UNIVERSITATEA “POLITEHNICA” TIMIŞOARA
FACULTATEA DE AUTOMATICĂ ŞI CALCULATOARE

Contribuții la optimizarea resurselor în rețele locale fără fir

Teză de doctorat

Doctorand: as. ing. Sebastian Fuicu

Coordonator: prof. dr. ing. Crişan Strugaru

2009

Cuprins

Introducere

Obiectivele tezei

Domeniul comunicațiilor mobile a căpătat o dezvoltare fără precedent în ultimii ani, marcând apariția pe piață a numeroase tipuri de dispozitive mobile (notebook-uri, pda-uri, smartphone-uri) fiecare fiind însoțit de o paletă largă de aplicații software, de la cele industriale până la divertisment. Pentru a putea comunica cu alte sisteme, aceste dispozitive trebuie conectate pentru a forma o rețea, iar pentru a beneficia din plin de avantajul mobilității, conexiunile trebuie să fie wireless. Există la ora actuală mai multe standarde de comunicație wireless pentru transmisia de date: GSM-GPRS, WLAN 802.11, Bluetooth. Dintre acestea, in lucrarea de față a fost aleasă ca şi modalitatea de conexiune, standardul 802.11. Motivele care au stat la baza acestei opțiuni țin de ratele de transfer suportate de acest standard si costurile reduse de exploatare (nefiind nevoie de un abonament lunar si nici o limită in ceea ce priveşte cantitatea de date vehiculate). Un alt criteriu pentru alegerea unui anumit standard au fost gradul de mobilitate al un dispozitiv mobil. S-a avut în vedere o rețea care să funcționeze cu preponderență in interiorul unor clădiri iar arhitectura rețelelor să fie una de tip celular. Astfel, standardul care corespunde cel mai bine acestor cerințe este WLAN 802.11.

Principala particularitate a rețelelor wireless este aceea că, mediul fizic folosit în cazul lor sunt undele radio. Acest mediu are câteva trăsături specifice: este un mediu care nu are o delimitare clară în spațiu, nu este protejat față de interferențele cu alte semnale, are o topologie care se poate modifica uşor, nu putem avea certitudinea că orice stație este „auzită" de către a orice altă stație, iar modul de propagare a semnalelor poate varia în timp şi poate prezenta asimetrii.

Din păcate, odată cu dezvoltarea explozivă a comunicațiilor wireless s-a constatat că unul din principalele protocoalele de comunicații şi anume TCP-ul, nu a fost gândit să funcționeze peste conexiuni wireless. Problema majora apare datorită faptului că mecanismele interne pentru controlul congestiei, cu care a fost prevăzut, vor reacționa în mod eronat în cazul pierderilor de pachete şi a întârzierilor introduse de conexiunile wireless, interpretându-le ca pe un fenomen de apariție a congestiei, rezultatul fiind diminuarea drastică a ratelor de transfer. Efectul este prelungirea duratei de comunicație şi deci, ocuparea canalului de comunicație un timp mai îndelungat. În literatura de specialitate au fost propuse diverse metode de corectarea a acestui fenomen nedorit. În cadrul standardului 802.11 au fost prevăzute diverse mecanisme de gestionare a resurselor

dar care sunt implementate doar in primele nivele din modelul OSI, adică Nivelul Fizic si Nivelul Legătură de Date. Pentru a optimiza funcționarea unei astfel de rețele, în lucrarea de față este propusă o soluție de tip *cross layer* , adică este introdus pe Nivelul Aplicație un software menit să monitorizeze parametrii rețelei şi să ia măsuri locale sau globale în vederea optimizării performanțelor acesteia. Pe parcursul lucrării vor fi definiți rând pe rând şi aceşti parametrii de performanță fără de care o analiză calitativă a funcționării rețelei nu s-ar putea face. Monitorizarea se va face pe nivele 2, 4 şi 7 corespunzătoare modelului OSI, adică Nivelul Legătură de Date, Nivelul Transport si Nivelul Aplicație. Toate tehnicile de analiză şi metodele de gestionare a resurselor puse la punct pe parcursul lucrării sunt parte integrantă dintr-o aplicație software complexă dezvoltată sub forma unui framework software care permite diverselor aplicații care rulează într-o rețea wireless să interacționeze cu rețeaua în aşa fel, încât aceasta să aloce optimul de resurse pentru rularea lor. Acest framework a primit denumirea de ***UFRM (Unified Framework for Resources Management).***

Ideea dezvoltată în această lucrare este de a încerca, pe cât posibil, ca într-o rețea 802.11 să se evite apariția erorilor şi întârzierilor pentru a nu declanşa mecanismele de control ale congestiei implementate de TCP. Pentru a putea evalua soluțiile propuse este nevoie să existe nişte metode de analiză calitativă a unei conexiuni TCP. ***Elaborarea unor astfel de metode de analiză a unei conexiuni TCP a fost un alt obiectiv al lucrării de față.***

Atunci când în interiorul unei rețele 802.11, un dispozitiv mobil se află în mişcare este posibil ca, la un moment dat, conexiunea stabilită cu unul dintre access point-urile acelei rețele să se întrerupă din cauza scăderii puterii semnalului sub o anumită valoare. În această situație este nevoie ca acel dispozitiv să încerce o asociere la un alt access point, care oferă un semnal mai puternic. Acesta mecanism de reasociere se numeşte în literatura de specialitate *handover.* **În lucrarea de față este propusă o metodă care să ducă la optimizarea procedeului de handover, astfel încât, comunicația aflată în desfăşurare printr-o conexiune wireless să nu aibă prea mult de suferit atunci când se impune comutarea la un alt access point.**

În domeniul acesta al aplicațiilor mobile, un loc tot mai important îl ocupă conceptul de *context-aware*, adică dezvoltarea de sisteme hard-soft care să fie capabile să identifice un anumit context în care utilizatorul îşi desfăşoară activitatea, oferindu-i servicii particularizate în funcție de acel context. Unul dintre elementele cele mai importante în determinarea contextului şi furnizarea acelor servicii particularizate este aflarea poziției unui utilizator într-un anumit spațiu fizic. ***Deşi problema acesta a***

localizării a fost studiată în ultimii ani nu există soluţii standardizate, bine puse la punct, care să fie uşor de implementat şi exploatat. De aceea, în lucrarea de faţă este elaborată o metodă de localizare care să corespundă acestor cerinţe, pe care am denumit-o „metoda poligonului".

Un alt aspect critic în cazul dispozitivelor mobile este autonomia. Aici resursele de energie sunt limitate şi dacă nu sunt exploatate în mod judicios consumul de energie prea mare poate să ducă la indisponibilitatea frecventă a unui anumit dispozitiv, acest lucru putând avea efecte negative şi asupra altor dispozitive aflate în reţea, dispozitive cu care acesta avea stabilite conexiuni. Într-o situaţie de genul aceste este de dorit ca reţeaua să fie înştiinţată de iminenţa unui astfel de eveniment.

Conceptul de resursă poate fi extins şi la capacitatea de transfer a unui reţele wireless, această resursă fiind distribuită la nivelul BSS-urilor (Basic Service Set), fiecare BSS fiind construit în jurul unui AP. Şi acest tip de resursă este limitat, el trebuind să fie împărţit între toţi utilizatorii reţelei. ***Gestionarea capacităţii de transfer globale a unei reţele 802.11 şi distribuirea acesteia în funcţie de necesităţile şi nivelul de prioritate al diverşilor utilizatori este un alt aspect vizat în lucrarea de faţă.***

Toate aceste aspecte legate de calitatea unei conexiuni, ratele de transfer suportate, autonomia unui dispozitiv mobil, localizarea unui dispozitiv mobil, oferirea unor informaţii referitoare la un anumit tip de context au fost prevăzute ca şi mecanisme sau servicii implementate într-un sistem unificat de gestiune a resurselor unei reţele wireless, denumit ***UFRM (Unified Framework for Resources Management).***

Structura tezei

Diagrama de mai jos redă structura pe capitole a tezei şi arată înlănțuirea logică a acestora, furnizând câteva elemente cheie referitoare la fiecare capitol în parte.

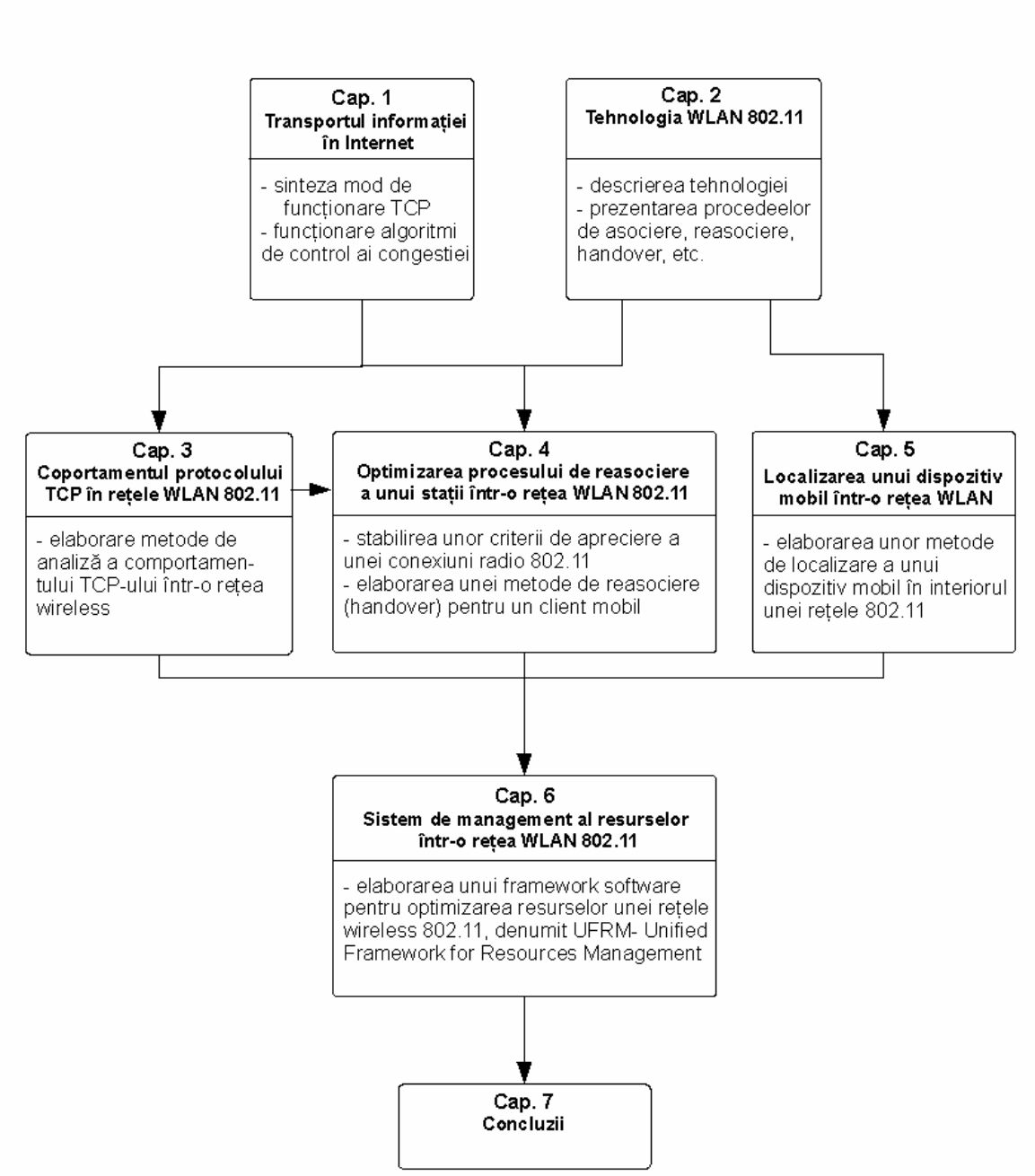


Figura 1 Structura pe capitole a tezei

Organizarea pe capitole este următoarea:

Capitolul 1

Sunt descrise principalele mecanisme care intervin atunci când informaţia este transportată printr-o reţea care are la bază stiva de protocoale TCP/IP, accentul punându-se pe mecanismele de control ale congestiei implementate de TCP. Aceste mecanisme sunt esenţiale pentru a asigura o bună funcţionare a reţelei, cu condiţia ca toţi participanţii la trafic să ţină cont de indicatorii ca semnalează apariţia fenomenului de congestie undeva în reţea şi să ia masurile necesare pentru a nu contribui la agravarea fenomenului. Pentru ca aceste măsuri sa fie respectate de toţi participanţii la trafic s-a folosit soluţia implementării acestor mecanisme la nivelul protocolului TCP.

Capitolul 2

Acest capitol face o prezentare a tehnologiei IEEE 802.11, pentru a oferi informaţiile necesare înţelegerii diverselor soluţii prezentate în această lucrare, soluţii care vizează îmbunătăţiri ai unor parametrii de funcţionare într-o reţea WLAN. Standardul 802.11 prezintă unele similitudini cu standardul GSM în ceea ce priveşte organizarea reţelei într-o arhitectură de tip celular. În interiorul unei celule WLAN, mai mulţi clienţi vor fi deserviţi de un AP (Access Point). Standardul prevede la nivelul Legătură de Date mecanisme de asociere, dezasociere sau reasociere, atunci când se trece dint-o celulă în alta (adică de la un AP la altul).

Capitolul 3

Protocolul TCP implementează o serie de mecanisme pentru controlul congestiei. Aceste mecanisme au fost proiectate înaintea răspândirii pe scară largă a comunicaţiilor wireless. Caracteristic conexiunilor wireless, faţă de alte tipuri de conexiuni este aceea că ele sunt mult mai vulnerabile la perturbaţii provenite din mediul exterior şi de aceea rata de apariţie a erorilor este mai mare decât în celelalte cazuri. În cazul unor conexiuni sigure, la care probabilitatea de apariţie a erorilor este mică, singura cauză de pierdere a pachetelor este fenomenul de congestie. Ca reacţie la apariţia acestui fenomen, TCP-ul declanşează nişte proceduri prin care este diminuată rata de transfer, pentru a reduce numărul de pachete care urmează să traverseze zona congestionată. În cazul conexiunilor wireless, pierderile de pachete şi întârzierile sunt interpretate ca o apariţie a congestiei şi deci se reduc ratele de transfer practicate între sursă si destinaţie.

Pentru a putea evalua gradul în care acest fenomen afectează o comunicație TCP desfăşurată peste o conexiune wireless, au fost elaborate două metode de analiză a comportamentului TCP-ului. Scopul acestor metode este acela de a putea aprecia calitatea unei conexiuni wireless. Cu cât o conexiune wirelss este mai bună cu atât mecanismele de control ale congestiei sunt mai rar invocate.

Capitolul 4

Pentru ca utilizatorul să beneficieze deplin de avantajul mobilității oferit de o conexiune wireless, trebuie să i se asigure acoperire în diversele zone din interiorul clădirii unde îşi desfăşoară activitatea. Deoarece raza de acoperire asigurată de o conexiune wireless de tip 802.11 este doar de câțiva zeci de metri în interiorul unei clădiri, se impune ca pentru a asigura o acoperire completă este necesar, ca în cazul deplasării dispozitivului mobil să se ofere un mecanism de comutare de la un *access point* la altul. În cadrul standardului 802.11 acest mecanism este implementat la nivel MAC. Din păcate, în implementarea existentă procedeul de comutare (handover) se declanşează când conexiunea fizică între client si AP se întrerupe. Etapele care urmează, adică identificarea unui nou AP disponibil, negocierea realizării unei noi conexiuni, etc., vor avea ca efect prelungirea duratei cât conexiunea la nivel legătură de date este întreruptă, ceea ce are repercusiuni asupra conexiunilor inițiate de protocoalele de pe nivele superioare. În acest capitol este prezentată o metodă care a fost proiectată pentru a obține o îmbunătățire a procesului de reasociere, adică de comutare a unui client de la un *access point* la altul, mecanism consacrat in telefonia mobilă sub denumirea de handover. Acest mecanism este necesar pentru a menține o conexiune de date existentă în parametrii optimi, ceea ce se traduce printr-o pierdere a cât mai puține pachete precum şi generarea de întârzieri cât mai mici. În situația ideală, o conexiune stabilită de către protocoalele de pe nivele superioare nu ar trebui să sufere o depreciere a calității in urma procesului de handover.

Capitolul 5

Descrie o metodă proprie, de localizare a unui dispozitiv mobil într-o rețea 802.11, metodă bazată pe citirea puterii semnalului provenit de la *acces point-uri* şi pe care am denumit-o *metoda poligonului*. Față de alte metode existente, aceasta se remarcă prin simplitate în implementare şi o precizie satisfăcătoare.

La elaborarea ei s-au realizat serii extinse de măsurători pentru a pune în evidență impactul pe care îl au diversele tipuri de AP-uri asupra puterii semnalului recepționat precum şi influența distanței asupra semnalului recepționat. S-a adaptat formula de

propagare a semnalului radio în spațiul liber pentru a obține rezultate care să minimizeze efectul perturbațiilor asupra semnalului. Un alt aspect vizat a fost punerea în evidență a modului de variație a semnalului atunci când dispozitivul mobil părăseşte încăperea şi a modului de variație atunci când în încăpere sunt prezente persoane.

Țelul urmărit în cazul procedeului de localizare elaborat în această teză, a fost acela de a obține un raport favorabil între resursele alocate de sistem pentru a rula algoritmii de localizare şi eficiența acestor algoritmi.

Capitolul 6

În acest capitol este descris un sistem care a fost conceput pentru a facilita optimizarea resurselor unor dispozitive, care interacționează folosind ca infrastructură o rețea de tip wireless 802.11. Pentru implementarea unui astfel de sistem s-a făcut o analiză din mai multe perspective:

- comportamentul protocoalelor implicate
- resursele care trebuie optimizate
- aplicațiile care pot fi rulate pe dispozitivele mobile
- modalitatea de culegere a informațiilor referitoare la contextul în care se desfăşoară o anumite activitate

Sistemul a fost gândit sub forma unui framework sub denumirea de **Unified Framework for Resources Management (UFRM)**, rolul lui fiind acela de a superviza activitățile clienților unei rețele wireless 802.11 şi de a gestiona resursele rețelei. La implementarea lui au fost folosite mecanismele elaborate în capitolele 3, 4 şi 5.

Capitolul 7

Capitol de concluzii.

Capitolul 1 Transportul informației în Internet

1.1 Introducere

Pe la mijlocul anilor 60', în America, puternicele calculatoare ale vremii respective (*mainfraim*-uri) aparţineau fie unor universităţi fie altor instituţii, nu puteau comunica între ele, deoarece fabricanţii acestor calculatoare folosea propriul software de reţea. În acea vreme sigurele reţele de care putem discuta, erau formate din terminale conectate la aceste *mainfraim*-uri şi care rulau un software proprietar. În 1957, Uniunea Sovietică reuşeşte să lanseze primul satelit artificial (Sputnik). Ca răspuns la acest eveniment, în cadrul Departamentului de apărare al SUA (DoD), ia naştere Advenced Reasearch Projects Agency (ARPA), care trebuia să recupereze handicapul apărut.

În acest context, ARPA a devenit interesată de găsirea unei modalităţi de interconectare a calculatoarelor. În felul acesta ar fi devenit mult mai facil schimbul de informaţii cu caracter ştiinţific şi nu numai.

În 1967, la întrunirea celor de la Association for Computing Machinery (ACM), ARPA îşi prezintă proiectul pentru o mică reţea care să interconecteze calculatoare, proiect care primeşte numele ARPANET. Ideea iniţială era ca fiecare calculator, sa fie conectat la un IMP (Interface Message Processor), iar mai multe IMP-uri să fie conectate intre ele. Această soluţie rezolva problema interconectării calculatoarelor produse de firme diferite. În figura 1-1 este prezentată o fotografie istorică reprezentând un IMP.

Figura 1-1 Interface Message Processor (IMP)

Astfel, ARPANET ia naştere în 1969 prin conectarea a patru universităţi: the University of California at Santa Barbara (UCSB), the University of California at Los Angeles (UCLA), Stanford Research Institute (SRI) şi the University of Utah. Software-ul care asigura comunicarea între noduri era numit Network Control Protocol (NCP). În figura 1-2 este înfăţişată diagrama originală care stabilea modul de conectare al celor patru noduri:

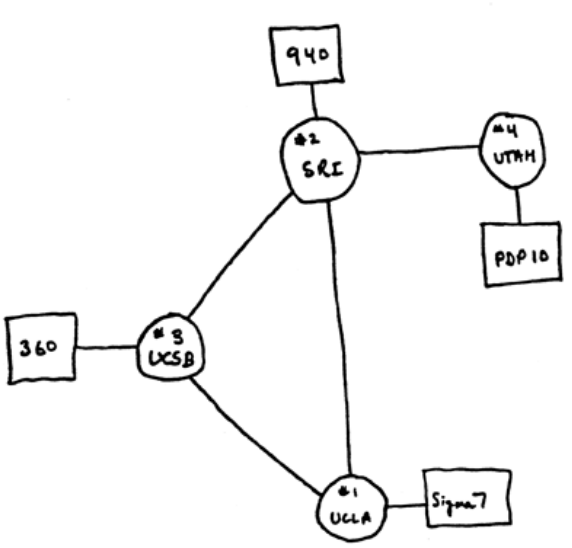


Figura 1-2 Diagrama originală care arată modul de conectare al primelor noduri din ARPANET

În 1971 existau deja 15 noduri, asigurând conexiuni pentru 23 de calculatoare.

În 1972, doi cercetări din grupul celor care lucrau la dezvoltarea ARPANET-ului, Vint Cerf şi Bob Kahn, lansează ideea conectării mai multor reţele între ele, luând astfel naştere Interneting Project. Problemele care trebuiau rezolvate erau cele legate de dimensiunea pachetelor, tipul interfeţelor precum şi ratele de transfer. Cei doi propun ideea unui dispozitiv denumit *gateway*, care să realizeze trecerea pachetelor dintr-o reţea în alta.

Înainte de a merge mai departe, să ne întoarcem câţiva ani în urmă, în anul 1961 când Leonard Kleinrock publică primul articol referitor la o modalitate noua de transfer a datelor între calculatoare, numită comutare de pachete. Această tehnică contrasta cu clasica tehnologie a comutării de circuite folosită în sistemul telefonic. În 1964 publică o carte numită „Communication Nets" prin care încearcă să convingă ca noua sa idee este potrivită pentru a realiza comunicaţia între calculatoare.

Astfel, în 1969 când ia fiinţă ARPANET-ul, software-ul care făcea ca reţeaua să funcţioneze, denumit Network Control Protocol, implementa principiile comutării de pachete.

În 1973, Vint Cerf şi Bob Kahn îmbunătăţesc protocolul pentru a asigura conexiune punct la punct. Noua versiune primeşte numele de TCP. Noul protocol îngloba concepte precum încapsularea pachetelor şi funcţionalităţi de *gateway*. Apare şi ideea de a transfera responsabilitatea de control a erorilor de la IMP la maşina gazdă.

În 1977 se ia decizia de a „sparge" TCP-ul în două, creându-se încă un protocol numit Internet Protocol (IP). Acest nou protocol va avea ca responsabilitate dirijarea pachetelor, în timp ce TCP va realiza funcţiile de segmentare şi reasamblare a pachetelor precum, control al fluxului de date şi de detecţie a erorilor. Astfel ia naştere stiva de protocoale TCP/IP.

1.2 Protocoale şi standarde

Prin protocol se înţelege un set de reguli pe care doua entităţi care comunică între ele trebuie să le respecte pentru ca schimbul de informaţii să poată avea loc. Prin entitate se înţelege orice fel de dispozitiv capabil să transmită şi să recepţioneze informaţii. Protocolul stabileşte ce se va comunica, cum se va comunica şi momentele de timp când se va comunica. Un protocol este caracterizat prin trei elemente: sintaxă, semantică şi sincronizări.

Standardele sunt esenţiale pentru a asigura interoperabilitatea protocoalelor la nivel mondial. Ele se împart în două categorii: standarde *de facto* şi standarde *de jure*. Standardele de facto nu au fost iniţial aprobate de nici o instituţie ci ele au fost adoptate datorită răspândirii şi utilizării lor pe scară largă. In schimb, standardele de jure sunt avizate din momentul creării lor de către un organism internaţional abilitat.

La începutul anilor ’70 marii producători de echipamente de calcul de la acea vreme aveau propriile soluţii de arhitecturi de reţea: IBM a dezvoltat SNA-ul, iar DEC propunea DNA-ul. Marele neajuns al acestor implementări era acela că nu permitea interconectarea cu alte sisteme, decât dacă producătorii cădeau de acord şi dezvoltau o extensie a acelei arhitecturi care să permită acest lucru. Ca o reacţie la această situaţie ISO(International Standards Organisation) a început în acea perioadă elaborarea unui model care să permită interconectarea „sistemelor deschise”.

Când se pune problema proiectării unui sistem de comunicaţii digital există doua probleme fundamentale de care trebuie să se ţină seama:

- nici o tehnologie nu poate îndeplini toate constrângerile
- utilizatorii doresc un mod universal de interconectare

Pentru a reduce complexitatea programelor care trebuie să ruleze într-o reţea s-a recurs la organizarea acestora sub forma unei ierarhii, această ierarhie căpătând denumirea de stivă de protocoale. Ea se prezintă sub forma unor straturi, rolul fiecărui strat este acela de a oferi servicii stratului imediat superior [Tan96]. În felul acesta un anumit strat nu trebuie să cunoască detaliile de implementare ale straturilor inferioare. Într-o astfel de ierarhie un anumit nivel N de pe o maşină va dialoga doar cu nivelul N de pe altă maşină. Între două niveluri adiacente există o interfaţă care defineşte ce servicii oferă nivelul inferior celui superior. Acest mod de organizare este avantajos şi atunci când se aduc modificări unui anumit nivel. Aceste modificări vor putea fi făcute fără ca să impună modificări şi în cadrul celorlalte niveluri cu condiţia ca interfaţa dintre ele să fie clar definită.

Serviciile specifică ce operaţii poate să îndeplinească un anumit nivel, dar nu indică cum anume să fie implementate aceste operaţii. Aceasta este problema protocoalelor care vor fi elaborate pentru a funcţiona pe acel nivel. Deci protocoalele sunt implementări ale serviciilor. Protocoalele pot fi modificate, dar serviciile trebuie să rămână neschimbate.

Există două categorii de servicii pe care unele niveluri le pot oferi nivelului adiacent superior şi acestea sunt:

- servicii orientate pe conexiune
- servicii fără conexiuni

Serviciul orientat pe conexiune numit şi circuit virtual are ca model sistemul telefonic. Reţeaua telefonică este o reţea de tipul cu comutare de circuite, adică între doi abonaţi care poartă o convorbire, centrala stabileşte un circuit fizic dedicat. În cazul unei reţele de date a fost adoptată denumirea de circuit virtual prin analogie cu modul de funcţionare a unei reţele telefonice. Mai întâi trebuie stabilită o conexiune, iar datele care vor fi transmise între sursă şi destinaţie vor urma toate acelaşi traseu. La încheierea comunicaţiei conexiunea trebuie eliberată.

În contrast, serviciul fără conexiuni este modelat pe baza serviciului poştal. Fiecare pachet (datagrama) va circula spre destinaţie independent de celelalte pachete, ele conţinând adresa completă a destinaţiei. În cazul serviciului cu circuit virtual este suficient ca doar primul pachet să conţină adresa destinaţiei, ea fiind folosită la stabilirea circuitului, pentru restul pachetelor fiind suficient să se indice doar numărul de identificare al circuitului virtual.

Modelul OSI (Open System Interconnection)

Modelul a fost propus în anul 1983 de către International Organization for Standardization, şi care în traducere liberă ar putea fi numit *Model de referinţă pentru sisteme de telecomunicaţii deschise interconectării de echipamente sau subsisteme neomogene din punct de vedere hardware şi sofware* [Str00]. Scopul acestui set de standarde era să definească o modalitate uniformă de conectare a unor sisteme cu caracteristici diferite. Modelul prezintă la nivel de principiu serviciile care trebuie asociate fiecărui nivel neimpunând însă soluţii concrete de implementare a acestora. Această implementare se va materializa prin protocoalele alese de proiectanţi. Modelul OSI prezintă 7 nivele, după cum se vede şi în figura 1-3.

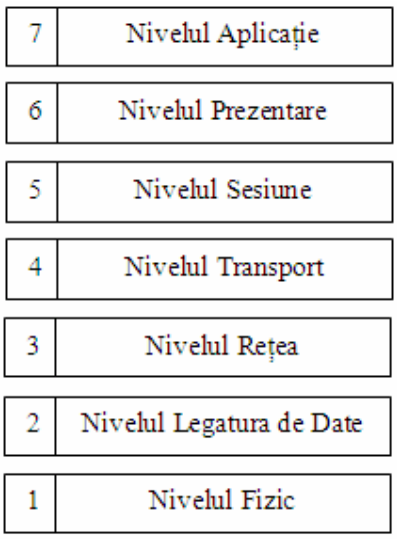


Figura 1-3 Modelul OSI

Pe scurt, atribuţiile şi caracteristicile fiecărui nivel sunt următoarele [Str00]:

Nivelul Fizic

Serviciul pe care acest nivel îl pune la dispoziţie este acela de a transporta un şir de biţi de la un capăt la celălalt al unei legături fizice. Această legătură fizică poate

fi reprezentată prin fire metalice, fibre optice sau canale radio. Nivelul Fizic va trebui să stabilească legătura între cele două puncte de capăt, să o menţină şi să o întrerupă atunci când se impune acest lucru.

Nivelul Legătură de Date

În lumea reală, în general, orice canal de comunicaţii poate fi afectat de perturbaţii. Aceasta însemnă că atunci când Nivelul Fizic transportă datele acestea pot fi afectate de erori. Pentru a realiza o comunicaţie sigură între două puncte a fost necesar să se introducă Nivelul Legătură de Date care va fi responsabil cu detecţia şi eventual corecţia erorilor care pot apărea pe Nivelul Fizic. Astfel, Nivelul Legătură de Date face uz de serviciul oferit de Nivelul Fizic, transformând acest serviciu dintr-unul nesigur într-unul sigur.

Nivelul Legătură de Date

Organizează datele care trebuie trimise sub forma unor cadre şi acestea sunt predate Nivelului Fizic spre a fi transportate. Pe lângă datele propriu-zise un cadru conţine câmpuri cu informaţii necesare pentru a transforma Nivelul Fizic în unul sigur. Unele câmpuri pot conţine numere de secvenţă, pentru a verifica că datele sosesc în ordinea corectă, iar alte câmpuri pot conţine sume de control pentru detecţia erorilor.

Modul cum sunt manipulate aceste cadre depinde de protocolul care este implementat în cadrul Nivelului Legătură de Date.

La acest nivel trebuie să se practice şi un control al fluxului de date, adică să se rezolve problema comunicării între două noduri dintre care unul este mai lent şi altul mai rapid.

Nivelul Reţea

Acest nivel trebuie să îndeplinească sarcina mai complexă de a transporta date între două noduri neadiacente, adică informaţia va trebui să tranziteze noduri intermediare.

Pentru a-şi îndeplini funcţia Nivelul Reţea organizează datele sub forma unor pachete denumite PPDU (Protocol Packet Data Unit). Şi Nivelul Transport organizează sub o anumită formă datele pe care le manipulează, acest mod de organizare numindu-se TPDU (Transport Protocol Data Units).

Când doreşte să transmită date, Nivelul Transport apelează la serviciile oferite de Nivelul Reţea. Acesta preia TPDU-ul şi îl introduce în câmpul de date al unui PPDU. La rândul său Nivelul Reţea face apel la serviciul oferit de Nivelul Legătură de Date care preia pachetul şi îl introduce în câmpul de date al frame-urilor pe care el le foloseşte pentru a transporta informaţia. Acest mecanism este pus în evidenţă de figura 1-4. Spunem că are loc o încapsulare a datelor. La rândul sau Nivelul Legătură de Date face uz de serviciul oferit de Nivelul Fizic pentru a transporta frame-ul către următorul nod.

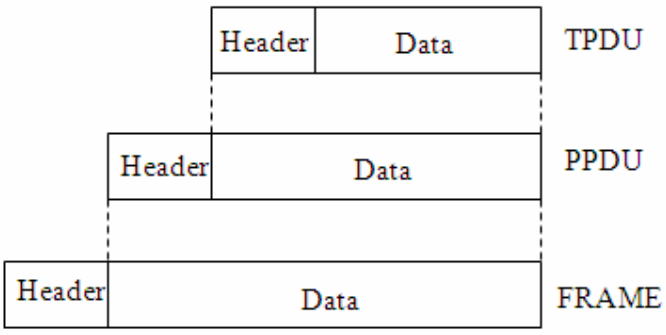


Figura 1-4 Modul de încapsulare al datelor practicat în modelul OSI

Cu alte cuvinte Nivelul Reţea este responsabil de dirijarea pachetelor de la sursă la destinaţie trecând prin noduri intermediare. Pentru a putea îndeplini această funcţie de dirijare, pachetul folosit de Nivelul Reţea este prevăzut cu un câmp in partea de Header, care reprezintă adresa nodului destinaţie. Când datele tranzitează nodurile intermediare, este nevoie ca de fiecare dată să se verifice valoarea acestui câmp de adresă. Pentru a se realiza acest lucru este nevoie să se recurgă mai întâi la extragerea PPDU-ului din cadrul frame-ului. Dacă adresa este cea a nodului curent atunci din câmpul de date al pachetul este extras TPDU-ul şi este furnizat Nivelului Transport. Dacă datele nu sunt destinate nodului curent atunci pachetul este încapsulat din nou într-un frame şi se apelează la serviciul oferit de Nivelul Fizic pentru a-l transporta mai departe către destinaţie.

Nivelul Transport

Funcţia pe care trebuie să o îndeplinească Nivelul Transport este de a oferii servicii Nivelului Sesiune. Aceasta presupune faptul de a accepta date de la Nivelul Sesiune. Dacă lungimea datelor pe care Nivelul Transport trebuie să le transmită depăşeşte lungimea câmpului de date din interiorul unui pachet, atunci aceste date trebuie fragmentate. Rămâne în sarcina Nivelului Transport de a se asigura că pachetele ajung în ordine la destinaţie şi că nici unul nu s-a pierdut. Am putea spune şi despre Nivelul Transport că transformă serviciul oferit de Nivelul Reţea dintr-unul nesigur în unul sigur.

Nivelul Transport spunem că este de tipul capăt la capăt deoarece o instanţă a protocoalelor de pe acest nivel trebuie să existe doar la nivelul nodurilor care comunică între ele, pe când în cazul Nivelului Reţea, protocoalele de pe acest nivel trebuie să se regăsească în fiecare nod intermediar de pe traseul parcurs de date. Aceste principii sunt puse în evidenţă de figura 1-5.

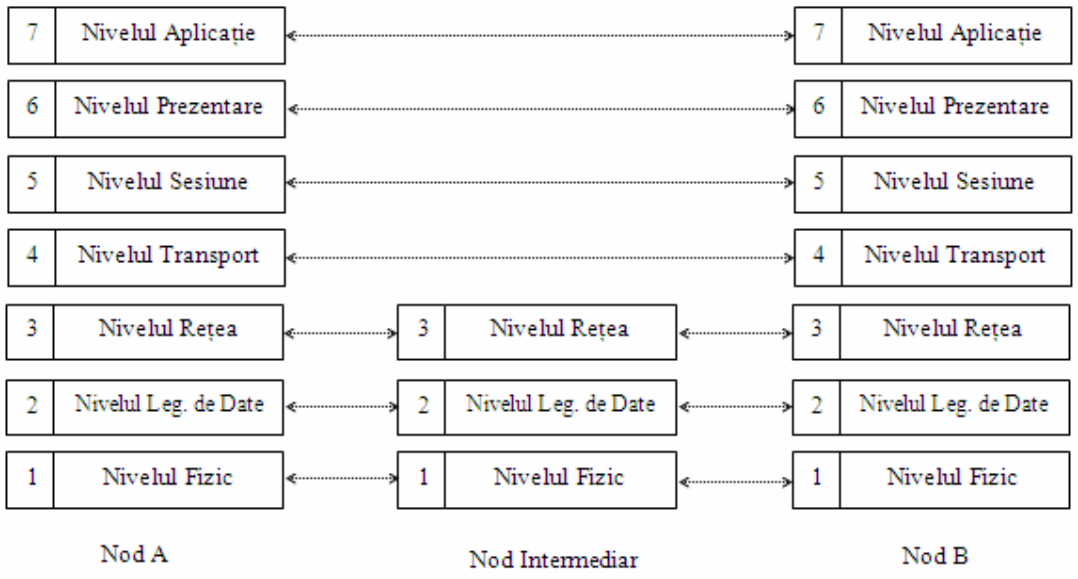


Figura 1-5 Comunicarea între diversele niveluri ale modelului OSI

Nivelul Sesiune

Nivelul sesiune a fost gândit pentru a permite utilizatorilor să stabilească sesiuni, adică o modalitate de sincronizare şi de control al dialogului între două procese care comunică la distanță. De asemenea acest nivel mai are în atribuții restabilirea legăturii în cazul unor întreruperi.

Nivelul Prezentare

Are de a face cu sintaxa şi semantica informațiilor transmise. Acest nivel procesează informațiile pentru a le face compatibile între două aplicații diferite, asigurând o independență între aplicații şi Nivelul Transport. Operațiile tipice pe care acest nivel le realizează sunt de conversie, formatare, criptare şi compresie.

Nivelul Aplicație

Conține toate protocoalele şi aplicațiile care interacționează direct cu utilizatorul oferind o interfață pentru accesul acestuia la rețea.

Protocolul X.25

X.25 este protocolul care se apropie cel mai bine de modelul OSI. El a fost dezvoltat la inițiativa ITU (International Telecommunications Union), devenind popular în special în Europa, fiind adoptat de companiile de telecomunicații.

Protocolul X.25 este de fapt un set de standarde care acoperă primele trei niveluri din modelul OSI [Str00, CT92]. Aceste standarde se referă la legătura dintre un terminal sau un calculator şi o rețea cu comutare de pachete. Pentru Nivelul Fizic există protocolul X.21, iar pentru Nivelul Legătură de date protocolul HDLC (High Level Data Link Control).

Pentru Nivelul Rețea, X.25 specifică modul de conectare al unui terminal (DTE - Data Circuit Terminating Equipment), cu un echipament terminator al unei

rețele (DCE – Data Circuit Terminating Equipment). Prin acest dispozitiv DCE se asigură accesul la rețeaua cu comutare de pachete, după cum este ilustrat în figura 1-6.

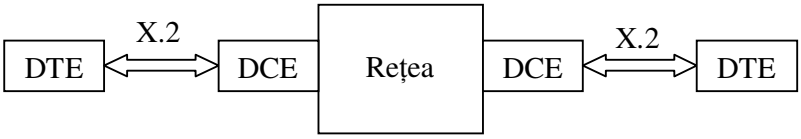


Figura 1-6 Accesul la rețea al unui dispozitiv, folosind protocolul X.25

Nu vom insista cu detalii de implementare pentru că nu acesta este scopul acestei expuneri. Pentru informații suplimentare a se consulta bibliografia [CT92].

1.2.1 Stiva de protocoale TCP/IP

După cum s-a văzut şi din partea introductivă a acestui capitol, stiva TCP/IP a evoluat o dată cu Internetul, iar protocoalele din cadrul ei intră în categoria standardelor *de facto.*

Nu toate nivelurile din modelul OSI se regăsesc şi in modelul TCP/IP după cum arată şi figura de mai jos. Nivelul Prezentare şi Nivelul Sesiune nu există, Nivelul Fizic împreună cu cel de Date, sunt cuprinse într-unul singur numit Nivel de Acces la Mediu, iar Nivelul Rețea poartă denumirea de Nivelul Internet, aşa cum se observă şi în figura 1-7.

Nivelul de Acces la Mediu

La acest nivel stiva TCP/IP nu defineşte un anume protocol. Ideea este de a suporta toate standardele de pe acest nivel (ex. Ehernet, Frame Relay, ATM, rețele bazate pe fibră optică, rețele fără fir, etc.)

Nivelul Internet

Protocolul care funcționează pe acest nivel este IP-ul. Tipul de serviciu oferit de acest protocol este de tipul comutare de pachete. Atunci când nivelul superior face apel al serviciile oferite de acest nivel pentru a transmite date, acestea vor fi încapsulate în pachete, iar IP nu ține cont de faptul că pachetele au sau nu legătură unele cu altele şi le dirijează în mod independent unele de celelalte.

Nivelul Transport

Pot exista două tipuri de servicii pe care Nivelul Transport le poate oferi. Unul dintre servicii este orientat pe conexiune, fără erori şi care furnizează octeții în ordinea în care au fost trimişi. Celălalt tip de serviciu nu oferă nici o garanție în privința ordinii în care au fost trimise datele. Primul tip de serviciu este oferit de protocolul TCP, al doilea de către protocolul UDP.

Nivelul Aplicație

La acest nivel se găsesc toate aplicațiile şi protocoalele care asigură accesul utilizatorului la resursele rețelei.

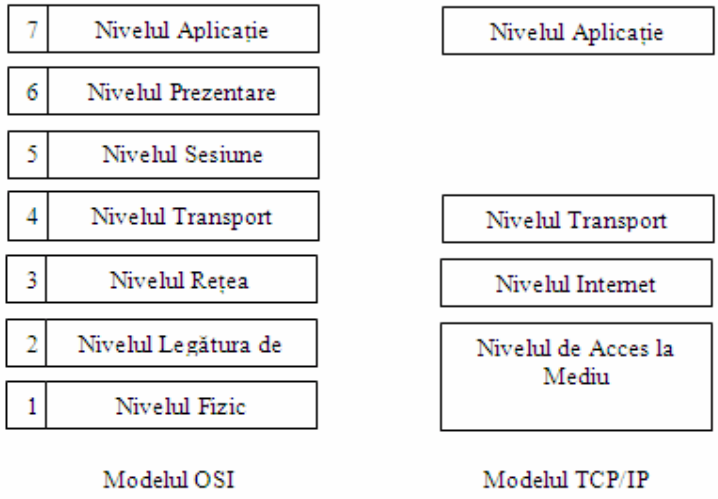


Figura 1-7 Modelul OSI vs. stiva TCP/IP

1.2.2 Concluzii privind o comparaţie între modelul OSI şi stiva TCP/IP

După cum am văzut, modelul OSI este unul conceptual el nespecificând care anume protocoale să fie utilizate, de aceea vom folosi ca şi exemplu de implementare, protocolul X.25 iar discuţia se va purta între X.25 şi TCP/IP.

Când facem această analiză comparativă vom lua în discuţie criterii intrinseci care ţin de modul de implementare şi funcţionare al protocoalelor şi criterii extrinseci care ţin de momentul apariţiei lor, de felul cum au fost percepute de comunitatea ştiinţifică, etc.

Soluţia adoptată de X.25 pentru gestionarea unei comunicaţii între două noduri este de tipul „*hop-by-hop*”. Când cele două noduri vor să stabilească o conexiune, mai întâi este setat un circuit virtual care presupune alocarea resurselor de-a lungul căii de urmat, iar datele care urmează a fi transmise urmează toate această cale stabilită în faza de iniţiere a conexiunii. Soluţia adoptată în cazul IP-ul este total diferită: nu se stabileşte în prealabil nici o cale pentru date ci fiecare pachet de date va fi dirijat în mod independent de celelalte.

Cele două stive diferă prin modul de tratare a erorilor şi modul de gestionare a stării de congestie. În TCP se practică un control al erorilor de tip „*end to end*”, mergându-se pe ideea că reţeaua nu este întru totul sigură şi de aceea cel mai bine este ca verificările să se facă la capete, considerându-se că dacă s-ar păstra şi verificările de pe parcurs acestea ar fi redundante. În X.25 se practică o verificare a integrităţii datelor în fiecare nod intermediar şi la fiecare nivel al stivei. Singurul avantaj al acestei abordări este că erorile vor fi depistate mai repede. În schimb necesitatea de a transmite mesaje de confirmare între fiecare două noduri adiacente nu face decât să conducă la o încărcare inutilă a reţelei. Acest mod de abordare era potrivit în perioada anilor ’70 când mediile de comunicaţie nu erau foarte sigure.

În ceea ce priveşte controlul congestiei aici X.25 prezintă un avantaj care rezidă din maniera în care se realizează o conexiune: mai întâi sunt alocate resursele, iar apoi are loc schimbul de date. Deci congestia s-ar manifesta prin incapacitatea de a stabili la un moment dat o conexiune (ceea ce nu este propriu–zis o congestie) şi nu

prin degradarea în timp a parametrilor (ex. rată de transfer, întârzieri) în care se desfăşoară o comunicaţie.

O altă diferenţă între cele două abordări este în ceea ce priveşte starea unei conexiuni. X.25 se încadrează în categoria *stateful,* iar TCP/IP poate fi caracterizat prin termenul de *state less*. Aceasta are legătură cu modul de realizare al unei conexiuni. În cazul lui X.25 toate nodurile intermediare trebuie să memoreze starea unui anumit circuit virtual. În cazul în care unul dintre aceste noduri „cade", transmisia se întrerupe şi trebuie restabilită. În cazul IP-ului am văzut că nu se stabileşte în prealabil nici un fel de traseu, fiecare pachet fiind dirijat în mod independent. În cazul în care un nod ar cădea, pur şi simplu pachetele sunt dirijate pe alt traseu fără să se întrerupă comunicaţia.

Rezumând cele spuse mai sus putem concluziona că în cazul lui X.25 plasarea „inteligenţei" s-a făcut la nivelul reţelei care este văzută ca un sistem complex şi autonom la care se pot conecta noduri, dar care au o contribuţie minimă la operaţiile realizate de către reţea. În schimb, în cazul TCP-ului, nodurile terminale participă la aproape toate operaţiile realizate de reţea.

Un motiv pentru care modelul OSI nu s-a bucurat de succes a fost acela că în momentul apariţie lui, TCP/IP-ul era destul de răspândit fiind asociat deja cu sistemul de operare UNIX, firmele oferind deja produse pe baza lui. În acest moment prea puţine firme au fost dispuse să facă noi învestiţii care să suporte o nouă stivă de protocoale, iar când totuşi au fost realizate anumite implementări acestea au fost greoaie şi cu multe defecte.

O altă deosebire esenţială între cele două modele este aceea că în cazul OSI mai întâi a fost elaborat modelul, care a fost foarte bine descris, introducând trei concepte fundamentale şi făcând clar delimitările între ele. Acestea sunt:

a) servicii
b) interfeţe
c) protocoale

Serviciul defineşte funcţiile pe care le realizează nivelul, **interfaţa** spune cum anume se face accesul la aceste funcţii, iar **protocolul** reprezintă implementarea acestor funcţii.

În cazul TCP/IP –ul nu a existat această delimitare. De fapt, aici protocoalele au apărut înaintea modelului. S-a văzut cum TCP/IP-ul a apărut ca şi parte a sistemului de operare UNIX, fiind în mod natural însuşit de către comunitatea ştiinţifică. Doar mai târziu s-a încercat şi asocierea unui model acestor protocoale.

Poate unul dintre cel mai mari neajunsuri ale modelului TCP/IP este că pentru Nivelul Fizic şi Nivelul Legătură de Date, el defineşte un singur nivel numit Nivel de Acces la Mediu, fără să descrie serviciile care trebuie să le îndeplinească acest nivel şi nici interfaţa de acces la el. Singurul lucru care se precizează este că *host*-ul trebuie să se conecteze printr-un anumit protocol la mediul fizic. Deci Nivelul Internet, reprezentat prin protocolul IP nu are nici un fel de informaţii despre ce se întâmplă sub el.

1.3 Principalele protocoale din stiva TCP/IP

În acest capitol vor fi prezentate pe scurt două protocoale din stiva TCP/IP pentru a servi ca bază de discuţie pentru capitolele următoare.

1.3.1 Protocolul IP

Orice calculator conectat la Internet şi care vrea să poată comunica cu orice alt calculator conectat de asemenea la Internet, trebuie să ruleze o instanţă a protocolului IP. După cum s-a văzut în rândurile de mai sus, protocolul IP [RFC791] este independent de structura Nivelului Legătură de Date, el putând rula peste orice protocol de pe nivelul imediat inferior. Pentru a putea transporta informaţia protocolul face uz de nişte structuri de date numite pachete sau datagrame.

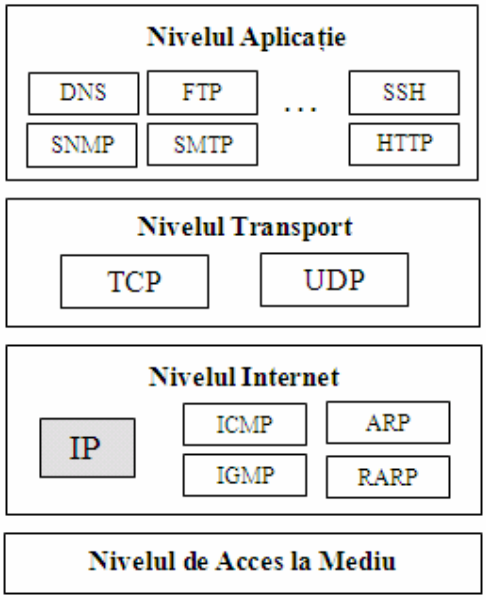


Figura 1-8 Poziţia protocolului IP în interiorul stivei TCP/IP

Atunci când protocoalele de pe Nivelul Transport doresc să transmită un mesaj, ele vor face uz de serviciile oferite de Nivelul Internet (figura 1-8), adică de protocolul IP. De exemplu dacă se doreşte transmiterea unui mesaj care depăşeşte dimensiunea maximă a unui pachet IP, atunci mesajul este fragmentat şi pachetele care rezultă în urma fragmentării vor fi trimise de către IP în mod independent fără a considera că acestea au legătură unele cu altele. Este astfel posibil ca unele pachete să se piardă sau să ajungă în altă ordine decât au fost trimise. În cadrul protocolului IP nu există nici un mecanism pentru a solicita o eventuală retransmisie a pachetelor pierdute sau de a ordona pachetele recepţionate. Există o sumă de control care se calculează atunci când este transmis un pachet, iar în cazul în care un pachet este afectat de eroare este înlăturat din reţea şi se emite un mesaj de eroare către nodul sursă, dar acesta este singurul mecanism prevăzut în cadrul protocolului pentru detecţia erorilor. Restul problemelor enunţate mai sus vor cădea în seama protocolului TCP, Nivelul Reţea fiind însărcinat doar cu dirijarea pachetelor. Pentru a funcţiona,

este necesar ca o instanţă a acestui protocol să existe pe fiecare maşină implicată în transferul de date de la un nod sură la un nod destinaţie.

Structura unei pachet este ilustrată in figura 1-9 [Ste94, Com00].

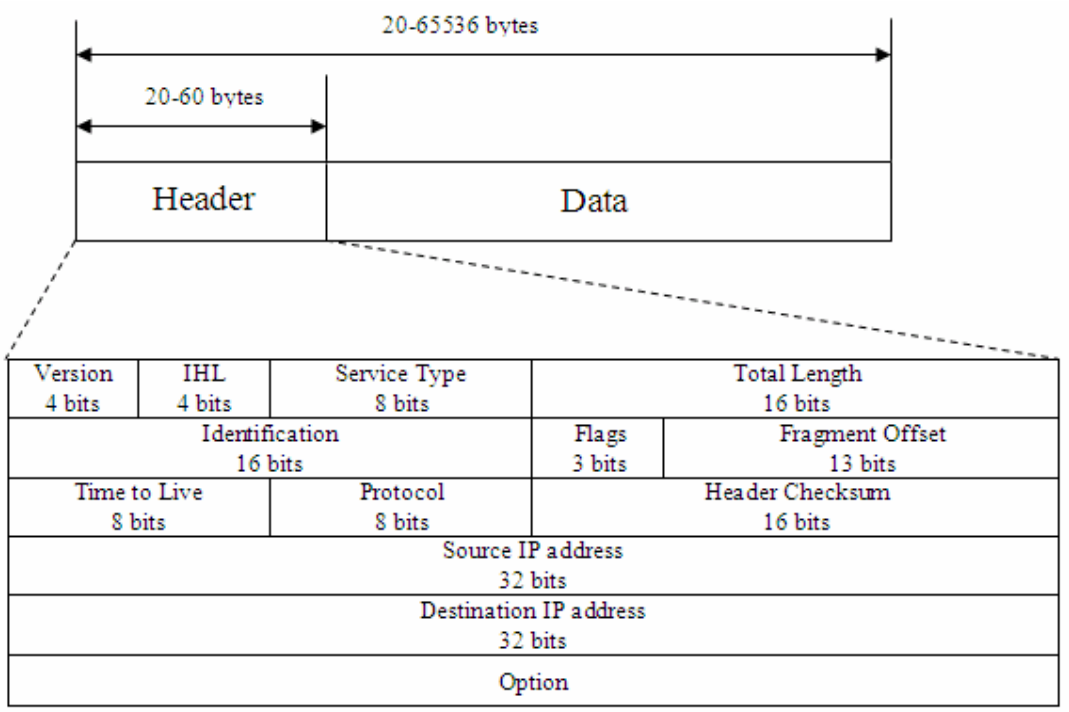


Figura 1-9 Structura unui pachet IP

În cele ce urmează vom discuta succint semnificaţia fiecărui câmp din structura header-ului:

Version: Versiunea prezentată în acest capitol este 4, dar există deja elaborată şi versiunea 6, care în următorii ani va deveni dominantă.

***IHL*:** Câmpul ne spune lungimea header-ului exprimată în număr de cuvinte de 4 octeţi. Dacă lungimea este 20, atunci valoare lui IHL este 5.

Service Type: Câmpul acesta este împărţit la rândul lui în mai multe subcâmpuri (figura 1-10).

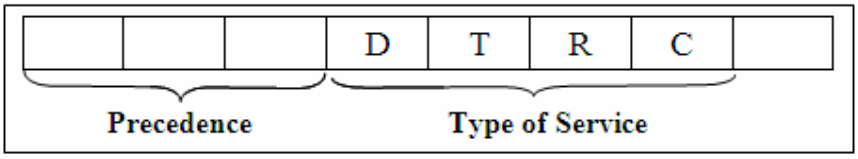


Figura 1-10 Structura câmpului Service Type

***Precedence*:** a fost gândit pentru a defini prioritatea unui pachet. Astfel dacă la un moment dat un router ar fi fost nevoit să ignore anumite pachete, atunci ar fi făcut-

o cu cele care au prioritate mai mică. În versiunea 4 a protocolului IP, acest subcâmp nu mai este folosit.

Type of Service: acest subcâmp este format din 4 biți. Fiecare dintre ei are o anumite semnificație şi doar unul poate fi setat la un moment dat. Semnificație fiecărui bit este dată în tabelul din figura 1-11.

ToS	Semnificație
0000	Normal
0001	Minimizează costul
0010	Maximizează siguranța
0100	Maximizează capacitatea de transfer
1000	Minimizează întârzierea

Figura 1-11 Semnificația biților din subcâmpul Type of Service

***Total Length*:** Acest câmp conține lungimea totală a pachetului. Dacă se doreşte să se afle lungimea datelor, se scade din lungimea totală valoarea câmpului Header Length înmulțită cu 4.

***Identification*:** Folosit în procesul de fragmentare a pachetelor.

***Flags*:** Folosit de asemenea în procesul de fragmentare a pachetelor.

***Time to Live*:** Acest câmp este folosit pentru a stabili numărul maxim de hop-uri (router-e) prin care un pachet poate trece. Fiecare router care procesează pachetul decrementează câmpul cu o unitate. Când valoare ajunge la zero, pachetul este eliminat din rețea şi un mesaj de eroare este generat către nodul care avea adresa trecută în câmpul *Source IP Address*. Valoarea de inițializare a acestui câmp este de obicei dublul numărului maxim de router-e care se pot interpune între sursă şi destinație. Este nevoie de acest mecanism deoarece în absența lui şi în anumite circumstanțe (router-e corupte) anumite pachete ar putea călători la infinit în rețea, consumând inutil resursele rețelei.

***Protocol*:** Prin acest câmp se identifică protocolul de nivel superior care face uz de datagramele IP pentru a-şi transporta datele. Unele valorile posibile sunt arătate în figura 1-12.

Protocol	Valoare
TCP	6
UDP	17
ICMP	1
OSPF	89
EGP	8
Ipv6	41

Figura 1-12 Valori posibile pentru câmpul Protocol

***Checksum*:** Sumă de control aplicată pachetului.

***Source Address*:** Câmpul conține adresa nodului care a trimis pachetul.

***Destination Address*:** Câmpul conține adresa nodului căruia îi este destinat pachetul.

Fiecare protocol de pe Nivelul Legătură de Date are propriul format pentru frame-urile utilizate la transportul informaţiei. Când un pachet IP traversează diferite reţelele el trebuie să fie încapsulat în aceste frame-uri ale Nivelului Legătură de Date. Fiecare frame acceptă o anumită dimensiune maximă pentru câmpul de date. Astfel, dacă dimensiunea pachetului depăşeşte această valoare, pachetul va trebui să fie fragmentat. În urma acestui proces trebuie ca fiecare fragment să conţină informaţiile necesare pentru dirijare, astfel ca fiecare fragment să poată ajunge la destinaţie unde aceste fragmente vor fi reasamblate.

Nodul care realizează fragmentarea unui pachet va trebui să modifice trei câmpuri: *Flags, Fragmentation Offset,* şi *Total Length.*

Câmpul *Identification* are şi el un rol important în procesul de fragmentare. Fiecare pachet primeşte un număr de identificare care va fi stocat în acest câmp. Acest număr este generat prin incrementare cu unu pentru fiecare nou pachet trimis. Valoarea de la care se porneşte este una pozitivă aleasă în mod aleator. Astfel, pachetele vor fi identificate în mod unic folosind această etichetă şi adresa sursei. Atunci când este necesar ca un pachet să fie fragmentat, fragmentele care au rezultat vor avea acelaşi număr de identificare cu cel al pachetului din care provin. În acest fel va putea fi refăcut pachetul original.

