla, sinó que s'ha deixat al màxim.

Per a la realització de les proves s'ha extret la bateria de l'STA 1, ja que s'han volgut obtenir les dades de quan no està carregant.

Per a les proves de PSM la targeta de l'STA 1 s'ha configurat amb els paràmetres següents:

Listen Interval	200ms
-----------------	-------

#### **4.2.2.3. Telèfon mòbil (STA 4)**

En cap cas no s'ha modificat la intensitat de la brillantor de la pantalla, sinó que s'ha deixat al màxim.

Per a la realització de les proves s'ha extret la bateria de l'STA 4, ja que s'han volgut obtenir les dades de quan no està carregant.

#### **4.2.3. Estat dels dispositius durant les proves**

És important destacar que durant la realització de les proves els diversos escenaris han requerit estats diferents en els dispositius. Així, per exemple, s'han pres mesures amb l'STA 1 i l'STA 4 com a referència amb la interfície WLAN desactivada, per tal de comparar aquestes dades amb les mesures posteriors.

Per cada prova es detalla la informació referent als estats relatius a l'escenari de prova corresponent.

#### **4.2.4. Tipificació de les proves**

Per tal d'obtenir un cert nivell de precisió en les mesures s'ha buscat un compromís de temps per a dur-les a terme, així com un nombre determinat de repeticions de les mostres per a ponderar les dades.

- Temps de captura: 10 min.
- Repetició de les proves: tres preses per escenari.
- Hora de captura: s'ha intentat prendre les mostres que requereixen trànsit en hores d'activitat màxima de la xarxa (en general, entre les 16.00 i les 20.00 h).

S'ha decidit utilitzar 10 minuts com a temps de captura, ja que s'ha comprovat empíricament que períodes de temps superiors no desvien el resultat. Això es deu al fet que l'alimentació dels dispositius a mesurar es duu a terme mitjançant l'analitzador de potència, que subministra un senyal d'alimentació

contínua. És a dir, que el fet de no treballar amb bateria fa que no es pateixi l'efecte de la descàrrega en les mesures.

### 4.3. Relació entre consum i durada de la bateria

El resultat de les proves empíriques sobre el temps de captura permet obtenir la potència mitjana del dispositiu.

Per a determinar el temps de durada de la bateria s'ha fet un càlcul senzill que ha permès extrapolar posteriorment els resultats a altres dispositius mòbils.

Per a poder extrapolar les dades només cal conèixer el consum de les targetes de xarxa IEEE 802.11 del dispositiu desconegut i les característiques de la seva bateria.

L'energia de la bateria s'obté de l'operació resultant de multiplicar-ne la capacitat pel voltatge:

- Energia de la bateria = 4,8 Ah x 11,1 V = 53,28 Wh

El temps restant de bateria s'obté de l'operació resultant del quocient de l'energia de la bateria i la potència mitjana obtinguda de les mesures:

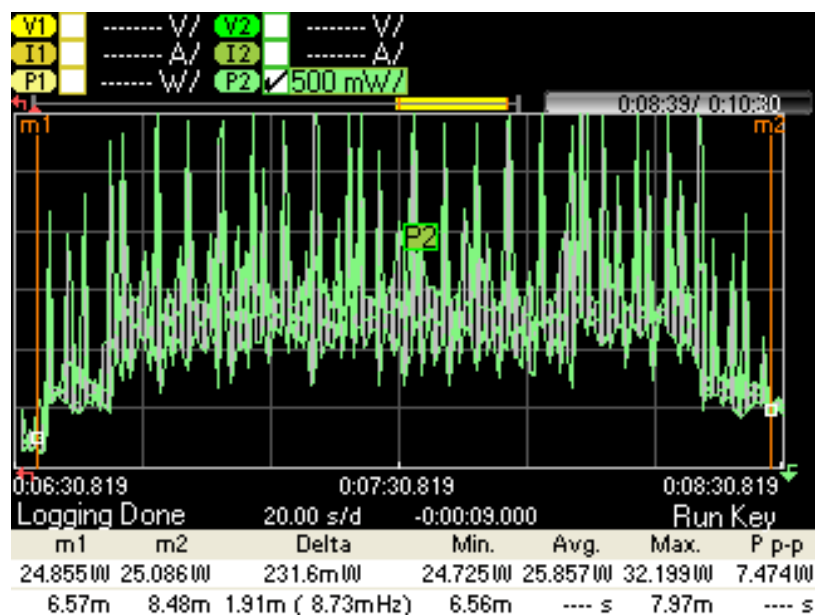
- Temps restant de bateria (TRB) = Energia bateria / potència mitjana

L'increment de bateria assolit a les proves s'ha obtingut mitjançant una senzilla operació. Per exemple, per a conèixer el percentatge obtingut en habilitar el PSM, cal fer el càlcul següent:

- $[(\text{TRB\_PSM} - \text{TRB\_BASE}) \times 100] / \text{TRB\_BASE}$

On TRB\_PSM és el temps de bateria de la mesura PSM i TRB\_BASE és el temps de bateria sense habilitar mecanismes d'estalvi d'energia. En cas de fer el càlcul amb U-APSD, només cal substituir TRB\_PSM per TRB\_UAPSD.

La figura 4.3 mostra les dades que proporciona l'analitzador de potència connectat a una estació (STA 1).



**Figura 4.3.** Exemple de captura amb l'analitzador de potència Agilent N6705A

A l'Annex IV es detalla la informació continguda a les captures de l'analitzador de potència, així com altres dades que s'han obtingut amb aquesta potent eina.

#### 4.4. Proves i resultats

A continuació es mostren els resultats de les proves realitzades als diversos escenaris.

L'objectiu és obtenir el màxim d'informació en relació amb l'efecte d'habilitar els diferents mètodes d'estalvi d'energia, PSM o U-APSD, en els dispositius de xarxa que ho permeten.

Per a cada escenari es detalla la topologia, els elements que hi intervenen i altres dades rellevants com l'existència de trànsit durant el temps de captura, l'AP al qual està vinculada l'STA i la xarxa a què dona accés l'AP.

La informació que s'ha recollit mitjançant el PWA és la potència mitjana.

A la taula d'informació de cada escenari es mostra el resultat de fer la mitjana de les tres captures d'aquestes dades. És a dir, que la potència mitjana mostrada a la taula és la mitjana de la potència mitjana dels resultats empírics obtinguts en tres captures de 10 minuts de durada cadascuna.

A la taula 4.3 es detalla la llegenda informativa de les taules.

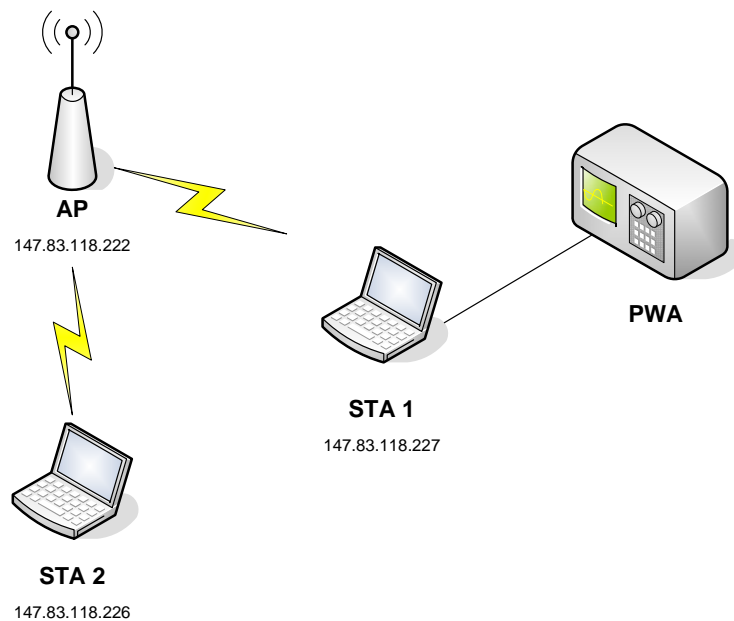
**Taula 4.3.** Taula de mostra

Access Point	Access Point al qual s'associa l'STA
Estacions	Estacions que intervenen
Xarxa	Xarxa a la qual es connecta l'AP
Trànsit	Existència de trànsit a la xarxa
PSM habilitat	Indica l'activació de PSM
U-APSD habilitat	Indica l'activació d'U-APSD
Potència mitjana	Mitjana de la potència mitjana mesurada
Temps de vida de la bateria	Temps de bateria en hores
Increment de bateria	Increment sobre la mesura base de bateria

#### 4.4.1. Impacte de l'activació de la interfície WLAN

##### 4.4.1.1. Topologia

La figura 4.4 mostra un esquema de la topologia i els elements que participen en l'escenari d'aquestes proves.



**Fig. 4.4.** Escenari de proves

#### 4.4.1.2. Pla de proves

S'han dut a terme cinc tandes de proves.

Les proves parteixen d'un escenari en què la targeta de xarxa es troba desactivada i progressivament s'habiliten la targeta i diversos paràmetres.

Les proves són:

- P1: WLAN STA 1 deshabilitada.
- P2: WLAN STA 1 habilitada, sense vincular-se a l'AP i sense connectar l'AP a una xarxa externa.
- P3: WLAN STA 1 habilitada, vinculada a l'AP, sense connectar l'AP a una xarxa externa i sense habilitar mecanismes d'estalvi d'energia.
- P4: WLAN STA 1 habilitada, vinculada a l'AP, sense connectar l'AP a una xarxa externa i amb PSM actiu.
- P5: WLAN STA 1 habilitada, vinculada a l'AP, sense connectar l'AP a una xarxa externa i amb PSM i U-APSD actius.

#### 4.4.1.3. Resultats esperats

De les proves realitzades, s'espera extreure la informació següent:

- Les proves P1, P2 i P3 han de permetre contrastar les mesures posteriors, ja que serviran com a referència. P1 ha de mostrar el consum de l'estació STA 1, mentre que P2 ha de permetre determinar el consum de la targeta de xarxa WLAN i P3 ha de fer referència a la potència mitjana de l'STA 1 en connectar-se a un dels AP Cisco Aironet de l'escenari de proves.
- Les proves P4 i P5 es realitzen amb l'objectiu de demostrar que una vegada establert l'enllaç amb l'AP, sense que aquest estigui connectat a cap xarxa i, per tant, només s'aporti el trànsit entre l'AP i l'STA 1, no hi ha cap diferència de consum entre el fet de tenir habilitat PSM o U-APSD, ja que en cas que no hi hagi trànsit assignat com a prioritari per a U-APSD, la WLAN sempre actua com a PSM.
- El consum d'energia ha de ser inferior a la referència P3 en els casos P4 i P5.



#### 4.4.1.4. Resultats de les proves

##### 4.4.1.4.1. P1: WLAN STA 1 deshabilitada

Com a referència per a les mesures posteriors s'han realitzat una sèrie de captures sense connectar la interfície WLAN, cosa que ha permès contrastar posteriorment les dades capturades.

La taula 4.4 mostra les dades obtingudes d'aquestes mesures base.

**Taula 4.4.** Dades sense associat a l'AP i sense WLAN

Access Point	No associat a cap AP
Estacions	STA 1, STA 2
Xarxa	No aplicable en aquest escenari
Trànsit	No
PSM habilitat	No
U-APSD habilitat	No
Potència mitjana	22,530 W
Temps de vida de la bateria	2,364 hores
Increment de bateria	No aplicable

##### 4.4.1.4.2. P2: WLAN STA 1 habilitada, sense vincular-se a l'AP i sense connectar l'AP a una xarxa externa

Una vegada obtingudes les dades anteriors (vegeu l'apartat 4.4.1.4.1), que serviran com a referència, cal activar la interfície WLAN de l'STA 1 per a poder mesurar el consum de la targeta de xarxa, contrastant-lo amb la mesura anterior. La taula 4.5 recull les dades esmentades.

**Taula 4.5.** Dades sense associat a l'AP amb WLAN habilitat

Access Point	No associat a cap AP
Estacions	STA 1, STA 2
Xarxa	No aplicable en aquest escenari
Trànsit	No
PSM habilitat	No
U-APSD habilitat	No
Potència mitjana	24,548 W
Temps de vida de la bateria	2,170 hores
Increment de bateria	-8,20 %

Les taules següents mostren la informació referent a un escenari sense trànsit, és a dir, sense que l'AP estigui connectat a cap xarxa. S'han pres mesures habilitant els diversos mètodes d'estalvi d'energia per a poder verificar que existeix un consum associat a l'activitat pròpia de l'AP, com ara els beacon, o de l'STA, on es produeixen probe request.

#### 4.4.1.4.3. P3: WLAN STA 1 habilitada, vinculada a l'AP, sense connectar l'AP a una xarxa externa i sense habilitar mecanismes d'estalvi d'energia

La taula 4.6 recull les dades relacionades amb el consum d'energia de l'STA 1 quan aquesta està vinculada a l'AP sense habilitar mecanismes d'estalvi d'energia, tenint en compte que no hi ha trànsit exterior, ja que l'AP no està connectat a cap xarxa externa (tal com es mostra a la figura 4.4).

**Taula 4.6.** Dades de l'STA 1 associada a l'AP sense estalvi d'energia habilitat

Access Point	Cisco Aironet 1131 AG
Estacions	STA 1, STA 2
Xarxa	No aplicable en aquest escenari
Trànsit	No
PSM habilitat	No
U-APSD habilitat	No
Potència mitjana	24,863 W
Temps de vida de la bateria	2,142 hores (TRB_BASE)
Increment de bateria	-1,29 %

#### 4.4.1.4.4. P4: WLAN STA 1 habilitada, vinculada a l'AP, sense connectar l'AP a una xarxa externa i amb PSM actiu

La taula 4.7 recull les dades relacionades amb el consum d'energia de l'STA 1 quan aquesta està vinculada a l'AP amb el mecanisme d'estalvi d'energia PSM habilitat (vegeu el Capítol 1), tenint en compte que no hi ha trànsit exterior, ja que l'AP no està connectat a cap xarxa externa (tal com es mostra a la figura 4.4).

**Taula 4.7.** Dades de l'STA 1 associada a l'AP amb PSM habilitat

Access Point	Cisco Aironet 1131 AG
Estacions	STA 1, STA 2
Xarxa	No aplicable en aquest escenari
Trànsit	No
PSM habilitat	Sí
U-APSD habilitat	No
Potència mitjana	24,762 W
Temps de vida de la bateria	2,151 hores (TRB_PSM)
Increment de bateria	0,42 %

#### 4.4.1.4.5. P5: WLAN STA 1 habilitada, vinculada a l'AP, sense connectar l'AP a una xarxa externa i amb PSM i U-APSD actius

La taula 4.8 recull les dades relacionades amb el consum d'energia de l'STA 1 quan aquesta està vinculada a l'AP amb el mecanisme d'estalvi d'energia U-APSD habilitat (vegeu el Capítol 1), tenint en compte que no hi ha trànsit exterior, ja que l'AP no està connectat a cap xarxa externa (tal com es mostra a la figura 4.4).

Cal recordar que en les situacions en què no hi ha trànsit per a gestionar pel mecanisme EDCA d'U-APSD, aquest es comporta de manera anàloga a PSM.

**Taula 4.8.** Dades de l'STA 1 associada a l'AP amb U-APSD habilitat

Access Point	Cisco Aironet 1131 AG
Estacions	STA 1, STA 2
Xarxa	No aplicable en aquest escenari
Trànsit	No
PSM habilitat	Sí
U-APSD habilitat	Sí
Potència mitjana	24.723 W
Temps de vida de la bateria	2,155 hores (TRB_UAPSD)
Increment de bateria	0,60 %

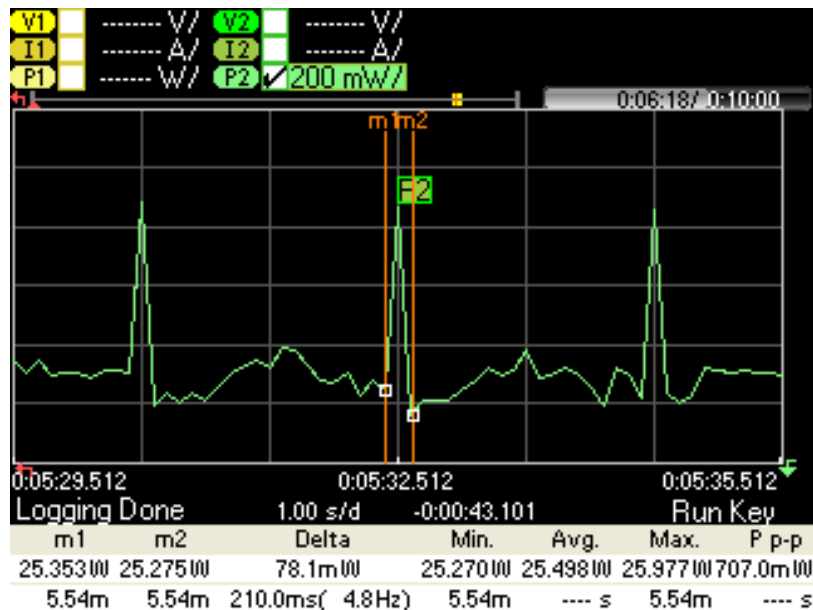
#### 4.4.1.5. Conclusió de les proves

En analitzar els resultats de les proves reflectides a les taules 4.4 (prova P1) i 4.5 (prova P2), s'observa un augment del consum de potència directament relacionat amb la interfície WLAN, el qual es tradueix una disminució del temps restant de bateria.

Aplicant els càlculs esmentats a l'apartat 4.3 es pot deduir l'impacte de la bateria en habilitar la targeta de xarxa, tal com s'aprecia en detall a la taula 4.5.

D'altra banda, aquests resultats demostren que encara que no s'estigui connectat a una xarxa, efectivament existeix un trànsit que prové, per exemple, dels probe request de la targeta WLAN o dels beacon de l'STA.

La figura 4.5 mostra l'activitat relacionada amb un probe request. En aquesta figura s'aprecia en detall la durada del Listening Interval en el cas de PSM, que tal com s'ha configurat és de 200 ms.



**Fig. 4.5.** Paràmetre Listening Interval de PSM

L'impacte del consum en habilitar U-APSD a l'AP no és gaire elevat, ja que com que no funciona cap cua per la manca de trànsit en aquest escenari, els camps pertinents són buits.

Tot i que apareixen els camps WME relacionats amb WMM, l'AP no està gestionant trànsit amb U-APSD.

#### 4.4.2. Efecte del trànsit multicast/broadcast a la xarxa

En els escenaris següents s'ha connectat l'AP Cisco Aironet 1131 AG a diferents xarxes per tal que la targeta WLAN de l'STA 1 en rebí el trànsit de gestió i de dades.

##### 4.4.2.1. Xarxa EPSC

A la figura 4.6 es destaquen les dades aportades per l'opció *Protocol Hierarchy* de l'aplicació Wireshark (vegeu l'Annex VI), capturades per l'estació de monitoratge STA 2 durant 10 minuts. Aquestes dades permeten caracteritzar el trànsit existent a la xarxa.

Mitjançant l'aplicació d'uns senzills filtres a l'aplicació Wireshark és possible diferenciar el trànsit broadcast i multicast. Del nombre total de trames capturades, 633 pertanyen al trànsit multicast (57 trames IGMP), i 1.613 al trànsit broadcast (862 trames ARP i 281 trames IPX). En els 10 minuts emprats per a dur a terme la caracterització s'obtenen:

- Trames multicast: 2,688 trames/segon.
- Trames broadcast: 1,055 trames/segon.

Ethernet	100,00%	4640	568016	0,006	0	0	0,000
Internet Protocol	40,04%	1858	399330	0,004	0	0	0,000
User Datagram Protocol	27,78%	1289	298614	0,003	0	0	0,000
Cisco Hot Standby Router Protocol	12,61%	585	36214	0,000	585	36214	0,000
Hypertext Transfer Protocol	7,22%	335	100052	0,001	335	100052	0,001
NetBIOS Datagram Service	2,20%	102	26656	0,000	0	0	0,000
Data	2,46%	114	99998	0,001	114	99998	0,001
Common Unix Printing System (CUPS) Browsing Protocol	0,56%	26	5954	0,000	26	5954	0,000
Simple Network Management Protocol	0,17%	8	939	0,000	8	939	0,000
BACnet Virtual Link Control	0,26%	12	756	0,000	0	0	0,000
Building Automation and Control Network NPDU	0,26%	12	756	0,000	0	0	0,000
NetBIOS Name Service	0,73%	34	3332	0,000	34	3332	0,000
Domain Name Service	0,22%	10	2046	0,000	10	2046	0,000
PacketCable	0,02%	1	930	0,000	1	930	0,000
Bootstrap Protocol	1,34%	62	21737	0,000	62	21737	0,000
Internet Group Management Protocol	1,96%	91	4872	0,000	64	3252	0,000
Protocol Independent Multicast	0,58%	27	1620	0,000	27	1620	0,000
Transmission Control Protocol	8,32%	386	87516	0,001	262	51236	0,001
Protocol Independent Multicast	1,79%	83	7100	0,000	83	7100	0,000
Internet Control Message Protocol	0,19%	9	1228	0,000	9	1228	0,000
Address Resolution Protocol	27,41%	1272	76330	0,001	1272	76330	0,001
Logical-Link Control	23,79%	1104	66240	0,001	0	0	0,000
Internetwork Packet eXchange	6,88%	319	20770	0,000	0	0	0,000
Service Advertisement Protocol	6,27%	291	18810	0,000	291	18810	0,000
IPX Routing Information Protocol	0,60%	28	1960	0,000	28	1960	0,000
Configuration Test Protocol (loopback)	1,75%	81	4860	0,000	0	0	0,000
Internet Protocol Version 6	0,09%	4	332	0,000	0	0	0,000
User Datagram Protocol	0,09%	4	332	0,000	0	0	0,000
Data	0,04%	2	154	0,000	2	154	0,000

Fig. 4.6. Caracterització del trànsit de la xarxa EPSC

#### 4.4.2.1.1. Topologia

La figura 4.7 mostra un esquema de la topologia i els elements que participen en l'escenari d'aquestes proves.

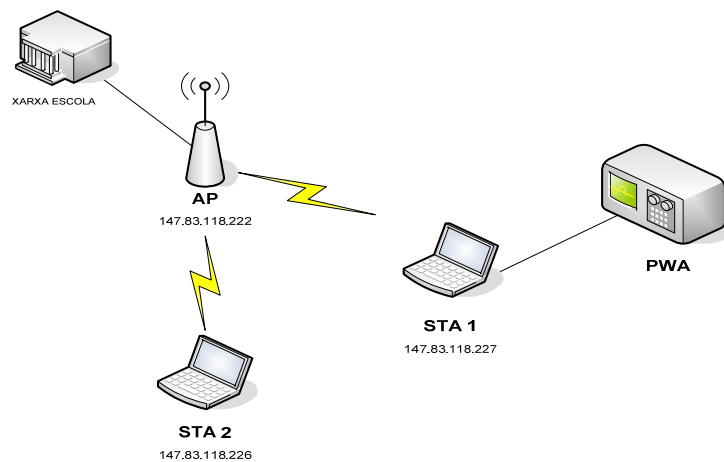


Fig. 4.7. Escenari de proves de la xarxa EPSC

#### 4.4.2.1.2. Pla de proves

S'han dut a terme tres tandes de proves.

Les proves parteixen d'un escenari en què l'STA 1 es troba amb els mecanismes d'estalvi d'energia deshabilitats. L'STA 1 es troba associada en tot moment a un dels AP Cisco Aironet 1131 AG, el qual s'ha connectat a la xarxa de l'Escola.

Les proves són:

- P1: WLAN STA 1 habilitada sense mecanismes d'estalvi d'energia.
- P2: WLAN STA 1 habilitada amb PSM activat.
- P3: WLAN STA 1 habilitada amb U-APSD activat.

#### 4.4.2.1.3. Resultats esperats

De les proves realitzades, s'espera extreure la informació següent:

- L'habilitació de PSM ha de mostrar una reducció del consum de l'STA 1.
- L'habilitació d'U-APSD no ha de mostrar gaire diferència en la reducció de consum respecte de PSM, ja que només hi ha trànsit multicast o broadcast que no és tractat per aquest mecanisme.

#### 4.4.2.1.4. Resultats de les proves

##### 4.4.2.1.4.1. P1: WLAN STA 1 habilitada sense mecanismes d'estalvi d'energia

La taula 4.9 recull les dades de consum de l'STA 1 en un entorn sense trànsit, amb l'AP connectat a la xarxa de l'Escola (EPSC).

**Taula 4.9.** Dades de l'STA 1 associada a l'AP amb xarxa EPSC, sense mecanismes d'estalvi d'energia habilitats

Access Point	Cisco Aironet 1131 AG
Estacions	STA 1, STA 2
Xarxa	EPSC
Trànsit	No
PSM habilitat	No
U-APSD habilitat	No
Potència mitjana	24.977 W
Temps de vida de la bateria	2,133 hores (TRB_BASE)
Increment de bateria	No aplicable

#### 4.4.2.1.4.2. P2: WLAN STA 1 habilitada amb PSM activat

La taula 4.10 recull les dades de consum de l'STA 1 en un entorn sense trànsit, amb l'AP connectat a la xarxa de l'Escola (EPSC).

**Taula 4.10.** Dades de l'STA 1 associada a l'AP amb xarxa EPSC, amb el mecanisme d'estalvi d'energia PSM habilitat

Access Point	Cisco Aironet 1131 AG
Estacions	STA 1, STA 2
Xarxa	EPSC
Trànsit	No
PSM habilitat	Sí
U-APSD habilitat	No
Potència mitjana	24.824 W
Temps de vida de la bateria	2,146 hores (TRB_PSM)
Increment de bateria	0,60 %

#### 4.4.2.1.4.3. P3: WLAN STA 1 habilitada amb U-APSD activat

La taula 4.11 recull les dades de consum de l'STA 1 en un entorn sense trànsit, amb l'AP connectat a la xarxa de l'Escola (EPSC).

**Taula 4.11.** Dades de l'STA 1 associada a l'AP amb xarxa EPSC, amb el mecanisme d'estalvi d'energia U-APSD habilitat

Access Point	Cisco Aironet 1131 AG
Estacions	STA 1, STA 2
Xarxa	EPSC
Trànsit	No
PSM habilitat	Sí
U-APSD habilitat	Sí
Potència mitjana	24.826 W
Temps de vida de la bateria	2,145 hores (TRB_UAPSD)
Increment de bateria	0,56 %

#### 4.4.2.1.5. Conclusió de les proves

Una vegada realitzades les proves i en comparar-ne els resultats amb els que s'han obtingut a l'apartat 4.4.1.4, s'observa que aquests varien en connectar l'AP a la xarxa de l'escola. Així, el trànsit que circula per la xarxa multicast i

broadcast, que arriba a l'AP i a través d'ell a l'STA 1, genera un augment de consum.

En comparació amb aquests valors s'observa el següent augment de consum en relació amb el cas sense trànsit d'una xarxa connectada a l'AP:

- Sense mecanismes d'estalvi: 0,42 %
- Amb PSM: 0,27 %
- Amb U-APSD: 0,46 %

Si només es té en compte el consum d'aquest apartat, es pot observar que l'habilitació dels mecanismes d'estalvi PSM i U-APSD queda reflectida per un lleuger increment de la bateria (d'un 0,60 i un 0,65 %, respectivament).

Els resultats d'U-APSD no són apreciables perquè no s'està generant un trànsit específic relacionat amb veu, que és la disciplina de cua EDCA habilitada a les proves, per la qual cosa s'està aplicant PSM al trànsit rebut.

La raó que justificaria aquest comportament és, tal com s'apunta a [19], que quan no hi ha trànsit mapat a una de les AC d'EDCA, el trànsit es tracta com a PSM.

#### **4.4.2.2. Xarxa EPSC amb ARP-CACHE activat a l'AP Cisco Aironet 1131 AG**

##### **4.4.2.2.1. Topologia**

La topologia d'aquest escenari és idèntica a la que es mostra a la figura 4.7 de l'apartat anterior. La diferència introduïda és l'activació de l'opció ARP-CACHE (*Address Resolution Protocol Cache*) a l'AP, que redueix el trànsit ARP a l'STA 1.

##### **4.4.2.2.2. Pla de proves**

Es realitzen tres tandes de proves per tal de contrastar la informació amb els resultats de l'apartat anterior.

Les proves es fan connectant l'AP a la xarxa EPSC, ja que no ha estat possible accedir a la configuració dels AP de l'Escola (xarxa Eduroam), perquè els gestiona una empresa privada.

Les proves parteixen d'un escenari en què l'STA 1 es troba amb els mecanismes d'estalvi d'energia deshabilitats. L'STA 1 es troba associada en tot moment a un dels AP Cisco Aironet 1131 AG, el qual s'ha connectat a la xarxa de l'Escola. A l'AP s'habilita l'opció ARP-CACHE.



Durant les proves s'envien, des d'una altra estació connectada a la VLAN 118 de l'EPSC, pings a l'STA 1, per tal que hi hagi trànsit ARP dirigit cap aquesta estació.

Les proves són:

- P1: WLAN STA 1 habilitada sense mecanismes d'estalvi d'energia.
- P2: WLAN STA 1 habilitada amb PSM activat.
- P3: WLAN STA 1 habilitada amb U-APSD activat.

#### 4.4.2.2.3. Resultats esperats

Amb aquestes proves es pretén demostrar que el trànsit ARP suposa un impacte en el consum energètic de l'STA 1. Per tant, si s'habilita l'opció ARP-CACHE als AP (vegeu l'Annex V), en gestionar el trànsit ARP a l'AP i si els paquets no arriben a l'STA 1, es produeix un augment de la vida de la bateria proporcional al trànsit eliminat.

A la figura 4.6 es destaca el trànsit ARP capturat.

#### 4.4.2.2.4. Resultats de les proves

##### 4.4.2.2.4.1. P1: WLAN STA 1 habilitada sense mecanismes d'estalvi d'energia

La taula 4.13 recull les dades de consum de l'STA 1 en un entorn sense trànsit, amb l'AP connectat a la xarxa de l'Escola (EPSC).

**Taula 4.13.** Dades de l'STA 1 associada a l'AP amb ARP-CACHE, xarxa EPSC

Access Point	Cisco Aironet 1131 AG
Estacions	STA 1, STA 2
Xarxa	EPSC
Trànsit	No
PSM habilitat	Sí
U-APSD habilitat	Sí
Potència mitjana	24.826 W
Temps de vida de la bateria	2,145 hores (TRB_BASE)
Increment de bateria	0,56 %

#### 4.4.2.2.4.2. P2: WLAN STA 1 habilitada amb PSM activat

La taula 4.14 recull les dades de consum de l'STA 1 en un entorn sense trànsit, amb l'AP connectat a la xarxa de l'Escola (EPSC), sense trànsit ARP i amb el mecanisme d'estalvi d'energia PSM activat.

**Taula 4.14.** Dades de l'STA 1 associada a l'AP amb ARP-CACHE, xarxa EPSC, amb PSM habilitat

Access Point	Cisco Aironet 1131 AG
Estacions	STA 1, STA 2
Xarxa	EPSC
Trànsit	Sí
PSM habilitat	Sí
U-APSD habilitat	No
Potència mitjana	24.689 W
Temps de vida de la bateria	2,158 hores (TRB_PSM)
Increment de bateria	0,60 %

#### 4.4.2.2.4.3. P3: WLAN STA 1 habilitada amb U-APSD activat

La taula 4.15 recull les dades de consum de l'STA 1 en un entorn sense trànsit, amb l'AP connectat a la xarxa de l'Escola (EPSC), sense trànsit ARP i amb el mecanisme d'estalvi d'energia U-APSD activat.

**Taula 4.15.** Dades de l'STA 1 associada a l'AP amb ARP-CACHE, xarxa EPSC, amb U-APSD habilitat

Access Point	Cisco Aironet 1131 AG
Estacions	STA 1, STA 2
Xarxa	EPSC
Trànsit	Sí
PSM habilitat	Sí
U-APSD habilitat	Sí
Potència mitjana	24,649 W
Temps de vida de la bateria	2,161 hores (TRB_UAPSD)
Increment de bateria	0,74 %

#### 4.4.2.2.5. Conclusió de les proves

L'anàlisi dels resultats de les proves d'aquest apartat, contrastats amb els resultats de l'apartat 4.4.2.1 (és a dir, la comparació de l'impacte del trànsit

ARP en el rendiment de la bateria de l'STA 1) demostra que en habilitar l'opció ARP-CACHE a l'AP es produeix un augment de la durada de la bateria.

En concret, en relació amb els diferents mecanismes d'estalvi d'energia estudiats en aquest treball i els resultats de l'apartat 4.4.2.1, s'han obtingut els resultats següents, que mesuren l'increment de la durada de la bateria:

- Sense mecanismes d'estalvi: 0,61 %
- Amb PSM: 0,84 %
- Amb U-APSD: 0,72 %

Si es contrasten les dades de la figura 4.6, es pot deduir el consum produït pel trànsit ARP i, d'aquesta manera, calcular l'equivalència de consum:

- Total de trames 4640 → consum 24,826 W
- Trames ARP 1272 → consum 6,805 W
- Trames ARP STA 1 42 → consum STA1 0,224 W
- Consum teòric sense trànsit ARP:  $24,804 \text{ W} - 0,224 \text{ W} = 24,580 \text{ W}$
- Consum real (vegeu la taula 4.15): 24,649 W

Per tant, es pot afirmar que el consum s'acosta bastant a la realitat. Només hi ha una diferència de 69 mW, atribuïbles a la resta de trànsit de la xarxa en el moment de prendre les mesures.

Per últim, de la mateixa manera que a les proves dels apartats anteriors, no hi ha una gran diferència entre l'habilitació de PSM i U-APSD.

#### 4.4.2.3. Xarxa Eduroam

A la figura 4.8 es destaquen les dades aportades per l'opció *Protocol Hierarchy* de l'aplicació Wireshark (vegeu l'Annex VI), capturades per l'estació de monitoratge STA 2 durant 10 minuts. Aquestes dades permeten caracteritzar el trànsit existent a la xarxa de l'Escola.

Tal com es mostra en aquesta figura, en un mateix període de 10 minuts es produeix una diferència de trànsit 7,5 vegades major (de 35.098 a 4.640 trames) en el cas de la xarxa Eduroam. Aquest comportament és degut en part al trànsit de trames de gestió IEEE 802.11 (beacon), ja que hi ha 7.672 trames de gestió IEEE 802.11, 5.238 de les quals són beacons, cosa que en comparació amb el nombre de beacons de l'escenari EPSC (470), mostra una relació d'11 a 1.

IEEE 802.11 Radiotap Capture header	100,00%	35098	4591365	0,051	12	6053	0,000
IEEE 802.11 wireless LAN	98,82%	34684	4542549	0,050	17218	617190	0,007
IEEE 802.11 wireless LAN management frame	21,86%	7672	1121700	0,012	7672	1121700	0,012
Data	1,79%	630	86308	0,001	630	86308	0,001
Logical-Link Control	26,11%	9164	2717351	0,030	0	0	0,000
Internet Protocol	20,45%	7179	2448434	0,027	0	0	0,000
Transmission Control Protocol	12,62%	4431	2013175	0,022	2688	325429	0,004
Data	3,76%	1318	1119266	0,012	1318	1119266	0,012
Hypertext Transfer Protocol	0,97%	341	496249	0,006	329	489334	0,005
MSN Messenger Service	0,03%	10	1965	0,000	10	1965	0,000
Secure Socket Layer	0,16%	55	58800	0,001	55	58800	0,001
BitTorrent	0,05%	17	11229	0,000	13	10119	0,000
Etheric	0,00%	1	109	0,000	1	109	0,000
TCP Encapsulation of IPsec Packets	0,00%	1	128	0,000	0	0	0,000
User Datagram Protocol	7,72%	2709	431157	0,005	0	0	0,000
NetBIOS Name Service	4,36%	1531	208223	0,002	1531	208223	0,002
Data	1,87%	658	86486	0,001	658	86486	0,001
Hypertext Transfer Protocol	0,76%	267	66564	0,001	267	66564	0,001
Bootstrap Protocol	0,23%	79	30672	0,000	79	30672	0,000
Domain Name Service	0,19%	68	15653	0,000	68	15653	0,000
NetBIOS Datagram Service	0,19%	67	18664	0,000	0	0	0,000
SMB (Server Message Block Protocol)	0,19%	67	18664	0,000	0	0	0,000
Network Time Protocol	0,01%	2	270	0,000	2	270	0,000
Teredo IPv6 over UDP tunneling	0,11%	37	4625	0,000	0	0	0,000
Internet Group Management Protocol	0,08%	28	2726	0,000	28	2726	0,000
Internet Control Message Protocol	0,03%	11	1376	0,000	11	1376	0,000
Spanning Tree Protocol	0,60%	210	20790	0,000	210	20790	0,000
Internet Protocol Version 6	2,27%	796	145246	0,002	0	0	0,000
User Datagram Protocol	1,77%	622	122386	0,001	0	0	0,000
Internet Control Message Protocol v6	0,50%	174	22860	0,000	174	22860	0,000
Address Resolution Protocol	2,64%	928	95150	0,001	928	95150	0,001
Cisco Wireless LAN Context Control Protocol	0,12%	43	4465	0,000	43	4465	0,000
Cisco Discovery Protocol	0,02%	8	3266	0,000	8	3266	0,000
Malformed Packet	1,15%	402	42763	0,000	402	42763	0,000

Fig. 4.8. Trànsit sense fils de la xarxa Eduroam

#### 4.4.2.3.1. Topologia

La figura 4.9 mostra un esquema de la topologia i els elements que participen en l'escenari d'aquestes proves.

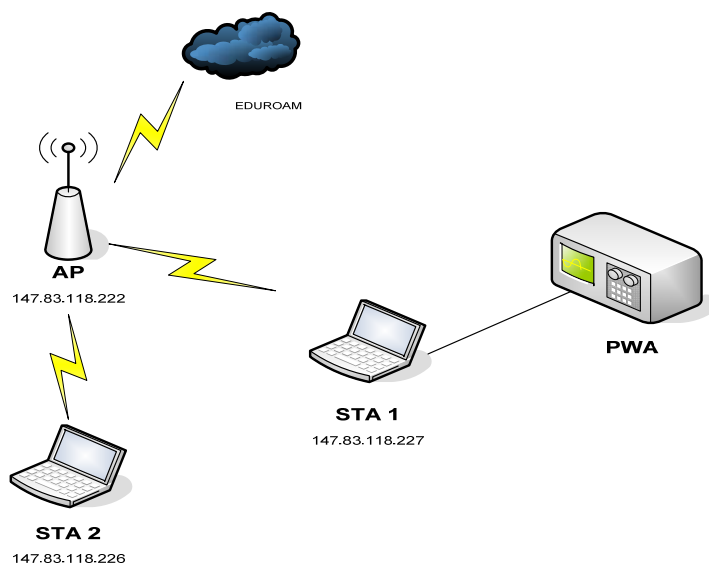


Fig. 4.9. Escenari de proves sota la xarxa Eduroam

#### **4.4.2.3.2. Pla de proves**

Es realitzen tres tandes de proves, amb l'objectiu principal de comprovar l'efecte del trànsit generat en un entorn sense fils extern al laboratori de proves.

Les proves parteixen d'un escenari en què l'STA 1 es troba amb els mecanismes d'estalvi d'energia deshabilitats. L'STA 1 es troba associada en tot moment a un dels AP Cisco Aironet 1131 AG, el qual s'ha connectat a la xarxa sense fils de l'Escola, Eduroam.

Les proves es realitzen al passadís central de l'edifici C4 de l'EPSC, en el mateix marge horari (entre les 16.00 i les 18.00 h) que les captures realitzades per a caracteritzar el laboratori 325.

Les proves són:

- P1: WLAN STA 1 habilitada sense mecanismes d'estalvi d'energia.
- P2: WLAN STA 1 habilitada amb PSM activat.
- P3: WLAN STA 1 habilitada amb U-APSD activat.

#### **4.4.2.3.3. Resultats esperats**

De les proves realitzades, s'espera extreure la informació següent:

- Es preveu un decrement de la bateria, causat pel trànsit multicast i broadcast generat pels equips dels alumnes, en relació amb les proves realitzades al laboratori 325.

#### **4.4.2.3.4. Resultats de les proves**

##### **4.4.2.3.4.1. P1: WLAN STA 1 habilitada sense mecanismes d'estalvi d'energia**

La taula 4.16 recull les dades de consum de l'STA 1 en un entorn amb trànsit generat per les comunicacions entre els equips que treballen a la zona de cobertura, amb l'AP connectat a la xarxa sense fils de l'Escola (Eduroam).

**Taula 4.16.** Dades de l'STA 1 associada a l'AP, xarxa Eduroam, sense mecanismes d'estalvi d'energia habilitats

Access Point	Cisco Aironet 1131 AG
Estacions	STA 1, STA 2
Xarxa	Eduroam
Trànsit	Sí
PSM habilitat	No
U-APSD habilitat	No
Potència mitjana	25,563 W
Temps de vida de la bateria	2,084 hores (TRB_BASE)
Increment de bateria	No aplicable

#### 4.4.2.3.4.2. P2: WLAN STA 1 habilitada amb PSM activat

La taula 4.17 recull les dades de consum de l'STA 1 en un entorn amb trànsit generat per les comunicacions entre els equips que treballen a la zona de cobertura, amb l'AP connectat a la xarxa sense fils de l'Escola (Eduroam), amb el mecanisme d'estalvi d'energia PSM activat.

**Taula 4.17.** Dades de l'STA 1 associada a l'AP, xarxa Eduroam, amb PSM habilitat

Access Point	Cisco Aironet 1131 AG
Estacions	STA 1, STA 2
Xarxa	Eduroam
Trànsit	Sí
PSM habilitat	Sí
U-APSD habilitat	No
Potència mitjana	25,294 W
Temps de vida de la bateria	2,106 hores (TRB_PSM)
Increment de bateria	1,04 %

#### 4.4.2.3.4.3. P3: WLAN STA 1 habilitada amb U-APSD activat

La taula 4.18 recull les dades de consum de l'STA 1 en un entorn amb trànsit generat per les comunicacions entre els equips que treballen a la zona de cobertura, amb l'AP connectat a la xarxa sense fils de l'Escola (Eduroam), amb el mecanisme d'estalvi d'energia U-APSD activat.

**Taula 4.18.** Dades de l'STA 1 associada a l'AP, xarxa Eduroam, amb U-APSD habilitat

Access Point	Cisco Aironet 1131 AG
Estacions	STA 1, STA 2
Xarxa	Eduroam
Trànsit	Sí
PSM habilitat	Sí
U-APSD habilitat	Sí
Potència mitjana	25,248 W
Temps de vida de la bateria	2,110 hores (TRB_UAPSD)
Increment de bateria	1,24 %

#### 4.4.2.3.5. Conclusió de les proves

De les proves es pot deduir que hi ha una relació directa (tal com se suposava) entre el trànsit multicast i broadcast i el major consum energètic que s'ha trobat a les mesures.

L'anàlisi dels resultats de les proves d'aquest apartat, contrastats amb els resultats de l'apartat 4.4.2.1 (és a dir, la comparació de l'impacte del trànsit captat a la xarxa EPSC amb el trànsit sense fils de la xarxa Eduroam), demostra empíricament que hi ha un decrement de la durada de la bateria, relacionat amb el trànsit de gestió IEEE 802.11 (p. ex., trames beacon).

En concret, en relació amb els diferents mecanismes d'estalvi d'energia estudiats en aquest treball i els resultats de l'apartat 4.4.2.1, s'han obtingut els resultats següents, que mesuren el decrement de la durada de la bateria:

- Sense mecanismes d'estalvi: 2,29 %
- Amb PSM: 1,89 %
- Amb U-APSD: 1,67 %

Per últim, s'aprecia una eficàcia major a l'hora d'habilitar els mecanismes d'estalvi d'energia, de manera que es produeixen increments de bateria de fins a un 1,24% (indicació estimada de l'efecte de l'aplicació del mecanisme PSM en un entorn amb una quantitat major de beacon).

És a dir, que l'augment d'activitat es tradueix en un decrement de la bateria, però això queda lleugerament contrarestat per la proporció de trànsit que és gestionat pels mecanismes d'estalvi d'energia.

### 4.4.3. Impacte de l'aplicació de trànsit cap a / des de l'estació de proves

#### 4.4.3.1. Trànsit FTP

La figura 4.10 mostra el trànsit que caracteritza les proves de trànsit realitzades al laboratori 325. Aquesta captura s'ha pres seguint la metodologia ja esmentada d'aquest projecte. El trànsit característic en aquest escenari és unicast, ja que prové de les proves d'FTP.

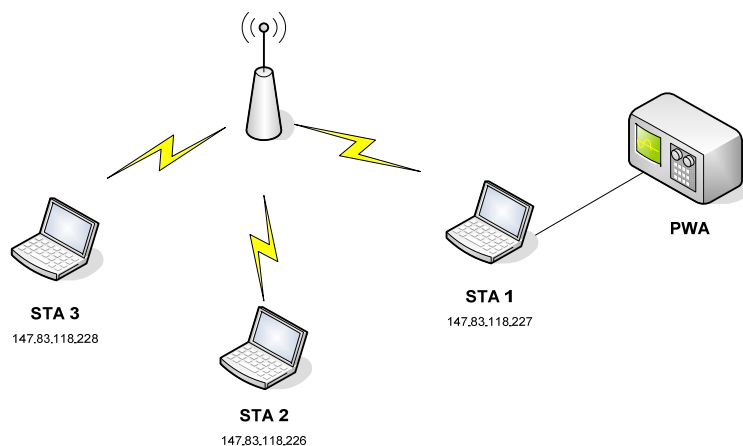
☐ IEEE 802.11 Radiotap Capture header	100,00%	17867	3120368	0,059	0	0	0,000
☐ IEEE 802.11 wireless LAN	100,00%	17867	3120368	0,059	5013	186911	0,004
IEEE 802.11 wireless LAN management frame	64,96%	11607	1771131	0,033	11607	1771131	0,033
☐ Logical-Link Control	6,93%	1238	1160963	0,022	0	0	0,000
☐ Internet Protocol	6,83%	1220	1157644	0,022	0	0	0,000
☐ Transmission Control Protocol	5,78%	1032	1103078	0,021	322	41670	0,001
Data	3,77%	673	1037994	0,020	673	1037994	0,020
Secure Socket Layer	0,08%	15	6705	0,000	15	6705	0,000
MSN Messenger Service	0,01%	1	102	0,000	1	102	0,000
File Transfer Protocol (FTP)	0,02%	4	442	0,000	4	442	0,000
☐ Hypertext Transfer Protocol	0,10%	17	16165	0,000	15	14524	0,000
☐ User Datagram Protocol	0,96%	172	53125	0,001	0	0	0,000
Hypertext Transfer Protocol	0,55%	98	31232	0,001	98	31232	0,001
☐ NetBIOS Datagram Service	0,04%	8	2263	0,000	0	0	0,000
☐ SMB (Server Message Block Protocol)	0,04%	8	2263	0,000	0	0	0,000
Bootstrap Protocol	0,15%	26	13856	0,000	26	13856	0,000
Data	0,06%	10	1145	0,000	10	1145	0,000
NetBIOS Name Service	0,14%	25	3635	0,000	25	3635	0,000
Simple Network Management Protocol	0,02%	3	489	0,000	3	489	0,000
Domain Name Service	0,01%	2	505	0,000	2	505	0,000
Internet Group Management Protocol	0,05%	9	797	0,000	9	797	0,000
Internet Control Message Protocol	0,04%	7	644	0,000	7	644	0,000
Cisco Discovery Protocol	0,01%	2	796	0,000	2	796	0,000
Address Resolution Protocol	0,07%	13	1197	0,000	13	1197	0,000
☐ Internet Protocol Version 6	0,02%	3	1326	0,000	0	0	0,000
☐ User Datagram Protocol	0,01%	1	1084	0,000	0	0	0,000
Internet Control Message Protocol v6	0,01%	2	242	0,000	2	242	0,000
Data	0,05%	9	1363	0,000	9	1363	0,000

**Fig. 4.10.** Caracterització del trànsit a l'escenari FTP



#### 4.4.3.1.1. Topologia

L'escenari de proves FTP consisteix, tal com es mostra a la figura 4.11, en una connexió punt a punt entre l'STA 1 i l'STA 3, monitorada per l'STA 2 i enllaçada per l'AP Cisco Aironet 1131 AG.



**Fig. 4.11.** Escenari de proves FTP

#### 4.4.3.1.2. Pla de proves

S'han realitzat una sèrie de transmissions FTP per a comprovar l'efecte del trànsit real punt a punt sobre l'escenari.

Es realitzen tres tandes de proves, sense habilitar cap mecanisme d'estalvi d'energia, habilitant PSM i U-APSD i transmetent en sentit ascendent i descendent fitxers de dades de 50 MB, 100 MB i 1.000 MB per a cada tanda.

#### 4.4.3.1.3. Resultats esperats

De les proves realitzades, s'espera extreure la informació següent:

- Comprovar la diferència de consum en la transmissió i recepció de dades mitjançant la targeta WLAN. Segons dades del fabricant (vegeu [20]), la targeta WLAN Linksys WUSB600N té un consum de 480 mW en transmissió, i de 300 mW en recepció.
- Així mateix, s'espera observar un consum superior com més gran sigui l'ocupació del canal (és a dir, que els fitxers de 1.000 MB han de presentar el major consum de les proves).

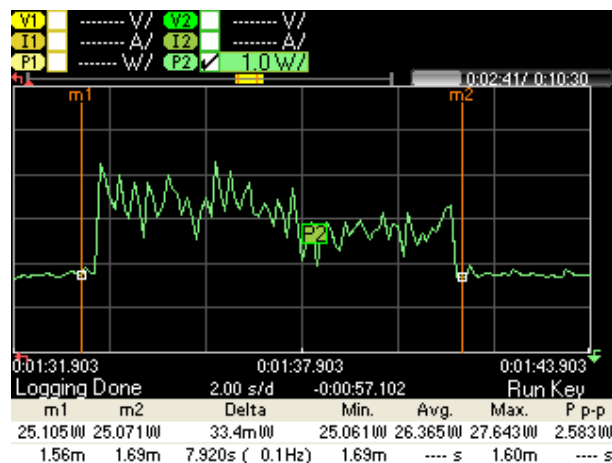
#### 4.4.3.1.4. Resultats de les proves

La taula 4.19 recull els resultats de les proves realitzades en aquest escenari. Tal com es pot observar, els resultats es distribueixen per mecanisme d'estalvi d'energia i per sentit de la transmissió (*ftp up* per al trànsit ascendent i *ftp down* per al descendent).

**Taula 4.19.** Resultats de les mesures de consum en transmissions FTP

TRÀNSIT TCP		Potència (W)	
Fitxer	Mecanisme	ftp down	ftp up
50MB	Cap	25,054	25,253
100MB	Cap	25,186	25,387
1GB	Cap	25,851	26,123
50MB	PSM	24,964	25,222
100MB	PSM	25,147	25,328
1GB	PSM	26,035	26,365
50MB	WMM	25,033	25,266
100MB	WMM	25,757	26,174
1GB	WMM	26,135	26,365

Les figures 4.12 i 4.13 mostren captures corresponents al trànsit de pujada i baixada, respectivament, de la transferència FTP d'1 GB amb el mecanisme de PSM activat.



