Simulació de xarxes WSN (Wireless Sensor Network) per a la detecció d'incendis

Víctor López Muñoz

Enginyeria Tècnica d'Informàtica de Gestió

Àrea: ADMINISTRACIÓ DE XARXES I SISTEMES OPERATIUS **(AXSO)**

Consultor: Carles Estorach Espinós

Curs acadèmic: Setembre 2013 – Gener 2014

Índex de continguts

Índex d'il·lustracions

1. Introducció

Estem immersos a la gran xarxa d'Internet. És una eina indispensable per moltes empreses, una font de coneixement per a estudiants e investigadors, un magatzem de dades per a empreses de màrqueting, en definitiva és un medi molt present a les nostres vides i cada vegada més ficat en la nostra vida privada.

Dintre de la immensitat d'aplicacions existents al mercat que utilitzen Internet, no poden quedar fora les aplicacions WSN. Són precisament les xarxes de sensors sense fils les que poden treure més profit de la gran Xarxa que és Internet.

Mitjançant aquestes petites xarxes de sensors es poden monitoritzar per exemple, paràmetres de temperatura, humitat, moviment, lluminositat, etc. A banda d'aquesta característica, també permeten anticipar-se a esdeveniments com poden ser incendis, inhalació de gasos tòxics, moviments d'estructures.

Existeixen infinitats d'aplicacions que podrien utilitzar les WSN. La versatilitat d'aquesta tecnologia fa que sigui molt interesant realitzar un treball relacionat amb les xarxes de sensors sens fils.

1.1 Objectius

El principal objectiu cercat en aquest TFC és l'aprenentatge d'un nou sistema de xarxes sense fils. A banda d'això, també es pretén millorar la tasca de detecció d'incendis mitjançant una tecnologia que sembla esta feta a mida per aquest tipus de tasca.

Es vol estudiar un sistema on es pugui monitoritzar les temperatures dels diferents nodes. Es pensa en crear un entorn virtual mitjançant eines de simulació que mostra un escenari d'una xarxa WSN.

2. Planificació

S'exposen les tasques portades a terme per a l'execució del projecte. Al començament es va pensar realitzar el projecte utilitzant l'SO TinyOS, però després d'un període de proves e instal·lacions vaig veure que la preparació de l'entorn de treball m'estava portant bastant temps degut a molts problemes que em sortien en la instal·lació. La idea era poder simular la xarxa virtual de sensors sense fils en TinyViz, però els problemes trobats sobretot amb l'intèrpret de Python em van fer prendre la decisió d'utilitzar Contiki.

2.1 Tecnologies disponibles

Tenint en compte la restricció temporal per a realitzar el TFC, s'han avaluat diversos SOs existents per a realitzar la simulació d'una WSN. A banda de l'elecció de l'SO cal dir que també calen altres eines que a continuació es detallen:

- Màquina Virtual VMware-Player per a poder executar l'SO i simular l'escenari d'una xarxa WSN.
- Contiki. És l'SO escollit ja que permet simular escenaris virtuals d'una xarxa WSN d'una forma molt més còmode en comparació amb els altres sistemes. Els components es programen en llenguatge C i disposa d'exemples per a poder comprovar el funcionament dels diferents components d'una WSN.
- InstantContiki 2.7. L'SO emprat per a realitzar l'estudi del projecte.
- Ubuntu 12.04.2 LTS.
- Compilador gcc. Per a poder compilar el codi desenvolupat en llenguatge C.

Cal indicar que en aquest període també es va realitzar el Pla de Treball.

Data d'inici	22/09/2013
Data de finalització	4/10/2013

2.2 Familiarització amb l'SO Contiki

En aquesta fase de la planificació el projecte es centra en el funcionament de l'SO i de com es programen els nodes sensors i els nodes gateways i encaminadors.

S'estudia l'estructura de codi que ha de tenir una mota i com es comuniquen entre les altres motes i el propi encaminador o gateway. També s'analitza com es comunica l'encaminador i un PC mitjançant el protocol TCP/IP.

Iniciació en la simulació i creació de motes virtuals i execució de les mateixes revisant els paràmetres de sortida tant del port sèrie com dels canals de radio freqüència.

Es realitzen proves de funcionament amb aplicacions existents dintre del mateix SO i amb codi propi dissenyat.

Realització de la PAC2.

Data d'inici	5/10/2013
Data de finalització	8/11/2013

2.3 Simulació

Coneixement del protocol RPL i proves amb encaminadors i nodes. També es prova la lectura de les sondes mitjançant protocol HTTP.

S'aprofundeix una mica en el sistema de missatges de control que utilitza el protocol RPL i també en l'estudi dels paquets de dades enviats pels nodes sense fils.

Es porta a terme la realització de la PAC3.

Data d'inici	9/11/2013
Data de finalització	13/12/2013

2.4 Documentació

Durant aquests dies es prepara tota la documentació relacionada amb el TFC i la presentació. Acabo unes proves de lectura de temperatura realitzades que no vaig poder entregar a la PAC3 perquè no estaven acabades i entrega.

3. L'incendi

Per a que existeixi una combustió han de estar presents sempre els següents tres elements: l'oxigen, la calor i el combustible. A banda de l'existència d'aquests elements, també és molt important la seva proporció.

Figura 1. Triangle de foc

En altres paraules es podria dir que existeix foc si existeix també un combustible, per exemple les mateixes fulles seques d'alguna part del bosc o camp, un comburent, en el nostre cas l'oxigen, i una energia d'activació, com pot ser una cigarreta, un cristall que mitjançant els rajos de sol fa un efecte lupa i es converteix en un perillós element que proporciona una energia d'activació, un llampec, etc…

Si no hi ha combustible, el foc no pot prosperar, si la proporció de comburent és molt petita, el foc no pot començar, de la mateixa manera que si no existeix un focus de calor suficientment potent, el foc tampoc pot començar.

A banda d'aquests tres requisits, també hem de tenir en compte la proporció de comburent necessària per a que existeixi el foc. A l'atmosfera, l'oxigen es troba present en una proporció d'un 21%, juntament amb altres gasos. Per tant, en cas d'incendi en un bosc, el comburent es troba en aquesta proporció. Una de les tècniques emprades per exterminar un incendi és baixar la proporció de comburent, en aquest cas l'oxigen, per sota del 14%. Per sota d'aquest llindar, la combustió es retarda molt i fins i tot, arriba a desaparèixer.

En condicions normals, no és fàcil que existeixi un incendi en un bosc, però introduint una energia d'activació, és molt fàcil que es declari l'incendi i la seva propagació dintre d'un escenari on es compleixen les condicions descrites anteriorment. L'estiu és l'estació de l'any on podem trobar escenaris amb un alt risc d'incendi, ja que el combustible, en aquest cas les plantes, fulles, fusta dels arbres… està molt més sec. En altres paraules, el bosc a l'estiu, té un combustible molt més eficient que no pas a l'hivern.

També cal dir que el mateix temps a l'estiu, alguna vegada fa el paper d'energia d'activació en un incendi, degut a les altes temperatures existents en aquesta època.

En aquest projecte realitzarem un estudi amb nodes sensors de temperatura, tot i que no és l'eina més ràpida per a la detecció d'incendis, és un excel·lent instrument per a indicar si existeix un incendi. Un sensor de fum seria molt més eficient per a la detecció d'un incendi, ja que el temps de detecció d'un sensor de fum és més curt que el temps de detecció d'una sonda de temperatura. La sonda de temperatura detecta un augment de la temperatura quan ja existeix flama, mentre que un sensor de fum està detectant un incendi en el començament de la combustió. Aquesta diferència de temps en la detecció pot ser vital a l'hora de portar a terme la tasca d'extinció de l'incendi.

La següent il·lustració reflecteix les fases de l'incendi en les que actua cada detector. La corba és per a un incendi iniciat per sòlids.

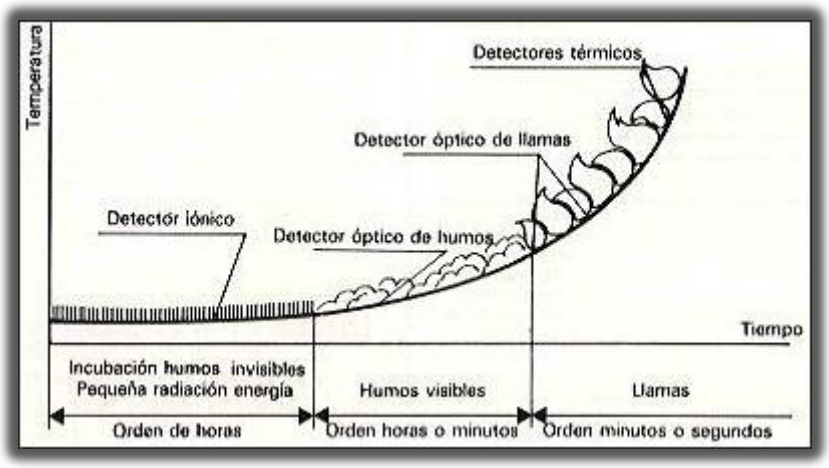

Figura 2. Temps de detecció d'un incendi iniciat per sòlids

4. Que és una WSN

Gràcies a les xarxes de sensors sense fils (WSN) podem crear xarxes de dispositius de captura constant que ens permeti emmagatzemar dades i controlar una sèrie de paràmetres ambientals o no. Per exemple, es pot controlar la temperatura d'un terreny en concret, o la humitat de la terra d'aquest mateix terreny. També es pot controlar si una estructura és estable o no, per exemple un pont, mitjançant sensor de moviment molt sensibles. Aquestes són només unes poques aplicacions en comparació amb la infinitat d'aplicacions on podríem utilitzar les WSN.

Bàsicament i explicat d'una manera molt simple, una xarxa de sensors sense fils (Wireless Sensors Network) és una xarxa de dispositius amb sensors que treballen en una tasca comú. Els sensors tenen la capacitat de capturar esdeveniments presents en un ambient, per exemple la qualitat de l'aire, detecció de substàncies contaminants etc… i comunicar les dades de les lectures efectuades pels sensors cap a un element coordinador com pot ser un node gateway. Una vegada arribades les dades al node coordinador, es poden transmetre cap a Internet, i monitoritzar les lectures dels sensors en temps real.

Les xarxes WSN tenen algunes característiques adaptades de les xarxes ad-hoc. A banda d'aquestes característiques adaptades, també tenen altres característiques pròpies. Algunes d'aquestes característiques son:

- Topologia dinàmica: És una topologia canviant.

- No s'utilitza infraestructura de xarxa: Una WSN no necessita tenir una infraestructura per poder operar. Els nodes poden actuar com emissors, receptor, i fins i tot a vegades com encaminadors.
- Consum energètic: Un node sensor ha de tenir un consum molt baix. Per tant, el seu processador i el seu transceptor de radio han de tenir un consum molt baix.

4.1 Components del sistema.

En una xarxa de sensors sense fils, trobarem els següents components:

- Nodes sense fils o motes.
- Gateway / Portes d'enllaç / Coordinador / Encaminadors.
- Estacions base.

