

# **COMUNICARE WIRELESS TIMP REAL PENTRU REȚELE DE SENZORI**

Teză destinată obținerii  
titlului științific de doctor inginer  
la  
Universitatea Politehnica Timișoara  
în domeniul CALCULATOARE ȘI TEHNOLOGIA  
INFORMAȚIEI

**ing. Valentin STÂNGACIU**

Conducător științific:	prof. univ. dr. ing. Vladimir Ioan Crețu
Referenți științifici:	prof. univ. dr. ing. Nicolae Țăpuș
	prof. univ. dr. ing. Sergiu Nevevschi
	prof. univ. dr. habil. ing. Mihai V. Micea

Ziua susținerii tezei: 19.06.2015

Seriile Teze de doctorat ale UPT sunt:

- |   |  |
|---|--|
| 1. Automatică                               | 10. Știința Calculatoarelor                |
| 2. Chimie                                   | 11. Știința și Ingineria Materialelor      |
| 3. Energetică                               | 12. Ingineria sistemelor                   |
| 4. Ingineria Chimică                        | 13. Inginerie energetică                   |
| 5. Inginerie Civilă                         | 14. Calculatoare și tehnologia informației |
| 6. Inginerie Electrică                      | 15. Ingineria materialelor                 |
| 7. Inginerie Electronică și Telecomunicații | 16. Inginerie și Management                |
| 8. Inginerie Industrială                    | 17. Arhitectură                            |
| 9. Inginerie Mecanică                       | 18. Inginerie civilă și instalații         |

Universitatea Politehnică Timișoara a inițiat seriile de mai sus în scopul diseminării expertizei, cunoștințelor și rezultatelor cercetărilor întreprinse în cadrul Școlii doctorale a universității. Seriile conțin, potrivit H.B.Ex.S Nr. 14 / 14.07.2006, tezele de doctorat susținute în universitate începând cu 1 octombrie 2006.

Copyright © Editura Politehnica – Timișoara, 2015

Această publicație este supusă prevederilor legii dreptului de autor. Multiplicarea acestei publicații, în mod integral sau în parte, traducerea, tipărirea, reutilizarea ilustrațiilor, expunerea, radiodifuzarea, reproducerea pe microfilme sau în orice altă formă este permisă numai cu respectarea prevederilor Legii române a dreptului de autor în vigoare și permisiunea pentru utilizare obținută în scris din partea Universității Politehnica Timișoara. Toate încălcările acestor drepturi vor fi penalizate potrivit Legii române a drepturilor de autor.

România, 300159 Timișoara, Bd. Republicii 9,  
Tel./fax 0256 403823  
e-mail: editura@edipol.upt.ro

## Cuvânt înainte

Teza de doctorat a fost elaborată pe parcursul activității mele în cadrul laboratorului DSP Labs, departamentul de Calculatoare, al Universității Politehnica Timișoara, unde sunt membru al echipei de cercetare din 2007; această echipă devenind o a doua familie pentru mine. Pe această cale doresc să le mulțumesc în mod deosebit domnului prof. dr. habil. ing. Mihai V. Micea, conducătorul laboratorului și echipei de cercetare, și colegilor și prietenilor mei as. dr. ing. Dan Chiciudean, ș.l. dr. ing. Răzvan D. Cioargă, care m-au "adoptat" și integrat în această nouă familie.

Pe parcursul acestei cercetări, și nu numai, am fost coordonat, îndrumat și sprijinit în primul rând de conducătorul de doctorat domnul prof. dr. ing. Vladimir I. Crețu, căruia îi datorez deosebite mulțumiri. Apoi aș dori să mulțumesc domnilor prof. dr. ing. Mircea Stratulat și domnului prof. Ionel Jian, pentru îndrumarea și colaborarea pe care am avut-o pe durata acestei activități de doctorat și nu numai.

Aș dori să mulțumesc de asemenea tuturor colegilor din departament și din universitate cu care am colaborat, sau m-am consultat pe parcursul acestei activități de cercetare. Mulțumiri deosebite, datorez și domnului prof. dr. ing. Toma L. Dragomir pentru șansa acordată prin colaborarea în cadrul grantului POSDRU-ATTRACTING.

O adâncă recunoștință se îndreaptă spre părinții mei, îndeosebi mama mea, pentru educația oferită și pentru tot ceea ce a făcut pentru mine.

De asemenea mulțumesc familiei și prietenilor pentru suportul oferit.

Această lucrare a fost finanțată parțial din proiectul strategic POSDRU/159/1.5/S/137070 (2014) al Ministerului Educației Naționale, România, co-finanțat din Fondul Social European – Investește în oameni, în cadrul Programului Operațional Sectorial Dezvoltarea Resurselor Umane 2007-2013.

Stângaciu, Valentin

**Comunicare wireless timp real pentru rețele de senzori**

Teze de doctorat ale UPT, Seria 14, Nr. 27, Editura Politehnica, 2015, 126 pagini, 53 figuri, 19 tabele.

Cuvinte cheie:

wireless sensor networks, real-time, communication, low-level protocol, time analysis, measurement framework, real-time systems, medium access protocol, real-time wireless communication

ISSN: 2069-8216

ISSN-L: 2069-8216

ISBN: 978-606-554-964-7

Rezumat,

Lucrarea de fata abordează tematica rețelelor de senzori cu comunicație wireless orientate în special spre aplicații critice timp real. Lucrarea este structurată în 8 capitole și are ca principală preocupare protocoalele de comunicație la nivel de acces la mediu pentru rețele de senzori. Prima parte a lucrării este destinată unei introduceri în domeniu precum și pentru a prezenta un studiu al literaturii actuale în domeniu. Astfel, în capitolul 2 se prezintă o scurtă sinteză a domeniului sistemelor timp real în care se pune accentul pe sistemele de operare timp real destinate microsystemelor incorporate. De asemenea principalul sistem de operare descris în această secțiune este nucleul HARETICK, acesta fiind cel folosit în cadrul cercetării doctorale a autorului. Tot în această secțiune se prezintă și aspecte generale ale dezvoltării de aplicații pentru sistemele timp real. Considerații generale despre rețelele de senzori sunt prezentate în capitolul 3 iar în capitolul 4 sunt descrise cele mai însemnate protocoale de acces la mediu pentru rețele de senzori. Tot în această secțiune se evidențiază principalele probleme ce apar în cadrul acestui tip de comunicație. Capitolul 5 este destinat unei analize realizate asupra protocolului de acces la mediu standardizat IEEE 802.15.4 și se arată astfel principalele probleme ce apar atunci când se dorește aplicarea acestui protocol în sisteme timp real. În capitolul 6 se prezintă un model de timp și un cadru de măsurare a componentelor de timp ce formează întârzierile cauzate de modulele radio în cadrul comunicației la nivel MAC. Se prezintă de asemenea și multe rezultate experimentale și comportamentele temporale ale componentelor de timp descrise. Capitolul 7 reprezintă principalele contribuții ce constă în propunerea unei noi scheme de acces la mediu ce îmbunătățește aspecte legate de TDMA și de asemenea, pe baza acestei scheme, se introduce un nou protocol de acces la mediu.

# Cuprins

<b>Listă acronime .....</b>	<b>7</b>
<b>Listă tabele .....</b>	<b>9</b>
<b>Lista de figuri .....</b>	<b>10</b>
<b>1 Introducere .....</b>	<b>12</b>
1.1 Tema și domeniile abordate .....	12
1.2 Motivația alegerii temei .....	13
1.3 Scopul și principalele obiective .....	14
1.4 Contextul de realizare a cercetării .....	15
1.5 Structura lucrării.....	15
<b>2 Sisteme timp real .....</b>	<b>17</b>
2.1 Sisteme timp real .....	17
2.2 Sisteme de operare în timp real pentru dispozitive embedded	18
2.3 Nucleul HARETICK.....	22
2.4 Generalități despre dezvoltarea de aplicații pentru sisteme hard real time .....	25
2.5 Concluzii .....	26
<b>3 Rețele de senzori wireless.....</b>	<b>27</b>
3.1 Noțiuni introductive.....	27
3.2 Aplicații ale rețelelor wireless de senzori .....	35
3.3 Rutarea informației în rețele wireless de senzori .....	38
3.4 Concluzii .....	42
<b>4 Protocoale MAC pentru rețele de senzori.....</b>	<b>44</b>
4.1 Considerații generale.....	44
4.2 Protocoale MAC bazate pe scheme de acces cu coliziuni .....	45
4.3 Protocoale MAC bazate pe scheme de acces fără coliziuni.....	46
4.4 Standardul IEEE 802.15.4.....	50
4.5 Concluzii .....	57
<b>5 Analiză experimentală a protocolului ZigBee și a standardului IEEE 802.15.4 din perspectiva timpului real și a interoperabilității .....</b>	<b>58</b>
<b>6 Model temporal și cadru de măsurare pentru comunicație wireless.....</b>	<b>63</b>
6.1 Model temporal pentru comunicația wireless .....	63
6.2 Cadru de măsurare a parametrilor de timp.....	68
6.3 Analiza experimentală a parametrilor de timp.....	71
6.4 Concluzii .....	84

<b>7 Greedy TDMA: Protocol timp-real eficient de acces la mediu</b>	<b>86</b>
7.1 Politica de acces Greedy TDMA .....	86
7.2 Rezultate experimentale .....	91
7.3 Protocolul de acces la mediu GTDMA-MAC .....	102
7.4 Concluzii .....	108
<b>8 Concluzii și perspective .....</b>	<b>110</b>
8.1 Concluzii .....	110
8.2 Sinteza contribuțiilor .....	111
8.3 Perspective de dezvoltare .....	112
<b>Referințe bibliografice .....</b>	<b>113</b>
<b>Publicații .....</b>	<b>123</b>

## Listă acronime

ACK (Acknowledge).....	107
ARQ (Automatic Repeat Request)...	40
BRAIN (Background Robotic Activity Induction Node).....	37
BS (Base Station) .....	41, 48
CAP (Contention Access Period) .....	54
CCA (Clear Channel Assessment)...	64
CFP (Contention Free Period).....	54
CH (Cluster Head).....	30
CN (Cluster Node).....	30
CORE-TX (Collaborative Robotic Environment - the Timisoara eXperiment) .....	36
CRC (Cyclic Redundancy Check) ....	106
CSMA (Carrier Sense Multiple Access) 39	
CSMA/CA (Carrier Sense Multiple Access Collision Avoidance) .....	44
CSMA/CD (Carrier Sense Multiple Access Collision Detection) .....	44
CSV (Comma-Separated Values) ...	70
CTS (Clear To Send) .....	46
DMS (Dynamic Modulation Scaling)	27
DNS (Domain name System).....	33
DTR (Data Transfer Rate) .....	67
DVS (Dynamic Voltage Scaling) .....	27
DW-MAC (Demand Wakeup MAC) ..	46
EDETA (Energy-efficient Adaptive Hierarchical and robust Architecture) .....	35
EDF (Earliest Deadline First).....	23
EIA (Electronic Industries Association) 32	
FCF (Frame Control Field) .....	56
FDMA (Frequency Division Multiple Access) .....	45, 48
FENP (Fixed Execution Non Preemptive) 23	
FFD (Full Functional Device) .....	51
FIFO (First in First out) .....	21
FIQ (Fast Interrupt) .....	24, 70
GAF (Geographical Adaptive Fidelity) 40	
GTS (Guaranteed Time Slot) .....	49
GTS (Guaranteed Time Slots).....	54
Hard Real Time (HRT) .....	22
HARETICK (Hard Real Time Compact Kernel) .....	22
HARP (Hierarchical Adaptive and Reliable Routing Protocol).....	41
HDLC (High-Level Data Link Control) 32	
HTTP (Hypertext Transfer Protocol)	33
I2C (Inter-Integrated Circuit) .....	66
IP (Internet Protocol) .....	32
Kernel Plug-Ins (KPI),.....	18
LEACH (Low-Energy Adaptive Clustering Hierarchy) .....	39
MAC (Medium Access Control) .....	13
MAM (Memory Accelerator Module)	25
MaRTE OS (Minimal Real-Time Operating System for Embedded Applications).....	21
ModX (Modul eXecutabil) .....	23
NH (Network Head) .....	30
OSI (Open System Interconnection)	31
PAN (Personal Area Networks).....	51
PARSECS (Predictable Architecture for Sensor Communication Systems)	33
PLL (Phase-Locked Loop) .....	66
Real Time Clock (RTC) .....	22
RFD (Reduced Funcional Device)....	51
RI-MAC (Received Initiated MAC)...	46
RISC (Reduced Instruction Set Computer).....	24
ROM (Read Only Memory) .....	19
RPAR (Real-time Power-Aware Routing) 40	
RTH-MAC (Real Time Hybrid MAC)..	48
RTS (Request To Send).....	46
SFD (Start of Frame).....	66
SHR (Synchronisation Header).....	57
Soft Real Time (SRT) .....	22
SPI (Serial Peripheral Interface)	26, 66
sTMDA (Stealing TDMA) .....	50
TAS-MAC (Traffic-Adaptive Synchronous MAC) .....	50
TCP (Transmit Control Protocol).....	32
TDM (Time-Division Multiplexing)...	47
TDMA (Time Division Multiple Access) 39, 46	
TDMA-ASAP (TDMA with Adaptive Slot- Stealing and Parallelism) .....	49

TS (Time Slot).....	47	WCET (Worst Case Execution Time)	18
TU (Time Unit) .....	87	WIT (Wireless Intelligent Terminal)	37
USART	(Universal	WLAN (Wireless Local Area Network)	
	Synchronous/Asynchronous		40
	Receiver/Transmitter) .....	WSN (Wireless Sensor Network) ....	27
VMS (Velocity Monotonic Scheduling)	66	Z-MAC (Zebra MAC) .....	49
	46		



## Listă tabele

Tabel 4-1. Structura generală a unui pachet IEEE 802.15.4 (sursa: [92]) .....	55
Tabel 4-2. Modulile de adresare specificare de Frame Control Field .....	56
Tabel 5-1. Configurația modulului XBee pentru comunicație broadcast.....	59
Tabel 5-2. Captură pachet baliză ZigBee .....	59
Tabel 5-3. Captură pachet date ZigBee cu adresare broadcast.....	60
Tabel 5-4. Configurația modulului XBee pentru comunicație punct-la-punct cu un alt modul existent .....	61
Tabel 5-5. Captură pachet date ZigBee cu adresare directă .....	61
Tabel 6-1. Structura unei înregistrări a unu eveniment în memorie .....	70
Tabel 6-2. Descrierea evenimentelor .....	72
Tabel 6-3. Calculul componentelor de timp din evenimentele măsurate .....	73
Tabel 6-4. Total măsurători efectuate.....	74
Tabel 6-5. Exemplificare set măsurători cu valori medii .....	75
Tabel 7-1. Parametrii de intrare simulator.....	92
Tabel 7-2. Mediul de simulare generat.....	93
Tabel 7-3. Rezultatele simulării.....	93
Tabel 7-4. Parametrii de intrare simulator.....	95
Tabel 7-5. Structura pachetului din sesiunea de date.....	105
Tabel 7-6. Structura pachetului din sesiunea de sincronizare .....	106
Tabel 7-7. Structura pachetului ACK.....	108

## Lista de figuri

Figura 2-1 Arhitectura QNX.....	20
Figura 2-2 Structura HARETICK (sursa: [6]) .....	22
Figura 2-3 Contextele de execuție ale nucleului HARETICK (sursa [21]).....	24
Figura 3-1 Schema bloc a unui nod dintr-o rețea de senzori .....	28
Figura 3-2 Exemplu rețea organizată pe clustere .....	30
Figura 3-3 Exemplu rețea ad-hoc.....	31
Figura 3-4 Modelul de referință OSI .....	32
Figura 3-5 Platforma CORE-TX .....	37
Figura 4-1 Divizarea timpului in sloturi TDMA .....	47
Figura 4-2 Divizarea timpului in frame-uri TDMA .....	47
Figura 4-3 Topologiile definite de standardul IEEE 802.15.4 (sursa: [103]).....	52
Figura 4-4 Exemplu de organizare pe clustere în IEEE 802.15.4 (sursa: [103]) ....	53
Figura 4-5 Structură de superframe definit de IEEE 802.15.4 (sursa: [103]).....	53
Figura 4-6 Structură de superframe definită în de IEEE 802.15.4 ediția 2006 (a) cu perioadă fără coliziuni (b) cu perioadă de inactivitate.....	55
Figura 6-1 Schema bloc a sistemului de măsurare a evenimentelor de timp .....	69
Figura 6-2 Valori minime, medii și maxime pentru componenta $T_{transfer\_A}$ .....	76
Figura 6-3 Valori minime, medii și maxime pentru componenta $T_{TX\_ON}$ – platforma CC2420.....	77
Figura 6-4 Valori minime, medii și maxime pentru componenta $T_{TX\_ON}$ – platforma CC2500.....	78
Figura 6-5 Valori minime, medii și maxime pentru componenta $T_{TX\_SFD}$ – platforma CC2420.....	79
Figura 6-6 Valori minime, medii și maxime pentru componenta $T_{TX\_SFD}$ – platforma CC2500.....	79
Figura 6-7 Valori minime, medii și maxime pentru componenta $T_{TX\_PACK}$ .....	80
Figura 6-8 Valori minime, medii și maxime pentru componenta $T_{RX\_ON}$ – platforma CC2420.....	81
Figura 6-9 Valori minime, medii și maxime pentru componenta $T_{RX\_ON}$ – platforma CC2500.....	81
Figura 6-10 Valori minime, medii și maxime pentru componenta $T_{RX\_SFD}$ – platforma CC2420.....	82
Figura 6-11 Valori minime, medii și maxime pentru componenta $T_{RX\_SFD}$ – platforma CC2500.....	83
Figura 6-12 Valori minime, medii și maxime pentru componenta $T_{RX\_PACK}$ – platforma CC2500.....	84
Figura 7-1 Conceptul <i>slot-stealing</i> .....	87
Figura 7-2 Situație de conflict în <i>slot-stealing</i> .....	87
Figura 7-3 Discretizarea unui slot de timp în unități de timp.....	87
Figura 7-4 Asignarea sloturilor de timp în ordinea crescătoare a adreselor nodurilor .....	88
Figura 7-5 Conceptul Greedy TDMA.....	89
Figura 7-6 Exemplificare concept Greedy TDMA.....	90
Figura 7-7 Iterație cadre TDMA .....	90
Figura 7-8 Simulare Greedy TDMA cu 10 noduri .....	94
Figura 7-9 Simulare Greedy TDMA cu 30 noduri .....	94

Figura 7-10 Simulare Greedy TDMA cu 100 noduri timp de 100 cadre TDMA cu distribuție uniformă a valorilor aleatoare generate .....	96
Figura 7-11 Histogramă valori generate cu distribuție uniformă a) dimensiunea pachetului b) probabilitatea de transmisie.....	96
Figura 7-12 Simulare Greedy TDMA cu 100 noduri timp de 100 cadre TDMA cu distribuție gaussiană a valorilor aleatoare generate .....	97
Figura 7-13 Histogramă valori generate cu distribuție gaussiană a) dimensiunea pachetului b) probabilitatea de transmisie.....	97
Figura 7-14 Simulare Greedy TDMA cu 500 noduri timp de 100 cadre TDMA cu distribuție uniformă a valorilor aleatoare generate .....	98
Figura 7-15 Histogramă valori generate cu distribuție uniformă pentru 500 noduri si 100 cadre TDMA a) dimensiunea pachetului b) probabilitatea de transmisie .....	98
Figura 7-16 Simulare Greedy TDMA cu 500 noduri timp de 100 sloturi TDMA cu distribuție gaussiană a valorilor aleatoare generate .....	99
Figura 7-17 Histogramă valori generate cu distribuție gaussiană pentru 500 noduri si 100 cadre TDMA a) dimensiunea pachetului b) probabilitatea de transmisie.....	99
Figura 7-18 Simulare Greedy TDMA cu 1000 noduri timp de 100 cadre TDMA cu distribuție uniformă a valorilor aleatoare generate .....	100
Figura 7-19 Histogramă valori generate cu distribuție uniformă pentru 1000 noduri si 100 cadre TDMA a) dimensiunea pachetului b) probabilitatea de transmisie.....	100
Figura 7-20 Simulare Greedy TDMA cu 1000 noduri timp de 100 sloturi TDMA cu distribuție gaussiană a valorilor aleatoare generate .....	101
Figura 7-21 Histogramă valori generate cu distribuție gaussiană pentru 1000 noduri si 100 cadre TDMA a) dimensiunea pachetului b) probabilitatea de transmisie.....	101
Figura 7-22 Situație de conflict .....	102
Figura 7-23 Situație de conflict .....	103
Figura 7-24 Situație de conflict în cadrul slotului de timp .....	104
Figura 7-25 Structura GTDMA-MAC .....	104
Figura 7-26 Slot de timp GTDMA cu zonă de ACK.....	107
Figura 7-27 Mecanismul de ACK în GTDMA-MAC .....	107

# 1 Introducere

## 1.1 Tema și domeniile abordate

În ultimele două decenii o atenție deosebită a fost acordată microsistemelor digitale, a dispozitivelor încorporate (embedded) acestea devenind un adevărat necesar în viața de zi cu zi. Alături de sistemele de calcul de dimensiuni medii și mari, cu grad ridicat de computație, dispozitivele embedded își demonstrează utilitatea în majoritatea domeniilor de activitate deși acestea de multe ori au resurse mai puține decât clasicele sisteme de calcul. Din această categorie aparte a sistemelor încorporate fac parte și rețelele de senzori care au beneficiat de o mai mare atenție în ultimii 10 poate chiar 15 ani. Rețelele de senzori pot fi găsite într-o gamă largă de aplicații precum sisteme de monitorizare a mediului, sisteme de securitate, sisteme de detecție a incendiilor, sisteme robotizate de intervenție în caz de calamități, aplicații militare, platforme robotice.

O rețea de senzori este alcătuită în principal de microsisteme încorporate (noduri), fixe sau mobile, de obicei slabe în resurse hardware, cu alimentare în special limitată, pe baterie, cu capacități senzoriale superioare și de asemenea cu capacitate de comunicare fără fir. Aceste noduri, din punct de vedere hardware, sunt independente dar funcționalitatea acestora nu prezintă importanță decât atunci când acestea funcționează ca un tot unitar cu menirea de a îndeplini o funcție, un rol. Astfel apare necesitatea de a stabili o comunicare fiabilă între aceste noduri pentru a se putea realiza o rețea de senzori.

O atenție deosebită este acordată în ultima perioadă sistemelor timp real în cazul cărora nu doar rezultatul operației este important, dar în egală măsură este important și timpul necesar obținerii rezultatului. Un caz banal ar fi un sistem de detectarea incendiilor. Acesta ar trebui să alarmeze populația în faza incipientă a incendiului, nu după o întârziere considerabilă când orice efort este în zadar. Pentru a se impune acestor cerințe de timp nu doar microsistemele ce funcționează individual ca noduri ale rețelei de senzori trebuie să funcționeze în timp real ci și întreaga rețea să respecte niște cerințe stricte de timp. Pentru acesta este necesar ca o comunicare între nodurile unei rețele de senzori să respecte anumite cerințe de timp și în același timp să-și mențină scalabilitatea, toleranța la defecte și consumul de energie electrică scăzut.

O comunicare la nivelul unei rețele de senzori este realizată prin intermediul unei stive de protocoale ierarhizate pe nivele. Conform multor modele în acest sens aceasta stivă conține protocoale specializate pe accesul la mediu (protocoale MAC), corecția și detecția erorilor, rutarea informației etc.

## 1.2 Motivația alegerii temei

Protocoalele de acces la mediu sunt protocoalele localizate la baza unei stive de comunicație și sunt cele care influențează cel mai mult comportamentul temporal al întregii comunicații. Aceste protocoale, deși au fost mult timp în atenția cercetătorilor, s-au dezvoltat mai puțin în direcția sistemelor timp real. Majoritatea studiilor s-au axat pe reducerea consumului de energie electrică. Acest lucru este într-adevăr esențial într-o rețea de senzori dat fiind faptul că nodurile sunt alimentate de o baterie sau acumulator.

Într-o rețea de senzori cu cerințe de timp real, reducerea consumului de energie electrică nu se află pe primele poziții în ordinea priorităților caracteristicilor unui protocol de acces la mediu. În acest caz, al rețelelor de senzori ce funcționează în manieră timp real, protocoalele de acces la mediu trebuie să se concentreze asupra cerințelor de timp spre a oferi o bază de comunicație bine determinată în timp. Majoritatea protocoalelor de acest tip se rezumă doar la garanții de timp soft real-time oferind doar un trafic garantat statistic. Acest lucru constituie una dintre principalele probleme ale acestui domeniu.

De asemenea o altă problemă des întâlnită este aceea că în majoritatea situațiilor cercetătorii își validează și demonstrează soluția folosind medii de simulare care nu au capacitatea de a aproxima caracteristicile unui mediu de comunicație cu grad mare de instabilitate precum mediul de comunicație fără fir. În plus, se folosesc în multe situații cazuri particulare extrem de restrânse care în realitate nu sunt fezabile.

Mai mult decât atât, ținând cont de faptul că protocoalele de acces la mediu sunt la baza stivei de comunicație, acestea lucrează direct cu modulul radio al sistemului. Acest modul radio are la rândul lui un anumit grad de predictibilitate și introduce o serie de latențe. Întârzierile de timp ce sunt introduse de către modulele radio la nivelul sistemului ce compune un nod al rețelei sunt de cele mai multe ori ignorate, considerând o funcționare ideală a modulului radio. Un protocol de acces la mediu nu poate ignora aceste aspecte.

Prin această lucrare se va acționa asupra protocoalelor de comunicare de nivel MAC (Medium Access Control) în vederea obținerii unor garanții de timp în cadrul comunicării dintre nodurile unei rețele de senzori. Aceste protocoale nu doar că trebuie să asigure anumite cerințe stricte de timp dar trebuie să păstreze scalabilitatea și toleranța la defecte a întregii rețele de senzori. Aceste protocoale precum și protocoalele superioare trebuie să țină cont și de situația când nodurile rețelei sunt mobile (colective robotice). De asemenea, în aceasta lucrare se va acorda o importanță deosebită studiului acestor protocoale în cazuri reale, nu bazându-se doar pe simulări care de cele mai multe ori nu sunt conforme cu realitatea mai ales în acest domeniu al comunicației wireless dintre nodurile unei rețele de senzori.

### 1.3 Scopul și principalele obiective

Principalul scop al activității de cercetare este studiul și găsirea unei soluții a problemelor ce apar în nivelele de acces la mediu la comunicația din cadrul rețelelor de senzori.

Principalele obiective ale activității de cercetare sunt următoarele:

- [O1]. Efectuarea unui studiu amănunțit al problemelor ce apar pentru a realiza un model de protocol MAC de comunicare în timp real. Acest studiu presupune o amplă documentare în legătură cu protocoalele de comunicare existente și stabilirea limitărilor acestora în contextul de timp real.
  - [T1]. Realizarea unui studiu comparativ al protocoalelor MAC folosite la ora actuală în rețelele de senzori wireless.
  - [T2]. Se poate observa din literatură faptul că majoritatea cercetătorilor și-au validat rezultatele folosind medii de simulare. Din practică, în schimb se poate observa că în cazul comunicării wireless aceste simulatoare nu pot aproxima situațiile din mediul real și astfel multe din protocoalele existente pe hârtie și validate folosind simulatoare nu pot funcționa într-un mediu real în aceeași manieră cum funcționează în medii de simulare. Din acest motiv este necesar un studiu al problemelor apărute la simulatoarele de rețele precum și a situațiilor ce acestea nu le pot acoperi.
  - [T3]. Pentru ca un protocol de nivel MAC să poată oferi garanții de timp real în comunicare este necesar să se identifice penalitățile de timp introduse de modulele de comunicare wireless precum și nivelul de predictibilitate al acestora. În acest pas îmi propun studiul penalităților de timp și a nivelului de predictibilitate oferit de modulele de comunicare wireless existente pe piață.
- [O2]. Realizarea unui protocol de timp MAC care să fie orientat atât pe menținerea unui consum redus de energie electrică la nivelul unui nod cât și la nivelul întregii rețele, cu facilități de comunicare în timp real.
  - [T1]. Dezvoltarea unui protocol de tip MAC care să fie orientat atât pe menținerea unui consum redus de energie electrică la nivelul unui nod cât și la nivelul întregii rețele cu facilități de comunicare în timp real.
  - [T2]. Validarea protocolului folosind pentru început medii de simulare.
- [O3]. Realizarea practică a unui dispozitiv/modul/sistem de comunicare în timp real care să ofere facilități de comunicare hard real time, destinat amplasării acestuia pe nodurile unei rețele de senzori.
  - [T1]. Selectarea unor module / clase de module de comunicare wireless studiate anterior pentru a putea fi folosite la validarea practică a algoritmilor și tehnicilor dezvoltate.
  - [T2]. Realizarea unui modul de comunicație care să integreze toate cele propuse mai sus și să valideze din punct de vedere practic protocoalele dezvoltate și eventualele rezultate obținute prin simulări.

## 1.4 Contextul de realizare a cercetării

Această lucrare a fost finanțată parțial din proiectul strategic POSDRU/159/1.5/S/137070 (2014) al Ministerului Educației Naționale, România, co-finanțat din Fondul Social European – Investește în oameni, în cadrul Programului Operațional Sectorial Dezvoltarea Resurselor Umane 2007-2013.

Activitatea de cercetare s-a desfășurat în cadrul Centrului de Cercetare în Calculatoare și Tehnologia Informației (CCCTI), în cadrul laboratoarelor de Sisteme Timp-Real, Robotice și de Prelucrare Numerică a Semnalelor (STRRPNS), laboratoarele de prelucrarea numerică a semnalelor DSP Labs Timișoara din Cadrul Facultății de Automatică și Calculatoare, Departamentul Calculatoare din Universitatea Politehnica Timișoara.

## 1.5 Structura lucrării

Lucrarea de față este structurată după cum urmează:

- Capitolul 2 introduce succint cititorul în sisteme în timp real prezentându-i diferența între un sistem clasic și un sistem în timp real precum și diferența între dezvoltarea aplicațiilor clasice și dezvoltarea aplicațiilor care să respecte principiile sistemelor în timp real. De asemenea se prezintă și principalele sisteme de operare în timp real pentru sisteme embedded.
- În capitolul 3 se prezintă în mare considerații teoretice despre rețele de senzori wireless. În prima parte a acestui capitol se prezintă noțiuni de bază despre acest tip de rețele precum și despre comunicarea dintre sistemele numerice. Legat de comunicarea între sistemele numerice se prezintă modelul OSI și cum se regăsește acest model în comunicarea din cadrul rețelelor de senzori. Se arată de asemenea și importanța acestor rețele de senzori prin prezentarea unor aplicații realizate cu ajutorul rețelelor de senzori wireless. În continuare se pot găsi noțiuni despre protocoalele de rutare a informației în rețelele de senzori.
- În capitolul 4 se prezintă principalele protocoale de acces la mediu folosite în acest domeniu cu ridicarea principalelor probleme și impedimente. Se pune accent pe prezentarea mai pe larg a celei mai răspândite stive de protocoale ce se folosește în rețelele de senzori și anume stiva ZigBee. De asemenea se prezintă și protocolul de acces la mediu folosit de această stivă de protocoale și anume standardul IEEE 802.15.4.
- În capitolul 5 se prezintă o analiză experimentală a protocolului de acces la mediu IEEE 802.15.4 și a stivei ZigBee din perspectiva interoperabilității și a operării în timp real.
- În capitolul 6 se prezintă un model care definește principalele componente de timp ce compun întârzierea în transferul informației între nodurile unei rețele de senzori. De asemenea tot în această secțiune se prezintă și un sistem hardware/software de măsurare și analiză a acestor componente de timp. În partea finală a capitolului se prezintă câteva seturi de rezultate experimentale obținute prin analizarea modelului definit anterior aplicat comunicării între noduri realizate cu module radio de diferite tipuri.

- În prima parte a capitolului 7 se prezintă o schemă de acces la mediu, bazată pe schema de acces TDMA, denumită GTDMA. Această schemă generalizează tehnicile de slot-stealing prezentate în literatură și oferă o soluție omogenă de a utiliza acest tip de politică de acces atât în mediul cu fir cât și aplicațiile cu comunicație fără fir. Se prezintă de asemenea și rezultate experimentale obținute în urma simulării acestei politici. În ultima parte a acestui capitol se prezintă protocolul de acces GTDMA-MAC, bazat pe schema de acces GTDMA, ce reprezintă un model de protocol în timp real.
- Capitolul 8 conține lucrarea de față și prezintă o sinteză a contribuțiilor aduse în cadrul tezei de doctorat precum și principalele perspective de dezvoltare și cercetare viitoare.



## 2 Sisteme timp real

Sistemele în timp real constituie unul din principalele domenii care se conturează în lucrarea de față. În acest capitol se vor prezenta principalele caracteristici ale unui sistem în timp real. De asemenea ca și principal interes al acestei teze sunt sistemele încorporate. În acest sens în acest capitol se va face o trecere în revistă a principalelor sisteme de operare în timp real pentru sisteme încorporate. Se va pune accentul asupra nucleului HARETICK, acesta fiind sistemul de operare folosit în cadrul cercetării. În plus, în ultima parte se vor discuta scurte aspecte privind proiectarea și implementarea aplicațiilor pentru sisteme de operare în timp real, cu evidențierea principalelor diferențe față de proiectarea și implementarea aplicațiilor pe sisteme clasice.

### 2.1 Sisteme timp real

Sistemele în timp real fac parte dintr-o categorie aparte de sisteme în care timpul de răspuns este un parametru esențial. Acest lucru se datorează faptului că intrările și ieșirile unui astfel de sistem sunt strict dependente de mediul înconjurător, acesta din urmă fiind practic tot un sistem continuu în timp. Diferența dintre momentul de timp de când se aplică informație intrărilor unui sistem în timp real și momentul când rezultatul este disponibil la ieșire, este dependentă de aplicație dar este bine determinată și predictibilă. [1]

De asemenea un sistem în timp real poate fi definit ca fiind acel sistem în care o funcționare corectă a sistemului nu depinde doar de corectitudinea rezultatului logic al acestuia dar mai mult depinde de durata de timp necesară producerii acestui rezultat. Așadar parametrul esențial în sisteme în timp real este timpul. [2, 3]

Din acestea se poate deduce ca sistemele în timp real sunt în marea majoritate sisteme reactive (sisteme reflexe) ce au ca funcție interacțiunea cu mediul înconjurător în mod continuu. Acest tip de sisteme în principiu reacționează la semnalele din exterior generând la rândul lor semnale de ieșire folosite în principal pentru control. Sistemele în timp real acceptă totuși și interacțiune din partea unui operator uman având astfel și funcții de sisteme interactive [4].

Sistemele în timp real pot fi implementate atât prin sisteme de calcul clasice cu putere ridicată de calcul cât și prin sisteme de mici dimensiuni, prin sisteme embedded. În primul caz, al sistemelor de calcul clasice principiile de bază ale unui sistem în timp real pot fi implementate cu o mai mare ușurință dat fiind faptul că aceste sisteme de calcul pun la dispoziție bogate resurse de memorie, calculație iar consumul de energie nu prezintă importanță, acestea fiind alimentate de obicei de la rețea. Pe de altă parte, în cazul sistemelor embedded implementarea conceptelor de timp real prezintă o ridicată dificultate ceea ce a creat domenii importante de studii în lumea universitară. Dispozitivele embedded sunt ideale pentru monitorizare și control ceea ce le face cele mai mult vizate pentru aplicarea conceptelor de timp real.

Totuși, în ambele cazuri este necesară, pe lângă componentele hardware și existența unei puternice componente software care să asigure implementarea conceptelor de timp real. Această componentă este realizată de sisteme de operare în timp real ele fiind prezente atât în variante pentru calculatoarele clasice cât și pentru dispozitivele embedded. Această categorie aparte de sisteme de operare prezintă

anumite caracteristici menite să implementeze principiile sistemelor în timp real: predictibilitate, timp de răspuns, multitasking cu respectarea cerințelor stricte de timp, existența unei referințe interne de ceas în timp real [5].

În funcție de restricțiile de timp, sistemele în timp real pot fi [6]:

- Sisteme în timp real critice – îndeplinirea restricțiilor de timp, respectarea deadline-urilor este critică. O nerespectare a parametrilor de timp impuși poate duce la rezultate catastrofale.
- Sisteme în timp real stricte – nerespectarea unor parametri de timp poate duce la rezultate eronate dar întregul sistem își poate îndeplini funcția destinată.
- *Sisteme soft real Time (lejere)* – rezultatul procesului nu este grav afectat dacă nu se îndeplinesc cerințele de timp. Folosite cu precădere în cazul aplicațiilor multimedia [7].

Constrângerile de timp în sistemele în timp real pot fi constrângeri hard real-time (pentru aplicații critice) și constrângeri soft real time. Aceste constrângeri sunt definite pentru fiecare task/aplicație a unui sistem în timp real prin intermediul unor parametri [8]:

- Momentul de timp când un task devine disponibil pentru execuție
- Deadline – momentul de timp până când un task trebuie să își încheie execuția
- Timpul de răspuns – intervalul de timp de când taskul a fost lansat în execuție și momentul când acesta și-a încheiat execuția.
- WCET (Worst Case Execution Time) – Timpul maxim de execuție al unui task în cel mai rău caz. Acest timp este esențial și trebuie determinat cu o foarte mare precizie întrucât dacă acest timp este depășit este grav afectată funcționalitatea unui sistem în timp real.

## 2.2 Sisteme de operare în timp real pentru dispozitive embedded

Nodurile unei rețele de senzori sunt micro sisteme embedded alimentate în general din baterie care formând rețeaua de senzori trebuie să realizeze funcțiile impuse de aplicație. În general aplicațiile pentru care rețelele de senzori sunt destinate au cerințe de timp bine definite. Pentru îndeplinirea acestor condiții se uzitează ca și componentă software pentru nodurile rețelei sisteme de operare în timp real destinate dispozitivelor embedded. În această scurtă secțiune se vor trece în revistă principalele sisteme de operare în timp real pentru dispozitivele embedded.

Un important sistem de operare în timp real destinat și sistemelor embedded mai complexe este LynxOS cu un nucleu bazat pe nucleul UNIX. Acesta este folosit în industria aeronautică și la sistemele de siguranță critice, fiind un sistem extrem de stabil. Micronucleul pune la dispoziție mecanismele esențiale de planificare a proceselor, gestionare a întreruperilor și pentru sincronizarea proceselor. De asemenea nucleul mai oferă o multitudine de servicii organizate în module, numite Kernel Plug-Ins (KPI), precum sistem de fișiere, gestionarea sistemului de intrare/ieșire, stiva de protocoale TCP/IP, etc. Important de precizat este faptul că aceste KPI au libertatea de a crea oricâte procese și fire de execuție pentru a-și gestiona resursele. De asemenea documentația aferentă sistemului LynxOS precizează că acesta nu face nici o schimbare de context pentru trimiterea de mesaje între modulele KPI iar comunicarea între aceste module durează doar câteva instrucțiuni. Important de spus despre LynxOS este că

acesta este printre puținele sisteme de operare în timp real existente de tip hard real time [8].

În industria constructoare de automobile se remarcă și sistemul OSEK, acesta devenind chiar un standard în sisteme de operare în timp real. OSEK rulează pe dispozitive embedded, moderate în resurse hardware, destinate monitorizării și controlului diferiților parametri din vehicule. Fiind un sistem folosit în domeniul siguranței acesta este relativ rigid, prezintă totuși un anumit grad de scalabilitate, portabilitate, configurabilitate. Rigiditate apare datorită faptului că aplicațiile, nucleul și alte informații sunt alocate și definite la compilare. În final rezultă o imagine de sistem de operare ce poate fi încărcată într-o memorie chiar și de tip ROM (Read Only Memory) [9].

Un alt sistem de operare în timp real folosit în multe aplicații prezente este cel dezvoltat de către Wind River System. Este vorba despre VxWorks, un sistem de operare folosit în principal pe platforma de dezvoltare Tornado [5]. Sistemul dispune de un nucleu ce folosește pentru planificarea proceselor o metodă preemptivă bazată pe priorități (256 nivele de priorități) dar pune la dispoziție și planificare de tip Round-Robin. De asemenea ca și în QNX un task își poate modifica prioritatea de execuție. Sistemul are o facilitate importantă și anume pune la dispoziție un sistem de servire a întreruperilor hardware. Pentru o deservire cât mai rapidă, în VxWorks, rutinele de tratare a întreruperilor rulează într-un context propriu diferit de cel al taskurilor obișnuite [10]. În ceea ce privește comunicarea între procese sistemul oferă mecanisme precum memorie partajată, semafoare pentru excludere mutuală și sincronizare, cozi de mesaje și structuri de tip pipe. Nucleul sistemului este scalabil putând fi adăugate noi facilități sau de asemenea eliminate unele dintre acestea. Software-ul aferent sistemului de operare pune la dispoziție modalități de măsurare a timpului de execuție al programelor dezvoltate pentru acesta precum și ustensile software de determinare a procentajului din timpul procesor ocupat de anumite procese [5]. Adiacent nucleului se mai oferă servicii de interfață grafică cu utilizatorul, suport pentru sisteme multiprocesor, mecanisme de gestionare a memoriei, suport Java și sistem de fișiere. Începând cu VxWorks 5.x sistemul respectă cerințele standardului POSIX 1003.1b. Există de asemenea și versiunea VxWorks AE ce respectă standardele POSIX și APEX.

QNX este un sistem de operare în timp real folosit pentru aplicații critice de la instrumente medicale la routere, aplicații de control ale unor procese tehnologice precum și în aplicații de control trafic aerian [5]. Sistemul oferă un mediu multitasking, multiuser, cu servicii de rețelistică, cu planificare preemptivă a proceselor precum și cu modalități de schimbare rapidă de context. QNX este un sistem ce respectă standardul POSIX 1003,1-2001 privind sistemele de operare în timp real. De asemenea sistemul este construit modular putând fi astfel scalat să ruleze atât pe dispozitive slabe în resurse, gen sisteme embedded, precum și pe sisteme mari, cum ar fi stații de lucru clasice [11].

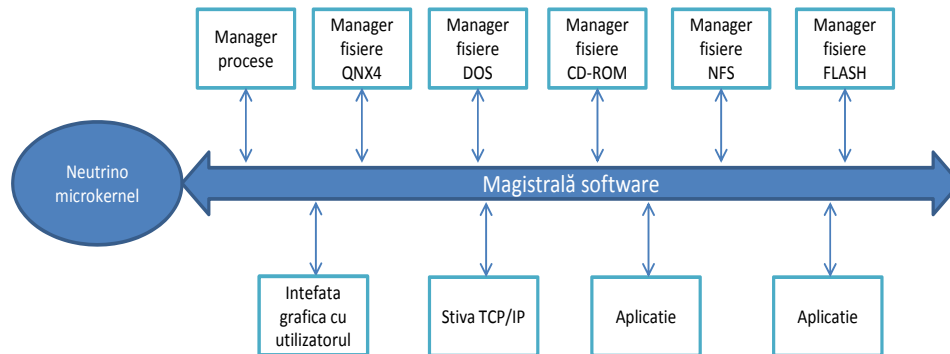


Figura 2-1 Arhitectura QNX

După cum se poate observa în Figura 2-1, sistemul constă într-un micronucleu, Neutrino microkernel ce controlează un grup de alte componente software [5]. Micronucleul, bazat pe nucleul UNIX, implementează doar funcțiile de bază de care are nevoie QNX pentru a funcționa (semnale, timere și planificatoare), alte funcții fiind opționale din acest punct de vedere și disponibile prin celelalte componente software. Avantajul acestei organizări este acela că în cazul unei componente defecte, aceasta poate fi restartată fără a fi nevoie o eventuală oprire a sistemului. De asemenea constituie un avantaj și faptul că orice componentă ce nu face parte din nucleu poate lipsi, reducând astfel semnificativ dimensiunea sistemului, putând fi astfel instalat pe orice dispozitiv sărac în resurse hardware. Printre mecanismele de timp real de care dispune nucleul putem aminti întreruperi încuibate ce permit tratarea întreruperilor în funcție de prioritate. Comunicarea între componente se face pe bază de mesaje și se formează între acestea un fel de magistrală software oferind facilitatea de care s-a vorbit mai sus prin care utilizatorul poate opri sau porni o componentă fără a afecta întregul sistem.

Sistemul QNX suportă implementări pentru o mare varietate de platforme: x86, ARM, MIPS, PowerPC, SH-4 precum și variante ale acestora de tip multiprocesor. Din punct de vedere al metodelor de planificare al proceselor pentru rulare, sistemul suportă un număr semnificativ de algoritmi: planificare de tip FIFO, planificare de tip Round-Robin, planificare adaptivă (dacă procesul își consumă cuanta de timp prioritatea îi este decrementată), planificare sporadică. Sistemul nu dispune de mecanisme de planificare a taskurilor cu execuție fixă în cadrul perioadei, așadar dacă se dorește o execuție periodică se pot folosi timerele puse la dispoziție precum și celelalte mecanisme.

QNX pune la dispoziția programatorului modalități prin care anumite operații pot deveni atomice minimizând astfel execuție de cod în plus pentru dezactivarea întreruperilor. Printre aceste operații se numără: adunare, scădere, setarea și resetarea unui bit dintr-un registru, complementarea unui bit [8].

Respectând standardul POSIX sistemul QNX folosește ca și mecanisme de sincronizare a proceselor și de comunicare între procese, mecanismele sugerate de acest standard.

Pe lângă aceste facilități se mai oferă și interfață grafică cu utilizatorul (cu impact foarte mic asupra performanței procesorului), servicii de rețelistică (stiva TCP/IP), sistem de fișiere, posibilitate de depanare a programelor. Ca limbaje de programare QNX suportă C și C++ și prin mediul Visual Micro-Edition chiar și limbajul Java. Se pune

la dispoziție și un mediu pentru programare vizuală pentru realizarea de interfețe grafice.

Tot pentru dispozitivele embedded sărace în resurse hardware este și MaRTE OS (Minimal Real-Time Operating System for Embedded Applications) dezvoltat de către Mario Aldea Rivas și Michael Gonzalez Harbour la universitatea din Cantabria, Spania. Sistemul este scris în mare parte în Ada și urmărește standardul POSIX.13, un subset al standardului POSIX orientat pe cerințele de timp real ale aplicațiilor. Sistemul deși încă este în lucru, suportă dezvoltarea de aplicații în Ada, C și C++ dar ținând cont că sistemul urmărește subsetul POSIX.13 nu este fezabil pentru aplicațiile critice de tip hard real time [12, 13].

La ora actuală majoritatea rețelelor wireless de senzori au ca și principală componentă software un sistem de operare în timp real dezvoltat de University of California Berkeley. Inițial a fost dezvoltat pentru modulele proprii dar apoi a început a fi folosit și de alte universități și cercetători. Este vorba de sistemul TinyOS care după cum îi spune și numele este format dintr-un nucleu de mici dimensiuni realizat special pentru sisteme embedded. Majoritatea studiilor protocoalelor de comunicație wireless în timp real sunt realizate folosind acest TinyOS.

