



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA

Implementació i simulació d'un model d'encaminament per a xarxes de sensors de radiofreqüència i subaquàtiques

Tesina de Màster en Enginyeria de Computadors
Dpt. d'Informàtica de Sistemes i Computadors

Jose Salvador Climent Bayarri

Dirigida per:
Juan Vicente Capella Hernández

Desembre de 2010

Índex

1	Introducció	5
1.1	Xarxes de sensors sense fil	6
1.2	Requeriments dels algorismes d'encaminament en WSN	6
1.3	Xarxes de sensors subaquàtiques	7
1.4	Objectius de la tesina	7
1.5	Estructura de la tesina	8
2	Treball relacionat	9
2.1	Protocols d'encaminament. Classificació	9
2.2	Transmissió en xarxes de sensors de radiofreqüència	9
2.2.1	La capa física	9
2.2.2	La capa d'accés al medi	10
2.2.3	La capa de xarxa. Protocols d'encaminament	10
2.2.4	Energy-efficient Adaptive Hierarchical and Robust Architecture	11
2.3	Transmissió en xarxes de sensors subaquàtiques	12
2.3.1	La capa física	13
2.3.2	La capa d'accés al medi	13
2.3.3	La capa de xarxa. Protocols d'encaminament	14
3	Simuladors utilitzats	16
3.1	NS-2	16
3.1.1	Arquitectura d'un node	17
3.1.2	Arquitectura d'un <i>mobilenode</i>	18
3.1.3	Modificacions del simulador	19
3.2	NS-3	19
3.2.1	Arquitectura del simulador	20
3.2.2	Modificacions del simulador	21
3.3	Generador de números aleatoris	22
3.4	Justificació de canvi de simulador	22
4	Implementacions d'EDETA	24
4.1	Implementació d'EDETA per al NS-2	24
4.2	Implementació d'EDETA per a entorns subaquàtics en NS-3	26
4.3	Contribucions a l'especificació de l'algorisme	27
4.3.1	Decisió del rol d'un node	28
4.3.2	Llindars d'acceptació de nodes	29
4.3.3	Planificació de l'enviament de les dades	30
5	Experimentació i resultats	32
5.1	Avaluació EDETA en radiofreqüència amb el simulador NS-2	32
5.1.1	Paràmetres generals de les simulacions i escenaris	32
5.1.2	EDETA. Avaluació de prestacions	34
5.1.3	Comparació d'EDETA i LEACH	35
5.2	Avaluació d'EDETA en el medi subaquàtic amb el simulador NS-3	35
5.2.1	Paràmetres generals de les simulacions i escenaris	35

5.2.2 Experiments	36
6 Conclusions, Publicacions i Treball Futur	38
6.1 Conclusions	38
6.2 Publicacions i altres contribucions	38
6.3 Treball futur	39

Capítol 1

Introducció

Les comunicacions sense fil han experimentat una gran expansió en l'última dècada. Les xarxes locals sense fil (WLANs) ofereixen un exemple d'aquest fenomen. L'especificació original de l'estàndard IEEE 802.11 oferí un ample de banda de 2 Mb/s[IEE03]. L'última revisió de l'estàndard disponible, el IEEE 802.11n, ofereix un màxim de 600 Mb/s[IEE09].

Altres xarxes estretament relacionades amb les WLANs són les xarxes sense fil personals (WPANs). Aquestes xarxes foren concebudes per a la intercomunicació de dispositius sense fil personals, com poden ser PDAs o telèfons mòbils. La major diferència entre les WPANs y les WLANs radica en que les primeres foren dissenyades des d'un principi posant atenció al cost i al consum energètic i amb unes majors restriccions pel que fa a la grandària de la xarxa.

Gràcies a estos avanços tecnològics, les xarxes de sensors sense fil han guanyat protagonisme en l'última dècada. Aquestes xarxes estan formades per nodes dotats de sensors, repartits per un àrea determinada i que cooperativament recol·lecten informació del medi, com poden ser temperatura, humitat, nivells de pol·lució, etc.

Adicionalment aquestos nodes tenen algun tipus de mitjà de comunicació sense fil, com pot ser un transceptor de radio, a més d'un microcontrolador i solen estar alimentats per bateries.

Aquesta última característica és clau. Al estar alimentats per bateries l'energia de que disposen els nodes és limitada i per tant, és molt important desenvolupar maquinari i algorismes de comunicació de baix consum.

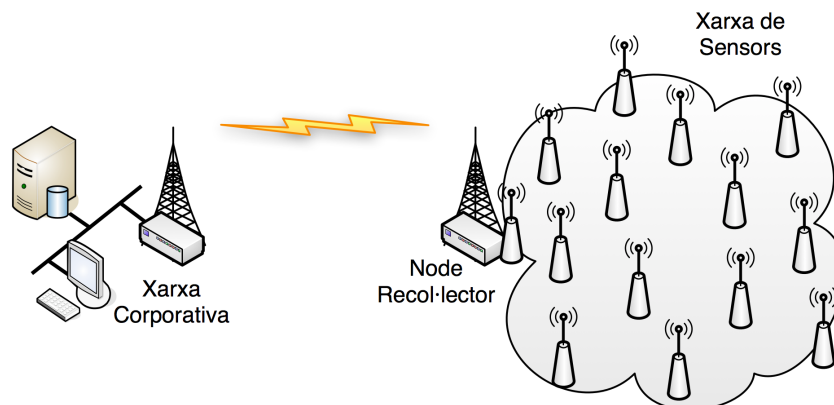


Figura 1.1: Exemple d'una xarxa de sensors sense fil

1.1 Xarxes de sensors sense fil

Considerem l'escenari de la Figura 1.1 on es poden veure una gran quantitat de nodes repartits cobrint una àrea determinada. Aquests nodes s'organitzen automàticament i col·laboren de forma coordinada per a recol·lectar informació del medi. A més, entre ells duen a terme agregació de dades. Es a dir, un node rep les dades del seu veí les processa i agrega a les seues, de forma que la quantitat de dades a enviar és menor.

Les dades van viatjant per la xarxa a través dels diferents nodes fins que arriben al node recol·lector. Aquest node recol·lector típicament fa la funció de pont entre la xarxa de sensors i una xarxa de propòsit més general com pot ser la xarxa local d'una organització o la pròpia Internet.

Com les dades han de donar diferents bots entre els nodes fins a arribar al node recol·lector, és necessari un algorisme d'encaminament que guie estes dades entre el núvol de nodes fins al recol·lector. Les decisions d'aquest algorisme tindran un gran impacte en el balanceig de càrrega, la fiabilitat, els retards i el consum d'energia, entre altres.

1.2 Requeriments dels algorismes d'encaminament per a les xarxes de sensors sense fils

L'encaminament en xarxes de sensors sense fil és diferent del d'altres xarxes sense fil, degut les característiques úniques d'aquest tipus de xarxes[AKK04].

Tràfic asimètric En les xarxes de sensors, les dades solen fluir des dels nodes sensors cap als recol·lectors. Per tant, es crea una asimetria en el tràfic al concentrar l'arreglada de dades als nodes recol·lectors. Poden haver casos en els quals un requeriment siga que la comunicació es done en el sentit invers, des del node recol·lector fins a un o uns nodes concrets, o entre dos nodes qualsevol, però estos casos no són els majoritaris.

Recursos limitats Com els nodes sensors tenen relativament poca capacitat de calcul i emmagatzematge, els algorismes d'encaminament no poden utilitzar operacions que necessiten una alta capacitat de càlcul, així com emmagatzemar grans quantitats de dades.

Temps de vida El temps de vida dels nodes és crucial en aquest tipus de xarxes. Els nodes solen estar alimentats per bateries i s'espera que la xarxa estiga funcionant durant llargs períodes de temps sense haver de fer cap intervenció de manteniment. Un algorisme d'encaminament no sols ha d'intentar mantenir baixa la sobrecarrega de paquets deguda al manteniment de les rutes, també ha de procurar encaminar els paquets per rutes energèticament eficients.

Escalabilitat El número de nodes en una xarxa d'aquest tipus sol ser elevat i varia depenent de l'àrea a cobrir, el factor de replicació i, evidentment, l'aplicació per a la qual s'està dissenyant la xarxa. Per tant, el manteniment de taules de rutes pot ser no és la millor aproximació degut a la gran quantitat de memòria que es podria arribar a consumir i es fa necessari l'encaminament distribuït.

Agregació Hi ha moltes vegades que l'aplicació per a la qual es dissenya la xarxa de sensors permet agregació de dades. En l'agregació de dades la informació es veu reduïda per una funció (com pot ser la suma, el màxim, el mínim, etc.) o concatenant diferents mesures en un únic paquet, reduint la sobrecàrrega.

Tolerància a fallades L'esgotament de les bateries o l'exposició a interferències externes són alguns dels problemes als que estan exposats els nodes. Les xarxes de sensors sense fil han de poder tolerar fallades en nodes puntuals. Evitant que la xarxa caiga per la fallada d'un node. Per a aconseguir açò, l'algorisme d'encaminament ha de proveir de redundància a les rutes de dades.

Temps real Poden haver aplicacions de xarxes de sensors que necessiten temps acotats per a recollir l'informació dels sensors, processar-la i enviar-la.

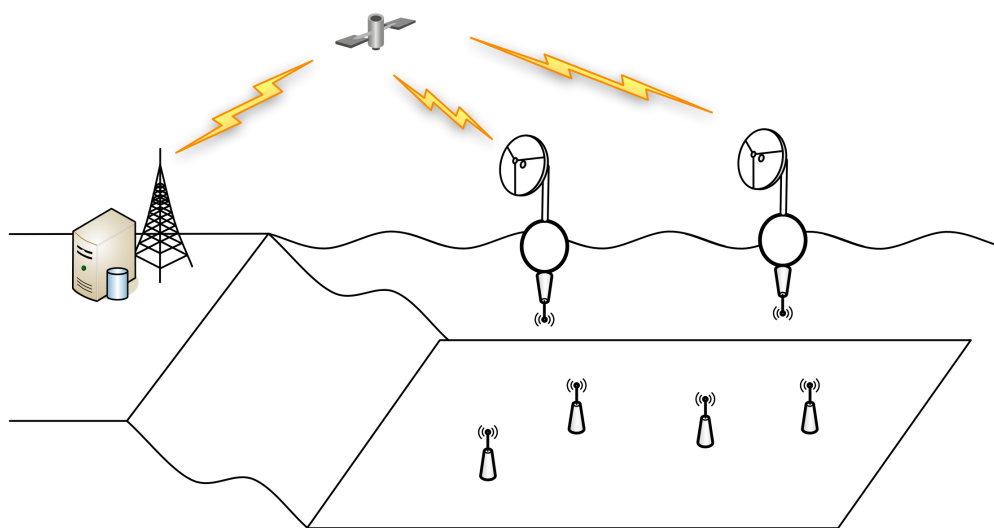


Figura 1.2: Exemple d'una xarxa de sensors sense subaquàtica

1.3 Xarxes de sensors subaquàtiques

Durant l'última dècada s'han fet grans avanços en la investigació en xarxes de sensors terrestres. En els últims anys, la comunitat científica està interessant-se cada vegada més en les xarxes de sensors subaquàtiques, fent-se nombrosos avanços a nivell físic i d'accés al medi.

A diferència de les xarxes de sensors terrestres, on es transmet per mitjà d'ones de radiofreqüència, en aquest tipus de xarxes s'utilitzen les ones acústiques. Actualment, l'ample de banda de que es disposa en aquest tipus de xarxes és molt reduït. Això, unit a altres problemes com pot ser el gran retard de propagació de la senyal, ha originat que també s'hagen d'investigar protocols d'accés al medi adequats per a aquest tipus de xarxes.

Degut als avanços en aquestes àrees, comença a ser factible la utilització d'aquest tipus de xarxes en aplicacions amb desenes o fins i tot centenars de node. Per tant, és necessària la utilització d'algorismes d'encaminament que tinguin en compte els nous reptes que representen les xarxes subaquàtiques.

Hi ha moltes aplicacions en les que es poden utilitzar les xarxes de sensors subaquàtiques. Es poden utilitzar en aplicacions científiques per a monitoritzar un àrea d'interès i recollir les dades; com ocorre en la indústria petrolífera, on és important tindre un registre de l'activitat sísmica en el pous de petroli. O també es poden utilitzar per a implementar sistemes d'alarmes en el cas en que ocorregui un sisme submarí, per exemple.

La Figura 1.2 mostra un exemple d'aquest tipus de xarxes. Es poden veure els nodes repartits pel fons marí i una sèrie de nodes recollidors en la superfície que arrepleguen la informació i l'envien (ací ja per radiofreqüència) a una estació base en terra ferma.

1.4 Objectius de la tesina

Els objectius d'aquesta tesina són per una part la implementació i avaluació d'escenaris d'aplicació d'un algorisme d'encaminament per a xarxes de sensors anomenat EDETA [Cap10][CBOC10] mitjançant simulació amb el NS-2 (Network Simulator) [NS-10b], àmpliament utilitzat en la comunitat científica. Els principals objectius que persegueix aquest protocol són minimitzar el consum d'energia, qüestió fonamental en xarxes de sensors sense fil; sense oblidar l'aportació d'altres característiques cada volta més demandades en les noves aplicacions de les xarxes de sensors sense fil com són escalabilitat, flexibilitat, fiabilitat, seguretat i temps real.

Per altra part, altres objectius importants són l'adaptació del citat protocol al camp de les xarxes de sensors subaquàtiques, sense perdre de vista les característiques de baix consum, escalabilitat i flexibilitat. Així com implementar un nou model per al entorn subaquàtic en el simulador NS-3

[NS-10c] i avaluar la nova proposta mitjançant simulació, re-alimentant la configuració de paràmetres del protocol per a la millora de prestacions i entreveient noves especificacions, requisits i reptes en aquest camp que definiran la línia de la futura tesi doctoral.

1.5 Estructura de la tesina

La resta del document s'organitza de la següent forma:

Al Capítol 2 s'introduirà una classificació d'algorismes d'encaminament per a, seguidament, passar a analitzar l'estat de la investigació de les capes física, accés al medi i de xarxa de la pila de protocols de les xarxes de sensors terrestres i subaquàtiques. L'anàlisi es centrarà sobretot en els algorismes d'encaminament per a ambdós tipus de xarxes.

Seguidament, al Capítol 3 s'introduiran els simuladors utilitzats NS-2 i NS-3. Es comentarà breument l'arquitectura d'ambdós simuladors i les modificacions que necessàriament s'han hagut de dur a terme per a poder realitzar el treball d'aquesta tesina. També es farà una breu descripció del generador de números aleatoris que inclouen els simuladors i per últim, es descriuran els motius per els que es va decidir passar a desenvolupar per al NS-3.

A continuació, al Capítol 4 es descriurà com s'ha dut a terme la implementació d'EDETA en els dos simuladors, explicant les diferències d'implementació més importants. També es detallaran les contribucions a l'algorisme derivades de la implementació i prova de l'especificació original.

En el Capítol 5 es detallen els experiments fets per a validar la implementació d'EDETA, tant en la implementació del NS-2 com en la del NS-3, explicant amb detall els paràmetres utilitzats, així com els models d'energia i de propagació dels dos casos.

Per a finalitzar, al Capítol 6 s'exposen les conclusions de tot el treball realitzat així com les publicacions relacionades amb ell i línies de treball futur de cara a la realització de la tesi doctoral.

Capítol 2

Treball relacionat

2.1 Protocols d'encaminament. Classificació

Els protocols d'encaminament es poden classificar en quatre categories: proactius, reactius, geogràfics i jeràrquics. Els protocols proactius intenten mantenir sempre actualitzada l'informació d'encaminament com per exemple DSDV o OLSR. El problema d'aquests protocols és que creen una gran quantitat de sobrecàrrega cada vegada que la topologia de la xarxa canvia. Això en una xarxa de sensors on aquests poden fallar, les condicions del canal de transmissió són molt variables, etc. origina una sobrecàrrega excessiva.

La aproximació contrària és la dels protocols reactius. Aquests protocols obtenen la informació d'encaminament en el moment exacte en el que es necessita enviar un paquet i no abans. Això fa que la sobrecarrega d'aquests protocols siga molt menor però, per contra, tenen una major latència perquè s'ha d'obtenir la informació de la ruta just abans d'enviar el paquet. Exemples d'aquest tipus de protocols poden ser AODV o DSR.