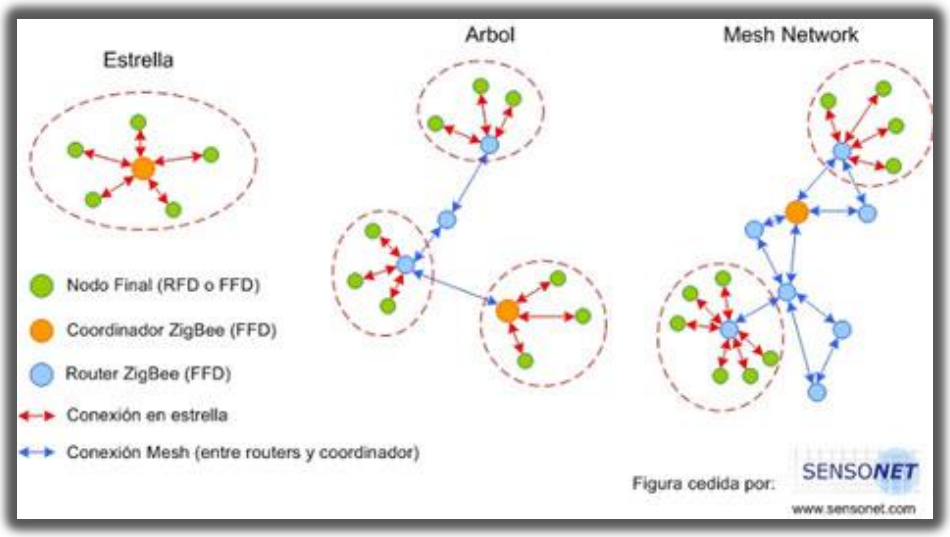

Figura 3. Components d'una WSN

4.1.1 Node sense fils o motes

Son dispositius que capten informació que prové de l'entorn on es troben. Aquests elements poden processar aquesta informació i enviar.la cap a un destinatari. Una mota es caracteritza pel seu baix nivell de consum d'energia i per un preu reduït. Un dels protocols més utilitzats per a la comunicació per radio freqüència de les motes és ZigBee. Normalment el node sense fils té el següent hardware:

- Alimentació.
- Processador.
- Sistema de comunicació sense fils.
- Sensors.
- Memòria.

4.1.1.1 Alimentació

El consum d'energia be donat per la suma del consum del sensor, el processament de la dada i la comunicació. On es consumeix més energia en una mota és en la transmissió de la informació. Avui dia les motes existents permeten utilitzar bateries de tipus AA de 1.5 V. Normalment amb quatre bateries el node pot tenir una autonomia de més de quatre anys.

Està molt relacionat amb aquest apartat el tipus de protocol que utilitzem per a realitzar les transmissions de dades. El més emprat és ZigBee. Aquest protocol té les següents característiques:

- Quan està transmetent dades té un consum de 30 mA, mentre que quan es troba en estat de repòs, el consum és de 0,3 µA (molt poc).
- Quan es realitza un enviament amb aquest protocol, immediatament després d'haver realitzat l'enviament, el sistema entra en estat de repòs. Aquesta situació facilita l'estalvi d'energia de la mota i el baix consum de la mateixa.
- La velocitat de transferència pot arribar fins a 250 Kbps. És una velocitat molt petita, però si tenim en compte que la quantitat de dades enviades és molt petita i a més l'enviament de dades es realitza en un període de temps més o menys llarg, arribem a la conclusió de que la velocitat de transferència no és molt important en aquests sistemes.

Aquestes són algunes de les característiques més importants del protocol ZigBee. Cal destacar que amb l'estàndard 802.15.4 es té un consum en la transmissió de 2 mA. Bastant menys que amb el protocol ZigBee. Si el volum de dades a transmetre és petit, llavors estem en condicions d'escollir l'estàndard 802.15.4. Caldrà tenir en compte la topologia de la xarxa per escollir entre ZigBee o l'estàndard 802.15.4.

4.1.1.2 Processador

Aquest és el component que pot interpretar i processar les dades per a poder-les transmetre cap a una estació.

Els processadors de baix consum més emprats dintre d'una mota o node són:

- Intel 8051. És un microcontrolador desenvolupat per Intel l'any 1980.
- Atmel AVR. Microcontroladors RISC d'Atmel.

- MSP430. És una família de microcontroladors fabricat per Texas Instruments.

4.1.1.3 Sistema de comunicació sense fils

És un dispositiu que permet enviar i rebre dades mitjançant radio freqüència amb altres dispositius dins del seu rang de transmissió.

Els nodes utilitzen la banda ISM. Aquestes bandes es poden utilitzar sense necessitat de llicència. Sempre s'ha de respectar les regulacions que limiten els nivells de potència transmesa.

Les xarxes WSN utilitzen les següents bandes:

- 433 – 464 MHz ISM band Europe.
- 902 – 928 MHz ISM band America.
- 2.4 – 2.5 GHz WLAN, ZigBee.

ISM (Industrial, Scientific, Medicine)

Les tasques d'emissió i recepció les coordina el transceptor. Els estats d'operació d'aquest component són:

- Emetre
- Rebre
- Dormir.
- Inactivitat.

Quan no s'està enviant ni rebent, les comunicacions de radio estan des habilitades (apagades). Alguns dels sistemes més coneguts de comunicació per radio son Chipcon CC2420, Xemics XE1205, Chipcon CC1020.

4.1.1.4 Sensors

Són dispositius hardware capaços de produir una resposta davant d'un canvi en un entorn físic, com pot ser humitat, temperatura.

Aquests dispositius detecten canvis físics a l'entorn on es troben. La senyal analògica detectada pel sensor, es convertida per un convertidor analògic digital a una senyal digital. Posteriorment, aquesta senyal és enviada al controlador per a ser processada.

Generalment, els sensor solen ser components molt petits que tenen un baix consum d'energia. Existeixen tres tipus de sensors:

- *Passius unidireccionals*. Tenen bastant definida la direcció des de on han de captar la informació.
- *Actius*. Son els sensors que han de generar algun esdeveniment en particular per a poder realitzar la mesura o presa de dades. Un exemple d'aquest tipus de sensor pot ser un sonar.
- *Passius omnidireccionals*. Són els sensors que són capaços de captar dades sense necessitat de produir cap esdeveniment dintre de l'entorn. Utilitzen l'energia per amplificar la senyal analògica captada.

4.1.1.5 Memòria

El tipus de memòria més emprat són la memòria integrada en el chip del microcontrolador amb la memòria flash.

Aquestes memòries flash permeten que la informació que emmagatzemen no es perdi si es desconnecta de la corrent. Segons la utilització existeixen dos tipus de memòries:

- Per emmagatzemar dades recollides per l'aplicació.
- Per emmagatzemar el programa del dispositiu.

4.1.2 Gateway, porta d'enllaç, coordinador

Aquest component és l'encarregat de col·lectar dades de mesures de la distribució de nodes. El node coordinador o Gateway és el responsable de fer la interconnexió entre la xarxa de sensors i una xarxa de dades (TCP/IP). El node coordinador no té cap element sensor. La funcionalitat d'aquest node és la d'actuar com a pont entre dues xarxes de diferent tipus. A aquest component també se'l coneix amb el nom de porta d'enllaç.

4.1.3 Estació base

És l'ordinador que recol·lecta les dades. Normalment totes les dades recollides pels nodes van cap a un Servidor dins d'una base de dades.

5. Exposició de diferents SO

Cal fer un estudi dels diferents SO per a xarxes WSN. A continuació es detallen uns quants, però existeixen molt més Sos dedicats a aquesta tasca.

5.1 Coneixement d'alguns SO existents.

Descartat TinyOS i TOSSIM per a realitzar la simulació, començo una fase de descobriments d'altres sistemes existents igualment factibles per a l'anàlisi del funcionament d'una xarxa WSN.

TinyOs. En principi es pensava en utilitzar aquest SO per a programar les motes i realitzar les simulacions, però una vegada insta-lat l'entorn de treball, vaig veure que el simulador TOSSIM, que és el simulador que utilitza TinyOS quan no tens una mota físicament i vols realitzar una simulació virtual del seu comportament, estava molt enfocat no a simular el mon real, si no al comportament del mateix SO. A banda d'això, les aplicacions en TinyOS s'escriuen en un dialecte de C anomenat nesC, cal familiaritzar-se amb nesC per a poder desenvolupar aplicacions sota TinyOS.

El simulador TOSSIM només suporta motes micaz. La compilació d'una mota simulada pot llançar molts errors, i una vegada compilada, s'ha d'iniciar l'intèrpret de Python que segons la versió de Python instal·lada, també provoca molts errors a l'aplicació. Davant d'aquestes situacions, vaig optar per canviar d'SO, ja que havia de complir les dades proposades al pla de treball

VisualSense. És un entorn específic per a xarxes de sensors. Es va desenvolupar gràcies a un projecte de la universitat de Berkley anomenat Ptolemy. El primer producte d'aquest projecte es va aconseguir l'any 1993. VisualSense no pertany al projecte Ptolemy. Un segon projecte anomenat Ptolemy II va iniciar-se l'any 2000, però no va ser fins a l'any 2004 que VisualSense va començar a formar part de Ptolemy II.

Aquesta eina és un entorn de modelatge. Es poden simular situacions bastants fidels a la realitat, com per exemple, un obstacle que atenuï les senyals que interfereixen amb l'obstacle, modelar els nodes sensors mitjançant diagrames de blocs, e inclús modelar i simular aspectes físics de l'entorn com poden ser la temperatura , lluminositat, etc.

Existeix una aplicació anomenada Viptos que permet crear a partir de diagrames de blocs existents a VisualSense, el codi en llenguatge nesC de la mota i poder efectuar simulacions en TOSSIM.

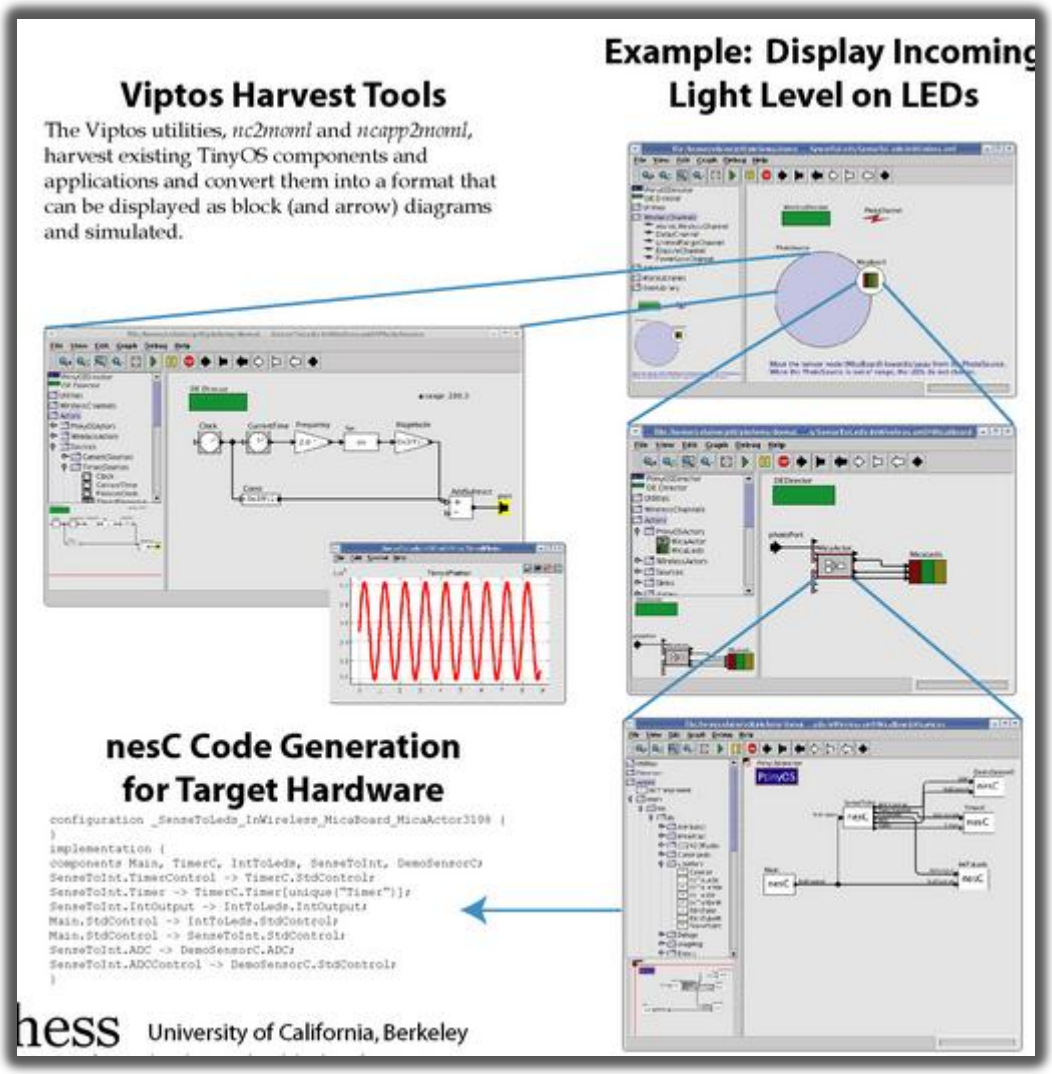

Figura 4. Viptos

Em semblava una eina molt interesant, i de fet em continua semblant molt interessant, però la corba d'aprenentatge és molt elevada i és necessari disposar de més temps per aconseguir un coneixement bàsic d'aquest entorn.