Deși practic nu este un sistem de operare în timp real precum cele prezentate până acum, sistemul TinyOS prezintă anumite caracteristici de planificări de procese precum și suport pentru a proiecta anumite procese ce au nevoie de rulare în timp real. TinyOS este un sistem de operare bazat pe evenimente și este folosit pe sisteme embedded orientate pe rețele de senzori. Sistemul este dezvoltat pentru a susține execuția concurentă a proceselor necesare controlului rețelelor de senzori [14].

În ceea ce privește planificatorul componentelor software necritice, acesta este structurat pe două nivele: un planificator pentru task-uri și unul pentru evenimente. Planificatorul este de fapt o simplă stivă FIFO (First in First out) și se bazează pe regula că un task nu poate întrerupe pe altul, dar un eveniment poate întrerupe un task, și evenimentele se pot întrerupe între ele, în funcție de prioritatea lor de tratare a întreruperii (un eveniment mai prioritar îl poate întrerupe pe unul mai puțin prioritar).

Este de asemenea important de menționat și sistemul de operare pentru module încorporate FreeRTOS. Acesta este un sistem de operare în real conceput de către Richard Barry și dezvoltat de către Real Time Engineers Ltd. Sistemul suportă un număr de 30 de arhitecturi hardware de procesoare devenind astfel extrem de popular, lucru dovedit de numărul mare de descărcări (peste 113.000) în anul 2014. Principalele avantaje sunt în primul rând gratuitatea acestuia sub licență GPL, amprenta mică de memorie (aprox. 4-9 KB) și bineînțeles număr mare de arhitecturi pentru care este disponibil. FreeRTOS are un grad mare de portabilitate, acesta fiind dezvoltat în cea mai mare parte în limbajul C. Din punct de vedere al timpului real, acesta este un sistem de operare bazat pe priorități și pe întreruperi de ceas, cu grad sporit de predictibilitate. Un alt avantaj al sistemului este acela că este extrem de bine documentat și foarte ușor de folosit datorită API-ului pus la dispoziție. Totuși un important dezavantaj rămâne numărul limitat de politici de planificare pentru task-uri. Producătorii au dezvoltat de asemenea și variante comerciale ale acestuia cum ar fi SafeRTOS și OpenRTOS [15-18].

### 2.3 Nucleul HARETICK

HARETICK (Hard Real Time Compact Kernel) este un nucleu de operare hard real time dezvoltat în cadrul Laboratoarelor de Prelucrare Numerică a Semnalelor (DSPLabs) din Departamentul de Calculatoare din cadrul Facultății de Automatică și Calculatoare, Universitatea Politehnică Timișoara. Este necesară o scurtă trecere în revistă a acestui nucleu deoarece întreaga activitate de cercetare-dezvoltare realizată de autor este strâns legată de acesta.

HARETICK este un nucleu de sistem de operare timp real, hibrid, single-user și multitasking ce permite existența a două contexte concurente de execuție a proceselor: un context non-preemptiv Hard Real Time (HRT) pentru procese perfect periodice cu cerințe stricte de timp, un context timp real preemptiv și un context Soft Real Time (SRT) destinat task-urilor lejere. Prioritatea cea mai mare în timpul execuției o are contextul HRT iar sistemul poate garanta că toate procesele de tip HRT planificate pentru execuție își vor respecta cerințele de timp chiar și în cazul de execuție cele mai defavorabile. O consecință a acestui aspect este faptul că sistemul permite o singură întrerupere și anume întreruperea de ceas de timp real – Real Time Clock (RTC) [19].

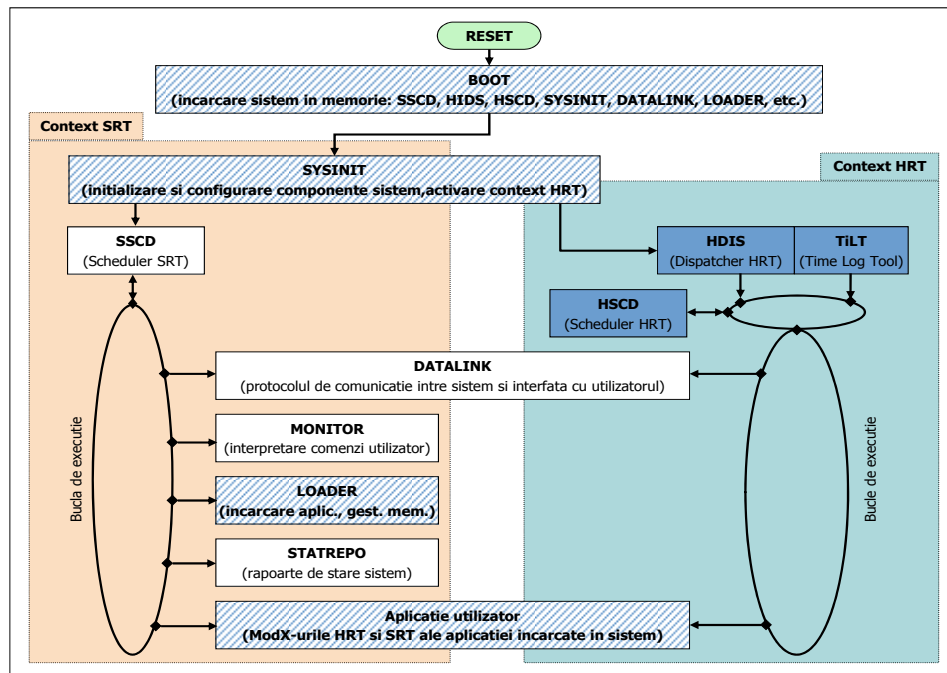


Figura 2-2 Structura HARETICK (sursa: [6])

Unul din principalele avantaje ale sistemului este arhitectura modulară a acestuia (Figura 2-2). În acest sens, o caracteristică importantă a acestuia este simplitatea și faptul că s-a încercat pe cât posibil să se reducă dimensiunea nucleului. De aceea pentru contextul strict de execuție nu se realizează nici un fel de analize de planificabilitate și fezabilitate și de asemenea sistemul nu dispune de exemplu de anumite mecanisme de protecție împotriva unelor operații realizate de aplicații

utilizator. HARETICK, fiind construit pentru sisteme embedded, merge pe ideea că dezvoltatorul de aplicații poartă întreaga răspundere pentru funcționarea în parametrii a sistemului. De asemenea *"componentele software cu specificații temporale stricte (HRT) ale sistemului și ale aplicațiilor dezvoltate pentru acesta sunt proiectate și implementate sub formă de module software independente, care comunică între ele prin intermediul parametrilor de intrare/ieșire și al variabilelor globale"* [6].

În ceea ce privește gestionarea timpului sistemul folosește structuri specializate iar baza de timp este implementată de ceasul timp-real care în același timp este și singura sursă predictibilă de întrerupere din sistem.

Inițial în HARETICK erau prezente două tipuri de task-uri: task-uri HRT, Hard Real Time, denumite și ModX (Modul eXecutabil) și task-uri SRT. Prin aceasta sistemul își atribuie caracteristica de sistem de operare hibrid. Comutarea între task-uri se realizează prin componente software dedicate. Task-urile HRT sunt planificate într-o manieră non-preemptivă pe baza algoritmului de planificare FENP (Fixed Execution Non Preemptive). Acest algoritm planifică taskurile cu o execuție fixă în cadrul perioadei de planificare adăugând astfel suportul pentru execuția taskurilor ce necesită a fi perfect periodice [20]. Sistemul este încărcat pe platforma pe care va fi executat după ce în prealabil se face o analiză statică din mai multe puncte de vedere: analiză a planificabilității task-urile, a memoriei utilizate, a resurselor folosite. După ce din analiză rezultă că întregul sistem poate rula corect conform cerințelor se creează o imagine sistem care este apoi încărcată pe procesorul gazdă.

Despre comunicația dintre task-uri se poate spune că sistemul HARETICK oferă această facilitate prin intermediul parametrilor de intrare/ieșire și al variabilelor globale. Deoarece ModX-urile se execută în manieră non-preemptivă și implementează astfel operații de tip atomic, se elimină necesitatea implementării de tehnici pentru sincronizarea controlului acceselor concurente la resursele partajate ale nucleului precum și problemele legate de inversarea priorității [6]. Totuși probleme apar în cazul în care se dorește comunicarea între taskuri din contexte diferite de execuție. În acest caz sunt necesare anumite mecanisme de sincronizare și de controlul accesului la resurse partajate. Acest lucru s-a realizat prin intermediul memoriei partajate gradate de anumite variabile cu rol asemănător unui mutex [19].

În final nucleului HARETICK i se adaugă un nou context de execuție. Acesta este de asemenea un context timp real, care garantează și el că fiecare task își va respecta deadline-ul iar întregul sistem va funcționa conform parametrilor impuși. Acest context este de tip preemptiv putând fi întrerupt doar de către contextul HRT care își menține prioritatea superioară în sistem. În cadrul acestui nou context taskurile sunt planificate în maniera EDF (Earliest Deadline First). În ordinea priorităților se află deci contextul HRT cu cea mai mare prioritate în sistem, contextul EDF cu o prioritate mai mică decât HRT și contextul SRT care își menține cea mai scăzută prioritate în sistem [21, 22]. O imagine mai clară se poate găsi în Figura 2-3.

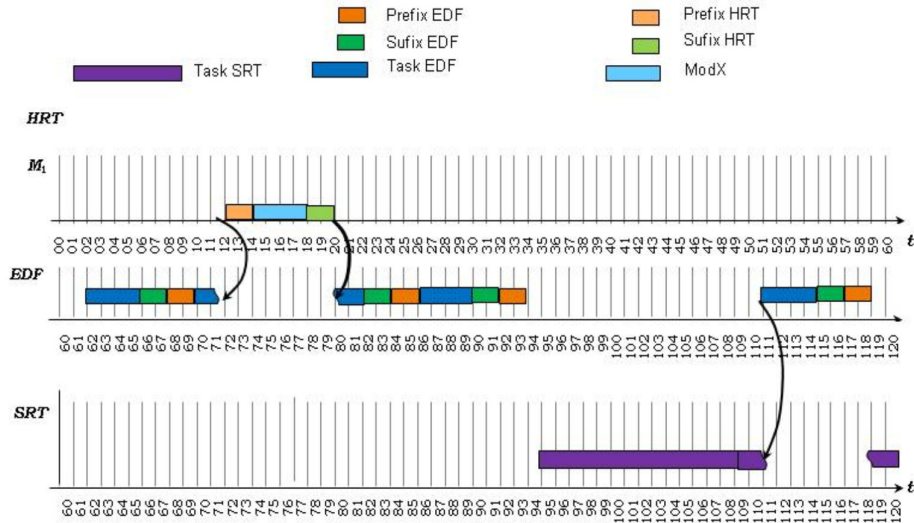


Figura 2-3 Contextele de execuție ale nucleului HARETICK (sursa [21])

Adăugarea acestui nou context oferă nucleului HARETICK un mai mare grad de flexibilitate din punct de vedere al numărului de taskuri și al cerințelor de timp impuse de acestea.

Inițial HARETICK a fost implementat pe mai multe platforme embedded printre care procesoare din familia Motorola respectiv DSP56303EVM [23], Motorola MSC8101ADS [24]. Recent HARETICK a fost implementat într-o variantă mai extinsă pe procesoare din familia ARM7 NXP Philips LPC2xxx [25].

Fiind o arhitectură intens folosită în cadrul acțiunilor de cercetare dezvoltare prezentate de autor în rândurile următoare se vor prezenta câteva aspecte legate de arhitectura ARM7 respectiv familia de procesoare NXP Philips LPC2xxx.

Procesoarele ce au ca fundament ARM7 sunt procesoare de tip RISC (Reduced Instruction Set Computer) bazate pe arhitectură de tip Von Neumann sau Harvard, cu structuri de tip pipeline (bandă de asamblare) ce au ca și caracteristică de bază simplitatea fără a pierde din performanță. De asemenea procesoarele bazate pe această arhitectură oferă performanțe destul de ridicate fără o mare creștere a consumului de energie. Pentru implementarea HARETICK s-a ales o variantă îmbunătățită a clasicei arhitecturi ARM7 și anume ARM7TDMI-S. Aceasta aduce în plus modul Thumb (caz în care se înjumătățește dimensiunea unei instrucții cu afectând performanța), facilități în plus legate de depanarea codului precum și o unitate aritmetică și logică mai dezvoltată și mai rapidă. [26]

Principalele caracteristici care au facilitat implementarea HARETICK pe această arhitectură sunt:

- Arhitectura RISC
- Pipeline simplu cu 3 stagii
- Set extins de registre (16 registre generale)
- Moduri de lucru
- Întrerupere cu deservire rapidă FIQ (Fast Interrupt)
- Set de registre diferit în funcție de modul de lucru



Dintre aceste caracteristici modurile de lucru împreună cu unele seturi de registre diferite în funcție de modul de lucru se dovedesc extrem de importante pentru implementarea unui sistem de operare în timp real. Spre exemplu modul de lucru FIQ conține propriul subset de registre ceea ce ajută la eliminarea salvării de context în momentul în care HARETICK trebuie să schimbe contextul HRT. De asemenea fiecare mod de lucru dispune de propria stivă gestionată prin mecanisme hardware (seturi de registre de stivă diferite).

Totuși la prima vedere arhitectura pare să ofere multe facilități în implementarea unui sistem de operare în timp real, există unele caracteristici care îngreunează scăzând gradul de predictibilitate. Astfel anumite mecanisme ale arhitecturii cum ar fi componenta MAM (Memory Accelerator Module), responsabilă cu îmbunătățirea accesului la memoria FLASH internă, duce la apariția unor timpii de acces ce nu pot fi calculați cu exactitate din cauza folosirii unor mecanisme de predicție. De asemenea ținând cont că un esențial parametru de timp în sisteme în timp real este WCET-ul, în cazul acestei arhitecturi setul de instrucțiuni are un grad scăzut de ortogonalitate, îngreunând procesul de calculare al acestui parametru [27].

## **2.4 Generalități despre dezvoltarea de aplicații pentru sisteme hard real time**

După cum s-a enunțat și mai sus, scopul acestei activități de cercetare este acela de a ajunge la o comunicare wireless în timp real între nodurile unei rețele de senzori ce realizează aplicații critice cu cerințe stricte de timp. Pentru ca acest lucru să se poată realiza nu este suficient ca pe fiecare nod să ruleze un sistem de operare hard real time ci trebuie ca și aplicațiile ce rulează folosind acest sistem de operare să fie dezvoltate și implementate după anumite principii menite să le integreze într-un sistem de operare de acest gen. Nu orice aplicație poate rula în condiții stricte de timp. De asemenea adaptarea unei aplicații ce nu a fost concepută pentru sisteme hard real time nu este posibilă și o reproiectare din punct de vedere software este necesară.

Una dintre principalele deosebiri între realizarea unei aplicații clasice față de realizarea unei aplicații destinată unui sistem hard real time este aceea că în cazul aplicației clasice pe lângă funcționarea corectă a acesteia se urmărește de obicei a se obține viteze cât mai mari, iar codul se urmărește a fi cât mai scalabil urmărind diverse tehnici consacrate de implementare; în cazul aplicațiilor destinate pentru sisteme hard real time nu viteza de execuție sau codul scalabil sunt vizate ci se urmărește predictibilitatea. Pentru a se realiza acest lucru o importantă practică este aceea de a se programa, identifica și calcula cazurile cele mai defavorabile. Această practică se spune că este programare orientată pe WCET [28]. Calcularea acestui parametru impune de multe ori o dificultate extrem de ridicată și se poate realiza în situațiile în care durata de execuție a codului depinde foarte puțin de datele de intrare sau în situația în care nu este posibil acest lucru să se trateze mai multe seturi de intrări la fel. Se demonstrează că o practică bună este aceea de a folosi biblioteci și bucați de cod care deja au un WCET calculat.

De asemenea comunicarea între procese în cazul sistemelor de timp real are un impact puternic asupra predictibilității și timpului de execuție a aplicațiilor. Acest lucru este clar dovedit în [19, 29] unde aplicându-se o proiectare și implementare bazată pe cazurile cele mai defavorabile, pe WCET, se obține atât un sistem de comunicare între

taskurile timp real din contexte diferite bazat pe memorie partajată și protejată de variabile de gardare, precum și un driver perfect predictibil, ce rulează în context HRT, pentru comunicarea de tip SPI (Serial Peripheral Interface).

Rezultate asemănătoare s-au obținut și în [30] unde s-a realizat un driver de asemenea perfect predictibil pentru a gestiona în manieră hard real time module de comunicare wireless de tip XBEE [31]. Acest driver s-a folosit în cadrul unei rețele de senzori pentru a se adăuga facilitatea de comunicare wireless folosind acest tip de module. Problema acestei situații este aceea că deși driverul din cadrul fiecărui nod, pentru acest tip de module wireless, este hard real-time, perfect predictibil, totuși modulele wireless în sine nu oferă aceste garanții bazându-se pe clasicul protocol de acces la mediu 802.15.4 iar nivelele superioare de comunicație fiind asigurate de stiva ZigBee [32].

## 2.5 Concluzii

Principalul domeniu de interes al acestei lucrări este reprezentat de sistemele în timp în real și mai mult decât atât rețelele de senzori care să funcționeze ca un sistem în timp real. Cea mai importantă caracteristică a unui sistem timp real este aceea că nu doar rezultatul corect al unei operații este important ci și intervalul de timp în care sistemul furnizează acel rezultat. Asemănător se poate spune și despre rețelele de senzori în timp real. Nu doar operațiunea sau procesul în sine pe care îl monitorizează sau controlează o rețea de senzori este important ci și timpul de răspuns al rețelei de senzori.

O rețea de senzori este formată din microsisteme digitale cu capabilități computaționale limitate, cu posibilitate de comunicare și cu capacități senzoriale diferite în funcție de aplicație.

Primul pas spre a obține o rețea de senzori în timp real este asigurarea unei funcționări în timp real la nivelul fiecărui nod al rețelei. Pentru aceasta, cea mai importantă componentă este sistemul de operare ce rulează pe fiecare nod al rețelei. Acest sistem de operare trebuie să ruleze după principiile operării în timp real. După cum s-a arătat în acest capitol, există o mulțime de sisteme de operare în timp real orientate spre sistemele încorporate. Unul dintre aceste sisteme de operare este și nucleul HARETICK care a fost special dezvoltat pentru a servi operării în context hard real-time pentru sisteme încorporate cum sunt și nodurile unei rețele de senzori.

Următorul pas spre o funcționare timp real a unei rețele de senzori este folosirea unor tehnici de proiectare, dezvoltare și implementare a tuturor modulelor software ce rulează pe nodurile rețelei, tehnici care să fie orientate pe principiile operării în timp real. Programarea pe sistemele în timp real este diferită de programarea clasică prin faptul că principala caracteristică a modului software rezultat trebuie să fie predictibilitatea.

După ce se constată că la nivelul unui nod al rețelei componentele software funcționează în manieră timp real, atunci se pune problema ca întreaga rețea de senzori să funcționeze în aceeași manieră. Pentru ca acest lucru să se poată realiza este necesar ca și comunicarea dintre nodurile rețelei să se realizeze după principiile operării în timp real. Acest aspect face principalul obiect al acestei lucrări.

## **3 Rețele de senzori wireless**

Un alt important domeniu abordat de prezenta lucrare este reprezentat de rețelele de senzori în special cele în care nodurile rețelei au capabilități de comunicare fără fir. Capitolul debutează cu o descriere a domeniului de rețele de senzori și prezintă considerații teoretice generale despre acestea punându-se accent pe partea de comunicație.

Rețelele de senzori sunt prezente într-o gamă largă de aplicații în ultima perioadă iar câteva dintre principalele aplicații și implementări existente sunt prezentate de asemenea în acest capitol. În ultima parte a acestei secțiuni se prezintă nivelul din stiva de comunicație responsabil cu rutarea informației evidențiindu-se progresul realizat de cercetători în ultima perioadă în acest domeniu.

### **3.1 Noțiuni introductive**

O posibilă definiție a unei rețele de senzori ar putea fi: o rețea de senzori wireless – WSN (Wireless Sensor Network) - este formată dintr-un număr mare de senzori inteligenți, referiți în continuare ca și noduri, dispuși pe o suprafață în exterior sau în interiorul unei clădiri, cu capacități de comunicare wireless, mobili sau fixi, care prin acțiuni colaborative formează o rețea de senzori cu scopul de a implementa o anumită aplicație ca un tot unitar [33].

Un nod are ca și componentă centrală un microcontroller de obicei cu putere mică de calcul, cu consum redus de energie electrică și care se obișnuiește să aibă facilități de managementul consumului de energie (power management). Este de preferat să se folosească arhitecturi optimizate pentru consumul redus de energie, ținând cont că în ultima perioadă s-au dezvoltat mecanisme precum DVS (Dynamic Voltage Scaling) sau DMS (Dynamic Modulation Scaling).

Microcontrolerul asigură întreaga funcționare a nodului. La acesta sunt conectate celelalte blocuri după cum se poate observa în Figura 3-1.

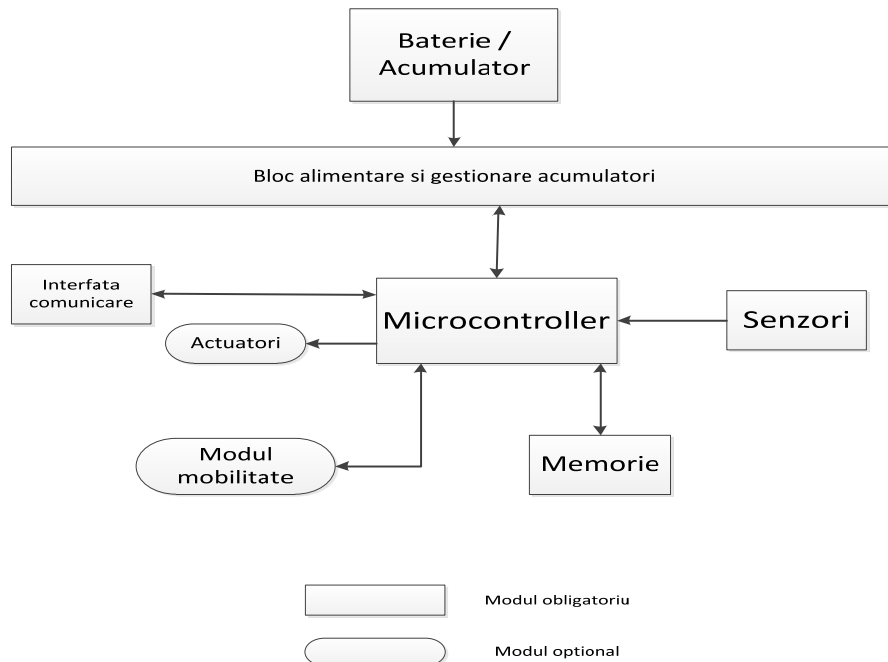


Figura 3-1 Schema bloc a unui nod dintr-o rețea de senzori

O configurație minimală a unui nod este formată dintr-un microcontroller, sursă de alimentare (baterie / acumulator), senzori și interfață de comunicare cu sau fără fir în funcție de tipul rețelei. Într-o asemenea configurație minimă rețeaua de senzori poate fi folosită în principiu doar pentru monitorizare într-o topologie fixă cu un grad scăzut de dinamism. În schimb, în cazul în care se adaugă diferite module opționale cum ar fi cele din Figura 3-1 caracteristicile rețelei se schimbă. Spre exemplu, în situația adăugării unor actuatori rețeaua primește și facilitate de control nu doar de monitorizare, iar în cazul adăugării și a unui modul de mobilitate rețeaua devine puternic dinamică. Deși o astfel de rețea este de dorit în multe situații, dinamismul introduce și o serie de probleme atât pe parte de mobilitate și consum de energie cât mai ales pe partea de comunicație wireless.

O astfel de rețea este formată dintr-un număr foarte mare de noduri și din acest considerent o caracteristică ce trebuie îndeplinită de fiecare nod este costul. Se optează de obicei pe o soluție cu un preț cât mai scăzut dat fiind numărul mare de noduri. În același timp trebuie să se țină cont de faptul că în marea majoritate a situațiilor nodurile sunt alimentate pe baterie și se dorește ca durata de viață a unui nod să fie cât mai lungă. Pentru a se realiza acest lucru nu doar că pe parte hardware ar trebui să se folosească componente care sunt optimizate pentru consum redus de energie dar și din punct de vedere software trebuie să se țină cont de tehnici de programare și proiectare software orientate pe reducerea consumului de energie. În cazul aplicațiilor în timp real toate tehnicile atât de reducere a consumului de energie cât și cele de reducere costului nu trebuie să afecteze cerințele stricte de timp impuse de aplicație [34, 35].

Rețelele wireless de senzori se pot clasifica după mai multe criterii [36]:

- dimensiunea rețelei sau distanța dintre nodul bază (în cazul unei rețele organizate după o topologie coordonată) și cele mai îndepărtate noduri ale

rețelei – rețelele pot fi single-hop sau multi-hop. În primul caz, single-hop, fiecare nod comunică doar cu coordonatorul rețelei. O astfel de rețea este de obicei de mici dimensiuni dar se caracterizează prin simplitate deoarece nu este nevoie de algoritmi complecși care să realizeze rutarea informației. Nodurile comunică doar atunci când au ceva de transmis nodului coordonator și astfel își economisesc mult din energia din baterie. În cazul rețelelor multi-hop puține noduri sunt în raza de comunicație a coordonatorului rețelei și astfel ele nu își pot transmite datele direct la coordonator ci doar prin noduri intermediare fiind necesară implementarea unor algoritmi specializați care să realizeze rutarea informației de la nodul transmițător până la nodul receptor, de obicei coordonatorul rețelei.

- densitatea rețelei și dependența de date – rețele cu agregare sau fără agregare. În sistemele fără agregare fiecare nod transmite coordonatorului toate datele pe care le deține fără să se aplice vreo transformare asupra lor. În cazul unei rețele cu multe noduri acest lucru generează trafic ridicat ceea ce implicit duce la scăderea duratei de viață a rețelei în urma consumului mare a fiecărui nod. În sistemele cu agregare nodurile sunt grupate de obicei în funcție de localizare, datele sunt agregate și prelucrate local iar coordonatorului i se transmite doar un rezultat și astfel traficul scade considerabil. De obicei această prelucrare a datelor se rezuma ori la o medie ori la eliminarea datelor redundante.
- Modul de distribuție a nodurilor – rețele deterministe sau dinamice. Rețelele deterministe din punct de vedere al distribuției nodurilor sunt acele rețele în care plasarea nodurilor este cunoscută și fixă. În această situație întreaga gestiune a rețelei devine mult mai facilă, mulți algoritmi (de exemplu de rutare, localizare) ne mai fiind necesari. O astfel de rețea are un grad mare de rigiditate. Rețelele dinamice din punct de vedere al distribuției nodurilor sunt acele rețele unde nu se cunoaște de la început poziția fiecărui nod. Motivul necunoașterii poziției este fie plasarea în mod aleatoriu a nodurilor, fie nodurile au și capacitate de mobilitate. În acest caz toate operațiunile realizate în cadrul rețelei se complică simțitor dar se oferă un mare grad de flexibilitate și scalabilitate.
- politica de control – cu autoconfigurare sau fără autoconfigurare. În cazul rețelelor fără autoconfigurare nodurile nu au capacitatea de a se auto-organiza ci au nevoie de o entitate de control care să comande ce să facă la fiecare pas. Acesta politică de control nu poate fi folosită decât pentru rețele de mici dimensiuni și unde nu se dorește scalabilitate. Cea de-a doua politică de organizare și control ce se referă la rețele cu autoconfigurare care are rolul de permite nodurilor rețelei să se organizeze atât din punct de vedere al comunicării cât și din punct de vedere al sarcinilor. În acest caz, o entitate superioară de control nu trebuie decât să traseze sarcina în linii mari iar nodurile rețele se auto organizează și, formând un mediu colaborativ, decid în colectiv fiecare pas.

Tot în funcție de politica de control se mai disting două tipuri de rețele asemănătoare celor cu autoconfigurare și fără autoconfigurare. Este vorba de categoria rețelelor ad-hoc sau a celor gestionate de un coordonator.

După cum se va vedea în cuprinsul acestei lucrări majoritatea activităților de cercetare au ca obiect studiul rețelelor wireless de senzori coordonate. În această configurație există noduri cu funcții reduse în ceea ce privește comunicarea și noduri mai evolute care au rolul de a le conduce pe celelalte formându-se astfel o structură ierarhizată de noduri. În general un nod cu funcții evolute de comunicare împreună cu nodurile pe care le conduce formează un *cluster*.

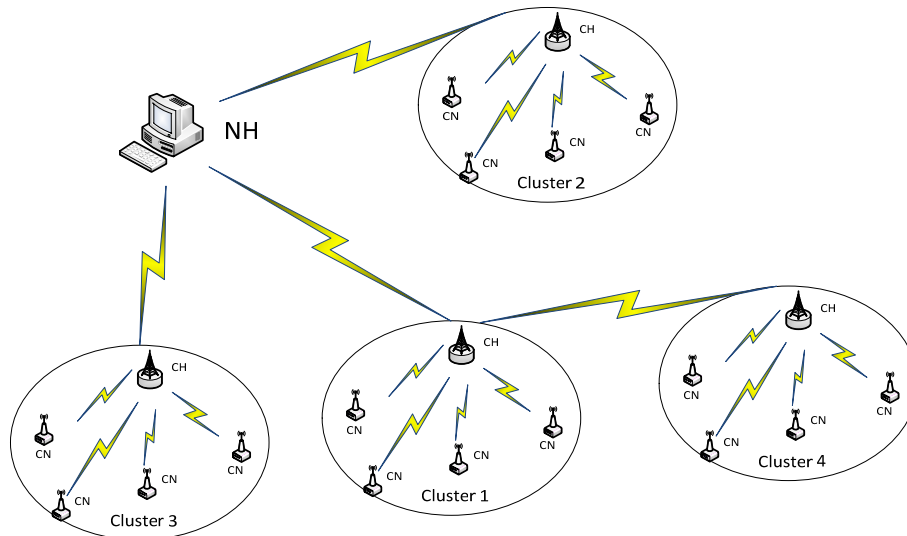


Figura 3-2 Exemplu rețea organizată pe clustere

În Figura 3-2 se prezintă un exemplu de rețea organizată în structuri de clustere. Nodurile desemnate cu CN (Cluster Node) reprezintă noduri de tip frunză ale unui cluster. Acesta nu pot comunica decât cu nod de tip CH (Cluster Head) și în concluzie au capabilități reduse de comunicare. Fiecare nod CN trebuie să fie în raza de acoperire a nodului CH ce conduce clusterul din care face parte și în principiu nu poate comunica decât cu acesta. Așadar un cluster este format dintr-un coordonator sau conducător de cluster (CH) și un număr de noduri de tip CN. Coordonatorul unui cluster este de obicei un nod mai bogat în resurse decât nodurile pe care le conduce putând implementa nu doar funcții de bază de comunicare ci și funcții mai evaluate cum ar fi rutarea informației. Deși sunt mai bogate în resurse, nodurile de tip CH sunt și cele la care durata de viață a bateriei poate să scadă mult mai repede față de celelalte noduri deoarece acestea având și funcția de rutare de informație consumă multă energie electrică pentru comunicarea pe radio. În figura prezentată mai sus se observă cum clusterul 4 nu este în directă rază de acoperire a conducătorului rețelei și astfel CH1 trebuie să realizeze rutarea informației între conducătorul rețelei și CH4.

O rețea de senzori într-o asemenea structură este formată din mai multe clustere, toate fiind conduse de către un coordonator de rețea semnificat în figură prin NH (Network Head). Conducătorul rețelei comunică doar cu conducătoarele de clustere și nu direct cu nodurile pe care acestea le gestionează. De obicei acest tip de nod dispune de multe resurse cum ar fi energia electrică, memorie, putere de calculație. În unele situații NH este chiar un calculator clasic și nu neapărat un sistem embedded.

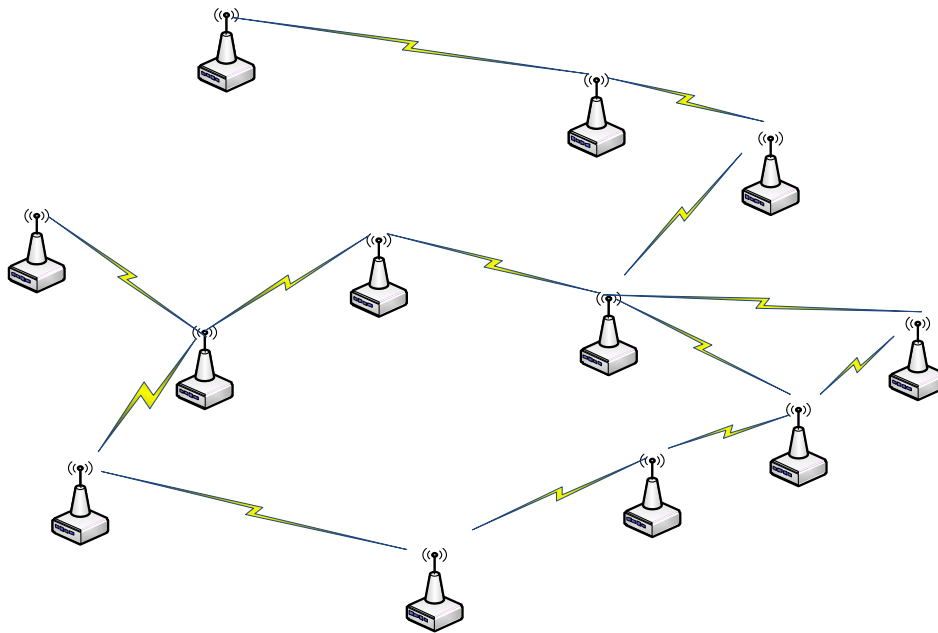


Figura 3-3 Exemplu rețea ad-hoc

Rețelele de tip ad-hoc caracterizează cel mai mult ideea de rețea wireless de senzori în general. Rețelele de acest tip sunt acelea care nu au nevoie de un coordonator pentru a-și îndeplini funcția. Prin procese colaborative senzorii care în principiu sunt dispuși aleatoriu se auto-organizează pentru a realiza funcția destinată. Acest tip de rețele prezintă un grad mare de scalabilitate și adaptabilitate. Există și aici 2 situații de organizare. O situație în care toate nodurile sunt în aceeași zonă de acoperire, adică orice nod poate comunica în mod direct cu oricare alt nod. De obicei această situație este destul de greu de obținut. Un alt caz este acela când raza de acoperire a nodurilor este mică și astfel orice nod nu poate comunica direct cu orice alt nod astfel apărând situația când unele noduri trebuie să realizeze și funcții de rutare a informației. În acest caz rețeaua are facilități întâlnite la o rețea de tip mesh [37, 38].

În ceea ce privește protocoalele de comunicație wireless din rețelele de senzori, în general acestea sunt organizate pe nivele fiind puternic influențate de clasicul model de referință OSI (Open System Interconnection) [39]:

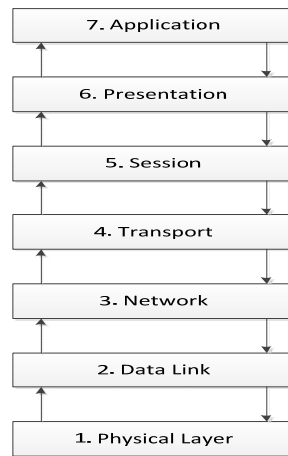


Figura 3-4 Modelul de referință OSI

Privind stiva de referință OSI de jos în sus, primul nivel al unei stive de protocoale ce respectă acest model este reprezentat de nivelul fizic (Physical Layer). Acest nivel precizează caracteristicile electrice, mecanice și procedurale ce au rolul de a realiza conexiunea fizică între mai multe module de comunicație, pe un anumit mediu fizic. Aceste protocoale se mai numesc și protocoale de linie și definesc caracteristicile canalului fizic pe care se efectuează comunicarea. Unul dintre protocoalele de linie cele mai întâlnite este și protocolul RS-232 [40] standardizat de EIA (Electronic Industries Association). Acest arhicunoscut protocol serial este încă folosit cu mult succes în industrie, în unele situații fiind de neînlocuit.

Nivelul 2 al modelului OSI este reprezentat de nivelul numit data-link. Scopul acestui nivel este de a asigura o legătură punct la punct între 2 noduri ale rețelei conectate în mod direct. De asemenea un protocol de nivel 2 trebuie să asigure și secvențierea corectă a pachetelor precum și detecția erorilor apărute la nivelul fizic iar în unele cazuri, când este posibil, să realizeze și corecția acestora. Din această categorie fac parte protocoale precum HDLC (High-Level Data Link Control) [41] sau bine-cunoscutul protocol Ethernet [42].

Nivelul rețea (network layer) este responsabil cu asigurarea transferului unor secvențe (pachete) de date, de dimensiuni variabile (datagrame) între două noduri ale rețelei fără a fi o legătură directă între ele. Nivelul implementează astfel o comunicare de tip end-to-end. Este important de menționat faptul că protocoalele din acest nivel nu este necesar să garanteze un transfer fiabil. În acest nivel OSI se încadrează protocoalele de rutare a informației precum și partea de adresare a elementelor din rețea. Un exemplu reprezentativ al acestui nivel ar fi protocolul IP (Internet Protocol).

Următorul nivel al modelului OSI este nivelul transport care are principalul rol de a garanta o comunicare fiabilă între nodurile rețelei prin metode precum controlul fluxului, controlul retransmișilor, detecții și corecții de erori apărute în protocoalele de pe nivelele inferioare precum și mecanisme de tip ACK/NACK. Protocolul cel mai reprezentativ al acestui nivel este protocolul TCP (Transmit Control Protocol) folosit de obicei împreună cu protocolul IP ca protocol rețea.

Nivelul de pe poziția 5 a modelului OSI reprezintă nivelul sesiune (session layer). Protocoalele de pe acest nivel se ocupă cu gestiunea conexiunilor dintre entitățile rețelei. Protocoalele care implementează acest nivel sunt protocoale orientate pe



conexiune. Principalele roluri ale acestui nivel sunt: realizarea, întreținerea, menținerea și încheierea conexiunilor. De asemenea protocoalele din acest nivel trebuie să distingă și cauza încheierii unei conexiuni pentru a încerca reluarea conexiunii în cazul în care aceasta nu s-a încheiat în urma unei cereri specifice din partea unui alt nivel.

Următoarele două nivele sunt nivel specifice aplicației. Nivelul 6, nivelul prezentare este cel care oferă nivelului aplicație sintaxa necesară încapsulării datelor. Tot nivelul prezentare se ocupă și de criptarea datelor atunci când aplicația cere acest lucru. Ultimul nivel al stivei OSI, nivelul aplicație, este cel mai apropiat de utilizatorul final și este puternic dependent de necesitățile utilizatorului. Exemple de protocoale de nivel aplicație: HTTP (Hypertext Transfer Protocol), DNS (Domain name System), Telnet.

Important de menționat despre modelul OSI este faptul că protocoalele de pe fiecare nivel sunt puternic dependente de protocoalele de pe nivelele inferioare. Spre exemplu, un protocol de pe nivelul 3, un protocol de rutarea informației, care prezintă un grad mare de predictibilitate și care depinde de un protocol de acces la mediu de nivel 2 și 1 care nu are mecanisme ce îi conferă predictibilitate atunci nici protocolul de rutare în cauză nu va fi predictibil.

Dat fiind faptul că nodurile unei rețele de senzori wireless sunt sisteme embedded sărace în resurse, protocoalele de comunicație nu respectă ad litteram acest model atât din considerentul că în acest caz poate nu sunt necesare toate nivelele cât și din cauza resurselor puține. De asemenea protocoalele ce se folosesc în acest tip de rețele trebuie să țină cont de resursele puține oferite de platforma ceea ce face imposibilă folosirea unor protocoale existente pentru sisteme de calcul clasice, protocoale ce pot rezolva multe din problemele dintr-o rețea cum ar fi spre exemplu obiectul acestei lucrări, comunicarea în timp real. Un astfel de exemplu de protocol ce oferă garanții de timp real dar nu este aplicabil în rețelele de senzori wireless compuse din sisteme embedded este [43] în care computația necesară algoritmului este rezolvată de un procesor cu două nuclee la o viteză de 1.8 GHz. Aceste resurse nu pot fi întâlnite pe nodurile unei rețele de senzori cel puțin din considerente de consum de energie electrică și preț.

Important de menționat este faptul că există o legătură puternică între rețelele de senzori cu comunicare pe fir și cele cu comunicarea wireless. Cel puțin teoretic unele idei folosite cu succes în cazul comunicării pe fir cum ar fi protocolul hard real time PARSECS (Predictable Architecture for Sensor Communication Systems) [44] se pot aplica și în cazul comunicării wireless. Totuși există o mare diferență între mediul electric și mediul wireless de care trebuie să se țină cont.

Principalele nivele care se regăsesc în protocoalele de comunicație din cadrul unei rețele de senzori wireless sunt:

- nivelul fizic (Physical Layer) – se ocupă cu accesul la mediul fizic de comunicație. Acest nivel este obligatoriu să fie implementat pe un nod al unei rețele. Aici se regăsesc cu precădere părți din protocoalele de tip MAC
- nivelul data link – acest nivel asigură comunicația de tip punct la punct între 2 noduri ale rețelei aflate în aceeași arie de acoperire, în acest caz realizându-se comunicație directă între cele 2 noduri. Aici se regăsesc de asemenea protocoale de tip MAC cât și protocoale pentru detecția și corecția erorilor și controlul retransmisiilor [45].
- nivelul rețea și nivelul transport – în general aceste nivele fuzionează în cazul comunicației în rețelele wireless de senzori. Acest nivel rezultat asigură o comunicație fiabilă punct la punct (end-to-end) în cadrul rețelei. Aici se regăsesc cu precădere protocoalele de rutarea informației.
- nivelul aplicație – este nivelul unde datele utile se extrag din nivelele inferioare oferindu-se aplicației doar ceea ce este necesar. Aici se mai regăsesc de multe ori și protocoalele de sincronizare [46].

Într-o comunicare în timp real între nodurile unei rețele de senzori wireless nu doar transportul cu succes al unui mesaj între sursă și destinație este important ci și durata de timp necesară pentru a realiza această operație. Această caracteristică este diferența esențială între o rețea clasică de senzori și o rețea de senzori cu comunicare în timp real [47].

Pentru a se putea realiza o comunicare în timp real, atât dacă este vorba de comunicare hard real-time sau soft real-time, trebuie ca protocoalele de pe fiecare nivel prezentat mai sus să respecte cerințele de timp real. Acest lucru se poate realiza când protocoalele de pe fiecare nivel lucrează în manieră timp real, dar cu mai multă ușurință dacă există o legătură destul de strânsă între nivele, protocoalele ținând cont de cerințele, capacitățile, rolurile și posibilele probleme ale celorlalte protocoale.

Un parametru esențial în sistemele în timp real este timpul. În cazul comunicării wireless între 2 noduri ale unei rețele de senzori durata de timp dintre transmisia unui pachet de date de un nod și recepția acestui pachet de către nodul destinatar se poate aproxima astfel [48]:

$$T(S, N) = T_C + T_t + T_{pp} + T_p + T_q + T_s \quad (3-1)$$

unde:

$T(S, N)$  – Timpul de transmisie un pachet de la nodul S la nodul N

$T_C$  – timpul necesar nodului S de a obține acces pe mediul de comunicație

$T_t$  – timpul transmisiei pachetului pe mediul de comunicație – depinde de dimensiunea pachetului și codarea de linie aplicată (a tipului de modulație)

$T_{pp}$  – timpul de propagare a semnalului de la nodul S la nodul N care depinde de factorii fizici ai mediului și de perturbații

$T_p$  – timpul de procesare care depinde de algoritmi de procesare a datelor implementați în rețea

$T_q$  – timpul cât un pachet stă în coada de așteptare pentru a fi transmis, acest parametru este influențat de gradul de încărcare a rețelei, de traficul din rețea

$T_s$  – timpul de comutare a unui nod din starea de inactivitate (sleep) în starea de funcționare și invers

Caracteristicile comunicării în timp real diferă substanțial de caracteristicile comunicării non real-time. Într-o comunicare non real-time, un punct de interes major îl reprezintă de obicei lățimea de bandă, fiind de dorit ca aceasta să fie cât mai mare și astfel să se poată transfera date cât mai multe într-un timp cât mai scurt. Acest parametru nu prezintă interes în cazul comunicării în timp real. În schimb aici interesează ca un mesaj să ajungă de la sursă la destinație într-un anumit timp limită iar acesta să nu fie depășit, în caz contrar degradând puternic funcționalitatea sistemului sau chiar anulând-o. De asemenea în cazul comunicării în timp real un factor de interes poate fi și procentajul pachetelor pierdute (miss ratio). Pentru îndeplinirea acestui obiectiv este necesar ca timpul de transmisie prezentat mai sus să aibă un grad cât mai mare de predictibilitate. Pentru aceasta trebuie în primul rând identificată componenta ce oferă impredictibilitatea acestui timp și găsirea unor metode de a elimina această proprietate.

În rețelele hard real-time faptul că un mesaj și-a depășit deadline-ul poate avea urmări catastrofale la nivelul sistemului și astfel este necesară o valoare maximă și absolută a deadline-ului fiecărui mesaj. Se observă că în cazul comunicării în timp real se poate face o analogie cu planificarea task-urilor în timp real, lucrurile fiind în multe situații similare. În rețelele care nu sunt hard real-time de obicei este de interes să se cunoască acel miss ratio și este de dorit ca acesta să nu depășească o anumită valoare (spre exemplu în sistemele multimedia).

Așadar în literatură traficul într-o rețea de senzori cu comunicare wireless în timp real traficul se împarte în: trafic garantat și trafic statistic. Se spune că traficul este garantat atunci când fiecare pachet de date ajunge la destinație înainte de expirarea deadline-ului. Traficul statistic este acela în care se spune că nu mai mult de un anumit procent din pachetele transmise nu ajung la destinație în deadline-ul specificat. De asemenea traficul într-o rețea wireless de senzori poate fi împărțit în trafic determinist sau trafic probabilistic. Traficul determinist este acela folosit în sisteme cu comunicație hard real-time unde se garantează atât o lățime minimă de bandă, un deadline pentru fiecare pachet și o întârziere maximă într-o comunicație end-to-end. Trafic probabilistic pe de altă parte este acel trafic în care se garantează doar o anumită calitate a serviciului – QoS (Quality of Service) cu o anumită probabilitate. Acest lucru este pretabil rețelelor de senzori wireless cu cerințe soft real-time [49].

### 3.2 Aplicații ale rețelelor wireless de senzori

Înainte de a intra în profunzime în problemele ce apar în comunicarea din cadrul rețelelor de senzori se va face o scurtă trecere în revistă a unor aplicații importante implementate folosind rețele wireless de senzori precum și platforme deja existente care se pot configura astfel încât să poată implementa aplicații diferite.

O importantă aplicabilitate a rețelelor wireless de senzori este aceea de a monitoriza parametrii unui mediu și de a trimite datele adunate la o entitate superioară rețelei care are de obicei rolul de a le centraliza, interpreta și afișa utilizatorului. De obicei aceste rețele au un efect pasiv în sensul că adună date din mediul înconjurător dar nu acționează asupra acestuia.

