

CONTRIBUȚII REFERITOARE LA CONFIGURAREA OPTIMALĂ PRIN PRISMA PERFORMANȚĂ-FIABILITATE A UNOR REȚELE DE DISPOZITIVE DE ACHIZIȚIE A DATELOR CU APLICABILITATE LA EXCAVATOARELE CU CUPE

Teză destinată obținerii
titlului științific de doctor inginer
la
Universitatea "Politehnica" din Timișoara
în domeniul calculatoarelor
de către

ing. Laurenția Timar

Șef comisie:	prof.dr.ing. Octavian Proștean
Conducător științific:	prof. dr.ing. Mircea Vlăduțiu
Referenți științifici:	prof.dr. ing. Mircea Petrescu prof.dr.ing. Vladimir Crețu prof.dr.ing. Daniela Popescu

Ziua susținerii tezei: 18 decembrie 2008

Cuvânt înainte

Teza de doctorat a fost elaborată pe parcursul mai multor ani de cercetare. În această perioadă am fost profesoară de Informatică la Liceul Nikolaus Lenau, apoi am colaborat cu firma Romínex, iar din 2001 lucrez la firma Siemens Automotive, actuală Continental.

Nefiind parte a unui colectiv de cercetare dedicat exclusiv studiului, teza am realizat-o în timpul meu liber, cele mai importante acumulări fiind în perioadele de creștere ale ultimilor doi copii.

Teza are un grad accentuat de aplicabilitate. Metodele și sistemul propuse în această teză sunt funcționale și pot fi aplicate în mai multe domenii. Din acest motiv cercetarea mea am protejat-o prin înaintarea unei cereri de brevet de invenție.

Trebuie să-l amintesc pe conducătorul meu de doctorat, prof.dr.ing. **Mírcea Vlăduțiu** care m-a susținut în toți acești ani și care după analiza cercetării m-a indemnă să înaintez brevetul de invenție.

În timpul căutărilor mele am găsit sprijin în colectivul de cercetare de la firma **Romínex**, sub îndrumarea domnului prof. dr. ing. **Cioară Titus**, care mi-a înlesnit participarea la redactarea a două articole la care sunt coautoare. Îi mulțumesc încă o dată pe această cale!

În aceste clipe gândul meu se îndreaptă spre domnul rector, prof. dr. ing. **Ioan Carțis** care m-a încurajat în toți acești ani, sfătuindu-mă să nu renunț și explicându-mi că pot să fiu un cercetător de excepție dar și o mamă bună și devotată în același timp.

Mulțumiri deosebite se cuvin domnului director **Teofil Ghirasim**, care m-a susținut și m-a propus să fac parte din colectivul de cercetare de la firma domniei sale. Le mulțumesc și angajaților și partenerilor săi!

În toți acești ani am fost susținută de minunata mea **familie**. Le mulțumesc tuturor și nu știu dacă le voi putea răsplăti vreodată pentru răbdarea și bucuria cu care au fost alături de mine! Le amintesc pe **Marioara**, soacra mea și pe **Cornelia**, actuala mea mamă, prietene de suflet.

Mulțumesc sorei mele **Aurora**, soțului meu **Lucian** și celor trei băieți: **Andrei, Gabriel și Paul**!

Îi amintesc și le mulțumesc **prietenilor și prietenelor** mele, care m-au ajutat și s-au rugat pentru mine în toți acești ani!

Îi mulțumesc cu pioșenie bunului **Dumnezeu** și **Maicii Sale** care mi-au dat multă putere și nu m-au părăsit nicio clipă!

Timișoara, decembrie 2008

Laurenția Timar

**Dedic această teză mamei mele,
Marioara.**

Timar, Laurenția

Configurații referitoare la configurarea optimală prin prisma performanță-fiabilitate a unor rețele de dispozitive de achiziție a datelor cu aplicabilitate la excavatoarele cu cupe

Teze de doctorat ale UPT, Seria X, Nr. YY, Editura Politehnica, 200Z, 168 pagini, 39 figuri, 27 tabele.

ISSN:1842-7707

ISBN (13): 978-973-625-775-9

Cuvinte cheie:

rețea de senzori, dispozitive de achiziție a datelor, protocol de comunicație, sincronizare, rata de eșantionare, baza de date, benchmark, performanță, fiabilitate

Rezumat,

În teză sunt prezentate metode de configurare optimală ale unei rețele de dispozitive de achiziție a datelor care sunt montate pe un excavator cu rotor port cupe într-o carieră de exploatare la suprafață a lignitului. S-au creat algoritmi de comunicație, care includ sincronizarea nodurilor rețelei. Datele care circulă în rețea

sunt stocate într-un sistem de achiziție a datelor, care conține o bază de date și furnizează rezultate atât sub formă grafică, cât și sub formă numerică.

Pentru validarea sistemului s-au creat aplicații de tip benchmark, care compară diferite scenarii din prisma performanță-fiabilitate și propun soluții optime pentru rețeaua creată.

Cuprins

CUPRINS	4
LISTA FIGURILOR.....	7
LISTA TABELELOR.....	10
GLOSAR ACRONIME.....	11
1. INTRODUCERE	12
1.1. OBIECTIVELE GENERALE ALE TEZEI	12
1.2. MOTIVAREA ȘI CONTEXTUL CERCETĂRII	12
1.3. DEFINIREA TEZEI ȘI MOTIVĂRILE TEHNICE	14
1.4. CERINȚELE ȘI PROBLEMELE CERCETATE ÎN CADRUL RDEA	15
1.5. ORGANIZAREA TEZEI	17
2. ANALIZA STADIULUI ACTUAL	19
2.1. DEFINIREA RDEA.....	19
2.2. SINCRONIZAREA ÎN TIMP	23
2.3. ALGORITMI PENTRU SINCRONIZAREA ÎN TIMP	26
2.3.1. <i>Network Time Protocol (NTP)</i>	27
2.3.2. <i>Reference Broadcast Synchronization (RBS)</i>	28
2.3.3. <i>The Flooding Time Synchronization Protocol (FTSP)</i>	31
2.3.4. <i>Timing-sync Protocol for Sensor Networks (TPSN)</i>	32
2.3.5. <i>Lightweight tree-based synchronization (LTS)</i>	35
2.4. ANALIZA ALGORITMILOR PENTRU SINCRONIZAREA ÎN TIMP	36
2.5. SISTEM ADAPTIV	38
2.6. METRICI.....	39
2.7. METRICI DE PERFORMANȚĂ.....	41
2.7.1. <i>Mărimea optimă a pachetului de date</i>	41
2.7.2. <i>Variația vitezei în funcție de numărul de noduri din rețea</i>	41
2.7.3. <i>Variația parametrilor în funcție de lățimea de bandă din rețea</i>	42
2.7.4. <i>Dimensiunea buffer-ului intern al unui nod de rețea</i>	44
2.8. METRICI DE FIABILITATE.....	44
2.8.1. <i>Determinarea ratei de succes SR (succes ratio)</i>	45
2.8.2. <i>Determinarea numărului de retransmisii în funcție de pachetele pierdute sau cu erori</i>	45
2.9. CONCLUZII	45
3. METODE DE COMUNICARE ÎN REȚELE DE DISPOZITIVE DE ACHIZIȚIA DATELOR	48
3.1. NIVELUL APLICAȚIE	48
3.2. DEFINIRE CADRE APLICAȚIE.....	48

3.3.	DESCRIERE ALGORITMI DE COMUNICARE	49
3.4.	ALGORITMI DE SINCRONIZAREA NODURILOR	51
3.4.1.	<i>Algoritmul 1 : Centrele de sincronizare sunt setate inițial</i>	52
3.4.2.	<i>Algoritmul 2 : Ierarhia se construiește pe un calculator personal</i>	55
3.4.3.	<i>Algoritmul 3 : Fiecare nod își construiește local ierarhia</i>	58
3.4.4.	<i>Analiza comparativă a algoritmilor de sincronizare propuși</i>	60
3.5.	NIVELUL REȚEA. DESCRIERE ALGORITMI DE COMUNICARE	61
3.5.1.	<i>IEEE 802.15.4</i>	63
3.5.2.	<i>Protocolul ZigBee</i>	65
3.5.3.	<i>Topologia ZigBee</i>	65
3.5.4.	<i>Adresarea</i>	67
3.5.5.	<i>Metoda de rutare AODV</i>	67
3.5.6.	<i>Interogarea și setarea informației nodurilor</i>	68
3.5.7.	<i>Moduri de lucru oferite de ZigBee</i>	68
3.6.	DESCRIERE RDEA	69
3.7.	TIMPUL DE RESINCRONIZARE	71
3.8.	COMUNICAREA SERIALĂ LA NIVEL REȚEA	72
3.9.	STRUCTURA GENERALĂ DAD	73
3.10.	SCHEMA ELECTRICĂ DAD	76
3.11.	MODUL DE FUNCȚIONARE AL DAD	78
3.12.	COEXISTENȚA MAI MULTOR REȚELE	79
3.13.	RELATED WORK	80
3.14.	ANALIZA COMPARATIVĂ A ALGORITMILOR PREZENTAȚI	85
3.15.	CONCLUZII	87
4.	IMPLEMENTAREA OPTIMALĂ, PRIN PRISMA PERFORMANȚĂ-FIABILITATE, A METODELOR DE COMUNICARE ÎN REȚELE DE ACHIZIȚIA DATELOR, CU APLICABILITATE LA EXCAVATOARELE CU CUPE	89
4.1.	ANALIZA SOLUȚIILOR EXISTENTE PRIN PRISMA PARAMETRULUI VIBRAȚII CA ELEMENT CRITIC AL FUNCȚIONĂRII FIABILE A EXCAVATORULUI	89
4.2.	SOLUȚII POTENȚIALE DE CREȘTERE A FIABILITĂȚII VIBRAȚIILOR	98
4.3.	NUCLEUL ARM	100
4.3.1.	<i>Watchdog timer</i>	102
4.4.	PROGRAMAREA MICROCONTROLLERULUI	104
4.5.	PROGRAMAREA NODURILOR DAD	108
4.6.	PROGRAMAREA RDEA ÎN CAZUL COMUNICĂRII RADIO	111
4.6.1.	<i>Cadrele API</i>	113
4.6.2.	<i>Configurare RDEA</i>	115
4.7.	BAZA DE DATE	115
4.8.	CONCLUZII	118
5.	EVALUAREA INDICILOR DE PERFORMANȚĂ ȘI FIABILITATE PENTRU REȚELE DE DISPOZITIVE DE ACHIZIȚIA DATELOR APLICATE LA EXCAVATOARELE CU CUPE	120
5.1.	BENCHMARK	120
5.2.	ANALIZA ÎNTINDERII RDEA	125
5.3.	MONTAREA DAD ȘI A SENZORILOR PE EXCAVATOR	125

5.4.	ANALIZA TIMPULUI DE EȘANTIONARE PENTRU FIECARE PARAMETRU FIZIC.....	126
5.5.	MĂRIMEA OPTIMĂ A PACHETULUI DE DATE.....	127
5.6.	VARIAȚIA TIMPULUI DE TRANSMISIE ÎN FUNCȚIE DE VOLUMUL DE DATE TRANSPORTAT	130
5.7.	DIMENSIUNEA BUFFER-ULUI INTERN AL UNUI NOD DE REȚEA	135
5.8.	DECIZII LUATE ÎN CADRUL PROIECTULUI PENTRU MĂRIREA PERFORMANȚEI	138
5.9.	DETERMINAREA RATEI DE SUCCES SR (SUCCES RATIO).....	138
5.10.	DETERMINAREA NUMĂRULUI OPTIM DE RETRANSMISII ÎN CADRUL RDEA.....	140
5.11.	ANALIZA TIMPULUI DE TRANSMISIE ÎN CADRUL RDEA	141
5.12.	ANALIZA VITEZEI DE TRANSMISIE ÎN CADRUL RDEA	142
5.13.	DECIZII LUATE ÎN CADRUL PROIECTULUI PENTRU MĂRIREA FIABILITĂȚII	143
5.13.1.	<i>Analiza topologiei rețelei</i>	<i>145</i>
5.13.2.	<i>Prima propunere de topologie</i>	<i>146</i>
5.13.3.	<i>A doua propunere de topologie</i>	<i>146</i>
5.14.	CONCLUZII	147
6.	CONCLUZII	148
6.1.	ASPECTE GENERALE LEGATE DE CERCETAREA EFECTUATĂ	148
6.2.	CONTRIBUȚII PERSONALE	153
	BIBLIOGRAFIE	155
	ANEXA 1	160
	ANEXA 2	162
	ANEXA 3	172

Lista figurilor

Fig. 1.2-1 Sistem eterogen format din părți mecanice și electrice	13
Fig. 1.3-1 Sistemul pe care este instalată RDEA, rețeaua de dispozitive de achiziție de date (DAD).....	14
Fig. 2.1-1 Eșantionarea unui semnal analogic	20
Fig. 2.1-2 Întârzierea de propagare.....	21
Fig. 2.1-3 Raza de acțiune	22
Fig. 2.2-1 Diferența(offset) ceasurilor.....	24
Fig. 2.2-2 Decalarea (drift) ceasurilor	25
Fig. 2.3-1 Ierarhia protocolului NTP.....	27
Fig. 2.3-2 Protocolului RBS sincronizează doar receptori	29
Fig. 2.3-3 Regresia liniară folosită pentru alegerea frecvenței relative a ceasurilor din sistem	30
Fig. 2.3-4 Distribuția gaussiană a diferenței temporare a perechilor de noduri dintr-o rețea[35]	30
Fig. 2.3-5 Timpii folosiți în algoritmul FTSP [19].....	31
Fig. 2.3-6 Pachete de date transmise în rețea folosind FTSP[19]	31
Fig. 2.3-7 Ierarhia protocolului TPSN	33
Fig. 2.3-8 Schimb de mesaje între două noduri.....	34
Fig. 2.3-9 Parcurgerea în lățime.....	35
Fig. 2.5-1 Plasarea cercetării în cadrul științei.....	38
Fig. 2.7-1 Cadru atașat unui caracter transmis	42
Fig. 2.7-2 Eșantionarea și citirea datelor de la senzori	44
Fig. 2.8-1 Rata de succes în transmisie (Success Rate)	45
Fig. 3.2-1 Cadrul de date	48
Fig. 3.4-1 Tabloul de date A	57
Fig. 3.4-2 Tabloul de date D	57
Fig. 3.4-3 Lista clustere de sincronizare	57
Fig. 3.5-1 Plasarea protocolului radio în cadrul modelului OSI.....	62
Fig. 3.5-2 Evidențierea nivelului fizic și subnivelului MAC în cadrul IEEE 802.15.4.....	63
Fig. 3.5-3 Reprezentarea celor 16 canale de comunicație oferite de standardul IEEE802.15.4, figură preluată din [61].....	64
Fig. 3.5-4 Reprezentarea unei configurații uzuale folosind IEEE802.15.4 ...	64
Fig. 3.5-5 Topologie ZigBee de tip Mesh	66
Fig. 3.6-1 RDEA și mesh-ul XBee	70
Fig. 3.8-1 Comunicarea serială în cadrul RDEA.....	72
Fig. 3.9-1 Structură generală DAD	73
Fig. 3.9-2 Configurația DAD pe module.....	74
Fig. 3.9-3 Configurația generală a unui DAD.....	75
Fig. 3.10-1 Schema electrică DAD.....	76
Fig. 3.10-2 Schema modulară cu mai multe plăci DAD.....	77
Fig. 3.10-3 DAD-uri grupate în două rețele.....	78
Fig. 3.12-1 Rețea hibridă formată dintr-o rețea serială și o rețea radio	80
Fig. 3.13-1 Rețea ZigBee într-un spital [50]	81
Fig. 3.13-2 Rețea ZigBee folosită în domeniul casnic[42]	82
Fig. 4.1-1 Amplasarea senzorului și a unui camere de filmat pe excavator. 91	

Fig. 4.1-2	Puntea tensiometrică și modulele radio	91
Fig. 4.1-3	Puntea tensiometrică și DAD-ul dedicat	92
Fig. 4.1-4	Montarea punții tensiometrice pe structura metalică	92
Fig. 4.1-5	Fișier cu date înregistrate de la puntea tensiometrică	93
Fig. 4.1-6	Graficul mărimilor din fișierul de date	93
Fig. 4.1-7	Secvență din MatLab care prelucrează datele din fișierul prezentat în Fig. 4.1-5	94
Fig. 4.1-8	Grafic din MatLab care prelucrează datele realizând o mediere a valorilor	95
Fig. 4.1-9	Amplasarea punții tensiometrice și a senzorului de accelerație pe macara	96
Fig. 4.1-10	Rețea radio formată din 2 transmițători și un receptor	97
Fig. 4.1-11	Fișier cu date înregistrate de la 2 senzori	97
Fig. 4.3-1	Detaliere DAD cu partea de ARM, adaptată din [13]	101
Fig. 4.5-1	Captură de ecran din timpul depanării în Keil	109
Fig. 4.5-2	Captura cu flash magic folosit la încărcarea fișierului .hex în microcontroler	110
Fig. 4.5-3	Configurarea comunicării seriale	111
Fig. 4.5-4	Comunicarea serială	111
Fig. 4.6-1	Structura unui cadru API	113
Fig. 4.6-2	Configurarea unui XBee folosind aplicația X_CTU	114
Fig. 4.7-1	Tabele din cadrul bazei de date a proiectului	116
Fig. 4.7-2	Datele din fișierul de stocare, înainte de introducerea în baza de date	118
Fig. 4.7-3	Afișarea datelor din baza de date folosind interogări (queries)	118
Fig. 5.1-1	Captură cu benchmark-ul folosind 6 DAD-uri	121
Fig. 5.1-2	Interogare senzori	121
Fig. 5.1-3	Alegere DAD analizat	122
Fig. 5.1-4	Alegere parametri senzor	122
Fig. 5.1-5	Vizualizare parametri senzor	123
Fig. 5.1-6	Vizualizare zonă perturbații	123
Fig. 5.1-7	Detaliere zonă perturbații	124
Fig. 5.1-8	Detalierea zonă perturbații la nivel valoric	124
Fig. 5.5-1	Timpii de transmisie a 16k octeți de date în pachete de 16 octeți	127
Fig. 5.5-2	Retransmisiile pachetelor în cazul ilustrat Fig. 5.5-1	127
Fig. 5.5-3	Analiza dimensiune pachet în funcție de timpul de transmisie ..	129
Fig. 5.5-4	Timpii de transmisie a 16k octeți de date în pachete de 48 octeți	129
Fig. 5.5-5	Retransmisiile pachetelor în cazul ilustrat în Fig. 5.5-4	129
Fig. 5.6-1	Timpul total de transmisie obținut 9,766s	130
Fig. 5.6-2	Retransmisia pachetelor în cazul experimentului din Fig. 5.6-1 ..	131
Fig. 5.6-3	Timpii de transmisie pentru o rețea cu 3 noduri	132
Fig. 5.6-4	Retransmisia pachetelor în cazul experimentului din Fig. 5.6-3 ..	132
Fig. 5.6-5	Timpii de transmisie pentru o rețea cu 5 noduri	132
Fig. 5.6-6	Retransmisia pachetelor în cazul experimentului din Fig. 5.6-5 ..	133
Fig. 5.6-7	Premisele analizei pentru o rețea cu n=5 noduri	134
Fig. 5.6-8	Variația timpilor de transmisie	135
Fig. 5.7-1	Ștampila de timp atașată datelor	137
Fig. 5.9-1	Rata de succes pentru cinci noduri cu vibrații	140
Fig. 5.10-1	Rata pachetelor pierdute pentru cinci noduri cu vibrații	141