Ns-2. Es un simulador de xarxes basada en esdeveniments discrets, és a dir, simula amb unes certes variables que donen sentit al sistema però que no varien el seu comportament durant el transcurs del temps.

Pot treballar amb simulacions TCP i UDP i suporta diferents protocols d'encaminament de missatges i multi difusió en xarxes sense fils i en xarxes cablejades també. Dintre d'aquest entorn de simulació existeix una extensió per a WSN. Aquesta extensió s'anomena *Mannasim* i serveix com escenari per provar algoritmes i protocols .

Avrora. Es composa d'un conjunt d'eines de simulació i anàlisi de programari dissenyat per a nodes WSN i Mica2 i microcontroladors AVR, dissenyada la seva arquitectura per estudiants al Norwegian Institute of Technology.

Contiki. És un SO open source que entre altres tasques permet utilitzar-se en sistemes amb molt poca memòria. Aquest SO té un simulador WSN que es diu cooja. Dintre d'aquest simulador es poden crear motes, configurar tipus de xarxes i protocols, com per exemple el protocol TCP/IP (en IPv4 i IPv6), UDP i també configurar components que adoptin el comportament d'un gateway dintre d'una xarxa de sensors sense fils. Està desenvolupat en llenguatge C i té una amplia comunitat de desenvolupadors.

Disposa de molta documentació, es poden compilar diferents tipus de motes (micaZ, sky, esb, Z1), el seu entorn de treball es bastant còmode, ja que disposa de diverses finestres on es mostra la sortida que genera la mota, l'entorn de la xarxa (network), la línia de temps on es pot veure en quin moment tenen lloc les comunicacions i un petit tauler de control de la simulació.

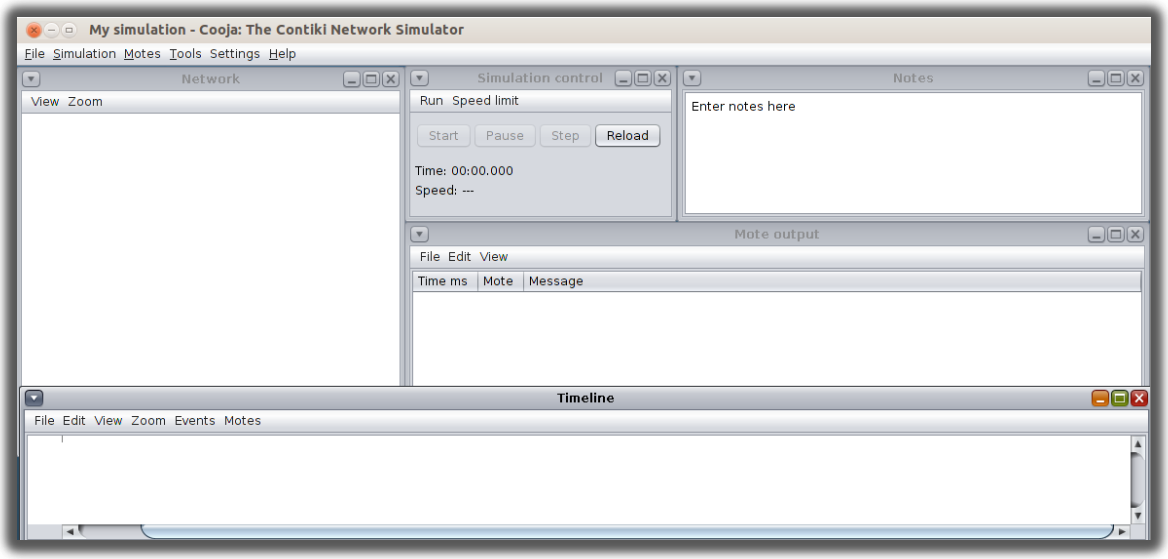

Figura 5. Entorn de simulació cooja

6. SO utilitzat per a la simulació de la WSN. Contiki

L'SO escollit per realitzar l'entorn simulat serà Contiki, ja que el seu simulador cooja conté molts exemples de compilació de motes, exemples de comunicacions entre components (motes – gateways), protocols TCP/IP i UDP i encara que quasi tota la documentació existent a la xarxa està en Anglès, resulta bastant assequible entendre els conceptes d'aquest simulador.

6.1 Configuració de l'entorn de treball

La plataforma on s'executarà Contiki serà una màquina virtual VMware Player, i el sistema instal·lat a la VM (Virtual Machine) és InstantContiki 2.7. Aquest sistema s'executa sota Linux, concretament en una distribució d'Ubuntu 12.04.2 LTS.

Cal tenir instal·lat també el compilador gcc que ens permetrà compilar un programa font en C i generar un programa executable binari en el llenguatge de la màquina on s'ha d'executar.

L'execució de cooja, el simulador per a WSN del SO Contiki, es realitza des d'un terminal i ubicats en la següent ruta:

user@ubuntu:~/contiki/tools/cooja$ ant run

Amb aquesta ordre executarem el simulador cooja.

7. Utilització del simulador Cooja

En aquest apartat es mostra la manera de com compilar una mota ja sigui del tipus Tmote Sky, micaz, Z1, ESB, Wismote a l'SO Contiki. L'estructura del codi d'una mota segueix el següent ordre:

```
#include "contiki.h"

 /* PROCESS() Aquesta declaració defineix el nom del procés. */

PROCESS(el_meu_proces, "Meu_process");


/* AUTOSTART_PROCESS()  selecciona quin o quins processos s'executaran quan el mòdul està carregat.  */

AUTOSTART_PROCESSES(&el_meu_proces);


/* PROCESS_THREAD() conté el codi del procés. */

PROCESS_THREAD(el_meu_proces, ev, data)

{

  /* El codi que es fica aquí, abans del procés BEGIN(), s'executa cada cop el procés es cridat */
```

```
  PROCESS_BEGIN();

  /* Inicialitzacions. */

  while(1) {

    PROCESS_WAIT_EVENT();

    /* Codi que s'executa mentre que no es detecta cap esdeveniment. */

  }

  PROCESS_END();

}
```

A continuació es mostra com es compila el programa en llenguatge C per poder crear la mota a posteriori.

7.1 Exemple de creació d'una mota o node

En aquest exemple es compilarà el codi del programari des d'una consola, per a poder crear una Tmote sky. La funcionalitat d'aquesta mota de l'exemple és recopilar dades dels sensors i també alguns valors del perfil d'energia.

El programa a compilar s'anomena sky-collect.c i es troba dintre del directori d'exemples de l'SO Contiki.

La primera instrucció per a realitzar la compilació és:

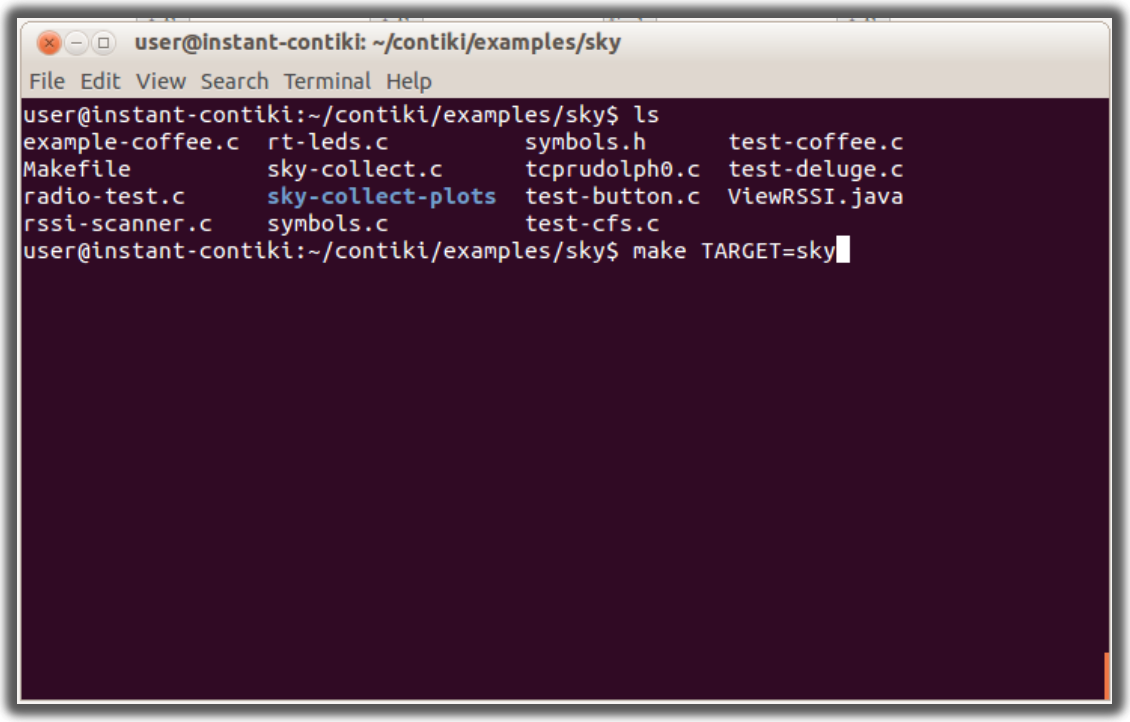

Figura 6. Compilació

A la imatge es pot veure el contingut del directori sky on es troba el programari a ser compilat i seguidament la instrucció utilitzada per la compilació. L'ordre make ha de tenir el paràmetre TARGET=sky per a poder indicar.li la plataforma a la que pertany la mota.