Els protocols basat en l'encaminament geogràfic, partixen de la base de que el node coneix la seua posició en l'escenari. Això en les xarxes de sensor terrestres és fàcil d'aconseguir gracies als sistemes de posicionament per satèl·lit com el GPS, però en les xarxes subaquàtiques no es pot disposar d'aquesta tecnologia perquè la senyal del GPS es veu altament atenuada per l'aigua. Estos protocols necessiten d'infraestructura addicional per a localitzar els nodes, el que pot encarir el cost de la xarxa. Un bon resum dels mètodes de localització en xarxes subaquàtiques es pot trobar en [CSCE06] i en les terrestres en [MWH02]

L'últim grup és el que més interessa per al propòsit d'aquesta tesina ja que EDETA es classifica dins d'ell. Els algorismes d'encaminament jeràrquic intenten evitar la sobrecarrega de missatges en la xarxa, així com permetre cobrir una major àrea a mesurar, gràcies als agregats o *clusters* de nodes. El que intenten aquests algorismes és realitzar un consum eficient de l'energia transmetent al node líder de l'agregat o *clusterhead* i dur a terme agregació de dades per a reduir el número de paquets que arriben al node recol·lector.

2.2 Transmissió en xarxes de sensors de radiofreqüència

2.2.1 La capa física

Degut a la multitud d'avanços que s'han fet en aquesta capa és possible la transmissió a grans velocitats (sobretot comparant-ho amb les velocitats de les xarxes subaquàtiques com es vorà més endavant). Per exemple una tecnologia estàndard com és IEEE 802.15.4 permet una velocitat de transmissió de 250 kbps [IEE06], un ample de banda més que suficient per a la majoria d'aplicacions d'aquestes xarxes.

En l'actualitat s'estan fent avanços en el que s'anomena la radio cognitiva [ALVM06] i que permet als nodes la adaptació dels paràmetres de transmissió i recepció per tal d'evitar interferències al canal.

2.2.2 La capa d'accés al medi

Hi ha nombrosos articles amb diferents propostes sobre algorismes per a la capa d'accés al medi i ací es van a esmentar alguns dels més bàsics i que ajuden a entendre la problemàtica d'aquest tipus de xarxes. Per a un estudi més complet es recomana consultar [DEA06].

Hi ha propostes basades en CSMA com la de [WC01]. En aquest treball els autors adapten aquest mecanisme d'accés al medi al tipus de trafic més característic d'aquest tipus de xarxes, on solen haver períodes de baixa utilització de la xarxa junt amb altres períodes amb una gran utilització. Extrauen que la utilització de períodes d'escoltar el canal, junt amb retards aleatoris contribueixen a evitar que les col·lisions es produeixin de forma repetitiva.

En [SCI⁺01] els autors proposen un accés híbrid TDMA-FDMA i proposen una fórmula per a decidir quan és millor, des del punt de vista de l'energia, utilitzar un accés amb un major nombre de canals FDMA o per el contrari afavorir més la transmissió mitjançant TDMA.

El protocol SMACS, *Self organization Medium Access Control for Sensor network*, [SGAP00] es un protocol distribuït que permet als nodes descobrir els seus veïns i establir els períodes de transmissió i recepció per a comunicar-se entre ells. El descobriment de nodes veïns i l'assignació de canals es fan alhora, d'aquesta forma, quan els nodes acaben de construir la xarxa ja tenen assignats els canals de transmissió. Els autors defineixen un enllaç de comunicació entre dos nodes com a un parell de ranures de temps operant a una freqüència determinada, es a dir, fan una combinació de TDMA i FDMA.

2.2.3 La capa de xarxa. Protocols d'encaminament

Al igual que en la capa d'accés al medi, hi ha nombrosos protocols d'encaminament per a les xarxes de sensors sense fil, ací s'esmenten alguns dels algorismes jeràrquics i geogràfics més representatius. Un resum exhaustiu amb una extensa classificació d'algorismes d'encaminament es pot trobar en [ACDFP09] i en [AY07] hi ha un altre resum però especialitzat en algorismes basats en clusters.

LEACH [HCB02], *Low Energy Adaptive Clustering Hierarchy*, és un protocol d'encaminament jeràrquic i és, possiblement, el protocol més popular d'aquest tipus. En ell, els nodes decideixen, mitjançant una funció distribuïda, quins van a ser clusterheads. Aquests són els encarregats de recollir la informació dels nodes i l'envien directament al recol·lector. Com els clusterheads tenen una gran sobrecàrrega, l'estructura de la xarxa es trenca i es torna a refer al cap d'un cert temps mitjançant la funció distribuïda anterior, per a equilibrar el consum d'energia dels nodes.

En [LR05] es proposa PEGASIS, *Power-Efficient Gathering in Sensor Information Systems*. Aquest protocol no forma clusters sinó, cadenes de nodes, on cada node transmet i rep dades del seu veí i solament un node de la cadena pot enviar al mateix temps. El protocol fa agregació de dades a cada bot en la cadena fins que les dades apleguen al recol·lector.

TEEN [MA01], *Threshold sensitive Energy Efficient sensor Network protocol* és un protocol jeràrquic dissenyat per a oferir una resposta ràpida als canvis en els paràmetres que estiga mesurant la xarxa de sensors. Els nodes s'organitzen en clusters per nivells. Primerament els nodes s'agrupen en clusters i després els clusterheads d'aquests clusters s'agrupen en clusters de segon nivell fins que s'aplega al recol·lector que és el clusterhead de més nivell. Una vegada s'han format els clusters es distribueixen dos llindars per als paràmetres que es volen estudiar. Quan un node detecta que el valor del paràmetre que està mesurant sobrepassa el llindar dur envia la informació cap al recol·lector i, a partir d'ací sols tornarà a enviar un nou canvi si la mesura canvia tant com marca el llindar bla.

Com a exemple de protocols geogràfics es té MECN, *Minimum Energy Communication Network* [RM02]. Configura i manté una xarxa de sensors utilitzant la posició per a encaminar els missatges de forma energèticament eficient. El protocol identifica una regió d'encaminament per a cada node. Aquesta regió compren els nodes en els que és més eficient enviar el missatge a través d'ells que de forma directa al node destí. Així MECN intenta trobar una sub-xarxa que tinga el menor número de nodes i que requereixi la menor energia per a la transmissió entre dos nodes.

Un altre exemple de protocols geogràfics és GAF, *Geographic Adaptive Fidelity* [XHE01]. Aquest protocol conserva l'energia ficant en estat de baix consum els nodes que no són necessaris per a l'encaminament dels paquets. Crea una graella virtual de l'àrea a cobrir i cada node utilitza la seua posició per a associar-se al punt de la graella que tinga més pròxim. Tots els nodes associats a un mateix punt es consideren equivalents a efectes d'encaminament de paquets, així sols cal que un node

dels assignats a un punt estiga despert per a encaminar els paquets. Els nodes van passant-se el torn d'encaminadors per a distribuir el consum d'energia.

2.2.4 Energy-efficient aDaptive hiErarchical and robuT Architecture

EDETA [Cap10][CBOC10] és un protocol d'encaminament jerarquitzat. EDETA aprofita el fet de que la major part del trafic va dirigit al node recol·lector i, encara que permet la comunicació entre dos nodes qualsevol de la xarxa, la comunicació amb el recol·lector és la més optimitzada, perquè és la més freqüent. El protocol és energèticament eficient i capaç d'adaptar-se als canvis de la xarxa. Amb tot, s'aconsegueix un protocol d'encaminament per a xarxes de sensors sense fil robust.

Es dedica un apartat a aquest algorisme d'encaminament perquè, com s'ha dit abans, tot el treball d'aquesta tesina es centra en aquest protocol i per tant, és imprescindible entrar a comentar-lo més a fons.

EDETA és jerarquitzat perquè està basat en dos nivells, el primer format per clusters i el segon format per un arbre dinàmic. L'algorisme selecciona els clusters mitjançant una funció distribuïda i torna a refer l'estructura de la xarxa cada cert número de rondes.

La funció distribuïda decideix si un node va a ser clusterhead basant-se en el número de nodes de la xarxa, el número aproximant de clusterheads que volem que hi haja i l'energia restant dels altres nodes de la xarxa.

Els clusters es comuniquen amb el node recol·lector mitjançant una estructura d'arbre, on el node recol·lector és l'arrel de l'arbre, cada clusterhead és un dels vèrtex de l'arbre i els nodes associats al clusterhead són les fulles.

A més, el protocol suporta més d'un node recol·lector per a permetre major escalabilitat i ser més tolerant a fallades.

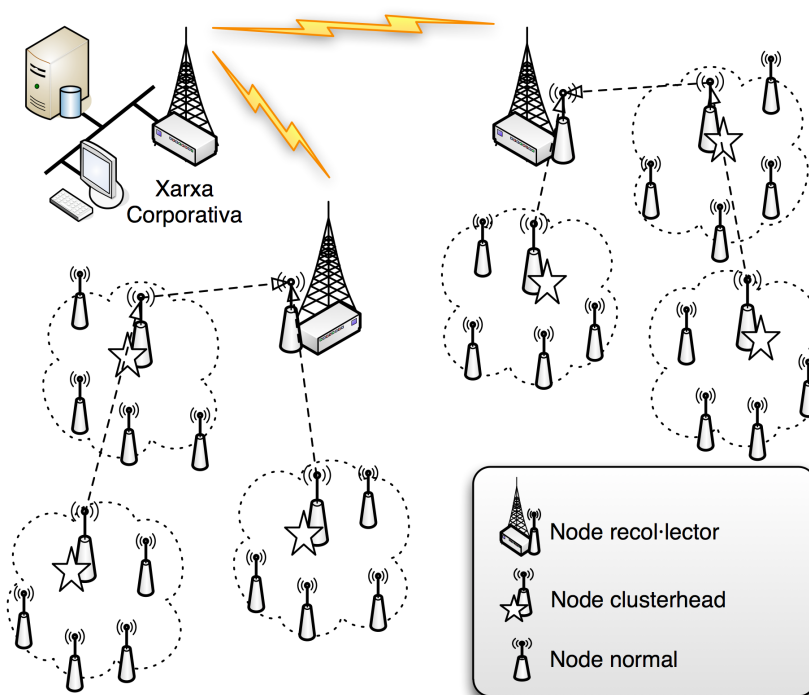


Figura 2.1: Exemple d'una xarxa EDETA

Un exemple del tipus de xarxa format per EDETA es pot veure a la Figura 2.1 on tenim una xarxa formada per dos nodes recol·lectors, cada un arrel del seu arbre. Els clusters estan marcats amb un símbol puntejat i els clusterheads es diferencien dels nodes normals perquè tenen una estrella. Els nodes normals s'han unit als clusterheads més propers i estos s'han organitzat en forma d'arbre per a enviar

les dades al node recol·lector més pròxim. Per últim, els nodes recol·lectors envien la informació que reben a un servidor central per al posterior anàlisi de les dades.

L'algorisme té dos fases: la fase d'inicialització i la de funcionament. En la primera es configura la xarxa i en la segona es recol·lecten les dades dels sensors.

En la fase d'inicialització els nodes decideixen quin va a ser el seu rol (node normal o clusterhead). A continuació, difonen el seu rol a la xarxa i els nodes normals s'uneixen al cluster que tinguen més pròxim. Alhora, els clusterheads aprenen, gràcies a les difusions dels altres clusterheads, quins són els seus veïns i formen l'arbre. Seguidament, els clusterheads planifiquen, de forma distribuïda, els temps d'enviament de les dades. Una vegada fet tot açò, l'algorisme està preparat per a passar a la següent fase.

En la fase de funcionament, els clusterheads recol·lecten les dades dels nodes integrants del seu cluster i les envien al seu pare en l'arbre, de tal forma que al final les dades arriben al node recol·lector. Una vegada un node en qüestió ha enviat les seues dades entra en un mode de baix consum fins a la següent ronda.

Per últim, després d'un cert número de rondes que es pot configurar, es trenca tota l'estructura i el procés de formació de la xarxa comença de nou. Així s'intenta repartir i balancejar el consum energètic entre tots els nodes de la xarxa.

EDETA-e

EDETA-e és un subconjunt d'EDETA que permet al dissenyador del sistema obtenir més control sobre la formació de la xarxa i els retards.

Amb EDETA-e és el dissenyador del sistema, i no el protocol, el que decideix quin node va a ser clusterhead. Així, s'ha de decidir on va a situar els clusterheads i de quin tipus de font d'energia disposaran. Com en EDETA-e els clusters estan fixes, sols cal executar una vegada la fase d'inicialització.

2.3 Transmissió en xarxes de sensors subaquàtiques

Existeixen diferents tecnologies que permeten la comunicació dels sensors en les xarxes subaquàtiques. Per una part tenim les ones de radio convencionals però, el seu major problema resideix en que la seua atenuació en el medi subaquàtic és molt gran i, encara que hi ha diferents estudis que demostren que és viable utilitzar-les per a comunicacions de molt poca distància [FRD06], no es solen utilitzar degut a que requereixen antenes molt grans i una gran potència de transmissió. Per altra part, també existeix la possibilitat d'utilitzar comunicacions òptiques perquè no pateixen la gran atenuació de les ones de ràdio i tenen un ample de banda molt més gran però, requereixen visió directa i aigua molt clara, a més, sofreixen altres problemes com la dispersió.

En l'actualitat la forma de transmissió més utilitzada per a la comunicació entre sensors subaquàtics es mitjançant les ones acústiques. Però aquesta forma de transmissió no està lliure de problemes. Per una part tenim que l'atenuació del senyal varia depenent de la distància i la freqüència de transmissió [Sto07]. A més la propagació del senyal també varia amb la distància i pot ser esfèrica o cilíndrica depenent de si la transmissió és a curta o llarga distància.

L'atenuació del senyal també varia amb la profunditat. A major profunditat menor atenuació. Es més, en aigües poc profundes¹, existeix el fenomen conegut com a *multipath*, en el que l'ona de transmissió es propaga per diferents camins; el que comporta una gran degradació de la senyal i genera interferència entre els símbols transmesos (ISI, de l'angles *InterSymbol Interference*). Aquest fenomen també depèn de l'orientació de la propagació del senyal. En un canal de transmissió en vertical, aquest fenomen pràcticament no existeix, en canvi en els canals horitzontals és un problema molt gran.

A més, les comunicacions acústiques són molt sensibles al soroll ambient, sobretot en aigües poc profundes, el soroll provinent dels vaixells o simplement de les ones de la superfície poden interferir notablement en les comunicacions.

Un altre problema a esmentar és la velocitat de transmissió del senyal. Les ones de radio es propaguen en l'aire a quasi la velocitat de la llum, en canvi, les ones acústiques es propaguen a

¹La bibliografia es refereix a aigües poc profundes quan tenen menys de 100 m. de profunditat i a aigües profundes quan tenen més [APM04]

1500 m/s en l'aigua. Això són 5 ordres de magnitud menys, el que comporta uns grans retards de propagació.

Un altra diferència a tindre en conte està en el cost energètic de les comunicacions. En les xarxes de comunicació terrestres, el cost energètic de la recepció és més elevat que el de la transmissió. En les xarxes acústiques passa totalment el contrari, el cost energètic ve dominat per la transmissió[FGS⁺06].

Tots aquestos problemes no existien amb les xarxes de sensors terrestres i això ha fet que s'hagen tornat a dissenyar totes les capes de la pila de protocols per a permetre una comunicació eficient entre els sensors subaquàtics. A continuació es va a fer un xicotet resum de l'estat de la investigació en les capes física, d'accés al medi i de xarxa.

2.3.1 La capa física

En la capa física, s'han estat investigant quines modulacions són les més adients per a la transmissió en el medi subaquàtic. Modulacions senzilles com FSK s'utilitzen degut a la seua fàcil implementació i eficiència energètica però, per contra, tenen un aprofitament molt baix del ample de banda del canal, entre 0.5 i 1 bits per Hz. Recentment, s'ha vingut estudiant la utilització de modulacions més complexes com OFDM i combinant-les amb MIMO s'ha aconseguit la transmissió a 12 Kbps en aigües de poca profunditat [CSS08].

Com s'explicarà més endavant al capítol 5 el model de la capa física que utilitza el simulador NS-3 per a xarxes de sensors subaquàtiques és l'implementat per el FuNLab a la Universitat de Washington[tea10]. Aquest model correspon al WHOI micro-modem, desenvolupat al Woods Hole Oceanographic Institution[FGS⁺06]

2.3.2 La capa d'accés al medi

En la capa d'accés al medi (MAC) també s'han fet nombrosos estudis. Les MAC es poden dividir en dos grans grups: els que són lliures de contenció i els que no ho són. Les que són lliures de contenció inclouen TDMA, FDMA i CDMA, on els usuaris accedeixen al medi de forma separada per temps, freqüència o codi respectivament. Per altra part, entre les que estan basades en contenció podem trobar ALOHA, MACA i CSMA.