Fig. 5.11-1 Timpul transmiterii pentru cinci noduri cu vibrații	142
Fig. 5.12-1 Viteza transmiterii pentru cinci noduri cu vibrații	143
Fig. 5.13-1 Alegerea rutei în cazul vecinătății cu o sursă de interferențe [53]	144
Fig. 5.13-2 RDEA formată din 2 mesh-uri XBee.....	145
Fig. 6.1-1 Comunicarea dintre baza de date și aplicația web.....	150
Fig. 6.1-2 Topologia ZigBee clasică	151
Fig. 6.1-3 Noua topologie DigiMesh	151

Lista tabelelor

Tab. 4.2-1 Mărimile fizice măsurate și senzorii aferenți lor	99
Tab. 4.6-1 Comparație între dispozitive și protocoale actuale [54,60].....	112
Tab. 4.6-2 Comenzile API utilizate	114
Tab. 5.3-1 Locul de amplasare al senzorilor pe excavator	125
Tab. 5.4-1 Tipul senzorilor, rata de citire și dimensiunea eșantioanelor...	126
Tab. 5.5-1 Analiza dimensiunii optime a pachetului	128
Tab. 5.5-2 Analiza dimensiune pachet în funcție de timpul de transmisie	128
Tab. 5.6-1 Analiza timpilor de transmisie de la diferite noduri din rețea ..	130
Tab. 5.6-2 Analiza timpilor de transmisie într-o rețea cu 3 noduri	131
Tab. 5.6-3 Analiza timpilor de transmisie într-o rețea cu 5 noduri	133
Tab. 5.6-4 Timpii de transmisie comparativi.....	134
Tab. 5.7-1 Timpii reali de transmitere ai unui pachet de date	135
Tab. 5.7-2 Cantitatea de date rezultată în urma eșantionării	136
Tab. 5.7-3 Îmbunătățire pentru mărirea performanței	137
Tab. 5.9-1 Rata de succes pentru două noduri cu vibrații	139
Tab. 5.9-2 Rata de succes pentru trei noduri cu vibrații	139
Tab. 5.9-3 Rata de succes pentru cinci noduri cu vibrații.....	140
Tab. 5.10-1 Rata pachetelor pierdute pentru cinci noduri cu vibrații	140
Tab. 5.11-1 Timpul transmiterii pentru cinci noduri cu vibrații	141
Tab. 5.12-1 Viteza transmiterii pentru cinci noduri cu vibrații	142

Glosar acronime

Acronim	Denumire
AES	Advanced Encryption Standard
ARM	Advanced RISC Machine
CAN	Controller Area Network
CCA	Clear Channel Assessment
COTS	Commercial-Off-The-Shelf
CRC	Cyclic Redundancy Check
CSMA/CA	Carrier Sense, Multiple Access with Collision Avoidance
DAD	Dispozitiv de Achiziția Datelor
DSSS	Direct Sequence Spread Spectrum
ED	Energy Detection
ETT	Expected Transmission Time
ETX	Expected Transmission Count
FHSS	Frequency Hopping Spread Spectrum
FTSP	Flooding Time Synchronization Protocol
GPS	Global Positioning System
I2C	Inter Integrated Circuit
INS	Inertial Navigation System
IRQ	Interrupt Request
LPC	Low Pin Count
LQI	Link Quality Indication
LTS	Lightweight Tree-based Synchronization
MAC	Medium Access Control
MSB	Most Significant Bit
NIMS	Networked Infomechanical Systems
NTP	Network Time Protocol
OSI	Open System Interconnection
PC	Program Counter
PWM	Pulse Width Demodulator
QoS	Quality of Service
RBS	Reference Broadcast Synchronization
RDEA	Rețea de dispozitive electronice asincrone
RISC	Reduced Instruction Set Computing
RSIR	Reset Source Identification Register
RTC	Real Time Clock
RTOS	Real Time Operating System
RTT	Round Trip Time
SAD	Sistem de Achiziția Datelor
SPI	Serial Peripheral Interface
STM	Synchronous Transfer Mode
TPSN	Time-sync Protocol for Sensor Networks
UART	Universal Asynchronous Receiver/Transmitter
VIC	Vector Interrupt Controller
WDT	Watchdog Timer
WLAN	Wireless LAN
WCETT	Weighted cumulative ETT

1. INTRODUCERE

1.1. Obiectivele generale ale tezei

Teza de față urmărește cercetarea, prin prisma performanță-fiabilitate, a comunicației într-o rețea de echipamente de achiziție. Transmiterea datelor în rețea se face folosind un canal unic de comunicație. Comunicația se realizează în timp real, fiind supusă perturbațiilor și interferențelor care generează erori în transmisie.

Sistemele existente până în prezent furnizează multe soluții, dar acuratețea lor este în general la nivel de secundă.

În teza de față am urmărit crearea unei metode, a unui algoritm care să realizeze eșantionarea la milisecundă a datelor de la echipamente de achiziție și proiectarea unui dispozitiv care să poată îndeplini cerințele.

Cercetarea analizează variația mai multor parametri, cum ar fi: numărul de echipamente din rețea, baud rate-ul, mărimea pachetului de date transmise, structura pachetului de date, numărul de pachete primite, tipul de erori care pot apărea, frecvența pachetelor care conțin erori, frecvența de transmisie a pachetelor, înlănțuirea pachetelor care conțin erori. Parametrii studiați sunt introduși într-un benchmark de analiză.

Scopul tezei îl reprezintă furnizarea unui rezultat, al unui optim, care să permită atât performanța sistemului cât și fiabilitatea lui.

Voi aduce argumente pentru motivarea temei alese, voi puncta direcțiile de cercetare și contribuția personală pe care o aduce această teză.

Teza nu urmărește să ia o problemă teoretică pe care să o aplice într-o dezbateră științifică adăugând anumite cuvinte cheie uzitate în industrie. Punctul de pornire îl constituie rezolvarea unei probleme concrete din domeniul industrial, iar studierea diverselor situații a facilitat direcția întrebărilor și demonstrațiilor teoretice.

1.2. Motivarea și contextul cercetării

Punctul de pornire al cercetării l-a constituit o problemă din lumea reală. Pe un excavator port cupe folosit în minele de lignit de suprafață, vezi Fig. 1.2-1, considerat de noi ca fiind un sistem eterogen format din părți mecanice și electrice, s-a cerut montarea unor echipamente electronice cu ajutorul cărora să se poată urmări funcționarea fiabilă a ansamblului.



Fig. 1.2-1 Sistem eterogen format din părți mecanice și electrice

Menționez că pe excavator sunt montate mai multe sisteme informatice de comandă și control care monitorizează și atenționează operatorul uman asupra limitelor de manevrare ale utilajului, ajungând până la oprirea automată a utilajului. Acest fel de echipamente nu fac obiectul cercetării mele.

Cercetarea mea urmărește monitorizarea excavatorului din punct de vedere al fiabilității, al respectării parametrilor și exigențelor date de producător. Se stochează zilnic un istoric al funcționării care este consultat ulterior de specialiști. Simplist vorbind cercetarea creează o "cutie neagră a excavatorului". În urma consultării datelor stocate electronic se poate decide dacă excavatorul este exploatat conform datelor din fișa tehnică, deoarece este posibil ca în anumite intervale de lucru excavatorul să fie suprasolicitat, iar în alte perioade să lucreze în gol.

Se analizează mai mulți parametri care diagnostichează dacă anumite subansamble necesită revizie sau schimbare, iar în cazul unei avarii se reface scenariul care a dus la acea situație critică.

Excavatoarele cu roată port-cupe din minele de exploatare a lignitului de suprafață au un preț ridicat iar întreținerea lor este costisioasă. Montarea și realizarea unui astfel de mastodont durează luni, dacă nu chiar ani. Defectarea excavatorului pune în pericol vieți omenești, iar repararea lui poate dura de la câteva săptămâni la câțiva ani.

Din acest motiv, producătorii doresc evitarea defectării excavatorului cât și prelungirea duratei lui de viață prin creșterea randamentului la exploatare.

Specialiștii mecanici au definit un set de mărimi fizice care trebuie urmărite și care pot să dea o imagine completă a funcționării normale a sistemului eterogen. De aici a pornit cercetarea. Sarcina tezei a fost: căutarea de metode de a măsura mărimile fizice indicate, dimensionarea lor, transmiterea și stocarea lor. S-au efectuat studii, cercetări și analize care au dus la soluția finală.

1.3. Definirea tezei și motivările tehnice

Factorul critic l-a constituit eșantionarea la milisecundă. Există grupe de parametri care trebuie citați împreună. Citirea lor crează un volum mare de date. Pentru o analiză corectă este necesară cunoașterea exactă a cronologiei citirii evenimentelor. Aceste premise au dus la căutarea de metode, iar în final la elaborarea de algoritmi care să realizeze eșantionarea la nivel de milisecundă, sincronizarea dispozitivelor pentru furnizarea unei cronologii necesare realizării istoricului evenimentelor și gestionarea unui volum mare de date într-un interval de timp scurt.

Bazându-mă pe algoritmi creați, am realizat un sistem de echipamente de achiziția datelor care să răspundă cerințelor date, echipamentele folosind dispozitive electronice de ultimă oră.

Ideea de cercetare urmărită în această teză constă în căutarea de metode de creare, dimensionare și management al unei rețele de dispozitive electronice asincrone (RDEA), care să poată asigura monitorizarea mărimilor mecanice ale excavatorului din punctul de vedere al performanței dar și al fiabilității.

RDEA este formată din mai multe dispozitive de monitorizare și achiziție de date (DAD).

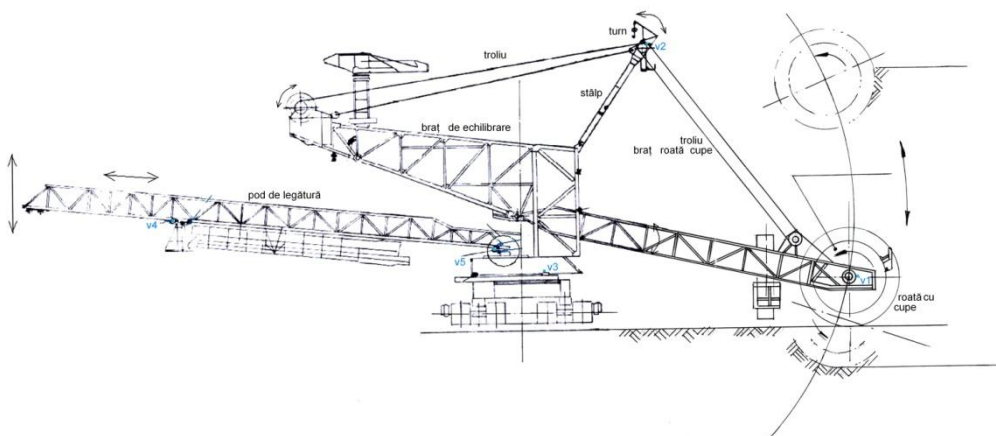


Fig. 1.3-1 Sistemul pe care este instalată RDEA, rețeaua de dispozitive de achiziție de date (DAD)

În Fig. 1.3-1 este reprezentat schematic un excavator port-cupe și câteva puncte V1,...,V5 care reprezintă nodurile rețelei în cadrul căreia se desfășoară studiul de față.

Nodurile V1,...,V5 sunt punctele de pornire care monitorizează parametri fizici. Fiecare parametru are o caracteristică proprie, care determină o anumită rată de eșantionare a valorilor variațiilor lui. Un nod V_i poate supraveghea mai mulți parametri diferiți sau identici. Datele culese de la nodurile rețelei trebuie gestionate și stocate.

Pentru comunicare am ales varianta unei rețele cu difuzare, bazată pe un canal unic, în care toate nodurile pot primi informații și pot transmite în întreaga

rețea, unui singur nod sau unui grup de noduri [1]. Datele transmise de la toate stațiile operabile din rețea, sunt stocate într-o bază de date. Baza de date este analizată de specialiști care interpretează informația având în vedere performanța prin prisma fiabilității. Datele stocate stau la baza mai multor analize, dintre care amintesc:

- Determinarea regimului de funcționare, urmărindu-se mărirea randamentului în exploatare.
- Determinarea stării de uzură a anumitor subansamble și diagnosticarea lor pentru revizie sau înlocuire.
- Se poate calcula cantitatea de minereu extrasă.
- În cazul unei avarii se poate reconstitui ultimul scenariul determinându-se ce a dus la provocarea lui.

Având în vedere cerințele particulare ale acestui sistem nu am putut folosi o soluție clasică, existentă. Am studiat metode și algoritmi iar în final am creat algoritmi care să soluționeze gestionarea rețelei propuse în Fig. 1.3-1.

Pentru validarea soluției am creat sisteme de achiziția datelor. Sistemele au fost montate pe excavator și pe ele s-au implementat algoritmi creați în această teză. Având în vedere originalitatea și amploarea pe care a luat-o cercetarea, în final soluția propusă este protejată de un brevet de invenție[87].

În plus, la metodele propuse în această teză am atașat metrici de performanță și fiabilitate, care după definire au fost testate într-un benchmark creat special pentru analiza algoritmilor creați, iar în urma testărilor am propus soluții pentru buna funcționare a metodei realizate în această teză care este implementată pentru excavatorul prezentat în Fig. 1.2-1.

Sistemul propus în această disertație, pornește de la analiza unor parametri fizici, și furnizează metode de gestionarea unui volum mare de date eșantionate la milisecundă, construind un istoric de funcționare care este extrem de util proprietarului excavatorului.

Din punctul meu de vedere, chiar dacă se efectuează o analiză preliminară, valorile datelor și interpretarea lor sunt transparente, fiind important ca datele să fie citite și să ajungă nealterate în baza de stocare.

1.4. Cerințele și problemele cercetate în cadrul RDEA

Pentru definirea metodei de transmisie a datelor și realizarea cronologiei evenimentelor, am urmărit mai multe direcții de cercetare:

- definirea parametrilor fizici și găsirea frecvenței optime de eșantionare, astfel încât, pe baza eșantioanelor să se descrie variația reală a fenomenului. Dacă datele depășesc anumite limite, se modifică perioada de eșantionare, iar transmiterea se face mai rapid sau mai lent, în funcție de situație. Acest mod de eșantionare este descris în literatură sub numele de eșantionare adaptivă [51].
- fiecare nod, al rețelei, reprezentate în Fig. 1.3-1 prin V_i , are un buffer intern de stocare în care sunt memorate eșantioanele. Din acest buffer sunt extrase datele, sunt asamblate pachete și sunt transmise în rețea. Dacă frecvența de transmitere a datelor în rețea este mai mică decât frecvența de eșantionare a datelor de la nivelul nodurilor V_i , bufferul intern este suprascris, iar datele sunt pierdute.

- Se urmărește corelația dintre viteza maximă de înregistrare, volumul de date înmagazinat și capacitatea de memorie disponibilă.
- în rețeaua cu difuzare propusă transmiterea se poate realiza serial sau radio (wireless). Indiferent de modalitatea de implementare aleasă, principiile transmisiei rămân aceleași.

Metoda propusă în această teză urmărește evitarea supraîncărcării, adică a situațiilor în care volumul datelor transmise de nodurile din rețea este mai mare decât rata de transfer și se pot pierde cadre. O gestionare bună trebuie să evite pierderea datelor, prin asigurarea unui protocol dinamic de comunicație care să permită modificarea ratei de transfer sau corelarea timpilor de transmisie și recepție.

Cercetarea urmărește desfășurarea unei comunicații optimale în rețea, care vizează atât performanța dar și fiabilitatea, propunând în final o soluție considerată de noi optimă din perspectiva dualității cerințelor.

Punctul de start l-a constituit asamblul de mărimi fizice de tip mecanic și electric: tensiuni, curenți, vibrații, temperaturi, unghiuri de înclinație, poziție geografică.

Am studiat forma de stocare a mărimilor fizice în format electronic, precizia lor, frecvența de eșantionare.

Am cercetat protocoale de funcționare și metodele potrivite pentru RDEA propusă. Soluția oferită de mine permite funcționarea dinamică a RDEA, având în vedere posibilitatea apariției de noi noduri în rețea sau pierderea legăturii cu cele existente.

În teza de față nu urmăresc elaborarea unui standard, dar caut o soluție care să permită circulația unui volum maxim de date într-o rețea de dispozitive de achiziția datelor, cu condiția ca la un moment dat un singur nod să transmită. În plus, achiziția datelor se face cu o acuratețe mărită față de cercetările efectuate până acum în domeniu. Eșantionarea datelor se face la milisecundă, iar sincronizarea nodurilor din RDEA trebuie realizată tot la nivel de milisecundă. Sistemele de achiziție nu sunt calculatoare personale. Ele sunt dispozitive cu resurse limitate. Pentru elaborarea unui soluții optime am avut în vedere utilizarea și crearea unor algoritmi care nu consumă excesiv resursele nodurilor din RDEA, studiind în acest fel raportul performanță/fiabilitate la nivelul nodurilor din rețea.

Pentru demonstrarea cercetării și aplicarea soluției prezentate în această teză am realizat o aplicație încapsulată (embedded) care funcționează în timp real. Pe lângă funcționarea normală, care a fost testată și validată s-au creat și scenarii critice. Fiecare funcționare și scenariu a fost analizat comparativ. Am creat un benchmark care a testat metricile de performanță și fiabilitate propuse în această teză, special pentru algoritmi prezențați.

Bazându-mă pe rezultatele teoretice realizate și pe algoritmi creați am urmărit realizarea practică a unui sistem format din dispozitive de achiziție care să realizeze constrângerile și cerințele din algoritmi. Am căutat dispozitive cu ajutorul cărora se pot implementa metodele. Acolo unde nu s-au găsit echipamente dedicate scopului urmărit am adaptat sau creat special altele. După ce dispozitivele au fost create, l-am configurat și integrat în rețea.

Concluzionând pot spune că:

- am analizat variația mărimilor fizice monitorizate și am creat metode pentru stocarea valorilor furnizate de ele
- am creat metoda de comunicare în rețea
- am furnizat o soluție teoretică care înglobează cercetarea funcționării fiabile și performante a sistemului eterogen existent, utilizând metode și algoritmi de comunicare
- am elaborat o metodă de gestionare a resurselor rețelei, bazată pe algoritmi de comunicare și sincronizare
- soluția teoretică propusă este dinamică, iar RDEA are caracteristicile unui sistem adaptiv, aceste caracteristici fiind modelate în cadrul metodei
- am demonstrat validitatea soluției propuse creând o aplicație practică
- am propus metrici de performanță și fiabilitate pentru verificarea soluției teoretice propuse
- bazându-mă pe protocol, am ales dispozitivele de achiziționarea datelor și acolo unde nu au existat, am creat echipamentul necesar
- am configurat dispozitivele de achiziția datelor conform studiilor realizate în cercetare
- am dimensionat rețeaua formată din dispozitivele de achiziția datelor
- am creat diverse scenarii de funcționare a rețelei și le-am analizat comparativ, folosind un benchmark

Variația parametrilor și analizele s-au efectuat atât din punctul de vedere al performanței, cât și din punctul de vedere al fiabilității. Teza prezintă o soluție optimă de funcționare a RDEA.

În final, dar nu în ultimul rând, urmăresc să generalizez cercetarea astfel încât soluția prezentată să se poată utiliza și în alte situații și domenii, nu doar pentru excavatorul prezentat în Fig. 1.3-1.

În dizertație, pe mai departe, voi folosi denumirile de DAD pentru un sistem de achiziția datelor, și RDEA pentru rețeau de DAD-uri amplasate pe excavator.

1.5. Organizarea tezei

Teza de față este organizată astfel:

Capitolul2: "*Analiza stadiului actual*", prezintă algoritmi de sincronizare folosiți în rețelele radio, definește metrici de performanță și fiabilitate, evidențiază caracteristici ale unui sistem adaptiv, iar în final încadrează cercetarea curentă în domeniul științei.