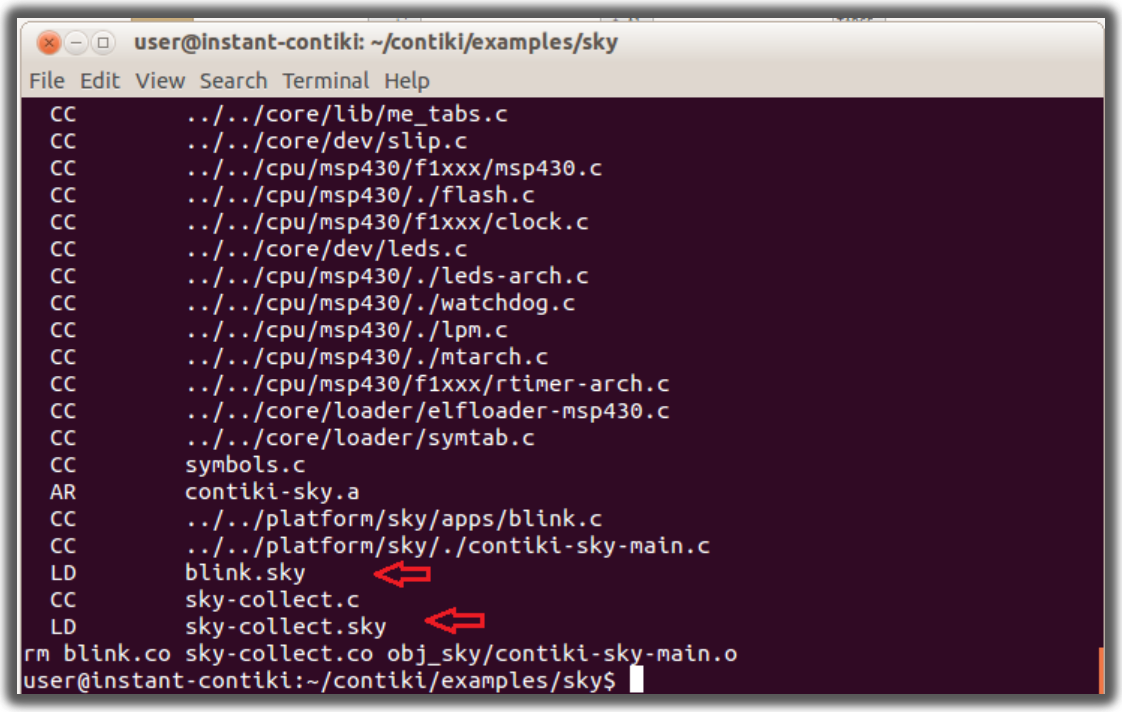

Figura 7. Resultat de compilació

El resultat es pot veure a la imatge a dalt. Amb una fletxa vermella estan indicats els dos programes compilats. Sky-collect.sky serà la mota que recopilarà dades i blink.sky que simula una mota que activa el led de la placa. A l'entorn de simulació cooja s'introduiran tots dos tipus de motes. Per a executar cooja cal fer el que s'ha indicat unes línies més a dalt a l'apartat 3.2.

user@ubuntu:~/contiki/tools/cooja$ ant run

Ara, una vegada en marxa el simulador, queda crear l'entorn a simular. Per a això anem a File> New Simulation. En aquest punt sortirà una finestra on es ficarà el nom que li volem donar a la simulació, seguidament escollim els paràmetres referents a la transmissió per radi, el retard de la mota que li volem donar i ja es pot crear la nova simulació.

La GUI que sortirà serà un entorn més o menys semblant a la imatge número 4 de l'apartat 4.1

Ara queda crear les motes que de moment ha estat compilat el codi des d'un terminal. Aquests passos són bastant intuïtius i no caldrà fer moltes especificacions amb imatges.

Escollim el menú Motes > Add Mote > Create new mote type, i aquí s'ha d'escollir una mota de tipus sky. A la finestra emergent que surt, hem de seleccionar "Browse" i situar-nos en la carpeta sky (/contiki/examples/sky). Dintre d'aquest directori ens trobarem amb els dos arxius amb extensió ".sky". Si escollim, per exemple, blink.sky i fem clic a Open anirem a una finestra des d'on podrem crear la mota o motes. En aquesta finestra s'ha de seleccionar el botó Create i automàticament ens sortirà una finestra indicant el número de motes que volem crear i on i com les volem crear (posicions X, Y aleatòries, manualment, en el·lipse, linear). Jo selecciono random positioning i li dic que faci dues motes. Add motes.

Si es vol afegir una altre mote sky, però aquesta vegada utilitzant la compilació del programa sky-collect.c, llavors caldrà seleccionar l'arxiu sky-collect.sky.

Ja muntat l'escenari es pot executar fent clic al botó Start del panell "Simulation Control"

7.2 Visualització de l'escenari simulat

Com ja s'indica a l'apartat anterior, la simulació es posa en marxa mitjançant el Panell "Simulation Control". A banda d'aquest panell, també es pot veure un gràfic de línia de temps que indica les senyals de radio enviades, cap a quins nodes, l'estat dels leds.

Al panell "mote outpout" es visualitza la sortida que genera la/les mota/es durant les comunicacions efectuades a la simulació. Bàsicament, en aquest entorn podem veure el comportament de la WSN. A més, des de la finestra network, podem escollir quins paràmetres volem veure, i també podem inspeccionar les motes fent clic amb el botó dret damunt d'una mota.

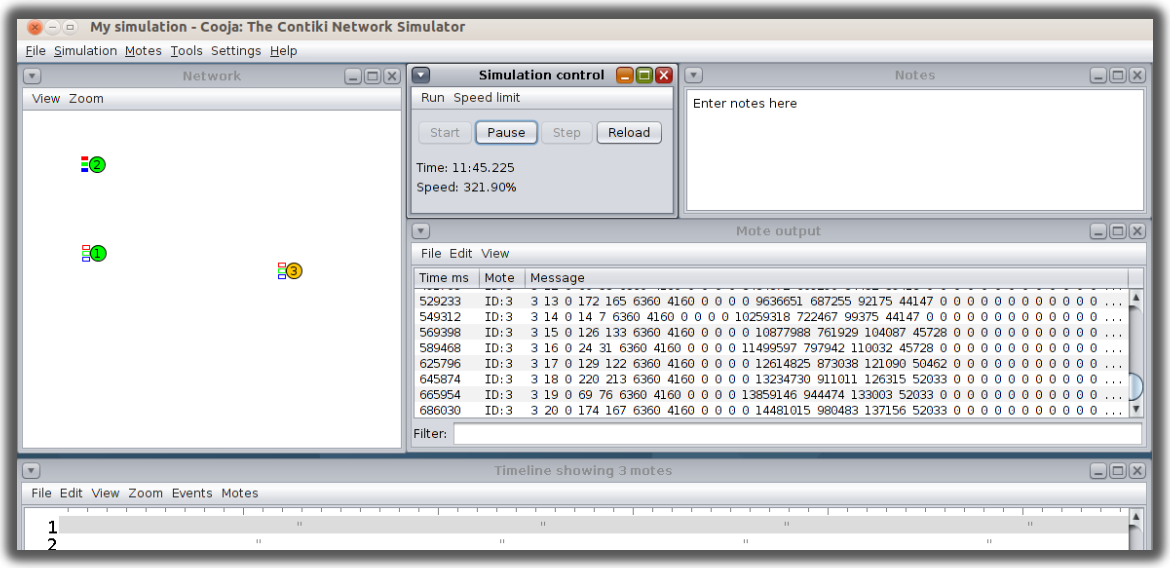

Figura 8. Simulació

7.3 Codi d'una mota i d'un element coordinador

En aquests moments estic desenvolupant una mota, intentant aprofitant codi d'altres tipus de motes i adaptant el codi al que jo vull fer. Cal dir que és molt necessari tenir un mínim domini del llenguatge de programació C per a poder desenvolupar el node en si. De tota manera existeix molta informació relacionada amb C.

L'element coordinador és una mota també, però sense sensor i ha de tenir la capacitat de poder treballar en xarxes amb diferents protocols. Amb protocol TCP/IP quan ha de transmetre dades de l'element coordinador cap a l'estació base mitjançant SLIP (Serial Line IP) i utilitzant l'estàndard 802.15.4 quan s'ha de comunicar amb els nodes de la WSN. A banda d'això, una vegada creat aquest escenari, ha d'existir alguna aplicació que pugui emmagatzemar les dades recollides.

8. Protocol d'encaminament en WSN (IPv6)

El protocol d'encaminament escollit per aquest treball és l'RPL. Està basat en un encaminament Vector-Distància i està definit per a xarxes WSN basades en IPv6.

Aquest és un protocol robust, que proporciona una alta escalabilitat, és a dir, davant d'un augment de treball, el protocol reacciona satisfactòriament, i està pensat per a utilitzar-se en ambients hostils.

Exemples de xarxes que utilitzen aquest protocol, són les xarxes de sensors sense fils, xarxes que controlen tota una flota de camions d'una empresa de transports, xarxes

que controlen el tràfic d'una ciutat mitjançant sensors sense fils que es poden trobar als automòbils, xarxes que controlen els aparcaments a una ciutat, etc.

Molts dels components d'aquestes xarxes es troben a la intempèrie, treballen amb nivells d'energia molt baixos i durant les 24 hores del dia, és per això que precisen d'un robust protocol per poder treballar en aquestes condicions. Aquest tipus de xarxes s'anomenen "Low power and Lossy Networks" (LLNs). Una traducció aproximada podria ser xarxes de baix consum amb pèrdues de paquets d'informació.

8.1 El protocol RPL

Les sigles RPL provenen de "Routing Protocol for Low power and Lossy Networks". Aquest protocol és una proposta de protocol d'encaminament per a xarxes (LLNs) feta pel grup de treball ROLL de la IETF. Es pot dir que RPL és un protocol d'encaminament per a WSNs (Wireless Sensor Networks) basades en IPv6.

Aquest projecte té finalitzats varis RFC ("Petició de comentaris"). Els RFC són documents que contenen una proposta oficial per a un nou protocol de la xarxa Internet.

En el nostre cas, uns dels primers RFC d'aquest projecte, el protocol RPL, són:

RFC5673 que fa referència als requeriments de l'encaminament en un escenari industrial.

RFC5548 que fa referència als requeriments de l'encaminament en un escenari urbà.

A continuació s'enumeren algunes definicions que ajuden a entendre el funcionament d'aquest protocol. Cal indicar que un graf es pot entendre com una representació abstracta d'un conjunt d'elements u objectes on alguns parells d'aquests elements estan connectats mitjançant uns vincles. Als objectes o elements els anomenen vèrtexs (per exemple els nodes sensors, encaminadors, gateways) i als vincles que els uneixen s'anomenen arcs o arestes (per exemple, la connexió sense fils). A continuació es detallen dos tipus de grafs.

- DAG. És un graf dirigit on tots els camins estan orientats de tal manera que no existeixen cicles. Tots els vèrtexs del graf tenen camins orientats cap a un o varis nodes arrel.
- DODAG. Un DAG amb un node arrel.

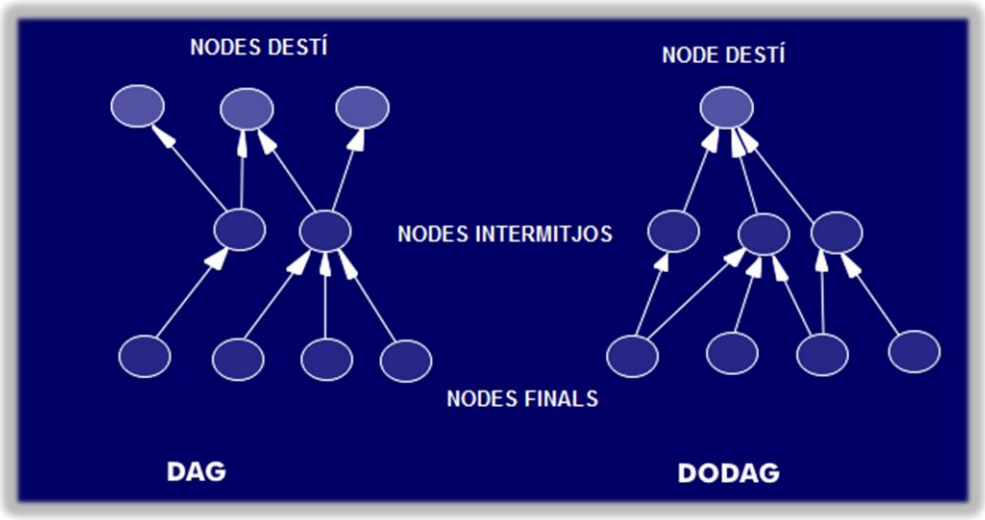

Figura 9. Representació graf DAG i DODAG

El protocol RPL es basa en la definició DODAG. RPL a la vegada utilitza una sèrie de missatges de control que són els següents:

- DIS: És utilitzada per a poder veure si existeixen DODAG a l'entorn i poder sol·licitar missatge DIO dels nodes RPL.
- DIO: Aquest senyal és enviat pel node RPL per a poder mostrar informació sobre el DODAG. Aquest senyal es pot fer servir per a que un node conegui als seus veïns.
- DAO: Actualitzen les taules d'encaminament dels nodes pares. També serveixen per propagar informació d'adreçament des dels nodes finals cap al node destí.