Câmpul *Flags* conţine 3 biţi. Primul este rezervat, iar următorii doi sunt notaţi cu D (*do not fragment*), respectiv M(*more fragment*). Dacă bitul D are valoarea 1, atunci pachetul nu poate fi fragmentat. Dacă nu există nici o posibilitate de a transmite mai departe pachetul fără a fi fragmentat, atunci este distrus. Dacă valoarea bitului D este 0 atunci pachetul poate fi fragmentat. Daca bitul M are valoarea 1 aceasta semnifică că pachetul nu este ultimul fragment ci mai sunt şi altele. Dacă valoarea este 0, atunci fragmentul este ultimul.

Câmpul *Fragmentation Offset* indică poziţia relativă a unui fragment în cadrul unui pachet. Poziţia este indicată sub formă de deplasament exprimat ca multiplu de 8 octeţi.

Header-ul unui pachet IP conţine o parte fixă şi una variabilă. După cum am văzut partea variabilă ocupă 20 de octeţi, iar parte variabilă este reprezentată de câmpul *Options* şi poate să ocupe maxim 40 de octeţi. În cele ce urmează vom descrie structura acestor opţiuni.

1.3.2 Protocolul TCP

Protocolul TCP [RFC793] are sarcina de a transforma Nivelul Reţea, reprezentat în cazul nostru prin protocolul IP, dintr-un nivel nesigur într-unul sigur. Pachetele care sunt transportate folosind protocolul IP se pot pierde, pot fi afectate de erori sau pot să sosească în altă ordine decât cea în care au fost trimise. Toate aceste probleme sunt gestionate de către protocolul TCP care rulează pe Nivelul Transport (figura 1-13). Tot protocolul TCP este responsabil şi de implementarea anumitor mecanisme de control al fluxului de date, adaptându-l la resursele disponibile în acel moment.

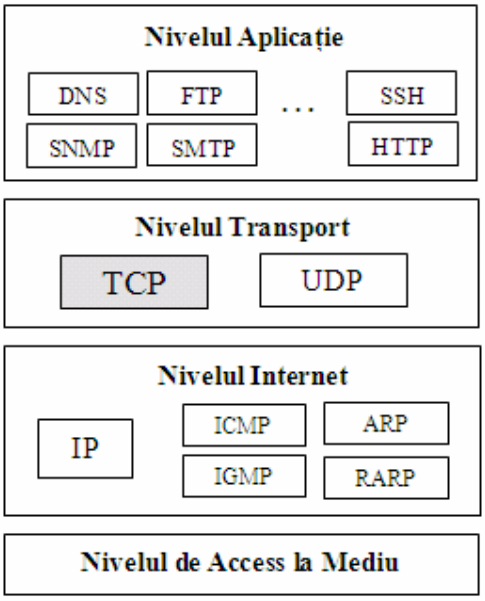


Figura 1-13 Poziția protocolului TCP în interiorul stivei TCP/IP

Spre deosebire de protocolul IP, protocolul TCP este de tipul capăt la capăt, adică este necesar să existe o instanță a acestui protocol doar pe mașina sursă şi pe mașina destinație, nu şi în nodurile intermediare care vor fi tranzitate de către pachete.

Serviciile oferite de către TCP sunt [Ste94, Com00]:

Stream Data Service

Aceasta presupune că pachetele care ajung la destinație să fie recepționate exact în ordinea în care au fost trimise. Adică, dacă trebuie transmis un mesaj care din cauza lungimii va fi fragmentat în mai multe pachete, la receptor se verifică sosirea tuturor pachetelor şi se ordonează în ordinea în care ele au fost trimise, astfel încât să se obțină mesajul original. Pentru a putea furniza acest serviciu, TCP face uz de buffer-e atât pe partea de transmisie, cât şi pe partea de recepție.

Full-duplex service

Acest serviciu presupune că transferul de informații între două noduri poate fi făcut simultan în ambele direcții. Când un pachet pleacă de la unul din noduri către celălalt, transportă şi un mesaj de confirmare pentru un pachet recepționat anterior. Acest procedeu poartă denumirea de *piggybacking*.

Reliable service

TCP foloseşte un mecanism de confirmări pentru a se asigura că pachetele nu au fost afectate de erori şi ca au sosit în ordinea corectă.

Formatul unui pachet TCP este redat în figura 1-14.

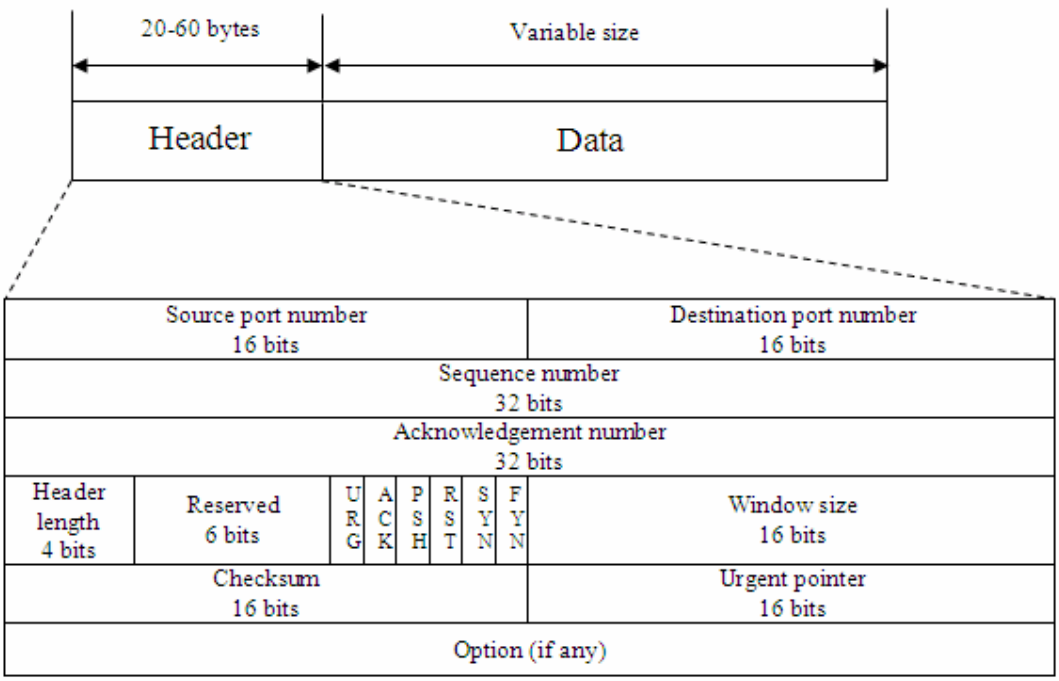


Figura 1-14 Formatul unui pachet TCP

Source port number **:** Numărul portului folosit de către aplicaţia care rulează pe maşina care trimite pachetele.

Destionation port number **:** Numărul portului folosit de aplicaţia care rulează pe maşina care primeşte pachetele.

Sequence number: TCP numerotează fiecare octet trimis şi foloseşte acest câmp pentru a indica numărul de secvenţă al primului octet de date din pachet. Când se iniţiază o comunicaţie între două maşini, numărul de secvenţă pentru primul octet trimis este ales aleator.

Acknowledgement number: Acest câmp este folosit pentru a confirma octeţii recepţionaţi. El se calculează însumând la valoarea din câmpul *Sequence number* al pachetului primit, dimensiunea câmpului de date recepţionat plus 1. Astfel valoarea acestui câmp reprezintă de fapt următoarea valoare pentru câmpul *Sequence number* care va fi folosită de către maşina care recepţionează pachetul de confirmare.

Sistemul acesta de confirmare este de tip *piggybacking* deoarece confirmarea octeţilor primiţi se poate face simultan cu trimiterea unui pachet care conţine alte date, dar pot fi trimise şi pachete care să conţină doar confirmări, nu şi date.

Header length: Conţine dimensiunea header-ului exprimată in cuvinte de 32 de biţi.

Control field: Conţine 6 biţi a căror semnificaţie va fi explicată atunci când se va discuta modul cum se iniţiază şi se desfăşoară o sesiune de comunicaţie.

Window size: reprezintă spaţiul disponibil existent în buffer-ul de recepţie.

1.4 Controlul congestiei în Internet

1.4.1 Principii generale

Atunci când resursele reţelei nu mai reuşesc să facă faţă traficului şi ea devine suprasolicitată, parametrii în care se desfăşoară traficul se degradează tot mai mult, în felul acesta instalându-se congestia. Congestia se manifestă prin întârzieri tot mai mari, printr-un număr mare de pachete pierdute şi în final se poate ajunge la colaps total, adică blocarea reţelei.

În funcţie de locul unde apare congestia există două categorii de congestie [KW03]:

- Congestie care apare din cauza suprasolicitării server-elor de aplicaţii (Server Side Congestion): apar prea multe cereri din partea clienţilor la un moment dat astfel încât noile cereri pentru conexiuni vor fi respinse.
- Cealaltă categorie de congestie apare pe partea de client, atunci când mai mulţi clienţi împart în comun aceleaşi conexiune fizice. În acest caz congestia apare în nodurile intermediare care nu mai pot să gestioneze numărul mare de conexiuni care apar la un moment dat.

Într-o reţea bazată pe comutarea de pachete resursele sunt distribuite la nivelul fiecărui nod din reţea. Aceste resurse pot fi definite prin trei elemente: capacitatea de procesare a informaţiilor, dimensiunea buffer-elor şi capacitatea de transport a liniilor.

Luând în considerare capacitatea de transport şi întârzierile care apar, instalarea congestiei arată grafic ca în figura 1-15

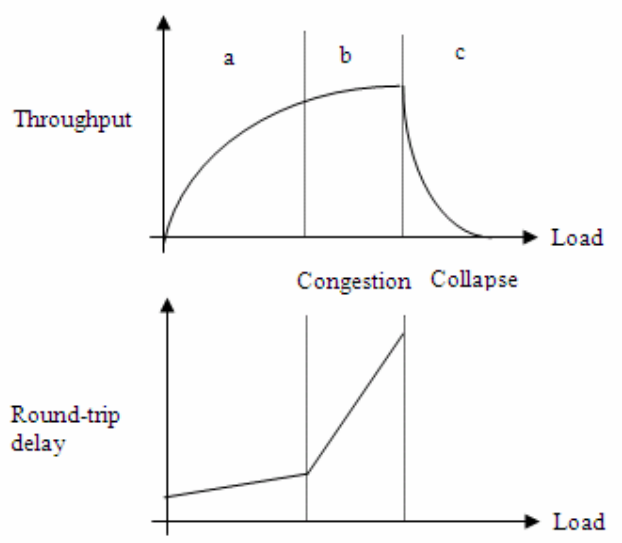


Figura 1-15 Evoluţia capacităţii de transfer şi a întârzierilor într-o reţea în care se manifestă fenomenul de congestie

Se observă că atunci când reţeaua lucrează în parametrii optimi, ea răspunde corect atunci când apare o încărcare mai mare (zona a). La început apare o creştere exponenţială, apoi o zonă în care reţeaua nu mai reacţionează la o creştere

suplimentară a încărcării (zona b), pentru ca apoi dintr-un anumit punct capacitatea de transfer să scadă brusc, iar dacă nu sunt luate măsuri, se ajunge la colaps (zona c).

În ceea ce priveşte întârzierea introdusă de reţea se observă că se păstrează o valoare aproximativ constantă în zona a, apoi pe măsură ce reţeaua începe să fie congestionată (zona b) întârzierea introdusă de reţea devine tot mai mare.

Pentru controlul congestiei există două abordări şi anume [Tan96, GC01]:

- un control în buclă deschisă
- un control în buclă închisă (cu *feedback*)

În primul caz se stabilesc de la început parametrii în care va funcţiona reţeaua, luându-se măsuri în faza de proiectare pentru prevenirea apariţiei problemelor, deoarece controlul se va face fără informaţii despre situaţia concretă de la un moment dat din reţea. Controlul în buclă deschisă se face de obicei la periferia reţelei, prin supravegherea traficului (*traffic policing*) şi prin formarea traficului (*traffic shaping*) pentru nodul care a primit acces la resursele reţelei. Formarea traficului presupune uniformizarea ratei medii de transmisie a datelor. Două exemple de algoritmi pentru formarea traficului sunt cel al găleţii găurite (*leaky bucket)* şi cel al găleţii cu jeton (*token bucket*). Vom descrie succint pe fiecare în parte.

Algoritmul găleţii găurite

Algoritmul poartă această denumire deoarece se face analogie cu o găleată care are un orificiu pe fund. Indiferent de debitul cu care apa intră în găleată, ea se va scurge prin orificiu cu un debit constant. La un moment dat, dacă găleata continuă să fie alimentată ea se poate umple şi apa se pierde. Acelaşi principiu poate fi aplicat şi în cazul unei transmisii de date. Pachetele care sunt trimise dintr-un anumit nod în reţea sunt trecute printr-o „găleată găurită", adică un buffer de tip FIFO care acceptă pachete la orice rată de transfer, dar le transmite mai departe cu o rată fixă, comportament ilustrat în figura 1-16. Dacă buffer-ul se umple atunci pachetele care sosesc ulterior se pierd. În felul acesta se realizează în mod implicit şi controlul traficului, adică urmărirea dacă un utilizator depăşeşte parametrii de trafic care i-au fost atribuiţi. Marele avantaj al acestei metode este acela că nu permite traficului în rafală care ar putea veni din partea nodului transmiţător să pătrundă sub această formă în reţea, aceasta fiind principală cauză a apariţiei congestiei, chiar şi atunci când resursele reţelei ar părea ca sunt suficiente.

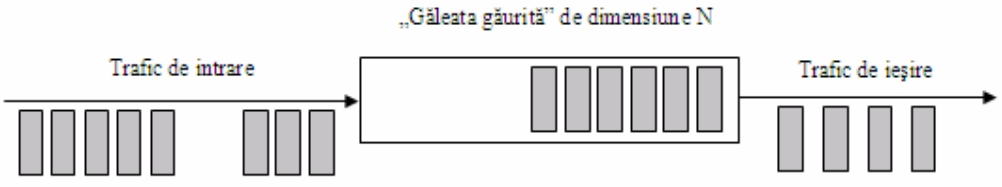


Figura 1-16 Algoritmul găleţii găurite

Algoritmul găleţii cu jeton

Acest algoritm se aseamănă în mare măsură cu cel anterior, dar spre deosebire de acesta permite flexibilitate în ceea ce priveşte rata traficului de ieşire. În cazul găleţii găurite rata acestui trafic era fixă. Algoritmul funcţionează în felul următor: găleata acumulează jetoane generate cu o rată de un jeton la ΔT secunde. Pentru ca un pachet să poată fi trimis el trebuie să găsească un jeton în găleată, pe care să-l

distrugă. Acest mod de abordare a problemei permite ca în momentul în care la intrare avem date în rafală, iar în găleată avem jetoane disponibile, atunci datele vor fi transmise tot în rafală, dar lungimea rafalei are maxim valoarea egală cu numărul de jetoane din găleată, aşa cum reiese şi din figura 1-17.

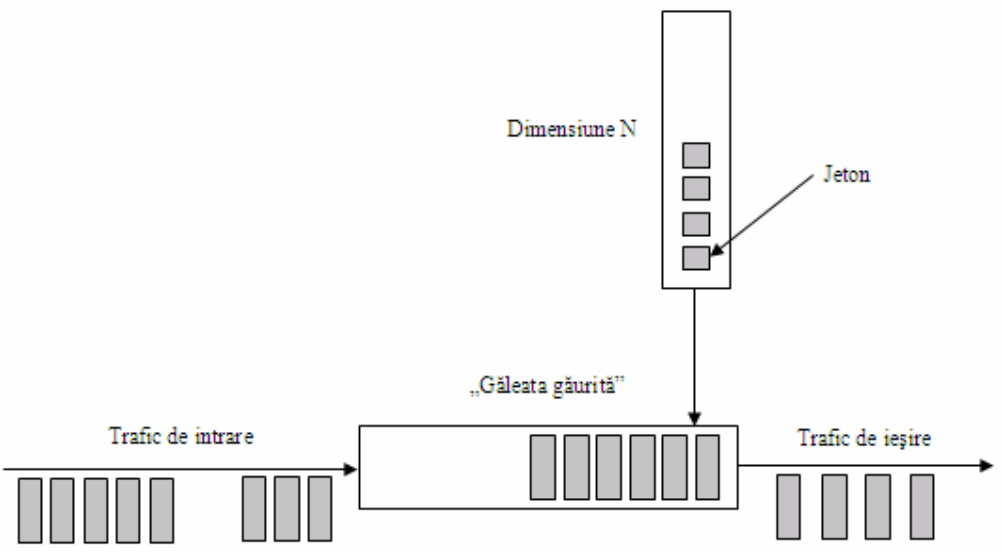


Figura 1-17 Algoritmul găleţii cu jeton

Al doilea mod de control al congestiei este cel în buclă închisă. Aici măsurile care sunt întreprinse se bazează pe informaţii culese în permanenţă din interiorul reţelei. Informaţiile primite pot fi implicite sau explicite. Un tip de informaţie implicită ar putea fi numărul de pachete pierdute, sau întârzierile din reţea. Informaţiile explicite sunt cele generate în mod special pentru a avertiza despre apariţia congestiei. Aceste informaţii pot fi pachete suplimentare care să conţină date despre congestie sau ar putea fi folosite anumite câmpuri în cadrul pachetelor şi care sunt setate cu anumite valori atunci când apare congestia. A doua metodă este de dorit deoarece nu conduce la o încărcare suplimentare a reţelei cu pachete, în momentul când acest lucru este cel mai puţin de dorit.

Toate discuţiile care urmează se vor referi la controlul congestiei în buclă închisă.

Pentru a se evita funcţionarea reţelei în zona de congestie este nevoie să se folosească procedee de monitorizare şi control al congestiei. Când se pune problema controlului congestiei două mecanisme trebuie luate în discuţie: evitarea congestiei (*congestion avoidance*) şi ieşirea din congestie (*congestion recovery*). Aceste mecanisme pot fi implementate pe de o parte la nivelul router-elor, care reprezintă nodurile intermediare atunci când are loc un transfer de date între sursă şi destinaţie, cât şi la nivelul sursei şi a destinaţiei, adică al celor două noduri între care se desfăşoară transferul. Se spune că în acest caz realizăm un control capăt la capăt.

Într-o reţea ideală, pentru ca cele două tipuri de control să fie eficiente este nevoie ca pe de o parte reţeaua să ofere feedback pentru ca resursele să fie folosite în mod eficient, iar pe de altă parte este nevoie ca fluxurile de date să fie protejate unele faţă de altele, în cazul în care unii utilizatori ar avea tendinţa să acapareze mai multe resurse decât cele care le-ar fi alocate. Această protejare a fluxurilor de date se poate face prin mecanisme de tip QoS.

Principalul avantaj al unei reţele bazate pe comutarea de pachete este faptul că toate resursele reţelei vor fi folosite împreună de către toţi utilizatorii, iar reţeaua va încerca să aloce maximul de resurse disponibile fiecărui utilizator în parte. Ceea ce creează probleme este natura impredictibilă şi în rafale a traficului, aceasta putând să conducă la apariţia congestiei [Jac88, ZS91]. Pentru scurte momente de timp reţeaua poate deveni supraîncărcată şi pentru a se evita intrarea în congestie este necesar ca într-un anume fel utilizatorii să fie înştiinţaţi de acest lucru şi să treacă la o diminuare a încărcării reţelei, evitându-se astfel apariţia congestiei. Acest mecanism va funcţiona făcând presupunerea că toţi utilizatorii reţelei vor coopera şi vor lua în considerare semnalele care avertizează asupra apariţiei congestiei. Pentru a se asigura acest lucru au fost implementate mecanisme de evitare a congestiei chiar în protocoalele de comunicaţie în Internet [Jac88].

1.4.2 Controlul congestiei de tip *end to end* practicat de TCP

TCP-ul este un protocol de tip capăt la capăt, aceasta însemnând că pentru a se realiza o comunicaţie între două noduri la nivel de protocol TCP, este nevoie să existe câte o instanţă a lui doar în nodurile sursă şi destinaţie, el făcând apel la serviciile oferite de către protocolul IP. În felul acesta cele două noduri care comunică la nivel de protocol TCP au senzaţia că sunt conectate direct, fără a mai exista noduri intermediare, de aceea se va observa că şi unele mecanisme implementate de către acest protocol seamănă cu cele folosite de protocoalele de la Nivelul Legătură de Date.

Protocolul TCP oferă două funcţii importante:

- Un transfer sigur al datelor
- Controlul congestiei

Pentru un transfer sigur al datelor, TCP-ul trebuie să implementeze următoarele mecanisme:

- Un mecanism de retransmisie al datelor
- Recepţia pachetelor în ordinea în care au fost trimise
- Multiplexarea traficului
- Controlul fluxului

Multiplexarea traficului înseamnă posibilitatea de a realiza conexiuni multiple între aceleaşi două noduri. Acest lucru este posibil în TCP prin folosirea numerelor de porturi. O conexiune realizată între două noduri va putea fi identificată în mod unic prin perechile (adresa IP sursă, număr port sursă) şi (adresă IP destinaţie, număr port destinaţie). Un acelaşi nod va putea avea la un moment dat mai multe conexiuni stabilite simultan, ele deosebindu-se prin numerele de port folosite.

Pe lângă mecanismul de control al congestiei, TCP-ul implementează şi un mecanism de control al fluxului numit fereastră glisantă (*sliding window*). Aceasta permite ca mai multe pachete să fie trimise fără a se aştepte confirmarea fiecărui pachet în parte. Numărul de pachete care pot fi trimise fără a aştepta confirmarea lor depinde de dimensiunea ferestrei.

Dacă în cadrul unei conexiuni unul dintre noduri tinde să transmită date mai repede decât poate celălalt nod să le proceseze, atunci datorită mecanismului cu fereastră glisantă, fluxul de date va fi adaptat în funcţie de nodul mai lent.

În figura 1-18 este reprezentat modul de gestionare al pachetelor la nivelul ferestrei glisante atât la transmiţător, cât şi la receptor.

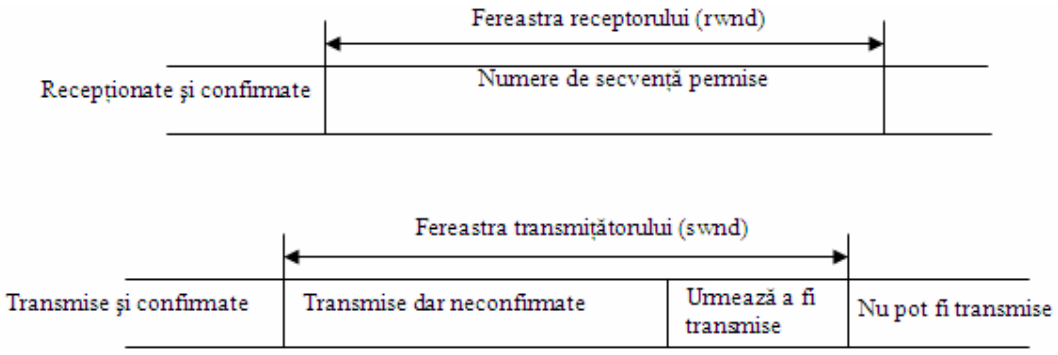


Figura 1-18 Evoluţia ferestrei glisante

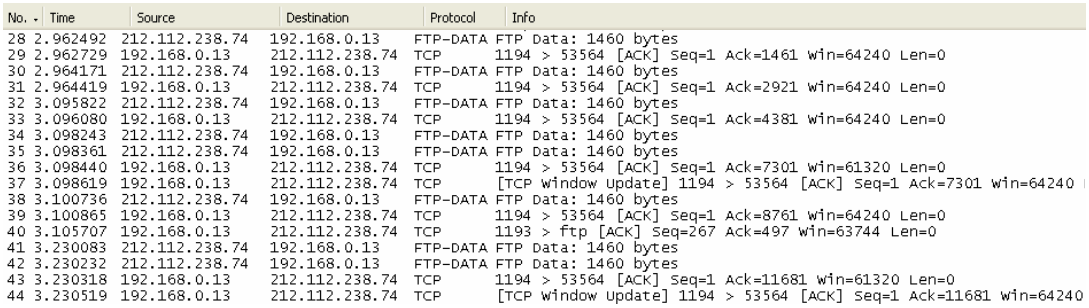
Atunci când un pachet este recepţionat se verifică dacă numărul lui de secvenţă coincide cu numărul de secvenţă de la începutul ferestrei receptorului, adică este următorul număr de secvenţă aşteptat. Daca numărul nu coincide dar se află în interiorul ferestrei, atunci este introdus în buffer dar nu este confirmat, iar receptorul trimite un pachet de confirmare care conţine în câmpul ACK aceiaşi valoare cu cea din pachetul de confirmare corespunzător ultimului pachet de date valid.

Dacă numărul de secvenţă nu este cel aşteptat şi nici nu se află în interiorul ferestrei de recepţie, atunci pachetul este ignorat. În momentul când soseşte pachetul aşteptat, atunci acesta este confirmat, iar limita din stânga a ferestrei se deplasează spre dreapta ducând la o micşorare a ferestrei. Se spune că fereastra se închide. Limita din dreapta a ferestrei se va deplasa spre dreapta doar în momentul în care pachetele care au fost confirmate sunt scoase din buffer pentru a fi procesate. Pachetele de confirmare trimise de receptor conţin întotdeauna şi dimensiunea la acel moment a ferestrei receptorului. Fereastra glisantă permite de asemenea şi depistarea pachetelor duplicat.

Retransmisia datelor se realizează fie când pachetele ajung la destinaţie dar sunt afectate de erori, fie când s-au pierdut pe drum. Pentru ca procedeul de retransmisie să funcţioneze este nevoie de folosirea unor timere şi a unui mecanism de confirmări pozitive (*positive acknowledgements*). Confirmarea pozitivă înseamnă că sunt confirmate doar pachetele care au ajuns neafectate de erori. Pentru a optimiza mecanismul de retransmisie se foloseşte o metodă cumulativă de confirmări, care permite confirmarea printr-un singur mesaj a unui grup de pachete consecutive.

Pentru transmisia confirmărilor sunt folosite pachete de date, procedeul purtând denumirea de *piggybacking*. Există şi posibilitatea de a transmite distinct doar pachete de confirmare.

În figura 1-19 este redată o captură de trafic care pune în evidenţă modul cum variază dimensiunea ferestrei receptorului.



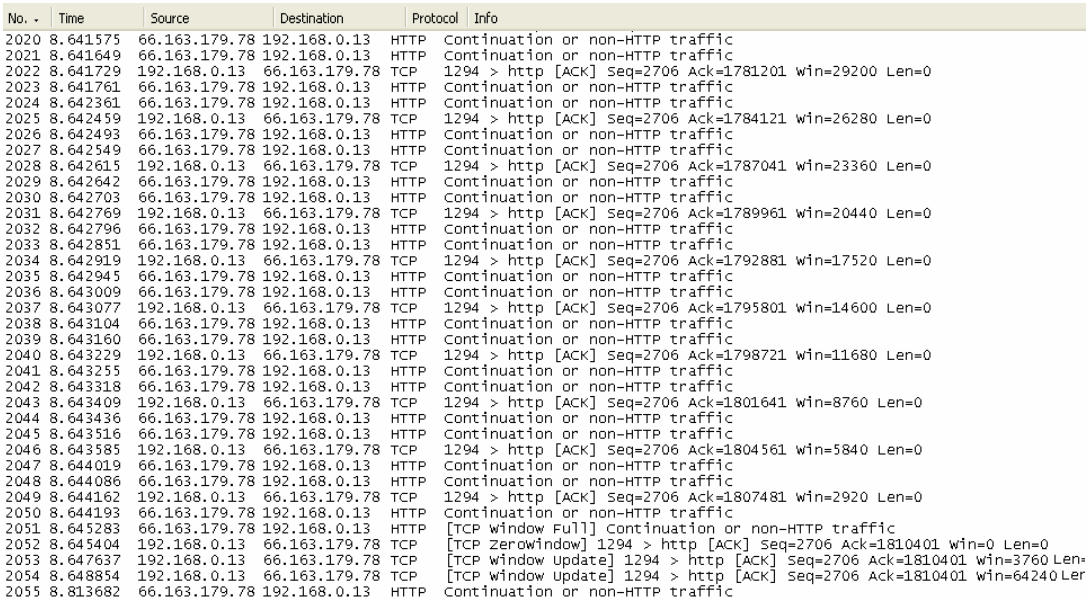
No.	Time	Source	Destination	Protocol	Info
28	2.962492	212.112.238.74	192.168.0.13	FTP-DATA	FTP Data: 1460 bytes
29	2.962729	192.168.0.13	212.112.238.74	TCP	1194 > 53564 [ACK] Seq=1 Ack=1461 Win=64240 Len=0
30	2.964171	212.112.238.74	192.168.0.13	FTP-DATA	FTP Data: 1460 bytes
31	2.964419	192.168.0.13	212.112.238.74	TCP	1194 > 53564 [ACK] Seq=1 Ack=2921 Win=64240 Len=0
32	3.095822	212.112.238.74	192.168.0.13	FTP-DATA	FTP Data: 1460 bytes
33	3.096080	192.168.0.13	212.112.238.74	TCP	1194 > 53564 [ACK] Seq=1 Ack=4381 Win=64240 Len=0
34	3.098243	212.112.238.74	192.168.0.13	FTP-DATA	FTP Data: 1460 bytes
35	3.098361	212.112.238.74	192.168.0.13	FTP-DATA	FTP Data: 1460 bytes
36	3.098440	192.168.0.13	212.112.238.74	TCP	1194 > 53564 [ACK] Seq=1 Ack=7301 Win=61320 Len=0
37	3.098619	192.168.0.13	212.112.238.74	TCP	[TCP Window Update] 1194 > 53564 [ACK] Seq=1 Ack=7301 Win=64240
38	3.100736	212.112.238.74	192.168.0.13	FTP-DATA	FTP Data: 1460 bytes
39	3.100865	192.168.0.13	212.112.238.74	TCP	1194 > 53564 [ACK] Seq=1 Ack=8761 Win=64240 Len=0
40	3.105707	192.168.0.13	212.112.238.74	TCP	1193 > ftp [ACK] Seq=267 Ack=497 Win=63744 Len=0
41	3.230083	212.112.238.74	192.168.0.13	FTP-DATA	FTP Data: 1460 bytes
42	3.230232	212.112.238.74	192.168.0.13	FTP-DATA	FTP Data: 1460 bytes
43	3.230318	192.168.0.13	212.112.238.74	TCP	1194 > 53564 [ACK] Seq=1 Ack=11681 Win=61320 Len=0
44	3.230519	192.168.0.13	212.112.238.74	TCP	[TCP Window Update] 1194 > 53564 [ACK] Seq=1 Ack=11681 Win=64240

Figura 1-19 Evoluția ferestrei receptorului

Liniile 29, 31, 33 corespund unor pachete de confirmare care anunță o fereastră a receptorului de dimensiune 64240 octeți. Pachetul din linia 36 confirmă pachetele de date din liniile 34 şi 35, dar anunță o fereastră de dimensiune 61320, adică cu 2920 mai mică decât valoarea anterioare, reprezentând dimensiunea însumată a pachetelor din liniile 35 şi 35. Aceasta însemnă că nodul destinație nu a reuşit să proceseze pachetele de date.

Se observă în liniile 37 şi 44 nişte pachete de confirmare de tipul *Window update* care au rolul doar de a anunța o modificare a dimensiunii ferestrei receptorului deoarece doar în acest moment transmițătorul a reuşit să proceseze pachetele primite şi să le scoată din buffer-ul de recepție. Se observă că valoarea din câmpul ACK a pachetelor care fac actualizarea dimensiunii ferestrei receptorului este identică cu cea din pachetul ACK anterior.

În figura 1-20 este redat un exemplu care se referă la situația când fereastra receptorului se „închide". Acest moment este marcat prin setarea câmpului *Window size* din header-ul TCP, la valoarea zero.



No.	Time	Source	Destination	Protocol	Info
2020	8.641575	66.163.179.78	192.168.0.13	HTTP	Continuation or non-HTTP traffic
2021	8.641649	66.163.179.78	192.168.0.13	HTTP	Continuation or non-HTTP traffic
2022	8.641729	192.168.0.13	66.163.179.78	TCP	1294 > http [ACK] Seq=2706 Ack=1781201 Win=29200 Len=0
2023	8.641761	66.163.179.78	192.168.0.13	HTTP	Continuation or non-HTTP traffic
2024	8.642361	66.163.179.78	192.168.0.13	HTTP	Continuation or non-HTTP traffic
2025	8.642459	192.168.0.13	66.163.179.78	TCP	1294 > http [ACK] Seq=2706 Ack=1784121 Win=26280 Len=0
2026	8.642493	66.163.179.78	192.168.0.13	HTTP	Continuation or non-HTTP traffic
2027	8.642549	66.163.179.78	192.168.0.13	HTTP	Continuation or non-HTTP traffic
2028	8.642615	192.168.0.13	66.163.179.78	TCP	1294 > http [ACK] Seq=2706 Ack=1787041 Win=23360 Len=0
2029	8.642642	66.163.179.78	192.168.0.13	HTTP	Continuation or non-HTTP traffic
2030	8.642703	66.163.179.78	192.168.0.13	HTTP	Continuation or non-HTTP traffic
2031	8.642769	192.168.0.13	66.163.179.78	TCP	1294 > http [ACK] Seq=2706 Ack=1789961 Win=20440 Len=0
2032	8.642796	66.163.179.78	192.168.0.13	HTTP	Continuation or non-HTTP traffic
2033	8.642851	66.163.179.78	192.168.0.13	HTTP	Continuation or non-HTTP traffic
2034	8.642919	192.168.0.13	66.163.179.78	TCP	1294 > http [ACK] Seq=2706 Ack=1792881 Win=17520 Len=0
2035	8.642945	66.163.179.78	192.168.0.13	HTTP	Continuation or non-HTTP traffic
2036	8.643009	66.163.179.78	192.168.0.13	HTTP	Continuation or non-HTTP traffic
2037	8.643077	192.168.0.13	66.163.179.78	TCP	1294 > http [ACK] Seq=2706 Ack=1795801 Win=14600 Len=0
2038	8.643104	66.163.179.78	192.168.0.13	HTTP	Continuation or non-HTTP traffic
2039	8.643160	66.163.179.78	192.168.0.13	HTTP	Continuation or non-HTTP traffic
2040	8.643229	192.168.0.13	66.163.179.78	TCP	1294 > http [ACK] Seq=2706 Ack=1798721 Win=11680 Len=0
2041	8.643255	66.163.179.78	192.168.0.13	HTTP	Continuation or non-HTTP traffic
2042	8.643318	66.163.179.78	192.168.0.13	HTTP	Continuation or non-HTTP traffic
2043	8.643409	192.168.0.13	66.163.179.78	TCP	1294 > http [ACK] Seq=2706 Ack=1801641 Win=8760 Len=0
2044	8.643436	66.163.179.78	192.168.0.13	HTTP	Continuation or non-HTTP traffic
2045	8.643516	66.163.179.78	192.168.0.13	HTTP	Continuation or non-HTTP traffic
2046	8.643585	192.168.0.13	66.163.179.78	TCP	1294 > http [ACK] Seq=2706 Ack=1804561 Win=5840 Len=0
2047	8.644019	66.163.179.78	192.168.0.13	HTTP	Continuation or non-HTTP traffic
2048	8.644086	66.163.179.78	192.168.0.13	HTTP	Continuation or non-HTTP traffic
2049	8.644162	192.168.0.13	66.163.179.78	TCP	1294 > http [ACK] Seq=2706 Ack=1807481 Win=2920 Len=0
2050	8.644193	66.163.179.78	192.168.0.13	HTTP	Continuation or non-HTTP traffic
2051	8.645283	66.163.179.78	192.168.0.13	HTTP	[TCP Window Full] Continuation or non-HTTP traffic
2052	8.645404	192.168.0.13	66.163.179.78	TCP	[TCP ZeroWindow] 1294 > http [ACK] Seq=2706 Ack=1810401 Win=0 Len=0
2053	8.647637	192.168.0.13	66.163.179.78	TCP	[TCP Window Update] 1294 > http [ACK] Seq=2706 Ack=1810401 Win=3760 Len=
2054	8.648854	192.168.0.13	66.163.179.78	TCP	[TCP Window Update] 1294 > http [ACK] Seq=2706 Ack=1810401 Win=64240 Ler
2055	8.813682	66.163.179.78	192.168.0.13	HTTP	Continuation or non-HTTP traffic

Figura 1-20 „Închiderea" ferestrei receptorului

Trebuie remarcat şi în acest exemplu modul cumulativ de confirmare a pachetelor de date şi de asemenea trebuie observată dinamica ferestrei receptorului.

Dimensiunea datelor transportate de fiecare pachet provenit de la server-ul web (66.163.179.78) este de 1460 de octeţi. Dacă se face diferenţa valorilor din câmpul ACK de la două pachete de confirmări consecutive se obţine valoarea 2920, adică sunt confirmate ambele pachete de date. Se mai observă că deşi pachetele sunt confirmate, dimensiunea ferestrei receptorului scade, deoarece datele nu au putut fi procesate în ritmul în care au fost confirmate.

Pentru a se îmbunătăţi mecanismul de retransmisie a pachetelor de date, atunci când există pierderi multiple a fost adăugat la TCP mecanismul de *Selective Ackowledgement* (SACK) [RFC2018]. Prin acest mecanism receptorul poate informa transmiţătorul despre pachetele care au ajuns cu bine la destinaţie, transmiţătorul ştiind exact în acest moment care pachete trebuie să fie retransmise. Pentru a transmite aceste informaţii suplimentare este folosit câmpul *Options* din header-ul TCP.

TCP-ul gestionează pentru fiecare conexiune patru tipuri de timere. Dintre acestea pe noi ne interesează doar cel care joacă un rol în retransmiterea pachetelor şi pe acest îl vom descrie în continuare.

Pentru realizarea retransmiterii pachetelor care s-au pierdut este importantă determinarea timpului scurs din momentul trimiterii pachetului şi până în momentul recepţionării pachetului de confirmare pentru acel pachet (*round-trip time* sau RTT). Am văzut că datorită mecanismului de confirmări cumulative, nu există întotdeauna o corespondenţă unu la unu între un pachet de date şi pachetul de confirmare. În acest caz se ia în considerare pachetul de ACK care acoperă şi numărul de secvenţă al pachetului de date pentru care se face măsurătoarea. Pe baza acestui RTT este calculat intervalul de timp cât se aşteaptă pentru retransmisia pachetelor (*Retrasmission Timeout* – RTO).

Iniţial, conform specificaţiilor din RFC 793, perioada pentru retransmisie era calculată in felul următor:

$$R = \alpha R + (1 - \alpha)M$$

unde M este valoarea măsurată pentru RTT, iar R este o estimare pentru RTT. Valoarea recomandată pentru α este 0,9. Astfel, noua valoarea estimată este alcătuită în proporţie de 90% din vechea estimare şi 10% din noua valoare calculată. Această estimare este recalculată de fiecare dată când o nouă valoare pentru RTT este măsurată. RTO este calculat pe baza acestei estimări cu formula:

$$RTO = \beta R$$

iar valoarea recomandată pentru β este 2.

În [ZS91] se arată că această metodă de calcul pentru RTO nu este potrivită deoarece nu se comportă bine atunci când apar variaţii mai mari pentru RTT. Jacobson propune o altă abordare, în care să se ţină cont în calcularea lui RTO, nu doar de estimarea R ci şi de variaţia lui RTT. Asfel avem:

$$Err = M - A$$
$$A = A + gErr$$
$$D = D + h(|Err| - D)$$
$$RTO = A + 4D$$

mnde M reprezintă valoare măsurată pentru RTT, A este valoarea medie estimată pentru RTT, iar D este deviaţie medie. Err este diferenţa dintre valoarea măsurată a lui RTT şi valoarea estimată.

Sa urmărim cum reacţionează TCP-ul atunci când are loc un timeout şi la ce intervale de timp se fac retransmisiile (figura 1-21).

No.	Time	Source	Destination	Protocol	Info
206	60.239355	10.23.3.11	10.23.3.21	TELNET	Telnet Data ...
207	60.239586	10.23.3.21	10.23.3.11	TELNET	Telnet Data ...
208	60.239662	10.23.3.11	10.23.3.21	TCP	32773 > telnet [ACK] Seq=182 Ack=639 Win=5840 Len=0 TSV=32001
209	63.590087	10.23.3.11	10.23.3.21	TELNET	Telnet Data ...
210	63.790488	10.23.3.11	10.23.3.21	TELNET	[TCP Retransmission] Telnet Data ...
211	64.210496	10.23.3.11	10.23.3.21	TELNET	[TCP Retransmission] Telnet Data ...
212	65.050499	10.23.3.11	10.23.3.21	TELNET	[TCP Retransmission] Telnet Data ...
213	66.730490	10.23.3.11	10.23.3.21	TELNET	[TCP Retransmission] Telnet Data ...
214	69.970590	10.23.3.11	10.23.255.255	CUPS	ipp://10.23.3.11/printers/EPSONEPL5200+ (idle)
215	70.090474	10.23.3.11	10.23.3.21	TELNET	[TCP Retransmission] Telnet Data ...
216	76.810506	10.23.3.11	10.23.3.21	TELNET	[TCP Retransmission] Telnet Data ...
217	90.250501	10.23.3.11	10.23.3.21	TELNET	[TCP Retransmission] Telnet Data ...
218	100.970779	10.23.3.11	10.23.255.255	CUPS	ipp://10.23.3.11/printers/EPSONEPL5200+ (idle)
219	117.130511	10.23.3.11	10.23.3.21	TELNET	[TCP Retransmission] Telnet Data ...
220	122.130439	10.23.3.11	10.23.3.21	ARP	Who has 10.23.3.21? Tell 10.23.3.11
221	122.130578	10.23.3.21	10.23.3.11	ARP	10.23.3.21 is at 00:e0:00:1c:31:80
222	131.970592	10.23.3.11	10.23.255.255	CUPS	ipp://10.23.3.11/printers/EPSONEPL5200+ (idle)
223	162.970780	10.23.3.11	10.23.255.255	CUPS	ipp://10.23.3.11/printers/EPSONEPL5200+ (idle)
224	170.890502	10.23.3.11	10.23.3.21	TELNET	[TCP Retransmission] Telnet Data ...
225	170.891512	10.23.3.21	10.23.3.11	TELNET	Telnet Data ...
226	170.891592	10.23.3.11	10.23.3.21	TELNET	Telnet Data ...
227	170.891807	10.23.3.21	10.23.3.11	TELNET	Telnet Data ...
228	170.930487	10.23.3.11	10.23.3.21	TCP	32773 > telnet [ACK] Seq=186 Ack=643 Win=5840 Len=0 TSV=43071
229	170.930689	10.23.3.21	10.23.3.11	TELNET	Telnet Data ...
230	170.930768	10.23.3.11	10.23.3.21	TCP	32773 > telnet [ACK] Seq=186 Ack=870 Win=6432 Len=0 TSV=43071

Figura 1-21 O situaţie de generare a timeout-ului

După trimiterea pachetelor de date în liniile 206 şi 207, urmează confirmarea lor în linia208. În acest moment a fost întreruptă legătura dintre cele două calculatoare care comunicau. Se observă că primul timeout se obţine după aproximativ 200 msec de la trimiterea pachetului de date din linia 209. Următoarea retransmisie se face după aprox. 420 msec, apoi după 840 msec, după 1640 msec şi aşa mai departe. Se observă ca în această situaţie TCP-ul recurge la o dublare a intervalul de timeout de la o retransmisie la alta. Acest procedeu poartă denumirea de *exponential backoff*.

O situaţie deosebită de care trebuie să se ţină cont este aceea când un pachet de date este trimis, dar nu se primeşte confirmarea pentru el. După un anumit interval de timp se generează timeout, pachetul este retransmis, iar RTO este dublat după cum s-a văzut în exemplul de mai sus. Să presupunem acum că am primit confirmarea pentru pachetul de date transmis. Întrebarea este dacă această confirmare se referă la primul pachet de date sau la cel retransmis. Este nevoie să ştim acest lucru pentru a recalcula valoarea pentru RTO. În această situaţie nu se mai recalculează RTO-ul ci se foloseşte până la următoarea măsurătoare valoarea dublată rezultată în urma timeout-ului.

Protocolul de tip capăt la capăt (*end to end*) care implementează mecanisme de evitare a congestiei şi de ieşire din congestie este TCP. Şi protocolul UDP este tot de tip *end to end*, dar el nu are implementate mecanisme pentru controlul congestiei. Dintr-o perspectivă globală, acest lucru nu este deranjant, deoarece ponderea aplicaţiilor care folosesc protocolul TCP este cu mult mai mare decât al celor care folosesc protocolul UDP (aproximativ 90% din trafic este reprezentat de TCP).

Primele mecanisme de control a congestiei au început să fie introduse în anul 1987. Până la acea dată nu s-a pus problema implementării unor mecanisme care să vizeze in mod direct controlul congestiei şi măsurile luate până la acea dată se rezumau la diferite tehnici de folosire eficientă a resurselor reţelei. Unele dintre aceste probleme, rezolvate de-a lungul timpului au fost [Ste94]:

Delayd ACKs
Se referă la faptul de a nu transmite imediat confirmarea pentru un pachet de date sosit corect ci să se mai aştepte, în ideea că la un moment dat vor fi date de transmis, iar confirmarea va fi trimisă împreună cu un pachet de date.

Silly Window Syndrome
Reglementează modul în care trebuie făcută actualizarea dimensiunii ferestrei receptorului, astfel încât să se evite trimiterea de pachete mici atunci când actualizarea dimensiunii ferestrei receptorului s-ar face cu valori mici.

Nagle Algorithm
A fost creat pentru a rezolva transmiterea pachetelor de mici dimensiuni pe linii cu întârzieri mari. Regula impusă de acest algoritm este că dacă datele care trebuie să fie transmise sunt mai puţine decât valoarea MSS, atunci ele sunt reţinute până se primeşte confirmarea ultimului pachet de date transmis.

Pentru controlul congestiei [RFC2001, RFC2581, RFC2914, RFC3390] TCP-ul foloseşte patru algoritmi: ***Slow start***, ***Congestion avoidance***, ***Fast retransmit*** şi ***Fast recovery*** .
Slow start intervine în faza de iniţiere a conexiunii. Rolul lui este de a creşte treptat rata de transmisie a pachetelor fără a depăşi capacitatea de transport a reţelei sau viteza receptorului de a procesa pachetele primite.

Dacă în cadrul unei conexiuni s-ar ţine cont doar de dimensiunea ferestrei receptorului, atunci rata maximă de transfer practicată ar fi direct proporţională cu dimensiunea ferestrei de recepţie şi poate fi aproximată prin relaţia $\frac{rwnd}{RTT}$, unde ***rwnd*** înseamnă ***receiver advertised window***. În continuare vom folosi aceiaşi denumire ***rwnd*** şi pentru variabila internă utilizată de către protocolul TCP pentru desemnarea dimensiunii ferestrei receptorului.

Mecanismul de controlul al congestiei mai introduce încă o variabilă numită ***cwnd*** (***congestion window***), care va fi folosită împreună cu variabila ***rwnd***. Astfel, cantitatea de date care poate fi trimisă la un moment dat în reţea fără a se aştepta confirmarea lor este dată de minimul dintre ***cwnd şi rwnd*** şi poartă numele de ***swnd*** (***sender window***).

$$swnd = \min(cwnd, rwnd)$$

cwnd (*congestion window*) reprezintă cantitatea de date care ar putea fi trimisă în reţea fără a se aştepta confirmarea pachetelor. ***cwnd*** este impusă de transmiţător, iar ***rwnd*** (*receiver advertised window*) reprezintă dimensiunea ferestrei pusă la dispoziţie de către receptor, cu alte cuvinte, dimensiunea spaţiului liber din buffer-ului de recepţie.

Algoritmul *Slow start* mai foloseşte încă o variabilă pentru controlul congestiei şi aceasta se numeşte ***ssthresh***. Pe baza acestei variabile se va hotărî când să se treacă de la *Slow start* la *Congestion avoidance*.

Iniţial, valoare pentru ***cwnd*** este de maxim **2*SMSS** (***Sender Maximum Segment Size***), adică de două ori dimensiunea maximă a unui pachet de date pe care îl

va trimite acel nod, iar pentru *ssthresh* nu există o limită superioară, unele implementări iau o valoare egală cu dimensiunea ferestrei receptorului.

Atâta timp cât ***cwnd < ssthresh***, va fi folosit algoritmul *Slow start,* iar când ***cwnd > ssthresh***, atunci va intra în acțiune algoritmul *Congestion avoidance*.

Pe lângă algoritmii menționați mai sus, o comunicație TCP mai este guvernată de încă doi algoritmi: *Fast retransmit* şi *Fast recovery*.

Condițiile de trecere de la un algoritm de control al congestiei la altul sunt redate în figura 1-22.

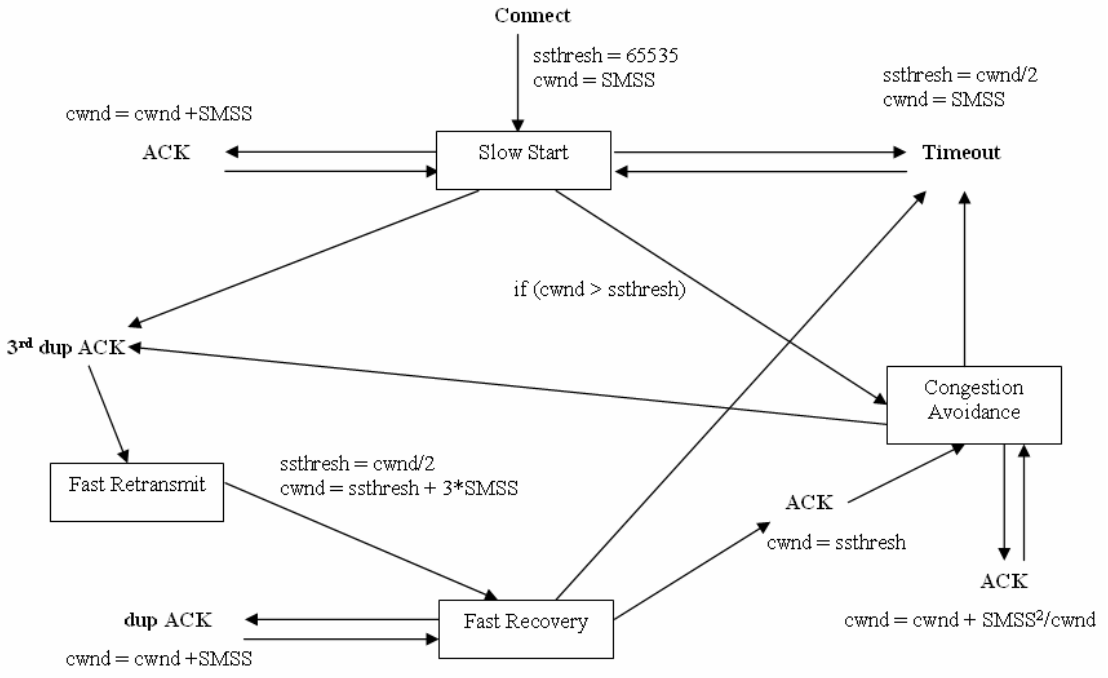


Figura 1-22 Condițiile de trecere de la un algoritm de control al congestiei la altul

Săgețile indică modul în care controlul unei anumite conexiuni TCP poate să fie preluat de unul sau altul din cei patru algoritmi în funcție de condițiile de trafic. Simplificând diagrama din figura 1-22, posibilitățile de trecere de la un algoritm de control al congestiei la altul sunt redate în figura 1-23.

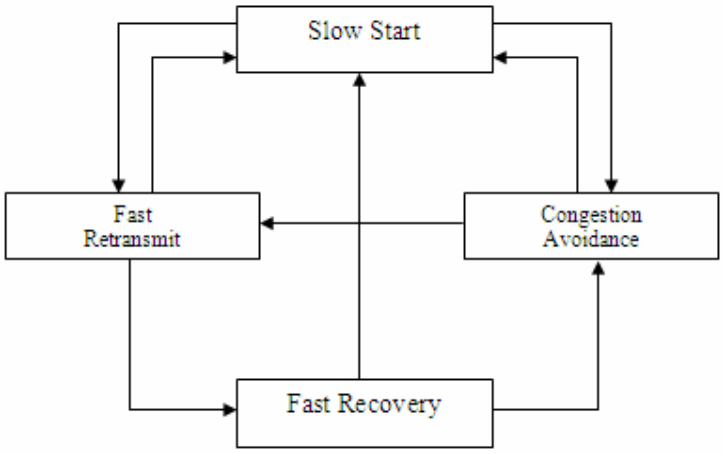


Figura 1-23 Relația dintre cei patru algoritmi de control ai congestiei

O transmisie debutează cu *Slow start*. În această fază pentru fiecare confirmare a unui pachet, variabila *cwnd* este incrementată cu cel mult valoarea lui SMSS. Nu sunt luate în considerare decât pachetele de tip ACK neduplicat Astfel, în această fază se realizează o creştere exponenţială a ratei cu care sunt transmise pachetele.

In figura 1-24 este redat modul cum funcţionează algoritmul *Slow start*.

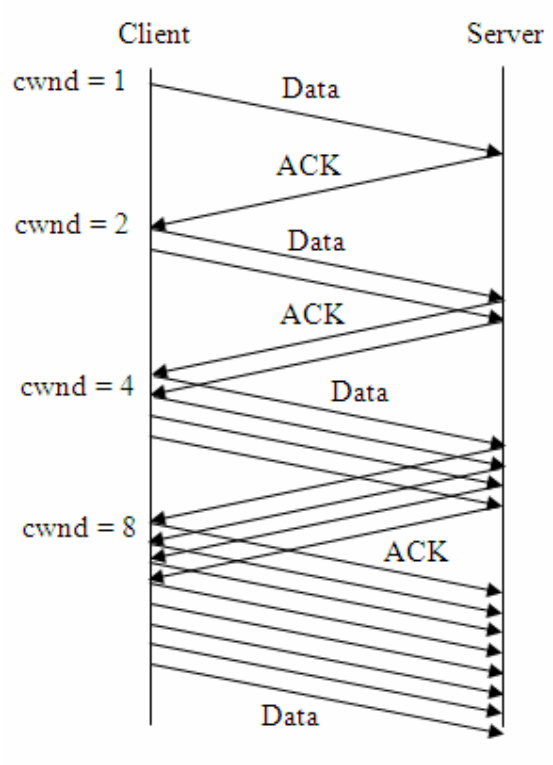


Figura 1-24 Algoritmul Slow Start

Pentru simplificare s-a considerat că dimensiunea iniţială a lui *cwnd* este 1 reprezentând un segment, iar apoi pentru fiecare pachet de confirmare primit se incrementează *cwnd* cu câte o unitate. În realitate valoarea lui *cwnd* este menţinută în octeţi şi este incrementată de fiecare dată cu dimensiunea segmentului de date care a fost confirmat.

În momentul în care *cwnd* devine mai mare decât *sstresh* se trece la *Congestion avoidance* şi se rămâne la acest algoritm până este detectată congestia.

Pe toată durata cât transmisia este controlată de *Congestion avoidance,* creşterea lui *cwnd* se va face odată cu sosirea fiecărui pachet de confirmare neduplicat, după formula:

$$cwnd = cwnd + \frac{SMSS \times SMSS}{cwnd}$$

Ideea este de a găsi folosi o formulă care să permită incrementarea lui *cwnd* cu un MSS la fiecare interval de timp egal cu RTT, ceea ce formula de mai sus aproximează suficient de bine.

Dinamica celor patru algoritmi pentru controlul congestiei este redată în figura 1-25.

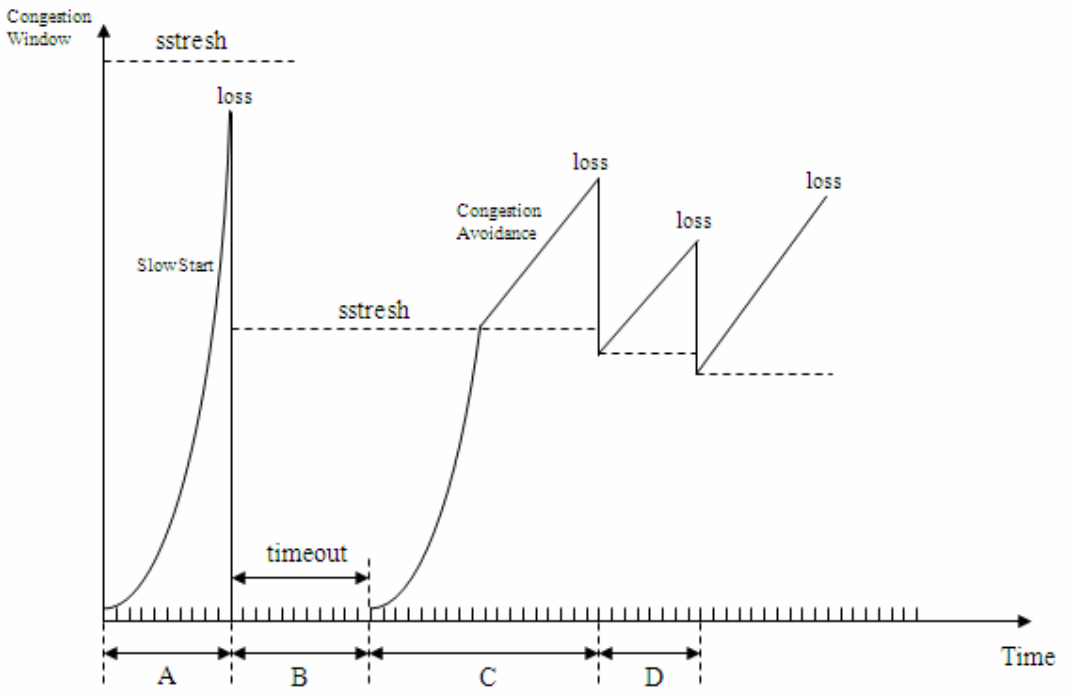


Figura 1-25 O posibilă evoluție în timp a algoritmilor de control ai congestiei

În desenul de mai sus trecerea de la *Slow start* la *Congestion avoidance* se observă în zona C . În zona A nu se mai ajunge de la *Slow start* la *Congestion avoidance* deoarece apar pierderi de pachete înainte de a se atinge pragul *sstresh* stabilit la inițierea conexiunii. Ce se întâmplă în această situație vom prezenta în continuare.