Capitolul3 : "*Metode de comunicare în rețele de dispozitive de achiziția datelor*", prezintă cerințele și constrângerile necesare definirii rețelei. Aceste constrângeri le analizez și propun metode teoretice de soluționare a lor. Prezentarea, cu analogie la sistemul de referință OSI, se realizează pe două niveluri,

nivelul aplicație și nivelul rețea. La nivelul aplicație sunt prezentate structura și tipul cadrelor care circulă în rețea, algoritmi de comunicare, algoritmi de sincronizarea DAD-urilor în rețea. La nivel rețea prezint protocoalele care realizează transmisia pachetelor în rețea. Tot în acest capitol detaliez funcționarea și structura DAD-urilor și prezint tipuri de rețele care folosesc aceste dispozitive.

Capitolul4: "Implementarea optimală, prin prisma performanță-fiabilitate a metodelor de comunicare în rețele de dispozitive de achiziția datelor cu aplicabilitate la excavatoarele cu cupe", prezintă etapele de proiectare și implementare ale RDEA, pornind de la traductoare, configurația hardware de achiziție, aplicația (software) de monitorizare, crearea rețelei și stocarea datelor în baza de date. Tot în acest capitol prezint câteva proiecte de cercetare la care am participat ca și coautoare și care au fost fructificate prin două articole.

Capitolul5 : "Evaluări ale indicilor de performanță și fiabilitate pentru rețele de dispozitive de achiziția datelor, aplicate la excavatoarele cu cupe", realizează validarea sistemului prin crearea unei rețele de dimensiuni reduse, dar care satisface cerințele definite în cadrul teoretic. În cadrul rețelei achiziționez datele, iar cu ajutorul unui benchmark testez diferitele configurații și realizez analiza comparativă a metricilor propuse în capitolul 2.

Capitolul6: "*Concluzii*", prezintă rezultatele analizelor realizate în capitolul 5. Capitolul cuprinde două părți. În prima parte, 6.1, furnizez concluziile și soluția elaborată în urma analizei din cadrul tezei. Prezint direcții viitoare de cercetare și aplicabilitatea tezei în alte domenii decât cel minier cu excavatorul cu cupe. În partea a doua, 6.2, creionez contribuțiile personale pe care le-am adus prin crearea acestei teze.

2. ANALIZA STADIULUI ACTUAL

În acest capitol încadrez cercetarea din teză în sfera științei actuale, determinând domeniul în care se va dezvolta cercetarea.

2.1. Definirea RDEA

Pentru elaborarea metodei prezentate în această teză am studiat tehnici de abordare în cazul modelării unui sistem încapsulat (embedded) în timp real. Dintre abordările studiate amintesc limbajele de modelare Rosetta, CHARON și Giotto.

Rosetta este un limbaj de modelare pentru specificarea și analiza sistemelor eterogene complexe. Sunt interesante modurile de abordare ale unui sistem complex și fațetele prin prisma cărora este specificat[65].

CHARON, este un limbaj de modelare folosit pentru designul sistemelor hibride. Permite analize legate de parametri în cadrul funcționării continue sau discrete, descrierea realizându-se atât arhitectural cât și funcțional [67].

Giotto furnizează un model matematic, un formalism pentru modelarea sistemelor în timp real. Giotto furnizează cadrul citirii senzorilor sau apelării periodice a task-urilor. În cadrul modelării se definesc specificații cum ar fi tipurile porturilor care intervin în sistem[81].

Limbajele de modelare prezentate evoluează la nivel experimental. Rosetta a modelat un întrerupător (actuator) , circuite integrate și câteva plăci care conțin circuite. CHARON implementează studiul asupra unui sistem format din două vehicule care comunică prin intermediul a patru porturi. Giotto a înaintat cel mai mult cu experimentele practice realizând comanda la distanță a unui aeromodel, un elicopter comandat la distanță, cât și a unei escadrile de avioane de vânătoare [82].

În afară de modelarea unui sistem încapsulat am studiat interacțiunea lui cu mediul înconjurător.

O caracteristică a sistemelor încapsulate (embedded) este integrarea lor, prin intermediul senzorilor, în procesele fizice, devenind parte a mediului înconjurător. Design-ul unui sistem încapsulat urmărește configurarea unui sistem de calcul astfel încât să facă față variațiilor parametrilor fizici din mediul înconjurător. Procesoarele moderne permit interacțiunea dintre codul sursă și caracteristicile fizice ale dispozitivelor de calcul. Tendința este de a utiliza componente larg utilizate COTS (Commercial-off-the-shelf) în aplicații diferite. Prețul lor este accesibil, structura și funcționarea cunoscute. Diferența se realizează prin programe (software). Modificările realizate prin program sunt flexibile și mai ușor de realizat decât cele care implică hardware-ul.

Trebuie să atrag atenția că specificarea și implementarea sistemelor încapsulate (embedded) trebuie privită diferit față de design-ul sistemelor de calcul cu care ne-am obișnuit deja[64].

Deosebirile constau în:

- siguranța, prin furnizarea unor calități ale serviciului (QoS) furnizat cunoscute, eficiente și scalabile

- adaptabilitatea, care permite reconfigurarea sistemului și funcționarea lui în parametri nominali.

-minimizarea prețurilor, prin folosirea unor componente COTS, iar design-ul este realizat modular ceea ce va implica costuri reduse la modificarea anumitor module sau în cazul unor versiuni îmbunătățite (upgrade) [83].

Toate aceste direcții trebuie colaborate cu cerințele de funcționare în timp real al sistemelor și politicile de redresare în cazul apariției erorilor.

Studierea acestor articole a ilustrat modul de analiză al cerințelor și definirea ipotezelor inițiale, modul în care trebuie pusă problema.

RDEA este o rețea formată din dispozitive de monitorizare și achiziție a datelor care au aceeași configurație hardware. DAD-urile sunt stații independente care transmit sau recepționează date. O stație poate transmite spre o altă stație, în toată rețeaua, sau doar spre un anumit grup de stații. Este vorba de transmiterea broadcast, unicast sau multicast.

Comunicarea în rețea este dinamică. Fiecare stație are o adresă unică. La un moment dat una sau mai multe stații pot părăsi rețeaua sau pot reveni în rețea. În tot acest timp rețeaua trebuie să rămână funcțională.

Eșantionarea datelor unei stații se poate modifica dinamic, în timpul funcționării. Transmiterea datelor se face în mod asincron.

Prin eşantionarea (sampling) unui semnal, se înțelege metoda de reducere de la semnal continuu la semnal discret prin selectarea datelor la un interval de timp corespunzător pentru a reprezenta cât mai fidel procesul care este măsurat [38,39]. Un eşantion se referă, caracterizează, o valoare sau un set de valori în timp sau în spațiu. Eşantionarea urmăreşte variaţia şi consistenţa mărimilor fizice măsurate, în raport cu anumite puncte de referinţă sau cu aceleaşi valori dar măsurate la intervale diferite de timp. În cadrul cercetării se va realiza eşantionarea periodică, adică se vor citi valorile la intervale egale de timp.

În fig. 2.1-1 este un exemplu de eşantionare periodică unui semnal analogic. Perioada de eşantionare este constantă. Folosind eşantioanele create se poate reface forma semnalului analogic sau se pot determina toate caracteristicile lui principale, cum ar fi amplitudinea și perioada.

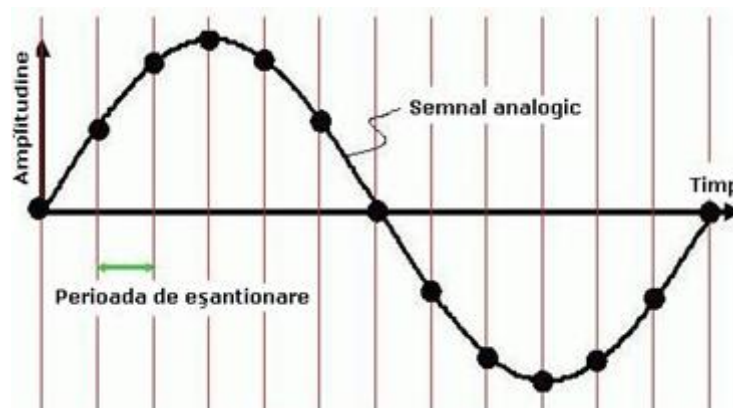


Fig. 2.1-1 Eşantionarea unui semnal analogic

Dacă mărimea fizică măsurată își schimbă, chiar și periodic, evoluția, se poate întâmpla ca eșantionarea utilizată să descrie doar o parte a fenomenului. În acest caz este necesară schimbarea perioadei de eșantionare. Mergând mai departe se poate discuta despre anticiparea unei rate de eșantionare bazată pe valorile înregistrate deja. Predicția eșantionării nu trebuie să cuprindă doar un anumit mecanism sau dispozitiv, ci având în vedere că lucrăm cu o rețea, este recomandat să se extindă abordarea.

Eșantioanele de date (samples) sunt încapsulate într-un format necesar protocolului, formând pachete, care pot avea lungimi diferite. Pachetele sunt împărțite în cadre și transmise în rețea. În rețea pot circula două sau mai multe tipuri de cadre, cum ar fi cadre de date și cadre de control. Toate cadrele de date au o lungime maximă stabilită. În funcție de mărimea pachetului pe care îl poartă, cadrele de date pot avea lungime maximă sau nu. Cadrele de control sunt folosite pentru gestionarea rețelei. O stație poate transmite și recepționa atât cadre de date cât și cadre de control.

Există un singur canal de comunicație, la care au acces toate stațiile. Prin el se transmite și se recepționează. Acest canal este aerul, dacă se folosesc module de transmisie radio; sau cablul în cazul transmisiei seriale. Principiile de bază ale comunicării rămân aceleași. La un moment dat trebuie să transmită doar o singură stație. Dacă transmit mai multe stații, apar coliziuni, pachetele se amestecă și datele se alterează. Trebuie gestionată planificarea următoarei stații care va ocupa canalul.

În RDEA toate nodurile monitorizează, „ascultă” canalul de transmisie. Din acest motiv este necesar să se știe dacă la un moment dat un nod transmite sau nu.

Față de transmisia serială, în situația emiterii radio apar mai multe probleme. Din acest motiv mă voi opri asupra lor.

În cazul transmiterii radio, în rețea pot apărea întârzieri de propagare, datorate parcurgerii mai multor stații, numite hopuri, până la destinație.

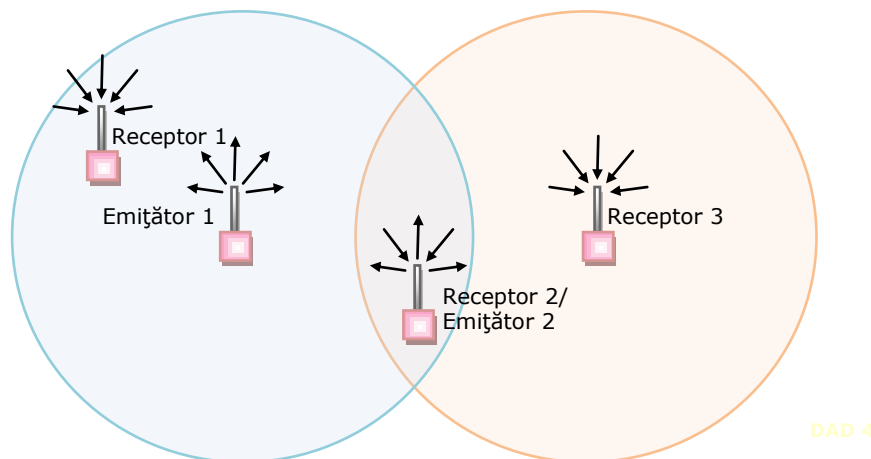


Fig. 2.1-2 Întârzierea de propagare

În Fig. 2.1-2 este prezentată întârzierea de propagare. Emitătorul 1 trimite date în rețea. Receptorul 1 și Receptorul 2 sunt în raza lui de acțiune și vor primi

pachetele. Receptorul 3 nu este în raza de acțiune a transmițătorului. În acest caz Receptorul 2, după primirea datelor le va transmite Receptorului 3. Stația care recepționează date și apoi le retransmite este notată în Fig. 2.1-2 ca Receptor2/Emițător2.

Rețelele în care datele trec prin mai multe noduri pentru a ajunge la un receptor se numesc rețele multi-hop. La celălalt pol, există rețele uni-hop sau single-hop sau cu 1 hop, cum este cazul comunicării între Emițător1, Receptor1 și Receptor2/Emițător2, ilustrat în Fig. 2.1-2.

O altă problemă, specifică rețelelor radio (wireless) este raza de acțiune. Aceasta este ilustrată în Fig. 2.1-3.

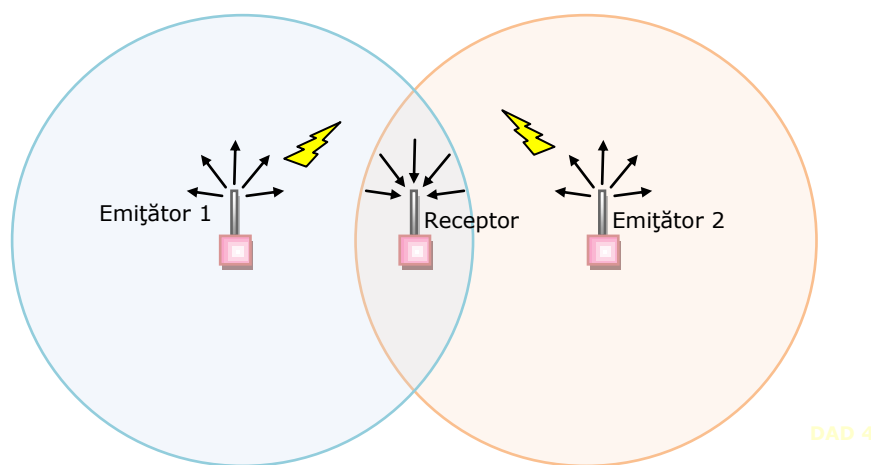


Fig. 2.1-3 Raza de acțiune

Receptorul se află atât în raza de acțiune a Emițătorului 1, cât și în raza de acțiune a Emițătorului 2. Emițător 1 și Emițător 2 nu pot comunica direct, "nu se văd". La un moment dat Emițător 1 și Emițător 2 încep să emită date în rețea. Stația Receptor recepționează date de la ambele stații. Datele care ajung sunt amestecate și nu pot fi folosite. În rețea trebuie să existe o stație care supervizează comunicarea, determinând care stație transmite și pentru cât timp. Această stație poartă numele de coordonator sau dispecer.

Se poate întâmpla ca un pachet transmis să nu ajungă la destinație. Stația care a transmis pachetul trebuie să știe dacă pachetul a ajuns la destinație sau nu. Există metode de tratare a acestei situații folosind pachete de control, de confirmare, de la receptor spre transmițător sau mecanisme cu expirarea timpului, timeout.

În timpul transmisiei, chiar dacă pachetul ajunge la destinație, informația din el se poate altera, datorită erorilor apărute în timpul transmisiei. Situația se rezolvă prin ignorarea sau retransmisia pachetului.

Fiecare stație transmite direct la SAD (Sistem de Achiziția Datelor) eșantioanele, iar acolo sunt stocate.

Un deziderat al cercetării este ca sistemul de comunicație implementat de mine să funcționeze dinamic. Se realizează o analiză preliminară a datelor eșantionate și se modifică rata de eșantionare în funcție de rezultatele analizei. Din

definirea rețelei se desprind multe direcții de studiu. În această teză voi căuta o metodă de comunicație care să realizeze dezideratele prezentate până acum, având în vedere scopul pentru care a fost creată rețeaua. Voi studia structura cadrelor de date și influența lor în fluidificarea traficului, voi aplica metrici de performanță și fiabilitate și voi urmări dinamica rețelei.

În continuare în capitolul 2 sunt prezentate problemele legate de timp și sincronizare în rețelele radio. Sunt prezentați algoritmi general folosiți, iar în final se conturează direcția abordării mele.

2.2. Sincronizarea în timp

Pentru proiectul meu sincronizarea este necesară deoarece se creează o rețea, RDEA din mai multe noduri, care monitorizează un sistem, de dimensiuni mari. Datele primite de la parametri monitorizați sunt prelucrate și fuzionează în cadrul analizei. Din acest motiv este important să știu precis momentul în care a avut loc un eveniment pentru a putea modela funcționarea întregului sistem. Cronologia evenimentelor stă la baza analizei și necesită timp precis.

Benveniste ne îndeamnă să gândim sincron, dar să acționăm asincron („think synchronously – act asynchronously”), adică o situație asemănătoare proiectul realizat în această teză, care implementează o rețea de noduri sincrone care interacționează prin comunicații asincrone [66].

Aceleași sfaturi le oferă și Caspi, după o experiență de 50 de ani în domeniu. Rețelele de senzori sunt în strânsă legătură cu mediul înconjurător, care variază asincron. Dar în cadrul implementării trebuie să se folosească un anumit paralelism, care poate fi realizat în diverse moduri, deoarece mai mulți parametri urmăresc același fenomen. În altă ordine de idei el atrage atenția asupra riscurilor pe care le poate cauza sincronizarea. Unul dintre ele îl reprezintă eșantionarea care poate să „piardă” anumite evenimente dacă se desfășoară la o rată prea mare [68]. Din acest motiv, pentru a evita astfel de situații, în aplicația mea am realizat un sampling dinamic.

Cel mai precis sistem de măsurare al timpului este ceasul atomic. El a fost creat pentru prima dată în jurul anului 1950 [37, 38]. Ceasul atomic nu se bazează pe secundă, ci pe oscilațiile izotopului de cesiu 133. Frecvența atomului de cesiu este foarte stabilă, în comparație cu rotația Pământului. Într-o secundă izotopul de cesiu 133 creează 9192 631 770 de oscilații. Folosirea unui ceas bazat pe acest principiu introduce o eroare de 1 secundă la 15 milioane de ani [38, 39]. Astfel în jurul anului 1960 izotopul de cesiu a fost adoptat ca unitatea de măsurare a timpului, înlocuind secunda care era folosită până atunci.

Timpul indicat de ceasul atomic este folosit de semnalele GPS, de sistemele distribuite de televiziune, sau serverele de timp atomic.

O soluție ar fi să echipăm fiecare placă cu un modul GPS. Am avea pe toate stațiile o bază de timp egală și timp precis. Nu putem recurge la această soluție deoarece este voluminoasă și scumpă. Mai mult, în rețeaua noastră nu avem nevoie de o acuratețe temporară strictă, ci mai mult de o cronologie strictă a evenimentelor, determinată de sincronizarea stațiilor.

Voi prezenta modul cum materializează un computer noțiunea de timp.

Orice calculator are un ceas intern, „internal clock”, care urmărește ca circuitele din care este format să funcționeze în același timp. Pe lângă ceasul

sistemului există un oscilator hardware cu cuarț. Circuitul se bazează pe rezonanța mecanică a cuarțului care vibrează și folosește vibrațiile pentru a crea un semnal electric cu o frecvență precisă. În acest fel se creează o serie de tacturi care sunt folosite de ceasul intern pentru sincronizarea operațiilor efectuate de circuitele calculatorului.

Cu toate acestea, uneori ceasurile interne ale computerelor se pot decala. Astfel un ceas al computerului poate să aibă decalări neuniforme, în diferite ocazii.

Într-o rețea de calculatoare, prin sincronizare se urmărește ca baza de timp a tuturor stațiilor din sistem să fie identică. Se creează un timp global pentru toată rețeaua și acest timp global se scurge la fel la toate stațiile. Prin sincronizarea în timp a două ceasuri se înțelege potrivirea ceasurilor ca la un moment dat să indice exact aceeași oră.