**Fig. 4.12.** Captura de trànsit de pujada FTP

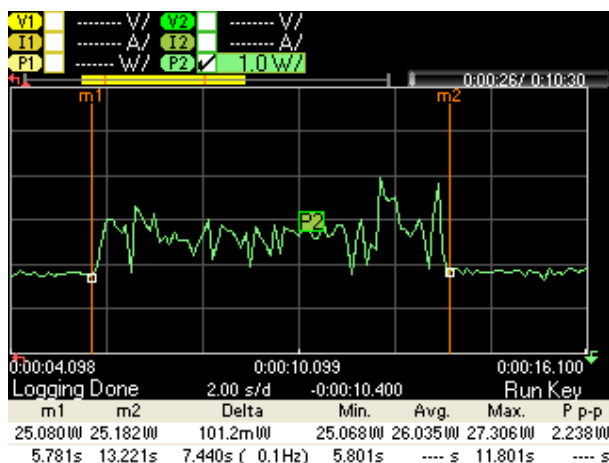


Fig. 4.13. Captura de trànsit de baixada FTP

#### 4.4.3.1.5. Conclusió de les proves

En observar els resultats que es recullen a la taula 4.19 s'arriba a la conclusió que s'han complert totes les hipòtesis plantejades a l'apartat 4.4.3.1.3.

A la taula 4.20 es recull el consum mitjà en sentit ascendent i descendent i, a més, es contrasta amb els valors proporcionats pel fabricant (vegeu [19]). Per a aconseguir-ho, s'ha restat el valor de pujada amb el de baixada (segons el fabricant, uns 180 mW).

**Taula 4.20.** Comparativa de potència d'emissió i recepció de la targeta Linksys WUSB600N

Fitxer	Mecanisme	Diferència de consum WLAN (W)
50MB	Cap	0,199
100MB	Cap	0,201
1GB	Cap	0,272
50MB	PSM	0,258
100MB	PSM	0,181
1GB	PSM	0,330
50MB	WMM	0,233
100MB	WMM	0,417
1GB	WMM	0,23

Del resultat de la taula anterior s'obté una potència mitjana de 256 mW de diferència entre la potència en sentit ascendent i descendent, cosa que demostra empíricament que el valor proporcionat pel fabricant, de 180 mW, no s'acosta als valors reals, sinó que se supera en 76 mW.

Un altre factor a tenir en compte a l'hora de considerar aquesta diferència és que el trànsit mesurat a l'estudi també inclou les trames de control (ACK) TCP. És a dir, que el trànsit mesurat consisteix en les dades i la gestió resultants del fet de confirmar l'arribada dels paquets o d'enviar les ordres i els altres paràmetres de la comunicació FTP. Així, quan es mesura el consum provinent de la descàrrega d'un fitxer, també s'està mesurant part del trànsit ascendent.

No obstant això, si es compara el resultat de la mitjana de la potència en el cas del trànsit amb paquets de 50 MB, la diferència és només de 50 mW. Sembla possible concloure, doncs, que el fabricant ha fet les mesures de consum transmetent i rebent trànsit amb fitxers petits, i sense trànsit addicional com en el cas d'aquestes proves.

#### **4.4.3.2. Trànsit VoIP**

Les proves de trànsit VoIP s'han dut a terme mitjançant el programa Simple Network Tester (vegeu l'Annex VI per a obtenir-ne informació més detallada) per a simular l'enviament de trucades amb el còdec de veu G.711.

L'escenari segueix la mateixa caracterització de trànsit que es mostra a la figura 4.10.

##### **4.4.3.2.1. Topologia**

L'escenari segueix la mateixa topologia que es presenta a la figura 4.11.

##### **4.4.3.2.2. Pla de proves**

S'han dut a terme una sèrie de transmissions UDP per a simular, mitjançant l'aplicació SNT (vegeu l'Annex VI), el trànsit de trucades VoIP i, d'aquesta manera, comprovar l'efecte del trànsit real punt a punt sobre l'escenari.

S'han dut a terme tres tandes de proves sense habilitar cap mecanisme d'estalvi d'energia, habilitant PSM i després U-APSD. S'ha fet la prova i realitzant 1 i 3 trucades i finalment una trucada constant, durant un període de 10 minuts.

S'ha habilitat la prioritat de veu AC\_VO a l'AP i a la tarjeta WLAN quan ha estat necessari. Així mateix, s'ha mapat el trànsit provinent de l'STA 1 cap a l'STA 3 com a CoS 6, per a produir el procés de gestió d'accés al mitjà EDCA.

##### **4.4.3.2.3. Resultats esperats**

De les proves realitzades, s'espera extreure la informació següent:

- S'espera observar un augment del consum com més llarga sigui la durada de la trucada.
- En aquest escenari de proves es preveu una diferència notable entre el consum habilitant PSM i U-APSD, ja que es pretén fer funcionar, en fer injecció de trànsit emulant AC\_VO, el mecanisme EDCA i les funcions d'U-APSD (vegeu el Capítol 1).

#### 4.4.3.2.4. Resultats de les proves

La taula 4.21 mostra els resultats de les tandes de proves en relació amb el temps de les trucades, i també el mecanisme d'estalvi d'energia actiu.

**Taula 4.21.** Resultat de les proves de potència en escenari de VoIP

Nombre de trucades	Mecanisme		
	Cap	PSM	WMM
1 trucada	24,676 W	24,283 W	24,202 W
3 trucades	25,192 W	25,021 W	24,764 W
Constant	25,957 W	25,757 W	25,129 W

#### 4.4.3.2.5. Conclusió de les proves

Dels resultats obtinguts es conclou:

- Tal com s'havia previst, la durada de la trucada va lligada al consum mesurat, de manera que en analitzar la taula 4.21 s'observa un increment d'1,2 W entre el consum associat a transmetre durant tot el temps de proves, 10 minuts, i fer una sola trucada, 2 minuts.
- Com més temps, més trànsit i millor rendiment d'U-APSD. Al contrari que en les proves dels escenaris anteriors, aquí s'observa una diferència important (vegeu la taula 4.22, on s'extrapola la potència de la taula 4.21 a hores de bateria) entre l'increment de bateria en habilitar PSM i U-APSD. És a dir que, tal com es mostra, en el millor cas, amb trànsit constant i U-APSD activat, s'han guanyat 4,09 minuts de bateria (127,21 – 123,12 minuts).

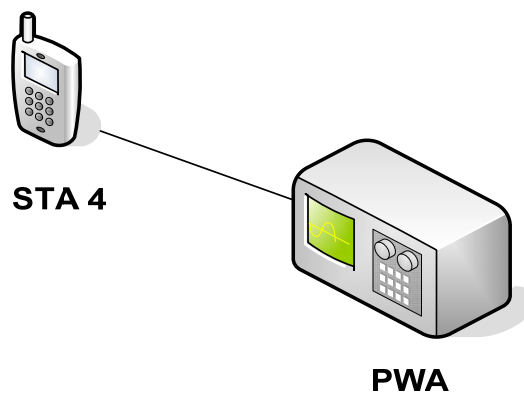
**Taula 4.22.** Extrapolació en temps de bateria de les mesures realitzades

Nombre de trucades	Mecanisme		
	Cap	PSM	WMM
1 trucada	129,54 minuts	131,64 minuts	132,06 minuts
3 trucades	126,84 minuts	127,74 minuts	129,06 minuts
Constant	123,12 minuts	124,08 minuts	127,21 minuts

#### 4.4.4. Avaluació del consum d'un terminal mòbil (STA 4)

##### 4.4.4.1. Topologia

La figura 4.14 mostra la topologia prevista per a aquest escenari.



**Fig. 4.14.** Escenari de proves STA 4

##### 4.4.4.2. Pla de proves

L'objectiu d'aquest escenari és extrapolar les mesures obtingudes del telèfon mòbil Nokia N95, STA 4, a l'escenari 4.4.3.2.

En cada cas s'ha fet una única trucada en un període de mesura de 10 minuts, i s'ha considerat el cas de la interfície WLAN.

##### 4.4.4.3. Resultats esperats

De les proves realitzades, s'espera extreure la informació següent:

- L'objectiu a assolir és identificar el consum teòric de l'estació STA 4 i deduir quina reducció de consum s'obtingria en habilitar U-APSD.

#### 4.4.4.4. Resultats de les proves

La taula 4.23 mostra els resultats de les mesures obtingudes.

**Taula 4.23.** Mesures de potència de l'STA 4

Interfície	Potència (W)
UMTS	0,673
WIFI	0,78
WIFI+AP+Conversa	0,793
GSM	0,665

La durada de la bateria de l'STA 4 amb la interfície WLAN habilitada (extrapolada de les dades proporcionades a l'Annex V) en fer una trucada de 2 minuts és la següent:

$$- (3,7 \text{ V} \times 0,95 \text{ Ah}) / 0,793 \text{ W} = 4,43 \text{ hores}$$

Per tant, en revisar l'increment de temps a la bateria de l'STA 1 quan es fa una trucada de VoIP a la taula 4.22, s'obtenen els següents guanys, comparant els resultats de l'STA 1 i aplicant-los a l'STA 4:

- PSM 1,64 %
- WMM 2,01 %

que extrapolat a l'STA 4 dóna un increment de temps de bateria de:

- PSM: 4,35 minuts.
- WMM: 5,34 minuts.

En comparació, el temps de durada de la bateria en utilitzar la interfície UMTS seria de:

$$- (3,7 \text{ V} \times 0,95 \text{ Ah}) / 0,673 \text{ W} = 5,22 \text{ hores}$$

#### 4.4.4.5. Conclusió de les proves

Arran de les proves realitzades i dels resultats extrapolats, a primer cop d'ull no s'aprecia un gran guany en la durada de la bateria. No obstant això, aquests resultats no són precisos, ja que ha calgut fer moltes aproximacions en no poder comprovar com es comportaria el trànsit generat pel mòbil en un escenari real.

A més, cal tenir en compte que els resultats extrapolats de l'apartat 4.3.2.2 se ceneixen a un escenari i unes característiques molt específics.

#### 4.4.5. Proves de consum sobre el dimoni UMIP modificat

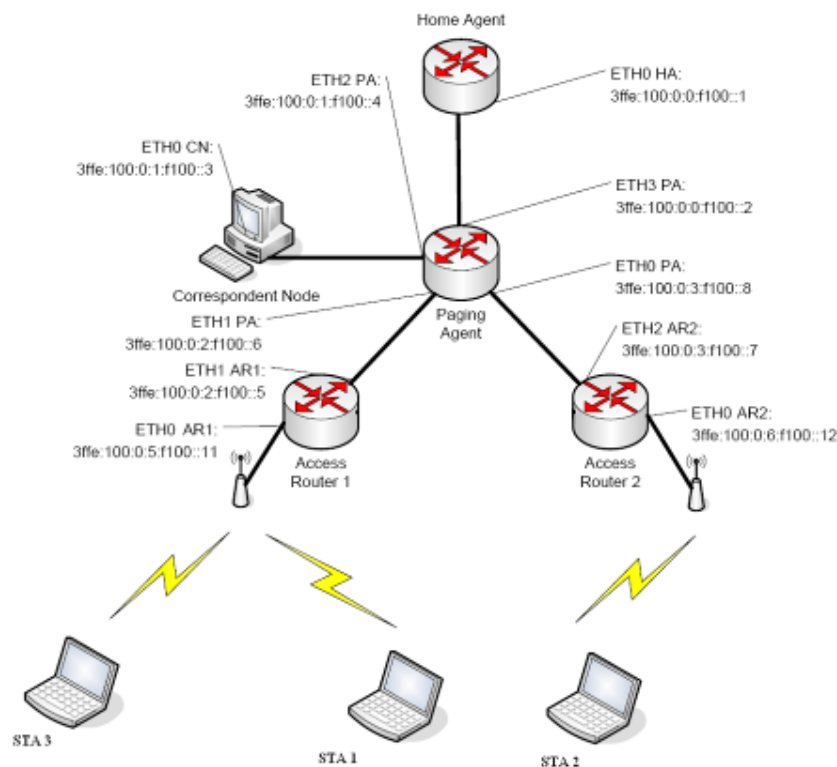
Finalment, s'han realitzat mesures de consum sobre la maqueta MIPv6 sobre una sèrie d'escenaris.

La finalitat d'aquestes proves és demostrar com la supressió de trànsit a l'MN (tant de RtrAdv com del mecanisme de Neighbor Discovery) i l'aplicació d'IP Paging permeten estalviar energia.

A més, es realitzen dues tandes de proves per a verificar com la implementació dels mecanismes d'estalvi d'energia PSM i U-APSD fa més eficient el protocol.

##### 4.4.5.1. Topologia

La figura 4.15 mostra l'escenari general que s'ha utilitzat per a les proves:





**Fig. 4.15.** Escenari general de proves MIPv6

#### 4.4.5.2. Pla de proves

S'han dividit les proves en dues tandes:

Proves sense PSM ni U-APSD activats:

- P1: Captura del consum de l'MN durant 60 segons en l'estat actiu.
- P2: Captura del consum de l'MN durant 60 segons en l'estat dormant.
- P3: Captura del consum de l'MN durant 60 segons en l'estat actiu realitzant un traspàs de xarxa.
- P4: Captura del consum de l'MN durant 60 segons en l'estat dormant realitzant un traspàs de xarxa amb canvi d'àrea de paging.
- P5: Captura del consum de l'MN durant 60 segons en l'estat dormant realitzant un traspàs de xarxa sense canvi d'àrea de paging.

Proves amb PSM i U-APSD activats:

- P6: Captura del consum de l'MN durant 60 segons en l'estat actiu.
- P7: Captura del consum de l'MN durant 60 segons en l'estat dormant.
- P8: Captura del consum de l'MN durant 60 segons en l'estat actiu realitzant un traspàs de xarxa.
- P9: Captura del consum de l'MN durant 60 segons en l'estat dormant realitzant un traspàs de xarxa amb canvi d'àrea de paging.
- P10: Captura del consum de l'MN durant 60 segons en l'estat dormant realitzant un traspàs de xarxa sense canvi d'àrea de paging.

#### 4.4.5.3. Resultats esperats

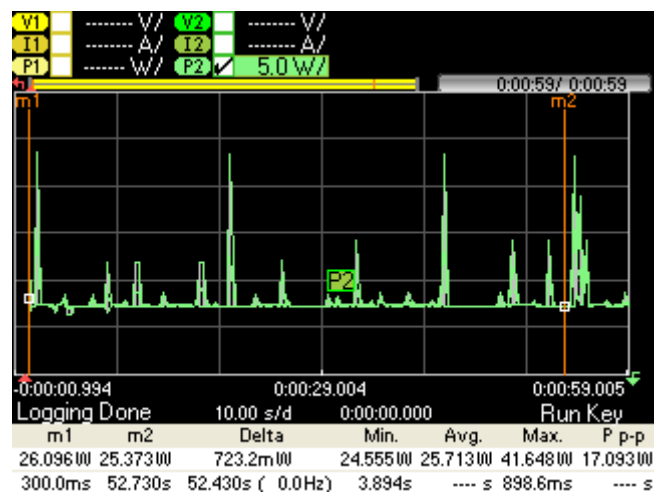
Els resultats previstos per aquestes proves són els següents:

- El consum d'energia ha de mostrar un comportament periòdic quan el node és en l'estat actiu, ja que rep a intervals regulars els RtrAdv del router.
- El consum d'energia hauria de ser tenir una forma plana quan el node es troba en l'estat dormant, ja que no processa RtrAdv.

- El consum d'energia en els escenaris on s'ha eliminat el trànsit de RtrAdv i el mecanisme de Neighbor Discovery (node en dormant) ha de ser inferior quan el node es troba en l'estat actiu.
- El consum d'energia quan s'aplica l'algoritme de IP Paging (canvi de xarxa sense canviar d'àrea de paging) ha de ser inferior a quan no se n'aplica cap (canvi d'àrea de paging).
- El consum d'energia ha de ser inferior a tots els escenaris on s'aplica PSM i U-APSD.

#### 4.4.5.4. Resultat de les proves

##### 4.4.5.4.1. P1: Captura del consum de l'MN durant 60 segons en estat actiu



**Fig. 4.16.** Recepció dels Router Advertisement en estat actiu

A la figura 4.16 es pot apreciar com apareixen periòdicament els pics de recepció dels Router Advertisement. L'interval configurat (entre 3 i 8 segons) per a l'enviament d'aquests paquets des de l'AR1 coincideix amb els valors trobats empíricament.

Si es contrasta el consum quan es produeixen els Router Advertisement amb el consum de l'STA 1 a l'apartat 4.4.1.4.3, s'observa que es produeix un decrement de la durada de la bateria (vegeu la taula 4.24).

**Taula 4.24.** Contrast de consum per recepció de Router Advertisement

Potència (W)	Temps de bateria (hores)
24,863 (vegeu l'apartat 4.4.1.4.3)	2,142
25,713	2,072

Aquestes dades es tradueixen en una reducció de la bateria del 3,26%.

#### 4.4.5.4.2. P2: Captura del consum de l'MN durant 60 segons en estat dormant

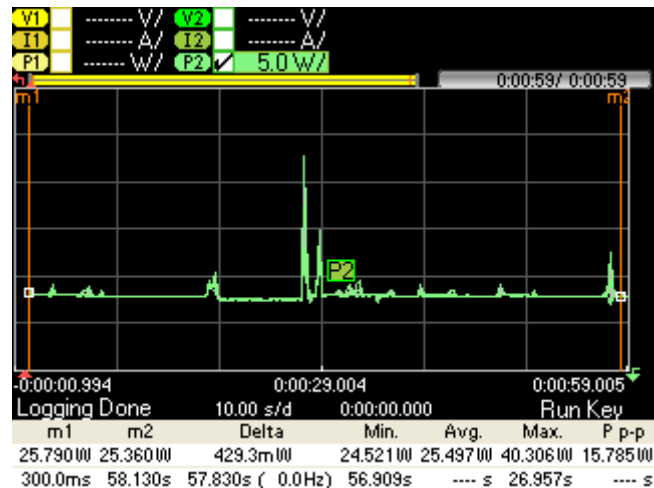


Fig. 4.17. Node en estat dormant

A la figura 4.17 es pot apreciar com el node en dormant elimina els pics provocats pels Router Advertisement. Únicament apareix un pic de consum als 30 segons, que és quan es calcula la xarxa en què es troba l'MN mitjançant l'SSID. Els valors trobats coincideixen amb els esperats.

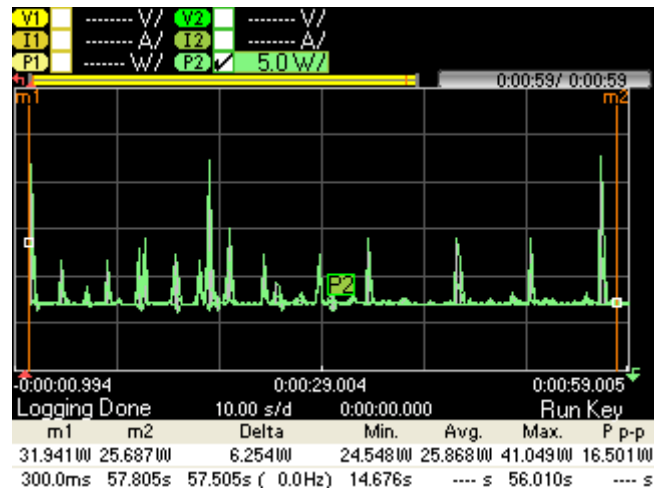
El consum en mitjana és inferior a la prova P1.

En analitzar les dades i comparar-les amb les obtingudes a l'apartat 4.4.5.4.1, es pot deduir el següent mitjançant un senzill càlcul:

**Taula 4.25.** Contract de consum de node en l'estat dormant amb l'estat actiu

Access Point	Cisco Aironet 1131 AG
Estacions	STA 1, STA 2, STA 3
Xarxa	Maqueta MIPv6
Trànsit	Sí
PSM habilitat	No
U-APSD habilitat	No
Potència mitjana	25,497 W
Temps de vida de la bateria	2,089 hores
Increment de bateria	0,82 %

#### 4.4.5.4.3. P3: Captura del consum de l'MN durant 60 segons en estat actiu realitzant un traspàs de xarxa



**Fig. 4.18.** Consum total de l'MN durant un traspàs sense PSM

A la figura 4.18 es mostra com el traspàs de xarxes a l'estat actiu mostra un patró calcat a la P1 (tan sols s'hi afegeix un pic a causa del traspàs entre xarxes).

**Taula 4.26.** Increment de consum per el traspàs entre xarxes amb node actiu

Access Point	Cisco Aironet 1131 AG
Estacions	STA 1, STA 2, STA 3
Xarxa	Maqueta MIPv6
Trànsit	Sí
PSM habilitat	No
U-APSD habilitat	No
Potència mitjana	25,868 W
Temps de vida de la bateria	2,059 hores
Increment de bateria	- 0,62 %

Els valors trobats demostren que el traspàs a una nova xarxa incrementa únicament el consum en un 0,62 % en el cas de mantenir el node en l'estat actiu.

#### 4.4.5.4.4. P4: Captura del consum de l'MN durant 60 segons en estat dormant realitzant un traspàs de xarxa amb canvi d'àrea de paging

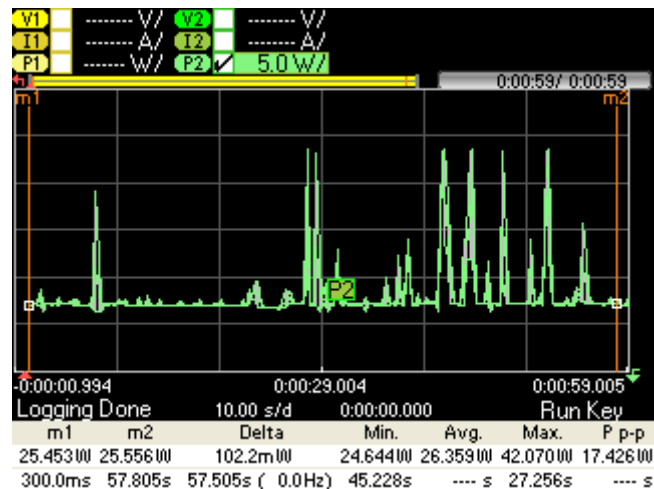


Fig. 4.19. Canvi d'àrea de paging

A la figura 4.19 es comprova que el canvi d'àrea de paging provoca un consum afegit pel procés de traspàs i la nova formació de CoA. Cal recordar que tot i que l'MN forma una nova CoA, l'HA considera que el node segueix en dormant.

El resultat trobat coincideix amb l'esperat.

En analitzar les dades i comparar-les amb les obtingudes a la prova P3, es pot deduir el següent:

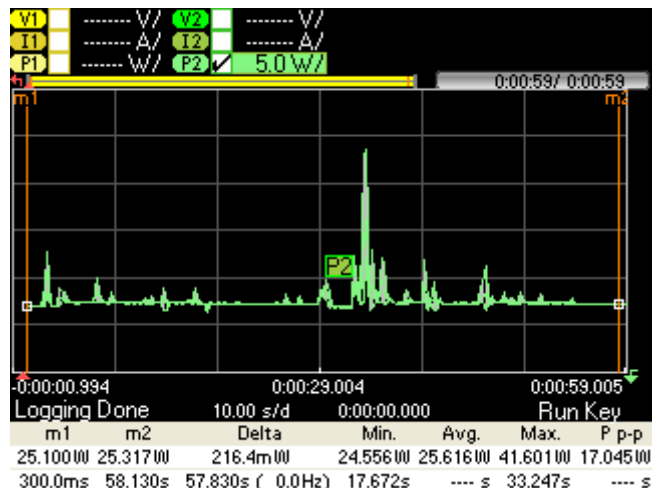
**Taula 4.27.** Contrast de consum per canvi d'àrea de paging entre node actiu i node dormant

Access Point	Cisco Aironet 1131 AG
Estacions	STA 1, STA 2, STA 3
Xarxa	Maqueta MIPv6
Trànsit	Sí
PSM habilitat	No
U-APSD habilitat	No
Potència mitjana	26,359 W
Temps de vida de la bateria	2,021 hores
Increment de bateria	- 1,8 %

El increment de bateria respecte la prova P3 demostra fins a quin punt el càlcul de la distància entre àrees de paging consumeix recursos del sistema.

Com es veu en els següents apartats, el veritable estalvi d'energia apareix quan no es canvia d'àrea de paging i s'apliquen mecanismes d'estalvi d'energia.

#### 4.4.5.4.5. P5: Captura del consum de l'MN durant 60 segons en estat dormant realitzant un traspàs de xarxa sense canvi d'àrea de paging



**Fig. 4.20.** Canvi de xarxa sense canviar d'àrea de paging

Contràriament a la figura anterior (figura 4.19), en aquesta figura es pot comprovar que el canvi de xarxa sense canviar d'àrea de paging fa que el terminal es comporti com un node dormant durant tota la captura i només apareix un pic als 30 segons (temps que triga a fer la lectura de SSID).

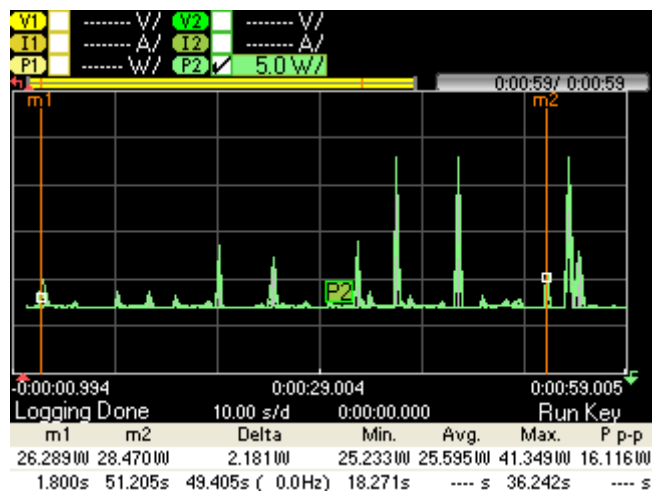
El resultat trobat coincideix amb l'esperat.

En analitzar les dades i comparar-les amb les obtingudes a l'apartat 4.4.5.4.4, es pot deduir el següent mitjançant un senzill càlcul:

**Taula 4.28.** Contract de consum per un canvi de xarxa amb i sense canvi d'àrea de paging

Access Point	Cisco Aironet 1131 AG
Estacions	STA 1, STA 2, STA 3
Xarxa	Maqueta MIPv6
Trànsit	Sí
PSM habilitat	No
U-APSD habilitat	No
Potència mitjana	25,616 W
Temps de vida de la bateria	2,079 hores
Increment de bateria	3,41 %

#### 4.4.5.4.6. P6: Captura del consum de l'MN durant 60 segons en estat actiu



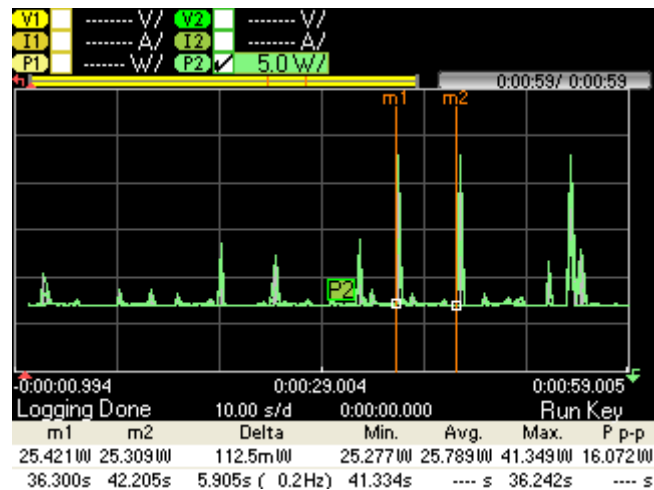
**Fig. 4.21.** Consum de l'MN durant 60 segons

La figura 4.21 és un calc de la figura de la P1. El resultat és l'esperat, ja que consum és menor pel fet de tenir PSM i U-APSD activats.

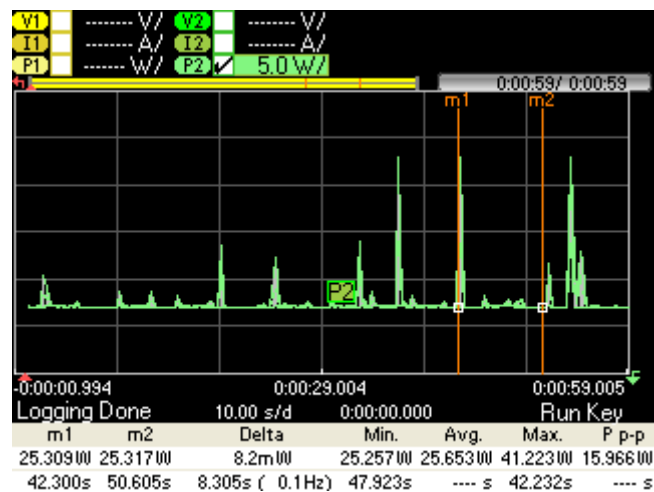
En analitzar les dades i comparar-les amb les obtingudes a l'apartat 4.4.5.4.1, es pot deduir el següent mitjançant un senzill càlcul:

**Taula 4.29.** Contract de consum node actiu amb i sense mecanisme PSM activat

Access Point	Cisco Aironet 1131 AG
Estacions	STA 1, STA 2, STA 3
Xarxa	Maqueta MIPv6
Trànsit	Sí
PSM habilitat	Sí
U-APSD habilitat	Sí
Potència mitjana	25,595 W
Temps de vida de la bateria	2,081 hores
Increment de bateria	0,43 %



**Fig. 4.22.** Interval de 6 segons durant el qual es reben els RtrAdv



**Fig. 4.23.** Interval de 8 segons durant el qual es reben els RtrAdv

Les figures 4.22 i 4.23 mostren l'interval durant el qual es reben els RtrAdv. En aquest cas, 6 i 8 segons.

Aquestes mesures coincideixen amb el debug del dimoni UMIP:

```
Sat Apr 25 19:35:10 md_recv_ra: rtr_addr_new: 3ffe:100:0:5:f100:0:0:11
Sat Apr 25 19:35:15 md_recv_ra: rtr_addr_new: 3ffe:100:0:5:f100:0:0:11
Sat Apr 25 19:35:24 md_recv_ra: rtr_addr_new: 3ffe:100:0:5:f100:0:0:11
```



#### 4.4.5.4.7. P7: Captura del consum de l'MN durant 60 segons en estat dormant

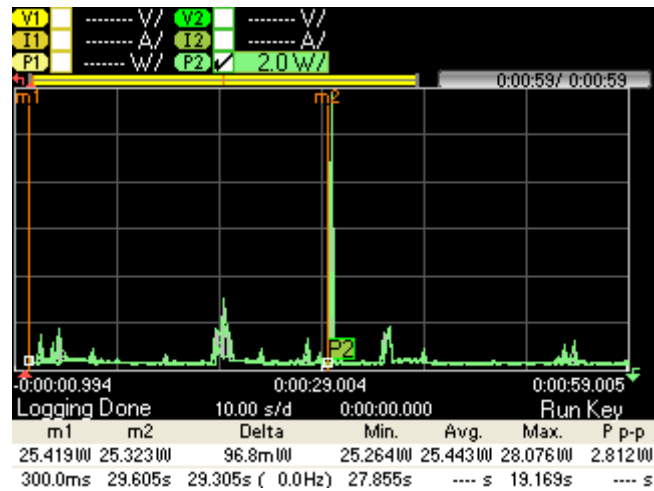


Fig. 4.24. Consum de l'MN durant 30 segons en estat dormant

La figura 4.24 mostra el mateix patró que el trobat a la prova P2. L'únic consum visible apareix als 30 segons pel fet de llegir l'SSID, calcular la xarxa en què es troba i decidir si canvia o no d'àrea de paging.

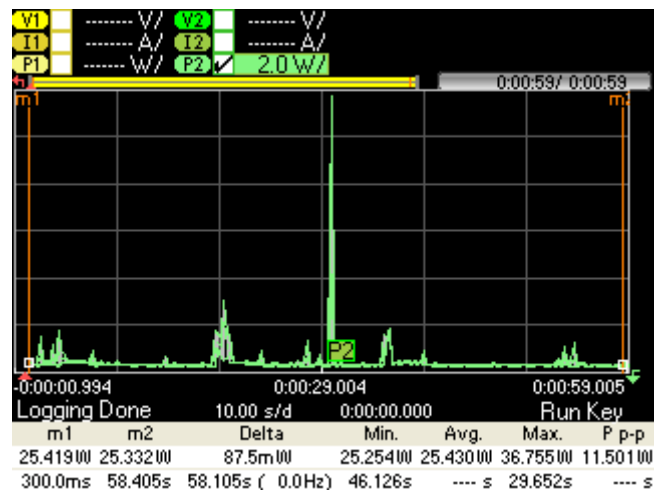


Fig. 4.25. Consum de l'MN durant 60 segons en estat dormant

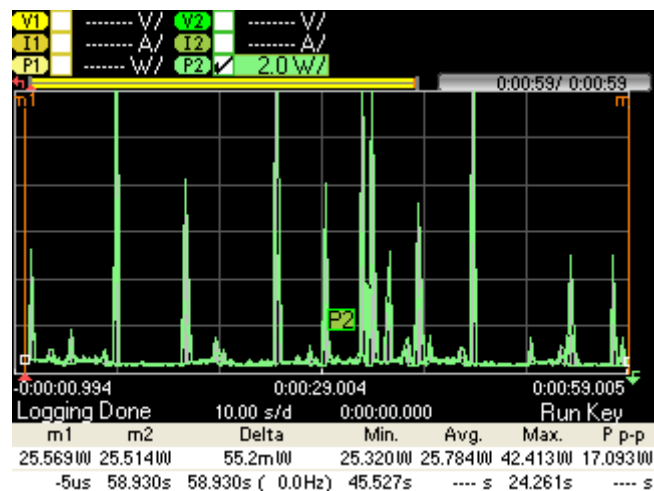
La figura 4.25 mostra el consum total durant els 60 segons de captura. El resultat és l'esperat, ja que el consum és menor pel fet de tenir PSM i U-APSD activats.

En analitzar les dades i comparar-les amb les obtingudes a l'apartat 4.4.5.4.2, es pot deduir el següent mitjançant un senzill càlcul:

**Taula 4.30.** Contract de consum node dormant amb i sense mecanisme PSM activat

Access Point	Cisco Aironet 1131 AG
Estacions	STA 1, STA 2, STA 3
Xarxa	Maqueta MIPv6
Trànsit	Sí
PSM habilitat	Sí
U-APSD habilitat	Sí
Potència mitjana	25,430 W
Temps de vida de la bateria	2,095 hores
Increment de bateria	0,28 %

**4.4.5.4.8. P8: Captura del consum de l'MN durant 60 segons en estat actiu realitzant un traspàs de xarxa.**



**Fig. 4.26.** Traspàs entre xarxes

La figura 4.26 és un calc de la figura de la P3. El resultat és l'esperat, ja que el consum és menor pel fet de tenir PSM i U-APSD activats.

Si es compara el consum amb el resultat de la taula 4,18, s'observa que hi ha un guany de temps de bateria del 0,34%.

#### 4.4.5.4.9. P9: Captura del consum de l'MN durant 60 segons en estat dormant realitzant un traspàs de xarxa amb canvi d'àrea de paging

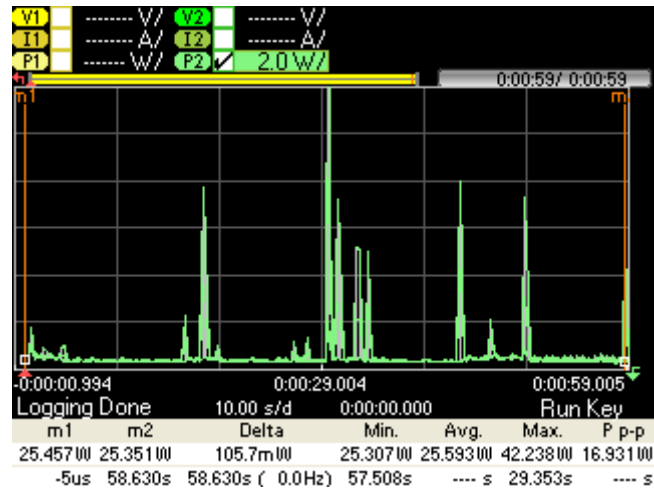


Fig. 4.27. Canvi d'àrea de paging

La figura 4.27 és un calc de la figura de la P4. El resultat és l'esperat, ja que el consum és menor pel fet de tenir PSM i U-APSD activats.

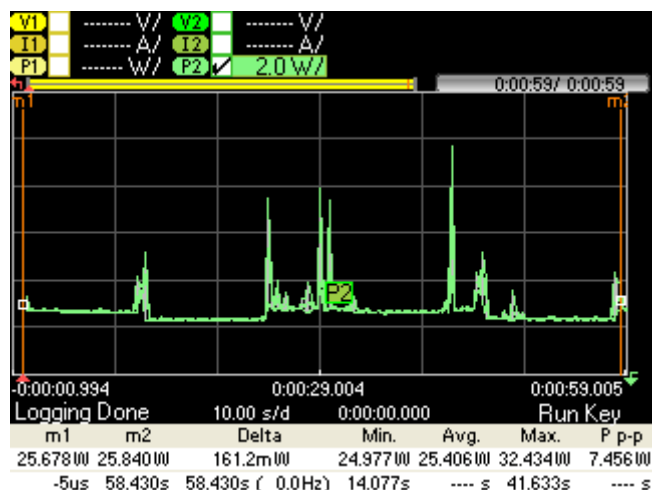
En analitzar les dades i comparar-les amb les obtingudes a l'apartat 4.4.5.4.4, es pot deduir el següent mitjançant un senzill càlcul:

**Taula 4.31** Contract de consum node dormant amb traspàs d'àrea de paging amb i sense mecanisme PSM activat

Access Point	Cisco Aironet 1131 AG
Estacions	STA 1, STA 2, STA 3
Xarxa	Maqueta MIPv6
Trànsit	Sí
PSM habilitat	Sí
U-APSD habilitat	Sí
Potència mitjana	25,593 W
Temps de vida de la bateria	2,081 hores
Increment de bateria	2,96 %

Al contrari que a la prova P4 (on el consum era superior a la prova P3), en aquest cas el consum de bateria és inferior a la prova P8. Aquest fet s'explica per el mecanisme de PSM que aprofita molt més els temps de silenci per estalviar bateria.

#### 4.4.5.4.10. P10: Captura del consum de l'MN durant 60 segons en estat dormant realitzant un traspàs de xarxa sense canvi d'àrea de paging



**Fig. 4.28.** Consum de l'MN dormant amb traspàs, sense canvi d'àrea de paging

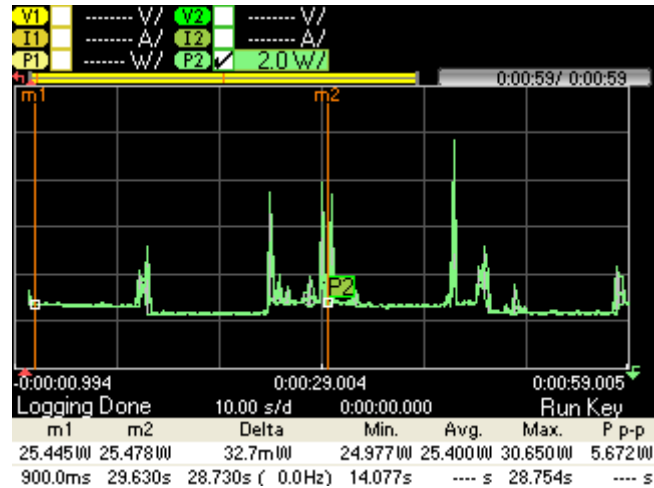
La figura 4.28 és un calc de la figura de la P5. El resultat és l'esperat, ja que el consum és menor pel fet de tenir PSM i U-APSD activats.

En analitzar les dades i comparar-les amb les obtingudes a l'apartat 4.4.5.4.5, es pot deduir el següent mitjançant un senzill càlcul:

**Taula 4.32.** Contract de consum node dormant sense traspàs d'àrea de paging amb i sense mecanisme PSM activat

Access Point	Cisco Aironet 1131 AG
Estacions	STA 1, STA 2, STA 3
Xarxa	Maqueta MIPv6
Trànsit	Sí
PSM habilitat	Sí
U-APSD habilitat	Sí
Potència mitjana	25,406 W
Temps de vida de la bateria	2,097 hores
Increment de bateria	0,86 %

Un altre resultat interessant és comparar els resultats dels capítols 4.4.5.3.9 i 4.4.5.3.10, on hi ha un guany de bateria del 0,78% (un minut més de bateria).



**Fig. 4.29.** Efecte d'un traspàs

La figura 4.29 mostra de nou com apareix un pic de consum als 30 segons pel traspàs de xarxa.

La següent captura del debug d'UMIP demostra com s'ha realitzat aquest canvi de xarxa:

```
Sat Apr 25 20:05:43 ssid_handler: llegint SSID des de 3ffe:100:0:6:f100:0:0:12 on iface 5
Sat Apr 25 20:05:43 ssid_handler: llegint SSID des de fe80:0:0:6:f100:0:0:12 on iface 5
Sat Apr 25 20:05:43 ssid_handler: rtr_addr_new: 3ffe:100:0:6:f100:0:0:12
Sat Apr 25 20:05:43 ssid_handler: SSID router: 3ffe:100:0:6:f100:0:0:12
Sat Apr 25 20:05:43 ssid_handler: Direccions iguals, node segueix en dormant
Sat Apr 25 20:05:43 md_update_router_stats: add coa 3ffe:100:0:6:21e:e5ff:fedf:9dd7 on
interface (5)
Sat Apr 25 20:05:43 md_update_router_stats: rtr_addr_new: 3ffe:100:0:6:f100:0:0:12
SSID llegit fe80:000:0000:0005:f100:0000:0000:0011
Sat Apr 25 20:06:13 ssid_handler: llegint SSID des de 3ffe:100:0:5:f100:0:0:11 on iface 5
Sat Apr 25 20:06:13 ssid_handler: llegint SSID des de fe80:0:0:5:f100:0:0:11 on iface 5
Sat Apr 25 20:06:13 ssid_handler: rtr_addr_new: 3ffe:100:0:6:f100:0:0:12
Sat Apr 25 20:06:13 ssid_handler: SSID router: 3ffe:100:0:5:f100:0:0:11
Sat Apr 25 20:06:13 ssid_handler: Direccions diferents, comprovant distancia de paging
Sat Apr 25 20:06:13 calculate_distance: coord_res = -1,1
Sat Apr 25 20:06:13 ssid_handler: Distance = 2
Sat Apr 25 20:06:13 ssid_handler: DISTANCE_PAG_AREA = 3
Sat Apr 25 20:06:13 ssid_handler: Salt inferior a distancia paging, rtr_addr_new:
3ffe:100:0:5:f100:0:0:11
Sat Apr 25 20:06:13 md_add_backup_router: add new backup router fe80:0:0:5:f100:0:0:11 on
interface ra0 (5)
Sat Apr 25 20:06:14 md_change_default_router: add new router fe80:0:0:5:f100:0:0:11 on
interface ra0 (5)
Sat Apr 25 20:06:14 md_expire_router: expiring router fe80:0:0:6:f100:0:0:12 on iface ra0 (5)
```

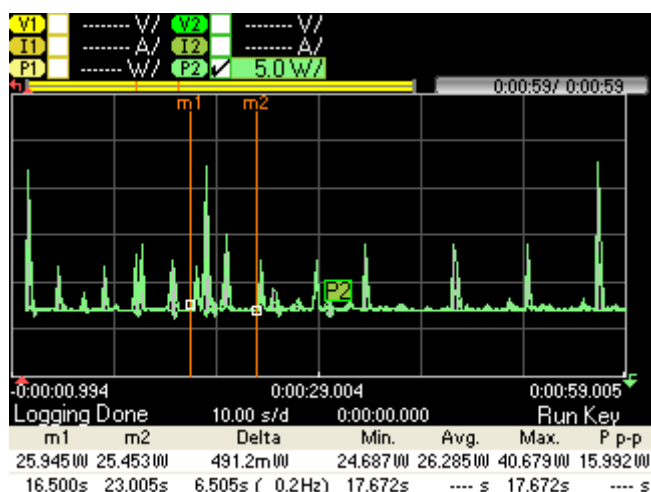
```

Sat Apr 25 20:06:14 md_update_router_stats: add coa 3ffe:100:0:6:21e:e5ff:fedf:9dd7 on
interface (5)
Sat Apr 25 20:06:14 md_update_router_stats: rtr_addr_new: 3ffe:100:0:5:f100:0:0:11
SSID llegit fe80:000:0000:0005:f100:0000:0000:0011
Sat Apr 25 20:06:14 ssid_handler: llegint SSID des de 3ffe:100:0:5:f100:0:0:11 on iface 5
Sat Apr 25 20:06:14 ssid_handler: llegint SSID des de fe80:0:0:5:f100:0:0:11 on iface 5
Sat Apr 25 20:06:14 ssid_handler: rtr_addr_new: 3ffe:100:0:5:f100:0:0:11
Sat Apr 25 20:06:14 ssid_handler: SSID router: 3ffe:100:0:5:f100:0:0:11
Sat Apr 25 20:06:14 ssid_handler: Direccions iguals, node segueix en dormant

```

#### 4.4.5.4.11. Consum del traspàs entre xarxes

En aquest darrer apartat es mostra el consum produït pel procés del traspàs entre xarxes del protocol 802.11 activant i desactivant el mecanisme d'estalvi d'energia PSM.



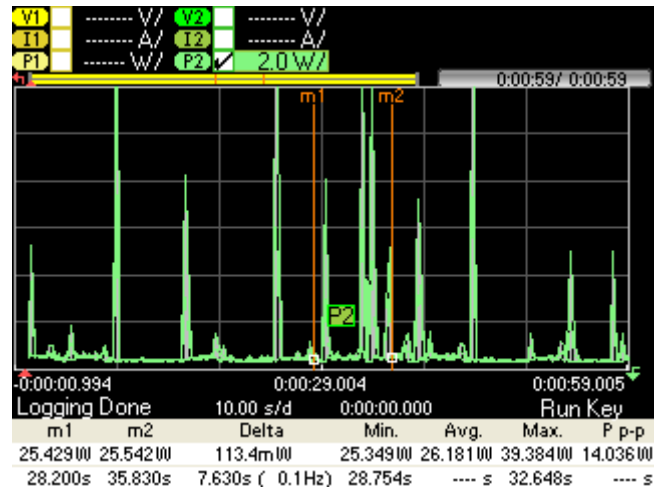
**Fig. 4.30.** Traspàs sense PSM activat

A la figura 4.30 es mostra el consum del traspàs sense PSM activat.

En analitzar les dades i comparar-les amb les obtingudes a l'apartat 4.4.1.4.3, es pot deduir el següent:

**Taula 4.33.** Contract de consum traspàs de xarxa amb consum mitjà sense PSM

Access Point	Cisco Aironet 1131 AG
Estacions	STA 1, STA 2, STA 3
Xarxa	Maqueta MIPv6
Trànsit	Sí
PSM habilitat	No
U-APSD habilitat	No
Potència mitjana	26,285 W
Temps de vida de la bateria	2,027 hores
Increment de bateria	-5,36 %



**Fig. 4.31.** Consum de l'MN durant un traspàs

A la figura 4.31 es mostra el consum del traspàs amb PSM activat. El resultat és l'esperat, ja que el consum és inferior que sense PSM i U-APSD activats.

Tal com s'ha pogut observar, un traspàs té un consum de 26,181 W, cosa que es tradueix en un increment de consum del 5,9% en contrast amb un període d'inactivitat (vegeu la taula 4.7).

Si es compara el consum amb el resultat de la taula 4.19, s'observa que hi ha un guany de temps de bateria del 0,49%.

Comparant les figures 4.30 i 4.31, es conclou que el mecanisme d'estalvi d'energia PSM redueix també el consum de bateria per els traspassos del protocol 802.11.

#### 4.4.5.5. Conclusions

Els resultats de les proves són els següents:

**Taula 4.34.** Resultat de les proves de consum del dimoni modificat

Prova	Potència mitjana	Comparació amb	Increment de bateria	Temps de vida de la bateria
P1	25,713 W	Consum mitjà	- 3,26 %	2,072
P2	25,497 W	P1	0,82 %	2,089
P3	25,868 W	P1	- 0,62 %	2,059
P4	26,359 W	P3	-1,8 %	2,021
P5	25,616 W	P4	3,41 %	2,079
P6	25,595 W	P1	0,43 %	2,081
P7	25,430 W	P2	0,28 %	2,095
P8	25,784 W	P3	0,34 %	-
P9	25,593 W	P4	2,96 %	2,081
P10	25,406 W	P5	0,86 %	2,097
Traspàs sense PSM	26,285 W	Consum mitjà	-5,36 %	2,027
Traspàs amb PSM	26,181 W	Traspàs sense PSM	0,49 %	-

Es pot comprovar com els resultats trobats coincideixen amb els esperats a l'apartat 4.4.5.3.

La supresió del trànsit multicast al demostrador MIPv6 (Router Advertisements i mecanisme de Neighbor Discovery) permet estalviar fins a un 0,82 % de bateria si es compara amb el funcionament per defecte del protocol. Aquest valor té lògica, ja que l'MN deixa de processar els RtrAdv que rep cada 3-8 segons des del router de la xarxa on es troba connectat.

Atès que la diferència de consum entre P1 i P2 és tan sols del 0,82 % (en un cas es llegeixen els RtrAdv cada 3-8 segons i en l'altre l'SSID cada 30 segons), no es recomana disminuir gaire l'interval de lectura de l'SSID, ja que pot provocar més consum que si es llegissin els RtrAdv directament.

Una diferència de consum tant petita s'explica per el fet de que cada cop que llegim l'SSID i hi ha un canvi de xarxa, es calcula la distància de paging de l'adreça llegida amb l'anterior registre guardat a l'MN. Aquest càlcul té una demanda més gran de procés de CPU que no pas la recepció i procés del RtrAdv (fet que explica els resultats de la prova P4).



El valor de lectura de l'SSID haurà de ser doncs superior a l'arribada dels RtrAdv per tal de tenir un guany d'energia.

Per contra, l'aplicació de l'algoritme d'IP Paging en el node dormant permet un estalvi de fins a un 3,41 % de bateria en el cas de no canviar d'àrea de paging. Aquest estalvi és molt beneficiós, sobretot en àrees de paging dinàmiques (vegeu [11]), on difícilment hi ha salts d'àrea de paging si el moviment de l'MN és continu. Aquest estalvi suposa un increment de (segons la taula 4.34):

$$\text{Increment} = ((\text{Temps bateria P10}) - (\text{Temps bateria P9})) * 60 = 0,96 \text{ minuts}$$

Aquest resultat demostra doncs la virtut de mantenir el node en l'estat dormant durant els traspessos de xarxa (molt freqüents en un dispositiu mòbil) tot i la pèrdua de temps de resposta quan l'MN rep trànsit.