Când receptorul primeşte un pachet neafectat de erori, dar care are nu are numărul de secvență egal cu cel de la începutul ferestrei şi care se află totuşi în interiorul ferestrei, este obligat să trimită imediat un pachet ACK duplicat, adică să retransmită ultimul pachet de confirmare valid. Toate pachetele de date care vor sosi la receptor de acum încolo, vor determina transmisia unui pachet ACK duplicat. Pentru transmițător, recepția unor astfel de pachete ACK duplicat poate să însemne fie că s-au pierdut pachete de date, fie ca pachetele au ajuns la destinație, dar nu în ordinea în care au fost transmise, din cauză că au urmat căi diferite. A doua situație nu este gravă, pentru că în final toate pachetele vor ajunge la destinație şi de aceea, în acest caz nu vor fi luate măsuri speciale. Pentru a face distincție între cele două cauze posibile care au generat recepția de pachete ACK duplicat, TCP-ul aşteaptă recepția a cel puțin trei pachete consecutive de ACK duplicat pentru a reacționa. În momentul în care au fost recepționate trei astfel de pachete duplicat, este foarte probabil că pierderea unuia sau mai multor pachete de date a fost cauza transmiterii de pachete ACK duplicat. În acest moment este folosit algoritmul *Fast retransmit* pentru a retransmite pachetul de date care pare să se fi pierdut. Algoritmul poartă denumirea de *Fast retransmit* deoarece după recepția celor trei pachete de ACK duplicat se trece imediat la retransmisia datelor fără a se mai aştepta generarea unui timeout.

O altă situație posibilă este aceea când pierderea unor pachete de date să nu fie sesizată prin recepția pachetelor ACK duplicat ci prin expirarea unui timer setat la valoarea RTO. Pentru fiecare pachet de date expediat în rețea, TCP-ul măsoară timpul scurs de la transmiterea lui şi dacă într-un anumit interval de timp nu se primeşte ACK pentru acel pachet, atunci este generat un eveniment de tip timeout.

Atât recepția unor pachete ACK duplicat, cât şi apariția unui timeout sunt indicii că undeva în rețea a apărut fenomenul de congestie. În ambele situații variabila *sstresh* este recalculată după formula:

$$ssthresh = \max\left(\frac{\min(cwnd, rwnd)}{2}, 2 \times SMSS\right)$$

În plus, dacă congestia este indicată de un timeout, atunci şi variabila *cwnd* este adusă la valoarea inițială şi intră în acțiune algoritmul *Slow start*. Această situație este surprinsă în zonele A şi B.

Revenim la cazul în care pierderea pachetelor de date a fost semnalată prin pachete ACK duplicat. În această situație, după retransmiterea pachetului lipsă prin *Fast retransmit* nu se trece la *Slow start* deoarece faptul că au fost recepționate pachete ACK duplicat este un indiciu că după pachetul sau pachetele pierdute, la destinație au continuat să sosească pachete de date, dar care nu au numărul de secvență aşteptat şi acestea au determinat transmisia de pachete ACK duplicat.

În această situație de la *Fast retransmit* se trece la *Fast recovery* (zonele C şi D). Aceasta presupune parcurgerea următorilor paşi:

1. S-a retransmis pachetul care s-a pierdut
2. Se setează valoarea *cwnd* = *sstresh* + 3*SMSS,
3. Pentru fiecare pachet ACK duplicat se incrementează *cwnd* cu valoarea SMSS.
4. Când se recepționează un ACK care confirmă datele retransmise se setează *cwnd* la valoarea lui *ssthresh* şi în felul acesta se trece la *Congestion avoidance.*

1.4.3 Controlul congestiei la nivelul router-elor

Mecanismele de control ale congestiei prezentate mai sus se bazează pe faptul că rețeaua este văzut ca un *black box*, fără a avea nici un fel de informații din interior, totul bazându-se pe măsurători efectuate din punctele de capăt ale unei conexiuni.

Există totuşi o limită în ceea ce priveşte controlul congestiei de tip capăt la capăt. Pentru a îmbunătății controlul congestiei trebuie luate măsuri şi la nivelul router-elor [RFC2914, Hui00, HMP00]. În ceea ce priveşte acest tip de control, există două clase de algoritmi pentru controlul congestiei la nivelul router-elor. Acestea sunt: *queue management* şi algoritmi pentru *scheduling*. *Queue managemet* se referă la modul în care sunt gestionate lungimile cozilor la nivelul router-elor şi prin acest mecanism se hotărăşte care pachete să fie eliminate din cozile de aşteptare. Algoritmii pentru *scheduling* sunt folosiți pentru a împărți o anumită capacitate de transmisie la mai multe fluxuri de date.

Folosirea cozilor este necesară deoarece traficul în Internet, după cum am mai spus-o, este în rafală (*burst*), iar cozile au rolul de a absorbi cantitatea de date care nu poate fi procesate în ritmul în care este transmisă.

Înainte de implementarea acestor mecanisme gestiunea cozilor la nivelul router-elor se făcea foarte simplu fixând o dimensiune maximă pe care cozile o puteau lua, iar în momentul în care această valoare era atinsă, pachetele care soseau erau respinse, până când dimensiunea cozii scădea, datorită faptului că pachetele erau expediate spre destinaţiile lor. Această manieră de gestionare a cozilor se numeşte *tail drop*, deoarece pachete respinse erau cele mai noi venite. Metoda prezintă două dezavantaje importante: pe de o parte dădea posibilitatea unor conexiuni să monopolizeze întreaga coadă, iar a doua problemă care apărea era aceea că aceste cozi puteau rămâne pline pentru o perioadă destul de lungă de timp pentru că trebuia să treacă un anumit interval de timp până când cel care trimitea pachete în reţea sesiza că acestea s-au pierdut datorită fenomenului de congestie şi lua măsuri pentru scăderea ratei de transmitere a pachetelor.

Soluţia pentru a rezolva cele două probleme a fost aceea ca router-ul să înceapă să înlăture pachete din coadă înainte ca aceasta să devină plină, iar modul în care sunt alese pachetele eliminate să se facă oarecum intr-un mod aleator. Această manieră de abordare a problemei poartă numele de AQM (*active queue management*).

O variantă de implementare a acestor principii s-a materializat prin algoritmul numit RED (*Random Early Detection*) [RFC2309, FJ93]. Acest algoritm înlătură pachete din coadă pe baza calculării unei probabilităţi. Această probabilitate se foloseşte de estimarea lungimii medii a cozii de aşteptare. Estimarea se bazează pe felul cum a fost încărcată coada de aşteptare la momentul anterior estimării. Astfel, dacă coada a fost puţin încărcată în trecutul apropiat, atunci probabilitatea de eliminare a pachetelor din coadă este mică, iar dacă coada a fost mult încărcată, probabilitatea de eliminare a pachetelor care sosesc va creşte.

Pe lângă calcularea acestei probabilităţi RED mai realizează încă o operaţie şi anume aceea de a decide care dintre noile pachete care sosesc să fie eliminate din coadă. Cea mai simplă metodă este aceea de a elimina pachetele în mod aleator, în felul acesta nefiind defavorizate doar anumite conexiuni.

În decursul timpului au fost dezvoltate diverse variante de AQM [RR03]: Adaptive-RED , Dynamic-RED , Stabilized-RED, Blue, fiecare diferind prin modul de calculare a parametrii implicaţi în gestionarea cozilor.

O facilitate suplimentare prezentă la unele router-e care implementează algoritmi de tip AQM este aceea că router-ul poate indica posibila apariţie a congestiei nu doar prin eliminarea pachetelor din coada de aşteptare ci şi prin setarea unor biţi în headerul pachetelor, această metodă se numeşte ECN (Explicit Congestion Notification) [Kuz05, RFC3168, Flo01]. Pentru IPv4 sunt folosiţi biţii 6 şi 7 din câmpul TOS, iar pentru pachetele Ipv6 aceşti biţi corespund octetului Traffic Class, fiind folosite exact aceleaşi poziţii pentru cei doi biţi. Ei poată denumirile de ECT (ECT-Capable Transport) şi CE.

ECN		
ECT	CE	
0	0	Not- ECT
0	1	ECT(1)
1	0	ECT(0)
1	1	CE

Figura 1-26 Semnificaţia biţilor ECN

Combinaţiile celor doi biţi dau cele patru coduri care au semnificaţie prezentată în figura 1-26. Not-ECT semnifică că cel care a trimis pachetul cu acest cod nu suportă mecanismul ECN. Codurile ECT(1) şi ECT(0) sunt folosite de o sursă pentru a indica că are implementat mecanism ECN, iar CE este setat de router pentru a indica apariţia congestiei.

Mecanismul ECN prevede folosirea încă a două flag-uri la nivelul headerelor TCP. Astfel, doi biţi din cei şase rezervaţi capătă denumirile ECE (ECN-Echo) şi CWR. Algoritmul ECN va funcţiona în felul următor:

- în prima fază nodul sursă trimite pachete care au setat în headerul IP unul din cel două coduri ECT (0) sau ECT(1), înştiinţând că suportă ECN.
- un router care detectează începutul unei congestii şi sesizează că pachetele care ar trebuie să le elimine din coadă au setat unul din cele două coduri ECT, în loc să le elimine, le va marca folosind codul CE şi la va direcţiona spre destinaţie.
- când pachetul ajunge la destinaţie şi este pregătit pachetul de confirmare pentru cel primit, acestui pachet de ACK i se va seta flag-ul ECE din headerul TCP.
- transmiţătorul recepţionează acest pachet de ACK, iar faptul că flag-ul ECE se găseşte setat este un indiciu că apare fenomenul de congestie şi trebuie luate măsuri.
- următorul pachet de date care îl va trimite va avea flag-ul CWR setat pentru a confirma recepţia pachetului de ACK cu flag-ul ECE setat.

Avantajul acestei tehnici este acela ca sursa se va comporta, din punctul de vedere al măsurilor care trebuiesc luate, exact ca şi în cazul în care congestia ar fi fost indicată prin pierderea de pachete, dar în cazul acesta se reacţionează mult mai rapid, înainte de a se ajunge la pierderea de pachete.

1.5 Concluzii

Scopul acestui capitol a fost acela de a face o scurtă prezentare a principalele protocoale implicate în transportul informaţiei la nivelul Internetului, pentru ca apoi să fie prezentate în detaliu mecanismele de control ale congestiei implementate în protocolul TCP. Aşa cum se va vedea mai clar şi din capitolele următoare (capitolul 2 şi capitolul 3), aceste mecanisme au o influenţă negativă atunci când comunicaţia dintre un nod sursă şi destinaţie implică tranzitarea unei reţele wireless de tip 802.11, ducând, în unele circumstanţe, la o degradare a ratelor de transfer.

În final au fost enumerate pe scurt principalele metode de control ale congestiei implementate la nivelul router-elor. Spre deosebire de tehnicile de control ale congestiei implementate de TCP, acestea nu sunt de tip *end-to-end*, deoarece sunt implementate la nivelul nodurilor intermediare tranzitate de pachete. Acestea nu au o influenţa negativă asupra parametrilor în care se desfăşoară o comunicaţie la nivel TCP, care în partea finală foloseşte pentru conexiune o reţea 802.11 şi de aceea ele fost prezentate succint.

Capitolul 2 Tehnologia WLAN 802.11

Acest standard face parte din grupul de standarde 802.x LAN. Wireless LAN a fost conceput pentru interconectarea sistemelor de calcul folosind ca mediu de transmisie undele radio. Avantajul acestui tip de transmisie constă în primul rând într-o mobilitate sporită, iar uneori poate conduce şi la costuri mai mici de implementare a unei reţele în locurile greu accesibile. Iniţial reţelele WLAN au fost gândite pentru extinderea reţelelor locale cablate, apoi, datorită multiplelor avantaje oferite de acestea ele au început să înlocuiască treptat reţelele LAN clasice.

Acest capitol face o prezentare a tehnologiei IEEE 802.11, pentru a oferi informaţiile necesare înţelegerii diverselor soluţii prezentate în această lucrare, soluţii care vizează îmbunătăţiri ale unor parametrii care caracterizează o conexiune TCP, atunci când conexiune tranzitează o reţea 802.11.

Principala particularitate a reţelelor wireless este aceea ca mediul fizic folosit în acest caz sunt undele radio. Acestea au proprietăţi total diferite de ale celorlalte medii fizice folosite în comunicaţiile de date:

- este un mediu care nu are o delimitare clară în spaţiu
- nu este protejat faţă de interferenţele cu alte semnale
- are o topologie care se poate modifica uşor
- nu putem avea certitudinea că orice staţie este „auzită" de către a orice altă staţie
- modul de propagare a semnalelor poate varia în timp şi poate prezenta asimetrii

2.1 Privire generala

În 1997 IEEE a adoptat primul standard pentru wireless LAN, denumit IEEE Std. 802.11-1997, cu rate de transfer de până la 2 Mbps. De atunci au fost ratificate mai multe amendamente sau standarde care prevedeau rate de transfer mai mari prin îmbunătăţiri aduse mediului fizic de transmisie. Aceste noi standarde au fost IEEE 802.11b, IEEE 802.11g şi IEEE 802.11 a. Spre deosebire de 802.11b şi 802.11g care operează în banda de 2,4GHz, 802.11a funcţionează în banda de 5GHz. Cu toate acestea, pentru toate cele trei noi standarde s-au păstrat nemodificate specificaţiile pentru subnivelul MAC, prezente la standardul iniţial 802.11. De fiecare dată când se dorea o nouă funcţionalitate era creat un nou grup de lucru de către IEEE care propunea un nou amendament la standardele deja existente. Fiecărui standard sau amendament i s-a asociat o nou literă. Astfel, 802.11e a introdus suport pentru transmisii de date şi voce, extinzând standardul iniţial cu facilităţi de tip QoS (Quality of Service), 802.11s a introdus facilitaţi de comunicare pentru arhitecturi de tip mesh, 802.11n a adus îmbunătăţiri asupra ratelor de transfer, iar 802.11w a adus modificări pentru a oferi o securitate sporită, etc.

Amendamentele care au avut contribuţia cea mai mare în adoptarea pe scară largă a reţelelor WLAN au fost 802.11a, 802.11b, 802.11g. Din acest motiv o să subliniem câteva particularităţi pentru fiecare din aceste standarde. Acestea diferă între ele prin ratele de transfer practicate, prin metoda de acces la mediul fizic şi prin banda de frecvenţă folosită.

Referitor la ratele de transfer trebuie făcute câteva precizări. Valorile prezentate de standard reprezintă cantitatea totală de informaţie vehiculată în unitate de timp. În aceasta cantitate, partea utilă de informaţie reprezintă doar o fracţie, undeva in jur de 45-50%. Deci rata efectivă de transfer se poate obţine, fără teama de a introduce erori mari, înjumătăţind valoare prevăzută de standard.

IEEE	Rată de transfer	Frecvenţă	Anul ratificării
802.11	2 Mb/s	2,4 GHz	1997
802.11b	11 Mb/s		1999
802.11g	54 Mb/s		2003
802.11a		5,2 GHz	1999

Figura 2-1 Principalele amendamente la standardul 802.11

802.11b

Foloseşte ca metodă de acces la mediu DSSS (*Direct Sequence Spread Spectrum*) în banda de 2,4 GHz. Lăţimea de bandă avută la dispoziţie este de 97MHz, împărţită în 14 canale, cu doar 3 canale nesuprapuse (figura 2-2). Lăţimea fiecărui canal este de 22MHz, cu o distanţă între purtătoare de doar 5MHz. Rata maximă de transfer este de 11Mbps, dar ca valoare efectivă se obţine maxim 5Mbps.

802.11g

Este o extensie a standardului 802.11b, operează tot în banda de 2,4GHz, dar ca metodă de acces la mediul fizic este folosită tehnologia OFDM (*Orthogonal Frequency Division Multiplexing*). Lăţimea de bandă oferită este la fel ca şi în cazul lui 802.11b, adică de 97MHz, impărţită în 14 canale, cu 3 canale nesuprapuse. Rata maximă de transfer este de 54Mbps, dar ca valoare efectivă maximă se obţine 22Mbps. Datorită compatibilităţii dintre cele două standarde, un dispozitiv 802.11g va putea comunica cu un dispozitiv 802.11b, dar la rate de transfer de maxim 11Mbps.

802.11a

Operează în banda de 5GHz şi de aceea compatibilitatea cu standardele 802.11b şi 802.11g nu este posibilă. Metoda de acces la mediul fizic este tot OFDM, dar datorită lăţimii de banda mai mari (300 MHz) s-au putut obţine astfel mai multe canale, existând 8 canale nesuprapuse, faţă de 3 în cazul benzii de 2,4GHz. Rata maximă de transfer este tot de 54Mbps, iar ca rată de transfer efectivă se obţine un maxim de 27Mbps, mai mare decât în cazul lui 802.11g.

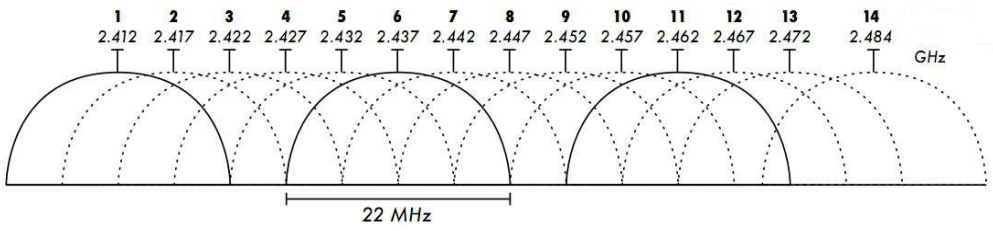


Figura 2-2 Repartizarea canalelor în cazul benzii de 2.4MHz

2.2 Topologii posibile pentru o rețea 802.11

O rețea locală de tipul 802.11 se bazează pe o arhitectură de tip celular. O celulă poarte denumirea de BSS (*Basic Service Set*) şi este controlată de către un AP (*Access Point*), acesta are un rol de releu pentru staţiile (STA în terminologie 802.11) din interiorul unui BSS, după cum se va vedea în continuare.
Există trei tipuri de topologii pentru o rețea de tip WLAN. Acestea sunt:

- Independent basic service set (IBSS)
- Basic service set (BSS)
- Extended service set (ESS)

Prin „service set" se înțelege un grup format dintr-un anumit număr de dispozitive echipate cu interfețe 802.11.

Independend basic sevice set

În acest tip de topologie rețeaua WLAN este alcătuită din dintr-un grup de staţii care comunică direct unele cu altele şi de aceea mai este numită şi rețea ad-hoc (figura 2-3). La acest tip de configuraţie nu este necesara prezenţa unui Access Point, iar standardul nu prevede un număr limita de staţii care sa facă parte dintr-o rețea ad-hoc.

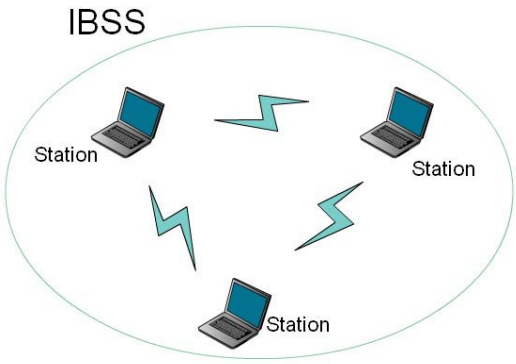


Figura 2-3 Arhitectura unui IBSS

Basic service set

În această situaţie staţiile nu vor mai comunica direct între ele, ci doar cu un dispozitiv specializat, numit Acceess Point (AP). Astfel se creează o topologie de tip celular, o celulă fiind alcătuită dintr-un AP şi staţiile conectate la el (figura 2-4). În acest caz comunicaţia între staţii se realizează prin intermediul AP-ul. De obicei, în acest tip de configuraţie, AP-ul beneficiază de o legătură uplink la o rețea Ethernet, conectând în acest fel staţiile din interiorul BSS-ului la această rețea externă.

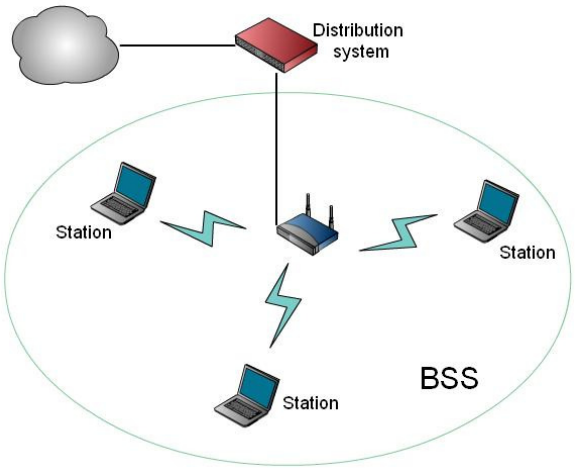


Figura 2-4 Arhitectura unui BSS

Extended sevice set

Mai multe AP-uri pot fi conectate între ele prin intermediul unei infrastructuri (ex: Ethernet), iar această infrastructură, conform standardul WLAN, are denumirea de DS (Distribution System). În felul acesta se crează o colecție de BSS-uri interconectate (figura 2-5), care poartă denumirea de ESS (Extended Service Set).

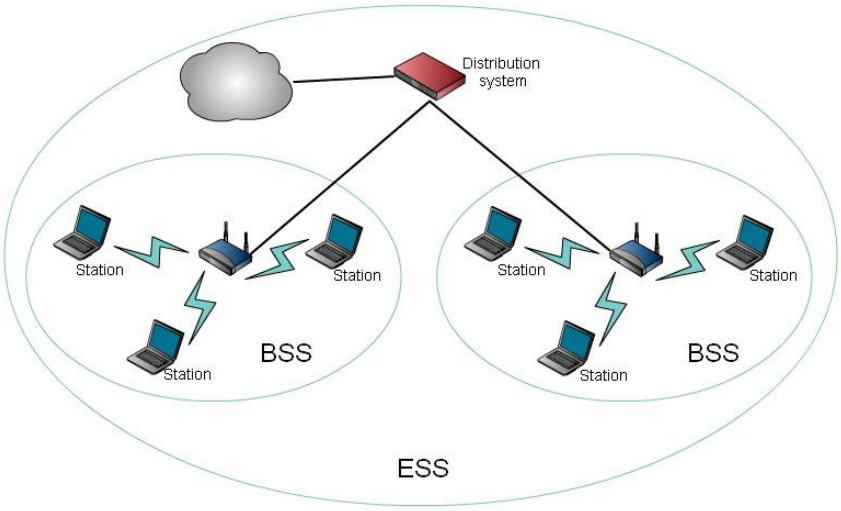


Figura 2-5 Arhitectura unui ESS

2.3 Serviciile oferite de o rețea 802.11

Standardul specifică nouă tipuri de servicii care trebuie implementate de echipamentele care vor să fie conforme cu acesta. Dintre acestea, şase sunt folosite pentru operații de management şi trei pentru transferul datelor [Gas02].

1) *Distribution*
Acest serviciu este folosit de către staţiile mobile atunci când se află intr-o arhitectură de tip *infrastructure*. Atunci când un *access point* primeşte un *frame*, pentru a stabili care este destinatarul acelui *frame* trebuie sa facă apel la serviciul *distribution.* Toate frame-urile care tranzitează un *access point* sunt dirijate către destinaţie făcându-se apel la cest serviciu.

2) *Integration*
Integration este un serviciu oferit de către *distribution system*, atunci când o reţea 802.11 trebuie sa se conecteze la un alt tip de reţea, de exemplu o reţea cablată.

3) *Association*
Într-o arhitectura de tip *infrastructure* staţiile mobile trebuie sa fie luate în evidenţă de către un *access point.* Aceste proces de „înregistrare" sau „asociere" este posibil făcând apel la serviciul *association.* Procesul de asociere este esenţial, deoarece oferă mecanismele necesare ca atunci când în cadrul unei arhitecturi in care există mai multe BSS-uri conectate printr-un sistem de distribuţie să se poate identifica *access point-ul* la care este conectat nodul destinaţie.
Staţiile care nu sunt asociate la nici un *access point* nu fac parte din reţeaua wireless, fiind in incapacitate de a comunica cu oricare dintre nodurile reţelei.

4) *Reassociation*
Acest serviciu este invocat în situaţia in care există mai multe BSS-uri, fiecare BSS având propriul *access* point (AP), iar o staţie se deplasează in zona acoperită de acele AP-uri. Atunci când staţia sesizează că puterea semnalului provenit de la AP-ul la care ea este asociată scade, iar un alt AP oferă o calitate mai bună a semnalului, atunci este declanşat procesul de reasociere prin care staţia va fi asociată de acum încolo la nou AP.

5) *Dissassociation*
Acest serviciu este invocat atunci când o staţie doreşte să părăsească reţeaua. Nu este neapărat necesar ca o staţie care părăseşte reţeaua să facă apel la acest serviciu, deoarece subnivelul MAC a fost proiectat să trateze şi o situaţie de genul acesta, când dintr-un motiv oarecare o staţie părăseşte reţeaua fără sa anunţe acest lucru în mod explicit.

6) *Authentication*
Oferă mecanismele de acces limitat la resursele reţelei prin aceea că doar staţiile care s-au autentificat au dreptul, după ce procedura de asociere a avut loc cu succes, sa beneficieze de serviciile oferite de AP-ul la care s-a făcut asocierea.

7) *Deauthentication*
Acest procesul marchează momentul de încheiere a perioadei in care staţia a fost autentificată.

8) *Privacy*
Oferă mecanismele de protecţie împotriva interceptării nedorite a datelor vehiculate în reţeaua wireless.

2.4 Subnivelul MAC

2.4.1 Metode de acces la mediul fizic

Standardul 802.11 acoperă zona nivelului fizic precum şi ce a nivelului legătură de date din modelul de referinţă OSI. Din nivelul legătură de date, standardul tratează doar subnivelul MAC, conform figurii 2-6.

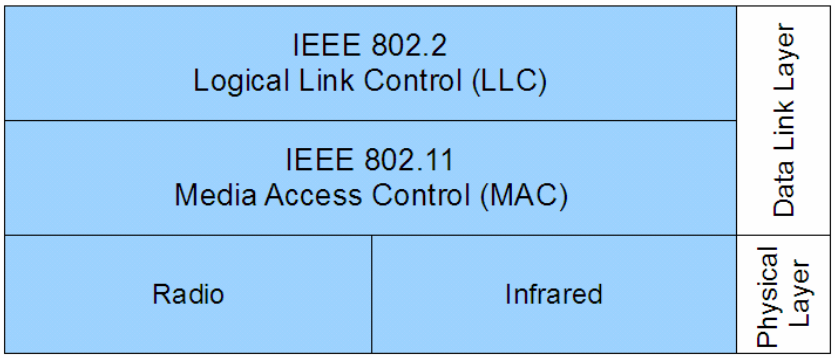


Figura 2-6 Subnivelul MAC gestionat de standardul 802.11

Subnivelul MAC interacţionează cu nivelul fizic conform figurii. Deşii standardul prevede şi undele infraroşii ca posibil mediu de transmisie, nu au fost dezvoltate dispozitive din această categorie. Motivul pentru care aceasta tehnologie nu a prins teren in cazul standardului 802.11, a fost acela ca undele infraroşii au limitări foarte mari in ceea ce priveşte propagarea, ele putând fii foarte uşor obturate, neavând capacitatea de a penetra obstacolele ca si undele radio. Un alt dezavantaj este acela că interfeţele IrDA existente, permit doar rate mici de transfer, pana la 115kbit/s.

Avantajul folosirii undele radio ca mediu de transmisie pentru standardul 802.11 este acela ca undele radio pot penetra obstacolele si de asemenea, permit rate de transfer mai mari. Dezavantajul folosirii undelor radio este acela ca pot fii afectate de interferente.

Conform standardului, subnivelul MAC defineşte două metode de acces la mediul fizic. Cea de bază este *Distributed Coordination Function* (DCF), iar *Point Coordination Function* (PCF) este opţională. Metoda DCF este de tip asincron, iar PCF este de tip sincron. De fapt, PCF se bazează în funcţionare pe mecanisme oferite de DCP. Prezenţa metodei PCF nu este obligatorie şi de aceea în momentul de faţă ea nu este implementată deoarece este prea complexă şi deci ar duce la ridicarea costurilor de fabricaţie.

În cazul WLAN subnivelul MAC trebuie să îndeplinească următoarele operaţii:

- fragmentarea pachetelor
- transmisia pachetelor
- retransmisia pachetelor
- confirmarea pachetelor

În cazul Ethernet metoda de acces la mediu era CSMA/CD (Carrier Sense Multiple Access with Collision Detection), ceea ce înseamnă ca sunt detectate cazurile de coliziune, adică de transmitere simultană de pachete. Pentru un WLAN această metodă de abordare nu este practică datorită specificului diferit al mediul de transmisie. Astfel este mult uşor de implementat un mecanism pentru evitarea situaţiei de coliziune, decât pentru detecţia ei. Metoda *Carrier Sense Multiple Access with Collision Detection* (CSMA/CD) folosită în reţelele Ethernet nu ar fi practică în acest caz din două motive:

- necesită implementarea unui mecanism full duplex de comunicaţie între staţii, ceea ce ar conduce la costuri ridicate de producţie
- nu există certitudinea că staţiile se „aud" toate între ele, adică este posibil ca cel care transmite să creadă că mediul este liber, dar de fapt în zona receptorului mediul să fie ocupat

Următoarea figură ilustrează problema cunoscută în literatura de specialitate sub denumirea: *the hidden node problem*. Aceasta apare atunci când unul din nodurile care vrea să obţină canalul de comunicaţie pentru a transmite date, „ascultă" să vadă daca acesta este liber, conform specificaţiilor prevăzute în standard.

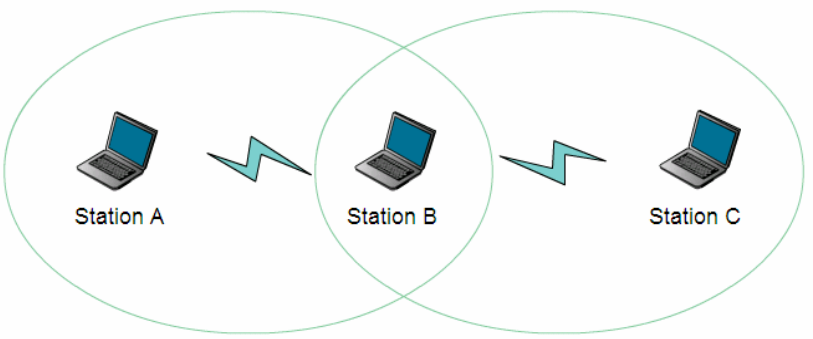


Figura 2-7 "The hidden node problem"

În figura 2-7 este redată situaţia când, din cauza distanţei, nodul A nu sesizează că nodurile B şi C comunică. Acelaşi lucru se întâmplă şi cu nodul C când nodurile A şi B comunică.

DCF are la bază protocolul CSMA/CA (Carrier Sense Multiple Access with Collision Avoidance). O staţie care doreşte să transmită „ascultă" mediul şi în cazul în care este liber trece la transmisie. Dacă mediul este ocupat atunci amână transmisia pentru mai târziu.

Există două modalităţi de a detecta dacă mediul fizic este liber sau nu. Prima metodă se bazează pe detectarea prezenţei altor transmisii prin ascultarea propriu-zisă a mediului (*Physical Carrier Sense*), analizând toate pachetele transmise de celelalte staţii, iar a doua metodă se bazează pe o ascultare virtuală a mediului (*Virtual Carrier Sense*).

În continuare vom descrie primul mecanism. O transmisie între două staţii se desfăşoară în două etape:

- Staţia care transmite ascultă mediul. Daca acesta este ocupat amână transmisia, iar dacă este liber pentru o perioada de timp egala cu DIFS (Distributed Inter Frame Space) poate trece la transmiterea pachetele. Deoarece există o probabilitate destul de mare ca două staţii care sesizează că mediul este liber să încerce să transmită simultan, există un mecanism de evitare a unor astfel de situaţii, prin care staţiile mai aşteptă un interval de timp aleatoriu, după care încep să transmită.

- Staţia care recepţionează pachetele verifică suma de control care le însoţeşte, iar apoi le confirmă printr-un pachet de tip ACK. Dacă sursa primeşte pachetele de confirmare înseamnă că nu a avut loc nici o coliziune. Dacă nu se primeşte confirmarea înseamnă că a avut loc o coliziune şi pachetul care nu a fost confirmat este retransmis.

În 802.11 sunt practicate confirmările pozitive (*positive acknowledge*). Aceasta înseamnă că vor fi confirmate doar pachetele care ajung la destinaţie fără să fie afectate de erori (figura 2-8). Daca un pachet a fost afectat de eroare, atunci la destinaţie acesta este ignorat.

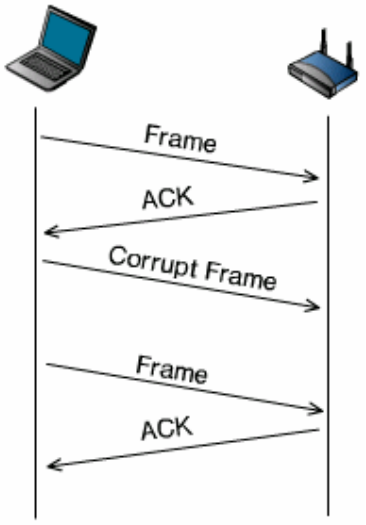


Figura 2-8 Confirmările pozitive practicate în 802.11

Dacă în procesul de transmisie ar fi implicat doar mecanismul de tip *Physical Carrier Sense*, atunci o comunicaţie s-ar desfăşura în felul următor. Staţia care doreşte să transmită urmăreşte mediul până când detectează că mediul liber. Din acest moment mai aşteaptă un interval de timp egal cu valoarea *DIFS (Distributed InterFrame Space)***,** după care începe să transmită datele. DIFS este intervalul minim pe durata căruia mediul de transmisie trebuie să fie liber, pentru ca o staţie vrea să transmită date să o poată face şi are o durată de 128 µs. După ce a fost transmis pachetul de date, receptorul aşteaptă un interval de timp numit *SIFS (Short Inter Frame Space)* pentru a transmite pachetul de confirmare şi având o durată de 28 µs. Acest interval de timp SIFS are o durată fixă şi este folosit pentru a separa două transmisii aparţinând aceluiaşi dialog între două staţii. Este ales în aşa fel încât să îi permită staţiei

transmiţătoare să treacă din modul de transmisie în modul de recepţie. În realitate, în momentul când o staţie simte că mediul este liber, nu trece automat la transmisia de date, deoarece este posibil ca mai multe staţii să facă acelaşi lucru simultan. Pentru a rezolva acest aspect lucrurile se desfăşoară în felul următor.

Cât timp are loc transmisia celelalte staţii îşi amână tentativele de a transmite, până când mediul devine liber pentru un interval de timp egal cu DIFS şi apoi aplică un algoritm de tip *backoff* menit să soluţioneze problema accesului simultan la mediu de către staţiile care doresc acces in acelaşi timp. Metoda presupune că fiecare staţie să aleagă un număr aleator cuprins intre 0 si o valoare denumita *CW (Contention window)*, iar apoi să aştepte un interval de timp egal cu produsul dintre acel număr şi un interval de timp numit *Slot time*, cu alte cuvinte să aştepte un anumit număr de slot-uri de timp şi doar apoi, dacă mediul a rămas în continuare liber, să treacă la transmisia de date.

$$BackoffTime = Random() \times SlotTime$$

Pentru a reduce probabilitatea unor coliziuni, situaţie care apar frecvent pentru cazul descris de către „*the hidden node problem*”, standardul a prevăzut şi metoda *Virtual Carrier Sense*, descrisă in continuare (figura 2-9).

O staţie care vrea să transmită date, mai întâi trimite un scurt pachet de control numit *RTS (Request to Send)*, care include adresa sursei, adresa destinaţiei şi durata transmisiei care va avea loc, această durată incluzând şi recepţia pachetului de confirmare, în scopul de a rezerva mediul pentru propria transmisie. Dacă mediul este liber, atunci staţia destinaţie răspunde cu un pachet numit *CTS (Clear to Send)*, care conţine aceleaşi informaţii legate de durata transmisiei.

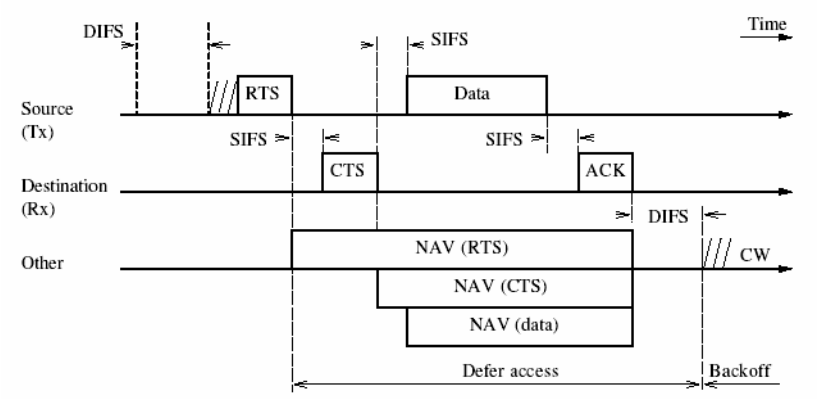


Figura 2-9 ***Virtual Carrier Sense***

Când staţiile învecinate recepţionează fie un pachet RTS fie un pachet CTS îşi setează un indicator numit *NAV (Network Allocation Vector)* în conformitate cu informaţia de timp conţinută în aceste pachete. Acesta este de fapt un timer care este decrementat şi doar când ajunge la zero staţia poate încerca să transmită din nou, dacă mediul este liber. Daca una dintre staţii nu recepţionează pachetul RTS, nefiind în aria de acoperire a acelei staţii, atunci ea va recepţiona pachetul CTS, care vine ca răspuns

la RTS. Prin acest mecanism este rezolvată şi „problema nodului ascuns”. Chiar dacă staţia C nu „aude” pachetul de tip RTS, ea va recepţiona pachetul CTS trimis de staţia B (figura 2-10). Pe baza informaţiei din acest pachet îşi va seta indicatorul NAV.

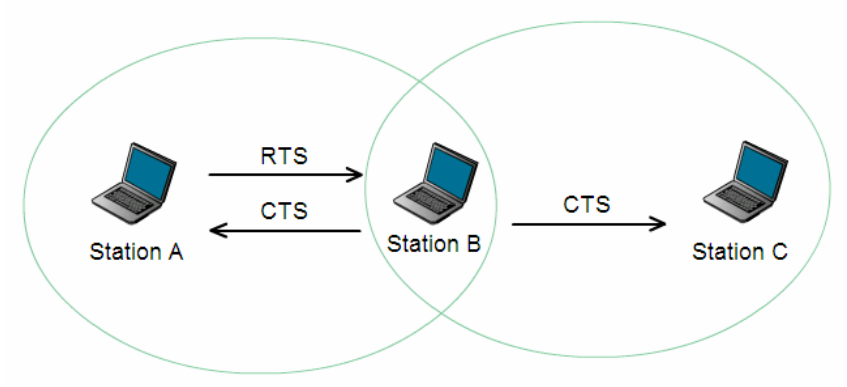


Figura 2-10 Eliminarea lui “the hidden node problem” prin mecanismul RTS-CTS

O staţie care vrea să transmită va aştepta un interval de timp egal cu valoarea dată de NAV, iar apoi apelează la algoritmul de tip *backoff* pentru a calcula momentul transmisiei. Mecanismul oferit de timer-ul NAV nu implică neapărat folosirea pachetelor RTS/CTS. Există situaţii când pachetele de date conţin informaţii de timp care duc la actualizarea timer-ului NAV. În figura 2-11 sunt sintetizate condiţiile care trebuie îndeplinite pentru ca o staţie sau un access point să poată trece la transmisia unui frame.

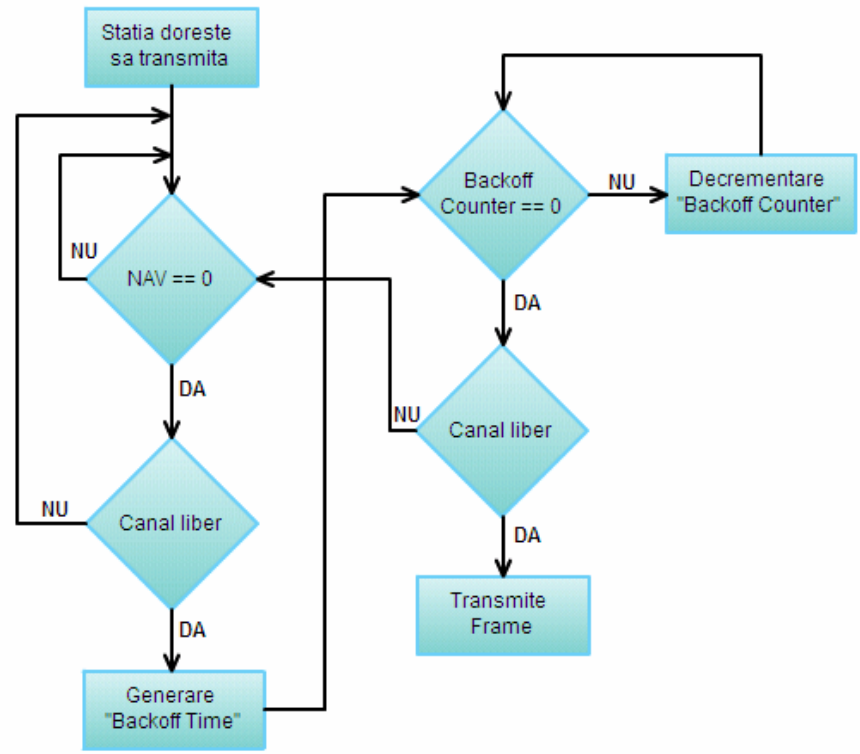


Figura 2-11 Transmisia unui frame, conform DCF (Distributed Coordination Function)

Pentru cazul ilustrat în figura 2-9, presupunem că înainte de transmisia datelor au fost parcurşi toţi paşii prezentaţi în diagrama din figura 2-11, pentru a determina dacă sunt îndeplinite toate condiţiile pentru a transmite date.

Dacă este activat mecanismul RTS/CTS capacitatea de transfer a reţelei este diminuată. De aceea acest mecanism este eficient doar în cazul în care există o densitate relativ mare de staţii şi există riscul apariţiei fenomenului *the hidden node problem*.

2.4.2 Scanarea activă şi pasivă

Procesul prin care se identifică reţelele wireless disponibile se numeşte scanare. În cadrul procesului de scanare este interesant de prezentat faptul ca există două modalităţi de a realiza acest lucru, denumite *scanare pasivă* şi *scanare activă*.

Scanarea presupune că staţia care realizează acest lucru nu face altceva decât să treacă de pe un canal pe altul şi să aştepte transmisia unor frame-uri speciale numite *beacon frames*. Aceste frame-uri conţin toate informaţiile necesare despre reţeaua respectivă. Scanarea pasivă este avantajoasă din punctul de vedere al consumului de energie, ea nepresupunând transmisia din partea staţiei a nici unui tip de frame-uri.

În cadrul scanării active, pe fiecare canal în parte, staţia trimite frame-uri de tipul *Probe Request*. Dacă există un AP care funcţionează pe canalul respectiv, atunci el răspund printr-un frame de tipul *Probe Response*.

2.4.3 Autentificarea

Autentificarea a fost introdusă pentru a preveni accesul neautorizat al unor staţii la o reţea wireless. Există posibilitatea de a dezactiva sistemul de autentificare, în acest caz spunem ca avem de a face cu un sistem deschis. Dacă se optează pentru un sistem cu autentificare atunci avem la dispoziţie una din metodele puse la dispoziţie de standard, dar care nu fac obiectul acestei prezentării. Odată procesul de autentificare încheiat, staţia poate trece la următorul pas, care este *asocierea* la un anumit AP. Există posibilitatea ca o staţie să realizeze autentificarea faţă de mai multe AP-uri, chiar dacă apoi asocierea se va realiza doar faţă de un singur AP. Acest proces se numeşte preautentificare şi poate fi folositor în procesul de roaming, atunci când o staţie îşi schimbă asocierea de la un AP la alt AP, ducând în felul acesta la o economie de timp.

2.4.4 Procedura de asociere

Odată ce a fost încheiată procedura de autentificare, staţia poate opta oricând pentru a se asocia la un anumit AP. Etapele parcurse pentru asociere sunt redate în figura 2-12.

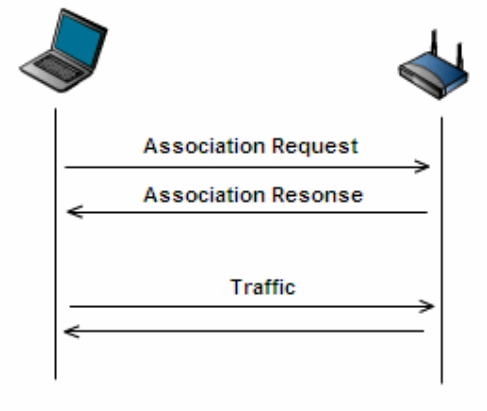


Figura 2-12 Procedura de asociere la un AP

Staţia transmite un frame de tip *Association Request*, iar dacă procedura de autentificare a decurs cu succes, AP-ul va răspunde printr-un frame de tip *Association Response*. Dacă autentificarea nu s-a realizat încă, sau nu s-a realizat cu succes, atunci răspunsul AP-ului este un frame de tip *Deauthentication*. În urma procesului de asociere staţiei îi este furnizat un număr numit *Authentication ID*, acesta fiind un identificator logic cu ajutorul căruia staţia este diferenţiată faţă de restul staţiilor care sunt asociate la acelaşi AP.

2.4.5 Procedura de reasociere

Atunci când o staţie pierde conexiunea cu AP-ul la care este în mod curent asociată începe procesul de reasociere la un alt AP. În primul rând are loc o scanare pentru identificarea AP-urilor disponibile care fac parte din acelaşi ESS, apoi are loc o procedură de asociere clasică, ca în subcapitolul 2.4.4.

2.4.6 Roaming-ul

Prin roaming se înţelege procesul prin care o staţie se deplasează dintr-un BSS în altul fără a se întrerupe conexiunea realizată de protocoalele de pe nivele superioare. Această funcţie este similară cu procesul de *handover* prezent în telefonia mobilă, atunci când un abonat trece dintr-o celulă în alta.

Standardul nu specifică exact cum se să se realizeze acest lucru, dar defineşte elementele care să fie folosite în acest scop. Acestea sunt scanarea activă sau pasivă şi reasocierea. În cazul reţelelor 802.11

2.4.7 Sincronizarea

Toate staţiile din interiorul unei BSS trebuie să fie sincronizate după un ceas comun. Acest lucru este posibil printr-o funcţie de sincronizare numită *Timing Syncronization Function* (TSF) şi care impune ca fiecare staţie să menţină un TSF timer. AP-ul din interiorul unui BSS este considerat *the timing master* şi este responsabil cu sincronizarea staţiilor. La intervale periodice de timp transmite scurte pachete denumite *beacons*, care conţin copii ale propriului TSF timer. Transmiterea acestor pachete se face la intervale regulate de timp date de un parametru numit *Beacon Period*.

2.4.8 Fragmentarea şi reasamblarea pachetelor

Dimensiunea pachetelor folosite de protocoalele dintr-o reţea locală sunt de câteva sute de octeţi. Cel mai lung pachet folosit în Ethernet are dimensiunea de 1518 octeţi. În cazul reţelelor wireless, dimensiuni mari ale pachetelor nu sunt potrivite din mai multe motive:

- datorită ratei mari a erorilor prezente în acest tip de reţea, cu cât pachetele sunt mai mari cu atât creşte probabilitatea de a fi afectate de erori.
- dacă pachetul a fost afectat de erori retransmisia lui va duce la o încărcare mare a reţelei

Pentru a putea face ca reţeaua wireless să poată interacţiona cu o reţea cablată Ehternet se foloseşte un procedeu de fragmentare a pachetelor (figura 2-13). Protocolul folosit este de tipul transmite şi aşteaptă. După transmisia unui fragment se aşteaptă recepţia pachetului de confirmare şi abia apoi se trimite următorul fragment. Dacă nu se reuşeşte trimiterea unui fragment după mai multe tentative nereuşite se abandonează transmisia întregului pachet din care făcea parte fragmentul. Standardul prevede ca unei staţii să nu i se permită transmiterea către o altă adresă atâta timp cât se încearcă retransmisia unui fragment.

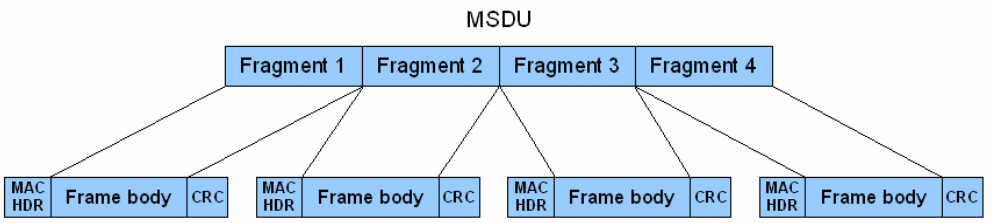


Figura 2-13 Fragmentarea unui frame de dimensiune prea mare

Dimensiunea fragmentelor nu trebuie să depăşească o anumită valoare numită *Fragmentation Threshold*. De asemenea fragmentele au dimensiune fixă, excepţie făcând ultimul fragment care poate să aibă dimensiune mai mică.

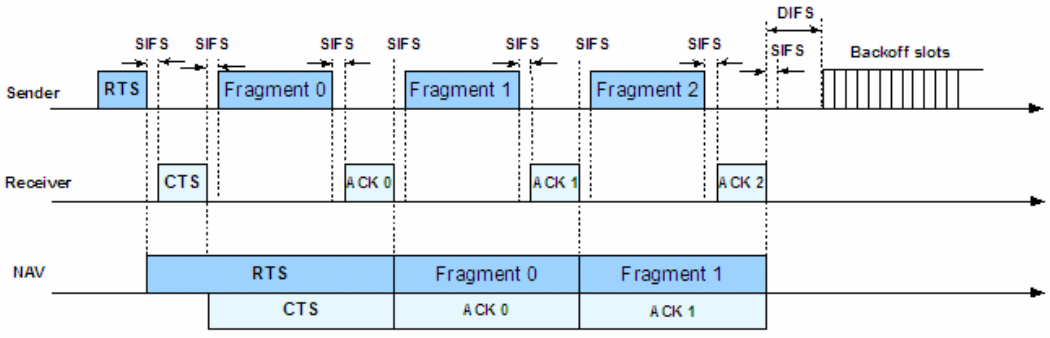


Figura 2-14 Fragmentele sunt transmise sub forma de *burst*

Intervalul de timp dintre un pachet de confirmare şi următorul fragment este egal cu SIFS, astfel că staţia care transmite va reţine rezervat canalul pe toată durata transmiterii fragmentelor, deoarece, aşa cum prevede standardul, dacă celelalte staţii

nu găsesc canalul liber pentru un interval de timp cel puțin egal cu DIFS, nu vor încerca să transmită date. Dacă în plus este activat si mecanismul RTS/CTS atunci rezervarea canalului se face la început prin pachetele RTS şi CTS, iar apoi fiecare fragment împreună cu pachetul de confirmare vor actualiza valoarea pentru timerul NAV, pe care stațiile care urmăresc mediul şi-l vor seta în mod corespunzător. Ultimul fragment va avea valoarea 0 pentru NAV.

2.4.9 Formatul frame-urilor

În cazul nivelului legătură de date, pachetele poartă denumirea de frame-uri, dar pentru simplificare exprimării am folosit până acum tot denumirea de pachete.

Standardul prevede trei tipuri diferite de frame-uri:

- *Data Frames*
- *Control Frames*
- *Managent Frames*

Data Frames sunt folosite pentru transmisia datelor. **Control Frames** sunt folosite pentru controlul accesului la mediul fizic (ex. RTS, CTS, ACK), iar **Management Frames** sunt transmise la fel ca frame-urile de date, dar conțin informații pentru managemetul resurselor (ex. *Beacon Frames*). Fiecare tip de frame este format la rândul lui din mai multe subtipuri.

Frame-urile au următorul format general (figura 2-15):

Figura 2-15 Formatul general al unui frame

Preamble

Conține două tipuri de informații:

- *Synch*: o secvență formată din 80 de biți de zero şi unu care alternează şi care este folosită de către circuitele de pe nivelul fizic pentru a selecta cea mai apropiată antenă şi pentru a se sincroniza în vederea recepției frame-ului.
- *SFD*: *Start Frame Delimiter* reprezintă o secvență de 16 biți sub forma 0000 1100 1011 1101

PLCP Header

Acest header este transmis întotdeauna cu o rată de transfer de 1Mbps şi conține informații necesare nivelulului fizic pentru decodarea frame-ului. Aceste informații sunt:

- *PLCP_PDU Length Word*: reprezintă numărul de octeți prezenți în pachet. Această informație este utilă pentru a determina cu exactitate sfârsitul pachetului.
- *PLCP Signaling Field*: care coține codificată rata de transfer care se doreşte a fi folosită
- *Header Error Check Field*: câmp pentru detecția erorilor la nivelul header-ului

- *MAC Data*

În acest câmp sunt prezente frame-urile propriu-zise folosite de către subnivelul MAC. Formatul general este redat în figura 2-16.

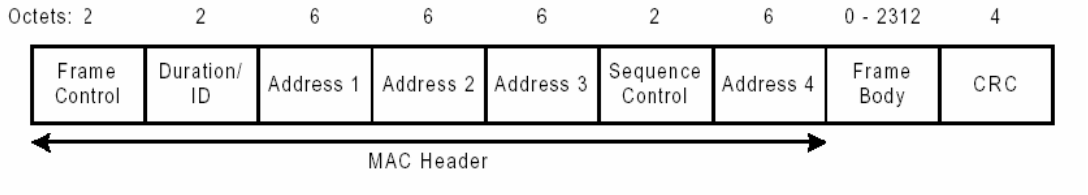


Figura 2-16 Structura MAC Header

Fiecare frame este format dintr-un header, o zonă de informaţii şi un câmp pentru controlul erorilor. Header-ul conţine la rândul lui mai multe subcâmpuri. Unele din aceste subcâmpuri pot să lipsească din anumite frame-uri.

Subcâmpul *Frame Control* are următoarea structură (figura 2-17):

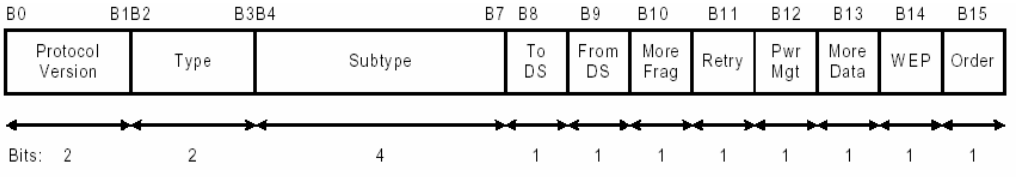


Figura 2-17 Subcâmpul Frame Control

Type şi *Subtype*
Indică tipul frame-ului după cum se observă în tabelul din figura 2-18.

ToDS

Este setat cu valoarea 1 atunci când frame-ul trebuie redirectat de AP către *Distribution System* sau pentru orice frame de date trimis de către o staţie către AP-ul la care este asociat şi are valoarea 0 în celelalte cazuri.

FromDS

Este setat cu valoarea 1 când frame-ul provine de la *Distribution System.*

More Fragments

Este setat pe 1 când mai există fragmente de trimis aparţinând aceluiaşi frame.

Retry

Indică faptul că fragmentul este o retransmisie. Acest câmp este folosit de receptor pentru a identifica fragmentele duplicat atunci când s-a pierdut pachetul de confirmare.

Type Value b3 b2	Type Description	Subtype Value b7 b6 b5 b4	Subtype Description
00	Management	0000	Association Request
00	Management	0001	Association Response
00	Management	0010	Reassociation request
00	Management	0011	Reassociation response
00	Management	0100	Probe request
00	Management	0101	Probe response
00	Management	0110-0111	Reserved
00	Management	1000	Beacon
00	Management	1001	Announcement traffic indication message (ATIM)
00	Management	1010	Disassociation
00	Management	1011	Authentication
00	Management	1100	Deauthentication
00	Management	1101-1111	Reserved
01	Control	0000-1001	Reserved
01	Control	1010	Power Save (PS)-Poll
01	Control	1011	RTS
01	Control	1100	CTS
01	Control	1101	Acknowledgment (ACK)
01	Control	1110	Contention-Free (CF)-End
01	Control	1111	CF-End+CF-Ack
10	Data	0000	Data
10	Data	0001	Data+CF-Ack
10	Data	0010	Data+CF-Poll
10	Data	0011	Data+CF-Ack+CF-Poll
10	Data	0100	Null data (no data transmitted)
10	Data	0101	CF-Ack (no data transmitted)
10	Data	0110	CF-Poll (no data transmitted)
10	Data	0111	Data+CF-Ack+CF-Poll
10	Data	1000-1111	Reserved
11		0000-1111	Reserved

Figura 2-18 Tipurile de frame-uri definite de standardul 802.11

Ne întoarcem acum la câmpurile din MAC Header.