Benveniste descrie noțiunile de sincronism, asincronism și sincronizare. Prezintă trei domenii de interes în realizarea unui sistem în timp real și premisele legate de ele. Aceste domenii sunt:

- crearea de arhitecturi hard și soft care interacționează asincron între ele
- realizarea unei structuri hardware sincrone
- crearea sistemelor de control în timp real [66].

În literatura de specialitate au fost definite mărimi care determină sau aproximează diferența dintre timpul real și timpul dispozitivului, în cazul nostru al DAD-ului. Dintre ele amintesc: diferența (offset), distorsionarea (skew) și decalarea (drift)[20].

Diferența (offset) reprezintă diferența de timp: numărul de minute, secunde sau milisecunde dintre ceasurile a două DAD-uri din rețea față de momentul t , sau diferența ceasului unui DAD față de timpul real t . Această diferență apare de obicei deoarece ceasurile nu pornesc în același timp și prezintă o desincronizare.

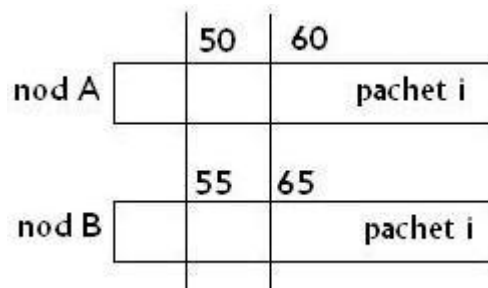


Fig. 2.2-1 Diferența(offset) ceasurilor

În Fig. 2.2-1 este ilustrată diferența ceasurilor a două noduri A și B. Nodurile merg sincron, transmit pachete de date, dar nodul B a intrat în rețea mai târziu decât nodul A și are o întârziere de 5 unități.

Matematic, diferența (offset) este reprezentată în (2.2-1), prin scăderea timpilor celor două ceasuri.

$$offset(\delta) = C_A(t) - C_B(t) \quad (2.2-1)$$

unde $C_A(t)$ reprezintă timpul local al DAD-ului A, iar
unde $C_B(t)$ reprezintă timpul local al DAD-ului B.

(2.2-1) reprezintă diferența (offset-ul) de timp dintre două noduri de rețea

Există situații când frecvența cuarțurilor se modifică în timp. Această modificare poartă numele de decalare (drift) a timpului.

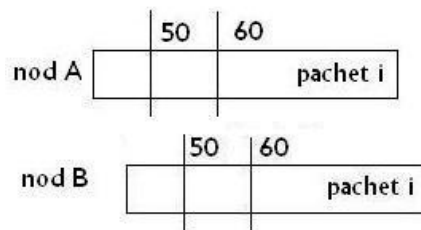


Fig. 2.2-2 Decalarea (drift) ceasurilor

În Fig. 2.2-2 este reprezentată decalarea ceasurilor a două noduri A și B. Nodurile au pornit în același timp, dar pe parcursul funcționării a apărut, din motive diverse, decalarea (drift).

S-au definit metrici pentru măsurarea frecvenței ceasurilor, cum ar fi precizia sau stabilitatea. Prin precizie se înțelege diferența dintre ce indică oscilatorul de cuarț și ce ar trebui să indice. Aceasta mai poartă numele de eroare de frecvență și valorile ei sunt de obicei specificate de producători [35]. Stabilitatea este tendința oscilatorului de a rula la aceeași frecvență în timp. Stabilitatea frecvenței este determinată de factori naturali, cum ar fi temperatura sau tensiunea de alimentare. Precizia și stabilitatea frecvenței pot influența în timp valoarea ceasului.

În concluzie, ceasurile rețelei de-a lungul funcționării ajung să măsoare timpul diferit, motiv pentru care se recurge la sincronizare. În cadrul RDEA pot exista DAD-uri care lucrează la frecvențe diferite. Aceasta nu este un impediment pentru cercetare. Pentru realizarea scopului dorit: sincronizarea în timp a nodurilor din rețea, corelarea diferențelor (offset) de timp este necesară și suficientă.

Există mai multe abordări ale sincronizării. Vom adopta clasificarea propusă în [20], având trei tipuri de sincronizare: ordonarea evenimentelor și a mesajelor, corelarea ceasurilor locale sau sincronizarea întregii rețele.

În cazul ordonării evenimentelor nu se urmărește sincronizarea ceasurilor stațiilor din rețea [20]. Algoritmii corelează evenimentele și stabilesc dacă evenimentul E1 a avut loc înainte sau după evenimentul E2. Această modalitate nu este potrivită abordării mele. Deși RDEA are dimensiuni mici spre medii, se vehiculează un număr mare de evenimente și mesaje. Ordonarea lor ar necesita o putere de calcul mai mare decât sincronizarea propriu zisă a tuturor stațiilor.

În cazul celui de-al doilea mod de sincronizare, algoritmii permit stațiilor să funcționeze independent, având fiecare ceasul său intern și deținând, în plus informații despre decalarea (drift) și diferența (offset) ceasurilor celorlalte stații din sistem. La un moment dat, folosind aceste informații, o stație X își poate converti ceasul și să se sincronizeze cu orice altă stație Y. Această metodă permite sincronizarea perechilor de stații. Tehnica se poate folosi în teză la intrarea unei noi stații în rețea și la sincronizarea anumitor perechi de stații. Presupunem că întreaga

rețea este sincronizată. La un moment dat o stație nouă intră în rețea. Se pot aplica algoritmi din această clasă pentru sincronizarea noii stații cu o alta care este deja sincronizată. În acest fel întreaga rețea va deveni sincronizată.

Ultima metodă furnizează algoritmi care mențin tot timpul întreaga rețea sincronizată. Este o metodă costisitoare și consumatoare de resurse pentru fiecare nod al rețelei.

O mare problemă în rețelele radio este faptul că orice resursă ce este folosită pentru sincronizare reduce din resursele disponibile pentru îndeplinirea sarcinii de bază a nodului.

În cadrul RDEA este necesară sincronizarea nodurilor pentru a furniza cronologia evenimentelor. Datele de la senzori sunt prelucrate de specialiști mecanici.

Pentru realizarea sincronizării, voi analiza câțiva dintre algoritmi existenți pentru a putea elabora o metodă potrivită rețelei noastre.

2.3. Algoritmi pentru sincronizarea în timp

La definirea rețelei am prezentat ipoteza canalului unic de comunicare și indiferent dacă transmisia este radio sau prin fir, principiile de bază sunt aceleași. Teoria dezvoltată aici va fi validată prin implementarea practică a rețelei, în cadrul căreia se va folosi comunicarea radio. Din acest motiv, pe mai departe, analiza va acorda atenție sincronizării din rețelele radio, de senzori.

Din literatura de specialitate, Sivrikaya în [20] adoptă o clasificare a algoritmilor după mai multe criterii, pe care le putem aplica în studiul nostru și să creăm un model general. O alternativă o constituie funcționarea desincronizată a rețelei, în care fiecare stație se bazează pe ceasul propriu local, dar deține informații despre ceasul global sau un tabel cu parametri legați de timpul celorlalte ceasuri din sistem. La un moment dat stația își poate sincroniza ceasul cu alt ceas din sistem, sau îl ajustează folosind ceasul global.

Sincronizarea se poate face între transmițător și receptor, dar și doar între receptori; se poate face atât la nivel MAC (*Medium Access Control*) cât și la nivel utilizator, în cadrul aplicației.

Am creionat un model general de sincronizare cu mai multe opțiuni, folosit ca reper în cadrul studierii algoritmilor.

În alegerea modului de sincronizare, există mai multe direcții generale valabile. Din grupul de cerințe oferit de Sivrikaya în [20] și Elson în [34], am extras elementele care trebuie urmărite în sincronizarea RDEA: precizia, robustețea, dimensiunea rețelei și ipotezele funcționale de configurare ale rețelelor.

Precizia: realizarea dezideratului poate să se materializeze prin abordări diferite, care au fost amintite anterior, pornind de la ordonarea evenimentelor până la o sincronizare totală cu acuratețe de ordinul microsecundelor.

Robustețea: sincronizarea trebuie să fie adaptabilă la condițiile interne ale rețelei, în ciuda dinamicității și a premizelor ce ar putea duce la o partiționare, ruptură a rețelei.

Dimensiunea rețelei prin prisma numărului de noduri cuprinse în procesul de sincronizare: o sincronizare globală este scumpă, consumatoare de resurse și greu de realizat. Se preferă o sincronizare între două sau mai multe noduri. Dar în cazul unei rețele în care datele fuzionează, singura modalitate de obținere a unor rezultate valide o constituie prima variantă.

Ipotezele funcționale de configurare ale rețelelor: sincronizarea timpului trebuie să funcționeze fără cunoașterea în prealabil a infrastructurii.

Protocoloalele studiate în cadrul algoritmilor de sincronizare sunt:

Network Time Protocol (NTP)

Reference Broadcast Synchronization (RBS)

Timing-sync Protocol for Sensor Networks (NTSP)

The Flooding Time Synchronization Protocol (FTSP)

Leightweight Tree-based Synchronization (LTS)

Am ales acești algoritmi deoarece se ocupă în special cu sincronizarea în rețelele radio și rețeaua analizată în special în această teză este de acest fel.

2.3.1. Network Time Protocol (NTP)

Este protocolul utilizat în internet încă din anii '80 care a menținut și menține ceasurile rețelelor într-un sincronism perfect [28]. Protocolul crează o ierarhie pe mai multe niveluri, precum este ilustrată în Fig. 2.3-1.

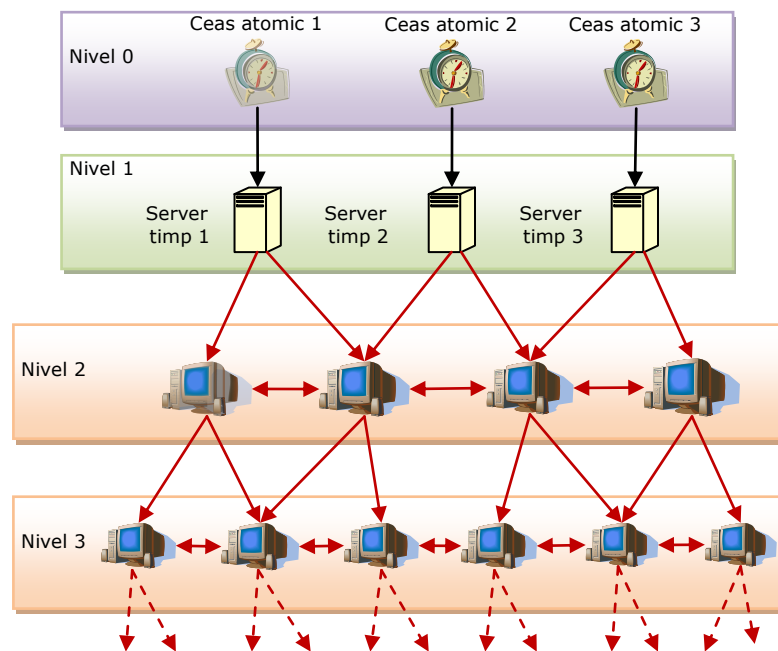


Fig. 2.3-1 Ierarhia protocolului NTP

Nivelul 0, rădăcina ierarhiei, este constituită din dispozitive care măsoară exact timpul: ceasuri atomice, ceasuri GPS sau ceasuri radio. Acest nivel este conectat direct la nivelul inferior care conține servere de timp, vezi Fig. 2.3-1. De la ele pornesc mesaje care conțin informații temporare spre computerele de pe

nivelurile inferioare, care pot ajunge până la 16. Fiecare computer primește informații NTP de la două computere de pe nivelul superior și schimbă mesaje și cu computerele învecinate lui din rețea.

În Fig. 2.3-1 sunt prezentate nivelurile ierarhiei.

Acest protocol realizează a treia formă de sincronizare prezentată în 2.2, în toată rețeaua, în mod continuu, realizând sincronizarea în fază a tuturor ceasurilor din rețea. Precizia oferită de NTP este de ordinul milisecundelor.

2.3.2. Reference Broadcast Synchronization (RBS)

Transmiterea unui pachet în rețea presupune mai multe etape, consumatoare de timp, invizibile de obicei utilizatorului, dar existente. În cazul comunicației radio, pachetele asamblate de CPU sunt transmise antenei transmițătorului și de acolo prin aer (radio) antenei receptorului, după care ajung la CPU-ul acestuia.

În literatură, majoritatea cercetătorilor [20, 28, 29, 30, 31] amintesc de timpul de trimitere al pachetului, timpul de acces, timpul de propagare și timpul de recepție.

Timpul de trimitere (send time) al pachetului, care se află încă la stația care emite, este timpul necesar creării mesajului și direcționării lui spre canalul de acces care emite în rețea.

Când o stație se hotărăște să transmită se planifică un task. La nivelul aplicației se assemblează pachetul și este trecut spre nivelurile inferioare (nivelul MAC), pentru a putea fi predat interfeței care emite în rețea.

Acest timp depinde de sistemul de operare, de încărcarea procesorului, de schimbarea de context, de întârzierile din aplicație; el este nedeterminist și poate varia de la câteva microsecunde până la câteva secunde.

Timpul de acces (access time) reprezintă timpul cât pachetul așteaptă în canalul de transmisie, înainte de a se produce transmisia bit cu bit, propriu zisă. Această întârziere este specifică rețelelor radio, deoarece sunt proiectate având un mediu comun pentru transmisia pachetelor. Este cea mai nedeterministă parte din transmisie și poate varia de la milisecunde la secunde.

Timpul de propagare (propagation time) reprezintă timpul necesar pachetului să ajungă de la transmițător la receptor prin mediu radio, adică de la interfața de rețea a stației care a emis la interfața de rețea a stației care recepționează. Este timpul cel mai determinist și pentru rețele radio care se întind pe maxim 300 de metri este de ordinul nanosecundelor, putând fi considerat neglijabil.

Timpul de primire (receive time) al pachetului este timpul necesar pentru recompunerea mesajului din biții sosiți și predarea lui de la nivelul MAC la nivelul aplicației al receptorului. Are aceeași valoare ca și timpul de trimitere, efectuându-se operațiile în ordine inversă de la interfața de rețea, prin nivelul MAC până la nivelul aplicației.

Avantajul metodei este că se elimină nedeterminismul de la transmițător, așa cum este ilustrat în Fig. 2.3-2, adică timpul de trimitere și timpul de acces, care uneori pot ajunge la valori de secunde nu sunt considerate. Sincronizarea se face doar între receptorii din rețea iar în acest caz întârzierile aleatorii de la transmițător

sunt eliminate. Rămân doar timpul de propagare, care este neglijabil, și timpul de primire.

Forma cea mai simplă a RBS conține un transmițător și doi receptori. Transmițătorul trimite un mesaj de sincronizare celor 2 receptori. În mesaj nu este conținută nici o informație de timp. La sosirea mesajului fiecare stație reține timpul ei local. Apoi cei doi receptori își trimit unul altuia timpul la care au recepționat mesajul de la transmițător și își estimează diferența (offset) ceasurilor.

Algoritmul RBS constă în transmiterea unui mesaj de sincronizare tuturor receptorilor din rețea (broadcast) și poate fi extins în două direcții: prima prin folosirea mai multor receptori, nu doar doi; iar a doua prin trimiterea mai multor mesaje de sincronizare și determinarea la receptori a mai multor timpi locali la primirea mesajului.

În această situație fiecare receptor conține mai mulți timpi locali de la mesajele primite de la transmițător și timpii locali ai tuturor celorlalți receptori. Urmează procesul de sincronizare, al receptorilor prin care fiecare stație calculează diferența (offsetul) dintre ceasul ei și celelalte din sistem și își reglează ceasul local în acord cu informațiile de la celelalte ceasuri, pe care le deține.

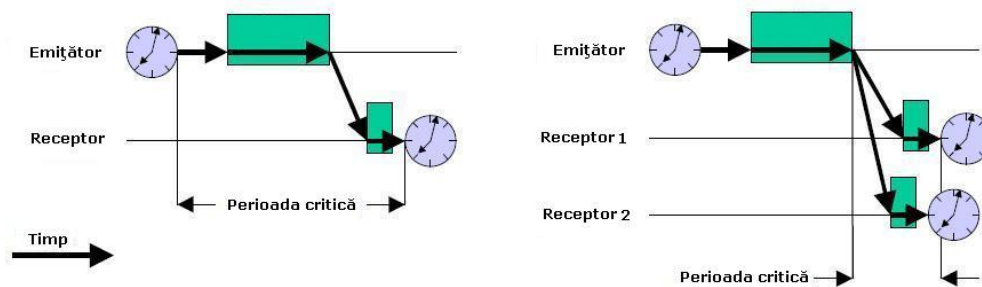


Fig. 2.3-2 Protocolului RBS sincronizează doar receptori

Un receptor are o listă cu una sau mai multe valori de timp locale ale ceasurilor tuturor receptorilor din rețea. Dacă diferența dintre două valori este zero, atunci nodurile sunt sincronizate. Dar, diferența este de obicei diferită de zero iar valoarea constituie eroarea de sincronizare. Având n valori se dorește o valoare aproximativă folosită la ajustarea ceasurilor pentru a deveni sincronizate.

Pentru corectarea acestor tipuri de erori fiecare stație aplică regresia liniară, vezi Fig. 2.3-3 a celor mai mici pătrate pentru a determina frecvența relativă față de ceasurile stațiilor din sistem, necesară în ajustarea ceasului ei.

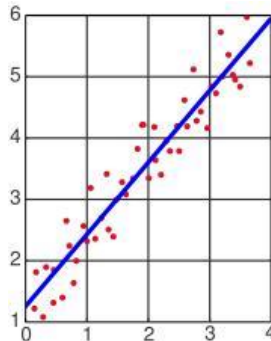


Fig. 2.3-3 Regresia liniară folosită pentru alegerea frecvenței relative a ceasurilor din sistem

Pentru analiza performanței algoritmului RBS se trasează distribuția normală (gaussiană) a timpilor de recepție a perechilor de noduri din rețea [35]. S-a creat o rețea de 6 noduri dintre care unul emite iar ceilalți $n=5$ sunt receptorii care vor fi sincronizați. Nodul emițător a trimis $m=160$ de pachete într-un interval de 3 minute, cu o frecvență de transmitere a pachetelor cuprinsă între 200ms și 2 secunde. Pentru fiecare transmisie s-au reținut 10 valori, reprezentând diferența (offset) timpilor combinațiilor posibile de perechi ale celor 5 noduri. Valorile obținute sunt ilustrate în Fig. 2.3-4.

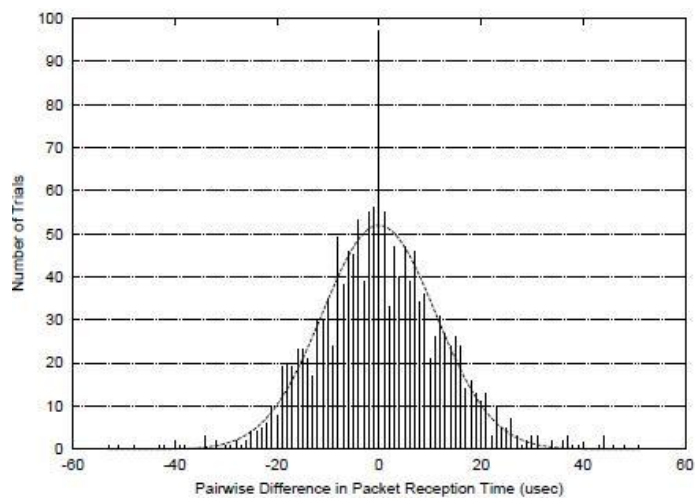


Fig. 2.3-4 Distribuția gaussiană a diferenței temporare a perechilor de noduri dintr-o rețea[35]

Autorii RBS estimează acuratețea protocolului la $11\mu\text{s}$, pe platforma lor de testare, MICA, dar alte teste sugerează o acuratețe de $29.1\mu\text{s}$ [19], într-o rețea cu un singur hop.