9. Aplicació WSN utilitzant protocol RPL

L'aplicació estudiada en aquest treball és una aplicació basada en un servidor Web que treballa amb el protocol d'Internet IPv6.

L'escenari s'ha simulat amb motes tipus Sky, i aquestes motes proporcionen els valors dels sensors de temperatura. Per poder transmetre la informació dels nodes cap a Internet, hi ha un encaminador que funciona amb RPL. Aquest encaminador fa de pont entre la xarxa sense fils dels nodes sensors que es comuniquen mitjançant l'estàndard 802.15.4 i la xarxa Internet que utilitza el conjunt de protocols TCP/IP.

Cal indicar que el codi de la mota ha sigut modificat per poder aconseguir certa variabilitat en la mesura de les temperatures. D'aquesta forma l'aplicació pot treballar amb diferents valors de temperatura dintre d'un rang determinat. Quan la temperatura

arriba a un cert valor s'encén el led de color vermell de la mota indicant que el node ha fet una lectura de temperatura alta.

9.1 Detecció de missatges de control RPL

Treballant en l'aplicació amb el simulador cooja de l'SO Contiki es poden identificar els diferents tipus de missatges de control d'RPL.

El primer missatge que s'identifica és DIS. Els nodes sensors, utilitzen aquest missatge per saber si existeix un graf DODAG. Cada node sensor provarà pel seu compte però d'una manera ordenada i periòdica, si existeix un DODAG. Mitjançant aquest tipus de missatge, s'està sol·licitant informació del DODAG als nodes que l'estan rebent.

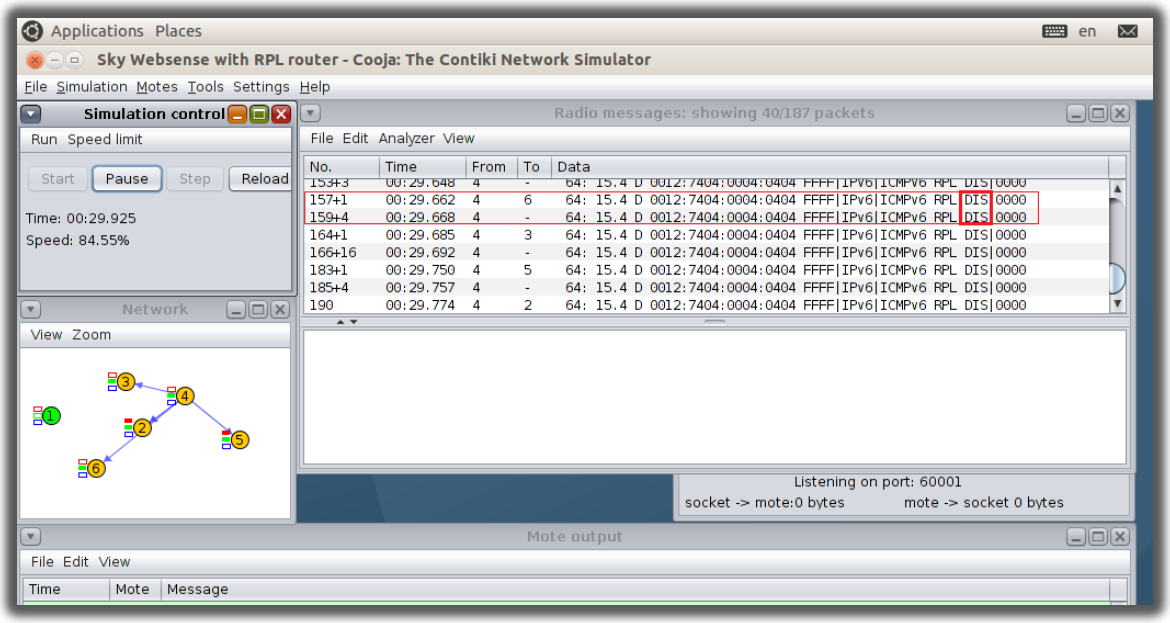

Figura 10. Missatge de control DIS en RPL

A la figura 2 es mostra la configuració de la xarxa. El node numero 1 de color verd és l'encaminador. L'encaminador encara no està activat, per tant, de moment no hi ha un graf DODAG. Es pot veure com el node sensor 4 envia un DIS al node numero 6. Com que no rep informació del DODAG per part del node numero 6, llavors envia un broadcast de missatge DIS.

El següent pas serà connectar l'encaminador per poder veure el comportament de la xarxa. Només connectar-se l'encaminador, el que fa es un broadcast del missatge DIO. Els broadcast dels missatges DIO són enviats pels nodes periòdicament. Quan un node rep aquest missatge el primer que fa es afegir a la taula de veïns al node que ha enviat aquest missatge. Després existeixen dues possibles opcions:

- El node que rep el missatge no coneix al DODAG. Llavors es processa de la següent manera:
    - Afegeix com a "best parent" al remitent del missatge. *Com a "best parent" s'entén com el node veí amb el que s'aconsegueix la millor ruta per arribar al node arrel.*
    - Assignació d'adreça IP si el node no la té.
    - Configuració del timer per a poder enviar un broadcast DIO amb la informació del nou DODAG.
    - Enviament de missatge DAO al node que ha enviat el missatge DIO avisant que és el "best parent" del node.

- El node que rep el missatge coneix al DODAG. Llavors es processa de la següent manera:
    - S'afegeix al remitent com a parent.
    - Es compara el remitent amb els altres parents i es comprova si és el "best parent".
    - Si és així, llavors el node envia un missatge DAO al remitent dient.li que és el seu "best parent".

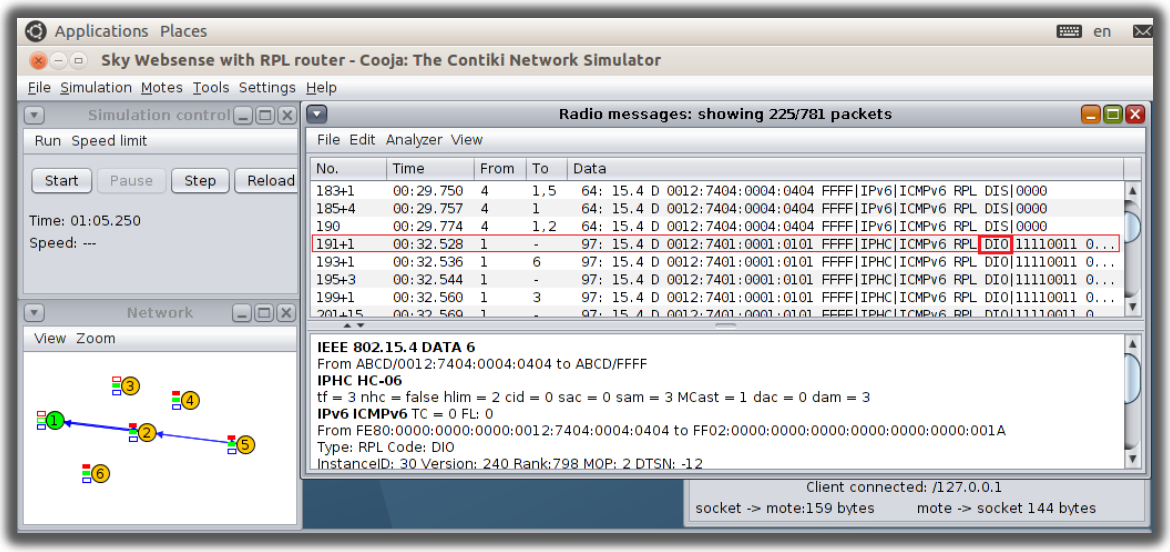

Figura 11. Missatge de control DIO en RPL

I finalment, els missatges DAO poden ser activats per algunes de les següents situacions:

- Si el node canvia de "best parent".
- Si es rep un missatge DIO del "best parent".
- Quan es rep un missatge DAO d'un node fill.

- Quan s'afegeix un node a una xarxa nova.
- Si el "best parent" té un *rank* massa elevat. El rank d'un node mostra numèricament la qualitat del camí per poder arribar del node en qüestió cap al node arrel (o sink). El node amb rank més baix és el propi node arrel i té un rank amb un valor de 256.

Cal indicar que un missatge DAO sempre serà enviat cap a un "best parent".

Quan un node rep un DAO, realitza les següents accions en funció del significat del missatge:

- Si s'ha rebut el missatge perquè el node ja no és el "best parent" del remitent, llavors treu al remitent de la taula d'encaminament i a totes les entrades que el tinguin com a next-hop.
- Si el missatge indica que el node és el "best parent" el remitent llavors afegeix al node a la taula d'encaminament, i a més, el missatge DAO és reenviat cap al "best parent".

Els missatges DAO són enviats d'un node cap al seu "best parent" i aquest el torna a enviar al seu "best parent". El missatge és reenviat de "best parent" a "best parent" fins arribar al node arrel, o l'encaminador en aquest cas. És d'aquesta manera que l'encaminador rep tots els missatges DAO i aprèn a arribar a tots els nodes existents a la xarxa.

En la següent figura es pot veure un exemple d'enviament de missatge DAO des d' un node (node 5) que prèviament havia rebut un missatge DIO del seu "best parent" (node 2). Cal aclarir que en aquest moment de la simulació el "best parent" del node 5 és el node 2 perquè és el node que té un rank més baix (718). Els altres nodes veïns del node 5 tenen el rank més alt (>718).