Duration

Conține informația de timp folosită pentru actualizarea lui NAV. Toate stațiile sunt obligate să monitorizeze header-ele tuturor frame-urilor care sunt vehiculate în rețea. Valoare prezentă în acest câmp reprezintă durată în microsecunde a comunicației care se află în desfăşurare. Dacă valoarea curentă prezentă în NAV este mai mică decât valoare citită, atunci NAV-ul se actualizează cu noua valoare.

Adress Fields

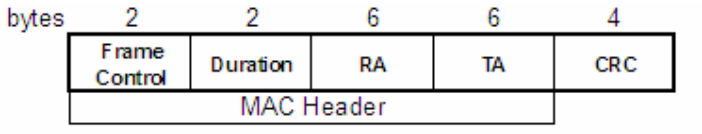
Aceste câmpuri sunt setate în funcție de valoarea subcâmpurilor ToDS şi FromDS şi reprezintă în principiu adresa transmițătorului şi a receptorului.

Sequence Control

Este folosit pentru numerotarea fragmentelor atunci când are loc procesul de fragmentare. Este compus din două subcâmpuri numite *Fragment Number* şi *Sequence Number*, care definesc numărul frame-ului şi numărul fragmentului dintr-un anumit frame.

În continuarea va fi prezentată succint structura câtorva din cele mai folosite frame-uri.

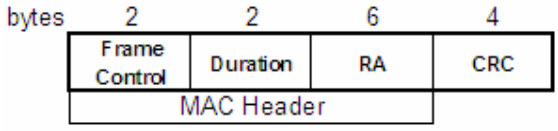
RTS Frame



RA (*Receiver Address*) este adresa staţiei către care se doreşte să se transmită un frame de date sau management, iar *TA* (*Transmitter Address*) este adresa staţiei care a transmis acest frame.

Duration este timpul exprimat în milisecunde necesar pentru transmisia frame-ului de date sau de management la care se adaugă durata un frame CTS, plus durata un frame de confirmare (ACK) şi plus trei intervale SIFS.

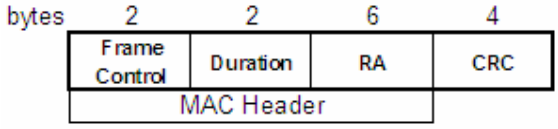
CTS Frame



Este frame-ul cu care care se răspunde la un frame RTS. Valoarea pentru RA este copiată din câmpul TA al frame-ului RTS.

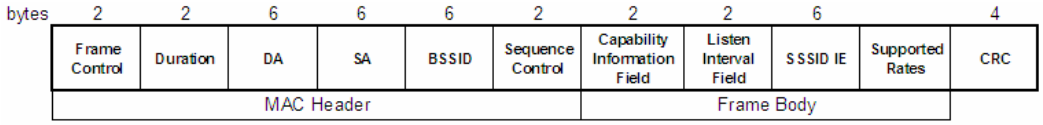
Valoarea pentru *Duration* se obţine scăzând din valoarea citită din frame-ul RTS a timpului necesar trasmiterii frame-ului CTS şi a intervalul SIFS aferent.

ACK Frame

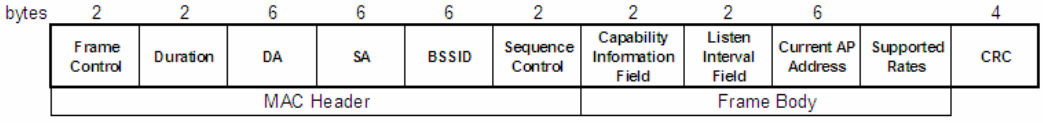


Valoarea pentru RA este copiată din câmpul Address2 al frame-ului pe care îl confirmă.

Association Request Frame



Reassociation Request Frame



2.5 Concluzii

Scopul acestui capitol este acela de a oferi o descriere a mecanismelor intime implicate în funcționarea unei rețele de tip 802.11. Aceste informații corelate cu cele prezentate în capitolul 1, referitoare la algoritmii de control ai congestiei implementați în protocolul TCP, oferă imaginea clară a cauzelor unui comportament ineficient al TCP-ului într-o rețea wireless de tip 802.11, în anumite circumstanțe, când calitatea conexiunii cu un AP se degradează, fie din cauza distanței prea mari, fie din cauza încărcării AP-ului cu prea mulți clienți.

Capitolul 3 Comportamentul protocolului TCP în rețele WLAN 802.11

3.1 Introducere

S-a văzut în subcapitolul 1.4.2 că protocolul TCP implementează un set de mecanisme pentru controlul congestiei. Acestea au fost proiectate, când nu existau comunicații wireless de date, iar legăturile prin fir ajunseseră la un nivel la care probabilitatea de apariție a erorilor datorate canalului de comunicație devenise foarte mică, iar rarele erori nu afectau pachete consecutive de date. În acest context, atunci când confirmările pentru mai multe pachete de date întârziau să apară, era un indiciu că undeva în rețea, pe traseul dintre sursă şi destinație, se manifesta un fenomen de congestie. Ca măsură luată, protocolul TCP răspundea prin invocarea unor algoritmi care a aveau rolul de a reduce traficul pentru a nu mai alimenta zona de rețea congestionată cu pachete care să înrăutățească situația. Pentru ca efectul să fie cel scontat, adică fenomenul de congestie să dispară, era necesar ca reacția să fie una globală, comună tuturor stațiilor ale căror pachete tranzitau zona congestionată. Acest lucru s-a obținut prin implementarea mecanismelor de control ale congestiei chiar în interiorul protocolului TCP.

Odată cu apariția comunicațiilor wireless lucrurile au suferit nişte modificări. Canalele de comunicație radio sunt mult mai vulnerabile la perturbații decât conexiunile prin cablu astfel că rata de corupere a pachetelor a este cu mult mai mare. Protocolul TCP reacționează la acest fenomen ca şi cum pachetele s-ar fi pierdut din cauza apariție congestiei şi invocă mecanismele prevăzute pentru această situație. Rezultatul este diminuarea ratei de transfer. Consecințele sunt nedorite, pentru că procedând astfel, pe de o parte durata comunicației creşte, pentru că aceiaşi cantitate de informație va fi trimisă cu o cadență mai mică, iar pe de altă parte mediul de comunicație fiind ocupat mai mult timp, va creşte probabilitatea de apariție a unor noi erori care să afecteze transmisia.

Pentru a corecta acest comportament al protocolului TCP au fost propuse de-a lungul timpului mai multe soluții.

În [XP99] se face o analiză pentru o rețea 802.11b şi se pun în evidență performanțele unor transmisii TCP şi UDP în situația unui trafic bidirecțional între echipamente eterogene. În [VP03] este studiat comportamentul TCP-ului în situația în care apare o degradare a puterii semnalului cauzând pierderi multiple ale aceluiaşi pachet TCP.

În [GR05] este investigată situația când mai multe stații accesează simultan acelaşi AP într-o configurație de tip *infrastructure*. Aici problema principală este faptul că pachetele de date şi cele de *ACK* concurează pentru obținerea aceluiaşi canal de comunicație, ducând la o degradare a ratelor de transfer.

Există situații în care într-o rețea wireless între o stație şi rețeaua cablată se interpun mai multe noduri intermediare într-o arhitectură de tip ad-hoc. Această configurație poartă denumirea de rețea *wireless multihop* şi comportamentul TCP-ului într-o astfel de arhitectură este analizat în [FZ03, NH05, KK05].

Atunci când o rețea 802.11 funcționează în configurație ad-hoc apar câteva probleme caracteristice acestui mod de operare. Aceste aspecte sunt studiate în [Ana03, CD05, WG05].

3.2 Metode de analiză a comportamentului protocolului TCP in rețele WLAN

Scopul acestui capitol este de a propune o serie de metode pentru evaluarea comportamentului protocolului TCP intr-o rețea wireless de tipul 802.11. În cele ce urmează vor fi prezentate câteva tehnici, elaborate pentru a oferi posibilitatea evaluării din mai multe perspective a modului de reacție al protocolului TCP.

Pentru a putea analiza comportamentul protocolului TCP au fost propuse două metode. Ele diferă prin cantitatea de informații culese şi nivelul dificultății de implementare. Acestea sunt:

a) Monitorizarea variabilelor interne gestionate de către algoritmii de control ai congestiei [FAM06]
b) Analiza pachetelor unei conexiuni TCP [FM07]

Prima variantă presupune o intervenție la nivelul kernel-ului sistemului de operare pentru a avea acces direct la variabilele responsabile de evoluția algoritmilor de control ai congestiei. Aceasta înseamnă că este nevoie de acces direct şi cu privilegii de administrator asupra maşinii pe care se va face monitorizarea, dar oferă un spectru larg de informații.

A doua metodă presupune monitorizarea traficului dintre două sisteme care comunică, la nivel de pachete TCP. Avantajul metodei este acela că poate fi implementată pe orice fel de platformă, indiferent de sistemul de operare folosit, dar oferă o viziune mai limitată asupra calității unei conexiuni TCP.

3.2.1 Monitorizarea variabilelor interne gestionate de către algoritmii de control ai congestiei

Testele au fost efectuate pentru transferuri de date între un calculator PentiumIV, cu 2GB RAM şi un laptop-ul cu procesor Intel Centrino, tot cu 2GB RAM, acesta beneficiind de interfață wireless 802.11g. Calculatorul cu procesor Pentium IV a fost conectat prin cablu la un access point. Conexiunea laptop-ului la calculatorul de tip desktop s-a făcut prin intermediul unui access point LinkSys WRT54GS.

Ca sistem de operare a fost ales FreeBSD-ul deoarece acesta reprezintă o portare a sistemului de operare BSD, a cărui stivă TCP/IP este considerată un standard de facto de către comunitatea ştiințifică.

Metoda aleasă pentru investigarea comportamentului protocolului TCP este aceea de citire direct din kernel a variabilelor care sunt folosite de către cei patru algoritmi de control ai congestiei descrişi în subcapitolul 1.4.2. Avantajul acestei metode este acela că permite o acuratețe mult mai mare în ceea ce priveşte investigația făcută. Ca să putem face o interpretare cât mai corectă a comportamentului protocolului TCP avem nevoie şi de variabilele *ssthresh* şi *cwnd*.

Sistemul de operare dispune de o opțiune numită TCP_DEBUG, care atunci când este setată permite citirea diferiților parametrii implicați într-o comunicație TCP/IP.

Traficul realizat între cele două calculatoare folosite în test a fost efectuat de către două programe, unul pe post de server care a rulat pe calculatorul PentiumIV, iar

celălalt având rol de client şi care a rulat pe laptop. Cantitatea de date transferată pentru fiecare măsurătoare a fost de aproximativ 20MB. Datele au fost transmise în două moduri: într-o situație, cantitatea de 20MB s-a împărțit în 5 fragmente de câte 4MB fiecare, iar în doua situație împărțirea s-a făcut în 20000 de fragmente de câte 1KB fiecare. Dimensiunea de 1KB a fost aleasă pentru a fi mai mică decât dimensiunea maximă a frame-urilor folosite de protocoalele Ethernet şi 802.11, astfel nemaifiind necesară fragmentarea pachetelor la Nivelul Legătură de Date.

Un alt parametru care a diferit a fost puterea semnalului. Pentru o serie de măsurători transmisiile s-au făcut asigurând o calitate bună a semnalului radio (>-50dBm), plasând laptop-ul în apropierea AP-ului (Zona3), iar pentru celelalte măsurători semnalul a variat între -80dBm şi -70dBm. Aceste puteri s-au obținut plasând laptop-ul în altă încăpere (Zona 2), la distanță de locația AP-ului care a rămas în Zona 3, conform figurii 3-1.

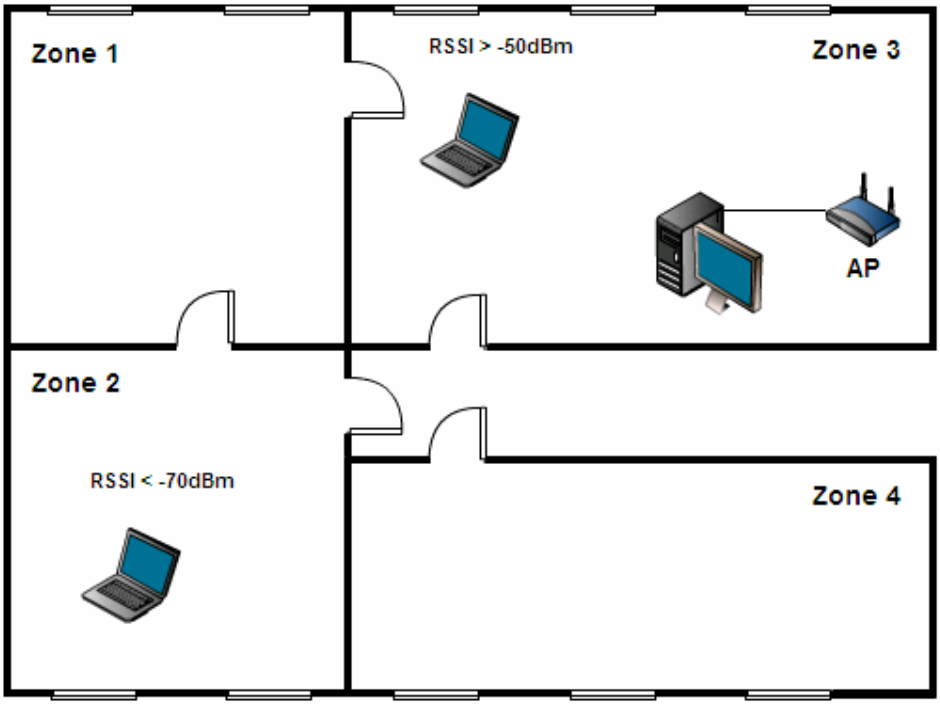


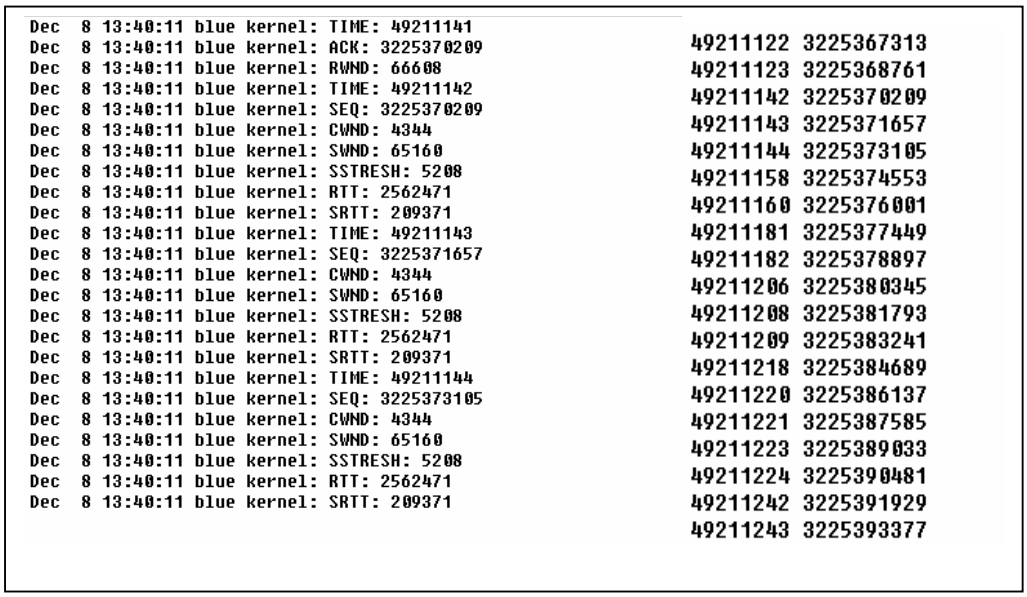
Figura 3-1 Schema amplasării echipamentelor pe durata testelor

Au fost făcute trei tipuri de măsurători:

a) comunicație având semnal radio slab (< -70dBm) şi buffer de transmisie de dimensiune mică (1KB)
b) comunicație având semnal radio slab (< -70dBm) şi buffer de transmisie de dimensiune mare (4MB)
c) comunicație având semnal radio puternic (> -50dBm) şi buffer de transmisie de dimensiune mare (4MB)

Un fragment din fişierul de captură arată ca în figura 3-2 (imaginea din stânga). De aici au fost extrase variabilele *seq*, *ack*, *cwnd* şi *ssthresh*, creându-se pentru fiecare în parte câte un fişier având 2 coloane. Pe prima coloană a fost trecut

timpul, iar pe a doua valorile luate de respectiva variabilă. Imaginea din dreapta din cadrul figurii 3-2 reprezintă un fragment din fişierul creat pentru variabile *seq.*



```
Dec  8 13:40:11 blue kernel: TIME: 49211141
Dec  8 13:40:11 blue kernel: ACK: 3225370209
Dec  8 13:40:11 blue kernel: RWND: 66608
Dec  8 13:40:11 blue kernel: TIME: 49211142
Dec  8 13:40:11 blue kernel: SEQ: 3225370209
Dec  8 13:40:11 blue kernel: CWND: 4344
Dec  8 13:40:11 blue kernel: SWND: 65160
Dec  8 13:40:11 blue kernel: SSTRESH: 5208
Dec  8 13:40:11 blue kernel: RTT: 2562471
Dec  8 13:40:11 blue kernel: SRTT: 209371
Dec  8 13:40:11 blue kernel: TIME: 49211143
Dec  8 13:40:11 blue kernel: SEQ: 3225371657
Dec  8 13:40:11 blue kernel: CWND: 4344
Dec  8 13:40:11 blue kernel: SWND: 65160
Dec  8 13:40:11 blue kernel: SSTRESH: 5208
Dec  8 13:40:11 blue kernel: RTT: 2562471
Dec  8 13:40:11 blue kernel: SRTT: 209371
Dec  8 13:40:11 blue kernel: TIME: 49211144
Dec  8 13:40:11 blue kernel: SEQ: 3225373105
Dec  8 13:40:11 blue kernel: CWND: 4344
Dec  8 13:40:11 blue kernel: SWND: 65160
Dec  8 13:40:11 blue kernel: SSTRESH: 5208
Dec  8 13:40:11 blue kernel: RTT: 2562471
Dec  8 13:40:11 blue kernel: SRTT: 209371
```

```
49211122 3225367313
49211123 3225368761
49211142 3225370209
49211143 3225371657
49211144 3225373105
49211158 3225374553
49211160 3225376001
49211181 3225377449
49211182 3225378897
49211206 3225380345
49211208 3225381793
49211209 3225383241
49211218 3225384689
49211220 3225386137
49211221 3225387585
49211223 3225389033
49211224 3225390481
49211242 3225391929
49211243 3225393377
```

Figura 3-2 Fişier de captură conţinând variabilele citite direct din kernel

a) Comunicaţie având semnal radio slab (< -70dBm) şi buffer de transmisie de dimensiune mică

Primele grafice se referă la numerele de secvenţă care însoţesc pachetele de date şi pachetele de tip *ACK*. Urmărirea numerelor de secvenţă poate să pună în evidenţă numărul şi frecvenţa retransmisiilor. În absenţa erorilor este normal ca numerele de secvenţă precum şi confirmările să aibă o evoluţie constant crescătoare.

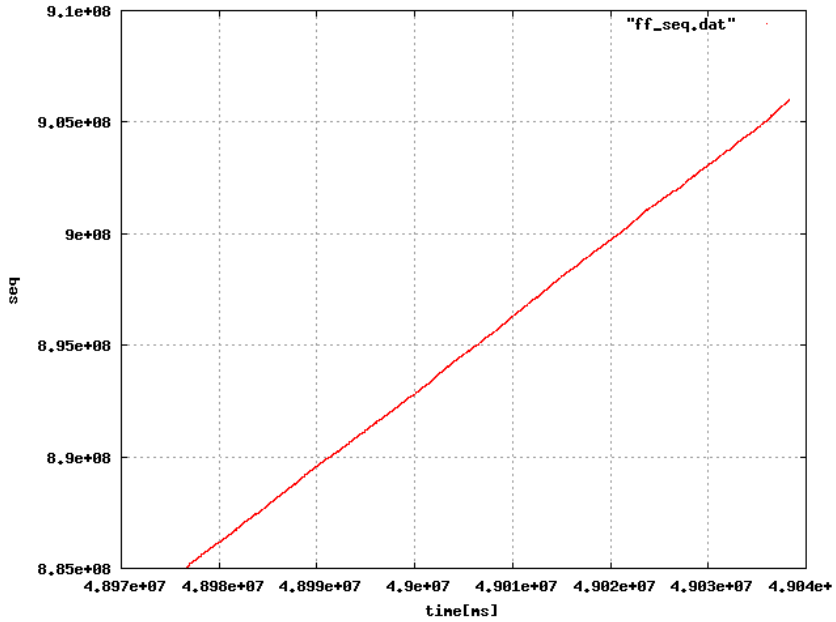


Figura 3-3 Evoluția variabilei *seq* pentru cazul a)

La prima vedere pare că în cazul graficelor din figurile 3-3 şi 3-4 nu există retransmisii de pachete şi rata de transfer are o valoare constantă.

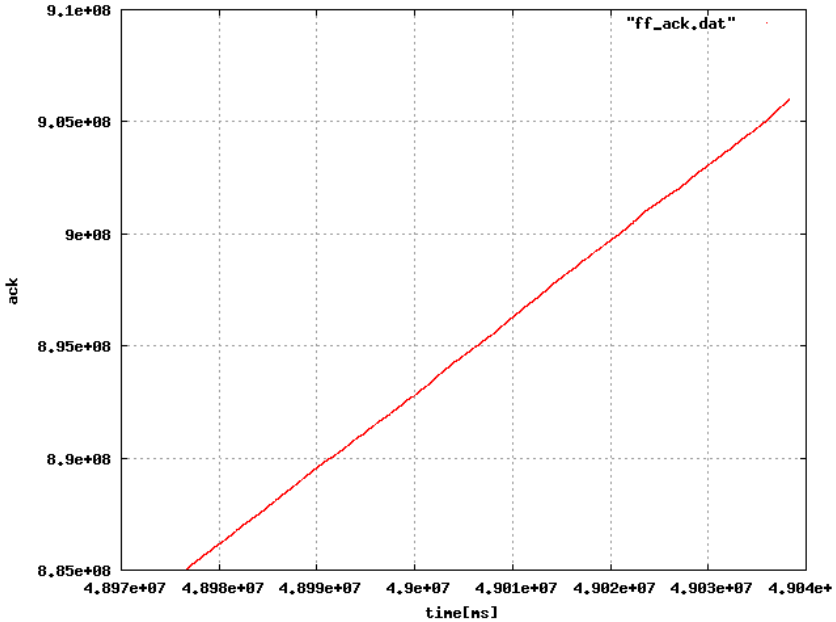


Figura 3-4 Evoluția variabilei *ack* pentru cazul a)

Pentru a observa momentele retransmisiilor trebui trecut la un alt nivel de granularitate şi pentru a obține acest efect se realizează un zoom pe axa timpului, fapt pus în evidență de figura 3-5. Momentele retransmisiilor apar pe grafic ca nişte mici spike-uri orientate în jos. Adâncimea spike-ului arată durata în timp dintre prima transmisie a pachetului şi momentul retransmiterii lui.

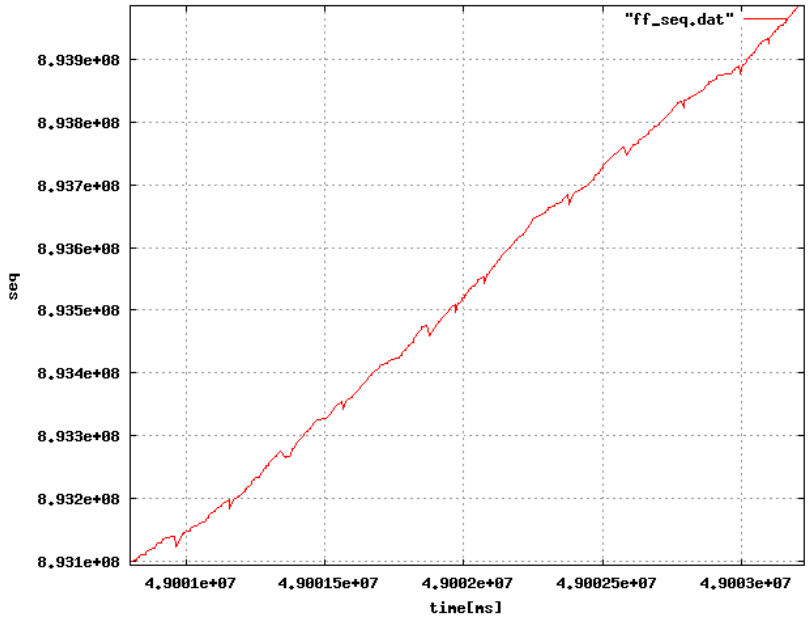


Figura 3-5 Detaliu pe axa timpului privind evoluția variabilei *seq*

Fişierul după care a fost realizate graficele de la cazul a) are aproximativ 9000 de linii. Din analiza lui rezultă 438 de retransmisii. Din totalul retransmisiilor o parte se datorează apariției timeout-ului, obținut pe baza RTO-ului, iar cealaltă fracție se datorează pachetelor de tip *ACK* duplicat (trei pachete consecutive), după cum s-a văzut în subcapitolul 1.4.2.

În cazul în care retransmisia este generată de un timeout, atunci se trece de la *Congestion Avoidance* la *Slow Start*. Este situația cea mai defavorabilă, deoarece *cwnd* ia din nou valoarea de start egală cu SMSS (Sender Maximum Segment Size). După cum s-a văzut, *cwnd* indică dimensiunea ferestrei transmițătorului, deci cantitatea de date care va putea fi trimisă la un moment dat în rețea fără ca transmițătorul să fie nevoit să aştepte sosirea confirmărilor.

Dacă retransmisia este generată de trei pachet de tip *ACK* duplicat atunci se trece de la *Fast Retransmit* urmat de *Fast Recovery*. Înainte de a trece la *Fast Recovery*, *ssthresh* primeşte valoare lui *cwnd* împărțită la doi (figura 1-22), iar după încheierea lui *Fast Recovery*, *cwnd* primeşte valoarea lui *ssthresh*, permițându-se astfel trecerea din nou la *Congestion Avoidance*.

Revenind la analiza fişierului de captură, contorizând numărul trecerilor lui *cwnd* la valoare de start, s-a obținut numărul 149, rezultă că diferența până la 438 se datorează trecerilor de la *Slow Start* sau *Congestion Avoidance*, la *Fast Retransmit*. Acest tip de evaluare cantitativă va fi făcută şi pentru celelalte doua cazuri, b) şi c), sinteza rezultatelor fiind făcută în tabelul din figura 3-18.

Următoarele două grafice (figurile 3-6 şi 3-7) înfățişează evoluția variabilei *cwnd*. Pentru o mai mare claritate s-a făcut afişarea în două moduri. Primul grafic este făcut cu puncte marcând exact valorile pe care *cwnd* le ia, iar al doilea grafic foloseşte linii pentru a unii aceste puncte, punând mai bine în evidență modul de variație al lui *cwnd.* Zona cu puncte mai dese, din partea de mijloc a graficului, reprezintă treceri de la *Congestion Avoidance* la *Fast Retransmit* şi *Fast Recovery,* urmate apoi din nou de reveniri la *Congestion Avoidance.*

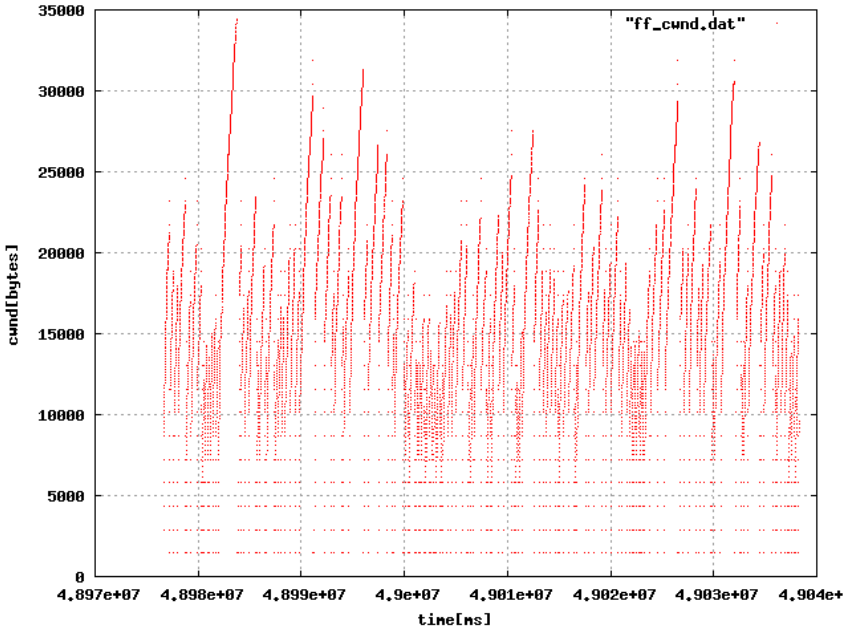


Figura 3-6 Evoluția variabilei *cwnd* pentru cazul a)

În partea inferioară a celor două grafice din figurile 3-6 şi 3-7 se observă trecerile la *Slow Start* fie din faza de *Congestion Avoidance*, fie printr-o reinițializare a lui *Slow Start*.

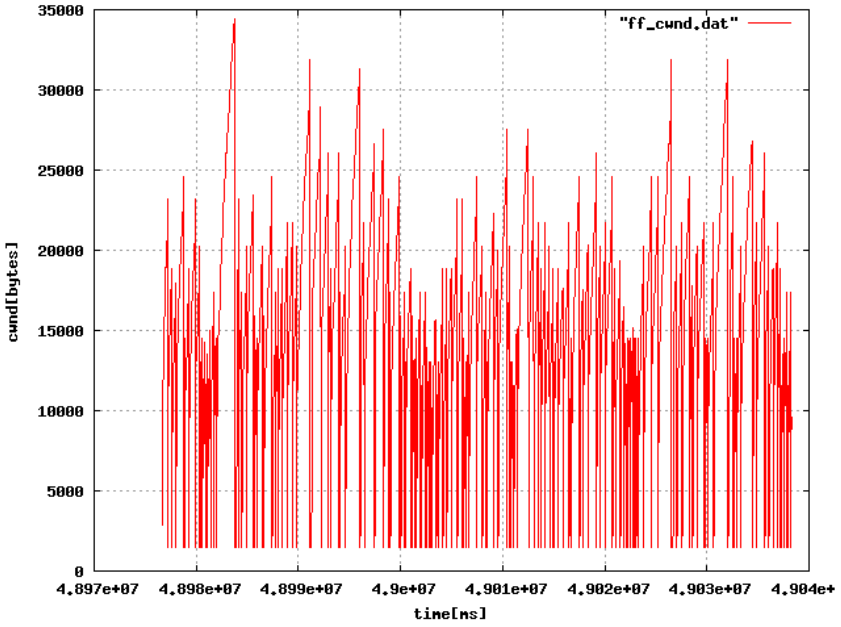


Figura 3-7 Evoluția variabilei *cwnd* pentru cazul a)

Pentru a se vedea mai clar modul de variație al lui *cwnd* s-a făcut un zoom pe axa timpului pentru aceiaşi variabilă *cwnd*. Se observă acum mai clar momentele de trecere de la *Slow Start* la *Congestion Avoidance*, iar apoi revenirile la *Slow Start*,

reprezentate pe grafic ca nişte căderi bruşte la valori mici. Valoare lui *cwnd* cu care debutează *Slow Start* este 1460 octeţi.

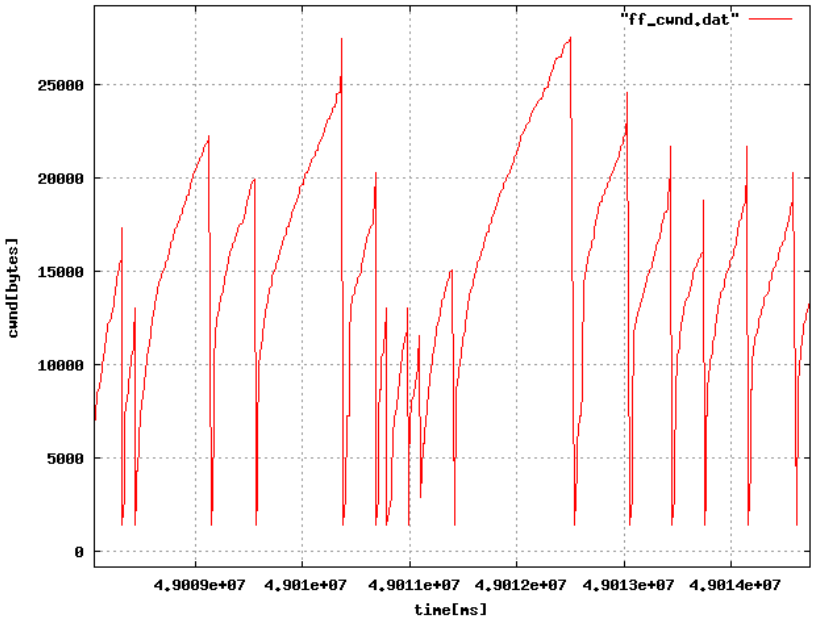


Figura 3-8 Detaliu privind evoluţia variabilei *cwnd*

În acest grafic se observă că nu avem nici o trecere la *Fast Recovery*, ceea ce înseamnă că în această porţiune nu au existat pachete de *ACK* duplicat care să se repete de cel puţin 3 ori, graficul surprinzând doar momente de timeout, marcând trecerea de la *Slow Start* la *Congestion Avoidance*. În secvenţa înfăţişată, majoritatea trecerilor la *Slow Start* sunt făcute din faza de *Congestion Avoidance*, pentru că se observă modul liniar şi nu exponenţial de variaţie a lui *cwnd* în momentul trecerii la valoarea iniţială, de start, impusă de *Slow Start*.

Graficul următor prezintă modul de variaţie al variabilei *ssthresh*, variabilă care marchează pragul de trecere de la *Slow Start* la *Congestion Avoidance*.

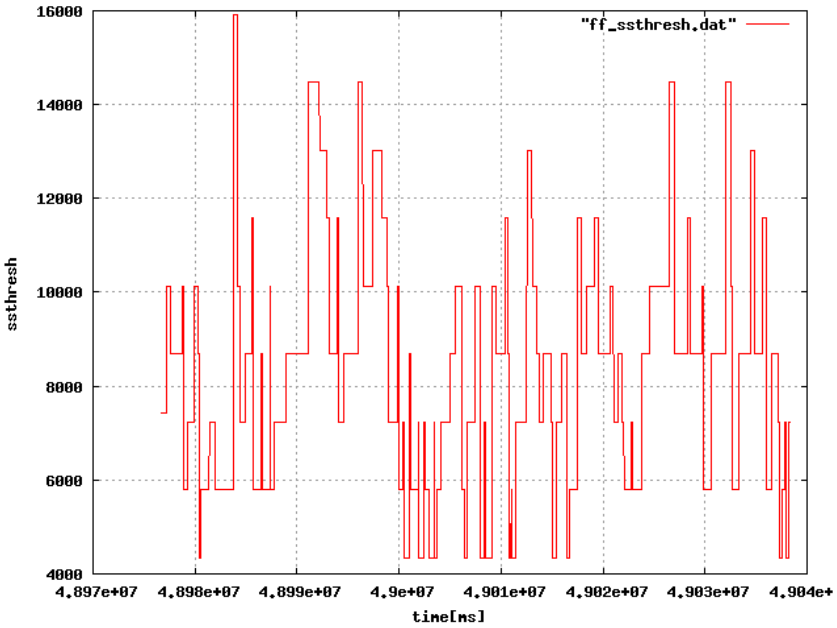


Figura 3-9 Evoluția variabilei *ssthresh* pentru cazul a)

Valorile cât mai mari şi menținerea constantă a acestora pentru o durată cât mai lungă de timp, reprezintă un indicator despre calitatea bună a unei conexiuni TCP, prin acesta înțelegând rate de transfer ridicate. Această variabilă se modifică de fiecare dată când se trece de la *Congestion Avoidance* la *Slow Start* sau *Fast Retransmit*, conform figurii 1-22.

b) Comunicație având semnal radio slab (< -70dBm) şi buffer de transmisie de dimensiune mare

În general graficele arată similar cu cele de la cazul anterior a), ceea ce diferă este numărul de retransmisii care se reflectă în valorile variabilei *seq*.

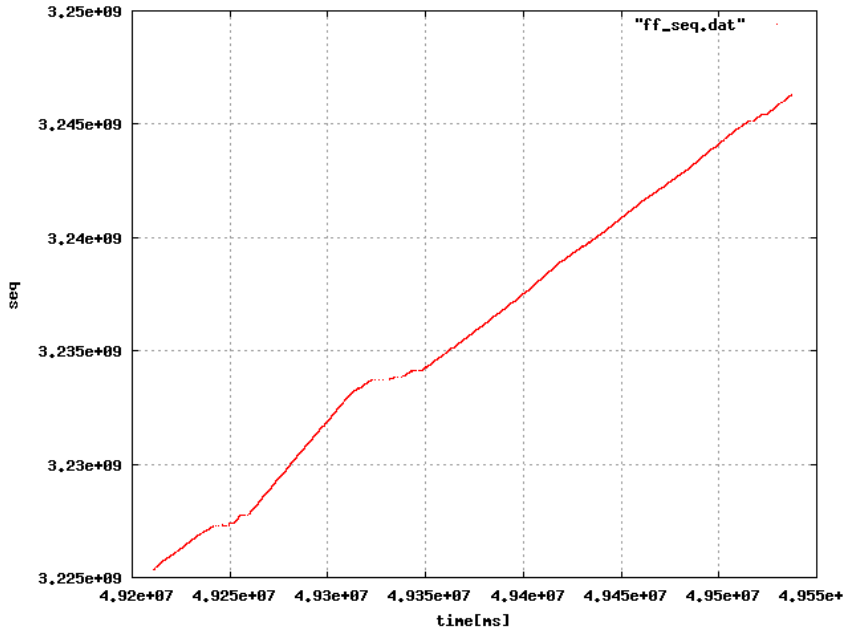


Figura 3-10 Evoluția variabilei *seq* pentru cazul b)

În figura 3-11 este înfățișat un detaliu, reprezentând porțiunea din graficul 3-10, unde are loc o diminuare a ratei de transfer. Această micşorare apare pe graficul 3-10, ca două porțiuni unde panta cu care evoluează *seq* este mai mică decât în restul graficului.

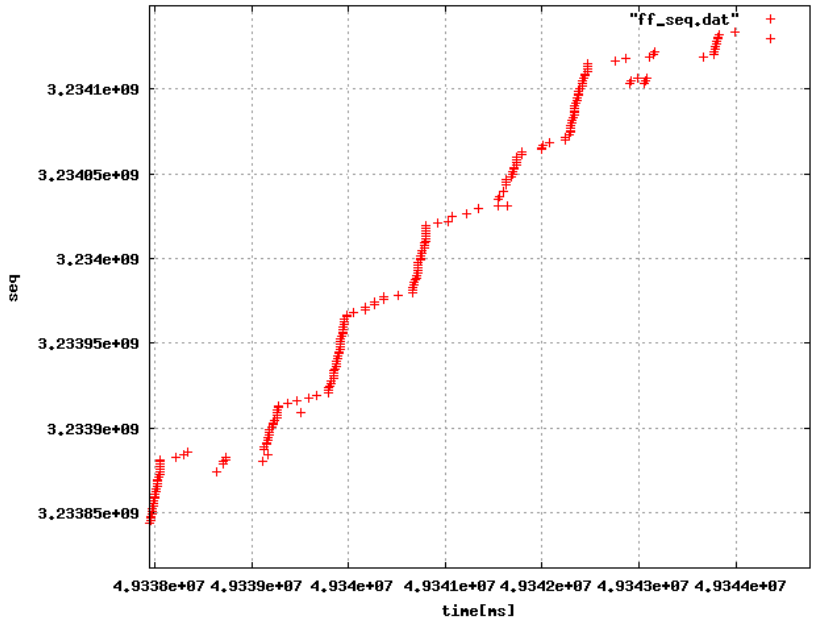


Figura 3-11 Detaliu privind evoluția variabilei seq pentru cazul b)

Privind în detaliu (figura 3-11) una din aceste zone se remarcă, de fapt, că zona este compusă din mai multe porțiuni succesive unde rata de transfer scade.

În cazul graficelor din figurile 3-12 şi 3-13, comparativ cu cele din figurile 3-6 şi 3-7 se remarcă un număr mai mic de reveniri ale variabilei *cwnd* la valoarea de start. Din nou a fost reprezentată variabila *cwnd* în două feluri (reprezentare prin puncte şi reprezentare prin linii) pentru a se observa mai clar modul ei de evoluție.

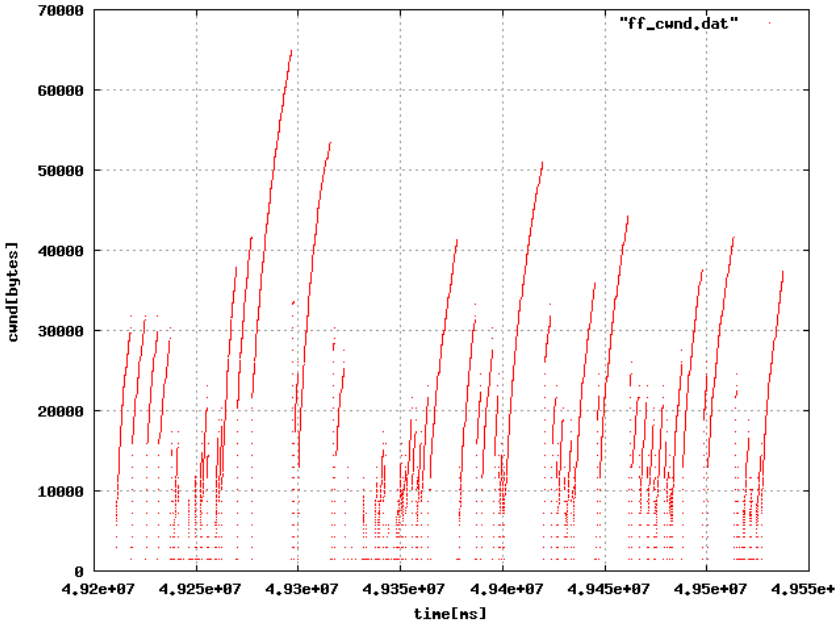


Figura 3-12 Evoluția variabilei *cwnd* pentru cazul b)

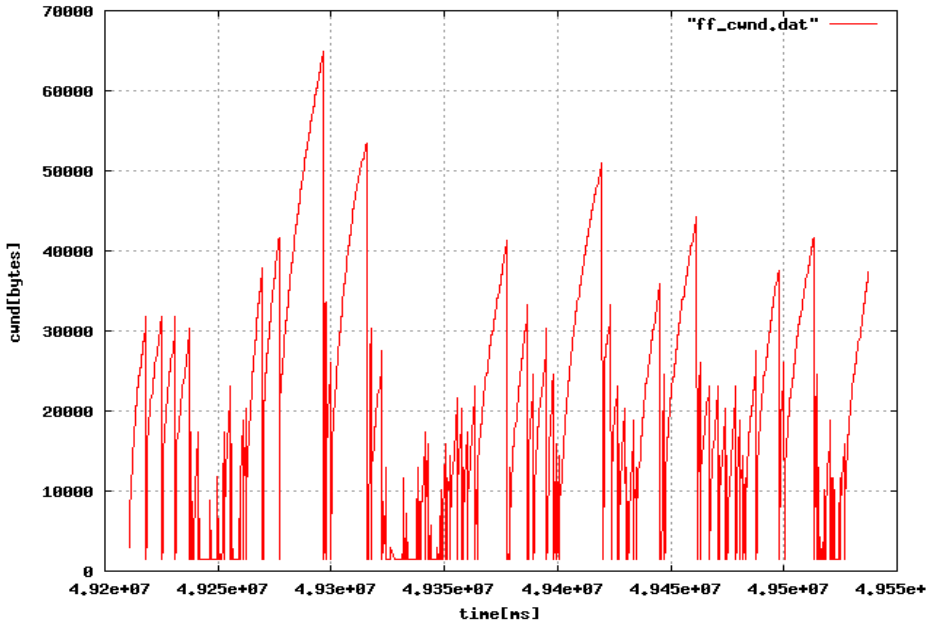


Figura 3-13 Evoluția variabilei *cwnd* pentru cazul b)

În cazul lui *ssthresh* se remarcă porţiunile de timp mai lungi unde variabila îşi păstrează valoarea constantă, semn că protocolul TCP a evoluat mai mult în zona *Congestion Avoidance*.

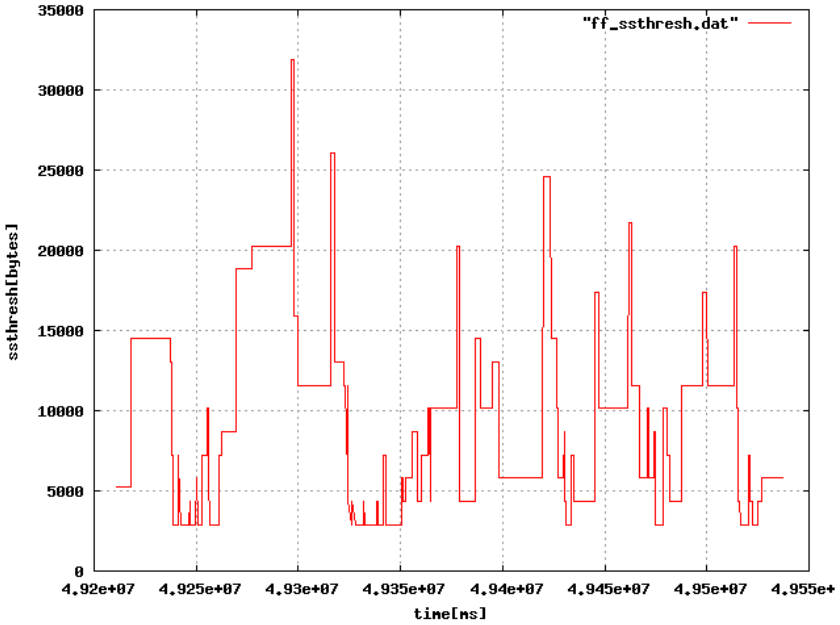


Figura 3-14 Evoluţia variabilei *ssthresh* pentru cazul b)

c) Comunicaţie având semnal radio puternic (> -50dBm) şi buffer de transmisie de dimensiune mare

Acesta este cazul când s-au obţinut cele mai bune rezultate privind calitatea conexiunii TCP. O sa discutăm doar evoluţia variabilelor *cwnd* şi *ssthresh*.

Se remarcă în figura 3-15 cum în prima jumătate a graficului, există o porţiune scurtă când variabila *cwnd* are reveniri dese la *Slow Start*, pentru ca apoi lucrurile să se îmbunătăţească şi acesta să reuşească să ajungă la valori mari, datorită rămânerii mai îndelungate a TCP-ului la *Congestion Avoidance*.

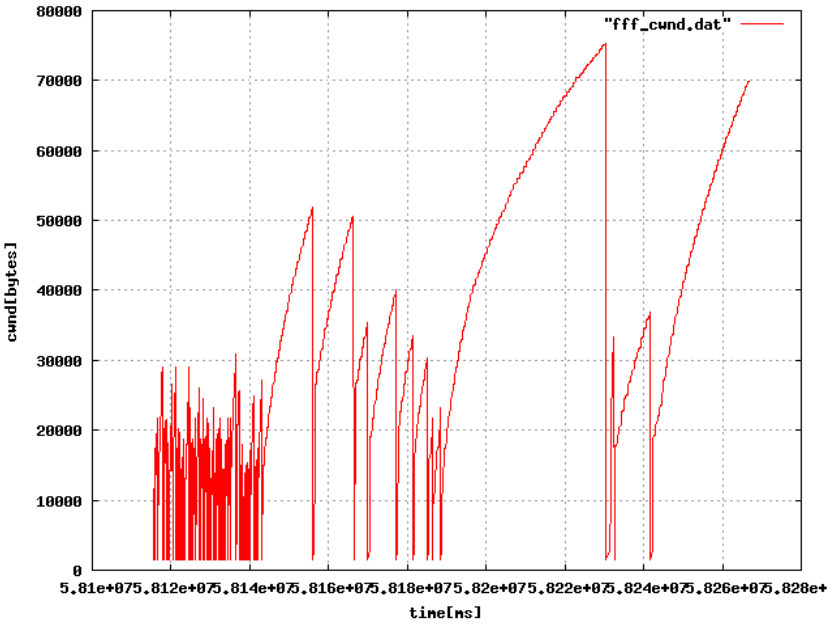


Figura 3-15 Evoluția variabilei *cwnd* pentru cazul c)

În figura 3-16 este detaliată prima porțiune a graficului din figura 3-15, pentru a se observa mai clar trecerile la *Slow Start*.

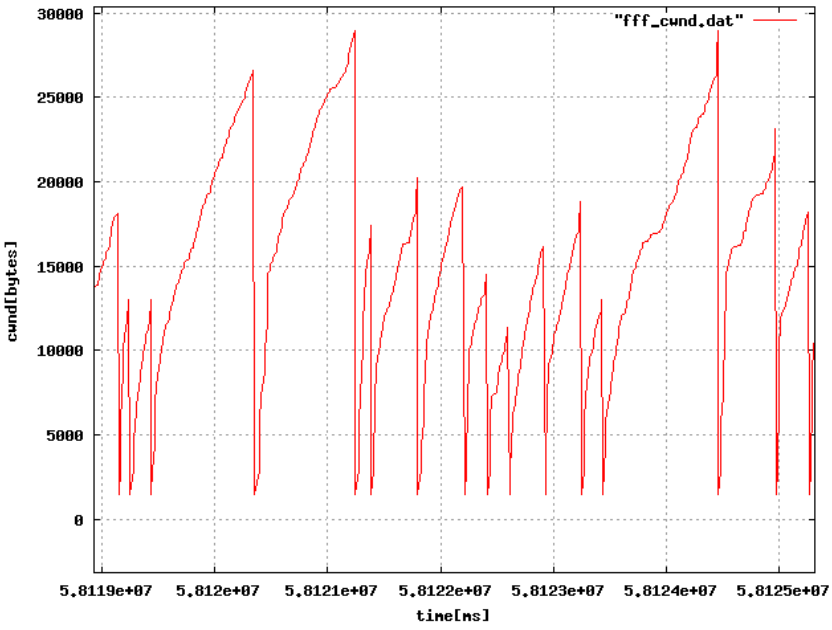


Figura 3-16 Detaliu privind evoluția variabilei *cwnd*

Foarte sugestiv este şi graficul din figura 3-17 care înfățişează variația lui *ssthresh*. Atâta timp cât *ssthresh* işi menține valoarea constantă, însemnă că ne aflăm în faza de *Congestion Avoidance*, ceea ce-i permite variabilei *cwnd* să aibă o evoluție ascendentă şi astfel rata de transfer să crească.

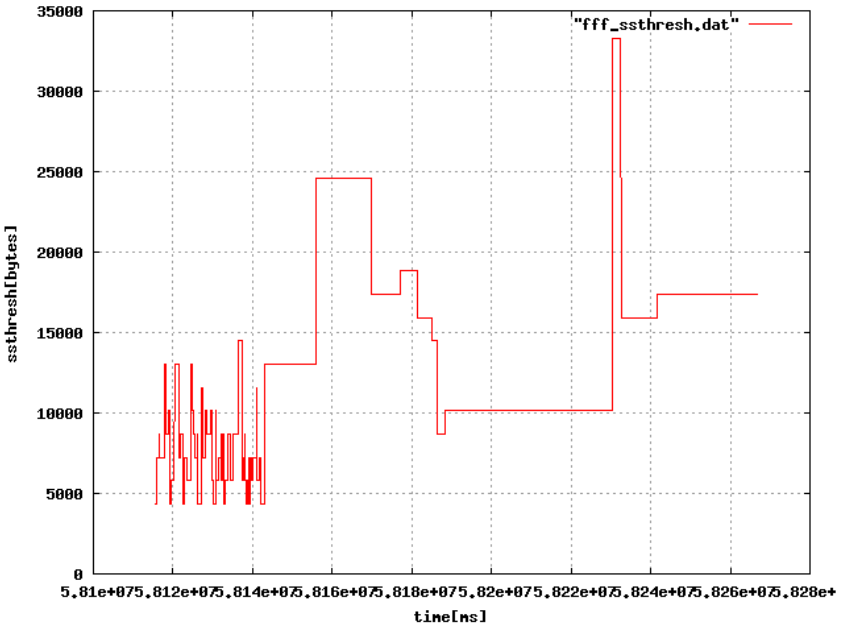


Figura 3-17 Evoluția variabilei *ssthresh* pentru cazul c)

Rezultatele testelor efectuate sunt sistematizate în tabelul din figura 3-18.

	seq (retransmisii)	*cwnd* (reveniri la *Slow Start*)	*ssthresh* (valoarea medie expr. în nr. de pachete)	Rata medie de transfer pt. date [Kb/sec]
a	438	149	4,3	1100
b	341	117	6,8	2300
c	104	33	13,6	5900

Figure 3-18 Rezultatele testelor pentru metoda 1

Pe coloana *seq* sunt trecute numărul total de retransmisii care au avut loc. Pe coloana *cwnd* sunt contorizate numărul de treceri la *Slow Start*. Pe coloana următoare s-a făcut o medie a valorilor luate de variabila *ssthresh*. Media s-a exprimat în număr de pachete şi nu în număr de octeți, iar pe ultima coloană s-a calculat rata medie de transfer a datelor utile. Se observă că, cu cât semnalul radio este mai slab cu atât calitatea conexiunii TCP scade, lucru care se remarcă prin valorile redate în tabel. Calitatea cea mai bună a unei conexiuni a fost obținută pentru cazul c), iar cea mai slabă calitate a fost înregistrată în cazul a).

Metoda elaborată oferă indicatori diverşi cu privire la calitatea unei conexiuni TCP, pentru că sunt monitorizate pe de o parte numerele de secvență şi numerele de confirmare, iar pe de altă parte variabilele *cwnd* şi *ssthresh*, preluându-se valorile lor direct din kernel. Având aceste informații, pot fi făcute reprezentări grafice diverse, care să o ofere o imagine intuitivă asupra evoluției diverşilor parametrii, dar pot fi făcute şi analize mai elaborate privind evoluția variabilelor, ca în tabelul din figura 3-18.