Si es compara els valors del node en l'estat actiu realitzant un traspàs (P3) amb el node en l'estat dormant i fent un canvi d'àrea sense canvi de paging (P10) es pot tenir una referència del total d'estalvi d'energia per el fet d'aplicar PSM, IP Paging i eliminar trànsit multicast:

$$\text{Increment} = ((\text{Temps bateria P10}) - (\text{Temps bateria P3})) * 60 = 2,28 \text{ minuts}$$

Si s'extrapola aquest guany a un escenari de VoIP com el de l'apartat 4.4.3.2, s'obté un increment de bateria del 3,21 %, cosa que es tradueix en 4,04 minuts més de conversa. Si se suma aquest guany al del demostrador MIPv6 (amb els mecanismes PSM i U-APSD, sense trànsit multicast i realitzant un canvi de xarxa sense canvi d'àrea de paging) s'obtenen 6,32 minuts de conversa addicionals.

Adicionalment, es pot concloure que el consum total d'un sistema en què s'hagin habilitat els mecanismes d'estalvi d'energia PSM i U-APSD, que tingui el trànsit correctament configurat i mapat i el paging del qual s'hagi modificat de la manera estudiada quedarà reduït de manera important, sense que això suposi una càrrega econòmica per a l'usuari final del producte.

A més dels càlculs i les proves realitzats simulant trànsit de VoIP i de les mesures preses a l'STA 4, es pot deduir que aquestes mesures augmentarien considerablement la vida de les bateries sempre que també s'utilitzés un còdec de veu que reduís la mida dels paquets enviats per a optimitzar la transmissió, i que el mecanisme U-APSD estigués configurat correctament.

La conclusió final, des d'un punt de vista teòric i una vegada fetes les proves, és que s'obté el següent augment de vida de la bateria (per exemple en un terminal telefònic com l'STA 4):

Vida de la bateria sense habilitar mecanismes d'estalvi:

- $(3,7 \text{ V} \times 0,95 \text{ Ah}) / 0,793 \text{ W} = 4,43 \text{ hores}$

Per tant, en revisar l'increment de temps de bateria quan es fa una trucada de VoIP a la taula 4.22, amb PSM, U-APSD i les modificacions fetes al MIPv6 per a afegir paging, s'obté un increment de bateria de:

- $4,43 + 2,01\%$  (vegeu l'apartat 4.4.4.4)  $+ 0,76\%$  (vegeu l'apartat 4.4.5.4.10)  $= 4,55 \text{ hores}$

Aquest resultat és interessant i gens menyspreable, ja que es tracta de gairebé 7,5 minuts més de conversa, tal com s'obté en restar les 4,55 hores resultants del càlcul anterior de la vida total de la bateria de l'STA 4.

- $(4,55 \text{ hores} - 4,33 \text{ hores (hores de bateria totals de l'STA 4)}) \times 60 = 7,38 \text{ minuts.}$

És important recordar que els resultats mostrats a la taula 4.23 no inclouen cap resultat sobre un escenari amb mecanismes d'estalvi d'energia i que, per tant, s'han extrapolat tots els valors a partir de la mesura d'un intent de trucada a l'STA 1 des de l'STA 4 vinculada a l'AP.

Són, per tant, resultats aproximats que en un futur estudi resultarien interessants d'ampliar mitjançant un terminal mòbil per a desenvolupadors que permetés modificar els paràmetres de QoS del mateix.



## CAPÍTOL 5. Conclusions

### 5.1. Fases del projecte i resultats obtinguts

Tot seguit, es detallen les fases del projecte i els resultats obtinguts en cadascuna d'elles.

S'ha fet coincidir cadascun dels capítols amb les fases d'estudi, desenvolupament, implementació i assaig.

En la primera fase, s'han estudiat els protocols IPv6, MIPv6 i els mecanismes d'estalvi d'energia PSM i U-APSD (denominat WMM-PS per la Wi-Fi Alliance). La informació recopilada es troba al Capítol 1.

En una segona fase, s'ha actualitzat el demostrador MIPv6 creat a [11] amb una nova implementació del dimoni per a donar a suport al protocol Mobile IP versió 6 anomenat UMIP. Els detalls de l'elecció de la nova implementació es recullen a l'Annex I.

Al mateix temps, s'ha dut a terme un estudi dels drivers existents amb codi obert que permeten modificar els paràmetres de QoS i l'estalvi d'energia de les targetes amb suport per als mecanismes PSM i U-APSD. A partir d'aquesta informació s'ha fet una tria de les targetes que ofereixen millor suport i que a la vegada estan certificades per la Wi-Fi Alliance. Els detalls de l'elecció d'aquestes targetes es recullen a l'Annex II.

En una tercera fase, s'ha implementat el nou dimoni UMIP al demostrador MIPv6 i s'han realitzat les pertinents proves de validació. A més, per tal de facilitar la feina per a propers projectes, s'ha elaborat un manual d'instal·lació del demostrador tant per les parts ja existents com per a les creades en aquest projecte. Tots aquests detalls es poden consultar a l'Annex III.

A continuació, s'han implementat sobre el dimoni UMIP les modificacions requerides per aquest projecte, per tal d'estudiar-ne l'estalvi d'energia. Els detalls d'aquest procés es poden trobar al Capítol 2.

Finalment, per tal de realitzar correctament les proves d'assaig amb la targeta de xarxa triada a l'Annex II, s'han creat tres manuals:

Un primer, recollit a l'Annex IV, detalla l'ús i les funcions que pot realitzar l'analitzador de potència Agilent N6705A. El segon manual, recollit a l'Annex V, explica pas a pas la configuració dels drivers de la targeta de xarxa triada per tal de modificar els valors de QoS. Finalment, el tercer manual, recollit a l'Annex VI, detalla el software utilitzat i la seva configuració per a les proves.

En la darrera fase del projecte, s'han validat les modificacions realitzades sobre el demostrador MIPv6 amb el dimoni modificat. Se'n poden consultar els detalls al capítol 3.

Al mateix temps, s'han realitzat les proves d'assaig de la targeta de xarxa amb l'analitzador de potència Agilent N6705A per a diversos escenaris i comparant els resultats amb els del dispositiu mòbil Nokia N95. Un dels escenaris de prova ha estat amb el demostrador MIPv6, i amb l'MN com a protagonista. Tots els detalls d'aquestes proves es recullen al Capítol 4.

Es pot concloure, doncs, que s'ha aconseguit complir els objectius marcats per aquest TFC.

Per una banda, s'ha actualitzat i modificat correctament el dimoni UMIP al demostrador MIPv6 per al suport a nodes dormants existent a [11], i s'han implementat solucions per a l'estalvi d'energia.

Per l'altra, s'han localitzat els drivers i les targetes amb suport per als mecanismes PSM i U-APSD, certificades per la Wi-Fi Alliance, i s'ha realitzat un pla de proves per a estudiar l'impacte d'aquests mecanismes sobre la bateria d'un dispositiu mòbil.

Els resultats mostren que és possible eliminar els Routers Advertisements del demostrador MIPv6 i, malgrat això, conservar les funcionalitats del protocol i reduir el consum de bateries fins a 1,02 minuts.

També es verifica que els mecanismes PSM i U-APSD redueixen el consum de bateria quan s'apliquen al demostrador MIPv6. El consum total de bateria es redueix en 0,54 minuts a la maqueta si s'elimina el trànsit multicast i s'apliquen els mecanismes esmentats.

Si s'afegeixen els mecanismes de PSM i U-APSD junt a la supressió del trànsit multicast i l'aplicació de l'algoritme de IP Paging al demostrador de MIPv6, l'increment de bateria arriba fins els 2,28 minuts.

Per altra banda, es comprova que les diferències entre PSM i U-APSD són més grans com més trànsit circula per la xarxa, sobretot si es tracta de trànsit VoIP i prioritzat a l'AP. Aquesta diferència pot suposar un increment de fins a 4,09 minuts de bateria.

Finalment, es poden extrapolar els resultats en els escenaris de VoIP sobre els del demostrador MIPv6 amb tots els estalvis d'energia activats, cosa que suposa un increment de fins a 6,32 minuts de bateria.

Tots aquests estalvis de bateria tenen un cost zero per a l'usuari, ja que aquest només ha d'activar les funcionalitats detallades en aquest projecte.

## 5.2. Futurs desenvolupaments

El demostrador MIPv6 s'ha actualitzat amb èxit i les funcionalitats implementades per [11] són totalment operatives.

En futurs desenvolupaments, i tal com es comenta al capítol 2, s'haurà de poder localitzar correctament l'MN quan canviï de xarxa sense canviar d'àrea de paging. Per a aconseguir-ho, l'HA hauria d'enviar els paquets BRR directament cap als AR que formen l'àrea de paging i seran aquests els encarregats de fer l'enviament final cap a l'MN. En aquest procés, l'HA enviaria la MAC de l'MN a als AR, cosa que permetria conèixer la MAC del MN destí del missatge. Això evitaria la necessitat de crear a priori regles estàtiques a la taula de neighbors, tal com s'ha fet en aquest projecte per a provar que es poden enviar paquets a l'MN sense utilitzar el mecanisme de Neighbor Discovery.

D'altra banda, es proposa millorar el mecanisme per a despertar l'MN abans de l'enviament de qualsevol tipus de trànsit. Per a aquest procés, l'HA hauria d'implementar un buffer de dades més gran per als seus nodes MN i comprovar que realment no es perden els primers paquets.

Finalment, seria interessant aplicar el mateix mecanisme d'enviament directe cap a la MAC que s'aplica per a l'enviament de trànsit des de l'AR cap a l'MN en el sentit invers. Perquè aquest mecanisme fos funcional, es podria utilitzar l'SSID de l'AP per a informar sobre la seva MAC, de manera anàloga a com s'aplica en aquest projecte per a informar sobre l'adreça *Global-Link* de l'AR (vegeu el Capítol 2).

Actualment existeixen diverses línies d'investigació que avancen sobre el model existent d'U-APSD, com per exemple la implementació SU-APSD (*Static Unscheduled Automatic Power Save Delivery*), les característiques principals de la qual consisteixen a limitar el retard en sentit descendent a la capa MAC i minimitzar l'overhead produït pels senyals requerits per U-APSD. També hi ha la implementació AU-APSD (*Adaptive Unscheduled Automatic Power Save Delivery*), una proposta de modificació genèrica que parteix de la idea d'habilitar totes les cues del mecanisme EDCA i, a partir d'aquest punt, ajustar paràmetres com ara el temps màxim de servei (vegeu [22] i [24]). L'aparició de dispositius que incorporin aquestes tècniques obre les portes a noves proves per a determinar la seva eficiència de manera empírica.

Un altre punt interessant a desenvolupar en un futur seria la realització d'un estudi amb telèfons mòbils en els quals sigui possible modificar paràmetres de QoS relacionats amb PSM i U-APSD, ja que en aquest estudi s'han extrapolat les dades obtingudes a partir d'un ordinador amb WLAN i, per tant, seria interessant poder obtenir dades més específiques.

Durant la realització d'aquest estudi l'ús de l'analitzador de potència Agilent N6705A ha facilitat la tasca de mesurar el consum de l'estació de proves, però a més cal tenir en compte que en un futur aquesta potent màquina pot permetre la creació d'escenaris amb diversos dispositius mòbils connectats, ja que admet fins a quatre connexions. A més, aquest dispositiu permetrà contrastar les

dades dels datasheets dels fabricants i realitzar proves en altres xarxes amb trànsit real que utilitzin VoIP sobre WiFi.

### **5.3. Impacte mediambiental del projecte**

La realització d'aquest treball no només no ha tingut un impacte negatiu en el medi ambient sinó que, per la seva naturalesa, contribueix a protegir-lo.

Les implementacions portades a terme sobre el demostrador MIPv6 al Capítol 2 han demostrat que és possible allargar la vida de les bateries, amb el consegüent estalvi per el medi ambient.

Els resultats trobats al Capítol 4 mostren igualment els beneficis de l'ús de mecanismes d'estalvi d'energia.

Per a la realització de les proves s'han utilitzat els ordinadors portàtils dels projectistes i s'ha aprofitat la maqueta ja existent per als escenaris de MIPv6. D'altra banda, el consum elèctric ha quedat minimitzat per l'ús de mecanismes d'estalvi d'energia.

A l'hora de mesurar el consum, s'ha utilitzat un analitzador de potència Agilent N6705A. Aquest analitzador s'ha adquirit no només per a la realització d'aquest treball, sinó també per a compartir-lo entre diversos grups de recerca, cosa que ha suposat una reducció en l'impacte mediambiental.

## Capítol 6. Abreviacions

Nom complet	Sigles acrònims
Access Class BackGround	AC_BK
Access Class Best Effort	AC_BE
Access Class List	ACL
Access Class Video	AC_VI
Access Class Voice	AC_VO
Access Point	AP
Access Router	AR
ACKnowledgement	ACK
Address Resolution Protocol	ARP
Address Resolution Protocol Cache	ARP-CACHE
Association ID code	AID
Authentication, Authorization and Accounting	AAA
Automatic Power Save Delivery	APSD
Binding Acknowledge	BAck
Binding Cache	BC
Binding Refresh Request	BRR
Binding Update	BU
Binding Update List	BUL
Bridge group Virtual Interface	bVI
Care-of-Address	CoA
Carrier Sense Multiple Access with Collision Avoidance	CSMA/CA
Class of Service	CoS
Command Line Interface	CLI
Contention Period	CP
Correspondent Node	CN
Delivery Traffic Indication Map	DTIM
Differentiated Services Code Point	DSCP
Enhanced Distributed Channel Access	EDCA
Escola Politècnica Superior de Castelldefels	EPSC
Estació de proves	STA
File Transfer Protocol	FTP
Foreign Link	FL
Global System for Mobile communications	GSM
Graphical User Interface	GUI
Home Address	HoA
Home Agent	HA
Home Link	HL



Nom complet	Sigles acrònims
HyperText Transfer Protocol	HTTP
International Telecommunication Union Standardization sector	ITU-T
Internet Protocol	IP
Listen Interval	LI
Media Access Control address	MAC
Mobile Internet Protocol version 6	MIPv6
Mobile Node	MN
More Data	MD
Neighbor Advertisement	NghAdv
Neighbor Solicitation	NghSol
PoWer Analyzer	PWA
Power Save Mode	PSM
Power Save poll	PS-Poll
Quality Of Service	QoS
Real time Transport Protocol	RTP
Router Advertisement	RtrAdv
Router Solicitation	RtrSol
Service Period	SP
Service Set Identifier	SSID
Simple Net Tester	SNT
Sistema Operatiu	SO
Temps Restant de Bateria	TRB
Traffic Indication Map	TIM
Transmission Control Protocol	TCP
Universal Mobile Telecommunications System	UMTS
Universal Serial Bus	USB
Universitat Politècnica de Catalunya	UPC
Unscheduled Automatic Power Save Delivery	U-APSD
Usagi Mobile Internet Protocol	UMIP
User Datagram Protocol	UDP
Virtual Local Area Network	VLAN
Voice over Internet Protocol	VoIP
WiFi MultiMedia	WMM
WiFi MultiMedia Power Save	WMM-PS
Wireless Local Area Network	WLAN
Wireless Multimedia Extensions	WME

## REFERÈNCIES I BIBLIOGRAFIA

- [1] Perkins, C. *IP Mobility Support*. RFC 2002. Octubre del 1996.
- [2] Johnson, D.; Perkins, C; Arkko, J. *Mobility Support in IPv6*. RFC 3775. Juny del 2004.
- [3] Deering, S.; Hinden, R. *Internet Protocol Version 6 (IPv6) Specification*. RFC 2460. Desembre del 1998.
- [4] Thomson, S.; Narten, T. *IPv6 Stateless Address Autoconfiguration*. RFC 2462. Desembre del 1998.
- [5] Narten, T.; Nordmark, E.; Simpson, W. *Neighbor Discovery for IP Version 6 (IPv6)*. RFC 2461. Desembre del 1998.
- [6] Vidal, R.; Paradells, J. *Geopaging: a novel multicast approach to IP paging delivery*. Barcelona: Proceedings on the IEEE International Symposium on Personal, Indoor and Mobile Communications (PIMRC), setembre del 2004.
- [7] Vidal, R.; Paradells, J.; Casademont, J. "Labelling mechanism to support distance-based dynamic location updating in cellular networks". *IEE Electronics Letters*, octubre del 2003, vol. 39, núm. 20, p. 1471-1472.
- [8] Arkko, J.; Devarapalli, V.; Dupont, F. *Using IPsec to Protect Mobile IPv6 Signaling Between Mobile Nodes and Home Agents*. RFC 3776. Juny del 2004.
- [9] Conta, A.; Deering, S. *Generic Packet Tunneling in IPv6 Specification*. RFC 2473. Desembre del 1998.
- [10] Devarapalli, V.; Wakikawa, R.; Petrescu, A.; Thubert, P. *Network Mobility (NEMO) Basic Support Protocol*. RFC 3963. Gener del 2005.
- [11] Rubio Albaladejo, V. *Xarxa cel·lular de 4G basada en IPv6: desenvolupament d'un demostrador*. TFC. Castelldefels: EPSC, juliol del 2006.
- [12] Solomon, J.. *Mobile IP: The Internet Unplugged*. New Jersey: Prentice Hall, 1997.
- [13] Soliman, H. *Mobile IPv6. Mobility in a Wireless Internet*. Boston: Addison-Wesley, abril del 2004.
- [14] W. Ruff, A. *Network Security 1 and 2 Companion Guide*. Indianapolis: Cisco Press, 2007.
- [15] Wi-Fi Alliance. *Certified products list* [en línia]. Disponible a: <<http://www.wi-fi.org>>.

[16] IEEE Standards Association. *802.11 Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications* [en línia]. Disponible a: <<http://standards.ieee.org/getieee802/802.11.htm>>.

[17] Sanz de la Rica Mann, J. *Estudio del consumo de baterías en dispositivos móviles IEEE802.11: técnicas de ahorro, caracterización y evaluación*. TFC. Castelldefels: EPSC, juny del 2008.

[18] IEEE Standards Association. *802.11e/D4.4 Draft Supplement to Part 11: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications: Enhancements for Quality of Service (QoS)* [en línia]. Disponible a: <<http://standards.ieee.org/getieee802/802.11.htm>>.

[19] Pérez-Costa, X. [et al.]. *On distributed power saving mechanisms of wireless LANs 802.11e U-APSD vs 802.11 power save mode*. Alemanya: Elsevier, febrer del 2007.

[20] Linksys by Cisco. *Dual-Band Wireless-N USB Network Adapter WUSB600N* [en línia]. Disponible a: <<http://www.linksysbycisco.com/US/en/support/WUSB600N>>.

[21] Agilent Technologies. *Agilent Technologies DC Power Analyzer Model N6705A User's Guide* [en línia]. Disponible a: <<http://cp.literature.agilent.com/litweb/pdf/N6705-90001.pdf>>.

[22] Pérez-Costa, X. [et al.]. *AU-APSD: Adaptive IEEE802.11e Unscheduled Automatic Power Save Delivery* [en línia]. Disponible a: <<http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=04024462>>.

[23] Chen, Y. [et al.]. *Power management for VoIP over IEEE 802.11 WLAN* [en línia]. Wireless Communications and Networking Conference, 2004. Disponible a: <<http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=04024462>>.

[24] Pérez-Costa, X. [et al.]. *SU-APSD: Static IEEE 802.11e Unscheduled Automatic Power Save Delivery* [en línia]. Disponible a: <<http://xavierperezcosta.com/publications/SU-APSD.pdf>>.

[25] Cisco Systems, Inc. *Cisco Aironet 1130AG Series IEEE 802.11A/B/G Access Point Data Sheet* [en línia]. Disponible a: <[http://www.cisco.com/en/US/prod/collateral/wireless/ps5678/ps6087/product\\_data\\_sheet0900aecd801b9058.pdf](http://www.cisco.com/en/US/prod/collateral/wireless/ps5678/ps6087/product_data_sheet0900aecd801b9058.pdf)>.

[26] Ralink Technology, Corp. Driver Linux del chipset RT2870 USB [en línia]. Disponible a: <[http://www.ralinktech.com.tw/data/drivers/2009\\_0424\\_RT2870\\_Linux\\_STA\\_V2.1.1.0.tgz](http://www.ralinktech.com.tw/data/drivers/2009_0424_RT2870_Linux_STA_V2.1.1.0.tgz)>

[27] Blanco Sastre, I. *Veü sobre IP sobre WiFi: desenvolupament de millores per a reduir el consum de bateries*. TFC. Castelldefels: EPSC, 2007.



# ANNEXOS

**TÍTOL DEL TFC/PFC: Xarxa cel·lular de 4G basada en IPv6: desenvolupament d'un demostrador (II)**

**TITULACIÓ: Enginyeria Tècnica de Telecomunicació, especialitat Telemàtica**

**AUTORS: Jaume Giribert Peraire  
Miguel Madrid Villar**

**DIRECTOR: Rafael Vidal Ferré  
DATA: 25 de febrer de 2003**



# ÍNDEX

<b>ANNEX I. IMPLEMENTACIONS DE MIPV6 .....</b>	<b>123</b>
<b>I.1 Projecte WIDE: MIPv6 per a Linux i BSD .....</b>	<b>123</b>
I.1.1 Projecte SHISA – Plataformes BSD .....	123
I.1.2 Projecte USAGI (antic MIPL2) – Plataformes Linux.....	124
I.1.3 Projecte TAHI .....	125
I.1.4 Projecte Nautilus6 .....	125
<b>I.2 Ajuda a les implementacions .....</b>	<b>126</b>
<b>I.3 Altres implementacions .....</b>	<b>127</b>
I.3.1. Windows .....	127
I.3.2. Dispositius Cisco .....	128
<b>I.4 Comparativa entre les implementacions .....</b>	<b>128</b>
<b>I.5 Tria de la implementació a utilitzar .....</b>	<b>129</b>
<b>ANNEX II. ESTUDI DE DRIVERS DE CODI OBERT I TARGETES DE XARXA SENSE FILS AMB CERTIFICACIÓ WMM-PS.....</b>	<b>131</b>
<b>II.1. Projectes de codi obert.....</b>	<b>131</b>
II.1.1. Projecte ATH9K.....	131
II.1.2. Projecte RALINK.....	132
II.1.3. Projecte IWLWIFI .....	132
<b>II.2. Altres implementacions .....</b>	<b>132</b>
II.2.1. Windows .....	132
<b>II.3. Paràmetres de QoS dels diferents projectes.....</b>	<b>133</b>
<b>II.4. Conclusió .....</b>	<b>133</b>
<b>ANNEX III. INSTAL·LACIÓ I CONFIGURACIÓ DE MIPV6 .....</b>	<b>135</b>
<b>III.1. Topologia de xarxa.....</b>	<b>135</b>
<b>III.2. Característiques de maquinari i programari.....</b>	<b>136</b>
<b>III.3. Instal·lació de MIPv6 en Ubuntu-Linux.....</b>	<b>137</b>
III.3.1. Modificació del kernel .....	138
III.3.2. Instal·lació del dimoni MIPv6 .....	148
III.3.2.1. Possibles errors d'instal·lació.....	150
III.3.3. Configuració del dimoni mip6d .....	151

III.3.4.	Instal·lació del dimoni Radvd .....	153
III.3.5.	Configuració de rutes estàtiques .....	155
III.4.	Configuració dels nodes del demostrador MIPv6 .....	155
III.4.1.	Configuració dels node PA i dels routers AR1 i AR2 .....	156
III.4.1.1.	Part comuna .....	156
III.4.1.2.	Configuració del PA .....	156
III.4.1.3.	Configuració de l'AR1 .....	157
III.4.1.4.	Configuració de l'AR2 .....	157
III.4.2.	Configuració del Correspondent Node .....	158
III.4.3.	Configuració del Home Agent .....	159
III.4.4.	Configuració del Mobile Node .....	161
III.4.5.	Configuració de l'AP .....	163
III.5.	Posada en marxa de la maqueta .....	163
III.5.1.	Notes finals .....	164
III.6.	Anàlisi i proves del funcionament del demostrador MIPv6 .....	165
III.6.1.	Proves d'IPv6 .....	165
III.6.2.	Proves de MIPv6 .....	165
III.6.2.1.	Arrencar el dimoni .....	165
III.6.2.2.	Enviament de paquets ICMPv6 .....	170
<b>ANNEX IV. MANUAL DE CONFIGURACIÓ DE L'ANALITZADOR DE POTÈNCIA AGILENT N6705A .....</b>		<b>172</b>
IV.1.	Característiques generals de l'analitzador de potència Agilent N6705A .....	173
IV.2.	Mòduls de potència de l'analitzador de potència Agilent N6705A .....	173
IV.2.1.	Mòdul de potència 1 .....	174
IV.2.2.	Mòdul de potència 2 .....	174
IV.2.3.	Instal·lació general d'un mòdul de potència .....	174
IV.3.	Obertura de la carcassa .....	175
IV.4.	Instal·lació del mòdul de potència .....	176
IV.5.	Configuració de l'analitzador de potència Agilent N6705A .....	178
IV.5.1.	Configuració de la font d'alimentació .....	178

---

IV.5.2.	Connexió de dispositius .....	184
IV.5.3.	Configuració dels paràmetres de captura .....	185
IV.5.4.	Captura i manipulació de les mesures .....	187
IV.5.5.	Emmagatzematge de dades .....	189
IV.5.6.	Control d'errors i emergències.....	192
<b>ANNEX V. CONFIGURACIÓ DEL MAQUINARI UTILITZAT.....</b>		<b>194</b>
V.1.	Access Point Cisco Aironet 1131 AG .....	194
V.1.1.	Connexió al dispositiu .....	195
V.2.	Connexió per port sèrie .....	196
V.3.	Connexió per Telnet .....	197
V.4.	Connexió per http.....	197
V.4.1.	Configuració dels paràmetres bàsics .....	199
V.5.	SSID .....	199
V.6.	BVI.....	199
V.7.	Interfícies físiques .....	200
V.8.	QoS .....	200
V.9.	ACL .....	200
V.10.	ARP CACHE .....	201
V.11.	Mapatge DSCP.....	201
V.12.	Modificació del DTIM.....	202
V.13.	Targeta de xarxa Linksys WUSB600N.....	202
V.13.1.	Instal·lació de la targeta .....	203
V.13.1.1.	Habilitació de paràmetres QoS .....	206
V.13.2.	Habilitació del mode monitor.....	206
V.14.	Ordinador portàtil Sony VAIO S5M .....	207
V.15.	Telèfon Nokia N95 .....	209
V.16.	Ordinador portàtil BENQ S72 .....	211
V.17.	Ordinador portàtil Sony VAIO FW11M.....	212
V.18.	Analitzador de potència Agilent N6705A .....	213



<b>ANNEX VI. CONFIGURACIÓ DEL PROGRAMARI UTILITZAT .....</b>	<b>216</b>
VI.1. Wireshark .....	216
VI.2. Kismet.....	218
VI.3. Iperf .....	221
VI.4. Vsftp.....	222
VI.5. Simple Network Tester .....	223
<b>ANNEX VII. MODIFICACIONS AL CODI DEL DIMONI UMIP .....</b>	<b>226</b>
VII.1 Modificacions a ip6mh.h.....	227
VII.2 Modificacions a proc_sys.c.....	227
VII.3 Modificacions a proc_sys.h.....	228
VII.4 Modificacions a movement.c.....	228
VII.5 Modificacions a movement.h .....	238
VII.6 Modificacions a mn.c .....	238
VII.7 Modificacions a mn.h .....	246
VII.8 Modificacions a ha.c .....	246

## ANNEX I. Implementacions de MIPv6

Per tal de muntar el demostrador, cal valorar les diferents possibilitats existents avui dia per tal de tenir suport per al protocol MIPv6.

Per a facilitar-ne l'elecció, en aquest document es recullen les diverses implementacions amb més suport de la comunitat i en base als projectes existents.

Per a cadascun d'ells se'n recalquen les particularitats (SO, funcionalitats, suport, estat del projecte, etc.) i, finalment, se'n fa una comparativa per a justificar l'opció triada en aquest projecte.



### I.1 Projecte WIDE: MIPv6 per a Linux i BSD

**WIDE** (*Widely Integrated Distributed Environment*) és un consorci format l'any 1988 al Japó per diverses empreses i els àmbits de la indústria i la recerca acadèmica per tal d'impulsar la recerca i el desenvolupament en les tecnologies d'Internet.

Actualment, unes 100 corporacions i 40 universitats treballen en el projecte WIDE, en una àmplia varietat d'activitats relacionades amb les tecnologies de la propera generació d'Internet. Se'n pot trobar més informació a la seva pàgina web: <http://www.wide.ad.jp/>.

Una de les recerques en què treballa WIDE és promoure el desplegament de la tecnologia IPv6 a través dels projectes **SHISA**, **USAGI**, **TAHI** i IPv6 Fix.

En col·laboració estreta amb els quatre projectes anteriors, el projecte **Nautilus6** s'encarrega de l'aplicació pràctica de les comunicacions mòbils per a IPv6.

#### I.1.1 Projecte SHISA – Plataformes BSD



El projecte WIDE va desenvolupar dues implementacions de Mobile IPv6 per als projectes següents:

- KAME project (KAMEIP): implementació en codi obert de la pila IPv6, IPsec i MobileIPv6 per a plataformes BSD. Se'n pot trobar més informació a la seva pàgina web: <http://www.kame.net/>.
- InternetCAR project (SFCMIP): implementació en codi obert per a l'enviament de dades en automòbils.

A la primavera de 2004, es van unir les dues implementacions en una de sola, anomenada SHISA. Se'n pot trobar més informació a la seva pàgina web: <http://www.mobileip.jp/>.

SHISA va ser implementada a la plataforma KAME per tal de donar suport a MobileIPv6 en plataformes BSD. El codi d'aquesta implementació es pot descarregar de la pàgina de KAME.

A data de març de 2009 no existeix cap altra implementació pública de SHISA, tot i que la seva pàgina anuncia que SHISA es portarà a altres sistemes operatius.

Les versions del sistema operatiu admeses són les següents:

- FreeBSD 4.0: des d'aquesta versió, totes les versions posteriors de FreeBSD ja tenen integrada (i activada per defecte) en el seu kernel la implementació de KAME. Se'n pot trobar més informació a: <http://www.ipv6.org/impl/freebsd.html>.
- OpenBSD 2.7 i posteriors.
- NetBSD 1.5 i posteriors (també integrada en el seu kernel).
- BSD/OS 4.2 i posteriors.

Les darreres versions de KAME són:

- kame-20090301-freebsd54-snap
- kame-20090301-netbsd20-snap
- kame-20090301-openbsd36-snap

### I.1.2 Projecte USAGI (antic MIPL2) – Plataformes Linux



El projecte USAGI (*UniverSAl playGround for IPv6*) és avui dia la implementació d'IPv6, IPsec (tant per a IPv4 com per a IPv6) i MIPv6 per a sistemes basats en Linux més activa. La seva pàgina web és: <http://linux-ipv6.org/>.

Anteriorment, la implementació principal per a Linux de la pila de protocols MIPv6 era MIPL (*Mobile IPv6 for Linux*), desenvolupada pel projecte GO-Core (inscrit a la Helsinki University of Technology – HUT) en cooperació amb el projecte USAGI.

La darrera versió de MIPL (MIPL2 o versió 2.0) es va reescriure completament perquè inclogués la majoria de les funcionalitats en l'espai d'usuari (mitjançant un dimoni) i només una mínima capa de suport per al kernel de Linux.

Aquesta darrera versió és compatible només fins al kernel 2.6.16 de Linux. La versió anterior de MIPL (v1.1) només era compatible amb el kernel 2.4.26. Totes dues versions ja no disposen de suport i el projecte es considera tancat.

El codi MIPL (juntament amb el codi per a aplicar el patch al kernel) es pot trobar a <http://www.mobile-ipv6.org/> o a <http://www.mipl.mediapoli.com/>.

Atès que el projecte MIPL va col·laborar amb USAGI, aquest últim és ara l'encarregat d'oferir suport, manteniment i continuïtat al projecte mitjançant el programari UMIP (Usagi Mobile IP).

UMIP és, per tant, la branca activa del projecte MIPL2.

La darrera versió d'UMIP és la 0.4, i és compatible amb el kernel 2.6 de Linux. Se'n pot descarregar el codi a la seva pàgina web: <http://umip.linux-ipv6.org/>.

### I.1.3 Projecte TAHI



El projecte TAHI s'encarrega de les proves de verificació de la tecnologia vinculada a IPv6 desenvolupant tests d'interoperabilitat i conformitat.

Està estretament vinculat als projectes USAGI i SHISA, ja que s'encarrega de dur a terme les proves de qualitat amb la tecnologia desenvolupada per aquests dos projectes.

La seva pàgina web és: <http://www.tahi.org/>.

### I.1.4 Projecte Nautilus6



El projecte Nautilus6 és l'encarregat d'aportar suport al protocol MIPv6 (NEMO o Network Mobility i les seves extensions) per als projectes USAGI i SHISA, mitjançant diferents implementacions.

Les darreres implementacions per a **SHISA** són les següents:

- **MIPv6/NEMO** (RFC 3963), ja inclosa dins el projecte mateix.
- **TARZAN**: implementació d'FMIPv6 (draft 3 de l'RFC 4068).  
La seva pàgina web és: <http://software.nautilus6.org/TARZAN>.

Les darreres implementacions per a **USAGI** són les següents:

- **NEPL** (*NEMO Platform for Linux*) for UMIP: implementació de NEMO (RFC 3963) per al kernel 2.6 de Linux i ja inclosa dins el projecte mateix.  
La seva pàgina web és: <http://software.nautilus6.org/NEPL-UMIP/>.
- Implementació de Dual Stack Mobile IPv6 (**DSMIPv6**) per al kernel 2.6 de Linux (<http://tools.ietf.org/html/draft-ietf-mext-nemo-v4traversal-05>).  
La seva pàgina web és: <http://software.nautilus6.org/DSMIP/index.php>.

- Implementació de **Multiple Care-of-Addresses** per al kernel 2.6 de Linux (<http://tools.ietf.org/html/draft-ietf-monami6-multiplecoa-09>). La seva pàgina web és: <http://software.nautilus6.org/MCoA/index.php>.
- **FMIPv6**: implementació del Fast Handover per a MIPv6 (FMIPv6) segons RFC 4068 per al kernel 2.6 de Linux. La seva pàgina web és: <http://www.fmipv6.org/>.

## I.2 Ajuda a les implementacions

En el seu esforç de donar a conèixer aquest protocol i de facilitar-ne l'ús, el projecte Nautilus6 ha desenvolupat dues eines força interessants que ajuden a implementar el MIPv6 de manera més simple.

### Haiku (<http://software.nautilus6.org/HAiku/index.php>)

Es tracta d'una interfície web que facilita l'administració dels diferents MN connectats a un HA.

Permet afegir i esborrar de manera intuïtiva en base a una interfície web (vegeu la figura I.1):

#### Mobile Hosts

Name	Description	HoA	
mn01	host	2001:db8:2ae9:b03a::5	✖ Destroy
xxxx		2001:db8:2ae9:b03a::6	✖ Destroy
mymobile	My mobile node.	2001:db8:2ae9:b03a::7	✖ Destroy
n770	Nokia N770	2001:db8:2ae9:b03a::8	✖ Destroy
coin	oab	2001:db8:2ae9:b03a::9	✖ Destroy
rhp		2001:db8:2ae9:b03a::a	✖ Destroy
mobilepoint		2001:db8:2ae9:b03a::b	✖ Destroy
KMobile	Mobile node to be used on laptop	2001:db8:2ae9:b03a::c	✖ Destroy
mnm	mnm	2001:db8:2ae9:b03a::d	✖ Destroy
mn1		2001:db8:2ae9:b03a::f	✖ Destroy
tes	hahaha	2001:db8:2ae9:b03a::10	✖ Destroy

#### Mobile Routers

Name	Description	HoA	MNP	
MRx		2001:db8:2ae9:b03a::e	2001:db8:2ae9:b03c::	✖ Destroy

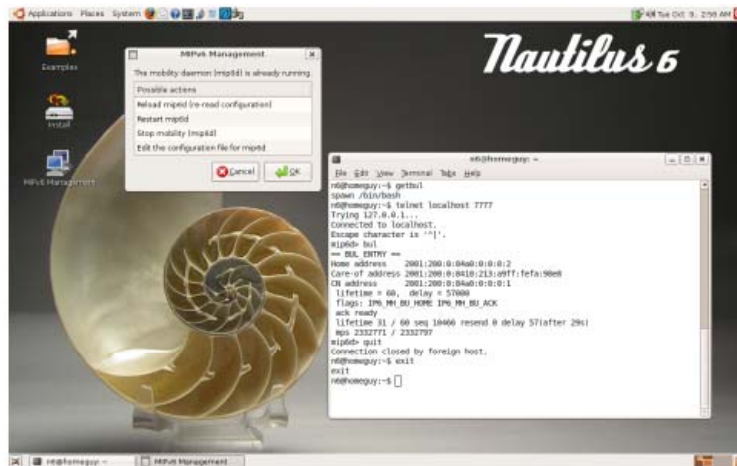
[+ Create MN](#)

Fig. I.1. Detall de Haiku amb els MN inscrits a un HA

## Homeguy (<http://software.nautilus6.org/homeguy/index.php/>)

Actualment ofereixen un servei “ready-to-use” de MIPv6 en un Live-CD que conté una distribució Ubuntu Gutsy 7.10 amb suport complet per a Mobile IPv6 amb eines d’assaig i programari multimèdia:

- Eines d’escaneig de xarxa: Wireshark, mip6tester, Scapy6, iperf, traceroute6, ping6, radvd, racoon2, sonar.
- Programari multimèdia: Totem, SIP Communicator.



**Fig. II.2.** Escriptori de Homeguy

Una vegada instal·lat, es pot configurar perquè actuï com un CN, MN o HA.

Es disposa de l’opció, mitjançant un simple registre, de crear en línia i configurar Mobile Nodes propis (per a descarregar-los després i executar-los sense instal·lar) mitjançant el Home Agent Service, que no és altra cosa que un servidor web que executa l’eina Haiku i actua d’HA: <https://op-ha.nautilus6.org/>.

## I.3 Altres implementacions

### I.3.1. Windows

Microsoft va desenvolupar un projecte d’implementació de Mobile IPv6 temps enrere, però actualment aquest projecte no es troba disponible. Suposadament, té suport tant per al protocol Mobile IPv6 com per a IPSec. Se’n pot trobar més informació a [http://technet.microsoft.com/ca-es/library/bb878106\(en-us\).aspx](http://technet.microsoft.com/ca-es/library/bb878106(en-us).aspx).

Windows només ofereix les funcionalitats de CN per a les versions de Windows Server (2000, 2003 i 2008) mitjançant l’ordre:

*netsh interface ipv6 set mobility correspondentnode=enabled*

### I.3.2. Dispositius Cisco

Una darrera opció és la implementació de Mobile IPv6 amb aparells Cisco. A partir de l'IOS versió 12.4, aquests dispositius tenen implementades funcionalitats pròpies d'aquest protocol.

Aquesta implementació de NEMO no conté suport per a IPSec (sempre que no s'activin protocols d'enrutament IPv6).

Se'n pot trobar un manual d'implementació a la pàgina web del fabricant:  
<http://www.cisco.com/en/US/docs/ios/ipv6/configuration/guide/ip6-mobile.html/>.

## I.4 Comparativa entre les implementacions

Amb la informació facilitada per les diferents webs de les implementacions, s'ha creat la següent taula comparativa amb les funcionalitats de cadascun dels projectes:

Característica	Linux (USAGI)	BSD (SHISA)	Windows	Cisco
Funcions d'MN	X	X		X
Funcions d'HA	X	X		X
Funcions de CN	X	X	X	X
Tunneling IPv6	X	X		X
Route Optimization	X	X		X
Return Routability	X			
IPSec (RFC 3776)	X	X		
Network Mobility NEMO (RFC 3963)	X	X		X
Fast Handover FMIPv6 (RFC 4068)	X	X		
Dual Stack Mobile IPv6 (DSMIPv6)	X			
Multiple Care-of-Addresses	X			
Extension to Sockets API for Mobile IPv6 (RFC 4584)		X		
Hierarchical MIPv6 – HMIPv6 (RFC 4140)	suport	X		
Servidor intern per a visualitzar la BUL	X			
Interfície web HA	X	X		
DHAAD	X	X		
Live CD	X			

## I.5 Tria de la implementació a utilitzar

A l'hora de triar la millor implementació de totes les esmentades per a actualitzar la maqueta existent, es va dubtar en un principi entre SHISA i USAGI:

- Tots dos projectes són actius, tot i que Nautilus6 ha creat eines per a facilitar l'ús d'USAGI.
- Hi ha alguns RFC que SHISA implementa i USAGI no, i viceversa.
- SHISA ja s'inclou per defecte en algunes plataformes BSD.

Finalment, es va triar la implementació d'USAGI pels motius següents:

- Un seguiment diari dels dos projectes va demostrar que USAGI rebia més actualitzacions.
- Les aportacions de Nautilus6 al projecte USAGI simplifiquen molt les tasques d'implementació i permeten aportar interfície gràfica a l'HA.
- Linux és l'opció més recomanable, ja que disposa d'una comunitat més gran d'usuaris (sobretot de les distribucions Debian/Ubuntu).
- El projecte USAGI és molt més simple de modificar (en cas de programar noves característiques), ja que gran part del seu codi treballa com un dimoni a la capa d'usuari. Això significa que en cas de modificar part del seu codi, no caldrà recompilar quasi mai el kernel del sistema.
- Al tractar-se de un projecte basat en MIPL2, es pot aprofitar la feina realitzada en treballs anteriors (modificacions per a admetre nodes dormants, paging, etc.).
- A partir del kernel 2.6.2x de Linux, ja estan inclosos els headers de MIPv6 i, per tant, qualsevol distribució amb aquest kernel o superior és capaç d'admetre MIPv6.

S'ha triat el sistema operatiu Linux Ubuntu com a base per a la instal·lació d'USAGI, ja que era el que utilitzava Nautilus6 per al seu Live-CD Homeguy.





## ANNEX II. ESTUDI DE DRIVERS DE CODI OBERT I TARGETES DE XARXA SENSE FILS AMB CERTIFICACIÓ WMM-PS

### II.1. Projectes de codi obert

S'ha dut a terme un estudi de mercat amb l'objectiu de trobar la solució que s'adapti millor a les necessitats d'aquest projecte. Així doncs, s'ha cercat una targeta amb suport de la certificació WMM-PS (vinculada al suport de QoS, per exemple, 802.11e), 802.11n, i que disposi d'un driver de codi obert per a Linux.

En el mercat actual es poden trobar principalment tres solucions que compleixin aquests requisits, basades respectivament en els chipsets Atheros, Intel i Ralink.

Excepte Ralink, que proporciona el seu propi driver per a Linux, la resta de solucions reben suport no oficial, és a dir, que és una comunitat externa de desenvolupadors qui proporciona els drivers.

Es tracta de projectes en desenvolupament constant i que no disposen de suport oficial, cosa que de vegades en limita la utilitat. Per exemple, la qualitat de servei és un camp encara poc desenvolupat en aquests projectes. El projecte associat a les targetes basades en el xip d'Atheros és Madwifi, mentre que el projecte de les targetes Intel s'anomena IWLwfi.

#### II.1.1. Projecte ATH9K

El projecte per als xips d'Atheros és el més veterà, amb una comunitat que ja fa anys que hi treballa. La versió actual s'anomena ath9k i compta amb el suport semioficial d'Atheros, que disposa del seu propi driver propietari però col·labora amb els projectes de desenvolupament de drivers integrats en el kernel de Linux mateix.

Tal com es pot veure, el suport de QoS i, particularment, les opcions relacionades amb WMM estan integrats en les versions de kernel 2.6.22 i superiors. Per tal d'habilitar el suport per a WMM cal habilitar les opcions següents per al kernel:

- *CONFIG\_NETDEVICES\_MULTIQUEUE*: aquesta opció activa la possibilitat d'assignar diverses cues de transmissió a un dispositiu de xarxa.
- *CONFIG\_NET\_SCHED*: en habilitar aquesta opció és possible assignar diferents disciplines de cua als dispositius de xarxa.

Aquest projecte es coneixia com a Madwifi i és un dels més estesos i amb un major nombre de desenvolupadors.

### II.1.2. Projecte RALINK

Aquest fabricant ofereix drivers de codi obert per a Linux de manera gratuïta, cosa que permet, per exemple, operar amb les opcions de qualitat de servei com WMM-PS en les targetes basades en el chipset RT2870, com ara la targeta WUSB600N de Linksys.

Alguns dels paràmetres que permet modificar o activar són:

- *WMMCapable*: activa el WMM.
- *APSDCapable*: activa el mecanisme d'APSD (*Automatic Power Save Delivery*).
- *PSMode*: permet triar diversos modes d'operació per al mecanisme d'estalvi d'energia PSM.
- *APSDC*: controla les categories d'accés del mecanisme EDCA (*enhanced distributed channel access*), que corresponen a BE (best effort), BK (background), VI (video) i VO (VoIP). EDCA s'implementa en IEEE.802.11e.

### II.1.3. Projecte IWLWIFI

Aquest projecte de codi obert té com a objectiu desenvolupar drivers per a les targetes basades en els xips Intel.

Pensant especialment en 802.11n, permet operar amb targetes basades, per exemple, en el xip 4965 o superior, encara que de manera molt limitada pel que fa a QoS, per la qual cosa no resulta gaire útil per a aquest estudi.

## II.2. Altres implementacions

### II.2.1. Windows

Els tres chipsets esmentats disposen de drivers per a Windows XP o Windows Vista. Malgrat ser totalment funcionals, aquests drivers no són de codi obert, cosa que en limita la utilitat per a aquest estudi, tot i que permeten accedir i modificar certs paràmetres de QoS com ara l'habilitació de WMM-PS en el cas de Ralink i les targetes basades en el seu xip RT2870.

### II.3. Paràmetres de QoS dels diferents projectes

La taula II.1 mostra quins paràmetres permet activar cadascun dels projectes esmentats.

**Taula II.1.** Paràmetres QoS dels chipsets analitzats

Projecte	WMM	PSM	APSD
RALINK	SÍ	SÍ	SÍ
IWLWIFI	NO	SÍ	NO
ATH9K	NO	SÍ	NO

### II.4. Conclusió

A raó dels resultats d'aquest estudi s'ha seleccionat per la seva flexibilitat i les seves possibilitats el projecte de Ralink.

El xip RT2870 és una de les solucions 802.11n més esteses actualment en el mercat, i el seu driver permet treballar amb els paràmetres de QoS necessaris per a dur a terme aquest projecte.

A més, el seu xip compta amb la certificació WMM-PS entre d'altres, requisit per a l'elecció de la targeta.

Entre les opcions possibles, s'ha adquirit una targeta sense fils USB Linksys WUSB600N, que compleix tots els requisits.

La taula II.2 mostra la llista de targetes que s'han considerat, així com alguna de les seves característiques.

**Taula II.2.** Llista de targetes analitzades

Nom (fabricant)	Chipset	Certificats	Consum	Potència (N)	Driver Linux	Preu
DWA-645 (D-LINK)	ATHEROS AR5416	IEEE 802.11b/g/n	Tx: 930 mA Rx: 450 mA Sb: 450 mA EIRP: 17 dbm	Tx (2,4 GHz): 17 dBm	MADWIFI (1)	47,30€
TEW-644UB (TRENDnet)	RALINK 2870	IEEE 802.11b/g/n	Tx: 320 mA Rx: 250 mA	Tx (2,4 GHz): 12 dBm Rx (2,4 GHz): --63 dBm	RALINK (2)	24,90€
WUSB600N (LINKSYS)	RALINK 2870	IEEE 802.11b/g/n	Tx: 480 mA Rx: 300 mA	Tx (2,4 GHz): 16 dBm Rx (2,4 GHz): -91 dBm G: 2dB +/- 1 dB	RALINK (2)	55,62€
HWUN-1 (HAWKING)	RALINK 2870	IEEE 802.11b/g/n	Tx: 480 mA Rx: 300 mA	Tx (2,4 GHz): 16 dBm	RALINK (2)	51,99€
SMCWUSBS-N (SMC)	RALINK 2870	IEEE 802.11b/g/n		Tx (2,4 GHz): 14 dBm	RALINK (2)	29,75€
WMP300N V.1 (LINKSYS)	BROADCOM	IEEE 802.11b/g/n		Tx (2,4 GHz): 10 dBm Rx (2,4 GHz): -70 dBm	BROADCOM(3)	59,79€
AR9001U-2NG (ATHEROS (OEM))	ATHEROS	IEEE 802.11b/g/n	Tx: 380 mA Rx: 250 mA	Tx (2,4 GHz): 14 dBm Rx (2,4 GHz): --64 dBm	MADWIFI (1)	32,95€
AWUS047N (ALFA)		IEEE 802.11b/g/n		Tx (2,4 GHz): 14 dBm Rx (2,4 GHz): --63 dBm		ND

## Annex III. Instal·lació i configuració de MIPv6

En aquest annex es recullen amb un alt grau de detalls els passos a seguir per a crear un demostrador MIPv6 amb la implementació d'USAGI – UMIP.