O rețea de senzori wireless deja folosită în multe aplicații este EDETA (Energy-efficient Adaptive Hierarchical and robust Architecture). Această platformă tratează și încearcă să rezolve multe din problemele apărute în rețelele de senzori wireless: cum ar fi toleranța la situația în care un nod dispăre din rețea, consumul de energie electrică, autoconfigurare, autogestiune a rețelei, scalabilitate, etc. Platforma este orientată în general pe monitorizarea mediului astfel că nodurile rețelei analizează diferiți parametri ai mediului și trimit rezultatele către un concentrator de date. Un nod al rețelei este asemănător ca și configurație cu ceea ce s-a prezentat mai sus.

Deși tratează multe probleme chiar și în ceea ce privește timpul, platforma totuși nu oferă garanții suficiente de timp real. Una din funcțiile pentru care a fost configurată această platforma este aceea de a monitoriza nivelul de poluare din mediu înconjurător [50].

Aceași platforma poate fi găsită implementând și altă aplicație cum ar fi un sistem de detecția incendiilor. Față de prima variantă arhitectura unui nod este semnificativ diferită dar în ansamblu, ideea întregii platforme rămâne neschimbată. Sistemul central care adună datele de la întreaga rețea este un PC și tot acesta are și rolul de a genera alarma de incendiu. Se pun și aici probleme de restricții de timp dar într-o măsură redusă și în concluzie rețeaua nu realizează funcțiile descrise în timp real. În ambele situații autorii prezintă rețeaua într-o organizare pe clustere [51].

O importantă apreciere trebuie acordată acestor lucrări deoarece avem de a face cu o implementare fizică a rețelei nu doar cu simulări care de cele mai multe ori ignoră unii factori reali de o importanță deosebită. Platforma își dovedește astfel funcționalitatea într-un mediu real, nu doar într-un mediu simulat.

Un alt exemplu de aplicabilitate ar fi un sistem de monitorizare, control și predicție a consumului de energie electrică într-o clădire inteligentă. Un astfel de sistem are o caracteristică în plus față de platforma prezentată anterior și anume aceea că sistemul are și rol de control, devenind un sistem activ. Sistemul se prezintă sub forma unei rețele de senzori wireless cu noduri ce au ca funcție măsurarea consumului consumatorului la care este legat, transmiterea datelor de consum către concentratorul de date precum și decuplarea consumatorului de la rețea în urma intervenției factorului uman. Nodurile din această rețea au ca și componentă de comunicație un modul de tip XBee care au la bază mult cunoscuta stivă ZigBee. Referitor la consumul de energie nodurile sunt gândite pentru consum redus de energie dar numai pentru a nu influența simțitor consumul în clădirea în care au fost instalate, nodurile fiind alimentate de la rețea [52, 53].

Tot un caz de rețea de senzori wireless activă ar fi și un sistem de detecția și stingerea incendiilor. Un astfel de sistem este conceput ca o rețea wireless de senzori cu noduri eterogene. Există astfel mai multe tipuri de noduri: noduri cu funcție de monitorizare, noduri cu funcție de stingere a incendiului și noduri intermediare cu funcție de a realiza rutarea informației și de asemenea de a amplifica semnalul comunicației. Aceste noduri sunt slabe în resurse și dimensiuni. Ca și stivă de protocoale de comunicație wireless se folosește din nou stiva ZigBee. În acest caz nu se pune deloc problema operării în timp real deși aplicația pe care o implementează ar necesita acest lucru încadrându-se la categoria de aplicații de siguranță publică [54].

În cadrul Laboratoarelor de Prelucrare Numerică a Semnalelor (DSPLabs) din Departamentul de Calculatoare al Facultății de Automatică și Calculatoare, Universitatea Politehnică Timișoara se află în dezvoltare o platformă robotică - CORE-TX (Collaborative Robotic Environment - the Timisoara eXperiment) – complexă, bazată pe rețele wireless senzori care are ca scop studiul teoretic și aplicativ al mediului colaborativ al microsistemelor de timp real autonome și integrarea acestora în aplicații de zi cu zi. Această platformă este orientată în mare parte către aplicații cu cerințe de timp real, mai mult către aplicații critice cu cerințe stricte de timp (hard real-time).

Sistemul CORE-TX este dezvoltat ca o platformă complexă clădită pe mai multe niveluri. Se disting trei niveluri principale: un nivel al microsistemelor autonome cu inteligență incorporată, un nivel de comunicație și un nivel de supervizare și control al întregului sistem, după cum reiese din Figura 3-5[55]:

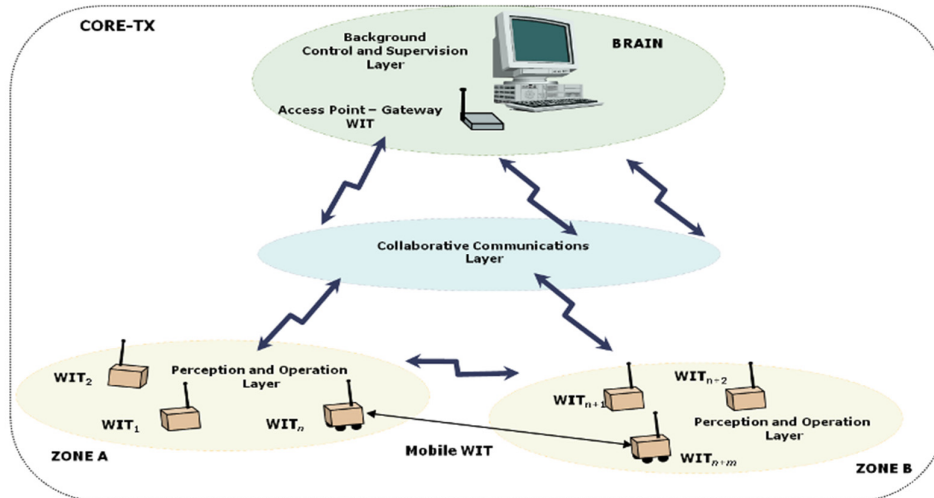


Figura 3-5 Platforma CORE-TX

Platforma este structurată pe mai multe nivele. Nivelul de supervizare și control reprezintă nivelul superior de abstractizare al modelului CORE-TX și este reprezentat de entitatea BRAIN (Background Robotic Activity Induction Node), implementată cu ajutorul unui calculator gazdă (PC), cu conectivitate sporită pentru comunicații, pe care rulează un sistem de pachete software originale. Entitatea BRAIN este situată pe un calculator PC extrem de performant. Acesta supraveghează toate unitățile WIT (Wireless Intelligent Terminal), este responsabilă, totodată, și pentru interfața cu utilizatorul. Entitatea este conectată la rețeaua de WIT-uri printr-un dispozitiv de tip gateway (care este de fapt un simplu WIT cu capacitatea de conectare la PC). La nivelul percepției și operării, sistemul CORE-TX interacționează cu mediu înconjurător prin intermediul microsistemelor autonome cu inteligență incorporată, denumite WIT. Aceste micro sisteme pot avea funcții de percepție, de operare sau combinate. Un modul WIT este alcătuit din mai multe plăci interconectate la nivel fizic pentru a forma un singur bloc iar sursa de alimentare este asigurată de un acumulator. Modulele WIT, care sunt practic nodurile rețelei wireless, în funcție de necesitate pot avea sau nu anumite capacități printre care și cea de mobilitate. O configurație minimă este aceea ca WIT-ul să aibă capacități senzoriale, de comunicare și de gestiune a acumulatorului și a consumul de energie [56].

### 3.3 Rutarea informației în rețele wireless de senzori

În ceea ce privește protocoalele de comunicație din rețelele wireless de senzori, printre cele mai studiate de către comunitatea academică sunt protocoalele de rutare a informației. Această secțiune are menirea de a prezenta succint problematica rutării informației în rețelele wireless de senzori.

Referindu-ne la modelul OSI relatat mai sus, protocoalele de rutare se găsesc în nivelul 3 din stiva de protocoale, în nivelul rețea.

În general nodurile unei rețele de senzori sunt dispuse aleatoriu, în număr foarte mare pe o suprafață pentru a monitoriza / controla parametrii unui mediu. Pentru a se putea asigura transmiterea datelor către un data concentrator este necesar ca unele noduri să primească și funcția de rutare a informației deoarece nu toate nodurile sunt în directă arie de comunicație a data concentratorului. Aceste noduri trebuie să suporte execuția unor algoritmi specializați în rutarea informației, realizându-se în final o comunicare de tip *multihop*. De asemenea, chiar dacă unele noduri sunt în directă arie de comunicație a unui data concentrator sau coordonator uneori este de preferat să se facă o transmisie multihop decât una directă din considerente de consum de energie, consumul de energie electrică la transmisie de obicei crește puternic odată cu puterea de emisie [57].

O primă și importantă clasificare a protocoalelor de rutare în rețelele de senzori se poate face după momentul și modul în care se creează rutele. Astfel rutarea poate fi pro activă și reactivă [58]. Rutarea pro activă presupune stocarea de către un nod a întregii organizări a rețelei și implicit a tuturor rutelor posibile. Astfel se elimină orice penalitate de timp în momentul rutării informației deoarece se cunoaște în orice moment configurația rețelei și a rutei ce trebuie stabilită. O astfel de rutare este de altfel extrem de rigidă și necesită o cantitate mare de memorie pe fiecare nod al rețelei pentru stocarea întregii tabele de rutare. De asemenea este foarte greu să se țină cont de situația când un nod dispare din rețea. Diametral opusă este rutarea reactivă în care fiecare rută este stabilită la cerere. În această situație nu se stochează nici o rută în nodurile rețelei dar se introduce o puternică penalitate de timp pentru descoperirea rutelor care este de asemenea și impredictibilă. Acest tip de rutare oferă o mare scalabilitate și adaptabilitate a rețelei dar penalitatea totală de timp devine extrem de mare punându-se uneori problema pierderii conectivității unor noduri din cauza numărului mare de mesaje schimbate în vederea stabilirii rutelor. De obicei majoritatea algoritmilor folosesc o tehnică rezultată din fuziunea celor 2 tehnici prezentate mai sus, rutarea pro activă respectiv rutarea reactivă [59]. Un astfel de exemplu este și soluția propusă în [60] unde nu se folosesc tabele liniare de rutare ci mai degrabă se folosesc arbori de rutare. Aceștia se refac periodic sau la cerere dar se și stochează pentru a înlătura penalități de timp inutile în urma cărora se obține un rezultat deja obținut anterior.

Un alt criteriu de clasificare a rutării în rețelele de senzori este organizarea rețelei [61].

- Flat routing – este un tip de rutare în care rețeaua este în principiu de tip ad-hoc, nu este organizată în ierarhii de tip cluster. În acest caz practic se poate vorbi de rețele ad-hoc cu caracteristici de rețea mesh; fiecare nod poate avea și funcție de router. De asemenea acest tip de rutare se pretează cel mai mult rețelelor wireless de senzori cu nodurile dispuse aleatoriu și mai mult acelor rețele în care nodurile pot fi și mobile.
- Hierarchical routing – este un tip de rutare întâlnit la rețelele organizate în structuri de clustere. În această situație doar nodurile de tip CH pot să aibă

funcție de rutare. Această situație are ca principale avantaje reducerea consumului de energie și scalabilitatea dar un puternic dezavantaj este acela că nodurile de tip CH pot să dispară mai repede din rețea din cauza consumului ridicat de energie electrică, ele fiind nevoite să intermedieze și să ruteze mesajele în rețea și implicit clusterul pe care îl conduc devine izolat de restul rețelei.

- Adaptive routing – o rutare în funcție de datele ce se transmit prin rețea. Scopul este ca rutele să se stabilească în așa fel încât să se elimine redundanța la comunicare, adică să se elimine situația când un nod rutează pachete diferite dar cu conținut identic sau aproape identic.
- Multipath routing – rutarea în care se încearcă să se găsească mai multe rute pentru ca nodul emițător să atingă nodul receptor final. Acest lucru ajută la îmbunătățirea fiabilității întregii rețele. De asemenea tot în acest caz în care există mai multe rute între două noduri se pot lua decizii de selecție a unei rute în funcție de mai mulți parametri cum ar fi: energia consumată pentru transportul pachetului, timpul, calitatea legăturilor, etc.
- Query-Based routing – Este un tip de comunicare în care nodul cu rol de data concentrator lansează interogări și cereri pentru recepția datelor iar nodurile care au datele respective disponibile le transmit către acesta.
- Negotiation-Based routing – asemănător tehnicii adaptive routing și această tehnică are menirea de a elimina redundanța datelor din pachetele ce circulă prin rețea dar se introduce și un proces de negociere pentru a se stabili rutele, datele și nodurile ce sunt somate să transmită informațiile deținute precum și informațiile ce trebuie rutate.

Majoritatea studiilor realizate de cercetători referitor la rutarea în rețele wireless de senzori sunt orientate asupra rețelelor organizate în structuri de clustere, fiind oarecum mai ușor de a le gestiona. De asemenea, nodurile fiind sisteme embedded alimentate pe baterie se pune și problema reducerii consumului de energie, acest lucru fiind un aspect puternic studiat în cadrul protocoalelor de rutare a informației. După un studiu aprofundat a literaturii de specialitate se poate constata că principalele direcții de studiu în problematica rutării în rețele wireless de senzori sunt reducerea consumului de energie, îmbunătățirea vitezei de comunicație, oferirea unui minim suport de timp real și foarte puțin garantarea respectării unor restricții de timp.

În acest sens putem aminti și tehnica LEACH (Low-Energy Adaptive Clustering Hierarchy) care după cum îi spune și numele este orientată pe configurarea unei rețele de senzori sub formă de clustere. Partea interesantă a tehnicii este aceea că nu se bazează pe o rețea deja organizată sub formă de clustere ci pe o rețea cu noduri dispuse aleatoriu. În prima fază rețeaua trece printr-o fază de autoorganizare în care se formează clusterelor. Din multitudinea de noduri, pe baza unor parametri, fiecare nod decide când trebuie să devină CH și când trebuie să renunțe la acest rol. După ce se decid coordonatorii de cluster, fiecare nod decide cui cluster i se arondează. Acest lucru se realizează în manieră CSMA (Carrier Sense Multiple Access) după care, în momentul în care un cluster s-a format, comunicarea continuă mai departe în manieră TDMA (Time Division Multiple Access). Cu cât crește numărul nodurilor dintr-un cluster cu atât crește și dimensiunea frame-ului TDMA. Nodul CH își ține permanent activă interfața radio pentru a putea primi oricând un pachet de la nodurile arondate clusterului. Nodurile frunză ale clusterului nu pot comunica decât cu nodul CH. Comunicarea între clusterelor rețelei se realizează în manieră CSMA. Dezavantajele acestei abordări sunt acelea că nodul CH are șanse foarte mari să își termine energia din baterie într-un timp relativ scurt din cauza solicitării modulului de comunicație și de asemenea faptul că între clusterelor se comunică într-o manieră cu coliziuni ceea ce

reduce puternic predictibilitatea întregului sistem. Tot o problemă poate fi și numărul maxim de noduri dintr-un cluster limitate fiind de o dimensiune maximă a frame-ului TDMA. Autorii au validat tehnica LEACH folosind simulări ceea ce din păcate practica invalidează în multe cazuri rezultatele obținute folosind simulatoare [62].

O problemă apărută în tehnica prezentată mai sus și anume faptul că nodurile de tip CH își mențin pornită interfața de comunicare wireless crescând astfel consumul total de energie și scăzând astfel durata de viață a nodului și implicit a clusterului pe care îl conduce, este tratată și în [63] unde autorii se orientează spre a opri interfața radio cât mai mult posibil obținând o reducere semnificativă a consumului de energie printr-o tehnică denumită GAF (Geographical Adaptive Fidelity). De asemenea tehnica prezentată de autori se încadrează și în categoria algoritmilor de rutare adaptivă în funcție de informațiile ce se transmit din nod în nod și se încearcă și aici să se reducă transmisia de date redundante prin localizarea nodurilor. Și în acest caz se menține validarea algoritmului folosind simulatoare, autorii având totuși viziunea asupra faptului că în acest domeniu mediul fizic poate avea un cuvânt greu de spus, putând da peste cap întreaga gândire a tehnicilor folosite.

O abordare asemănătoare, deși nu este specifică rețelelor de senzori ci rețelelor wireless în general, este și algoritmul Span [64]. Acesta elimină ideea ca algoritmul de rutare să se folosească și de poziția geografică a nodului în rețea și reduce și mai mult consumul de energie electrică. Ideea de bază este că se alege dinamic câte un coordonator, pentru o perioadă finită de timp în care el trebuie să servească rutării informației între nodurile vecine. După o perioadă de timp se poate decide ca nodului să i se ia acest rol. Ca și protocol MAC folosit este IEEE 802.11, un protocol MAC folosit în principal pentru rețele de tip WLAN (Wireless Local Area Network) [65]. Simularea este din nou singura metodă de a valida abordarea propusă.

În exemplele prezentate până acum nu se pune problema timpului real, a predictibilității. O abordare în acest sens care completează soluțiile prezentate anterior este reprezentată de algoritmul SPEED. Acesta are ca principal scop oferirea unor garanții de timp real. Algoritmul se referă strict la rutare și folosește principiile enunțate mai sus, cum ar fi rutarea în funcție de localizarea geografică a nodurilor. În plus, folosind mecanisme de feedback algoritmul poate garanta un delay end-to-end. Pe lângă faptul că oferă garanții de timp real, SPEED funcționează în cazul rețelelor fără coordonator, fără sincronizare prin baliză, practic fiind pretabil rețelelor ad-hoc [66]. Autorii din nou folosesc protocolul IEEE 802.11 ca și protocol de acces la mediu care în primul rând nu este gândit pentru rețele de senzori iar pe de altă parte nu oferă garanții de timp real. Ca și în soluții precedente validarea se face cu ajutorul simulatoarelor.

O abordare mai pragmatică în care se folosesc nu doar simulatoare dar și noduri fizice care formează o rețea wireless de senzori este și cea care se referă la protocolul RPAR (Real-time Power-Aware Routing) [67]. Acest protocol oferă garanții de timp real, mai exact se orientează pe soft real time în care se poate garanta de fapt un anumit procent maxim de pachete pierdute. Autorii introduc noțiuni precum managementul vecinilor, estimarea întârzierilor și în plus realizează un important studiu din care rezultă impactul puterii de emisie asupra întârzierilor apărute în comunicație, protocolul încercând să facă un compromis între acești parametri. De asemenea se folosesc și mecanisme de retransmisie automată a pachetelor – ARQ (Automatic Repeat Request). Verificarea și validarea algoritmului precum și aceste studii se realizează atât prin intermediul simulatoarelor dar se folosesc și module wireless fizice observându-se astfel discrepanța dintre rezultatele obținute prin simulări și cele obținute prin măsurători ale implementării fizice. Totuși pentru ca algoritmul să funcționeze autorii fac unele presupunții ce oferă rețelei un grad mare de rigiditate: nodurile sunt fixe, se cunoaște poziția lor exactă (obținută chiar prin GPS), se cunoaște întreaga structură a



rețelei. O rețea de senzori se caracterizează în primul rând prin dinamicitate și astfel algoritmul acesta se poate aplica doar pentru anumite cazuri particulare. Algoritmul nu ignoră nivelele inferioare ale stivei de protocoale ci ține cont de caracteristicile protocolului MAC ce îl transportă: se folosește ca și protocol MAC protocolul B-MAC. Nodurile sunt reprezentate de module de tip MICA2 care au ca și componentă software de bază sistemul de operare TinyOS ce oferă un mic suport pentru timp real.

Similar protocolul RPAR este și algoritmul de rutare pentru rețele de senzori prezentat în [48]. Algoritmul are 4 module "*neighbour management, power management, location management, routing management*" prin care se combină mai multe tehnici și abordări anterioare rezultând o manieră de a ruta informația în rețele de senzori și de a oferi minime garanții de timp real având în vedere și reducerea consumului de energie. În prima parte a lucrării autorii precizează că se ia în calcul deadline-ul transmisiei de pachete dar nu se dă și o metodă de a garanta acest deadline ci totul se bazează pe rezultatele obținute din măsurători pe baza simulatoarelor. Modulele folosite ca și noduri ale rețelei de senzori sunt modulele MICAz. De asemenea se consideră o rețea fixă în care se cunoaște a priori poziția nodurilor.

Tot în categoria de rutare soft real time se încadrează și QERT, un algoritm de rutare specializat pentru rețele wireless de senzori ce oferă garanții soft real time. Rețelele pe care se orientează acest algoritm sunt acele rețele organizate în topologii orientate pe clustere unde este necesară prezența unui coordonator sau BS (Base Station). Algoritmul este și orientat pentru reducerea consumului de energie în ceea ce privește comunicarea în sensul că rutarea informației se face pe calea cea mai eficientă din punct de vedere al consumului de energie, nu neapărat cea mai scurtă cale, cu cel mai mic număr de hopuri intermediare [68].

O abordare hard real time a problematicei rutării este redată în [34] în care se propune o metodă de rutare a informației care să eficientizeze consumul de energie electrică în rețelele de senzori care funcționează cu cerințe stricte de timp. Metoda se bazează pe planificarea mesajelor după o manieră non-preemptivă de tip EDF. Protocolul MAC folosit este din nou clasicul 802.11 ceea ce nu oferă garanții de timp real și în plus nu este pretabil rețelelor de senzori iar rezultatele sunt obținute doar pe baza simulărilor.

Prin folosirea facilitațiilor DMS și DVS se încearcă și mai mult economisirea energie electrice la nivel de nod și se studiază impactul pe care acest lucru îl are asupra unei comunicații end-to-end în cazul rețelelor în timp real. Autorii lucrării [69] se orientează asupra rețelelor cu funcție strictă de monitorizare de mediu și agregare de date cu o configurație fixă cunoscută inițial.

Tot în domeniul reducerii energiei consumate și măririi duratei de viață a rețelei se află și algoritmul HARP (Hierarchical Adaptive and Reliable Routing Protocol). Algoritmul funcționează în cazul rețelelor organizate în structuri de clustere și pe lângă îmbunătățirile aduse la consumul de energie, algoritmul încearcă să aducă mici rezolvări problemelor de apar la nivelul nodurilor de CH și anume faptul ca acesta este cel care își consumă cel mai repede bateria din cauza solicitării interfeței de comunicare, acesta fiind cel ce realizează rutarea între nodurile din clusterul ce îl conduce și alte clustere. Comparativ cu tehnica LEACH, algoritmul HARP aduce importante îmbunătățiri și pentru a include noduri noi în rețea și pentru a construi rute alternative în situația în care vechile rute nu mai sunt valabile. Și în această situație măsurătorile nu se bazează pe cazuri reale ci pe simulări [70].

Tot la capitolul rutare se încadrează oarecum și [71] deși autorii prezintă mai mult algoritmi de nivel de acces la mediu. Considerentul pentru această încadrare este faptul că lucrarea prezintă o idee interesantă legată de comunicarea de tip broadcast în sensul că se transformă în mai multe comunicări unicast cu scopul de reduce consumul

de energie. Practic, în loc de a face o emisie broadcast cu putere maximă, este posibil să fie mai eficient mai multe emisii unicast, din nod în nod care să echivaleze emisia broadcast.

Majoritatea protocoalelor existente se referă mai mult la situația în care nodurile rețelelor sunt statice și mai puțin în cazul în care acestea sunt mobile. Un exemplu în care se tratează această situație este prezentat în [72].

Nu în ultimul rând putem aminti și protocolul WirelessHART, un protocol comercial, standardizat IEEE, destinat rețelelor industriale de senzori. Se referă la cazul particular de rețea statică, fixă, cunoscută perfect din punct de vedere al plasării nodurilor. Standardul se folosește cu precădere în situația în care nodurile rețelei transmit doar date către un concentrator, un sink. Rețeaua este de tip industrial, protocolul necesită un manager de rețea care este un sistem de calcul mai puternic, nu poate fi vorba de sistem embedded [73].

### 3.4 Concluzii

Rețelele de senzori wireless sunt compuse din sisteme incorporate de mici dimensiuni, cu sau fără mobilitate, cu capacități hardware limitate, alimentate de o baterie sau acumulator care au posibilitate de comunicare radio. În plus, majoritatea acestor sisteme incorporate pot avea capacități senzoriale diferite în funcție de tipul de aplicație.

Din punct de vedere al aplicabilității, rețelele de senzori pot fi întâlnite într-o gamă largă de aplicații precum sisteme de monitorizare a mediului, sisteme de securitate, sisteme de detecție a incendiilor, sisteme robotizate de intervenție în caz de calamități, aplicații militare, platforme robotice. O parte din aceste aplicații au cerințe de timp real, adică rețeaua trebuie să răspundă evenimentelor monitorizate și controlate într-un timp bine definit asemănător oricărui sistem în timp real. Pentru ca acest lucru să fie îndeplinit, este necesar ca nu doar fiecare nod să funcționeze în manieră timp real ci și întreaga rețea să îndeplinească criteriile unui astfel de sistem în timp real. Acest lucru implică faptul că și comunicația dintre noduri trebuie să funcționeze după principiile operării în timp real.

Majoritatea protocoalelor de comunicație în cadrul rețelelor de senzori sunt orientate în primul rând pe reducerea consumului de energie și pe îmbunătățirea rate de transfer între nodurile rețelei și mai puțin pe îndeplinirea unor cerințe stricte de timp. Totuși protocoalele superioare din stiva de comunicație au fost dezvoltate în ultima perioadă și spre direcția operării în timp real. Protocoalele precum cele cu rolul de rutare a informației au primit poate cea mai mare atenție în cercetarea din ultima perioadă.

Se pot trage astfel următoarele concluzii despre algoritmi și tehnicile de rutare a informației în rețelele wireless de senzori:

- Studiile sunt axate în marea majoritate pe cazuri particulare, în speță pe rețelele organizate pe clustere

- Majoritatea algoritmilor consideră că protocoalele ce le transportă, respectiv protocoalele MAC de nivel inferior nu aduc nici o penalitate de timp și oferă garanții de timp real
- Validarea tehnicilor și a algoritmilor se face cu ajutorul simulatoarelor ignorându-se puternic situațiile reale
- Se consideră în multe cazuri că rețeaua este fixă, nodurile nu sunt mobile și se cunoaște cu exactitate poziția nodurilor. Aceste presupunții reduc semnificativ gradul de aplicabilitate ale rețelelor de senzori
- Studiile sunt orientate în mare parte pe reducerea consumului de energie, nodurile unei rețele wireless de senzori sunt alimentate pe baterie
- Nu se acordă în multe cazuri o mai mare importanță situațiilor cu cerințe stricte de timp. Aceste situații pot fi semnificativ întâlnite în rețelele de senzori, acestea având rol de monitorizare și control a mediului în care sunt dispuse, o aplicație cu cerințe clare de timp real
- Puținele studii referitoare la timp real sunt orientate pe comunicație în timp real statistică și mai puțin pe comunicație garantată

## 4 Protocoale MAC pentru rețele de senzori

Principala preocupare a lucrării de față o constituie protocoalele de comunicație de acces la mediu pentru rețelele de senzori cu comunicație fără fir acest capitol fiind rezervat unei analize teoretice și critice a acestui subdomeniu. Se prezintă tipurile de protocoale existente și se extrag principalele probleme existente la ora actuală. Se pune totuși accent pe standardul IEEE 802.15.4 descriindu-se mai detaliat deoarece este prezent în multe din actualele implementări de stive de comunicație pentru acest tip de rețele de senzori.

### 4.1 Considerații generale

Protocoalele de nivel MAC sunt în principal protocoale de nivel 2 din modelul de referință OSI specializate pe accesul la mediul de comunicație. Principalul rol al acestuia este de a asigura o legătură directă, punct la punct între 2 noduri ale rețelei, o conexiune stabilă de acest tip fiind vitală pentru nivelele superioare, cum ar fi spre exemplu protocoalele de rutare [74].

În funcție de modul de acces la mediu protocoalele MAC se pot clasifica în:

- protocoale bazate pe coliziune (contention based). În această categorie se încadrează protocoalele bazate pe CSMA. În cazul acestor protocoale, ținând cont că mediul de comunicație este în marea majoritate comun tuturor nodurilor, coliziunile datorate faptului că mai multe noduri emit simultan pot să apară cu o probabilitate foarte mare (mai ales în cazul unui număr mare de noduri). Astfel de protocoale în principiu încearcă să reglementeze comunicația astfel încât, după aplicarea unei politici de tratare a coliziunilor, fiecare nod să poată accesa mediul de comunicație pentru a-și transmite datele deținute. Avantajul acestui tip de protocoale este simplitatea în implementarea pe nodurile rețelei, dar un puternic dezavantaj constituie impredictibilitatea.
- protocoale bazate pe planificarea transmisiilor (scheduling based). În această categorie se încadrează protocoalele de tip TDMA. Aceste protocoale împart timpul astfel încât fiecărui nod i se atribuie o cantitate de timp în care poate transmite fără să se suprapună peste transmisia altor noduri. Acest tip de protocoale oferă un grad mare de predictibilitate dar implementarea devine greoaie: este necesară ori sincronizarea ceasului tuturor nodurilor rețelei sau existența unui coordonator care prin intermediul unei balize să ofere baza de timp.

În marea majoritate a situațiilor, pentru rețelele wireless de senzori se folosește atât prima variantă cât și cea de-a doua iar în unele abordări se folosește o combinație a celor două.

Metoda de acces la mediu CSMA/CA (Carrier Sense Multiple Access / Collision Avoidance) funcționează astfel: înainte ca un nod să transmită, acesta verifică dacă mediul de comunicație este ocupat, în acest caz își întârzie transmisia până când mediul devine liber. Acest tip de protocoale are avantajul că se implementează ușor, poate fi folosit pe module half-duplex și funcționează foarte bine în situații în care mediul de comunicație nu este intens folosit. Metoda CSMA/CD (Carrier Sense Multiple Access / Collision Detection) pe de altă parte nu verifică dacă mediul de comunicație este liber

ci transmite direct. Acesta în schimb detectează dacă a apărut coliziune și doar în acest caz își întârzie transmisia. Această metodă necesită ca modulul de comunicație al nodului să fie full-duplex [75].

## 4.2 Protocoale MAC bazate pe scheme de acces cu coliziuni

Un protocol larg folosit în multe dintre platformele de senzori wireless apărute în ultima perioadă este acela oferit de standardul IEEE 802.15.4. Standardul este practic derivat din clasicul IEEE 802.11 folosit pentru rețelele WLAN [76], modificându-l astfel încât să ofere facilitățile necesare rețelelor de senzori. Protocolul definește atât nivelul 1, nivelul fizic (PHY), cât și nivelul 2, nivelul de acces la mediu (MAC).

Înainte de apariția standardului IEEE 802.15.4 au existat unele abordări ale problemei comunicației în cadrul rețelelor de senzori și nu numai din partea comunității academice. Practic, studiind literatura de specialitate se poate deduce că și standardul 802.15.4 a fost, practic, o continuare a muncii depuse de cercetătorii din mediul academic.

În domeniul comunicării hard real-time, deși nu este vorba de mediul wireless, a apărut de la început ideea de a adapta protocoalele de tip CSMA pentru a oferi garanții de timp real, celelalte având un grad mai ridicat de dificultate în implementare mai ales din cauza lipsei de resurse hardware ale nodurilor unei rețele.

O abordare este și cea denumită *Virtual Time CSMA* [77]. În această lucrare se prezintă un algoritm de comunicare cu garanții hard real-time, nu în mediul wireless dar cu mari șanse de a se aplica și în acest domeniu. Autorii prezintă un protocol de nivel MAC ce se bazează pe ideea de CSMA. Ideea de bază este de a semna temporal pachetele care circulă prin rețea folosind un timp virtual. Metoda lor poate oferi garanții de timp real doar în situația în care nodurile au ceasul intern sincronizat, sincronizarea realizându-se cu acest timp virtual. Abordarea prezentată de autori ar putea fi studiată și în cazul comunicării în rețelele wireless de senzori.

Ca și în cazul protocoalelor de rutare abordările soft real-time sunt mult mai răspândite decât cele hard real-time. O abordare de tip soft real-time poate fi reprezentată de așa numitele *splitting protocols*. Acest tip de protocoale se apropie de politica de tip TDMA de acces la mediu, în sensul că multiplexarea mai multor participanți la comunicare se face în timp, astfel că mediul de comunicare este împărțit în sloturi timp de unde și denumirea protocoalelor. În astfel de situații nodurile nu inițiază comunicarea decât la începutul unui slot de timp și nu în alt moment al acestuia. [78, 79]

Diferența față de TDMA este aceea că și în cazul *splitting protocols* pot apărea coliziuni dar acestea apar doar la începutul unui slot de timp pe când în cazul TDMA fiecărui nod  $i$  se alocă un slot de timp și doar el îl folosește. Spre exemplu un astfel de protocol, bazat strict pe ideea clasică de TDMA este prezentată în [80] unde baza de timp este dată de un nod master, practic toată structura prezentată de autori poate fi asemuită cu structura de cluster. Problema de bază a acestei abordări este nodul master care fiind foarte solicitat din punct de vedere al comunicării consumul de energie este mare și astfel durata de viață a acestuia este mică, implicit durata de viață a întregii rețele fiind mică, restul nodurilor neputând comunica fără semnalele oferite de master.

O combinație interesantă între TDMA și FDMA (Frequency Division Multiple Access) este prezentată în [81] în vederea obținerii unor garanții de timp pentru o abordare

soft real time în cadrul comunicației într-o rețea de senzori cu menținerea unor mecanisme de a reduce consumul de energie.

Tot în domeniul soft real time se încadrează și protocolul RAP. Acesta încearcă să îmbunătățească protocolul oferit de standardul 802.11. Se prezintă practic o arhitectură de comunicare soft real time pentru rețele de senzori la scară mare. Autorii introduc o politică de planificare a pachetelor ce se transmit prin rețea denumită VMS (Velocity Monotonic Scheduling) care se bazează pe noțiunea de *packet velocity* (viteza pachetului). Ideea vitezei pachetului este dată de faptul că un pachet, ca să-și îndeplinească deadline-ul trebuie să se transmită de-a lungul rețelei cu o anumită *viteză*. Algoritmul nu se referă doar la accesul la mediu ci și la partea de rutare și se aduc importante îmbunătățiri de timp standardului 802.11 [82].

Deși nu oferă garanții de timp real protocolul S-MAC (Sensor MAC) este un protocol folosit poate la fel de mult ca și IEEE 802.15.4 în rețelele de senzori. Protocolul se bazează direct pe standardul IEEE 802.11 preluând de la acesta secvențele RTS (Request To Send), CTS (Clear To Send), DATA, ACK. Ceea ce a adus nou acest protocol este ideea de a permite nodurilor să intre într-o stare de consum redus de energie electrică, oprindu-și modulul radio pe perioada în care ele nu sunt implicate în comunicație. Punctele importante pe care autorii le urmăresc la acest protocol sunt scalabilitatea, consumul redus de energie, adaptabilitatea pentru sisteme embedded slabe în resurse [83].

S-MAC s-a dovedit a fi un protocol extrem de bun pentru rețele de senzori, evident în aplicațiile unde nu sunt necesare facilități de timp real. De aceea protocolul a fost îmbunătățit rezultând B-MAC [84] care practic îmbunătățește tot ceea ce oferă S-MAC, fiind apoi inclus în nucleul de operare TinyOS.

Tot pe baza lui S-MAC a apărut o nouă abordare în protocoalele de comunicare pentru rețele wireless de senzori. Până acum, abordarea logică, transmițătorul era cel ce inițiază transferul de informație, în sensul că atunci când un nod avea ceva de transmis acesta inițiază procedurile necesare. Noua abordare constă în ideea ca un nod receptor să inițieze transferul de informație, această tehnică fiind prezentă în literatură sub denumirea de *receiver initiated communication*. În această situație, când un nod este pregătit să primească date, anunță în vecinătatea lui și așteaptă ca un alt nod să-i transmită informație. Această abordare este prezentă în protocolul RI-MAC (Receive Initiated MAC) care în afară de această abordare se bazează pe mecanismele prezente în S-MAC respectiv B-MAC [85]. Tot din această serie face parte și DW-MAC (Demand Wakeup MAC) care aduce ca noutate față de RI-MAC ideea că nodurile sunt în marea majoritate a timpului în stare de *sleep*. Când un nod devine activ și este pregătit să recepționeze anunță vecinătatea lui despre această stare și dacă un alt nod are ceva de transmis acesta iese din starea de *sleep* și transmite. În cazul în care nimeni nu are nimic de transmis nodul ce s-a anunțat disponibil să primească trece din nou într-o stare de *sleep* [86].

### 4.3 Protocoale MAC bazate pe scheme de acces fără coliziuni

Protocoalele de acces la mediu bazate pe scheme de acces fără coliziuni sunt acele protocoale care conform clasificării din secțiunea 4.1 fac parte din categoria protocoalelor bazate pe planificarea transmisiilor.

Din categoria protocoalelor bazate pe planificarea transmisiilor fac parte protocoale de tip TDMA (Time Division Multiple Access). Aceste protocoale împart timpul astfel încât

fiecărui nod  $i$  se atribuie o cantitate de timp în care poate transmite fără să se suprapună peste transmisia altor noduri. Acest tip de protocoale oferă un grad mare de predictibilitate dar problemele apar în faza de implementare din principalul motiv că este necesară sincronizarea ceasului intern al tuturor nodurilor rețelei. Sincronizarea timpului se poate face atât prin protocoale existente specializate [46] sau prin folosirea unei topologii în care rețeaua este sincronizată cu ajutorul nodurilor de tip coordonator [87, 88].

Dintre cele două categorii, în cazul unui sistem în timp real, a doua categorie de protocoale prezintă cel mai mare interes. Lucrarea de față se va concentra astfel asupra protocoalelor de tip TDMA.

Protocoalele de tip TDMA au la bază metoda de multiplexare în timp TDM (Time-Division Multiplexing). Această schemă a fost folosită pentru prima dată în sisteme de telecomunicații telegrafice de către Jean-Maurice-Émile Baudot în anul 1870. Conceptul de TDMA a avut însă cea mai pregnantă aplicabilitate în sistemele digitale de transmisie prin satelit [89] și mai ales în sistemele de telefonie mobilă [90].

Protocoalele bazate pe metoda de multiplexare TDMA se pot aplica pentru orice mediu de comunicație cu sau fără fir. În cazul unei scheme TDMA se consideră faptul că mai multe dispozitive sunt conectate la același mediu de comunicație comun. De asemenea se consideră că ceasul intern al participanților la comunicație este sincronizat. Astfel, într-un protocol bazat pe TDMA, mediul de comunicație este împărțit în sloturi egale de timp și fiecărui dispozitiv participant la comunicație, denumit nod,  $i$  se atribuie un astfel de slot pe o durată de timp limitată sau nelimitată. Lungimea unui slot TDMA este dată de durata de timp necesară unui nod pentru a transmite un pachet de date. Capacitatea unui slot TDMA este astfel egală cu un pachet de date [91].

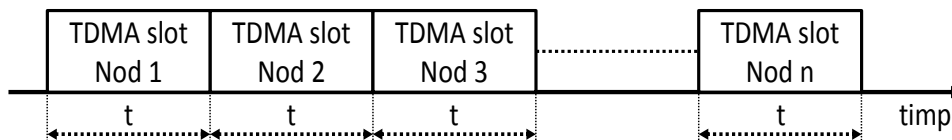


Figura 4-1 Divizarea timpului în sloturi TDMA

În Figura 4-1 este prezentat modul de divizare al timpului într-o schema TDMA. Fiecărui nod al rețelei îi este asignat un slot TDMA. Astfel fiecare nod are dreptul de a transmite informație doar în slotul TDMA alocat. De asemenea se poate observa că fiecare slot TDMA are aceeași durată care se alege de obicei ca fiind timpul transmisiei unui pachet de date de dimensiune maximă. Construcția prezentată în Figura 4-1 este repetitivă și formează un cadru (frame) TDMA.

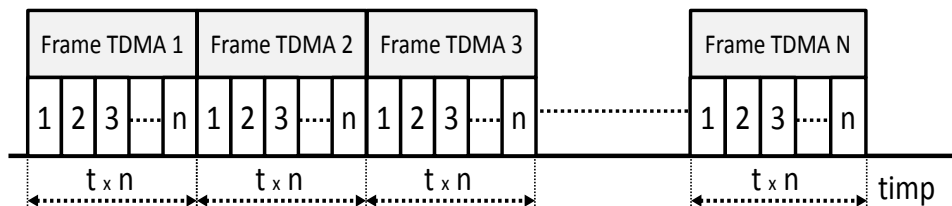


Figura 4-2 Divizarea timpului în frame-uri TDMA

Așadar, unitatea de bază în comunicarea într-o schema TDMA este cadrul (frame-ul) TDMA. În cadrul unui frame TDMA fiecare nod al rețelei are șansa de a comunica

fără a apărea coliziuni pentru că fiecare nod are un slot de timp alocat în fiecare frame TDMA așa cum se arată în Figura 4-2. De asemenea, se poate observa că dimensiunea unui cadru TDMA depinde strict de numărul de noduri ale rețelei și de dimensiunea unui slot de timp TDMA:

$$F_s = t \cdot n \quad (4-1)$$

Prin egalitatea (4-1) se definește dimensiunea unui cadru TDMA, notată prin  $F_s$  unde  $t$  este dimensiunea unui slot de timp TDMA și  $n$  numărul de sloturi de timp dintr-un cadru TDMA.

Principalele avantaje ale unui protocol TDMA sunt în primul rând legate de predictibilitatea comunicației deoarece printr-o asemenea schemă se poate oferi ușor o comunicație în timp real la nivelul unei rețele de senzori. Principalul dezavantaj al TDMA este acela că este ineficient în cazul în care unele noduri ale rețelei transmit date foarte rar și astfel ocuparea canalului de comunicație este mică. De asemenea cu cât rețeaua este mai mare cu atât numărul sloturilor de timp dintr-un cadru TDMA este mai mare ceea ce poate duce la timpi mari de comunicație. Totuși cel mai mare dezavantaj al unui protocol TDMA este necesitatea sincronizării nodurilor rețelei. Acest dezavantaj poate fi ușor trecut cu vederea în cazul unei rețele de senzori organizate ierarhic sub formă de clustere în care un nod coordonator sincronizează celelalte noduri ce le are în subordine prin transmisii periodice de mesaje de tip baliză [36, 92].

Multe dintre îmbunătățirile ce se aduc protocoalelor bazate pe TDMA sunt orientate pe reducerea timpilor de comunicație în cazul unei rețele cu număr mare de noduri. De asemenea unele studii s-au concentrat și pe eliminarea timpilor de inactivitate în care unele noduri nu au date de transmis, slotul de timp alocat acestora rămâne nefolosit în contrast cu alte noduri care au un volum mare de date de transmis dar un singur slot de timp pe cadru TDMA nu este suficient.

În cazul rețelilor de senzori cu comunicație fără fir în care modulele radio au posibilitatea de a comunica pe mai multe canale de frecvență [93, 94] există tendințe de îmbunătățire a ratei de transfer și de micșorarea unui cadru TDMA prin folosirea unor protocoale care să îmbine tehnica TDMA cu tehnica FDMA (Frequency Division Multiple Access). O schemă de tip FDMA se încadrează în clasa protocoalelor bazate pe multiplexarea mediului prin divizarea acestuia în canale de frecvențe diferite.

Așadar folosirea unui protocol care să combine cele două metode ar putea duce la rezultate mai bune în termeni de timpi de răspuns și lățime de bandă față de clasicul TDMA. Un asemenea exemplu ar fi protocolul RTH-MAC (Real Time Hybrid MAC) [81]. O rețea care comunică folosind acest protocol este în general o rețea formată din noduri statice cu organizare ierarhică sub formă de clustere cu coordonator orientată pe aplicații de agregare de date cu unul sau mai multe noduri cu funcție de data concentrator (sink). Nodurile rețelei sunt sincronizate în timp. Mediul de comunicație este împărțit atât în mai multe canale de frecvență cât și în sloturi de timp apărând astfel îmbinarea celor două metode de multiplexare. Dezavantajul folosirii unui asemenea protocol este acela că în primul se restricționează tipul de topologie ce se poate folosi, nu se pot folosi noduri mobile iar nodul ce coordonează întreaga rețea, nodul BS (Base Station) trebuie să țină în memorie întreaga configurație a rețelei, întreaga topologie. În plus, folosirea unor module radio care să poată funcționa pe mai multe canale de frecvență poate ridica semnificativ costul unui nod. Totuși, poate cea mai mare problemă este aceea că protocolul nu ține cont întârzierile foarte mari ce pot



apărea la schimbarea frecvenței de comunicare a modulului radio ceea ce ar putea introduce o mare impredictibilitate întregului sistem.

Tot o combinație între tipurile de acces la mediu este prezentată și în [95] unde se introduce protocolul Z-MAC (Zebra MAC) care folosește o schemă de comunicare bazată pe CDMA și TDMA. Fiecare nod al rețelei are un slot TDMA asigurat dar accesul la sloturi se face prin politica CDMA. Orice nod poate comunica peste sloturile deținute de alte noduri dar cu limitarea că nodul proprietar al unui slot de timp are prioritate maximă de comunicare în cadrul slotului. Chiar și așa, proprietarul slotului de timp, în momentul în care are date de transmis în propriul slot de timp accesează mediul de comunicație printr-o operațiune CCA ceea ce reduce din start gradul de predictibilitate al comunicării. Autorii acestei lucrări împart zonele de comunicare în zone cu nivel mic de coliziuni și zone cu nivel mare de coliziuni prin mecanisme de detecție a coliziunilor. După cum se știe, în comunicația fără fir nu este posibil să se detecteze coliziunile așa cum se poate face în cazul comunicației cu fir. Așadar metodele adiționale care încearcă să detecteze coliziunile nu pot garanta detectarea unei astfel de situații. Autorii protocolului Z-MAC încearcă detectarea coliziunilor prin măsurarea timpilor de *backoff* în cazul unei operațiuni de CCA dar această soluție nu poate garanta apariția unei coliziuni pentru că operațiunea CCA funcționează pe bază de măsurare de zgomot pe canalul respectiv. Un zgomot pe canalul de comunicație nu înseamnă neapărat coliziune ci poate foarte bine să fie o perturbație prezentă. De asemenea, se mai încearcă o detecție a coliziunilor prin detectarea pachetelor de date eronate. Din punct de vedere al timpului real Z-MAC nu poate fi folosit din principalul motiv că folosește politici de acces bazate pe analiza coliziunilor.

În standardul de acces la mediu cel mai des folosit în rețele de senzori, IEEE 802.15.4 există o îmbunătățire care folosește într-un mod foarte limitat, ideea de TDMA prin definirea sloturilor GTS (Guaranteed Time Slot) [96]. O îmbunătățire directă a standardului este propusă în [97] prin introducerea protocolului RRMAC în care se consideră o rețea cu o topologie arborescentă cu nod coordonator organizată conform standardului. Protocolul asignează sloturi de timp TDMA doar acelor noduri care au de transmis pachete sensibile din punct de vedere al timpului. Se modifică astfel superframe-ul IEEE 802.15.4 iar sincronizarea timpului este rezolvată prin mesaje periodice de tip baliză. În plus, se introduce în superframe-ul IEEE 802.15.4 o perioadă de timp, implementată folosind TDMA și denumită *beacon only period*, pentru accesul nodurilor coordonator în care fiecare să-și poată emite propria baliză în slotul de timp alocat. Deși protocolul aduce o îmbunătățire, aceasta nu este suficientă și nu rezolvă problemele de timp real.