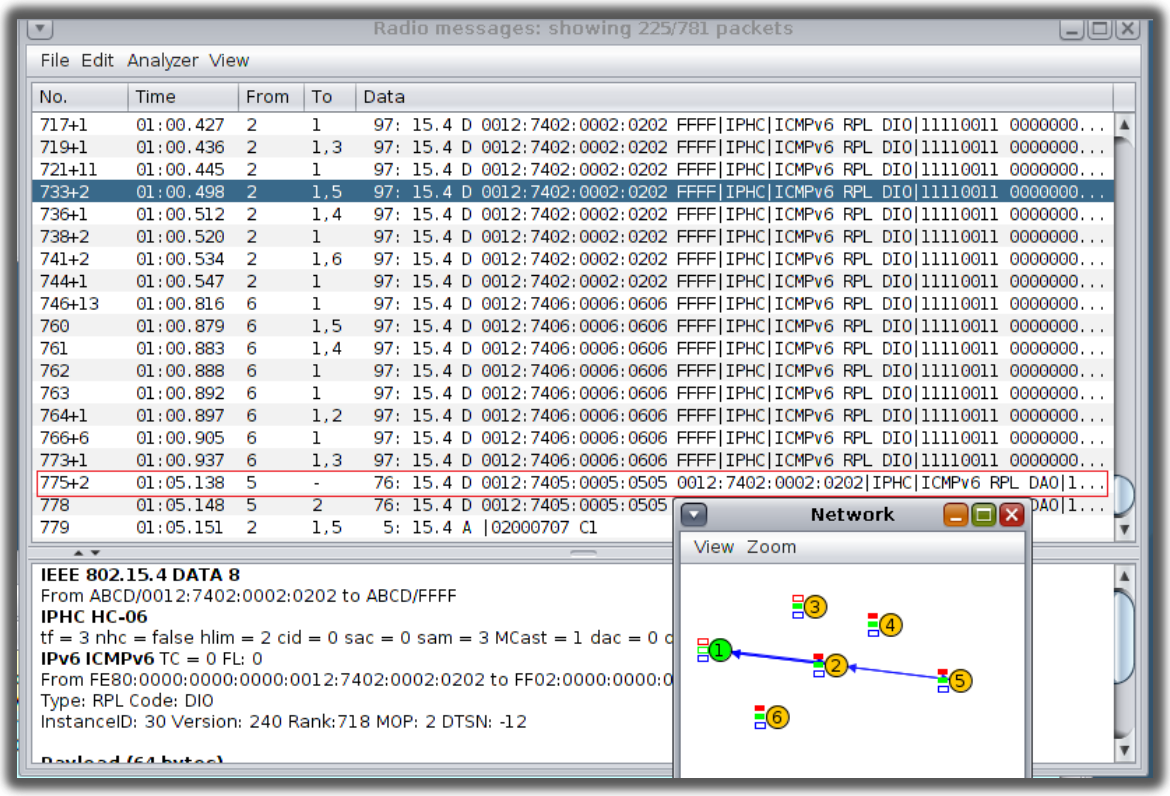

Figura 12. Log de missatges de radio

La línia marcada amb el cursor de color blau indica el moment en que el node 2 envia un missatge DIO cap al node 5. Com que la mota 2 és la "best parent" de la mota 5, el pròxim missatge que enviarà la mota 5 quan disposi del seu temps de transmissió, serà un missatge DAO, ja que al haver rebut el missatge DIO del seu node pare (node 2), s'activarà automàticament l'ordre d'enviament de missatge DAO. El rectangle de color vermell mostra l'enviament del missatge DAO de la mota 5.

10. L'encaminador

Després de veure tota la interacció dels missatges de control del protocol RPL per a xarxes WSN amb IPv6 per a poder estructurar un DODAG, és el moment de començar a veure els resultats del treball dels missatges de control.

El primer que s'ha de fer per a poder veure l'encaminador és connectar.se a ell mitjançant un terminal. En aquesta aplicació, la manera d'activar l'encaminador és la següent:

Quan l'aplicació està en execució s'ha d'obrir un terminal nou i anar a la carpeta de l'aplicació. Hem d'executar la següent instrucció:

..$ make connect-router-cooja

Quan s'executa aquesta instrucció, el que realment s'està fent és executar el Makefile que es troba a la carpeta de l'aplicació, que en aquest cas és ...**:~/contiki/examples/ipv6/sky-websense$**

Dintre d'aquest directori es troba l'arxiu Makefile. En aquest arxiu existeix un tros de codi igual que el que es mostra a continuació:

```
connect-router-cooja:    $(CONTIKI)/tools/tunslip6
        sudo $(CONTIKI)/tools/tunslip6 -a 127.0.0.1 aaaa::1/64
```

Llavors quan escrivim per consola connect-router-cooja, la instrucció que realment estem executant és tunslip6 al path que ens indica l'arxiu Makefile.

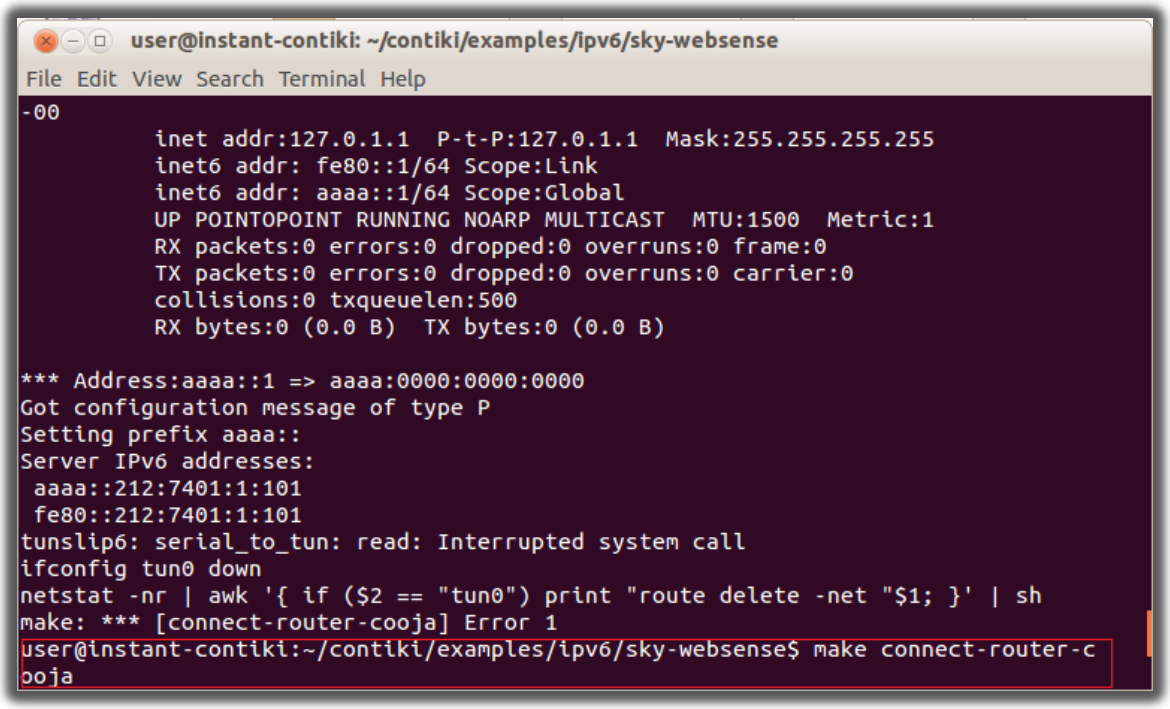

Figura 13. Execució de l'encaminador

Com es pot observar, quan executem l'encaminador amb make connect-router-cooja, realment el que estem fent és executar el programa tunslip6. Aquest programa és una eina que té l'SO Contiki que ens permet crear una connexió entre un protocol de comunicació sèrie que encapsula el protocol IP i una interfície de xarxa virtual. En altres paraules, permet comunicar un port amb una interfície virtual de xarxa, en el meu cas la interfície és tun0. D'aquesta manera es pot treballar per una banda amb el protocol IP, en aquest cas IPv6, i per l'altre amb l'estàndard 802.15.4.

Per tant, el nostre encaminador està utilitzant contínuament el programa tunslip6 per poder realitzar les transmissions de dades des dels nodes cap al nostre ordinador.

10.1 L'Encaminament a l'aplicació

En la taula d'encaminament un node té les rutes cap als nodes de la seva sub-DODAG. El node arrel té en la seva taula d'encaminament tots els nodes del DODAG (veure figura 6).

Si un node X ha d'enviar un missatge a altre node Y, i aquest node Y no està present en la taula d'encaminament del node X, llavors el missatge s'envia pel camí on el next-hop és el "best parent". Si aquest altre node no el té tampoc en la seva taula d'encaminament, l'enviarà al seu "best parent". Aquesta situació es pot anar repetint fins arribar al node arrel, que ell sap com arribar a tots els nodes del DODAG.

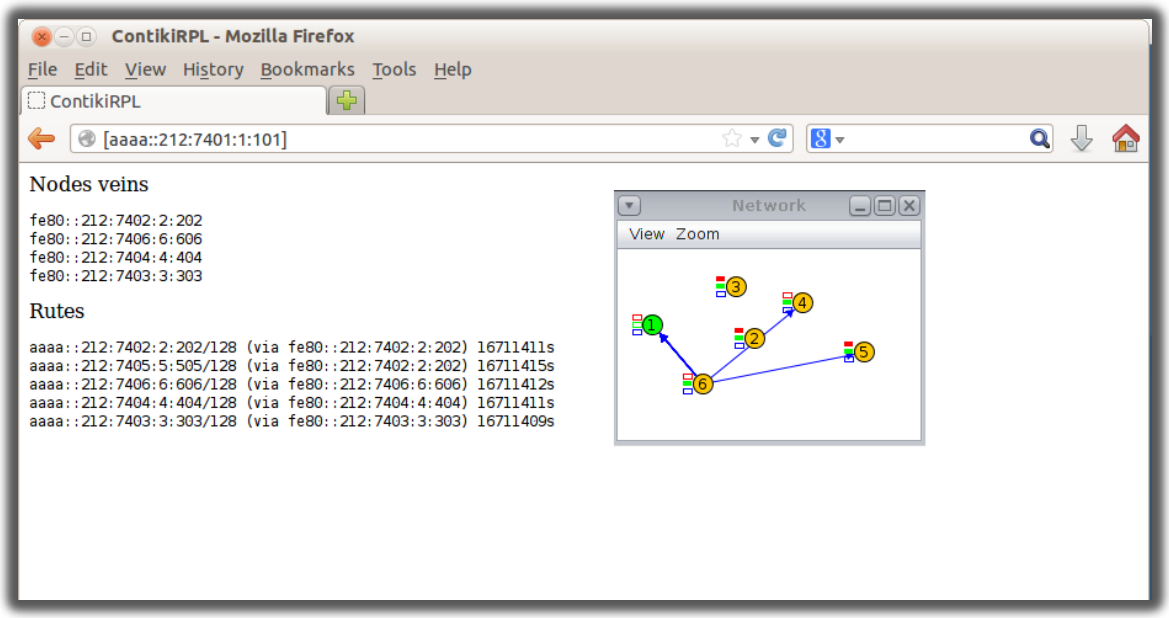

Figura 14. Taules de nodes veïns i rutes dins l'encaminador

11. Descripció de l'aplicació utilitzada

L'aplicació que s'està utilitzant per a simular l'escenari d'un control d'incendis en aquest cas, és sky-websense de l'SO Contiki 2.7. És una aplicació pensada per a motes tipus Sky i es basa en un servidor web accessible via IPv6 que s'utilitza per a proporcionar els valors de lectura dels sensors de la xarxa WSN.

L'aplicació treballa amb un encaminador que fa de pont entre la xarxa de sensors i Internet.

Per a poder obtenir les lectures de temperatura dels diferents nodes sensors, cal accedir al navegador i dirigir-se al node en qüestió indicant la seva adreça.

Per exemple, per accedir al node sensor número 2, hem d'indicar.li la següent adreça:

http:// [aaaa::0212:7402:0002:0202].

Per simplificar les adreces i fer menys cansada la tasca de realització de proves amb el codi, s'han abreviat les adreces registrant-les a etc/hosts. Dintre d'aquest fitxer s'han afegit les línies de les adreces a abreujar. A sota es mostra un exemple per abreujar les adreces.

For sky-websense

aaaa::0212:7402:0002:0202 Node_2

.....

Figura 15. Abreujament d'adreces

11.1 Accés a les dades d'un node sensor

Per poder accedir a la lectura del valor de temperatura o lluminositat d'un node en concret, hem d'obrir una connexió al host del node. Els nodes funcionen com petits servidors que el que fan és tornar els valors demanats pel client. A continuació es fa un seguiment del log de missatges de radio del simulador cooja per poder veure les dades intercanviades entre els dos extrems de la comunicació (node 1 i node 3).

A l'exemple es demanarà les dades del node número 3.

Comencem executant l'aplicació amb l'aplicació tunslip6 en execució (quan s'ha executat en un terminal la instrucció make connect-router-cooja). De fet aquesta execució es pot deixar treballant encara que l'aplicació sky-websense no estigui executant.se. Tunslip6 fa de pont entre el port de l'ordinador i la interfície de xarxa virtual de la simulació.