### III.1. Topologia de xarxa

El demostrador utilitzat en aquest projecte segueix la topologia següent:

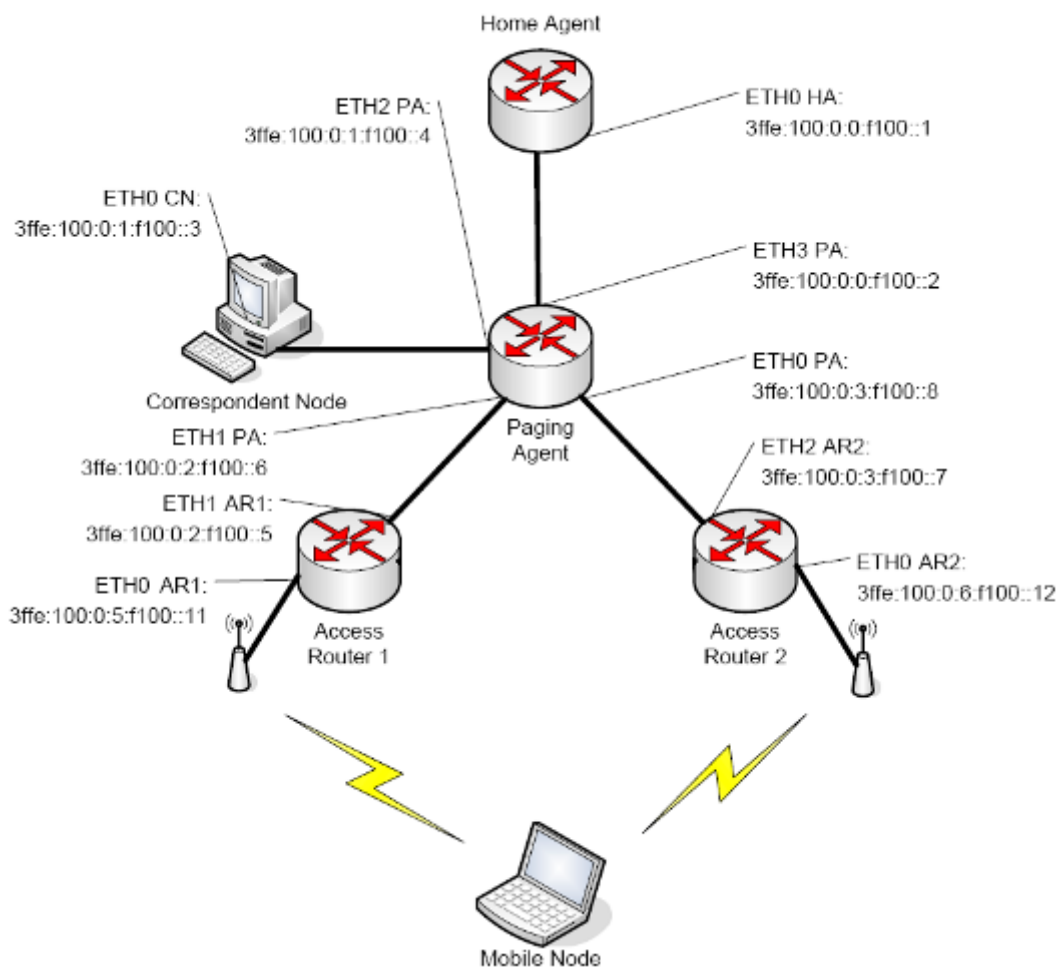


Fig. III.1. Topologia de xarxa

Tot i que l'àmbit de la maqueta no ho requereix, s'han configurat les interfícies dels sis terminals amb adreces *Global-Link* i *Local-Link* amb un prefix de xarxa de 64 bits, ja que el dimoni MIPv6 només accepta adreces globals.

En cas de no utilitzar el dimoni MIPv6, n'hi hauria prou d'emprar les adreces *Local-Link*.

### III.2. Característiques de maquinari i programari

El demostrador MIPv6 està format per sis ordinadors amb característiques pròpies de CPU, memòria RAM, interfícies de xarxa i sistemes operatius. A la taula següent es recull tota la informació d'aquests equips:

**Taula III.1.** Característiques tècniques dels equips utilitzats

<b>Home Agent</b>	
CPU	Intel ® Pentium ® 4 CPU 3.00 GHz
Targeta ETH0	Realtek Semiconductor Co., Ltd. RT8139
Versió del SO	Ubuntu Linux 7.10 kernel 2.6.22
Driver ETH0	Realtek Semiconductor Co., Ltd. RTL-8139/8139C/8139C+ (rev 10)
<b>Correspondent Node</b>	
CPU	Intel ® Pentium ® 4 CPU 1.60 GHz
Targeta ETH0	Broadcom Corporation: device 8401
Versió del SO	Fedora Core 4 kernel 2.6.15
Driver ETH0	Broadcom Corporation BCM4401 100Base-T (rev 01)
<b>Paging Agent</b>	
CPU	Intel ® Pentium ® 4 CPU 1.80 GHz
Targeta ETH0	3Com Corporation 3C905C-TX Fast Etherlink for PC Management NIC
Targeta ETH1	3Com Corporation 3C905C-TX Fast Etherlink for PC Management NIC
Targeta ETH2	3Com Corporation 3C905C-TX Fast Etherlink for PC Management NIC
Targeta ETH3	3Com Corporation 3C905C-TX Fast Etherlink for PC Management NIC
Versió del SO	Fedora Core 4 kernel 2.6.15
Driver ETH0	3Com Corporation 3c905C-TX/TX-M [Tornado] (rev 78)
Driver ETH1	3Com Corporation 3c905C-TX/TX-M [Tornado] (rev 78)
Driver ETH2	3Com Corporation 3c905C-TX/TX-M [Tornado] (rev 78)
Driver ETH3	3Com Corporation 3c905C-TX/TX-M [Tornado] (rev 78)

<b>AR1</b>	
CPU	Intel ® Pentium ® 4 CPU 3.00 GHz
Targeta ETH0	D-Link System Inc DFE-538TX 10/100 Ethernet Adapter
Targeta ETH1	ASUSTeK Computer Inc.: device 811d
Versió del SO	Fedora Core 4 kernel 2.6.15
Driver ETH0	D-Link System Inc RTL8139 Ethernet (rev 10)
Driver ETH1	Intel Corporation 82541GI/PI Gigabit Ethernet Controller (rev 05)
<b>AR2</b>	
CPU	Intel ® Pentium ® 4 CPU 3.00 GHz
Targeta ETH0	D-Link System Inc DFE-538TX 10/100 Ethernet Adapter
Targeta ETH2	ASUSTeK Computer Inc.: device 811d
Versió del SO	Fedora Core 4 kernel 2.6.15
Driver ETH0	D-Link System Inc RTL8139 Ethernet (rev 10)
Driver ETH2	Intel Corporation 82541GI/PI Gigabit Ethernet Controller (rev 05)

S'ha procurat conservar les parts de la maqueta anterior que ja funcionaven per tal de no repetir la feina feta, però malgrat això se n'expliquen igualment els detalls de la configuració amb el sistema operatiu escollit a l'Annex I.

### III.3. Instal·lació de MIPv6 en Ubuntu-Linux

Un dels avantatges de la implementació actual de MIPv6 és que gran part de les funcionalitats del protocol es poden modificar des de l'espai d'usuari (dimoni `mipd6`) i, per tant, no cal recompilar el kernel cada vegada que es volen afegir modificacions al dimoni.

En l'anterior implementació MIPL (vegeu l'Annex I) i fins a la versió 2.6.16 del kernel de Linux, s'havia d'executar un patch sobre el codi font del kernel (respectant-ne la versió corresponent) abans de compilar-lo i instal·lar-lo a la màquina.

D'ençà la versió 2.6.20 del kernel de Linux, el suport per a MIPv6 està integrat dins del seu propi codi font i, per tant, no cal aplicar-hi cap patch.

Aquest fet permet actualitzar el demostrador amb nou maquinari (com ara noves targetes de xarxa amb estalvi d'energia), ja que sol ser necessari actualitzar el kernel per a obtenir-ne la compatibilitat.



En cas contrari, s'hauria de dependre de terceres persones per a esperar un patch compatible amb la versió del kernel.

### III.3.1. Modificació del kernel

Tal com s'ha comentat, el primer pas serà donar suport al protocol MIPv6 des del kernel de Linux.

En aquest manual es pressuposa que l'usuari ja ha instal·lat i configurat l'ordinador amb una distribució Linux-Ubuntu i que totes les ordres s'executaran com a root (executant l'ordre **#sudo su**).

1. En primer lloc, és important instal·lar totes les biblioteques necessàries per a posar a punt la maqueta:

```
#apt-get install gcc build-essential libtool autoconf automake libncurses5  
libncurses5-dev libpcap-dev indent flex bison
```

2. A continuació, cal comprovar quina versió hi ha instal·lada:

```
#uname -a (que retorna una descripció de tot el sistema)
```

```
#uname -r (que retorna únicament la versió numèrica del kernel)
```

En el cas d'Ubuntu, els kernels amb l'etiqueta *generic* es refereixen a tots els kernels compatibles amb màquines d'arquitectura x86 (en general, Intel).

3. Seguidament, cal descarregar des de la pàgina oficial la versió desitjada del kernel de Linux :

<http://www.kernel.org/>

A l'hora de donar suport al protocol MIPv6, es recomana descarregar una versió superior a la que ja hi hagi instal·lada.

També es pot fer el mateix amb l'ordre següent (com a root):

```
#apt-get install linux-source-[versió desitjada del kernel]
```

4. A continuació, cal descomprimir el contingut a la carpeta `/usr/src`:

```
#tar -xjvf [nom arxiu].tar.bz2 -c /usr/src
```

5. Una vegada descomprimit, cal situar-se en el directori `/usr/src/[nou kernel]` i executar l'ordre següent:

```
#cp /boot/config-[versió del kernel que fem servir] .config
```

Aquesta ordre copiarà el fitxer de configuració del kernel (p. ex., *config-2.6.27-7-generic*) a un nou fitxer (*.config*) a la carpeta */usr/src/[nou kernel]*. Aquest pas estalviarà temps a l'hora de configurar el kernel.

6. Dins el directori */usr/src/[nou kernel]*, cal executar l'ordre següent:

**#make menuconfig**

i apareixerà la pantalla següent:

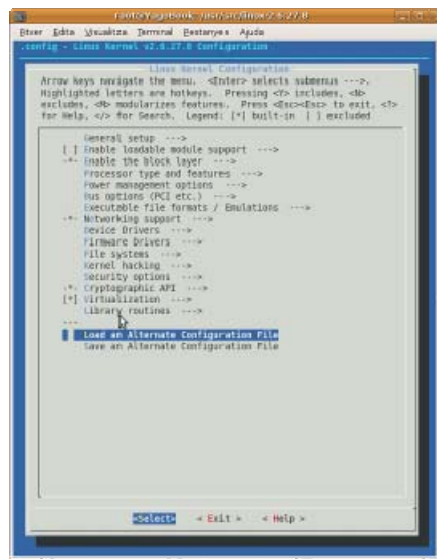


Fig. III.2. Menú *menuconfig*

7. A continuació, es carrega la configuració que s'ha guardat al punt 4. D'aquesta manera, ja només cal configurar la part de MIPv6.

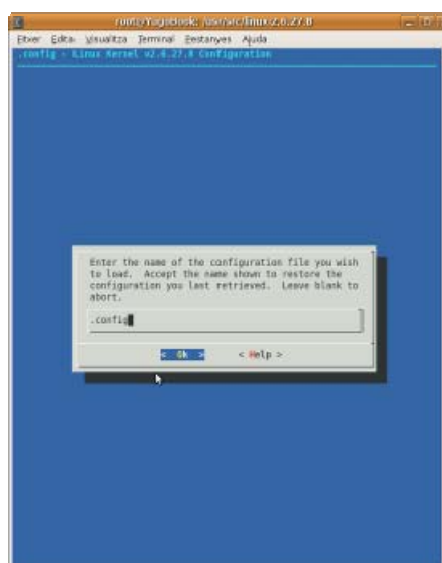
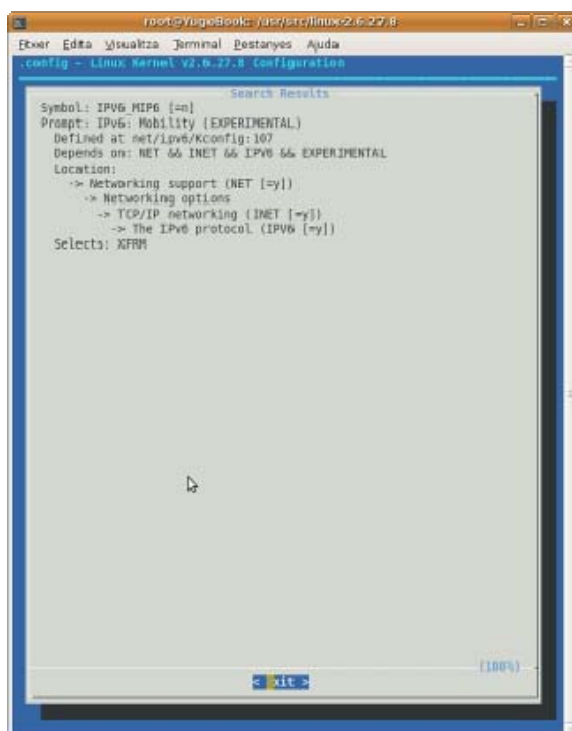


Fig. III.3. Càrrega de *.config*

8. Dins el menú de configuració, la tecla / s'utilitza per a trobar els símbols, i la tecla ? per a obtenir-ne més informació.



**Fig. III.4.** Cerca del driver



**Fig. III.5.** Detalls del driver

9. Una vegada familiaritzats amb l'entorn, es disposa de dues opcions a l'hora d'instal·lar-ne el suport: com a *module* (M), cosa que obliga a carregar els drivers de suport cada vegada que s'engega l'ordinador, o bé com a *yes* (Y), que instal·la el driver permanentment.

En el cas d'aquest estudi, sembla més convenient fer la instal·lació com a *module*, ja que d'aquesta manera s'obté més control per a experimentar amb la maqueta.

Per a tenir el suport de MIPv6 al kernel, cal activar el símbols següents.

Suport de MIPv6 bàsic per a tots els nodes:

**CONFIG\_EXPERIMENTAL**  
**CONFIG\_SYSVIPC**  
**CONFIG\_PROC\_FS**  
**CONFIG\_NET**  
**CONFIG\_INET**  
**CONFIG\_IPV6**  
**CONFIG\_IPV6\_MIP6**  
**CONFIG\_XFRM**  
**CONFIG\_XFRM\_USER**  
**CONFIG\_XFRM\_SUB\_POLICY**  
**CONFIG\_INET6\_XFRM\_MODE\_ROUTEOPTIMIZATION**

Suport per al Home Agent i el Mobile Node:

**CONFIG\_IPV6\_TUNNEL**  
**CONFIG\_IPV6\_MULTIPLE\_TABLES**  
**CONFIG\_IPV6\_ADVANCED\_ROUTER**

Suport únicament per al Mobile Node:

**CONFIG\_IPV6\_SUBTREES**  
**CONFIG\_ARPD**

Suport per a IPSec (IP Secure):

**CONFIG\_INET6\_ESP**  
**CONFIG\_NET\_KEY**  
**CONFIG\_NET\_KEY\_MIGRATE**

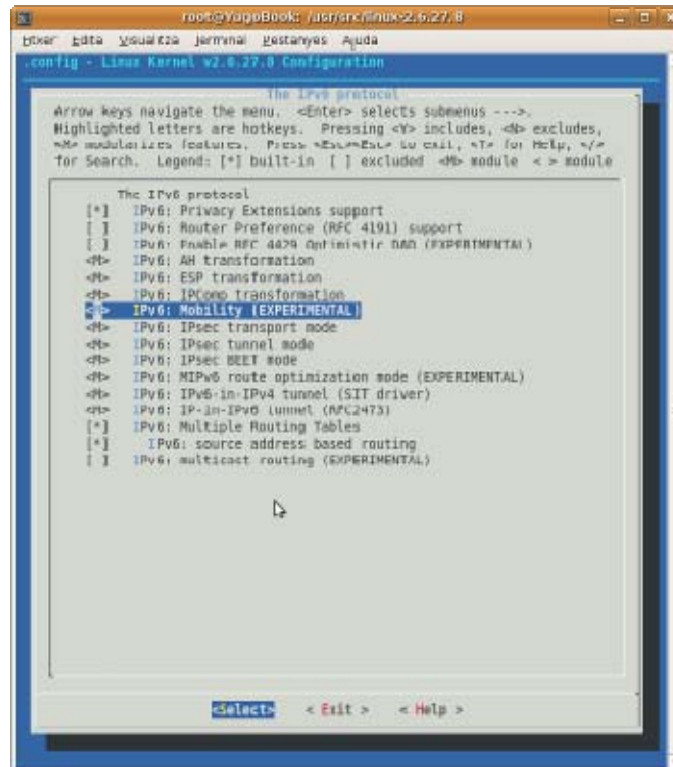


Fig. III.6. Activació del mòdul MIPv6

És possible que molts d'aquests símbols ja estiguin activats per defecte a la instal·lació bàsica del sistema. Com a exemple, en el cas d'aquest estudi i partint d'una instal·lació Ubuntu 8.04 amb kernel 2.6.27-7, només ha calgut activar els símbols següents:

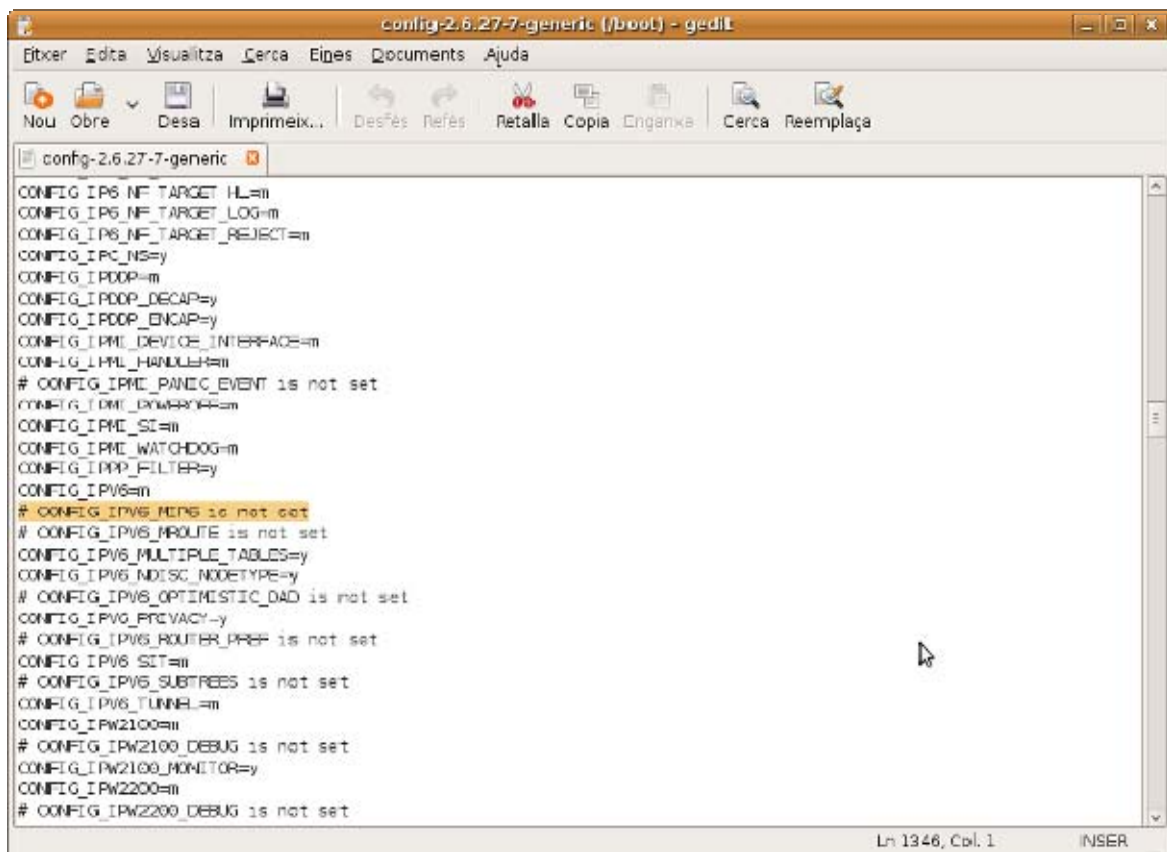
**CONFIG\_IPV6\_SUBTREES**  
**CONFIG\_IPV6\_MIP6**  
**CONFIG\_XFRM\_SUB\_POLICY**  
**CONFIG\_NET\_KEY\_MIGRATE**  
**CONFIG\_ARPD**

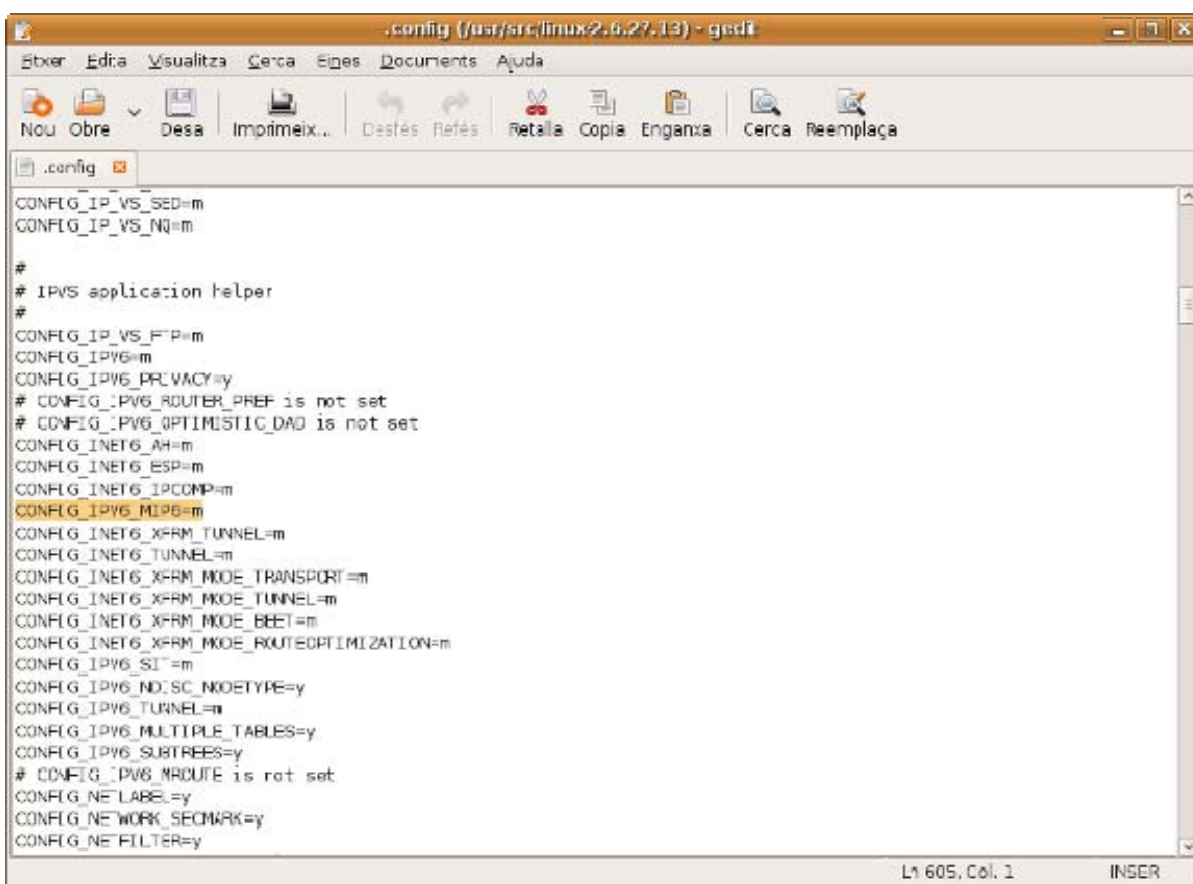
10. Una vegada configurat el nou kernel, cal sortir del menú i dir que es volen guardar els canvis:



**Fig. III.7.** Menú per a guardar la configuració

En aquest punt, cal comprovar si el fitxer `.config` que s'ha guardat al directori `/usr/src/linux-source[nou kernel]` té els símbols requerits activats:



**Fig. III.8.** Fitxer de configuració abans de la modificació del kernel

```
CONFIG_IP_VS_SED=m
CONFIG_IP_VS_NQ=m

#
# IPVS application helper
#
CONFIG_IP_VS_FIP=m
CONFIG_IPV6=m
CONFIG_IPV6_PRIVACY=y
# CONFIG_IPV6_ROUTER_PREF is not set
# CONFIG_IPV6_OPTIMISTIC_DAD is not set
CONFIG_INET6_AH=m
CONFIG_INET6_ESP=m
CONFIG_INET6_IPCOMP=m
CONFIG_IPV6_MIP6=m
CONFIG_INET6_XFRM_TUNNEL=m
CONFIG_INET6_TUNNEL=m
CONFIG_INET6_XFRM_MODE_TRANSPORT=m
CONFIG_INET6_XFRM_MODE_TUNNEL=m
CONFIG_INET6_XFRM_MODE_BEET=m
CONFIG_INET6_XFRM_MODE_ROUTEOPTIMIZATION=m
CONFIG_IPV6_SIT=m
CONFIG_IPV6_NDISC_NDOETYPE=y
CONFIG_IPV6_TUNNEL=m
CONFIG_IPV6_MULTIPLE_TABLES=y
CONFIG_IPV6_SUBTREES=y
# CONFIG_IPV6_MROUTE is not set
CONFIG_NET_LABEL=y
CONFIG_NETWORK_SECMARK=y
CONFIG_NETFILTER=y
```

**Fig. III.9.** Fitxer de configuració després de la modificació del kernel

11. Seguidament, s'executen les ordres següents:

**#make**

**#make modules**

**#make modules-install**

**#make install**

```

root@YugoBook: /usr/src/linux-2.6.27.13
Etter  edita  Visualitzat  Ferrina  Bastanyes  Ajuda
INSTALL /lib/firmware/korg/k1212.dsp
INSTALL /lib/firmware/ess/m maestro3_ assp_kernel.fw
INSTALL /lib/firmware/ess/m maestro3_ assp_minisrc.fw
INSTALL /lib/firmware/yamaha/ds1_cir1.fw
INSTALL /lib/firmware/yamaha/ds1_dsp.fw
INSTALL /lib/firmware/yamaha/ds1e_ctl.fw
INSTALL /lib/firmware/eni26/loader.fw
INSTALL /lib/firmware/eni26/firmware.fw
INSTALL /lib/firmware/eni26/bitstream.fw
INSTALL /lib/firmware/eni62/loader.fw
INSTALL /lib/firmware/eni62/bitstream.fw
INSTALL /lib/firmware/eni62/spdif.fw
INSTALL /lib/firmware/eni62/aid1.fw
INSTALL /lib/firmware/kaweth/new_code.bin
INSTALL /lib/firmware/kaweth/trigger_code.bin
INSTALL /lib/firmware/kaweth/new_code_fix.bin
INSTALL /lib/firmware/kaweth/trigger_code_fix.bin
INSTALL /lib/firmware/ti_3419.fw
INSTALL /lib/firmware/ti_5852.fw
INSTALL /lib/firmware/edgeport/boot.fw
INSTALL /lib/firmware/edgeport/boot2.fw
INSTALL /lib/firmware/edgeport/down.fw
INSTALL /lib/firmware/edgeport/down2.fw
INSTALL /lib/firmware/edgeport/down3.bin
INSTALL /lib/firmware/whiteheat_loader.fw
INSTALL /lib/firmware/whiteheat.fw
INSTALL /lib/firmware/keyspan_pda/keyspan_pda.fw
INSTALL /lib/firmware/keyspan_pda/xircan_pgs.fw
INSTALL /lib/firmware/vicam/firmware.fw
INSTALL /lib/firmware/cpia2/slv8672_vp4.bin
INSTALL /lib/firmware/sb16/mulaw_main.csp
INSTALL /lib/firmware/sb16/alaw_main.csp
INSTALL /lib/firmware/sb16/ina_adpcm_init.csp
INSTALL /lib/firmware/sb16/ina_adpcm_playback.csp
INSTALL /lib/firmware/sb16/ina_adpcm_capture.csp
DEPMOD 2.6.27.13
root@YugoBook: /usr/src/linux-2.6.27.13# make install
sh /usr/src/linux-2.6.27.13/arch/x86/boot/install.sh 2.6.27.13 arch/x86/boot/bzImage
System.map */boot*
root@YugoBook: /usr/src/linux-2.6.27.13#

```

Fig. III.10. Execució de *make install*

12. Una vegada compilat i instal·lat el nou kernel, cal anar al directori `/boot` i comprovar que s'hagin creat tres nous fitxers amb la versió del nou kernel instal·lat:

**config-[versió kernel]**

**system.map-[versió kernel]**

**vmlinuz-[versió kernel]**



```

root@YugoBook: /boot
Etixer Edita Visualitza Terminal Pestanyes Ajuda
-rw-r--r-- 1 root root 2234224 2008-12-08 20:27 vmlinuz-2.6.27.2.old
-rw-r--r-- 1 root root 2212080 2008-12-08 15:32 vmlinuz-2.6.27.7
-rw-r--r-- 1 root root 2244272 2008-10-24 10:29 vmlinuz-2.6.27-7-generic
-rw-r--r-- 1 root root 2216208 2009-02-02 18:19 vmlinuz-2.6.27.8
-rw-r--r-- 1 root root 2216208 2009-02-02 18:18 vmlinuz-2.6.27.8.old
root@YugoBook:/boot# mkinitramfs -o initrd.img-2.6.27.13 2.6.27.13
root@YugoBook:/boot# ls -l
total 2452/6
-rw-r--r-- 1 root root 8182432 2008-12-08 16:01 2.6.27
-rw-r--r-- 1 root root 507665 2008-10-24 10:29 abi-2.6.27-7-generic
-rw-r--r-- 1 root root 93893 2009-02-02 20:35 config-2.6.27.13
-rw-r--r-- 1 root root 96473 2008-12-08 21:52 config-2.6.27.2
-rw-r--r-- 1 root root 96528 2008-12-08 20:27 config-2.6.27.2.old
-rw-r--r-- 1 root root 93958 2008-12-08 15:32 config-2.6.27.7
-rw-r--r-- 1 root root 91364 2008-10-24 10:29 config-2.6.27-7-generic
-rw-r--r-- 1 root root 93892 2009-02-02 18:19 config-2.6.27.8
-rw-r--r-- 1 root root 93892 2009-02-02 18:18 config-2.6.27.8.old
drwxr-xr-x 2 root root 4096 2009-02-02 18:25 grub
-rw-r--r-- 1 root root 52622891 2009-02-02 20:31 initrd.img-2.6.27.13
-rw-r--r-- 1 root root 52610650 2008-12-08 22:16 initrd.img-2.6.27.2
-rw-r--r-- 1 root root 52613078 2008-12-08 20:30 initrd.img-2.6.27.7
-rw-r--r-- 1 root root 8182016 2008-11-25 13:06 initrd.img-2.6.27-7-generic
-rw-r--r-- 1 root root 52631271 2009-02-02 18:20 initrd.img-2.6.27.8
-rw-r--r-- 1 root root 124152 2008-09-11 22:11 memtest86+.bin
-rw-r--r-- 1 root root 1000092 2009-02-02 20:35 System.map-2.6.27.13
-rw-r--r-- 1 root root 1007528 2008-12-08 21:52 System.map-2.6.27.2
-rw-r--r-- 1 root root 1006489 2008-12-08 20:27 System.map-2.6.27.2.old
-rw-r--r-- 1 root root 998421 2008-12-08 15:32 System.map-2.6.27.7
-rw-r--r-- 1 root root 1029585 2008-10-24 10:29 System.map-2.6.27-7-generic
-rw-r--r-- 1 root root 999832 2009-02-02 18:19 System.map-2.6.27.8
-rw-r--r-- 1 root root 999832 2009-02-02 18:18 System.map-2.6.27.8.old
-rw-r--r-- 1 root root 1073 2008-10-24 10:31 vmcoreinfo-2.6.27-7-generic
-rw-r--r-- 1 root root 2217328 2009-02-02 20:35 vmlinuz-2.6.27.13
-rw-r--r-- 1 root root 2236080 2008-12-08 21:52 vmlinuz-2.6.27.2
-rw-r--r-- 1 root root 2234224 2008-12-08 20:27 vmlinuz-2.6.27.2.old
-rw-r--r-- 1 root root 2212080 2008-12-08 15:32 vmlinuz-2.6.27.7
-rw-r--r-- 1 root root 2244272 2008-10-24 10:29 vmlinuz-2.6.27-7-generic
-rw-r--r-- 1 root root 2216208 2009-02-02 18:19 vmlinuz-2.6.27.8
-rw-r--r-- 1 root root 2216208 2009-02-02 18:18 vmlinuz-2.6.27.8.old
root@YugoBook: /boot#

```

Fig. III.11. Llista de fitxers creats

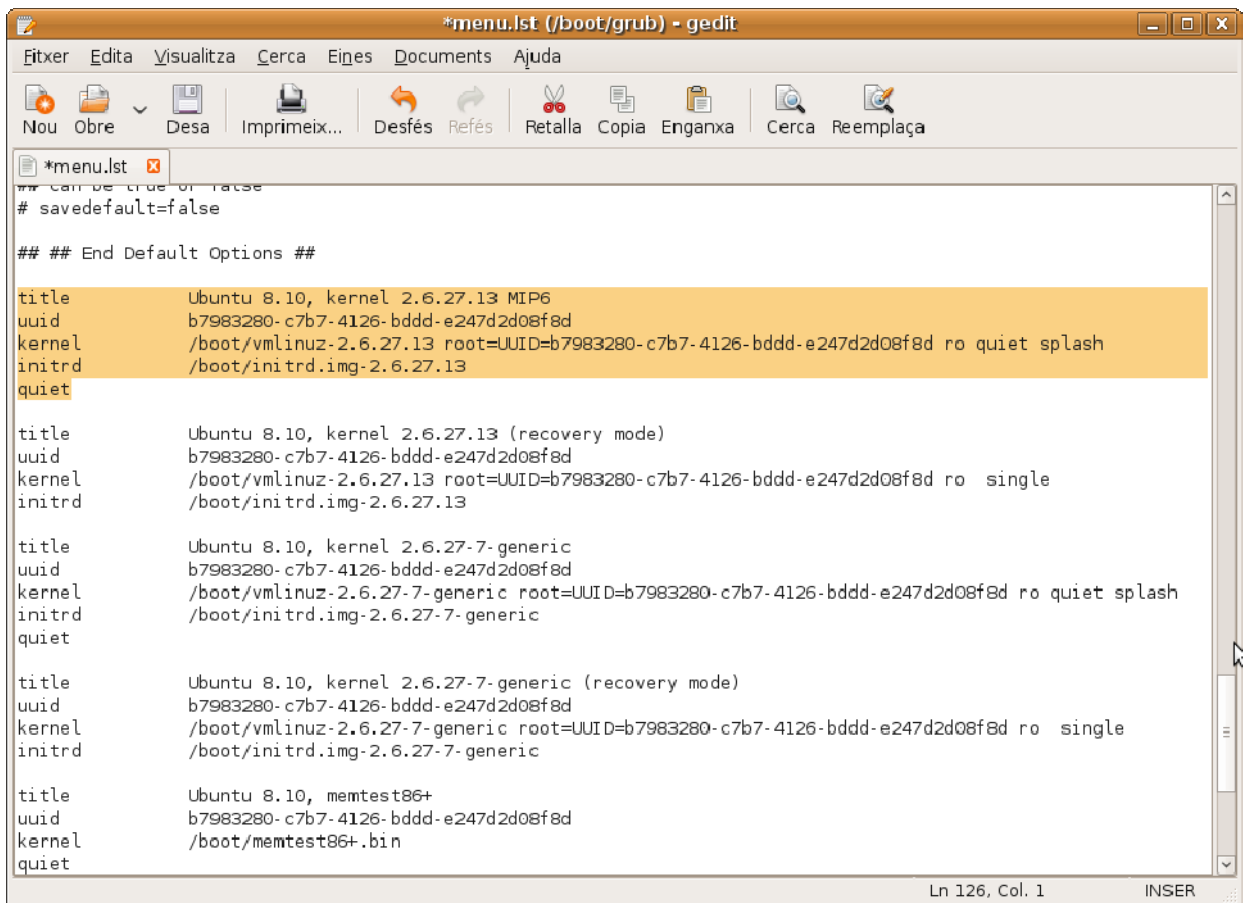
13. Si es disposa d'un kernel superior a la versió 2.6.16, cal executar l'ordre següent:

**#mkinitramfs -o initrd.img-[versió del kernel] [versió del kernel]**

O bé, si el kernel és inferior a la versió 2.6.16:

**#mkinitrd -o initrd.img[versió del kernel] [versió del kernel]**

14. Finalment, només cal editar el fitxer d'inici de GRUB situat a */boot/grub/menu.lst* per a carregar a l'inici el kernel amb suport per a MIPv6:



**Fig. III.12.** GRUB modificat

Aquest procés es pot fer automàticament amb l'ordre següent:

### #update-grub

15.A continuació, s'edita el fitxer (*/etc/rc.local*) i s'hi afegeixen les línies següents:

```

sudo insmod /lib/modules/2.6.27.13/wlan/wl.ko
sudo insmod /lib/modules/2.6.27.13/kernel/net/ipv6/tunnel6.ko
sudo insmod /lib/modules/2.6.27.13/kernel/net/ipv6/ip6_tunnel.ko
sudo insmod /lib/modules/2.6.27.13/kernel/net/ipv6/ah6.ko
sudo insmod /lib/modules/2.6.27.13/kernel/net/ipv6/esp6.ko
sudo insmod /lib/modules/2.6.27.13/kernel/net/ipv6/ipcomp6.ko
sudo insmod /lib/modules/2.6.27.13/kernel/net/ipv6/ipv6.ko
sudo insmod /lib/modules/2.6.27.13/kernel/net/ipv6/mip6.ko
sudo insmod /lib/modules/2.6.27.13/kernel/net/ipv6/sit.ko
sudo insmod /lib/modules/2.6.27.13/kernel/net/ipv6/xfrm6_mode_beet.ko
sudo insmod /lib/modules/2.6.27.13/kernel/net/ipv6/xfrm6_mode_ro.ko
sudo insmod /lib/modules/2.6.27.13/kernel/net/ipv6/xfrm6_mode_transport.ko
sudo insmod /lib/modules/2.6.27.13/kernel/net/ipv6/xfrm6_mode_tunnel.ko
sudo insmod /lib/modules/2.6.27.13/kernel/net/ipv6/xfrm6_tunnel.ko
sudo insmod /lib/modules/2.6.27.13/kernel/net/key/af_key.ko

```

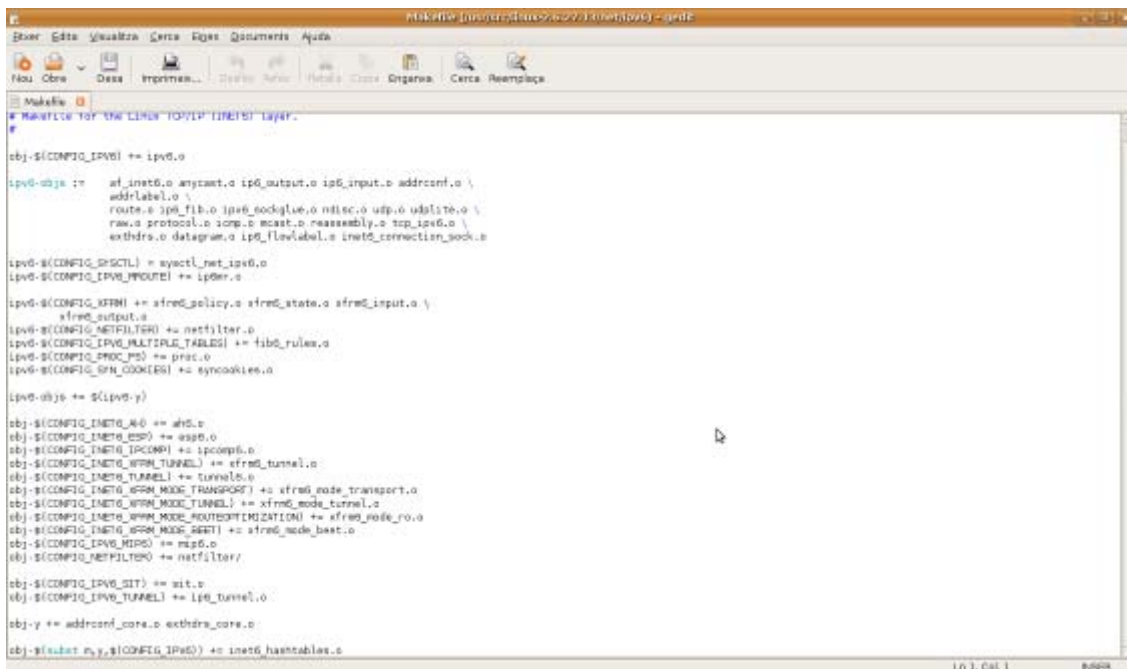
```
sudo insmod /lib/modules/2.6.27.13/kernel/net/xfrm/xfrm_ipcomp.ko
sudo insmod /lib/modules/2.6.27.13/kernel/net/xfrm/xfrm_user.ko
```

Cal tenir en compte que aquest fitxer serà diferent en funció dels mòduls que es necessitin carregar per a cada configuració i de la versió del kernel que s'instal·li.

Perquè tingui efecte, cal que al final del fitxer hi hagi la línia següent:

```
exit(0)
```

Per a saber com són els fitxers \*.o de sortida per a carregar els mòduls, n'hi ha prou d'editar els fitxers *makefile* corresponents del codi font (*/usr/src/[versió del kernel]*) i veure quin nom se'ls dona.



```
Makefile [usr/src/linux-2.6.27.13/net/xfrm] - gvim
# Makefile for the LINUX TOP/VP/INETS/layer.
#
obj-$(CONFIG_IPV6) += ipv6.o

ipv6-objs := af_inet6.o anycast.o ipv6_output.o ipv6_input.o addrconf.o \
            addrlabel.o \
            route.o ip6_fib.o ip6_sockglue.o nhsec.o udp.o udplite.o \
            raw.o protocol.o icmp.o mcast.o reassembly.o tcp_ipv6.o \
            exthdrs.o datagram.o ip6_flowlabel.o inet6_connection_sock.o

ipv6-$(CONFIG_SHOCTL) += syactl_net_ipv6.o
ipv6-$(CONFIG_IPV6_MROUTE) += ipdr.o

ipv6-$(CONFIG_XFRM) += xfrm_policy.o xfrm_state.o xfrm_input.o \
                    xfrm_output.o
ipv6-$(CONFIG_NETFILTER) += netfilter.o
ipv6-$(CONFIG_IPV6_MULTIPLE_TABLES) += fibs_rules.o
ipv6-$(CONFIG_IPV6_MROUTE) += mroute.o
ipv6-$(CONFIG_IPV6_MROUTE_MULTIPLE_TABLES) += mroute_multi.o
ipv6-$(CONFIG_IPV6_MROUTE_MULTIPLE_TABLES) += mroute_multi.o
ipv6-objs += $(ipv6-y)

obj-$(CONFIG_INET6_AH) += ah.o
obj-$(CONFIG_INET6_ESP) += esp.o
obj-$(CONFIG_INET6_IPCOMP) += ipcomp.o
obj-$(CONFIG_INET6_XFRM_TUNNEL) += xfrm_tunnel.o
obj-$(CONFIG_INET6_TUNNEL) += tunnel.o
obj-$(CONFIG_INET6_XFRM_MODE_TRANSPORT) += xfrm_mode_transport.o
obj-$(CONFIG_INET6_XFRM_MODE_TUNNEL) += xfrm_mode_tunnel.o
obj-$(CONFIG_INET6_XFRM_MODE_ROUTEBYPASS) += xfrm_mode_router.o
obj-$(CONFIG_INET6_XFRM_MODE_SIT) += xfrm_mode_sit.o
obj-$(CONFIG_IPV6_MIP6) += mip6.o
obj-$(CONFIG_NETFILTER) += netfilter.o

obj-$(CONFIG_IPV6_SIT) += sit.o
obj-$(CONFIG_IPV6_TUNNEL) += ip6_tunnel.o

obj-y += addrconf_core.o exthdrs_core.o

obj-$(subst %,y,$(CONFIG_IPV6)) += inet6_hashtables.o
```

Fig. III.13. Fitxer *makefile* del mòdul

16. Finalment, després de seguir tots els passos anteriors i de reiniciar la màquina amb el nou kernel, cal executar l'ordre següent per a carregar els mòduls corresponents a MIPv6:

```
#sh /etc/rc.local
```

A partir d'aquest punt ja es pot començar a instal·lar el dimoni MIPv6.

### III.3.2. Instal·lació del dimoni MIPv6

Tal com s'ha explicat a l'Annex I, s'ha triat finalment el dimoni d'USAGI, anomenat UMIP, per a executar el protocol MIPv6.

En primer lloc, cal descarregar el codi font del dimoni des de la seva pàgina web:

<http://umip.linux-ipv6.org/>

i descomprimir-ne el contingut a qualsevol carpeta:

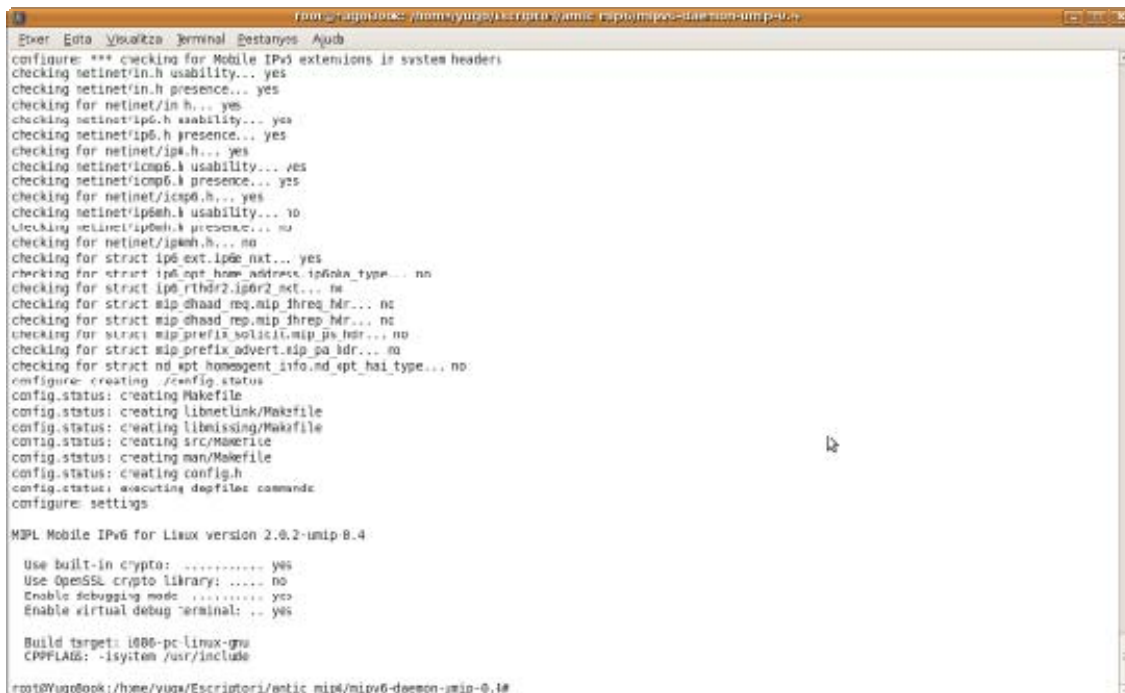
**#gunzip mipv6-daemon-umip-0.4.tar.gz**

S'ha de tenir en compte que només cal instal·lar aquest protocol en les màquines CN, MN i HA. Per tal de no trobar cap mena de problema durant la compilació i instal·lació, cal haver instal·lat totes les biblioteques especificades al primer pas de l'apartat III.3.1 d'aquest annex.

Seguidament, cal aplicar els canvis necessaris per a donar suport a nodes dormants i paging, tal com es detalla a l'Annex VII.

A continuació, cal situar-se a la carpeta on s'ha descomprimit el dimoni i executar l'script `./configure` per a passar-li com a paràmetre la ubicació del codi font del kernel que s'ha compilat anteriorment i si es vol instal·lar el servidor local per a visualitzar la BUL (*Binding Update List*):

**#CPPFLAGS='-isystem /usr/include' ./configure --enable-vt**



```

root@yugbook: /home/yugbook/Entic/mip6/mipv6-daemon-umip-0.4
Ever Eota Visualta Terminal Destanyos Ajuda
configure: *** checking for Mobile IPv6 extensions in system headers
checking netinet/in.h usability... yes
checking netinet/in.h presence... yes
checking for netinet/in.h... yes
checking netinet/ip6.h usability... yes
checking netinet/ip6.h presence... yes
checking for netinet/ip6.h... yes
checking netinet/ip6.h usability... yes
checking netinet/ip6.h presence... yes
checking for netinet/ip6.h... yes
checking netinet/icmp6.h usability... yes
checking netinet/icmp6.h presence... yes
checking for netinet/icmp6.h... yes
checking netinet/ip6m.h usability... no
checking netinet/ip6m.h presence... no
checking for netinet/ip6m.h... no
checking for struct ip6_ext_ip6e_nxt... yes
checking for struct ip6_opt_hop_address_ip6oat_type... no
checking for struct ip6_rthdr2_ip6r2_sel... no
checking for struct mip_dhaddr_req_mip_dhreq_hdr... no
checking for struct mip_dhaddr_req_mip_dhreq_hdr... no
checking for struct mip_prefix_solicit_mip_ps_hdr... no
checking for struct mip_prefix_advert_mip_pa_hdr... no
checking for struct nd_spt_homesegnt_info_nd_spt_hai_type... no
configure: creating ./config.status
config.status: creating Makefile
config.status: creating libnetlink/Makefile
config.status: creating libmissing/Makefile
config.status: creating src/Makefile
config.status: creating man/Makefile
config.status: creating config.h
config.status: executing depfiles commands
configure: settings

MPL Mobile IPv6 for Linux version 2.0.2-umip-0.4

Use built-in crypto: ..... yes
Use OpenSSL crypto library: ..... no
Enable debugging mode: ..... yes
Enable virtual debug terminal: .. yes

Build target: i866-pc-linux-gnu
CPPFLAGS: -isystem /usr/include

root@yugbook:/home/yugbook/Entic/mip6/mipv6-daemon-umip-0.4#

```

**Fig. III.14.** Execució de `configure`

Una vegada configurat, i en cas que es tracti del codi modificat en el projecte, s'executen les ordres següents dins del directori. En cas contrari, es passa directament a executar l'ordre **#make**:

**#cd src**

**#gedit Makefile**

i a cada lloc on hi hagi la paraula *gcc* s'afegeix al darrere *-lpcap*. Aquesta ordre permet donar suport al codi de les biblioteques *libpcap* que s'utilitzen en algunes de les funcions creades per a donar suport de nodes dormants i paging.

```
CC = gcc -lpcap
CCDEPMODE = depmode=gcc3 -lpcap
CPP = gcc -E -lpcap
ac_ct_CC = gcc -lpcap
```

Una vegada editat, s'executen les ordres següents:

**#cd ..**

**#make**

Per a compilar el codi.

**#make install**

Per a instal·lar el dimoni al sistema.

### III.3.2.1. Possibles errors d'instal·lació

Si durant la compilació es produeixen errors amb el fitxer *scan*, caldrà esborrar qualsevol referència d'aquest fitxer al Makefile del directori */src* (per a la versió 0.4 del protocol).

Si durant la compilació es produeixen errors amb el fitxer *conf\_changes*, caldrà esborrar qualsevol referència d'aquest fitxer al Makefile del directori */src* (per a la versió 0.4.14 del protocol).

Si durant la compilació s'obté l'error següent:

```
init.Tpo -c -o inet6_rth_init.o inet6_rth_init.c
inet6_rth_init.c: In function , inet6_rth_init:
inet6_rth_init.c:26: error: , "NULL" undeclared (first use in this function)
inet6_rth_init.c:26: error: (Each undeclared identifier is reported only once
inet6_rth_init.c:26: error: for each function it appears in.)
```

caldrà editar els fitxers següents dins del directori */libmissing*:

**-inet6\_rth\_getaddr.c**

`-inet6_rth_init.c`  
`-inet6_opt_find.c`

i afegir a cadascun d'ells, a la declaració de les capçaleres, la definició següent:

**#define NULL 0**

Qualsevol altre tipus d'error que es pugui trobar durant la instal·lació estarà vinculat a un error en les biblioteques necessàries per a fer la compilació correctament .