În ultima perioadă a apărut în literatură un concept care poate fi folosit pentru a îmbunătăți politica de acces TDMA cu menținerea avantajelor de timp real pe care le oferă aceasta. Conceptul tinde să îmbunătățească situația în care unele noduri au de transmis volume mari de date pe când altele transmit rar puține date. Această situație duce, după cum s-a prezentat și mai sus, la existența unor sloturi de timp din cadrul TDMA care nu sunt folosiți o perioadă îndelungată și astfel capacitatea de transmisie de date poate să scadă drastic. Conceptul, sub denumirea generică de *slot stealing* se referă la posibilitatea unui nod care are un volum mare de date de transmis să "fure" un slot de timp al unui nod care nu trebuie să transmită date. Astfel capacitatea de transmisie de date se poate îmbunătăți. Conceptul introduce și probleme cum ar fi politica de a *fura* sloturile de timp.

Un prim exemplu de protocol ce folosește acest concept este TDMA-ASAP (TDMA with Adaptive Slot-Stealing and Parallelism) prezentat inițial în [98] și îmbunătățit mai târziu în [99]. Autorii îmbunătățesc capacitatea de transmisie a datelor într-o politică TDMA prin permiterea anumitor noduri să emită atât în slotul de timp alocat lor cât și

în sloturile de timp alocate altor noduri ce nu au date de transmis în cadrul TDMA curent. Totuși restricțiile impuse sunt mari. Se consideră pentru început notația cu Y a nodului care poate emite și fura slotul de timp la unui nod X. Protocolul se poate aplica strict doar în rețelele organizate pe clusetere cu un nod coordonator. De asemenea noduri X și Y trebuie să fie vecini direcți, cu alte cuvinte, cele două noduri trebuie să poată comunica direct. În plus nodurile X și Y trebuie să facă parte din același cluster, adică să aibă același nod părinte. Mai mult decât atât se consideră o topologie fixă.

Tot pe aceeași idee se prezintă și sTMDA (Stealing TDMA) unde un nod poate accede la sloturile de timp ale altor noduri dar pe bază de cerere. Deși se îmbunătățește utilizarea cadrului TDMA autorii introduc o serie de mecanisme prin care se trimit pachete suplimentare între noduri pentru a se putea realiza mecanismul de *slot stealing* acest lucru introducând mult trafic adițional [100].

Un protocol care folosește aceeași idee dar are în vedere informații venite de la protocoalele superioare din stiva de protocoale, în speță informațiile provenite de la protocoalele de rutare, este și TAS-MAC (Traffic-Adaptive Synchronous MAC) în care nodurilor rețelei le sunt asignat sloturi de timp din cadrul TDMA în funcție de cerințele de trafic și de rutele active [101]. Validarea protocolului se face prin medii de simulare care modelează destul de succint parametrii din pagina de catalog ale modulelor radio CC2420.

#### **4.4 Standardul IEEE 802.15.4**

Multe dintre platformele de senzori wireless apărute în ultima perioadă folosesc protocoalele de comunicare oferite de stiva ZigBee. Aceasta stivă de protocoale a fost concepută în 1998 și standardizată apoi în 2003 cu o importantă revizuire în 2006. Standardul ZigBee oferă o soluție finală de comunicare pentru rețele de tip PAN în special pentru aplicații specifice platformelor de senzori wireless cum ar fi: monitorizarea și controlul mediului industrial, sisteme de securitate, sisteme folosite în case inteligente precum și în diferite operațiuni de automatizare a unor procese. Standardul a fost dezvoltat și este în continuare întreținut de un consorțiu industrial denumit ZigBee Alliance. Acesta este și singura entitate de certificare a produselor ce folosesc standardul ZigBee [102].

Raportându-ne la stiva OSI, standardul ZigBee definește nivelul rețea și un nivel aplicație. Nivele de acces la mediu, respectiv nivelul fizic și nivelul data-link, sunt asigurate de standardul IEEE 802.15.4. După cum se observă stiva ZigBee nu definește și nivelele superioare nivelului rețea (cu excepția nivelului aplicație). De asemenea standardul asigură comunicarea între nodurile aceleși rețele nu și între rețele diferite, mai precis asigură doar o comunicare intra-PAN.

Prin nivelul rețea, stiva ZigBee oferă posibilitatea ca o rețea de senzori să fie structurată sub formă de topologii precum: topologie stea, arbore sau mesh. În situația în care o rețea este organizată printr-o topologie de tip stea, nodul central este un nod de tip ZigBee coordonator care are rolul de a iniția și configura rețeaua și de a asigura comunicația cu nodurile periferice. Într-o rețea de tip arbore nodul coordonator este responsabil pentru inițierea și configurarea rețelei, mai exact pentru configurarea nodurilor de tip router care la rândul lor devin mici coordonatori pentru nodurile inferioare ale arborelui. Nodurile de tip router și nodurile de tip coordonator sunt cele responsabile cu asigurarea rutării informației.

Din punct de vedere al adresării, fiecare modul din cadrul unei rețele ZigBee are mai multe adrese: adresa PAN, adresa de rețea și adresa extinsă. Adresa PAN este un identificator pe 16 biți a rețelei PAN din care modulul face parte. Doar modulele care au aceeași adresă PAN pot comunica între ele. Acest lucru face posibilă existența a mai multor rețele ZigBee, independente, în aceeași locație. Fiind la nivel de rețea această adresă PAN nu are efect și asupra frecvențelor folosite de nivelul fizic. După ce un nod se atașează unui coordonator sau a unui router din aceeași rețea acesta primește o adresă pe 16 biți, adresa de rețea. Această adresă este valabilă pe toată durata de viață a rețelei până la următoarea reconfigurare. De asemenea, în general, fiecare modul mai dispune de o adresă proprie extinsă, de obicei pe 64 biți care în cele mai multe situații este specifică fiecărui modul în parte.

Se face convenția că într-o rețea ZigBee nu poate exista decât un singur modul de tip coordonator și acesta va avea adresa de rețea, fixă cu valoarea 0x0000. De asemenea pentru transmisie de tip broadcast adresa de rețea a modulului destinație va avea valoarea 0xFFFF iar valoarea extinsă în acest caz este 0x000000000000FFFF.

După cum s-a menționat mai sus, nivelele de acces la mediu, respectiv nivelul fizic și nivelul datalink sunt oferite de standardul IEEE 802.15.4. Standardul este practic derivat din clasicul IEEE 802.11 folosit pentru rețelele WLAN, modificându-l astfel încât să ofere facilitățile necesare rețelelor de senzori. Protocolul definește atât nivelul 1, nivelul fizic (PHY), cât și nivelul 2 (datalink), nivelul de acces la mediu (MAC).

Referitor la nivelul fizic, acesta specifică frecvențele folosite în comunicație precum: modulația, viteza de transfer și politica de acces la mediu (cu sau fără coliziuni). Nivelul fizic specifică în principiu două benzi de frecvență: banda de 2.4 GHz și banda de 915 MHz și împarte aceste benzi în 26 de canale de frecvență. Primele 10 canale sunt asignate benzii de 915 MHz. Primul canal are asignată frecvența de 906 MHz iar ultimul din această bandă are frecvența de 924 MHz cu o distanță de 2 MHz între canale. Următoarele canale, de la 11 la 26 se referă la banda de 2.4 GHz. Canalul 11 începe de la frecvența de 2.405 MHz iar ultimul canal are o frecvență de 2.480 MHz. Distanța între canale în banda de 2.4 GHz este de 5 MHz.

Această secțiune, de nivel PHY, nu prezintă interes lucrării de față, fiind necesară doar în cazul realizării unui modul de radio-frecvență care să respecte aceste cerințe, ceea ce ar implica realizarea unor module deja existente și folosite cu succes.

Prima variantă a standardului [103] (ediția din 2003) trasează primele elemente de organizarea unei rețele wireless de senzori, aceste elemente fiind folosite atât de companii în realizarea spre vânzare a diferitelor soluții integrate de comunicare wireless pentru rețele de senzori cât și de către cercetătorii din mediul academic cu scopul de a aduce îmbunătățiri. Din aceste motive acest standard va fi succint prezentat în paragrafele următoare.

Entitatea de bază definită este *nodul*. Standardul definește 2 tipuri de noduri: noduri de tip FFD (Full Functional Device) și RFD (Reduced Functional Device). Nodurile de tip RFD sunt noduri cu funcționalitate redusă din punct de vedere al comunicării. Acestea pot fi asemănată cu nodurile de tip CN dintr-un cluster. Acestea sunt noduri mai mult cu funcție de monitorizare și control a mediului decât cu funcții evaluate de comunicare. Acestea nu pot realiza funcții de rutare a informației, nu pot fi conducători de cluster și nu li se permite decât să comunice cu coordonatorul subrețelei (a clusterului) din care fac parte. Nodurile de tip FFD sunt noduri cu funcții evaluate de comunicare. Acestea pot oferi și funcții de monitorizare și control a mediului dar în același timp au ca scop principal realizarea unor funcții adiționale în ceea ce privește comunicarea: coordonator PAN (Personal Area Networks), rutare, transmisie baliză, sincronizare, etc. Noțiunea de PAN definită de standard poate fi asemuită cu noțiunea de cluster, fiind de fapt o subrețea. Coordonatorul de PAN poate fi identificat cu nodul

de tip CH. Practic acesta și nodurile pe care le coordonează formează acest PAN. Restricția care se aplică este aceea că nodurile de tip RDF nu au voie să discute decât cu nodul FFD de care aparțin [103]. Aceste reguli au la bază ideea că nodurile de tip RFD pot fi sisteme embedded extrem de slabe în resurse, cu necesitatea de a funcționa o perioadă lungă de timp pe baterie iar nodurile de tip FFD fiind sisteme embedded, sau sisteme de calcul clasice, la care consumul de energie electrică nu este pe primul loc, pot fi chiar și sisteme alimentate de la rețea.

În funcție de aplicație standardul prevede 2 tipuri de topologii: o topologie de tip stea în care centrul este reprezentat de un nod de tip FFD iar frunzele sunt reprezentate de noduri de tip RFD și o topologie de tip mesh sau peer-to-peer în care mai multe noduri sunt de tip FFD dar care nu toate trebuie să îndeplinească funcție de coordonator ci, din punct de vedere al comunicării, se ocupă cu rutarea informației între noduri.

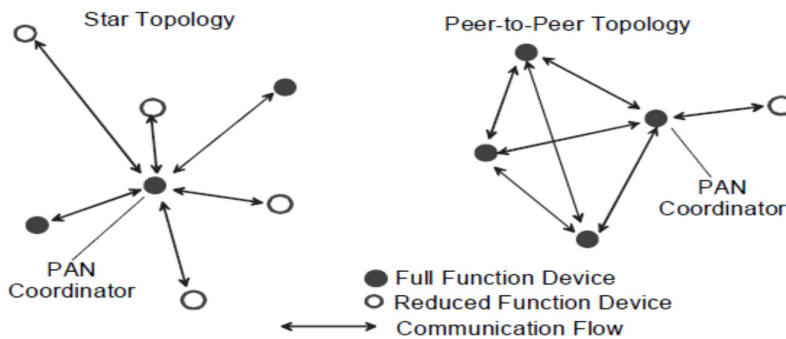


Figura 4-3 Topologiile definite de standardul IEEE 802.15.4 (sursa: [103])

În Figura 4-3 [103], preluată din documentul standardului IEEE 802.15.4 ediția 2003, se pot observa cele 2 topologii permise de standard. Interesant este faptul că într-o rețea pot exista mai multe noduri de tip FFD dar care nu este obligatoriu să îndeplinească funcția de coordonator de PAN, ele fiind folosite în topologia peer-to-peer pentru a realiza rutarea informației și a permite comunicarea și cu alte noduri în afară de nodul coordonator.

O altă caracteristică definită de standard este aceea de clusterare în interiorul unui PAN, situație exemplificată de standard în Figura 4-4 (preluată din standard) [103]:

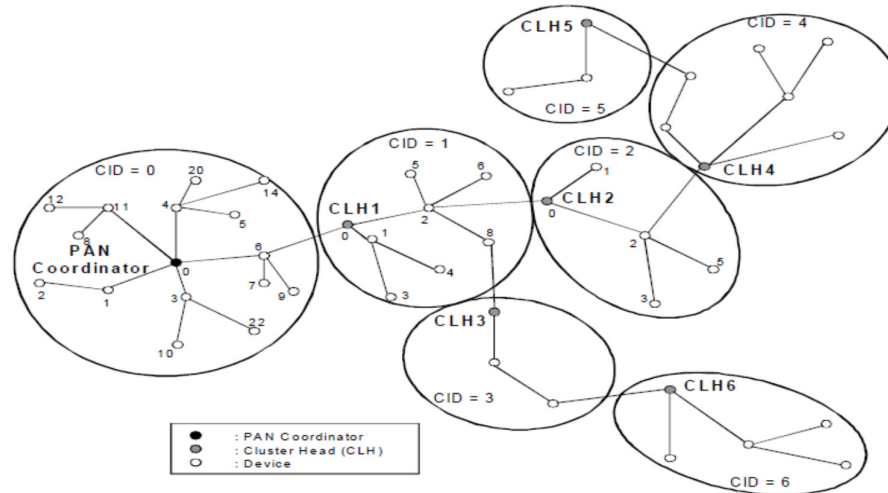


Figura 4-4 Exemplu de organizare pe clustere în IEEE 802.15.4 (sursa: [103])

Partea interesantă în structura de mai sus este aceea că toată rețeaua din figură constituie un PAN, care este necesar a fi un nod de tip FFD, care în același timp este și coordonator pentru propriul cluster. Acesta deci îndeplinește atât funcția de conducător de cluster cât și cea de conducător al întregului PAN. Celelalte clustere sunt conduse de câte un nod de tip CH care implicit după standard trebuie să fie de tip FFD. De asemenea se poate observa că și în interiorul unui cluster, în afară de coordonatorul de cluster, există noduri de tip FFD, acestea fiind necesare pentru realizarea funcției de rutare a informației (spre exemple în clusterul 0, nodul 6; în clusterul 1, nodul 2 și nodul 8; etc.)

La nivelul comunicării nodul ce îndeplinește funcție de coordonator (PAN sau cluster) conduce comunicarea prin intermediul unei balize, acesta stabilind astfel baza de timp. Distanța în timp dintre 2 balize constituie o structură de tip *superframe*, situație exemplificată în standard prin Figura 4-5 [103]:

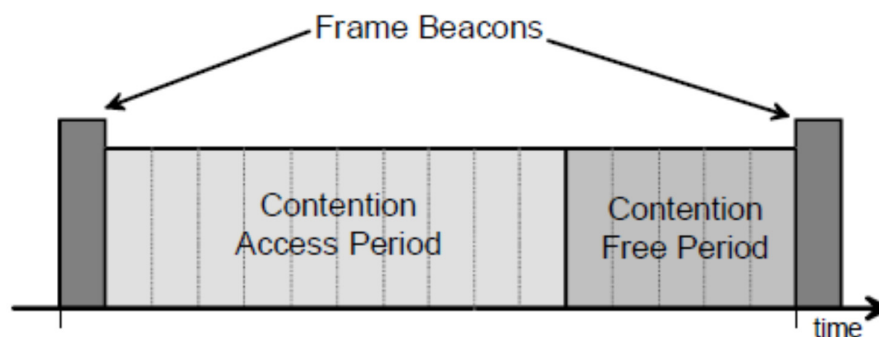


Figura 4-5 Structură de superframe definit de IEEE 802.15.4 (sursa: [103])

Un astfel de superframe este compus din 16 cuante de tip (sloturi) egale ca dimensiune în timp. Doar în cadrul unei astfel de cuante de timp un nod poate să

comunicare. Modul în care un nod reușește să obțină mediul ca să poată comunica în cadrul unui slot este realizat în tehnici bazate pe coliziuni: CSMA-CA (Carrier Sense Multiple Access – Collision Avoidance). Restricția este doar aceea că nodurile nu au dreptul de a iniția comunicație decât la începutul unui slot de timp și dacă un nod a început să emită la începutul slotului și nu a fost coliziune, obținând astfel mediul de comunicare, nici un alt nod nu are dreptul de a comunica pe toată durata aceluia slot de timp. Aceste sloturi de timp, conform standardului, pot fi împărțite în două tipuri: sloturi de timp cu acces bazat pe coliziuni și sloturi de timp garantate – GTS (Guaranteed Time Slots). Astfel, conform figurii, superframe-ul se împarte în CAP (Contention Access Period) și CFP (Contention Free Period) [103].

În perioada CAP obținerea mediului se face prin mecanisme bazate pe coliziuni, mai exact prin CSMA/CA.

Pentru această lucrare prezintă interes perioada CFP, perioada CAP având un mare grad de impredictibilitate din cauza mecanismelor de acces la mediu bazate de coliziuni. Pe de altă parte perioada CFP oferă un mic suport de timp real în ceea ce privește comunicația. Sloturile ce aparțin perioadei CFP sunt definite de standard ca fiind GTS în sensul că după ce un nod obține un astfel de slot, acesta este garantat pe toată perioada comunicării. Aici poate apărea problema obținerii acestui slot (mecanism realizat prin algoritmi bazați pe coliziuni) și de asemenea poate apărea problema introdusă în mod direct de standard și anume ca nu pot exista decât maxim 7 sloturi de tip GTS într-un superframe, limitând astfel caracteristicile de timp real ale întregii rețele [103]. O îmbunătățire în atribuirea sloturilor garantate de către coordonator este prezentată în [87] în care metoda prezentată de autori oferă garanții soft real time în ceea ce privește transmiterea în rețea a mesajelor periodice. Există și situații în care folosind sloturile GTS se poate ajunge la o comunicare predictibilă, în timp real, dar numai după o analiză în prealabil, offline, a necesităților de comunicare a rețelei și astfel să se planifice transmiterea mesajelor conform cu rezultatele analizei. Această situație funcționează doar într-o rețea în care configurația nu se schimbă oferind rețelei un grad foarte mare de rigiditate și fără a fi scalabilă [104].

În această primă variantă a standardului nu se specifică nimic legat de reducerea consumului de energie, mai mult este necesar ca nodurile cu funcție de coordonator să-și țină modulul radio pornit pe toată durata funcționării ceea ce scurtează durata de viață a nodului. O posibilă rezolvare a acestei probleme apare în ediția următoare a standardului, în 2006. Această ediție specifică faptul că un superframe poate fi împărțit în 2 tipuri: superframe cu sloturi de tip GTS și superframe fără sloturi de tip GTS. În situația fără sloturi GTS, perioada corespunzătoare lor este înlocuită cu o perioadă în care nodul coordonator poate să intre în moduri de funcționare cu consum redus după cum se poate vedea în Figura 4-6 [96]. De asemenea în perioada CAP, unde accesul la mediu se face prin tehnica CSMA/CA, în cazul în care un nod nu reușește să obțină mediul de comunicare, acesta intră din nou într-un mod de consum redus în așteptarea eliberării mediului de comunicație, și reîncearcă în următorul *masterframe*. Acest lucru poate să ducă la un timp foarte lung de obținerea accesului la mediu, iar o posibilă rezolvare este dată de 2 metode bazate pe priorități prezentate în [105].

Reducerea consumului de energie este și mai mult îmbunătățită în [106] unde nodurilor ce comunică după standardul IEEE 802.15.4 li se adaugă mecanismul de DMS și se modifică dinamic puterea de emisie adăugând un plus ediției din 2006 a standardului [96].

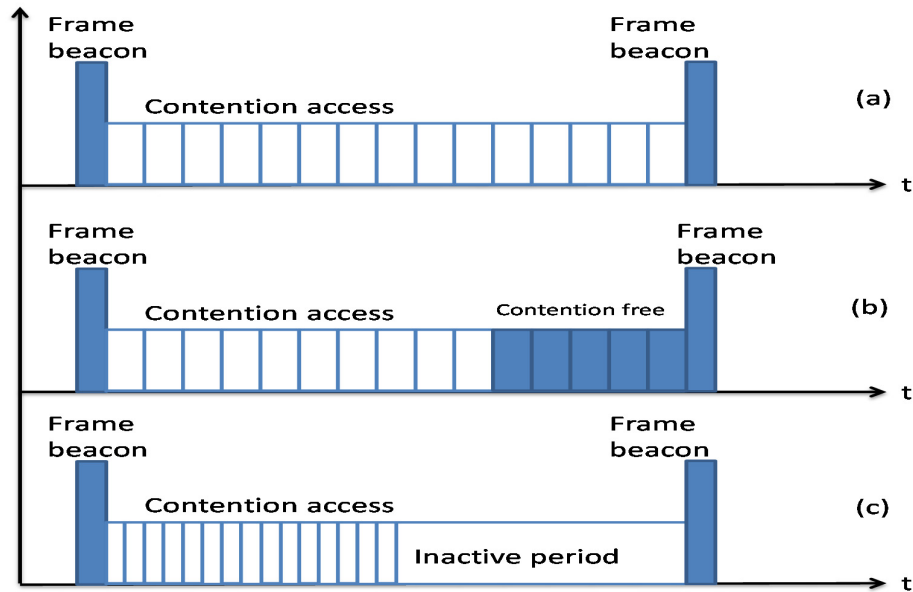


Figura 4-6 Structură de superframe definită în de IEEE 802.15.4 ediția 2006 (a) cu perioadă fără coliziuni (b) cu perioadă de inactivitate

Apare totuși limitarea directă în standard că dacă se folosesc sloturi garantate atunci nu poate exista perioadă în care coordonatorul sau nodul de tip router să intre în mod de consum redus de energie și invers.

Edițiile următoare ale standardului nu mai completează aceste noțiuni, referitoare la nivelul MAC ci se ocupă doar de nivelul fizic; în [107] se adaugă noi specificații nivelului fizic (noi modulații și frecvențe) iar în edițiile [92, 108] se adaugă doar noi benzi de frecvențe nivelului fizic.

Standardul IEEE 802.15.4, care se referă și la nivelul rețea din stiva de protocoale, definește și un sistem de împachetare a informației. Astfel structura generală a unui pachet definit de standard este:

Tabel 4-1. Structura generală a unui pachet IEEE 802.15.4 (sursa: [92])

Field number	Field size [bytes]	Field description
1	1	Frame length
2	2	Frame control field (FCF)
3	1	Sequence number
4	0/2	Destination PAN identifier
5	0/2/8	Destination address
6	0/2	Source PAN identifier
7	0/2/8	Source Address
8	n	Frame payload
9	2	Frame checksum

Primul câmp al pachetului reprezintă lungimea întregului pachet pe o dimensiune de 1 octet.

Al doilea câmp și poate cel mai important câmp al pachetului este reprezentat de FCF (Frame Control Field). Acest câmp, cu o dimensiune de 2 octeți, decide exact cum va fi structura pachetului în cauză. Printre cele mai importante aspecte ce sunt decise de către acest câmp sunt: tipul pachetului, dacă se folosește sau nu securitate, dacă se cere sau nu confirmare, dacă este sau nu pachet segmentat, dacă este un pachet inter-PAN sau intra-PAN (dacă se transmite în aceeași rețea sau între rețele diferite) și modul de adresare a nodului sursă precum și modul de adresare a nodului destinație.

Pentru tipul pachetului sunt rezervați de către standard un număr de 3 biți, existând astfel 8 tipuri diferite de pachete posibile. Standardul definește doar 4 tipuri de pachete restul fiind rezervate. Tipurile de pachete definite de standard sunt următoarele:

- Pachet de tip *beacon* (*baliza*). Acest pachet este trimis de către coordonator (sau router după caz) la începutul fiecărui superframe și are atât rol de sincronizare cât și de a transmite specificații generale ale rețelei către celelalte noduri.
- Pachet de tip *data*. Acest pachet este rezervat pentru transmisiile de date și este folosit de către nivelele superioare ale comunicației.
- Pachet de tip *ack*. Acest pachet este folosit pentru confirmare
- Pachet de tip *MAC command*. Acest pachet este destinat pentru a trimite comenzi specifice nivelului rețea între nodurile rețelei.

După cum se observă în Tabel 4-1 fiecare pachet adresează atât nodul sursă cât și nodul destinație. Această adresare poate include și adresa rețelei (a PAN-ului) din care nodul sursă sau destinație face parte. Modul de adresare atât a nodului de destinație cât și a nodului sursă este descris de 2 câmpuri specifice de dimensiune 2 biți fiecare. Astfel, prin FCF se pot specifica următoarele moduri de adresare (atât pentru sursă cât și pentru destinație):

Tabel 4-2. Modurile de adresare specificare de Frame Control Field

Valoare câmp mod adresare $b_1b_0$	Descriere mod adresare
00	Câmpul de adresă este nespecificat
01	Rezervat
10	Câmpul de adresare conține adresa PAN și adresa scurtă, pe 16 biți a nodului
11	Câmpul de adresare conține adresa PAN și adresa extinsă, pe 64 de biți a nodului

Următorul câmp din pachetul general este câmpul ce conține un număr de secvență pe 1 octet. Acest câmp este necesar pentru a asigura o secvențiere corectă a pachetelor în comunicare conform specificațiilor unui protocol de nivel rețea. Câmpurile 4,5,6 și 7 care sunt hașurate în Tabel 4-1 au o dimensiune variabilă în funcție de modul de adresare specificat în câmpul FCF conform cu Tabel 4-2.



Penultimul câmp este câmpul de date propriu zis de o dimensiune variabilă.

Întregul pachet este protejat de o sumă de control pe 2 octeți transportată de ultimul câmp al pachetului. Suma de control este de fapt generată folosind algoritmul CRC16. Acest câmp este necesar pentru a asigura validitatea întregului pachet.

În standard mai este prevăzut faptul că înainte de a se trimite orice pachet conform structurii din Tabel 4-1 mai este necesar să se trimită o secvență denumită SHR (Synchronisation Header). Această secvență presupune două componente. O prima componentă este reprezentată de o secvență de preambul de 4 octeți (32 de biți de 0) necesară pentru a sincroniza nodurile din vecinătate și de a le anunța despre transmisia pachetului. În cazul în care nodurile sunt într-un mod cu consum redus, cu modulul radio într-o stare latentă, secvența de preambul are și rolul de a reactiva nodurile respective pentru a putea face față recepției noului pachet. A doua componentă a SHR este reprezentată de secvența SFD (Start Of Frame Delimiter) care are rolul de a încheia secvența de sincronizare. Câmpul de SFD are o lungime de 1 octet cu o valoare fixă 0xA7. Pentru a înlătura sincronizările false standardul prevede faptul că după ce un modul a recepționat secvența de sincronizare, pentru a începe recepționarea pachetului propriu zis de date, acesta trebuie să distingă cu succes și octeții din SFD. În caz contrar va ignora orice recepție până la următoarea recepție a 32 de biți de 0 consecutivi.

## 4.5 Concluzii

Referitor la nivelul MAC al comunicării în rețelele wireless de senzori, din câteva exemple prezentate anterior, se pot trage mai multe concluzii. Pentru început se poate constata faptul că majoritatea protocoalelor de tip MAC se orientează mai mult asupra consumului de energie decât asupra garanțiilor de timp real. Chiar și protocoalele MAC orientate spre aplicații ce oferă garanții de timp real se rezumă doar la garanții de tip soft real-time, putând astfel doar aproxima un trafic garantat statistic.

O problemă intens întâlnită este aceea că în marea majoritate a situațiilor cercetătorii își validează soluția folosind simulatoare iar în cazul unor protocoale de nivel jos precum cele de acces la mediu, simulatoarele, indiferent de complexitate, nu au capacitatea de a reda un mediu real de comunicație. Mai mult decât atât, penalitățile de timp introduse de modulul radio în sine sunt de cele mai multe ori ignorate considerând o funcționare ideală a modulului radio [109].

În multe situații se consideră chiar și cazuri particulare extrem de restrânse care nu pot fi acceptate în realitate (rețele în care poziția nodurile se cunoaște cu exactitate, adăugarea unui modul GPS fiecărui nod pentru localizare, comunicarea în medii lipsite de perturbații, măsurători doar în cazuri favorabile, etc.).

Majoritatea implementărilor actuale de rețele de senzori folosesc ca și pachet de protocoale de comunicație stiva ZigBee având ca protocol de acces la mediu standardul IEEE 802.15.4. Aceasta deși oferă o soluție integrată și funcțională pentru organizarea comunicației la nivelul unei rețele de senzori nu este orientată pe aplicații critice în timp real. Mai mult, protocolul de acces la mediu folosit de asemenea nu este gândit pentru acest tip de aplicații.

## **5 Analiză experimentală a protocolului ZigBee și a standardului IEEE 802.15.4 din perspectiva timpului real și a interoperabilității**

După cum s-a prezentat în capitolele anterioare stiva ZigBee având ca bază standardul IEEE 802.15.4 este folosit într-un număr mare de aplicații. De asemenea acesta este în continuare dezvoltat, corectat și întreținut de companiile de formează consorțiul ZigBee Alliance. În marea majoritate a aplicațiilor această stivă este adecvată, ea oferind practic o soluție integrată de comunicare pentru rețele de senzori.

Totuși se poate demonstra ușor că această stivă de protocoale nu se poate folosi în domeniul timp real și mai mult se poate arăta faptul că stiva ZigBee nu folosește în totalitate standardul IEEE 802.15.4 și astfel o coabitare între o rețea ZigBee și o altă rețea având la bază standardul IEEE 802.15.4 ar fi extrem de dificilă dacă nu chiar imposibilă. Scopul acestui capitol nu este acela de a anula și critica rezultatele și platforma oferită de stiva ZigBee ci de a arăta faptul că utilizarea acesteia în aplicații cu cerințe stricte de timp nu poate fi folosită. Mai mult decât atât, în varianta actuală a standardului IEEE 802.15.4 protocoalele oferite de acesta nu sunt adecvate unui mediu de comunicare în timp real.

În literatura de specialitate se pot găsi multe lucrări de cercetare în care se studiază lipsa de fiabilitate a standardului IEEE 802.15.4 în WSN, în special în sisteme de timp real.

Spre exemplu în lucrarea [110] autorii arată că în multe situații, chiar și într-o rețea formată din puține noduri, folosirea protocolului 802.15.4 poate să conducă la situații în care pierderea de pachete este foarte ridicată. Principala cauză ar fi mecanismele de reducerea consumului de energie prevăzute de standard. Autorii propun o soluție de a rezolva această problemă dar evident cu costuri suplimentare: o creștere impredictibilă a timpului de transmisie a mesajului de la sursă la destinație ceea ce nu este de acceptat într-un sistem în timp real. Aceeași problemă a fost dezbătută și în [111] unde îmbunătățirea s-a realizat prin optimizarea unor parametrii din partea de CSMA/CA a standardului dar cu aceleași dezavantaje ca și în situația anterioară. Este important de menționat faptul că în ambele lucrări protocolul oferit de standard a fost evaluat folosind atât medii de simulare cât și folosind module de comunicare reale cu scopul de a evidenția cât mai clar impredictibilitatea la care este supusă comunicarea în cazul folosirii acestui standard.

Fiind de o popularitate foarte mare, pentru a evalua protocolul acestui standard unii cercetători au creat un simulator dedicat [112] și din nou s-a evidențiat faptul că într-un mediu de timp real nu poate fi folosit chiar dacă dispune de extensia cu sloturile GTS. Folosirea acestor sloturi nu rezolvă problema în timp real fie din cauza faptului că accesul la sloturile GTS se face tot în manieră cu coliziuni fie pentru că aceste sloturi sunt puține și astfel nu pot fi utilizate când rețeaua are un număr mare noduri.

Pentru a realiza o analiză practică a protocolului ZigBee și a protocolului IEEE 802.15.4 s-au folosit două tipuri de module wireless [113]. Primul modul este reprezentat de XBee Series 2 [114] produs de Digi International Inc, acest modul fiind un modul certificat ZigBee de către autoritatea ZigBee Alliance. Acest modul are implementată stiva ZigBee peste care este dezvoltat un nivel aplicație specific producătorului prin care utilizatorul poate accesa stiva ZigBee. Celălalt tip de modul

## Analiză experimentală a protocolului ZigBee și a IEEE 802.15.4 59

wireless folosit este modulul CC2420 [93] produs de ChipCon sub Texas Instruments. Acest modul oferă hardware specializat pentru implementarea standardului IEEE 802.15.4 și este compatibil pentru ca utilizatorul să-și implementeze stiva ZigBee pe un procesor gazdă atașat modulului CC2420.

Rolul modulului XBee în cadrul experimentelor a fost acela de a fi o referință a stivei ZigBee în timp ce modulul CC2420 a fost folosit pentru a recepționa pachetele emise de modulul XBee spre a fi decodificate conform standardului IEEE 802.15.4.

Modulul XBee a fost configurat cu firmware de coordonator cu următoarele setări:

Tabel 5-1. Configurația modulului XBee pentru comunicație broadcast

Număr serial (adresa extinsă pe 64 biți)	0x0013A20040649EC2
Function set / version (versiune firmware)	ZigBee Coordinator AT / 1020
Canalul de frecvență	0xB (11 - 2.405 MHz)
Adresă destinație	0xFFFF (Broadcast)
Adresă PAN	0x0234

Pentru utilizarea modulului CC2420 s-a folosit un sistem tip MPR2400 MicaZ produs de Crossbow Technology [115]. Acesta este practic format dintr-un modul CC2420 și un microcontroller de tip Atmel ATMEGA128L [116]. În general aceste module folosesc ca și componente software sistemul de operare embedded TinyOS [14]. Pentru a putea evalua experimental cele propuse a fost necesară îndepărtarea acestei componente software și realizarea unui software adecvat acestui experiment. Acest software pe lângă gestionarea și configurarea modulluui CC2420 a mai avut rolul de a decodifica și afișa pachetele de tip 802.15.4 recepționate. Pentru a putea realiza acest lucru, modulul CC2420 a fost configurat și în aceeași bandă cu modulul XBee și i s-au dezactivat mecanismele de recunoaștere automată a adresei lăsând activ doar mecanismul de calcul hardware a sumei de control reprezentată prin polinomul CRC16.

Imediat după ce s-a pus în funcțiune modulul XBee acesta a început să emită, cu o cadență de 2 secunde, o baliză specifică protocolului ZigBee (nu o baliză specifică protocolului definit de 802.15.4). În următorul tabel se prezintă un exemplu de asemenea pachet capturat de către modulul CC2420:

Tabel 5-2. Captură pachet baliză ZigBee

Pachet ZigBee în format hexazecimal	19 41 88 37 34 02 FF FF 00 00 04 00 FF FF 00 00 01 68 08 F0 80 0E C0 F0 06
<b>Decodarea pachetului conform 802.15.4</b>	
Lungime pachet	25
Număr de secvență	55
Tip pachet	Data
Adresa PAN a modului destinație	0x0234
Adresa modulului destinație	FFFF (broadcast)
Adresa modulului sursă	0000 (coordonator)
Lungimea datelor utile	15
Date utile în format hexazecimal	04 00 FF FF 00 00 01 68 08 F0 80 0E C0 F0 06

## 60 Analiză experimentală a protocolului ZigBee și a IEEE 802.15.4 – 5

Făcând o analiză a acestui pachet se poate observa faptul că deși acest pachet este o baliză, stiva ZigBee folosește un pachet de tip data din 802.15.4 pentru a transporta această baliză și nu se folosește tipul de pachet special desinat acestui lucru oferit de protocol, tipul *beacon*. Se folosește totuși corect partea de adresă a standardului 802.15.4.

Faptul că acest pachet, care este de fapt o baliză, este încapsulat într-un pachet de date din punct de vedere al standardului duce la situația ca fiecare modul radio compatibil cu standardul să recepționeze acest pachet ca și un pachet de date și să-l interpreteze ca atare. Mai mult, dacă în aceeași locație se află atât o rețea ZigBee cât și o rețea compatibilă la nivel 802.15.4 fără stivă ZigBee pot apărea probleme atât de interoperabilitate cât chiar și de funcționare individuală a celor două rețele. Rețeaua ZigBee emite o baliză periodică specifică, sub forma unui pachet de date 802.15.4, acesta la rândul lui fiind interpretat ca atare de rețeaua fără stivă ZigBee ceea ce poate duce la o interpretare greșită și implicit la creșterea inutilă a consumului de energie.

Un alt set de experimente a avut ca scop urmărirea unor pachete de date utilizator emise de rețeaua ZigBee. Astfel s-a ales un pachet de date de dimensiune 3 octeți sub forma *0x61, 0x62, 0x63* (mai exact tripletul "abc"). În urma trimiterii acestui pachet de date utilizator de pe rețeaua ZigBee s-a constatat captura următorului pachet de către modulul CC2420:

Tabel 5-3. Captură pachet date ZigBee cu adresare broadcast

Pachet ZigBee în format hexazecimal	27 41 88 73 34 02 FF FF 00 00 04 00 FF FF 00 00 0F A4 08 E8 11 05 C1 E8 21 0A 0B 00 13 A2 00 40 64 9E C2 61 62 63 06
Decodarea pachetului conform 802.15.4	
Dimensiune pachet	39
Număr de secvență	115
Tip pachet	Data
Adresă PAN a modului destinație	0x0234
Adresă de rețea a modului destinație	FFFF (broadcast)
Adresă de rețea a modului sursă	0000 (coordonator)
Dimensiunea datelor utile	29
Date utile	04 00 FF FF 00 00 0F A4 08 E8 11 05 C1 E8 21 0A 0B 00 13 A2 00 40 64 9E C2 <b>61 62 63</b> 06
Date utilizator	61 62 63

Analizând acest pachet se poate constata faptul că din punct de vedere al standardului nu există nici o diferență între un pachet de tip baliză ZigBee și un pachet de date ZigBee. Practic stiva ZigBee reduce standardul la a folosi doar pachetul de tip data oferit de acesta. Un alt lucru esențial de observat este că secvența de octeți (în hexazecimal) *13A20040649EC2* este prezentă în câmpul de date al pachetului. Aceasta reprezintă adresa extindă a modului XBee (vezi Tabel 5-1) care conform standardului

## Analiză experimentală a protocolului ZigBee și a IEEE 802.15.4 61

se poate încapsula folosind unul din modurile de adresare a standardului, din nou observându-se o ocolire a acestui protocol.

În cele două experimente s-a folosit adresa de broadcast ca și adresă destinație. În această situație toate modulele compatibile 802.15.4 din raza de acțiune au putut recepționa aceste pachete. În următorul experiment se folosește o adresare exactă a unui modul realizându-se o comunicație end-to-end. Pentru aceasta modulul XBee este configurat astfel:

Tabel 5-4. Configurația modulului XBee pentru comunicație punct-la-punct cu un alt modul existent

Număr serial (adresa extinsă pe 64 biți)	0x0013A20040649EC2
Function set / version (versiune firmware)	ZigBee Coordinator AT / 1020
Canalul de frecvență	0xB (11 - 2.405 MHz)
Adresă extinsă a nodului de destinație	0x0013A20040521954
Adresă de rețea a nodului destinație	0x2E83
Adresă PAN	0x0234

Pachetul trimis astfel de coordonator păstrând același câmp de date utilizator este următorul:

Tabel 5-5. Captură pachet date ZigBee cu adresare directă

Pachet ZigBee în format hexazecimal	2F 61 88 6C 34 02 83 2E 00 00 44 00 83 2E 00 00 0F 32 00 E8 11 05 C1 E8 E1 EA 00 54 19 52 40 00 A2 13 00 00 13 A2 00 40 64 9E C2 61 62 63 09
<b>Decodarea pachetului conform 802.15.4</b>	
Dimensiune pachet	47
Număr de secvență	108
Tip pachet	Data
Adresă PAN a modulului destinație	0x0234
Adresă de rețea a modulului destinație	0x2E83
Adresă de rețea a modulului sursă	0000 (coordonator)
Dimensiunea datelor utile	36
Date utile	44 00 <b>83 2E</b> 00 00 0F 32 00 E8 11 05 C1 E8 E1 EA 00 <b>54</b> <b>19 52 40 00 A2 13</b> 00 00 <b>13 A2 00 40</b> <b>64 9E C2 61 62 63</b> 09
Date utilizator	61 62 63

## 62 Analiză experimentală a protocolului ZigBee și a IEEE 802.15.4 – 5

---

Acest pachet este tot un pachet de date 802.15.4. Se poate observa că se folosește o adresare scurtă pe 16 biți în pachetul de bază al standardului. Mai important de observat este că în datele utile adăugate de nivelul stivei ZigBee se regăsesc atât adresele extinse pe 64 de biți ale fiecărui modul cât și adresele de rețea pe 16 biți. Nu se poate vorbi despre o redundanță neapărat din punct de vedere al acestor adrese deoarece aceste adrese se referă la nodul final al comunicației. În cazul în care era necesară o operație de rutare, aceste adrese nu ar mai fi fost identice cu cele din câmpurile de adrese ale standardului 802.15.4.

Modul de împachetare al stivei ZigBee aduce o serie de dezavantaje în contextul comunicării într-o rețea de senzori wireless. În exemplul precedent s-a trimis un câmp de date utilizator de dimensiune 3 octeți iar stiva ZigBee a introdus o informație în plus de 33 de octeți necesari protocolului pe lângă cei deja 10 octeți introduși de protocolul standardului 802.15.4. Se observă un total de 43 octeți de informație adițională celor 3 octeți utili.

Problema în acest caz este în primul rând creșterea pachetului de date pe un mediu extrem de instabil și supus zgomotelor și perturbațiilor, cum este mediul wireless ceea ce poate duce la o scădere considerabilă a ratei de succes de transmisie a unui pachet de la sursă la destinație.

A doua problemă introdusă de modul de împachetare a datelor apare în momentul în care se dorește să se folosească module wireless mai slabe în resurse. De obicei, se tinde ca o rețea wireless de senzori să fie compusă dintr-un număr de sute sau chiar mii de noduri. În acest caz este de dorit ca costul per nod să fie destul de scăzut. Modulele radio folosite în experimente au fost CC2420 care pot transmite și recepționa un număr de maxim 128 octeți. În cazul în care s-ar folosi ca și modul radio un circuit de tip CC2500 [94], un circuit semnificativ mai ieftin și cu un consum de energie mult mai mic, această stivă de protocoale ar fi aproape imposibil de implementat din simplul motiv că aceste module radio pot transmite și recepționa maxim 64 octeți.

În concluzie, în cadrul unei rețele de senzori cu comunicație wireless, în care numărul nodurilor este ridicat și în care costul per nod se dorește a fi cât mai mic protocoalele de comunicare folosite trebuie să țină cont de aceste aspecte mai ales în contextul folosirii lor într-un mediu de timp real.

## 6 Model temporal și cadru de măsurare pentru comunicație wireless

Ca și în orice sistem în timp real și în comunicarea în timp real parametrul esențial este timpul. În capitolul 2.1 s-a arătat un model de a determina componentele de timp ale întârzierii produse de transmiterea unui pachet de la nodul sursă la nodul destinație. Modelul a fost descris în [48] din perspectiva rutării informației într-o rețea de senzori și astfel și componentele de timp ce formează întârzierea totală se referă la acest nivel din stiva de protocoale. Acest model poate fi adaptat și în cazul în care ne referim la nivelele ce compun accesul la mediu. În acest capitol se va prezenta un astfel de model temporal menit în primul rând să analizeze comportamentul temporal al penalităților de timp introduse de modulele de comunicare wireless. [117]

### 6.1 Model temporal pentru comunicația wireless

Pentru început trebuie făcute unele precizări pentru a defini modelul de descriere a parametrilor de timp. Se consideră faptul că nodurile rețelei sunt half duplex și deci au un singur modulator radio astfel încât nu pot recepționa și transmite mesaje simultan. Practic această presupunție nu face o discrepanță mare față de realitate deoarece majoritatea modulelor radio prezente pe piață, adecvate rețelelor de senzori, au această caracteristică de half duplex.

De asemenea se consideră o subrețea a unei rețele de senzori, subrețea ce este formată din două noduri, identice, ce pot comunica direct, fiecare fiind în raza de acțiune a celuilalt. Pentru simplitate se consideră că cele două noduri sunt formate dintr-un microcontroller gazdă și un modul radio. Microcontrollerul gazdă se ocupă atât de implementarea driver-ului pentru a gestiona modulul radio cât și de a implementa nivelul de acces la mediu. Din acest considerent, modelul de descriere a timpului trebuie să considere nu doar timpul efectiv de transport al unui mesaj de la un nod A la un nod B ci și întârzierile cauzate de procesarea datelor la nivelul microcontrollerului gazdă, acesta fiind influențat la rândul lui de procesarea la nivelul driverului și procesarea la nivelul protocolului de acces la mediu. Tot pentru a simplifica descrierea modelului considerăm că nodul A este nod emițător și nodul B este nod receptor.

Așadar construcția următoare descrie componentele de timp ce intervin în întârzierea totală a transportului unui pachet de date de la un nod A la un nod B.

$$T(A, B) = T_{transfer_A} + T_{TXON_A} + T_{TX\_SFD_A} + T_{RX\_SFD_B} + T_{RX\_PACK_B} + T_{transfer_B} \quad (6-1)$$

Termenii ce compun timpul total de transmisie a unui pachet de date de la un nod A la un nod B se pot defini astfel:

- $T_{transfer\_A}$  – intervalul de timp necesar ca procesorul gazdă al nodului A să transfere pachetul de date către modulul radio prin interfața de comunicare cu acesta
- $T_{TXON\_A}$  – intervalul de timp necesar modulului radio al nodului A pentru a-și dezactiva componenta de recepție și a-și activa componenta de transmisie
- $T_{TX\_SFD\_A}$  – intervalul de timp necesar modulului radio la nodului A pentru a transmite secvența SFD. Acest interval de timp poate include și procedura CCA (Clear Channel Assessment)
- $T_{RX\_SFD\_B}$  – intervalul de timp necesar pentru ca secvența SFD transmisă de nodul A să fie recepționată de nodul B
- $T_{RX\_PACK\_B}$  – intervalul de timp necesar nodului B să recepționeze întregul pachet trimis de nodul A
- $T_{transfer\_B}$  – intervalul de timp necesar procesorului gazdă al nodului B să transfere pachetul de date nou recepționat de la modulul radio atașat către memoria sa internă

Construcția (6-1) se referă la întârzierea totală datorată transferului unui pachet prin mediul wireless de la un nod transmițător A la un nod receptor B. Ținând cont de faptul că într-o rețea de senzori cu comunicare wireless în timp real, atât comunicarea cât și fiecare nod trebuie să funcționeze în manieră de timp real este necesar să se țină cont și de componentele de timp ce apar la nivelul fiecărui nod, în acest caz la nivelul nodului transmițător A și la nivelul nodului receptor B. Aceste componente de timp sunt absolut necesare astfel încât atât driverul ce rulează pe microcontrolerul gazdă pentru a gestiona modulul radio cât și componenta software ce implementează protocolul de acces la mediu să funcționeze într-un sistem de operare în timp real.