Havíem dit que es volia fer una consulta de la temperatura del node 3. En aquest cas la temperatura detectada al node és de 5ºC.

Figura 16. Lectura d'un node

Per veure en detall l'intercanvi de dades entre l'encaminador i el node 3 hem d'analitzar la finestra del log dels missatges de radio, i en concret, els payloads. Llavors es comença analitzant a partir de la línia número 239 que és on comença el diàleg HTTP.

No faré tot l'anàlisi de la comunicació perquè seria molt extens, però si que mostraré punts claus de la comunicació. A la figura 9 es pot observar en el payload quines dades ha enviat el node 1 al node 3.

Veient el log podem començar a intuir que està passant en aquest moment de la comunicació, ja que es pot apreciar la següent informació:

GET / HTTP /1.1

Host: node_3

User-A

Això és un inici de diàleg entre client i servidor per a obtenir un recurs amb el URL. La informació a obtenir es troba en el host node_3 i el nom del client és, segons la figura 10, Mozilla (línia 241).

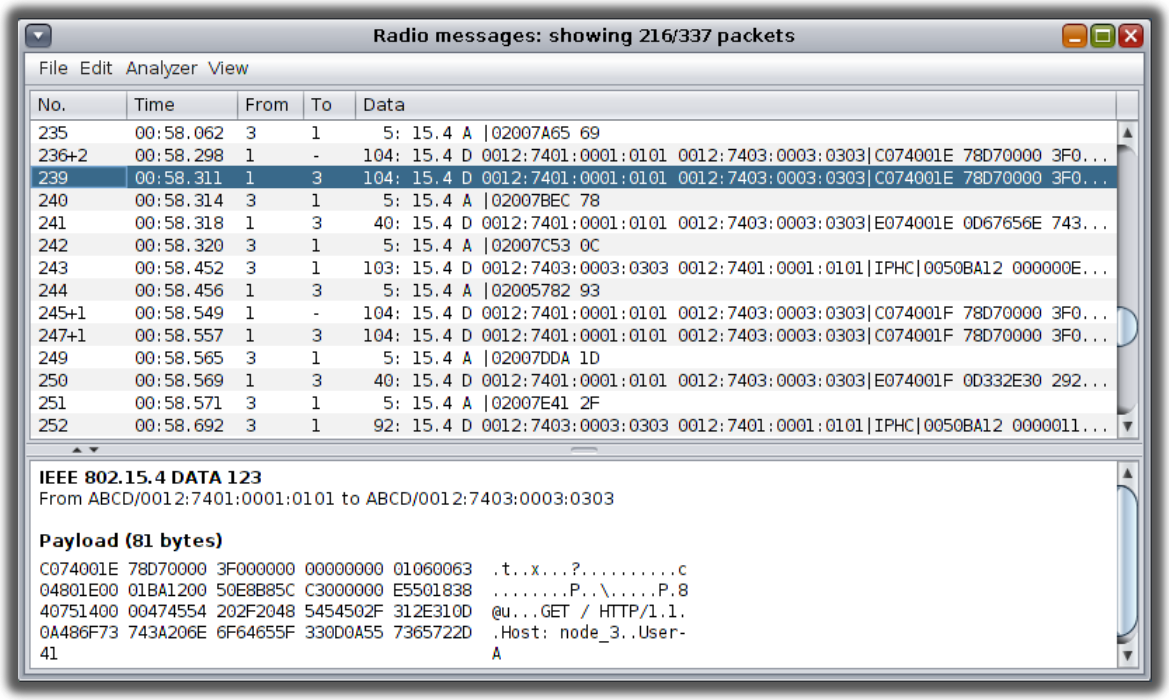

Figura 17. Log de missatges de radio

La línia 240 és una confirmació que envia el node 3 al node 1 conforme les dades han estat rebudes satisfactòriament.

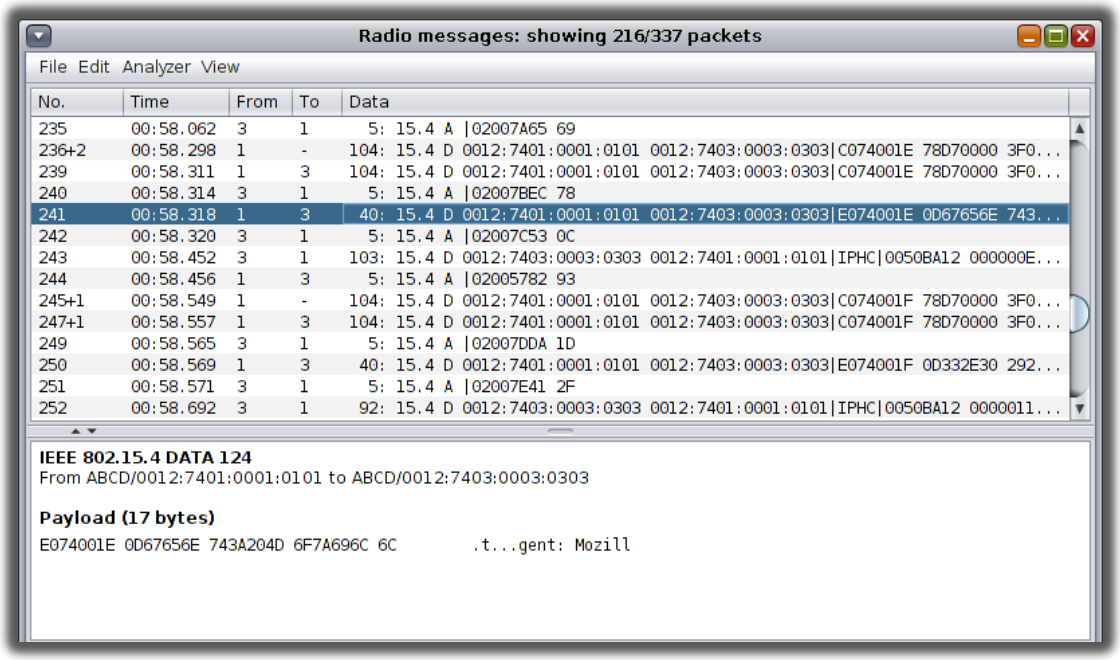

Figura 18. Nom del client (veure el Payload línia 241)

Per veure la resposta del node, en aquest cas, el servidor, mirem la línia 243. El Payload d'aquesta línia conté les següents dades. (veure Figura 11).

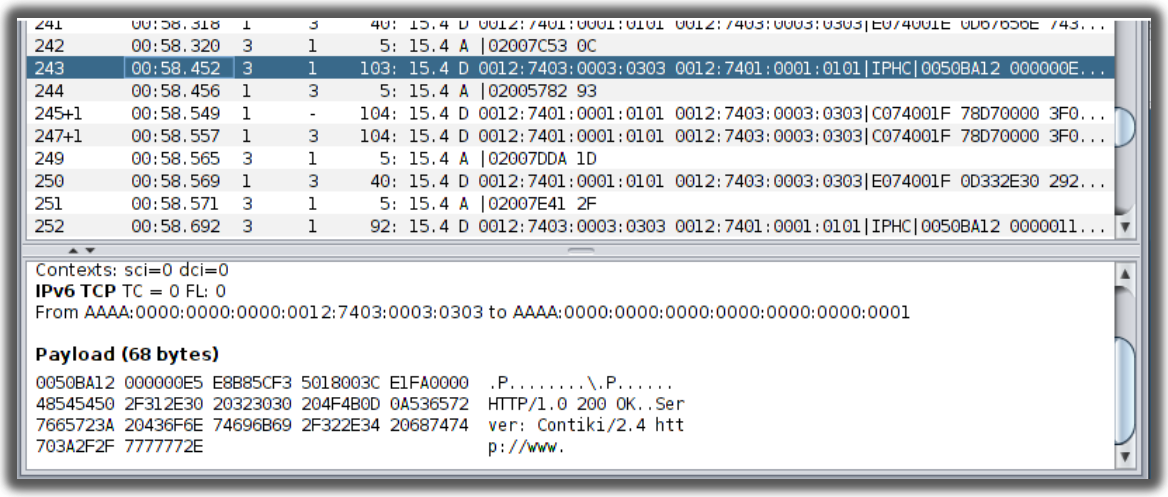

Figura 19. Missatge de connexió correcte al node 3

Establerta la comunicació entre els nodes, comença un intercanvi de missatges i dades per poder portar a terme la comunicació entre ells i així obtenir les dades que vol el client.

Finalment, es pot observar com el host envia les dades de temperatura cap al client a la figura 12 en el Payload de la línia 299.

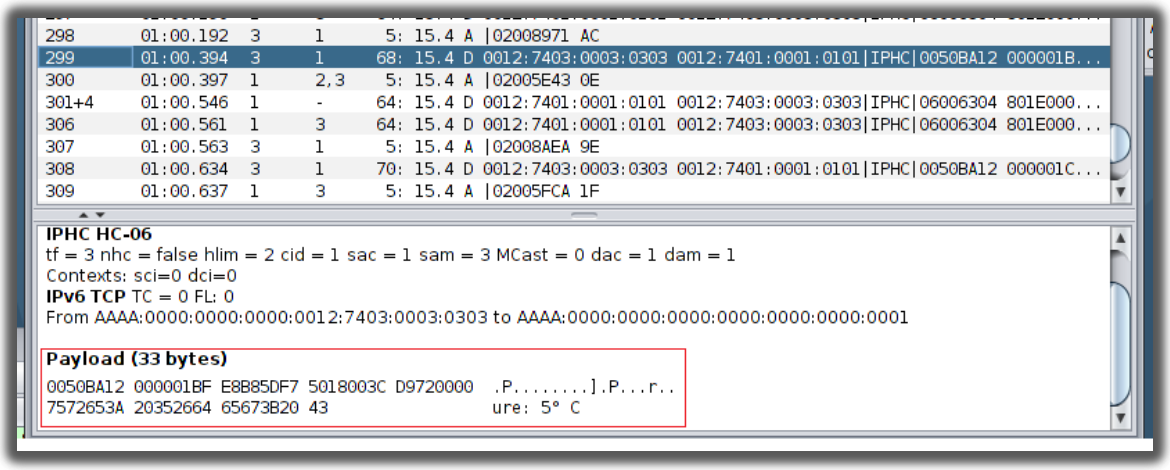

Figura 20. Enviament del valor temperatura al client

11.2 Modificació del codi de la mota o node sensor

Per a poder tenir un escenari més pròxim a la realitat, el codi dels nodes sensors (sky-websense.c) han sigut modificats de tal forma que puguin donar temperatures diferents. D'aquesta manera es pot treballar en situacions de lectures de temperatures màximes.

```
static int

get_temp(void)

{

  /*return ((sht11_sensor.value(SHT11_SENSOR_TEMP) / 10) - 396) / 10;*/

        return (unsigned short)rand() % 41+10;

}
```

Amb aquesta modificació s'aconsegueix obtenir valors de temperatura dintre del rang entre 10ºC i 50ºC. Variant el rang numèric es pot variar el rang de temperatures.

Al mateix arxiu, s'ha afegit codi per poder controlar el rang de temperatures amb els colors dels led i així poder controlar visualment si la lectura de temperatura es correcte. Es considera que una temperatura correcte si no arriba als 48 graus. D'aquesta forma es pot fer un control visual de l'entorn. La modificació en qüestió és la següent:

```
if(/*get_temp()*/temperature[sensors_pos]>48){

printf("Alarma. Se ha alcanzado una temperatura de: %d grados Celsius\n",temperature[sensors_pos]);

        leds_on(LEDS_RED);

        leds_off(LEDS_GREEN);

}
```

```
else{
printf("Temperatura de: %d grados Celsius\n",temperature[sensors_pos]);
        leds_on(LEDS_GREEN);
        leds_off(LEDS_RED);
 }
sensors_pos = (sensors_pos + 1) % HISTORY;
```

Si es detecta una temperatura per sobre dels 48 graus, s'encendrà el led de color vermell de la mota que ha donat aquesta lectura indicant visualment que la mota ha donat una lectura de alta de temperatura.