### III.3.3. Configuració del dimoni mip6d

Tal com s'ha apuntat en aquest annex, aquesta versió del dimoni de MIPv6 permet modificar els paràmetres de configuració des de l'espai d'usuari, sense haver de recompilar el kernel cada vegada.

Per a fer aquestes modificacions, s'utilitza el fitxer de configuració *mip6d.conf* que es troba a a */usr/local/etc/*. Aquest fitxer conté paràmetres que permeten configurar cadascun dels nodes que utilitza el protocol (CN, MN i HA).

**Taula III.2.** Configuració del dimoni MIPv6

Paràmetres	Opcions	Descripció
<b>Paràmetres comuns</b>		
<i>NodeConfig</i>	CN   HA   MN	Rol del node dins la xarxa
<i>DebugLevel</i>	0 - 10	Detall de debug. A partir de 3 és el màxim
<i>DoRouteOptimizationCN</i>	Enabled   Disabled	Activa el mecanisme de RO amb els MN
<b>Paràmetres comuns a l'HA i a l'MN</b>		
<i>Interface</i>	Ethx MnIfPreference IfType	Interfície per la qual funciona el protocol
<i>UseMnHaIPsec</i>	Enabled   Disabled	Activa l'opció d'IPSec entre l'HA i l'MN
<i>KeyMngMobCapability</i>	Enabled   Disabled	Activa l'opció de claus dinàmiques per a protegir la senyalització
<i>IPsecPolicySet</i>	HomeAgentAddress HomeAddress IPsecPolicy	Defineix la política de segurat per a l'HA i l'HoA
<b>Paràmetres de l'HA</b>		
<i>HaMaxBindingLife</i>	Número (segons)	Limita el temps màxim d'un BU per l'MN



<i>SendMobPfxAdvs</i>	Enabled   Disabled	Activa l'enviament dels missatges Mobile Prefix Advertisement a l'MN
<i>SendUnsolMobPfxAdvs</i>	Enabled   Disabled	Activa l'enviament dels MP Advertisement periòdicament.
<i>MinMobPfxAdvInterval</i>	Número (segons)	Defineix l'interval mínim entre paquets MPA
<i>MaxMobPfxAdvInterval</i>	Número (segons)	Defineix l'interval màxim entre paquets MPA
<i>BindingAclPolicy address</i>	Allow   Deny	Defineix el comportament de l'HA respecte una adreça concreta d'un MN sobre l'acceptació del seu BU
<i>DefaultBindingAclPolicy</i>	Allow   Deny	Defineix el comportament de l'HA respecte els MN sobre l'acceptació del seu BU
<b>Paràmetres de l'MN</b>		
<i>MnMaxHaBindingLife</i>	Número (segons)	Temps màxim durant el qual són vàlids els BU amb l'HA
<i>MnMinHaBindingLife</i>	Número (segons)	Temps mínim durant el qual són vàlids els BU amb l'HA
<i>MnDiscardHaParamProb</i>	Enabled   Disabled	Ignora paquets ICMPv6 per a evitar atacs DoS
<i>SendMobPfxSol</i>	Enabled   Disabled	Activa la transmissió de l'MP solicitation cap a l'HA
<i>DoRouteOptimizationMN</i>	Enabled   Disabled	Activa el mecanisme d'RO amb el CN
<i>MnUseAllInterfaces</i>	Enabled   Disabled	Indica que totes les interfícies s'utilitzen per mobilitat
<i>UseCnBuAck</i>	Enabled   Disabled	Indica si el bit ACK dels BU s'utilitzarà amb els CN
<i>MnRouterProbes</i>	Nombre de vegades	Fixa el nombre de proves amb NUD que s'enviaran abans d'assumir que s'ha perdut connectivitat amb el router
<b>Paràmetres de l'MN</b>		
<i>MnRouterProbeTimeout</i>	Número (segons)	Fixa el temps durant el qual es realitzarà el

		NUD
<i>MnHomeLink "name"</i>	HomeAddress HomeAgentAddress MnRoPolicy	Informació de l'HA, la seva HoA i si utilitza o no la política d'RO

### III.3.4. Instal·lació del dimoni Radvd

Els nodes del demostrador que actuen com a routers (PA, AR1, AR2 i HA) han d'anunciar-se a la xarxa. Concretament, han d'anunciar el seu prefix de xarxa, la seva IP i la seva interfície mitjançant paquets ICMPv6 anomenats Router Advertisements.

Així, les màquines que es connectin a les interfícies d'aquests nodes poden autoconfigurar la seva adreça IPv6 mitjançant el prefix de xarxa anunciat i un identificador d'interfície generat a partir de la seva adreça MAC.

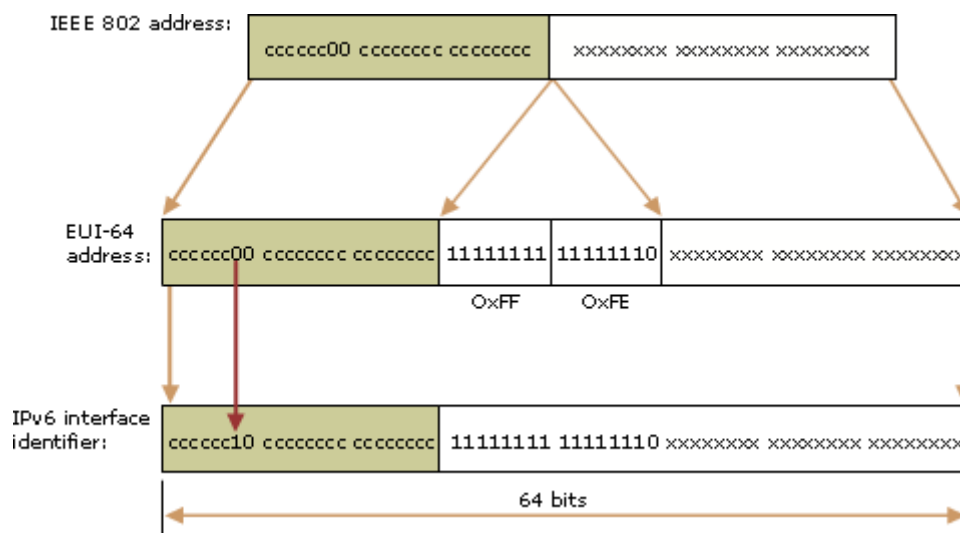


Fig. III.15. Formació de l'adreça IPv6 amb MAC

:

Per a donar una adreça *Local-Link* a una interfície, cal aplicar l'ordre següent:

```
#ifconfig ethx add fe80::2GG:HHFF:FE:IIJJ:KKLL/64
```

Tenint en compte que la MAC és GG:HH:II:JJ:KK:LL.

En Linux, el dimoni Radvd (*Router ADvertisement Daemon*) és l'encarregat de difondre aquests missatges periòdicament o quan són sol·licitats per un missatge *router solicitation*.

Per a instal·lar el dimoni Radvd en Ubuntu, cal executar l'ordre següent com a root:



## #apt-get install radvd

Una vegada instal·lat, podem configurar-ne els paràmetres des del fitxer *radvd.conf* situat a */etc/*:

```
Interface ethx      //Interfície per la qual es difondran els RtrAdv
{
AdvSendAdvert on;   //activa l'enviament de missatges
MinRtrAdvInterval 3; //interval mínim entre missatges
MaxRtrAdvInterval 10; //interval màxim entre missatges
AdvHomeAgentFlag on; //indica que es pot fer activa l'opció de MIPv6
AdvHomeAgentInfo on; // s'envia la informació de l'HA
Prefix a:b:c:d::/x //prefix de xarxa que s'anunciarà
{
AdvOnLink on;      //indica que el prefix es pot utilitzar per a determinar
                  //l'enllaç
AdvAutonomous on; //indica que el prefix es pot utilitzar per a
                  //autoconfigurar adreces
AdvRouterAddr on; //activa la difusió de l'adreça IPv6
};
};
```

Es recomana que els intervals d'enviament de missatges RA màxim i mínim siguin petits si es tracta d'una maqueta MIPv6, ja que això facilita el traspàs entre xarxes.

Una vegada configurat, es pot activar el dimoni amb les ordres següents:

```
#radvd
```

```
#service radvd start
```

Per a aturar el dimoni, el més recomanable és utilitzar l'ordre següent:

```
#killall radvd
```

S'aconsella que a l'hora d'executar aquest dimoni s'utilitzi l'ordre següent, ja que mostrarà per pantalla els missatges de debug:

### **#radvd -m stderr**

Per tal que el dimoni arrenqui, és imprescindible que cadascuna de les interfícies que anunciïn els Router Advertisements tinguin configurades la seva adreça IPv6 global i local.

### **III.3.5. Configuració de rutes estàtiques**

Per a simplificar el demostrador, s'han utilitzat rutes estàtiques entre els nodes "fixos" per a crear taules d'enrutament i tenir connectivitat entre tots ells.

Per a fer-ho s'ha utilitzat l'ordre **route**, que permet modificar les taules d'enrutament tant per a IPv4 com per a IPv6.

En aquest cas, l'ordre serà la següent:

```
#route -A inet6 add 3ffe:100:0:x::/64 gw 3ffe:100:0:x:f100:0:0:x dev ethx
```

Aquesta ordre indica que si es vol comunicar amb una adreça amb prefix *3ffe:100:0:x*, caldrà enrotar-ne els paquets cap a l'adreça definida després del paràmetre *gw* i utilitzant la interfície de xarxa definida després de *dev*.

Per a cada node, s'hauran d'afegir les rutes cap a les xarxes que no estiguin directament connectades a les seves interfícies. En aquest cas, només els nodes CN, HA, AR1 i AR2 necessitaran aquesta configuració.

Per a visualitzar les rutes afegides a un node cal utilitzar l'ordre següent:

```
#route -n -A inet6
```

### **III.4. Configuració dels nodes del demostrador MIPv6**

Una vegada explicats els diversos programaris que s'utilitzen per a posar a punt la maqueta, es defineix per a cadascun d'ells la seva configuració única per tal de reproduir, si calgués, el demostrador de manera exacta.

A mode de resum, es pot utilitzar la taula següent per a saber quina configuració cal aplicar a cadascun dels nodes del demostrador:

**Taula III. 3.** Resum de la configuració per a la maqueta MIPv6

Nodes	Router	Forward	radvd	mip6d	Rutes estàtiques
MN				X	
HA	X	X	X	X	X
CN				X	X
AR1	X	X	X		X
AR2	X	X	X		X
PA	X	X			X

S'aconsella incloure cadascuna de les ordres descrites en els apartats III.4.1, III.4.2, III.4.3 i III.4.4 en el fitxer */etc/rc.local* per a cadascuna de les màquines.

D'aquesta manera, cada vegada que s'engegui la màquina només caldrà executar l'ordre següent per a tenir-les configurades correctament:

```
#sh /etc/rc.local
```

### III.4.1. Configuració dels node PA i dels routers AR1 i AR2

Per tal d'actuar com a router IPv6, els nodes AR1 i AR2 s'han de configurar de manera diferent a com es fa amb els altres nodes.

#### III.4.1.1. Part comuna

Ordres perquè actuïn com a router i facin forwarding:

```
# echo "1" > /proc/sys/net/ipv6/conf/all/forwarding
# echo "0" > /proc/sys/net/ipv6/conf/all/autoconf
# echo "0" > /proc/sys/net/ipv6/conf/all/accept_ra
# echo "0" > /proc/sys/net/ipv6/conf/all/accept_redirects
```

#### III.4.1.2. Configuració del PA

Configurar l'adreça IPv6 global i local (tal com s'explica a l'apartat III.3.4 d'aquest annex).

```
#Ifconfig eth0 up
#Ifconfig eth1 up
#Ifconfig eth2 up
#Ifconfig eth3 up
#ifconfig eth0 add 3ffe:100:0:3:f100:0:0:8/64
#ifconfig eth1 add 3ffe:100:0:2:f100:0:0:6/64
#ifconfig eth2 add 3ffe:100:0:1:f100:0:0:4/64
```

```
#ifconfig eth3 add 3ffe:100:0:0:f100:0:0:2/64
```

Rutes estàtiques.

```
#route -A inet6 add 3ffe:100:0:6::/64 gw 3ffe:100:0:3:f100::7 dev eth0
#route -A inet6 add 3ffe:100:0:5::/64 gw 3ffe:100:0:2:f100::5 dev eth1
```

### III.4.1.3. Configuració de l'AR1

Configurar l'adreça IPv6 global i local (tal com s'explica a l'apartat III.3.4 d'aquest annex).

```
#Ifconfig eth0 up
#Ifconfig eth1 up
#Ifconfig eth0 add 3ffe:100:0:5:f100:0:0:11/64
#Ifconfig eth1 add 3ffe:100:0:2:f100:0:0:5/64
```

Configurar l'adreça IPv4 a la interfície connectada a l'AP per a tenir connectivitat.

```
#Ifconfig eth0 147.83.118.169 netmask 255.255.255.0
```

Rutes estàtiques.

```
#route -A inet6 add 3ffe:100:0:1::/64 gw 3ffe:100:0:2:f100::6 dev eth1
#route -A inet6 add 3ffe:100:0:0::/64 gw 3ffe:100:0:2:f100::6 dev eth1
#route -A inet6 add 3ffe:100:0:3::/64 gw 3ffe:100:0:2:f100::6 dev eth1
#route -A inet6 add 3ffe:100:0:6::/64 gw 3ffe:100:0:2:f100::6 dev eth1
```

Codi *radvd.conf*.

```
interface eth0
{
    AdvSendAdvert on;
    MinRtrAdvInterval 3;
    MaxRtrAdvInterval 10;
    AdvHomeAgentFlag off;
    prefix 3ffe:100:0:5::/64
    {
        AdvOnLink on;
        AdvAutonomous on;
        AdvRouterAddr on;
    };
};
```

### III.4.1.4. Configuració de l'AR2

Configurar l'adreça IPv6 global i local (tal com s'explica a l'apartat III.3.4 d'aquest annex).

```
#ifconfig eth0 up
#ifconfig eth2 up
#ifconfig eth0 add 3ffe:100:0:3:f100:0:0:7/64
#ifconfig eth2 add 3ffe:100:0:6:f100:0:0:12/64
```

Configurar l'adreça IPv4 a la interfície connectada a l'AP per a tenir connectivitat.

```
#ifconfig eth0 147.83.118.170 netmask 255.255.255.0
```

Rutes estàtiques.

```
#route -A inet6 add 3ffe:100:0:0::/64 gw 3ffe:100:0:3:f100::8 dev eth2
#route -A inet6 add 3ffe:100:0:1::/64 gw 3ffe:100:0:3:f100::8 dev eth2
#route -A inet6 add 3ffe:100:0:2::/64 gw 3ffe:100:0:3:f100::8 dev eth2
#route -A inet6 add 3ffe:100:0:5::/64 gw 3ffe:100:0:3:f100::8 dev eth2
```

Codi *radvd.conf*.

```
interface eth0
{
    AdvSendAdvert on;
    MinRtrAdvInterval 3;
    MaxRtrAdvInterval 10;
    AdvHomeAgentFlag off;
    prefix 3ffe:100:0:6::/64
    {
        AdvOnLink on;
        AdvAutonomous on;
        AdvRouterAddr on;
    };
};
```

### III.4.2. Configuració del Correspondent Node

Configurar l'adreça IPv6 global i local (tal com s'explica a l'apartat III.3.4 d'aquest annex).

```
#ifconfig eth0 up
#ifconfig eth0 add 3ffe:100:0:1:f100:0:0:3/64
```

Carregar els mòduls del kernel necessaris per al funcionament de MIPv6.

```
sudo insmod /lib/modules/2.6.27.13/wlan/wl.ko
sudo insmod /lib/modules/2.6.27.13/kernel/net/ipv6/tunnel6.ko
```

```

sudo insmod /lib/modules/2.6.27.13/kernel/net/ipv6/ip6_tunnel.ko
sudo insmod /lib/modules/2.6.27.13/kernel/net/ipv6/ah6.ko
sudo insmod /lib/modules/2.6.27.13/kernel/net/ipv6/esp6.ko
sudo insmod /lib/modules/2.6.27.13/kernel/net/ipv6/ipcomp6.ko
sudo insmod /lib/modules/2.6.27.13/kernel/net/ipv6/ipv6.ko
sudo insmod /lib/modules/2.6.27.13/kernel/net/ipv6/mip6.ko
sudo insmod /lib/modules/2.6.27.13/kernel/net/ipv6/sit.ko
sudo insmod /lib/modules/2.6.27.13/kernel/net/ipv6/xfrm6_mode_beet.ko
sudo insmod /lib/modules/2.6.27.13/kernel/net/ipv6/xfrm6_mode_ro.ko
sudo insmod /lib/modules/2.6.27.13/kernel/net/ipv6/xfrm6_mode_transport.ko
sudo insmod /lib/modules/2.6.27.13/kernel/net/ipv6/xfrm6_mode_tunnel.ko
sudo insmod /lib/modules/2.6.27.13/kernel/net/ipv6/xfrm6_tunnel.ko
sudo insmod /lib/modules/2.6.27.13/kernel/net/key/af_key.ko
sudo insmod /lib/modules/2.6.27.13/kernel/net/xfrm/xfrm_ipcomp.ko
sudo insmod /lib/modules/2.6.27.13/kernel/net/xfrm/xfrm_user.ko

```

Per al CN, cal configurar les rutes estàtiques següents:

```

#route -A inet6 add 3ffe:100:0:0::/64 gw 3ffe:100:0:1:f100::4 dev eth0
#route -A inet6 add 3ffe:100:0:3::/64 gw 3ffe:100:0:1:f100::4 dev eth0
#route -A inet6 add 3ffe:100:0:2::/64 gw 3ffe:100:0:1:f100::4 dev eth0
#route -A inet6 add 3ffe:100:0:5::/64 gw 3ffe:100:0:1:f100::4 dev eth0
#route -A inet6 add 3ffe:100:0:6::/64 gw 3ffe:100:0:1:f100::4 dev eth0

```

El fitxer de configuració *mip6d.conf* és el següent:

```

NodeConfig CN;
DebugLevel 10;
DoRouteOptimizationCN enabled;

```

### III.4.3. Configuració del Home Agent

Configurar l'adreça IPv6 global i local (tal com s'explica a l'apartat III.3.4 d'aquest annex).

```

ifconfig eth0 up
ifconfig eth0 add 3ffe:100:0:0:f100:0:0:1/64

```

Carregar els mòduls del kernel necessaris per al funcionament de MIPv6.

```

sudo insmod /lib/modules/2.6.27.13/wlan/wl.ko
sudo insmod /lib/modules/2.6.27.13/kernel/net/ipv6/tunnel6.ko
sudo insmod /lib/modules/2.6.27.13/kernel/net/ipv6/ip6_tunnel.ko
sudo insmod /lib/modules/2.6.27.13/kernel/net/ipv6/ah6.ko
sudo insmod /lib/modules/2.6.27.13/kernel/net/ipv6/esp6.ko

```

```

sudo insmod /lib/modules/2.6.27.13/kernel/net/ipv6/ipcomp6.ko
sudo insmod /lib/modules/2.6.27.13/kernel/net/ipv6/ipv6.ko
sudo insmod /lib/modules/2.6.27.13/kernel/net/ipv6/mip6.ko
sudo insmod /lib/modules/2.6.27.13/kernel/net/ipv6/sit.ko
sudo insmod /lib/modules/2.6.27.13/kernel/net/ipv6/xfrm6_mode_beet.ko
sudo insmod /lib/modules/2.6.27.13/kernel/net/ipv6/xfrm6_mode_ro.ko
sudo insmod /lib/modules/2.6.27.13/kernel/net/ipv6/xfrm6_mode_transport.ko
sudo insmod /lib/modules/2.6.27.13/kernel/net/ipv6/xfrm6_mode_tunnel.ko
sudo insmod /lib/modules/2.6.27.13/kernel/net/ipv6/xfrm6_tunnel.ko
sudo insmod /lib/modules/2.6.27.13/kernel/net/key/af_key.ko
sudo insmod /lib/modules/2.6.27.13/kernel/net/xfrm/xfrm_ipcomp.ko
sudo insmod /lib/modules/2.6.27.13/kernel/net/xfrm/xfrm_user.ko

```

Ordres perquè faci el *forward* correctament i el *nd\_proxy*.

```

echo 1 > /proc/sys/net/ipv6/conf/eth0/forwarding
echo 0 > /proc/sys/net/ipv6/conf/eth0/autoconf
echo 0 > /proc/sys/net/ipv6/conf/eth0/accept_ra
echo 0 > /proc/sys/net/ipv6/conf/eth0/accept_redirects
echo 1 > /proc/sys/net/ipv6/conf/all/forwarding
echo 1 > /proc/sys/net/ipv6/conf/all/proxy_ndp

```

Per a l'HA, cal configurar les rutes estàtiques següents:

```

route -A inet6 add 3ffe:100:0:1::/64 gw 3ffe:100:0:0:f100:0:0:2 dev eth0
route -A inet6 add 3ffe:100:0:2::/64 gw 3ffe:100:0:0:f100:0:0:2 dev eth0
route -A inet6 add 3ffe:100:0:3::/64 gw 3ffe:100:0:0:f100:0:0:2 dev eth0
route -A inet6 add 3ffe:100:0:4::/64 gw 3ffe:100:0:0:f100:0:0:2 dev eth0
route -A inet6 add 3ffe:100:0:5::/64 gw 3ffe:100:0:0:f100:0:0:2 dev eth0
route -A inet6 add 3ffe:100:0:6::/64 gw 3ffe:100:0:0:f100:0:0:2 dev eth0

```

El fitxer de configuració *mip6d.conf* és el següent:

```

NodeConfig HA;

## If set to > 0, will not detach from tty
DebugLevel 10;

## List of interfaces where we serve as Home Agent
Interface "eth0";

DoRouteOptimizationCN disabled;

##
## IPsec configuration
##

UseMnHaIPsec disabled;

```

```
## Key Management Mobility Capability
KeyMngMobCapability disabled;

HaMaxBindingLife 100;

MnMaxHaBindingLife 100;
```

El fitxer de configuració *radvd.conf* és el següent:

```
interface eth0
{
    AdvSendAdvert on;
    MinRtrAdvInterval 3;
    MaxRtrAdvInterval 10;
    #AdvIntervalOpt on;
    AdvHomeAgentFlag on;
    #AdvHomeAgentInfo on;
    #HomeAgentLifetime 10000;
    #HomeAgentPreference 20;
    #AdvIntervalOpt off;

    prefix 3ffe:100:0:0::/64
    {
        AdvOnLink on;
        AdvAutonomous on;
        AdvRouterAddr on;
    };
};
```

#### III.4.4. Configuració del Mobile Node

Només cal configurar l'adreça IPv6 local (tal com s'explica a l'apartat III.3.4 d'aquest annex), ja que l'adreça global s'autoconfigura amb els paquets Router Advertisement provinents de l'AR al qual es connecta.

Carregar els mòduls del kernel necessaris per al funcionament de MIPv6.

```
sudo insmod /lib/modules/2.6.27.13/wlan/wl.ko
sudo insmod /lib/modules/2.6.27.13/kernel/net/ipv6/tunnel6.ko
sudo insmod /lib/modules/2.6.27.13/kernel/net/ipv6/ip6_tunnel.ko
sudo insmod /lib/modules/2.6.27.13/kernel/net/ipv6/ah6.ko
sudo insmod /lib/modules/2.6.27.13/kernel/net/ipv6/esp6.ko
sudo insmod /lib/modules/2.6.27.13/kernel/net/ipv6/ipcomp6.ko
sudo insmod /lib/modules/2.6.27.13/kernel/net/ipv6/ipv6.ko
sudo insmod /lib/modules/2.6.27.13/kernel/net/ipv6/mip6.ko
sudo insmod /lib/modules/2.6.27.13/kernel/net/ipv6/sit.ko
sudo insmod /lib/modules/2.6.27.13/kernel/net/ipv6/xfrm6_mode_beet.ko
sudo insmod /lib/modules/2.6.27.13/kernel/net/ipv6/xfrm6_mode_ro.ko
sudo insmod /lib/modules/2.6.27.13/kernel/net/ipv6/xfrm6_mode_transport.ko
```



```
sudo insmod /lib/modules/2.6.27.13/kernel/net/ipv6/xfrm6_mode_tunnel.ko
sudo insmod /lib/modules/2.6.27.13/kernel/net/ipv6/xfrm6_tunnel.ko
sudo insmod /lib/modules/2.6.27.13/kernel/net/key/af_key.ko
sudo insmod /lib/modules/2.6.27.13/kernel/net/xfrm/xfrm_ipcomp.ko
sudo insmod /lib/modules/2.6.27.13/kernel/net/xfrm/xfrm_user.ko
```

Ordres perquè faci l'*autoconf* i accepti *RtrAdv*:

```
echo 0 > /proc/sys/net/ipv6/conf/all/router_solicitations
echo 0 > /proc/sys/net/ipv6/conf/all/router_solicitation_interval
echo "0" > /proc/sys/net/ipv6/conf/all/forwarding
echo "1" > /proc/sys/net/ipv6/conf/all/autoconf
echo "1" > /proc/sys/net/ipv6/conf/all/accept_ra
echo "1" > /proc/sys/net/ipv6/conf/all/accept_redirects
```

El fitxer de configuració *mip6d.conf* és el següent:

```
NodeConfig MN;

## If set to > 0, will not detach from tty
DebugLevel 3;

## Support route optimization with other MNs
DoRouteOptimizationCN enabled;

## Use route optimization with CNs
DoRouteOptimizationMN disabled;

UseCnBuAck disabled;

MnDiscardHaParamProb disabled;

MnMaxHaBindingLife 60;
MnMaxCnBindingLife 60;

#Interface "eth1";

MnRouterProbes 1;

MnHomeLink "eth1" {
    HomeAgentAddress 3ffe:100::f100:0:0:1;
    HomeAddress 3ffe:100::223:6cff:fe86:c027/64;
}

##
## IPsec configuration
##
```

```
UseMnHalPsec disabled;

## Key Management Mobility Capability
KeyMngMobCapability disabled;

Interface "eth1" { MnIfPreference 1; }
```

### III.4.5. Configuració de l'AP

La configuració bàsica dels AP Cisco Aironet 1130 es detalla a l'Annex V d'aquest treball.

Per a donar suport als nodes dormants, cal canviar l'SSID que anuncien els AP per un que reflecteixi en quin router estan situats:

```
si el router que anuncia l'AP és 3ffe:100:0:5:f100:0:0:11, l'SSID serà:

0005f100000000000011

si el router que anuncia el AP és 3ffe:100:0:6:f100:0:0:12, l'SSID serà:

0006f100000000000012
```

Atès que els AP estaran molt junts durant les proves, és important que tots dos se separin per una franja mínima de cinc canals per tal de no tenir interferències i obtenir un traspàs al més ràpid possible.

### III.5. Posada en marxa de la maqueta

En aquest apartat es descriuen en estricte ordre els passos a seguir per a posar en marxa el demostrador.

Els AP només s'han d'endollar a les interfícies corresponents de cada AR.

1. Engregar les màquines PA, AR1, AR2 i CN.
2. Executar l'ordre següent per tal de tenir cada màquina configurada:

```
#sh /etc/rc.local
```

3. Comprovar la connectivitat entre els quatre nodes amb l'eina ping6.
4. Executar el següent a AR1 i AR2:

```
#radvd -m stderr
```

i tot seguit endollar els AP a les seves interfícies corresponents.

5. Si s'està executant la versió modificada del protocol, cal assignar a les interfícies dels AR connectats als AP una nova adreça *Local-Link* (tal com s'especifica al Capítol 2).

Per exemple, si l'adreça de la interfície de l'AP és *3ffe:100:0:6:f100:0:0:12*, cal afegir una nova interfície amb:

```
#ifconfig ethx add fe80:0:0:6:f100:0:0:12/64
```

6. Engregar la màquina HA i executar el codi de *rc.local* a l'HA per a tenir el node configurat correctament.
7. Arrencar el dimoni UMIP a l'HA amb l'ordre:

```
#mip6d
```

8. Arrencar el dimoni Radvd a l'HA amb l'ordre:

```
#radvd -m stderr
```

9. Engregar la màquina MN i executar el codi de *rc.local* de l'MN per a tenir el node configurat correctament.
10. Esperar que la interfície connectada a l'AR s'autoconfiguri amb una adreça IPv6 global i arrencar el dimoni UMIP:

```
#mip6d
```

Una vegada seguits aquests passos, el demostrador MIPv6 és totalment operatiu.

### III.5.1. Notes finals

Si es desitja fer traspassos entre AP, es recomana desactivar qualsevol dimoni de monitoratge de xarxa com el *network monitor* o semblant, ja que impedeixen fer un traspàs ràpid.

Per a fer els traspassos entre AP, idealment cal introduir l'ordre següent des de consola:

```
#lwconfig ethx ssid "nom ssid"
```

Finalment, si no es volen seguir tots els passos explicats anteriorment per tal de configurar el kernel per a tenir suport per a MIPv6, es pot descarregar i instal·lar la versió modificada de Linux-Ubuntu oferta per al projecte Nautilus (vegeu l'Annex I).

## III.6. Anàlisi i proves del funcionament del demostrador MIPv6

En aquest darrer apartat s'analitza el comportament del demostrador per tal de verificar que la maqueta funciona correctament tant pel que fa al protocol IPv6 com al protocol MIPv6.

### III.6.1. Proves d'IPv6

Les proves de validació del protocol IPv6 consisteixen bàsicament a assegurar que es té connectivitat entre tots els nodes de la maqueta i que el procés d'autoconfiguració de l'MN funciona correctament.

La taula següent reflecteix les proves realitzades, juntament amb els resultats obtinguts:

**Taula III. 4.** Resultat de les proves de connectivitat IPv6

Descripció	Resultat
Connectivitat MN amb HA	Correcte
Connectivitat MN amb CN	Correcte
Connectivitat AR1 amb AR2	Correcte
Connectivitat AR1 amb HA	Correcte
Connectivitat AR2 amb HA	Correcte

### III.6.2. Proves de MIPv6

A continuació es comenten els trets més importants del funcionament del dimoni UMIP, amb exemples típics de situacions previstes pel protocol MIPv6.

El funcionament del codi modificat i les proves realitzades s'expliquen amb detall als Capítols 2 i 3 d'aquest estudi.

La manera d'actuar del dimoni depèn del rol que assumeixi cadascun dels nodes (CN, MN i HA).

#### III.6.2.1. Arrencar el dimoni

Quan s'activa el dimoni a l'HA, apareix la figura següent:

```
mip6d[6703]: MIPL Mobile IPv6 for Linux v2.0.2-umip-0.4 started (Home Agent)
```

```

Sun Apr 19 20:05:06 main: MIPL Mobile IPv6 for Linux started in debug mode, not detaching
from terminal
Sun Apr 19 20:05:06 conf_show: config_file = /usr/local/etc/mip6d.conf
Sun Apr 19 20:05:06 conf_show: vt_hostname = localhost
Sun Apr 19 20:05:06 conf_show: vt_service = 7777
Sun Apr 19 20:05:06 conf_show: mip6_entity = 2
Sun Apr 19 20:05:06 conf_show: debug_level = 10
Sun Apr 19 20:05:06 conf_show: debug_log_file = stderr
Sun Apr 19 20:05:06 conf_show: PolicyModulePath = [internal]
Sun Apr 19 20:05:06 conf_show: DefaultBindingAcIPolicy = 0
Sun Apr 19 20:05:06 conf_show: NonVolatileBindingCache = disabled
Sun Apr 19 20:05:06 conf_show: KeyMngMobCapability = disabled
Sun Apr 19 20:05:06 conf_show: UseMnHalPsec = disabled
Sun Apr 19 20:05:06 conf_show: MnMaxHaBindingLife = 100
Sun Apr 19 20:05:06 conf_show: MnMaxCnBindingLife = 420
Sun Apr 19 20:05:06 conf_show: MnRouterProbes = 0
Sun Apr 19 20:05:06 conf_show: MnRouterProbeTimeout = 0.000000
Sun Apr 19 20:05:06 conf_show: InitialBindackTimeoutFirstReg = 1.500000
Sun Apr 19 20:05:06 conf_show: InitialBindackTimeoutReReg = 1.000000
Sun Apr 19 20:05:06 conf_show: UseCnBuAck = disabled
Sun Apr 19 20:05:06 conf_show: DoRouteOptimizationMN = enabled
Sun Apr 19 20:05:06 conf_show: MnUseAllInterfaces = disabled
Sun Apr 19 20:05:06 conf_show: MnDiscardHaParamProb = disabled
Sun Apr 19 20:05:06 conf_show: SendMobPfxSols = enabled
Sun Apr 19 20:05:06 conf_show: OptimisticHandoff = disabled
Sun Apr 19 20:05:06 conf_show: SendMobPfxAdvs = enabled
Sun Apr 19 20:05:06 conf_show: SendUnsolMobPfxAdvs = enabled
Sun Apr 19 20:05:06 conf_show: MaxMobPfxAdvInterval = 86400
Sun Apr 19 20:05:06 conf_show: MinMobPfxAdvInterval = 600
Sun Apr 19 20:05:06 conf_show: HaMaxBindingLife = 100
Sun Apr 19 20:05:06 conf_show: DoRouteOptimizationCN = disabled
Sun Apr 19 20:05:06 xfrm_cn_init: Adding policies and states for CN
Sun Apr 19 20:05:06 xfrm_ha_init: Adding policies and states for HA
Error abrindo archivo /proc/net/dev
Sun Apr 19 20:05:06 ha_if_addr_setup: My ip: 3ffe:100:0:0:f100:0:0:1
Sun Apr 19 20:05:06 ha_if_addr_setup: Joined anycast group 3ffe:100:0:0:fdff:ffff:ffff:ffe on
iface 3

```

**Fig. III.16.** Execució del dimoni UMIP a l'HA

D'inici es mostra la configuració que s'ha donat a l'HA mitjançant el fitxer *mip6d.conf* corresponent. Una vegada carregada la configuració, el node s'uneix al grup *anycast* que porta el prefix de xarxa de l'HA (en aquest cas, *3ffe:100:0:0*).

Quan s'activa el dimoni al CN, apareix la figura següent:

```

[root@CN ~]# mip6d
mip6d[3114]: MIPL Mobile IPv6 for Linux v2.0.1 started (Correspondent Node)
main: MIPL Mobile IPv6 for Linux started in debug mode, not detaching from terminal
conf_show: config_file = /usr/local/etc/mip6d.conf
conf_show: mip6_entity = 0
conf_show: debug_level = 10
conf_show: PolicyModulePath = [internal]
conf_show: DefaultBindingAcIPolicy = 0
conf_show: NonVolatileBindingCache = disabled
conf_show: KeyMngMobCapability = disabled
conf_show: UseMnHalPsec = enabled

```

```

conf_show: MnMaxHaBindingLife = 262140
conf_show: MnMaxCnBindingLife = 420
conf_show: MnRouterProbes = 0
conf_show: MnRouterProbeTimeout = 0.000000
conf_show: InitialBindackTimeoutFirstReg = 1.500000
conf_show: InitialBindackTimeoutReReg = 1.000000
conf_show: UseCnBuAck = disabled
conf_show: DoRouteOptimizationMN = enabled
conf_show: MnUseAllInterfaces = disabled
conf_show: MnDiscardHaParamProb = disabled
conf_show: SendMobPfxSols = enabled
conf_show: SendMobPfxAdvs = enabled
conf_show: SendUnsolMobPfxAdvs = enabled
conf_show: MaxMobPfxAdvInterval = 86400
conf_show: MinMobPfxAdvInterval = 600
conf_show: HaMaxBindingLife = 262140
conf_show: DoRouteOptimizationCN = enabled
xfrm_cn_init: Adding policies and states for CN

```

**Fig. III.17.** Execució del dimoni UMIP a l'MN

D'inici es mostra la configuració que s'ha donat al CN mitjançant el fitxer *mip6d.conf* corresponent. Com que es tracta d'un node "passiu", el dimoni UMIP del CN no actua de manera especial fins que no s'inicia el mecanisme de *Route Optimization* entre MN i CN.

Quan s'activa el dimoni a l'MN, apareix la figura següent :

```

mip6d[10262]: MIPL Mobile IPv6 for Linux v2.0.2-umip-0.4 started (Mobile Node)
Sun Apr 19 20:05:46 main: MIPL Mobile IPv6 for Linux started in debug mode, not detaching
from terminal
Sun Apr 19 20:05:46 conf_show: config_file = /usr/local/etc/mip6d.conf
Sun Apr 19 20:05:46 conf_show: vt_hostname = localhost
Sun Apr 19 20:05:46 conf_show: vt_service = 7777
Sun Apr 19 20:05:46 conf_show: mip6_entity = 1
Sun Apr 19 20:05:46 conf_show: debug_level = 3
Sun Apr 19 20:05:46 conf_show: debug_log_file = stderr
Sun Apr 19 20:05:46 conf_show: PolicyModulePath = [internal]
Sun Apr 19 20:05:46 conf_show: DefaultBindingAclPolicy = 0
Sun Apr 19 20:05:46 conf_show: NonVolatileBindingCache = disabled
Sun Apr 19 20:05:46 conf_show: KeyMngMobCapability = disabled
Sun Apr 19 20:05:46 conf_show: UseMnHalPsec = disabled
Sun Apr 19 20:05:46 conf_show: MnMaxHaBindingLife = 60
Sun Apr 19 20:05:46 conf_show: MnMaxCnBindingLife = 60
Sun Apr 19 20:05:46 conf_show: MnRouterProbes = 1
Sun Apr 19 20:05:46 conf_show: MnRouterProbeTimeout = 0.000000
Sun Apr 19 20:05:46 conf_show: InitialBindackTimeoutFirstReg = 1.500000
Sun Apr 19 20:05:46 conf_show: InitialBindackTimeoutReReg = 1.000000
Sun Apr 19 20:05:46 conf_show: UseCnBuAck = disabled
Sun Apr 19 20:05:46 conf_show: DoRouteOptimizationMN = disabled
Sun Apr 19 20:05:46 conf_show: MnUseAllInterfaces = disabled
Sun Apr 19 20:05:46 conf_show: MnDiscardHaParamProb = disabled
Sun Apr 19 20:05:46 conf_show: SendMobPfxSols = enabled
Sun Apr 19 20:05:46 conf_show: OptimisticHandoff = disabled
Sun Apr 19 20:05:46 conf_show: SendMobPfxAdvs = enabled
Sun Apr 19 20:05:46 conf_show: SendUnsolMobPfxAdvs = enabled
Sun Apr 19 20:05:46 conf_show: MaxMobPfxAdvInterval = 86400
Sun Apr 19 20:05:46 conf_show: MinMobPfxAdvInterval = 600

```

```
Sun Apr 19 20:05:46 conf_show: HaMaxBindingLife = 262140  
Sun Apr 19 20:05:46 conf_show: DoRouteOptimizationCN = enabled
```

```

Sun Apr 19 20:05:46 xfrm_cn_init: Adding policies and states for CN
Sun Apr 19 20:05:46 xfrm_mn_init: Adding policies and states for MN
Error abriendo archivo /proc/net/dev
Sun Apr 19 20:05:46 conf_home_addr_info: HoA address 3ffe:100:0:0:223:6cff:fe86:c027
Sun Apr 19 20:05:46 conf_home_addr_info: HA address 3ffe:100:0:0:f100:0:0:1
Sun Apr 19 20:05:46 __tunnel_add: created tunnel ip6tnl1 (8) from
3ffe:100:0:0:223:6cff:fe86:c027 to 3ffe:100:0:0:f100:0:0:1 user count 1
Sun Apr 19 20:05:46 conf_home_addr_info: Home address 3ffe:100:0:0:223:6cff:fe86:c027
Sun Apr 19 20:05:46 flag_hoa: set HoA 3ffe:100:0:0:223:6cff:fe86:c027/128 iif 8 flags 12
preferred_time 4294967295 valid_time 4294967295
Sun Apr 19 20:05:46 conf_home_addr_info: Added new home_addr_info successfully

```

**Fig. III.18.** Execució del dimoni UMIP al CN

D'inici es mostra la configuració que s'ha aplicat al CN mitjançant el fitxer *mip6d.conf* corresponent. Seguidament s'afegeixen les adreces HoA (la pròpia del node) i la de l'HA. En cas de no haver especificat aquestes adreces, aquesta informació es recolliria per DHAAD (*Dynamic Home Agent Address Discovery*).

A continuació s'afegeix el túnel IPv6 entre l'HA i l'MN, que porta com a identificador *ip6tnl*. Es pot executar l'ordre **ifconfig** per a apreciar com s'ha creat aquest túnel:

```

ip6tnl1 Link encap:UNSPEC HWaddr 3F-FE-01-00-00-00-00-05-00-00-00-00-00-00-00-00
inet6 addr: 3ffe:100::223:6cff:fe86:c027/128 Scope:Global
inet6 addr: fe80::223:6cff:fe86:c027/64 Scope:Link
UP POINTOPOINT RUNNING NOARP MTU:1460 Metric:1
RX packets:0 errors:0 dropped:0 overruns:0 frame:0
TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
collisions:0 txqueuelen:0
RX bytes:0 (0.0 B) TX bytes:0 (0.0 B)

```

**Fig. III.19.** Interfície del túnel a l'MN

Aquest túnel és l'encarregat de transmetre els paquets dirigits des de i cap a l'adreça HoA. Els missatges del protocol MIPv6 s'envien per la interfície connectada a l'AR.

Una vegada obtinguda tota la informació, es passa a escoltar els Router Advertisements que arriben a l'MN per a decidir si es troba a la xarxa de l'HA mateix o en una xarxa externa.

En cas de trobar-se en una xarxa externa (*foreign net*), l'MN anuncia a l'HA la seva adreça CoA (l'adreça que l'AR ha donat a l'MN) mitjançant un missatge MIPv6 Binding Update.

Una de les opcions amb UMIP és la de poder veure en tot moment les BUL de l'MN, ja que incorpora una màquina virtual que permet visualitzar-les.

Per a poder veure les BUL cal executar l'ordre següent en una nova consola:

```
#telnet open localhost 7777
```



Dins del servidor cal executar l'ordre:

**#bul**

i s'observarà la BUL actual de l'MN.

### III.6.2.2. Enviament de paquets ICMPv6

La darrera de les proves bàsiques realitzades consisteix a enviar paquets ICMPv6 (ping6) des de l'MN cap al CN i viceversa.

Per a veure aquests paquets a l'MN, cal executar el programari Wireshark a la interfície ip6tnl creada a l'hora d'arrencar el dimoni UMIP.

```
[root@CN ~]# ping6 3ffe:100:0:0:223:6cff:fe86:c027
PING 3ffe:100:0:0:223:6cff:fe86:c027(3ffe:100::223:6cff:fe86:c027) 56 data bytes
64 bytes from 3ffe:100::223:6cff:fe86:c027: icmp_seq=0 ttl=62 time=1060 ms
64 bytes from 3ffe:100::223:6cff:fe86:c027: icmp_seq=1 ttl=62 time=60.7 ms
64 bytes from 3ffe:100::223:6cff:fe86:c027: icmp_seq=2 ttl=62 time=77.5 ms
64 bytes from 3ffe:100::223:6cff:fe86:c027: icmp_seq=3 ttl=62 time=93.6 ms
64 bytes from 3ffe:100::223:6cff:fe86:c027: icmp_seq=4 ttl=62 time=125 ms
64 bytes from 3ffe:100::223:6cff:fe86:c027: icmp_seq=5 ttl=62 time=134 ms
64 bytes from 3ffe:100::223:6cff:fe86:c027: icmp_seq=7 ttl=62 time=486 ms
64 bytes from 3ffe:100::223:6cff:fe86:c027: icmp_seq=8 ttl=62 time=297 ms
64 bytes from 3ffe:100::223:6cff:fe86:c027: icmp_seq=10 ttl=62 time=234 ms
64 bytes from 3ffe:100::223:6cff:fe86:c027: icmp_seq=12 ttl=62 time=291 ms
64 bytes from 3ffe:100::223:6cff:fe86:c027: icmp_seq=13 ttl=62 time=124 ms
```

**Fig. III.20.** Enviament de ping des del CN cap a l'MN

Per a visualitzar els paquets al CN, n'hi ha prou de visualitzar la interfície de xarxa connectada al PA.

Es pot apreciar com a l'hora d'enviar el primer paquet ICMPv6 (figura III.20), aquest té un TTL superior als altres.

Això és degut al fet que el primer paquet és l'encarregat de "descobrir" la ruta cap a l'MN passant per l'HA, ja que aquest últim és l'encarregat d'enviar cap a l'MN els paquets que li arriben, mitjançant el túnel ip6tnl.

Quan el primer paquet ICMPv6 arriba a l'HA, aquest ha de modificar les seves regles d'enrutament per a tunelitzar el trànsit del CN cap a l'MN.

A la figura següent es mostra el trànsit dels paquets de dades per la interfície ip6tnl1 de l'HA.

No. -	Time	Source	Destination	Protocol	Info
1	0.000000	3ffe:100:0:1:f100::3	3ffe:100::223:6cff:fe	ICMPv6	Echo request
2	0.099864	3ffe:100::223:6cff:fe86:c027	3ffe:100:0:1:f100::3	ICMPv6	Echo reply
3	0.231976	3ffe:100:0:1:f100::3	3ffe:100::223:6cff:fe	ICMPv6	Echo request
4	0.235902	3ffe:100::223:6cff:fe86:c027	3ffe:100:0:1:f100::3	ICMPv6	Echo reply
5	1.243933	3ffe:100:0:1:f100::3	3ffe:100::223:6cff:fe	ICMPv6	Echo request
6	1.331862	3ffe:100::223:6cff:fe86:c027	3ffe:100:0:1:f100::3	ICMPv6	Echo reply
7	2.239915	3ffe:100:0:1:f100::3	3ffe:100::223:6cff:fe	ICMPv6	Echo request
8	2.355889	3ffe:100::223:6cff:fe86:c027	3ffe:100:0:1:f100::3	ICMPv6	Echo reply
9	3.253384	3ffe:100:0:1:f100::3	3ffe:100::223:6cff:fe	ICMPv6	Echo request
10	3.376096	3ffe:100::223:6cff:fe86:c027	3ffe:100:0:1:f100::3	ICMPv6	Echo reply
11	4.263086	3ffe:100:0:1:f100::3	3ffe:100::223:6cff:fe	ICMPv6	Echo request
12	4.407880	3ffe:100::223:6cff:fe86:c027	3ffe:100:0:1:f100::3	ICMPv6	Echo reply
13	5.258243	3ffe:100:0:1:f100::3	3ffe:100::223:6cff:fe	ICMPv6	Echo request
14	5.427902	3ffe:100::223:6cff:fe86:c027	3ffe:100:0:1:f100::3	ICMPv6	Echo reply
15	6.263060	3ffe:100:0:1:f100::3	3ffe:100::223:6cff:fe	ICMPv6	Echo request
16	6.451865	3ffe:100::223:6cff:fe86:c027	3ffe:100:0:1:f100::3	ICMPv6	Echo reply
17	7.259921	3ffe:100:0:1:f100::3	3ffe:100::223:6cff:fe	ICMPv6	Echo request
18	7.267863	3ffe:100::223:6cff:fe86:c027	3ffe:100:0:1:f100::3	ICMPv6	Echo reply
19	8.272131	3ffe:100:0:1:f100::3	3ffe:100::223:6cff:fe	ICMPv6	Echo request
20	8.291867	3ffe:100::223:6cff:fe86:c027	3ffe:100:0:1:f100::3	ICMPv6	Echo reply

Frame 1 (120 bytes on wire, 120 bytes captured)  
 Linux cooked capture  
 Packet type: Sent by us (4)  
 Link-layer address type: 769  
 Link-layer address length: 0  
 Source: <MISSING>  
 Protocol: IPv6 (0x86dd)  
 Internet Protocol Version 6  
 0110 .... = Version: 6  
 .... 0000 0000 .... = Traffic class: 0x00000000  
 .... 0000 0000 0000 0000 = Flowlabel: 0x00000000  
 Payload length: 64  
 Next header: ICMPv6 (0x3a)  
 Hop limit: 62  
 Source: 3ffe:100:0:1:f100::3 (3ffe:100:0:1:f100::3)  
 Destination: 3ffe:100::223:6cff:fe86:c027 (3ffe:100::223:6cff:fe86:c027)  
 Internet Control Message Protocol v6  
 Type: 128 (Echo request)

**Fig. III. 21.** paquets de dades del CN a l'MN a la interfície túnel de l'HA

## ANNEX IV. MANUAL DE CONFIGURACIÓ DE L'ANALITZADOR DE POTÈNCIA AGILENT N6705A

L'analitzador de potència Agilent N6705A és un sistema que combina les capacitats d'anàlisi i captura de dades d'un oscil·loscopi amb un generador d'alimentació elèctrica multisortida. És possible connectar fins a quatre mòduls de potència diferents a l'analitzador, cosa que permet modificar segons les necessitats de l'escenari de proves els diversos paràmetres de tensió o d'intensitat, així com de potència, de sortida del dispositiu.

En aquest manual es mostra de manera resumida i general el procés d'instal·lació i configuració del dispositiu perquè funcioni en un entorn de proves semblant al que s'ha utilitzat en aquest estudi. Per tant, resulta molt senzill extrapolar aquestes instruccions per a la connexió d'altres dispositius.

Vegeu [21] per a obtenir informació més detallada sobre el dispositiu.

La figura IV.1 mostra una imatge frontal de l'analitzador de potència Agilent N6705A.



**Fig. IV.1.** Analitzador de potència Agilent N6705A

## IV.1. Característiques generals de l'analitzador de potència Agilent N6705A

Les característiques principals del dispositiu s'enumeren a la taula IV.1.

**Taula IV.1.** Característiques de l'analitzador de potència Agilent N6705A

Potència màxima de sortida	600W
Memòria interna d'emmagatzematge	64Mbytes
Interfície USB	USB 2.0
Interfície de xarxa	10/100 LAN
Sortides d'alimentació	4

En el manual disponible a la pàgina web d'Agilent es pot trobar una llista de característiques més detallada (vegeu [21]).

## IV.2. Mòduls de potència de l'analitzador de potència Agilent N6705A

L'analitzador de potència disponible durant la realització d'aquest estudi conté dos mòduls de potència, cosa que el dota de certa flexibilitat a l'hora de configurar els diferents escenaris.

La figura IV.2 mostra tots dos mòduls.



**Fig. IV.2.** Mòduls de potència de l'analitzador Agilent N6705A

En els apartats següents es detallen les característiques de tots dos mòduls, que són programables i ofereixen sortides variables de tensió i intensitat.