3.2.2 Analiza pachetelor unei conexiuni TCP

Următoarea metodă se bazează pe folosirea unui program de tip analizor de pachete numit şi sniffer, cu ajutorul căruia poate fi interceptat traficul realizat de către calculatorul pe care este instalat acest program. Pentru studiul de faţă a fost folosit Wireshark-ul, care este cel mai răspândit şi mai cunoscut program de acest gen. Metoda de lucru este de a realiza pe toată durata conexiunii TCP o captură folosind acest program. În figura 3-19 este înfăţişat un fragment dintr-o captură, unde este evidenţiat un exemplu de apariţie a congestiei precum şi situaţia unor pierderi multiple de pachete de date:

No.	Time	Source	Destination	Protocol	Info
41	3.230083	212.112.238.74	192.168.0.13	FTP-DATA	FTP Data: 1460 bytes
42	3.230232	212.112.238.74	192.168.0.13	FTP-DATA	FTP Data: 1460 bytes
43	3.230318	192.168.0.13	212.112.238.74	TCP	1194 > 53564 [ACK] Seq=1 Ack=11681 Win=61320 Len=0
44	3.230519	192.168.0.13	212.112.238.74	TCP	[TCP Window Update] 1194 > 53564 [ACK] Seq=1 Ack=11681 Win=6424(
45	3.237148	212.112.238.74	192.168.0.13	FTP-DATA	[TCP Previous segment lost] FTP Data: 1460 bytes
46	3.237260	192.168.0.13	212.112.238.74	TCP	[TCP Dup ACK 43#2] 1194 > 53564 [ACK] Seq=1 Ack=11681 Win=64240
47	3.237320	212.112.238.74	192.168.0.13	FTP-DATA	FTP Data: 1460 bytes
48	3.237381	192.168.0.13	212.112.238.74	TCP	[TCP Dup ACK 43#3] 1194 > 53564 [ACK] Seq=1 Ack=11681 Win=64240
49	3.373386	212.112.238.74	192.168.0.13	FTP-DATA	[TCP Retransmission] FTP Data: 1460 bytes
50	3.373522	192.168.0.13	212.112.238.74	TCP	[TCP Dup ACK 43#4] 1194 > 53564 [ACK] Seq=1 Ack=11681 Win=64240
51	3.506696	212.112.238.74	192.168.0.13	FTP-DATA	[TCP Previous segment lost] FTP Data: 1460 bytes
52	3.506828	192.168.0.13	212.112.238.74	TCP	[TCP Dup ACK 43#5] 1194 > 53564 [ACK] Seq=1 Ack=11681 Win=64240
53	3.641570	212.112.238.74	192.168.0.13	FTP-DATA	[TCP Retransmission] FTP Data: 1460 bytes
54	3.641764	192.168.0.13	212.112.238.74	TCP	1194 > 53564 [ACK] Seq=1 Ack=14601 Win=61320 Len=0 SLE=23361 SRE
55	3.641913	192.168.0.13	212.112.238.74	TCP	[TCP Window Update] 1194 > 53564 [ACK] Seq=1 Ack=14601 Win=6424(
56	4.148838	212.112.238.74	192.168.0.13	FTP-DATA	[TCP Retransmission] FTP Data: 1460 bytes
57	4.149022	192.168.0.13	212.112.238.74	TCP	1194 > 53564 [ACK] Seq=1 Ack=18981 Win=59860 Len=0 SLE=23361 SRE
58	4.149186	192.168.0.13	212.112.238.74	TCP	[TCP Window Update] 1194 > 53564 [ACK] Seq=1 Ack=18981 Win=6362(
59	4.282285	212.112.238.74	192.168.0.13	FTP-DATA	[TCP Retransmission] FTP Data: 1460 bytes
60	4.282550	192.168.0.13	212.112.238.74	TCP	1194 > 53564 [ACK] Seq=1 Ack=20441 Win=64240 Len=0 SLE=23361 SRE
61	4.417267	212.112.238.74	192.168.0.13	FTP-DATA	[TCP Retransmission] FTP Data: 1460 bytes
62	4.417397	192.168.0.13	212.112.238.74	TCP	[TCP Dup ACK 60#1] 1194 > 53564 [ACK] Seq=1 Ack=20441 Win=64240
63	4.417446	212.112.238.74	192.168.0.13	FTP-DATA	FTP Data: 1460 bytes
64	4.417506	192.168.0.13	212.112.238.74	TCP	[TCP Dup ACK 60#2] 1194 > 53564 [ACK] Seq=1 Ack=20441 Win=64240
65	4.552960	212.112.238.74	192.168.0.13	FTP-DATA	[TCP Previous segment lost] FTP Data: 1460 bytes

Figura 3-19 Captură realizată cu programul Wireshark

În liniile 41 şi 42 sunt transmise date care sunt confirmate în linia 43. În linia 44 are loc doar o actualizare a ferestrei receptorului. În linia 45 este recepţionat un pachet de date în afara numărului de secvenţă aşteptat. El determină transmiterea primului pachet de confirmare duplicat. În continuare sursa continuă să trimită pachete de date, fiecare din ele determinând emiterea a câte unui pachet de confirmare duplicat. Faptul că sunt pachete de confirmare duplicat rezultă din valoare câmpului *ACK*. În momentul când se recepţionează cel de-al treilea pachet duplicat se trece la retransmisia pachetului pierdut. Acest lucru se observă în linia 53. Aparent, retransmisia are loc după recepţia celui de-al patrulea pachet de tip *ACK* duplicat. Aceasta s-a întâmplat deoarece pachetele de date au fost trimise în rafală. Cum pentru fiecare pachet de date recepţionat după un pachet de date pierdut, se răspunde cu un pachet de tip ACK duplicat, determină ca trimiterea de pachete ACK să se facă tot în rafală. Chiar dacă, după recepţia celui de-al treilea pachet de tip ACK duplicat, TCP retransmite pachetul de date lipsă, deja cel de-al patrulea pachet ACK duplicat era deja generat şi a fost trimis înainte ca pachetul de date retransmis să ajungă la destinaţie.

Din fişierul de captură astfel obţinut se va genera prin filtrare o variantă care va conţine doar traficul TCP care ne interesează pe noi, deoarece captura va conţine absolut toate pachetele care ajung la interfaţa noastră de reţea. Apoi, fişierul se va exporta în format text ca în figura 3-20. Datele care ne interesează vor fi extrase cu ajutorul unor programe de tip parser de date.

```
"27","0.484468","172.16.254.102","172.16.254.251","TCP","[TCP Dup ACK 14#5] 3141 > 44074 [ACK] Seq=1 Ack=5793
"28","0.484514","172.16.254.251","172.16.254.102","TCP","44074 > 3141 [ACK] Seq=17377 Ack=1 Win=5840 Len=1448
"29","0.522256","172.16.254.102","172.16.254.251","TCP","[TCP Dup ACK 14#6] 3141 > 44074 [ACK] Seq=1 Ack=5793
"30","0.538286","172.16.254.102","172.16.254.251","TCP","[TCP Dup ACK 14#7] 3141 > 44074 [ACK] Seq=1 Ack=5793
"31","0.538324","172.16.254.251","172.16.254.102","TCP","44074 > 3141 [ACK] Seq=18825 Ack=1 Win=5840 Len=1448
"32","0.590546","172.16.254.102","172.16.254.251","TCP","3141 > 44074 [ACK] Seq=1 Ack=17377 Win=20272 Len=0 TS
"33","0.590595","172.16.254.251","172.16.254.102","TCP","44074 > 3141 [ACK] Seq=20273 Ack=1 Win=5840 Len=1448
"34","0.620029","172.16.254.102","172.16.254.251","TCP","3141 > 44074 [ACK] Seq=1 Ack=18825 Win=23168 Len=0 TS
"35","0.633647","172.16.254.102","172.16.254.251","TCP","3141 > 44074 [ACK] Seq=1 Ack=20273 Win=26064 Len=0 TS
"36","0.665797","172.16.254.102","172.16.254.251","TCP","3141 > 44074 [ACK] Seq=1 Ack=21721 Win=28960 Len=0 TS
"37","0.717052","172.16.254.251","172.16.254.102","TCP","44074 > 3141 [PSH, ACK] Seq=21721 Ack=1 Win=5840 Len=
"38","0.737207","172.16.254.251","172.16.254.102","TCP","44074 > 3141 [ACK] Seq=23169 Ack=1 Win=5840 Len=1448
"39","0.739043","172.16.254.102","172.16.254.251","TCP","3141 > 44074 [ACK] Seq=1 Ack=23169 Win=31856 Len=0 TS
"40","0.745730","172.16.254.251","172.16.254.102","TCP","44074 > 3141 [ACK] Seq=24617 Ack=1 Win=5840 Len=1448
"41","0.755401","172.16.254.251","172.16.254.102","TCP","44074 > 3141 [ACK] Seq=26065 Ack=1 Win=5840 Len=1448
"42","0.764991","172.16.254.251","172.16.254.102","TCP","44074 > 3141 [ACK] Seq=27513 Ack=1 Win=5840 Len=1448
"43","0.784285","172.16.254.102","172.16.254.251","TCP","3141 > 44074 [ACK] Seq=1 Ack=24617 Win=34752 Len=0 TS
"44","0.784342","172.16.254.251","172.16.254.102","TCP","44074 > 3141 [ACK] Seq=28961 Ack=1 Win=5840 Len=1448
"45","0.795304","172.16.254.251","172.16.254.102","TCP","44074 > 3141 [ACK] Seq=30409 Ack=1 Win=5840 Len=1448
"46","0.825258","172.16.254.102","172.16.254.251","TCP","3141 > 44074 [ACK] Seq=1 Ack=26065 Win=37648 Len=0 TS
"47","0.825313","172.16.254.251","172.16.254.102","TCP","44074 > 3141 [ACK] Seq=31857 Ack=1 Win=5840 Len=1448
"48","0.852472","172.16.254.102","172.16.254.251","TCP","3141 > 44074 [ACK] Seq=1 Ack=27513 Win=40544 Len=0 TS
"49","0.852518","172.16.254.251","172.16.254.102","TCP","44074 > 3141 [PSH, ACK] Seq=33305 Ack=1 Win=5840 Len=
"50","0.890694","172.16.254.102","172.16.254.251","TCP","3141 > 44074 [ACK] Seq=1 Ack=28961 Win=43440 Len=0 TS
"51","0.890737","172.16.254.251","172.16.254.102","TCP","44074 > 3141 [ACK] Seq=34753 Ack=1 Win=5840 Len=1448
"52","0.931382","172.16.254.102","172.16.254.251","TCP","3141 > 44074 [ACK] Seq=1 Ack=30409 Win=46336 Len=0 TS
```

Figura 3-20 Fişier text generat pe baza unei capturi făcute cu Wireshark

Din acest fişier vor fi contorizate numărul retransmisiilor. Se vor considera retransmisii acele pachete de date care au numărul de secvenţă identic cu al unui pachet de date anterior trimis. Numărul total al retransmisiilor cuprinde atât situaţia de timeout când se va trece la *Slow Start*, cât şi cazul recepţiei consecutive a trei pachete de tip *ACK* duplicat, când se trece la *Fast Retransmit*. Dacă se doreşte decelarea numărului total de retransmisii şi obţinerea separată a numărului de treceri la *Slow Start* şi *Fast Retransmit*, se poate proceda în două feluri.

a) Se identifică momentele de timeout.

Atunci când se identifică un pachet de date retransmis se compara valoare de pe coloana unde se înregistrează timpul cu cea a ultimei transmisii a aceluiaşi pachet de date. Dacă diferenţa este mai mare de 200ms se trage concluzia că a avut loc un timeout. Valoare de 200ms este dependentă de platforma pe care este implementată stiva TCP.

b) Se identifică cele trei pachete de tip *ACK* duplicat

Atunci când se identifică un pachet de date retransmis se urmăreşte dacă înaintea lui au fost recepţionate trei pachete de tip *ACK* consecutive, care au valoarea din câmpul *ACK* identică cu cea a pachetului retransmis.

Această metodă este uşor de implementat pe orice tip de platformă şi cu ajutorul ei poate fi făcută o evaluare rapidă şi destul de elocventă în ceea ce priveşte calitatea unei conexiuni TCP. Dacă nu este nevoie să se facă analize rafinate, atunci ea este o metoda care poate fi folosită.

În urma măsurătorilor efectuate tot pentru cele trei cazuri luate în calcul la metoda anterioară, s-a obţinut rezultatele prezentate în figura 3-21 . Capturile de pachete au fost efectuate în paralel cu măsurătorile realizate la metoda 1 şi se observă că rezultatele reflectă aceiaşi tendinţă înregistrată şi cu prima metodă, pentru toate cele trei cazuri a), b) şi c).

	Retransmisii generate de time-out	*Retransmisii generate de „duplicate ACKs”*
a	135	274
b	128	211
c	47	83

Figura 3-21 Rezultatele testelor pentru metoda 2

Retransmisiile generate de time-aut, reprezintă treceri la *Slow Start,* iar retransmisiile generate de pachetele de tip ACK duplicat, marchează treceri la *Fast Retransmit*, fie din faza de *Congestion Avoidance*, fie chiar din faza de *Slow Start*.

3.3 Concluzii

Cu ajutorul celor două metode de analiză a comportamentului protocolului TCP, pe care le-am elaborat a putut fi pus în evidenţă comportamentul protocolului TCP rulând peste o reţea wireless 802.11. Cu ajutorul datelor culese s-au putut trasa grafice şi s-a putut face o analiză calitativă privind evoluţia în timp a acestui protocol. Se poate trage concluzia, care era de aşteptat şi care este în concordanţă cu alte studii prezente în bibliografie, că protocolul TCP, în anumite circumstanţe, nu are un comportament eficient atunci când rulează pe echipamente conectate la reţea prin legături wireless. Soluţia pe care o vom prezenta în capitolul 4 încearcă să elimine aceste „circumstanţe nefavorabile” şi propune o rezolvare care nu necesită intervenţia la nivelul stivei de protocoale, în schimb presupune existenţa unui sistem de asistenţă care să monitorizeze în permanenţă calitatea unei conexiuni şi să intervină, în scopul de a menţine legătura în parametrii cât mai buni, pentru a diminua cât se poate de mult apariţia erorilor.

A doua metodă de investigaţie, prin analiza pachetelor, oferă o imagine mai puţin exactă asupra nivelului de degradare a unei conexiunii TCP, deoarece obţinem doar momentele şi numărul trecerilor la *Slow Start* sau *Fast Retransmit*, fără a avea o idee despre ce valori au avut variabilele *cwnd* şi *ssthresh*. Pentru a face o analiză mai rafinată este util de ştiut şi care au fost valorile medii pentru variabilele *cwnd* şi *ssthresh*.

Metoda de analiză folosind citirea variabilelor din kernel oferă o imagine mai complexă asupra calităţii unei conexiuni TCP dar este ceva mai dificil de implementat, pe când a doua metoda care foloseşte un analizor de pachete, poate fi rulată pe orice sistem, existând variante de Wireshark pentru orice tip de platformă.

Aceste metode au fost aplicate în acest caz pentru conexiuni realizate folosind o reţea 802.11, dar ele au caracter general, putând fi folosite pentru orice alt tip de reţea.

Capitolul 4 Optimizarea procesului de reasociere a unui staţii într-o reţea WLAN 802.11

4.1 Introducere

În capitolul 3 s-au elaborat două metode de analiză a comportamentului protocolului TCP într-o reţea wireless 802.11. Scopul acestor metode, în lucrarea de faţă, este acela de a ne oferi posibilitatea de a face o analiză calitativă asupra unei soluţii de îmbunătăţire a comportamentului TCP-ului într-o reţea wireless.

Soluţia propusă nu presupune intervenţia la nivelul stivei de protocoale, deci ea nu va necesita modificarea nici unuia dintre protocoalele direct răspunzătoare de calitatea unei conexiuni. Protocoalele la care facem referire sunt TCP-ul şi implementarea de pe subnivelul MAC al lui 802.11. Ideea care stă la baza soluţiei propuse este de a încerca menţinerea unei conexiuni wireless la un nivel la care apariţia erorilor să fie cât mai redusă.

Degradarea calităţii unei conexiuni wireless apare în două situaţii:

- staţia se află la distanţă prea mare de AP-ul la care este asociată
- în interiorul aceluiaşi BSS sunt prea mulţi clienţi

Soluţia propusă este parte integrantă din UFRM (Unified Framework for Resource Management), care va fi descris în detaliu în capitolul 6. Deci mecanismul implementat presupune existenţa unor aplicaţii software care să ruleze pe dispozitivul pentru care se doreşte menţinerea în parametrii optimi a unei conexiuni TCP.

Când un client wireless (STA) funcţionează într-o configuraţie de tip *infrastructure* el va fi asociat la un anumit AP. Acel AP defineşte un *basic service set* (BSS). Tot traficul care pleacă sau vine către STA se va realiza prin AP-ul la care STA este asociat. Dacă se doreşte acoperirea unei zone mai mari, atunci mai multe AP-uri sunt conectate printr-un *distribution system* (DS) formând astfel un *extended service set* (ESS). Când un client se deplasează şi pierde conectivitatea cu AP-ul său, atunci el va fi nevoit să se reasocieze la un alt AP din acelaşi ESS. În literatura de specialitate acest proces poartă mai multe denumiri: ***roaming***, ***handover*** sau ***handoff***. Termenul de roaming încetăţenit în cazul reţelelor GSM şi prin el se înţelege mecanismul prin care un abonat poate beneficia de serviciile oferite de un alt operator, atunci când părăseşte aria de acoperire a operatorului la care el este abonat. În cazul nostru, roaming-ul reprezintă procesul prin care un client 802.11 migrează dintr-un BSS în altul. Pentru a nu exista confuzii, va fi folosit de acum în acolo termenul de handover.

Folosindu-ne de acest mecanism pus la dispoziţie de standardul 802.11 vom propune o metodă care vizează păstrarea în parametrii optimi a unei conexiuni wireless pentru un dispozitiv mobil aflat sau nu în mişcare. Metoda fiind integrată în UFRM, va fi posibilă iniţierea un proces de handover chiar dacă dispozitivul nu se află în mişcare, dar se determină pe baza unor criterii, că pentru a menţine conexiunea în parametrii optimi este mai bine să se facă asocierea cu un alt AP. Criteriile care ar putea fi luate în calcul pentru a se lua decizia de handover vor fii discutaţe în detaliu în cadrul capitolului 6. Criteriul principal este puterea semnalului recepţionat de la

AP-urile din zonă, dar poate fi luată în considerare şi încărcarea cu clienţi a AP-urilor. Acest gen de informaţie nu poate fi obţinut direct, el fiind disponibil doar prin intermediul UFRM. Pentru soluţia implementată şi testată în acest capitol s-a folosit ca şi criteriu pentru luarea deciziilor, doar puterea semnalului recepţionat (RSSI – Received Signal Strength Indicator).

Procedura de reasociere este de obicei realizată în mod automat de către driver-ul care controlează funcţionarea interfeţei wireless. Atunci când conectivitatea cu AP-ul curent se pierde, începe procesul de scanare prin care sunt căutate AP-uri cu acelaşi SSID ca cel al vechiului AP, dar care să ofere o calitate mai bună a semnalului. Din păcate acest mod de funcţionare al mecanismului de reasociere va conduce inevitabil la întreruperea temporară a unei eventuale conexiuni aflate în desfăşurare.

Soluţia propusă urmăreşte tocmai acest aspect al întreruperii temporare a unei conexiuni şi încearcă să reducă cât mai mult cu putinţă durata procesului de handover. Motivul pentru care a fost ales agest gen de optimizare este acela de a servi ca mecanism de bază în implementarea framework-ului propus în capitolul 6 al acestei lucrări.

Handover-ul poate fi realizat în două moduri [RL03]:

- Un handover care presupune continuitatea conexiunii şi care evită degradarea excesivă a semnalului, făcând ca resursele reţelei să fie disponibile fără întreruperi sesizabile. Acest tip de handover este caracteristic reţelelor GSM.

- Un handover care nu necesită continuitate în ceea ce priveşte accesul la resursele reţelei. Acesta tip de handover este întâlnit în cazul reţelelor 802.11. Să presupunem că un utilizator aflat într-o anumită zonă părăseşte acea zonă care deservită de un AP şi se îndreaptă spre o altă zonă, deservită de un alt AP. Pe perioada deplasării, el nu are nevoie de conexiune la reţea, iar în momentul în care îşi reia lucrul în noua locaţie, se realizează efectiv procesul de handover, când dispozitivul mobil pe care îl foloseşte sesizează că vechiul AP nu mai este disponibil şi recurge la căutarea unui nou AP la care se conectează. În cazul acestui tip de handover, conexiunea este temporar întreruptă, lucru care în unele cazuri poate să nu deranjeze, dar pentru un anumit gen de aplicaţii acest aspect poate să fie critic.

Conform specificaţiilor din standard şi a implementărilor existente, reţelele 802.11 nu sunt capabile să realizeze un handover fără întreruperea conexiunii, după cum se observă din figura 4-24. La început staţia se asociază la AP1, prin perechea de mesaje *Association Request* şi *Association Response*. La un moment dat conexiunea se întrerupe, moment în care driver-ul plăcii de reţea începe o scanare pentru a identifica AP-urile disponibile din aria sa. În acest caz presupunem că singurul AP disponibil este AP2, urmând să se realizeze procesul de reasociere cu acesta. La un moment dat se întrerupe şi conexiunea cu AP2. Urmează acelaşi proces de scanare şi o noua reasociere cu AP3, singurul AP disponibil la momentul întreruperii legăturii cu AP2.

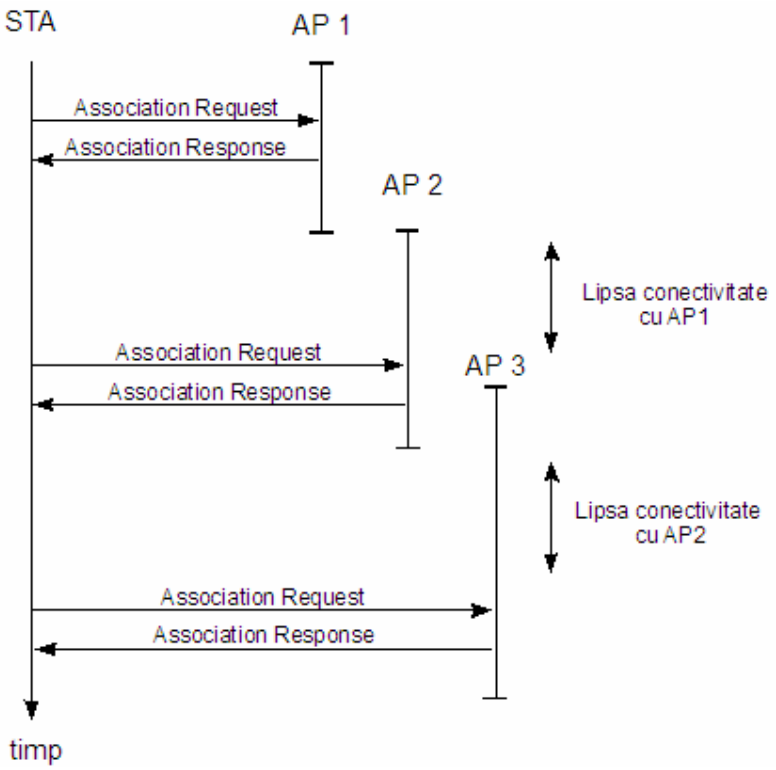


Figura 4-1 Procedura de handover

Chiar dacă un handover fără întreruperea conexiunii nu este caracteristic reţelelor 802.11, este posibilă în schimb o ameliorare a procesului de handover. Să analiză din nou figura 4-24. În momentul pierderii conexiunii cu AP2, există deja semnal suficient de puternic provenit de la AP3. Dacă nu s-ar aştepta până la întreruperea totală a conexiunii cu AP2, ci în momentul în care semnalul începe să se degradeze, s-ar putea realiza reasocierea la AP3. Din păcate, pentru a descoperi noul access point disponibil, trebuie să se efectueze acea scanare, care inevitabil impune întreruperea temporară a conexiunii, deoarece standardul nu permite o scanare în timp ce o conexiune se află în desfăşurare. Un pas înainte spre o reasociere mai rapidă ar fi acela dacă s-ar cunoaşte deja lista AP-urilor disponibile şi să se poată alege direct unul din acea listă. Problema care apare este cum se poate obţine acea listă şi să existe garanţia că ea este actualizată. Întocmirea acestei liste este unul din mecanismele implementate de metoda descrisă în acest capitol. Evident că aceste îmbunătăţiri îşi vor avea preţul.

Pe durata procesului de handover şi în funcţie de natura lui (cu sau fără întreruperea conexiunii) trebuie analizat în ce măsură este influenţată buna funcţionare a protocolul TCP. Acest lucru va fi făcut cu ajutorul metodelor de investigaţie descrise în capitolul 4. Chiar dacă evaluarea se va face doar pentru protocolul TCP, este evident că îmbunătăţirile obţinute se vor resimţi şi asupra protocolului de transport pereche şi anume UDP-ul.

Înainte de a trece la descrierea propriu-zisă a metodei propuse, se vor prezenta cele mai interesante soluţii care au fost identificate în literatura de specialitate şi care vizează procedeul aceste de handover.

Într-o reţea 802.11 este inevitabil faptul că unii clienţi suportă rate de transfer mai mari decât alţii. Asociind aceşti clienţi la acelaşi AP poate să ducă la o degradare

a conexiunii pentru utilizatori care suportă rate mai mari de transfer, deoarece subnivelul MAC asigură şanse egale tuturor clienţilor de a obţine accesul la canalul de comunicaţie şi astfel pentru a transmite aceiaşi cantitate de date, utilizatorii mai lenţi vor ţine ocupat mai mult timp mediul de transmisie. În [FS09] este studiat impactul pe care îl are asupra performanţelor clienţilor unui AP atunci când este permisă asocierea la acel AP unui client care suportă rate de transfer mai scăzute. Mai mult, autorii propun chiar soluţia de a impune clientului „lent" să realizeze procesul de handover către un alt AP.

În [LLG08] se face delimitarea între aşa numitul *horizontal handoof (HHO)* şi *vertical handoff (VHO)*. VHO are o importanţă majoră atunci când procedura de handoff se realizează între reţele wireless eterogene unde tehnologia de access la mediul fizic poate să difere. VHO este responsabil de păstrarea conexiunii la nivelul protocoalelor de pe nivele superioare. Chiar dacă articolul vizează reţele wireless eterogene, modelul analitic dezvoltat poate oferi soluţii şi în cazul reţelelor 802.11 în sensul de a implica nu doar subnivelul MAC în procesul de handoff ci şi protocoalele de pe nivele superioare. Un studiu asemănător din perspectiva reţelelor wireless eterogene este oferit şi în [BCI07]. Aici este propus un mecanism de tip *cross-layer* pentru implica protocoalele de pe mai multe nivele în procesul de handover.

Autorii din [RBP07] demonstrează că în anumite circumstanţe când reţeaua wireless este supraîncărcată, apar procese de tip handoff, chiar dacă se observă că nu există mobilitate din partea clienţilor care generează procese handover. Studiul a fost făcut prin analiza unui reţele formată din 55 de AP-uri şi peste 1200 de clienţi, aceştia fiind de fapt participanţii la o conferinţă. Autorii susţin că acest fenomen poate apare în orice reţea 802.11 menită să gestioneze un număr mare de conexiuni simultane când pierderea repetată a unor pachete de date poate să declanşeze procedura de handoff.

Pentru a obţine un maxim de performanţă pentru toţi clienţii unei reţele wireless este necesar să existe un mecanism de distribuire uniformă a traficului global realizat în reţea, între AP-urile care deservesc reţeaua. Rezolvare acestei probleme a fost studiată în [BH06], dar în momentul de faţă nu există soluţii standardizate pentru rezolvarea ei.

Pentru a obţine un handover fără întreruperi ale conexiunii aflate în desfăşurare, autorii din [ADH06] propun un protocol numite SMesh, implementat peste o reţea mesh formată din AP-uri 802.11 cu firmware-ul modificat, pentru a suporta noul protocol. AP-urile monitorizează în permanenţă calitatea conexiunilor avute cu clienţii lor şi mai multe, îşi distribuie aceste informaţii între ele pentru a stabili care dintre AP-uri se află în vecinătatea clientului care trebuie deservit. Tot un protocol pentru îmbunătăţirea procesului de handoff este propus şi în [WB05]. Aici decizia de migrare a unei staţii de la un AP la altul este luată de către AP-uri pe baza unui protocol de comunicaţie între AP-uri.

În [VV03] este realizat un extins studiu experimental asupra efectelor pe care procesul de handover îl are asupra fluxului de date schimbat între STA şi AP, din punct de vedere al întârzierilor şi pierderilor introduse, oferind o înţelegere mai clara asupra acestor mecanisme. Tot un studiu experimental este realizat şi în [MSA03], unde sunt analizate în mod detaliat etapele parcurse pentru realizarea procesului de handover.

4.2 Descrierea metodei propuse

După cum se ştie, interfaţa dintre placa de reţea wireless şi sistemul de operare este asigurată de driver-ul asociat acestei plăci. În funcţie de sistemul de operare folosit există diverse variante de implementare, dar indiferent de platforma pe care placa de reţea wireless funcţionează, driver-ul trebuie să respecte specificaţiile oficiale prezente în documentele care descriu standardul IEEE 802.11 [Std1, Std2].

Pentru exemplificare a fost aleasă soluţia adoptată de sistemul de operare Windows. Aici, pentru a oferi un mod uniform de acces la diferite tipuri de interfeţe de reţea este oferit un nivel intermediar de abstractizare între driver-ul nativ al plăcii de reţea şi sistemul de operare. Acest nivel intermediar poată denumirea de **NDIS** (**Network Driver Interface Specification)** şi este de fapt un API (Application Programming Interface) pentru plăcile de reţea. NDIS a început ca un proiect dezvoltat la de Microsoft împreună cu 3Com Corporation. Este folosit cu precădere pe platformele Windows, dar există şi variante open-source rulând pe sisteme Linux, FreeBSD sau NetBSD.

Raportat la modelul de referinţă OSI, NDIS rulează pe nivelul 2, formând de fapt subnivelul LLC (Logical Link Control), care rulează imediat peste subnivelul MAC.

Pentru fiecare tip de interfaţă de reţea există un set de operaţii care pot fi efectuate. Aceste operaţii sunt specificate prin aşa numiţii identificatori de obiecte NDIS (OIDs - Object Identifiers). Prin aceşti identificatori se specifică tipul de operaţie care este cerută driver-ului plăcii de reţea.

În cazul interfeţei wireless, lista completă a acestor identificatori este redată în tabelul din figura 4-2.

OID_802_11_BSSID
OID_802_11_SSID
OID_802_11_NETWORK_TYPES_SUPPORTED
OID_802_11_NETWORK_TYPE_IN_USE
OID_802_11_TX_POWER_LEVEL
OID_802_11_RSSI
OID_802_11_RSSI_TRIGGER
OID_802_11_INFRASTRUCTURE_MODE
OID_802_11_FRAGMENTATION_THRESHOLD
OID_802_11_RTS_THRESHOLD
OID_802_11_NUMBER_OF_ANTENNAS
OID_802_11_RX_ANTENNA_SELECTED
OID_802_11_TX_ANTENNA_SELECTED
OID_802_11_SUPPORTED_RATES
OID_802_11_DESIRED_RATES
OID_802_11_CONFIGURATION
OID_802_11_STATISTICS
OID_802_11_DISASSOCIATE
OID_802_11_POWER_MODE
OID_802_11_BSSID_LIST_SCAN
OID_802_11_BSSID_LIST
OID_802_11_PRIVACY_FILTER
OID_802_11_RELOAD_DEFAULTS
OID_802_11_AUTHENTICATION_MODE
OID_802_11_ENCRYPTION_STATUS
OID_802_11_ADD_WEP
OID_802_11_REMOVE_WEP
OID_802_11_ADD_KEY
OID_802_11_REMOVE_KEY
OID_802_11_ASSOCIATION_INFORMATION
OID_802_11_TEST
OID_802_11_CAPABILITY
OID_802_11_PMKID
OID_802_11_MEDIA_STREAM_MODE

Figura 4-2 OIDs – Object Identifiers

Să urmărim care sunt etapele parcurse atunci când un client încearcă să se asocieze la un access point. AP-ul la care se doreşte asocierea va fi identificat prin doi parametrii. Unul poartă denumirea de SSID (Service Set Identity), iar celălalt se numeşte BSSID (Basic Service Set ID). În cazul AP-urilor BSSID reprezintă adresa MAC a interfeţei de reţea wireless din interiorul AP-ului. SSID reprezintă un nume (şir de caractere) care poate fi asociat unei reţele wireless configurată într-o arhitectură de tip ESS. După cum s-a văzut în capitolul 2, un ESS este alcătuit din mai multe BSS-uri, fiecare BSS având la bază un access point. Deci fiecare AP din interiorul unui ESS va avea asociat un BSSID unic, dar în schimb toate vor fi setate cu un SSID comun.

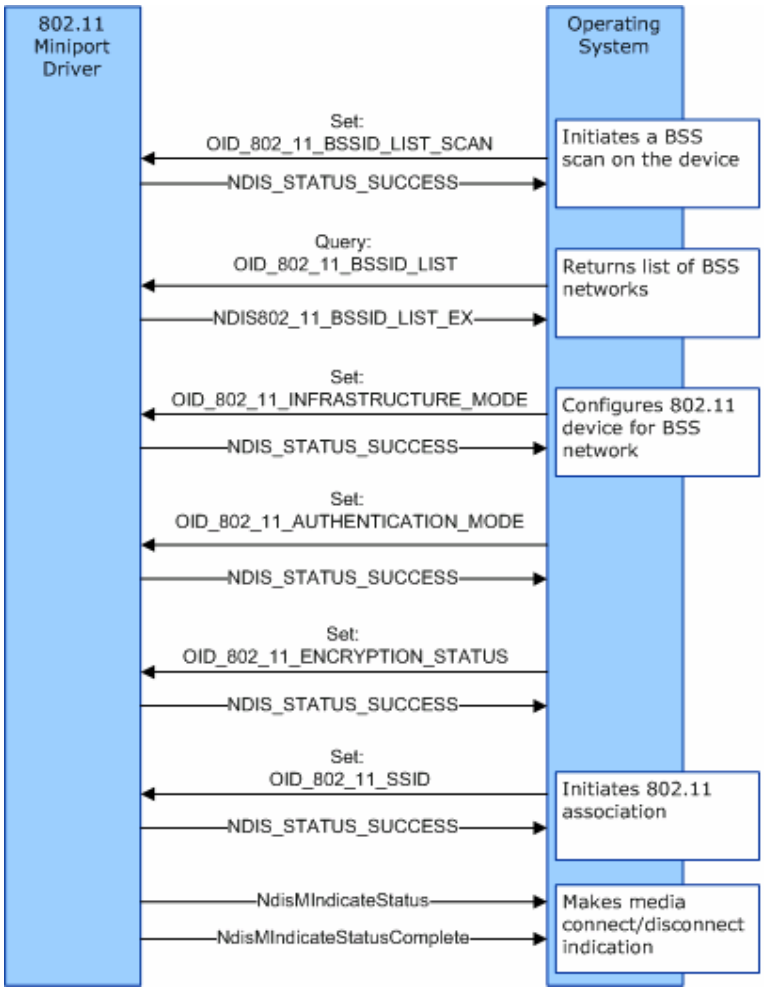


Figura 4-3 Etapele parcurse de driver-ul plăcii de reţea pentru a realiza asocierea la un AP

- În prima fază are loc o scanare pentru identificare AP-urilor valide. Fiecare AP creează în jurul său propriul BSS (Basic Service Set). Atunci când vom folosi acronimul BSS, de fapt este vorba de access point-ul care îl defineşte.
- Se stabileşte apoi modul de operare al clientului (*infrstructure* sau *ad-hoc*). În cazul nostru este vorba de o configuraţie de tip *infrastrucute*, deoarece clientul urmează să se asocieze la un AP, căruia i se va subordona.
- Următoarea etapă constă în stabilirea modului de autentificare şi de criptare a datelor care vor fi vehiculate între client si AP.
- Se setează apoi numele reţelei (SSID) la care se doreşte asocierea. Din lista de AP-uri obţinute la început, folosind un anumit criteriu (de exemplu AP-ul cu semnalul cel mai puternic) se alege acel AP care are setată aceiaşi valoare pentru SSID şi care satisface criteriul definit.

Dacă se doreşte reasocierea la un alt AP este suficient să se modifice BSSID folosind OID_802_11_BSSID. Chiar dacă BSSID nu se află în lista obţinută în faza de scanare, asocierea va fi permisă atâta timp câte există potrivire în ceea ce priveşte SSID-ul. Pe acest mecanism se bazează şi metoda propusă, adică se vor realiza reasocieri cu AP-uri care nu există în lista menţinută de driver-ul plăcii de reţea.

Principiul soluției [FSM09] propuse este cât se poate de simplu şi el necesită existența încă a unei plăci de rețea wireless în sistemul pentru care se dorește optimizarea. Aceasta a doua placă va fi folosită doar pentru monitorizarea AP-urilor din zona unde activează dispozitivul nostru. Ca soluție tehnică pentru dotarea sistemului cu încă o interfață de rețea wireless s-a ales varianta unei plăci pe portul USB.

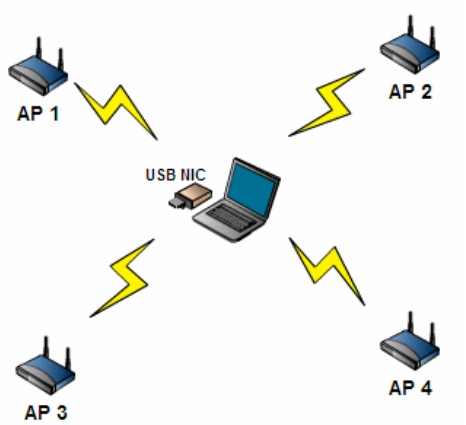


Figura 4-4 Laptop dotat cu două plăci de rețea wireless

Aplicația de test a fost făcută să scaneze de cinci ori pe secundă pentru a descoperi eventualele modificări ale puterii semnalului provenit de la cele patru AP-uri. În momentul în care diferența de putere dintre semnalul provenit de la AP-ul curent şi semnalul de la unul dintre celelalte AP-uri devine mai mare de 20dBm se realizează reasocierea cu noul AP. Pentru evaluarea eficienței metodei în a menține o conexiune TCP în parametrii optimi, s-a folosit pentru test metoda citirii directe din kernel a variabilelor utilizate de algoritmii de control ai congestiei. Pentru că metoda presupune ca pe calculatorul gazdă să ruleze sistemul de operare FreeBSD, s-a rulat acesta ca şi maşină virtuală folosind programul VMware, sistemul de operare nativ de pe laptop fiind WindowsXP.

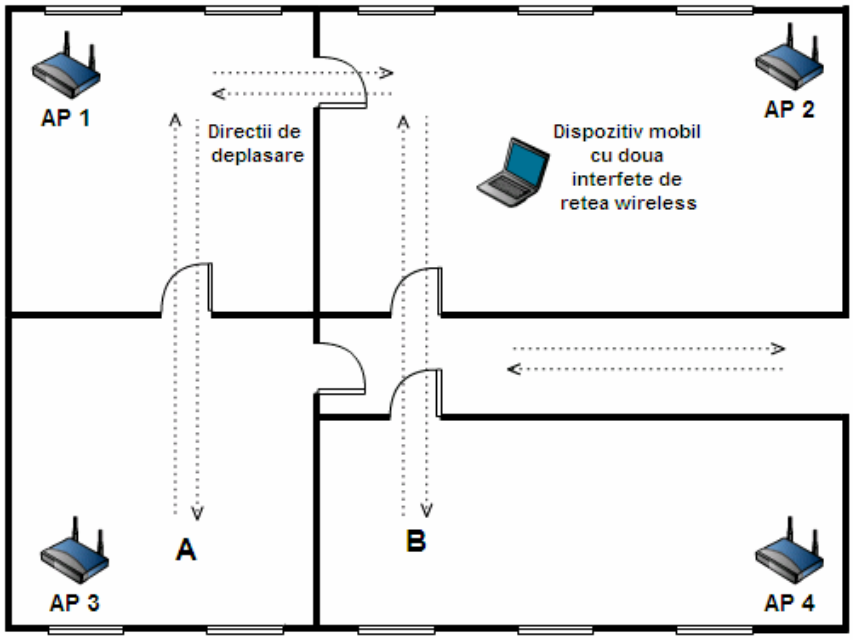


Figura 4-5 Schema amplasării dispozitivelor wireless implicate în efectuarea testelor

4.3 Rezultatele testelor

În cadrul testului s-au folosit patru AP-uri dispuse ca în figura 4-5. S-a iniţiat o conexiune pentru a realiza un transfer continuu de date grupate în blocuri de câte 4MB. Deplasarea s-a făcut din punctul A în punctul B şi invers. Viteza de deplasare a fost cea a unui mers normal. S-au făcut o set de măsurători fără a folosi placa wireless suplimentară şi un set având sistemul de asistenţă pentru handover activat.

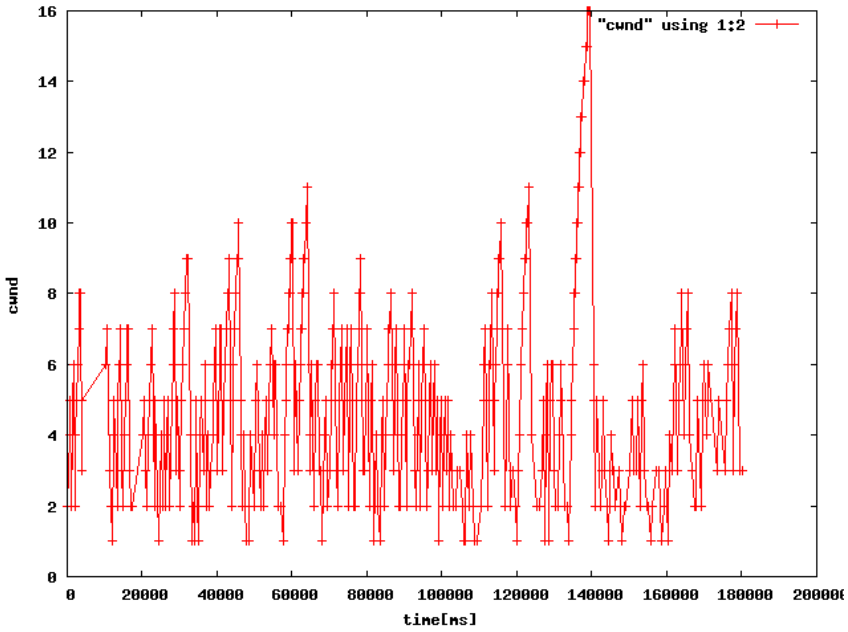


Figura 4-6 Evoluţia variabilei *cwnd* cu sistemul de asistenţă dezactivat

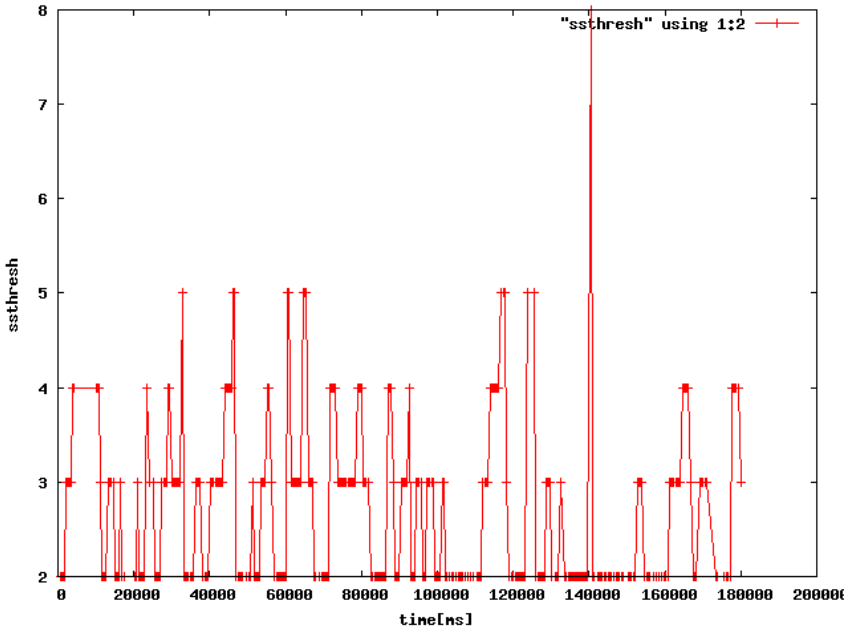


Figura 4-7 Evoluţia variabilei *ssthresh* cu sistemul de asistenţă dezactivat

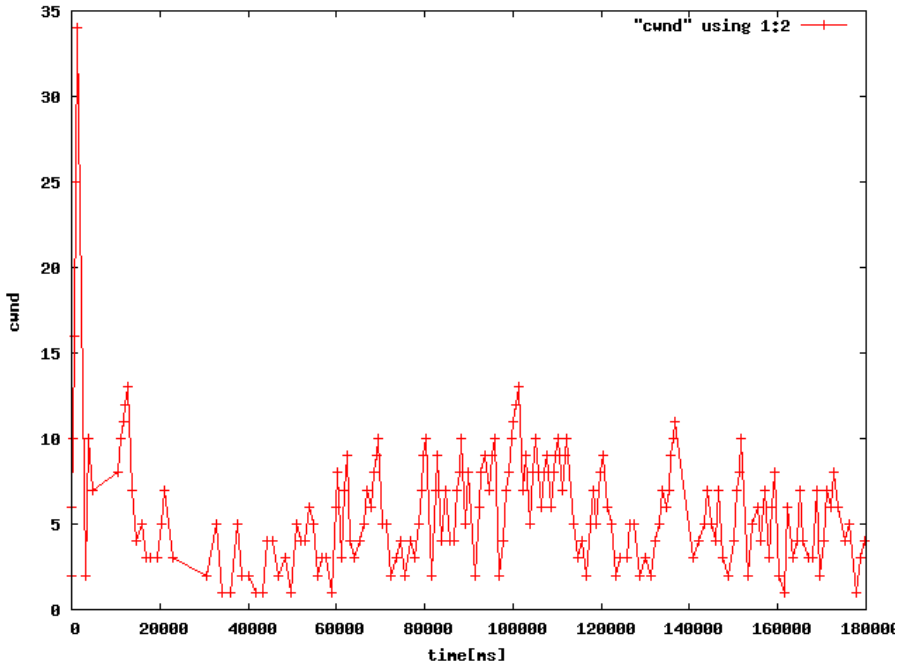


Figura 4-8 Evoluția variabilei *cwnd* cu sistemul de asistență activat

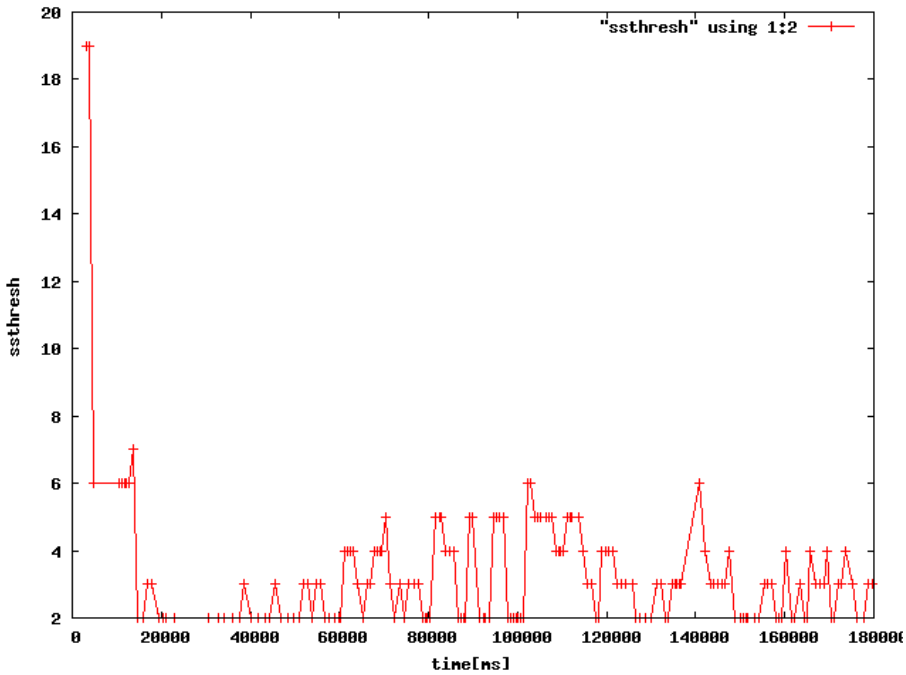


Figura 4-9 Evoluția variabilei *ssthresh* cu sistemul de asistență activat

Graficele de mai sus au fost realizate folosind o secvență din fişierele generate cu ajutorul metodei de citire a variabilelor direct din kernel. Primele două grafice (figurile 4-6 şi 4-7) înfățişează evoluția variabilelor *cwnd* şi *ssthresh* fără ca sistemul de asistență să fi fost activat, iar celelalte două grafice (figurile 4-8 şi 4-9) au fost realizate cu sistemul de asistență activat.

Analizând cele două perechi de grafice se observă o îmbunătăţire simţitoare în ceea ce priveşte menţinerea unei conexiuni TCP de bună calitate atunci când este folosit sistemul de asistenţă la reasociere.

4.4 Concluzii

Pentru a menţine o conexiune radio de bună calitate trebuie prevăzute mecanisme prin care un dispozitiv mobil sa fie capabil, fie independent, fie asistat sa se asocieze la AP-ul care oferă calitatea cea mai bună a conexiunii. In manieră clasică acest deziderat se realizează prin faptul ca un dispozitiv mobil monitorizează in permanenţă calitatea semnalului primit de la AP-ul la care este asociat. În momentul când acest semnal îşi pierde drastic din putere, dispozitivul mobil ar trebui sa înceapă procesul de reasociere, adică să scaneze mediul de transmisie pentru a detecta prezenţa altor AP-uri care să ofere o calitate mai bună a semnalului. Daca este detectat un astfel de AP, atunci asocierea cu vechiul AP va fi anulată şi se va realiza o nouă asociere. În practică se constată ca lucrurile nu stau chiar aşa. Se poate demonstra că procesul de reasociere se declanşează doar în momentul când conexiunea cu AP-ul curent se întrerupe. Acest lucru duce la o întrerupere temporară a conexiunii TCP şi mai grav, daca degradarea conexiunii nu merge până la acel nivel la care ea să se întrerupă, va continua să existe, dar în parametrii mult inferiori, chiar daca în zonă există AP-uri care ar putea oferi alternative mai bune din punct de vedere al calităţii semnalului. Acest lucru se traduce prin pierderi de cadre, ceea ce va atrage retransmisii atât la nivelul MAC, dar şi la nivel de protocol TCP, generând o comunicaţie de proastă calitate şi un consum suplimentar de energie, deci o autonomie redusă.

În cadrul acestui capitol a fost prezentată şi testată o metodă pusă la punct pentru a oferi asistenţă unui dispozitiv mobil în cadrul procesului de reasociere. Rezultatele testelor demonstrează eficienţa metodei propuse, care se remarcă totodată şi prin simplitate.

Capitolul 5 Localizarea unui dispozitiv mobil într-o rețea WLAN 802.11

5.1 Introducere

Capacitatea de localizarea este un aspect esențial atunci când locația unui dispozitiv mobil reprezintă un parametru important al unui sistem de tip *context*-aware, sistem din care acel dispozitiv face parte. Folosind locația dispozitivului se pot face diverse optimizări care pot influența atât performanța acelui dispozitiv, dar şi performanța sistemului în ansamblu.

Problema localizării este una de larg interes [FMSA09], ea având diverse domenii de aplicabilitate. Cele mai răspândite metode de localizare se bazează pe citirea puterii semnalului recepționat [WKC07, Lee07, EXM04, LKH+06] şi determinarea distanței față de sursă, dar au fost propuse, de-a lungul timpului şi sisteme de localizare bazate pe alte principii. De exemplu în [HHS+02] sunt folosite ultrasunetele pentru localizare, în [WHF+92] s-au folosit undele infraroşii, iar în [RA07] radiația luminoasă. Un loc aparte îl ocupă metodele de localizare care folosesc pentru determinarea distanței față e un emițător timpul de propagare a informației (Time Of Arrival – TOA), calcularea acestui timp bazându-se pe valoarea obținută pentru RTT (Round Trip Time). O altă abordare deosebită este realizată în [NN04] unde pentru determinarea locației sunt folosite unghiurile sub care este recepționat semnalul de către receptor, unghiurile calculându-se față de un sistem de referință. Deci metoda folosită are la bază principiile **triangulației**. În acest caz este necesară folosirea de antene direcționale. Metodele care folosesc distanța față de nişte puncte de referință pentru a determina locația unui anumit dispozitiv in spațiu, se bazează pe principiile **trilaterației**.

În categoria metodelor bazate pe determinare puterii semnalului există o clasă care foloseşte ca principiu de bază realizarea în prealabil a unei hărți care cuprinde valorile semnalului recepționat, asociate cu distanța reală față de emițător, în cât mai multe puncte din spațiul unde se doreşte localizarea. Această fază fiind una de antrenare a sistemului. Apoi, distanța unui dispozitiv mobil față de sursa de semnal este determinată prin compararea puterii citite cu valorile conținute pe acea hartă cu puterile semnalului in diverse puncte [Kjæ08, TK07]. O abordare mai complexă este realizată în [Bol08] unde algoritmul de localizare este rulat pe smarphone-uri şi în felul acesta pot fi luate în considerare la calcularea poziției, pe lângă semnalul din rețele 802.11 şi informațiile provenite din GSM şi Bluetooth. O altă lucrare în care sunt folosite mai multe standarde de comunicație radio pentru a localiza un dispozitiv în interiorul dar şi în exteriorul clădirilor este [RD07]. În [ZHK+07] se încearcă punerea la punct a unui sistem care să permită localizarea în interiorul unui spațiu, folosind un singur access point şi o hartă a semnalului. Precizia care s-a dorit să fie atinsă, a fost aceea de a indica prezența într-una din încăperile unei locuințe. Un mod de abordare interesant este prezentat în [PA06]. Aici autorii propun un sistem de localizare care să permită restricționarea accesului neautorizat la rețeaua wireless, dar implementarea lui este condiționată de existența unui sistem centralizat de control al access point-urile şi ele trebuie să permită reglarea puterii semnalului emis în mod dinamic. Autorii din [CPB+08] propun o metodă de a construi, pe baza puterii semnalului recepționat, liste de proximitate care conțin persoane şi dispozitive .

În [KK08] se încearcă găsirea unei soluții de localizare care să nu interfereze foarte mult cu aplicațiile în timp real care rulează pe acelaşi dispozitiv. Pentru a clasifica tipurile de proximități este introdus un aşa numit grad de similaritate, de fapt o metodă statistică de evaluare.

5.2 Noțiuni generale privind localizarea folosind sisteme radio

5.2.1 Măsurarea puterii semnalului recepționat

Semnalul emis de o antenă se va propaga în toate direcțiile, dar nu cu aceiaşi putere, aceasta depinzând de tipul antenei folosite. Un rol important in studiul antenelor îl ocupă antena izotropă. Ea este un model ideal şi reprezintă un radiator ipotetic punctiform, care generează unde electromagnetice cu o distribuție uniformă în spațiu.

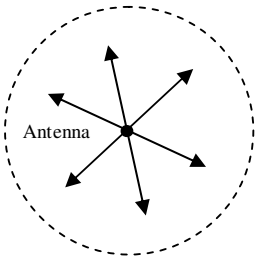


Figura 5-1 Antena izotropă

Una dintre principalele caracteristici ale antenelor reale este directivitatea, care reprezintă neuniformitatea distribuției puterii radiante în diferite direcții. Caracteristica de radiație arată cum este distribuită puterea radiată în spațiul din jurul antenei.

Un alt tip de antenă, reală de această dată, este dipolul electric. Caracteristica de directivitate este redată în figura 5-2. Cu ajutorul dipolului sunt create antenele omnidirecționale, care emit uniform radiație electromagnetică în planul orizontal, perpendicular pe elementul radiant.

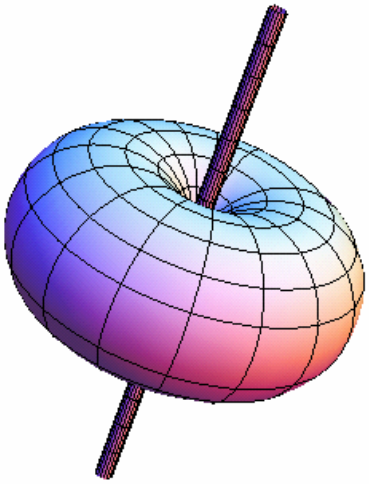


Figure 5-2 Modul de propagare al undelor electromagnetice în cazul dipolului

Pentru orice tip de comunicație radio, semnalul este atenuat odată cu distanța. Formula care exprimă puterea semnalului la o anumită distanță de emițător se numeşte formula de propagare în spațiul liber. Aceasta este:

$$\frac{P_t}{P_r} = \frac{(4\pi d)}{\lambda^2} = \frac{(4\pi f d)^2}{c^2}$$

unde

P_t – puterea semnalului emis

P_r – puterea semnalului recepționat

d – distanța până la receptor

λ – lungimea de undă a semnalului

f – frecvența semnalului

c – viteza luminii

Dacă se cunosc cele două puteri poate fii obținută distanța față de emițător la care se face recepția.

$$d = \sqrt{\frac{P_t}{P_r}}\left(\frac{c}{4\pi f}\right) \qquad (1)$$

Pentru măsurarea puterii semnalului s-a folosit un laptop cu interfață wireless. Ca emițător a fost folosit un AP de tipul LinkSys WAP55AG. Programul pentru citirea puterii a fost scris in Visual C++ şi el foloseşte funcțiile oferite de **NDIS** (**Network Driver Interface Specification).** Detalii despre această interfață au fost oferite în capitolul 4. Cu ajutorul programului poate fi monitorizată puterea semnalului provenit de la AP-ul curent sau pot fi monitorizate în paralel puterile semnalelor de la toate AP-urile aflate în zona de recepție a laptop-ului. Apoi, luând în calcul doar puterile semnalelor provenite de la AP-urile care se află în aceiaşi încăpere cu dispozitivul mobil se determină distanța acestuia față de AP-uri. Apoi, folosind metoda trilaterației este determinată poziția mobilului în interiorul acelui spațiu. Coordonatele AP-urilor sunt exprimate față de un punct de referință din interiorul încăperii, stabilit în prealabil şi față de care va fii exprimată şi poziția dispozitivului mobil.

În realitate, semnalul radio emis de AP este afectat de reflexii, interferențe şi atenuări datorate geometriei încăperii în care sunt instalate AP-urile, precum şi obstacolelor care pot să se interpună între emițător şi receptor (ex. mobilier, aparatură sau persoane aflate în încăpere). Dacă receptorul se deplasează relativ constant față de emițător, într-un proces continuu de depărtare sau apropiere față de sursă, puterea semnalului recepționat ar trebui să scadă sau să crească constant. În realitate valorile recepționate arată ca în figura 5-3.

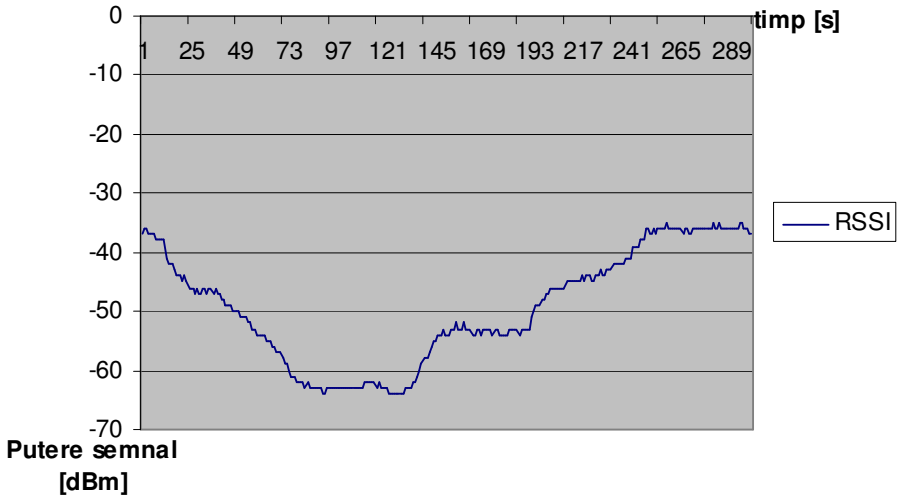


Figura 5-3 Valoarea reală a puterii semnalului recepționat exprimată în dBm

Pentru citirea puterii semnalului au fost făcute două interogări pe secundă folosind funcțiile oferite de interfața NDIS. Semnalul este exprimat în dBm. Se observă cum pe graficul care arată puterea semnalului apar acele oscilații. Pentru a „curăța" semnalul a fost aplicat un filtru trece jos. Rezultatul aplicării filtrului se observă în figura 5-4.