Așadar întârzierea totală ce influențează componenta software a nodului A, nodul transmițător, se pot descrie astfel:

$$T(A) = T_{transfer} + T_{TXON} + T_{TX\_SFD} + T_{TX\_PACK} + T_{RX\_ON} \quad (6-2)$$

Componentele de timp din construcția (6-2) au următoarea semnificație:

- $T_{transfer}$  – intervalul de timp necesar ca procesorul gazdă să transfere pachetul de date către modulul radio prin interfața de comunicare cu acesta
- $T_{TX\_ON}$  – intervalul de timp necesar modulului radio pentru a-și dezactiva componenta de recepție și a-și activa componenta de transmisie
- $T_{TX\_SFD}$  – intervalul de timp necesar modulului radio pentru a transmite secvența SFD. Acest interval de timp poate include și procedura CCA
- $T_{TX\_PACK}$  – intervalul de timp necesar modulului radio pentru a transmite întregul pachet
- $T_{RX\_ON}$  – intervalul de timp necesar modulului radio pentru a-și dezactiva componenta de transmisie și a-și reactiva componenta de recepție



În aceeași manieră este necesar să se studieze și întârzierea totală ce influențează componenta software a nodului B, nodul receptor:

$$T(B) = T_{RX\_PACK} + T_{transfer} \quad (6-3)$$

Componentele de timp din construcția (6-3) se definesc astfel:

- $T_{RX\_PACK}$  – intervalul de timp necesar modulului receptor să primească întregul pachet din momentul în care acesta primește cu succes secvența SFD
  - $T_{transfer}$  – intervalul de timp necesar procesorului gazdă să transfere pachetul de date nou recepționat de la modulul radio atașat către memoria sa internă
- O analiză temporală a acestor componente de timp este următoarea:
- componentele de timp  $T_{transfer\_A}$ ,  $T_{transfer\_B}$  și  $T_{transfer}$  sunt componente predictibile și calculabile în cazul majorității modulelor radio. Acest lucru este posibil deoarece modulul radio comunică cu procesorul gazdă printr-o magistrală de obicei standardizată care permite calcularea exactă a timpilor de comunicație (ex: magistrala SPI sau I2C). Aceste componente de timp depind de dimensiunea pachetelor și de viteza magistralei de comunicație dintre procesorul gazdă și modulul radio
  - componentele de timp  $T_{TX\_ON\_A}$ ,  $T_{TX\_ON\_B}$ ,  $T_{TX\_ON}$  și  $T_{RX\_ON}$  sunt parametri de timp de depind de caracteristicile fiecărui modul radio. Acestea sunt strict dependente de modulul radio. De obicei această valoare este oferită de producător în pagina de catalog a modulului radio
  - componentele de timp  $T_{TX\_SFD\_A}$ ,  $T_{TX\_SFD\_B}$  reprezintă durata de timp necesară modulului pentru a trimite secvența SFD și pentru a realiza operațiunea CCA. De asemenea acest timp include întreaga secvență SHR. Acest parametru este puternic dependent de mediul wireless prezentând un caracter impredictibil
  - componentele de timp de tip  $T_{RX\_SFD\_B}$  sunt componente ce depind atât de caracteristicile mediului wireless la momentul transmisiei cât și de bit rate-ul și modulația modulului radio

Pe lângă aceste componente de timp se mai poate defini cu aproximație durata de transmisie a unui octet pe mediul wireless. Acesta se poate defini ca fiind momentul de timp de când modulul A a terminat de transmis pachetul până când modulul B a terminat de recepționat acest pachet. Acest timp se poate nota cu  $T_{BYTE}$ .

Modelul temporal descris anterior poate fi analizat mai mult prin descompunerea și descrierea amănunțită a termenilor din ecuațiile anterioare. Astfel se pot identifica mai exact componentele de timp ce induc impredictibilitate sistemului.

Astfel, prima componentă de timp ce apare în modelul temporal descrie timpul necesar pentru a se transfera pachetul de date de la procesorul gazdă al nodului către modulul radio al acestuia ( $T_{transfer\_A}$ ,  $T_{transfer\_B}$ ,  $T_{transfer}$ ). Acest tip de parametru poate fi descompus astfel:

$$T_{transfer} = 8 \cdot P_S \cdot T_{transfer\_BIT} \quad (6-4)$$

În ecuația (6-4) se definește parametrul  $P_S$  ca fiind dimensiunea pachetului exprimată în octeți și parametrul  $T_{transfer\_BIT}$  ce reprezintă timpul necesar pentru a transfera 1 bit de informație. Ținând cont de faptul că ultimul termen este cunoscut și

constant în marea majoritate a cazurilor, se poate deduce că întreaga componentă  $T_{transfer}$  este calculabilă folosind ecuația (6-4) devenind astfel predictibilă. Se poate trage această concluzie din cauza faptului că în marea majoritate a cazurilor calea de comunicație dintre procesorul gazdă și modulul radio este reprezentată de o magistrală pe care se transferă date folosind protocoale de linie cum ar fi SPI (Serial Peripheral Interface), USART (Universal Synchronous/Asynchronous Receiver/Transmitter) sau I2C (Inter-Integrated Circuit) care sunt perfect calculabile și predictibile.

Următoarea componentă de timp ce va fi analizată definește intervalul de timp necesar pentru ca modulul radio să își schimbe starea din receptor în transmițător ( $T_{TXON\_A}$ ,  $T_{TXON}$ ). În marea majoritate a cazurilor se constată practic că această schimbare de stare nu durează mult dar în cazul cel mai defavorabil, această componentă de timp, conform paginilor de catalog a celor mai multe module radio [93, 94], poate să fie egală cu timpul de recalibrare a sub modulului PLL (Phase-Locked Loop) intern al modulului radio. Acest timp este de cele mai multe ori precizat de către producător în pagina de catalog a modulului radio. În cazul în care se dorește funcționarea acestui modul radio în maniera timp real, acest scenariu defavorabil trebuie luat în calcul rezultând astfel următoarea egalitate.

$$T_{TXON} = T_{PLL} \quad (6-5)$$

O componentă de timp similară este și  $T_{RXON}$  care definește timpul necesar modulului radio pentru a-și schimba starea internă din transmițător în receptor. În cea mai mare parte din timpul de funcționare a unui modul radio, acesta se găsește în starea de receptor. În momentul în care acestuia i se cere să efectueze o transmisie, modulul trebuie să treacă în starea internă de transmisie. Imediat după ce transmisia s-a încheiat acesta trebuie să își schimbe starea înapoi în recepție. În cel mai defavorabil caz și schimbarea din starea de transmisie în starea de recepție poate determina o recalibrare a modulului PLL intern. Ținând cont de aceste observații se pot defini următoarele egalități:

$$T_{RXON} = T_{PLL} \quad (6-6)$$

$$T_{RXON} = T_{TXON} \quad (6-7)$$

Una dintre componentele de timp ce introduce un grad mare de impredictibilitate sistemului este componenta ce descrie timpul necesar modulului radio pentru a transmite secvența SFD (Start of Frame) identificată prin  $T_{TX\_SFD}$ . Această componentă include operațiunea CCA și transmisia efectivă a acestei secvențe. Impredictibilitatea este dată de durata operațiunii CCA ce depinde în cea mai mare parte de starea de ocupare a canalului de comunicație precum și de zgomotul prezent pe canal. Astfel componenta  $T_{TX\_SFD}$  este descrisă în următoarea construcție:

$$T_{TX\_SFD} = T_{CCA} + SFD\_SEQ\_SIZE \cdot \frac{1}{DTR} \quad (6-8)$$

În ecuația (6-8) a fost introdus termenul  $T_{CCA}$  ce reprezintă durata de timp a operațiunii CCA,  $SFD\_SEQ\_SIZE$  ce reprezintă dimensiunea secvenței SFD exprimată

în octeți precum și DTR (Data Transfer Rate) ce reprezintă viteza de transfer. Acest ultim termen reprezintă o caracteristică de bază a modulelor radio prezentată de producător în pagina de catalog.

Următoarea componentă în discuție ( $T_{RX\_SFD}$ ) descrie intervalul de timp necesar modulului radio pentru a recepționa secvența SFD. Acest interval de timp este reprezentat de latența internă a modulului radio, parametru ce în majoritatea cazurilor este prezent în pagina de catalog modulului:

$$T_{RX\_SFD} = L_{RX} \quad (6-9)$$

O altă componentă de timp predictibilă este reprezentată de termenul  $T_{RX\_PACK}$ . Predictibilitatea acestei componente este asigurată de faptul că depinde doar de dimensiunea pachetului de date și de viteza de transfer:

$$T_{RX\_PACK} = P_S \cdot \frac{1}{DTR} \quad (6-10)$$

Folosind parametrii de timp detaliați în (6-4), (6-5), (6-6), (6-7), (6-9) și înlocuind în ecuația (6-1) se obține un model de timp mult mai detaliat care descrie întârzierea totală ce apare la transferul unui pachet de date de la nodul transmițător A la nodul receptor B:

$$T(A, B) = P_S \cdot \left( 16 \cdot T_{transfer\_BIT} + \frac{1}{DTR} \right) + T_{PLL} + T_{CCA} + SFD\_SEQ\_SIZE \cdot \frac{1}{DTR} + L_{RX} \quad (6-11)$$

În modelul rezultat din (6-11) majoritatea componentelor sunt predictibile mai puțin termenii  $T_{CCA}$  și  $T_{PLL}$  care trebuie analizați separat de către dezvoltatorul de aplicație în timp real.

Asemănător, aplicând ecuațiile (6-4), (6-6), (6-7), (6-8) și înlocuind în modelul descris în (6-2) se obține un model extins pentru a exprima întârzierea totală introdusă de modulul radio la nivelul operării interne a nodului transmițător în manieră timp real:

$$T(A) = P_S \cdot \left( 8 \cdot T_{transfer\_BIT} + \frac{1}{DTR} \right) + 2 \cdot T_{PLL} + T_{CCA} + SFD\_SEQ\_SIZE \cdot \frac{1}{DTR} \quad (6-12)$$

Aceeași metodă se poate aplica și pentru a obține un model extins pentru intervalul de timp ce exprimă întârzierea totală introdusă de modulul radio la nivelul operării interne a nodului receptor în manieră timp real. Pentru obținerea acestuia se folosește ecuația (6-3) în care se înlocuiesc termenii din (6-4) și (6-10) rezultând:

$$T(B) = P_s \cdot \left( 8 \cdot T_{transfer\_BIT} + \frac{1}{DTR} \right) \quad (6-13)$$

## 6.2 Cadru de măsurare a parametrilor de timp

Parametrii de timp ce au fost descriși în capitolul anterior pot fi măsurați pentru a se demonstra experimental modelul. Problema ce apare în acest caz este găsirea unui număr suficient de canale de măsurare a timpilor și memorarea evenimentelor apărute. Această memorare a evenimentelor de timp apărute trebuie să nu blocheze sistemul de captură a evenimentelor de timp.

Una dintre soluțiile tehnice pentru a realiza acest cadru de măsurare a evenimentelor de timp este folosirea unui microcontroller cu un număr mare de canale de captură de evenimente externe de timp. Un astfel de procesor poate fi și un microcontroller NXP LPC2294 [25]. Acest procesor dispune în periferia sa de 2 module timer, fiecare având 4 canale de captură de timp fiecare și în plus oferă 4 canale de întreruperi externe putându-se obține astfel un număr total de 12 canale de captură de timp. Un canal de captură de timp ce aparține unui modul timer poate capta un eveniment atât pe front crescător cât și pe front descrescător. Nu același lucru se poate spune și despre canalele de întrerupere externă acestea putând fi configurate să capteze evenimente doar pe unul din fronturi, crescător sau descrescător. În concluzie, folosind acest microcontroller se pot captura un număr de 20 de evenimente de timp diferite după cum se poate observa pe schema bloc din figura următoare.

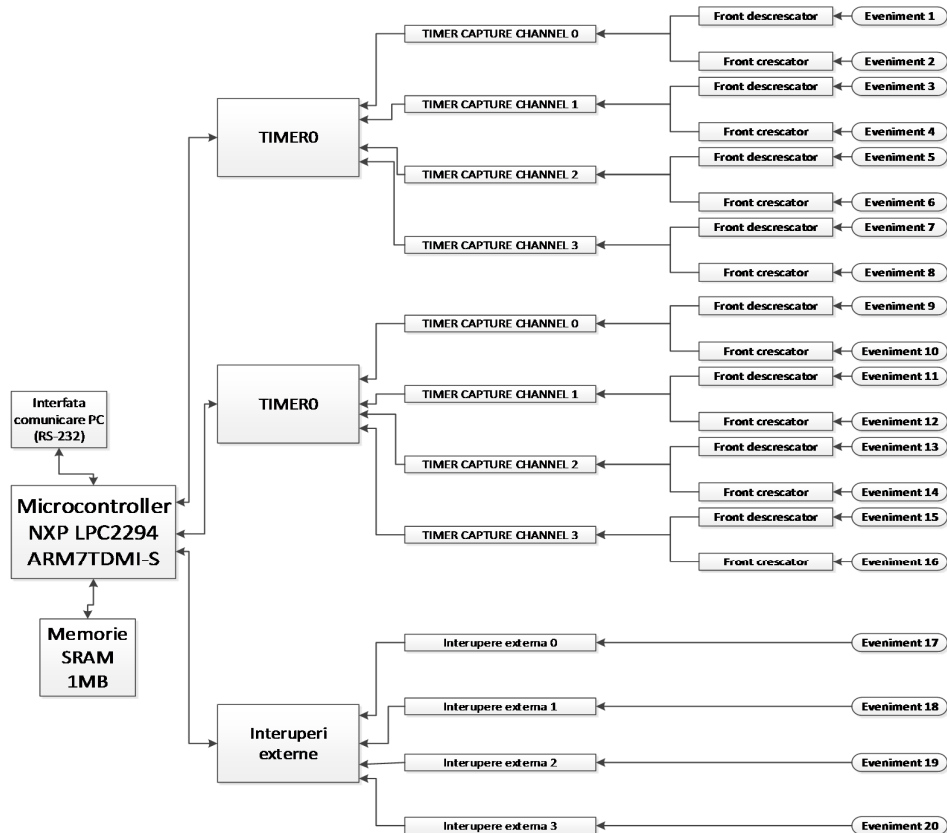


Figura 6-1 Schema bloc a sistemului de măsurare a evenimentelor de timp

Odată capturat un eveniment de timp, acesta trebuie stocat într-o memorie iar pentru a captura toate evenimentele și a măsura cât mai multe tipuri de comunicație este necesar să se folosească o memorie cât mai mare și cât mai rapidă. Pentru aceasta s-a ales placa de dezvoltare produsă de Olimex și LPC-H2294 [118]. Această placă este echipată cu un procesor NXP LPC 2294 și cu două tipuri de memorii paralele: o memorie de tip FLASH de 4 MB și o memorie SRAM de o capacitate 1 MB. Din considerente de viteză s-a ales a se folosi memoria SRAM la care timpii de acces sunt mici, de aproximativ 12 ns [119].

Procesorul ales, NXP LPC2294, are la bază o arhitectură ARM7TDMI-S care are ca și caracteristici principale existența mai multor moduri de funcționare și o bandă de asamblare cu 3 stagii. Fiecare mod de funcționare are propria zonă de stivă definită prin 2 registre. Practic schimbarea între modurile de funcționare nu presupune nici o operație suplimentară asupra memoriei de program. Cel mai important mod de funcționare este modul caracteristic întreruperii rapide FIQ (Fast Interrupt). Acest mod are un set de registre dedicate astfel că schimbarea în și din acest mod de funcționare se poate face foarte repede. În cazul în care se programatorul folosește de aceste facilități corect, o trecere în și din modul FIQ nu ar trebui să implice salvări și restaurări din stivă [25].

Principalele motive pentru care s-a ales acest procesor au fost: existența unui număr mare de canale de captură de evenimente de timp, existența unei magistrale externe paralele pentru conectarea memoriilor externe, existența unei plăci de dezvoltare adecvate pentru acest tip de aplicație și cel mai important lucru: un sistem de întreruperi ce permite multe surse de întreruperi vectorizate, nevectorizat sau foarte rapide, în toate cazurile latențele fiind foarte mici și predictibile. În cazul unei întreruperi vectorizate de tip IRQ latența este de maxim 25 cicluri de tact iar latența unei întreruperi rapide de tip FIQ este de 12 cicluri de tact. Este important de menționat că întreruperile de FIQ nu pot fi vectorizate astfel că nu poate exista decât o singură rutină de tratare a întreruperii pentru toate sursele de întrerupere de tip FIQ [26].

În situația de față procesorul își formează tactul intern de la un cristal de cuarț de 14.7456 MHz și rulează la o frecvență de 58,9824 MHz. În aceste condiții durata maximă a latenței de tratare a întreruperii de tip IRQ este de aproximativ 424 ns iar latența de deservire a unei întreruperi de tip FIQ este de aproximativ 230 ns. Comparate cu vitezele de lucru ale modulelor radio pentru care a fost realizat acest cadru de măsurare, latențele prezentate anterior nu pot influența valorile măsurate. După cum se observă în schema bloc sunt folosite ambele timere ale procesorului, ambele având frecvența de numărare de 14.7456 MHz și fiind sincronizate cu o eroare de o perioadă de tact, diferența dintre ele fiind de maxim 1.

Procesorul sistemului de măsură are ca rol doar captura și stocarea în memorie a fiecărui eveniment de tip cu ștampila de timp a acestuia, reprezentată aici de ceasul sistem. Pentru situația în care se dorește realizarea unui set de măsurători lung în timp valoarea timer-ului de 32 de biți a fost extinsă printr-o valoare tot pe 32 de biți cu ajutorul unei locații de memorie. La fiecare depășire a valorii maxime a timer-ului se incrementează această parte mai semnificativă a timpului sistem. Așadar ceasul sistem este reprezentat pe o valoare totală de 64 de biți. Fiecare înregistrare în memorie are următoarea structură:

Tabel 6-1. Structura unei înregistrări a unui eveniment în memorie

Denumire câmp	Dimensiune câmp [octeți]
Identificator unic eveniment	1
Ștampila de timp, parte semnificativă	4
Ștampila de timp, partea mai puțin semnificativă	4
<b>TOTAL</b>	<b>9</b>

După cum se observă în Tabel 6-1 dimensiunea unei înregistrări în memoria SRAM a unei capturi de timp a unui eveniment este de 9 octeți permițând astfel o sesiune maximă de măsurători de 116.508 evenimente.

Sistemul embedded prezentat în această secțiune are rolul de a captura și stoca evenimentele de timp cu ștampila de timp asociată. În Figura 4-1 se poate observa existența blocului de comunicare cu un PC gazdă. Prin intermediul unei interfețe seriale de mare viteză (460.800 bps), sistemul poate transmite, la cerere, în format CSV (Comma-Separated Values) toate măsurătorile. Programul de pe PC-ul gazdă are rolul de face calculele necesare și de a centraliza datele. În urma descărcării datelor de pe sistem, acestea sunt ulterior șterse din memoria internă a sistemului embedded pentru a face loc unor noi măsurători.

### 6.3 Analiza experimentală a parametrilor de timp

Validarea experimentală a modelului de timp prezentat în secțiunea 6.1 poate fi realizată folosind cadrul de măsurare descris în secțiunea 0. Pentru această validare s-au folosit două platforme. Prima platformă este aceeași cu cea folosită la experimentele din secțiunea 5. A doua platformă este compusă dintr-o placă Olimex LPC-H2294 la care s-a atașat modulul radio CC2500 integrat pe modulul de dezvoltare Texas Instruments eZ430-RF2500 [120]. Pentru fiecare caz s-au folosit câte două platforme de același tip pentru a se putea comunica între ele (2 sisteme cu CC2420 și 2 sisteme cu CC2500).

Atât pentru modulul radio CC2420 cât și pentru CC2500 s-a folosit calcularea hardware a șumei de control CRC16, mecanismele de recunoaștere a adresei fiind dezactivate. În cadrul ambelor platforme magistrala de comunicare dintre procesorul gazdă și modulul radio este reprezentată de magistrala SPI. În toate cazurile, modulele au fost configurate să folosească canalul 11 din banda de 2,4 GHz, mai exact frecvența 2405 MHz.

Toate sistemele din ambele platforme au fost programate să anunțe o serie de evenimente descrise în Tabel 6-2. Evenimentele descrise în acest tabel se referă strict la fiecare sistem în parte. Spre exemplu evenimentul 5 este lansat când modulul radio al sistemului a terminat de transmis pachetul. Acest lucru se referă doar la faptul că modulul radio al sistemului a terminat de modulat acel pachet în mediul wireless nu și faptul că pachetul a ajuns la destinație.

Așadar fiecare platformă, având câte 2 sisteme identice, vor fi denumite în continuare nodul A și nodul B. Pentru a măsura componentele de timp prezentate în secțiunile anterioare se va face convenția că nodul A este nodul emițător și nodul B este nodul receptor.

De asemenea, în unele situații s-a încercat și bruierea canalului de comunicație. Pentru aceasta s-a folosit un modul de comunicație XBee configurat pe canalul 12, pe 2410 MHz.

Tabel 6-2. Descrierea evenimentelor

Eveniment	Descriere eveniment	Canalul de captură	Front	Tip op
1	Modulul radio a primit secvența SFD	Întrerupere externă	Crescător	Recepție
2	Modulul radio a primit tot pachetul. Procesorul gazdă a început descărcarea acestuia prin SPI	TIMER Capture Channel 0	Crescător	
3	Procesorul gazdă a finalizat descărcarea pachetului prin SPI	TIMER Capture Channel 0	Descrescător	
4	Procesorul gazdă a început transferului pachetului din memorie internă către modulul radio prin SPI	TIMER Capture Channel 1	Crescător	Transmisie
5	Procesorul gazdă a finalizat transferul pachetului către modulul radio prin SPI. Procesorul gazdă comandă modulul radio să își oprească modulul de recepție și să-și pornească modulul de transmisie	TIMER Capture Channel 1	Descrescător	
6	Modulul radio a inițiat procedura de transmisie și începe transmisia secvenței SFD	TIMER Capture Channel 2	Crescător	
7	Modulul a terminat transmisia secvenței SFD	TIMER Capture Channel 2	Descrescător	
8	Modulul a transmis întregul pachet (a terminat de modulat întregul pachet). Procesul gazdă comandă modulul radio să-și oprească modulul de transmisie și să-și pornească modulul de recepție	TIMER Capture Channel 3	Crescător	
9	Modulul radio a terminat schimbarea în modul de recepție	TIMER Capture Channel 3	Descrescător	

Aceste evenimente sunt folosite pentru a calcula componentele de timp descrise în secțiunea 6.1 . În următorul tabel se va arăta cum folosind evenimentele descrise mai sus se pot obține componentele de timp din secțiunea 6.1 .



Tabel 6-3. Calculul componentelor de timp din evenimentele măsurate

Nr. Crt.	Componenta de timp	Descriere componente de timp	Formulă de calcul
1	$T_{transfer_A}$	intervalul de timp necesar ca procesorul gazdă al nodului A să transfere pachetul de date către modulul radio prin interfața de comunicare cu acesta	$EV_{5_A} - EV_{4_A}$
2	$T_{TXON_A}$	intervalul de timp necesar modulului radio al nodului A pentru a trece în modul transmisie	$EV_{6_A} - EV_{5_A}$
3	$T_{TX\_SFD_A}$	intervalul de timp necesar modulului radio al nodului A pentru a transmite secvența SFD	$EV_{7_A} - EV_{6_A}$
4	$T_{TX\_PACK_A}$	intervalul de timp necesar modulului radio al nodului A pentru a transmite întregul pachet (pentru a modula întregul pachet)	$EV_{8_A} - EV_{7_A}$
5	$T_{RX\_ON_A}$	intervalul de timp necesar modulului radio pentru a intra în mod recepție	$EV_{9_A} - EV_{8_A}$
6	$T_{RX\_SFD_B}$	intervalul de timp necesar pentru ca secvența SFD transmisă de nodul A să fie recepționată de nodul B	$EV_{1_B} - EV_{7_A}$
7	$T_{RX\_PACK_B}$	intervalul de timp necesar nodului B să recepționeze întregul pachet trimis de nodul A	$EV_{2_B} - EV_{1_B}$
8	$T_{transfer_B}$	intervalul de timp necesar procesorului gazdă al nodului B să transfere pachetul de date nou recepționat de la modulului radio atașat către memoria sa internă	$EV_{3_B} - EV_{2_B}$
9	$T_{BYTE}$	aproximare a timpului de transmisie a unui octet	$EV_{2_B} - EV_{8_A}$
10	T(A,B) - măsurat	intervalul de timp (măsurat) total de transmisie a unui pachet de la nodul A la nodul B	$EV_{3_B} - EV_{4_A}$

## 74 Model temporal și cadru de măsurare pentru comunicație wireless – 6

Componentele de timp prezentate în secțiunea 6.1 se pot măsura folosind cadrul de măsurare din secțiunea 6.2 prin utilizarea evenimentelor descrise în Tabel 6-2. Cu ajutorul ștampilelor de timp ale acestor evenimente se vor obține componentele de timp descrise în Tabel 6-3. Spre exemplu, conform Tabel 6-3, componenta de timp  $T_{\text{transfer\_A}}$  se obține prin scăderea din ștampila de timp a evenimentului 5 generat de nodul transmițător A a ștampilei de timp a evenimentului 4 generat de nodul receptor B. Evenimentul 5 generat de nodul A și evenimentul 4 generat de nodul B sunt descrise în Tabel 6-2, atât semnificația lor cât și canalul și frontul folosit pentru captura ștampilei de timp.

Experimentele au constatat în măsurarea timpilor descriși mai sus pentru fiecare din cele două platforme. S-a procedat prin a efectua 100 de măsurători pentru fiecare dimensiune de pachet. Astfel pentru platforma cu sisteme cu modulele radio CC2420 s-a considerat o dimensiune maximă de 64 octeți de date utile iar în cazul platformelor cu CC2500 s-a considerat o dimensiune maximă de 50 octeți de date utile. Prin Tabel 6-4 se poate totaliza numărul de măsurători efectuate. Pentru fiecare set de măsurători datele au fost centralizate și s-au calculat valorile minime, maxime și medii pentru fiecare componentă de timp în parte.

Tabel 6-4. Total măsurători efectuate

Platforma	Dim. maxima a pachetului [octeți]	Număr seturi de măsurători	Număr măsurători/ set/ dimensiune pachet	Număr componente de timp	TOTAL
CC2420	64	6	100	10	384.000
CC2500	50	4	100	10	200.000
<b>TOTAL MĂSURĂTORI</b>					<b>584.000</b>

### 6.3 - Analiza experimentală a parametrilor de timp 75

Tabel 6-5. Exemplificare set măsurători cu valori medii

Dim. pachet	T <sub>transfer_A</sub>	T <sub>rxon_A</sub>	T <sub>txsfd_A</sub>	T <sub>tx_pack_A</sub>	T <sub>rxon_A</sub>	T <sub>rx_sfd_B</sub>	T <sub>rx_pack_B</sub>	T <sub>byte</sub>	T <sub>transfer_B</sub>	T(A,B)
	[μs]	[μs]	[μs]	[μs]	[μs]	[μs]	[μs]	[μs]	[μs]	[μs]
1	96,8221	13,696	350,535	474,3456	6,508	2,171	480,9075	8,733	104,688	1048,820
2	103,0599	13,695	350,536	506,4819	6,507	2,151	512,9347	8,603	111,060	1093,438
3	109,2977	13,696	350,535	538,2304	6,508	2,287	544,7876	8,844	117,434	1138,038
4	115,5373	13,696	350,535	570,3526	6,507	2,266	576,8281	8,742	123,808	1182,671
5	121,7740	13,694	350,536	602,4909	6,507	2,187	608,8847	8,581	130,183	1227,261
6	128,0219	13,697	350,529	634,2231	6,510	2,204	640,8834	8,864	136,555	1271,891
7	134,2482	13,697	350,534	666,3608	6,507	2,244	672,8319	8,715	142,928	1316,483
8	140,4860	13,697	350,535	698,4972	6,509	2,239	704,8211	8,563	149,300	1361,078
9	146,7244	13,697	350,533	730,2307	6,507	2,218	736,8605	8,848	155,674	1405,708
10	152,9622	13,697	350,533	762,3684	6,507	2,205	768,8761	8,713	162,048	1450,322
11	159,1987	13,697	350,536	794,4750	6,507	2,244	800,8477	8,616	168,421	1494,945
12	165,4371	13,696	350,535	826,2377	6,507	2,163	832,8864	8,812	174,796	1539,515
13	171,6749	13,695	350,536	858,3761	6,507	2,244	864,8349	8,703	181,167	1584,154
14	177,9127	13,697	350,534	890,4778	6,507	2,214	896,8621	8,598	187,542	1628,763
15	184,1526	13,695	350,534	922,2466	6,505	2,242	928,8262	8,821	193,916	1673,367
16	190,3890	13,697	350,532	954,3844	6,507	2,204	960,8738	8,694	200,287	1717,984
17	196,6254	13,697	350,535	986,3634	6,507	2,168	992,9057	8,710	206,661	1762,592
18	202,8632	13,697	350,535	1018,252	6,507	2,286	1024,768	8,801	213,036	1807,186
19	209,1017	13,696	350,535	1050,390	6,507	2,250	1056,819	8,678	219,409	1851,812
20	215,3408	13,697	350,534	1082,310	6,507	2,155	1088,875	8,720	225,780	1896,384
21	221,5800	13,694	350,535	1114,262	6,505	2,174	1120,890	8,802	232,154	1941,029
22	227,8171	13,697	350,532	1146,398	6,508	2,313	1152,754	8,669	238,528	1985,643
23	234,0522	13,696	350,536	1178,430	6,507	2,269	1184,814	8,653	244,900	2030,270
24	240,2886	13,699	350,531	1210,272	6,509	2,182	1216,897	8,807	251,275	2074,874
25	246,5305	13,695	350,534	1242,406	6,507	2,179	1248,872	8,645	257,647	2119,460
26	252,7656	13,698	350,533	1274,320	6,510	2,196	1280,847	8,724	264,023	2164,065
27	259,0013	13,698	350,536	1306,274	6,509	2,190	1312,866	8,782	270,393	2208,686
28	265,2418	13,696	350,535	1338,414	6,507	2,101	1344,955	8,641	276,768	2253,298
29	271,4708	13,699	350,545	1370,244	6,510	2,150	1376,879	8,786	283,140	2297,886
30	277,7181	13,697	350,533	1402,284	6,507	2,151	1408,905	8,772	289,514	2342,521
31	283,9552	13,696	350,535	1434,422	6,507	2,216	1440,840	8,633	295,888	2387,131
32	290,1930	13,697	350,536	1466,460	6,507	2,217	1472,871	8,628	302,262	2431,777
33	296,4301	13,697	350,534	1498,291	6,507	2,143	1504,919	8,771	308,634	2476,359
34	302,6666	13,699	350,531	1530,436	6,505	2,154	1536,887	8,606	315,008	2520,947
35	308,9064	13,697	350,533	1562,209	6,507	2,224	1568,797	8,812	321,382	2565,540
36	315,1435	13,697	350,535	1594,299	6,507	2,287	1600,751	8,739	327,755	2610,170
37	321,3820	13,695	350,535	1626,437	6,507	2,165	1632,895	8,623	334,129	2654,804
38	327,6198	13,696	350,535	1658,241	6,507	2,215	1664,828	8,801	340,502	2699,397
39	333,8589	13,695	350,536	1690,307	6,507	2,183	1696,864	8,740	346,875	2744,014
40	340,0960	13,695	350,533	1722,446	6,507	2,268	1728,771	8,593	353,250	2788,614
41	346,3325	13,697	350,535	1754,180	6,507	2,219	1760,826	8,865	359,621	2833,232
42	352,5716	13,695	350,534	1786,316	6,508	2,216	1792,841	8,740	365,994	2877,853
43	358,8087	13,697	350,534	1818,453	6,506	2,221	1824,811	8,579	372,368	2922,441
44	365,0472	13,694	350,535	1850,183	6,509	2,284	1856,728	8,828	378,741	2967,030
45	371,2836	13,696	350,535	1882,322	6,509	2,214	1888,828	8,720	385,116	3011,674
46	377,5221	13,695	350,534	1914,459	6,509	2,244	1920,786	8,572	391,487	3056,271
47	383,7613	13,696	350,534	1946,192	6,507	2,257	1952,775	8,841	397,863	3100,889
48	389,9984	13,696	350,535	1978,331	6,505	2,187	1984,844	8,700	404,235	3145,497
49	396,2368	13,695	350,536	2010,309	6,507	2,180	2016,815	8,686	410,609	3190,073
50	402,4740	13,697	350,535	2042,199	6,506	2,179	2048,860	8,840	416,982	3234,727
51	408,7118	13,697	350,533	2074,338	6,507	2,170	2080,852	8,683	423,356	3279,321
52	414,9489	13,696	350,535	2106,386	6,507	2,195	2112,853	8,662	429,728	3323,957
53	421,1846	13,696	350,535	2138,206	6,507	2,223	2144,826	8,843	436,100	3368,566
54	427,4278	13,696	350,533	2170,346	6,507	2,179	2176,836	8,669	442,475	3413,149
55	433,6616	13,697	350,535	2202,381	6,507	2,181	2208,848	8,648	448,847	3457,771
56	439,8994	13,695	350,535	2234,214	6,509	2,282	2240,745	8,813	455,219	3502,377
57	446,1385	13,696	350,534	2266,353	6,507	2,204	2272,825	8,676	461,596	3546,995
58	452,3756	13,695	350,535	2298,280	6,507	2,108	2304,901	8,730	467,969	3591,586
59	458,6141	13,694	350,534	2330,223	6,508	2,224	2336,815	8,816	474,343	3636,226
60	464,8519	13,697	350,533	2362,361	6,507	2,137	2368,882	8,658	480,714	3680,817

În Tabel 6-5 se exemplifică un set de astfel de măsurători în care sunt obținute valorile medii ale componentelor de timp. Valorile sunt exprimate în microsecunde.

În continuare se va face o analiză a datelor experimentale pentru fiecare componentă de timp în parte.

Primele componente de timp ce vor fi analizate din punct de vedere al rezultatelor obținute experimental sunt reprezentate de timpii de transfer pe magistrala SPI între procesorul gazdă și modulul radio. Timpii în discuție sunt următorii:  $T_{transfer\_A}$ ,  $T_{transfer\_B}$ . Magistrala SPI fiind o magistrală sincronă, timpul necesar transferului unui pachet de date de la procesorul gazdă către modulul radio și invers poate fi calculat și astfel este predictibil. În Figura 6-2 se prezintă timpul de transfer al unui pachet de la procesorul gazdă către modulul radio (valori minime, medii și maxime) în cazul folosirii platformei CC2420, mai exact este prezentată componenta  $T_{transfer\_A}$  pentru această platformă.

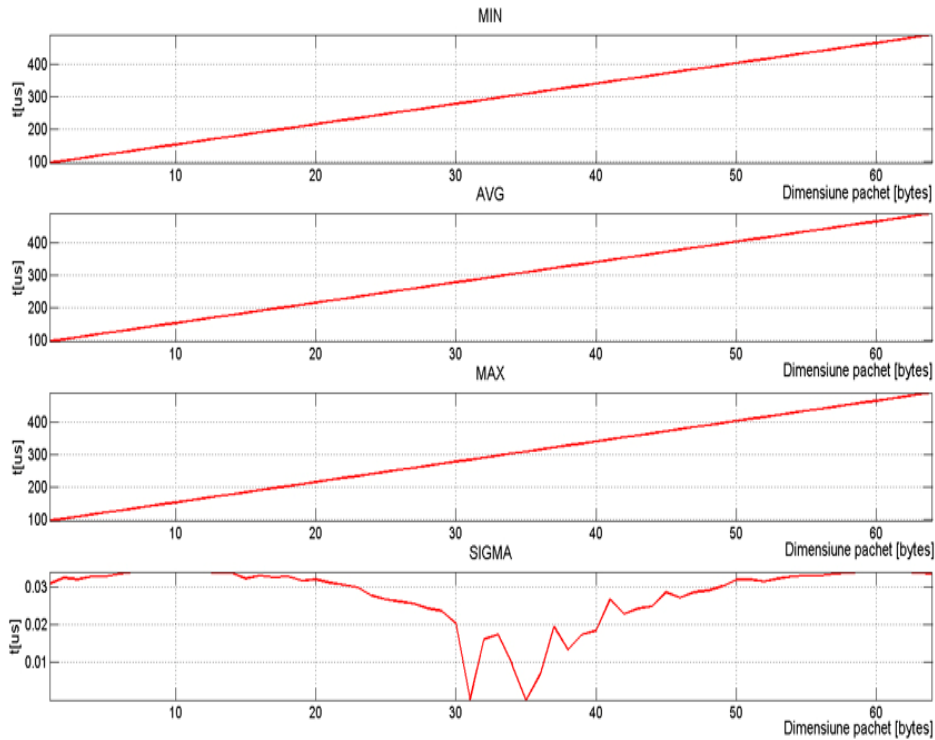


Figura 6-2 Valori minime, medii și maxime pentru componenta  $T_{transfer\_A}$

Celelalte componente de timp asemănătoare componentei prezentate în Figura 6-2, respectiv componenta  $T_{transfer\_B}$  pentru platforma CC2420 cât și componentele  $T_{transfer\_A}$  și  $T_{transfer\_B}$  pentru platforma CC2500, sunt identice ca și evoluție în timp, ceea ce este evident din moment ce acestea sunt calculabile și predictibile.

Următorul timp analizat este intervalul de timp necesar modulului radio pentru a trece din modul recepție în modul transmisie. Din această categorie fac parte componentele de tip  $T_{TX\_ON}$ . La marea majoritate a modulelor radio existente această componentă este de obicei furnizată de producător, ca valoarea maximă, în pagina de catalog a modulului. În cazul platformei cu modulul CC2420, această componentă de

### 6.3 - Analiza experimentală a parametrilor de timp 77

timp poate ajunge la 192  $\mu\text{s}$  și la 730,6  $\mu\text{s}$  în cazul platformei CC2500. În ambele situații s-a luat în calcul cazul cel mai defavorabil. Din punct de vedere experimental, evoluția acestui timp este descrisă în Figura 6-3 și Figura 6-4.

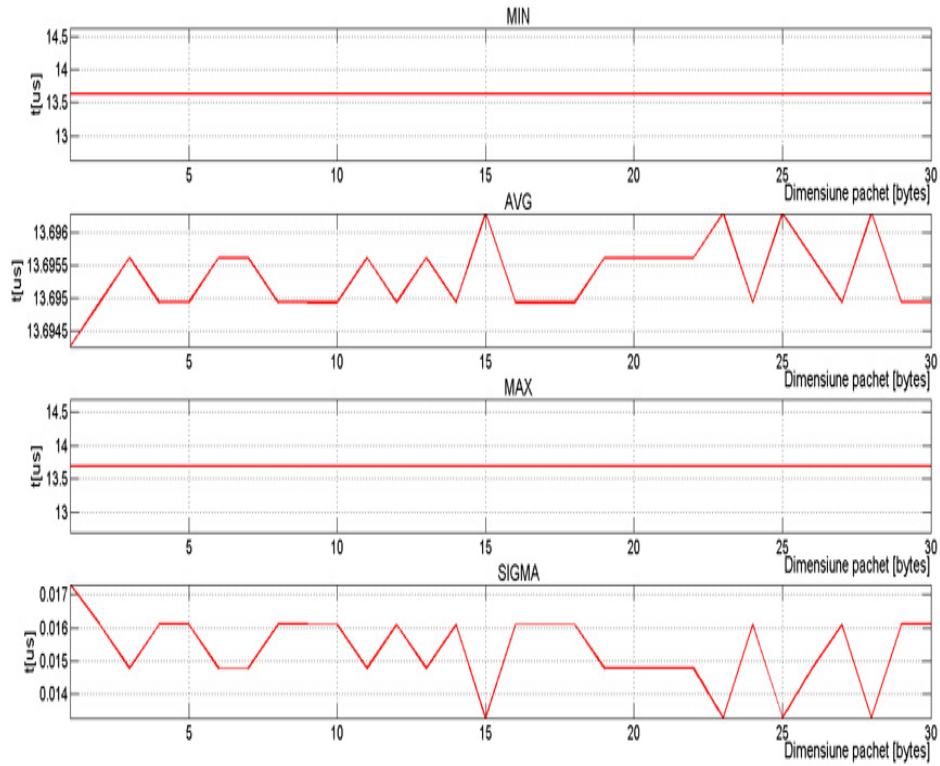


Figura 6-3 Valori minime, medii și maxime pentru componenta  $T_{TX\_ON}$  – platforma CC2420

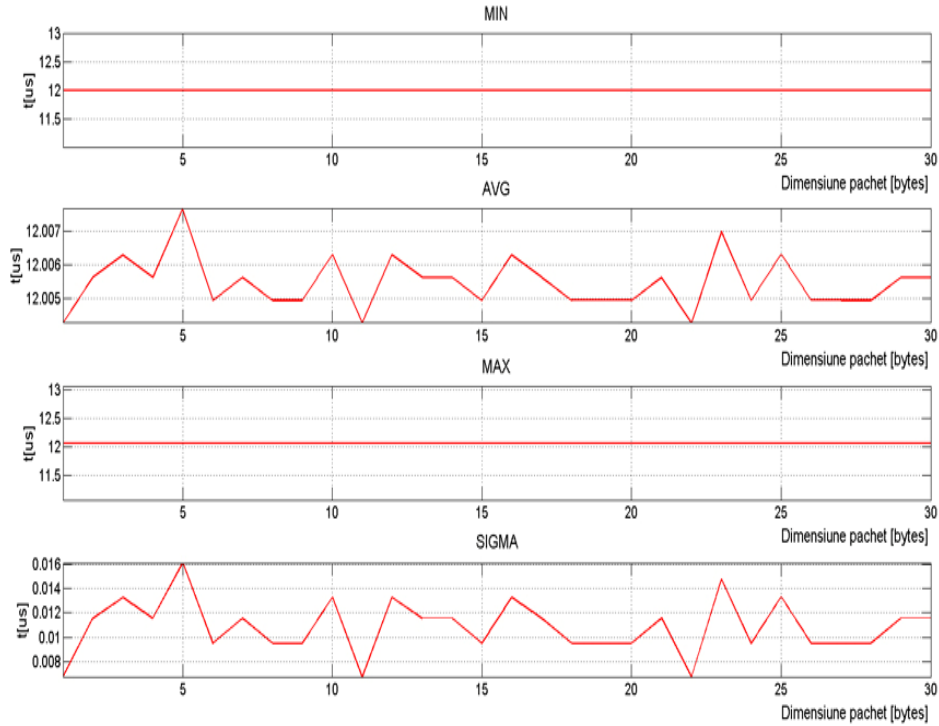


Figura 6-4 Valori minime, medii și maxime pentru componenta  $T_{TX\_ON}$  – platforma CC2500

Se poate observa și din măsurătorile obținute faptul că se menține o valoare maximă constantă. Această valoare este cea mai importantă de luat în considerare într-un mediu de timp real.

Altă componentă de timp analizată este  $T_{TX\_SFD}$ . Aceasta reprezintă intervalul de timp necesar modulului radio de a transmite secvența SFD. Acest timp conține de fapt întreaga secvență de preambul (din care face parte și secvența SFD) și în plus conține și operațiunea CCA. Din acest punct de vedere se poate trage o prima concluzie cum că această componentă este puternic influențată de mediul radio, fiind oarecum proporțională cu gradul de ocupare a mediului pe frecvențele de lucru ale modulelor radio. De asemenea acest interval de timp mai poate fi influențat și de modulația folosită de modulul radio precum și de capacitatea modulului de "a rezista" perturbațiilor din mediul de comunicație.

Așadar, într-un mediu liniștit din punct de vedere al perturbațiilor și folosind un modul radio cu performanțe mai ridicate, această componentă de timp poate avea o evoluție conform Figura 6-5. Într-o situație opusă, cu un modul radio mai slab și într-un mediu perturbat pe frecvențele de comunicație, această componentă se poate prezenta ca în Figura 6-6.

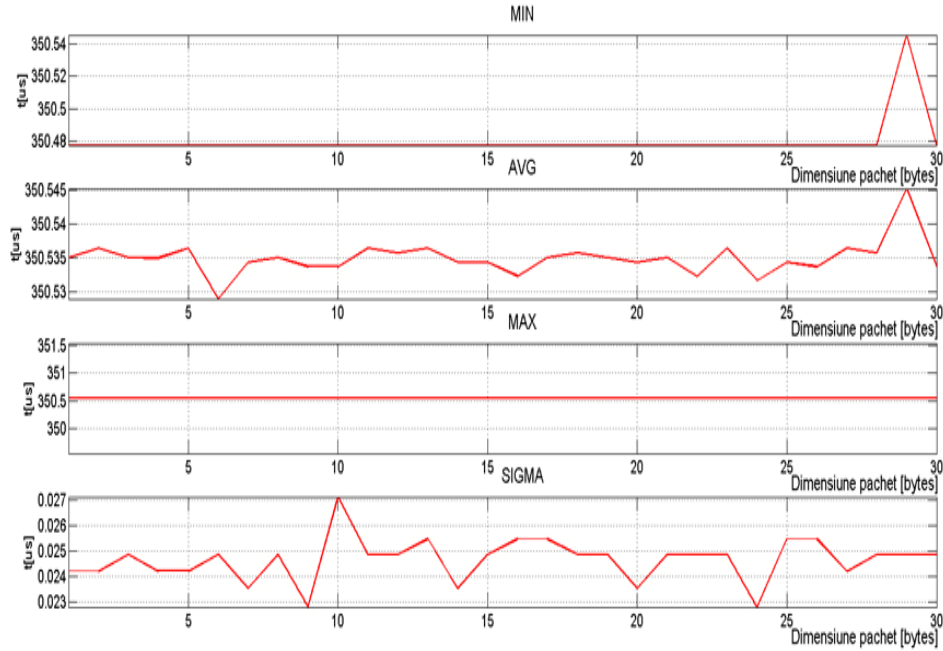


Figura 6-5 Valori minime, medii și maxime pentru componenta  $T_{TX\_SFD}$  – platforma CC2420

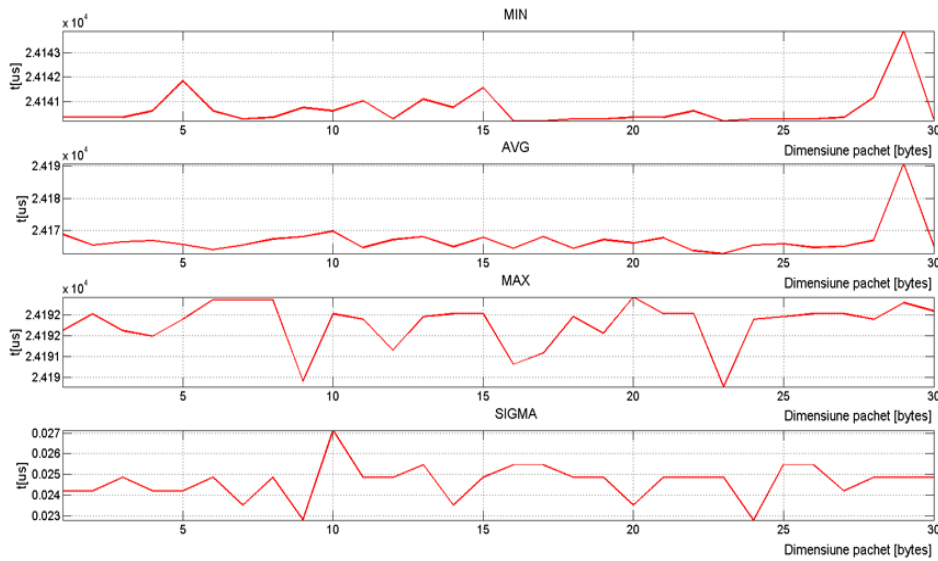


Figura 6-6 Valori minime, medii și maxime pentru componenta  $T_{TX\_SFD}$  – platforma CC2500

Următoarea componentă de timp supusă analizei este reprezentată de intervalul de timp necesar modulului radio pentru a transmite întregul pachet pe mediul radio.