De entre les que són lliures de contenció, FDMA no es aplicable degut al poc ample de banda de que es disposa (de l'ordre d'uns pocs kHz). Per altra part, utilitzar TDMA en una xarxa amb centenars de nodes pot resultar impracticable degut a que és un protocol centralitzat (i per tant resulta difícil la seua escalabilitat) i necessita d'una molt bona sincronització entre els nodes. CDMA és una de les millors alternatives en este grup. Una de les dificultats que comporta l'aplicació de CDMA ve donada per la reducció de l'ample de banda efectiu deguda a la utilització de codis de transmissió. Un altra ve deguda al problema conegut com *near-far problem*. Consisteix en que la senyal d'un node es pot vore emmascarada quan arriba al receptor si un altre node, prop del receptor, està emetent al mateix temps; encara que siga amb un altre codi de transmissió.

De entre les que no són lliures de contenció, ALOHA s'utilitza en comunicacions amb satèl·lits on els retards també són molt elevats. ALOHA pur té l'avantatge de que és fàcil d'implementar però la utilització del canal no és molt elevada i sols aplega al 18%. En [CSC07] els autors proposen unes variacions a ALOHA per a xarxes subaquàtiques que milloren el seu rendiment. Per altra part, MACA està basat en el mecanisme RTS/CTS. Amb això s'aconsegueixen evitar els problemes del node exposat i ocult. Per últim, CSMA es basa en escoltar primer el medi de transmissió; si el canal està lliure transmetre i sinó executar el mecanisme de *backoff*. El major desavantatge d'estos dos últims protocols ve determinat per el retard de propagació, que fa que un node haja d'escoltar el canal durant molt de temps abans de començar a transmetre.

En [XC06] els autors estudien el comportament de l'accés aleatori i del mecanisme RTS/CTS per a xarxes de sensors subaquàtiques. Apleguen a la conclusió de que l'accés aleatori i RTS/CTS tenen el mateix rendiment en xarxes amb molt poc de tràfic i baixa densitat de nodes però, quan la densitat és alta RTS/CTS es comporta millor que l'accés aleatori. Un altre factor que han estudiat és la distancia de transmissió, quan la distancia de transmissió augmenta, el rendiment de RTS/CTS disminueix. Per contra, l'accés aleatori no és veu afectat. Per últim, també analitzen la grandària del paquet i apleguen a la conclusió de que en xarxes amb grandàries de paquets grans RTS/CTS es comporta

millor que l'accés aleatori. Per contra, l'accés aleatori es comporta millor quan la grandària de paquet es xicoteta.

2.3.3 La capa de xarxa. Protocols d'encaminament

Hi ha una gran quantitat de protocols d'encaminament proposats per a xarxes de sensors terrestres. Però degut a la gran diferència en el mitjà de transmissió entre les xarxes terrestres i les subaquàtiques es fa necessari el desenvolupament de nous protocols per a aquest tipus de xarxes.

En [Ald08] l'autor estudia el comportament de DSDV, AODV i DSR en el medi subaquàtic. Per a fer-ho, a utilitzat el simulador NS-2 modificant els paràmetres de atenuació, propagació i freqüència de transmissió de la capa física per a intentar adaptar-lo al nou medi de transmissió i com a mètode d'accés al medi ha utilitzat CSMA/CA. Dels resultats obtinguts conclou que AODV és el que obté un major rendiment en quant a transmissió de paquets i retràs extrem a extrem. D'aquest estudi cal assenyalar que les simulacions no s'han realitzat utilitzant un model físic del medi de transmissió subaquàtic, sinó que s'han modificat alguns dels paràmetres del model sense fils del simulador NS-2. Per això, és possible que els resultats de les simulacions no siguin totalment acurats.

En [TTB94] els autors proposen un algorisme d'encaminament basat en taules. Cada node manté una taula amb informació d'encaminament sobre tots els nodes de la xarxa. Aquesta informació s'obté enviant paquets de control als nodes veïns cada vegada que hi ha un canvi a la xarxa. Quan un node vol enviar un paquet a un altre node, primer crea un circuit virtual i una vegada està establert aquest circuit envia el o els paquets. Aquest algorisme d'encaminament pot ser útil quan es requereix enviar molta informació de dades entre dos nodes de la mateixa xarxa; al crear el circuit virtual, els autors s'asseguren de que hi ha ample de banda i recursos reservats per a enviar les dades. Com a contrapartida es té que l'algorisme no és gens escalable perquè el manteniment de les taules de rutes quan hi ha centenars de nodes pot ser impracticable. A més, si la informació a enviar no és molt elevada, el sobrecost d'establir el circuit virtual pot ser molt gran.

En [XG02] els autors proposen un algorisme d'encaminament que intenta mitigar l'efecte que l'elevat retard de propagació té en el rendiment de la xarxa. Per a dur-ho a terme, proposen un algorisme d'encaminament centralitzat en el que un node, anomenat node principal, és l'encarregat d'establir els camins i reservar l'ample de banda necessari per a la transmissió de paquets. Per a aconseguir mitigar l'efecte del retard de propagació, proposen la utilització del protocol de la finestra lliscant per al control de congestió però, amb la particularitat de que els paquets de reconeixement han d'anar per un canal diferent als paquets de dades. Els autors diuen a l'article que el número de canals necessari per a una xarxa de 16 nodes és de 7, en canvi, al passar a una xarxa de 60 nodes aquest número s'incrementa fins a 12. Per tant, el número de canals necessaris és el que més limita l'escalabilitat d'aquest protocol.

Els autors de [FAC⁺05] proposen la utilització d'una versió de AODV modificada anomenada AODV-BI, junt amb el protocol d'accés al medi MACAW. Fan dos modificacions a AODV, una d'elles consisteix en que el node destí aprofita la informació del paquet Route-Request per a establir un camí al node font en el cas en que necessite contestar-li a les dades. En el AODV original, si el node destí ha de contestar al node font deu de tornar a sol·licitar el descobriment de ruta. I la segona modificació consisteix en que els nodes emmagatzemen la informació de les rutes sol·licitades i sols l'esborren quan al enviar un paquet per una ruta ja coneguda es genera un error d'encaminament. En el protocol original les rutes no s'emmagatzemen. Els resultats mostren que AODV-BI es comporta millor que AODV en quant a termes de latència i pèrdua de paquets.

En [XCL06] els autors presenten l'algorisme VBF (*Vector-Based Forwarding Protocol for Underwater Sensor Networks*). Cada paquet porta la posició de l'emissor, el receptor i el node que reenvia el missatge. Introdueixen el concepte de camí de reenviament que ve especificat pel vector que uneix l'emissor i el receptor. Si un node determina que està suficientment prop del camí de reenviament afegeix la seua posició al paquet i el reenvia. A més, proposen un algorisme d'adaptació automàtica que permet a cada node determinar si ha de reenviar el paquet basant-se en la densitat del nodes de la seua zona. Els autors proposen que cada node base el calcul de la seua posició a partir del seus nodes veïns, sense cap tipus d'infraestructura addicional però, no aprofundeixen en com dur a terme aquesta localització.

QUELAR [HF10] és un algorisme d'encaminament adaptatiu basat en l'aprenentatge. Quan un node necessita transmetre dades, afegeix a més informació sobre el seu estat. Cada vegada que un

node rep un paquet, encara que no siga per a ell, llegeix la informació afegida per l'emissor i actualitza el seu estat i la seua funció d'encaminament. El autors comparen el seu rendiment amb l'algorisme VBF i conclouen que obté el mateix rendiment en termes de paquets correctament enviats i eficiència en l'encaminament però, amb major eficiència energètica.

En [PMA06] els autors proposen un algorisme d'encaminament centralitzat i basat en la localització. El node gestor de la xarxa calcula la topologia i els camins de dades i l'encaminament es du a terme mitjançant circuits virtuals. A més, proposen un mètode per a que els nodes pugen reparar els camins de dades en el cas en que un dels nodes caiga. El major problema d'aquest protocol ve donat per l'algorisme que executa el node central per a calcular les rutes perquè és NP-complet. Els autors asseguren que és computacionalment factible trobar una solució per a xarxes de fins a 100 nodes.

Capítol 3

Simuladors utilitzats

En aquest capítol es va a fer una introducció als simuladors NS-2 i NS-3. No es pretén replicar la documentació existent d'ambdós simuladors i per tant, sols es descriuran breument la seua arquitectura i les modificacions que se'ls ha hagut de fer per a implementar EDETA.

En ambdós programes la simulació està guiada per esdeveniments. El simulador manté una llista d'esdeveniments ordenada cronològicament i en el seu bucle principal la recorre i va processant els esdeveniments en ordre. Un exemple es pot veure en la Figura 3.1. Hi ha tres esdeveniments A, B i C i el temps entre els tres no és necessari que siga el mateix.

En una simulació d'aquest tipus, no cal que tots els esdeveniments de la simulació es creen a l'inici. Conforme avança la simulació, un esdeveniment pot crear un o més esdeveniments, sempre amb un temps major o igual que el de l'esdeveniment que està processant-se. Els nous esdeveniments s'afegeixen a la llista d'esdeveniments respectant l'ordre cronològic.

La simulació acabarà quan no queden més esdeveniments que processar o quan s'aplegue a un estat prèviament configurat, com pot ser un temps de simulació màxim.

La resta del capítol està estructurat de la següent forma: Primerament es descriu l'arquitectura del simulador NS-2 i les modificacions que se li han fet. A continuació es farà el mateix amb el simulador NS-3. Seguidament es descriurà breument el generador de números aleatoris d'ambdós simuladors i per a acabar, es justificarà el perquè d'un canvi de simulador (del NS-2 al NS-3) per a adaptar EDETA al medi de transmissió subaquàtic.

3.1 NS-2

NS-2 (per les seues sigles en angles Network Simulator 2)[NS-10b] és un simulador d'esdeveniments discrets utilitzat en la investigació de xarxes de comunicació. A la Figura 3.2 es pot veure la seua arquitectura bàsica. Consta d'un executable que rep com a paràmetre d'entrada el nom d'un fitxer amb ordres per al simulador i a l'eixida s'obté un fitxer amb les traces de la simulació.

El simulador, utilitza el paradigma orientat a objectes i està desenvolupat en dos llenguatges: C++ i OTcl (Object-oriented Tool Command Language). C++ s'encarrega de definir el mecanisme intern dels objectes de la simulació i OTcl configura la simulació connectant i configurant els objectes, així com programant els esdeveniments. C++ i OTcl interactuen entre ells utilitzant TelCL. Cada objecte de C++ accessible des de OTcl té la seua corresponent variable en aquest últim llenguatge i l'anomenen *handler*. Un handler és sols una cadena de caràcters en OTcl (per exemple `_o25`) i, en principi, no

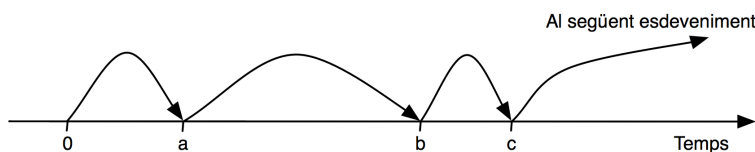


Figura 3.1: Com avança el relloctge en una simulació guiada per esdeveniments

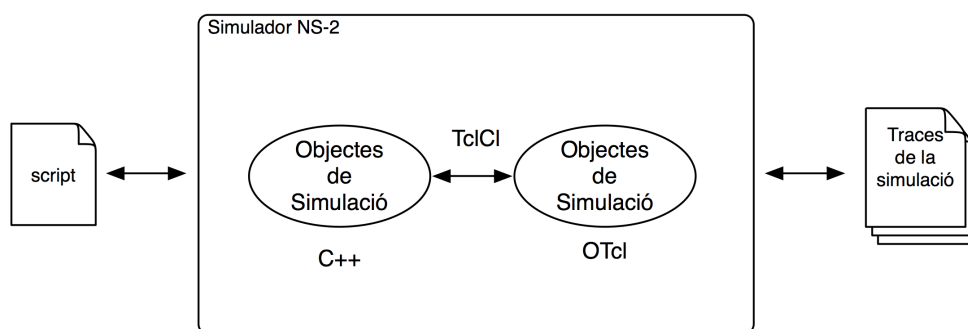


Figura 3.2: Arquitectura bàsica del simulador NS-2

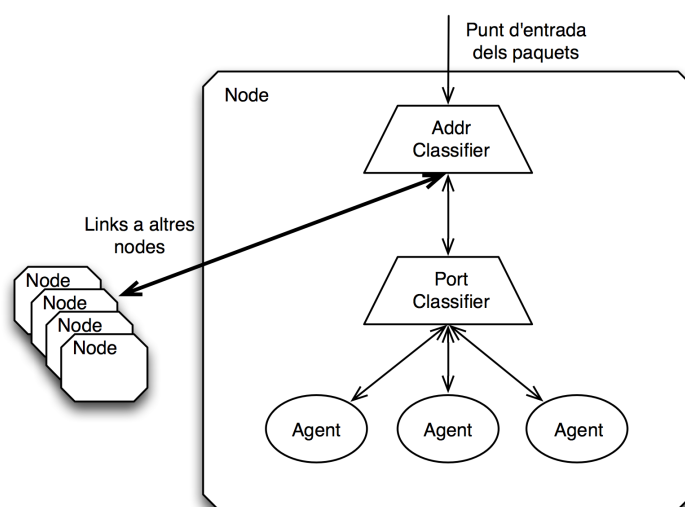


Figura 3.3: Diagrama d'un objecte node en NS-2

conté ninguna funcionalitat. La funcionalitat ve definida en l'objecte C++ associat a aquest handler. S'ha dit que en principi no conté cap funcionalitat perquè l'ideal és implementar-la en C++ perquè, OTcl és un llenguatge interpretat i molt més lent que el primer. Però OTcl també permet la definició de les seues pròpies classes, mètodes i variables.

3.1.1 Arquitectura d'un node

Aquest objecte està dissenyat per a xarxes cablejades però és necessari descriure'l ja que l'objecte *mobilenode* que deriva d'aquest es el que s'utilitza en la simulació de xarxes sense fils.

Un node en NS2 fa dos papers. Com a encaminador, re-encamina els paquets basant-se en la seua taula d'encaminament. Com a node, envia els paquets al protocol de transport especificat en la capçalera dels paquets.

Com es pot veure a la Figura 3.3, un node és un objecte compost. Els paquets primerament apleguen al objecte AddrClassifier. Si el paquet no és per a aquest node el re-encaminarà cap al seu destí basant-se en la taula d'encaminament. Sinó, enviarà el paquet a un altre objecte, anomenat PortClassifier, encarregat d'enviar el paquet al protocol de nivell superior (objectes de tipus Agent en NS2) especificat per el port del paquet.

Com s'ha dit al començament, aquestos nodes no estan dissenyats per a la simulació de xarxes sense fil i per això és van introduir els *mobilenode*.

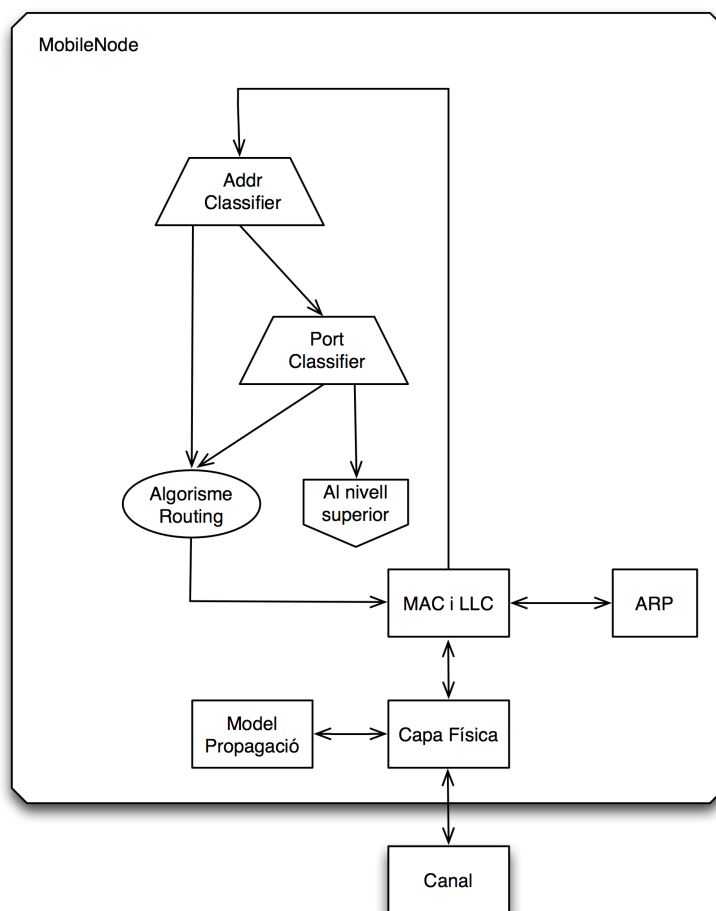


Figura 3.4: Diagrama d'un objecte MobileNode en NS-2

3.1.2 Arquitectura d'un *mobilenode*

L'objecte *mobilenode* deriva de l'objecte Node, és per tant, un objecte node amb funcionalitat afegida per a la comunicació sense fils i per a tenir la capacitat de moure's en la topologia.