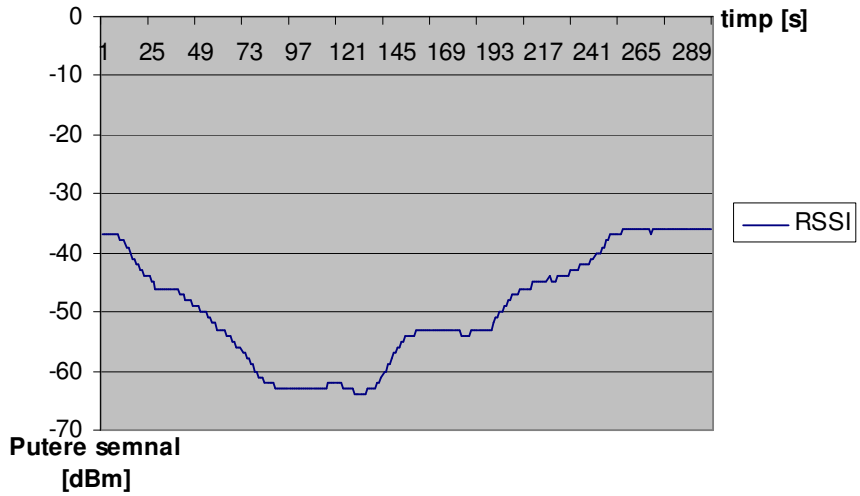


Figura 5-4 Forma semnalului după aplicarea unui filtru trece jos.

Pentru a elimina oscilațiile puterii semnalului s-a construit un filtru bazat pe calcularea unei medii ponderate a puterii semnalului recepționat.

$$RSSI_{ponderat} = (1 - f)RSSI_{med} + fRSSI \qquad (2)\text{, unde}$$

RSSI - Received Signal Strength Indicator

$$RSSI_{med} = \frac{\sum_{i=n-500}^{n} RSSI_i}{n}$$

Valorile pe care le poate lua f sunt în intervalul [0,1]. Dacă valoarea lui f este 1, atunci media ponderată coincide cu valoarea instantanee a puterii semnalului recepţionat, adică $RSSI$, iar dacă f este 0, atunci $RSSI_{ponderat}$ este egal cu media aritmetică a ultimelor 500 de valori recepţionate pentru $RSSI$. Cu cât valoarea lui f este mai mare, cu atât $RSSI_{ponderat}$ reacţionează mai prompt la modificări ale lui $RSSI$. Dacă se doreşte implementarea unui filtru care să elimine oscilaţiile nedorite ale puterii semnalului recepţionat, atunci se alege un f cât mai mic, doar că în cazul acesta sistemul nu va mai reacţiona prompt în cazul în care dispozitivul mobil se află în mişcare, iar modificarea puterii semnalului recepţionat este una reală.

Pentru a calcula distanţa faţă de emiţător trebuie cunoscută puterea cu care acesta emite. De obicei producătorii nu specifică puterea cu care AP-urile emit, aşa că aceasta valoare trebuie dedusă prin alte mijloace. Metoda folosită în această lucrare a fost aceea de a măsura valoarea puterii semnalului în imediata vecinătate a emiţătorului.

Pentru a determina exact puterea cu care emite AP-ul s-au făcut o serie de măsurători în imediata vecinătate a acestuia. Pentru a valida corectitudinea măsurătorilor, pe lângă laptop-ul pe care a fost instalat si rulat programul de determinare a puterii semnalului s-a folosit şi un dispozitiv specializat, un analizor pentru reţele wireless (EherScope Series II), produs de firma Fluke Networks. Acesta este capabil, pe lângă alte funcţii, să măsoare puterea semnalului provenit de la un AP. Fiind un dispozitiv special proiectat pentru asemenea operaţii de diagnosticare a reţelelor wireless, a fost folosit ca şi etalon pentru măsurătorile efectuate.

O întrebare la care s-a căutat răspuns a fost aceea de a vedea dacă echipamente fabricate de producători diferiţi sau chiar diferite modele provenite de la acelaşi producător să emită cu puteri diferite. Pentru testări s-au folosit 4 modele de AP-uri, provenite de la două firme, Linksys şi Dlink. Cele 2 modele de la firma Linksys sunt WRT54GS, WAP55AG şi WAP54G, iar cele doua modele provenite de la Dlink sunt DWL-G700AP şi DI-624.

De fapt nu se va putea măsura puterea cu care emite un AP ci se va măsura valoarea puterii semnalului în imediata vecinătate a acestuia. Pentru a determina această putere, s-au plasat atât laptop-ul cât şi analizorul wireless EherScope la distanţa de 50cm faţă de fiecare AP în parte. Pentru fiecare AP s-au făcut serii de câte 300 de citire, cu câte 2 citiri pe secundă, apoi s-a făcut media acestor citiri. S-a ales distanţa de 50cm pentru că s-a observat experimental că începând cu această distanţă şi depărtându-ne de AP-ul emiţător, media valorilor măsurate pentru puterea semnalului începe să scadă. În intervalul 0-50cm media valorilor măsurate rămâne constantă. Rezultatele măsurătorilor sunt redate mai jos.

WRT54GS	WAP55AG	WAP54G	G700AP	DI-624
-32dBm	-30dBm	-33dBm	-29dBm	-32dBm

Figura 5-5 Măsurarea puterii semnalului în imediata vecinătate a diverselor tipuri de AP-uri

De aici tragem concluzia că nu toate AP-urile emit cu aceiaşi putere, aceasta variind de la un model la altul. Deci este nevoie de o fază de calibrare a sistemului de localizare, fază în care pentru fiecare AP în parte să fie determinată puterea cu care acesta emite. Este vorba despre P_t din ecuația (1). În sistemul nostru de localizare, această putere P_t va fi asimilată cu una din valorile prezentate în tabelul din figura 5-5.

În următoare etapă a fost necesar să se verifice dacă formula teoretică de propagare a semnalului radio în spațiul liber poate fi folosită pentru a determina distanța unui dispozitiv mobil față de un AP.

Pentru a valida măsurătoarea făcută s-a procedat în felul următor. Dintre cele 5 modele de AP-uri s-a ales pentru început modelul WAP55AG.

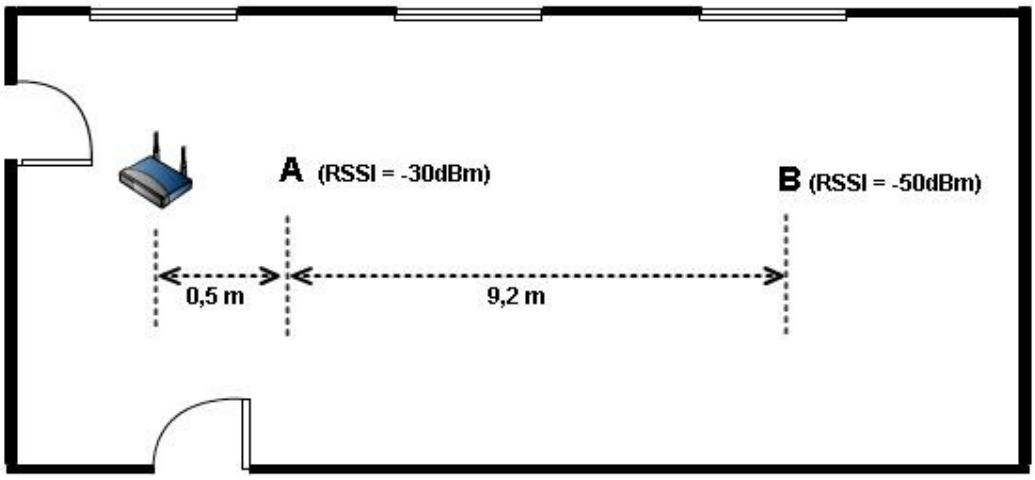


Figura 5-6 Verificarea formulei de propogare a semnalului radio în spațiul liber

Laptop-ul cu care se citeşte puterea semnalului a fost depărtat de access point până în momentul în care puterea semnalului a scăzut la -50dBm (Figura 5-6). S-a marcat locul şi s-a măsurat distanța până la AP. Înlocuind în formula (1), Pt cu valoarea de -30dBm şi Pr cu valoarea de -50dBm, valori transformate în prealabil în mW, s-a obținut distanța de 10m. Deci, conform formulei, ar fi trebuit ca laptop-ul să se găsească acum la distanța de 10m față de AP. Distanța reală, obținută prin măsurare a fost de 9,2 m. Deci eroarea obținută a fost de 0,8m, ceea ce este suficient de mulțumitor.

S-a procedat în mod similar şi cu restul AP-urilor, erorile obținute variind între 0.7 şi 1.5 m. Pentru a elimina aceste erori s-a introdus un factor de corecție k, specific fiecărui AP în parte. Astfel formula (1) devine $d = \sqrt{\frac{P_t}{P_r}}\left(\frac{c}{4\pi f}\right) + k$. (3)

Reamintim un aspect legat de aceste măsurători: valorile obținute pentru puterea semnalului au fost obținute prin calcularea mediei unui set de 200 de citiri succesive. S-a procedat în felul acesta pentru a elimina erorile care apar datorită oscilațiilor puterii semnalului datorare reflexiilor din interiorul încăperii. Într-o situație reală, când se încearcă localizarea unui dispozitiv mobil aflat în mişcare, va trebui aplicată altă metodă pentru atenuarea acestor erori.

Pentru a observa comportamentul semnalului emis de AP-uri şi eventual pentru a putea pune în evidenţă un anumit şablon de variaţie a semnalului s-au făcut mai multe serii de măsurători experimentale [MFG07, MF06], fiecare din ele urmărind să pună în evidenţă un anumit aspect. Au fost în final izolate câteva cazuri de test şi ele sunt descrise în cele ce urmează.

Cazul 1
Date iniţiale: - distanţă fixă faţă de AP
- AP-uri de diverse tipuri
Obiective: **-** observarea modului de variaţie a puterii semnalului pentru diverse modele de AP-uri

În acest caz s-au folosit toate cele 5 tipuri de AP-uri, dar vor fi redate doar rezultatele pentru patru din ele. Dispozitivul mobil (laptop-ul) s-a aflat de fiecare dată la aceiaşi distanţă de 6 m faţă de fiecare din ele. În graficele prezentate este redată variaţia distanţei, obţinută aplicând formula (3). Pentru fiecare caz în parte s-au făcut aproximativ 3500 de citiri, câte două pe secundă.

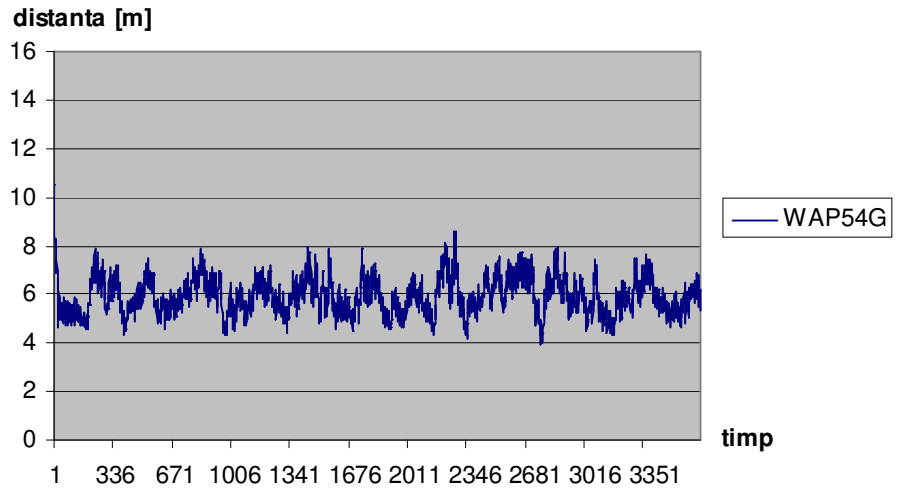


Figura 5-7 Valori oscilante, obţinute pentru distanţa dintre un AP, model WAP54G şi dispozitivul mobil care se află la distanţa fixă de 6m

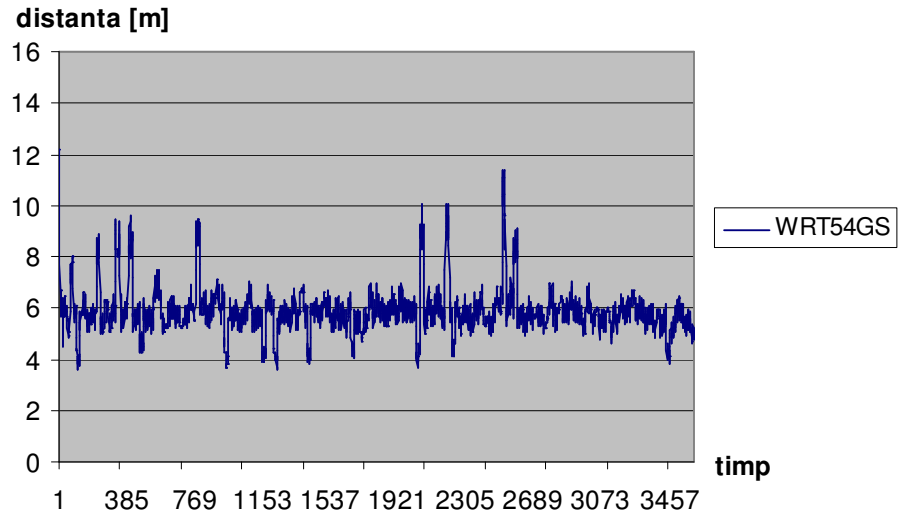


Figura 5-8 Valori oscilante, obținute pentru distanța dintre un AP, model WRT54GS şi dispozitivul mobil care se află la distanța fixă de 6m

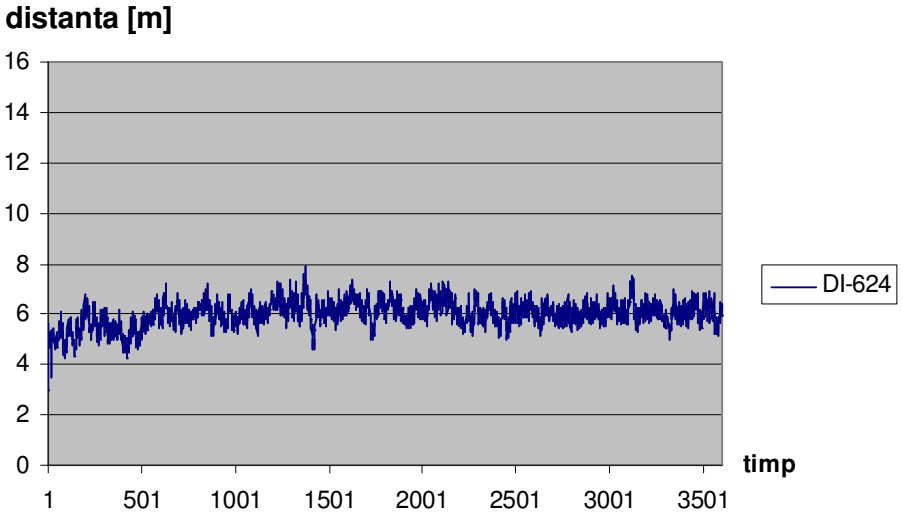


Figura 5-9 Valori oscilante, obținute pentru distanța dintre un AP, model DI-624 şi dispozitivul mobil care se află la distanța fixă de 6m

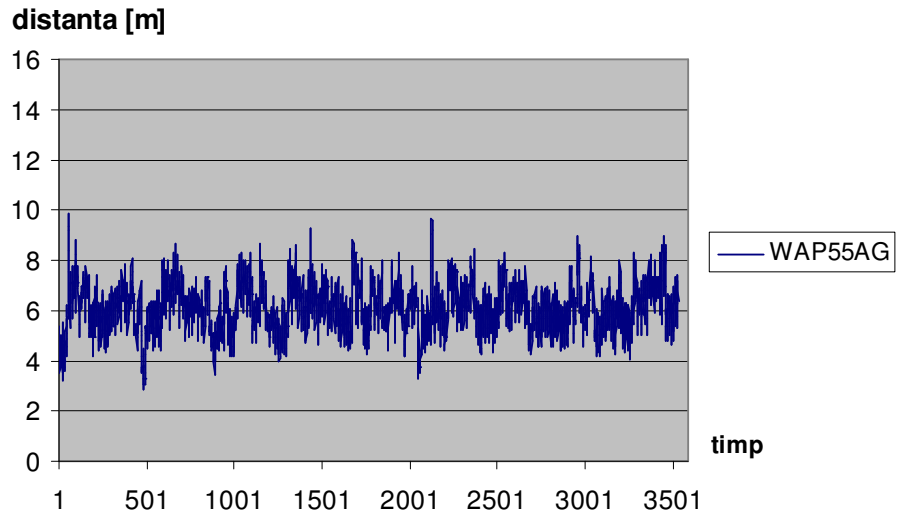


Figura 5-10 Valori oscilante, obținute pentru distanța dintre un AP, model WAP55AG şi dispozitivul mobil care se află la distanța fixă de 6m

Concluzii: Chiar dacă toate cele patru AP-uri au fost situate la aceiaşi distanță față de dispozitivul mobil, se observă că valorile obținute pentru distanță oscilează. Acest lucru se datorează faptului că puterea semnalului recepționat nu este constantă şi cum distanța se calculează pe baza acestei puteri, apar şi oscilații ale distanței chiar dacă dispozitivul mobil are o poziție fixă.

Variația puterii semnalului recepționat este un aspect care a fost pus în evidență, dar pe de altă parte se observă că modul de variație diferă şi de la un model de AP la altul. Se observă ca sunt dispozitive pentru la care oscilațiile sunt mai ample, altele la care acestea sunt mai reduse. Concluzia este că pentru fiecare dispozitiv în parte trebuie ales cu grijă acel factor f care intervine în media ponderată din formula (2), astfel încât să se uniformizeze rezultatele pentru calcularea distanțelor, indiferent de tipul AP-ului luat în considerare.

Cazul 2

Date inițiale: - măsurători efectuate la diverse distanțe față de un AP
- este păstrat acelaşi tip de AP de la o măsurătoare la alta

Obiective: **-** punerea în evidență a modului de variație a puterii semnalului funcție de distanța față de emițător

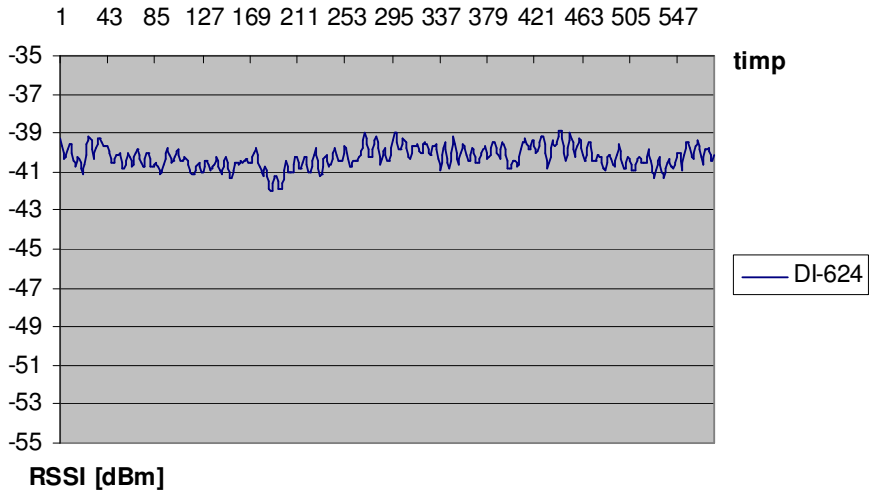


Figura 5-11 RSSI recepționat la distanța de 3m față de AP

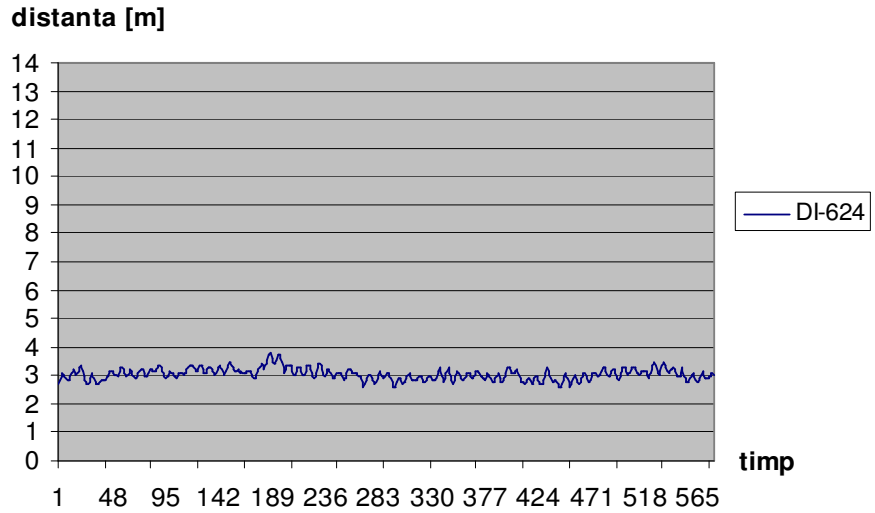


Figura 5-12 Valori oscilante pentru distanța calculată, atunci când distanța reală este de 3m

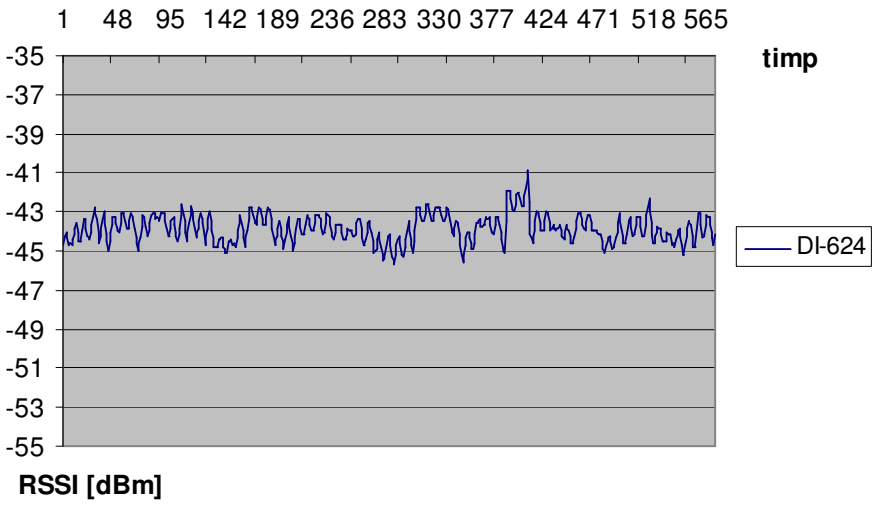


Figura 5-13 RSSI recepționat la distanța de 5m față de AP

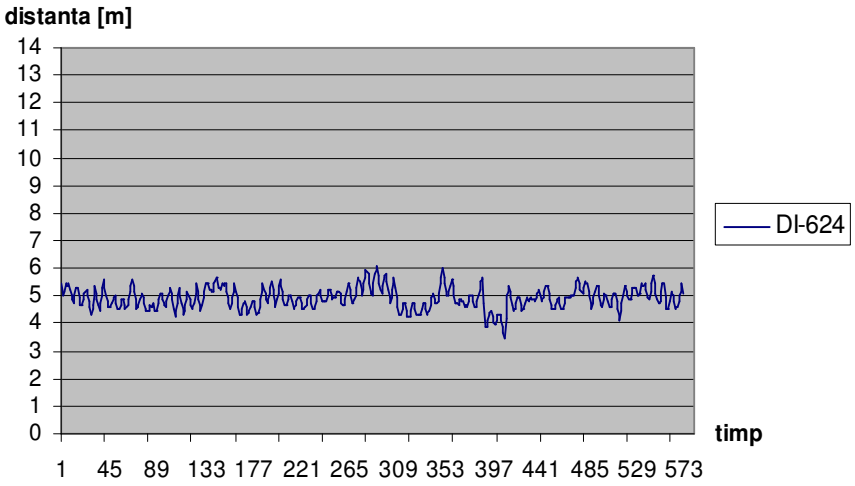


Figura 5-14 Valori oscilante pentru distanța calculată, atunci când distanța reală este de 5m

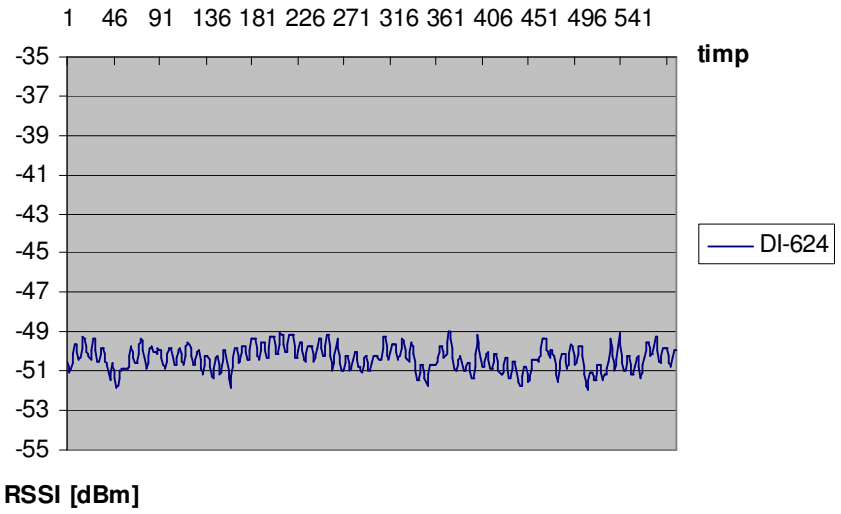


Figura 5-15 RSSI recepționat la distanța de 10m față de AP

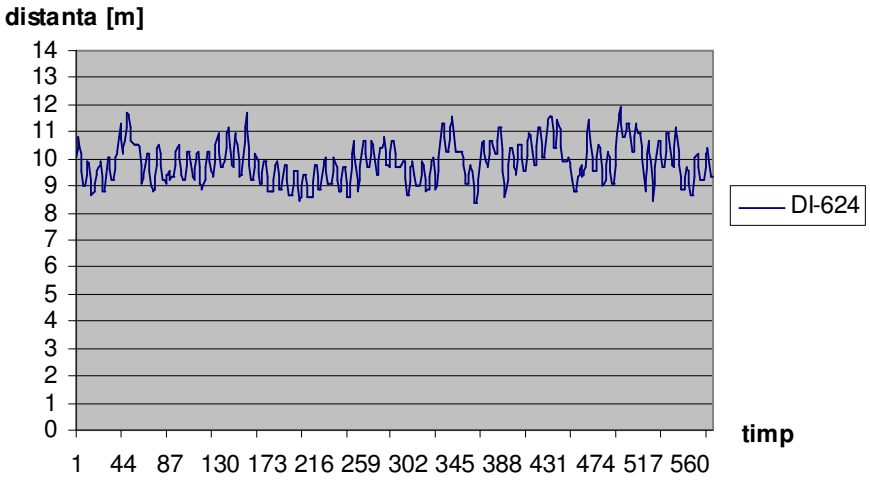


Figura 5-16 Valori oscilante pentru distanța calculată, atunci când distanța reală este de 10m

Concluzii: Acest set de măsurători a încercat să pună în evidență modul diferit de variație a puterii semnalului funcție de distanță. Se observă relativ ușor cum odată cu creşterea distanței apar şi variații mai ample ale puterii semnalului recepționat.

Pentru a reduce efectul acestora, ar trebui aleasă o valoare mai mică pentru factorul f din suma ponderată. Acest lucru poate să aducă oarecare îmbunătățiri ale preciziei de localizare, atunci când distanța dintre sursă şi receptor este mai mare, dar pentru distanțe mai mici, măsura este inutilă şi chiar duce la o reacție înceată a sistemului în cazul în care apare o deplasare a dispozitivului mobil.

Cazul 3

Date inițiale: - se foloseşte un acelaşi tip de AP
- dispozitivul mobil îşi modifică poziția trecând dintr-o încăpere în alta

Obiective: - încearcă să pună în evidență modul de variație a semnalului la trecerea dintr-o încăpere în alta precum şi variația determinată de obturarea semnalului de către diverse persoane care se deplasează prin sala în care au avut loc testele.

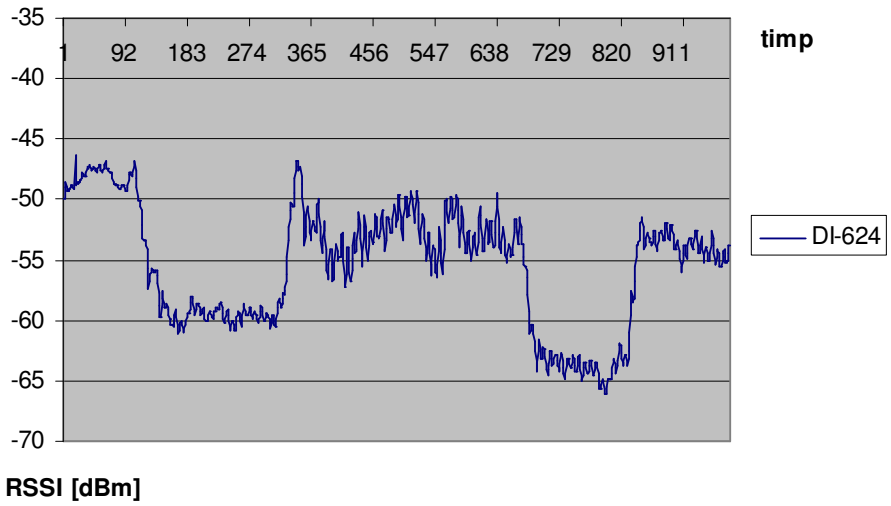


Figura 5-17 Variația semnalului la trecerea dintr-o încăpere în alta precum şi în cazul interpunerii între emițător şi receptor a unor persoane

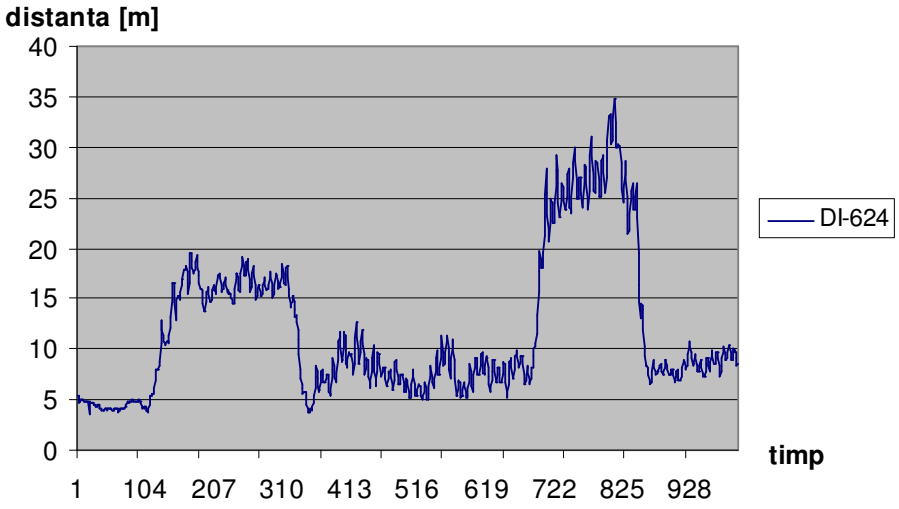


Figura 5-18 Variația distanței obținută pe baza semnalului din figura 5-17

În graficele 5-17 şi 5-18 este redată variația puterii semnalului şi variația distanței calculate pe baza acestei puteri. În prima parte a graficelor dispozitivul mobil se află în poziție fixă față de receptor. Urmează deplasarea în încăperea alăturată, urmată

concomitent cu intrarea în sală a mai multor persoane care se deplasează în apropierea laptop-ului cu care se fac măsurătorile. În a treia parte se revine în prima sală, iar personale continuă să se deplaseze în vecinătatea dispozitivului mobil. Din nou se trece în sala alăturată, dar se ajunge la o distanţă mai mare faţă de emiţător şi apoi, pe finalul graficului, se revine în prima sală, persoanele părăsind şi ele încăperea.

Prima problemă care o evidenţiază graficele din figura 5-17 se referă la modul în care semnalul este atenuat de către persoanele aflate în încăpere. Acest gen de atenuări conduc la variaţii ale puterii semnalului cu valori cuprinse între 3dBm şi 5dBm după cum se observă în graficul din figura 5-17.

O altă problemă care a trebuit să fie tratată a fost aceea de a determina când dispozitivul mobil părăseşte încăperea. Această problemă a fost rezolvată tot într-o manieră experimentală. Măsurătorile au fost făcute folosind laptop-ul cu interfaţă wireless cât şi analizorul FlukeNetworks pentru a putea compara rezultatele. Procesul de măsurare s-a realizat plasând AP-ul la diferite distanţe faţă de perete, apoi laptop-ul şi analizorul Fluke au fost plasate mai întâi de aceiaşi parte a peretelui ca şi AP-ul (figura 5-19, pozitia A) iar apoi de partea cealaltă a peretelui (poziţia B).

În urma mai multor seturi de măsurători s-a ajuns la următoarea concluzie: în funcţie de grosimea peretelui se obţin variaţii ale puterii semnalului cuprinse între 15dBm şi 20dBm. Aceste valori nu depind de distanţa dintre AP şi perete, lucru care era de aşteptat, dar în schimb depind de grosimea peretelui.

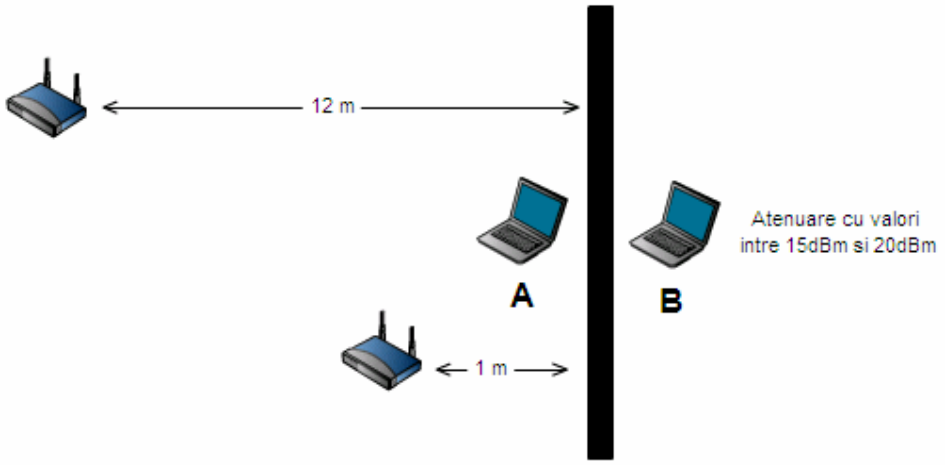


Figura 5-19 Măsurarea atenuării la trecerea semnalului printr-un perete

Rezultatele obţinute de noi sunt specifice clădirii în care au avut loc experimentele. Pentru implementarea unui sistem de localizare, trebuie să se ţină cont de acest parametru, care se referă la natura pereţilor din care este realizată clădirea şi care va determina alte valori de atenuare a semnalului.

Folosind aceste date obţinute pe cale experimentală poate fi trasă concluzia că dispozitivul mobil a părăsit încăperea doar atunci când faţă de toate AP-urile din încăpere semnalul scade cu valuri cuprinse între 15dBm şi 20dBm.

Concluzii: Realizând mai multe seturi de astfel de măsurători s-a putut trage concluzia că trecerea dintr-o încăpere în alta duce la variaţii ale semnalului cuprinse între 15dBm şi 20dBm, iar atenuările generate de persoanele din încăpere sunt în intervalul 3dBm şi 5dBm.

5.2.2 Trilaterația

Termenul defineşte o clasă de metode pentru determinarea poziției unui punct, cunoscându-se distanțele dintre punctul a cărui poziție trebuie determinată şi alte trei puncte cu poziție dată. După cum reiese şi din denumire, metoda este aplicabilă luând în calcul cel puțin trei puncte cu poziție cunoscută, însă numărul acestora poate să fie şi mai mare de 3.

Presupunem că avem trei puncte P_1, P_2 şi P_3, cu poziții date, iar M este punctul a cărui poziție trebuie determinată. Distanța dintre punctul M şi cele trei le notăm cu d_1, d_2, respectiv d_3.

În acest caz, punctul nostru se află la intersecția celor trei cercuri cu centrul în punctele cunoscute şi de raze d_1, d_2, d_3, ca în figura 5-20.

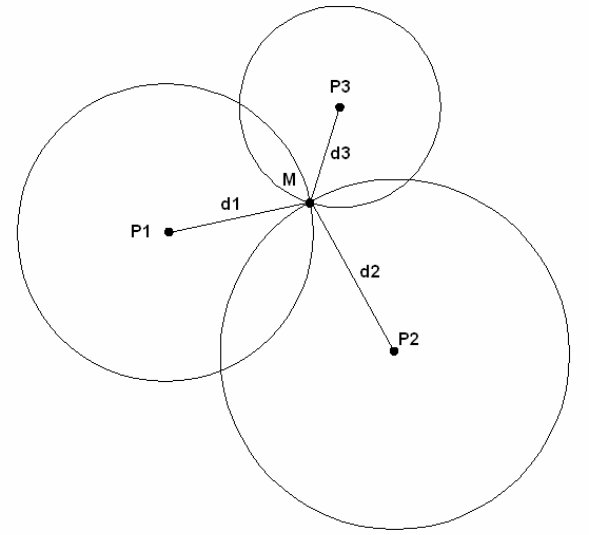


Figura 5-20 Principiul trilaterației când toate punctele sunt în plan

$$M:\begin{cases}(x_M-x_1)^2+(y_M-y_1)^2=d_1^2\\(x_M-x_2)^2+(y_M-y_2)^2=d_2^2\\(x_M-x_3)^2+(y_M-y_3)^2=d_3^2\end{cases} \qquad (4)$$

Punctul M, fiind punct de intersecție, trebuie să satisfacă ecuațiile cercurilor cu centrele în punctele P_1, P_2 şi P_3 şi de raze d_1, d_2 şi d_3. Rezolvând sistemul de trei ecuații se obțin coordonatele (x_M, y_M) ale punctului M..

Discuția s-a purtat pentru situația în care atât punctul M cât şi cele trei puncte fixe P_1, P_2 şi P_3 erau în acelaşi plan. Dacă ar trebui să luăm în calcul şi situația în care punctul M nu se află în acelaşi plan cu P_1, P_2 şi P_3, atunci am avea de-a face cu o aşezare tridimensională a celor patru puncte (figura 5-21) şi ecuațiile (*4*) trebuie rescrise pentru a descrie nişte sfere care se intersectează (5). Noul sistem de ecuații este următorul:

$$\Omega:\begin{cases}(x_M-x_{AP_1})^2+(y_M-y_{AP_1})^2+(z_M-z_{AP_1})^2=d_1^2\\(x_M-x_{AP_2})^2+(y_M-y_{AP_2})^2+(z_M-z_{AP_2})^2=d_2^2\\(x_M-x_{AP_3})^2+(y_M-y_{AP_3})^2+(z_M-z_{AP_3})^2=d_3^2\end{cases} \quad (5)$$

Rezolvându-l se obțin coordonatele (x_M, y_M, z_M) ale punctului M..

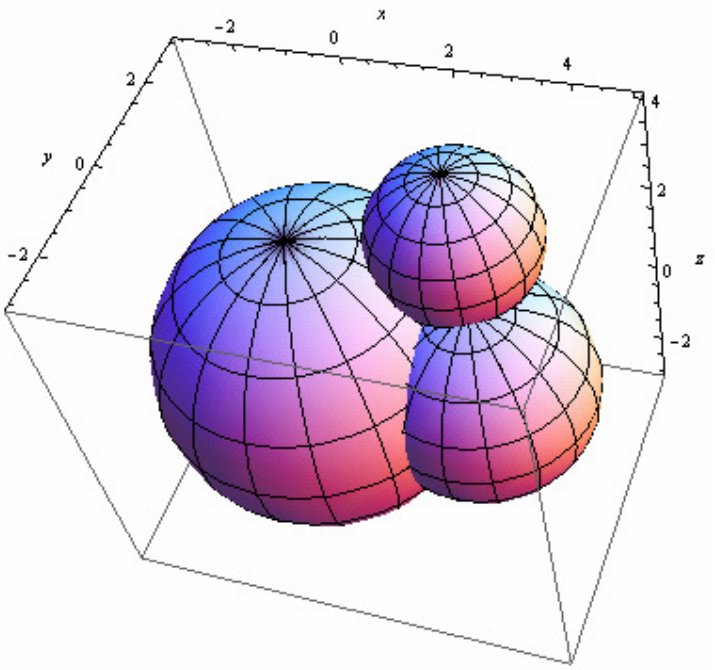


Figura 5-21 Principiul trilaterației aplicat în spațiul tridimensional

5.3 Metode de implementare a trilaterației

5.3.1 Modele teoretice

În cazul de față, rolul punctelor P_1, P_2 și P_3 este jucat de *access point*-uri, iar distanțele d_1, d_2 și d_3 sunt calculate măsurând puterea semnalului emis de către aceste *access point*-uri.

$$\begin{aligned}d_1&=\sqrt{\frac{P_{t1}}{P_{r1}}}\left(\frac{c}{4\pi f}\right)\\d_2&=\sqrt{\frac{P_{t2}}{P_{r2}}}\left(\frac{c}{4\pi f}\right)\\d_3&=\sqrt{\frac{P_{t3}}{P_{r3}}}\left(\frac{c}{4\pi f}\right)\end{aligned} \quad (6)$$

P_{ti} reprezintă puterea semnalului la ieşirea din emițător, iar P_{ri} este puterea semnalului citită la receptor, în cazul nostru punctul M.

Din păcate, într-o situație reală, calcularea distanței față de un emițător radio implică o serie de erori de măsurare inerente. În primul rând atenuarea semnalului nu se va face exact după formula care exprimă propagarea semnalului în spațiul liber, putând apărea perturbații şi atenuări temporare. O altă sursă de eroare putând fi chiar echipamentele folosite pentru citirea puterii semnalului. Chiar dacă în prealabil sunt realizate calibrări, nu există garanții clare că echipamentele nu vor introduce erori la citire.

Datorită acestor erori de citire, rezolvarea sistemului de ecuații (*4*) nu mai conduce la o soluție unică, iar intersecția cercurilor va delimita un întreg domeniu, notat cu D, ca în figura 5-22.

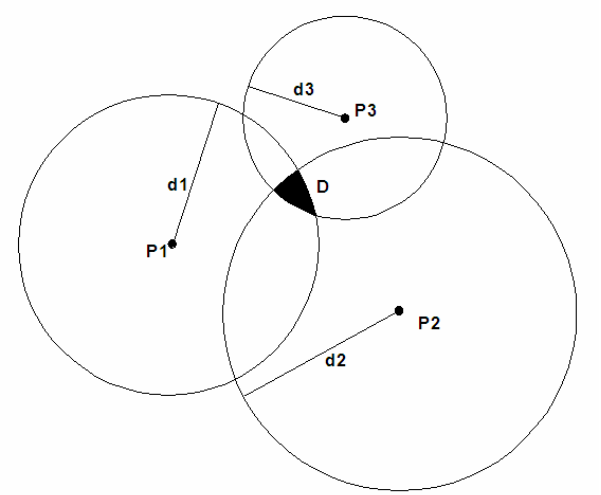


Figura 5-22 Intersecția cercurilor determină domeniul D

Astfel, sistemul de ecuații trebuie transformat într-un sistem de inecuații, care arată în felul următor:

$$D:\begin{cases}(x_M-x_1)^2+(y_M-y_1)^2\le d_1^2\\(x_M-x_2)^2+(y_M-y_2)^2\le d_2^2\\(x_M-x_3)^2+(y_M-y_3)^2\le d_3^2\end{cases} \qquad (7)$$

Soluția sistemului de inecuații va fi un domeniu D. Problema care se pune acum este aceea de a stabili cum anume va fi aproximată poziția punctului M în interiorul domeniului D. Un punct important prin proprietățile pe care le are în interiorul unui domeniu oarecare este centrul său de greutate, astfel că vom alege acest punct pentru a aproxima poziția punctului M în interiorul domeniului.

Folosind calculul integral, problema noastră s-ar putea rezolva în felul următor.

Metoda 1

$$x_M = \frac{\iint\limits_D x\rho(x,y)dxdy}{\iint\limits_D \rho(x,y)dxdy}$$
$$y_M = \frac{\iint\limits_D y\rho(x,y)dxdy}{\iint\limits_D \rho(x,y)dxdy} \qquad (8)$$

ρ este densitatea mediului.

Metoda 2

$$mass(D) = \iint\limits_D \rho(x,y)dxdy \qquad (9)$$

(*8*) exprimă masa domeniului D, iar ρ fiind densitatea mediului din domeniul respectiv. Centrul de greutate al acestui domeniu va fi dat de ecuaţiile următoare:

$$x_M = \frac{1}{mass(D)} \iint\limits_D x\rho(x,y)dxdy \qquad (10)$$
$$y_M = \frac{1}{mass(D)} \iint\limits_D y\rho(x,y)dxdy$$

Pentru cazul tridimensional, intersecţia sferelor va genera un domeniu Ω în spaţiu, care ar putea arata ca în figura 5-23.

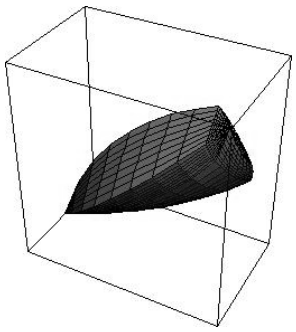


Figura 5-23 Domeniul Ω

Sistemul de inecuaţii care defineşte acest domeniu este:

$$\Omega : \begin{cases} (x_M - x_{AP_1})^2 + (y_M - y_{AP_1})^2 + (z_M - z_{AP_1})^2 \le d_1^2 \\ (x_M - x_{AP_2})^2 + (y_M - y_{AP_2})^2 + (z_M - z_{AP_2})^2 \le d_2^2 \\ (x_M - x_{AP_3})^2 + (y_M - y_{AP_3})^2 + (z_M - z_{AP_3})^2 \le d_3^2 \end{cases} \qquad (11)$$

Aici trebuie calculat centrul de greutate al domeniului Ω din spaţiu.

Metoda 1

$$x_M = \frac{\iiint_D x\rho(x,y,z)dxdydz}{\iiint_D \rho(x,y,z)dxdydz}$$
$$y_M = \frac{\iiint_D y\rho(x,y,z)dxdydz}{\iiint_D \rho(x,y,z)dxdydz} \qquad (12)$$
$$z_M = \frac{\iiint_D z\rho(x,y,z)dxdydz}{\iiint_D \rho(x,y,z)dxdydz}$$

Metoda 2

$$mass(\Omega) = \iint_\Omega \rho(x,y,z)dxdydz \qquad (13)$$

$$x_M = \frac{1}{mass(\Omega)} \iiint_\Omega x\rho(x,y,z)dxdydz$$
$$y_M = \frac{1}{mass(\Omega)} \iiint_\Omega y\rho(x,y,z)dxdydz \qquad (14)$$
$$z_M = \frac{1}{mass(\Omega)} \iiint_\Omega z\rho(x,y,z)dxdydz$$

După cum se observă, folosind calculul integral se obțin niște ecuații simple şi elegante (8, 9, 10, 12, 13, 14). Din păcate, este foarte greu de găsit un algoritm care să rezolve aceste ecuații, pentru cazul general, când domeniile D şi Ω ar putea lua orice formă [FMG08].

Din acest motiv s-a încercat elaborarea unor metode de calcul ale centrului de greutate care să poate fi transpuse cu uşurință sub forma unor algoritmi. Rezolvarea problemei face apel la geometrie şi ea va fi explicată imediat în subcapitolul 5.3.2.

Până acum am presupus că între emițător şi receptor nu se află nici un obstacol. Să analizăm totuşi cazul în care, în calea semnalului se interpune un obstacol. În figura 5-24, avem următoarea situație. În punctul A se află emițătorul, iar în punctul M citim puterea semnalului. Între emițător si receptor se interpune un obstacol de grosime w. Presupunem structura obstacolului necunoscută.

Puterea semnalului înainte de a penetra obstacolul este dată de formula:

$$P' = \frac{P_A}{d^2}\left(\frac{c}{4\pi f}\right)^2 \qquad (14)$$

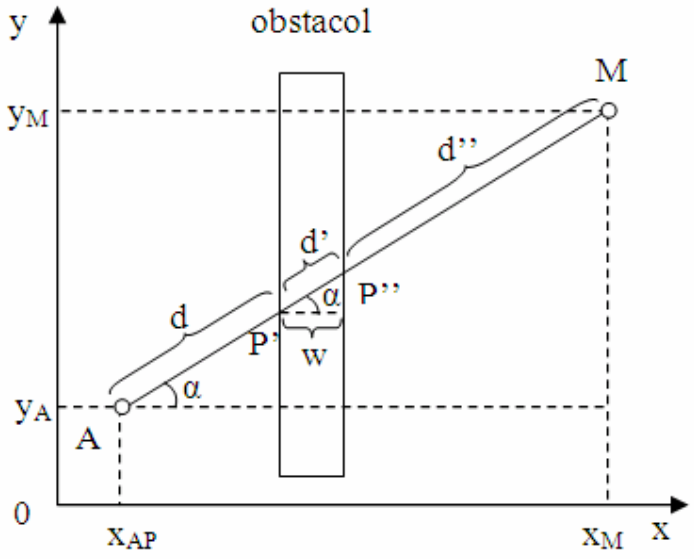


Figura 5-24 Propagarea semnalului printr-un obstacol

Pe partea cealaltă a obstacolului puterea semnalului este P''. Relaţia dintre P' şi P'' nu mai poate fi determinată cu formula propagării semnalului în spaţiul liber. Prin măsurători putem stabili pe cale experimentală un coeficient prin care să determinăm P'', cunoscând P'.

$$P''=kP'$$

Valoarea semnalului în punctul M este dată de formula:

$$P_M=\frac{P''}{d''^2}\left(\frac{c}{4\pi f}\right)^2 \quad (15)$$

Dacă obstacolul are o grosime mare atunci distanţa d' e dată de formula:

$$d'=\frac{w}{\cos\alpha}$$

În cazul nostru, în care încercăm să obţinem localizarea în interiorul unor clădiri, aceste obstacolele sunt reprezentate de obicei prin pereţi. Cum grosimea pereţilor este mult mai mică comparativ cu dimensiunea încăperilor, putem aproxima valoarea lui d' cu grosimea peretelui.

Dacă scriem acum ecuaţia cercului obţinem:

$$(x_M-x_{AP})^2+(y_M-y_{AP})^2+(z_M-z_{AP})^2=(d+d'+d'')^2 \quad (16)$$

Distanţa d' este neglijabilă, deci rămân în ecuaţie d şi d''. Pentru a putea obţine valorile d şi d'' ar trebui să putem obţine deduce valorile P' şi P'', iar cum acest lucru nu este posibil, trebuie găsită o altă soluţie. Dacă s-ar cunoaşte factorul de atenuare introdus perete atunci am putea scrie formula pentru distanţă în felul următor:

$$d+d'=\sqrt{\frac{P_A}{P_M+P_\Delta}}\left(\frac{c}{4\pi f}\right) \quad (17)$$

Unde P_Δ este diferenţa dintre P' şi P''. Cum anume se obţine această valoarea vom vedea în cele ce urmează.

5.3.2 Adaptări practice ale modelelor teoretice

Ținând cont de cele prezentate până acum, vom face o serie de simplificări pentru a putea obține un model funcțional [MFG+07, FMS+09], care să poată fi implementat. Prima simplificare este aceea că AP-urile cât şi punctul a cărui poziție vrem să o determinăm se află în acelaşi plan, deci vom aplica formulele din cazul bidimensional. A doua ipoteză simplificatoare este aceea că AP-urile folosite pentru localizare cât şi dispozitivul care trebuie localizat se află în aceiaşi încăpere.

Metoda poligonului

Pentru aplicarea metodei **[FMG10]** facem presupunerea că atât AP-urile cât şi dispozitivul mobil care trebuie localizat se află toate în acelaşi plan. Metoda nu este limitată la un anumit număr de AP-uri, cu cât sunt mai multe cu atât precizia în determinarea localizării creşte.

Pentru exemplificare considerăm că avem patru AP-uri dispuse ca în figura 5-25. Ele alcătuiesc vârfurile unui poligon convex.

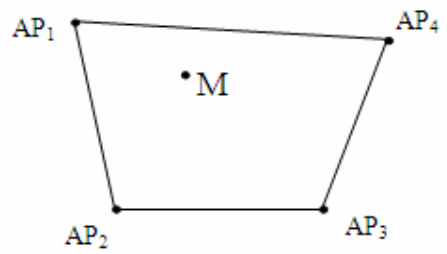


Figura 5-25 Dispunerea AP-urilor sub forma unui poligon convex

Distanța dintre M şi fiecare AP este dată de formula

$$d_i = \sqrt{\frac{P_{ti}}{P_{ri}}}\left(\frac{c}{4\pi f}\right)$$

Fiecare distanță obținută prin citirea puterii semnalului provenit de la cele patru AP-uri în punctul M va defini câte un cerc cu centrul în AP şi de rază d_i. Intersecția acestor cercuri arată ca în figura 5-26.

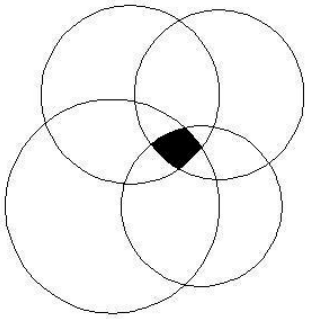


Figura 5-26 Domeniul generat în urma intersecției celor patru cercuri

Acum grupăm două câte două ecuațiile cercurilor şi realizăm un sistem pe care îl rezolvăm.

$$\begin{cases} (x-x_{APi})^2+(y-y_{APi})^2=d_i^2 \\ (x-x_{APi+1})^2+(y-y_{APi+2})^2=d_{i+1}^2 \end{cases}$$

În urma rezolvării sistemului rezultă două soluții (x_{M1}, y_{M1}) şi (x_{M2}, y_{M2}). Dintre cele două soluții o alegem pe aceea care satisface condiția:

$$\left(\frac{x_M-x_i}{x_{i+1}-x_i}-\frac{y_M-y_i}{y_{i+1}-y_i}\right)\left(\frac{x_{i-1}-x_i}{x_{i+1}-x_i}-\frac{y_{i-1}-y_i}{y_{i+1}-y_i}\right)>0$$

Mulțimea punctelor astfel obținute definesc la rândul lor un nou poligon. Deci domeniu D îl vom aproxima prin acest nou poligon. Centrul de greutate al poligonului este dat de formulele:

$$x_M=\frac{(x_1+x_2+...+x_n)}{n}$$
$$y_M=\frac{(y_1+y_2+...+y_n)}{n}$$

Pentru a implementa această metodă trebuie satisfăcută o singură condiție şi anume, AP-urile trebuie să formeze un poligon convex. Programul care implementează algoritmul de localizare folosind metoda poligonului a fost scris tot în Visual C++ şi el este de fapt o extensie a celui folosit pentru calcularea puterii semnalului recepționat. În prealabil s-au făcut toate calibrările descrise subcapitolul 5.2.1, iar modelul ales pentru AP-uri a fost DI-624.

S-au efectuat mai multe tipuri de măsurători introducând următoarele tipuri de elemente variabile:

a) numărul de persoane din încăpere
b) gradul de mobilitate al dispozitivului care trebuie localizat

Pentru toate tipurile de măsurători s-au folosit patru AP-uri. S-au marcat anumite puncte pe podeaua încăperii unde au avut loc măsurătorile, puncte cu poziție cunoscută. AP-urile au fost plasate în colțurile încăperii, ca în figura 5-27. Dimensiunile încăperii sunt Lungime=14m şi Lățime=6m. Referința a fost aleasă în colțul din stânga jos.

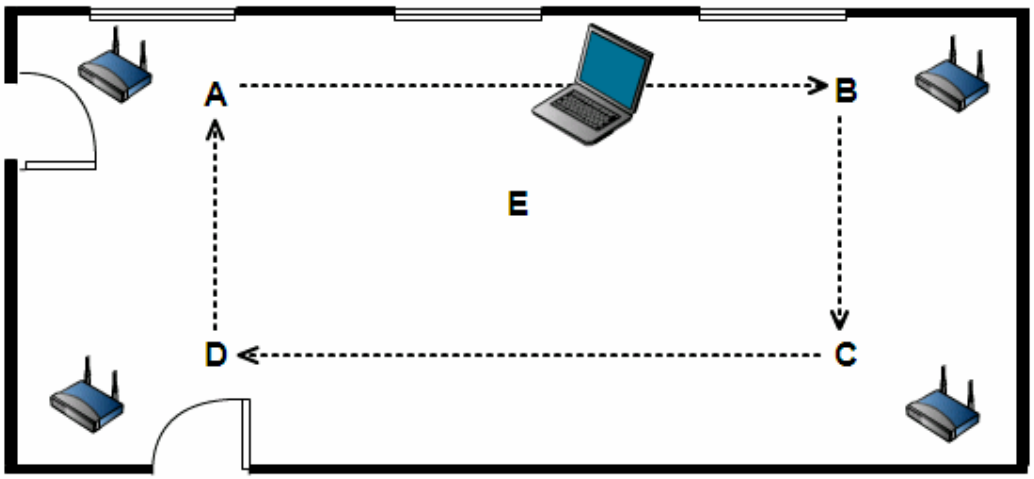


Figura 5-27 Schița realizării testelor pentru algoritmul de localizare

S-au făcut mai întâi măsurători statice. Laptop-ul a ocupat o poziție fixă într-unul din punctele marcate, mai întâi fără oameni în interiorul încăperii, apoi în încăpere au intrat 10 persoane care s-au deplasat în mod aleatoriu. S-a calculat apoi eroare de localizare, constând în diferența dintre valorile deduse prin măsurarea puterii şi valorile cunoscute pentru distanțe. S-au realizat mai multe serii de măsurători iar apoi s-a făcut media lor. Pentru primul caz, eroarea de poziționare a fost de 1,8 m, iar pentru al doilea caz, cu persoane în încăpere, eroarea a fost de 2,5 m.