În calculul de mai sus frecvența ceasurilor a fost considerată constantă. Acest lucru nu este posibil în practică. Există atât deformarea (skew) cât și decalarea (drift) ceasurilor introduse de frecvența oscilatorului.

În rețea se realizează sincronizarea lucrându-se cu un timp relativ, nu cu unul absolut ca în cazul protocolului anterior, NTP.

2.3.3. The Flooding Time Synchronization Protocol (FTSP)

Este un algoritm care urmărește sincronizarea tuturor ceasurilor nodurilor dintr-o rețea. Pentru aceasta se desemnează o stație, numită transmițător, care folosește baza de timp globală. Ștampila de timp este transmisă într-un mesaj în întreaga rețea (broadcast).

Receptorii rețin timpul de interceptare al mesajului, obținând o pereche de valori (timp global, timp local) care poartă numele de punct de sincronizare. Diferența dintre timpul global și cel local permite estimarea diferenței de ceas (offset) a fiecărui receptor. Are loc sincronizarea nodului local după transmițător.

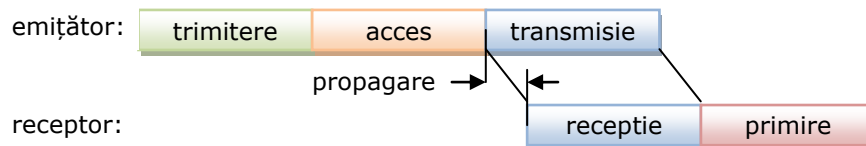


Fig. 2.3-5 Timpii folosiți în algoritmul FTSP [19]

În cazul FTSP, față de timpii definiți până acum, Maroti în [19], adaugă, în plus timpul de transmisie și timpul de recepție.

Prin timp de transmisie (transmission time) se înțelege timpul de când începe transmițătorul emisia și cât durează până transmite prin radio bit cu bit toate datele. Durata depinde de dimensiunea pachetului și de viteza de emisie a canalului.

În Fig. 2.3-5 sunt ilustrați timpii prezentați până acum și interdependența lor.

Timpul de recepție (reception time) reprezintă timpul necesar receptorului să primească datele transmise. Este egal cu timpul de transmisie și de obicei se suprapun, deoarece o dată pornită transmisia de la o stație, după un anumit timp, stația destinatară începe să recepționeze. Acest lucru este evidențiat în Fig. 2.3-5.

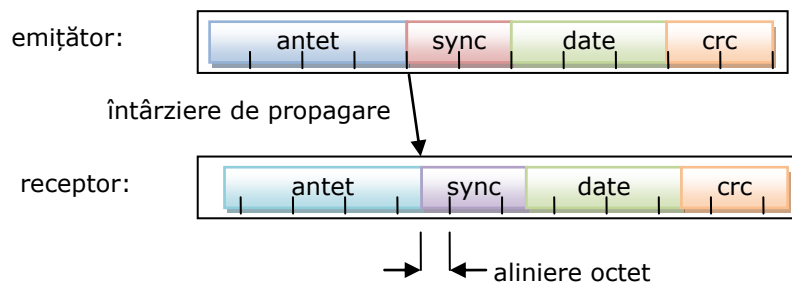


Fig. 2.3-6 Pachete de date transmise în rețea folosind FTSP[19]

În Fig. 2.3-6 este exemplificat un cadru de date trimis de transmițător și forma în care a fost recepționat de receptor. Mesajul transmis începe cu un antet (preambul). Urmează octeții de sincronizare, sync, octeții de date și octeții de CRC.

La primire, receptorul își sincronizează octeții de antet cu frecvența purtătorului său propriu. Pot apărea decalaje, cum se ilustrează în Fig. 2.3-6. Folosind octeții de sincronizare, sync, se poate calcula diferența de biți necesară pentru reconstituirea mesajului având alinierea octeților identică cu sursa emitentă.

Pentru creșterea preciziei se transmit mai multe mesaje cu ștampile de timp între transmițător și receptor. Ștampilele de timp sunt normalizate și prelucrate statistic folosind regresia liniară. Distribuția nu este gaussiană, dar erorile pot fi eliminate alegând minimumul ștampilelor normalizate. În final rămâne o singură ștampilă de timp la transmițător și una la receptor. Ștampila finală este transmisă într-un mesaj în întreaga rețea (broadcast). Eroarea aproximată prin regresie liniară reprezintă valoarea relativă a ștampilelor de timp dintre nodul receptor și transmițător. Mesajul conține ștampila de timp (diferența și distorsionarea) estimată de transmițător în urma schimbului de mesaje cu receptorul.

Această tehnică este folosită nu numai la protocoalele de sincronizare, dar și la determinarea limitelor acustice și la localizarea unei stații care transmite.

Dacă receptorii lucrează cu aceeași frecvență, iar diferența (offset) a ceasurilor tuturor receptorilor este constantă se poate utiliza o singură pereche transmițător-receptor pentru sincronizare, altfel algoritmul propune mai multe perechi transmițător-receptor. În cazul mai multor perechi transmițător-receptor se adună ștampilele de timp și se aplică algoritmul de regresie liniară pentru aproximarea ștampilei de timp.

O problemă ridicată de algoritm constă în alegerea transmițătorului. Astfel se formează o mulțime de noduri candidate (sincronized node) care pot deveni centru de sincronizare (sincronization - root) la un moment dat. Alegerea se face în funcție de identificatorul nodului. Nodul candidat cu cel mai mic identificator devine centru de sincronizare. Dacă la un moment dat centru de sincronizare părăsește rețeaua, alte noduri candidate încep să transmită, dar la primirea mesajelor se vor păstra ștampilele de timp doar ale nodului cu identificatorul minim. În acest fel a fost reales alt centru de sincronizare.

FTSP funcționează și pentru rețele radio multi-hop. Centru de sincronizare transmite mesaje în toată rețeaua. Un nod care nu este în raza lui de acțiune poate să recepționeze de la un nod sincronizat deja, aflat aproape de centru, și care este în raza lui de acțiune.

Autorii protocolului [19], după rezultatele experimentale pe platforma MICA2 la 7.37MHz, susțin că valoarea media a erorii a fost de $0.95\mu\text{s}$, cu o valoare absolută a erorii de $4,32\mu\text{s}$.

2.3.4. Timing-sync Protocol for Sensor Networks (TPSN)

TPSN are drept scop stabilirea unei baze de timp comune pentru fiecare nod din rețeaua radio. Protocolul funcționează în două etape. În prima etapă se crează o structură ierarhică, urmată de sincronizarea perechilor de noduri, care duc la realizarea sincronizării globale. În cele din urmă toate nodurile din rețea vor fi

sincronizate față de un nod de referință, pe care îl vom numi ca în algoritmul precedent, FTSP, centru de sincronizare.

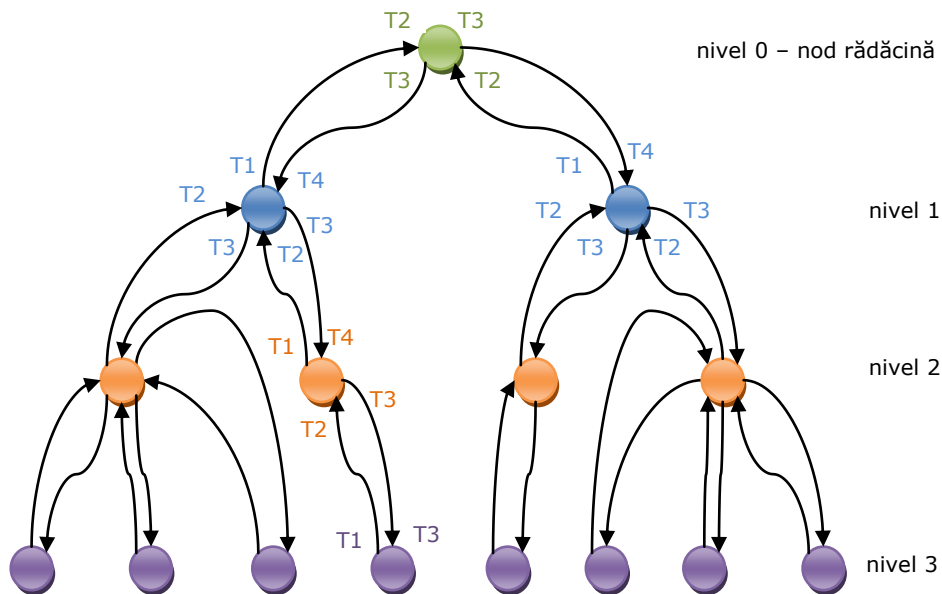


Fig. 2.3-7 Ierarhia protocolului TPSN

La crearea rețelei se construiește și ierarhia nodurilor necesară protocolului.

Se alege un nod ca fiind centru sau rădăcina de sincronizare. Lui i se atribuie nivelul 0 și el dă curs unei „faze de descoperire a nivelului”. Astfel trimite mesaje de tipul *level_discover* care conțin identificatorul nodului și numărul nivelului spre toate nodurile din rețea. La primirea unui mesaj receptorii verifică numărul nivelului din mesaj, îl incrementează, și-l atribuie și transmite mesajul mai departe spre alte noduri furnizând de această dată nivelul lor. Algoritmul asigură comunicarea dintre un nod la nivelul i cu cel puțin un nod aparținând nivelului $i-1$. Dacă un nod a primit pachetul de descoperire al nivelului și și-a stabilit nivelul va ignora toate celelalte pachete de acest fel care ajung la el.

A doua etapă constă în sincronizarea nodurilor. Un nod de pe nivelul i se sincronizează cu nodul de pe nivelul $i-1$. Se realizează sincronizarea între perechi de muchii ale structurii ierarhice ce a fost stabilită la faza anterioară, ilustrată în Fig. 2.3-7, iar în final toate nodurile vor fi sincronizate cu centru sau cu rădăcina.

Sincronizarea a două noduri folosește abordarea clasică a sincronizării dintre transmițător și receptor. În Fig. 2.3-8 este prezentat un schimb de mesaje între nodul A și nodul B . Voi arăta în continuare cum un astfel de schimb de mesaje dintre două noduri poate duce la o sincronizare a acestora. Nodul A se va sincroniza cu nodul B .

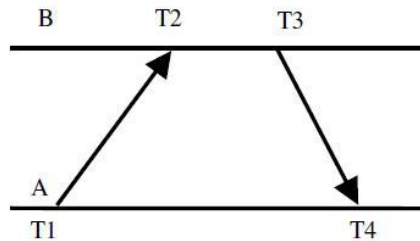


Fig. 2.3-8 Schimb de mesaje între două noduri

Notațiile T_1 , T_4 din figură reprezintă timpul măsurat de ceasul local aparținând nodului A. În mod similar notațiile T_2 și T_3 reprezintă timpul măsurat de ceasul local aparținând nodului B. La momentul T_1 nodul A trimite un pachet de sincronizare ce conține numărul nivelului nodului A și valoarea lui T_1 . Nodul B va recepționa acest pachet la momentul T_2 , unde $T_2 = T_1 + \Delta + d$. În această ecuație Δ reprezintă diferența (offset) de timp dintre ceasurile celor două noduri, iar d reprezintă întârzierea de propagare. La momentul T_3 , nodul B trimite înapoi către A un pachet de confirmare care conține nivelul nodului B, și valorile T_1 , T_2 și T_3 . Nodul A va recepționa acest pachet la momentul T_4 . Presupunând că diferențele dintre ceasuri și întârzierea propagării nu se modifică într-un timp atât de scurt, nodul A va efectua următorul calcul

(2.3-1) pentru a determina valorile Δ , respectiv d :

$$\Delta = \frac{(T_2 - T_1) - (T_4 - T_3)}{2} \quad (2.3-1)$$

$$d = \frac{(T_2 - T_1) + (T_4 - T_3)}{2}$$

Cunoscând diferența dintre ceasuri, nodul A va putea să își modifice propriul ceas astfel încât să se sincronizeze cu nodul B. Aceasta este o metodă prin care transmițătorul se sincronizează după receptor.

Centrul de sincronizare, nodul rădăcină inițiază procesul de sincronizare prin trimiterea unui pachet de tipul `time_sync`. La recepționarea pachetului de către un nod de la nivelul 1, acesta din urmă va aștepta un scurt timp după care va iniția schimbul bidirecțional de mesaje, cum este ilustrat în Fig. 2.3-8, cu nodul rădăcină, care în acest caz este nodul B. În momentul în care se recepționează pachetul de confirmare (*acknowledgement*), nodul A de pe nivelul 1 își ajustează ceasul după cel al nodului rădăcină. Nodurile ce aparțin nivelului 2 vor surprinde acest schimb de informații, acest lucru bazându-se pe faptul că fiecare nod din nivelul 2 are cel puțin un vecin ce aparține nivelului 1. În momentul în care a fost surprins un astfel de schimb de informații de către un nod, acesta din urmă va aștepta un scurt timp, după care începe și el un astfel de schimb de informații cu nodul rădăcină. Durata de așteptare este necesară pentru a se asigura că nodurile din nivelul 1 au reușit să se sincronizeze. La fiecare pachet de sincronizare se răspunde printr-un pachet de confirmare, *acknowledgement*, necesar pentru a împiedica formarea mai multor niveluri de sincronizare în rețea. Procesul este efectuat de fiecare nivel al structurii

ierarhice a rețelei, astfel că în final fiecare nod va fi sincronizat cu nodul rădăcină. Se mai ține cont și de faptul că de-a lungul unei rețele se pot pierde pachete, astfel încât un nod va aștepta un anumit timp, după care, dacă nu a recepționat pachetul de confirmare va proceda la o retransmitere a pachetului de sincronizare, acest proces repetându-se până când se efectuează cu succes un astfel de schimb de mesaje bidirecțional.

NTPS realizează sincronizarea perechilor după modelul transmițător-receptor. Autorii [28] susțin că sincronizarea a două noduri între ele se face în mai puțin de 20μs iar în cel mai rău caz în 50 μs.

2.3.5. Lightweight tree-based synchronization (LTS)

În literatură există o abordare diferită a sincronizării. Greunen și Rabaey în [20] nu urmăresc să mărească precizia ceasurilor locale ci caută o metodă de simplificare a sincronizării. Ei susțin că precizia nu este o constrângere în majoritatea rețelele radio de senzori și propun folosirea unor metode cât mai simple pentru realizarea unei scheme de sincronizare. Urmărind acest scop ei propun doi algoritmi.

Autorii admit o precizie la nivel de fracțiuni de secunda (milisecunde), dar economisesc puterea de calcul a nodurilor și schimbul excesiv de mesaje din rețea. În principiu, ei propun ca sincronizarea să se facă mai rar și la mai puține noduri, nu în toată rețeaua [36].

Primul algoritm este dedicat unei rețele multi-hop centralizate. Structura este asemănătoare cu cea prezentată în TPSN: se crează o topologie având un centru, o rădăcină urmată de sincronizarea bazată pe schimbul de mesaje între perechi de noduri. Autorii [36] observă că odată cu creșterea arborelui se mărește gradul de acumulare al erorilor. La construirea arborelui se urmărește ca acesta să aibă o adâncime minimă (cât mai puține niveluri). Grauen și Rabaey [36] propun pentru crearea arborele metode cum ar fi parcurgerea în lățime (breadth-first order), Distributed depth first search (DDFS) sau algoritmul "Echo". După folosirea lor rezultă un arbore cu adâncime minimă [36].

În Fig. 2.3-9 este exemplificată parcurgerea în lățime unde întotdeauna se verifică nodul rădăcină dat apoi nodurile învecinate lui, și apoi nodurile învecinate ale acestora până când se ajunge la toate nodurile nevizitate.

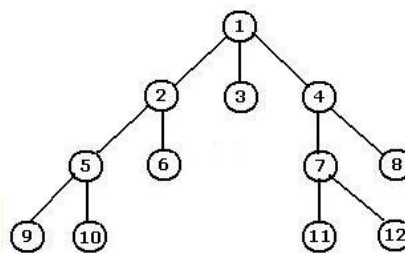


Fig. 2.3-9 Parcurgerea în lățime

Nodul rădăcină are posibilitatea de a iniția o resincronizare, dacă este necesar. După realizarea sincronizării nodurile frunză transmit nodului rădăcină un

mesaj prin care anunță adâncimea ierarhiei, iar această informație este folosită pentru calcularea timpului la care va avea loc resincronizarea.

Algoritmul mai propune o optimizare. Să se realizeze o structură la fiecare nod și dacă apar mai multe cereri de sincronizare să se proceseze ultima sosită, iar celelalte să se ignore.

Față de TPSN care construia arborele o singură dată la crearea rețelei, LTS propune crearea dinamică a ierarhiei înaintea fiecărei resincronizări. Acest lucru mărește robustețea față de dinamica rețelei.

Al doilea algoritm vizează rețelele distribuite și permite unui nod să ceară sincronizarea celui mai apropiat centru de sincronizare care poate să realizeze aceasta. Ierarhia nu este necesară și fiecare nod decide singur când trebuie să ceară o cerere de sincronizare.

În acest fel, toate nodurile, de la nodul de referință până la nodul i care a cerut sincronizarea se sincronizează, realizându-se o sincronizare parțială. Avantajul metodei este că sincronizarea se face la cerere și că scade mult numărul perechilor care efectuează sincronizarea; mai mult se evită crearea unor bucle de sincronizare. Algoritmul propune și agregarea sincronizărilor. Astfel, dacă un nod a cerut o sincronizare, iar un nod adiacent lui face o cerere, se va lărgi sincronizarea pentru a cuprinde ambele noduri emitente.

2.4. Analiza algoritmilor pentru sincronizarea în timp

Pornind de la creionarea protocolului de sincronizare al RDEA din paragraful 2.1 și analizând algoritmi prezentați voi face o analiză cu scopul găsirii unei soluții adecvate cerințelor.

RDEA este o rețea de mică spre medie (maxim 50 noduri). Constrângerile rețelei sunt legate de viteza și transmiterea datelor. Nodurile nu intră în stări de somn (low-power), iar economisirea puterii nu este o constrângere. În schimb, având în vedere cantitatea mare de date preluată de la senzori pot să se ridice probleme de stocare, legate de memoria disponibilă nodurilor. Din acest motiv algoritmi locali DAD-urilor, care folosesc structuri de date voluminoase ar trebui evitați. La fel și schimbul excesiv de mesaje între noduri. Se dorește ori o globalizare a ceasului ori o resincronizare înainte de evenimentele importante.

Folosirea NTP în cadrul RDEA ar însemna crearea de noduri care să primească informații de la un ceas exact, cum ar fi un GPS. Chiar dacă ar exista una sau mai multe stații care conțin GPS trebuie avută în vedere și posibilitatea defectării lor sau a părăsirii rețelei. În acel moment, rețeaua nu ar mai avea posibilitatea captării exacte a timpului.

Unul dintre dezavantajele NTP este că încearcă să determine timpul cât mai precis, iar aceasta implică multe resurse: timp, instrucțiuni de microcontroler, memorie. Nodurile RDEA nu au resurse restrânse, dar nu au fost dimensionate nici din abundență, de aceea trebuie gestionate corespunzător. Trebuie ales optimul dintre precizia oferită de protocol și puterea de calcul folosită pentru aceasta [29].

Protocolul NTP este adecvat pentru rețelele tradiționale. În rețelele radio pot apărea întârzieri de ordinul sutelor de milisecunde datorate nivelului MAC și al întârzierilor de propagare, ilustrate în Fig. 2.1-2, situații care nu sunt întâlnite în

rețelele clasice. Aceasta poartă numele de non-determinism. Un alt aspect îl reprezintă nodurile rețelelor de senzori care sunt mult mai des supuse defectării decât computerele dintr-o rețea.