La figura 3.4 mostra de forma simplificada els components d'aquesta classe. La pila de xarxa per al *mobilenode* consisteix en una capa d'enllaç, un mòdul ARP, una cua de prioritats, una MAC i una interfície de xarxa connectada a un canal. Tots estos components estan connectats entre ells amb OTcl.

La capa d'enllaç és la que s'encarrega de resoldre les direccions de xarxa a direccions MAC per als paquets enviats i envia els paquets rebuts als classificador per a que els envie al seu corresponent agent.

El mòdul ARP resol les peticions que li fa la capa d'enllaç i fa les consultes ARP pertinents.

La cola de prioritats s'utilitza com a buffer per a enviar els paquets i permet prioritzar i filtrar l'enviament de certs paquets.

La MAC té la funcionalitat típica, rep els paquets de la capa d'enllaç i els envia a la interfície de xarxa. Actualment hi ha diferents implementacions de MACs per al NS-2 i entre elles estan 802.11 i 802.15.4.

La interfície de xarxa la utilitza el *mobilenode* per a accedir al canal. Aquest element afegeix dades com la potencia de transmissió i la longitud d'ona per a que el node receptor pugui determinar si el paquet li ha aplegat, si ha ocorregut una col·lisió, etc.

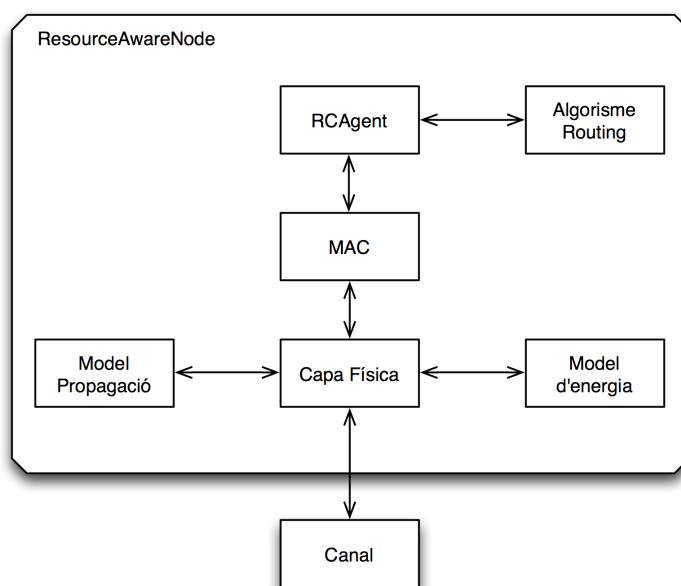


Figura 3.5: Diagrama d'un objecte ResourceAwareNode en NS-2

3.1.3 Modificacions del simulador

EDETA requereix una capa d'accés al medi senzilla, que simplement permeta enviar i rebre quan siga necessari i que permeta posar al node en mode de baix consum. Les MACs que hi ha disponibles en la distribució estàndard del NS-2 són massa complexes i suposarien una sobrecarrega important per a l'algorisme d'encaminament. Per exemple, els estàndards 802.11 i 802.15.4 són els que decideixen quan un node pot entrar en mode de baix consum o quan pot transmetre. EDETA requereix un control total d'aquestes funcionalitats i per això es fa necessari la utilització d'un altre tipus de capa d'accés al medi.

Es va optar per la utilització de les modificacions que varen fer els autors de LEACH per a implementar el seu protocol al simulador. Com ja s'ha dit abans, LEACH és un algorisme d'encaminament en xarxes de sensors jeràrquic [HCB02].

En les modificacions han definit un nou node al que han anomenat *ResourceAwareNode*. Com es pot veure en la Figura 3.5, en aquest node han introduït un model d'energia implementat per ells i han canviat l'entrega dels paquets. En un *mobilenode* els paquets rebuts s'entreguen a un classificador que s'encarrega d'entrega-los al protocol de nivell superior que escau. En aquest node el que s'ha fet és que s'entregue directament els paquet al objecte RCAgent, encarregat del nivell de xarxa.

Aquesta classe no realitza ninguna acció. Sols té implementats els mètodes *sendmsg* i *recv* que s'encarreguen d'enviar els paquets que es reben cap a l'algorisme d'encaminament i d'enviar els paquets que l'algorisme d'encaminament li entrega.

També han introduït uns canvis a la classe *wirelessPhy* per a que actualitze el nivell d'energia del nou node cada vegada que s'executa un canvi d'estat (transmissió, recepció, espera o baix consum).

A més a més, han implementat una nova MAC molt simple que s'encarrega de rebre i enviar les trames. Abans d'enviar una trama la MAC escolta el medi per a assegurar-se de que no hi ha cap altre node emetent i en la recepció de paquets és l'encarregada de descartar els paquets que han sofert alguna col·lisió.

3.2 NS-3

Al igual que el seu predecessor, el NS-3 és un simulador d'esdeveniments discrets per a xarxes de computadors[NS-10c].

Un dels objectius fonamentals del disseny del NS-3 ha sigut millorar el realisme dels models. El llenguatge en el que està implementat és C++ i triaren aquest llenguatge en part per a facilitar la inclusió de codi ja implementat en C. A més, NS-3 també facilita la possibilitat d'integrar-se en un banc de proves amb dispositius i aplicacions reals.

Un altre dels objectius ha sigut facilitar la depuració de codi i la utilització de llenguatges més moderns que el seu predecessor. Aquest objectiu ha fet que els autors del simulador evitaren la barreja de OTcl i C++, que és difícil de depurar i a més Tcl és un llenguatge antiquat i en desús. En lloc d'això, els models del NS-3 s'implementen en C++ per a millorar el rendiment i facilitar la depuració i consta d'un API basada en Python per a la implementació de scripts. Així, els usuaris poden escriure les simulacions amb programes fets en C++ o amb scripts en Python.

L'últim dels objectius principals ha sigut el manteniment de codi. El simulador NS-2 accepta models sense cap tipus d'estàndard d'escriptura de codi, amb poques proves de validació i amb cap consideració per el disseny del programari. Aquesta política ha originat que el NS-2 creixera ràpidament en número de models disponibles, el que està molt bé, però ha fet que la tasca de manteniment del simulador siga molt difícil. Al NS-3 s'ha intentat resoldre aquestos problemes amb un estàndard d'escriptura de codi més rigorós i processos de revisió de codi abans de la seua inclusió al simulador.

Degut al seu disseny, el NS-2 té alguns inconvenients que els autors del NS-3 estan intentant evitar. Un dels inconvenients del NS-2 és la dependència entre models. Si un model requereix de altres per a funcionar pot ser necessari editar el codi del model requerit i això pot originar que altres models deixen de funcionar. Per exemple, En les modificacions que s'han comentat abans sobre el simulador NS-2, s'ha dit que s'ha definit un nou objecte node anomenat *ResourceAwareNode*. Això implica que no es puguin utilitzar els algorismes d'encaminament DSDV o AODV perquè necessiten un objecte *MobileNode*. Per a evitar aquest problema, en NS-3 s'ha introduït un model d'agregació d'objectes. Els objectes es poden agregar a altres objectes en temps d'execució i existeix un mecanisme per a saber si un objecte d'un tipus determinat està agregat dins d'un altre objecte. Així un objecte de tipus node no és més que un objecte que agrega a altres objectes que poden ser un model d'energia, una pila de protocols, etc.

Un altre inconvenient de NS-2 està en el sistema de traces perquè és necessari editar fitxers del nucli del simulador per a poder variar l'eixida que es mostra en el fitxer de traça. En NS-3 el sistema de traces és molt flexible. Introdueixen dos objectes: El recol·lector de traces i l'origen de les traces. Un recol·lector de traces pot recollir traces de diferents objectes d'origen i a l'inrevés, un objecte origen de traces pot estar unit a més d'un recol·lector. A més, un recol·lector pot escriure la seua eixida a diferents fitxers, com pot ser un fitxer ASCII o pcap. I tot açò configurable des del script de simulació, sense editar res del codi del nucli del simulador.

La manca d'un sistema de gestió de memòria és un altre desavantatge. Si bé aquest problema ve degut a la utilització d'un llenguatge com C++ que no incorpora mecanismes de gestió de memòria com pot ser el *Garbage Collector* de Java, els autors de NS-3 l'han solucionat introduint els *Smart Pointers*. El concepte d'arrere dels Smart Pointers és el mateix que el dels *shared objects* de les llibreries Boost. Simplement duen un conte de les referències a un objecte i quan aquest conte aplega a zero s'allibera la memòria.

Per últim, NS-3 incorpora un sistema de *logging* molt flexible, estructurat en nivells (debug, warn, error, etc) i en tipus d'objecte a depurar, que facilita el tractament de la eixida del programa a l'hora de depurar.

Un estudi complet de totes les característiques del simulador es pot trobar al seu manual [np10a] i al seu tutorial[np10b].

3.2.1 Arquitectura del simulador

El simulador NS-3 està organitzat en mòduls, com es mostra a la Figura 3.6. El nucli del simulador és el que ofereix les facilitats per al programador que s'han esmentat abans, com els *Smart Pointers*, el sistema d'agregació d'objectes o el sistema de traces. Però també ofereix altres funcionalitats que no s'han esmentat com els objectes *Callback*, atributs d'objectes en temps d'execució, entre altres [np10a].

El mòdul "common" conté els tipus de dades relacionats amb la manipulació dels paquets i les seues capçaleres i el mòdul "simulator" conté les funcions per al tractament del temps i els diferents planificadors d'esdeveniments. El modul "node" conté la definició de la classe Node esmentada anteriorment.

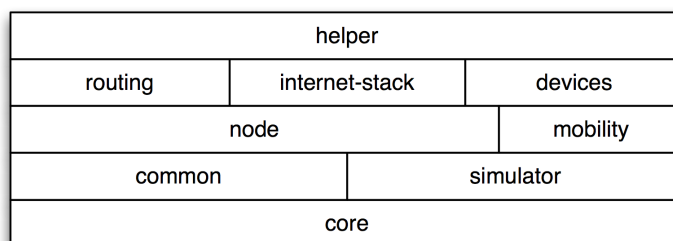


Figura 3.6: Diagrama dels mòduls del simulador NS-3

orment i altres classes bàsiques com “NetDevice” que esmentarem més endavant. El modul ‘mobility’ conté els models de mobilitat dels nodes, es poden destacar alguns com el model estàtic, el de velocitat constant, direcció aleatòria, etc.

Altres mòduls que també conté el simulador són el modul “routing” on es troben els models dels algorismes d’encaminament (OSLR, AODV, etc), el mòdul “internet-stack” on estan implementats els protocols TCP, UDP, IPv4 i IPv6. El mòdul “devices” consta dels models dels diferents dispositius que permeten l’accés dels nodes a la xarxa, com són els models de 802.11, un model de CSMA paregut a Ethernet o UAN, el model subaquàtic.

Per últim, el mòdul “helper” conté una serie de classes que defineixen un API de més alt nivell i facilita la programació de scripts als usuaris.

Passem ara a vore amb més detall algunes de les classes que componen els mòduls anteriors. Com ja s’ha dit abans, un node en NS-3 no és més que un objecte que agrega altres objectes. Segons el tutorial del simulador[[np10b](#)], un node és l’abstracció bàsica d’un computador i ofereix mètodes per a manipular-lo. Proposen que es pense en un node com en un ordinador al que se li pot afegir o llevar funcionalitats. Per exemple, a un ordinador se li poden afegir aplicacions o targetes d’expansió i això mateix se li pot fer a un node del simulador.

Una aplicació en el simulador no és com una aplicació del mon real, sinó que és un objecte que simula el tipus de tràfic que poden generar les aplicacions reals. Per exemple, la classe UdpEchoServerApplication, que deriva de la classe Application, genera un trafic UDP acorde als paràmetres proporcionats per l’usuari.

La classe Channel és l’encarregada de modelar el canal de transmissió. Com a exemple de tipus de canals es té la classe WifiChannel o, el que s’ha utilitzat per a la realització d’aquesta tesina UanChannel.

La classe NetDevice és l’encarregada de l’abstracció del maquinari de comunicació (per exemple una targeta Ethernet) i els seus drivers. Les classes que s’encarreguen de l’abstracció dels nivells superiors es comuniquen amb aquesta classe per a enviar i rebre paquets. Per “baix” de la classe NetDevice queden les classes que abstraen el canal (Channel), la capa física (Phy) i la d’enllaç (UanMac, WifiMac, etc.).

3.2.2 Modificacions del simulador

El codi del simulador consta de dos rames, la rama estable i la de desenvolupament. Com que la rama estable manca de funcionalitats per a poder implementar el protocol, com es la integració entre el model subaquàtic i el model d’energia, s’ha optat per utilitzar la rama de desenvolupament.

A pesar d’utilitzar aquesta rama, no totes les funcionalitats que es necessitaven per a la implementació d’EETA estaven presents. A la Figura 3.7 hi ha un diagrama de classes simplificat amb els mètodes que s’han modificat de la implementació original. Per una part, la integració entre el model subaquàtic i el model d’energia si que estava però no es disposava de mètodes per a posar el node en un estat de baix consum i es varen haver d’implementar.

A més, EETA necessita que les capes inferiors l’informen del nivell de senyal en que escolta als altres nodes per a prendre decisions i això també es va haver d’implementar.

Un altra modificació que es va haver de fer va ser afegir la possibilitat d’utilitzar diferents canals per a transmetre i rebre perquè el model original no ho suportava. Al dur a terme aquesta extensió,

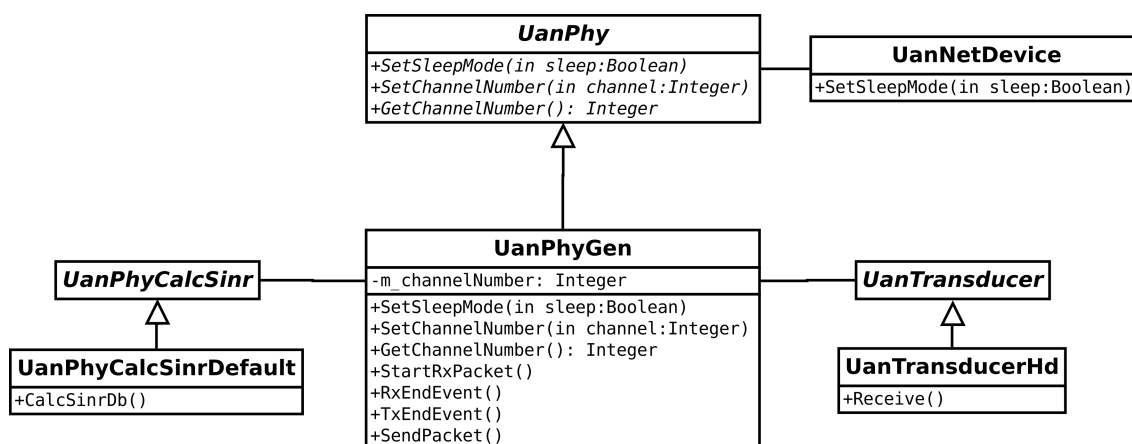


Figura 3.7: Diagrama de classes simplificat dels canvis fets al codi del NS-3

també es va haver de modificar el codi del tractament de les col·lisions i les interferències entre les diferents trames.

Aquestes modificacions s'han enviat als autors del model per a que les incloguen si ho estimen oportú. A més, com a resultat de les proves realitzades al model, també s'han enviat dos informes d'errors amb els seus corresponents pegats i ja han sigut acceptats i inclosos al simulador.

3.3 Generador de números aleatoris

Conèixer com es duu a terme la generació de números aleatoris en els simuladors és molt important per a poder assegurar que els resultats de les diferents simulacions són independents[UR08].

Tant el simulador NS-3 com NS-2 (aquest últim des de la versió 2.1b9), utilitzen el generador de números aleatoris MRG32k3a[LSC02] que té un període de 3.1×10^{57} números aleatoris. Com aquesta seqüència de números aleatoris és molt gran, es divideix en 1.8×10^{19} streams de números independents i cada un d'aquests streams és dividit a més en 2.3×10^{15} substreams. Cada substream té un període de 7.6×10^{22} números.