Pentru măsurătorile dinamice, laptop-ul s-a deplasat cu viteza de aproximativ 2m/s pe traseul marcat în figura 5-27. Pentru cazul în care nu au fost persoane în încăpere, media erorilor a fost de 2,8 m, iar pentru cazul în care au fost oameni în încăpere, media erorilor a fost de 3,2m.

Concluzia este că eroarea de localizare creşte dacă dispozitivul mobil se află în mişcare. Pentru cazul static, erorile de 1,8 şi 2,6 metric sunt considerate acceptabile.

5.4 Concluzii

Problema localizării unui dispozitiv în interiorul unui spațiu închis este una destul de dificilă, datorită diverselor perturbații pe care semnalul le suferă în urma interferențelor şi reflexiilor. Aceste perturbații au ca efect obținerea unor valori oscilante atunci când se citeşte puterea semnalului recepționat. Una dintre primele măsuri prezentate în cadrul capitolului a fost de a găsi o metodă de corectare a acestor valori eronate.

S-au făcut seturi extinse de măsurători pentru a pune în evidență impactul pe care îl au diversele tipuri de AP-uri asupra puterii semnalului recepționat precum şi influența distanței asupra semnalului recepționat. S-a adaptat formula de propagare a semnalului radio în spațiul liber pentru a obține rezultate care să minimizeze efectul perturbațiilor asupra semnalului.

Un alt aspect vizat a fost punerea în evidență a modului de variație a semnalului atunci când dispozitivul mobil părăseşte încăperea şi a modului de variație atunci când în încăpere sunt prezente persoane.

Pentru localizarea propriu-zisă a dispozitivului mobil este elaborată o metodă denumită „metoda poligonului”, care se remarcă prin simplitate şi uşurința în imple-

mentare. Ea va fi folosită în cadrul framework-ului descris în capitolul 6, unde îşi va demonstra mai clar aplicabilitatea.

Se va vedea că soluția propusă în această lucrare este cea a unei arhitecturi de tip „cross layer” prin care aplicații care rulează pe nivele superioare ale stivei de protocoale să poate interveni în procese executate pe nivele inferioare ale stivei.

Pentru a se putea lua o decizie referitoare la care AP ar oferi o conexiune optimă într-un anumit context, se poate lua în calcul valoarea distanței dintre AP-urile candidate şi dispozitivul mobil aflat in discuție. Mai mult, se pot pune la punct politici de tip anticipativ. O situație de acest gen ar fi aceea în care framework-ul UFRM (descris în capitol 6), pe baza unui algoritm preemtiv să sesizeze intenția dispozitivului mobil de a părăsi încăperea şi să acționeze în scopul de a asigura resursele necesare pentru a oferi o nouă conexiune, în parametrii optimi, pentru dispozitivul mobil, cu alte cuvinte, de a pregăti un AP care să poată prelua dispozitivul mobil.

Acelaşi framework UFRM, pentru a oferi utilizatorului servicii particularizate pentru necesitățile lui, într-un sistem de tip *context-aware*, are nevoie de un mecanism de localizare în spațiul în care acesta îşi desfăşoară activitatea.

Țelul urmărit în cazul procedeului de localizare elaborat în această teză, a fost acela de a obține un raport favorabil între resursele alocate de sistem pentru a rula algoritmii de localizare şi eficiența acestor algoritmi.

Capitolul 6 Sistem de management al resurselor într-o rețea WLAN 802.11

6.1 Introducere

Când discutăm despre conexiuni wireless, în ecuație intervin doi factori importanți care afectează direct calitatea serviciilor de care va beneficia utilizatorul dispozitivului mobil. Aceşti parametrii sunt autonomia şi rata de transfer obținută prin conexiunea wireless. Cei doi parametrii nu sunt independenți şi vom vedea cum autonomia este direct legată de calitatea conexiunii radio.

Intuitiv privind lucrurile, oricine poate deduce care ar fi această legătura dintre calitatea conexiunii radio si modul în care ea afectează descărcarea acumulatorului dispozitivului mobil. Astfel, putem spune că o conexiune de bună calitatea favorizează o durată de viață mai mare pentru acumulator, pe când o conexiune de proastă calitate duce la o descărcare mai rapidă a acestuia. Acest lucru este general valabil pentru orice tip de conexiune wireless, indiferent de standardul de comunicație folosit.

Trecând la o analiză mai detaliată a fiecărui tip de conexiune în parte vom vedea că există diferențe între modalitatea de gestionare a unei conexiuni wireless pentru diverse standarde de comunicație. De exemplu, in cazul rețelelor GSM dispozitivul mobil îşi adaptează puterea de transmisie în funcție de distanța față de antena receptoare, aceasta fiind principalul motiv pentru care acumulatorul unui astfel de dispozitiv se descarcă mai repede dacă distanța față de antenă creşte sau dacă semnalul este atenuat.

În cazul dispozitivelor care fac parte din standardul 802.11, nu toate dispozitivele beneficiază de o funcție de reglare a puterii semnalului emis, iar la cele care au această facilitate, reglajul nu se face în mod dinamic ci doar static, în momentul setării parametrilor generali de funcționare. Aici, calitatea conexiunii influențează direct comportamentul protocoalelor din stiva TCP/IP. O conexiune de calitate slabă are ca efect retransmisii la nivel legătură de date, dar şi o „încetinire" a protocolului TCP, protocol care interpretează întârzierile în transmiterea pachetelor ca pe o congestie care are loc undeva în rețea, el nefiind proiectat să funcționeze într-o rețea unde probabilitățile de apariție a erorilor este foarte mare şi astfel o sesiune de comunicație va dura mai mult din cauza micşorării ratei de transfer. Dacă avem o conexiune radio de proastă calitate, aceasta va genera numeroase retransmisii la nivelul MAC implementat de standardul 802.11, retransmisii care vor duce la un consum suplimentar de energie. În felul acesta se manifestă în cazul rețelelor 802.11 legătura dintre calitatea conexiunii şi descărcarea acumulatorului.

O altă modalitate de conservare a energiei acumulatorului este de a încerca o „armonizare" a aplicațiilor care rulează pe dispozitivul mobil. Aceasta se poate realiza printr-un framework care să gestioneze toate activitățile dispozitivului mobil. În mediile te tip „*corporate*" şi nu numai, există un număr considerabil de dispozitive mobile care interacționează între ele. Aici problema optimizării resurselor este una de ansamblu, care vizează interacțiunea tuturor resurselor implicate direct sau indirect într-un astfel de sistem complex.

În acest capitol este descris un sistem care a fost conceput pentru a facilita optimizarea resurselor unor dispozitive, care interacționează folosind ca infrastructură o rețea de tip wireless 802.11. Pentru implementarea unui astfel de sistem s-a făcut o analiză din mai multe perspective:

- comportamentul protocoalelor implicate
- resursele care trebuie optimizate
- aplicaţiile care pot fi rulate pe dispozitivele mobile
- modalitatea de culegere a informaţiilor referitoare la contextul în care se desfăşoară o anumite activitate

Atunci când se discută de optimizarea resurselor într-un sistem din acesta complex, în care intervin un număr mare de dispozitive, e clar că atunci când se pune problema optimizărilor trebuie avute în vedere şi diverse politici preferenţiale care vor favoriza anumiţi utilizatori privilegiaţi, în defavoarea altora cu privilegii mai reduse. Astfel de situaţii sunt inevitabile şi ele sunt întâlnite în orice domeniu în care se impune managementul resurselor de orice natură ar fi ele.

Sistemul propus se construieşte în jurul conceptului de dispozitiv. Dispozitivele sunt caracterizate printr-o serie de proprietăţi şi pot fi împărţite în două mari categorii, din punctul de vedere al modului de interacţiune cu UFRM (Unified Framework for Resources Management). Aceste două categorii sunt: **dispozitive mobile** şi **dispozitive pasive**.

Denumirea de **dispozitive mobile** sugerează în primul rând faptul că acestea îşi pot modifica oricând locaţia. Pe de altă parte ele sunt dispozitive capabile să transmită informaţiile de tip context în mod direct, nemijlocit, către UFRM, ele fiind capabile să ruleze o aplicaţii care să interacţioneze cu framework-ul nostru.

În cea de-a doua categorie, a **dispozitivelor pasive,** intră acele dispozitive care au o poziţie fixă, bine determinată şi care nu se modifică decât foarte rar. De exemplu, dacă este nevoie de a se muta în altă locaţie un anumit AP. Pe de altă parte, ele nu sunt capabile să furnizeze direct informaţii referitoare la context, acestea fiind culese în mod indirect fie prin intermediul altor dispozitive active cu care cele pasive se află în interacţiune. O altă posibilitate ar fii aceea ca informaţiile referitoare la context să fie actualizate periodic, prin intervenţia unor operatori umani.

6.2 Sisteme de calcul de tip „context-aware”

În sistemele de tip *context-aware* este aplicat un concept care exprimă capacitatea acestor sisteme de a descoperi şi de a ţine cont de diverşi factori sau parametrii care creează un anumit context în care o aplicaţie sau sistem de calcul funcţionează. Odată cu apariţia comunicaţiilor de tip wireless, folosirea dispozitivelor mobile a început sa penetreze tot mai accentuat piaţa. Spectrul aplicaţiilor care rulează pe acest tip de dispozitive a devenit tot mai larg, precum şi serviciile oferite utilizatorilor au început sa fie tot mai variata. Astfel că, mai mult ca niciodată, pentru o utilizare cât mai eficientă, dar şi pentru a oferi o varietate cât mai largă de servicii, a apărut ca o necesitate naturală implementarea acestui concept numit *context-aware*.

La început dispozitivele mobile (PDA-uri, laptop-uri, telefoane mobile) nu beneficiau de performanţe deosebite în ceea ce priveşte puterea de calcul şi autonomia. Îmbunătăţirea acestor doi parametrii a dus la un avânt fără precedent în ceea ce priveşte diversitatea aplicaţiilor şi serviciilor oferite de acestea.

În primul rând ar trebui stabilită o definiţie pentru termenul de „context”, care să faciliteze înţelegerea clasificărilor care vor fii făcute in cele ce urmează. Astfel, prin „context” se înţelege un concurs de împrejurări în care se produce un fenomen şi modul de intercondiţionare dintre elementele (entităţile) care interacţionează. Defini-

ţia este destul de largă, dar ea va fi particularizată pentru cazul reţelelor de tip wireless. Prin reţea de tip wireless înţelegem totalitatea dispozitivelor wireless care comunică direct sau indirect între ele, precum şi a aplicaţiilor care rulează pe astfel de dispozitive.

6.2.1 Definiţii ale conceptului de *context*

În încercarea de a defini conceptul de „context", unii autori au dat definiţii pornind de la exemplificări ale contextului:

- context computaţional: starea resurselor hardware şi software
- context utilizator: profilul utilizatorului, locaţia, interacţiuni sociale, etc.
- context fizic: temperatura, luminozitate, zgomot, etc.

În [CK00] este dată o definiţie pentru context: „contextul este mulţimea stărilor şi parametrilor mediului ambiant în care rulează o anumită aplicaţie şi care influenţează comportamentul acelei aplicaţii sau care determină apariţia unor evenimente in cadrul aplicaţiei, evenimente care pot avea un anumit interes pentru utilizator".

O alta definiţie a „contextului" ar fi următoarea: „ orice informaţie care poate fi folosită pentru a caracteriza situaţia unei entităţi (persoana, obiect, loc, aplicaţie software) şi care are relevanţă in interacţiunea dintre un utilizator si o aplicaţie".

În cele mai multe cazuri, sistemele de calcul deţin informaţii foarte sărace despre realitatea înconjurătoare. Singura modalitate de a le face „conştiente" de context este prin acţiunea utilizatorului care introduce în sistem informaţiile respective, o manieră deloc convenabilă, care necesită intervenţia directă a utilizatorului, fiind o acţiune consumatoare de timp şi mai mult, chiar susceptibilă de a genera erori, intrând cum am spus în ecuaţie şi factorul uman. În mod clasic, un sistem de calcul obişnuit deţine foarte puţine interfeţe cu exteriorul care să permită colectarea de informaţii, iar acestea necesită, cum spuneam, acţiunea explicită a utilizatorului pentru a face posibilă colectarea acestor informaţii. Scopul unui sistem de tip *context-aware* este acele de a elimina acest neajuns şi de a permite şi colectarea implicită a informaţiilor de tip context.

Informaţiile referitoare la context permit adaptarea la diverse cerinţe, discernerea între ce este important şi ce nu, la un moment dat. Ele ajută la organizarea multitudinii de informaţii, influenţează modul în care percepem informaţiile.

6.2.2 Principii de implementare a sistemelor de tip *context-aware*

Aplicabilitatea unor servicii de tip *context-aware* este variată, acoperind practic toate domeniile de activitate: educaţie, producţie, sănătate, comerţ, turism, etc. [SCP+05, DX08]. Simplificând lucrurile, putem spune că scopul acestor sisteme de tip *context-aware* este de a oferi resursa potrivită la momentul potrivit.

Modul de implementare a unui sistem de tip *context-aware* poate fi gândit din mai multe perspective [CK00]:

.

A. După tipul de colectarea informaţiilor care alcătuiesc contextul

- arhitectura care permite accesul direct la senzori
- arhitectura cu nivel intermediar

B. Din punct de vedere al modalităţii de colectare a informaţiilor

- colectare explicită sau manuală
- colectare implicită sau automată

C. După modul de gestionare a informaţiilor colectate

- structura distribuită
- structură centralizată

D. După modul de utilizare a informaţiilor

- sistem activ (aplicaţiile se adaptează dinamic în funcţie de anumite condiţii oferite de context)
- sistem pasiv (cere intervenţia utilizatorului pentru modificarea comportamentului odată cu actualizarea informaţiilor de context)

E. Din punct de vedere al tipului de informaţii colectate

- informaţii primare (direct de la senzori)
- informaţii procesate (informaţiile primare au fost prelucrate intr-o anumită formă şi apoi transmise mai departe)

F. Din punct de vedere al naturii informaţiilor

- informaţii de mediu (temperatură, luminozitate, zgomot, etc.)
- informaţii spaţiale şi temporale (coordonate, viteza, altitudine, data, ora, etc.)
- informaţii computaţionale (informaţii provenite de la diverse programe)
- informaţii sociale (care se referă la interacţiunea utilizatorului cu alţi semeni)

Prin senzor trebuie sa se înţeleagă nu doar echipamentele dedicate care intră in această categorie, ci orice sursă care poate furniza informaţii referitoare la context.

Scopul implementării unui sistem de tip *context-aware* este pe de o parte de a oferi servicii îmbunătăţite utilizatorului sau unui grup de utilizatori dar şi de a eficientiza serviciile deja existente şi de a optimiza funcţionarea în ansamblu a unui sistem de calcul.

În [BDR07] este prezentată evoluţia sistemelor de tip *context-aware*, începând cu prima soluţie de acest tip intitulată *Active Badge Location System*, continuând cu un sistem bazat pe tehnologia undelor infraroşii, capabil să redirecţioneze un apel telefonic către aparatul cel mai apropiat de utilizator, urmând apoi câteva soluţii pentru implementarea unor sisteme de ghidare a vizitatorilor în diverse instituţii. În cadrul acestor soluţii principala informaţie de tip context se referea la poziţie.

In [DAS01] se propune o organizare a informaţiilor contextuale în funcţie de natura lor. Autorii observă că în cazul scenariului prezentat de ei, elementele care

interacţionează se încadrează în 3 mari grupe: zona(locaţie), persoane (indivizi sau grupuri) şi entităţi (obiecte fizice sau entităţi software). Acestea pot fi complet caracterizate prin patru clase de proprietăţi: identitatea, coordonatele spaţiale, starea şi timpul.

În [MNV08] este introdus un nou concept şi anume *context-aware retrieval* care se referă la capacitatea unui sistem de tip *context-aware* de a gestiona un set de elemente în funcţie de care va fi luată o decizie: o colecţie de documente, un set de cerinţe din partea utilizatorului şi contextul în care se află utilizatorul. Pe baza acestor elemente utilizatorului ii va fi furnizat un anumit document. Autorii din [MSG+07] propun un sistem numit SATO (Service-aware Adaptive Transport Overlay) responsabil cu distribuirea informaţiilor într-o reţea în funcţie de context.

Între colecţia de informaţii reprezentând contextul şi beneficiarii acelor informaţii este util de multe ori să se introducă un nivel intermediar care va face o procesare primară a acelor informaţii. Un sistem de acest fel este descris în [PKK06], iar în [CSS05] este propusă o componentă de tip *middleware* pentru un sistem de control al unei case inteligente. Acelaşi principiu este aplicat de autorii din [SWS+04] pentru în cazul unei reţele de tip ad-hoc. În [ROP+05] se prezintă o platformă pentru dezvoltarea aplicaţiilor dependente de context pentru smartphone-uri. Autorii subliniază discrepanţa dintre facilităţile oferite de sistemele de operare care rulează pe astfel de dispozitive şi nevoile dezvoltatorilor de aplicaţii.

6.3 Descrierea arhitecturii UFRM (Unified Framework for Resources Management)

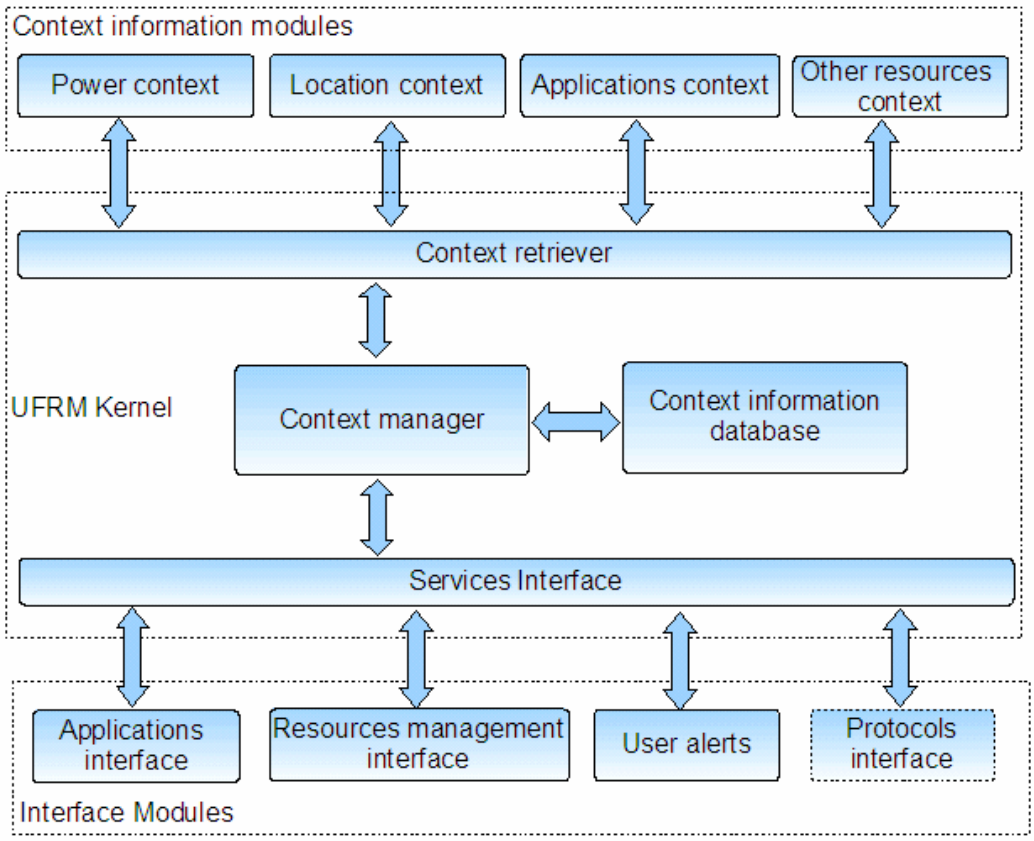


Figura 6-1 Arhitectura UFRM (Unified Framework for Resource Management)

Arhitectura este centrată în jurul unei baze de date (**Context Information Database**) în care sunt centralizate toate informaţiile referitoare la context. Culegerea acestor informaţii este realizată de către o serie de module (**Context Information Modules**) specializate pe culegerea unui anumit tip de informaţie. De exemplu modulul **Location Context** este responsabil cu obţinerea informaţiilor legate de locaţia diverselor dispozitive, iar **Power Context** oferă informaţii referitoare resursele de energie si consumul dispozitivelor mobile. Aceste module rulează în background, ca şi aplicaţii, la nivelul **dispozitivelor mobile** (notebook-uri, PDA-uri, etc.). Pentru **dispozitivele pasive** (access point-uri, imprimante, etc.) culegerea informaţiilor se poate face indirect, în urma interacţiunilor realizate între dispozitivele active şi cele pasive. Un exemplu de acest fel este aflarea numărului de clienţi pe care îi are la un moment dat un anumit AP. Un AP este un dispozitiv pasiv, care nu este capabil să furnizeze direct astfel de informaţii. Aceasta se realizează consultând o tabelă a conexiunilor active, tabelă asociată de către UFRM fiecărui AP. Actualizarea acestei tabele este făcută de UFRM pe baza informaţiilor colectate de la dispozitivele mobile, care anunţă prin intermediul modulului **Location context,** identificatorul AP-ul la care sunt asociate în acel moment.

6.3.1 Modelarea dispozitivelor de către UFRM

Dispozitivele mobile se împart în doua mari categorii: notebook-uri, care au putere de procesare mai mare şi PDA-urile care sunt mai limitate din punct de vedere al resurselor de calcul şi deci al tipurilor de aplicații pe care le pot rula. Indiferent cu care categorie avem de a face, structurile de date folosite pentru managementul unor astfel de dispozitive sunt redate în figura 6-2.

A) Modelarea dispozitivelor mobile

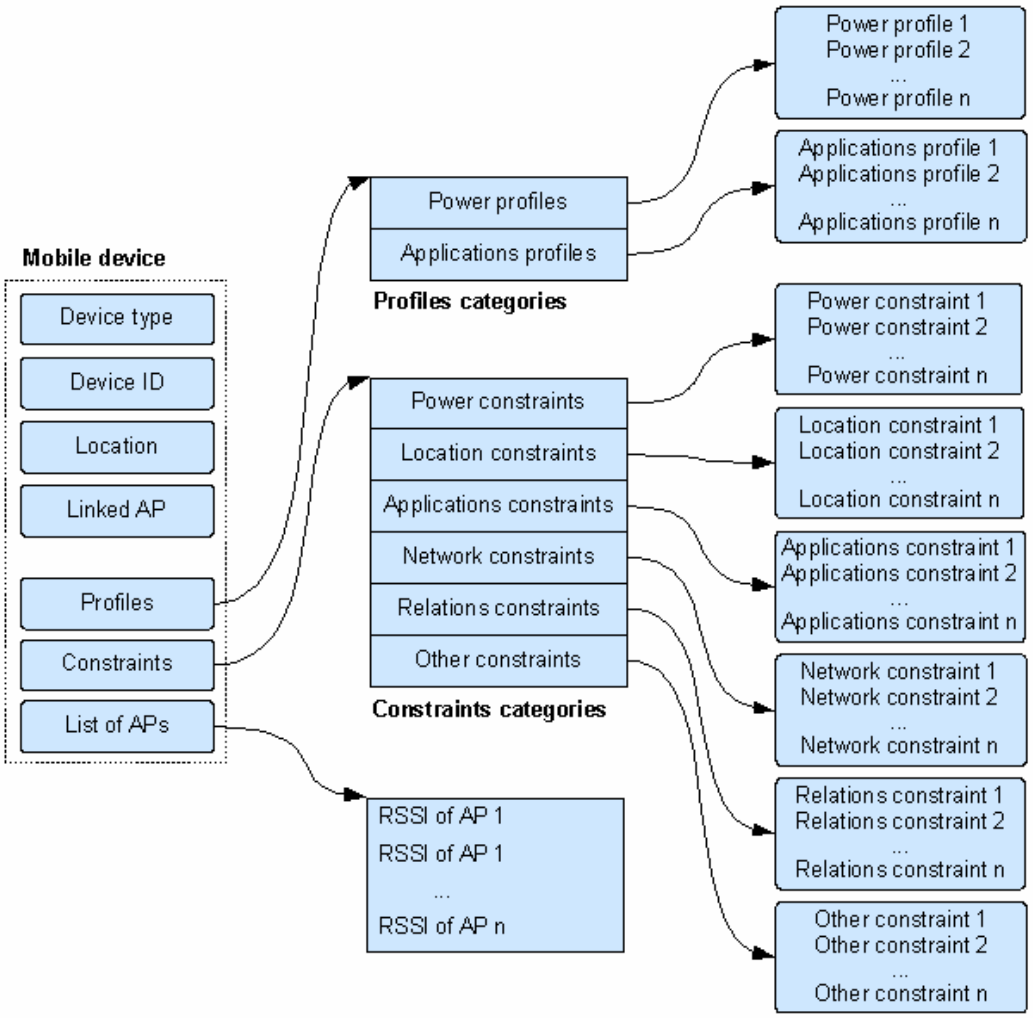


Figura 6-2 Structurile de date asociate unui dispozitiv mobil

Fiecare dispozitiv mobil este caracterizat prin: tip (Device type), identificator (Device ID), locație (Location), identificatorul AP-ului la care este asociat (Linked AP).

Device type: acest câmp specifică dacă dispozitivul mobil este notebook sau PDA.

Device ID: Dispozitivele pot fi identificate în mod unic prin adresa MAC, asociată plăcii de rețea, dar există situații când este mai convenabil să se folosească pentru identificare adresa IP. Oricum, ambele adrese sunt salvate în Device ID.

Location: este definită prin identificatorul zonei în care se află dispozitivul şi dacă este cazul prin coordonatele în interiorul acelei zone. Prin zonă se înțelege un anumit spațiu fizic. De obicei o zonă coincide cu spațiul delimitat de o anumită încăpere.

Linked AP: în acest câmp se specifică identificatorul (BSSID-ul) AP-ului la care este conectat respectivul dispozitiv mobil.

Profiles: defineşte o legătură către o tabelă care conține câte o intrare pentru fiecare categorie de profiluri. Cu ajutorul acestor profiluri se definesc o sumă de proprietăți de natură mai complexă care caracterizează dispozitivul la un moment dat şi care pot genera anumite tipuri de constrângeri. Există două mari categorii de profiluri: Power profiles şi Applications profiles.

Power profiles: pe baza informațiilor culese cu ajutorul modului **Power context** dispozitivele vor putea fi caracterizate printr-unul sau mai multe profiluri energetice. Fiecare profil va urmări evoluția unei anumite resurse de la nivelul dispozitivului monitorizat(acumulatorul, placa de rețea, microprocesorul, etc.). Pe baza profilului se va putea aproxima,de exemplu, care va fi durata de viață a acumulatorului sau care este consumul de energie pentru 1KB de date trimis, etc.

Applications profiles: aici sunt înregistrate numărul şi tipul aplicațiilor care rulează pe dispozitivul mobil.

Constraints: defineşte o legătură către o tabelă care are câte o intrare pentru fiecare categorie de constrângi. La rândul lor, aceste intrări sunt legături către liste (câte una pentru fiecare categorie de constrângeri) care conțin enumerări ale diverselor tipuri de constrângeri proprii fiecărei categorii.

Denumirea de constrângere trebuie privită într-un mod mai general. Sub această denumire vor fi surprinse şi posibile privilegii de care se pot bucura anumiți utilizatori. De exemplu, aici sunt înregistrate informații referitoare la nivelul de acces al utilizatorului la resursele rețelei. Se poate stabili o anumită rată de transfer care se încearcă să-i fie garantată unui anumit utilizator al rețelei, poate fi stabilită o listă cu aplicații care să nu aibă voie să fie rulate pe un anumit dispozitiv, sau poate fi blocat accesul la Internet, etc.

În cadrul UFRM sunt definite şase categorii de constrângeri:

- power constraints
- location constraints
- applications constraints
- network constraints
- relations constraints
- other constraints

Fiecare categorie la rândul ei cuprinde mai multe tipuri de constrângeri. Constrângerile pot fi predefinite sau generate de către modulul **Context manager**.

O categorie aparte de constrângeri o reprezintă **relațiile**. Prin **relație** se înțelege modul de interacționare cu celelalte dispozitive. Cu ajutorul **relațiilor** se pot construi **grupuri de interes**, adică dispozitive care trebuie să conlucreze pentru atingerea unui deziderat comun.

Relația devine activă în funcție de anumite condiții care se stabilesc asupra unor proprietăți. De exemplu dacă este pornită o anumită aplicație, atunci devine activă relația dintre dispozitivul care a pornit aplicația şi un alt dispozitiv.

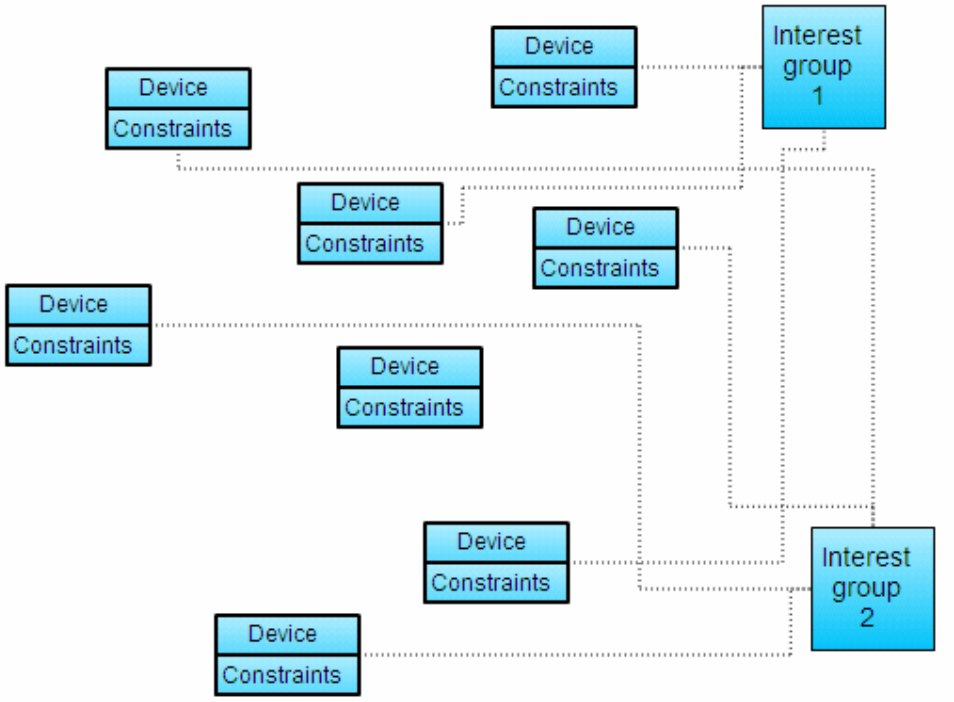


Figura 6-3 Modul de formare al grupurilor de interes

Pornind de la aceste constrângeri se va face gestiunea resurselor de către UFRM. Dacă pentru o anumită resursă nu există specificate anumite constrângeri, atunci se va ține cont de politicile generale care sunt specificate folosindu-se modulul **Resource Management Interface.**

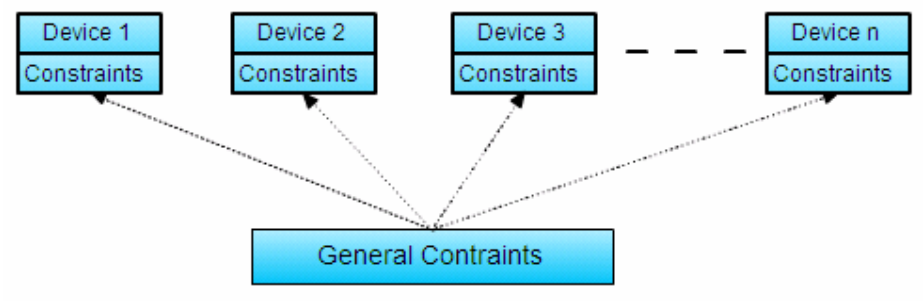


Figura 6-4 Funcționarea sistemului de constrângeri

Să presupunem următorul scenariu: avem un utilizator care are garantată o anumită rată de transfer. Dacă el foloseşte o aplicație de videochat este important ca şi interlocutorului său să-i fie garantată o rată de transfer similară, care să asigure o convorbire în parametrii optimi. Pe baza **relațiilor** definite static prin **Resource Management Interface** sau dinamic, de către **Context manager**, se va crea un **grup de interes** şi astfel va avea şi interlocutorul garantată o anumită rata de transfer. Asigurarea unei anumite rate de transfer se poate face prin migrarea unei părți din clienții care se află asociați la acelaşi AP cu utilizatorul privilegiat, spre alte AP-uri din vecinătate.

O altă situație ipotetică ar fi următoarea: un utilizator privilegiat are pornită o aplicație critică (generarea unor rapoarte). Dacă nivelul acumulatorului scade sub un prag de alarmă, conform cu profilul energetic asociat acelui utilizator, atunci prin modulul **User Alerts**, utilizatorul va fi avertizat despre situația apărută şi în acelaşi timp i se vor garanta conexiuni de calitate, pentru a-şi putea încheia cât mai repede activitățile aflate în desfăşurare.

List of AP: defineşte o referință către o structură care reprezintă o listă cu toate AP-urile accesibile de către acel dispozitiv mobil. Pe lângă BSSID-ul fiecărui AP, în listă mai sunt trecute şi ultimele valori citite pentru RSSI (Received Siganl Strenght Indicator).

B) Modelarea dispozitivelor pasive

Cealaltă categorie de dispozitive prezente în rețea sunt cele pasive. Aici am spus că sunt incluse AP-uri, imprimantele (dacă sunt prevăzute cu interfață de rețea), camere web, etc. Fiecare din ele vor avea asociate structuri de date specifice acelui dispozitiv. Vom exemplifica cu structurile de date folosite pentru gestionarea unui AP.

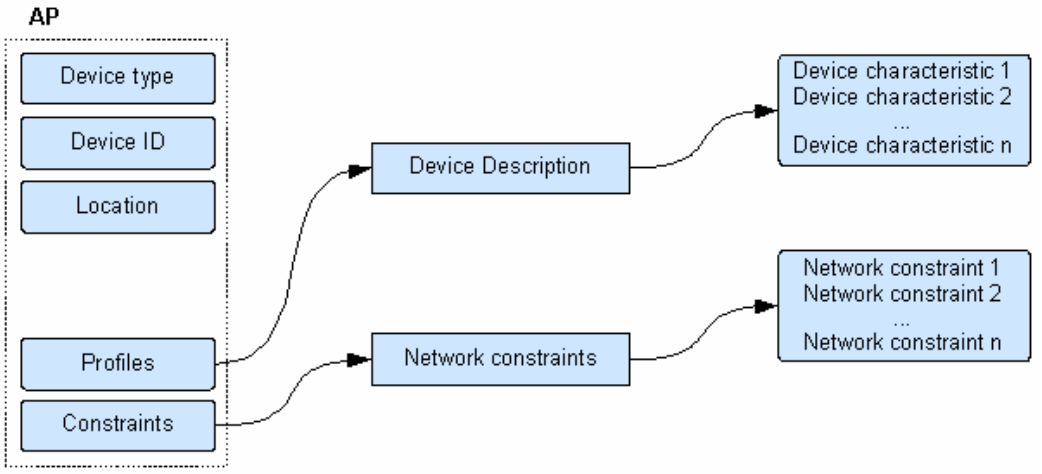


Figura 6-5 Structurile de date asociate unui AP

Se observă că din structura prezentă la dispozitivele mobile au dispărut o parte din câmpuri. De exemplu, **Linked AP** şi **List of APs**, nu îşi mai au sensul în această situaţie. O altă simplificare se constată în ceea ce priveşte tabelele care conţin categoriile de profiluri şi de constrângeri. Pentru **Profiles**, unica intrare care mai există s-a transformat într-o structură descriptivă de unde pot fi aflate caracteristicile tehnice ale AP-ului. Pentru **Constraints**, singura categorie păstrată se referă la **Network constraints**. Aici se pot specifica, de exemplu, numărul maxim de clienţi pe care dispozitivul poate să îi accepte sau tipul de trafic pe care îl poate suporta.

6.3.2 Modulul „Power context”

Acest modul asigură culegerea de informaţii referitoare la resursele de energie disponibile pe un anumit dispozitiv mobil. Acest modul a fost gândit să poată gestiona atât informaţii sub formă primară, de exemplu nivelul încărcării bateriei, dar şi informaţii deja procesate de un *power-aware framework* (PAF) implementat la nivelul acelui dispozitiv mobil. Acest PAF funcţionează independent de UFRM şi el a fost dezvoltat ca făcând parte iniţial dintr-o altă direcţie de cercetare. Ulterior s-a dovedit că informaţiile furnizate de acest *power-aware framework* (PAF) pot fi foarte utile atunci când se doreşte o analiză mai rafinată a profilului energetic al unui anumit dispozitiv mobil [MTF09, FMS+09]. PAF a fost conceput pentru optimizarea consumului unui sistem de calcul.

Structura PAF este următoarea:

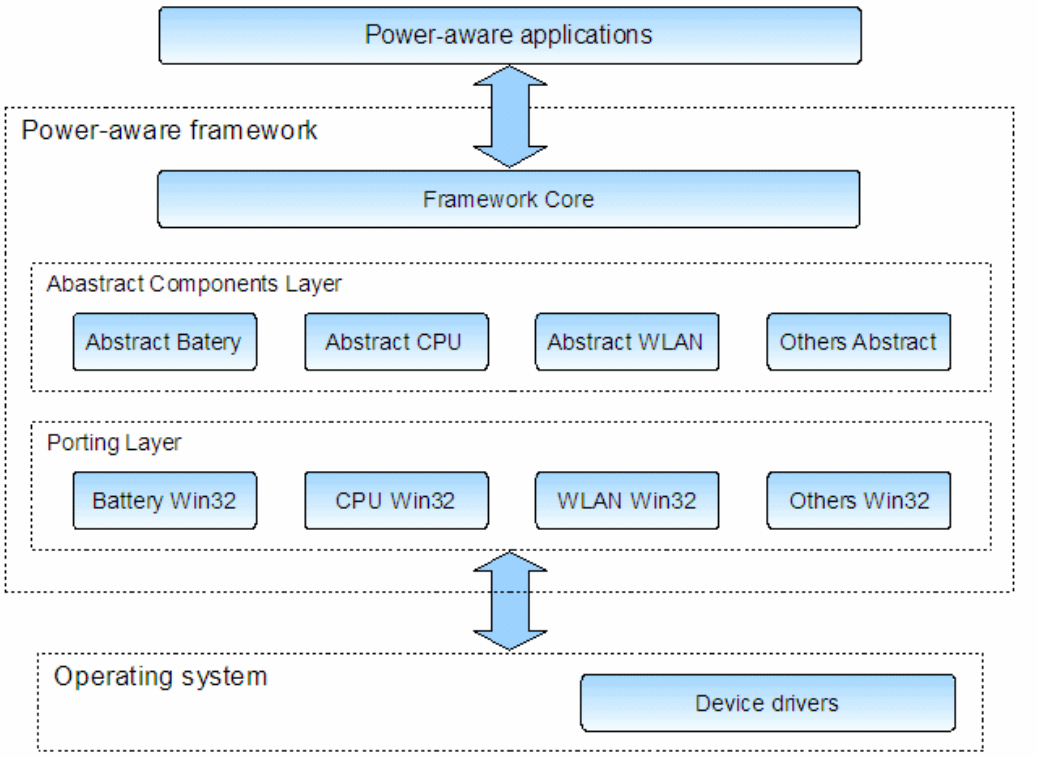


Figura 6-6 Arhitectura PAF(Power-Aware Framework)

PAF are o structură modulară, având mai multe nivele de abstractizare. La nivelul cel mai de jos se află Porting Layer, care va accesa direct serviciile puse la dispoziţie de către driverele sistemului de operare pentru a obţine informaţii despre diversele resurse care vor fi monitorizate [MTM+09, MTF+08]. Aceste resurse pot fi: acumulatorul, procesorul, placa de reţea wireless, etc. PAF va monitoriza prin intermediul driverelor consumul diverselor componente, încărcarea procesorului, gradul de utilizare al interfeţei wireless, va putea face unele predicţii în ceea ce priveşte evoluţia consumului, apoi va transmite aceste informaţii diverselor aplicaţii care vor fi interesate de aceste informaţii [MTFG08, MTFM08]. Aceasta este şi modalitatea prin care UFRM va beneficia de informaţiile de context referitoare la consum.

Ideea de bază din spatele PAF este aceea de profil energetic [MTFV08, MTF07]. Aceste profiluri sunt dispuse pe un nivel superior de abstractizare numit Abstract Components Layer. Elementele de pe acest nivel sunt independente de sistemul de operare folosit, ele putând fi portate şi pe alte platforme, rescriind doar componentele definite în Porting Layer, interfaţa dintre Porting Layer şi Abstract Component Layer rămânând nemodificată. Informaţiile culese de Abstract Component Layer sunt oferite apoi prin Framework Core aplicaţiilor care le solicită.

Importanţa unui astfel de *power-aware framework* a fost pusă în evidenţă printr-o serie de măsurători care au avut ca scop observarea consumului diverselor componente ale calculatorului [SFK+09, MTF08]. Metoda a constat în monitorizarea acumulatorului şi înregistrarea curbei de descărcare în diferite situaţii. Monitorizarea s-a făcut pentru un desktop PC, pentru un laptop si pentru un pocket PC. Au fost făcute mai multe serii de măsurători în diferite circumstanţe de utilizare (ex. număr de aplicaţii rulate simultan, comunicaţie wireless intensă, etc) şi apoi s-a făcut o medie a valorilor generate. Rezultatele testelor sunt rezumate în figura 6-7.

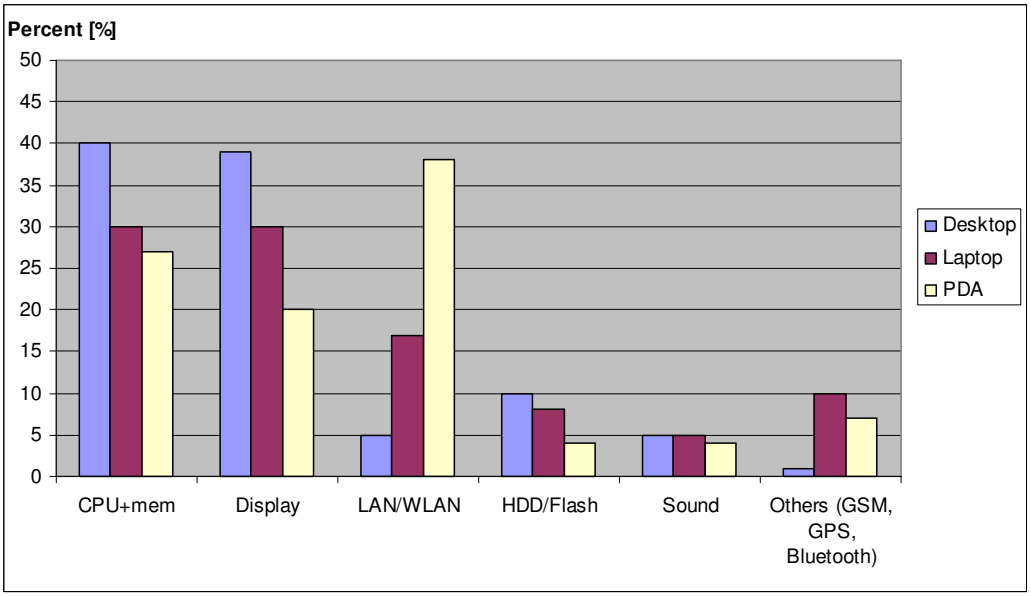


Figura 6-7 Distribuția consumului pentru diferite componente ale unui sistem

O importanță deosebită in cadrul PAF este acordată profilului energetic asociat plăcii de rețea wireless. Interacțiunea dintre UFRM şi PAF este redată în figura 6-8. Principalul indicator care este luat în calcul atunci când se încearcă realizarea unui profil energetic pentru placa de rețea wireless este cantitatea traficului total realizat prin această interfață. Această informație corelată cu un istoric al consumului şi cu celelalte profiluri energetice ne dă cu aproximație cantitatea de energie consumată pentru a transmite 1KB de informație, care va reprezenta o metrică pentru UFRM.

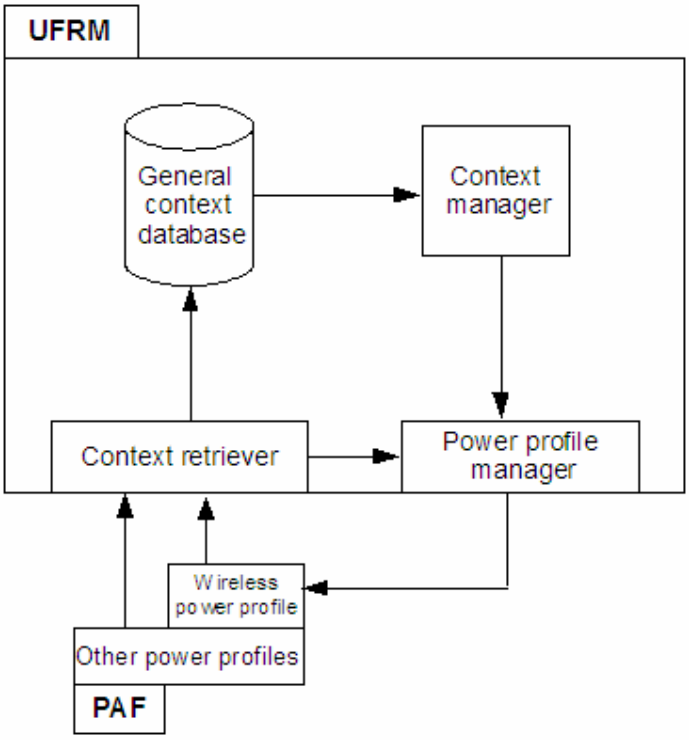


Figura 6-8 Interacţiunea dintre UFRM şi PAF

PAF joacă un rol important în cadrul UFRM, deoarece cu ajutorul lui sunt construite profiluri energetice pentru dispozitivele monitorizate [MTF09]. Pe baza profilurilor se poate monitoriza aportul la consumul de energie al unei anumite componente şi pot fi făcute predicţii ale consumului, lucru foarte important în luarea unor decizii.

6.3.3 Modulul „Location context”

Cu ajutorul acestui modul, UFRM oferă următorul set de servicii:

- localizarea cu diverse precizii a unui dispozitiv mobil în interiorul unui anumit spaţiu (clădire). Localizarea se va putea face la nivel de zone sau de coordonate în interiorul unei anumite zone.
- păstrarea unui istoric al deplasărilor în acel spaţiu.
- gestionarea resurselor aflate în evidenţa UFRM ţinând cont de poziţia şi deplasarea beneficiarilor acelor resurse.

Modul de localizare realizează determinarea poziţionării unui dispozitiv mobil într-un spaţiu în care se află instalate mai multe access point-uri. Pentru determinarea poziţiei, întreg spaţiul în care se poate deplasa dispozitivul mobil va fi împărţit în zone. În cele mai multe cazuri, o zonă va corespunde unei singure încăperi. Dacă aceea încăpere are dimensiuni foarte mari, atunci ea va fi împărţită în mai multe zone. Tehnica de localizare folosită este metoda trilateraţiei descrisă în capitolul 6.

Fiecărei zone îi este asociat cel puţin un *access point*. Acest *access point* asigură determinarea prezenţei unui dispozitiv în acea zonă, dar nu asigură localizarea cu precizie în interiorul acelei zone. Dacă acest lucru este necesar, atunci zonei respective în vor fi asociate cel puţin trei access point-uri. Această asociere presupune

prezența fizică a *access point*-urilor în acea zonă. Cu o astfel de configurație (figura 6-10) dispozitivul mobil poate să-şi determine poziția folosindu-se de citirea puterii semnalului (RSSI) emis de *access point*-urile din zona în care se află dispozitivul mobil. Pentru a simplifica exprimarea, atunci când ne vom referi la un access point, vom spune doar AP (Access Point), iar referirea la dispozitivul mobil se va face prin acronimul MD (Mobile Device).

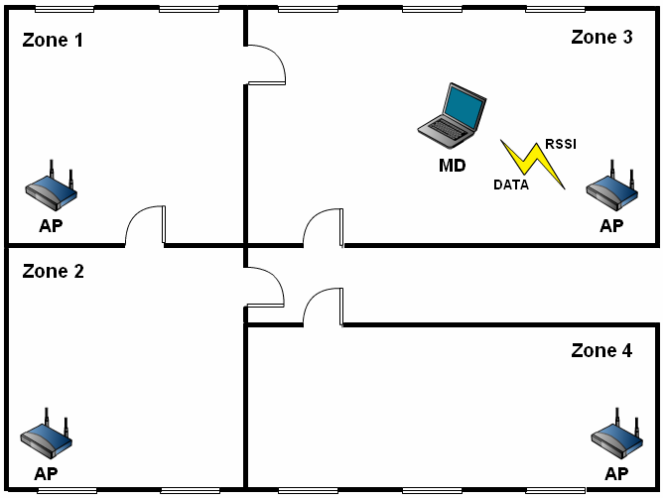


Figura 6-9 Localizarea doar la nivel de zonă

În configurație prezentată în figura 6-9, MD este capabil să identifice doar zona în care se află, nu şi coordonatele exacte în interiorul acelei zone. Pentru a-şi putea determina poziția exactă ar fi obligatoriu ca în zona respectivă să fie prezente cel puțin trei AP-uri, ca in figura 6-10, pentru a putea fi aplicat algoritmul de localizare prezentat în capitolul 5. Comunicația de date este posibilă cu oricare dintre cele trei AP-uri.

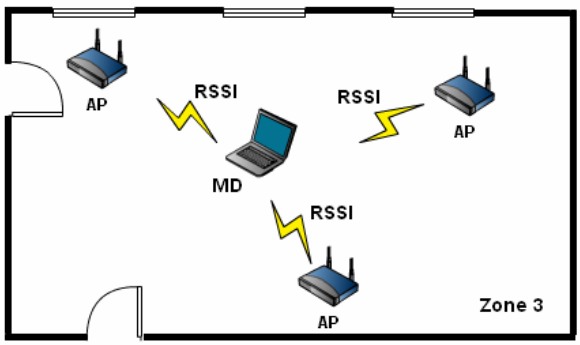


Figura 6-10 Localizarea la nivel de coordonate în interiorul unei zone

Până acum am luat în discuție varianta aceasta în care un dispozitiv mobil îşi determină poziția pe baza citirii puterii semnalului (RSSI) emis de către AP-urile din zona în care el operează. Pentru a putea realiza acest lucru este necesar ca dispozitivul care face scanarea pentru a determina prezența AP-urilor să îşi întrerupă pentru o scurtă durată de timp asocierea cu AP-ul la care este conectat. Acest lucru nu este de dorit în cazul unor anumite tipuri de aplicații.

O soluție de compromis ar fi dotarea MD-ului cu doua interfețe wireless, una pentru comunicația propriu zisă de date şi alta folosită doar pentru citirea puterii semnalului (RSSI) recepționat de la AP-uri.

Aceasta problema va avea în viitor o rezolvare foarte elegantă odată cu răspândirea pe piață a sistemelor de operare Windows 7. Ele vor pune la dispoziția utilizatorilor un API sub numele de Native Wifi, care va oferi funcții pentru gestionarea interfețelor WLAN. O facilitate nouă introdusă odată cu acest API este **Wireless Hosted Network**, care suportă două mecanisme importante:

- virtualizarea unei interfețe wireless, astfel că unei interfețe fizice să i se poată asocia mai multe interfețe logice.
- implementarea unui AP software, folosind o interfață wireless virtualizată.

Aceste două funcționalități oferă posibilitatea ca unei singure interfețe wireless fizice să i se poată asocia atât funcționalitățile unei interfețe wireless normale, cât şi funcționalitățile unui AP. Aceasta însemnă că maşina pe care este instalată acea interfață se va putea comporta ca un client wireless, asociindu-se la diverse AP-uri, dar va putea la rândul ei să se comporte ca un AP care va deservi la rândul lui diverşi clienți. Pe noi ne interesează foarte mult acest lucru, pentru că astfel vom putea să realizăm următorul gen de configurație.

Presupunem că avem un dispozitiv mobil (notebook) care dispune de o interfață wireless care este configurată să funcționeze atât ca şi client cât şi ca AP [FMS09]. Fiecare zonă din cadrul spațiului în care este implementat UFRM dispune în continuare de câte un AP arondat acelei zone, dar în plus pentru fiecare zonă vom avea distribuiți o serie de „senzori" wireless, de fapt nişte PDA-uri care să fie dotate cu o interfață wireless. Rolul acestor senzori este de a monitoriza în permanență o anumită zonă. Această monitorizare presupune executarea operațiilor de scanare pentru depistarea prezenței unor AP-uri. De fapt, căutarea va urmări evoluția în spațiu a dispozitivelor mobile la nivelul cărora a fost implementat un AP virtual. Dacă nu ar fi implementate aceste AP-uri virtuale, atunci acei „senzori" ar fi incapabili să sesizeze prezențe acelui dispozitiv mobil, pentru că după cum s-a arătat în capitolul 4, scanarea este posibilă doar pentru depistarea AP-urilor nu şi a clienților wireless. Presupunem situația din figura 6-11, unde zonele 3 şi 5 au fost prevăzute cu astfel de senzori.

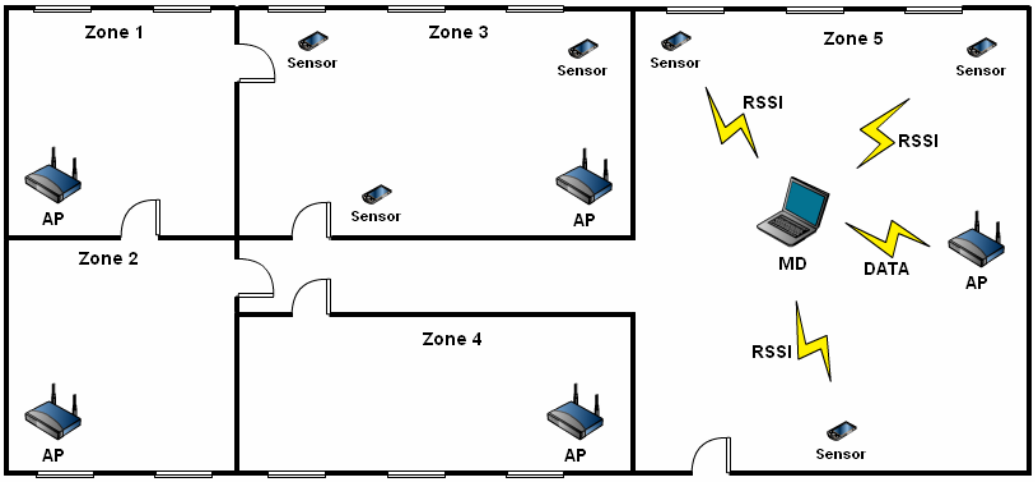


Figura 6-11 Localizarea folosind PDA-uri cu rol de "senzori"

Un alt avantaj al acestui tip de arhitectură este posibilitatea de a obţine o precizia mai mare în localizarea dispozitivului mobil. În varianta clasică, dispozitivul mobil îşi determină poziţia pe baza valorii puterilor semnalelor emise de către AP-uri. S-a văzut în capitolul 5 că puterea semnalului nu este staţionară ci au loc oscilaţii care sunt eliminate prin folosirea unui filtru software. Folosind senzorii introduşi în arhitectura noastră, lucrurile vor decurge în felul următor: fiecare senzor va obţine o valoare pentru RSSI. Presupunând ca avem trei senzori alocaţi zonei 5. În mod normal valorile pentru RSSI ar trebui să fie identice pentru toti trei senzorii, dar fiindcă citirile se fac asincron şi puterea de emisie a interfeţei wireless de pe dispozitivul mobil variază, atunci aceste valori vor fi diferite, conducând în cele din urmă la erorii de localizare. Dacă, în schimb, senzorii ar citi puterea semnalului emis de MD în acelaşi timp ele ar obţine aceiaşi valoare pentru RSSI. Astfel că se impune introducerea unei mecanism de sincronizare care să facă ca scanările realizate de senzorii dintr-o anumită zonă să fie simultane.