#### IV.2.1. Mòdul de potència 1

El mòdul de potència 1 de l'analitzador N6705A utilitzat per a la realització d'aquest estudi és un mòdul Agilent N6761A. Aquest mòdul és una font d'alimentació de precisió de tensió contínua, que opera en la regió dels mili i microampers.

La taula IV.2 en mostra les característiques principals.

**Taula IV.2.** Característiques del mòdul de potència N6761A

Tensió	50 V
Intensitat	1,5 A
Potència	50 W

#### IV.2.2. Mòdul de potència 2

El mòdul de potència 2 de l'analitzador N6705A utilitzat per a la realització d'aquest estudi és un mòdul Agilent N6774A, considerat pel fabricant com el model bàsic de font d'alimentació d'alt rendiment.

La taula IV.3 en mostra les característiques principals.

**Taula IV.3.** Característiques del mòdul de potència N6774A

Tensió	35 V
Intensitat	8,5 A
Potència	300 W

#### IV.2.3. Instal·lació general d'un mòdul de potència

A continuació es mostra el procés d'instal·lació d'un mòdul de potència a l'analitzador. Abans de començar la instal·lació és important recordar que es tracta d'un dispositiu electrònic complex, i que les operacions les ha de dur a terme personal qualificat i en un entorn lliure d'electricitat estàtica.

L'analitzador disposa d'una presa a terra que caldria connectar per a operar amb seguretat. Abans de manipular el dispositiu, convé assegurar-se de no

tenir càrrega estàtica. Així, és recomanable fer tota l'operació amb una polsera antiestàtica.

En els apartats següents es detalla el procés d'instal·lació d'un mòdul de potència.

### IV.3. Obertura de la carcassa

Per a instal·lar el mòdul cal començar retirant la carcassa superior de l'analitzador. Per a fer-ho cal localitzar i desmuntar dos cargols de rosca manual ubicats a la part posterior de l'aparell.

La figura IV.3 mostra amb claredat la localització a la part posterior de l'analitzador d'un dels cargols que cal retirar. L'altre cargol es troba a la mateixa ubicació, però a la part esquerra de l'aparell.



**Fig. IV.3.** Cargol de desencaix de la tapa superior de l'analitzador

Després d'haver afliixat els cargols cal retirar la tapa fent-la lliscar cap enrere.

La figura IV.4 mostra la tapa de l'analitzador retirada i els cargols de subjecció destacats amb un requadre vermell.



**Fig. IV.4.** Tapa superior de l'analitzador de potència Agilent N6705A

#### **IV.4. Instal·lació del mòdul de potència**

Una vegada retirada la tapa cal situar el mòdul de potència a l'interior de la carcassa, en paral·lel amb el mòdul ja preinstal·lat. La figura IV.5 mostra el mòdul ja ubicat. Per a subjectar-lo cal emprar dos cargols i, a continuació, connectar el cable verd numerat com a 2 al sòcol situat en el mòdul (vegeu la figura IV.6, on es destaca el sòcol).



**Fig. IV.5.** Ubicació dels mòduls de potència





**Fig. IV.6.** Sòcol de connexió del mòdul de potència

Una vegada s'ha instal·lat i connectat el mòdul cal tornar a tancar l'analitzador. Per a fer-ho, cal col·locar la tapa retirada damunt la part superior de l'analitzador i enroscar els cargols fermament.

A continuació cal connectar el dispositiu a la línia elèctrica mitjançant el cable proporcionat, que s'ha de connectar a la part posterior de l'analitzador. La figura IV.7 mostra el sòcol de connexió amb el cable d'alimentació.



**Fig. IV.7.** Cable d'alimentació



Finalment, per tal d'activar l'analitzador i deixar-lo a punt per a la configuració cal accionar l'interruptor de connexió que es mostra a la figura IV.8.



**Fig. IV.8.** Interruptor d'engegada de l'analitzador de potència Agilent N6705A

## **IV.5. Configuració de l'analitzador de potència Agilent N6705A**

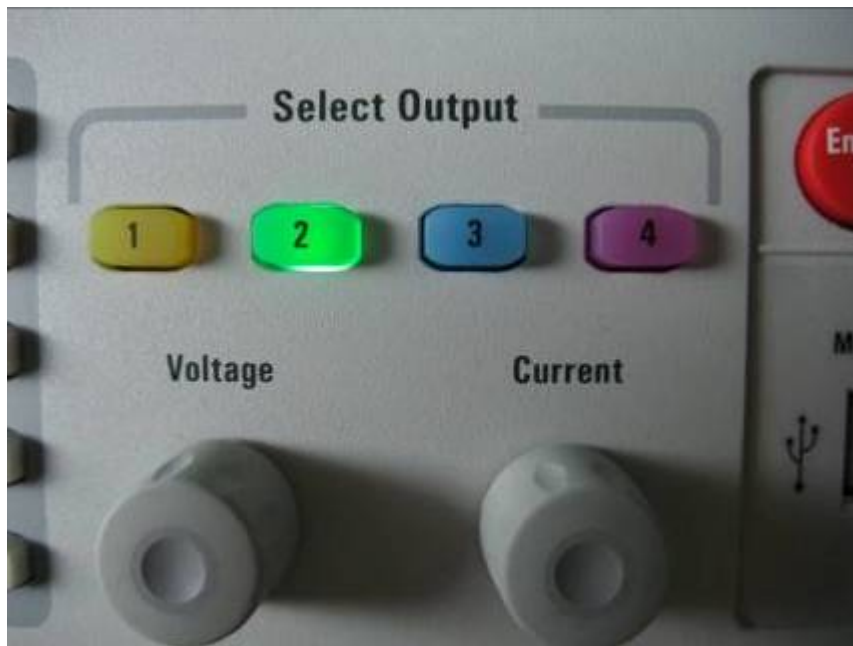
En aquest apartat es detalla el procés de configuració de l'analitzador, adequat a les necessitats de les proves de consum d'aquest projecte. El manual d'utilització que es pot descarregar de la pàgina web d'Agilent (vegeu [21]) explica aquest procés amb més profunditat i entra en altres tipus de detalls que no han estat necessaris per a aquest estudi.

### **IV.5.1. Configuració de la font d'alimentació**

L'analitzador de potència utilitzat per a la realització d'aquest treball consta de dos mòduls d'alimentació, tal com s'ha explicat en els apartats anteriors. La configuració d'aquests mòduls és idèntica i, per tant, encara que en aquest apartat s'esmentarà el mòdul de potència 2 com a exemple, les instruccions són idèntiques per al mòdul 1.

Tal com es mostra a la figura IV.9, l'analitzador permet treballar amb fins a quatre mòduls, que es poden seleccionar mitjançant els botons situats a la zona indicada com a *Select Output*. Una vegada s'ha engegat l'analitzador

(vegeu les instruccions de l'apartat IV.2.3.2) cal prémer el botó de selecció del mòdul de potència 2.



**Fig. IV.9.** Botons selectors del mòdul de potència

Una vegada s'ha seleccionat el mòdul 2, cal prémer el botó *Meter View* (vegeu la figura IV.10) per tal que es mostrin les dades de la configuració actual del mòdul a la pantalla de l'analitzador. La figura IV.11 mostra la pantalla d'informació.

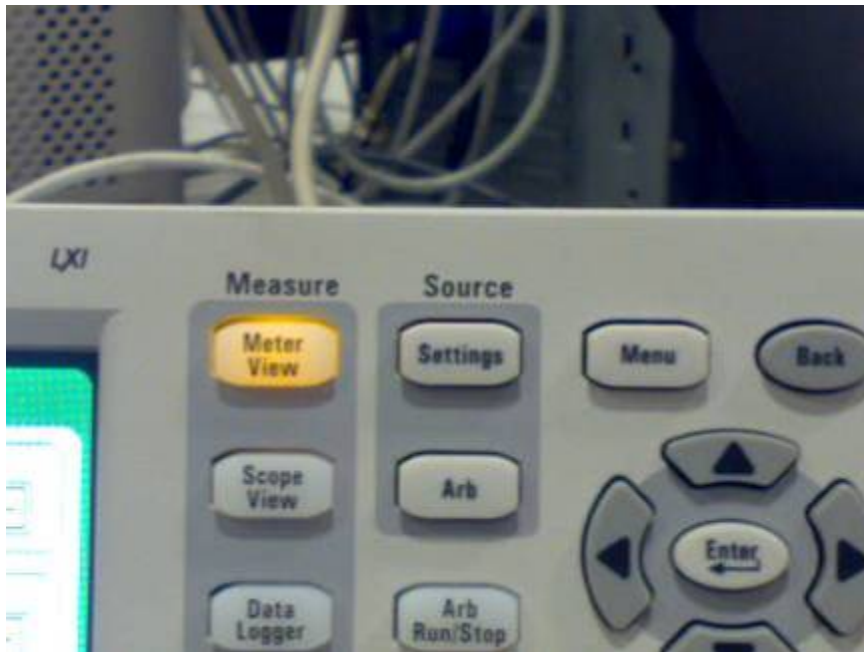


Fig. IV.10. Botó de selecció del mòdul de presentació

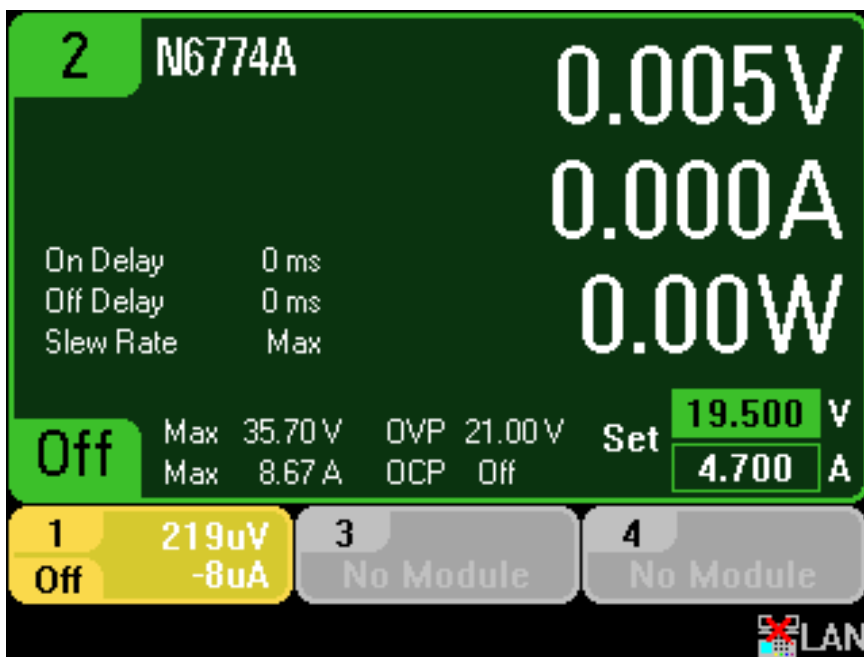


Fig. IV.11. Pantalla d'informació del mòdul de potència 2

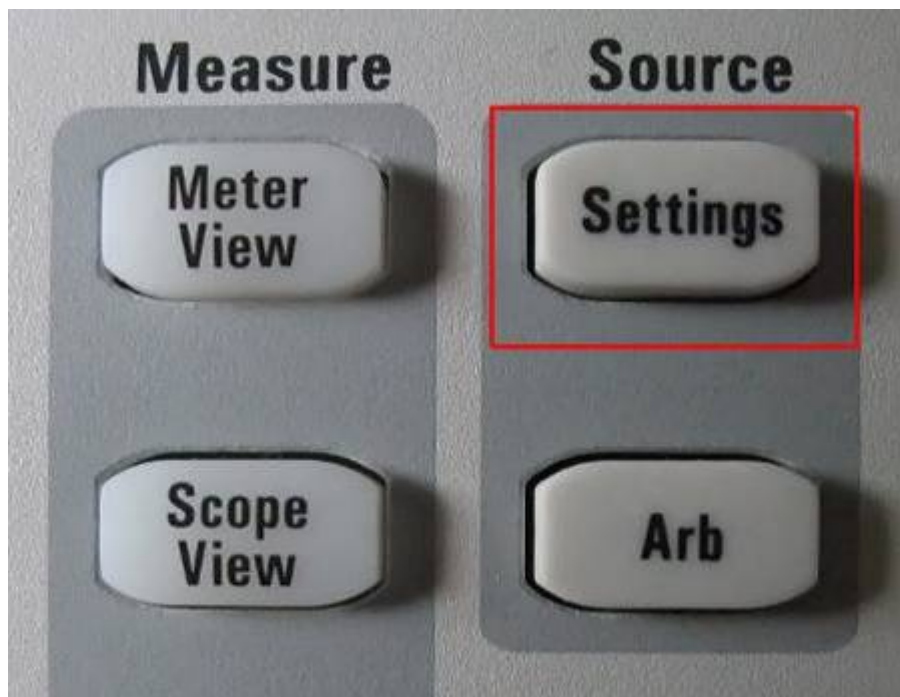
La pantalla d'informació mostra les dades relatives a la configuració d'alimentació dels dispositius connectats. Cal configurar bàsicament tres paràmetres: la tensió, la intensitat i la protecció contra pics de tensió.

Per a configurar els paràmetres de manera que s'adeqüin al dispositiu a connectar, cal saber quins paràmetres d'alimentació requereixen aquests

dispositius. En el cas d'aquest estudi, per a connectar l'estació de proves STA 1 cal configurar els paràmetres següents:

- Tensió: 19,5 V
- Intensitat: 4,7 A

Per a fer-ho cal prémer el botó *Settings* (vegeu-lo destacat a la figura IV.12) per tal que es mostri el menú de configuració a la pantalla de l'analitzador. Per tal d'introduir les dades de l'analitzador cal emprar el teclat numèric i el cursor de desplaçament que es mostren a la figura IV.13.



**Fig. IV.12.** Botó de configuració de paràmetres d'alimentació elèctrica



Fig. IV.13. Cursor i teclat numèric de l'analitzador de potència Agilent N6705A

A continuació cal introduir les dades indicades de tensió i intensitat de l'STA 1 en els camps *Voltage* i *Current*, respectivament, tal com es mostra a la figura IV.14.

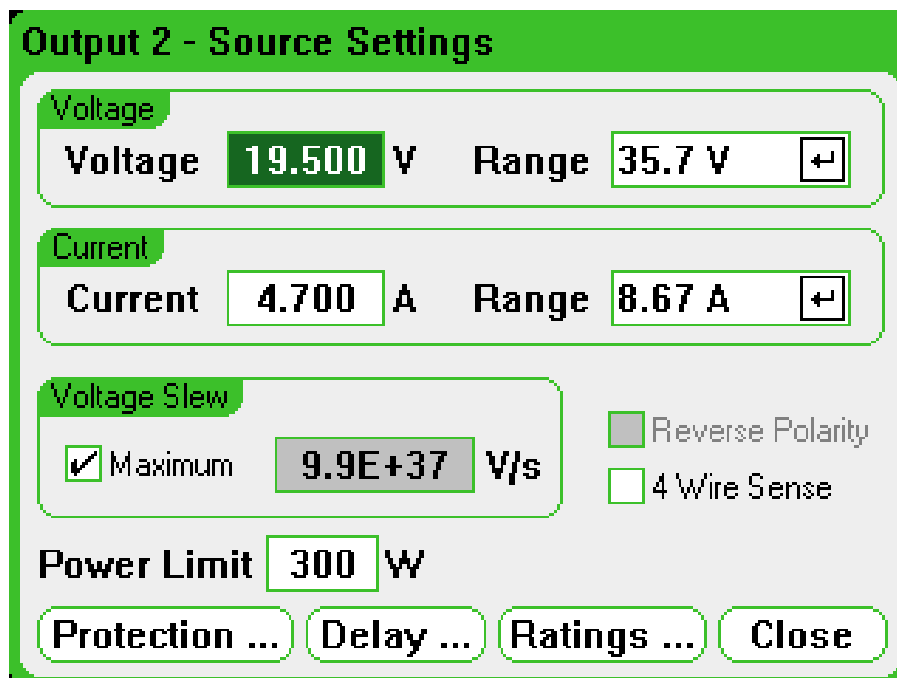


Fig. IV.14. Menú de configuració dels paràmetres elèctrics

El menú també disposa de l'opció *Protection*, que permet, entre altres paràmetres, assignar un valor màxim de tensió (vegeu la figura IV.15).

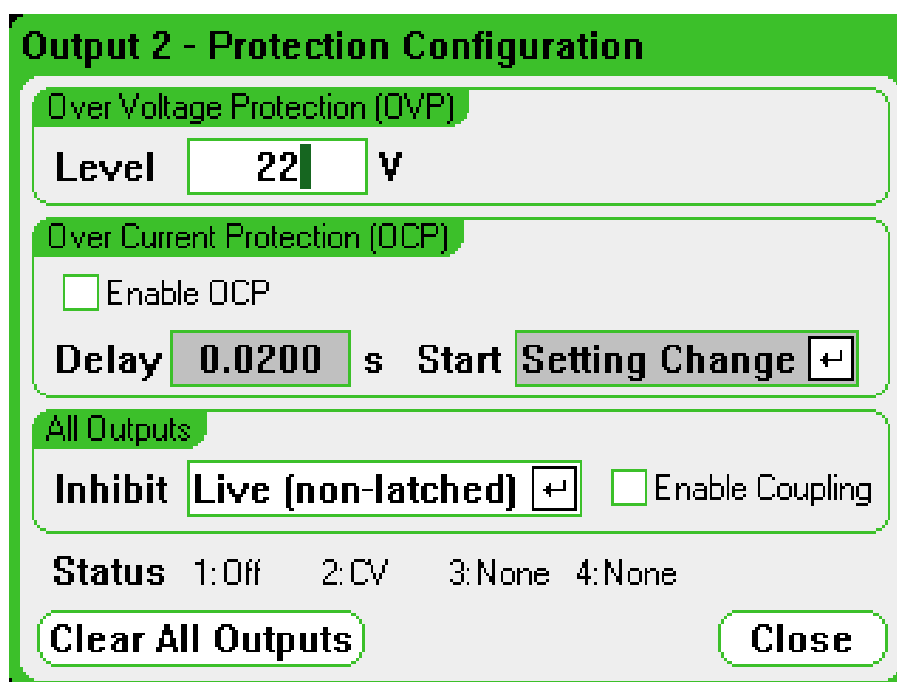


Fig. IV.15. Submenú de configuració de protecció elèctrica

Una vegada introduïts els paràmetres elèctrics, cal activar la sortida d'alimentació corresponent al mòdul 2 (vegeu-la destacada a la figura IV.16). Aquesta sortida és a la zona marcada com a *Power Supply Outputs*, a la part inferior dreta de l'analitzador.



Fig. IV.16. Botó selector de la sortida d'alimentació 2

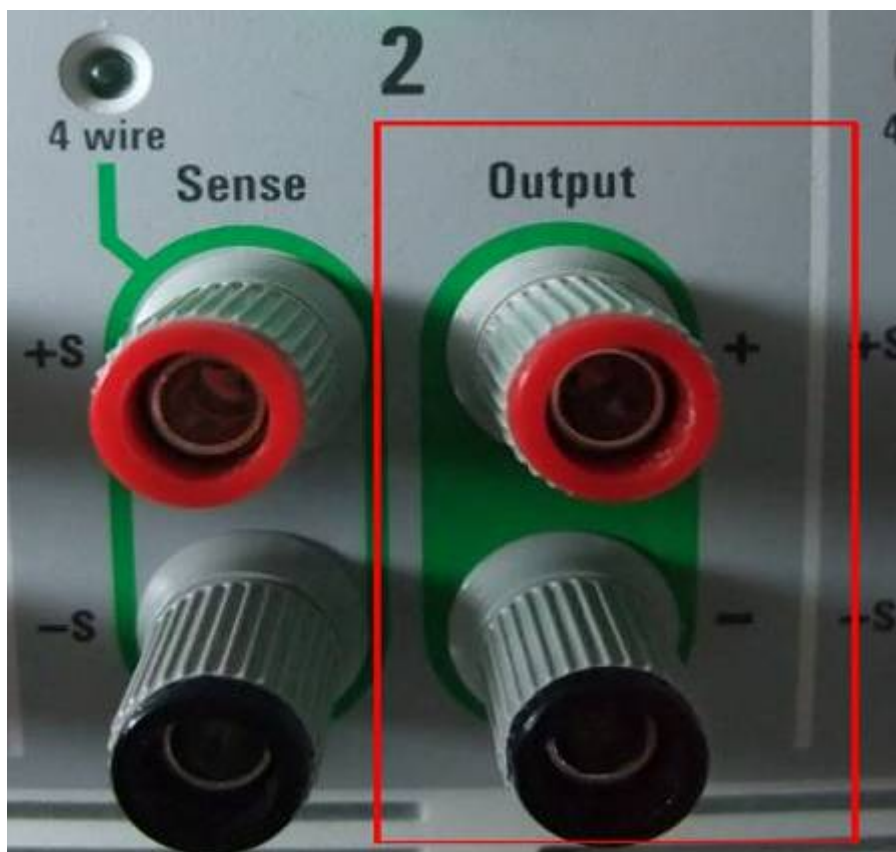
En aquest punt l'analitzador és a punt per a connectar-hi el dispositiu sobre el qual es duran a terme les proves.

## IV.5.2. Connexió de dispositius

Per a connectar a l'analitzador els dispositius de proves emprats en aquest estudi s'han utilitzat una sèrie de cables que ha calgut dissenyar amb aquest objectiu. Aquests cables s'han de connectar a les bananes femella que es mostren a la figura IV.17, on s'han destacat amb vermell les preses de sortida. La polaritat que segueixen els connectors és l'estandarditzada, en què el vermell correspon al pol positiu i el negre al negatiu.

Cal remarcar que el diàmetre dels cables de connexió s'ha de condicionar a la càrrega transportada per tal d'evitar sobreescalfaments, de manera que sigui capaç de suportar la intensitat total configurada (en aquest cas, per exemple, de 4,7 A). Aquestes dades de diàmetre, així com de l'efecte que produeixen en la tolerància del dispositiu a l'hora de prendre mesures, es poden consultar en el manual descarregable des del web d'Agilent (vegeu [21]).

Per a la realització d'aquestes proves s'ha emprat cable de diàmetre d'1,8 mm, que maneja intensitats de fins a 7,8 A sense problemes de sobreescalfament.



**Fig. IV.17.** Connectors de sortida del mòdul 2 de l'analitzador Agilent N6705A



### IV.5.3. Configuració dels paràmetres de captura

Una vegada s'ha configurat el mòdul de potència i s'ha connectat el dispositiu a l'analitzador, cal configurar l'aplicació de captació de dades Datalogger del dispositiu per tal de poder analitzar les captures.

L'analitzador de potència Agilent N6705A permet visualitzar les dades que s'estan analitzant en temps real i mostra la potència instantània. A més, també permet capturar per a un període de temps determinat la informació que es vulgui analitzar.

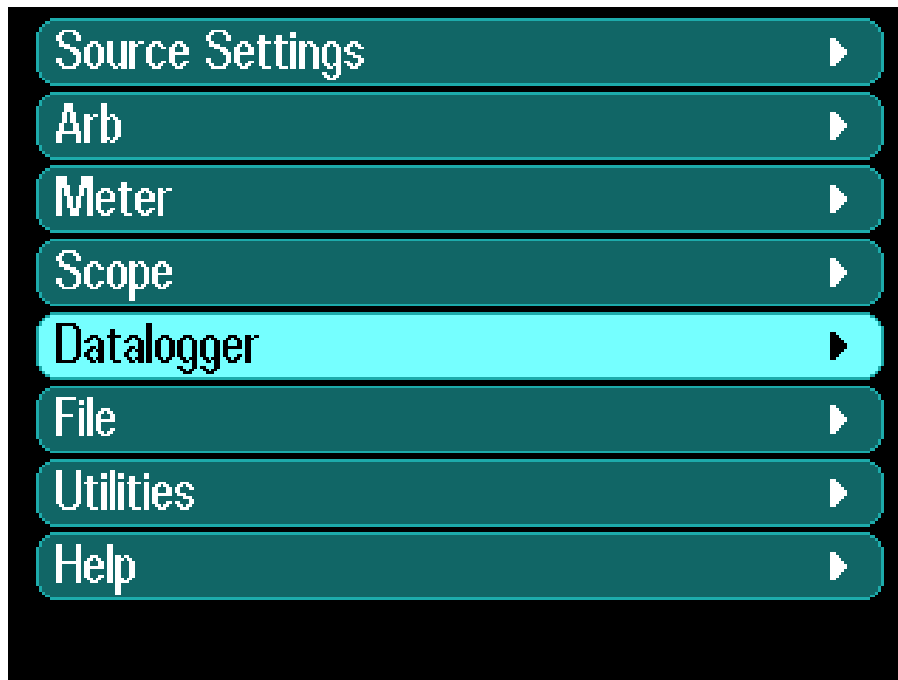
Aquest manual mostra l'opció de captura mitjançant *Datalogger*, que és la que més s'adequa a la mesura del consum de potència d'un dispositiu sotmès a proves durant un temps predeterminat.

Per a configurar el temps durant el qual s'han de capturar les mesures, cal accedir al menú de configuració de l'analitzador (a la figura IV.18 es mostra el botó corresponent a aquesta opció). Una vegada s'ha accedit a aquest menú (vegeu la figura IV.19), cal seleccionar l'opció *Datalogger*.



**Fig. IV.18.** Botó d'accés al menú de configuració





**Fig. IV.19.** Menú de configuració

En accedir a aquest submenú s'obre a la pantalla de l'analitzador el diàleg de propietats de captura, anomenat *Datalogger Properties* (vegeu la figura IV.20), on es mostren dues àrees principals que cal configurar. La primera d'aquestes àrees, anomenada *Display Trace*, permet seleccionar mitjançant els cursors i la tecla *Intro* la informació que es vol mostrar a les mesures. En el cas d'aquest estudi, cal seleccionar la potència en el mòdul 2 (*Power*).

El temps de captura i el període de mostratge es configuren a l'altra àrea principal del submenú, anomenada *Logging*. En aquesta àrea cal seleccionar la durada de la captura, que per defecte és de 30 segons.

El submenú presenta una tercera àrea que permet, per exemple, seleccionar el nom del fitxer en el qual es gravarà la informació capturada.

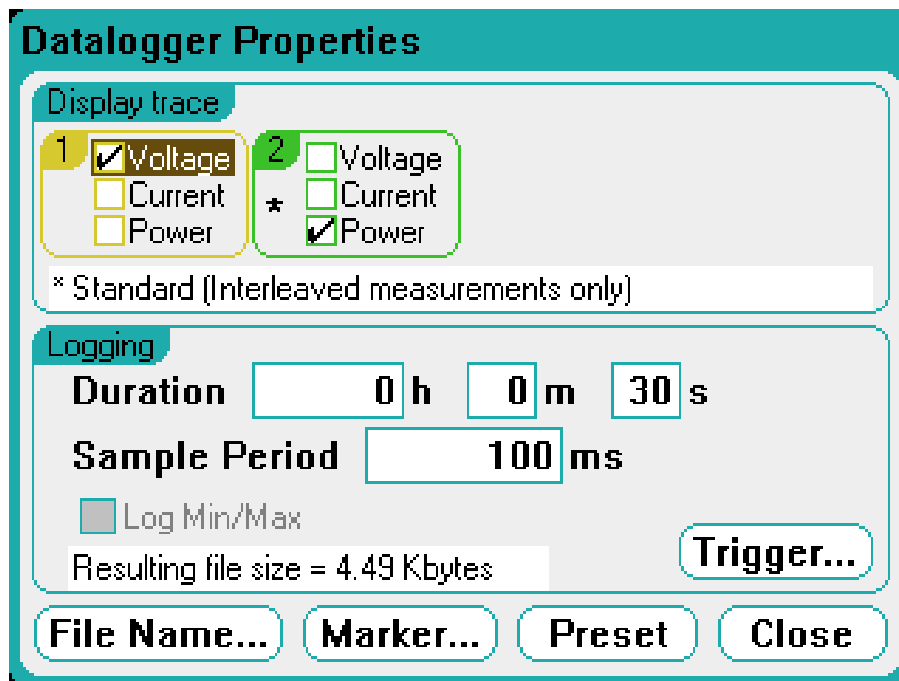


Fig. IV.20. Menú de propietats de captura de dades

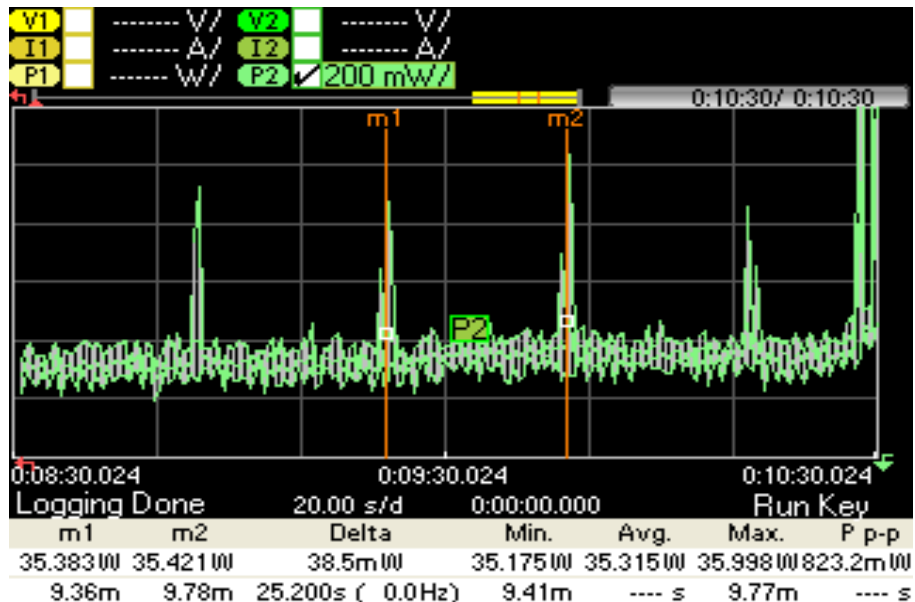
#### IV.5.4. Captura i manipulació de les mesures

Per a iniciar la captura de les mesures, en primer lloc cal seleccionar l'opció de captura mitjançant el botó *Data Logger* i, a continuació, posar-la en funcionament mitjançant el botó *Run/Stop* (a la figura IV.21 es mostren els dos botons).



Fig. IV.21. Botons d'ús de l'aplicació de captura de dades

La durada del temps de captura queda determinada pels paràmetres configurats anteriorment. Una vegada ha transcorregut aquest temps la captura s'atura i mostra les dades obtingudes. La figura IV.22 mostra un exemple de captura de consum de potència de l'estació de proves STA 1 durant 10 minuts.



**Fig. IV.22.** Exemple de captura de dades mitjançant *Datalogger*

A la captura s'observen nombroses dades com ara la potència mínima, mitjana i màxima que s'obté durant la captura, la potència pic a pic, el valor de les deltes o, mitjançant els marcadors, la potència en un punt concret de la captura.

Es disposa de dos marcadors (vegeu la figura IV.23) que es poden desplaçar per a acotar zones de la captura i obtenir informació complementària. Així mateix, també es poden variar els valors de la component vertical i horitzontal, és a dir, dels volts per divisió i el temps per divisió, per a visualitzar la mostra amb més detall.

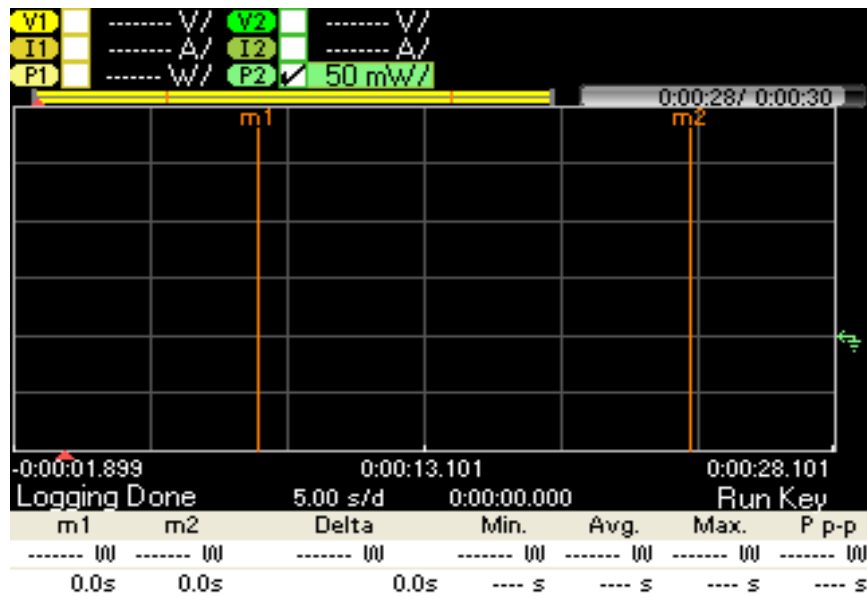


Fig. IV.23. Marcadors d'acotament

La figura IV.24 mostra els potenciòmetres que permeten manipular els marcadors i les components vertical i horitzontal.



Fig. IV.24. Potenciòmetres d'ajustament de l'analitzador de potència

#### IV.5.5. Emmagatzematge de dades

L'analitzador de potència disposa d'una memòria interna de 64MB de capacitat per a poder emmagatzemar captures de pantalla o de dades. Així mateix, també permet la connexió d'una memòria USB externa, gràcies a un connector USB 2.0 ubicat a la part frontal de l'analitzador.

La figura IV.25 mostra el connector USB.

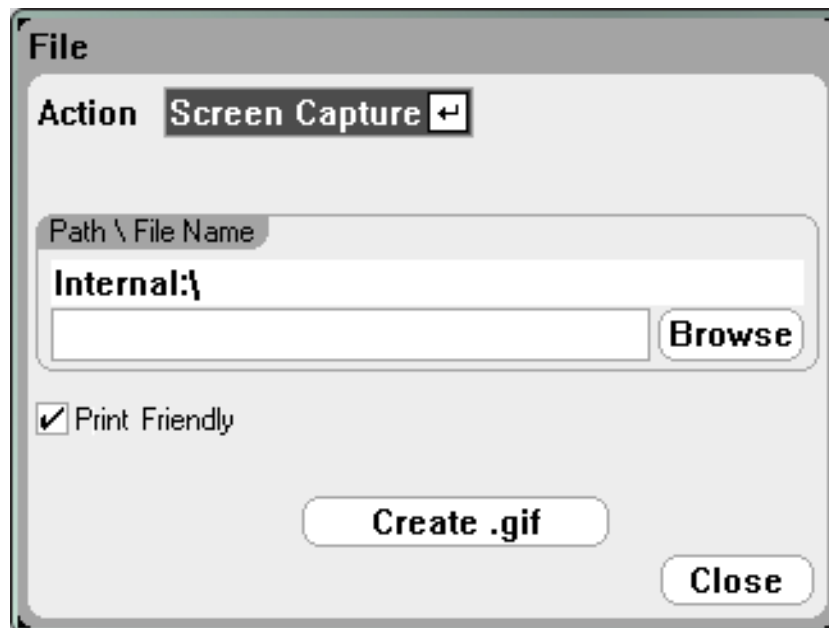


**Fig IV.25.** Connector USB 2.0

Per a poder emmagatzemar captures de pantalla a la memòria, ja sigui interna o externa, cal prémer el botó *File* (vegeu la figura IV.26) per a accedir al menú de captura de pantalla. Una vegada en aquest menú, cal seleccionar l'opció *Screen Capture* (vegeu la figura IV.27).

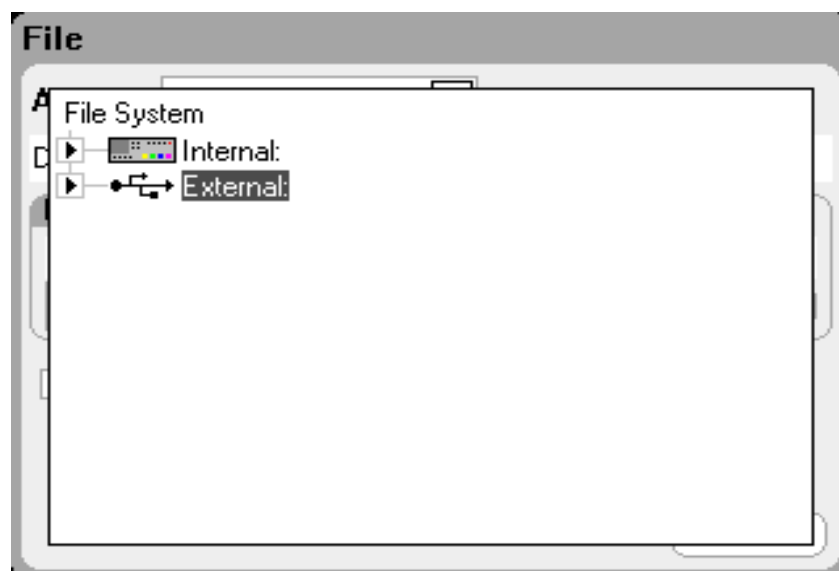


**Fig IV.26.** Botó d'accés al menú *File*



**Fig IV.27.** Accions permeses en el menú *File*

En seleccionar l'opció de capturar pantalla apareix un altre menú que permet triar la ubicació de la captura: cal triar *Internal* per a gravar la captura a l'analitzador mateix, o bé *External* per a gravar la captura o el fitxer a la memòria externa USB, tal com es mostra a la figura IV.28.



**Fig IV.28.** Menú de selecció del dispositiu d'emmagatzematge

Finalment, després de triar on es desaran les dades, cal donar-los un nom o identificador per a finalitzar el procés de captura o exportació de dades.

#### IV.5.6. Control d'errors i emergències

L'analitzador compta amb una sèrie de contingències per a situacions en què calgui deixar de subministrar alimentació als dispositius connectats.

El sistema disposa d'un botó d'aturada d'emergència, *Emergency Stop* (vegeu la figura IV.29), que apaga de manera immediata i sense retard les quadre sortides d'alimentació fins que es torna a prémer qualsevol botó, inclòs el botó d'aturada d'emergència mateix. Addicionalment, hi ha un altre sistema que permet fer l'aturada total del sistema, mitjançant els botons *On* i *Off* situats a la carcassa frontal de l'analitzador. Aquests botons, al contrari que el botó d'aturada d'emergència, aturen l'analitzador segons una configuració de retard prèvia, és a dir, que és possible programar quant de temps triga a engegar-se o apagar-se el dispositiu en prémer-los (vegeu la figura IV.30).



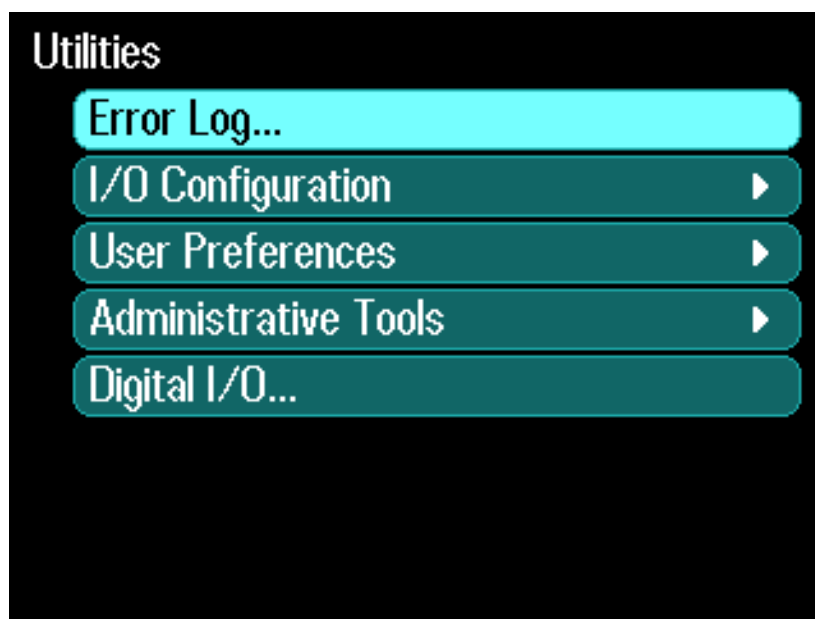
Fig IV.29. Botó d'aturada d'emergència



**Fig IV.30.** Sistema *On / Off* programable

Per últim, qualsevol error que es produeixi durant l'operació de l'analitzador de potència queda registrat a l'historial d'errors, al qual es pot accedir prement el botó *Menu* (vegeu la figura IV.18) i seguint les ordres *Utilities > Error Log* (vegeu la figura IV.31).

Els errors que apareixen a la llista es desen en ordre d'arribada, de manera que l'últim error de la llista és el més recent.



**Fig IV.31.** Menú d'historial d'errors



## ANNEX V. CONFIGURACIÓ DEL MAQUINARI UTILITZAT

Per a la realització d'aquest projecte s'han utilitzat una sèrie de dispositius IEEE802.11. Per a la part d'aquest estudi relativa a les mesures de consum, aquests dispositius s'han integrat en diversos escenaris de proves.

A la figura 1.1 del Capítol 4 es pot observar l'escenari de proves general, que està format pels elements següents: Access Point Cisco, targeta, telèfon mòbil, analitzador de potència Agilent i estacions de treball.

Aquests elements s'han configurat amb l'objectiu d'adequar-se a les diverses proves i escenaris. En aquest annex es descriu la configuració general d'aquests dispositius, mentre que en el Capítol 4 es descriu la configuració particular per als casos en què aquesta ha estat necessària.

### V.1. Access Point Cisco Aironet 1131 AG

Les característiques generals d'aquests AP (vegeu la figura V.1) es mostren a la taula V.1.



**Fig.V.1.** Access Point Cisco Aironet 1131AG.

**Taula V.1.** Característiques de l'AP Cisco Aironet 1131 AG

ACCESS POINT	
Descripció	Access Point Cisco Aironet
Model	1131 AG
Sistema operatiu	IOS 12.4(10b)JA
Certificacions	802.11 A/G/H/D/I WMM WPA/WPA2

### V.1.1. Connexió al dispositiu

L'Access Point Cisco Aironet 1131AG consta dels connectors físics següents:

- **Alimentació:** connexió de tipus jack per a una font d'alimentació de 48V DC i 380mA.
- **Consola:** connector femella RJ45 per a la connexió del cable de consola proporcionat pel fabricant.
- **Ethernet:** connector femella per a la connexió del dispositiu amb l'estació (STA) o dispositiu a partir del qual realitzarà la seva funció de punt d'accés; aquesta connexió és del tipus IEEE FastEthernet, és a dir, una connexió de 100Mbps.
- **Interfície ràdio. El dispositiu presenta dues interfícies ràdio:**
  - o **DotRadio 0: Connexió 802.11A**
  - o **DotRadio 1: Connexió 802.11G**

Tal com es pot observar, la connexió es realitza físicament pel port de consola o bé mitjançant una de les seves interfícies de xarxa, ja sigui per cable o mitjançant ràdio. La figura V.2 mostra els seus connectors.



**Fig. V.2.** Connectors de l'AP Cisco Aironet 1131 AG

## V.2. Connexió per port sèrie

La connexió per port sèrie a l'AP es realitza mitjançant un cable adaptador sèrie a consola, el qual es mostra a la figura V.3.

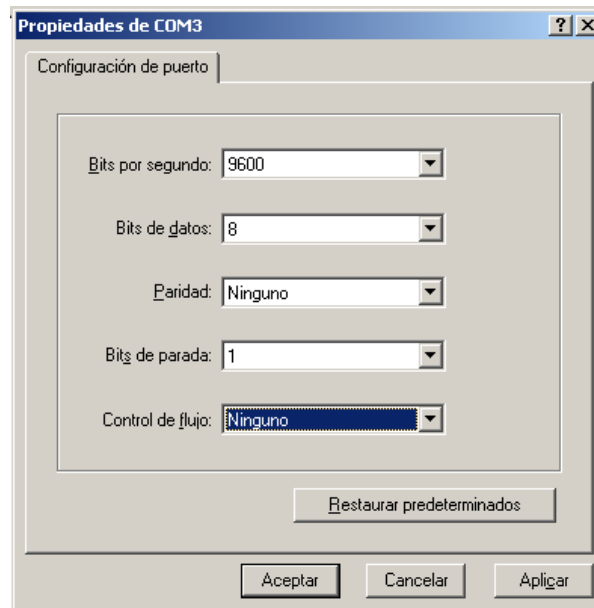


**Figura V.3.** Cable de consola DB-9 a RJ45

La connexió per consola (CLI) és l'accés de configuració més potent a l'AP, ja que permet modificar-ne totalment els paràmetres, des de l'adreça IP de les seves interfícies fins a les llistes d'accés, QoS i AAA, i també permet consultar dades d'interès com la configuració d'arrencada de l'AP.

La connexió es duu a terme en dos passos:

- En primer lloc, es connecta el cable al port COM1 de l'estació des de la qual es configura l'AP, i se'n connecta l'altre extrem al connector de consola de l'AP.
- A continuació, des de l'estació i mitjançant un programa d'emulació de terminal, es connecta amb l'AP mitjançant una configuració com la que es mostra a la figura V.4.



**Fig V. 4.** Dades de connexió sèrie de l'AP

### V.3. Connexió per Telnet

La connexió mitjançant l'aplicació Telnet es realitzarà contra l'adreça IP que s'hagi configurat tal com es mostra a l'Annex II.

Una vegada feta la connexió contra aquesta adreça IP, s'estarà accedint a la CLI de l'AP, i es podrà configurar segons la necessitat de l'escenari.

La figura V.5 mostra un exemple de connexió Telnet.

```
C:\>telnet 147.83.118.222

User Access Verification
Username: Cisco
Password:
0006f10000000000012#_
```

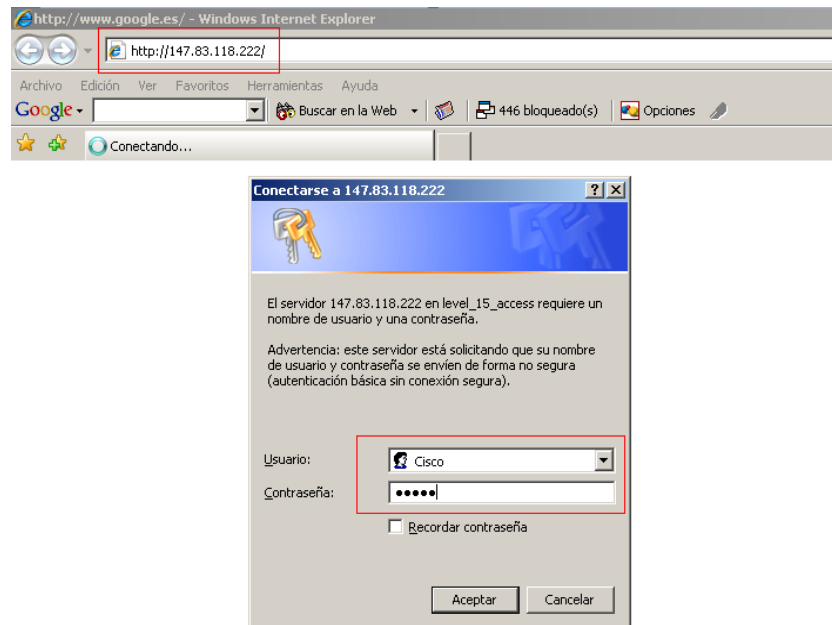
**Fig V.5.** Connexió Telnet a l'AP Cisco Aironet 1131 AG

### V.4. Connexió per http

Els AP de la sèrie 1130 disposen d'una interfície gràfica (GUI) que en facilita la configuració.

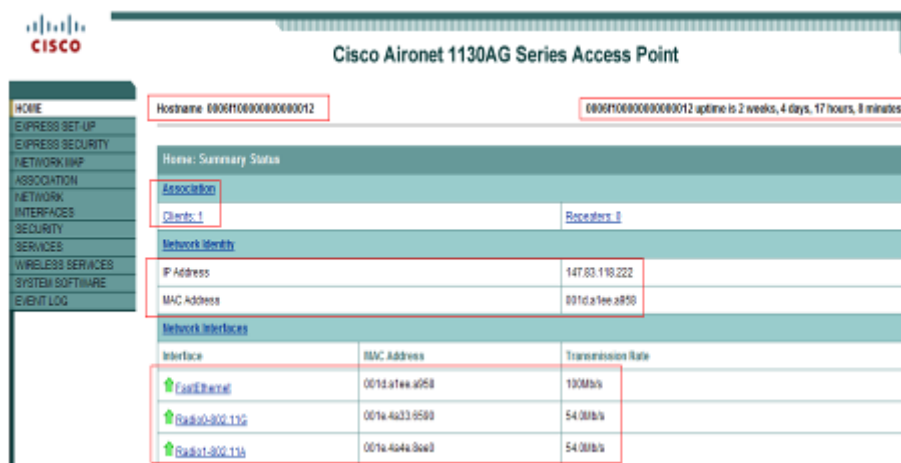
La connexió a la GUI es duu a terme mitjançant un navegador web per connexió http, introduint l'adreça IP configurada en el punt V.1.2.2 com a URL.

Una vegada autenticada la sessió apareix la GUI, tal com es pot observar a la figura V.6, en la qual s'han destacat en color vermell els camps relacionats amb l'accés.



**Fig V.6.** Diàleg d'accés

Una vegada establertes les credencials apareix la GUI, tal com es pot observar a la figura V.7, en la qual es mostra la pàgina inicial de la interfície gràfica. En aquesta figura s'han destacat alguns camps per tal d'apreciar la facilitat amb què es proporciona la informació. Es tracta, doncs, d'un entorn molt intuïtiu.



**Fig V.7.** Interfície gràfica de l'AP Cisco Aironet 1131 AG

Aquesta GUI no és tan potent com la interfície per línia d'ordres, però compleix correctament la funció d'indicar i configurar els paràmetres més bàsics del dispositiu.

#### V.4.1. Configuració dels paràmetres bàsics

En aquest apartat s'enumeren una sèrie de paràmetres bàsics que ha calgut configurar per tal que l'AP funcionés correctament durant les proves.

Atesa la potència de la interfície CLI es treballarà amb ella en la majoria de casos, tenint en compte que és molt fàcil de configurar mitjançant la interfície gràfica.

### V.5. SSID

L'identificador de xarxa sense fils SSID d'un AP Cisco Aironet es configura de manera global o específica per a cadascuna de les seves interfícies.

Per a la realització d'aquest estudi s'ha treballat amb la interfície ràdio 802.11G (denominada dotRadio 0 a la CLI de l'AP), i s'ha configurat i associat l'SSID a aquesta interfície.

Per a configurar-la cal connectar-se a l'AP mitjançant una connexió Telnet o bé per port sèrie, en el qual, després d'haver accedit a la CLI, es configura l'SSID mitjançant les ordres que es mostren a la figura V.8.

```
0006f100000000000012#configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
0006f100000000000012(config)#interface dot11Radio 0
0006f100000000000012(config-if)#ssid 0006f100000000000012
```

**Fig V.8.** Configuració de l'SSID de l'AP Cisco Aironet 1131 AG

### V.6. BVI

La bVI (*Bridge group Virtual Interface*) és una interfície virtual que permet assignar una sola adreça IP a l'AP per tal de connectar-s'hi a través de qualsevol de les seves interfícies mitjançant aquesta adreça.