En NS-2, cada objecte RNG instanciat correspon a un substream i per tant, a pesar de que els objectes RNG tenen mètodes per a assignar-los una llavor, no es deuen utilitzar perquè no es pot garantir que dos objectes RNG als que se'ls ha proporcionat ambdós llavors produeixin seqüències de números aleatoris totalment independents.

En NS-3, cada objecte RandomVariable utilitza un stream diferent del generador de números aleatoris. Al igual que en el simulador NS-2, si es vol proporcionar una llavor s'ha de proporcionar al generador de números aleatoris i no a cada un dels streams o substreams, per a assegurar que les seqüències de números aleatoris són totalment independents. A més, com que l'única forma rigorosa de dur a terme replicacions independents d'una simulació és utilitzar substreams diferents per a cada simulació, els autors del simulador introdueixen el concepte de número d'execució (*run number* en la documentació en anglès). Per tant, al configurar la simulació s'ha d'especificar el número d'execució i a partir d'ell el simulador seleccionarà els substreams per a cada RandomVariable.

3.4 Justificació de canvi de simulador

NS-3 aporta moltes millores a la simulació de xarxes de computador i xarxes sense fil respecte a NS-2. Les més rellevants s'han esmentat a l'apartat 3.2 i la resta es poden trobar a la seua pàgina web [NS-10c].

L'argument de pes que ha propiciat el canvi de simulador ha sigut la disponibilitat d'un model de transmissió subaquàtic. En NS-2 existeix un model del medi subaquàtic[NS-10a] que utilitza un conjunt de llibreries que s'afegeixen al simulador anomenades NS-Miracle. Tant aquest conjunt de llibreries com el model de transmissió per al medi subaquàtic no continuen mantenint-se per això, la utilització d'aquesta plataforma no pareix molt recomanable.

A més, existeix una forta dependència, donada per l'arquitectura del simulador, entre el model implementat d'EDETA en NS-2 i els models de les capes inferiors. La utilització de les llibreries NS-Miracle i del model subaquàtic anaven a forçar la reescriptura d'una part important del codi ja disponible. A això hi ha que afegir-li el fet de que hi hauria que tornar a familiaritzar-se amb el codi de les capes inferiors i la seua API.

Per la seua part, NS-3 disposa d'un model de transmissió subaquàtic inclòs en la seua distribució estàndard i el desenvolupament del model continua, afegint-li noves funcionalitats, així com corregint possibles errors.

Per tot açò, es va decidir que els beneficis de continuar amb el simulador NS-2 eren mínims degut a la necessitat de reescriure gran part del codi d'EDETA així com el fet d'haver de tornar a familiaritzar-se amb codi i APIs noves.

Capítol 4

Implementacions d'EDETA

EDETA és una màquina d'estats [Cap10][CBOC10]. L'algorisme canvia d'estat quan es donen certes condicions. La implementació en ambdós simuladors és molt pareguda, ja que en ambdós s'ha modelat com el que és, una màquina d'estats.

Per a dur a terme la implementació del protocol en els simuladors, s'han seguit les recomanacions de [Sto08], realitzant una implementació incremental, partint d'un model simple i anant afegint-li funcionalitat.

La resta del capítol està estructurat de la següent forma. Primerament, s'explicarà la implementació d'EDETA en el simulador NS-2 i a continuació, es descriurà la implementació en el simulador NS-3. Bàsicament ambdós implementacions són iguals però, al tornar a implementar EDETA des de zero per al NS-3, s'han fet alguns canvis en la implementació derivats de l'experiència obtinguda amb NS-2. Finalment, es descriuran les aportacions que s'han fet a l'especificació del protocol.

4.1 Implementació d'EDETA per al NS-2

Com s'ha esmentat en l'apartat 3.1.3 EDETA s'ha implementat utilitzant el model de node que els autors de LEACH varen publicar junt amb el seu algorisme.

Aquest model de node resulta més apropiat per al propòsit d'aquesta tesina perquè permet un control total per part de l'algorisme d'encaminament de la major part dels paràmetres de les capes física i d'enllaç, com poden ser el control dels modes de consum o el canvi de la direcció MAC del propi node.

La Figura 4.1 mostra un diagrama de classes simplificat de la implementació d'EDETA en NS-2. Com es pot observar, la classe `Edeta` deriva de la classe `Application` i no de la classe `Agent` com li correspondria a un protocol d'encaminament. Els autors del model de node varen decidir crear un objecte `RCAgent` genèric que simplement enviara els paquets a la capa superior (en aquest cas la de aplicació). En aquesta capa va ser on realment implementaren el seu protocol d'encaminament, anomenat LEACH, en OTcl. Al estar implementat en OTcl, les dades i paràmetres de capçalera dels paquets els varen modelar amb una llista de OTcl i per això la classe `RCAgent` espera un `String`, les llistes en OTcl són cadenes amb un cert format en C++. Per a facilitar el tractament d'aquestes cadenes s'ha hagut d'implementar una classe auxiliar que permet inserir i extraure informació mantenint el format de OTcl.

La classe `Edeta` conté un vector del tipus `EdetaState`. Cada un dels seus elements és un punter a una classe que representa un estat del protocol (`RoleElection`, `ChAdv`, `ListenAdv`, `Schedule`, `Polling`, `Sense`, `Dead`) i manté l'estat actual guardant l'índex que ocupa en el vector en l'atribut `currentState`. Implementa també el mètode `ChangeState` que permet la transició entre estats i espera com a paràmetre l'índex del nou estat.

Al llistat de codi 4.1 es pot veure la implementació d'aquest mètode. Rep com a paràmetre el nou estat al que es vol canviar i simplement s'actualitza la variable `currentState` al nou valor i es crida al mètode `initialize` del nou estat per a que comence la seua execució.

La classe `EdetaState` és la classe pare de les diferents classes que representen els estats del protocol. És una classe abstracta, ja que declara la majoria dels mètodes virtuals que deuen implementar les

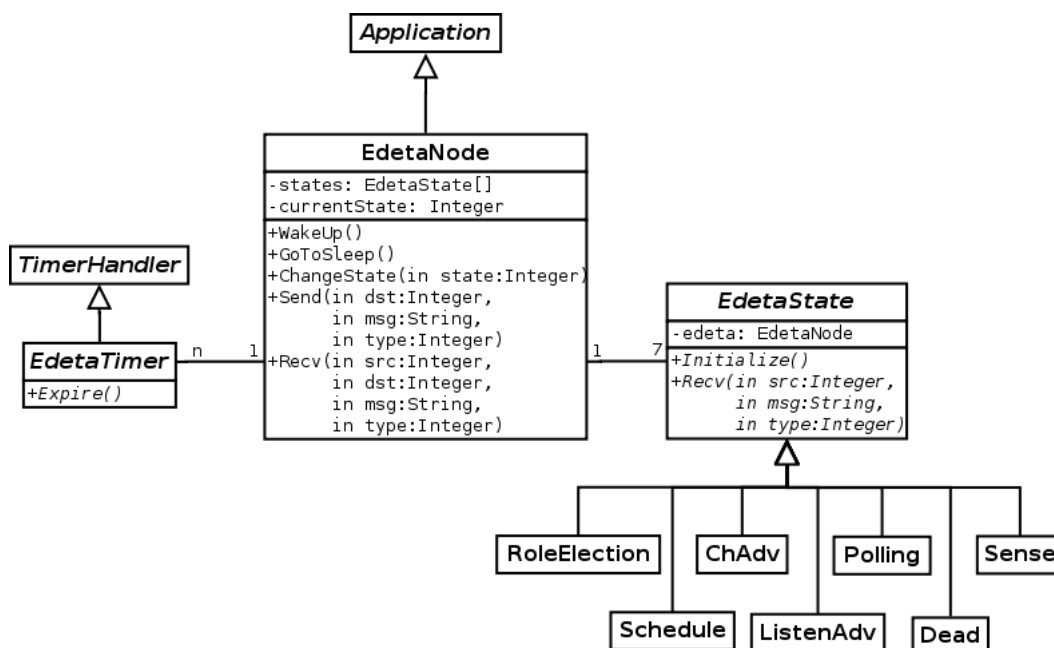


Figura 4.1: Diagrama de classes simplificat de la implementació d'EDETA en NS-2

Llistat de codi 4.1: Implementació del mètode changeState (NS-2)

```

1 void Edeta::changeState(int state)
2 {
3     currentState = state;
4     states[currentState]->initialize ();
5 }

```

Llistat de codi 4.2: Implementació d'EdetaTimer_Ack (NS-2)

```

1 class EdetaTimer_Ack : public EdetaTimer {
2 public:
3     EdetaTimer_Ack(EdetaState*** edetaState, int* currentState)
4     {
5         edetaState_ = edetaState;
6         currentState_ = currentState;
7     }
8     void expire(Event* e)
9     {
10        (*edetaState_)[*currentState_]->On_timer_Ack();
11    }
12 };

```

classes més especialitzades, les que realment implementen els estats del protocol. Conté també un punter a un objecte de tipus `Edeta` per a permetre l'accés als estats del protocol als mètodes `Send()`, `Recv()`, `WakeUp()` i `GoToSleep()`, entre altres. A més, implementa els mètodes virtuals per al tractament de les interrupcions provocades per els timers.

La classe `EdetaTimer` és la que representa els timers. La classe `EdetaNode` es la que instància tots els objectes d'aquest tipus que van a necessitar posteriorment els diferents objectes que representen els estats de l'algorisme. Això es va fer així per a permetre que un estat pugues configurar fàcilment timers per a un altre estat.

Per a dur a terme la implementació dels timers s'ha estès la classe `TimeHandler`, proporcionada pel simulador. Més concretament, el constructor de la classe `EdetaTimer` espera un punter al vector d'estats de la seua classe `Edeta` i un altre a l'atribut `currentState`. Això permet que, des del mètode `Expire`, que s'executa quan venç el temps d'un timer, es pugui cridar, de forma eficient, al mètode de l'estat actual del node encarregat del tractament de la interrupció d'aquest timer.

A títol il·lustratiu, al llistat de codi 4.2 es proporciona la definició i implementació de la classe `EdetaTimer_Ack`. En les línies 3-7 es defineix el constructor de la classe que espera els dos paràmetres esmentats abans. En les línies 8-11 és pot veure com és la implementació del mètode `expire` i com es crida al mètode adequat de l'estat en que està actualment el protocol.

Per últim, per al càlcul de números aleatoris s'utilitza l'objecte global `Random` del simulador NS-2, que cridant als seus mètodes retorna un número aleatori dins de la distribució especificada.

4.2 Implementació d'EDETA per a entorns subaquàtics en NS-3

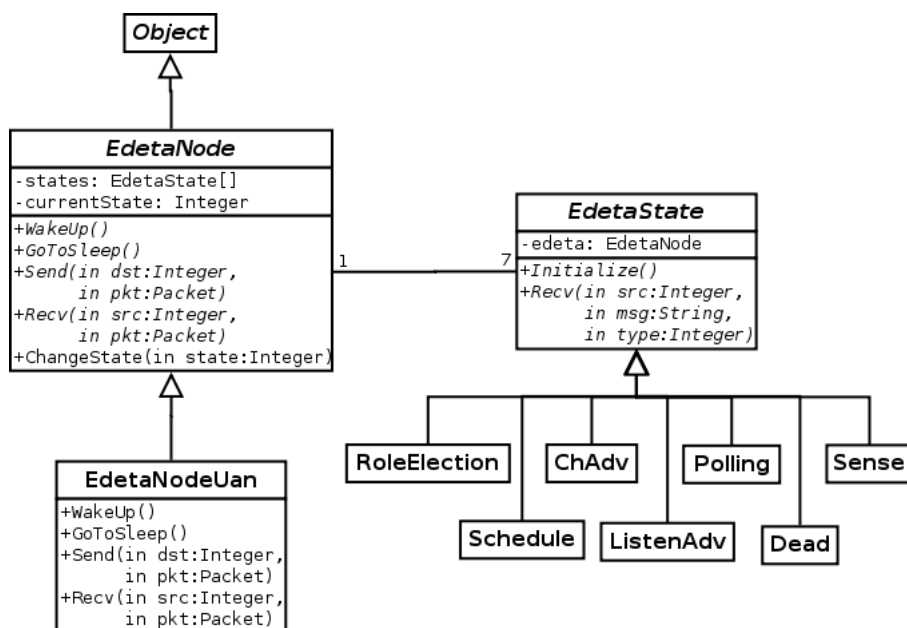


Figura 4.2: Diagrama de classes simplificat de la implementació d'EDETA en NS-3

La implementació d'EDETA per al simulador NS-3 és molt pareguda a la del simulador NS-2. Si més no, hi ha alguns canvis propiciats per la diferència entre les APIs dels simuladors i també deguts a l'experiència obtinguda en la implementació del l'algorisme en NS-2.

Hi ha quatre diferències fonamentals entre ambdós implementacions. La primera ve donada pel desig de fer la implementació del protocol el més independent possible de les capes inferiors en la pila de protocols. La implementació d'EDETA en el simulador NS-2 està molt lligada a les capes inferiors. Això està propiciat, en part, per la falta d'independència entre models que hi ha en aquest simulador.

Per això, com es pot veure a la Figura 4.2, en la nova implementació del model feta per al NS-3, s'ha afegit un nivell més d'herència per al objecte `EdetaNode`. Així, la classe `EdetaNode` és una classe

Llistat de codi 4.3: Utilització d'un timer (NS-3)

```

1  class EdetaNode : public Object
2  {
3      public:
4          Timer timer;
5
6          EdetaNode::Timer_expire()
7          {
8              //Accions a dur a terme quan el timer expira
9          }
10
11         EdetaNode::EdetaNode()
12         {
13             timer.SetFunction (&EdetaNode::Timer_expire, this);
14         }
15     }

```

abstracta que declara els atributs i mètodes que necessita la implementació de l'algorisme i la classe **EdetaNodeUan** és en la que s'implementen els mètodes declarats en la classe pare i encarregats de la comunicació amb les capes inferiors. Per tant, la classe **EdetaNodeUan** és l'encarregada de la interacció entre la implementació del model subaquàtic i la implementació del model d'encaminament.

Un altra diferència ve donada per la forma en que s'utilitzen els timers. El mètode `expire` de la classe **TimerHandler** del simulador NS-2 sempre s'ha de definir en cada implementació d'un nou timer (Llistat de codi 4.2, línies 8-11). Això força la necessitat de definir i implementar codi nou per a cada timer nou que es vulga utilitzar. En NS-3 la utilització dels timers és més senzilla.

Al llistat 4.3 hi ha un exemple d'utilització d'un objecte **Timer** en el simulador NS-3. En l'exemple es defineix una classe **EdetaNode** que inclou la definició del seu constructor (línies 11-14), la definició d'un mètode anomenat **Timer_expire** (línies 6-9) i la declaració d'un objecte de tipus **Timer** anomenat **timer** (línia 4). Al nou timer se li indica a quina funció ha de cridar utilitzant el mètode **SetFunction**, al que se li passa per paràmetre un punter al mètode que s'ha d'encarregar del tractament de la interrupció i un punter a l'objecte en el que es produirà la interrupció. D'aquesta forma s'evita que l'usuari del simulador haja de definir una classe nova per a cada timer diferent que necessite utilitzar.

La següent diferència d'implementació també té a veure amb els timers. Com s'ha dit en l'apartat anterior, en la implementació per al simulador NS-2 la classe **EdetaNode** es la que instància tots els objectes del tipus **EdetaTimer** que van a necessitar posteriorment els diferents objectes que representen els estats de l'algorisme. S'ha comprovat que aquest no és un bon disseny degut a que dificulta la identificació de quins timers utilitza un estat determinat.

Per això, en la nova implementació del model per al NS-3 s'ha optat per una aproximació totalment diferent. Cada estat es defineix i es configura els seus propis timers. Solament hi ha una excepció, el timer que s'encarrega de marcar el canvi entre estats, anomenat **timerSuperFrame** en la implementació NS-3, continua estant definit en la classe **EdetaNode**

Per a finalitzar, l'última diferència té a veure amb la generació dels números aleatoris. En lloc d'utilitzar un objecte global per a generar els números aleatoris, cada objecte **EdetaNode** té un objecte del tipus **RandomVariable**. Així, cada node té el seu propi generador de números aleatoris que, segons s'ha explicat a l'apartat 3.3, correspon a un substream (determinat per el número d'execució) dins d'un stream del generador de números aleatoris del simulador. Així és pot assegurar més rigorosament que els diferents experiments són totalment independents.

4.3 Contribucions a l'especificació de l'algorisme

L'especificació de l'algorisme s'ha fet d'una forma iterativa. Partint d'una primera especificació, s'han anat implementant i resolguen contratemps no previstos. Per això, el treball d'aquesta tesina no ha sigut solament d'implementació i prova d'un protocol, sinó també la proposició d'idees per a intentar millorar el seu funcionament.