Acest interval de timp se referă strict la modulul transmițător și nu are legătură cu intervalul de timp ajungerii la destinație a pachetului. Practic această componentă de timp este intervalul de timp necesar modulului transmițător pentru a modula întregul pachet. Componenta de timp în discuție este strict dependentă de performanțele modulului radio și de modulația aleasă pentru comunicare precum și de frecvența purtătoare. Astfel, intervalele de timp de acest fel sunt calculabile și predictibile. În Figura 6-7 se prezintă un astfel de interval de timp, pentru module de tip CC2420. Se menționează faptul că și în cazul folosirii platformei cu CC2500 evoluția acestei componente de timp este similară cu cea din Figura 6-7, diferind doar valorile efective ale timpilor măsurați.

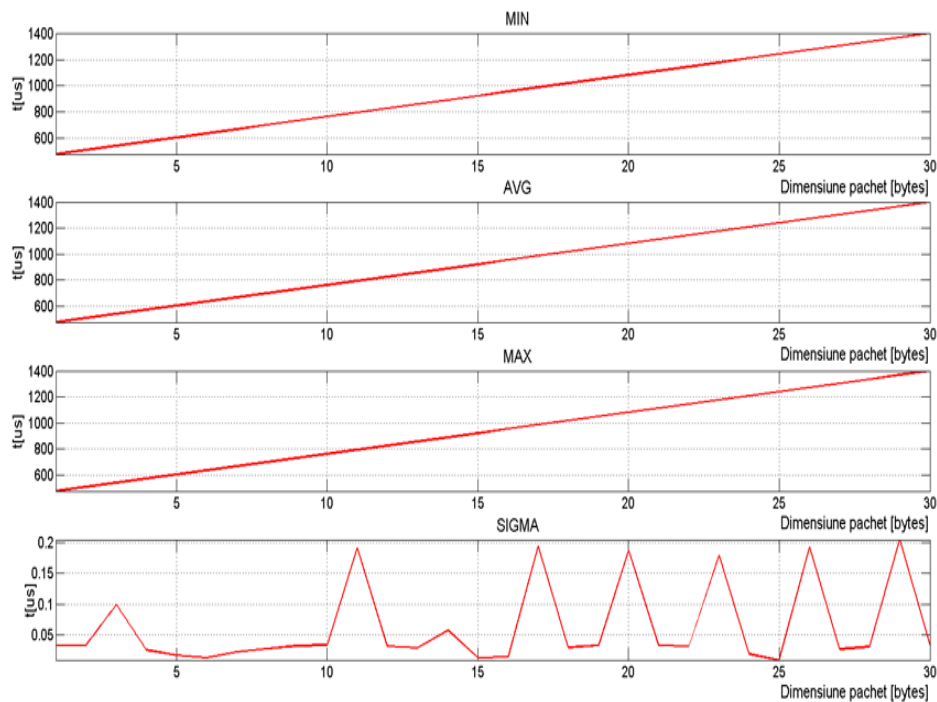


Figura 6-7 Valori minime, medii și maxime pentru componenta  $T_{TX\_PACK}$

Următorul timp analizat este intervalul de timp necesar modulului radio pentru a trece din modul transmisie în modul recepție. Din această categorie fac parte componentele de tip  $T_{RX\_ON}$ . La marea majoritate a modulelor radio existente această componentă este de obicei furnizată de producător, ca valoarea maximă, în pagina de catalog a modulului. În cazul platformei cu modulul CC2420, această componentă de timp poate ajunge la 192  $\mu s$  și la 742,5  $\mu s$  în cazul platformei CC2500. În ambele situații s-a luat în calcul cazul cel mai defavorabil. Totuși din măsurători se poate observa că în unele situații aceste valori maxime extrase și calculate folosind pagina de catalog pot fi chiar depășite (în cazul modulului radio CC2500).



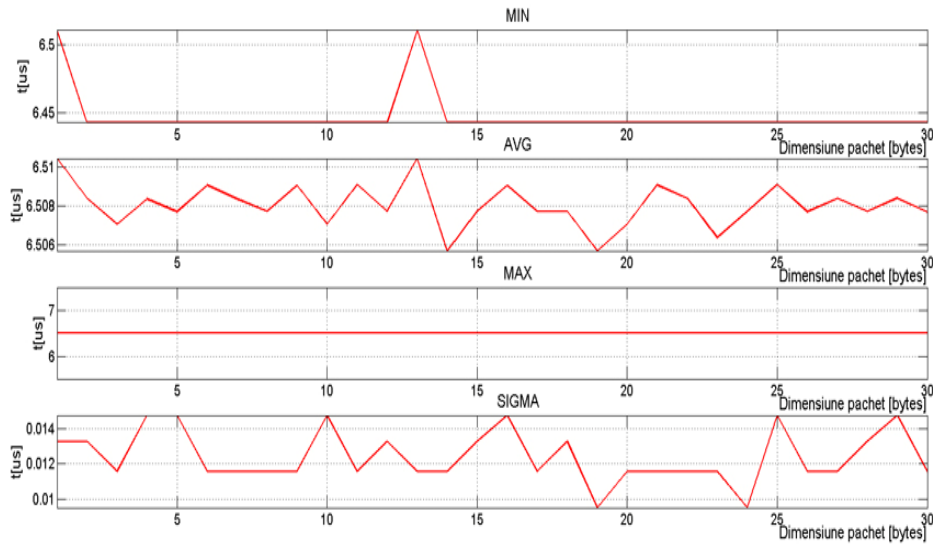


Figura 6-8 Valori minime, medii și maxime pentru componenta  $T_{RX\_ON}$  – platforma CC2420

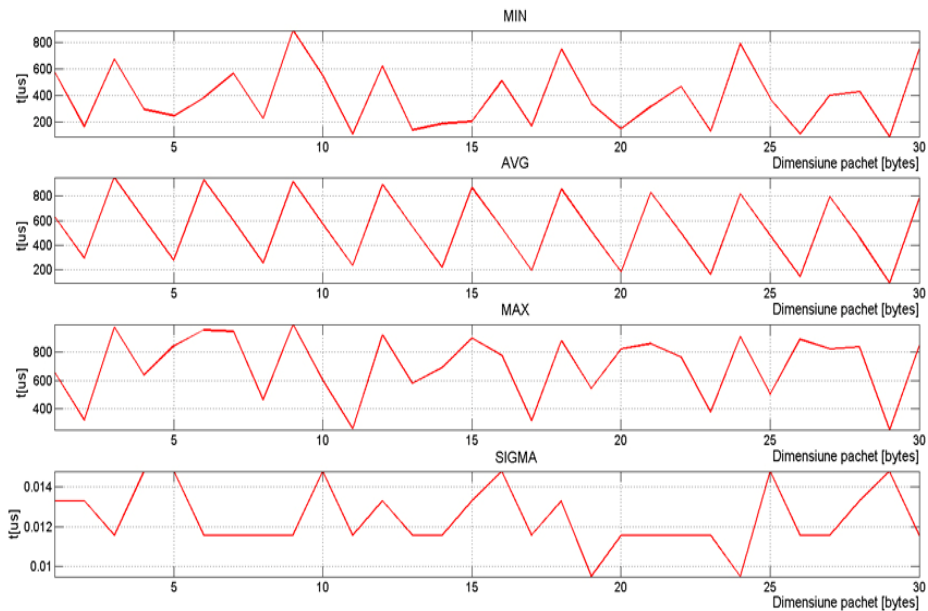


Figura 6-9 Valori minime, medii și maxime pentru componenta  $T_{RX\_ON}$  – platforma CC2500

În Figura 6-8 se poate observa cum operațiunea de schimbare din mod transmisie în mod recepție (în cazul platformei CC2420) prezintă niște valori maxime constante. În contrast cu această observație, se poate vedea cum în cazul platformei CC2500, nu doar că valorile maxime prezentate în pagina de catalog sunt depășite dar acestea sunt și extrem de instabile și impredictibile chiar și în valoare maximă.

## 82 Model temporal și cadru de măsurare pentru comunicație wireless – 6

O altă componentă de timp ce intră în categoria componentelor ce nu sunt calculabile și predictibile este și componenta ce reprezintă intervalul de timp necesar pentru ca secvența SFD transmisă de nodul A să fie recepționată de nodul B. Acest interval conține atât procedura CCA precum și secvența de preambul și sincronizare ceea ce îi conferă acestei componente caracterul de impredictibilitate din principalul motiv că durata procedurii CCA este puternic influențată de disponibilitatea canalului de comunicație și de nivelul de zgomot. În următoarele figuri se prezintă comportamentul acestei componente de timp în cazul ambelor platforme în discuție. Se poate observa din nou faptul că modulul radio CC2500 prezintă un mai mare grad de instabilitate și în cazul acestei componente de timp.

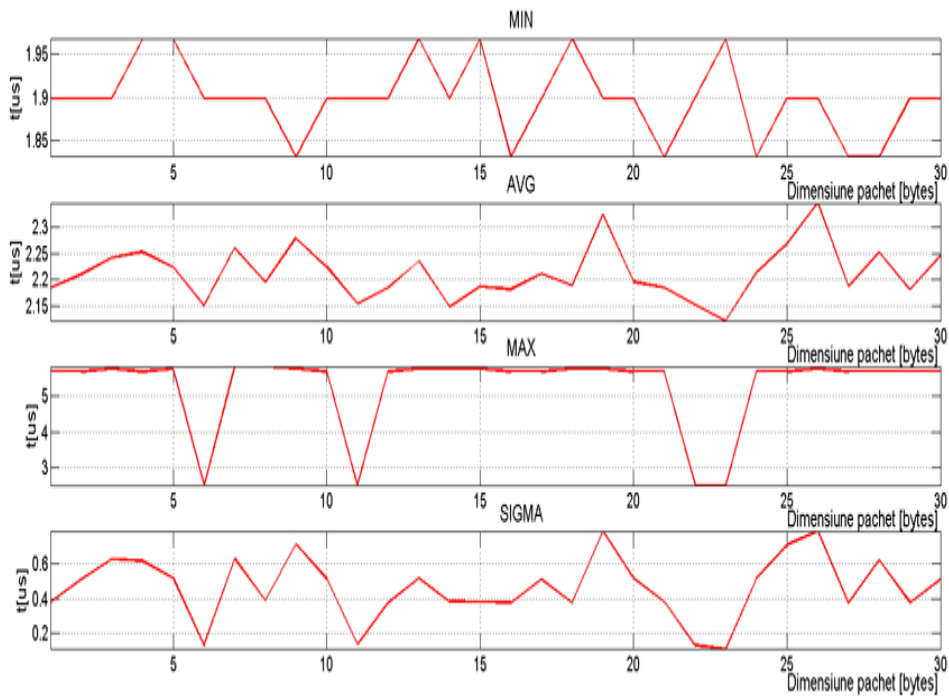


Figura 6-10 Valori minime, medii și maxime pentru componenta  $T_{RX\_SFD}$  – platforma CC2420

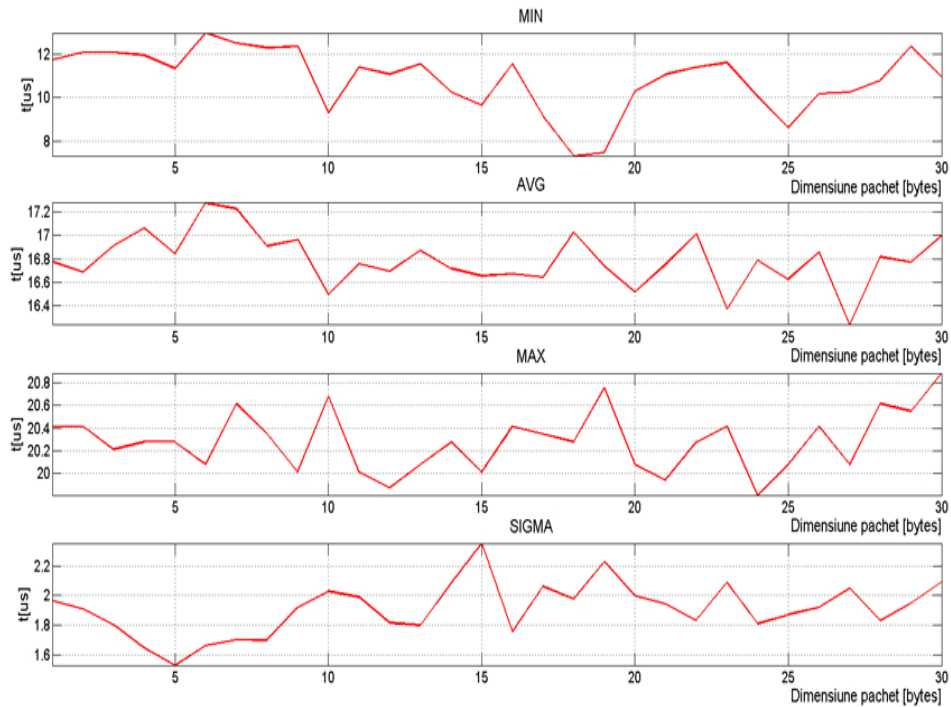


Figura 6-11 Valori minime, medii și maxime pentru componenta  $T_{RX\_SFD}$  – platforma CC2500

O ultimă componentă a cărei analiză prezintă o importanță ridicată este cea reprezentată de intervalul de timp necesar nodului B să recepționeze întregul pachet trimis de nodul A. Măsurarea acestei componente de timp începe din momentul în care modulul radio a primit secvența SFD și se termină în momentul în care modulul radio a primit întregul pachet de date. Această componentă de timp este calculabilă și predictibilă analog componentei  $T_{TX\_PACK}$  deoarece după ce s-a trimis secvența SFD modulul emițător începe imediat modularea pachetului de date iar modulul receptor o poate recepționa fără întârziere. În cazul în care după recepționarea secvenței SFD nu se primește restul pachetului de date se poate considera, după un timp maxim calculat ca și componenta  $T_{TX\_PACK}$ , că pachetul ori a ajuns corupt la destinație ori nu ajuns deloc. Această componentă de timp este dependentă de modulația folosită în comunicație, de frecvența purtătoare și de dimensiunea pachetului, astfel că se poate afla cu siguranță că după acest timp maxim emițătorul a terminat de transmis pachetul. Dacă după depășirea acestui interval de timp pachetul nu a ajuns la destinație atunci acesta poate fi considerat pierdut. În Figura 6-12 se poate vedea evoluția acestei componente de timp.

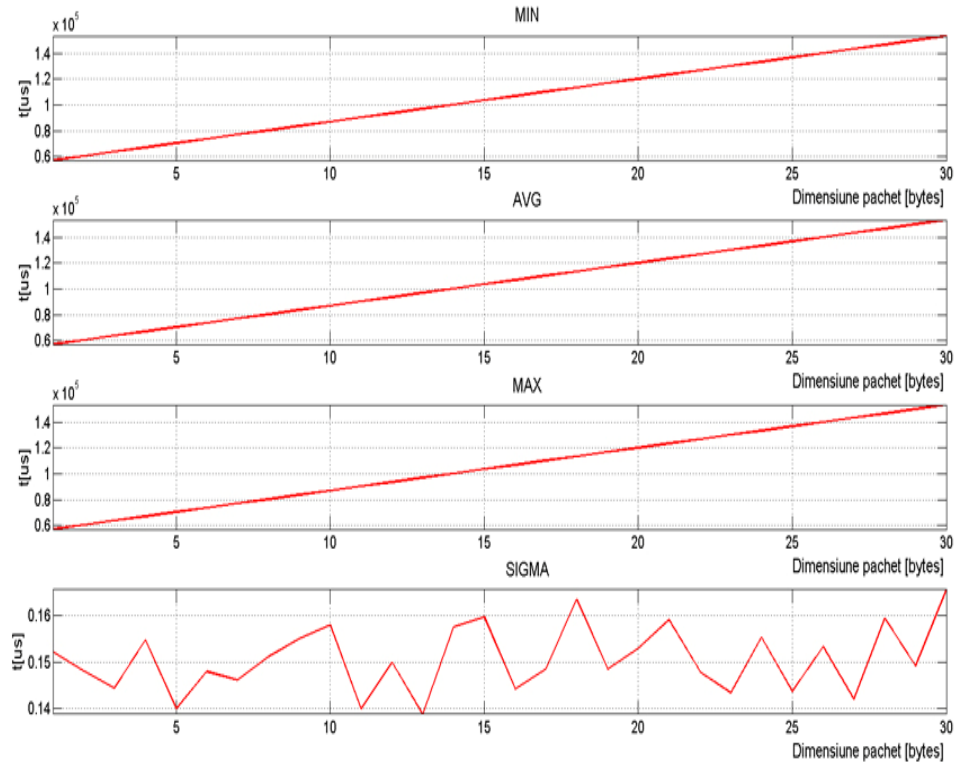


Figura 6-12 Valori minime, medii și maxime pentru componenta  $T_{RX\_PACK}$  – platforma CC2500

## 6.4 Concluzii

Gradul de predictibilitate al timpului de transfer al unui pachet de la un modul sursă la un modul destinație este dat de gradul de predictibilitate a fiecărui interval de timp ce compune acest interval. Cunoașterea acestui interval maxim implică cunoașterea maximelor fiecărei componente de timp. O parte din aceste componente de timp sunt perfect calculabile și astfel nu introduc impredictibilitate sistemului. Alte componente de timp uneori sunt strict dependente de modulul radio și astfel în cea mai mare parte se regăsesc în pagina de catalog al acestuia. Există totuși și componente de timp pentru care efortul de a stabili valori maxime este considerabil.

Cu ajutorul acestui model de timp un dezvoltator de aplicații folosind module radio și integrându-se într-un sistem în timp real, poate identifica, calcula și aproxima componentele de timp a întârzierilor introduse de comunicația wireless la nivel de acces la mediu. Mai mult decât atât se pot folosi ecuațiile definite anterior pentru a se putea calcula și aproxima penalitățile de timp introduse de modulul radio la nivelul sistemului de operare sau a firmware-ului ce rulează pe dispozitivul respectiv, în cazul de față nodurile unei rețele.

Majoritatea componentelor de timp descrise în secțiunea 6.1 sunt ori calculabile, și astfel nu afectează predictibilitatea sistemului, ori sunt furnizate de către producătorul modulului radio în pagina de catalog a acestuia. Din analiza făcută în acest capitol rezultă că există și componente de timp ce pot afecta predictibilitatea sistemului ceea ce nu este permis într-un sistem timp real.

Una din componentele de timp ce poate afecta comportamentul timp real al sistemului este  $T_{PLL}$ . Această componentă apare atunci când modulul radio are nevoie să își recalibreze PLL-ul intern. Acest lucru se poate întâmpla în trecerea din starea de recepție în starea de transmisie sau invers, când modulul radio își schimbă starea din transmițător în receptor. În marea majoritate a timpului de funcționare a unui modul radio, acesta se află în starea de recepție și doar atunci când i se cere explicit să transmită date acesta își schimbă starea pentru transmisie după care revine la starea de recepție. Mai mult, majoritatea modulelor radio, chiar și pentru transmisii succesive este necesară schimbarea de stare din transmițător în receptor și apoi din nou în transmițător. În unele situații, în cadrul acestor schimbări de stare, este necesar să se facă o recalibrare a sistemului PLL al modulului radio. Durata acestei calibrări variază mult dar o valoare maximă este dată de obicei în pagina de catalog a modulului radio.

Ținând cont de situația cea mai defavorabilă, în cazul de față durata de timp maximă de recalibrare a sistemului PLL oferită de producător, se poate menține predictibilitatea sistemului.

Dacă în cazul componentei de timp  $T_{PLL}$ , folosind situația cea mai defavorabilă, se păstrează caracterul timp real al sistemului, nu același lucru se poate afirma și despre componenta  $T_{CCA}$ . Această componentă, după cum s-a enunțat la începutul acestui capitol, reprezintă durata de timp pentru ca modulul radio să efectueze operațiunea CCA înainte să înceapă transmisia. În urma acestei operațiuni modulul radio decide dacă și când canalul de comunicație nu este ocupat. Durata de timp a acestei operațiuni depinde doar de gradul de ocupare al canalului de comunicație sau de nivelul de zgomot de pe acesta. Ținând cont de caracteristicile mediului de comunicație wireless se poate afirma faptul că această componentă de timp nu este deloc predictibilă. O soluție ar putea fi aproximarea acestei valori cu o valoare maximă dar rămâne problema găsirii acestei valori maxime. O altă soluție, adecvată pentru un sistem timp real, este găsirea unui protocol de comunicație unde această operațiune de CCA nu este necesară, cum ar fi protocoalele ce nu sunt bazate pe coliziuni (ex. TDMA, FDMA...).

## 7 Greedy TDMA: Protocol timp-real eficient de acces la mediu

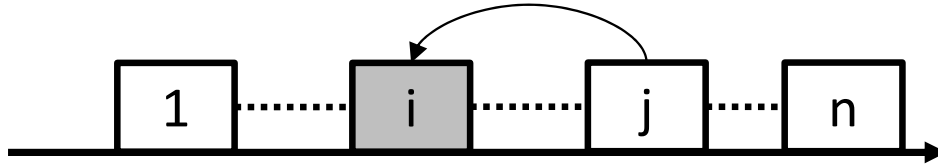
Într-o politică de acces de tip TDMA, conform celor prezentate anterior în capitolul 4.3, fiecărui nod participant la comunicație i se atribuie un slot de timp pe care se consideră că-i va fi asignat pe toată durata existenței acestui nod în respectivul mediu de comunicație. Politica TDMA oferă un mediu de comunicație predictibil și stabil atât pentru situația în care nodurile sunt conectate între ele prin fir dar și pentru situația când acestea comunică în mediul radio. Principalele dezavantaje ale acestei politici de comunicare sunt în primul rând necesitatea ca nodurile rețelei să fie configurate în prealabil prin stabilirea sloturilor de timp și prin sincronizarea ceasurilor interne, timpuri mari de răspuns în cazul unei rețele cu noduri numeroase dar și o puternică limitare a capacității de transmisie a canalului. Această limitare apare în situațiile în care unele noduri ale rețelei transmit un volum foarte mare de date pe când alte noduri transmit puține date și la interval mare de timp. În această situație gradul de ocupare al canalului de comunicație este mic deși cererea este foarte ridicată din partea unor noduri.

În următoarele paragrafe se va prezenta o schemă de comunicare menită să îmbunătățească aspectele legate de gradul de ocupare al canalului în situația prezentată anterior. Metoda este bazată pe conceptul de *slot-stealing*. În această secțiune se prezintă o politică de acces fără a fi particularizată pentru un mediu de comunicație. De asemenea conceptele vor fi prezentate în domeniul discret, folosind unități de timp, și nu în domeniul timp.

### 7.1 Politica de acces Greedy TDMA

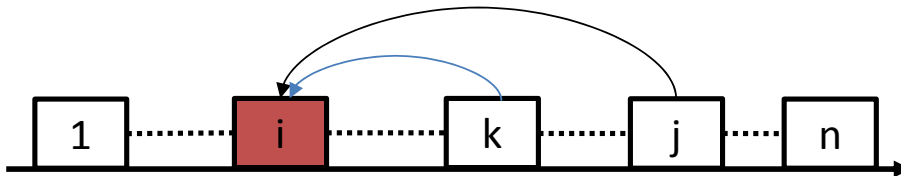
Schema de acces Greedy TDMA se bazează pe conceptul de *slot-stealing* aplicat politicii TDMA în care un nod al rețelei poate să folosească slotul de timp al unui alt nod atunci când acesta nu îl folosește. În acest mod, nodurile care și-au pregătit un volum mare de date pentru a fi transmise vor putea folosi mai multe sloturi de timp în cadrul aceluiași cadru TDMA pentru a-și transmite toate datele.

În continuare se va considera o rețea de  $n$  noduri într-o schemă TDMA clasică astfel că fiecare nod are asociat un slot de timp. De asemenea se consideră că fiecare nod are asociat un număr de ordine cu rol de adresă astfel că fiecărui nod  $i$  se va atribui slotul de timp corespunzător adresei sale (ex. nodul cu adresa 1 va primi slotul de timp 1, nodul cu adresa 2 va primi slotul de timp 2... etc). De asemenea se consideră că sloturile de timp au aceeași dimensiune. Cadrul TDMA va avea în această situație structura din Figura 4-1. Ideea de bază a conceptului de *slot-stealing*, prezentată și în Figura 7-1, este aceea că dacă considerând că un nod oarecare  $i$  nu are date de transmis în slotul de timp  $i$  asignat atunci un alt nod  $j$  care are date deja pregătite pentru transmisie în slotul de timp  $i$  poate utiliza slotul neutilizat  $i$  al nodului  $i$  pentru a transmite.

Figura 7-1 Conceptul *slot-stealing*

În plus în cadrul aceluiași frame TDMA când ajunge la rând slotul de timp al nodului  $j$  (considerând  $i < j$ ) acesta îl poate folosi în continuare dacă mai are date pregătite pentru a fi transmise chiar dacă în frame-ul TDMA curent a utilizat și slotul de timp  $i$  care într-o structură clasică TDMA nu ar fi avut dreptul să-l utilizeze. Astfel, în cazul în care nodul  $i$  nu a avut date de transmis nodul  $j$  a putut folosi două sloturi de timp din același frame TDMA pentru a transmite mai multe date pe care le-a avut pregătite în prealabil. În acest fel nu s-a pierdut slotul de timp al nodului  $i$  ceea ce s-ar fi întâmplat într-o politică clasică TDMA. În plus, volumul de date pe care nodul  $j$  l-a transmis folosind un singur cadru TDMA, într-o politică clasică acesta ar fi avut nevoie de două frame-uri TDMA.

Se consideră în continuare o situație asemănătoare dar în care există 2 noduri  $j$  și  $k$  (pentru care  $i < k < j$ ) care au pregătite multe date de transmis și în cazul în care nodul  $i$  nu își folosește slotul de timp alocat ambele noduri  $j$  și  $k$  ar folosi slotul liber al nodului  $i$ . În acest caz se poate ajunge la o situație de conflict exemplificat în Figura 7-2:

Figura 7-2 Situație de conflict în *slot-stealing*

Politica de acces Greedy TDMA oferă o soluție pentru problema prezentată anterior. Pentru descrierea noii politici de acces se va considera în primul rând că un slot de timp TDMA se împarte în TSs unități discrete de timp, unități ce vor fi referite în continuare ca TU (Time Unit) iar un slot de timp se va descrie ca în Figura 7-3:

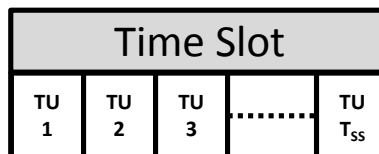


Figura 7-3 Discretizarea unui slot de timp în unități de timp

De asemenea se va defini și capacitatea unei unități de timp TU ca fiind numărul de octeți ce pot fi transferați în mediul de comunicație pe durata unei unități de timp TU și se va nota cu  $CAP_{TU}$  iar unitatea de măsura va fi octetul (byte). Ținând cont de

## 88 Greedy TDMA: Protocol timp-real eficient de acces la mediu – 7

numărul de noduri din rețea notat cu  $n$  se poate exprima numărul de unități de timp dintr-un frame TDMA:

$$T[\text{TU}] = TS_S \cdot n \quad (7-1)$$

Se mai definește în continuare și capacitatea unui slot de timp, adică numărul de octeți ce pot fi transferați într-un timeslot în funcție de dimensiunea acestuia în unități de timp TU:

$$CAP_{TS} = TS_S \cdot CAP_{TU} \quad (7-2)$$

În final rezultă modul în care se poate defini întreaga capacitate a unui frame TDMA (numărul de octeți ce se pot transmite prin mediul de comunicație într-un frame TDMA):

$$CAP_{\text{TDMA}} = CAP_{TS} \cdot n = TS_S \cdot n \cdot CAP_{TU} \quad (7-3)$$

În plus față de acestea se consideră că fiecare nod al rețelei are asignată o adresă proprie și unică iar pe baza acestei adrese  $i$  se alocă un slot de timp din cadrul TDMA. Așadar sloturile de timp din cadrul TDMA se alocă nodurilor în ordinea crescătoare a adreselor acestora. În acest sens spre exemplu nodul cu adresa 1 va avea asignat slotul de timp 1, nodul cu adresa 2 va avea asignat slotul de timp 2 ce urmează imediat după slotul de timp 1 după cum este prezentat în Figura 7-4:

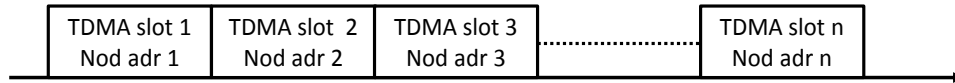


Figura 7-4 Asignarea sloturilor de timp în ordinea crescătoare a adreselor nodurilor

În termeni generali politica de acces Greedy TDMA propune faptul că un nod poate accede la slotul de timp al unui alt nod numai dacă acesta nu detectează o transmisie nici din partea nodului căruia dorește să-i folosească slotul de timp dar nici din partea nodurilor dintre acestea. În plus un nod poate accede doar la sloturile de timp ale nodurilor ce au sloturile de timp înaintea slotului acestuia. O altă restricție ar fi aceea că nodurile ce au date de transmis în slotul propriu asignat sunt obligate să-și înceapă transmisia la începutul slotului de timp asignat.



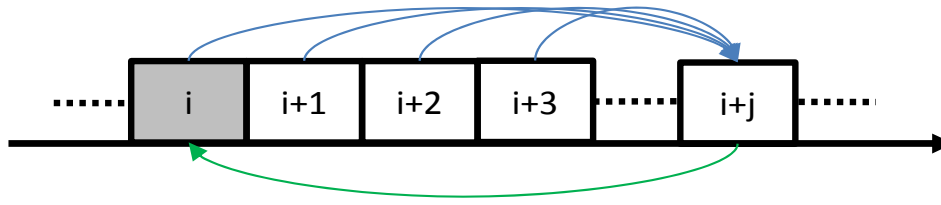


Figura 7-5 Conceptul Greedy TDMA

În Figura 7-5 se descrie în linii mari conceptul prezentat în acest paragraf și anume se arată că dacă în momentul slotului de timp  $i$  nodul  $N_i$  care are asignat acest slot de timp, nu are date pregătite pentru a fi transmise și astfel nu va folosi slotul de timp alocat și nodul  $N_{i+j}$  are date pregătite pentru a fi transmise atunci acesta din urmă își va putea transmite datele în slotul de timp curent al nodului  $N_i$  doar dacă se va asigura că nodurile dintre  $N_{i+j}$  și  $N_i$  nu vor accede la rândul lor la slotul de timp curent al nodului  $N_i$ . În acest fel se vor evita situațiile de conflict asemănătoare celei prezentate în Figura 7-2.

Dezavantajul acestei soluții este acela că detecția faptului că nodurile dintre  $N_{i+j}$  și  $N_i$  nu vor transmite în slotul de timp curent consumă unități de timp  $TU$  din maximumul  $TS_s$  al unui slot de timp rămânând mai puține unități de timp pentru transmisia efectivă a datelor. Astfel se mai impune o restricție pentru nodul  $N_{i+j}$  și anume că acesta nu va putea transmite datele decât dacă dimensiunea pachetului de date ce îl poate trimite este mai mică sau egală cu capacitatea rămasă a slotului de timp. Astfel dacă nodul  $N_{i+j}$  a așteptat un număr de  $w$  unități de timp pentru a se asigura de disponibilitatea slotului de timp atunci restricția de transmisie este descrisă de inecuația (7-4) unde  $P_s$  definește dimensiunea pachetului de date în octeți, având aceeași semnificație ca și în secțiunea 6.1 :

$$P_s \leq w \cdot CAP_{TU} \quad (7-4)$$

De asemenea trebuie precizat faptul că sloturile de timp trebuie respectate. Astfel în cazul în care nodul  $N_{i+j}$  își începe transmisia în slotul de timp al nodului  $N_i$  acesta este obligat să încheie transmisia înainte de începerea următorului slot de timp, în cazul de față slotul  $i+1$ . În plus se mai face precizarea că dacă nodul  $N_{i+j}$  a folosit slotul de timp al nodului  $N_i$  din cadrul TDMA curent, acesta nu își pierde dreptul folosirii slotului de timp  $i+j$  când vine momentul de timp oportun. Astfel în situația în care nodul  $N_{i+j}$  dispune de un număr foarte mare de date el va folosi sloturile de timp inactive și va putea emite de mai multe ori într-un cadru TDMA.

În cele ce urmează se va prezenta un exemplu pe un caz particular al celor descrise mai sus. Se va considera o rețea formată din 5 noduri și dimensiunea unui slot de timp de 8  $TU$ . De asemenea se consideră că slotul de timp curent este slotul 1 asignat nodului  $N_1$  care nu dispune de date pregătite pentru transmisie la fel ca și nodurile  $N_2$ ,  $N_3$  și  $N_4$ . În schimb se consideră că doar nodul  $N_5$  are date pregătite pentru transmisie. Acest exemplu este descris în Figura 7-6.

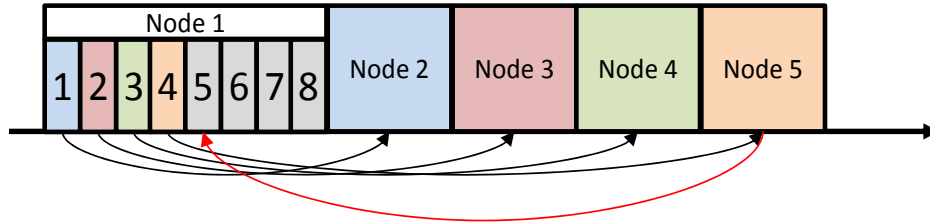


Figura 7-6 Exemplificare concept Greedy TDMA

În această stare nodul  $N_1$  nu va ocupa slotul de timp asignat 1 deoarece s-a considerat ca nu are date disponibile pentru a fi transmise lăsând astfel canalul neocupat. Într-o soluție TDMA clasică intervalul de timp al acestui slot de timp ar fi pierdut, canalul de comunicație fiind neocupat. În cazul de față însă nodurile vor avea următorul comportament:

- nodul 2 va detecta faptul că nodul 1 nu începe transmisia de date în TU 1 și ar putea ocupa ce a rămas din slotul de timp dacă ar avea datele pregătite începând cu TU 2;
- nodul 3 va detecta faptul că nodul 1 nu începe transmisia în TU 1 și că nici nodul 2 nu începe transmisia în TU 2 și astfel în cazul în care ar avea date disponibile și-ar putea anunța prezența prin începerea transmisiei în TU 3;
- nodul 4 va detecta faptul că nodul 1 nu începe transmisia în TU 1, că nodul 2 nu începe transmisia în TU 2 și că nici nodul 3 nu începe transmisia în TU 3. Din nou în cazul în care ar avea date disponibile acesta le-ar putea transmite începând cu TU 4;
- nodul 5 va detecta faptul că nodul 1 nu începe transmisia în TU 1 și că nodul 2, 3 și 4 nu încep transmisia în TU 2, TU 3 respectiv TU 4 și astfel începând cu TU 5 acesta poate ocupa restul slotului de timp. Considerând ca nodul 5 are datele pregătite pentru a fi transmise acesta mai trebuie să verifice starea de adevăr a inegalității (7-4). În cazul în care această inegalitate se verifică, mai exact pachetul de date pe care nodul 5 îl are disponibil pentru transmisie încapă în restul slotului de timp rămas atunci acest nod 5 își poate începe transmisia datelor începând cu nodul TU 5 și are la dispoziție TU 5 până la TU 8 inclusiv, un număr de 4 unități disponibile;

Se poate observa că aplicând această schemă de acces la mediu slotul de timp 1 nu rămâne nefolosit ca și în cazul TDMA. Totuși și în această situație o parte a slotului de timp 1 este pierdut pentru a se detecta inactivitatea nodurilor în acest slot de timp dar îmbunătățirea este substanțială.

O analiză mai amănunțită asupra celor prezentate până acum poate să scoată în evidență detaliul că toate nodurile au posibilitatea să folosească sloturi nefolosite ce nu le aparțin cu excepția nodului 1. Astfel nodul ce are asignat primul slot de timp dintr-un cadru TDMA nu va putea folosi sloturi de timp ale altor noduri deoarece nu există alte noduri înainte lui în cadrul TDMA. Totuși, o posibilă soluție poate fi găsită analizând Figura 7-7:

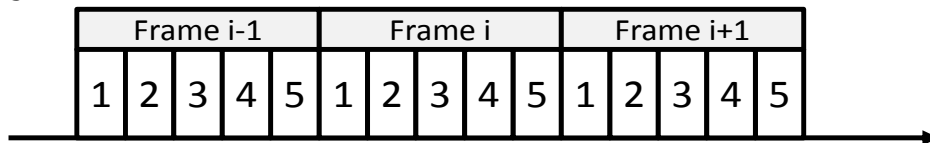


Figura 7-7 Iterație cadre TDMA

În situația prezentată în figura anterioară se consideră 3 iterații a cadrului TDMA în timp iar cadrul TDMA curent este  $i-1$ . Se consideră că în cadrul TDMA curent  $i-1$  nodul 5 nu are date pregătite pentru a fi transmise astfel slotul de timp 5 rămâne nefolosit. Totuși, nodul 1 ar avea date pregătite pentru a fi transmise și ar putea folosi slotul de timp al nodului 5 chiar dacă slotul de timp al nodului 1 expirat. Totuși aceasta îl poate folosi considerând iterația din figură și făcând abstracție de cadrul TDMA. Astfel, într-o privire de ansamblu în mod circular se observă că nodul 1 poate detecta slotul nefolosit al nodului 5 și astfel îl poate folosi.

După cele enunțate până aici se poate concluziona faptul că soluția Greedy TDMA prezintă sub o formă generalizată conceptul mult restrâns de *slot-stealing* prezent în unele abordări particulare prezente în literatură. Acest concept este menit spre a fi folosit indiferent de mediul de comunicație folosit. Greedy TDMA îmbunătățește gradul de acces la canalul de comunicație față de o schema TDMA clasică menținând predictibilitatea oferită de TDMA în contextul folosirii acesteia într-un mediu timp-real.

## 7.2 Rezultate experimentale

Validarea politicii de acces la mediu Greedy TDMA prezentată anterior presupune implementarea unui simulator care să ofere atât corectitudinea politicii propuse cât și termeni de comparație cu politica TDMA clasică. Simulatorul trebuie doar să implementeze cele 2 politici de acces la mediu fără să țină cont de vreun caz particular de organizare sau de vreun mediu fizic.

Ținând cont de aceste specificații s-a implementat un simulator pentru evaluarea și compararea Greedy TDMA. Simulatorul lucrează într-un mediu de timp discret și ține cont de parametrii introduși în secțiunea anterioară 7.1 .

Principalii parametri de intrare ai simulatorului sunt:

- Numărul de noduri ale rețelei: ținând cont că se asignează un slot timp pentru fiecare nod al rețelei acest parametru are și semnificație de dimensiunea de cadru TDMA.
- Numărul de unități de timp al unui slot de timp: acest parametru se identifică cu termenul TSS definit anterior cu unitatea de măsură 1 TU.
- Capacitatea unui slot de timp: acest parametru definește numărul de octeți ce pot fi transferați pe unitatea de timp și se identifică cu termenul CAPTU cu unitatea de măsură B/TU (octeți pe unitatea de timp).
- Numărul de cadre TDMA: acest parametru specifică practic durata simulării în cadre TDMA.
- Dimensiunea maximă a pachetului de date: parametrul specifică numărul maxim de octeți ce pot fi transferați de către un nod. Simulatorul consideră că nodurile sunt identice.
- Probabilitatea de transmisie: acest parametru reprezintă probabilitatea unui nod de a avea date disponibile pentru a fi transmise la un moment dat.

Simulatorul folosește un fișier de configurare în format *libconfig*, folosit în sistemele POSIX, în care parametrii definiți mai sus pot fi modificați pentru a simula scenarii diferite. Pentru a se putea genera un mediu de simulare valid este necesar ca dimensiunea maximă a pachetului de date să nu depășească capacitatea unui slot de timp astfel fiind necesară verificarea inegalității (7-5) în caz contrar simularea va fi oprită.

$$P_{S_{MAX}} \leq TS_S \cdot CAP_{TU} \quad (7-5)$$

Simularea decurge în 3 etape: generarea mediului de simulare, simularea comunicării într-o schema TDMA clasică folosind mediul de simulare generat și simularea comunicării într-o schema Greedy TDMA folosind același mediu de simulare.

Pentru a se putea obține termene de comparație între cele două politici de acces este necesar ca în ambele situații să se folosească aceleași date de intrare. Aceste date de intrare sunt reprezentate de niște variabile generate ce formează mediul de simulare. Așadar pentru fiecare nod se generează un set de două variabile pentru fiecare cadru TDMA ce se va simula. Variabilele reprezintă dimensiunea pachetului și probabilitatea de transmisie. Astfel pentru fiecare nod și pentru fiecare cadru TDMA al fiecărui nod se va genera aleatoriu dimensiunea pachetului ce îl are pregătit pentru transmisie cu valoarea maximă definită ca parametru de intrare și probabilitatea de transmisie a acestuia ce va ține cont de valoarea maximă definită de asemenea ca parametru de intrare.

Rezultatele și evaluările furnizate de simulator sunt:

- Numărul total de octeți transferați de fiecare nod: acest rezultat este același indiferent de politica de acces fie TDMA sau Greedy TDMA
- Numărul total de unități de timp folosite de fiecare nod: acest rezultat este de asemenea același indiferent de politica de acces
- Numărul total de sloturi de timp folosite de fiecare nod: acest rezultat este de asemenea același indiferent de politica de acces folosită
- Numărul de ordine al ultimului cadru TDMA folosit pentru transmisie pentru fiecare nod: această valoare diferă în funcție de politica de acces simulată și prezintă principalul termen de comparație. Această valoare este de așteptat ca în cazul unei politici TDMA să fie chiar ultimul cadru TDMA de simulare, iar în cazul unei politici Greedy TDMA este de așteptat să fie mai mic.

Simulatorul oferă mecanisme de detecție a coliziunilor. Lipsa unor astfel de mecanisme ar putea duce la o încălcare atât a schemei TDMA cât și a schemei Greedy TDMA. De asemenea simulatorul prezintă și limitarea privind faptul că primul nod al unei cadre TDMA nu are posibilitate de a accede la nici un slot al altui nod.

În cele ce urmează se vor prezenta primele rezultate obținute în cazul folosirii unor date care să implementeze exemplul folosit în secțiunea anterioară 7.1 și descris în Figura 7-6. Pentru aceasta se vor folosi parametrii de simulări din următorul tabel:

Tabel 7-1. Parametrii de intrare simulator

Număr de noduri	5
$TS_S$	16
$CAP_{TU}$	4
Număr cadre TDMA	8
Dimensiunea maximă a pachetului de date	64
Probabilitatea de transmisie a unui nod	60

Folosind parametrii de mai sus, în primul pas al simulării se va genera mediul de simulare folosit atât pentru a simula o schema TDMA cât și o schema Greedy TDMA.

Tabel 7-2. Mediul de simulare generat

	<b>Nod 1</b>	<b>Nod 2</b>	<b>Nod 3</b>	<b>Nod 4</b>	<b>Nod 5</b>
<b>Frame 1</b>	(57,88)	(51,79)	(43,45)	(19,52)	(19,85)
<b>Frame 2</b>	(57,45)	(25,88)	(28,41)	(34,93)	(33,95)
<b>Frame 3</b>	(18,61)	(47,29)	(2,50)	(18,27)	(22,48)
<b>Frame 4</b>	(25,27)	(57,7)	(22,47)	(46,76)	(36,93)
<b>Frame 5</b>	(56,47)	(58,73)	(30,45)	(14,29)	(26,52)
<b>Frame 6</b>	(29,62)	(39,42)	(12,27)	(34,48)	(1,4)
<b>Frame 7</b>	(61,25)	(12,35)	(41,37)	(36,30)	(44,3)
<b>Frame 8</b>	(33,1)	(19,22)	(29,38)	(44,93)	(25,8)

În Tabel 7-2 se prezintă mediul de simulare ce a fost generat după cum s-a prezentat anterior. În fiecare celulă a tabelului sunt trecute dimensiunea pachetului și probabilitatea sa de transmisie. După cum se observă este prezentat întregul mediu de simulare. Sper exemplu în cadrul TDMA 3 nodul 4 va avea un pachet de 18 octeți pregătit pentru transmisie cu o probabilitate 27.

Următorii doi pași ai simulării au constat în trecerea mediului de simulare printr-o schemă de acces de tip TDMA după care aceleași valori au fost folosite pentru a simula o schema de acces Greedy TDMA. În următorul tabel se prezintă rezultatele simulării:

Tabel 7-3. Rezultatele simulării

	<b>Total octeți transferați</b>	<b>Total unități de timp</b>	<b>Total sloturi de timp</b>	<b>Ultimul cadru TDMA utilizat</b>	
				<b>TDMA</b>	<b>Greedy TDMA</b>
Nod 1	232	61	5	8	8
Nod 2	174	45	5	8	6
Nod 3	207	55	8	8	4
Nod 4	121	32	5	7	5
Nod 5	118	32	5	8	6

Analizând rezultatele conținute în Tabel 7-3 se poate observa, comparând ultimul cadru TDMA utilizat într-o schemă TDMA clasică cu rezultatele obținute într-o schemă Greedy TDMA faptul că în cazul celei din urmă majoritatea nodurile și-au terminat datele pregătite mult mai repede față de schema TDMA clasică. Acest fapt duce în primul rând la o îmbunătățire a gradului de utilizare a întregului canal de comunicație. De asemenea prin faptul că majoritatea nodurilor și-au terminat transmisia datelor mult mai repede face ca aceste date să fie disponibile rețelei mult mai repede față de schema clasică.

Aceste rezultate se pot observa și mai bine aplicând simularea la un număr mai mare de noduri precum și măbind timpul de simulare la un număr mai mare de cadre de timp. Spre exemplu în Figura 7-8 se prezintă rezultatele unei simulări similare cu cea prezentată sub formă tabelară anterior dar când se aplică asupra unei rețele formate din 10 noduri timp de 30 de cadre TDMA/GTDMA. În aceeași manieră s-au obținut și rezultatele din Figura 7-9 aplicându-se simularea asupra unei rețele formate din 50 de noduri timp de 30 de cadre.