Per a configurar-la cal accedir a l'AP mitjançant connexió Telnet o per port sèrie i, una vegada a la CLI, introduir les ordres que es mostren a la figura V.9.

```
0006f100000000000012#configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
0006f100000000000012(config)#interface bVI 1
0006f100000000000012(config-if)#ip address 147.83.118.222 255.255.255.0
```

**Fig V.9.** Configuració de la interfície bVI

## V.7. Interfícies físiques

No cal configurar una adreça IP per a cada interfície, ja que tal com s'ha configurat a l'Annex II, la interfície virtual bVI en facilita la configuració general.

Com a referència, l'AP Cisco Aironet 1131 AG disposa de tres interfícies: d'una banda, dues interfícies sense fils (dotRadio 0, que és una interfície 802.11G, i dotRadio 1, que és una interfície 802.11A) i, de l'altra banda, una interfície Ethernet denominada FastEthernet 0.

## V.8. QoS

Per a habilitar la qualitat del servei a la interfície ràdio 802.11G de l'AP, cal accedir-hi mitjançant Telnet o per port sèrie i seguir les ordres que es mostren a la figura V.10.

```
0006f100000000000012#configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
0006f100000000000012(config)#interface dot11Radio 0
0006f100000000000012(config-if)#dot11 qos mode wmm
```

**Fig V.10.** Habilitació de QoS a l'AP Cisco Aironet 1131 AG interfície ràdio 802.11G

En habilitar el WMM s'activa l'UAPSD, i l'AP afegeix el camp CoS (*class of service*) a les capçaleres 802.11 dels paquets.

## V.9. ACL

Una ACL és una llista d'accés en entorn Cisco, una sèrie de normes o patró que s'aplica a una o diverses interfícies.

Per a configurar una ACL a l'AP Cisco Aironet 1131 AG es pot accedir a la configuració aquesta ACL tant a través de CLI com de GUI. La figura V.11

mostra com s'ha creat un filtre per a tot el trànsit multicast, és a dir, rang 224.0.0.0/24.

Filter Name:	Bloqueo_Multicast	
Default Action:	Forward All	
<b>IP Address</b>		
Destination Address:	0.0.0.0	Mask: 0.0.0.0
Source Address:	224.0.0.0	Mask: 255.255.255.0
		Action: Block

**Fig V.11.** Configuració de l'ACL multicast

En el cas d'aquest estudi, s'ha creat un filtre que permet tot el trànsit tret del multicast, és a dir que, tal com es mostra a la figura, l'acció per defecte és *Forward All*, i l'acció aplicada al trànsit multicast és la de bloquejar tot el trànsit.

## V.10. ARP CACHE

L'habilitació d'ARP CACHE es duu a terme des de la línia d'ordres després d'haver-se connectat a l'AP a través de Telnet o per port sèrie. Cal executar les ordres que es mostren a la figura V.12, on també es pot observar destacat amb color vermell el camp **optional**, que en cas d'utilitzar-se permet deshabilitar la caché d'ARP fins que l'AP coneix totes les adreces IP dels clients que hi estan connectats.

```
0005f100000000000011#configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
0005f100000000000011(config)#dot11 arp-cache optional
```

**Fig V.12.** Habilitació d'ARP-CACHE

## V.11. Mapatge DSCP

Per a fer les proves de VoIP ha calgut mapar el trànsit UDP generat per l'aplicació SNT, de manera que s'hi apliqués CoS de veu.

Per a fer-ho cal crear una classe anomenada voz (vegeu la figura V.13) que permet crear una regla que posteriorment es podrà aplicar (vegeu la figura V.14) a una política, de manera que la classe associï el CoS de veu al filtre creat a la classe, que consisteix en un match de l'adreça MAC de la targeta de xarxa Linksys WUSB600N emprada a l'estudi.



```
0005f1000000000000011#configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
0005f1000000000000011(config)#class-map voz
0005f1000000000000011(config-cmap)#match source-address mac 001e:e5df:9dd7
```

**Fig V.13.** Creació de classe de filtratge MAC de la targeta Linksys WUSB600N

```
0005f1000000000000011#configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
0005f1000000000000011(config)#policy-map voz
0005f1000000000000011(config-pmap)#class voz
0005f1000000000000011(config-pmap-c)#set cos 6
```

**Fig V.14.** Creació de política d'assignació de CoS de veu al trànsit filtrat

## V.12. Modificació del DTIM

El DTIM es modifica fàcilment des de la línia d'ordres. Per a fer-ho cal connectar-se a l'AP mitjançant Telnet o per port sèrie i a continuació modificar a la interfície adequada (en aquest cas, la interfície sense fils 802.11G) el valor de beacon, *dtim-period*.

A la figura V.15 es mostren la sèrie d'ordres que modifiquen el valor de DTIM i es destaca amb color vermell per a aquest cas un valor de DTIM de 2.

```
0005f1000000000000011#configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
0005f1000000000000011(config)#interface dot11Radio 0
0005f1000000000000011(config-if)#beacon dtim-period 2
```

**Fig V.15.** Modificació del DTIM a l'AP Cisco Aironet 1131 AG

## V.13. Targeta de xarxa Linksys WUSB600N

Per a dur a terme les proves d'aquest estudi s'ha hagut d'utilitzar una targeta de xarxa certificada per la Wi-Fi Alliance com a WMM-PS.

A l'Annex II s'enumeren tots els dispositius de xarxa que compleixen els requisits d'aquest estudi. D'entre tots aquests dispositius s'ha triat la targeta Linksys WUSB600N (vegeu la figura V.16) perquè reuneix els requisits esmentats i per la seva disponibilitat.



**Fig V.16.** Targeta de xarxa sense fils USB Linksys WUSB600N

A la taula V.2 s'enumeren algunes de les característiques més importants d'aquesta targeta.

**Taula V.2.** Característiques de la targeta de xarxa Linksys WUSB600N

TARGETA DE XARXA	
Fabricant	Linksys
Model	WUSB600N
Chipset	Ralink 2870 USB
Driver disponible per al sistema operatiu	LINUX <sup>2</sup> /WINDOWS
Certificacions Wi-Fi Alliance	WMM, WMM-PS, IEEE802.11ABGN, WPA, WPA 2, EAP

La targeta s'utilitza en un dels ordinadors portàtils que s'han emprat per a realitzar les proves (concretament en el portàtil anomenat STA 1, les característiques del qual es descriuen a l'apartat IV.3).

### V.13.1. Instal·lació de la targeta

Per a instal·lar la targeta a l'STA 1 (vegeu l'apartat IV.3) cal instal·lar el driver proporcionat pel fabricant del chipset Ralink RT2870.

<sup>2</sup> La targeta Linksys WUSB600N té suport oficial per part del fabricant del chipset (Ralink), però no disposa d'un driver específic del fabricant de la targeta (Cisco).

Tot el procés es duu a terme sobre un sistema operatiu Ubuntu 7.10 Nautilus 6.

Cal realitzar els passos següents:

- Instal·lar el paquet `build-essential`

Cal tenir instal·lat aquest paquet per a poder compilar el driver posteriorment. S'instal·la mitjançant l'aplicació `apt` d'Ubuntu.

- o `sudo apt-get install linux-headers-generic build-essential`

- Descarregar el driver des del web de Ralink

Cal descarregar del web de Ralink el driver adequat per al chipset de la targeta. En el cas d'aquest estudi, es tracta del chipset RT2870USB.

- o `Wget http://www.ralinktech.com.tw/data/drivers/2008_0925_RT2870_Linux_STA_v1.4.0.0.tar.bz2`

- Descomprimir i instal·lar el driver

Cal descomprimir el fitxer descarregat i, des del directori que s'ha creat, editar el fitxer `config.mk`, que es troba en el subdirectori `os/linux`.

- o `sudo tar -xvzf 2008_0925_RT2870_Linux_STA_v1.4.0.0.tar`

Cal modificar els camps següents del fitxer `config.mk`:

- o `HAS_NATIVE_WPA_SUPPLICANT_SUPPORT`
- o `HAS_WPA_SUPPLICANT`

Adicionalment, cal afegir en el fitxer `rt2870.h`, després de la línia:

- o `#define RT2870_USB_DEVICES`

la línia següent per a indicar el dispositiu (en aquest cas, la targeta Linksys WUSB600N):

- o `{USB_DEVICES(0x1737,0x0071)}`

- Una vegada fets aquests passos cal compilar el driver. Per a fer-ho, cal executar les ordres següents des del directori del driver:

- o `sudo make`
- o `sudo make install`

- A continuació, cal copiar el fitxer *RT2870STA.dat* generat en compilar al directori */etc/wireless*:
  - o `sudo cp RT2870STA.dat /etc/wireless/RT2870STA/`
- Verificació de la instal·lació

Cal connectar el dispositiu de xarxa USB a l'ordinador portàtil STA 1 i comprovar que funciona correctament amb les ordres **ifconfig** i **iwconfig**, tal com es mostra a la figura V.17.

```

root@aesir-laptop: /home/aesir
Archivo Editar Ver Terminal Solapas Ayuda
aesir@aesir-laptop:~$ sudo su
[sudo] password for aesir:
root@aesir-laptop:/home/aesir# ifconfig ra0
ra0      Link encap:Ethernet direcciónHW 00:1E:E5:DF:9D:D7
        inet6 addr: fe80::21e:e5ff:fedf:9dd7/64 Scope:Link
        UP DIFUSIÓN CORRIENDO MULTICAST MTU:1500 Metric:1
        RX packets:870 errors:0 dropped:0 overruns:0 frame:0
        TX packets:928 errors:0 dropped:0 overruns:0 carrier:0
        colisiones:0 txqueuelen:1000
        RX bytes:124327 (121.4 KB)  TX bytes:76584 (74.7 KB)

root@aesir-laptop:/home/aesir# iwconfig ra0
ra0      RT2870 Wireless  ESSID:"PC1"  Nickname:"RT2870STA"
        Mode:Auto  Frequency=2.412 GHz  Access Point: Not-Associated
        Bit Rate:1 Mb/s
        RTS thr:off  Fragment thr:off
        Encryption key:off
        Link Quality=10/100  Signal level:-256 dBm  Noise level:-77 dBm
        Rx invalid nwid:0  Rx invalid crypt:0  Rx invalid frag:0
        Tx excessive retries:0  Invalid misc:0  Missed beacon:0

root@aesir-laptop:/home/aesir# █

```

**Fig V.17.** Verificació de la instal·lació de la targeta Linksys WUSB600N

A continuació, cal fer una prova de connectivitat amb l'ordre **ping** per tal de comprovar que la comunicació de la targeta sigui correcta. La figura V.18 mostra el resultat d'aquesta prova.

```

root@aesir-laptop:/home/aesir# ping 147.83.118.222
PING 147.83.118.222 (147.83.118.222) 56(84) bytes of data.
64 bytes from 147.83.118.222: icmp_seq=1 ttl=255 time=1012 ms
64 bytes from 147.83.118.222: icmp_seq=2 ttl=255 time=1002 ms
64 bytes from 147.83.118.222: icmp_seq=3 ttl=255 time=1002 ms
64 bytes from 147.83.118.222: icmp_seq=4 ttl=255 time=1002 ms
64 bytes from 147.83.118.222: icmp_seq=5 ttl=255 time=1002 ms

--- 147.83.118.222 ping statistics ---
6 packets transmitted, 5 received, 16% packet loss, time 5010ms
rtt min/avg/max/mdev = 1002.121/1004.223/1012.481/4.225 ms, pipe 2
root@aesir-laptop:/home/aesir#

```

**Fig V.18.** Prova de connectivitat mitjançant ping

### V.13.1.1. Habilitació de paràmetres QoS

Per a habilitar la QoS a la targeta de xarxa, cal emprar les ordres **ifconfig**, **iwconfig** i **iwpriv**.

L'ordre **iwpriv** modifica directament paràmetres del driver del dispositiu segons especificacions del fabricant, és a dir, que és dependent del fabricant i no funcionarà de la mateixa manera per a tots els dispositius, encara que comparteixin el mateix chipset.

L'ordre **iwconfig** es comporta de manera anàloga a **ifconfig**, però és una ordre específica per a dispositius de xarxa sense fils. Aquesta ordre és part del sistema i, per tant, al contrari que l'ordre **iwpriv**, és comuna per a tots els dispositius sense fils.

Cal tenir en compte que mitjançant les ordres i instruccions següents és possible configurar qualsevol dispositiu USB basat en el chipset RT2870 de la companyia Ralink.

Les instruccions fan referència a una distribució Ubuntu Linux 7.10 Nautilus 6 amb kernel 2.6.24 o superior.

Les ordres per a la configuració de QoS són:

- *iwpriv ra0 set wmmcapable 1*

Amb l'ordre anterior s'haurà habilitat en el sistema WMM.

- *iwpriv ra0 set PSMMode {CAM, FAST\_PSP, MAX\_PSP}*

Mitjançant aquesta ordre s'habilita el mode d'estalvi de potència PSM i es defineix CAM com a mode d'estalvi de potència nul, FAST\_PSP com a mode d'estalvi de potència estàndard (Listen Interval de 200 ms) i MAX\_PSP com a mode màxim d'estalvi de potència (Listen Interval de 1000 ms), en què Listen Interval és el temps d'escolta abans d'entrar en mode PSM.

El període d'inactivitat es pot modificar amb l'ordre següent:

- *iwconfig ra0 power period X (la targeta desperta cada X ms)*

### V.13.2. Habilitació del mode monitor

Per a capturar el trànsit de beacon 802.11 mitjançant l'aplicació Wireshark o Kismet cal fer treballar la targeta de xarxa en mode monitor.

La figura V.19 mostra el procés d'activació del mode monitor, i també la seva verificació mitjançant l'ordre **iwconfig**.

```
root@aesir-laptop:/home/aesir# iwconfig ra0 mode monitor
root@aesir-laptop:/home/aesir# iwconfig ra0
ra0      RT2870 Wireless ESSID:""  Nickname:"RT2870STA"
        Mode:Monitor  Frequency=2.427 GHz  Access Point: 00:0F:24:D1:5A:20
        Bit Rate=48 Mb/s
        RTS thr:off   Fragment thr:off
        Encryption key:off
        Link Quality=94/100  Signal level:-66 dBm  Noise level:-87 dBm
        Rx invalid nwid:0  Rx invalid crypt:0  Rx invalid frag:0
        Tx excessive retries:0  Invalid misc:0  Missed beacon:0
```

**Fig V.19.** Habilitació del mode monitor

## V.14. Ordinador portàtil Sony VAIO S5M

L'ordinador Sony VAIO S5M correspon a l'estació de proves STA 1. L'STA 1 és l'estació de mesura de potència de l'escenari, és a dir, la màquina sobre la qual s'han habilitat les diverses opcions d'administració d'energia o estalvi de potència segons els diversos escenaris. La taula V.3 en mostra les característiques més importants.

**Taula V.3.** Característiques de l'STA 1

STA 1	
Descripció	Ordinador portàtil Sony VAIO
Model	S5M
Bateria	VGP-BPS2B 11.1V 4800mAh
CPU	Intel Pentium M 740 1,73 GHz
Sistema operatiu	Ubuntu 7.10 Nautilus 6
Targeta de xarxa	Linksys WUSB600N

La figura V.20 mostra una fotografia de l'STA 1.



**Fig V.20.** STA 1. Portàtil Sony VAIO S5M

Per a configurar les característiques elèctriques a l'analitzador de potència s'ha seguit la informació que proporciona el fabricant. La figura V.21 mostra una fotografia del carregador de dispositiu amb les seves característiques elèctriques.



**Fig V.21.** Característiques elèctriques de l'STA 1

Tal com es pot veure a la figura, l'STA 1 té uns requisits d'alimentació elèctrica de 19.5V i 4,7A.

Adicionalment, durant les proves s'ha mantingut la brillantor del monitor al seu nivell màxim.

## V.15. Telèfon Nokia N95

El telèfon mòbil Nokia N95 (vegeu la figura V.22) és l'estació de proves de telefonia WIFI STA 4. Mitjançant la connexió d'aquest dispositiu ha estat possible comparar els resultats de l'STA 1 amb els obtinguts amb un terminal de característiques molt diferents (i.e un portàtil vs un telèfon mòbil).



**Fig V.22.** Telèfon mòbil Nokia N95.

La taula V.4 en mostra les característiques més significatives d'aquest terminal.

**Taula V.4.** Característiques de l'STA 4

STA 4	
Descripció	Telèfon mòbil Wifi Nokia
Model	N-95
Bateria	BL-5F 3.7V 950mAh



Per a configurar les característiques elèctriques a l'analitzador de potència s'ha seguit la informació que proporciona el fabricant. La figura V.23 mostra una fotografia del carregador de dispositiu amb les seves característiques elèctriques.



**Fig V.23.** Característiques elèctriques de l'STA 4

Tal com es pot veure a la figura, l'STA 4 té uns requisits d'alimentació elèctrica de 5.0V i 890mA.

Per a dur a terme aquest estudi no ha calgut configurar el dispositiu més enllà de l'habilitació dels diversos modes de connexió sense fils, WIFI o 3G. Per a fer-ho, ha calgut seguir els passos següents:

- **Activació WIFI:** Menú > Eines > Ajustaments > Connexió > WLAN
- **Activació/desactivació UMTS/GSM:** Menú > Eines > Ajustaments > Telèfon > Xarxa

La figura V.24 mostra el menú d'habilitació de la interfície WLAN, mentre que a la figura V.25 es pot veure el menú d'habilitació de la interfície GSM/UMTS.



**Fig V.24.** Menú d'habilitació WLAN STA 4



**Fig V.25.** Menú d'habilitació GSM/UMTS STA 4

La resta de paràmetres s'han mantingut tal com els subministra el fabricant per defecte, tret de la brillantor de la pantalla, que s'ha mantingut al seu nivell màxim tal com s'ha configurat per a l'estació de proves STA 1.

## **V.16. Ordinador portàtil BENQ S72**

L'ordinador BENQ S72 correspon a l'estació de proves STA 2, la funció de la qual ha consistit a monitorar el trànsit de xarxa de l'escenari mitjançant el programari d'anàlisi i captura de trànsit Wireshark i Kismet.

Atesa la seva funció, no ha calgut aplicar-hi cap modificació o configuració addicional per a utilitzar-la durant aquest estudi, a banda de les qüestions relacionades amb el programari, que es tracten a l'Annex VI.

La taula V.5 mostra les característiques de l'STA 2.

**Taula V.5.** Característiques de l'STA 2

STA 2	
Descripció	Ordinador portàtil BENQ
Model	S72
Bateria	I304RJ 10.8V 4800 mAh
CPU	Intel Pentium M 730 1,60 GHz
Sistema operatiu	Ubuntu 7.10 Nautilus 6
Targeta de xarxa	Intel PRO 2200 BG

La figura V.26 mostra una fotografia de l'STA 2.



**Fig V.26.** STA 2. Portàtil BENQ S72

### **V.17. Ordinador portàtil Sony VAIO FW11M**

L'ordinador portàtil Sony VAIO FW11M correspon a l'estació de proves STA 3, que durant aquest estudi ha servit com a client FTP i servidor de SNT.

Atesa la funció de l'estació durant les proves, no ha calgut aplicar-hi cap configuració de maquinari o programari a banda de les qüestions relatives al client FTP o el servidor de SNT, que es tracten amb detall a l'Annex VI. La taula V.6 mostra les característiques de l'STA 3.

**Taula V.6.** Característiques de l'STA 3

STA 3	
Descripció	Ordinador portàtil Sony VAIO
Model	FW11M
Bateria	BPS13A/S 11,1V 4800mAh
CPU	Intel Core2 Duo P8200 2,26 GHz
Sistema operatiu	Windows Vista Home SP1
Targeta de xarxa	Intel WiFi Link 5100

La figura V.27 mostra una fotografia de l'STA 3.



**Fig V.27.** STA 2. Portàtil Sony VAIO FW11M

## V.18. Analitzador de potència Agilent N6705A

La configuració de l'analitzador de potència Agilent N6705A es documenta a l'Annex IV, ja que per la seva complexitat i potència aquest dispositiu requereix un document independent per a explicar-ne la configuració i les possibilitats per als diversos escenaris d'aquest estudi.

Per a la connexió a les estacions de proves en què s'han dut a terme les mesures de potència, STA 1 i STA 4, ha calgut confeccionar un cable de connexió, que es mostra a la figura V.28.



**Fig V.28.** Cable de connexió STA 4 a l'analitzador de potència

Les estacions es connecten mitjançant el cable de connexió a la sortida 2 de l'alimentació de l'analitzador de potència.

La figura V.29 mostra l'analitzador de potència connectat a l'STA 4.



**Fig V.29.** Connexió de l'analitzador de potència Agilent N6705A a l'STA 4



## ANNEX VI. CONFIGURACIÓ DEL PROGRAMARI UTILITZAT

Per a la realització d'aquest projecte ha calgut emprar una sèrie d'aplicacions per a monitorar la xarxa o generar trànsit.

Aquestes aplicacions s'han configurat per a adequar-les als diversos escenaris que es detallen en el Capítol 4, on també s'especifica, quan escau, la configuració específica per a cada escenari, mentre que la configuració general per als diversos escenaris s'analitza en aquest annex.

Es poden diferenciar dos tipus d'aplicacions. D'una banda, les relacionades amb el monitoratge, que es configuren per a capturar el trànsit sense fils de la xarxa i, de l'altra, les aplicacions de simulació de trànsit, que permeten modelar segons els requeriments dels diversos escenaris el trànsit que travessa la xarxa de proves (com ara l'aplicació Iperf, que permet modelar el trànsit UDP perquè simuli VoIP).

La taula 1.2 del Capítol 4 inclou una relació del programari utilitzat en aquest estudi.

### VI.1. Wireshark

Aquesta aplicació permet monitorar el trànsit que travessa una xarxa, mitjançant una de les interfícies de l'estació en què s'executa.

En aquest estudi, Wireshark s'ha instal·lat i configurat per a capturar trànsit WiFi 802.11 a l'estació de monitoratge STA 2, les característiques de la qual es detallen a l'Annex II.

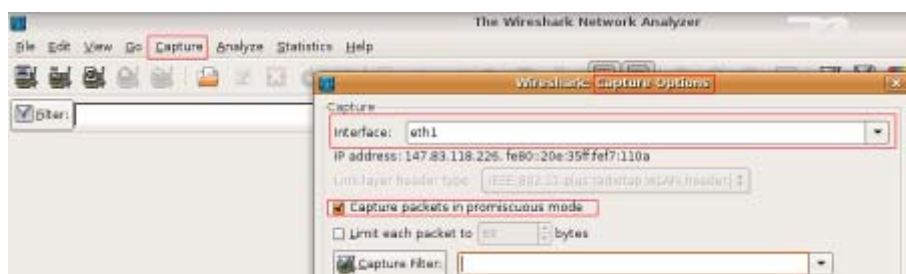
Wireshark requereix que s'habiliti la funció monitor a la targeta de xarxa mitjançant la qual accedeix al mitjà per a capturar trànsit 802.11, per a examinar els beacon i poder analitzar els paràmetres de QoS que transporten.

Per a aquest estudi s'ha utilitzat la versió 1.0.7 de Wireshark, que ve instal·lada per defecte a la versió de sistema operatiu Ubuntu 7.10 Nautilus 6 (el SO emprat en aquest estudi). No ha calgut instal·lar cap paquet extern ni modificar cap fitxer addicional per al seu funcionament.

La configuració per defecte de Wireshark permet dur a terme la captura de trànsit de la xarxa seguint la sèrie de passos que es descriuen a continuació:

- En primer lloc, cal accedir a la pestanya *Capture* i seleccionar *Options* en el menú.
- En la finestra que s'obté, cal deixar tots els camps per defecte tret del desplegable *Interface*, on s'ha de seleccionar la interfície per la qual es

monitorarà el trànsit de la xarxa i, a continuació, comprovar que estigui activada l'opció *Capture packets in promiscuous mode*. Aquesta opció permet capturar el trànsit de tot el segment LAN al qual estigui connectada la interfície. Si no s'activa aquesta opció, només es captura el trànsit que travessa la targeta. A la figura VI.1 es destaquen aquestes opcions amb un requadre vermell.



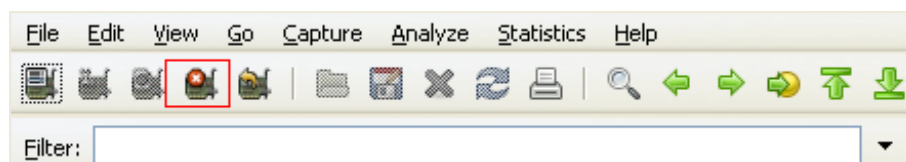
**Fig VI.1.** Menú *Options* de l'aplicació Wireshark

- A continuació, en aquest mateix menú *Options* se selecciona l'opció *Start* (vegeu la figura VI.2) per tal que l'aplicació comenci a capturar trànsit de la xarxa.



**Fig VI.2.** Opció *Start* del menú *Options* de l'aplicació Wireshark

- Per a aturar el procés de monitoratge i poder analitzar el trànsit capturat cal seleccionar l'opció *Stop the running live capture* de la barra d'eines (vegeu la figura VI.3, on es destaca amb vermell aquesta opció).



**Fig VI.3.** Opció d'aturada de captura de l'aplicació Wireshark

El resultat de la captura es mostra a la finestra de l'aplicació que es pot veure a la figura VI.4. En aquesta figura es destaca la informació transportada en un dels beacon de l'Access Point Cisco Aironet 1131 AG utilitzat a l'escenari de proves.



En el Capítol 4, per exemple, s'analitza en detall la informació obtinguda a les captures en relació amb cada escenari concret.

11	0.535953	Cisco-L1_3a:5a:94	Broadcast	IEEE 802 Beacon Frame, SN=3484, FN=0, Flags=..
12	0.651965	Cisco_33:65:90	Broadcast	IEEE 802 Beacon Frame, SN=2663, FN=0, Flags=..
13	0.840015	Cisco-L1_3a:5a:94	Broadcast	IEEE 802 Beacon Frame, SN=3487, FN=0, Flags=..
14	1.143965	Cisco-L1_f1:7f:ba	Broadcast	IEEE 802 Beacon frame, SN=1353, FN=0, Flags=..
15	1.143967	Cisco-L1_b8:46:63	Broadcast	IEEE 802 Beacon frame, SN=2445, FN=0, Flags=..
16	1.167992	Cisco-L1_3a:5a:94	Broadcast	IEEE 802 Beacon frame, SN=3490, FN=0, Flags=..
17	1.235540	Cisco-L1_f1:7f:ba	Broadcast	IEEE 802 Beacon frame, SN=1354, FN=0, Flags=..
18	1.263983	Cisco-L1_3a:5a:94	Broadcast	IEEE 802 Beacon frame, SN=3491, FN=0, Flags=..

# Frame 12 (207 bytes on wire, 207 bytes captured)				
# Radiotap Header v0, Length 25				
# IEEE 802.11 Beacon frame, Flags: .....				
# IEEE 802.11 wireless LAN management frame				

**Fig VI.4.** Captura de trànsit mitjançant l'aplicació Wireshark

## VI.2. Kismet

Kismet és una aplicació de detecció de xarxa passiva. A diferència d'altres aplicacions, no injecta trànsit artificial a la xarxa per a monitorar-la, i pot detectar passivament clients i punts d'accés sense fils.

Durant la realització de l'estudi, el treball amb aquesta aplicació ha permès verificar el tipus de trànsit associat a la xarxa monitorada, així com la vinculació dels clients als punts d'accés associats a cada escenari.

Aquesta aplicació es pot descarregar i instal·lar fàcilment mitjançant l'ordre **apt-get** de la distribució de Linux Ubuntu 7.10 Nautilus 6 (aquest és el SO instal·lat a l'estació de monitoratge STA 2, tal com es pot consultar a l'Annex I).

La figura VI.5 mostra com s'utilitza l'ordre **apt-get** i posteriorment s'edita el fitxer de configuració per a instal·lar i fer funcionar l'aplicació.

```
sudo apt-get install kismet
sudo gedit /etc/kismet/kismet.conf
```

**Fig VI.5.** Instal·lació i configuració de l'aplicació Kismet

Una vegada instal·lada l'aplicació caldrà modificar una sèrie de línies en el fitxer que s'ha obert, *kismet.conf*. Aquestes línies corresponen a la informació de la interfície que es vol establir per a monitorar la xarxa. A la figura VI.6 es destaquen les línies que ha calgut modificar.

```
# YOU MUST CHANGE THIS TO BE THE SOURCE YOU WANT TO USE
source=ipw2200,eth1,wlan
```

**Fig VI.6.** Informació d'interfície de xarxa del fitxer de configuració *kismet.conf*

Una vegada instal·lada i configurada l'aplicació, cal executar-la des de la línia d'ordres amb l'ordre **kismet**, sempre amb permisos d'administrador. Per a obtenir aquests permisos, cal executar l'ordre **sudo kismet**.

Una vegada iniciada l'aplicació, apareix la finestra en què es mostra la informació de xarxa que està capturant Kismet.

Cal tenir en compte que l'aplicació habilita de manera automàtica el mode monitor a la targeta de xarxa a través de la qual monitora la xarxa.

La figura VI.7 mostra el menú gràfic de Kismet.

Network List (SSID)						Info
Name	T	W	Ch	Pkts	Flags	IP Range
! 0005f100000000000011	A	N	006	73		0.0.0.0
! 0006f100000000000012	A	N	001	49	T4	147.83.118.228
! Lab_tel_rec	A	N	011	76	U4	147.83.119.53
bampla_lab301	A	Y	001	2		0.0.0.0
hpsetup	H	N	006	1		0.0.0.0
! secu	A	O	009	8		0.0.0.0

Ntwrks	6
Pckets	242
Cryptd	0
Weak	0
Noise	0
Discrd	0
Pkts/s	17
Elapsd	00:00:12

Status
Connected to Kismet server version 2007.01.R1 build 20050815211952 on localh
Sorting by SSID
Found IP 147.83.119.53 for Lab_tel_rec::00:0C:41:3A:5A:92 via UDP

**Fig VI.7.** Menú gràfic de Kismet

Per a accedir al menú d'opcions de l'aplicació cal prémer la tecla ?. La figura VI.8 mostra les diverses opcions que ofereix Kismet, mentre que a la figura VI.9 es pot veure la informació que s'obté en seleccionar un dels Access Point utilitzats durant l'estudi, amb SSID 0006f100000000000012.

```
KISMET PANELS INTERFACE
QUICK REFERENCE
Key  Action
e    List Kismet servers
z    Toggle fullscreen zoom of network view
m    Toggle muting of sound and speech
t    Tag (or untag) selected network
g    Group tagged networks
u    Ungroup current group
c    Show clients in current network
L    Lock channel hopping to the current network channel
H    Return to normal channel hopping
+/-  Expand/collapse groups
^L   Force a screen redraw.

POPOP WINDOWS
h    Help (What you're looking at now)
n    Name current network
i    Detailed information about selected network
s    Sort network list
l    Show wireless card power levels
d    Dump printable strings
r    Packet rate graph
a    Statistics
p    Dump packet type
f    Follow network center
w    Track alerts
x    Close popup window

Q    Quit
```

**Fig VI.8.** Menú d'opcions de l'aplicació Kismet

```

SSID      : 0006f100000000000012
Server    : localhost:2501
BSSID     : 00:1E:4A:33:65:90
Carrier   : IEEE 802.11b
Manuf     : Unknown
Max Rate  : 18.0
BSS Time  : 27ffc9f18c
Max Seen  : 11000 kbps
First     : Thu Apr 16 16:37:26 2009
Latest    : Thu Apr 16 16:45:16 2009
Clients   : 2
Type      : Access Point (infrastructure)
Info      : 0006f10000000000\000\000\000\000
Channel   : 1

```

**Fig VI.9.** Exemple d'informació d'Access Point obtinguda amb Kismet

Finalment, per a sortir de l'aplicació cal emprar la combinació de tecles Shift+q.

### VI.3. Iperf

Iperf és una eina que permet mesurar paràmetres com ara el *jitter* o el *throughput* d'una connexió extrem a extrem.

La seva capacitat per a generar trànsit bidireccional UDP amb mida dels paquets variable ha permès fer una aproximació al trànsit de xarxa d'una conversa telefònica VoIP.

Iperf és una aplicació client-servidor, és a dir, que en un extrem de la comunicació s'ha de configurar un servidor d'Iperf (vegeu la figura VI.10) i a l'altre extrem de la connexió s'han de configurar un o diversos clients (vegeu la figura V.11).

Les dades de configuració per a l'aproximació al model de conversa VoIP han estat les següents:

- Mida de paquets [-l]: 250
- Durada de la transmissió [-t]: -t 120
- Port emprat per la transmissió [-p]: -p 5001
- Protocol UDP [-u]: -u
- Habilitació client [-c]: -c
- Habilitació servidor [-s]: -s
- Interval de captura [-i]: 5
- Prova bidireccional [-d]: -d

```

root@aesir-laptop:/home/aesir/Desktop/jperf# iperf -s -u -p 5001 -i 5
-----
Server listening on UDP port 5001
Receiving 1470 byte datagrams
UDP buffer size: 108 KByte (default)
-----
[ 3] local 147.83.118.227 port 5001 connected with 147.83.118.228 port 1351
[ 3] 0.0- 5.0 sec 2.05 MBytes 3.43 Mbits/sec 0.711 ms 16/ 8594 (0.19%)

```

**Fig VI.10.** Exemple de servidor d'iperf.

```

C:\>iperf -c 147.83.118.227 -u -i 5 -p 5001 -t 120 -l 250 -b 54M
-----
Client connecting to 147.83.118.227, UDP port 5001
Sending 250 byte datagrams
UDP buffer size: 8.00 KByte (default)
-----
[1912] local 147.83.118.228 port 1349 connected with 147.83.118.227 port 5001
[ ID] Interval      Transfer      Bandwidth
[1912] 0.0- 5.0 sec 3.29 MBytes  5.52 Mbits/sec

```

**Fig VI.11.** Exemple de client d'iperf.

Per a la realització d'aquest estudi s'ha emprat la versió 2.0.3 d'aquesta aplicació.

## VI.4. Vsftp

Vsftp és una aplicació de servidor FTP per a sistemes Linux. En particular, per a l'entorn de proves d'aquest estudi s'ha instal·lat en una distribució Ubuntu Linux 7.10 Nautilus 6.

La instal·lació d'aquesta aplicació es duu a terme mitjançant APT, l'eina avançada d'empaquetament d'Ubuntu.

La figura VI.12 mostra l'ordre que permet instal·lar el servidor FTP.

```

root@aesir-laptop:/home/aesir/Desktop/jperf# apt-get install vsftp

```

**Fig. VI.12.** Instal·lació del servidor FTP Vsftp en Ubuntu 7.10 Nautilus 6

A continuació cal modificar el fitxer de configuració del servidor FTP segons les necessitats de l'escenari. Per a aquest estudi ha calgut modificar el fitxer que es pot trobar al directori `/etc/vsftpd.conf`.

La figura VI.13 mostra algunes de les opcions de configuració més interessants d'aquest fitxer, com ara `anonymous_enable` (que en aquest cas s'ha configurat a `NO` per motius de seguretat) o `local_enable`, que permet l'accés d'usuaris locals a les seves carpetes privades.

```
# Allow anonymous FTP? (Beware - allowed by default if you comment this out).  
#anonymous_enable=NO  
#  
# Uncomment this to allow local users to log in.  
local_enable=YES  
#  
# Uncomment this to enable any form of FTP write command.  
write_enable=YES
```

**Fig. VI.13.** Exemple de fitxer de configuració *vsftpd.conf*

En cas d'haver de reiniciar el servei, es pot fer fàcilment tal com s'indica a la figura VI.14.

```
root@aesir-laptop:/home/aesir/Desktop/jperfe# /etc/init.d/vsftpd restart  
* Stopping FTP server: vsftpd  
* Starting FTP server: vsftpd
```

**Fig. VI.14.** Reinici del servei vsftpd

El servidor queda configurat a l'espera de connexions.

## VI.5. Simple Network Tester

Simple Network Tester (SNT) és una aplicació que combina una sèrie d'eines per a obtenir informació de l'estat d'una xarxa, generar-hi trànsit o comprovar l'estat d'una connexió extrem a extrem.

Per a aquest estudi s'ha utilitzat l'opció de proves de VoIP per a obtenir informació detallada del jitter, la pèrdua de paquets i l'amplada de banda ocupada durant les proves.

Cal recordar que el jitter és la variació del retard, és a dir, la diferència entre el temps d'arribada real d'un paquet i el temps previst. Es recomana que tingui un valor inferior als 100ms.

Pel que fa al retard, és a dir, al temps que triga un paquet a arribar d'un extrem de la connexió a l'altre, no és recomanable que superi els 150ms.

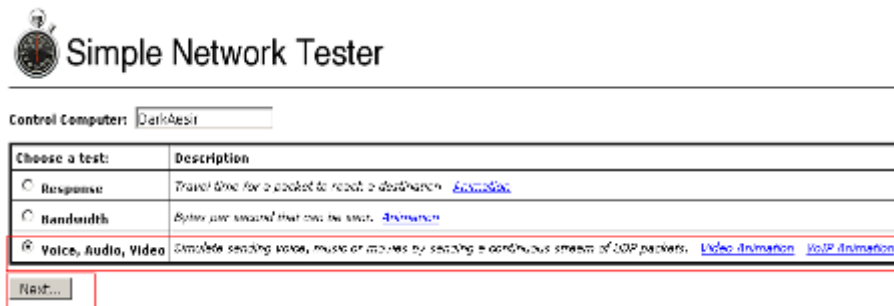
Finalment, la pèrdua de paquets en un entorn de comunicacions de VoIP no ha de superar el 5%.

Per a la realització d'aquestes proves s'ha instal·lat la versió 3.0 d'SNT a les estacions de proves STA 1 i STA 3, sobre els sistemes operatius Windows XP SP3 i Windows Vista SP1 respectivament.

Aquesta aplicació utilitza una interfície gràfica HTTP, però també disposa d'una versió de línia d'ordres.

Es pot instal·lar fàcilment descarregant-ne l'executable des de la seva pàgina web i seguint el procés d'instal·lació que s'ofereix per defecte.

Una vegada instal·lada, cal anar al menú *Inicia > Tots els programes > Simplenet Software* i executar Simple Net Tester per obrir en el navegador el menú que es mostra a la figura VI.15, on es destaca l'opció que permet dur a terme proves de VoIP.



**Fig. VI.15.** Menú d'inici d'SNT

Les proves de veu s'han dut a terme amb una configuració que simula el còdec G.711 de compressió d'àudio de l'ITU-T.

El còdec G.711 té les característiques següents:

- Mostres de so de 8 bits.
- Taxa de mostratge de 8.000 mostres per segon.
- Flux de dades de 64 Kbps.

Per tant, per a configurar l'aplicació de manera que simuli aquesta informació, cal tenir en compte la mida de càrrega dels paquets (de 160 bytes per a encaixar dues mostres de 10ms) i afegir-hi la mida de capçalera RTP que tindria un paquet de VoIP transmès en una xarxa real (12 bytes).

A l'hora de configurar l'aplicació no cal tenir en compte les capçaleres UDP i IP, ja que l'aplicació mateixa ja les té en compte per a mostrar els resultats.

La figura VI.16 mostra el menú de configuració de proves per a VoIP.

#### Streaming Test: Voice over IP and Video

Receiving Computer: panoramix (This is the control computer.)

Transmitting Computer:   Verify remote computer has Simple Network Tester.

Choose the kind of traffic you want to simulate:

**Voice over IP (VoIP)**  
 Simulated VoIP (RTP) using 160-byte voice samples & 12 byte headers sent 50 times / second. 200 millisecond jitter buffer.

**Video**  
 Simulate an MPEG-2 stream of 2M bits per second with 1K byte packets. 5 second jitter buffer duration.

**Custom stream**

Packet size:  Bytes. (Between 100 and 1500 bytes.)

Flow rate:  M Bits / second (Under 10 M Bits / second.)

Jitter buffer duration:  milliseconds  
 125 packets per second.

DiffServ code point:  [Help](#)

Layer 2 priority (802.1p):  [Help](#)

Test time limit is	<input type="text" value="120"/> seconds.
List progress every	<input type="text" value="1"/> seconds
Refresh the screen every	<input type="text" value="10"/> seconds

**Fig. VI.16.** Menú de proves de VoIP de l'aplicació SNT

L'aplicació disposa d'una opció anomenada *Response* que permet obtenir el càlcul del retard, entre d'altres. A la figura VI.17 es mostren destacades les opcions *Response* i *Bandwidth*. Aquesta darrera opció permet mesurar l'amplada de banda consumida per una transmissió simulada.



Control Computer:

Choose a test:	Description
<input type="radio"/> <b>Response</b>	Travel time for a packet to reach a destination. <a href="#">Animation</a>
<input type="radio"/> <b>Bandwidth</b>	Bytes per second that can be sent. <a href="#">Animation</a>
<input checked="" type="radio"/> <b>Voice, Audio, Video</b>	Simulate sending voice, music or movies by sending a continuous stream of UDP packets. <a href="#">Video Animation</a> <a href="#">VoIP Animation</a>

**Fig. VI.17.** Opcions de mesura de retard i amplada de banda de l'aplicació SNT



## ANNEX VII. Modificacions al codi del dimoni UMIP

En aquest annex es documenten tots els canvis implementats en el codi del dimoni UMIP, tant els realitzats a [11] com els d'aquest projecte.

Per a la realització d'aquest projecte ha calgut modificar els següents fitxers del codi font:

**Taula VII.1.** Fitxers modificats en aquest projecte

Fitxer	Ubicació
ip6mh.h	/include/netinet
pcap.h	/include/netinet (només cal afegir-lo)
proc_sys.c	/src/
proc_sys.h	/src/
movement.c	/src/
movement.h	/src/
mn.c	/src/
mn.h	/src/
ha.c	/src/

A continuació s'inclou el fitxer .diff que mostra les diferències entre el codi del dimoni UMIP per la versió 0.4 i el codi implementat en aquest treball, per a cadascun dels fitxers de la taula VII.1.

Per a generar aquest fitxer s'ha utilitzat el programari gratuït ExamDiff per a Windows. Se'n pot trobar més informació a la seva pàgina web: <http://www.prestosoft.com/examdiff/free/>.

Com a guia per a l'usuari, cal tenir en compte que les línies amb el signe < davant són les afegides, mentre que les línies amb el signe > són les suprimides.

La numeració que s'aprecia abans de cada canvi indica la línia d'inici i la línia final de la modificació (en cas d'existir més d'una línia).

## VII.1 Modificacions a *ip6mh.h*

```

16c16
< uint8_t ip6mh_reserved; /* Reserved */
71d70
< #define IP6_MH_BU_DORM 0x0100 /* MN enter in dormant mode */
73,74d71
< #define IP6_ADDR_PRE_UNIT 0x00000001 /* Increment of IP6 addr */
< #define IP6_ADDR_PRE_TEN 0x00000010 /*Increment of 10IP6 addr*/
81d77
< #define IP6_MH_BU_DORM 0x0001 /* MN enter in dormant mode */
83,84d78
< #define IP6_ADDR_PRE_UNIT 0x01000000 /* Increment of IP6 addr */
< #define IP6_ADDR_PRE_TEN 0x10000000 /*Increment of 10IP6 addr*/
100d93
< #define IP6_MH_BA_DORM 0x01 /* MN enter in dormant mode */

```

## VII.2 Modificacions a *proc\_sys.c*

```

29c29
< #include "proc_sys.h"
---
>
62,124d61
< }
<
< /*Funci—n que decide a partir de las estad'sticas del kernel si el MN debe cambiar al estado
dormant o no*/
< int go_to_dormant (int *last_val)
< {
< FILE *fp;
< char s[20];
< int ret, go = 0, new_val;
<
< fp = fopen (PROC_NET_IP6_DEV_STAT,"r");
< if (!fp)
< {
< printf("Error abriendo archivo %s\n",PROC_NET_IP6_DEV_STAT);
< return (go);
< }
< do
< {
< ret = fscanf(fp,"%s ",s);
< if (strcmp(s,"ip6tnl1:")==0)
< {
< fscanf(fp,"%d",&new_val);
< printf ("new_val = %d\n",new_val);
< printf ("last_val = %d\n",*last_val);
< if (new_val == *last_val)
< {
< go = 1;
< }
< *last_val = new_val;
< fclose (fp);
< return (go);
< }
< }while (ret>0);
< fclose (fp);

```

```

<     return (go);
< }
<
< /*Funci—n que recupera el par metro de configuraci—n Dt*/
< void distance_threshold (int *distance)
< {
<     FILE *fp;
<     char s[30];
<     int ret, threshold;
<
<     fp = fopen ("/usr/local/etc/paging.conf","r");
<     if (!fp)
<     {
<         printf("Error abriendo archivo %s\n",PROC_NET_IP6_DEV_STAT);
<         return;
<     }
<     do
<     {
<         ret = fscanf(fp,"%s ",s);
<         if (strcmp(s,"DistancePagingArea")==0)
<         {
<             fscanf(fp,"%d",&threshold);
<             printf ("distance_threshold = %d\n",threshold);
<             memcpy(distance,&threshold,sizeof(int));
<             fclose (fp);
<             return;
<         }
<     }while (ret>0);
<     fclose (fp);
<     return;

```

### VII.3 Modificacions a *proc\_sys.h*

```

17d16
< #define PROC_NET_IP6_DEV_STAT "/proc/net/dev"
22,25d20
<
< int go_to_dormant (int *last_val);
<
< void distance_threshold (int *distance);

```

### VII.4 Modificacions a *movement.c*

```

51,70d50
< //capçaleres afegides
< #include <inttypes.h>
< #include <time.h>
< #include <unistd.h>
< #include <signal.h>
< #include <sys/ioctl.h>
< #include <stdio.h>
< #include <string.h>
< #include <stdlib.h>
< #include <sys/socket.h>
< #include <netinet/in.h>
< #include <arpa/inet.h>
< #include <netinet/ether.h>
< #include <netinet/if_ether.h>

```

```

< #include <net/ethernet.h>
< #include <netpacket/packet.h>
< #include <sys/time.h>
< #include <netinet/ip6.h>
< //fi capçaleres afegides
<
91,92d70
< void ssid_handler();
<
111,306d88
< struct in6_addr mn_coa;
<
< pthread_mutex_t mutex; //thread timer
< pthread_cond_t cond; //thread timer
< pthread_t md_listener_ssid; //thread ssid
< int dormant_global=0; //si =1, estem en estat dormant
< const struct in6_addr *saddr_dormant; //guardem direccio del router anterior
< int iif_dormant; //guardem interficie del router anterior
< struct nd_router_advert *ra_dormant; // ra global per md_rcv_ra
< int len_dormant; // len global per md_rcv_ra
< struct nlmsg_hdr *n_dormant; //n global per process_addr
< void *arg_dormant; //arg global per process_addr
<
<
<
<
< /*Constantes del canvi de base para compensar el giro de los ejes*/
<
< int canvi_base[2][2]={{3,-1},{1,2}};
< int unitat[2][2]={{1,0},{0,1}};
<
< /*Funciones para el c lculo con matrices*/
<
< void matrix_product2 (int a[2][2],int b[2][2],int res[2][2])
< {
<     res[0][0] = a[0][0] * b[0][0] + a[0][1] * b[1][0];
<     res[0][1] = a[0][0] * b[0][1] + a[0][1] * b[1][1];
<     res[1][0] = a[1][0] * b[0][0] + a[1][1] * b[1][0];
<     res[1][1] = a[1][0] * b[0][1] + a[1][1] * b[1][1];
< }
<
< void matrix_product (int a[2][2],int b[2],int res[2])
< {
<     res[0] = a[0][0] * b[0] + a[0][1] * b[1];
<     res[1] = a[1][0] * b[0] + a[1][1] * b[1];
< }
<
< void matrix_sub (int a[][2], int b[][2], int res[][2], int level)
< {
<     int i;
<
<     for (i=0 ; i<level; i++)
<     {
<         res[i][0] = a[i][0] - b[i][0];
<         res[i][1] = a[i][1] - b[i][1];
<     }
< }
<
<
< void matrix_sum (int a[2], int res[2])
< {

```

```

<     res[0] += a[0];
<     res[1] += a[1];
< }
<
< /*Funci—n que devuelve el identificador de celda a partir de la direcci—n del AR*/
< void id_cell (struct in6_addr *router,int ids[21])
< {
<     int i=0, j;
<
<     for (j=0;j<3;j++)
<     {
<         ids[i++]=((int)((((__const uint8_t*)(router))[15-(j*3)]&0x7);
<         ids[i++]=((int)((((__const uint8_t*)(router))[15-(j*3)]>>3)&0x7);
<         ids[i++]=((int)((((__const      uint8_t*)(router))[15-(j*3)]>>6)&0x3)|((((__const
uint8_t*)(router))[14-(j*3)]&0x1)<<2));
<         ids[i++]=((int)((((__const uint8_t*)(router))[14-(j*3)]>>1)&0x7);
<         ids[i++]=((int)((((__const uint8_t*)(router))[14-(j*3)]>>4)&0x7);
<         if (i==21)
<             break;
<         ids[i++]=((int)((((__const      uint8_t*)(router))[14-(j*3)]>>7)&0x1)|((((__const
uint8_t*)(router))[13-(j*3)]&0x3)<<1));
<         ids[i++]=((int)((((__const uint8_t*)(router))[13-(j*3)]>>2)&0x7);
<         ids[i++]=((int)((((__const uint8_t*)(router))[13-(j*3)]>>5)&0x7);
<     }
< }
<
< /*Funci—n que obtiene las coordenadas de la celda a partir del identificador*/
< int get_coord (int number,int coord[2])
< {
<     switch (number)
<     {
<         case 0:
<             coord[0] = 0;
<             coord[1] = 0;
<             return (0);
<         case 1:
<             coord[0] = 0;
<             coord[1] = 1;
<             return (0);
<         case 2:
<             coord[0] = 1;
<             coord[1] = 0;
<             return (0);
<         case 3:
<             coord[0] = 1;
<             coord[1] = 1;
<             return (0);
<         case 4:
<             coord[0] = 1;
<             coord[1] = 2;
<             return (0);
<         case 5:
<             coord[0] = 2;
<             coord[1] = 1;
<             return (0);
<         case 6:
<             coord[0] = 2;
<             coord[1] = 2;
<             return (0);
<     }