4.3.1 Decisió del rol d'un node

A l'apartat 2.2.4 on s'explica el funcionament del protocol, es diu que un node, en l'estat d'iniciació, decideix quin va a ser el seu rol, node clusterhead o node normal. Hi ha dos aproximacions per a dur a terme aquesta decisió, prendre la decisió de forma centralitzada o distribuïda.

La decisió centralitzada té un gran avantatge, el node central encarregat de decidir el rol de tots els altres nodes conta amb la informació de tota la xarxa i pot prendre la decisió més òptima per a cada cas. A més, també permet fer la planificació dels temps d'enviament de dades per a tota la xarxa. Però té dos grans desavantatges:

El primer d'ells és com enviar la informació de tots els nodes al node central. En aquest punt no hi ha una estructura de xarxa construïda, un node qualsevol no sap com enviar les dades al node central i aquest últim tampoc té definides les rutes per a enviar els resultats dels seus càlculs. Seria necessària la utilització d'un altre algorisme d'encaminament per a aquesta tasca o utilitzar una tècnica d'inundació de missatges amb la sobrecàrrega que això suposa.

El segon desavantatge ve donat per els recursos de computació dels que necessitaria disposar el node central. Per a una xarxa de centenars o milers de nodes la tasca de decidir el rol de cada node pot resultar molt costosa computacionalment.

Per això, la decisió del rol en EDETA és distribuïda i cada node amb les dades de que disposa decideix el seu rol. Es va decidir utilitzar la mateixa fórmula que els autors de LEACH utilitzen per a l'elecció distribuïda del rol d'un node però, afegint també a l'equació l'energia de que disposen els restants nodes de la xarxa.

En LEACH un node decideix quin és el seu rol triant un número aleatori entre 0 i 1. Si aquest número és menor que el número calculat per l'expressió 4.1, aleshores el node passa a ser clusterhead.

$$T(n) = \begin{cases} \frac{P}{1 - P \times (r \bmod \frac{1}{P})} & \text{si } n \in G \\ 0 & \text{d'altra forma} \end{cases} \quad (4.1)$$

On $T(n)$ és el valor del llindar per a un node n , P és la proporció (en tant per u) de clusterheads que es vol en la xarxa, r és la ronda actual i G és el conjunt dels nodes de la xarxa que no han sigut clusterheads en les últimes $\frac{1}{P}$ rondes. D'aquesta forma, cada node serà clusterhead una ronda de les $\frac{1}{P}$ rondes. Com que cada vegada hi ha menys nodes elegibles per a ser clusterhead, la probabilitat de convertir-se en clusterhead ha de augmentar amb el número de ronda. I això s'aconsegueix en el denominador de l'expressió, que va disminuint el seu valor fins a ser igual que el numerador en l'última ronda; forçant als últims nodes que no han sigut clusterheads a ser-ho. Després de les $\frac{1}{P}$ rondes, tots els nodes tornen a ser elegibles per a convertir-se en clusterheads.

La forma en que els nodes en EDETA seleccionen el seu rol es basa en el mateix principi que LEACH però han de seguir dos passos. En el primer pas, els nodes calculen també un número aleatori entre 0 i 1 i si és menor que el número donat per l'expressió 4.2 aleshores poden passar al segon pas.

$$T(n) = \frac{c}{|N| - 2c} \times \alpha, n \in N \quad (4.2)$$

On $T(n)$ és el valor llindar per a un node n , c és el número de clustersheads que es vol a la xarxa, N és el conjunt de nodes de la xarxa i α es un paràmetre que dependrà de quan s'ha computat l'equació. Si l'equació es computa al inici de la fase de configuració, $\alpha = 1$. Si per el contrari es computa com a conseqüència de la demanda de més clusterheads es va incrementant el seu valor per a incrementar la probabilitat de que un node siga clusterhead¹.

Aquesta expressió el que intenta és donar un llindar basant-se en el número de nodes normals que són candidats a passar a ser clusterheads. En cada nova ronda hi haurà $|N| - 2c$ nodes candidats a ser clusterheads. El 2 de la expressió ve perquè es va estimar, basant-se en el seu consum d'energia, que un node que ha sigut clusterhead podrà tornar a ser-ho al cap de dos rondes sense ser-ho.

Per tant, si el número aleatori que calculen és major que el valor que retorna l'expressió 4.2 el node passa automàticament a ser un node normal. Si per el contrari el número calculat és menor, es continua amb el segon pas.

¹Aquest mecanisme de demanda de més clusterheads l'incorpora EDETA per a evitar la formació d'illes en la xarxa[CBOC10][Cap10]

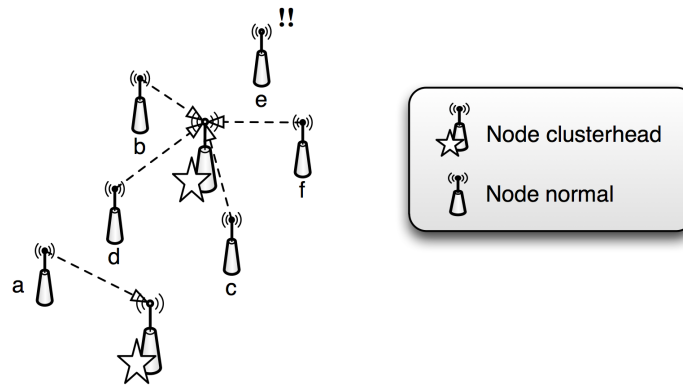


Figura 4.3: Exemple il·lustratiu del problema dels llindars

En el segon pas, els nodes comparen la seua energia restant amb l'energia que tenien els clusterheads de la macro-ronda anterior a l'inici de la fase de configuració². Aquesta comparació la fan mitjançant l'expressió 4.3

$$E(n) = E_T \frac{2T_{Config}}{2T_{Config} + MAX_{Rounds} \times T_{SuperFrame}}, n \in N \quad (4.3)$$

On E_T és la mitja de l'energia dels clusterheads que hi havia al voltant del node, N és el conjunt de nodes de la xarxa i T_{Config} , MAX_{Rounds} i $T_{SuperFrame}$ són tres paràmetres de configuració del protocol que s'expliquen breument a continuació:

T_{Config} Aquest paràmetre especifica la mitat del temps de la fase d'inicialització. Aquesta fase està dividida en l'especificació d'EDETA en dos subfases i aquest paràmetre especifica la durada d'aquestes subfases. Per tant, la fase d'inicialització dura $2T_{Config}$.

$T_{SuperFrame}$ Aquest paràmetre indica la durada d'una ronda en la fase de funcionament. Per tant, un node envia dades al seu clusterhead cada $T_{SuperFrame}$ segons.

MAX_{Rounds} Aquest paràmetre configura el número de rondes que els nodes poden estar en la fase de funcionament. Per tant, els nodes estan en la fase de funcionament $MAX_{Rounds} \times T_{SuperFrame}$ segons.

Per tant, el que l'expressió 4.3 el que intenta és especificar un valor d'energia mínim per a que un node passe a ser clusterhead. Per descomptat, aquest valor d'energia mínim ha d'anar ajustant-se a mesura que l'energia dels nodes es va consumint i això s'aconsegueix amb la variable E_T . Així si un node que en la macro-ronda anterior ha sigut clusterhead calcula un valor aleatori major que el que li especifica la equació 4.2, amb tota probabilitat no podrà tornar a ser clusterhead a la macro-ronda següent perquè és molt difícil que el seu nivell d'energia siga major que el valor retornat per la funció 4.3.

4.3.2 Llindars d'acceptació de nodes

Un clusterhead pot acceptar la unió de nodes normals fins a un número màxim de nodes determinat per paràmetre. Si sols s'implementa un llindar, pot ocórrer el cas de la Figura 4.3. En ella es veu com la majoria dels nodes han triat unir-se al clusterhead del centre, i aquest aplegat al seu límit d'acceptació de nodes; deixant "e" orfe, mentre que l'altre clusterhead podria acceptar més nodes. Aquesta situació es podria haver evitat si el node "d" s'haguera unit al clusterhead de baix, deixant espai per a "e" en el clusterhead de dalt.

²Els nodes disposen d'aquestes dades perquè un clusterhead en el missatge que envia per a anunciar el seu rol inclou el seu nivell d'energia.

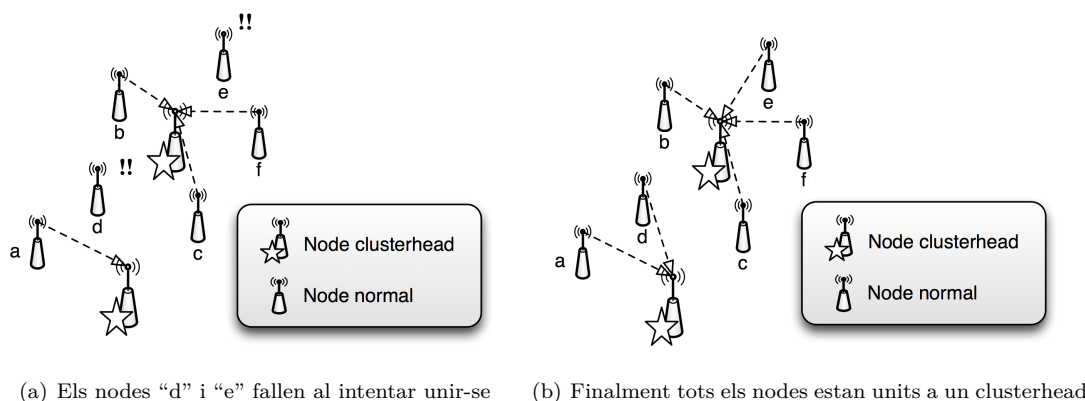


Figura 4.4: Exemple d'unió amb els dos llindars

Per a intentar evitar aquesta situació tant com siga possible sense introduir un sobrecost gran en la fase de configuració i intentar repartir els nodes de forma més equitativa, s'ha definit dos llindars: un llindar bla i un llindar dur. Un node podrà unir-se sense problemes a un clusterhead mentre aquest no arribe al seu llindar bla. En el moment un clusterhead aplegue al seu llindar bla rebutjarà totes les peticions d'unió informant que ha aplegat al seu llindar bla. Quan un node rep que ha sigut rebutjat a causa d'un llindar bla, intenta unir-se a un altre clusterhead que tinga aprop i si no en té cap, torna a intentar la unió assenyalant al missatge que és el seu últim recurs. Si un clusterhead que ha arribat al seu llindar bla rep un missatge amb el bit d'últim recurs activat accepta al node però, una vegada arriba al llindar dur ja no accepta més unions, encara que tinguen el bit d'últim recurs activat.

Per tant, amb aquestos dos llindars la seqüència d'acció podria haver sigut la de la Figura 4.4: A l'inici, Figura 4.4(a), els nodes "a", "b", "c" i "f" s'uneixen als clusterheads que han elegit però "e" i "d" reben un missatge de rebuig perquè el clusterhead que han seleccionat ha aplegat al seu llindar bla. Així que, a la Figura 4.4(b), "d" decideix provar amb l'altre clusterhead que escolta i aconsegueix unir-se. Mentrestant, "e" envia un altra petició al clusterhead indicant que és el seu últim recurs i aquest l'accepta.

Aquesta solució no evita que es pugui donar el cas de la Figura 4.3 però si que el mitiga en gran mesura sense introduir un sobrecost molt gran en la fase de configuració.

4.3.3 Planificació de l'enviament de les dades

A l'apartat 4.3.1 s'han esmentat les raons per les que s'ha decidit fer la planificació de l'estructura de la xarxa de forma distribuïda però, que passa amb l'enviament de les dades? EDETA defineix un temps d'enviament per a cada node i aquest temps s'ha de planificar.

Un altra vegada apareixen dos possibilitats: una planificació centralitzada i una distribuïda. La planificació centralitzada de l'enviament de les dades té menys dificultat perquè EDETA aborda aquesta qüestió quan l'estructura de la xarxa ja està definida. Per tant, els nodes podrien saber com enviar l'informació necessària al node recol·lector per a que fos este el que computara la planificació i la difonguera per la xarxa.

El major desavantatge d'aquesta aproximació està en la difusió de la planificació. Difondre la planificació de centenars o milers de nodes pot costar prou de temps en fer-se, obligant a tots els nodes de la xarxa a estar escoltant fins que els aplega la informació. A més, una vegada l'estructura de la xarxa està definida, una planificació centralitzada de l'enviament de les dades en EDETA no aporta més optimització que la planificació distribuïda. Els nodes poden saber quan van a tindre les seues dades i informar als seus superiors en la jerarquia de la xarxa, fent la planificació tots al mateix temps i de forma distribuïda. Per tant, com que la planificació centralitzada no aporta un clar avantatge front a la distribuïda, EDETA duu a terme una planificació distribuïda dividida en dos parts o nivells.

La primera part és la planificació intra-cluster, la segona és la planificació extra-cluster o planificació en l'arbre. En la planificació intra-cluster el clusterhead informa als nodes normals que se li han unit en quin temps li van a enviar les dades i això ho duu a terme en el mateix moment de la

unió. Quan un clusterhead accepta un node en el seu cluster li assigna la ranura de temps en la que li podrà enviar les dades. D'aquesta forma, alhora que es duu a terme la planificació de l'estructura de la xarxa també es duu a terme la planificació intra-cluster.

Quan l'estructura de la xarxa ja està definida es quan els clusterheads passen a la segona part de la planificació, la planificació extra-cluster o planificació en l'arbre. En aquesta part, els clusterheads ja saben en quin moment van a disposar de totes les dades dels seus nodes del cluster i poden informar d'eixe temps als seus clusterheads pares en la jerarquia d'arbre³. Una vegada un clusterhead obté els temps d'enviament dels clusterheads que té per davall en l'arbre pot computar el seu temps d'enviament i informar al seu pare. Aquest procediment es va repetint fins que s'aplega al node recol·lector.

³Hi ha que recordar, que EDETA té dos nivells de jerarquia, el primer nivell són els clusters i el segon nivell ve donat per un arbre que es forma entre els clusterheads.

Capítol 5

Experimentació i resultats

Una part molt important del disseny d'algorismes consisteix en l'experimentació. Mitjançant l'experimentació es pot analitzar el comportament de l'algorisme i decidir en base a això si es fa algun canvi a l'especificació per a intentar millorar el seu comportament. Evidentment, després de cada canvi s'han de tornar a repetir els experiments per a comprovar si l'efecte produït és el desitjat.

A més, la comparació de resultats front a altres algorismes també és molt important. Això permet analitzar els punts forts i febles de l'algorisme que s'està dissenyat front a altres algorismes prèviament proposats i així definir clarament quines són les seues aportacions principals.

5.1 Avaluació EDETA en radiofreqüència amb el simulador NS-2

Com s'ha explicat al Capítol 4 la primera implementació d'EDETA es va dur a terme al simulador NS-2 utilitzant el mateix model que utilitzaren els autors de LEACH per a la implementació del seu protocol. Aquest model és per a xarxes de radiofreqüència i per tant, tot aquest apartat tracta del comportament d'EDETA en aquest medi de transmissió.

A part de les simulacions d'EDETA, també s'han dut a terme simulacions de LEACH [HCB02] per a la comparació de resultats.

5.1.1 Paràmetres generals de les simulacions i escenaris

Els paràmetres que es van a detallar a continuació s'han utilitzat en les simulacions fetes amb NS-2 en tots els models.

La velocitat de transmissió es de 250 Kbps i la freqüència de la portadora del senyal és de 868 MHz. La distància màxima de transmissió s'ha configurat en 100 metres en les simulacions d'EDETA però, aquesta limitació no s'ha imposat per a LEACH. Aquest protocol assumeix que tots els nodes són capaços d'enviar la seua informació directament al node recol·lector sense importar la distància que els separe.

Totes les simulacions s'han dut a terme amb un node recol·lector al centre de l'escenari. S'han simulat escenaris amb 200, 250, 300, 350 i 400 nodes distribuïts de forma aleatòria. Per a cada un d'aquest número de nodes s'han fet escenaris quadrats amb laterals de 200, 250, 300, 350 i 400 metres. La Taula 5.1 resumeix els escenaris en els que s'han dut a terme les simulacions. Cada escenari (fila de la Taula 5.1) s'ha simulat varies vegades i s'ha extret una mitja de cada cas.