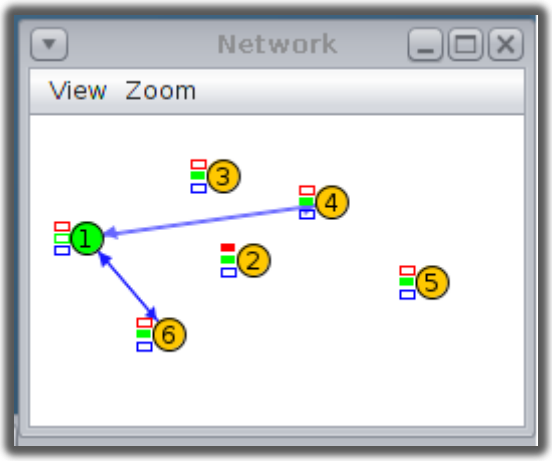

Figura 21. Control visual de temperatures

11.3 Recopilació de dades

Els nodes utilitzats a l'aplicació van enviant lectures de temperatures cada 5 segons. Llavors, és aquest l'interval de temps que ha de passar per a transmetre les lectures de les sondes cap a un arxiu log creat per poder manipular després les dades, ja sigui recopilant la informació mitjançant un programa que recol·lecti les dades importants de l'arxiu log, com poden ser el número de node i el valor de la temperatura, i posteriorment una base de dades, com per exemple PostgresSQL o MySQL, creant una taula on es pugui emmagatzemar aquesta informació. En aquest cas l'arxiu es diu logtemp.txt.

Cooja permet apuntar la sortida sèrie de la mota cap a un arxiu. Això es pot fer des de la finestra *Mote output* al Menú File > Append to file escollint el nom que li volem ficar a l'arxiu.

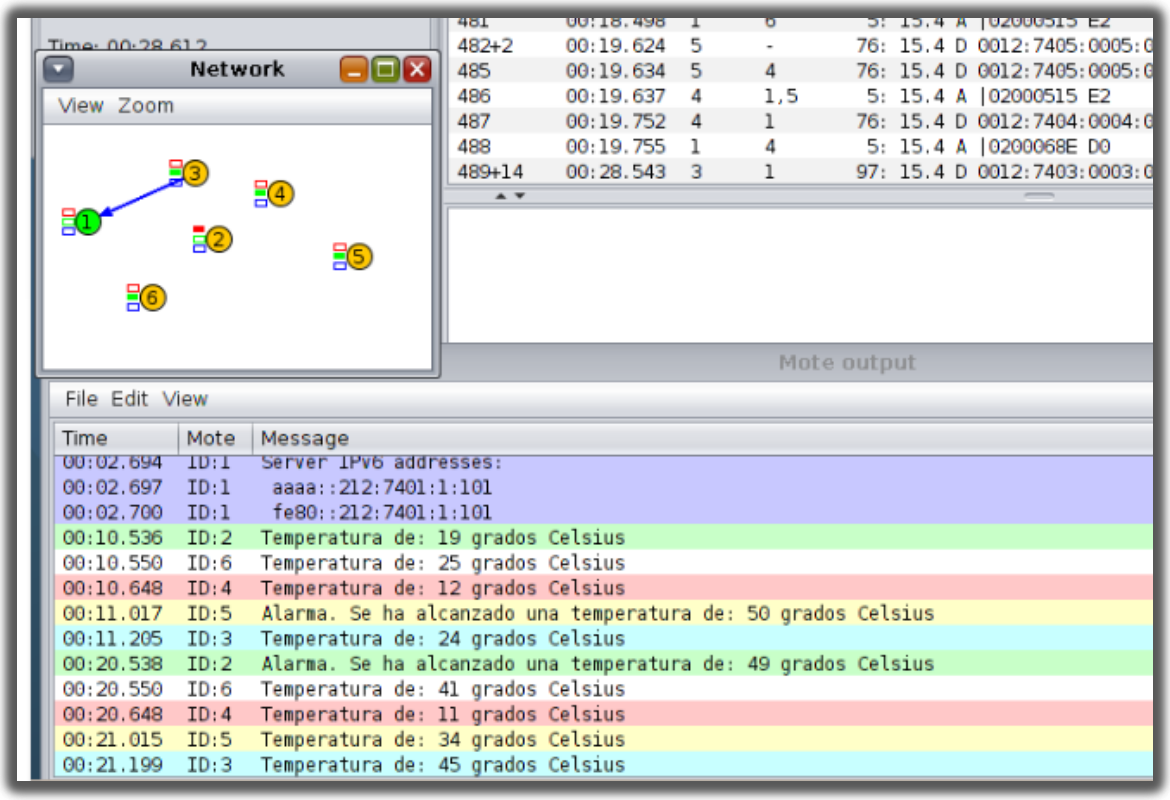

Figura 22. Sortida port sèrie.

L'arxiu log on es troben aquestes mateixes dades es diu logtemp.txt. En la següent figura es pot comprovar que el contingut de la finestra coincideix amb el contingut de l'arxiu. Es pot veure que quan la temperatura ha superat el rang, la sortida mostra un missatge d'alarma.

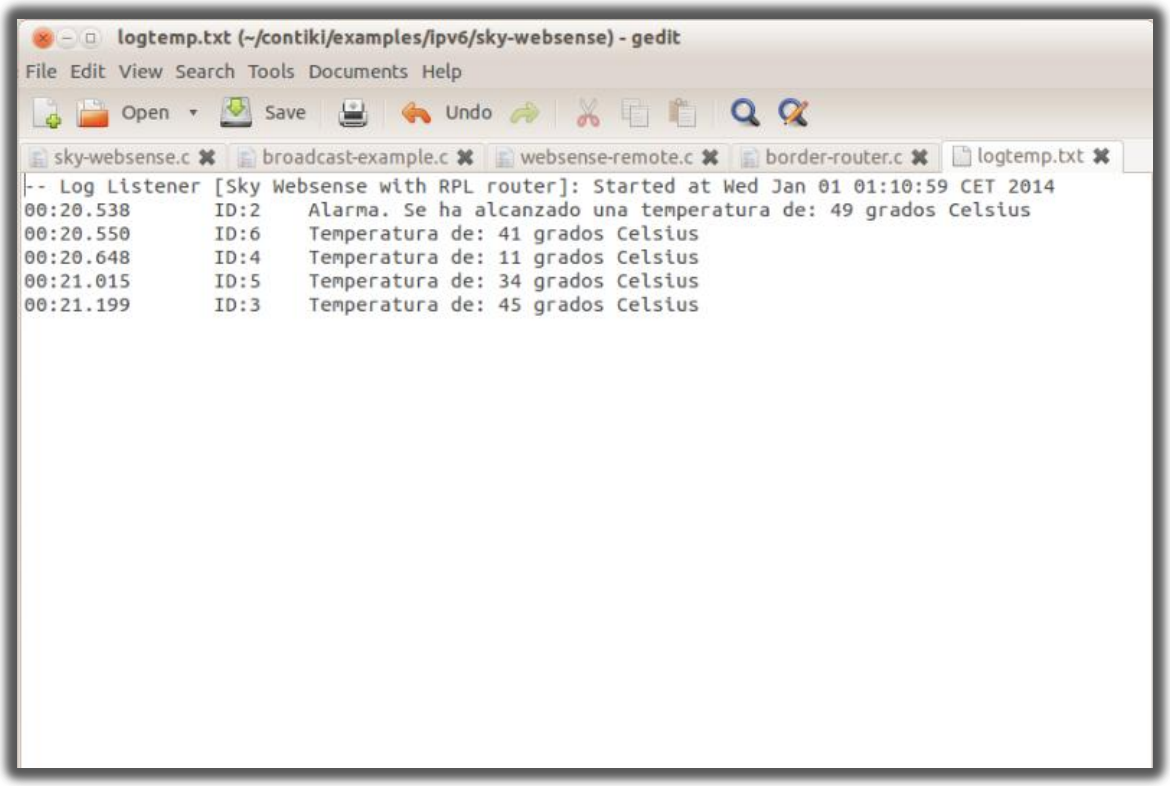

Figura 23. Fitxer log amb les temperatures recopilades.

12. Conclusions

Tot i que no han estat assolits els objectius que em vaig marcar al començament del semestre, he de dir que si que he aconseguit poder simular una xarxa WSN i el seu comportament utilitzant el simulador Cooja de Contiki. EL fet d'haver treballat amb aquest SO condiciona en part perquè és més difícil crear un entorn gràfic per a l'usuari, ja que l'has d'integrar en el simulador.

Penso que Contiki és un excel·lent SO per a realitzar proves amb motes físiques, ja que et permet primer simular la mota virtual i posteriorment provar el seu hardware.

És necessari aprofundir més en l'esquelet de Contiki per a poder entendre millor el seu funcionament. Només d'aquesta manera es pot arribar a desenvolupar aplicacions interessants.

Va ser una errada intentar simular les motes virtuals amb TOSSIM, ja que em va treure bastant més temps del que em pensava.

Les xarxes WSN són amb tota seguretat una tecnologia del futur. En poc temps començarem a treballar parametritzant el nostre entorn i monitoritzant esdeveniments ja siguin ambientals, industrials o constructius.

13. Treball futur

Queda penden el tractament de la recol·lecció de dades que envien els nodes sensors. Amb l'arxiu logtemp.txt es pot fer una recol·lecció de dades i inserir-les en una tabla de una base de dades i poder tenir un control de temperatures molt més acurat.

Es podria fer un historial per nodes, mitjanes de temperatures, comprovació de nodes que tenen lectures de temperatures més altes.

Sistema d'alarma més eficient on l'avís d'alarma no pugui passar desapercebut.

Com a tasca a realitzar es pot considerar la de provar l'aplicació ambo motes reals. Comprovar el comportament que tenen i comparar els resultats amb els resultats teòrics de la simulació.

Desenvolupament de codi per a poder controlar la vida de la bateria del node sensor. Per a poder simular aquesta situació cal tenir en compte el nombre de vegades que el node a enviat informació, el temps comunicant-se per RF (Radio Freqüència), rebent dades, etc.

També és molt necessari per poder portar a terme totes aquestes tasques, realitzar un estudi exhaustiu del funcionament del simulador Cooja.

14. Bibliografia / Web grafia

http://tools.ietf.org/html/rfc5548

Introducción a las redes de sensores
http://www.arcos.inf.uc3m.es/~sescolar/index_files/presentacion/wsn.pdf

http://tools.ietf.org/wg/6lowpan/draft-ietf-6lowpan-nd/

IETF The Internet Engineering Task Force

http://www.ietf.org/

http://www.sics.se/~adam/

Faludi, R. (2010). "Building wireless sensor networks: With ZigBee, XBee, Arduino, and processing". Farnham: O'Reilly Media.

http://arri.uta.edu/acs/networks/WirelessSensorNetChap04.pdf

http://www.mfbarcell.es/conferencias/wsn.pdf

Adam Dunkels. SICSlowpan – Internet-Connectivity for Low-power Radio Systems, 2010

ZigBee Alliance
http://www.zigbee.org/Home.aspx

http://www.cbe.berkeley.edu/research/briefswirelessxyz.htm

IEEE802.15.4
http://standards.ieee.org/getieee802/download/802.15.42006.pdf
http://www.ieee802.org/15/

Wireless Industrial networking alliance
http://wina.org/