```

```

<         case 7:
<             coord[0] = 7;
<             coord[1] = 7;
<             return (1);
<         }
<     return (-1);
< }
<
< /*Devuelve el valor de la distancia a partir de el vector calculado*/
< int get_distance (int distance[2])
< {
<     int res = 0;
<
<     if (distance[0]<0&&distance[1]>0)
<     {
<         distance[0] = distance[0]*-1;
<         res = distance[0]+distance[1];
<     }
<     else if (distance[0]>0&&distance[1]<0)
<     {
<         distance[1] = distance[1]*-1;
<         res = distance[0]+distance[1];
<     }
<     else
<     {
<         if (abs(distance[0])<abs(distance[1]))
<             res = abs(distance[1]);
<         else
<             res = abs(distance[0]);
<     }
<     return (res);
< }
<
< /*Funci—n que calcula la distancia entre celdas usando las direcciones de los ARs*/
< int calculate_distance (struct in6_addr *old_router, struct in6_addr *new_router)
< {
<     struct in6_addr aux;
<     int i,j, res, level = 21, num_cell[21];
<     int                                     coord_old[21][2],
coord_new[21][2],coord_sub[21][2],coord_mult[21][2],canvi_base2[2][2];
<     int coord_res[2]={0 , 0};
<
<     aux = *old_router;
<     id_cell(&aux,num_cell);
<     for (i=0;i<21;i++)
<     {
<         res = get_coord(num_cell[i],coord_old[i]);
<         if (res==-1)
<         {
<             dbg ("Problema recuperando coordenadas\n");
<             return(-1);
<         }
<     }
<     aux = *new_router;
<     id_cell(&aux,num_cell);
<     for (i=0;i<21;i++)
<     {
<         res = get_coord(num_cell[i],coord_new[i]);
<         if (res==-1)
<         {

```

```

<         dbg ("Problema recuperando coordenadas\n");
<         return (-1);
<     }
< }
< matrix_sub(coord_new,coord_old,coord_sub,level);
< for (i=0;i<21;i++)
< {
<     if (i==0)
<     {
<         matrix_product(unitat,coord_sub[i],coord_mult[i]);
<     }
<     else if (i==1)
<     {
<         matrix_product(canvi_base,coord_sub[i],coord_mult[i]);
<     }
<     else
<     {
<         for (j=0;j<i;j++)
<         {
<             matrix_product2(canvi_base,canvi_base,canvi_base2);
<         }
<         matrix_product(canvi_base2,coord_sub[i],coord_mult[i]);
<     }
<     matrix_sum(coord_mult[i],coord_res);
< }
< dbg ("coord_res = %d,%d\n",coord_res[0],coord_res[1]);
<
<     return (get_distance(coord_res));
< }
472d253
<
493d273
<         {
495d274
<         }
501d279
<
503d280
<
504a282
>
601,603d378
<
<         mn_coa = new->addr;
<
645c420,421
<         ifa->ifa_index);
---
>         ifa->ifa_index);
>
725,727d500
<     n_dormant = n; //guardem variables globals
<     arg_dormant = arg;
<
1396d1168
<
1401,1402d1172
<
<
1405,1410d1174

```

```

<
<         if (dormant_global == 1)
<         {
<             ssid_handler();
<         }
<
1441,1446d1204
<
<             char new_ssid[38];
<             FILE *fp;
<             int i;
<
<             struct in6_addr router;
1448,1481d1205
<             fp = popen("/sbin/iwgetid eth1", "r"); //posar aqui la interficie que es fa servir
<             if (fp == NULL) {
<                 MDBG2("No s'ha pogut llegir iwgetid\n" );
<
<             }
<
<                 while (fgets (&new_ssid[0],2,fp)!=NULL)
<                 {
<                     if (strcmp(&new_ssid[0],":")==0)
<                     {
<                         fgets (&new_ssid[0],2,fp);
<                         break;
<                     }
<                 }
<
<                 strcpy(new_ssid,"3ffe:100:0000:");
<
<                 for (i=14;i<38;i++)
<                 {
<                     if(i==18 || i==23 || i==28 || i==33 )
<                     strcpy(&new_ssid[i],":");
<                     else
<                     fgets (&new_ssid[i],2,fp);
<                 }
<
<
<
<                 inet_pton(AF_INET6, new_ssid, &router); //funcio que passa char a
adreça ipv6
<
<                 rtr_addr_new = router;
<                 dbg("rtr_addr_new:                %x:%x:%x:%x:%x:%x:%x:%x\n",
NIP6ADDR(&rtr_addr_new) );
<                 pclose(fp);
<
<
1491d1214
<
1905,1909d1627
<         iif_dormant = iif; //guardem valor variables globals
<         ra_dormant = ra;
<         len_dormant = len;
<         saddr_dormant = saddr;
<
1913,1959d1630
<

```



```

<      /*Obtengo la direcció IPv6 del AR que controla la celda visitada*/
<      struct prefix_list_entry *prefixes = NULL;
<      prefixes = prefix_list_get(&new->prefixes,&mn_coa,64);
<      if (prefixes!=NULL)
<      {
<          char new_ssid[38];
<          FILE *fp;
<          int i;
<
<          struct in6_addr router;
<
<          fp = popen("/sbin/iwgetid eth1", "r"); //posar aqui la interfície que es fa servir
<          if (fp == NULL) {
<              MDBG2("No s'ha pogut llegir iwgetid\n" );
<
<          }
<
<          while (fgets (&new_ssid[0],2,fp)!=NULL)
<          {
<              if (strcmp(&new_ssid[0],":")==0)
<              {
<                  fgets (&new_ssid[0],2,fp);
<                  break;
<              }
<
<          }
<
<          strcpy(new_ssid,"3ffe:100:0000:");
<
<          for (i=14;i<38;i++)
<          {
<              if(i==18 || i==23 || i==28 || i==33 )
<                  strcpy(&new_ssid[i],":");
<              else
<                  fgets (&new_ssid[i],2,fp);
<          }
<
<
<          inet_pton(AF_INET6, new_ssid, &router); //funcio que passa char a
adreça ipv6
<
<          rtr_addr_new = router;
<          dbg("rtr_addr_new:                %x:%x:%x:%x:%x:%x:%x:%x\n",
NIP6ADDR(&rtr_addr_new) );
<          pclose(fp);
<
<      }
<
1965d1635
<
2039,2175c1709
< void ssid_handler()
< {
< char new_ssid[38];
< FILE *fp;
< int i;
<
< char new_ssid2[38];
< FILE *fp2;
< int j;

```

```

<
< struct in6_addr router;
< struct in6_addr router2;
< struct md_inet6_iface *iface_dormant; //guardem variable iface per quan estiguem en
dormant
< struct md_router *new_dormant; //on guardem el valor de new dormant per funcions de
expire...
<
<     int distance;
<     //int DISTANCE_PAG_AREA = 1; descomentar aquest valor si no llegim be la distancia
de paging
<
< fp = popen("/sbin/iwgetid eth1", "r"); //posar aqui la interficie que es fa servir
< if (fp == NULL) {
<     MDBG2("No s'ha pogut llegir iwgetid\n" );
< }
<
<     while (fgets (&new_ssid[0],2,fp)!=NULL)
<     {
<         if (strcmp(&new_ssid[0],":")==0)
<         {
<             fgets (&new_ssid[0],2,fp);
<             break;
<         }
<     }
<
<     strcpy(new_ssid,"3ffe:100:0000:");
<
<     for (i=14;i<38;i++)
<     {
<         if(i==18 || i==23 || i==28 || i==33 )
<         strcpy(&new_ssid[i],":");
<         else
<         fgets (&new_ssid[i],2,fp);
<     }
<
<     inet_pton(AF_INET6, new_ssid, &router); //funcio que passa char a adreça ipv6
<
< fp2 = popen("/sbin/iwgetid eth1", "r"); //posar aqui la interficie que es fa servir
< if (fp2 == NULL) {
<     printf("No s'ha pogut llegir iwgetid\n" );
< }
<
<     while (fgets (&new_ssid2[0],2,fp2)!=NULL)
<     {
<         if (strcmp(&new_ssid2[0],":")==0)
<         {
<             fgets (&new_ssid2[0],2,fp2);
<             break;
<         }
<     }
<
<     strcpy(new_ssid2,"fe80:000:0000:");
<
<     for (j=14;j<38;j++)
<     {
<         if(j==18 || j==23 || j==28 || j==33 )
<         strcpy(&new_ssid2[j],":");
<         else
<         fgets (&new_ssid2[j],2,fp2);

```

```

<     }
<     printf("SSID llegit %s\n", new_ssid2);
<     inet_pton(AF_INET6, new_ssid2, &router2);
<
<     MDBG("llegint SSID des de %x:%x:%x:%x:%x:%x:%x:%x on iface %d\n",
<         NIP6ADDR(&router), iif_dormant);
<
<     MDBG("llegint SSID des de %x:%x:%x:%x:%x:%x:%x:%x on iface %d\n",
<         NIP6ADDR(&router2), iif_dormant);
<
<     pthread_mutex_lock(&iface_lock);
<     if ((iface_dormant = md_get_inet6_iface(&ifaces, iif_dormant)) != NULL &&
<         (new_dormant = md_create_router(iface_dormant, &router2, ra_dormant,
len_dormant)) != NULL)
<     {
<
<         /*Obtengo la direcció IPv6 del AR que controla la celda visitada*/
<
<         dbg("rtr_addr_new:                %x:%x:%x:%x:%x:%x:%x:%x\n",
NIP6ADDR(&rtr_addr_new));
<         dbg("SSID            router:      %x:%x:%x:%x:%x:%x:%x:%x\n",
NIP6ADDR(&router));
<
<         if (!memcmp(&rtr_addr_new,&router,sizeof(struct in6_addr)))
<         {
<             dbg("Direccions iguals, node segueix en dormant\n");
<
<             if (!md_check_expired_routers(iface_dormant, new_dormant) &&
!md_check_backup_routers(iface_dormant, new_dormant)){
<                 md_check_default_router(iface_dormant, new_dormant);
<                 pthread_mutex_unlock(&iface_lock);
<             }
<
<             else
<             {
<                 dbg("Direccions diferents, comprovant distancia de paging\n");
<
<                 distance = calculate_distance(&rtr_addr_new,&router);
<                 dbg ("Distance = %d\n",distance);
<                 dbg ("DISTANCE_PAG_AREA = %d\n",DISTANCE_PAG_AREA);
<
<                 if (distance<DISTANCE_PAG_AREA)
<                 {
<                     memcpy (&rtr_addr_old,&rtr_addr_new,sizeof(struct in6_addr));
<                     rtr_addr_new = router;
<                     dbg("Salt inferior a distancia paging, rtr_addr_new:
%x:%x:%x:%x:%x:%x:%x:%x\n", NIP6ADDR(&rtr_addr_new));
<
<                     if (!md_check_expired_routers(iface_dormant, new_dormant) &&
!md_check_backup_routers(iface_dormant, new_dormant)){
<                         md_check_default_router(iface_dormant, new_dormant);
<
<                         pthread_mutex_unlock(&iface_lock);
<
<                     }
<
<                     if (distance >= DISTANCE_PAG_AREA)
<                     {
<                         if (!md_check_expired_routers(iface_dormant, new_dormant)
&& !md_check_backup_routers(iface_dormant, new_dormant)){

```

```

<         md_check_default_router(iface_dormant, new_dormant);}
<         pthread_mutex_unlock(&iface_lock);
<         rtr_addr_new = router;
<         md_stop(1); //passem de dormant a actiu
<         md_start(0);
<         MDBG("Salt superior a distancia paging, node activat\n");
<         paging_nou_movement = 1; //evitem tornar a activar el node
quan passi a actiu a mn.c
<         }
<     }
< }
<
<     }
<     pclose(fp);
<     pclose(fp2);
< }
<
<
< int md_start(int dormant)
---
> int md_start(void)
2177,2179d1710
<     dormant_global=dormant;
<     if (dormant == 2) //activacio inicial
<     {
2185,2198d1715
<     dormant_global=0;
<     }
<
<     if (dormant == 0) //node passa a actiu despres de estar dormant
<     {
<     system ("echo 1 > /proc/sys/net/ipv6/conf/eth1/accept_ra"); //activem els RtrAdv
<     icmp6_handler_reg(ND_ROUTER_ADVERT, &md_ra_handler); //activem el "lector" de
RtrAdv
<     }
<
<     if (dormant == 1)
<     {
<     system ("echo 0 > /proc/sys/net/ipv6/conf/eth1/accept_ra"); //anulem els RtrAdv
<     ssid_handler(); //lectura de ssid
<     }
2202c1719
< void md_stop(int dormant)
---
> void md_stop(void)
2204,2215c1721,1725
<
<     if (dormant == 0)
<     {
<     MDBG("Parem serveis de node MN\n");
<     icmp6_handler_dereg(ND_ROUTER_ADVERT, &md_ra_handler); //desactivem el
"lector" de RtrAdv
<     MDBG("Serveis de node MN aturats\n");
<
<     }
<     if (dormant == 1)
<     {
<     dormant_global = 0; //no caldria, pero aixi ens asegurem
<     }

```

```

---
>   rtnl_close(&md_rth);
>   icmp6_handler_dereg(ND_ROUTER_ADVERT, &md_ra_handler);
>   icmp6_handler_dereg(ND_NEIGHBOR_ADVERT, &md_na_handler);
>   pthread_cancel(md_listener);
>   pthread_join(md_listener, NULL);

```

## VII.5 Modificacions a *movement.h*

```

8d7
< #include <inttypes.h>
20,34d18
< /*#define DISTANCE_PAG_AREA 2*/
<
< struct in6_addr rtr_addr_new;
< int paging_nou_movement;
<
<
< void matrix_product2 (int a[2][2],int b[2][2],int res[2][2]);
< void matrix_product (int a[2][2],int b[2],int res[2]);
< void matrix_sub (int a[][2], int b[][2], int res[][2], int level);
< void matrix_sum (int a[2], int res[2]);
< void id_cell (struct in6_addr *router,int ids[21]);
< int get_coord (int number,int coord[2]);
< int calculate_distance (struct in6_addr *old_router, struct in6_addr *new_router);
<
<
200,201c184,185
< int md_start(int dormant);
< void md_stop(int dormant);
---
> int md_start(void);
> void md_stop(void);

```

## VII.6 Modificacions a *mn.c*

```

37c37
< #include <unistd.h>
---
> #include <unistd.h>
39,58d38
< //Lliberies afegides
< #include <pcap.h>
< #include <signal.h>
< #include <sys/ioctl.h>
< #include <linux/types.h>
< #include <stdio.h>
< #include <string.h>
< #include <stdlib.h>
< #include <sys/socket.h>
< #include <netinet/in.h>
< #include <arpa/inet.h>
< #include <netinet/ether.h>
< #include <netinet/if_ether.h>

```

```

< #include <net/ethernet.h>
< #include <netpacket/packet.h>
< #include "proc_sys.h"
< //#include <netinet/pcap.h>
<
< //Fi lliberies afegides
<
108,117d87
< /*Defino las variables importantes*/
< struct bulentry *last_bule;
< static int i_dormant = 0;
< static pthread_t tid = '0';
< pthread_mutex_t dormant_mutex;
< static int DISTANCE_PAG_AREA;
< char *first_msg;
< int iface, msg_len;
< char old_ssid[21];
<
129c99
< static void mn_send_home_bu(struct home_addr_info *hai,int pagging_move);
---
> static void mn_send_home_bu(struct home_addr_info *hai);
132,170d101
< /*Funci—n que reenv'a el primer paquete saliente del nodo cuando se reestablece el toenel
entre el HA y el MN*/
<
< int send_first_msg()
< {
<     int packet_socket, ret;
<     struct sockaddr_ll addr;
<
<     if (msg_len!=0)
<     {
<         dbg("iface %d\n",iface);
<         packet_socket = socket(PF_PACKET,SOCK_DGRAM,ETH_P_ALL);
<         if (packet_socket == -1)
<         {
<             dbg("Error abriendo el socket\n");
<             return (-1);
<         }
<         bzero(&addr,sizeof(struct sockaddr_ll));
<         addr.sll_family = AF_PACKET;
<         addr.sll_protocol = htons(ETH_P_IPV6);
<         addr.sll_ifindex = iface;
<         addr.sll_hatype = ARPHRD_ETHER;
<         addr.sll_pkttype = PACKET_BROADCAST;
<         memset(&addr.sll_addr,0xff,6);
<
<         ret = sendto (packet_socket, first_msg, msg_len, 0, (struct sockaddr*)&addr,
sizeof (addr));
<         if (ret < 0)
<             dbg ("Error enviando paquete\n");
<         else
<             dbg("Datos enviados = %d\n",ret);
<         close(packet_socket);
<         msg_len = 0;
<         free(first_msg);
<     }
<     else
<         dbg("No hay datos para enviar\n");

```

```

<     return(0);
< }
<
<
374,375c305
<
< static int mn_send_bu_msg(struct bulentry *bule, int wakeup)
---
> static int mn_send_bu_msg(struct bulentry *bule)
378d307
<     static int last_val = -1;
393,417d321
<     MDBG("valor del wakeup %d\n", wakeup);
<     /*Decido si cambiar el estado dormant o al activo*/
<     if (go_to_dormant(&last_val) && wakeup==0)
<     {
<
<         bu->ip6mhbu_flags = bu->ip6mhbu_flags | IP6_MH_BU_DORM;
<         bule->lifetime.tv_sec = 0xffff << 2;
<         pthread_mutex_lock (&dormant_mutex);
<         i_dormant = 1;
<         MDBG("Node Dormant\n");
<         pthread_mutex_unlock(&dormant_mutex);
<
<     }else if (wakeup == 2)
<     {
<         bu->ip6mhbu_flags = bu->ip6mhbu_flags | IP6_MH_BU_DORM;
<         bule->lifetime.tv_sec = 0xffff << 2;
<
<     }else
<     {
<         pthread_mutex_lock (&dormant_mutex);
<         i_dormant = 0;
<         MDBG("Node Actiu\n");
<         pthread_mutex_unlock(&dormant_mutex);
<     }
<
449,476d352
<     pthread_mutex_lock(&dormant_mutex);
<     MDBG("Valor de i_dormant %d\n", i_dormant);
<     MDBG("Valor de tid %d\n", tid);
<     /*Si he cambiado al estado dormant, creo el thread que controlarà el tràfic saliente*/
<     if (i_dormant&&tid=='0')
<     {
<         MDBG("Passem a estat dormant\n");
<         pthread_create(&tid,NULL,wake_up,&bule->hoa);
<         pthread_detach(tid);
<         md_stop(0); //passem de actiu a dormant
<         md_start(1);
<         paging_nou_movement = 0;
<         MDBG("Node en dormant\n");
<     }
<     /*Si he cambiado al estado activo, cancelo ese thread*/
<     if (i_dormant==0 && tid!='0')
<     {
<         MDBG("Passem a estat actiu\n");
<         tid = '0';
<         if( paging_nou_movement == 0 ){ //per si venim de un canvi de paging, no
activem de nou la interfície, nomes si ve de dormant
<         md_stop(1); //passem de dormant a actiu

```

```

<         md_start(0);
<         }
<         MDBG("Node activat\n");
<
<     }
<     pthread_mutex_unlock(&dormant_mutex);
<
480,545d355
< /*Thread que controla el tráfico saliente del nodo*/
< void *wake_up (void *varg)
< {
<     const u_char *msg;
<     char errbuf[PCAP_ERRBUF_SIZE];
<     pcap_t* descr;
<     struct pcap_pkthdr hdr;
<     struct bpf_program fp;
<     bpf_u_int32 maskp;
<     bpf_u_int32 netp;
<     struct in6_addr *aux = varg;
<     char s[50], *dev;
<
<     dev = pcap_lookupdev(errbuf);
<     if (dev == NULL)
<     {
<         dbg("%s\n",errbuf);
<         pthread_exit("-1");
<     }
<     pcap_lookupnet(dev,&netp,&maskp,errbuf);
<     if ((descr = pcap_open_live(NULL,65535,1,-1,errbuf)) == NULL)
<     {
<         dbg("%s\n",errbuf);
<         pthread_exit("-1");
<     }
<     sprintf (s,"ip6 src %x:%x:%x:%x:%x:%x:%x:%x\n",NIP6ADDR(aux));
<     dbg ("%s\n",s);
<     if (pcap_compile(descr,&fp,s,0,netp) == -1)
<     {
<         dbg("Error compilando el filtro\n");
<         pthread_exit("-1");
<     }
<     if (pcap_setfilter(descr,&fp) == -1)
<     {
<         dbg("Error aplicando el filtro\n");
<         pthread_exit("-1");
<     }
<     if ((msg = pcap_next(descr,&hdr)) == NULL)
<     {
<         dbg("Error al capturar el paquete\n");
<         pthread_exit("-1");
<     }
<     first_msg = (char *)malloc(hdr.len);
<     if (first_msg == NULL)
<         dbg("Error en el malloc\n");
<     else
<     {
<         msg_len = hdr.len - sizeof(struct ether_header);
<         memcpy (first_msg,msg+sizeof(struct ether_header)+2,hdr.len);
<     }
<     if (i_dormant)
<     {

```



```

<         pthread_mutex_lock(&dormant_mutex);
<         i_dormant = 0;
<         pthread_mutex_unlock(&dormant_mutex);
<         last_bule->seq++;
<         last_bule->lifetime.tv_sec = 0xf << 2;
<         mn_send_bu_msg(last_bule,1);
<     }
<     MDBG("Estoy saliendo del thread\n");
<     pthread_mutex_lock(&dormant_mutex);
<     tid = '0';
<     pthread_mutex_unlock(&dormant_mutex);
<     pthread_exit(0);
< }
<
628c438
<         mn_send_bu_msg(bule,1);
---
>         mn_send_bu_msg(bule);
658c468
<         mn_send_home_bu(hai,1);
---
>         mn_send_home_bu(hai);
709c519
<         mn_send_bu_msg(bule,1);
---
>         mn_send_bu_msg(bule);
738c548
<         mn_send_bu_msg(bule,0);
---
>         mn_send_bu_msg(bule);
743d552
<             dbg ("Entro al else de bu_refresh\n");
909c718
< static void mn_send_home_bu(struct home_addr_info *hai,int pagging_move)
---
> static void mn_send_home_bu(struct home_addr_info *hai)
1001,1002d809
<
< /*Dependiendo de si estoy activo o dormido, y en el œltimo cado de si me he movido de la
frea de paging o no, env'o el BU o no*/
1004,1013c811,815
<         if (pagging_move)
<         {
<             bule->do_send_bu = 1;
<             mn_send_bu_msg(bule,pagging_move);
<             bul_update_timer(bule);
<         }
<         else{
<             bule->do_send_bu = 0;
<         }
<
---
>         bule->do_send_bu = 1;
>         mn_send_bu_msg(bule);
>         bul_update_timer(bule);
>         if (conf.OptimisticHandoff)
>             post_ba_bul_update(bule);
1015d816
<
1030,1042c831,833

```

```

<
<     /*Si estoy en modo dormant, modifico el tÃnel*/
<     pthread_mutex_lock(&dormant_mutex);
<     if (i_dormant)
<     {
<         tunnel_mod(hai->if_tunnel, &hai->primary_coa.addr, &hai->primary_coa.addr,
hai->primary_coa.iif, mn_ext_tunnel_ops, hai);
<         post_ba_bul_update(bule);
<     }else
<     {
<         tunnel_mod(hai->if_tunnel, &hai->primary_coa.addr, &hai->ha_addr, hai-
>primary_coa.iif, mn_ext_tunnel_ops, hai);
<     }
<     iface = hai->if_tunnel;
<     pthread_mutex_unlock(&dormant_mutex);
---
>     tunnel_mod(hai->if_tunnel, &hai->primary_coa.addr, &hai->ha_addr,
>         hai->primary_coa.iif, mn_ext_tunnel_ops, hai);
>
1050d840
<     last_bule = bule;
1066c856
<     mn_send_bu_msg(bule,0);
---
>     mn_send_bu_msg(bule);
1172d961
<
1213c1002
<     // First check authenticator
---
>     /* First check authenticator */
1219c1008
<     // Then sequence number
---
>     /* Then sequence number */
1223,1224c1012,1014
<         // In this case, ignore BA and resends BU.
<
---
>         /*
>         * In this case, ignore BA and resends BU.
>         */
1247c1037
<         mn_send_bu_msg(bule,0);
---
>         mn_send_bu_msg(bule);
1276c1066
<         // Don't resend BUs to this CN
---
>         /* Don't resend BUs to this CN */
1301,1302c1091,1093
<         // If BA was for home registration & succesful Send RO BUs to CNs for
this home address.
<
---
>         /* If BA was for home registration & succesful
>         * Send RO BUs to CNs for this home address.
>         */
1310c1101
<         // If status of BA is 0 or 1, Binding Update is accepted.

```

```

---
>          /* If status of BA is 0 or 1, Binding Update is accepted. */
1315,1316c1106,1108
<          // If BA was for home registration & succesful Send RO BUs to CNs for this
home address.
<
---
>          /* If BA was for home registration & succesful
>          * Send RO BUs to CNs for this home address.
>          */
1320c1112
<          // IP6_MH_BA_KEYM
---
>          /* IP6_MH_BA_KEYM */
1323c1115
<                                     // Inform IKE to send readdress msg
---
>                                     /* Inform IKE to send readdress msg */
1325c1117
<                                     // Inform IKE to renegotiate SAs
---
>                                     /* Inform IKE to renegotiate SAs */
1346d1137
<          dbg("Crida a les següents funcions de mn_rcv_ba\n");
1350,1367d1140
<          //Si estaba en estado dormant, reestablezco el t nel con el HA y reenvio mi primer
paquete
<          struct home_addr_info *hai;
<          hai = bule->home;
<
<          pthread_mutex_lock(&dormant_mutex);
<          if (i_dormant)
<          {
<              tunnel_mod(hai->if_tunnel, &hai->primary_coa.addr, &hai->primary_coa.addr,
<              hai->primary_coa.iif, mn_ext_tunnel_ops, hai);
<          }
<          else
<          {
<              tunnel_mod(hai->if_tunnel, &hai->primary_coa.addr, &hai->ha_addr,
<              hai->primary_coa.iif, mn_ext_tunnel_ops, hai);
<          }
<          if (send_first_msg()!=0)
<              dbg("Error enviando primer paquete\n");
<          pthread_mutex_unlock(&dormant_mutex);
1377,1400d1149
<          /*Dependiendo de si estoy activo o dormant, y si cambio de area de paging o no,
avisar  al HA con un BU o no*/
<          int paging_move=1;
<          int distance;
<
<          pthread_mutex_lock(&dormant_mutex);
<          if (i_dormant)
<          {
<              dbg("rtr_addr_old                                     :
<          %x:%x:%x:%x:%x:%x:%x:%x:%x\n",NIP6ADDR(&rtr_addr_old));
<              dbg("rtr_addr_new                                     :
<          %x:%x:%x:%x:%x:%x:%x:%x:%x\n",NIP6ADDR(&rtr_addr_new));
<              distance = calculate_distance(&rtr_addr_old,&rtr_addr_new);
<              if (distance<DISTANCE_PAG_AREA)
<                  paging_move = 0;

```

```

<         else
<         {
<             pagging_move = 2;
<             memcpy (&rtr_addr_old,&rtr_addr_new,sizeof(struct in6_addr));
<             dbg("New rtr_addr_old :
%x:%x:%x:%x:%x:%x:%x:%x:%x\n",NIP6ADDR(&rtr_addr_old));
<         }
<
<         dbg ("Distance = %d\n",distance);
<     }
<
<     pthread_mutex_unlock(&dormant_mutex);
<
1403c1152
<         mn_send_home_bu(hai,pagging_move);
---
>         mn_send_home_bu(hai);
1407c1156
<             mn_send_home_bu(hai,pagging_move);
---
>             mn_send_home_bu(hai);
2355d2103
<     {
2357,2365d2104
<         /*Si no estoy dormido, almaceno la direcci3n del AR para identificar la celda en
la que me encuentro*/
<         pthread_mutex_lock(&dormant_mutex); //afegida per el bloqueig
<         if (!li_dormant)
<         {
<             rtr_addr_old = rtr_addr_new;
<             dbg ("rtr_addr_old
%x:%x:%x:%x:%x:%x:%x:%x:%x\n",NIP6ADDR(&rtr_addr_old));
<         }
<         pthread_mutex_unlock(&dormant_mutex); //afegida per el bloqueig
<     }
2561,2575d2299
<
<     /*Si el BRR lo env'a el HA, se supone que es un mensaje de paging. Se cambia al
estado activo y se env'a BU para informar de la nuevo CoA*/
<     if (IN6_ARE_ADDR_EQUAL(cn,&last_bule->peer_addr))
<     {
<         struct home_addr_info *hai;
<         hai = last_bule->home;
<         last_bule->seq++;
<         last_bule->lifetime.tv_sec = 0xf << 2;
<         mn_send_bu_msg(last_bule,1);
<
<         tunnel_mod(hai->if_tunnel, &hai->primary_coa.addr, &hai->ha_addr,hai-
>primary_coa.iif, mn_ext_tunnel_ops, hai);
<         return;
<     }
<
<
2662,2666d2385
<     /*Inicializo las variables importantes*/
<     int distance = 0;
<
<     msg_len = 0;
<
2675,2682d2393

```

```

<   if (pthread_mutex_init(&dormant_mutex,NULL))
<       return -1;
<   distance_threshold (&distance);
<   if (distance != 0)
<       DISTANCE_PAG_AREA = distance;
<   else
<       DISTANCE_PAG_AREA = 1;
<
2706c2417
<   if (md_start(2) < 0)
---
>   if (md_start() < 0)
2741c2452
<   md_stop(i_dormant);
---
>   md_stop();

```

## VII.7 Modificacions a *mn.h*

```

116c116
< struct in6_addr rtr_addr_old;
---
>
119d118
< void *wake_up (void *varg);

```

## VII.8 Modificacions a *ha.c*

```

41,64d40
< //Capçaleres afegides
<
< #include <netinet/ip6.h>
< #include <stdio.h>
< #include <unistd.h>
< #include <string.h>
< #include <stdlib.h>
< #include <sys/socket.h>
< #include <netinet/in.h>
< #include <arpa/inet.h>
< #include <netinet/ether.h>
< #include <netinet/if_ether.h>
< #include <net/ethernet.h>
< #include <netpacket/packet.h>
< #include <inttypes.h>
< #include "movement.h"
< #include "proc_sys.h"
< #include <pcap.h>
< #define NULL 0
<
< /*#define MAX_MNS_DORMANT 100*/
<
< //Fi capçaleres afegides

```

```

<
87,88c63
<
< static pthread_mutex_t bu_worker_mutex, list_mutex;
---
> static pthread_mutex_t bu_worker_mutex;
91,92d65
< static struct in6_addr my_ip;
< static int DISTANCE_PAG_AREA;
97,315d69
< struct ha_rcv_bu_args {
<     struct list_head list;
<     struct in6_addr src;
<     struct in6_addr dst;
<     struct in6_addr remote_coa;
<     struct in6_addr bind_coa;
<     struct ip6_mh_binding_update *bu;
<     ssize_t len;
<     struct mh_options mh_opts;
<     struct timespec lft;
<     int iif;
<     int flags;      /* HA_BU_F_XXX */
<     int *statusp;  /* 0 or more than 0 is BA status, otherwise error */
< };
<
< /*Defino las estructuras que definen el listado de nodos dormant*/
< struct mn_dormant {
<     struct ha_rcv_bu_args bu_mn_dormant;
<     int rcv_bu;
<     pthread_t tid;
<     struct mn_dormant *next;
<     u_char *msg;
<     int len_msg;
< };
<
< struct list_mns_dormant {
<     struct mn_dormant *first;
< };
< /*Struct auxiliar para pasar al thread ha_rcv_bu_worker con toda la informaci—n necesaria
para retransmitir el paquete*/
< struct ha_rcv_bu_args_dormant {
<     struct ha_rcv_bu_args *args;
<     struct mn_dormant *mn;
< };
<
< /*Listado de nodos dormant*/
< static struct list_mns_dormant mns_dormant;
<
< /*Funci—n que envia un BRR a toda una area de paging*/
< void send_to_areapaging (struct in6_addr_bundle addrs,struct in6_addr *bind_coa)
< {
<     int i, distance;
<     struct iovec mh_vec;
<     struct in6_addr aux_min, aux_max, aux, reference;
<
<     int DISTANCE_PAG_AREA1 = 1;
<
<     addrs.dst = bind_coa;
<
<     if (!mh_create(&mh_vec, IP6_MH_TYPE_BRR))

```

```

<         pthread_exit("-1");
<
<     mh_send(&addrs, &mh_vec, 1, NULL, 0);
<     free_iov_data(&mh_vec, 1);
<     dbg ("ip6 bind_coa %x:%x:%x:%x:%x:%x:%x:%x:%x\n",NIP6ADDR(bind_coa));
<
<     aux.s6_addr32[0] = 0x0;
<     aux.s6_addr32[1] = 0x0;
<     reference.s6_addr32[0] = 0x0;
<     reference.s6_addr32[1] = 0x0;
<     reference.s6_addr32[2] = bind_coa->s6_addr32[0];
<     reference.s6_addr32[3] = bind_coa->s6_addr32[1];
<     dbg("Reference: %x:%x:%x:%x:%x:%x:%x:%x\n",NIP6ADDR(&reference));
<     aux_min = *bind_coa;
<     aux_max = *bind_coa;
<     for (i=0;i<15;i++)
<     {
<         aux_min.s6_addr32[1]-=(__uint32_t)IP6_ADDR_PRE_UNIT;
<
<         aux.s6_addr32[2] = aux_min.s6_addr32[0];
<         aux.s6_addr32[3] = aux_min.s6_addr32[1];
<         distance = calculate_distance(&reference, &aux);
<         dbg("Distance : %d\n",distance);
<         if (distance <= DISTANCE_PAG_AREA1)
<         {
<             dbg      ("Send      BRR      to      possible      new      CoA
%x:%x:%x:%x:%x:%x:%x:%x\n",NIP6ADDR(&aux_min));
<             addrs.dst = &aux_min;
<             if (!mh_create(&mh_vec, IP6_MH_TYPE_BRR))
<                 pthread_exit("-1");
<
<             mh_send(&addrs, &mh_vec, 1, NULL, 0);
<
<             free_iov_data(&mh_vec, 1);
<         }
<
<         aux_max.s6_addr32[1]+=(__uint32_t)IP6_ADDR_PRE_UNIT;
<         aux.s6_addr32[2] = aux_max.s6_addr32[0];
<         aux.s6_addr32[3] = aux_max.s6_addr32[1];
<         distance = calculate_distance(&reference, &aux);
<         dbg("Distance : %d\n",distance);
<         if (distance <= DISTANCE_PAG_AREA1)
<         {
<             dbg      ("Send      BRR      to      possible      new      CoA
%x:%x:%x:%x:%x:%x:%x:%x\n",NIP6ADDR(&aux_min));
<             addrs.dst = &aux_max;
<             if (!mh_create(&mh_vec, IP6_MH_TYPE_BRR))
<                 pthread_exit("-1");
<
<             mh_send(&addrs, &mh_vec, 1, NULL, 0);
<
<             free_iov_data(&mh_vec, 1);
<         }
<     }
< }
<
< /*Funci—n que inserta la informaci—n de un MN que ha cambiado al estado dormant en el
listado de nodos*/
< int insert_mn (struct ha_recv_bu_args *varg, pthread_t tid)
< {

```

```

< struct ha_recv_bu_args arg;
< struct mn_dormant *mn;
< struct in6_addr aux,aux2;
< memcpy (&arg,varg,sizeof(struct ha_recv_bu_args));
< pthread_mutex_lock(&list_mutex);
< mn = mns_dormant.first;
< if (mn == NULL)
< {
<     mns_dormant.first = malloc (sizeof(struct mn_dormant));
<     if (mns_dormant.first == NULL)
<     {
<         dbg("Error en el malloc!\n");
<         pthread_mutex_unlock(&list_mutex);
<         return (0);
<     }
<     mns_dormant.first->bu_mn_dormant = arg;
<     mns_dormant.first->tid = tid;
<     aux = mns_dormant.first->bu_mn_dormant.dst;
<     dbg ("ip6 mn %x:%x:%x:%x:%x:%x:%x:%x\n",NIP6ADDR(&aux));
<     mns_dormant.first->recv_bu = 0;
<     mns_dormant.first->next = NULL;
<     mns_dormant.first->msg = NULL;
<     mns_dormant.first->len_msg = 0;
< }else{
<     aux2 = arg.dst;
<     aux = mn->bu_mn_dormant.dst;
< if (IN6_ARE_ADDR_EQUAL(&aux,&aux2))
< {
<     dbg("MN ya est# en el listado...actualizando la informaci#n\n");
<     mn->bu_mn_dormant = arg;
<     mn->recv_bu = 0;
<     mn->tid = tid;
<     mn->msg=NULL;
<     mn->len_msg=0;
<     pthread_mutex_unlock(&list_mutex);
<     return(1);
<     }
<     while (mn->next!=NULL)
<     {
<         aux = mn->next->bu_mn_dormant.dst;
<         if (IN6_ARE_ADDR_EQUAL(&aux,&aux2))
<         {
<             dbg("MN ya est# en el listado...actualizando la informaci#n\n");
<             mn->next->bu_mn_dormant = arg;
<             mn->next->recv_bu = 0;
<             mn->next->tid = tid;
<             mn->next->msg=NULL;
<             mn->next->len_msg=0;
<             pthread_mutex_unlock(&list_mutex);
<             return(1);
<         }
<         mn = mn->next;
<     }
<     mn->next = malloc (sizeof(struct mn_dormant));
<     if (mn->next == NULL)
<     {
<         dbg("Error en el malloc!\n");
<         pthread_mutex_unlock(&list_mutex);
<         return (0);
<     }

```



```

<         mn->next->bu_mn_dormant = arg;
<         mn->next->recv_bu = 0;
<         mn->next->tid = tid;
<         mn->next->next = NULL;
<         mn->next->msg=NULL;
<         mn->next->len_msg=0;
<     }
<     pthread_mutex_unlock(&list_mutex);
<     dbg ("MN dormant insertado\n");
<     return (1);
< }
<
< /*Funci—n que borra un MN del listado de nodos dormant cuando cambia al estado activo*/
< int delete_mn (struct in6_addr varg)
< {
<     struct mn_dormant *mn, *aux;
<     struct in6_addr aux2, arg = varg;
<
<     mn = malloc(sizeof(struct mn_dormant));
<     pthread_mutex_lock(&list_mutex);
<     mn->next = mns_dormant.first;
<     while (mn->next!=NULL)
<     {
<         aux2 = mn->next->bu_mn_dormant.dst;
<         dbg("Estoy dentro del if\n");
<         if (IN6_ARE_ADDR_EQUAL(&aux2,&arg))
<         {
<             dbg("He encontrado el mn!!\n");
<             aux = mn->next;
<             if (aux == mns_dormant.first)
<             {
<                 mns_dormant.first = NULL;
<             }else if (aux->next == NULL)
<             {
<                 mn->next = NULL;
<             }else
<             {
<                 mn->next = aux->next;
<             }
<             free (aux);
<             pthread_mutex_unlock(&list_mutex);
<             dbg("MN borrado!\n");
<             return (1);
<         }
<     else
<         mn = mn->next;
<     }
<     pthread_mutex_unlock(&list_mutex);
<     return(0);
< }
<
<
<
< 477,479d230
<         /*Guardo mi direcci—n IP para despuŽs restablecer el tœnel con los nodos
dormant*/
<         memcpy(&my_ip,ha_addr,sizeof(struct in6_addr));
<         dbg("My ip: %x:%x:%x:%x:%x:%x:%x:%x\n", NIP6ADDR(&my_ip));
886,1044c637,650
< /*Funci—n que retransmite el mensaje capturado dirigido al MN dormant*/

```

```

< int send_msg(struct mn_dormant *mn,struct in6_addr bind_coa,int iface)
< {
<     int packet_socket, ret;
<     struct sockaddr_ll addr;
<
<     if (mn->len_msg!=0)
<     {
<
<         dbg("iface %d\n",iface);
<         packet_socket = socket(PF_PACKET,SOCK_DGRAM,ETH_P_ALL);
<         if (packet_socket == -1)
<         {
<             dbg("Error abriendo el socket\n");
<             return (-1);
<         }
<         bzero(&addr,sizeof(struct sockaddr_ll));
<         addr.sll_family = AF_PACKET;
<         addr.sll_protocol = htons(ETH_P_IPV6);
<         addr.sll_ifindex = iface;
<         addr.sll_hatype = ARPHRD_ETHER;
<         addr.sll_pkttype = PACKET_BROADCAST;
<         memset(&addr.sll_addr,0xff,6);
<
<         ret = sendto (packet_socket, mn->msg, (mn->len_msg), 0, (struct
sockaddr*)&addr, sizeof (addr));
<         if (ret < 0)
<             dbg ("Error enviando paquete\n");
<         else
<             dbg("Datos enviados = %d\n",ret);
<         close(packet_socket);
<     }
<     else
<         dbg("No hay datos para enviar\n");
<     return(0);
< }
<
< void copy_buffer(struct mn_dormant *user,struct pcap_pkthdr *phdr, u_char *pdata)
< {
<     int i;
<
<     struct ether_header *eptr;
<
<     /* Apuntamos el puntero a la cabecera Ethernet al comienzo del paquete*/
<     eptr = (struct ether_header *) (pdata + 2);
<
<     //Comprobamos que el paquete es de tipo IPv6*/
<     if (ntohs (eptr->ether_type) == ETH_P_IPV6)
<     {
<         dbg("Detectado paquete IPv6 dirigido a mn\n");
<         /*Ahora extraemos la cabecera IP, por lo que tenemos que desplazar el tamaño de la
cabecera Ethernet ya procesada*/
<
<
<         pdata+=sizeof (struct ether_header)+2;
<
<         dbg("Inserto el mensaje a enviar al mn\n");
<         user->len_msg = phdr->len - sizeof(struct ether_header) - 2;
<         user->msg = (u_char*)malloc(sizeof(u_char)*user->len_msg);
<         dbg("Mensaje insertado\n");
<         for (i=0;i<user->len_msg;i++)

```

```

<         {
<             user->msg[i] = pdata[i];
<         }
<     }
<     else{
<         dbg ("No es un paquete ipv6!\n");
<     }
<     return;
< }
<
<
< /*Thread que controla el tràfic dirigit a un MN dormant*/
< void *first_msg_to_mn_dormant(void *varg)
< {
<     struct    ha_recv_bu_args    *arg=(struct    ha_recv_bu_args*)malloc(sizeof(struct
ha_recv_bu_args));
<
<     struct in6_addr_bundle addr;
<     u_char *msg;
<     char errbuf[PCAP_ERRBUF_SIZE];
<     pcap_t* descr;
<     struct pcap_pkthdr hdr;
<     struct bpf_program fp;
<     bpf_u_int32 maskp;
<     bpf_u_int32 netp;
<     struct in6_addr aux, bind_coa;
<     struct mn_dormant *mn;
<     char s[50], *dev;
<     int bu_received=0;
<
<     memcpy(arg,varg,sizeof(struct ha_recv_bu_args));
<     pthread_setcancelstate(PTHREAD_CANCEL_ENABLE,NULL);
<     bind_coa = arg->bind_coa;
<     addr.src = (struct in6_addr*)malloc(sizeof(struct in6_addr));
<     addr.dst = (struct in6_addr*)malloc(sizeof(struct in6_addr));
<     memcpy(addr.src,&arg->src,sizeof(struct in6_addr));
<     memcpy(addr.dst,&arg->bind_coa,sizeof(struct in6_addr));
<     addr.remote_coa = NULL;
<     addr.local_coa = NULL;
<
<     dev = pcap_lookupdev(errbuf);
<     if (dev == NULL)
<     {
<         dbg("%s\n",errbuf);
<         pthread_exit("-1");
<     }
<     pcap_lookupnet(dev,&netp,&maskp,errbuf);
<     if ((descr = pcap_open_live(NULL,65535,1,-1,errbuf)) == NULL)
<     {
<         dbg("%s\n",errbuf);
<         pthread_exit("-1");
<     }
<     sprintf (s, "ip6 dst || src %x:%x:%x:%x:%x:%x:%x:%x\n",NIP6ADDR(&arg->dst));
<     if (pcap_compile(descr,&fp,s,0,netp) == -1)
<     {
<         dbg("Error compilando el filtro\n");
<         pthread_exit("-1");
<     }
<     if (pcap_setfilter(descr,&fp) == -1)
<     {

```

```

<     dbg("Error aplicando el filtro\n");
<     pthread_exit("-1");
< }
< if ((msg = pcap_next(descr,&hdr)) == NULL)
< {
<     dbg("Error al capturar el paquete\n");
<     pthread_exit("-1");
< }
<
<     pthread_mutex_lock (&list_mutex);
<     mn = mns_dormant.first;
<     while (mn!=NULL)
<     {
<         dbg("Estoy en el for buscando el mn!\n");
<         aux = mn->bu_mn_dormant.dst;
<         dbg ("aux %x:%x:%x:%x:%x:%x:%x:%x\n",NIP6ADDR(&aux));
<
<         if (IN6_ARE_ADDR_EQUAL(&aux,&arg->dst))
<         {
<             dbg("He encontrado el mn!\n");
<             break;
<         }else
<             mn = mn->next;
<     }
<     if (mn!=NULL)
<     {
<         bu_received = mn->recv_bu;
<         copy_buffer(mn,&hdr,msg);
<         dbg("len_msg %d\n",mn->len_msg);
<     }
<     pthread_mutex_unlock (&list_mutex);
<
<     if (!bu_received && mn!=NULL)
<     {
<         send_to_areapaging(addr,&bind_coa);
<     }
<     pthread_mutex_lock(&list_mutex);
<     mn->tid=NULL;
<     pthread_mutex_unlock(&list_mutex);
<     pthread_exit(0);
< }
< }
---
> struct ha_recv_bu_args {
>     struct list_head list;
>     struct in6_addr src;
>     struct in6_addr dst;
>     struct in6_addr remote_coa;
>     struct in6_addr bind_coa;
>     struct ip6_mh_binding_update *bu;
>     ssize_t len;
>     struct mh_options mh_opts;
>     struct timespec lft;
>     int iif;
>     int flags;          /* HA_BU_F_XXX */
>     int *statusp;      /* 0 or more than 0 is BA status, otherwise error */
> };
1048,1051c654
<
<     struct ha_recv_bu_args_dormant *args = varg;
<     struct mn_dormant *mn = args->mn;

```

```

<     struct ha_recv_bu_args *arg = args->args;
---
>     struct ha_recv_bu_args *arg = varg;
1218,1234d820
<
<         }
<     /*Modifico el tunel si el MN ha cambiado al modo dormant*/
<     else if (bu_flags & IP6_MH_BU_DORM)
<     {
<         bce->old_coa = bce->coa;
<         bce->coa = arg->bind_coa;
<         if (tunnel_mod(bce->tunnel, &arg->src, &arg->src, 0,
<             home_tnl_ops, &p) < 0) {
<             if (p.ba_status >= IP6_MH_BAS_UNSPECIFIED)
<                 status = p.ba_status;
<             else
<                 status = IP6_MH_BAS_INSUFFICIENT;
<             goto send_nack;
<         }
<         bcache_update_expire(bce);
<
1256,1260d841
<
<         /*Si recibo un BU con flag dormant, envio un BAcK con flag dormant*/
<         if (bu_flags & IP6_MH_BU_DORM)
<             ba_flags |= IP6_MH_BU_DORM;
<
1281,1293d861
<
<         /*Si el nodo ha cambiado a estado activo, le reenvio el primer paquete
dirigido hacia ŽI*/
<         if (mn!=NULL)
<         {
<             struct in6_addr aux;
<
<             aux = mn->bu_mn_dormant.dst;
<             pthread_mutex_lock(&list_mutex);
<             send_msg(mn,arg->remote_coa,bce->tunnel);
<             pthread_mutex_unlock(&list_mutex);
<             delete_mn(aux);
<         }
<
1322d889
<     struct ha_recv_bu_args_dormant *varg; //linia afegida
1331,1333d897
<     varg = (struct ha_recv_bu_args_dormant *)malloc(sizeof (struct
ha_recv_bu_args_dormant)); //linia afegida
<     varg->mn=NULL; //linia afegida
<
1336,1337d899
<
<
1369a932
>     memcpy(arg->bu, bu, len);
1372,1434d934
<     memcpy(arg->bu, bu, len);
<
<     /*Si el MN ha enviado un BU con flag dormant, lo agrego al listado de nodos dormant y
modifico el Lifetime en la Binding Cache*/
<     pthread_t tid;

```

```

<
<     if (bu->ip6mhbu_flags & IP6_MH_BU_DORM)
<     {
<         struct timespec lifetime;
<
<         pthread_create(&tid, NULL, first_msg_to_mn_dormant,arg);
<         if (!insert_mn(arg,tid))
<         {
<             dbg ("Error insertando mn_dormant\n");
<             pthread_cancel(tid);
<         }
<
<         dbg ("I received a BU with the dormant flag set\n");
<         dbg ("Home address      %x:%x:%x:%x:%x:%x:%x:%x\n",NIP6ADDR(&arg-
>dst));
<         dbg ("Care-of address  %x:%x:%x:%x:%x:%x:%x:%x\n",NIP6ADDR(&arg-
>remote_coa));
<         tssetsec (lifetime,ntohs(arg->bu->ip6mhbu_lifetime <<2));
<         dbg ("Lifetime %ld s\n",lifetime.tv_sec);
<     }
<     /*Si no ha enviado el flag dormant, compruebo si es un MN dormant que ha cambiado
al estado activo*/
<     else
<     {
<         struct in6_addr aux;
<         struct mn_dormant *mn;
<
<         dbg("Home address  %x:%x:%x:%x:%x:%x:%x:%x\n",NIP6ADDR(&arg->dst));
<         mn = (struct mn_dormant*)malloc(sizeof(struct mn_dormant));
<         if (mn == NULL)
<         {
<             dbg ("Malloc failed!\n");
<         }
<         else
<         {
<             pthread_mutex_lock (&list_mutex);
<             if (mns_dormant.first !=NULL)
<             {
<                 mn->next = mns_dormant.first;
<                 while (mn->next != NULL)
<                 {
<                     aux = mn->next->bu_mn_dormant.dst;
<                     dbg("Estoy en ha_recv_bu dentro del if\n");
<                     dbg                                     ("aux
%x:%x:%x:%x:%x:%x:%x:%x\n",NIP6ADDR(&aux));
<
<                     if (IN6_ARE_ADDR_EQUAL(&aux,&arg->dst))
<                     {
<
<                         dbg("He recibido el bu!!\n");
<                         mn->next->recv_bu=1;
<                         varg->mn = mn->next;
<                         break;
<                     }
<                     else
<                         mn = mn->next;
<                 }
<             }
<             pthread_mutex_unlock(&list_mutex);
<         }
<     }

```

```
<
<     varg->args=arg;
<
1438c938
<     if (pthread_create(&worker, NULL, ha_recv_bu_worker, varg)) {
---
>     if (pthread_create(&worker, NULL, ha_recv_bu_worker, arg)) {
1468,1469d967
<     int distance; //linia afegida
<
1476,1489d973
<
<     /*Inicio las variables*/
<     mns_dormant.first=NULL;
<
<     if (pthread_mutex_init(&list_mutex,NULL))
<         return -1;
<
<     distance_threshold(&distance);
<
<     if (distance !=NULL)
<         DISTANCE_PAG_AREA = distance;
<     else
<         DISTANCE_PAG_AREA = 1;
<
```