Anumiți autori [19], critică TPSN considerând că adaugă încărcare în rețea nefolosind transmisii broadcast, dar Srivastava and co. în [28] susțin că prin crearea arborelui prezentat în Fig. 2.3-7 TPSN elimină transmisiile inutile între noduri și oferă o performanță de 2 ori mai bună chiar decât RBS, care realizează sincronizarea doar între receptori.

TPSN deși folosește metoda de sincronizare clasică transmițător-receptor, realizează eșantionarea la nivelul de jos (MAC) al stivei modulului radio, chiar și în timpul recepției și transmisiei datelor. În plus transmite doar două mesaje pentru sincronizarea a două noduri, iar cantitatea de informație este mult mai mică decât în cazul RBS[19]. Doar pentru configurația minimă cu un transmițător și doi receptori, algoritmul RBS trimite 3 mesaje și recepționează 4. RBS este recomandat în general pentru rețele single-hop.

Dezavantajul TPSN este că folosește doar diferența (offset) timpului dintre două noduri. Nu estimează și decalarea (drift). O altă sursă de eroare o poate reprezenta timpul de propagare, așa cum este ilustrat în Fig. 2.3-8. Protocolul presupune că $d = d1 = d2$, unde $d1$ este timpul de propagare al mesajului de la nodul A la nodul B, iar $d2$ este timpul de propagare de la nodul B la nodul A. În practică acești timpi nu sunt întotdeauna egali. Aceste neajunsuri limitează performanțele. Dar, în afară de timpul de propagare, TPSN elimină toate celelalte surse de întârzieri.

FTSP folosește, la fel ca și TPSN eșantionarea la nivelul MAC eliminând o mare parte a întârzierilor. În afară de diferența (offset) timpului tratează și distorsionarea (skew) aplicând regresia liniară. FTSP se bazează pe o structură dinamică actualizându-se continuu nodurile și alegând dinamic centrul de sincronizare al rețelei prin așa zisa politică de alegeri, care de fapt realizează metode de grupare a nodurilor pentru a avea o rază de acțiune cât mai bună.

În concluzie:

NTP furnizează ora exactă dacă are acces la un server de timp sau la un GPS, în schimb necesită resurse importante pentru calcularea timpului. Un ceas local care funcționează cu un tact neregulat necesită corecții continue din partea procesorului în cazul acestui algoritm [34].

FTSP oferă cel mai bun timp dintre toți algoritmii prezentați în lucrare, este potrivit pentru o topologie dinamică de rețea. Dar folosind regresia liniară și metoda dinamică de re alegere a centrului de sincronizare consumă o mare parte din resursele nodurilor. În plus folosirea antetului (preamble) și a biților de sincronizare ale purtătorului (carrier) nu sunt necesari pentru tipul de rețea construit în acest proiect.

TPSN nu este potrivit pentru schimbări dinamice de topologii, deoarece ierarhia este creată o singură dată la crearea rețelei. Furnizează viteză bună de sincronizare, iar acuratețea estimării timpului folosește doar diferența (offset), netratând decalarea (drift) și deformarea (skew).

LTS are o acuratețe a sincronizării la nivel de milisecundă, în schimb propune tehnici simple de realizare a sincronizării. Propune resincronizarea la cerere, sau sincronizarea unui grup de noduri. Este un protocol dinamic, care înainte

de fiecare sincronizare realizează o structură ierarhică de adâncime minimă, ceea ce duce la un mare consum de resurse.

Pornind de la aceste premize vom defini în capitolele următoare un protocol de sincronizare potrivit pentru RDEA.

2.5. Sistem adaptiv

Un sistem încapsulat (embedded) este considerat adaptiv dacă își modifică funcționalitatea în deplin acord cu schimbările din mediul înconjurător menținându-și cu toate acestea randamentul constant [3]. Există grade și limite diferite ale adaptabilității, dar în momentul de față nu putem spune că s-a construit un sistem perfect adaptabil [33]. În general aceste sisteme au limitări și constrângeri de funcționalitate.

RDEA interacționează cu mediul înconjurător. Sensorii măsoară valori din natură (temperaturi, vibrații, tensiuni). Rețeaua acționează ca o colectivitate: nu este importantă informația de la un anumit senzor, dar este important evenimentul raportat, el fiind agregarea de la mai mulți senzori [4]. În cazul în care anumite noduri se strică, rețeaua trebuie să se folosească de restul stațiilor pentru a reuși să-și îndeplinească sarcina [32].

RDEA poate fi considerată un sistem cu cerințe adaptive: în funcție de procesele care au loc pe parcursul exploatării sistemului de dimensiuni mari este necesară ajustarea ratei de eșantionare în mod dinamic. O altă cerință este legată de sincronizare. Nodul care realizează sincronizarea în rețea se poate defecta sau poate părăsi rețeaua, dar rețeaua trebuie să funcționeze mai departe. Trebuie desemnat dinamic un nod care să dicteze sincronizarea.

Această caracteristică poartă numele de auto-reparare, auto-modificare sau auto-configurare, dată de diferiți autori [4, 27].

Proiectarea RDEA ar trebui să încorporeze adaptabilitatea la schimbările de mediu și de configurație. Rețeaua trebuie să funcționeze la parametri stabiliți indiferent de modificările dinamice din jurul ei. Putem spune că RDEA este o rețea radio adaptivă sau o rețea adaptivă de senzori [8].

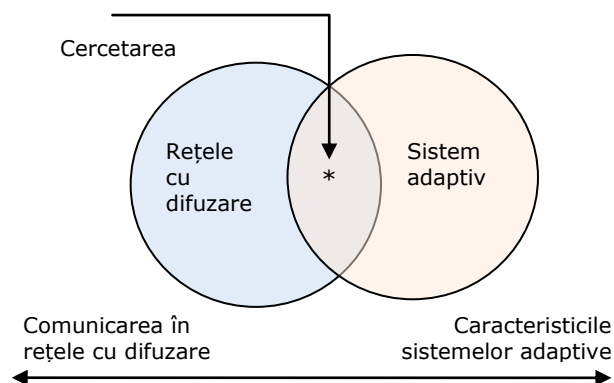


Fig. 2.5-1 Plasarea cercetării în cadrul științei

Cercetarea realizată de mine poate fi plasată la intersecția dintre cerințele comunicației într-o rețea cu difuzare și ipotezele realizării unui sistem adaptiv. Această încadrare este exemplificată în Fig. 2.5-1.

În rețea există noduri cu capacitate de calcul, sensibilitate și putere de analiză ridicată. Tendința actuală, pe care o urmăresc și eu în cadrul dezvoltării metodei, este de a reuși să se poată face diferența dintre o perturbare și un eveniment care trebuie monitorizat. Doar în caz de eveniment, datele corespunzătoare evenimentului sunt transmise mai departe sau se mărește perioada de eșantionare pentru detalierea informațiilor. Nodurile transmit doar datele cerute sau rezultate parțiale, mediate [4].

2.6. Metrici

La implementarea unui sistem încapsulat care funcționează în timp real se știe că acesta interacționează cu alte sisteme, dar și cu mediul înconjurător. Pentru succesul design-ului trebuie urmărit ca atât componentele hardware cât și cele software să interacționeze continuu și armonios. Ținta design-ului este ca sistemul realizat să furnizeze servicii de o calitate garantată, nu fluctuantă [84].

O metrică este o unitate de măsură, sau un parametru al unui sistem care este periodic investigat și permite prin analiza sa interpretarea funcționării sistemului. Primul pas pentru definirea unui cadru în care se aplică metrici care descriu performanța comunicării în Internet l-a făcut Paxson [5]. Urmându-i exemplul alți cercetători au stabilit cadre de aplicare a metricilor pentru aplicațiile sau proiectele lor. Neil și Fenton în [2], împart subiectul în două componente și anume definirea metricilor și metodologia, adică modul de colectare a lor. Este important modul în care se face măsurătoarea, din acest motiv, se pot trasa câteva linii generale [10].

Metodologia pentru determinarea unei metrici trebuie să aibă proprietatea de a fi repetabilă: dacă aceeași metodologie este folosită de mai multe ori în aceleași condiții, rezultatele măsurătorilor trebuie să fie identice. Cu toate acestea, chiar și cele mai bune metodologii de măsurare aplicate celor mai riguroase metrici vor crea erori.

Pentru aceasta, în dezvoltarea metodologiilor de măsurare trebuie să ne străduim să:

- minimizăm sursele de incertitudini și erori
- conștientizăm sursa incertitudinilor și a erorilor
- să măsurăm cantitatea de erori

Neil și Fenton creează un cadru general și studiază metrici generale ale software-ului, definite încă din anii '60 și ajung la concluzia că acestea urmăresc două direcții: evaluarea sau predicția. În lucrarea lor fac, propun o analiză a dezvoltării proceselor software bazată pe raportul efort/cost [2].

Având în vedere avântul tehnologic, sistemul proiectat trebuie să fie implementat într-o perioadă de timp rezonabilă, altfel soluția va fi învechită și nu va mai fi de actualitate. Sistemelor în timp real de cele mai multe ori li se cere să fie adaptive, reconfigurabile, performante și fiabile [84]. Aceste caracteristici le-am urmărit și eu în această teză, în cadrul design-ului dezvoltând caracteristicile

(feature) de adaptabilitate și reconfigurare urmate de analize ale implementării bazate pe ultimele două criterii.

Scopul cercetării mele a fost analiza rețelelor de dispozitive de achiziția datelor prin prisma performanță-fiabilitate, am căutat în literatură metrici care să illustreze această latură, studiind atât rețele clasice, cât și rețele radio. Negăsind nici o lucrare care să trateze această analiză m-am concentrat la abordarea individuală a celor două metrici.

În lucrările studiate de mine, cercetătorii care au analizat performanța rețelelor au urmărit:

- Mărimea cadrului de date, raportul dintre timpul de așteptare și timpul de recepție al datelor, capacitatea buffer-ului intern al nodului și modul de tratare al erorilor datorate zgomotelor. Rezultatele indică o performanță mărită pentru rețelele care funcționează dinamic [16].
- Puterea de acoperire a semnalului (coverage), rata de modulație și redundanța semnalului necesară unei anumite arii de acoperire[17].
- Timpul de viață al bateriei și constrângerile legate de tensiunea nodurilor din rețea [22].
- Analiza timpului de răspuns (response time) în LAN-uri cu aplicabilitate în industrie [23].

Studiind fiabilitatea cercetătorii au propus în general metrici noi, specifice domeniului activității lor de cercetare, care ilustrează adecvat caracteristicile rețelelor:

- s-au analizat și comparat metrici pentru găsirea căilor de comunicație (routing) în rețele cu mai multe hopuri prin studierea metricilor legate de calitatea legăturii (LQI) [9].
- s-a descris și implementat metrica ETX (Expected Transmission Count) în scopul găsirii căilor de comunicație optime într-o rețea multi-hop (routing) [18].
- s-a creat o metrică proprie, numită metrica legăturii care conține mai mulți parametri dintre care amintesc: ETX, ETT (Expected Transmission Time), WCETT(Weighted cumulative ETT), RTT (Routing Trip Time). Această metrică a fost analizată în două rețele de tip mesh(plasă) care au fost implementate pe diferite platforme și s-au studiat mecanismele de găsirea căilor de comunicație (routing)[21].
- s-au analizat metodele de grupare a nodurilor (clustering) pentru rețele radio cu aplicabilitate în industrie[24].
- s-a propus găsirea căilor de comunicație (routing) în funcție de importanța datelor transportate. Datele importante, folosite în aplicații cum ar fi prevenirea dezastrelor naturale sau aplicații militare, sunt transmise folosind o calitate superioară a serviciului (QoS) de transmitere față de datele obișnuite. Diferența constă nu numai în calitate, cât și în preț, primele fiind mai scumpe. [25]
- se propun arhitecturi de rețele radio care să conțină noduri redundante pentru menținerea disponibilității (availability) și a funcționării corecte [26]
- rețelele radio ad-hoc de senzori sunt importante dacă le privim ca veriga dintre lumea fizică și Internet. Din acest motiv multe cercetări tratează acest subiect. [77] prezintă un studiu referitor la calcularea puterii de acoperire, de expunere a unei rețele. Calculele pot fi folosite cu succes la stabilirea performanței rețelelor și la determinarea căilor de comunicație (routing).

Pornind de la aceste studii am definit metrice de performanță și fiabilitate adecvate RDEA proiectate de mine, pe care le prezint în continuare.

2.7. Metrice de performanță

- Mărimea optimă a pachetului de date
- Variația vitezei în funcție de numărul de noduri din rețea
- Lățimea de bandă folosită la transmisia datelor
- Dimensiunea buffer-ului intern al unui nod din rețea
- Timpul minim admis între două avarii

2.7.1. Mărimea optimă a pachetului de date

Eșantioanele citite de la senzori sunt grupate, pentru transmisie, mai multe într-un pachet. Mărimea unui pachet de date este unul dintre parametri studiați și analizați în această teză.

Formula (2.7-1) exprimă timpul de transfer al datelor.

$$\text{Timpul_transfer} = (\text{dim_header} + \text{dim_date}) \times \text{viteza_transmisie} \quad (2.7-1)$$

Urmăresc determinarea mărimii optime a pachetului.

Un pachet de date conține pe lângă datele propriu-zise, anumite informații de configurare grupate în header-ul pachetului. Pentru determinarea vitezei de transfer a datelor este nevoie de cantitatea de octeți din pachet transmisă la viteza pe care o furnizează dispozitivul, numită viteza de transmisie. Viteza de transmisiie este exprimată în octeți/sec sau octeți/msec.

Se pune problema frecvenței transmisiei. Vreau să studiez cum este optimă transmisia: pachete de dimensiuni mari transmise mai rar sau pachete de dimensiuni mici transmise des. Timpul de transfer are aceeași valoare, dar diferă valoarea dimensiunii pachetului și a vitezei de transmisie variază.

2.7.2. Variația vitezei în funcție de numărul de noduri din rețea

RDEA are o structură prestabilită, adică un număr cunoscut de noduri. Am studiat cum variază viteza de transmisie în cazul adăugării de noi noduri în rețea, tratând în mod special nodurile care generează un volum mare de date.

2.7.3. Variația parametrilor în funcție de lățimea de bandă din rețea

Între componentele fiecărui DAD comunicația se realizează folosind protocolul RS-232 [62].

În cadrul comunicației seriale RS-232 trebuie specificați patru parametri: baudrate-ul folosit pentru transmisie, numărul de biți care codifică caracterul, valoarea opțională a bitului de paritate și numărul de biți de stop folosiți.

În aplicația mea am lucrat cu baud rate variabil (9600, 19200, 38400, 57600, 11520), codificarea caracterelor a fost făcută pe 8 biți, nu s-au folosit biți de paritate și s-a adăugat un singur bit de stop.

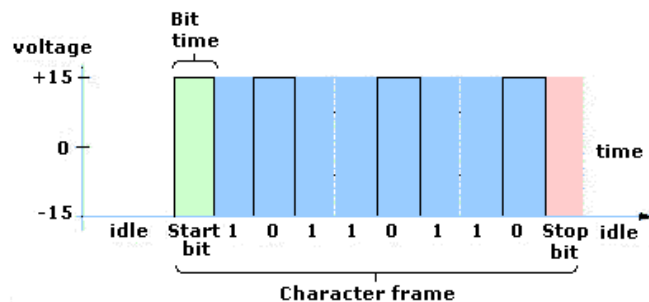


Fig. 2.7-1 Cadru atașat unui caracter transmis

În acest fel, fiecare caracter este împachetat într-un cadru al caracterului, ilustrat în Fig. 2.7-1, care începe cu un bit de start, urmează 8 biți care codifică caracterul și un bit de stop.

Rata de transmisie (baud rate) măsoară numărul maxim de caractere care circulă între două componente sau dispozitive care folosesc protocolul RS-232.

Rata de transmitere (baudrate) măsoară cât de repede se mișcă datele între dispozitive care folosesc comunicația serială. RS-232 folosește doar două niveluri de tensiune, +15V și -15V, reprezentate în Fig. 2.7-1. În această schemă de configurare pe două niveluri, rata de transmitere, baudrate, coincide cu numărul maxim de biți de informație, inclusiv biții de „control”, care sunt transmiși. Valoarea -15V reprezintă tensiunea negativă și +15V cea pozitivă, Fig. 2.7-1 indicând un semnal idealizat captat de la osciloscop.

Tensiunile negative reprezintă valori de 1, iar cele pozitive valori de 0. În cazul transmisiei RS-232 se transmite folosind logica inversată, și începând de la cel mai puțin semnificativ bit (LSB) până la cel mai semnificativ (MSB).

Bitul de start este folosit pentru marcarea începerii unui nou cadru care conține un caracter și reprezintă tranziția de la tensiunea negativă (-15V) la cea pozitivă (+15V).

Biții de stop pot fi 1, 1,5 sau 2. În aplicația mea am folosit un singur bit de stop, iar scopul lui este de a aduce linia la valori negative(-15V) și a o pregăti pentru o nouă transmisie.

În afară de modul de codificare al unui caracter interesează viteza maximă de transfer a datelor. Pentru acesta se pot aplica formulele lui Nyquist și Shannon [1].

Nyquist a demonstrat că există o limită suficientă de eșantionare a datelor, care dacă este folosită, semnalul de la care provin eșantioanele poate fi reconstituit în întregime.

Astfel, având un semnal cu N niveluri discrete care este transmis printr-un filtru de frecvențe joase având lățimea de bandă de B Hz, viteza maximă de transfer a datelor va fi dată de formula lui Nyquist (2.7.3-1):

$$\text{Viteza_maximă_transfer_date} = 2B \log_2 N \text{ biți/s} \quad (2.7.3-1)$$

$N = 2$, deoarece se lucrează cu valori binare

$B = 10\text{MHz}$, lățimea de bandă

$$\text{Viteza_maximă_transfer_date} = 20\text{M biți/s} = 20 \times 10^6 \text{ biți/s}$$

Formula lui Nyquist tratează un canal ideal, fără perturbații.

Din acest motiv, Shannon a propus o formulă care ține seama de zgomotul care intervine în transmisie, (2.7.3-2).

$$\text{Număr_maxim_biți/s} = B \log_2(1 + S/M) \quad (2.7.3-2)$$

$B = 10 \text{ MHz}$, lățimea de bandă

S/M – raportul semnal/zgomot. Se măsoară în dB. Considerând acest raport egal cu 1, determinăm

$$\text{Număr_maxim_biți/s} = 10\text{M biți/s} = 10 \times 10^6 \text{ biți/s}$$

Formulele lui Nyquist și Shannon se referă la o limitare superioară, pe care sistemele reale rareori o ating.

Voi determina viteza maximă reală de transfer a datelor.

În documentația tehnică pentru RS-232, la o rată de transmisie de 9600 baud, durata transmisiei primului bit și a fiecărui bit din secvența de biți durează în jur de 0.104 ms. Astfel, pentru un baud rate de 9600, un caracter se va transmite în aproximativ 1,046ms [62].

Voi prezenta calculul vitezei pentru un pachet de date având lungimea de 48 de octeți, calculând numărul de pachete transmise pe secundă în funcție de baud rate.

$$\text{Viteza_maximă_pachete/s} = \text{Baudrate} / (\text{Start_bit} + \text{Nr_biti} + \text{Parity_Bit} + \text{Stop_Bits})$$

$$\text{Viteza_maximă_pachete/s} = \text{Baudrate} / (1+8+0+1)$$

$$\text{Viteza_maximă_pachete/s} = \text{Baudrate} / 10 \quad (2.7.3-3)$$

Pentru $\text{Baudrate} = 19200 \text{ biți/sec}$

$$19200 / 10 = 1920 \text{ frame_bit / sec}$$

$$1920 / 48 \text{ octeti} = 40 \text{ pachete de lungime 48 octeti/ sec}$$

În cadrul transmisiei se urmărește o viteză cât mai mare posibilă. Din acest motiv am ales $\text{Baudrate} = 115200 \text{ biți/sec}$

În acest caz, $115200 / 10 = 11520$ frame_bit / sec

Pentru un pachet de date având mărimea de 48 octeți se transmit

$11520 / 48 = 240$ de pachete / s

Într-o secundă se transmit 240 de pachete de lungimea de 48 de octeți, la o rata de transmisie (baud rate) de 115200. Astfel se transmit 2,4 pachete de date la fiecare 10ms.