O altă problemă care trebuie soluţionată este modul de reprezentare a hărţii clădirii unde va fi implementat un sistem UFRM. Un prim pas a fost făcut şi am arătat că întreg spaţiul va fi împărţit în zone. O zonă, de obicei, corespunde unei încăperi de dimensiuni obişnuite. Dacă acea încăpere este foarte mare ea va fi împărţită în două sau mai multe zone. Fiecărei zone i se asociază un AP. Dacă se doreşte o localizare cu precizie în interiorul unei zone, atunci în interiorul acelei zone vor fi amplasate fie cei puţin trei AP-uri fie cel puţin trei senzori. Pentru framework-ul nostru este importantă stabilirea vecinătăţilor, adică să fie bine definit cu cine se învecinează fiecare zonă în parte. Aceste vecinătăţi sunt necesare pentru a stabili modul de atenuare a semnalului pe măsură ce ne depărtăm de un anumit AP. Am optat pentru o reprezentare sub formă de graf ponderat. Fiecărui arc i s-a dat o valoare, în funcţie de distanţa dintre zonele unite de acel arc. Valoarea reprezintă un coeficient de atenuare, care va fi determinat pe cale experimentală, prin măsurători. Astfel, pentru schiţa din figura 6-12 vom avea următorul graf ponderat.

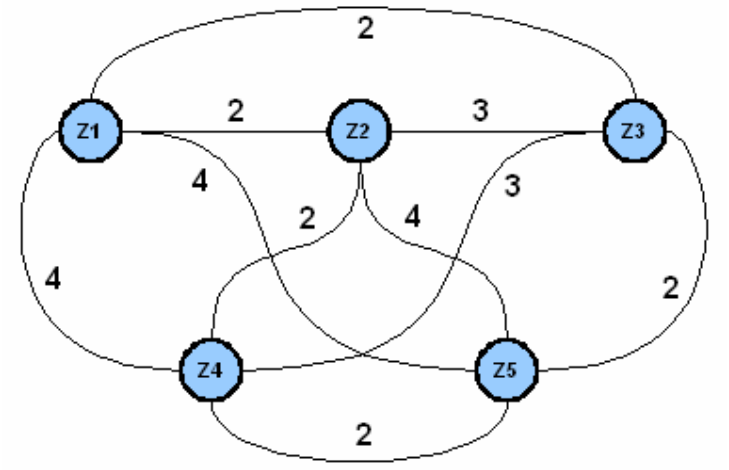


Figura 6-12 Graf ponderat reprezentând atenuările semnalului între diverse zone

Se observă cum pentru zonele 1 şi 2 sau 1 şi 3, care sunt învecinate, factorul de atenuare a fost ales 2, iar pentru zonele 1 şi 4 sau 2 şi 5, factorul de atenuare a fost ales 4. Aceste valori nu sunt cele reale folosite în implementarea framework-ului ci ele sunt folosite doar pentru a exemplifica principiul reprezentării vecinătăţii zonelor. Acest graf este unul neorientat, pentru că atenuarea este aceiaşi, indiferent din ce parte

am privi lucrurile. Acum să explicăm de ce a fost necesară definirea acestor vecinătăţi. Dacă la un moment dat se doreşte redistribuirea clienţilor unui anumit AP către alte AP-uri din sistem, atunci această hartă va fi folosită pentru a alege, în funcţie de nivelul de prioritate al fiecărui client, AP-ul la care va fi repartizat. Aceasta situaţie apare atunci când pentru un client cu prioritate mare trebuie eliberate resurse, în cazul nostru el ar avea nevoie de o anumită bandă de transfer garantată. Astfel, o parte din clienţii AP-ului la care este conectat utilizatorul cu prioritate mare vor fi repartizaţi către alte AP-uri din clădire. Vor fi alese acele AP-uri cât mai apropiate de AP-ul care trebuie descongestionat.

Pe lângă această hartă de vecinătăţi va putea fi folosită, dacă este necesar şi o hartă care va cuprinde modalităţile de deplasare dintr-o zonă în alta (figura 6-13).

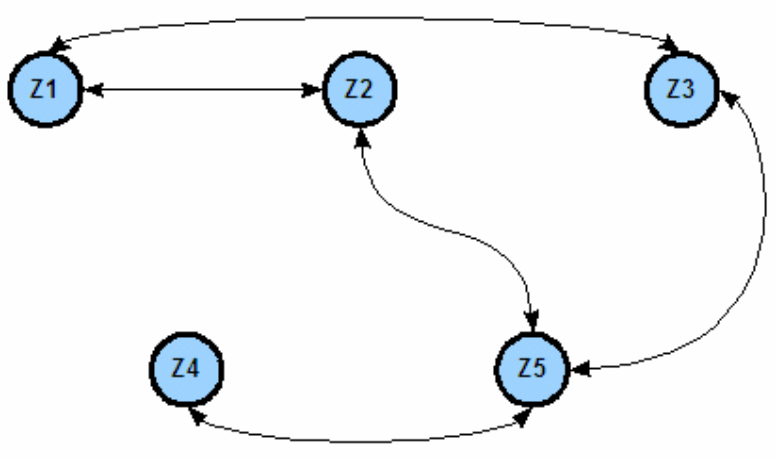


Figura 6-13 Graf reprezentând posibilităţile de acces între zone

Acest tip de hartă este utilă atunci când se implementează mecanismele care asigură păstrarea şi analiza istoricului deplasărilor unui anumit dispozitiv mobil. În acelaşi context, când in ecuaţie intervine un utilizator cu prioritate mare, dacă se deplasează dintr-o zonă spre altă zonă, având la dispoziţie istoricul mişcărilor sale, se va putea anticipa traseul pe care el îl va urma spre destinaţie şi astfel ar putea fi eliberate resurse în mod anticipat. Graful din figura de mai sus arată de exemplu că între zonele 1 şi 2 sau 1 şi 3 există acces direct, pe când între din zona 1 se poate ajunge în zona 5, doar trecând prin zona 2 sau din zona 1 se poate ajunge în zona 4 traversând pe rând zonele 2 şi 5.

6.3.4 Modulul „Applications Context”

Acest modul nu este de fapt unul de sine stătător. Culegerea de informaţii pentru acesta este făcută apelând la serviciile oferite de PAF. Scopul acestui modul este acela de a culege informaţii despre numărul şi tipul aplicaţiilor care rulează pe dispozitivul monitorizat.

6.3.5 Modulul „Other resources context”

Un exemplu de resursă care ar putea fi monitorizată este traficul realizat de către dispozitiv la Internet. Acest gen de informații pot fi culese de la proxy server-ul prin care se face accesul la Internet.

6.4 Implementarea UFRM

6.4.1 PAF – Power-Aware Framework

Atât PAF cât şi modulul de localizare au fost scris folosind Visual C++, deoarece a fost nevoie de acces la nivelul driverelor sistemului de operare.

Pentru PAF arhitectura este următoarea:

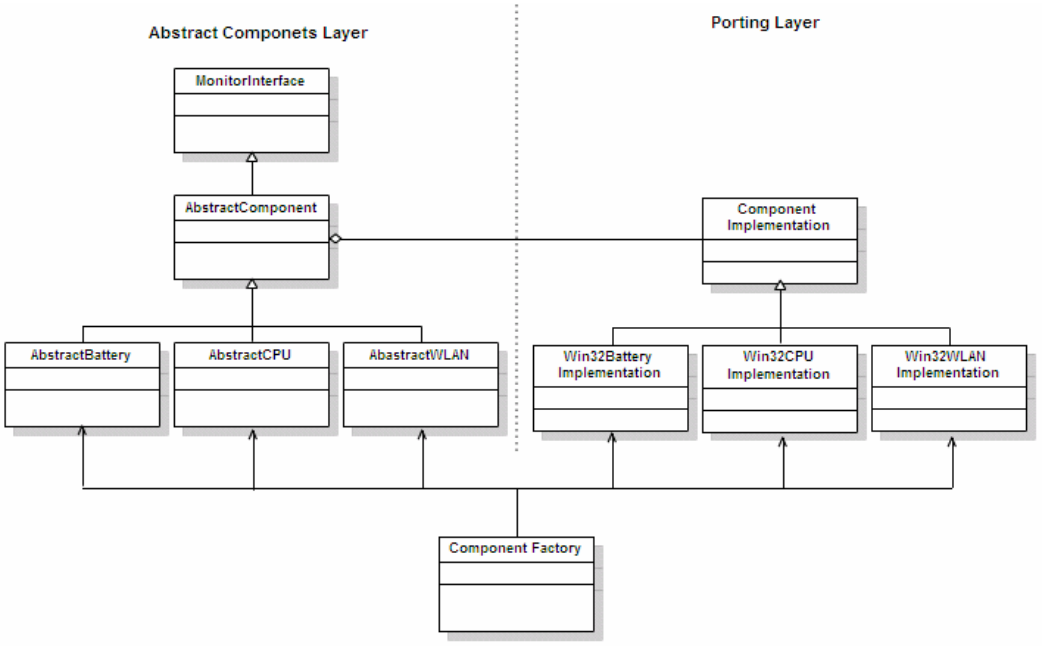


Figura 6-14 Arhitectura PAF

După cum se observă există un set de clase care implementează Porting Layer-ul. Să luăm de exemplu clasa Win32BateryImplementation. Interfața ei este compusă din metode de genul:

```
bool Initialize(void);
bool UpdateData(void);
long GetBatteryVoltage(void);
long GetBatteryCapacity(void);
long GetBatteryRate(void);
long GetBatteryCurrent(void);
long GetBatteryTemperature(void);
long GetBatteryLifeTime(void);
long GetBatteryFullLifeTime(void);
long GetBatteryFullCapacity(void);
```

Prin metoda UpdateData se reîmprospătează valorile parametrilor care caracterizează resursa monitorizată. Momentele de timp când se face acest refresh este dictat de Framework Core şi nu de aplicaţia care doreşte să obţină informaţii de la PAF.

Clasele de pe nivelul Abstract Component Layer asigură un acces uniform la resursele monitorizate. Fiecare resursă poate să fie caracterizată prin mai mulţi parametrii. Dacă luăm ca exemplu tot acumulatorul, aceşti parametrii sunt:

```
BatteryVoltage
BatteryCapacity
BatteryPower
BatteryCurrent
BatteryTemperature
BatteryLifetime
BatteryFullLifetime
BatteryFullCapacity
```

Pentru a avea acces la aceşti parametrii, clasa AbstractBatery implementează o interfaţă prin care se obţine acces uniform la oricare din parametrii resursei monitorizate. Metodele primesc ca şi parametru ID-ul resursei şi returnează valoarea curentă a acesteia.

```
virtual double GetParameterValue(int ParameterID);
virtual char* GetParameterName(int ParameterID);
virtual int GetParameterType(int ParameterID);
virtual int GetParameterStatus(int ParameterID);
```

6.4.2 Implementarea UFRM Kernel

Implementarea nucleului pentru UFRM s-a făcut folosind WCF (Windows Comunication Foundation) care este o platformă unificată pentru dezvoltarea aplicaţiilor orientate pe servicii (SOA – Service Oriented Applications). S-a optat pentru aceste tehnologii datorită gradului de abstractizare în spatele căruia sunt ascunse detaliile de implementare ale mecanismelor implicate în procesul de comunicaţie dintre aplicaţii rulând pe diverse sisteme de calcul.

Un serviciu de tip WCF poate fi privit ca un program care expune o colecţie de puncte de acces (*EndPoints*) la serviciile oferite. Un *Service EndPoint* este identificat prin trei elemente: *Address, Binding* şi *Contract.*

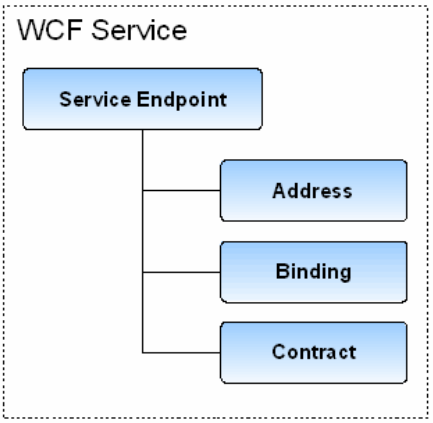


Figura 6-15 Structura unui serviciu WCF

Address

Identifică în mod unic un serviciu. Adresa cuprinde tipul de protocol folosit, numele maşinii gazdă şi dacă se specifică şi numărul portului, dacă este cazul.
[**transport**]://[**machine**][:optional **port**]

Contract

Există patru tipuri de contracte pe care un serviciu le expune în exterior. Acestea sunt:
Service Contract: expune serviciul propriu-zis.
Operation Contract: expune membrii serviciului.
Data Contract: descrie parametrii serviciului.
Fault Contracts: descrie condiţiile de apariţie a erorilor şi modul de tratare a acestora.

Bindings

Descrie modul de configurare al canalelor de comunicaţie.

Arhitectura claselor pentru UFRM Kernel este următoarea:

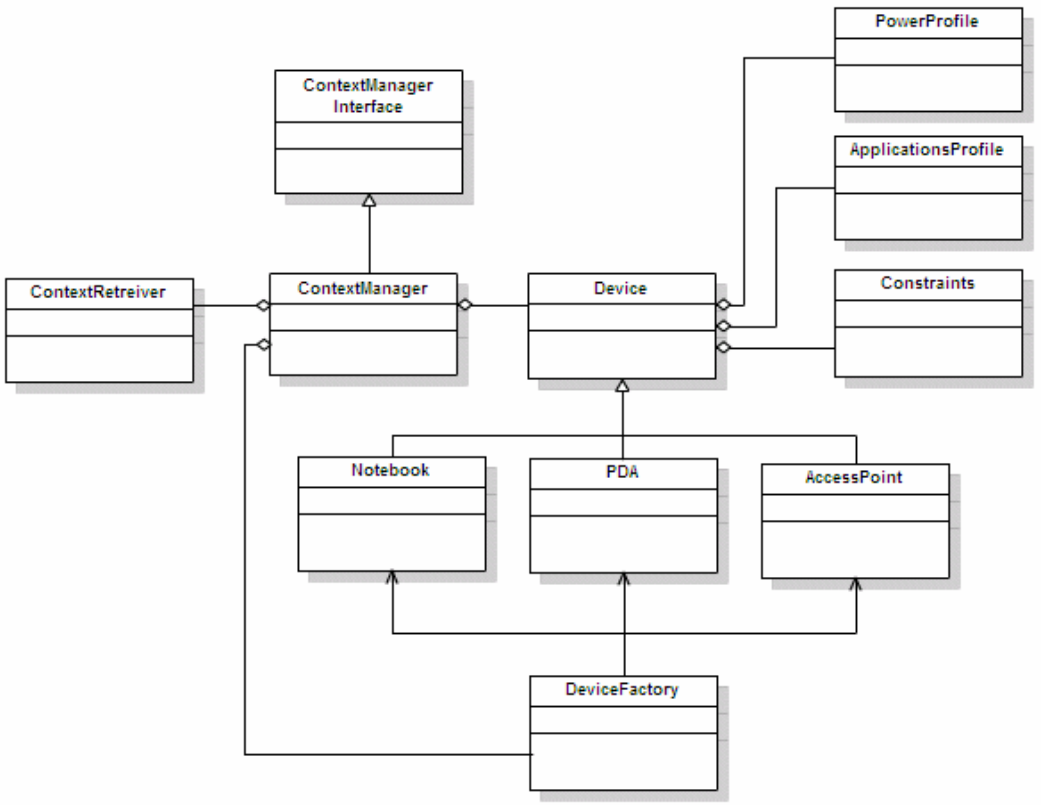


Figura 6-16 Arhitectura UFRM

6.5 Studii de caz şi concluzii

Pentru a exemplifica utilitatea framework-ului elaborat în acest capitol vor fi date câteva exemple de aplicaţii posibile care se pot construi în jurul acestuia. Serviciile oferite de aceste aplicaţii îşi găsesc utilitatea, în cadrul unor companii de dimensiuni mari, unde gestionarea eficientă a resurselor de calcul este un deziderat major.

În continuare vor fi prezentate câteva studii de caz privind unele din activităţile posibile într-o astfel de firmă.

6.5.1 Activităţi în cadrul unei firme

Descrierea infrastructurii:

- reţea 802.11
- fiecare angajat cu rol de conducere are în posesie un PDA sau un smartphone prevăzut cu interfaţă 802.11
- în plus, orice angajat poate avea în dotare un laptop
- la nivelul reţelei de calculatoare este implementat şi complet funcţional un sistem de tip UFRM

Desfăşurarea unei şedinţe

Să presupunem că urmează să se desfăşoare o şedinţă în cadrul unui anumit departament, şedinţă la care vor participa toţi conducătorii de grupuri din acel departament. Şeful de departament selectează din lista de angajaţi pe cei care urmează să-i convoace la şedinţă. Fiecare dintre ei va primi pe dispozitivul mobil aflat în posesie o înştiinţare referitoare la ora şi locaţia şedinţei.

La alegerea sălii se vor lua în considerare mai multe criterii:

- locaţia cea mai probabilă a participanţilor ţinând cont de un istoric al deplasărilor în cadrul firmei, pe care aceştia le-au avut în ultimul timp
- daca în zona sălii respective există resurse hardware corespunzătoare (ex: AP-uri, aparat de proiecţie, imprimantă) pentru a putea deservi toţi participanţii la şedinţă.
- dacă este cazul, unor participanţi li se vor trimite materiale pentru a fi tipărite folosind cea mai apropiată imprimantă de locaţia lor actuală. În acelaşi timp ei vor fii înştiinţaţi printr-un mesaj de unde să ridice materialul tipărit.
- odată ajunşi în sala de şedinţe, seful de departament va obţine o listă cu cei absenţi împreună cu locaţiile lor actuale, putând astfel lua hotărârea dacă aceştia să mai fie aşteptaţi sau şedinţa poate începe.
- pe durata şedinţei, dacă cineva doreşte să ia legătura cu unul din participanţi, va putea să identifice rapid şi locaţia acestuia.

Sistem de asistenţă pentru vizitatori

Presupunem situaţia unui vizitator care trebuie să participe la un interviu în cadrul unei firme. La intrarea în firmă, vizitatorul va primi un PDA căruia îi va fi asociată identitatea noului venit.

- în funcţie de persoana cu care trebuie să se întâlnească şi de poziţie curentă a acesteia în firmă, el va fi ghidat vocal spre acea persoană sau spre locul stabilit pentru întâlnire.
- la rândul ei, persoana cu care trebuie să se întâlnească vizitatorul va primi o înştiinţare prin care i se aduce la cunoştinţă că vizitatorul se află in firmă şi locaţia acestuia.

Acelaşi sistem poate fi aplicat şi în cazul noilor angajaţi, până aceştia se familiarizează cu arhitectura clădirii, dispunerea departamentelor în clădire, locaţia diverselor persoane cheie pentru activităţile pe care ei le au de desfăşurat în firmă.

Activităţi curente în cadrul unei firme

În cadrul firmei angajaţii vor fi împărţiţi după un sistem de privilegii. Unii vor avea garantată o anumită calitate a conexiunilor wireless, alţii vor beneficia de prioritate în ceea ce priveşte tipărirea unor materiale, alţii vor avea întâietate în ceea ce priveşte rezervarea unor spaţii pentru diverse activităţi, toate acestea în funcţie de natura activităţilor firmei şi de poziţia pe care o are angajatul în firmă.

Să analizăm acum câteva scenarii posibile:

Angajat cu bandă de comunicaţie garantată

În timpul deplasării prin firmă acestuia îi sunt eliberate resurse, în cazul nostru daca un AP care este cel mai apropiat de angajatul nostru, are o încărcare prea mare, atunci o parte din cei care sunt conectaţi la acest AP, adică cei cu privilegiile cele mai reduse, vor fi repartizaţi altor AP-uri mai îndepărtate. Acest proces nu implică acţiuni explicite ale utilizatorilor, toate etapele desfăşurându-se în mod total transparent.

Angajat cu prioritate la tipărirea diverselor materiale

Ţinând cont de locaţia curentă a acestuia i se vor pune la dispoziţie cele mai apropiate imprimante. Dacă utilizatorul se află în mişcare, analizând activităţile la care el urmează să participe, documentele vor fi direcţionate spre locaţia la care el se va afla la un moment dat.

Gestionarea resurselor

Sistemul propus va realiza şi o optimizare resurselor, prin acestea se pot urmări de exemplu următorii parametrii:

- autonomia unui anumit dispozitiv mobil
- daca starea bateriei este critică, va fi împiedicat să ruleze aplicaţii care ar duce la descărcarea rapidă a acumulatorului
- în cazul în care sunt rulate aplicaţii critice, iar acel utilizator beneficiază de privilegii sporite, atunci UFRM va optimiza consumul acelui dispozitiv mobil, astfel încât să-i confere o autonomie cât mai mare. Optimizări posibile sunt: garantare celor mai bune conexiuni wireless posibile, precum şi închiderea aplicaţiilor necritice.
- alerte către cei care asigură întreţinerea echipamentelor (ex. imprimantă fără tonner sau defectă, AP defect, etc.)

6.5.2 Asistență pentru un sistem de tip „warehouse management”

Sistemul propus de noi îşi poate găsi implementarea şi în cazul unor aplicații de tip „warehouse management”, adică gestionarea depozitelor mari de mărfuri.

În depozitele moderne există deja implementate sisteme de evidență a reperelor pe baza etichetelor cu coduri de bare sau RFID. Cititoarele pentru astfel de etichete au interfețe wireless 802.11 pentru a comunica datele în timp real în baza de date. Având în vedere că există deja o infrastructură 802.11 implementată şi care poate fi extinsă cu AP-uri suplimentare, o aplicabilitate imediată pentru sistemul nostru de management al resurselor ar fi aceea de optimizare a deplasării motostivuitoarelor şi dirijarea acestora în interiorul depozitului pentru localizarea mărfurilor dorite. În acelaşi timp poate fi monitorizat consumul şi starea acumulatorului pentru cititoarele de etichete, astfel încât să se poată face predicții privind autonomia de care va beneficia dispozitivul.

Capitolul 7 Concluzii

7.1 Contribuţii personale

Contribuţiile personale aduse în această teză sunt prezentate în cele ce urmează, ele evidenţiind spectrul larg de probleme care au fost abordate.

a) A fost realizată o sinteză personală asupra modului de funcţionare a principalelor protocoale din stiva TCP/IP evidenţiind acele aspecte care influenţează într-un mod nedorit comportamentul acestora în cazul în care la realizarea unei conexiuni intervin legături wireless într-o reţea de tip 802.11. Acelaşi tip de sinteză a fost aplicat şi în cazul standardului 802.11 pentru a putea identifica soluţii care să amelioreze efectul comportamentului eronat al protocolului TCP în anumite circumstanţe.

b) Protocolul TCP a fost proiectat să funcţioneze în reţele unde pierderea pachetelor datorate erorilor de transmisie are o probabilitate mică datorită fiabilităţii crescute a reţelelor bazate pe conexiuni cablate. Singura cauză care putea să genereze o astfel de situaţie era apariţia fenomenului de congestie. Acest lucru s-a schimbat, odată cu introducerea pe scară tot mai mare a reţelele wireless unde rata de apariţie a erorilor de transmisie este mult mai mare. TCP-ul interpretează acest fapt ca pe un fenomen de congestie, reacţionând în mod eronat. Pentru a investiga acest comportament au fost elaborate şi prezentate în capitolul 3, două metode de analiză a comportamentului protocolului TCP **[FM07, FAM06]**, care au fost aplicate în cazul unei reţele wireless 802.11. Aceste metode au un caracter general, ele putând fi folosite şi în cazul altor tipuri de reţele.

c) În reţelele WLAN 802.11 sunt prevăzute mecanisme pentru implementarea procedurii de handover (comutarea de la un AP la altul) care permite unui dispozitiv mobil să treacă dintr-un BSS în altul. Detaliile de realizare ale acestei proceduri sunt lăsate la latitudinea celor care implementează driver-ele pentru plăcile de reţea wireless. În toate implementările, driver-ul interfeţei de reţea wireless comandă comutarea de la un AP la altul doar în momentul în care vechea conexiune s-a întrerupt. În teză s-a propus o metodă de asistenţă **[FSM09]** care să ajute un dispozitiv mobil să realizeze procedura de handover ori de câte ori este nevoie, pentru a menţine o conexiune în parametrii optimi şi a nu declanşa mecanismele de control ale congestiei implementate la nivelul TCP-ului care au fost discutate în capitolul 1. Folosind metodele de analiză descrise în capitolul 3 s-a făcut o comparaţie între evoluţia unui dispozitiv cu şi fără sistemul de asistenţă activat. Din măsurătorile efectuate rezultă ca metoda propusă oferă îmbunătăţiri semnificative privind calitatea unei conexiuni TCP.

d) Capacitatea de localizarea este un aspect esenţial atunci când locaţia unui dispozitiv mobil reprezintă un parametru important al unui sistem de tip *context*-aware, sistem din care acel dispozitiv face parte. Folosind locaţia dispozitivului

se pot face diverse optimizări care vizează performanţa acelui dispozitiv, dar şi performanţa sistemului în ansamblu, putându-se oferi servicii particularizate în funcţie de necesităţile fiecărui utilizator în parte. Deşi problema aceasta al localizării a fost studiată în ultimii ani, nu există soluţii standardizate, bine puse la punct, care să fie uşor de implementat şi exploatat. În cadrul tezei a fost elaborat un algoritm original de localizare **[FMSA09, FMG10, MFG07, MFG+07, MF06]** a unui dispozitiv mobil în interiorul unei reţele de tip 802.11, algoritm caracterizat prin simplitate în implementare şi eficienţă.

e) S-a dezvoltat un sistem unificat de gestionare a resurselor **[FMS09, MTF09, MTM+09, FMS+09, SFK+09, MTFG08, MTFM08, MTFV08, MTF+08, MTF08, MTF07]** într-o reţea 802.11, denumit UFRM (Unified Framework for Resources Management) . Principalele mecanisme folosite de către acest sistem sunt:
   - determinarea locaţiei unui dispozitiv mobil
   - distribuirea uniformă a traficului între AP-urile care alcătuiesc reţeaua
   - păstrarea calităţii conexiunilor printr-un mecanism de handover asistat
   - monitorizarea consumului dispozitivelor mobile şi realizarea unor profiluri energetice

   Gestionarea capacităţii de transfer globale a unei reţele 802.11 şi distribuirea acesteia în funcţie de necesităţile şi nivelul de prioritate al diverşilor utilizatori este unul din aspectele vizate de UFRM. Un alt aspect vizat este optimizarea consumului la nivelul dispozitivelor gestionate.

   Pentru a pune în evidenţă posibilităţile de utilizare ale unui astfel de sistem unificat de gestionare a resurselor, au fost prezentate la finalul capitolului 6 câteva studii de caz.

7.2 Dezvoltări ulterioare

- Adăugarea la UFRM a posibilităţii de gestionare a diverse tipuri de senzori (ex. senzor de temperatură) pentru a îmbogăţi spectrul informaţiilor de context obţinute.
- Rafinarea metodei de localizare, încorporând mecanisme care să folosească standardul Bluetooth sau RFID.
- Dezvoltarea modulului „Protocols interface” pentru a permite generarea de semnalizări pe diverse niveluri ale stivei de protocoale, conform unui model de tip *cross-layer.*

Lista articolelor publicate

- **[FMS09] Sebastian Fuicu**, Marius Marcu, Bogdan Stratulat, „Using a WLAN Infrastructure as a Wireless Sensor Network in a Scalable Architecture”, 5th International Symposium on Applied Computational Intelligence and Informatics, 2009. SACI '09 , Timisoara, Romania, ISBN 978-1-4244-4478-6 [IEEEXplore]

- **[FSM09] Sebastian Fuicu,** Bogdan Stratulat, Marius Marcu, „An assistive system for reassociation management in a WLAN mesh environment”, Third International Conference on Next Generation Mobile Applications, Services and Technologies, 16-18 Sep 2009, Cardiff, Wales, UK. IEEE Computer Society Press. ISBN 978-0-7695-3786-3

- **[MTM+09]** Marius Marcu, Dacian Tudor, Horatiu Moldovan, **Sebastian Fuicu**, Popa Mircea, „Energy characterization of mobile devices and applications using power–thermal benchmarks”, **Microelectronics Journal Volume 40**, Issue 7, July 2009, Pages 1141-1153, IDS Number: 485GG

- **[FMG10] Sebastian Fuicu**, Marius Marcu, Anania Girban, "A mathematical model for Wireless LAN indoor positioning system", **Tensor Journal** of TENSOR Society on Differential Geometry & Its Applications and Mathematical Foundations of Information Sciences & Its Applications, edited by Tomoaki Kawaguchi, published by the Tensor Society, Chigasaki, Japan, Vol. 72 (2010) - in print, ISSN 0040-3504.

- **[FMSA09] Sebastian Fuicu**, Marius Marcu, Bogdan Stratulat, Anania Girban, „Effectiveness and Accuracy of Wireless Positioning Systems”, WSEAS TRANSACTIONS on COMPUTERS, Issue 9, Volume 8, September 2009

- **[MTF09]** Marius Marcu, Dacian Tudor, **Sebastian Fuicu, „**Towards a Network-Device Unified Framework for Power-Aware Wireless Applications **”,** The 5th International Wireless Communications and Mobile Computing Conference, IWCMC 2009

- **[FMS+09] Sebastian Fuicu**, Marius Marcu, Bogdan Stratulat, Iulia Stratulat, Anania Girban, „An open, low power framework for WLAN indoor positioning system”, 13th WSEAS International Conference on COMPUTERS, Rodos Island, Greece, July , 2009

- **[SFK+09]** Bogdan Stratulat, **Sebastian Fuicu**, Julia Klein, Marius Marcu, „A Survey of Power Saving Techniques for Wireless Communications", Scientific Bulletin of "Politehnica" University of Timisoara, Transactions on Automatic Control and Computer Science, Vol: 54(68) No: 1 / 2009, ISSN 1224-600X

- **[MTFG08]** Marius Marcu, Dacian Tudor, **Sebastian Fuicu**, Horatiu Moldovan, Voicu Groza, „A View on Mobile Terminal Power Efficiency of Wireless Communication", IEEE International Instrumentation and Measurement Technology Conference, I2MTC 2008, Victoria, Vancouver Island, Canada, May 12–15, 2008, ISI Proceedings No: BIP18

- **[MTFM08]** Marius Marcu, Dacian Tudor, **Sebastian Fuicu**, Mihai Micea, Silvia Copil, Florin Maticu, „Power Characterization of Multi-Threading Mobile Applications", Proceedings of the 12th WSEAS International Conference on Computers, WSEAS 2008, Heraklion, Greece, Jul. 2008, ISI Proceedings No: BIK27

- **[MTF+08]** Marius Marcu, Dacian Tudor, **Sebastian Fuicu**, Silvia Copil, Florin Maticu, Mihai Micea, „Power Efficiency Study of Multi-threading Applications for Multi-core Mobile Systems", WSEAS Transactions on Computers, Issue 12, Vol. 7, Dec. 2008, pp. 1875-1885, ISSN 1109-2750

- **[MTF08]** Marius Marcu, Dacian Tudor, **Sebastian Fuicu, „**A View on Power Efficiency of Multimedia Mobile Applications**",** International Joint Conferences on Computer, Information, and Systems Sciences, and Engineering, CISSE 2008, Dec. 2008

- **[FMG08] Sebastian Fuicu**, Marius Marcu, Anania Girban, „A mathematical model for Wireless LAN indoor positioning system", The 10th International Conference of TENSOR Society on Differential Geometry & Its Applications and Mathematical Foundations of Information Sciences & Its Applications, TENSOR 2008, Constanta

- **[MTFV08]** Marcu Marius, Tudor Dacian, **Sebastian Fuicu**, Horatiu Moldovan, Voicu Groza, „An Execution Framework for Power Characterization of Mobile Applications", IEEE International Instrumentation and Measurement Technology Conference, I2MTC 2008, Victoria, Vancouver Island, Canada, May 12–15, 2008 [IEEEXplore]

- **[MTF07]** Marius Marcu, Dacian Tudor, **Sebastian Fuicu, „**Power Efficiency Profile Evaluation for Wireless Communication Applications**",** International Joint Conferences on Computer, Information, and Systems Sciences, and Engineering, CISSE 2007, Dec. 2007, ISI Proceedings No: BIT20

- **[MFG+07]** Marius Marcu, **Sebastian Fuicu**, Anania Girban, Mircea Popa, „Experimental Test Cases for Wireless Positioning System", IEEE International Conference on "Computer as a Tool", EUROCON 2007, Warsaw, Poland, Sep. 2007, pp. 530-537, ISBN: 978-1-4244-0813-9, ISI Proceedings No: BHX65

- **[MFG07]** Marius Marcu, **Sebastian Fuicu**, Anania Girban, „Local Wireless Positioning System", 4th International Symposium on Applied Computational Intelligence and Informatics, 2007. SACI '07., Timisoara, Romania, May. 2007, pp. 171-176, ISBN 1-4244-1234-X, ISI Proceedings No: BGN60

- **[FM07] Sebastian Fuicu**, Marius Marcu, „An experimental approach of TCP mechanism in WLAN networks", Scientific Bulletin of "Politehnica" University of Timisoara, Transactions on Automatic Control and Computer Science, Vol: 52(66) No: 4 / 2007, pp. 159-162, ISSN 1224-600X

- **[MF06]** Marius Marcu, **Sebastian Fuicu, „**Wireless Local Positioning Systems: Issues and Challenges**",** Proceedings of the 2nd IEEE International Conference on Intelligent Computer Communication and Processing, ICCP 2006, vol. 2, Cluj, Romania, Sep. 2006, pp. 181-186, ISBN (10) 973-662-235-5

- **[FAM06] Sebastian Fuicu**, Alin Anton, Marius Marcu, "Experimental Measurements into TCP Congestion Mechanism over a Wireless 802.11 Network", Proceedings of the 7th International Conference on Technical Informatics, CONTI2006, Timisoara, Romania, Jun. 2006, pp. 317-322, ISBN (10) 973-625-321-X, (13) 978-973-625-321-8

Lista figurilor

Lista echipamentelor folosite

WRT54GS - Router wireless Linksys WRT54GS

- Standarde suportate: IEEE 802.11g, IEEE 802.3, IEEE 802.11b, IEEE 802.3u; Porturi: 1 x Wireless Access Point, 1 x DSL Internet, 4 x RJ-45 10/100 Full Duplex; Securitate: Filtrare adrese MAC, Internet Policy, Wi-Fi Protected Access 2 (WPA2), SPI firewall, WEP

WAP55AG - Linksys Dual Band 802.11a/g Wireless Access Point

- Standarde suportate: Ethernet, Fast Ethernet, IEEE 802.11b, IEEE 802.11g, IEEE 802.11a; Antene: 2 x External; Interfețe: 1 x network - Radio-Ethernet 1 x network - Ethernet 10Base-T/100Base-TX - RJ-45

WAP54G – Linksys Wireless-G Access Point

- Antene: 2 x Externe, Omnidirectionale; Porturi: 1 x 10/100 Auto-Cross Over (MDI/MDI-X); Standarde suportate: IEEE 802.11b , IEEE 802.3u , IEEE 802.11g , IEEE 802.3; Criptare: WEP 64/128-bit

DWL-G700AP - Access Point D-Link

- Wireless Access Point, 54Mbit/s, IEEE 802.11g/b, 64/128-Bit WEP, WPA and WPA2 wireless security, NAT with VPN Passthrough, DHCP Server/Client, IEEE 802.1x, 2dBi Gain detachable dipole antenna

DI-624 - Router wireless D-Link AirXtremeG

- standarde suportate: IEEE802.11b, IEEE802.3u, IEEE802.11g, IEEE 802.3
- CSMA/CA cu ACK; Frecventa: 2.4-2.462GHz, Tehnologia de modulare: Orthogonal Frequency Division Multiplexing (OFDM) si Complementary Code Keying (CCK); Rata transfer: 11/54/108; Antena: Single detachable reverse SMA; Raza de actiune: 100 m (interior), 400 m (exterior); VPN Pass Through/Multi-Sessions: PPTP, L2TP, IPSec

WUSB54G

- interfață de rețea wireless 802.11g, marca Linksys, cu conectare pe portul USB

EherScope Series II - Wireless Network Assistant

- Rezolvă problemele la frecvente de 2.4 GHz si 5 GHz oferind vizibilitate în rețelele 802.11 a/b/g
- Interfața intuitiva, Linux, display color cu operare *touch-screen*,
- Măsoară gradul de încărcare al rețelei si descoperă utilizatori care utilizează cel mai mult rețeaua, inclusiv access point-urile cele mai încărcate.
- Descoperă accesul neautorizat in rețea.
- Se poate verifica acoperirea curenta a rețelei in vederea extinderii acesteia.
- Vede toata infrastructura Wireless precum si diferiții clienți monitorizând puterea si calitatea semnalului

Notebook Lenovo X61

- Lenovo Notebook Thinkpad X61 Tablet Centrino Pro, Core 2 Duo L7500 (1.6 GHz), 2x1 GB, 160GB/5400rpm, Intel X3100, 256MB shared, 12.1 inch XGA touch TFT, Modem, Eth Giga, Intel 802.11n

Pocket LooX T830

- Smartphone Fujitsu-Siemens, Retea GSM 900 / 1800 / 1900, Ecran TFT touchscreen, 64K culori, Camera 2 MP, Rezoluție 1600x1200 pixeli, Timp stand by pana la 400 ore, Timp convorbire pana la 5 ore

Bibliografie

[ADH06] Y. Amir, C. Danilov, M. Hilsdale, R. Musăloiu-Elefteri, and N. Rivera, "Fast handoff for seamless wireless mesh networks," *Proceedings of the 4th international conference on Mobile systems, applications and services*, Uppsala, Sweden: ACM, 2006, pp. 83-95.

[Ana03] Giuseppe Anastasi, "IEEE 802.11 Ad Hoc Netwerks: Performace Measurements", ICDCSW'03, 2003

[BCI07] M. Bernaschi, F. Cacace, G. Iannello, and M. Vellucci, "Mobility Management for VoIP on Heterogeneous Networks: Evaluation of Adaptive Schemes," *IEEE Transactions on Mobile Computing*, vol. 6, 2007, pp. 1035-1047.

[BCQ07] C. Bolchini, C.A. Curino, E. Quintarelli, F.A. Schreiber, and L. Tanca, "A data-oriented survey of context models," SIGMOD Rec., vol. 36, 2007, pp. 19-26.

[BDR07] M. Baldauf, S. Dustdar, and F. Rosenberg, "A survey on context-aware systems," *Int. J. Ad Hoc Ubiquitous Comput.*, vol. 2, 2007, pp. 263-277.

[BH06] Y. Bejerano and S. Han, "Cell Breathing Techniques for Load Balancing in Wireless LANs," *Proceedings IEEE INFOCOM 2006. 25TH IEEE International Conference on Computer Communications*, Barcelona, Spain: 2006, pp. 1-13.

[Bol08] P. Bolliger, "Redpin - adaptive, zero-configuration indoor localization through user collaboration," *Proceedings of the first ACM international workshop on Mobile entity localization and tracking in GPS-less environments*, San Francisco, California, USA: ACM, 2008, pp. 55-60.

[CD05] Claude Chaudet, Dominique Dhoutaut, "Experiments of some performance issue with IEEE802.11 in ad hoc networks", WONS'05, 2005

[Che05] Yi-Chao Chen, "Sensor-Assisted Wi-Fi Indoor Location System for Adapting to Environmental Dynamics," MSWiM'05, October 10–13, 2005, Montreal, Quebec, Canada, 2005.

[Ciu06] Marc Ciurana, "Indoor Tracking in WLAN Location with TOA Measurements," MobiWAC'06, October 2, 2006, Torremolinos, Malaga, Spain, 2006.

[CK00] G. Chen and D. Kotz, *A Survey of Context-Aware Mobile Computing Research*, Dartmouth College, 2000.

[Com00] D. E Comer, "Internetworking with TCP/IP: Principles, Protocol and Architecture", 2000, Prentice Hall

[CPB+08] A. Carlotto, M. Parodi, C. Bonamico, F. Lavagetto, and M. Valla, "Proximity classification for mobile devices using wi-fi environment similarity," *Proceedings of the first ACM international workshop on Mobile entity localization and tracking in GPS-less environments*, San Francisco, California, USA: ACM, 2008, pp. 43-48.

[CSS05] Jonghwa Choi, Dongkyoo Shin, and Dongil Shin, “Research and implementation of the context-aware middleware for controlling home appliances,” *Consumer Electronics, IEEE Transactions on*, vol. 51, 2005, pp. 301-306.

[CT92] Valentin Cristea, Nicolae Țăpuş, „Rețele de calculatoare”, Ed. Teora, 1992

[DAS01] A.K. Dey, G.D. Abowd, and D. Salber, “A conceptual framework and a toolkit for supporting the rapid prototyping of context-aware applications,” Hum.-Comput. Interact., vol. 16, 2001, pp. 97-166.

[DX08] P. Dai and G. Xu, “Context-aware computing for assistive meeting system,” *Proceedings of the 1st international conference on PErvasive Technologies Related to Assistive Environments*, Athens, Greece: ACM, 2008, pp. 1-7.

[EXM04] E. Elnahrawy, Xiaoyan Li, and R. Martin, “The limits of localization using signal strength: a comparative study,” Sensor and Ad Hoc Communications and Networks, 2004. IEEE SECON 2004. 2004 First Annual IEEE Communications Society Conference on, 2004, pp. 406-414.

[FAM06] S. Fuicu, A. Anton, M. Marcu, "Experimental Measurements into TCP Congestion Mechanism over a Wireless 802.11 Network" ,Proceedings of the 7th International Conference on Technical Informatics, CONTI2006, Timisoara, Romania, Jun. 2006, pp. 317-322, ISBN (10) 973-625-321-X, (13) 978-973-625-321-8

[FJ93] S. Floyd, V. Jacobson “Random Early Detection Gateway for Congestion Avoidance” , IEEE, ACM 1993

[Flo01] Sally Floyd, “ A Report on Recent Developments in TCP Congestion Control”, IEEE Communications Magazine, April 2001

[FM07] S. Fuicu, M. Marcu, "An experimental approach of TCP mechanism in WLAN networks", Scientific Bulletin of “Politehnica” University of Timisoara, Transactions on Automatic Control and Computer Science, Vol: 52(66) No: 4 / 2007, pp. 159-162, ISSN 1224-600X

[FMG10] S. Fuicu, M. Marcu, A. Girban, "A mathematical model for Wireless LAN indoor positioning system", **Tensor Journal** of TENSOR Society on Differential Geometry & Its Applications and Mathematical Foundations of Information Sciences & Its Applications, edited by Tomoaki Kawaguchi, published by the Tensor Society, Chigasaki, Japan, Vol. 72 (2010) - in print, ISSN 0040-3504

[FMS09] S. Fuicu, M. Marcu, B. Stratulat, "Using a WLAN infrastructure as a wireless sensor network in a scalable architecture", 5th International Symposium on Applied Computational Intelligence and Informatics,Timisoara, Romania, May, 2009

[FMSA09] Sebastian Fuicu, Marius Marcu, Bogdan Stratulat, Anania Girban, „Effectiveness and Accuracy of Wireless Positioning Systems”, WSEAS TRANSACTIONS on COMPUTERS, Issue 9, Volume 8, September 2009

[FMS+09] Sebastian Fuicu, Marius Marcu, Bogdan Stratulat, Iulia Stratulat, Anania Girban, „An open, low power framework for WLAN indoor positioning system”, 13th WSEAS International Conference on COMPUTERS, Rodos Island, Greece, July , 2009

[FS09] X. Fafoutis and V. Siris, “Handover Incentives for WLANs with Overlapping Coverage,” *Wired/Wireless Internet Communications*, 2009, pp. 146-158.

[FSM09] S. Fuicu, B. Stratulat, M. Marcu, "An assistive system for re-association management in a WLAN mesh environment", 3rd International Conference on Next Generation Mobile Applications, Services and Technologies, Cardiff, UK 15-18 September 2009

[FMG08] S. Fuicu, M. Marcu, A. Girban, „A mathematical model for Wireless LAN indoor positioning system", The 10th International Conference of TENSOR Society on Differential Geometry & Its Applications and Mathematical Foundations of Information Sciences & Its Applications, TENSOR 2008, Constanta

[FZ03] Zhenghua Fu, Petros Zerfos, "The Impact of Multihop Wireless Channel on TCPThroughput and Loss", IEEE INFOCOM 2003

[Gas02] M. Gast, "802.11 Wireless Netwerks: The Definitive Guide", O'Reilly, 2002

[GC01] Panos Gevros, Jon Crowcroft, "Congestion Control Mechanisms and the Best Effort Service Model", IEEE Networ, May/June 2001

[GR05] Sumathi Gopal, Dipankar Raychaudhuri, "Experimental Evalution of TCP Simultaneous-Send Problem in 802.11 Wireless Local Area Networks", SIGCOMM'05 Workshop, August, 2005

[HHS+02] A. Harter, A. Hopper, P. Steggles, A. Ward, and P. Webster. "The anatomy of a context-aware application", Wireless Networks, Jan 2002.

[HMP00] S. Halabi, D. McPerson, "Internet Routing Architecture", 2000, Cisco Press

[Hui00] C. Huitema, "Routing in the Internet, 2nd Edition", 2000, Prentice Hall

[Jac88] Van Jacobson, "Congestion Avoidance and Control", Proceedings of SIGCOMM'88

[Kjæ08] Mikkel Baun Kjærgaard, "Efficient Indoor Proximity and Separation Detection for Location Fingerprinting," Mobilware '08, February 12-15, 2008, Innsbruck, Austria, 2008.

[KK05] V. Kawadia, P.R. Kumar, "Experimental Investigation into TCP Performance over Wireless Multihop Networks", SIGCOM'05 Workshop, August, 2005

[KK08] T. King and M.B. Kjærgaard, "Composcan: adaptive scanning for efficient concurrent communications and positioning with 802.11," Proceeding of the 6th international conference on Mobile systems, applications, and services, Breckenridge, CO, USA: ACM, 2008, pp. 67-80.

[Kuz05] A. Kuzmanovic "The Power of Explicit Congestion Notification", SIGCOMM'05

[KW03] K. Kuran, D. Woods, "The effects of badly behaved routers on Internet congestion", International Journal of Network Management, 2003

[Lee07] Dik Lun Lee, "A Model-Based WiFi Localization Method," *INFOSCALE 2007 June 6-8, 2007, Suzhou, China*, 2007.

[LKH+06] H. Lim, L. Kung, J.C. Hou, and H. Luo, "Zero-Configuration, Robust Indoor Localization: Theory and Experimentation," INFOCOM 2006. 25th IEEE International Conference on Computer Communications. Proceedings, 2006, pp. 1-12.

[LLG08] M. Liu, Z. Li, X. Guo, and E. Dutkiewicz, “Performance Analysis and Optimization of Handoff Algorithms in Heterogeneous Wireless Networks,” *IEEE Transactions on Mobile Computing*, vol. 7, 2008, pp. 846-857.

[MF06] M. Marcu, S. Fuicu, "Wireless Local Positioning Systems: Issues and Challenges", Proceedings of the 2nd IEEE International Conference on Intelligent Computer Communication and Processing, ICCP 2006, vol. 2, Cluj, Romania, Sep. 2006, pp. 181-186, ISBN (10) 973-662-235-5

[MFG+07] M. Marcu, S. Fuicu, A. Girban, M. Popa, "Experimental Test Cases for Wireless Positioning System", IEEE International Conference on "Computer as a Tool", EUROCON 2007, Warsaw, Poland, Sep. 2007, pp. 530-537, ISBN: 978-1-4244-0813-9

[MNV08] S. Mizzaro, E. Nazzi, and L. Vassena, “Retrieval of context-aware applications on mobile devices: how to evaluate?,” *Proceedings of the second international symposium on Information interaction in context*, London, United Kingdom: ACM, 2008, pp. 65-71.

[MSA03] A. Mishra, M. Shin, and W. Arbaugh, “An empirical analysis of the IEEE 802.11 MAC layer handoff process,” *SIGCOMM Comput. Commun. Rev.*, vol. 33, 2003, pp. 93-102.

[MSG+07] B. Mathieu, Meng Song, A. Galis, L. Cheng, K. Jean, R. Ocampo, M. Brunner, M. Stiemerling, and M. Cassini, “Self-Management of Context-Aware Overlay Ambient Networks,” *Integrated Network Management, 2007. IM '07. 10th IFIP/IEEE International Symposium on*, 2007, pp. 749-752.

[MTF+08] M. Marcu, D. Tudor, S. Fuicu, M. Micea, S. Copil-Crisan, F. Maticu, "Power Characterization of Multi-Threading Mobile Applications", Proceedings of the 12th WSEAS International Conference on Computers, WSEAS 2008, Heraklion, Greece, Jul. 2008, ISBN 960-8457-47-5

[MTF07] M. Marcu, D. Tudor, S. Fuicu, "Power Efficiency Profile Evaluation for Wireless Communication Application", International Joint Conferences on Computer, Information, and Systems Sciences, and Engineering, CISSE 2007, Dec. 2007.

[MTF08] Marius Marcu, Dacian Tudor, Sebastian Fuicu**, „**A View on Power Efficiency of Multimedia Mobile Applications**”,** International Joint Conferences on Computer, Information, and Systems Sciences, and Engineering, CISSE 2008, Dec. 2008

[MTF09] M. Marcu, S. Fuicu, D. Tudor , "Towards a Network-Device Unified Framework for Power-Aware Wireless Applications", International Wireless Communications & Mobile Computing, Leipzig, June 2009

[MFG07] Marcu Marius, Tudor Dacian, Sebastian Fuicu, Horatiu Moldovan, Voicu Groza, „An Execution Framework for Power Characterization of Mobile Applications”, IEEE International Instrumentation and Measurement Technology Conference, I2MTC 2008, Victoria, Vancouver Island, Canada, May 12–15, 2008 [IEEEXplore]

[MTFG08] Marius Marcu, Dacian Tudor, Sebastian Fuicu, Horatiu Moldovan, Voicu Groza, „A View on Mobile Terminal Power Efficiency of Wireless Communication”,

IEEE International Instrumentation and Measurement Technology Conference, I2MTC 2008, Victoria, Vancouver Island, Canada, May 12–15, 2008, ISI Proceedings No: BIP18

[MTFM08] Marius Marcu, Dacian Tudor, Sebastian Fuicu, Mihai Micea, Silvia Copil, Florin Maticu, „Power Characterization of Multi-Threading Mobile Applications", Proceedings of the 12th WSEAS International Conference on Computers, WSEAS 2008, Heraklion, Greece, Jul. 2008, ISI Proceedings No: BIK27

[MTFV08] Marcu Marius, Tudor Dacian, Sebastian Fuicu, Horatiu Moldovan, Voicu Groza, „An Execution Framework for Power Characterization of Mobile Applications", IEEE International Instrumentation and Measurement Technology Conference, I2MTC 2008, Victoria, Vancouver Island, Canada, May 12–15, 2008 [IEEEXplore]

[MTM+09] Marius Marcu, Dacian Tudor, Horatiu Moldovan, Sebastian Fuicu, Popa Mircea, „Energy characterization of mobile devices and applications using power–thermal benchmarks", **Microelectronics Journal** Volume 40, Issue 7, July 2009, Pages 1141-1153

[NH05] Kitae Nahm, Ahmed Helmy, "TCP over Multihop 802.11 Networks: Issues and Performance Enhancement", MobiHoc'05 May, 2005–12–12

[NN04] D. Niculescu and B. Nath, "VOR base stations for indoor 802.11 positioning," *Proceedings of the 10th annual international conference on Mobile computing and networking*, Philadelphia, PA, USA: ACM, 2004, pp. 58-69.

[PA06] Santosh Pandey and Farooq Anjum, "A Low-cost Robust Localization Scheme for WLAN," *WICON'06, The 2nd Annual International Wireless Internet Conference, August 2-5, 2006, Boston, MA, United States*, 2006.

[PKK06] S. Park, D. Kim, and B. Kang, "Context-aware Middleware Architecture for Intelligent Service in Mobile Environment," *Proceedings of the Sixth IEEE International Conference on Computer and Information Technology*, IEEE Computer Society, 2006, p. 240.

[RA07] Julian Randall and Oliver Amft, "LuxTrace: indoor positioning using building illumination," *Pers Ubiquit Comput (2007) 11:417–428*, 2007.

[RBP07] R. Raghavendra, E.M. Belding, K. Papagiannaki, and K.C. Almeroth, "Understanding handoffs in large ieee 802.11 wireless networks," *Proceedings of the 7th ACM SIGCOMM conference on Internet measurement*, San Diego, California, USA: ACM, 2007, pp. 333-338.

[RD07] L. Reyero and G. Delisle, "Always Best Located, a pervasive positioning system," *Wireless Pervasive Computing, 2007. ISWPC '07. 2nd International Symposium on*, 2007.

[RFC 2309] RFC 2309, "Recommendations on Queue Management and Congestion Avoidance in the Internet"

[RFC2001] RFC 2001, "TCP Slow Start, Congestion Avoidance, Fast Retransmit, and Fast Recovery Algorithms "

[RFC2018] RFC 2018 TCP Selective Acknowledgement Options

[RFC2481] RFC 3168, "The Addition of Explicit Congestion Notification (ECN) to IP

[RFC2581] RFC 2581, "TCP Congestion Control"

[RFC2914] RFC 2914, "Congestion Control Principles"

[RFC3390] RFC 3390, “Increasing TCP's Initial Window”

[RFC791] RFC 791, "Internet Protocol"

[RFC793] RFC 793, “Transmission Control Protocol”

[RL03] Pejman Roshan, Jonathan Leary “802.11 Wireless LAN Fundamentals”, Cisco Press, 2003, ISBN 1-58705-077-3

[ROP+05] M. Raento, A. Oulasvirta, R. Petit, and H. Toivonen, “ContextPhone: a prototyping platform for context-aware mobile applications,” *Pervasive Computing, IEEE*, vol. 4, 2005, pp. 51-59.

[RR03] S. Ryu, C. Rump, “Advanced in Internet congestion control”, IEEE Communications Surveys, 2003

[Sch03] J. H. Schiller, "Mobile Communications", 2003, Addison-Wesley

[Sch06] C. Philipp Schloter, “Wireless Symbolic Positioning using Support Vector Machines,” IWCMC’06, July 3–6, 2006, Vancouver, British Columbia, Canada, 2006.

[SCP+05] W. Schwinger, G. Ch, B. Pröll, W. Retschitzegger, and A. Schauerhuber, “Context-awareness in Mobile Tourism Guides – A Comprehensive Survey.”, 2005

[SFK+09] Bogdan Stratulat, Sebastian Fuicu, Julia Klein, Marius Marcu, „A Survey of Power Saving Techniques for Wireless Communications”, Scientific Bulletin of “Politehnica” University of Timisoara, Transactions on Automatic Control and Computer Science, Vol: 54(68) No: 1 / 2009, ISSN 1224-600X

[Sta05] W. Stallings, "Wireless Communications and Networks", 2005, Pearson Education International

[Std1] ANSI/IEEE Std 802.11, 1999 Edition, “Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications”

[Std2] IEEE Std 802.11g™-2003 (Amendment to IEEE Std 802.11™, 1999 Edition (Reaff 2003)

[Ste94] W.R. Stevens, “TCP/IP Illustrated, Vol.1”, 1994, Addison-Wesley

[Str00] C. Strugaru, “Sisteme de comunicaţii digitale”, 2000, Editura Orizonturi Universitare

[SWS+04] C. Sørensen, M. Wu, T. Sivaharan, G.S. Blair, P. Okanda, A. Friday, and H. Duran-Limon, “A context-aware middleware for applications in mobile Ad Hoc environments,” *Proceedings of the 2nd workshop on Middleware for pervasive and ad-hoc computing*, Toronto, Ontario, Canada: ACM, 2004, pp. 107-110.

[Tan96] A. S. Tanenbaum , “Computer Nerworks”, 3rd ed., 1996, Prentice-Hall PTR

[TK07] C.M. Takenga and K. Kyamakya, “Robust positioning system based on fingerprint approach,” *Proceedings of the 5th ACM international workshop on Mobility management and wireless access*, Chania, Crete Island, Greece: ACM, 2007, pp. 1-8.

[Van07] Konstantinos Vandikas, “Empirical-Based Analysis of a Cooperative Location-Sensing System,” Autonomics October 28-30, Rome, Italy, 2007.

[VP03] V.Vasudean, M.Parikh, “TCP and IEEE 802.11b Protocol Performance in Indoor Wireless Channels”, IEEE Sarnoff Symposium, 2003

[VV03] J. Vatn and J. Vatn, “An Experimental Study of IEEE 802.11b Handover Performance and Its Effect on Voice Traffic,” 2003.

[WB05] Jidong Wang and Lichun Bao, “Mobile context handoff in distributed IEEE 802.11 systems,” *Wireless Networks, Communications and Mobile Computing, 2005 International Conference on*, 2005, pp. 680-685 vol.1.

[WG05] Zhibin Wu, Sachin Ganu, “Experimental Investigation of PHY Layer Rate Control and Frequency Selection in 802.11 – based Ad-Hoc Networks”, SIGCOM’05 Workshop, August, 2005

[WHF+92] R. Want, A. Hopper, V. Falc~ao, and J. Gibbons. "The active badge location system", ACM Transactions on Information Systems (TOIS), Jan 1992.

[WKC07] K. Whitehouse, C. Karlof, and D. Culler, “A practical evaluation of radio signal strength for ranging-based localization,” *SIGMOBILE Mob. Comput. Commun. Rev.*, vol. 11, 2007, pp. 41-52.

[XP99] George Xylomenos and George C. Polyzos, “TCP and UDP Performance over a Wireless LAN”, PROCEEDINGS OF THE IEEE INFOCOM 1999, pp. 439–446

[ZHK+07] G.V. Zàruba, M. Huber, F.A. Kamangar, and I. Chlamtac, “Indoor location tracking using RSSI readings from a single Wi-Fi access point,” *Wirel. Netw.*, vol. 13, 2007, pp. 221-235.

[ZS91] L Zhang ,S. Shenker, “Observations on the Dynamics of a Congestion Control Algorithm, Proceedings of SIGCOMM’91