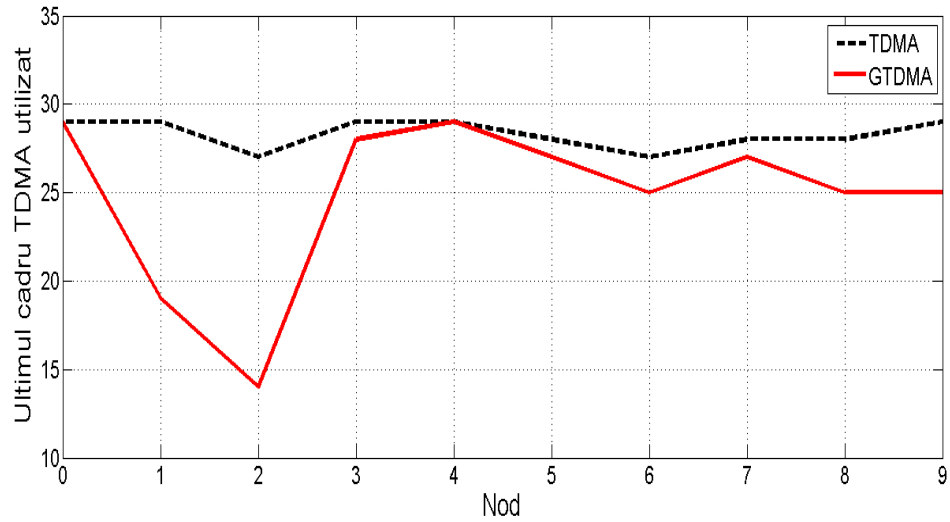


Figura 7-8 Simulare Greedy TDMA cu 10 noduri

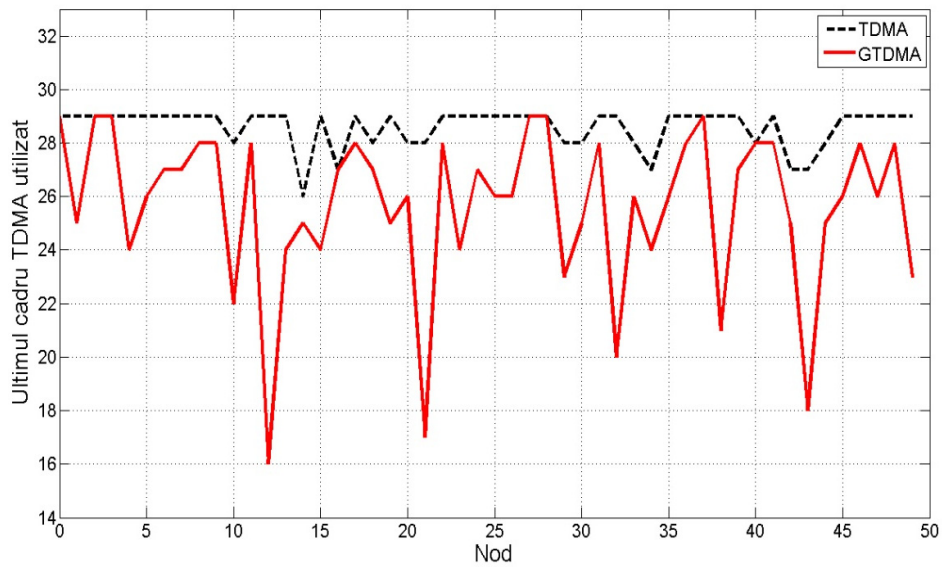


Figura 7-9 Simulare Greedy TDMA cu 30 noduri

Rezultatele se păstrează și în cazul în care se consideră o rețea cu un număr mai mare de noduri și un timp de simulare mai lung. Astfel în următorul caz se vor considera două situații în generarea aleatoare a mediului de simulare și anume se va considera în primul caz o generare cu distribuție uniformă a valorilor iar în cel de-al doilea caz o generare cu distribuție gaussiană. Astfel datele de intrare pentru simulator vor fi următoarele:

Tabel 7-4. Parametrii de intrare simulator

Număr de noduri	100
$TS_s$	16
$CAP_{TU}$	4
Număr cadre TDMA	100
Dimensiunea maximă a pachetului de date	64
Probabilitatea de transmisie a unui nod	80

În continuare se vor prezenta rezultatele obținute în urma simulării doar sub forma grafică, sub formă tabelară fiind mult prea mare îngreunând o analiză asupra rezultatului. Alături de rezultatele obținute se va prezenta și histograma valorilor generate pentru a se putea evidenția distribuția valorilor generate.

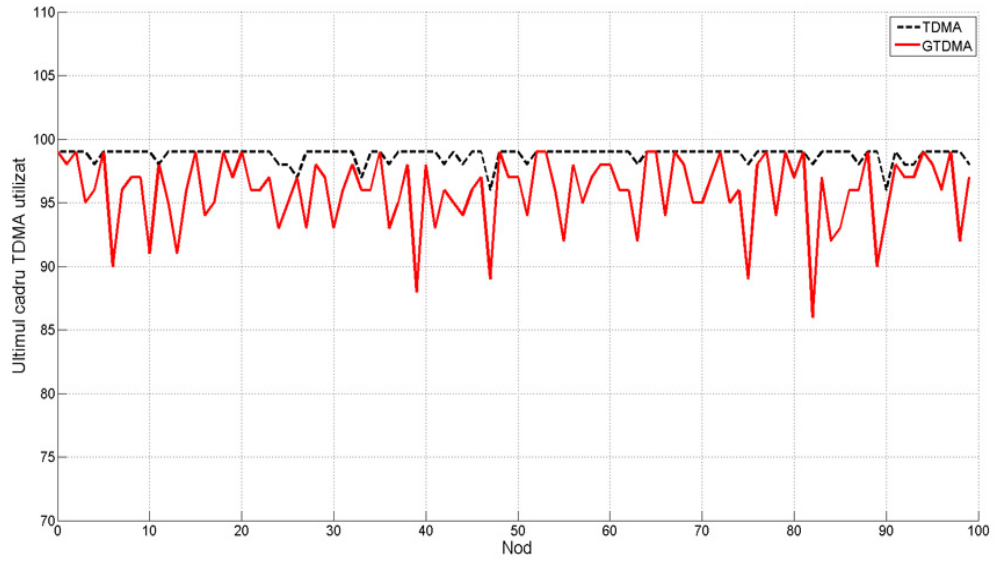


Figura 7-10 Simulare Greedy TDMA cu 100 noduri timp de 100 cadre TDMA cu distribuție uniformă a valorilor aleatoare generate

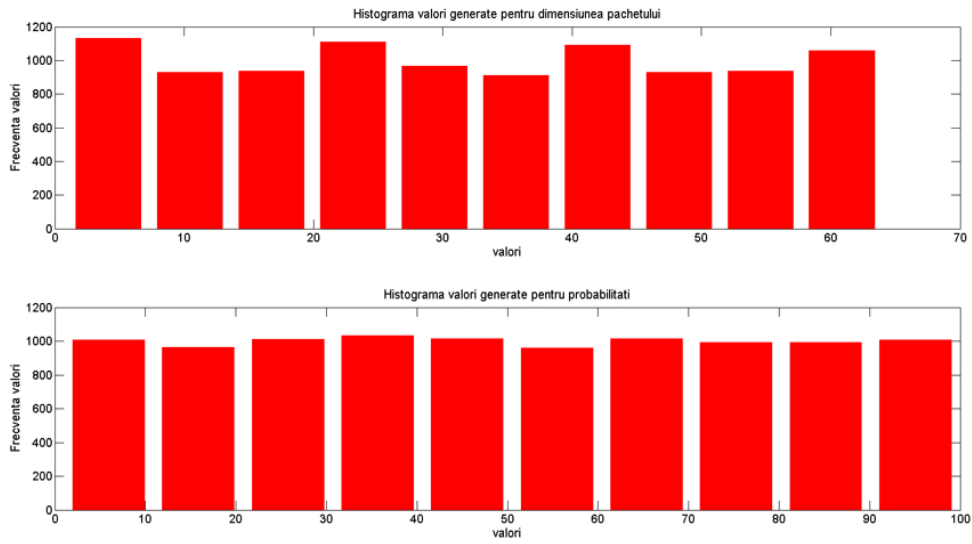


Figura 7-11 Histogramă valori generate cu distribuție uniformă a) dimensiunea pachetului b) probabilitatea de transmisie



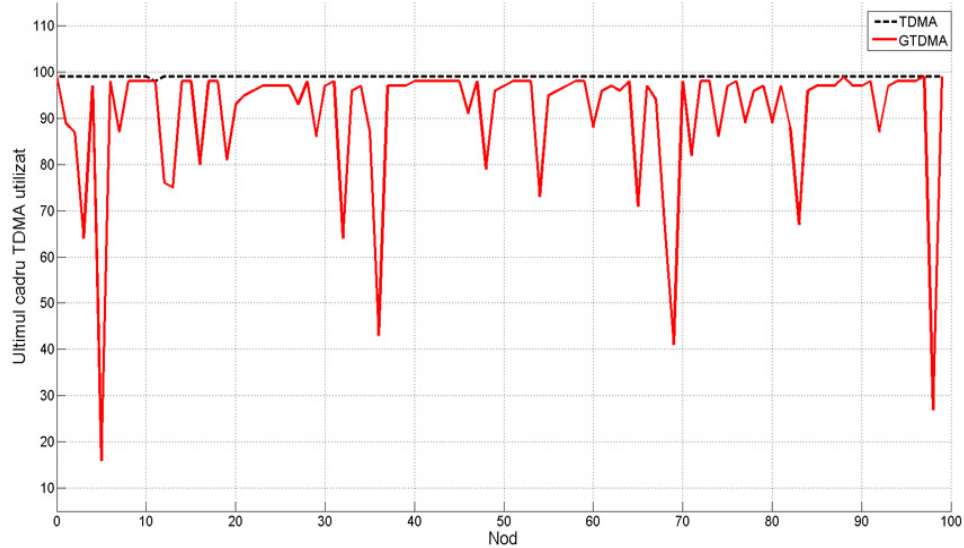


Figura 7-12 Simulare Greedy TDMA cu 100 noduri timp de 100 cadre TDMA cu distribuție gaussiană a valorilor aleatoare generate

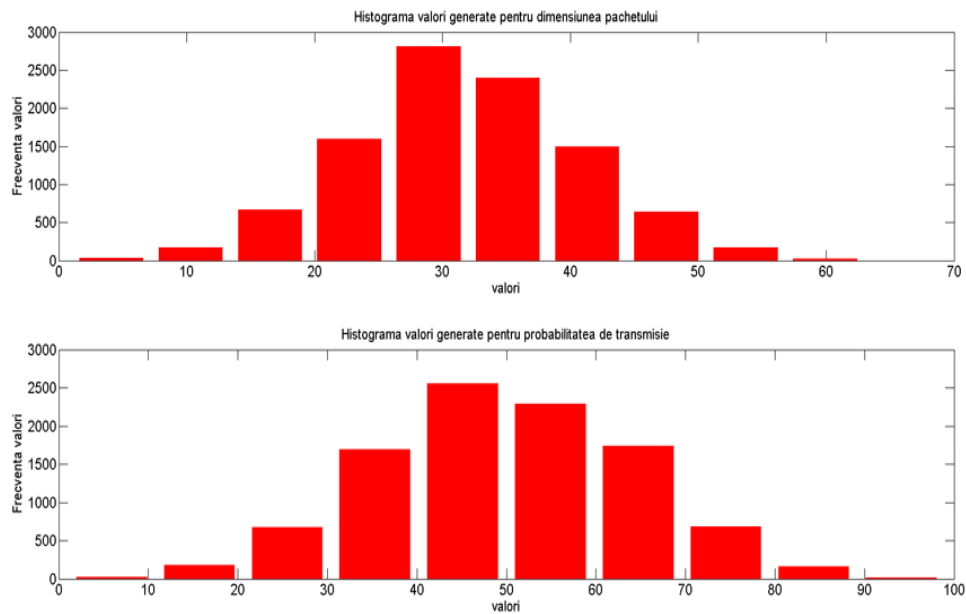


Figura 7-13 Histogramă valori generate cu distribuție gaussiană a) dimensiunea pachetului b) probabilitatea de transmisie

În același mod se poate analiza situația în care rețeaua este formată dintr-un număr de 500 de noduri menținând timpul de simulare de 100 cadre TDMA și menținând restul parametrilor de simulare din Tabel 7-4.

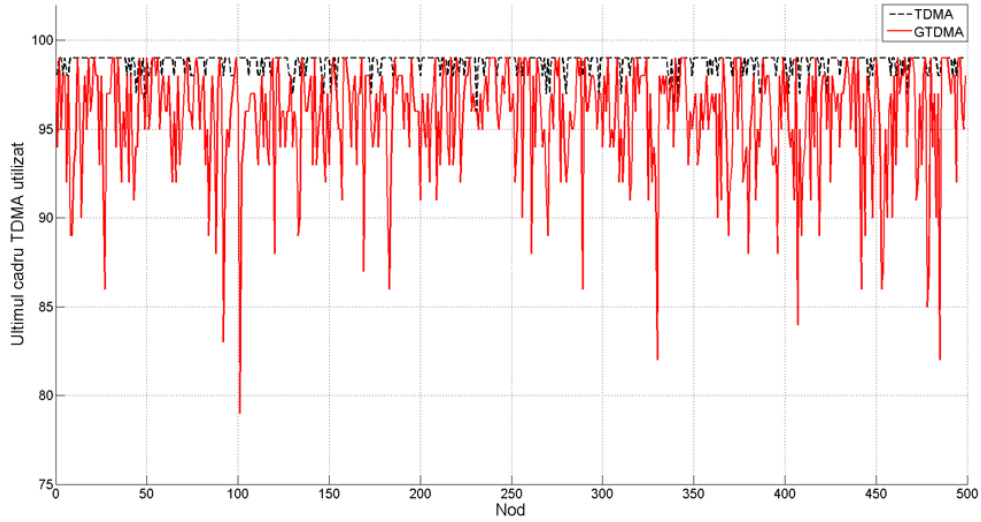


Figura 7-14 Simulare Greedy TDMA cu 500 noduri timp de 100 cadre TDMA cu distribuție uniformă a valorilor aleatoare generate

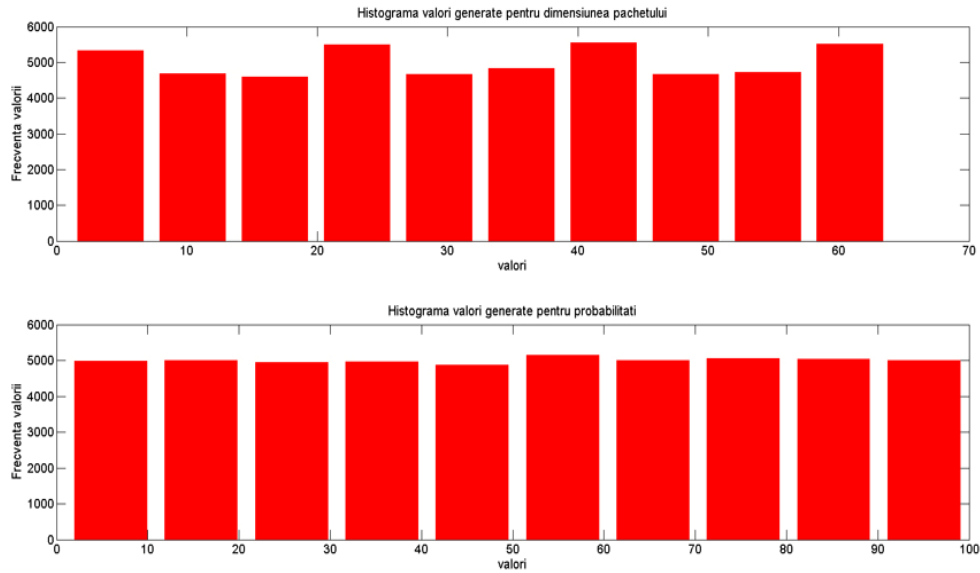


Figura 7-15 Histogramă valori generate cu distribuție uniformă pentru 500 noduri si 100 cadre TDMA a) dimensiunea pachetului b) probabilitatea de transmisie

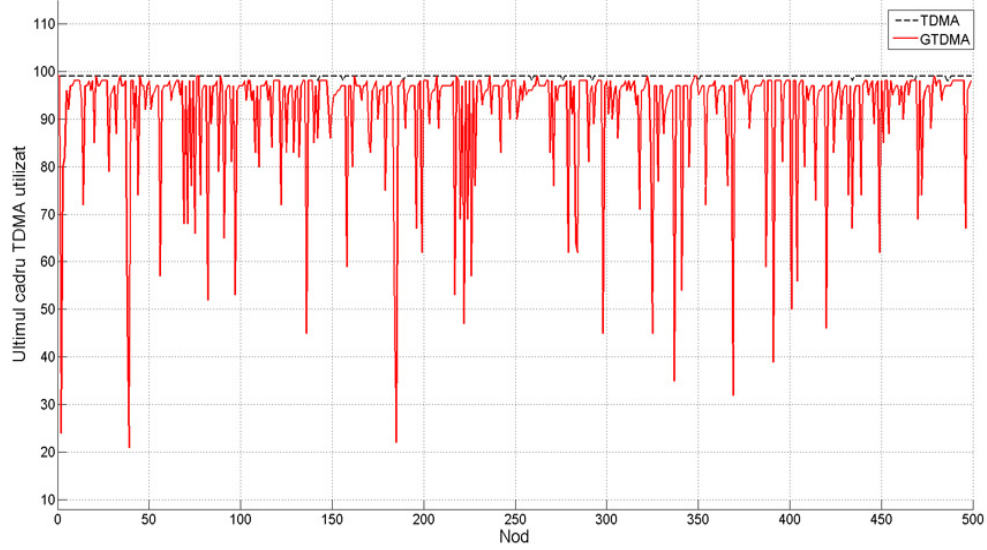


Figura 7-16 Simulare Greedy TDMA cu 500 noduri timp de 100 sloturi TDMA cu distribuție gaussiană a valorilor aleatoare generate

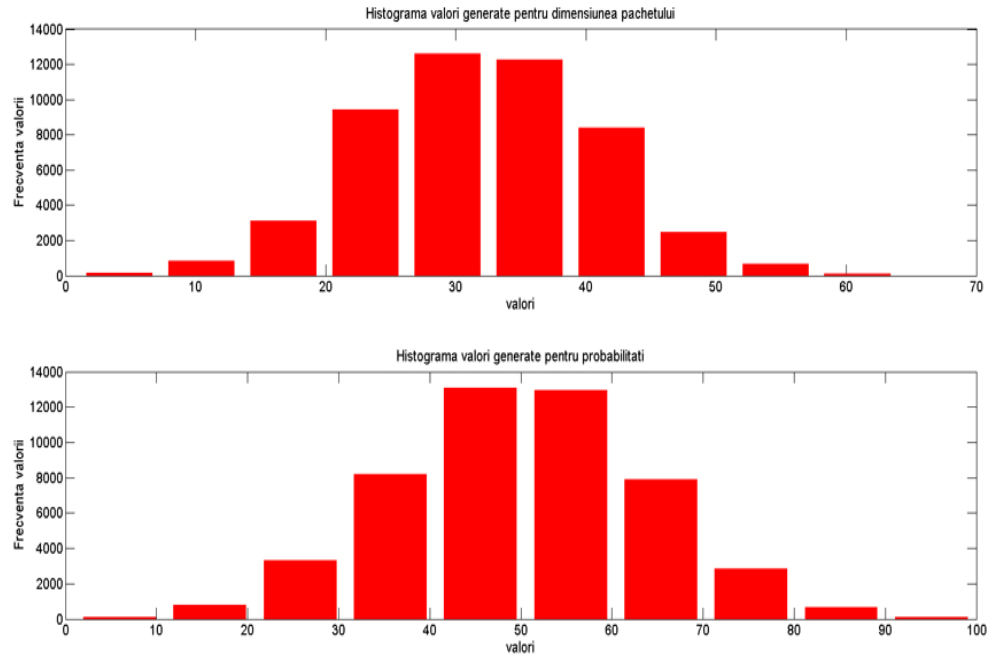


Figura 7-17 Histogramă valori generate cu distribuție gaussiană pentru 500 noduri si 100 cadre TDMA a) dimensiunea pachetului b) probabilitatea de transmisie

## 100 Greedy TDMA: Protocol timp-real eficient de acces la mediu – 7

În continuare se va prezenta aceeași situație ca și până acum dar considerând o rețea formată din 1000 noduri. Restul parametrilor de simulare au rămas aceeași.

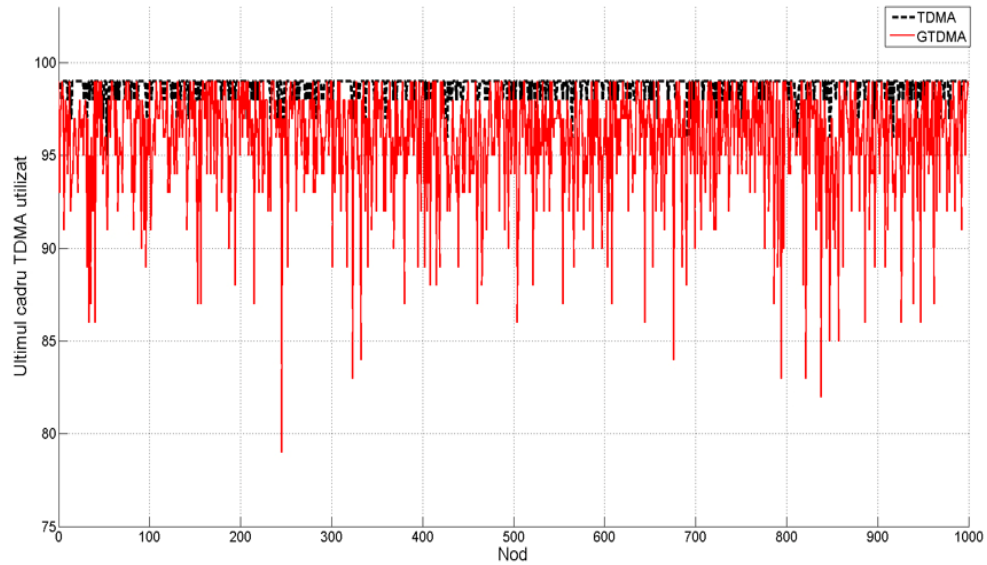


Figura 7-18 Simulare Greedy TDMA cu 1000 noduri timp de 100 cadre TDMA cu distribuție uniformă a valorilor aleatoare generate

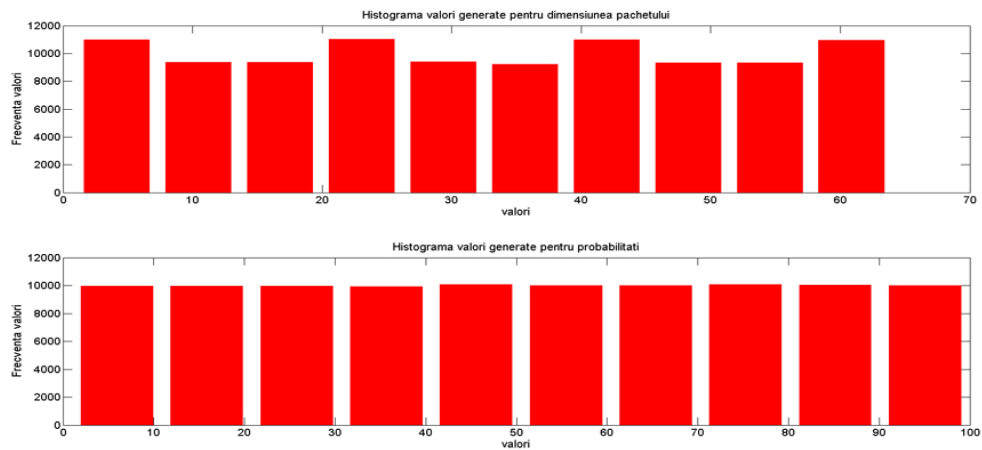


Figura 7-19 Histogramă valori generate cu distribuție uniformă pentru 1000 noduri si 100 cadre TDMA a) dimensiunea pachetului b) probabilitatea de transmisie

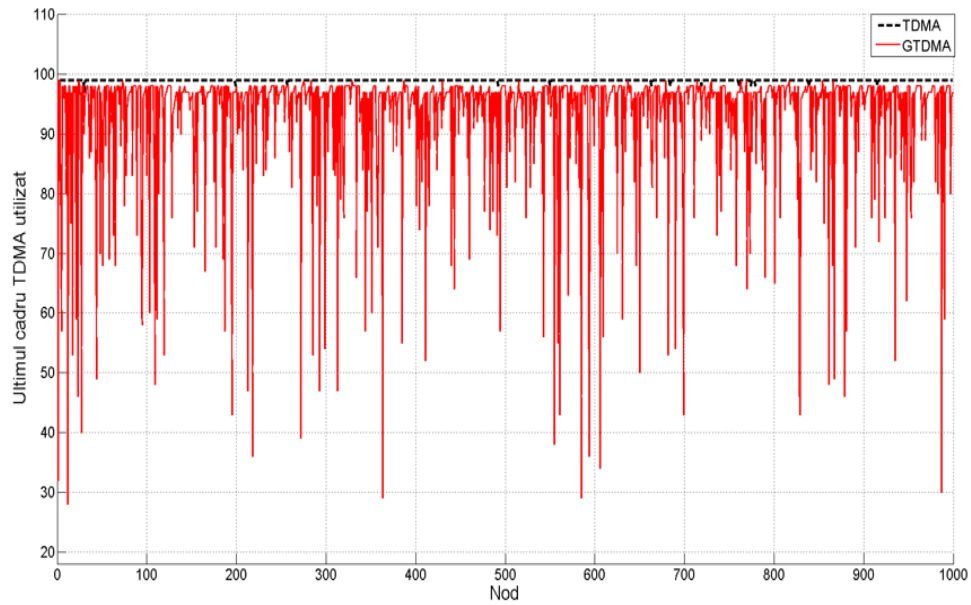


Figura 7-20 Simulare Greedy TDMA cu 1000 noduri timp de 100 sloturi TDMA cu distribuție gaussiană a valorilor aleatoare generate

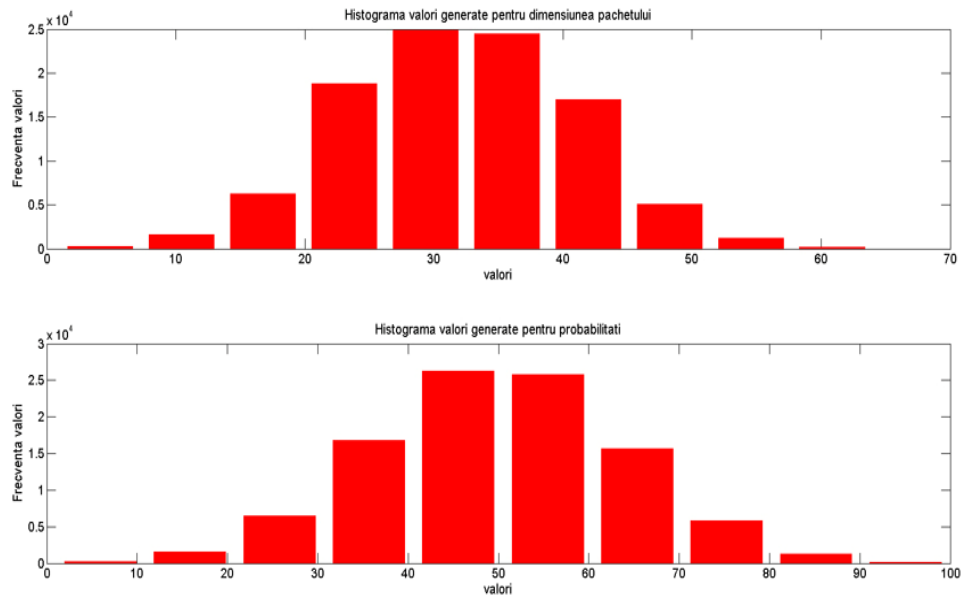


Figura 7-21 Histogramă valori generate cu distribuție gaussiană pentru 1000 noduri si 100 cadre TDMA a) dimensiunea pachetului b) probabilitatea de transmisie

### 7.3 Protocolul de acces la mediu GTDMA-MAC

În secțiunea 4.3 au fost prezentate câteva protocoale de acces la mediu, pentru rețele de senzori wireless, care pot fi folosite în medii timp real punându-se mai mult acces pe protocoale bazate pe TDMA. De asemenea se poate constata din literatură faptul că majoritatea protocoalelor de acces la mediu folosite în rețelele de senzori sunt bazate pe coliziuni fiind mult mai ușor de implementat.

Protocoalele bazate pe coliziuni nu pot fi folosite în aplicații cu cerințe de timp real din cauza impredictibilității introduse de coliziuni pe când protocoalele bazate pe TDMA oferă un suport puternic pentru timp real. În această secțiune se va propune un protocol de acces la mediu bazat pe politica de acces Greedy TDMA prezentată anterior.

Implementarea politicii de acces la mediu TDMA în cazul rețelelor de senzori wireless oferă suport de timp real dar în același timp este inefficientă dacă rețeaua este formată dintr-un număr foarte mare de noduri. După cum s-a prezentat anterior, o îmbunătățire este oferită de politica de acces Greedy TDMA. În următoarele paragrafe se vor prezenta câteva probleme ce pot apărea în cazul în care se folosește politică de acces Greedy TDMA pentru rețele de senzori wireless.

Se consideră, ca și în secțiunea anterioară, că fiecare nod al rețelei are o adresă unică pe baza căreia nodului i se atribuie un slot de timp din cadrul TDMA. Acestea sunt atribuite nodurilor pe toată perioada de funcționare a rețelei în ordinea crescătoare a adreselor.

Principala problemă în aplicarea schemei de acces la mediu GTDMA în comunicația fără fir apare în situația în care nu toate nodurile sunt în aria de acoperire a tuturor nodurilor. În acest caz nu se poate detecta corect dacă un nod folosește sau nu slotul de timp alocat.

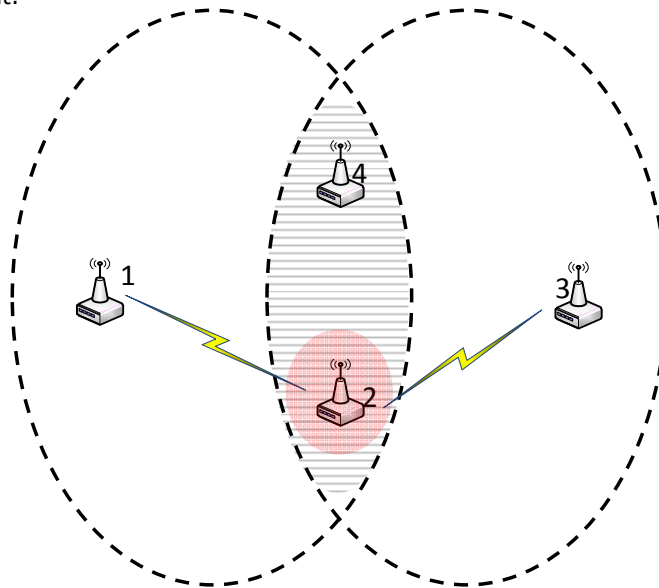


Figura 7-22 Situație de conflict

În Figura 7-22 se prezintă un exemplu de astfel de situație de conflict. Se poate observa că nodurile 2 și 4 se află atât în aria de acoperire a nodului 1 cât și a nodului 3 și astfel acestea pot recepționa toate mesajele tuturor nodurilor. În schimb nodul 1 nu se află în aria de acoperire a nodului 3 și invers. În acest caz nodul 3 nu poate recepționa mesajele nodului 1 și nici nodul 1 nu poate recepționa mesajele nodului 3. Se consideră că nodul 1 are date de transmis disponibile în slotul de timp 1 dar nodul 3 nu poate recepționa transmisia nodului 1 și conform unei scheme GTDMA, în cazul în care nu detectează o transmisie nici din partea nodului 2, nodul 3 va putea folosi slotul de timp 1 deoarece din punctul său de vedere nu există o transmisie de date în curs. În realitate se constată ca ambele noduri, nodul 1 și nodul 3, vor emite în același slot de timp iar astfel la nivelul nodului 2 și a nodului 4 se va crea conflict. În consecință mesajul transmis de nodul 1 va fi practic distrus din cauza transmisiei eronate a nodului 3.

O posibilă soluție la această problemă ar putea fi la prima vedere aplicarea unei restricții în sensul că un nod oarecare nu poate accede decât la slotul de timp al nodului adiacent când acesta nu are date disponibile de transmis în slotul său de timp. În acest sens se va analiza următoarea situație cu ajutorul Figura 7-23:

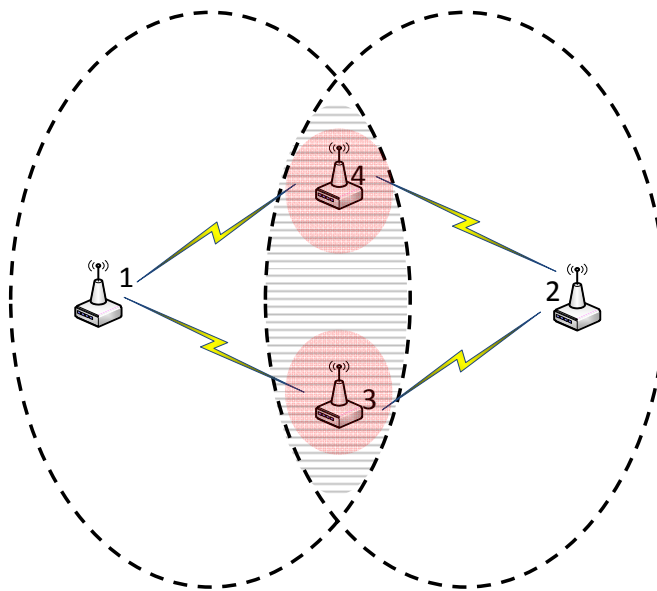


Figura 7-23 Situație de conflict

În situația de față se consideră același raționament folosind organizarea unei rețele din Figura 7-23. Nodul 1 are pregătite date pentru transmisie și își începe transmisia la începutul slotului de timp 1. În același timp, următorul nod din punct de vedere al adresei, nodul 2, nu detectează transmisia nodului 1, acesta nefiind în aria lui de acoperire. În această situație nodul 2, având date pregătite pentru transmisie, decide să folosească slotul de timp al nodului 1. Se creează din nou o situație de conflict la nivelul nodurilor 3 și 4. În slotul de timp 1 apar astfel două noduri în proces de transmisie conform Figura 7-24:

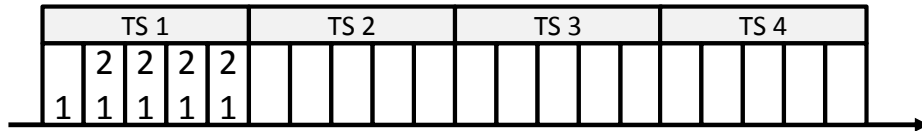


Figura 7-24 Situație de conflict în cadrul slotului de timp

Se observă în figura precedentă cum slotul de timp al nodului 1 este ocupat atât de nodul 1 cât și de nodul 2 în mod eronat. Așadar soluția prezentată anterior nu este valabilă.

O soluție finală în acest sens este aplicarea unei restricții astfel că un nod nu poate accede asupra sloturilor de timp ale unui alt nod decât dacă acesta se află atât în aria de acoperire a nodului căruia dorește să-i folosească slotul de timp cât și a nodurilor dintre cele două. Această arie de acoperire se poate stabili prin comunicații bazate pe schema de acces TDMA clasică. Astfel, un nod  $N_j$  poate folosi slotul de timp al nodului  $N_i$  ( $i < j$ ), doar dacă nodul  $N_j$  poate recepționa direct, într-o schemă TDMA clasică nu doar mesajele nodului  $N_i$  dar și mesajele nodurilor dintre cele două. În această soluție constă protocolul de acces la mediu GTDMA-MAC ce urmează a fi prezentat.

Protocolul GTDMA-MAC are la bază combinarea schemelor de acces TDMA și GTDMA astfel încât să rezolve problemele ce pot apărea în cazul comunicației fără fir, prezentate mai sus. În consecință protocolul GTDMA-MAC împarte mediul de comunicare în două sesiuni distincte: sesiunea de date (*Data Session*) și sesiunea de sincronizare (*Sync Session*). Sesiunea de date este formată dintr-un număr de cadre de timp bazate pe schema Greedy TDMA, denumite în continuare cadre GTDMA. Sesiunea de sincronizare este formată dintr-un număr de cadre de timp bazate pe schema de acces TDMA clasică, denumite în continuare cadre TDMA SYNC. În Figura 7-25 se prezintă o descriere a acestor două sesiuni de comunicare.

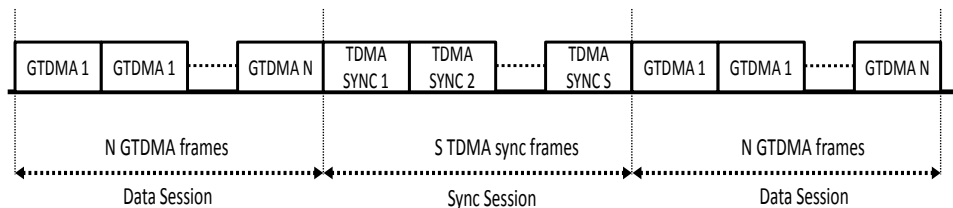


Figura 7-25 Structura GTDMA-MAC

Sesiunea de date este compusă dintr-un număr  $N$  de cadre GTDMA. Aceste cadre GTDMA vor fi folosite de către noduri după o schemă de acces Greedy TDMA prezentată anterior. Principalul scop al acestei sesiuni este transmiterea efectivă a datelor. Sesiunea de sincronizare este compusă dintr-un număr  $S$  de cadre TDMA clasice ce sunt folosite atât pentru rularea algoritmilor de sincronizare a ceasului intern al nodurilor precum și pentru ca fiecare nod al rețelei să-și refacă harta de acoperire a nodurilor din jurul său astfel încât să poată îndeplini cu succes restricția definită anterior.

Așadar principala noutate introdusă de protocolul GTDMA-MAC, pe lângă schema de acces Greedy TDMA, este sesiunea de sincronizare. În această sesiune fiecare nod este obligat să emită anumite informații și nodurile ce recepționează aceste date își vor putea realiza intern harta de acoperire. Cu ajutorul acestei hărți de acoperire nodurile



vor putea decide corect dacă pot sau nu să folosească sloturi de timp ale altor noduri prin detecția corectă a disponibilității sloturilor de timp.

Concret, într-o sesiune de comunicare din punct de vedere al transmisiei, fiecare nod este obligat să emită anumite informații de sincronizare într-o manieră TDMA clasică, adică fiecare nod în slotul de timp alocat. Din punct de vedere al recepției fiecare nod trebuie să "asculte" toate mesajele transmise de celelalte noduri și să își gestioneze o hartă internă a nodurilor pe care le-a putut recepționa corect. Astfel fiecare nod va putea aplica cu succes restricția definită mai sus și va putea decide care sloturi de timp lăsate libere de alte noduri le va putea folosi conform regulii.

Folosind structura protocolului definită în Figura 7-25 se poate reitera restricția descrisă anterior. Un nod  $N_j$  poate folosi slotul de timp al nodului  $N_i$  ( $i < j$ ), în sesiunea de date, doar dacă nodul  $N_j$  a recepționat corect, în sesiunea de sincronizare atât mesajele de sincronizare transmise de nodul  $N_i$  cât și mesajele de sincronizare ale nodurilor dintre cele două.

Sloturile de timp dintr-un cadru GTDMA sunt diferite ca și durată în timp de sloturile de timp dintr-un cadru TDMA SYNC, acestea din urmă fiind mai mici. Acest lucru este o consecință a faptului că structura pachetului de informație ce se transmite în sesiunea de date este diferită de structura pachetului de informație ce se transmite în sesiunea de sincronizare. Această situație apare din cauza faptului că în sesiunea de date pachetele de informație vor transporta date utile pentru aplicația dată, în timp ce în sesiunea de sincronizare pachetele de informație vor transporta date utile doar pentru funcționarea protocolului, acestea din urmă nefiind importante pentru aplicația în sine. Aceste pachete de informație vehiculate în sesiunea de date și sesiunea de sincronizare vor fi prezentate în următoarele paragrafe.

Pachetele transportate în cadrul sesiunii de date sunt practic pachetele utile date. Acestea au următoarea structură:

Tabel 7-5. Structura pachetului din sesiunea de date

Numărul câmpului	1	2	3	4	5	6	7
Denumirea câmpului	Packet size	Source address	Destination address	Sequence number	Pachet type	Payload	CRC
Dimensiunea câmpului [bytes]	1	2	2	2	1	n	2
Descrierea câmpului	Dimensiune a pachetului	Adresa sursa	Adresa destinație	Număr de secvență	Tipul pachetului	Datele utile	Sumă de control

În această structură se observă că împachetarea introduce 10 octeți în plus față de datele utile ceea ce se află cu mult sub numărul de octeți introduși spre exemplu de protocolul standardizat IEEE 802.15.4. Acest lucru este important în situația în care se folosesc module radio slabe în resurse care au o limită considerabilă a numărului maxim de octeți dintr-un pachet de date.

Dimensiunea pachetului descris în tabelul de mai sus este primul câmp al pachetului și cuprinde dimensiunea tuturor câmpurilor de urmează după acesta. Următoarele două câmpuri reprezintă adresa sursă, adică adresa nodului transmițător și adresa nodului destinație, a nodului receptor. Următorul câmp se referă la numărul de secvență a pachetului de date și este util pentru a se putea stabili secvențierea corectă a pachetelor de date de către protocolele superioare specializate pe acest aspect. Tipul de pachet specificat de câmpul 5 poate fi folosit de protocolele superioare

pentru a putea forma categorii de pachete. Informațiile utile sunt cuprinse în câmpul 6 și are următoarea dimensiune (notată cu  $n$ ):

$$n = P_s - 10 \quad (7-6)$$

Se poate observa din ecuația (7-6) că dimensiunea datelor utile ce pot fi transportate este dependentă de dimensiunea maximă suportată pentru un pachet de date de către modulul radio din care se scade numărul de octeți introduși de împachetare.

Întregul pachet este protejat de o sumă de control de tip CRC (Cyclic Redundancy Check). Majoritatea modulelor radio au în componența lor internă o unitate funcțională specializată pentru calcularea incrementală a unei astfel de sume de control. Este de dorit să se folosească o astfel de unitate funcțională pentru calcularea câmpului CRC decât o metodă software. Din acest punct de vedere protocolul nu specifică tipul polinomului lăsând astfel libertatea alegerii acestuia la nivelul implementatorului în funcție de capacitățile modulului radio folosit.

În cadrul sesiunii de sincronizare se vehiculează un alt tip de pachete. Aceste pachete este de dorit să fie cât mai scurte astfel încât să introducă cât mai puțin timp care nu este util comunicației. În acest mod se scurtează implicit și durata întregii sesiuni de sincronizare.

Tabel 7-6. Structura pachetului din sesiunea de sincronizare

Numărul câmpului	1	2	3	4
Denumirea câmpului	Packet size	Source address	Sync data	CRC
Dimensiunea câmpului [bytes]	1	2	s	2
Descrierea câmpului	Dimensiunea pachetului	Adresa sursa	Date de sincronizare	Sumă de control

După cum se poate observa în tabelul anterior, în pachetul de sincronizare se regăsesc ca date importante doar adresa nodului care a transmis pachetul și o serie de informații de sincronizare în general specifice aplicației. De asemenea și acest pachet este protejat de o sumă de control de tip CRC cu aceeași specificații ca și în cazul pachetului de date prezentat anterior.

Informațiile de sincronizare sunt în general dependente de protocoalele de sincronizare dar pentru buna funcționare a acestui protocol este necesar să conțină minim următoarele date:

- Numărul de pachete eronate recepționate față de ultimul cadru TDMA SYNC, practic față de ultima sesiunea de sincronizare
- Numărul de pachete valide recepționate față de ultimul cadru TDMA SYNC, practic față de ultima sesiunea de sincronizare
- Numărul de cadre GTDMA numărate față de ultima de sesiune de sincronizare

Un alt aspect important este acela legat de confirmarea mesajului (*acknowledge - ACK*). În cadrul marii majorități a protocoalelor de acces la mediu un nod ce a recepționat un mesaj de la un alt nod (*unicast*) va confirma acel mesaj printr-un scurt pachet dedicat acestei operațiuni. De asemenea această facilitate este în unele cazuri implementată în modulele radio. Astfel un modul radio, după ce a primit un pachet de

date valid adresat lui, va emite automat un pachet de tip ACK (Acknowledge). Și în cazul protocolului GTDMA-MAC este prevăzut un astfel de pachet. Acest pachet are o zonă dedicată în cuprinsul unui slot de timp GTDMA și anume înainte de sfârșitul acestuia. Acest lucru implică faptul că orice nod este obligat să încheie transmisia nu la sfârșitul slotului de timp ci la începutul zonei dedicate pentru ACK. În consecință, un slot de timp GTDMA se va modifica având următoarea structură finală descrisă în Figura 7-26.

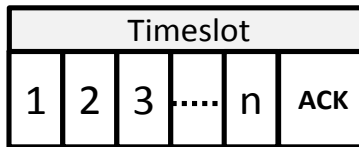


Figura 7-26 Slot de timp GTDMA cu zonă de ACK

Trebuie menționat că mecanismul de confirmare a recepției se poate utiliza doar în transmisiile de tip unicast, în care există un singur transmițător și un singur receptor adresat direct. Nu se poate folosi în general acest mecanism în cazul transmisiilor de tip broadcast unde există un singur transmițător și teoretic o infinitate de receptori. În cazul protocolului GTDMA-MAC propus acest mecanism funcționează precum este descris în Figura 7-27.

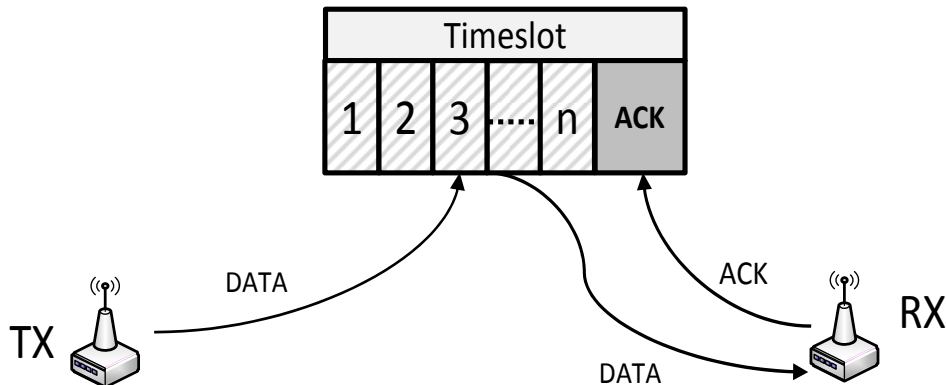


Figura 7-27 Mecanismul de ACK în GTDMA-MAC

Astfel se consideră că un nod oarecare va transmite în cadrul unui slot de timp oarecare, conform figurii de mai sus, fie că este sau nu deținătorul slotului de timp. Acest nod transmițător face astfel o transmisie de tip unicast în care este adresat direct nodul receptor. Acesta din urmă, imediat ce a primit mesajul și i-a verificat validitatea va emite un scurt pachet de date, în unitatea de timp (ACK) rezervată acestei operațiuni. Această unitate de timp este necesar să fie ultima din cadrul unui slot de timp și nu este neapărat necesar să aibă o dimensiune egală cu cea a unui TU, deși, pentru simplitate este de preferat acest lucru. Tot această unitate de timp pentru ACK încheie slotul de timp. Pachetul de date de tip ACK ce va fi transmis de nodul receptor

la sfârșitul recepției este de dorit să fie cât mai scurt și va avea astfel următoarea structură.