Știm că, în sistemele reale, în cazul transmisiei apar întârzieri datorate rețelei (network latency). Aceste întârzieri vor trebui analizate și luate în calcul. În această situație, în cadrul aplicației se va verifica dacă se poate realiza transmisia conform vitezelor calculate mai sus. Se va analiza influența întârzierii rețelei (network latency). Ținând cont de întârzierea rețelei, se vor propune soluții, cum ar fi transmiterea mai multor eșantioane grupate într-un pachet.

2.7.4. Dimensiunea buffer-ului intern al unui nod de rețea

DAD-urile citesc datele de senzori la frecvența furnizată de aceștia. Aceste date sunt stocate într-un buffer intern circular, unde li se face o analiză preliminară. În cadrul analizei se face deosebirea între fenomen și perturbație și se stabilește frecvența de transmitere a datelor în rețea.



Fig. 2.7-2 Eșantionarea și citirea datelor de la senzori

În Fig. 2.7-2 este reprezentat buffer-ul intern în care sunt stocate datele eșantionate de la dispozitivele de achiziție. Toate dreptunghiurile conțin date, care sunt analizate de fiecare DAD. Datele din acest buffer intern sunt citite la diverse intervale de timp. În Fig. 2.7-2 consider două intervale de timp, unul colorat cu galben (gri deschis) și altul cu verde (gri închis), dar pot exista mai multe. Astfel se consideră că într-un regim de funcționare normal datele se citesc mai rar, dreptunghiurile galbene, iar în situații de alertă frecvența citirii se mărește, lucru ilustrat prin dreptunghiurile verzi.

Având în vedere că se analizează mai multe eșantioane, valorile colectate pot să reprezinte medieri ale datelor brute, citite.

În cadrul analizei, în funcție de senzorii aleși, de perioada lor de eșantionare, de volumul de date citit trebuie să se analizeze dacă DAD-urile au nevoie de un suport de memorare suplimentar, pentru a nu se pierde eșantioanele generate.

2.8. Metrici de fiabilitate

- Rata de succes
- Numărul de retransmisii
- Apariția zgomotelor și tratarea lor
- Pachetele cu erori
- Pachete pierdute
- Raza de acțiune

2.8.1. Determinarea ratei de succes SR (succes ratio)

Procentul de succes (delivery ratio) este o metrică utilizată pentru determinarea eficienței unui serviciu furnizat de o rețea, cum ar fi transmiterea unui pachet. Această mărime urmărește costul datelor transmise (SendPachet) ținând seama și de numărul pachetelor retransmise.

Metoda de determinare este euristică [6] : La fiecare testare pentru fiecare nod din rețea se însumează numărul total de pachete recepționate și numărul total de pachete retransmise în rețea.

Se calculează rata de succes (SR) folosind expresia

$$SR = \frac{\text{Număr_total_pachete_transmise}}{\text{Număr_total_pachete_primate}}$$

Fig. 2.8-1 Rata de succes în transmisie (Succes Rate)

Pentru determinarea unui procent cât mai real se recomandă realizarea mai multor testări și furnizarea unei valori medii pentru SR.

2.8.2. Determinarea numărului de retransmisii în funcție de pachetele pierdute sau cu erori

Comunicarea în rețea poate produce pierderea pachetelor, datorită supraîncărcării transmisiei. O altă problemă o reprezintă apariția zgomotelor care pot altera informația din cadrul unui pachet. Coliziunile provoacă același efect. Alterarea conținutului pachetului este detectată de receptor prin informații cum ar fi mărimea pachetului sau suma de control (CRC). În acest caz se trimite înapoi emițătorului un cadru în care specifică, prin not_ack, că a primit un pachet care conține erori sau că nu a primit nici un pachet.

Se impune retransmisia pachetului.

Există mai multe moduri de abordare. Unul ar fi să se realizeze retransmiterea cadrului pierdut sau a celui alterat până ajunge corect și confirmarea va fi pozitivă, ack. În acest caz retransmiterea pachetului se poate face și de câteva zeci de ori.

O altă soluție ar fi acceptarea unui procent de pachete pierdute sau eronate, de exemplu 5%, iar pentru realizarea acestui deziderat trebuie analizat și furnizat un număr optim de retransmisii. Această analiză se va efectua în cadrul benchmark-ului.

2.9. Concluzii

În cadrul sistemelor încapsulate în timp real, cel mai important rol îl joacă timpul. Adăugarea constrângerilor temporale poate să transforme programarea în timp real într-un subiect dificil și greu de realizat. Cerința și propunerile de sisteme încapsulate în timp real devine tot mai pretențioasă, iar acestea trebuie să-și

garanteze fiabil funcționarea evitând dezastrele care pot apărea în cazul defectiunilor[75].

Se creionează, exact cum este descris și în [64], o linie diferită de dezvoltare și implementare a sistemelor încapsulate în comparație cu sistemele software clasice, uzuale. Această linie e determinată de caracteristicile sistemelor încapsulate în cadrul cărora interacționează atât componente hardware cât și aplicații (software). Majoritatea sistemelor interacționează cu mediul înconjurător. Timpul reprezintă o constrângere, care uneori este stringentă (hard real-time constrain). În majoritatea cazurilor la sistemele mari apare necesitatea aplicării algoritmilor de sincronizare. O altă diferență față de sistemele „obișnuite”, care este impusă celor „încapsulate”, o constituie fiabilitatea. Dacă sistemul eșuează, acest lucru poate fi fatal. Dar cu toate acestea trebuie avute în vedere cerințe de fiabilitate rezonabile, în funcție de cerințele sistemului. Nu rentează să punem o alarmă de 2000\$ pe o Dacie Logan.

S-a dezvoltat o întreagă disciplină care vizează aceste sisteme, cu formalisme sofisticate, tool-uri de verificare și validare, cadre de lucru (framework), medii de dezvoltare (middleware). S-au scris foarte multe cărți care realizează demonstrații laborioase, dar nu reușesc nici să ia în considerare, dărmite să mai demonstreze, toate aspectele legate de programarea în timp real. Judecând după gradul redus de aplicabilitate în practică, Wirth are mari îndoieli asupra eficienței tuturor acestor metode și formalisme sofisticate în lumea reală unde este necesară o putere mare de calcul. Wirth nu pornește o cruciadă împotriva fundamentărilor teoretice sau asupra tool-urilor eficiente, ci din experiența lui, propune soluții simple, îndeamnă la reducerea complexității. El oferă câteva sfaturi practice. În cazul folosirii întreruperilor, durata rutinei de tratare trebuie să fie mai mică decât orice constrângere de timp real. O altă sursă de erori o constituie definirea interfeței dintre sistem și mediul înconjurător. De multe ori, și aici intervin diverse standarde, interfața este prea stufoasă generând erori sau neînțelegeri. Interfața poate fi o sursă de erori, extrem de subtilă, care duce la neînțelegerea pe deplin a problemei. Există cazuri de erori care nu apar din funcționarea greșită a programului, ci datorită neînțelegerii cadrului de desfășurare al fenomenului. Sistemele încapsulate conțin părți mecanice, electrice și programe. Proiectantul unui astfel de sistem trebuie să fie și inginer mecanic și electric și calculatorist, toate într-un singur om pentru a înțelege, pe de o parte fenomenul, și pentru a-l putea implementa corect[75].

Cu cât teoria și implementarea sunt mai simple acestea denotă perspicacitatea inginerului.

Urmărind problematica specifică rețelelor formate din dispozitive de monitorizare și achiziția datelor, în vederea aplicării eficiente a acestora, atât prin prisma performanței cât și a fiabilității la excavatoare cu cupe, a fost analizată eșantionarea semnalelor culese în formă analogică de la senzori.

Sunt puse în relief modalitățile de încapsulare în pachete care sunt incluse într-un protocol de comunicare cu toate aspectele specifice importante prin prisma prelucrării ulterioare din teză în vederea dezideratului urmărit, al unei configurări optimale cu dominantă performanță-fiabilitate.

Astfel sunt analizate întârzierea de propagare precum și efectuarea sincronizării în timp, fiind stabilite constrângerile referitoare la decalarea impulsurilor de ceas (clock), întrucât unul dintre factorii critici este constituit de stabilitatea frecvenței la rândul ei aflată sub influența decisivă a factorilor reprezentați de temperatura mediului de exploatare și de valoarea tensiunii de alimentare.

Țintind aceleași deziderate primare ale tezei, au fost analizate 5 categorii de algoritmi de sincronizare în timp, fiind puși în relief parametri cum ar fi precizia, robustețea, dimensiunea rețelelor precum și ipotezele funcționale de configurare a rețelelor.

Luând în considerație referirile la aspectele de performanță și fiabilitate a mai multor surse de analiză din literatura de specialitate, am selectat parametri decisivi constând din : timpul de trimitere (send time), timpul de acces (access time), timpul de propagare (propagation time) și de receptare (receive time) care au fost urmăriți cu precădere de algoritmi Reference Broadcast Synchronization(RBS) și Flooding Time Synchronization Protocol (FTSP).

Cu precădere au fost focalizate elementele menite a asigura fiabilitatea transmisiei împreună cu cele care măresc performanța.

În acest capitol sunt analizate metricile de performanță și de fiabilitate prin prisma cărora sunt efectuate în cadrul lucrării evaluările din următoarele capitole ale lucrării.

În ceea ce privește performanța am analizat dimensiunile rețelei, respectiv a pachetelor de date, precum și lățimea de bandă și dimensiunea buffer-ului intern al unui nod de rețea.

În ceea ce privește fiabilitatea referirile au fost făcute la determinarea ratei de succes, numărul de retransmisii precum și tipul zgomotelor care pot afecta transmisia și modul de combatere al acestora. Sunt analizate pachetele eronate și cele pierdute.

3. METODE DE COMUNICARE ÎN REȚELE DE DISPOZITIVE DE ACHIZIȚIA DATELOR

Comunicarea în rețea se realizează pe mai multe niveluri, prezentate în teză în comparație cu modelul de referință OSI. În acest capitol este detaliată comunicarea realizată la nivel aplicație, și comunicarea la nivel rețea. Comunicarea la nivelurile inferioare: legătură de date și fizic este în general standardizată.

3.1. Nivelul aplicație

În cadrul nivelului aplicație sunt oferite servicii care gestionează protocolul de comunicație. Deoarece RDEA funcționează dinamic, în timp real, prezentând caracteristicile unui sistem adaptiv, având constrângeri de eșantionare simultană a datelor la nivel de milisecundă nu s-a putut folosi nici o aplicație existentă. Din acest motiv am construit un protocol la nivel înalt care să realizeze constrângerile date. În cadrul protocolului de comunicație am realizat sincronizarea nodurilor din rețea.

3.2. Definire cadre aplicație

La baza protocolului de comunicație stă definirea cadrelor, care sunt de mai multe tipuri:

- cadru de date
- cadru de confirmare
- cerere de sincronizare
- răspuns la cererea de sincronizare

Cadrelor de date nu au lungime fixă dar au o lungime maximă admisă și sunt folosite pentru transmiterea datelor în rețea. Eșantioanele din buffer-ul intern al unui nod al rețelei sunt grupate și transmise în rețea.

Structura unui cadru de date conține câmpurile: lungime cadru Length, identificator cadru FrameId, tipul cadrului FrameType, ștampila de timp TimeStamp, identificatorul senzorului SensorId, rata de eșantionare Rate, numărul de eșantioane SamplesNo și eșantioanele Sensor Data Frame. În Fig. 3.2-1 este reprezentat un cadru de date, unde câmpurile sunt grupate pe octeți.

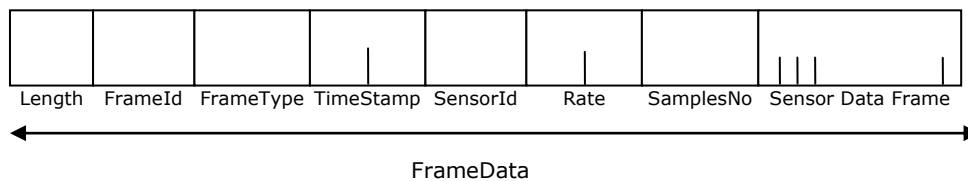


Fig. 3.2-1 Cadrul de date

Lungimea cadrului, Length, reprezintă numărul total de octeți ai cadrului de date.

Identificatorul cadrului, FrameId, este folosit pentru autentificare și necesar pentru situațiile de duplicat, pierdere sau retransmisii ale pachetelor.

Tipul cadrului, FrameType, face diferența între tipurile de cadre, enumerate mai sus.

Ștampila de timp, TimeStamp, reține momentul la care s-au citit eșantioanele. Acesta trebuie să conțină data și ora cu acuratețe de milisecundă.

Identificatorul senzorului, SensorId, furnizează identitatea și tipul senzorului de la care se citesc datele. În RDEA sunt mai multe tipuri de senzori. Un DAD monitorizează mai mulți senzori identici sau diferiți. În funcție de tipul lor se cunoaște mărimea eșantioanelor, limitele valorilor normale și alte caracteristici. Acest câmp este folosit la analiza datelor.

Rata de eșantionare, Rate, este necesară deoarece la achiziția datelor se folosesc rate de citire diferite.

Numărul eșantioanelor, Samples number, este o valoare necesară pentru verificare corectitudinii datelor.

Cea mai mare parte a datelor o reprezintă eșantioanele de date, care sunt grupate într-un cadru. În funcție de tipul senzorului eșantioanele pot să ocupe spațiu diferit: de la un octet în cazul turației roții sau al tensiunii în trolu până la 42 de octeți în cazul unei poziții GPS, vezi Tab. 4.2-1.

Din punctul de vedere al performanței este avantajos să se transmită mai multe eșantioane într-un cadru. Se citesc mai multe eșantioane, și se transmit deodată. Numărul eșantioanelor trimise este dat de câmpul Samples no.

Majoritatea cadrelor transmise în RDEA, sunt cadre de date. Numărul optim de eșantioane care trebuie conținut într-un cadru va fi studiat ulterior.

Cadrul de confirmare al primirii pachetelor dă informații despre starea pachetului: dacă a ajuns la destinație și nu conține erori, dacă a ajuns dar e alterat sau dacă s-a pierdut. În cazul alterării sau a pierderii pachetului cadrul trebuie retransmis.

Cadrul de sincronizare conține informații necesare sincronizării nodurilor din rețea, implicând un dialog între două noduri. Cu ajutorul acestui cadru se cere timpul necesar sincronizării de la un centru de sincronizare.

Cadrul de răspuns la sincronizare furnizează timpul de referință al nodului sursă.

Cadrelor de sincronizare și cadrul de răspuns la sincronizare sunt folosite de algoritmi de sincronizare a nodurilor din rețea.

3.3. Descriere algoritmi de comunicare

În cadrul RDEA se pot realiza conexiuni punct-la-punct, sau punct-la-multi-punct sau peer-to-peer. Serviciile oferite sunt garantate prin folosirea confirmărilor (ack) și a politicii de retransmisia datelor. Pentru evitarea încărcării transmisiei se folosesc mecanisme de timeout, care fluidizează comunicația.

Sampling_task reprezintă operația efectuată la nivelul nodului, al DAD-ului. Eșantioanele achiziționate de la senzori sunt stocate în buffer-ul intern al DAD-ului, cum am explicat în paragraful 2.7.4. Task-ul de sampling crează ștampila de timp, realizează analiza preliminară, în urma căreia decide rata de citire a datelor din buffer-ul intern și grupează mai multe eșantioane pe care le furnizează buffer-ului de transmisie.

În continuare prezint pseudocodul pentru sampling_task.

```

Sampling_task
While (1)
    Daca exista cererea de sincronizare (RTC_sincronized == TRUE)
        Citește timpul (Time = RTC_GetTime)
        Citește eșantionul (GetSample)
        Crește număr_eșantioane (Samples no++)
        Analiză_eșantioane
        Dacă există condiție de alertă
            Dacă se modifică rata
                // cazul de alertă, se mărește rata
            (if CurrentSample mod Configuration[AlertSamplesRate]==0)
                Stabilește rata citire regim alertă
                Mută eșantioanele în buffer-ul de transmisie
                Altfel
                    Dacă se modifică rata
                        // revenirea din alertă la normal
                    (if CurrentSample mod
Configuration[NormalSamplesRate]==0)
                        Stabilește rata citire
                        Mută eșantioanele în buffer-ul de transmisie
                        Δ //end dacă există condiție de alertă
                        Δ //end dacă există cerere de sincronizare
                    Δ //end while

```

Task_transmisie descrie în pseudocod crearea pachetelor de date, prin alcătuirea header-ului și adăugarea eșantioanelor.

După crearea ștampilei de timp și stabilirea numărului de eșantioane care vor face parte dintr-un cadru de date, urmează crearea cadrului, descris în paragraful 3.2. Atâta timp cât rata de citire a datelor este constantă se adaugă eșantioane la cadrul de date. Dacă rata de citire se modifică, cadrul se încheie, este transmis, iar noile eșantioane sunt introduse într-un cadru nou.

```

Task_transmisie
While (1)
    Dacă există eșantioane în buffer-ul intern
    //(if length_of_samples_buffer > 0)
    Dacă buffer-ul de eșantioane NU este plin
    //(if FrameData is not full)
    Extrage eșantion din buffer-ul (intern) de eșantioane
        Dacă este primul eșantion
            Completează header-ul :
                Adaugă identificatorul cadrului, FrameId
                Adaugă tipul cadrului, FrameType
                Adaugă ștampila de timp, TimeStamp

```

```

        Adaugă tipul senzorului, SensorId
        Adaugă rata de eşantionare, Rate
        length += length(eşantion)
        Rata_eşantionare_veche = Rate
        Adaugă eşantion la FrameData
        Continue; // extrage eşantion din buffer
    Altfel //nu este primul eşantion
        Dacă are aceeaşi rată de eşantionare
        (if Rata_eşantionare_veche == Rate)
            length += length(eşantion)
            Adaugă eşantion la FrameData
            Continue; // extrage eşantion din buffer
        Altfel // s-a schimbat rata de eşantionare
            Apelează_task_trimitere_FrameData
            △ //end dacă rata de eşantionare se păstrează
        △ //end dacă este primul eşantion
    Altfel // nu mai sunt eşantioane în buffer-ul intern
    Dacă un cadru FrameData aşteaptă să fie trimis
        Apelează_task_trimitere_FrameData
        △ //end dacă sunt eşantioane în buffer-ul intern
    △ //end while

```

Task_trimitere_FrameData descrie elementele implicate în transmiterea unui pachet: confirmări, lipsa confirmărilor care implică retransmiterea pachetului, situații de timeout.

```

Task_trimitere_FrameData
While (1)
    Dacă este permisă transmiterea
        Transmite_FrameData
        Dacă NU s-a primit confirmarea (Ack)
            Dacă timeout-ul a expirat
                Retransmite_FrameData
                Nr_retransmisii++
                Dacă s-a atins nr_max_retries
                    Break;
                Altfel //se mai pot face retransmisii
                    Continue;
            △ //end Not Ack
        △ //end este permisă transmisia
    △ //end while

```

Algoritmii de transmisie prezentați furnizează servicii la nivel aplicație. Ele trebuie să se bazeze pe un protocol existent la nivelul rețea. Serviciile descrise în pseudocod se pot implementa atât pentru transmisia radio cât și pentru comunicația serială, principiile rămânând aceleași.