Nodes	Lat. escen. (m)	Nodes	Lat. escen. (m)	Nodes	Lat. escen. (m)	Nodes	Lat. escen. (m)	Nodes	Lat. escen. (m)
	200		200		200		200		200
	250		250		250		250		250
200	300	250	300	300	300	350	300	400	300
	350		350		350		350		350
	400		400		400		400		400

Taula 5.1: Taula resum d'escenaris en els que s'han dut a terme les simulacions

Model d'energia

El model d'energia té en conte solament el consum d'energia de la radio. La radio pot estar en quatre estats diferents, transmissió, recepció, espera i baix consum i cada un d'aquests estats té un consum d'energia diferent. Per a calcular l'energia consumida en la transmissió d'un paquet s'ha utilitzat l'equació 5.1

$$E_{tx}(p) = P_{tx} \times t_{tx}(p) \quad (5.1)$$

On p és el paquet a enviar, P_{tx} és un paràmetre que especifica el valor de la potencia de transmissió i t_{tx} és el temps que la radio està en l'estat de transmissió, que ve donat per el temps que es tarda en transmetre un paquet i es calcula segons l'equació 5.2.

$$t_{tx}(p) = \frac{|p|}{v_{tx}} \quad (5.2)$$

On $|p|$ representa la grandària en bits del paquet a transmetre i v_{tx} és la velocitat de transmissió en bits per segon.

L'energia consumida per els estats de recepció, espera i baix consum ve donada per les equacions 5.3, 5.4 i 5.5, respectivament.

$$E_{rx} = P_{rx} \times t_{rx} \quad (5.3)$$

$$E_{idle} = P_{idle} \times t_{idle} \quad (5.4)$$

$$E_{sleep} = P_{sleep} \times t_{sleep} \quad (5.5)$$

On P_{rx} , P_{idle} i P_{sleep} són la potencia dels estats de recepció, espera i baix consum respectivament i t_{rx} , t_{idle} i t_{sleep} són els temps en els que la radio està en els estats de recepció, espera i baix consum respectivament.

Els paràmetres de consum dels nodes s'han extret de mesures preses a nodes reals al laboratori de Xarxes de Sensors i Sistemes Encastats del Grup de Sistemes Tolerants a Fallades de la Universitat Politècnica de València. El transceptor de ràdio consumeix en el mode de baix consum 3.6×10^{-6} watts, en recepció i espera 0.072 watts i en transmissió depèn de la distancia estimada fins al receptor. Aquesta estimació es fa utilitzant el model de propagació *Two-ray Ground* que va amb NS-2.

Model de propagació

El model de propagació és l'anomenat *Two-ray Ground* i que ve definit per l'equació 5.6. En aquest model es té en conte tant la propagació per la ruta directa entre dos nodes com la propagació per la reflexió del sòl.

$$P_r(d) = \frac{P_t G_t G_r h_t^2 h_r^2}{d^4 L} \quad (5.6)$$

On P_t i P_r són la potencia de transmissió i recepció. G_t i G_r representen el guany de l'antena en l'emissor i el receptor. H_t i H_r representen l'altura de les antenes respecte del sòl. El paràmetre d és la distancia entre els dos nodes i L és la pèrdua de potencia del sistema.

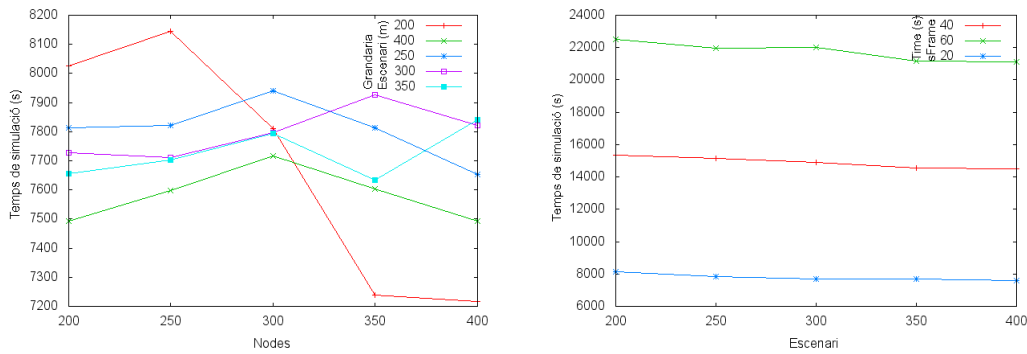
Aquest model dona bons resultats per a distancies llargues però no en les distancies curtes. Per això per a aquestes distancies s'utilitza l'equació de Friis 5.7

$$P_r(d) = \frac{P_t G_t G_r \lambda^2}{(4\pi)^2 d^2 L} \quad (5.7)$$

On P_t i P_r són la potencia de transmissió i recepció. G_t i G_r representen el guany de l'antena en l'emissor i el receptor. λ és el valor de la longitud d'ona de la portadora. El paràmetre d és la distancia entre els dos nodes i L és la pèrdua de potencia del sistema.

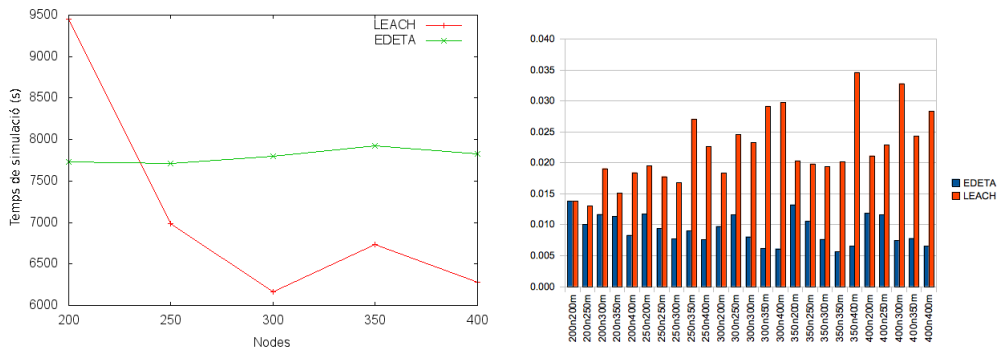
Per a decidir quin d'ambdós models s'utilitza en cada cas es mira la distancia d que separa els dos nodes. Si $d > d_c$ s'utilitza el model *Two-ray ground* i sinó s'utilitza l'equació de Friis. d_c representa la distancia llindar que senyala la utilització d'un model o un altre i ve determinada per l'equació 5.8

$$d_c = \frac{4\pi h_t h_r}{\lambda} \quad (5.8)$$



(a) Estudi del temps de vida de la xarxa variant el número de nodes i la grandària del escenari (b) Estudi del temps de vida de la xarxa variant el paràmetre $T_{SuperFrame}$ i la grandària de l'escenari

Figura 5.1: Estudi de vida de la xarxa. EDETA



(a) Estudi del temps de vida de la xarxa variant el número de nodes (b) Estudi dels paquets perduts d'EDETA i LEACH

Figura 5.2: Estudi comparatiu entre EDETA i LEACH

5.1.2 EDETA. Avaluació de prestacions

Les simulacions d'EDETA s'han fet utilitzant els paràmetres i els escenaris de l'apartat 5.1.1, als nodes s'els han assignat 20 joules d'energia i els següents paràmetres específics del protocol: els paràmetres T_{Config} i $T_{SuperFrame}$ s'han establert a 20 segons i $MaxRounds$ s'ha configurat a 20 rondes.

A la Figura 5.1(a) es pot apreciar el temps de vida de la xarxa front a diferent número de nodes i diferent grandària d'escenari. El paràmetre $T_{SuperFrame}$ és de 20 segons. Les simulacions amb major variabilitat en el temps de vida de la xarxa són les de l'escenari de 200 metres de costat. En aquestes simulacions el temps de vida de la xarxa disminueix notablement amb l'augment del número de nodes degut a les col·lisions durant el temps de configuració.

També s'ha estudiat l'efecte que té el paràmetre $T_{SuperFrame}$ sobre la duració de la xarxa. A la Figura 5.1(b) es mostra el temps de vida de la xarxa front a la grandària de l'escenari, s'ha fet variar el paràmetre $T_{SuperFrame}$ entre 20 i 60 segons i el número de nodes s'ha fixat a 250. El gràfic mostra com el paràmetre $T_{SuperFrame}$ és el major responsable en la variació del temps de la xarxa, augmentat en 7000 segons la duració per cada 20 segons d'increment del paràmetre $T_{SuperFrame}$, això suposa més d'un 30% en l'increment de vida de la xarxa. A més, ací també es pot apreciar com hi ha un lleuger descens del temps de vida de la xarxa al augmentar la grandària del escenari.

5.1.3 Comparació d'EDETA i LEACH

Com s'ha esmentat al començament del capítol, la comparació entre protocols és molt important per a saber si s'estan aconseguint millores respecte d'altres propostes. En aquest apartat es va a comparar EDETA amb LEACH. S'ha intentat que la comparació entre ambdós protocols siga el més justa possible. En LEACH una vegada s'ha fet l'estructura de la xarxa, els clusterheads van recollint informació dels seus nodes i enviant-la directament al recol·lector. Després d'un cert nombre de rondes (20 en la configuració original del protocol i en estes simulacions) l'estructura es trenca i es torna a refer. La configuració d'EDETA és la mateixa que la del apartat anterior, T_{Config} i $T_{SuperFrame}$ s'han establert a 20 segons i $MaxRounds$ s'ha configurat a 20 rondes. D'aquesta forma s'intenta igualar el màxim possible els paràmetres d'ambdós protocols.

A la Figura 5.2(a) es mostra el temps de vida de la xarxa front al número de nodes. Com es pot apreciar, LEACH és clarament superior en quant a temps de vida de la xarxa en l'escenari amb menys nodes. Però al augmentar el número de nodes, la vida de la xarxa cau considerablement. Per contra, EDETA es manté estable en tots els casos, millorant el comportament de LEACH a partir de 250 nodes.

També s'ha analitzat el ràtio de paquets perduts a la Figura 5.2(b). Es pot apreciar clarament com EDETA té un ratio de paquets perduts menor que LEACH i en alguns casos molt menor, arribant a disminuir-lo en més d'un 70%.

5.2 Avaluació d'EDETA en el medi subaquàtic amb el simulador NS-3

En aquest apartat es va a avaluar el comportament d'EDETA en el medi subaquàtic per mitja del model que s'ha implementat per al NS-3. Per a fer-ho, es va a simular el comportament d'EDETA-e en aquest nou medi de transmissió.

5.2.1 Paràmetres generals de les simulacions i escenaris

Al igual que en les simulacions del simulador NS-2, en aquestes la distància de transmissió també està limitada a 100 metres, la velocitat de transmissió és de 500 bps i la freqüència de la portadora és de 85KHz.

Totes les simulacions s'han dut a terme amb un node recol·lector al centre de l'escenari. S'han simulat tres escenaris quadrats: el primer amb 100 nodes i 100 metres de costat, el segon amb 200 nodes i 200 metres de costat i el tercer amb 250 nodes i 500 metres de costat.

S'han incrementat considerablement els paràmetres temporals del protocol per a fer front a l'increment en el temps de transmissió i, sobretot, el de propagació. El paràmetre T_{Config} s'ha establert a 9000 segons, $T_{SuperFrame}$ s'ha configurat a 250 segons i $MaxRounds$ s'ha quedat amb el mateix valor, 20 rondes.

Model d'energia

El model d'energia utilitzat és equivalent al del NS-2 però, evidentment, variant els paràmetres de consum que s'han extret del WHOI micromodem[FGS⁺06]. El transceptor de ràdio consumeix en el mode de baix consum 0.0058 watts, en recepció i espera 0.158 watts i en transmissió 50 watts.

Model de propagació

El model de propagació utilitzat en aquestes simulacions és l'anomenat model de Thorp. En ell l'atenuació total es calcula basant-se en la pèrdua associada a la disseminació de l'ona i la pèrdua per la absorció del senyal. L'aproximació de Thorp per a l'absorció del canal ve donada per l'equació 5.9

$$10 \log a(f) = \begin{cases} 0.11 \frac{f^2}{1+f^2} + 44 \frac{f^2}{4100+f} + 2.75 \times 10^{-4} f^2 + 0.003 & f \geq 0.4 \\ 0.002 + 0.11 \frac{f}{1+f} + 0.011 f & f < 0.4 \end{cases} \quad (5.9)$$

On $a(f)$ ve donat en db/km i f en KHz. Per tant, a les simulacions que es detallaran a continuació sols els afectarà la primera part de l'equació ja que la freqüència de la portadora és de 85 KHz en tots els casos.

La combinació de la atenuació total es calcula amb la fórmula 5.10

$$10 \log A(l, f) = k \times 10 \log l + l \times 10 \log a(f) \quad (5.10)$$

On el primer terme és l'atenuació provocada per la disseminació i el segon terme és la pèrdua per absorció del senyal. El coeficient de disseminació k defineix la geometria de la propagació del senyal i pot ser cilíndrica $k = 1$ o esfèrica amb $k = 2$. En la pràctica s'utilitza un valor intermedi $k = 1.5$ degut a variacions en la forma de la disseminació del senyal.

5.2.2 Experiments

Per a dur a terme els experiments amb aquest simulador s'ha procedit de la mateixa forma que amb el simulador NS-2, les simulacions de cada escenari s'han repetit varies vegades i s'ha extret una mitja de cada cas.

	100 nodes	200 nodes	250 nodes
Mitja	5.17	15.39	4.64
Mitja per node	0.05	0.15	0.05

Taula 5.2: Taula amb la mitja de paquets perduts

La Taula 5.2 fa un resum del número de paquets perduts per node. Com es pot veure, la mitja de paquets perduts per node és molt baixa 0.015 paquets per node en el cas de l'escenari amb 200 nodes i tan sols 0.05 paquets en el cas dels escenaris de 100 i 250 nodes.

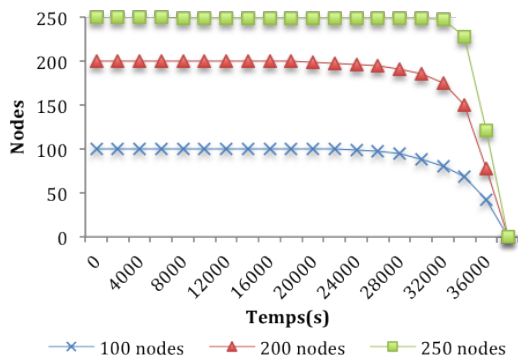
A la Figura 5.3(a) es mostra el temps de vida dels escenaris. L'algorisme té un comportament idoni per a aquest tipus de xarxes, la gran majoria dels nodes es queda sense energia al mateix temps; el que evita que la xarxa es divideixi en parts, trencant la comunicació amb cert número de nodes. Com el consum d'energia ve dominat per el mode de transmissió i tots els nodes estan, de mitja, el mateix temps en aquest estat, el consum d'energia dels nodes és paregut.

L'energia total consumida fins al segon 30000 es mostra a la Figura 5.3(b), on es poden apreciar clarament tres zones ben diferenciades. A la primera zona, des de l'inici de la simulació fins al segon 4000, es on es dona el major consum energètic en el menor temps. Aquesta part correspon a la fase d'inicialització on els nodes intenten unir-se als clusterheads i hi ha una gran activitat a la xarxa, el que provoca un increment en el consum molt gran. En la segona part, que compren des del segon 4000 fins al 18000, quasi no hi ha consum d'energia, degut a que els nodes en aquesta fase estan en un estat de baix consum perquè s'estan computant la planificació entre els clusterheads. Finalment, la tercera part mostra un creixement en el consum d'energia més gran que la segona però significativament més xicotet que la primera. En aquesta part els nodes estan en la fase de funcionament on es desperten regularment per a enviar les seues dades.

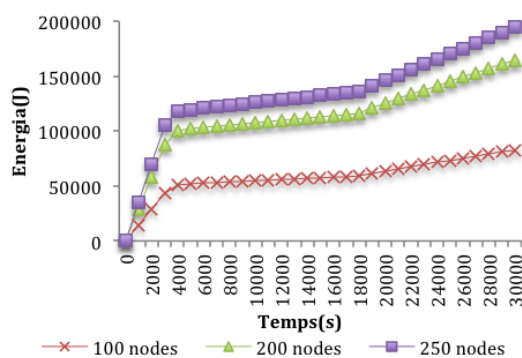
Per últim, a la Figura 5.3(c) s'estudia l'efecte del paràmetre $T_{SuperFrame}$ en el temps de vida de la xarxa. S'han fet simulacions incrementant el seu valor des dels 250 segons fins als 800 segons. El resultat obtingut es el que es mostra, el temps de vida de la xarxa creix linealment amb aquest paràmetre. A la gràfica es mostren els resultats per a l'escenari de 250 nodes per simplicitat, perquè els punts dels altres escenaris cauen aproximadament a les mateixes coordenades a la gràfica.

A més, s'ha calculat una regressió lineal per a extraure l'expressió 5.11 amb un índex de correlació $R^2 = 0.998$, el que permet predir de forma acurada el temps de vida de la xarxa en absència d'errors.