Tabel 7-7. Structura pachetului ACK

Numărul câmpului	1	2	3
Denumirea câmpului	Packet size	Source address	CRC
Dimensiunea câmpului [bytes]	1	2	2
Descrierea câmpului	Dimensiunea pachetului	Adresa sursa	Sumă de control

## 7.4 Concluzii

Pentru a satisface cerințele de timp impuse de o operare în timp real a unei rețele de senzori este necesar să se folosească anumite protocoale de comunicație care nu contravin principiilor unei astfel de operări. Astfel protocoalele bazate pe coliziuni nu pot fi folosite pentru a implementa o comunicație în timp real din cauza faptului că din pricina coliziunilor se pierde caracterul predictibil necesar. În acest sens singurele protocoale ce pot fi folosite în astfel de situații sunt cele care nu funcționează cu ajutorul politicilor bazate pe coliziuni. Din această categorie, după cum s-a arătat în secțiunile anterioare, fac parte, printre altele, protocoale bazate pe politica de acces TDMA. Aceste politici oferă o soluție sigură de a implementa protocoale de comunicație predictibile pentru sisteme în care este cerută operarea în timp real.

Politicile de acces TDMA au două mari dezavantaje: necesitatea sincronizării ceasurilor interne ale nodurilor rețelei și utilizarea precară a canalului de comunicație. Slaba utilizare a canalului apare în situația în care unele noduri ale rețelei nu doresc acces la canalul de comunicație și astfel slotul lor de timp asignat rămâne nefolosit. În același context există în rețea și noduri care au acumulat multe date ce trebuie transmise mai departe având astfel nevoie de multe sloturi de timp pentru a-și încheia întregul transfer de date. Fiecare nod având de obicei un singur slot asignat într-un cadrul TDMA, în cazul nodurile cu cerințe mari de comunicație datele vor fi transmise după multe iterații ale cadrului TDMA. Astfel unele noduri au nevoie mare de sloturi de timp pentru a transmite un volum mare de date pe când altele lasă sloturile de timp nefolosite. În această situație apare utilizarea slabă a canalului de comunicație.

În acest capitol s-a prezentat o soluție pentru rezolvarea celui de-al doilea dezavantaj al protocoalelor TDMA. Se propune astfel o schemă de comunicație bazată pe TDMA care îmbunătățește utilizarea canalelor radio. Astfel sloturile de timp nefolosite de către anumite noduri sunt preluate după anumite reguli de alte noduri ale rețelei ce au nevoie de a folosi mai multe sloturi de timp pentru a transfera un volum mare de date. Această îmbunătățire nu afectează predictibilitatea oferită de schema TDMA clasică. Schema nou introdusă, Greedy TDMA, este o schemă de comunicație cu reguli generale ce poate fi aplicată pe orice mediu de comunicație. Politica nou

introdusă este validată prin medii de simulare și se arată cum aceasta este superioară clasicei scheme TDMA în termeni de utilizare a mediului de comunicație.

Într-un mediu de comunicație cu fir, aplicarea acestei scheme nu prezintă probleme majore.

Aplicarea schemei de comunicație Greedy TDMA în mediul wireless presupune adăugarea unor mecanisme menite să adapteze schema de comunicație cerințelor canalului de comunicație. Se propune astfel protocolul de acces la mediu GTDMA-MAC care definește atât algoritmul de comunicație cât și împachetarea datelor. Prin împachetarea datelor protocolul nu introduce multă informație redundantă pe lângă datele utile fiind astfel superior din acest punct de vedere standardului IEEE 802.15.4. De asemenea menține anumite mecanisme definite de standard care pot fi folosite cu succes într-un mediu timp real.

Un alt aspect important de menționat este acela că protocolul GTDMA-MAC fiind bazat pe o schemă ce are la bază politica TDMA nu mai are nevoie de operațiuni adiționale de acces la mediul de comunicație, cum este operațiunea CCA descrisă în secțiunile anterioare. În capitolul 5 se arată cum această procedură CCA introduce o componentă de timp ce afectează predictibilitatea sistemului,  $T_{CCA}$ . În cazul folosirii protocolului GTDMA-MAC, această componentă este eliminată deoarece întregă procedură CCA nu mai este necesară.

## 8 Concluzii și perspective

### 8.1 Concluzii

În ultima perioadă rețelele de senzori au început să fie folosite tot mai mult în toate domeniile și astfel problemele legate de acestea sunt intens studiate de către cercetătorii din întreaga lume. Una dintre principalele probleme apărute în cadrul rețelelor de senzori este legată de comunicarea dintre nodurile rețelei și în particular comunicarea fără fir. Aceste rețele de senzori sunt formate din sisteme încorporate de mici dimensiuni, slabe în resurse, cu un cost scăzut care au capacități de comunicare radio.

Multe dintre aceste rețele de senzori pot fi folosite cu succes în aplicații critice. În acest tip de aplicații nu doar nodurile rețelei este necesar să funcționeze în timp real dar și întreaga rețea, ca tot unitar trebuie să funcționeze în aceeași manieră, de timp real. Pentru a se putea realiza acest lucru stiva de protocoale de comunicație trebuie să funcționeze după principiile operării în timp real a unui sistem. Referindu-ne la această stivă de protocoale de comunicare, toate nivelele acestei stive trebuie să funcționeze în timp real și în special nivelele inferioare precum nivelul de acces la mediu (format în general de nivelul fizic și nivelul data-link).

În vederea realizării unui protocol de acces la mediu, în timp real, trebuie ca dezvoltatorul acestui protocol de țină cont de timpii de transmitere a unui pachet de date de la un nod sursă la un nod destinație precum și de stările interne ale modulelor radio și a timpilor introduși de fiecare operațiune în parte. Mai mult decât atât este necesar ca protocolul de acces la mediu să trateze corect diferitele componente timp. Este de asemenea important să se stabilească regimul fiecărei componente de timp: dacă este calculabilă sau nu, dacă este predictibilă sau nu (și în acest caz să se stabilească un mod de aproximare), dacă este dată în foaia de catalog a modulului radio și de asemenea dacă se respectă valorile de timp date în foia de catalog a modulului radio.

În acest context s-a propus un model de a calcula componentele de timp implicate atât în transferul pachetelor de date între module cât și a componentelor de timp ce pot influența sistemul fiecărui nod în parte, componente ce se referă la tranziția stărilor interne a modulului radio folosit. De asemenea s-a dezvoltat și propus un sistem hardware/software de măsurare și analiză a acestor componente de timp.

Protocoalele de acces ce pot fi aplicate cel mai ușor pentru a putea obține o comunicație în timp real sunt protocoalele bazate pe divizarea timpului, mai exact protocoalele bazate pe schema de acces TDMA. Dezavantajele acestei scheme de acces au făcut ca acest tip de protocoale să fie puțin folosite în cadrul rețelelor de senzori.

Totuși, există soluții pentru ca acestea să fie folosite. Principalele modificări ce trebuie făcute asupra protocoalelor bazate pe TDMA se referă la îmbunătățirea utilizării mediului de comunicație dar cu păstrarea predictibilității. O astfel de soluție poate fi și tehnica slot-stealing prezentă în literatură dar numai în cazuri particulare. O generalizare a acestei tehnici de acces a dus la formarea politicii GTDMA. Această politică de acces are la bază ideea de a folosi sloturile de timp ale unor noduri care nu au date pregătite pentru a fi transmise de către nodurile care au acumulat un volum mare de date ce trebuie transmise. Practic acestea din urmă vor folosi atât slotul lor de timp asignat inițial cât și sloturile de timp ale nodurilor ce nu au nevoie de mediul de comunicație.

Aplicarea schemei de acces GTDMA presupune stabilirea unor restricții în comunicație radio pentru rețelele de senzori pentru a se putea păstra predictibilitatea și caracterul de timp real oferit de schema clasică TDMA. Aceste restricții au dus la formarea protocolului de acces la mediu GTDMA-MAC.

## 8.2 Sinteza contribuțiilor

Principalele obiective propuse pentru activitatea de cercetare doctorală sunt următoarele:

[O1]. *Efectuarea unui studiu amănunțit al problemelor ce apar pentru a realiza un model de protocol MAC de comunicare în timp real. Acest studiu presupune o amplă documentare în legătură cu protocoalele de comunicare existente și stabilirea limitărilor acestora în contextul de timp real.*

Îndeplinirea acestui obiectiv împreună cu sub-obiectivele sale este evidențiată de următoarele contribuții:

- Model temporal pentru comunicația wireless pentru a analiza penalitățile de timp introduse de modulele radio precum și pentru a analiza comportamentul temporal al acestora
- Cadru de măsurare a parametrilor de timp ce compun întârzierea totală în cadrul transmisiei și recepției datelor între modulele radio
- Noduri ale unei rețele de senzori folosind mai multe module radio existente pe piață
- Analiză experimentale a comportamentului temporal a unor module radio existente pe piață
- Identificarea componentelor de timp ce afectează predictibilitatea temporală a comunicației

Aceste contribuții s-au materializat în publicațiile [A9] și [A8]. De asemenea, acest obiectiv propus a fost îndeplinit și prin realizarea capitolelor teoretice 2, 3 și 4 a acestei lucrări.

[O2]. *Realizarea unui protocol de timp MAC care să fie orientat atât pe menținerea unui consum redus de energie electrică la nivelul unui nod cât și la nivelul întregii rețele, cu facilități de comunicare în timp real.*

Îndeplinirea acestui obiectiv este evidențiată de următoarele contribuții:

- Realizarea unei scheme de comunicație bazată pe schema TDMA ce îmbunătățește principalul dezavantaj al TDMA: slaba ocupare a mediului de comunicație
- Simulator în mediu discret pentru a valida și evalua experimental schema de acces Greedy TDMA propusă
- Adaptarea schemei de acces Greedy TDMA pentru comunicația wireless în cadrul unei rețele de senzori rezultând protocolul de acces la mediu GTDMA-MAC

Aceste contribuții s-au materializat în [A7].

[O3]. *Realizarea practică a unui dispozitiv/modul/sistem de comunicare în timp real care să ofere facilități de comunicare hard real time, destinat amplasării acestuia pe nodurile unei rețele de senzori.*

Acest obiectiv s-a îndeplinit prin următoarele contribuții ce s-au materializat prin [A9]și [A8]:

- Model temporal pentru comunicația wireless pentru a analiza penalitățile de timp introduse de modulele radio precum și pentru a analiza comportamentul temporal al acestora
- Noduri ale unei rețele de senzori folosind mai multe module radio existente pe piață
- Analiză experimentală a comportamentului temporal a unor module radio existente pe piață

Aceste contribuții au dus la îndeplinirea obiectivelor propuse pentru activitatea de cercetare. Pe lângă acestea s-a mai realizat și o analiză experimentală a celei mai utilizate stive de protocoale pentru rețelele de senzori, și anumite stivă ZigBee și s-a identificat și experimental principalele probleme ce fac ca această stivă de comunicație să nu poată fi folosită într-un mediu timp real. Această analiză s-a materializat prin [A5].

### **8.3 Perspective de dezvoltare**

Prima propunere de dezvoltare ulterioară ar fi implementarea protocolului GTDMA-MAC ce a fost propus pe platforme hardware existente și evaluarea lui și în mediul real.

De asemenea se dorește integrarea acestui protocol de comunicație în platforme CORE-TX și înlocuirea sistemului de comunicație existent bazat pe stiva ZigBee. Pentru a se putea realiza acest lucru este necesar ca protocolului GTDMA-MAC să i se adauge nivelele superioare de comunicație necesare, cum ar fi protocoale pentru detecția și corecția erorilor, planificarea retransmisiilor, protocoale de rutare a informației. Prin aceasta se dorește crearea unei stive de comunicație pentru rețele de senzori în timp real.



## Referințe bibliografice

- [1] J. Daintith and E. Wright, *A Dictionary of Computing*. USA: Oxford University Press, 2004, ISBN: 978-0-19-923400-4.
- [2] J. A. Stankovic, K. Ramamritham, M. Spuri, and G. C. Buttazzo, *Deadline Scheduling for Real-Time Systems: EDF and Related Algorithms*. MA, USA: Kluwer Academic Publishers Norwell, 1998, ISBN: 079238.
- [3] G. Butazzo and P. Gai, "Efficient EDF Implementation for Small Embedded Systems," in *Workshop on Operating Systems Platforms for Embedded Real-Time applications OSPERT*, Dresden, Germany, 2006.
- [4] T. Lev-Ami and S. S. Tyszberowicz, "Reactive and Real-Time Systems Course: How to Get the Most Out of it," *Real-Time Syst.*, vol. 25, no. 2-3, pp. 231-253, 2003.
- [5] G. C. Butazzo, *Hard Real-Time Computing Systems: Predictable Scheduling Algorithms and Applications (Real-Time Systems Series)*. Pavia, Italy: Springer, 2005, ISBN: 978-0387231372
- [6] M. V. Micea, "Proiectarea si implementarea sistemelor timp-real pentru aplicatii critice de achizitie si prelucrare numerica de semnal," PhD, Politehnica Timisoara, 2004.
- [7] V. Stangaciu, G. Macariu, M. V. Micea, V. Muresan, and B. Barry, "Requirements of a Real-Time Multiprocessor Operating System for Multimedia Applications," in *6th IEEE International Symposium on Applied Computational Intelligence and Informatics (SACI)*, Timisoara, Romania, 2011.
- [8] J. W. S. Liu, *Real-Time Systems* 1ed. New Jersey, USA: Prentice Hall, 2000, ISBN: 978-0130996510.
- [9] OSEK group, "OSEX/VDX Operating System, Specification version 2.2.3," 2005.
- [10] Wind River Systems Inc., "VxWorks Programmer's Guide, 5.5.," Alameda, CA, SUA 2002.
- [11] I. Ripoll, P. Pisa, L. Abeni, P. Gai, and A. Lanusse, "RTOS State of the Art Analysis, Deliverable D1.1 – RTOS Analysis," OCERA 2002.
- [12] M. A. Rivas and M. G. Harbour, "MaRTE OS: Overview and Linux Version," Talk for the Real-Time Systems Group, York Nov 2004.
- [13] M. A. Rivas and M. G. Harbour, "POSIX-Compatible Application-Defined Scheduling in MaRTE OS," in *14th Euromicro Conference on Real-Time Systems*, Vienna, Austria, 2002, pp. 67-75.
- [14] J. Mache, E. Dean, and K. Imber, "Sensor Network Lab Exercises Using TinyOS and MicaZ Motes, Part I," in *International Conference on Wireless Networks ICWN*, Monte Carlo Resort, Las Vegas, Nevada, USA, 2007, pp. 464-467.
- [15] C. S. Stangaciu, M. V. Micea, and V. I. Cretu, "Hard real-time execution environment extension for FreeRTOS," in *Robotic and Sensors*

- Environments (ROSE), 2014 IEEE International Symposium on*, 2014, pp. 124-129.
- [16] R. Inam, J. Maki-Turja, M. Sjodin, S. M. H. Ashjaei, and S. Afshar, "Support for hierarchical scheduling in FreeRTOS," in *Emerging Technologies & Factory Automation (ETFA), 2011 IEEE 16th Conference on*, 2011, pp. 1-10.
- [17] K. Andersson and R. Andersson, "A comparison between FreeRTOS and RTLinux in embedded real-time systems," Linköping University 2005.
- [18] R. Barry, *Using the FreeRTOS real time kernel: a practical guide*: Real Time Engineers Limited, 2009, ISBN: 978-1446169971.
- [19] M. V. Micea, C. Certejan, V. Stangaciu, R. Cioarga, V. Cretu, and E. Petriu, "Inter-Task Communication and Synchronization in the Hard Real-Time Compact Kernel HARETICK," in *IEEE International Workshop on Robotic and Sensors Environments*, Ottawa, Canada, 2008.
- [20] M. V. Micea, V. Cretu, and V. Groza, "Maximum Predictability in Signal Interactions With HARETICK Kernel," *IEEE Transactions on Instrumentation and Measurement*, vol. 55, no. 4, pp. 1317 - 1330, August 2006.
- [21] C. Certejan, "Mecanisme de timp real si planificare pentru HARETICK," BSc, Politehnica Timisoara, 2009.
- [22] C. Certejan, "MEDF-FENP Real Time Scheduling Algorithm," MSc., Politehnica Timisoara, 2011.
- [23] Motorola Inc., "DSP56307EVM User's Manual Rev. 3/1999, DSP56307EVMUM/D," Semiconductor Products Sector, DSP Division, Austin, USA 1999.
- [24] Motorola Inc., "MSC8101 Reference Manual: 16-Bit Digital Signal Processor," MSC8101RM/D Rev. 1 June 2001.
- [25] NXP Semiconductors, "LPC2119/2129/2194/2292/2294 User Manual," Koninklijke Philips Electronics N.V. May 2004.
- [26] T. Martin, *The Insider's Guide To The Philips ARM7-Based Microcontrollers, An Engineer's Introduction to the LPC2100 Series*. UK: Hitex Ltd., 2007, ISBN: 0-9549988 6.
- [27] V. Stangaciu, "Nucleul HARETICK de pe platforma CORE-TX si operarea acestuia in aplicatii de tip cladire inteligenta," BSc., Politehnica Timisoara, 2009.
- [28] P. Puschner, "Hard Real-Time Programming is Different," in *Proceedings of the 17th International Symposium on Parallel and Distributed Processing*, 2003, p. 116.2.
- [29] M. V. Micea, G. N. Carstoiu, L. Ungurean, D. Chiciudean, V. Cretu, and V. Groza, "Predictable Data Communication Interface for Hard Real-Time Systems," presented at the IEEE International Workshop on Robotic and Sensors Environments (ROSE), Ottawa, Canada, 2008.
- [30] M. V. Micea, V. Stangaciu, C. Stangaciu, and C. Filote, "Sensor-Level Real-Time Support for XBee-Based Wireless Communication," in *Advances in Intelligent and Soft Computing (AISC)*. vol. 145, F. L. Gaol and Q. V. Nguyen, Eds., ed Berlin, Germany: Springer Berlin Heidelberg, 2012, ISBN: 978-953-307-842-7, pp. 147-154.

- 
- [31] Digi International, "XBeeTM/XBee-PROTM OEM RF Modules. Product manual v1.xAx - 802.15.4 protocol," Digi International Inc. 2007.
  - [32] ZigBee Alliance, "ZigBee Specifications," version 1.0 r13 Dec 2006.
  - [33] L. B. Ruiz, J. M. Nogueira, and A. A. F. Loureiro, "Sensor Network Management," in *Handbook of Sensor Networks: Compact Wireless and Wired Sensing Systems*, M. Ilyas and I. Mahgoub, Eds., ed Boca Raton, Florida: CRC Press, 2005, ISBN: 0-8493-1968-4, pp. 57-84.
  - [34] G. Sudha Anil Kumar, G. Manimaran, and Z. Wang, "Energy-Aware Scheduling of Real-Time Tasks in Wireless Networked Embedded Systems," in *Real-Time Systems Symposium (RTSS). 28th IEEE International*, Tucson, AZ, 2007, pp. 15-24.
  - [35] G. Sudha Anil Kumar, G. Manimaran, and Z. Wang, "Energy-aware scheduling with deadline and reliability constraints in wireless networks," in *Broadband Communications, Networks and Systems (BROADNETS). Fourth International Conference on*, Raleigh, NC, USA, 2007, pp. 95-105.
  - [36] Q. Wang, H. Hassanein, and K. Xu, "A Practical Perspective on Wireless Sensor Networks," in *Handbook of Sensor Networks: Compact Wireless and Wired Sensing Systems*, M. Ilyas and I. Mahgoub, Eds., ed Boca Raton, Florida: CRC Press, 2005, ISBN: 0-8493-1968-4, pp. 167-194.
  - [37] B. S. Manoj and R. R. Rao, "Wireless Mesh Networks: Issues And Solutions," in *Wireless Mesh Networking - Architectures, Protocols and Standards*, Y. Zhang, J. Luo, and H. Hu, Eds., ed Boca Raton, New York, USA: Auerbach Publications, Taylor & Francis Group, 2006, ISBN: 978-0-8493-7399-2, pp. 4-46.
  - [38] I. F. Akyildiz, X. Wang, and W. Wang, "Wireless mesh networks: a survey," *Comput. Netw. ISDN Syst.*, vol. 47, no. 4, pp. 445-487, 2005.
  - [39] H. Zimmermann, "OSI Reference Model - The ISO Model of Architecture for Open Systems Interconnection," *IEEE Transactions on Communications*, vol. COM-28, no. 4, pp. 425-432, April 1980.
  - [40] Electronic Industries Association. Engineering Department, "Interface between data terminal equipment and data communication equipment employing serial binary data interchange," ed. Washington: Electronic Industries Association, Engineering Dept., 1969.
  - [41] International Organization for Standardization, "Information technology. Telecommunications and information exchange between systems. High-level data link control (HDLC) procedures. Elements of procedures 13239:2002," ed: ISO/IEC, 2002.
  - [42] IEEE Computer Society, "Carrier sense multiple access with Collision Detection (CSMA/CD) Access Method and Physical Layer Specifications," in *802.3*, ed: Institute of Electrical and Electronics Engineers, Inc., 2008.
  - [43] J. Yi, C. Poellabauer, X. S. Hu, and L. Zhang, "Minimum Bandwidth Reservations for Periodic Streams in Wireless Real-Time Systems," *IEEE Transactions on Mobile Computing*, vol. 10, no. 4, pp. 479-490, Apr 2011.
  - [44] M. V. Micea, G. N. Carstoiu, L. Ungurean, and D. Chiciudean, "PARSECS: A Predictable Data Communication System for Smart

- Sensors and Hard Real-Time Applications," *IEEE Transactions on Instrumentation and Measurement*, vol. 59, no. 11, pp. 2968-2981, November 2010.
- [45] E. Uhlemann, L. K. Rasmussen, and P.-A. Wiberg, "Wireless real-time communication using deadline dependent coding," in *ARTES - A Network for Real-Time Research and Graduate Education in Sweden*, H. Hansson, Ed., 1 ed Sweden Uppsala University, 2006, ISBN: 91-506-1859-8, pp. 397-415.
- [46] B. Statulat, S. Barzeianu, M. V. Micea, and V. Groza, "Wireless Synchronization Protocols for Collaborative Robotic and Sensor Environments," in *19th International Workshop on Robotics in Alpe-Adria-Danube Region (RAAD)*, Budapest, Hungary, 2010.
- [47] K. K. Z. Teng, "A Survey on Real-Time MAC Protocols in Wireless Sensor Networks," *Communications and Network*, vol. 2, no. 2, pp. 104-112, 2010.
- [48] A. Ali, L. A. Latiff, M. A. Sarijari, and N. Fisal, "Real-time Routing in Wireless Sensor Networks," in *28th International Conference on Distributed Computing Systems Workshops ICDCS*, 2008, pp. 114-119.
- [49] E. Uhlemann, "Adaptive Concatenated Coding for Wireless Real-Time Communications," PhD Thesis, School of Information Science, Computer and Electrical Engineering, Chalmers University of Technology, Goteborg, Sweden, 2004.
- [50] J. V. Capella, A. Bonastre, J. J. Serrano, and R. Ors, "A pollution monitoring system based on an energy efficient Wireless Sensor Networks architecture," in *17th Telecommunications forum TELFOR*, Serbia, Belgrade, 2009, pp. 1145-1148.
- [51] J. V. Capella, A. Bonastre, J. J. Serrano, and R. Ors, "A new robust, energy-efficient and scalable wireless sensor networks architecture applied to a wireless fire detection system," in *International Conference on Wireless Networks and Information Systems WNIS*, 2009, pp. 395-398.
- [52] M. Marcu, C. Stangaciu, A. Topirceanu, D. Volcinschi, and V. Stangaciu, "Wireless Sensors Solution for Energy Monitoring, Analyzing, Controlling and Predicting Sensor Systems and Software." vol. 57, G. Par and P. Morrow, Eds., ed: Springer Berlin Heidelberg, 2011, ISBN: 978-3-642-23583-2, pp. 1-19.
- [53] B. S. Popescu, A. Stancovici, V. Stangaciu, C. Certejan, and M. Marcu, "Intelligent wireless distributed network for power consumption monitoring and analysis," in *Science and Technology for Humanity, IEEE Toronto International Conference (TICSTH)*, Toronto, ON, Canada, 2009, pp. 730-735.
- [54] P. N. N. Reddy, P. I. Basarkod, and S. S. Manvi, "Wireless Sensor Network based Fire Monitoring and Extinguishing System in Real Time Environment," *International Journal Advanced Networking And Application*, vol. 3, no. 2, pp. 1070-1075, 2011.
- [55] B. Ciubotaru, R. Cioarga, D. Chiciudean, M. V. Micea, and M. Stratulat, "Connectivity Improvement in Wireless Sensor Networks Based on

- Mobile Nodes," in *Instrumentation and Measurement Technology Conference (IMTC)*, Warsaw, Poland, 2007.
- [56] M. V. Micea, "Sisteme timp-real incorporate în aplicații complexe de percepție artificială distribuită, medii colaborative robotizate și rețele de senzori inteligenți," Raport final Grant CEEEX Cod ET-07, 2006-2008.
- [57] W. Su, E. Cayirci, and Ö. B. Akan, "Overview of Communication Protocols for Sensor Networks," in *Handbook of Sensor Networks: Compact Wireless and Wired Sensing Systems*, M. Ilyas and I. Mahgoub, Eds., ed Boca Raton, Florida: CRC Press, 2005, ISBN: 0-8493-1968-4, pp. 314-329.
- [58] J. A. Stankovic, T. F. Abdelzaher, C. Lu, L. Sha, and J. C. Hou, "Real Time Communication and Coordination in Embedded Sensor Networks," *Proceedings of the IEEE*, vol. 91, no. 7, pp. 1002-1022, July 2003.
- [59] J. I. Goodman, A. I. Reuther, and D. R. Martinez, "Next-Generation Technologies to Enable Sensor Networks," in *Handbook of Sensor Networks: Compact Wireless and Wired Sensing Systems*, M. Ilyas and I. Mahgoub, Eds., ed Boca Raton, Florida: CRC Press, ISBN: 0-8493-1968-4, pp. 36-57.
- [60] Li-Ming He, "A Novel Real-Time Routing Protocol for Wireless Sensor Networks," in *10th ACIS International Conference on Software Engineering, Artificial Intelligences, Networking and Parallel/Distributed Computing SNPD*, 2009, pp. 411-416.
- [61] J. N. Al-Karaki and A. E. Kamal, "A Taxonomy of Routing Techniques in Wireless Sensor Networks," in *Handbook of Sensor Networks: Compact Wireless and Wired Sensing Systems*, M. Ilyas and I. Mahgoub, Eds., ed Boca Raton, Florida: CRC Press, 2005, ISBN: 0-8493-1968-4, pp. 117-140.
- [62] W. R. Heinzelman, A. Chandrakasan, and H. Balakrishnan, "Energy-efficient communication protocol for wireless microsensor networks," in *System Sciences, 2000. Proceedings of the 33rd Annual Hawaii International Conference on*, 2000.
- [63] Y. Xu, J. Heidemann, and D. Estrin, "Geography-informed energy conservation for Ad Hoc routing," in *Proceedings of the 7th annual international conference on Mobile computing and networking*, Rome, Italy, 2001, pp. 70-84.
- [64] B. Chen, K. Jamieson, H. Balakrishnan, and R. Morris, "Span: an energy-efficient coordination algorithm for topology maintenance in ad hoc wireless networks," *Wirel. Netw.*, vol. 8, no. 5, pp. 481-494, September 2002.
- [65] IEEE Computer Society, "Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications," in *802.11*, ed: Institute of Electrical and Electronics Engineers, Inc., 2007.
- [66] J. S. T. He, C. Lu and T. Abdelzaher, "SPEED: A Real-Time Routing Protocol for Sensor Networks," University of Virginia, Tech. Report CS-2002-09, March 2002.
- [67] O. Chipara, Z. He, G. Xing, Q. Chen, X. Wang, C. Lu, J. Stankovic, and T. Abdelzaher, "Real-time Power-Aware Routing in Sensor Networks,"

- in *Quality of Service, 2006. IWQoS 2006. 14th IEEE International Workshop on*, 2006, pp. 83-92.
- [68] A. A. Minhas, S. Jabbar, R. A. Akhtar, and W. Mehmood, "Query-based Energy-aware Protocol for Real-time Routing in Wireless Sensor Network," in *IEEE 10th International Conference on Computer and Information Technology CIT*, 2010, pp. 2861-2866.
- [69] G. S. A. Kumar, G. Manimaran, and Z. Wang, "End-to-End Energy Management in Networked Real-Time Embedded Systems," *Parallel and Distributed Systems, IEEE Transactions on*, vol. 19, no. 11, pp. 1498-1510, Nov 2008.
- [70] F. J. Atero, J. J. Vinagre, E. Morgado, and M. R. Wilby, "A Low Energy and Adaptive Architecture for Efficient Routing and Robust Mobility Management in Wireless Sensor Networks," in *31st International Conference on Distributed Computing Systems Workshops ICDCSW*, Minneapolis, Minnesota USA, 2011, pp. 172-181.
- [71] Y. Sun, O. Gurewitz, S. Du, L. Tang, and D. B. Johnson, "ADB: an efficient multihop broadcast protocol based on asynchronous duty-cycling in wireless sensor networks," presented at the Proceedings of the 7th ACM Conference on Embedded Networked Sensor Systems, Berkeley, California, 2009.
- [72] R. J. Punnoose, P. V. Nikitin, J. Broch, and D. D. Stancil, "Optimizing Wireless Network Protocols Using Real-Time Predictive Propagation Modeling," in *IEEE Radio and Wireless Conference RAWCON*, 1999, pp. 33-44.
- [73] S. Han, X. Zhu, A. K. Mok, D. Chen, and M. Nixon, "Reliable and Real-time Communication in Industrial Wireless Mesh Networks," in *17th IEEE Real-Time and Embedded Technology and Applications Symposium RTAS*, 2011, pp. 3-12.
- [74] D. Dewasurendra and A. Mishra, "Design Challenges in Energy-Efficient Medium Access Control for Wireless Sensor Networks," in *Handbook of Sensor Networks: Compact Wireless and Wired Sensing Systems*, M. Ilyas and I. Mahgoub, Eds., ed Boca Raton, Florida: CRC Press, 2005, ISBN: 0-8493-1968-4, pp. 314-329.
- [75] P. Brenner, "A Technical Tutorial on the IEEE 802.11 Protocol," Breezecom Wireless Communications July 1996.
- [76] C. R. Berger, V. Arbatov, Y. Voronenko, F. Franchetti, and M. Puschel, "Real-time software implementation of an IEEE 802.11a baseband receiver on Intel multicore," in *IEEE International Conference on Acoustics, Speech and Signal Processing ICASSP*, 2011, pp. 1693-1696.
- [77] W. Zhao and K. Ramamritham, "A Virtual Time CSMA Protocol for Hard Real Time Communication," *IEEE Transactions on Software Engineering*, vol. 13, no. 8, pp. 938-952, Aug 1987.
- [78] M. J. Markowski and A. S. Sethi, "Real-Time Wireless Communication Using Splitting Protocols," in *IEEE Global Telecommunications Conference GLOBECOM*, 1997, pp. 1621-1625.

- 
- [79] M. J. Markowski and A. S. Sethi, "Wireless MAC Protocols for Real-Time Battlefield Communications," in *Military Communications Conference MILCOM*, Monterey, Canada, 1997.
- [80] B. Huber and W. Elmenreich, "Wireless Time-Triggered Real-Time Communication," in *Workshop on Intelligent Solutions in Embedded Systems WISES*, 2004.
- [81] D. Abdeli, S. Zelit, and S. Moussaoui, "RTH-MAC: A real time hybrid MAC protocol for WSN," in *Programming and Systems (ISPS), 2013 11th International Symposium on*, 2013, pp. 153-162.
- [82] C. Lu, B. M. Blum, T. F. Abdelzaher, J. A. Stankovic, and T. He, "RAP: a real-time communication architecture for large-scale wireless sensor networks," in *Eighth IEEE Real-Time and Embedded Technology and Applications Symposium RTAS*, Proceedings. Eighth IEEE, 2002, pp. 55 - 66.
- [83] W. Ye, J. Heidemann, and D. Estrin, "An energy-efficient MAC protocol for wireless sensor networks " in *Twenty-First Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE (INFOCOM)*, 2002, pp. 1567-1576.
- [84] J. Polastre, J. Hill, and D. Culler, "Versatile low power media access for wireless sensor networks," presented at the Proceedings of the 2nd international conference on Embedded networked sensor systems, Baltimore, MD, USA, 2004.
- [85] Y. Sun, O. Gurewitz, and D. B. Johnson, "RI-MAC: a receiver-initiated asynchronous duty cycle MAC protocol for dynamic traffic loads in wireless sensor networks," presented at the Proceedings of the 6th ACM conference on Embedded network sensor systems, Raleigh, NC, USA, 2008.
- [86] Y. Sun, S. Du, O. Gurewitz, and D. B. Johnson, "DW-MAC: A Low Latency, Energy Efficient Demand-Wakeup MAC Protocol for Wireless Sensor Networks," in *9th ACM International Symposium on Mobile Ad Hoc Networking and Computing MOBIHOC*, 2008, pp. 53-62.
- [87] J. Francomme, G. Mercier, and T. Val, "A simple method for guaranteed deadline of periodic messages in 802.15.4 cluster cells for automation control applications," in *11th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*, Prague, Czech Republic, 2006, pp. 270-277.
- [88] G. Franchino and G. Buttazzo, "WBUST: A real-time energy-aware MAC layer protocol for wireless embedded systems," in *Emerging Technologies & Factory Automation (ETFA), 2012 IEEE 17th Conference on*, 2012, pp. 1-8.
- [89] T. C. Tozer and D. J. Withers, "Satellite communications," *Communications, Radar and Signal Processing, IEE Proceedings F*, vol. 133, no. 4, pp. 317-318, 1986.
- [90] European Telecommunications Standards Institute, "Digital cellular telecommunications system (Phase 2+); Multiplexing and multiple access on the radio path, 3GPP TS 45.002 version 12.3.0 Release 12," ed: ETSI, 2015.

- [91] A. Ephremides and O. A. Mowafi, "Analysis of a Hybrid Access Scheme for Buffered Users-Probabilistic Time Division," *Software Engineering, IEEE Transactions on*, vol. SE-8, no. 1, pp. 52-61, 1982.
- [92] IEEE Computer Society, "Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless personal Area Networks (LR-WPANs)," in *802.15.4d*, ed: Institute of Electrical and Electronics Engineers, Inc., 2009.
- [93] ChipCon Products from Texas Instruments, "CC2420 2.4 GHz IEEE 802.15.4 / ZigBee-ready RF Transceiver," Texas Instruments 2012.
- [94] ChipCon Products from Texas Instruments, "CC2500 Low-Cost Low-Power 2.4 GHz RF Transceiver," Texas Instruments 2011.
- [95] R. Injong, A. Warrier, M. Aia, M. Jeongki, and M. L. Sichitiu, "Z-MAC: A Hybrid MAC for Wireless Sensor Networks," *Networking, IEEE/ACM Transactions on*, vol. 16, no. 3, pp. 511-524, 2008.
- [96] IEEE Computer Society, "Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless personal Area Networks (LR-WPANs)," in *802.15.4*, ed: Institute of Electrical and Electronics Engineers, Inc., 2006.
- [97] K. Jungsook, L. Jaehan, C. Pelczar, and J. Byungtae, "RRMAC: A sensor network MAC for real time and reliable packet transmission," in *Consumer Electronics, 2008. ISCE 2008. IEEE International Symposium on*, 2008, pp. 1-4.
- [98] S. Gabriel, D. Mosse, and R. Cleric, "TDMA-ASAP: Sensor Network TDMA Scheduling with Adaptive Slot-Stealing and Parallelism," in *Distributed Computing Systems, 2009. ICDCS '09. 29th IEEE International Conference on*, 2009, pp. 458-465.
- [99] J. Yackovich, D. Mosse, A. Rowe, and R. Rajkumar, "Making WSN TDMA Practical: Stealing Slots Up and Down the Tree," in *Embedded and Real-Time Computing Systems and Applications (RTCSA), 2011 IEEE 17th International Conference on*, 2011, pp. 41-50.
- [100] C. Jenhui and J. Shuhua, "Improvement of Slots Utilization with a Stealing-TDMA Protocol for Ad Hoc Network," in *Vehicular Technology Conference, 2006. VTC-2006 Fall. 2006 IEEE 64th*, 2006, pp. 1-5.
- [101] H. Pei, L. Chin-Jung, and X. Li, "TAS-MAC: A traffic-adaptive synchronous MAC protocol for wireless sensor networks," in *Sensor, Mesh and Ad Hoc Communications and Networks (SECON), 2013 10th Annual IEEE Communications Society Conference on*, 2013, pp. 113-121.
- [102] J. Edgar H. Callaway, "Low Power Consumption Features of the IEEE 802.15.4 WPAN Standard," in *Handbook on Theoretical and Algorithmic Aspects of Sensor, Ad Hoc Wireless, and Peer-to-Peer Networks*, J. Wu, Ed., ed Boca Raton, Florida: Auernach Publications Taylor & Francis Group, 2006, ISBN: 0-8493-2832-2, pp. 572-583.
- [103] IEEE Computer Society, "Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless personal Area Networks (LR-WPANs)," in *802.15.4*, ed: Institute of Electrical and Electronics Engineers, Inc., 2003.



- 
- [104] Y. Seong-Eun, C. Poh Kit, K. Daeyoung, D. Yoonmee, P. Minh-Long, C. Eunchang, and H. Jaedoo, "Guaranteeing Real-Time Services for Industrial Wireless Sensor Networks With IEEE 802.15.4," *Industrial Electronics, IEEE Transactions on*, vol. 57, no. 11, pp. 3868-3876, 2010.
- [105] T. H. Kim and S. Choi, "Priority-based delay mitigation for event-monitoring IEEE 802.15.4 LR-WPANs," *Communications Letters, IEEE*, vol. 10, no. 3, pp. 213-215, March 2006.
- [106] B. Fateh and M. Govindarasu, "Energy-Aware Adaptive MAC Protocol for Real-Time Sensor Networks," in *Communications (ICC), 2011 IEEE International Conference on*, Kyoto, Japan, 2011, pp. 1-5.
- [107] IEEE Computer Society, "Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless personal Area Networks (LR-WPANs)," in *802.15.4a*, ed: Institute of Electrical and Electronics Engineers, Inc., 2007.
- [108] IEEE Computer Society, "Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless personal Area Networks (LR-WPANs)," in *802.15.4c*, ed: Institute of Electrical and Electronics Engineers, Inc., 2009.
- [109] S. Lohier, A. Rachedi, E. Livolant, and I. Salhi, "Wireless Sensor Network simulators relevance compared to a real IEEE 802.15.4 Testbed," in *Wireless Communications and Mobile Computing Conference (IWCMC), 2011 7th International*, 2011, pp. 1347-1352.
- [110] G. Anastasi, M. Conti, and M. D. Francesco, "The MAC unreliability problem in IEEE 802.15.4 wireless sensor networks," presented at the Proceedings of the 12th ACM international conference on Modeling, analysis and simulation of wireless and mobile systems, Tenerife, Canary Islands, Spain, 2009.
- [111] G. Anastasi, M. Conti, and M. Di Francesco, "A Comprehensive Analysis of the MAC Unreliability Problem in IEEE 802.15.4 Wireless Sensor Networks," *Industrial Informatics, IEEE Transactions on*, vol. 7, no. 1, pp. 52-65, 2011.
- [112] M. J. L. J. Zheng, "A comprehensive performance study of IEEE 802.15.4," in *Sensor Network Operations*, I. Press, Ed., ed: Wiley Interscience, 2006, ISBN: 0-471-71976-5, pp. 218-237.
- [113] V. Stangaciu, D. Pescaru, M. V. Micea, and V. Cretu, "Practical aspects regarding implementation of real time WSN applications based on IEEE 802.15.4," in *Telecommunications Forum (TELFOR), 2013 21st*, 2013, pp. 295-298.
- [114] Digi International, "XBee™ Series 2 OEM RF Modules Product manual v1.x.2x - ZigBee Protocol," Digi International Inc. 2007.
- [115] Crossbow Technology Inc, "MPR-MIB Users Manual, Revision B, PN7430-0021-07," Crossbow Technology Inc June 2006.
- [116] Atmel, "ATmega128 Datasheet - 8 bit AVR Microcontroller with 128K Bytes In-System programmable Flash," Atmel 2009.
- [117] V. Stangaciu, M. V. Micea, and V. Cretu, "Low-Level Communication Time Analysis in Real-Time Wireless Sensor Networks," presented at

## 122 Referințe bibliografice

---

- the IEEE International Workshop on Robotic and Sensors Environments (ROSE), Timisoara, Romania, 2014.
- [118] Olimex. (2005). *LPC-H2294 Get Started Guide*. Available: <https://www.olimex.com/Products/ARM/NXP/LPC-H2294/resources/H2294.chm>
- [119] Integrated Device Technology, "3.3V CMOS Static RAM 4 Meg (256 K x 16-Bit) IDT71V416S/L," IDT Integrated Device Technology, Santa Clara, CA 95054, October 2003.
- [120] Texas Instruments, "eZ430-RF2500 Development Tools User's Guide," Texas Instruments 2009.

## Publicații

### Publicații indexate ISI:

- [A1]. M. V. Micea, V. Stangaciu, C. Stangaciu, and C. Filote, "Sensor-Level Real-Time Support for XBee-Based Wireless Communication," in *Advances in Intelligent and Soft Computing (AISC)*. vol. 145, F. L. Gaol and Q. V. Nguyen, Eds., ed Berlin, Germany: Springer Berlin Heidelberg, 2012, ISBN: 978-953-307-842-7, pp. 147-154
- [A2]. B. S. Popescu, A. Stancovici, V. Stangaciu, C. Certejan, and M. Marcu, "Intelligent wireless distributed network for power consumption monitoring and analysis," in *Science and Technology for Humanity, IEEE Toronto International Conference (TICSTH)*, Toronto, ON, Canada, 2009, pp. 730-735
- [A3]. D. Stanescu, V. Stangaciu, I. Ghergulescu, and M. Stratulat, "Steganography on embedded devices," in *Applied Computational Intelligence and Informatics, 2009. SACI '09. 5th International Symposium on*, 2009, pp. 313-318.
- [A4]. M. V. Micea, C. Certejan, V. Stangaciu, R. Cioarga, V. Cretu, and E. Petriu, "Inter-Task Communication and Synchronization in the Hard Real-Time Compact Kernel HARETICK," in *IEEE International Workshop on Robotic and Sensors Environments*, Ottawa, Canada, 2008.
- [A5]. V. Stangaciu, D. Pescaru, M. V. Micea, and V. Cretu, "Practical aspects regarding implementation of real time WSN applications based on IEEE 802.15.4," in *Telecommunications Forum (TELFOR)*, 2013 21st, 2013, pp. 295-298.
- [A6]. M. V. Micea, C. S. Stangaciu, V. Stangaciu, and V. I. Cretu, "Improving the efficiency of highly predictable wireless sensor platforms with hybrid scheduling," in *Robotic and Sensors Environments (ROSE)*, 2012 IEEE International Symposium on, 2012, pp. 73-78.
- [A7]. V. Stangaciu, M. V. Micea, V. Cretu and V. Groza, "General Slot Stealing TDMA Scheme to Improve the Low Channel Utilization Factor", in *Intelligent Signal Processing (WISP)*, 2015 IEEE International Symposium on, 15-17 May, Italy, *in press*
- [A8]. V. Stangaciu, M. V. Micea, V. Cretu, "MAC-Level Communication Time Modeling and Analysis for Real-Time WSNs", submitted for review in *Advances in Electrical and Computer Engineering*, ISSN: 1582-7445, e-ISSN: 1844-7600.

- [A9]. V. Stangaciu, M. V. Micea, V. Cretu, "Low-Level Communication Time Analysis in Real-Time Wireless Sensor Networks", in Robotic and Sensors Environments (ROSE), 2014 IEEE International Symposium on, Timisoara, Romania, 2014

Patente indexate ISI:

- [B1]. M. Marcu, A. Stancovici, V. Stangaciu, C. Stangaciu, A. Topirceanu, B. Popescu, D. Volcinschi, S.O. Fuicu, "System For Measuring and Analyzing the Energy Consumption in Electric Devices By Using Consumption Signatures", Patent Number RO127698-A0, 2011

Publicații indexate BDI:

- [C1]. M. Marcu, C. Stangaciu, A. Topirceanu, D. Volcinschi, and V. Stangaciu, "Wireless Sensors Solution for Energy Monitoring, Analyzing, Controlling and Predicting Sensor Systems and Software." vol. 57, G. Par and P. Morrow, Eds., ed: Springer Berlin Heidelberg, 2011, ISBN: 978-3-642-23583-2, pp. 1-19.
- [C2]. V. Stangaciu, G. Macariu, M. V. Micea, V. Muresan, and B. Barry, "Requirements of a Real-Time Multiprocessor Operating System for Multimedia Applications," in 6th IEEE International Symposium on Applied Computational Intelligence and Informatics (SACI), Timisoara, Romania, 2011.
- [C3]. D. Stanescu, V. Stangaciu, and M. Stratulat, "Steganography on new generation of mobile phones with image and video processing abilities," in Computational Cybernetics and Technical Informatics (ICCC-CONTI), 2010 International Joint Conference on, 2010, pp. 343-347.
- [C4]. V. Stangaciu, O. Datcu, M. Micea, „INVERTA – Specification of Real-Time Scheduling Algorithms”, submitted for review in Bulletin of the Politehnica University of Timisoara – Transactions on Electronics and Communications, ISSN: 2065-2119

Alte publicații:

- [D1]. T. Tite, C. Mihai, G. Macariu, V. Stangaciu, M.V. Micea, V. Muresan, B. Barry, "moviOS: a Real-Time Multiprocessor Operating System for Multimedia Applications ", The Scientific Bulletin of the Electrical Engineering Faculty, 57(1),pp41-50,2012, ISSN 1224-600X.
- [D2]. M. Marcu, B. Popescu, A. Stancovici, V. Stangaciu, C. Certejan, "Power Characterization of Electric, Electronic and Computing Devices",The Scientific Bulletin of the Electrical Engineering Faculty,10 (1), pp.99-106,2009,ISSN 1843-6188