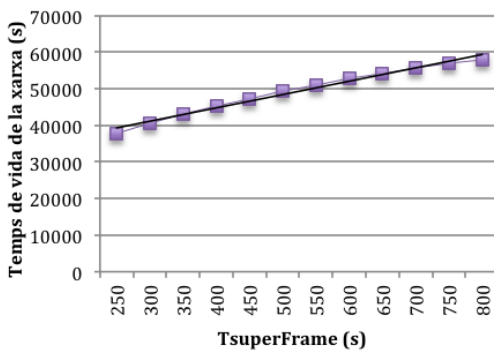
$$y = 1823x + 37485 \quad (5.11)$$



(a) Estudi del temps de vida de la xarxa variant el número de nodes



(b) Energia consumida



(c) Efecte del paràmetre $T_{SuperFrame}$ en el temps de vida de la xarxa

Figura 5.3: Estudi d'EDETA en el medi subaquàtic

Capítol 6

Conclusions, Publicacions i Treball Futur

6.1 Conclusions

L'objectiu principal d'aquesta tesina ha sigut el de la implementació i avaluació d'EDETA [Cap10] [CBOC10], un algorisme d'encaminament per a xarxes de sensors sense fil. Com a resultat d'aquesta avaluació s'han proposat canvis a l'especificació de l'algorisme sempre intentant respectar els requeriments bàsics a partir dels que va sorgir l'especificació del protocol: minimitzar el consum d'energia, aconseguir una gran escalabilitat i flexibilitat sense perdre fiabilitat i intentar conferir-li propietats dels protocols de temps real.

L'avaluació de l'algorisme s'ha dut a terme en dos medis de transmissió totalment diferents. Per una part, s'ha avaluat l'algorisme en les xarxes de sensors sense fil clàssiques, les que operen amb radiofreqüència. Per a fer-ho, s'han implementat un model de l'algorisme al simulador NS-2 i s'han conduït diferents proves per a comprovar la seua eficiència energètica, escalabilitat i fiabilitat. Dels resultats extrets de les proves es pot concloure que EDETA es un algorisme que té una bona escalabilitat. La grandària de l'escenari no afecta en gran mesura al temps de vida de la xarxa, encara que si que és sensible a la densitat de nodes. A més posseeix una molt bona fiabilitat, amb menys d'un 1.5% d'error en l'entrega de paquets. Tot açò sense perdre flexibilitat i oferint temps acotats en l'entrega de paquets en absència d'errors.

Per altra part, seguint el camí que està prenguen la comunitat científica en aquest tipus de xarxes, s'ha avaluat EDETA en el medi de transmissió subaquàtic mitjançant ones acústiques. Per a dur a terme aquest canvi de medi de transmissió, s'ha estudiat la continuació del desenvolupament per al simulador NS-2 o el canvi al nou simulador NS-3. Vistos els avantatges i desavantatges de cada alternativa es va decidir passar a utilitzar NS-3, degut a la major facilitat de programació i depuració que ofereix. Es va implementar el model d'EDETA per a xarxes subaquàtiques, així com es varen realitzar i estudiar noves propostes per a l'òptima adaptació del mateix al nou medi.

Els resultats extrets de les proves en el medi subaquàtic han mostrat una bona adaptació del protocol al nou medi. El protocol manté un baix percentatge de paquets perduts (1.5%) i ha mostrat un comportament estable en les proves realitzades. Aquesta adaptació ha sigut propiciada, sobretot, per la gran quantitat de paràmetres que posseeix l'algorisme i que han permès adaptar-lo sense fer modificacions a les seues especificacions.

6.2 Publicacions i altres contribucions

El present treball d'investigació ha donat lloc a dos publicacions en congressos internacionals:

- J.V. Capella, A. Bonastre, R. Ors, and S. Climent. A new energy-efficient, scalable and robust architecture for wireless sensor networks. In *3rd International Conference on New Technologies, Mobility and Security (NTMS), 2009* pages 1-6. IEEE 2010.

- S. Climent, J.V. Capella, A. Bonastre, and R. Ors. A new model for the NS-3 simulator of a novelty routing protocol applied to underwater wireless sensor networks. In *4th International ICST Conference on Simulation Tools and Techniques*. En revisió.

A més, durant el desenvolupament del model per al NS-3 s'ha contribuït amb dos informes d'errors amb els seus corresponents pegats de codi. Aquests informes ja han sigut acceptats per el grup de desenvolupament del simulador i el codi dels pegats inclòs a la versió de desenvolupament per a publicar-lo a la següent versió estable. A més, també s'ha enviat el codi amb les funcionalitats afegides al model subaquàtic. Els dos informes d'error i el codi de les funcionalitats afegides es pot consultar a la web del simulador [NS-10c] a l'apartat de seguiment d'errors.

6.3 Treball futur

EDETA ha demostrat ser un algorisme d'encaminament amb una gran adaptació a medis de transmissió ben diferents. Així i tot, hi ha certs aspectes que es poden optimitzar en l'adaptació d'aquest algorisme a les xarxes subaquàtiques.

El temps de configuració de la xarxa es prou elevat en el medi subaquàtic. Els alts retard de propagació, així com les extremadament baixes tasses de transmissió de dades obliguen a que els nodes necessiten alts temps de configuració, sobretot si es comparen amb els temps de configuració de les xarxes terrestres.

Una forma d'abordar aquest problema pot ser aprofitar el canvi que hi ha en el factor de consum dominant. En les xarxes terrestres l'estat de recepció té un consum elevat, per contra, en les xarxes subaquàtiques, l'estat que més consumeix és el de transmissió. En EDETA en una mateixa fase de configuració s'intenten configurar diferents aspectes de la xarxa disminuint al màxim l'estat de recepció i el fet de fer retransmissions dels paquets de configuració no és tan important ja que la transmissió és relativament "barata". En aquest nou paradigma, on els retards de transmissió i propagació són molt més elevats i el cost energètic ve dominat per la transmissió, les col·lisions són molt més probables i les retransmissions "cares". Per això, la divisió de la fase de configuració en sub-fases on es configure un aspecte del protocol cada vegada pot millorar els temps de configuració i reduir el consum d'aquesta part.

Com a futura línia per a la tesis doctoral hi ha diferents aspectes que es pretenen abordar. La integració de nodes mòbils en la xarxa formada per EDETA és un d'ells. La investigació en les xarxes subaquàtiques es mou cap a xarxes amb nodes mòbils i estendre EDETA per a suportar aquest tipus de xarxes mantenint les seues propietats d'escalabilitat, flexibilitat i fiabilitat constitueix un gran repte. A més, es pretén estudiar la integració de tècniques dels sistemes multi-agent per a conferir-li al protocol una major adaptabilitat en un entorn canviant. Aquestes tècniques, pròpies dels camps de la intel·ligència artificial, podrien conferir-li al protocol nous camps d'aplicació com el guiat de robots o vehicles submarins.

Bibliografía

- [ACDFP09] G. Anastasi, M. Conti, M. Di Francesco, and A. Passarella. Energy conservation in wireless sensor networks: A survey. *Ad Hoc Networks*, 7(3):537–568, 2009.
- [AKK04] J.N. Al-Karaki and A.E. Kamal. Routing techniques in wireless sensor networks: a survey. *Wireless Communications, IEEE*, 11(6):6 – 28, dec. 2004.
- [Ald08] OO Aldawibio. A review of current routing protocols for ad hoc underwater acoustic networks. In *Applications of Digital Information and Web Technologies, 2008. ICADIWT 2008. First International Conference on the*, pages 431–434. IEEE, 2008.
- [ALVM06] I.F. Akyildiz, W.Y. Lee, M.C. Vuran, and S. Mohanty. Next generation/dynamic spectrum access/cognitive radio wireless networks: a survey. *Computer Networks*, 50(13):2127–2159, 2006.
- [APM04] I.F. Akyildiz, D. Pompili, and T. Melodia. Challenges for efficient communication in underwater acoustic sensor networks. *ACM Sigbed Review*, 1(2):3–8, 2004.
- [AY07] A.A. Abbasi and M. Younis. A survey on clustering algorithms for wireless sensor networks. *Computer Communications*, 30(14-15):2826–2841, 2007.
- [Cap10] J.V. Capella. *Wireless Sensor Networks: A new efficient and robust architecture based on dynamic hierarchy of clusters*. PhD thesis, Universidad Politécnic de Valencia, 2010.
- [CBOC10] JV Capella, A. Bonastre, R. Ors, and S. Climent. A new energy-efficient, scalable and robust architecture for wireless sensor networks. In *New Technologies, Mobility and Security (NTMS), 2009 3rd International Conference on*, pages 1–6. IEEE, 2010.
- [CSC07] N. Chirdchoo, W.S. Soh, and K.C. Chua. Aloha-based MAC protocols with collision avoidance for underwater acoustic networks. In *INFOCOM 2007. 26th IEEE International Conference on Computer Communications. IEEE*, pages 2271–2275. IEEE, 2007.
- [CSCE06] V. Chandrasekhar, W.K.G. Seah, Y.S. Choo, and H.V. Ee. Localization in underwater sensor networks: survey and challenges. In *Proceedings of the 1st ACM international workshop on Underwater networks*, pages 33–40. ACM, 2006.
- [CSS08] M. Chitre, S. Shahabudeen, and M. Stojanovic. Underwater acoustic communications and networking: Recent advances and future challenges. *The State of Technology in 2008*, 42(1):103–114, 2008.
- [DEA06] I. Demirkol, C. Ersoy, and F. Alagoz. MAC protocols for wireless sensor networks: a survey. *Communications Magazine, IEEE*, 44(4):115–121, 2006.
- [FAC⁺05] KY Foo, PR Atkins, T. Collins, C. Morley, and J. Davies. A routing and channel-access approach for an ad hoc underwater acoustic network. In *OCEANS'04. MTTs/IEEE TECHNO-OCEAN'04*, volume 2, pages 789–795. IEEE, 2005.
- [FGS⁺06] L. Freitag, M. Grund, S. Singh, J. Partan, P. Koski, and K. Ball. The WHOI micro-modem: an acoustic communications and navigation system for multiple platforms. In *OCEANS, 2005. Proceedings of MTS/IEEE*, pages 1086–1092. IEEE, 2006.

- [FRD06] M.R. Frater, M.J. Ryan, and R.M. Dunbar. Electromagnetic communications within swarms of autonomous underwater vehicles. In *Proceedings of the 1st ACM international workshop on Underwater networks*, pages 64–70. ACM, 2006.
- [HCB02] W.R. Heinzelman, A. Chandrakasan, and H. Balakrishnan. Energy-efficient communication protocol for wireless microsensor networks. In *System Sciences, 2000. Proceedings of the 33rd Annual Hawaii International Conference on*, page 10. IEEE, 2002.
- [HF10] T. Hu and Y. Fei. QELAR: A Machine-Learning-Based Adaptive Routing Protocol for Energy-Efficient and Lifetime-Extended Underwater Sensor Networks. *Mobile Computing, IEEE Transactions on*, 9(6):796–809, 2010.
- [IEE03] IEEE. IEEE Standard for Information Technology- Telecommunications and Information Exchange Between systems- Local and Metropolitan Area Networks- Specific Requirements- part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications. *ANSI/IEEE Std 802.11, 1999 Edition (R2003)*, pages i–513, 2003.
- [IEE06] IEEE. IEEE 802.15.4 standard, 2006 revision, 2006.
- [IEE09] IEEE. Ieee standard for information technology–telecommunications and information exchange between systems–local and metropolitan area networks–specific requirements part 11: Wireless lan medium access control (mac) and physical layer (phy) specifications amendment 5: Enhancements for higher throughput. *IEEE Std 802.11n-2009 (Amendment to IEEE Std 802.11-2007 as amended by IEEE Std 802.11k-2008, IEEE Std 802.11r-2008, IEEE Std 802.11y-2008, and IEEE Std 802.11w-2009)*, pages c1–502, oct. 2009.
- [LR05] S. Lindsey and C.S. Raghavendra. PEGASIS: Power-efficient gathering in sensor information systems. In *Aerospace Conference Proceedings, 2002. IEEE*, volume 3, page 3. IEEE, 2005.
- [LSC02] P. L’ecuyer, R. Simard, E.J. Chen, and W.D. Kelton. An object-oriented random-number package with many long streams and substreams. *Operations Research*, 50(6):1073–1075, 2002.
- [MA01] Arati Manjeshwar and Dharma P. Agrawal. TEEN: A Routing Protocol for Enhanced Efficiency in Wireless Sensor Networks. *Parallel and Distributed Processing Symposium, International*, 3:30189a, 2001.
- [MWH02] M. Mauve, A. Widmer, and H. Hartenstein. A survey on position-based routing in mobile ad hoc networks. *Network, IEEE*, 15(6):30–39, 2002.
- [np10a] ns 3 project. *ns-3 Reference Manual*, August 2010.
- [np10b] ns 3 project. *ns-3 Tutorial*, August 2010.
- [NS-10a] NS-2 acoustic model team. NS-2 acoustic model. <http://www.dei.unipd.it/wdyn/?IDsezione=5216>, 2010.
- [NS-10b] NS-2 team. Network Simulator 2. <http://nsgam.isi.edu/nsgam/>, 2010.
- [NS-10c] NS-3 team. Network Simulator 3. <http://www.nsgam.org/>, 2010.
- [PMA06] D. Pompili, T. Melodia, and I.F. Akyildiz. A resilient routing algorithm for long-term applications in underwater sensor networks. In *Proc. of Mediterranean Ad Hoc Networking Workshop (Med-Hoc-Net)*. Citeseer, 2006.
- [RM02] V. Rodoplu and T.H. Meng. Minimum energy mobile wireless networks. *Selected Areas in Communications, IEEE Journal on*, 17(8):1333–1344, 2002.

- [SCI⁺01] E. Shih, S.H. Cho, N. Ickes, R. Min, A. Sinha, A. Wang, and A. Chandrakasan. Physical layer driven protocol and algorithm design for energy-efficient wireless sensor networks. In *Proceedings of the 7th annual international conference on Mobile computing and networking*, pages 272–287. ACM, 2001.
- [SGAP00] K. Sohrabi, J. Gao, V. Ailawadhi, and G.J. Pottie. Protocols for self-organization of a wireless sensor network. *IEEE personal communications*, 7(5):16–27, 2000.
- [Sto07] M. Stojanovic. On the relationship between capacity and distance in an underwater acoustic communication channel. *ACM SIGMOBILE Mobile Computing and Communications Review*, 11(4):34–43, 2007.
- [Sto08] I. Stojmenovic. Simulations in wireless sensor and ad hoc networks: matching and advancing models, metrics, and solutions. *Communications Magazine, IEEE*, 46(12):102–107, 2008.
- [tea10] FunLab team. Fundamentals of Networking Laboratory. University of Washington. <http://ee.washington.edu/research/funlab/uan/>, 2010.
- [TTB94] J.L. Talavage, T.E. Thiel, and D. Brady. An efficient store-and-forward protocol for a shallow-water acoustic local area network. In *OCEANS'94. Oceans Engineering for Today's Technology and Tomorrow's Preservation. Proceedings*, volume 1. IEEE, 1994.
- [UR08] M. Umlauf and P. Reichl. Experiences with the ns-2 network simulator—explicitly setting seeds considered harmful. In *Wireless Telecommunications Symposium, 2007. WTS 2007*, pages 1–5. IEEE, 2008.
- [WC01] A. Woo and D.E. Culler. A transmission control scheme for media access in sensor networks. In *Proceedings of the 7th annual international conference on Mobile computing and networking*, pages 221–235. ACM, 2001.
- [XC06] P. Xie and J.H. Cui. Exploring random access and handshaking techniques in large-scale underwater wireless acoustic sensor networks. In *Proceedings of the MTS/IEEE OCEANS Conference*. Citeseer, 2006.
- [XCL06] P. Xie, J.H. Cui, and L. Lao. Vbf: vector-based forwarding protocol for underwater sensor networks. *Networking 2006. Networking Technologies, Services, and Protocols; Performance of Computer and Communication Networks; Mobile and Wireless Communications Systems*, pages 1216–1221, 2006.
- [XG02] G.G. Xie and J.H. Gibson. A network layer protocol for uans to address propagation delay induced performance limitations. In *OCEANS, 2001. MTS/IEEE Conference and Exhibition*, volume 4, pages 2087–2094. IEEE, 2002.
- [XHE01] Y. Xu, J. Heidemann, and D. Estrin. Geography-informed energy conservation for ad hoc routing. In *Proceedings of the 7th annual international conference on Mobile computing and networking*, pages 70–84. ACM, 2001.