3.4. Algoritmi de sincronizarea nodurilor

La nivelul fiecărui nod din rețea se citesc unul sau mai mulți senzori identici sau diferiți. Eșantioanelor li se aplică o ștampilă de timp, indicând momentul în care

s-a făcut citirea. Ștampila de timp trebuie să conțină data și ora dată cu acuratețe de milisecundă.

Senzorii monitorizează un sistem de dimensiuni mari, iar datele de la senzori fuzionează. Este importantă cunoașterea momentului precis când a avut loc citirea. Aceasta implică sincronizarea nodurilor din rețea.

Bazându-mă pe algoritmi de sincronizare prezentați în paragraful 2.3, în urma analizei realizate în 2.4, definesc în continuare trei algoritmi de sincronizare potriviți cerințelor.

Există premise comune tuturor algoritmilor. Acestea sunt : inițial există un grup de noduri care conțin GPS și prin ele se realizează sincronizarea. Aceste noduri, pentru identificare, conțin în identificatorul nodului cuvântul cheie "GPS": GPS1, GPS2, ..., GPSn.

Nodul cu GPS furnizează timpul la nivel de milisecundă și poate fi folosit ca centru de sincronizare pentru rețea.

Rețeaua se împarte în noduri (clustere) de sincronizare, în jurul nodurilor GPS. Împărțirea se face pe baza distanței minime dintre noduri. Grupul centrelor de sincronizare, poate fi format și doar dintr-un singur element. În această situație întreaga rețea se sincronizează cu acel nod.

Dacă nodul GPS părăsește rețeaua sau în rețea nu există noduri GPS, sincronizarea se realizează după nodul conectat la un calculator personal, care furnizează timpul.

Ideea de a folosi unul sau mai multe centre de sincronizare care dețin GPS am regăsit-o în lucrarea [29].

Conversația nod-GPS se bazează pe metoda transmițător-receptor, prezentată în paragraful 2.3.4. și este folosită în toate cele trei metode. Se utilizează patru timpi: T1,T2,T3,T4, cu ajutorul cărora se calculează diferența (offsetul). Nodul care a transmis cererea se sincronizează cu nodul cu GPS.

Astfel în cadrul de sincronizare se transmite timpul T1, iar nodul de referință furnizează în "răspuns la cadrul de sincronizare" timpii T2 și T3.

În desfășurarea algoritmilor am considerat câteva scenarii, cum ar fi intrarea unui nod nou în rețea, sau părăsirea rețelei de către un nod. Aceste noduri, pot fi de mai multe tipuri: nod frunză, nod părinte care are noduri copii sau centru de sincronizare GPS. Am tratat aceste situații în cadrul algoritmilor.

3.4.1. Algoritmul 1 : Centrele de sincronizare sunt setate inițial

În faza premergătoare creării rețelei, fiecare DAD este configurat, memorând informații cum ar fi tipul senzorilor pe care îi monitorizează, rata de eșantionare, identificatorul plăcii, identificatorul senzorilor sau intervalul de timp dintre sincronizări. Printre datele de configurare se adaugă adresa nodului GPS, și adresa unui nod GPS de rezervă, (backup).

La pornirea rețelei, nodurile se sincronizează direct cu GPS-ul ales, iar resincronizarea se face la intervale de timp, cunoscute. Dacă GPS-ul ales nu mai funcționează, GPS-ul de rezervă este folosit în locul lui. Dacă și GPS-ul de rezervă colapsează sincronizarea se face după calculatorul personal, care furnizează timpul.

Un nod știe momentul la care să facă sincronizarea și transmite cererea de sincronizare GPS-ului. În cerere este conținut momentul T1, la care s-a generat cererea. Nodul cu GPS răspunde trimițând înapoi timpii T2 și T3, corespunzător situației din TPSN, paragraful 2.3.4. La sosire, nodul care a cerut sincronizarea generează timpul T4. Având cei patru timpi, nodul sursă se sincronizează cu nodul cu GPS.

Voi descrie algoritmul și îl voi prezenta în pseudocod.

Descrierea algoritmului:

La pornirea rețelei, fiecare nod are memorată adresa GPS-ului cu care se realizează sincronizarea. Nodul are memorată și adresa unui GPS secund, folosit ca nod de rezervă (backup).

Fiecare nod din rețea, la un interval de timp cunoscut trimite mesaj de sincronizare cu GPS. În mesaj este conținut T1.

Există situații în care se generează o cerere de sincronizare independent de intervalul stabilit .

GPS-ul primește mesajul. Generează T2.

Pregătește pachet, generează T3 trimite pachet.

Nodul sursă primește pachetul. Generează T4.

Calculează delta și d

Ajustează T1 cu T2-delta-d

Nodul a fost sincronizat cu centrul de sincronizare

În cadrul algoritmului se tratează și cazul pierderii pachetelor, situație în care pachetul este retransmis.

```
funcție configurare_dispozitiv
...
Adresa nod sincronizare = adresa_nod_GPS
Adresa nod sincronizare 2 = adresa_nod_GPS_2_de_backup
...
end
```

Funcția *configurare_dispozitiv* prezintă configurarea în cadrul căreia DAD-ul memorează adresa centrelor de sincronizare, adică a celor 2 noduri cu GPS.

```
Task sincronizare_nod
Ultima_sincronizare = 0
While (1)
  Dacă
    (T1 - Ultima_sincronizare > Timpul_dintre_două_sincronizări)
      sincronization_request = TRUE
  ⚡
  Dacă există cerere de sincronizare
    //(sincronization_request = TRUE)
    Reține T1
    Trimite mesaj de sincronizare la GPS
    Așteaptă răspuns t ms
    Dacă timeout și nr_retransmisii == MAX_RETRASMISII
      //(au trecut cele t ms)
      Așteaptă Y ms pentru o altă retransmisie
      continue;
    altfel
      nr_retransmisii = 1
```

```

        continue;
    Δ // timeout t
    Dacă timeout și nr_retransmisii == MAX_RETRASMISII
    //au trecut Y ms și nodul GPS este declarat invalid
    Dacă GPS==GPS_prime_nod
        GPS = GPS_secund //de backup
        continue;
    Dacă GPS == GPS_secund
        Dacă GPS_prime_nod este activ din nou
            GPS = GPS_prime_nod
            continue;
        altfel
            GPS = ceas_PC //nici un GPS nu funcționează
            continue;
    Δ // timeout Y
    Dacă a sosit pachet de răspuns de la GPS
        Dacă pachetul este valid
            Reține T1, T2, T3, T4
            Calculează Δ și d

$$\Delta = \frac{(T2 - T1) - (T4 - T3)}{2}$$


$$d = \frac{(T2 - T1) + (T4 - T3)}{2}$$

            Ajunstează T1 cu T2-Δ-d
            nr_retransmisii = 0
            sincronization_request = FALSE
            Ultima_sincronizare = T1
        Δ // pachet valid
        break;
    // dacă pachetul e invalid reiau întregul proces de
    sincronizare, și generez din nou T1

    Dacă pachetul este not_ack
        break; // pachetul a ajuns la GPS dar era alterat
    Δ // a sosit not_ack

    Δ // a sosit pachet
        Δ // există cerere de sincronizare
        end while
    end
end

```

Task_sincronizare_nod reprezintă etapele realizate de un nod oarecare de la cererea de sincronizare către un nod cu GPS și până la realizarea sincronizării.

Sincronizarea se realizează la cerere sau la expirarea unei perioade de timp, moment după care sincronizarea nodurilor nu mai este garantată.

Nodul sursă care dorește să se sincronizeze cu centrul de sincronizare trimite un mesaj, în care este conținut timpul la care a generat mesajul, T1. Dacă sosește un mesaj valid de la GPS, se generează T4, se extrag timpii T1, T2 și T3 din mesaj; se calculează Δ, d și urmează corectarea timpului. În final sursa va fi sincronizată cu nodul de sincronizare, cu care a schimbat mesajele.

Dacă pachetul sosit de la GPS este alterat, procesul de sincronizare este reluat de la început, prin generarea unui nou timp T1 și transmiterea unei noi cereri de sincronizare.

Dacă pachetul de la GPS nu ajunge, se așteaptă o anumită perioadă de timp, iar la scurgerea ei (timeout) se retransmite cererea de sincronizare și se așteaptă din nou o perioadă de timp. Dacă după scurgerea celui de-al doilea interval nu s-a primit nici un răspuns, nodul cu GPS este declarat invalid. Se verifică posibilitățile: dacă primul nod GPS1 este invalid se stabilește GPS2 pe post de centru de sincronizare. Dacă GPS2 este invalid se verifică dacă GPS1 este valid, în cazul în care a revenit în rețea, altfel sincronizarea se face după ceasul PC-ului.

```

task sincronizare_la_GPS
  while (true)
    Așteaptă mesaj de la orice nod
    Dacă s-a primit mesaj de un nod
      Dacă pachetul e valid
        Reține T2
        Așteaptă x ms
        Reține T3
        Construiește pachet cu T1, T2, T3
        Trimite pachet la nod
      altfel
        trimite not_ack la nod
    Δ // pachet valid
    Δ // s-a primit mesaj de la nod
  end while
end

```

Task_sincronizare_la_GPS reprezintă acțiunile întreprinse de nodul cu GPS care realizează sincronizarea. La primirea unui cadru de cerere de sincronizare se generează timpii T2 și T3, se formează un cadru răspuns la cererea de sincronizare, în care sunt introduși timpii T1, T2 și T3. Cadru este transmis nodului care a cerut sincronizarea.

Dacă pachetul cu cererea de sincronizare ajunge alterat, nodul cu GPS va transmite înapoi un cadru de confirmare negativă (not_ack) prin care arată că a primit un cadru invalid.

3.4.2. Algoritmul 2 : Ierarhia se construiește pe un calculator personal

Acest algoritm tratează separat transmisia radio și cea serială.

La crearea rețelei se creează ierarhia nodurilor.

În cazul comunicării radio, ierarhia rețelei se salvează pe un calculator personal. Calculatorul personal determină poziția nodurilor cu GPS, creează zonele de sincronizare și transmite fiecărui nod adresa centrului cu care să se sincronizeze.

După primirea informațiilor de la calculatorul personal, un nod știe la ce moment să facă sincronizarea și transmite cererea de sincronizare nodului cu GPS.

Există situații în care nodurile se defectează sau părăsesc rețeaua.

Dacă nodul este părinte, la ieșirea lui din rețea nodurile copii vor deveni deconectate. Aceste noduri se vor reconecta, în cazul transmisiei radio, având alte

adrese de rețea. După reconectare, coordonatorul C va primi un mesaj de conectare, numit „Join Notification”. Acest mesaj ajunge la calculatorul personal. Acesta reface ierarhia și le trimite tuturor nodurilor noi venite adresa centrului de sincronizare corespunzător lor.

Voi considera cazul în care nodul GPS părăsește rețeaua, mai precis cazul în care nu este identic cu coordonatorul C. Pachetele de cerere de sincronizare (sincronization_request) adresate nodului cu GPS nu vor primi confirmare, se va ajunge la expirarea timpului de așteptare (timeout), iar nodul va fi declarat invalid. Nodurile vor trimite o notificare calculatorului personal. Acesta va decreta alt nod GPS, dacă există, altfel sincronizarea se va face după ceasul său intern. Calculatorul personal reface ierarhia și transmite tuturor nodurilor din rețea adresa noului centru de sincronizare.

La revenirea în rețea a unui nod cu GPS prin comanda de conectare „Join notification” calculatorul personal va fi notificat că a revenit un centru de sincronizare. Calculatorul personal va reface ierarhia și va transmite adresa de sincronizare nodurilor.

În cazul comunicării seriale se aplică același algoritm. Calculatorul personal construiește inițial structura rețelei, prin adunarea informațiilor despre toate nodurile valide. În cazul acestei comunicări poate apărea cazul întreruperii legăturii fizice, a firului, dintre noduri, care nu se rezolvă așa de elegant ca în cazul radio, prin reconectarea nodurilor izolate, ci este necesară intervenția unui operator. În cazul întreruperii legăturii dintre noduri, calculatorul personal observă că o parte a ierarhiei este inaccesibilă și contactează un operator uman.

```
Tabela_noduri_GPS = [GPS1, GPS2, ..., GPSk]
// Tablou care conține toate centrele de sincronizare
//disponibile din sistem.

Funcție_creaza_clustere
//calculeaza distanta minima
Dacă Tabela_noduri_GPS e vidă
    Centrul_de_sincronizare = PC // ceasul PC-ului
altfel
For fiecare nod i din retea
    For fiecare GPS j din Tabela_noduri_GPS
        If D[i] > dist_nod_i_la_GPS_j
            Begin
                D[i] = dist_nod_i_la_GPS_j
                A[i] = GPSj
            end
        end
    initializeaza_pe_0_tabela_clustere
    For fiecare GPS j activ din Tabela_noduri_GPS
        For fiecare nod i din retea
            If A[i] == j
                Adauga nod i la cluster[j]
        end
    end
    ⊖ // tabela_noduri_GPS nu e vida
// tabela cu clustere gata
```

În cadrul funcției *crează_clustere* sunt prezentate câteva dintre structurile de date folosite. Acestea sunt tablourile A, D și lista centrelor(cluster) de sincronizare.

A	GPS3	GPS2	GPS3	GPS1	GPSi	...	GPS2
	nod1	nod2	nod3	nod4	nod5		nodn

Fig. 3.4-1 Tabloul de date A

D	3	1	2	5	2	...	1
	nod1	nod2	nod3	nod4	nod5		nodn

Fig. 3.4-2 Tabloul de date D

Tabloul de date A, reprezentat în Fig. 3.4-1, are n poziții corespunzătoare cu cele n noduri existente în rețea. Fiecare poziție din tablou, $A[i]$, are codificarea nodului cu GPS cel mai apropiat, cu care trebuie să se sincronizeze.

Tabloul de date D, reprezentat în Fig. 3.4-2, are n poziții corespunzătoare cu cele n noduri existente în rețea și este folosit pentru calculul distanței minime până la cel mai apropiat nod cu GPS.

Cu ajutorul tablourilor D și A se determină distanța minimă dintre un nod al rețelei și un nod care conține GPS, realizându-se o legătură între nodurile din rețea și nodurile cu GPS. După realizarea acestor legături prin parcurgerea tabloului A se creează centrele (clusterelor) de sincronizare, ilustrate în Fig. 3.4-3. Astfel, se creează o listă de noduri pentru fiecare nod cu GPS, iar nodurile din acea listă se vor sincroniza cu GPS-ul respectiv.

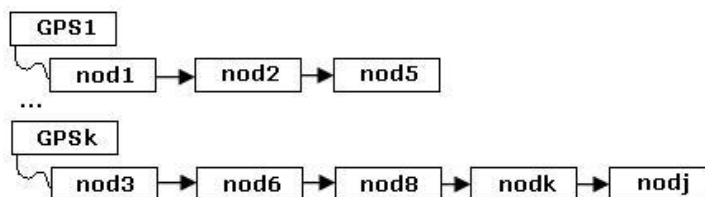


Fig. 3.4-3 Lista clusterelor de sincronizare

Fiecare listă creată, Fig. 3.4-3, reprezintă un cluster de sincronizare, conținând nodurile din rețea care se sincronizează cu GPS-ul corespunzător.

```

Funcție_formează_ierarhia
While (1)
  If nou_nod_GPS_prezent_in_retea
    Begin
      Activează_poz_nod_GPS_in_tabela_noduri
      Apelează_funcția_creează_clusterelor
      Sioncronization_request = TRUE
    End
  End
End // end while

```

Funcția `formează_ierarhia` este apelată la crearea rețelei, și ulterior, la cerere de către calculatorul personal. Dacă detectează un nod cu GPS care a reintrat în rețea, îl consideră activ, îl adaugă tabelii de noduri_GPS active și apelează funcția de creare a cluster-elor de sincronizare. Această funcție, distribuie nodurile

din rețea în jurul centrelor de sincronizare. În final se înaintează o propunere de resincronizarea întregii rețele.

```

Funcție sincronizare_clustere
    Pentru fiecare cluster din rețea
        Sincronizează nodurile din cluster cu centrul de
        sincronizare desemnat apelând task_sincronizare_nod

```

În cazul existenței mai multor noduri active cu GPS în rețea, se creează mai multe clustere de sincronizare. Sincronizarea rețelei se face prin sincronizarea fiecărui cluster; în cadrul lui, toate nodurile își reglează ceasurile în raport cu nodul GPS corespunzător. Sincronizarea unui nod cu nodul GPS se realizează prin apelarea task_sincronizare_nod prezentat în paragraful 3.4.1. Având în vedere că nodurile cu GPS furnizează timpul extrem de exact, sincronizarea separată a cluster-elor va duce la sincronizarea întregii rețele.

```

Funcția obtine_nou_nod_de_sincronizare
    Dist_min = MAX
    Centru_sincronizare = 0
    Foe fiecare nod i din rețea
        For fiecare GPS j activ din Tabela_noduri_GPS
            If Dist_min > dist_pana_la_GPS_j
                Begin
                    Dist_min = dist_pana_la_GPS_j
                    Centru_sincronizare = GPSj
                End
        GPS = Centru_sincronizare
    Adaugă nod_la_cluster_GPS

```

Funcția obține_nou_nod_de_sincronizare este apelată în situația în care nodul GPS părăsește rețeaua și nodurile din cluster-ul asociat lui nu mai au centru de sincronizare. În această situație fiecare nod își caută un nou centru de sincronizare, efectuând pașii descriși în funcție.

3.4.3. Algoritm 3 : Fiecare nod își construiește local ierarhia

Acest algoritm realizează sincronizarea fiecărui nod cu părintele lui, până la nodul cu GPS.

Algoritmii 1 și 2 lăsau libertatea rețelei de a-și crea drumurile între noduri.

În cazul acestui algoritm, fiecare nod își calculează drumul spre nodul cu GPS, bazându-se pe părintele lui. Calea de sincronizare și adresa nodului cu GPS sunt memorate local de fiecare stație.

Pentru sincronizare, un nod face cerere spre părintele lui. Părintele poate accepta cererea, sau poate refuza indicând că nici el nu este sincronizat. În acest caz nodul copil va aștepta o perioadă predefinită, la sfârșitul căreia consideră că nodul părinte este sincronizat și îi adresează din nou cerea de sincronizare.

Dacă un nod părinte, părăsește rețea, nodurile copii vor rămâne deconectate. În cazul comunicării radio, după reconectare, observată prin comanda „Join notification” trimisă coordonatorului, nodurile vor transmite comenzi de sincronizare spre nodul părinte.

În cazul acestui algoritm există un singur nod cu GPS și acesta este rădăcina(root) arborelui și este legat direct cu calculatorul personal.

Fiecare nod se sincronizează cu părintele lui.

Taskul sincronizare() este planificat sa se execute pe fiecare nod (mai puțin nodul rădăcină cu GPS) la fiecare T_y minute

Taskul Primește_pachete() se execută pe fiecare nod în parte.

Taskul Citeste_GPS() se execută doar pe nodul echipat cu GPS.

```

Task sincronizare ()
Ultima_sincronizare = 0
While (1)
    Dacă
        (T1 - Ultima_sincronizare > Timpul_dintre_două_sincronizări)
            sincronization_request = TRUE
    Δ
    Dacă există cerere de sincronizare
        // (sincronization_request = TRUE)
        Reține T1
        Trimite mesaj de sincronizare la Nod_parinte
        Așteaptă răspuns t ms
        Dacă timeout (au trecut cele t ms)
            Incrementează nr_retransmisii
            Dacă nr_retransmisii este depășit
                nr_retransmisii = 0
            //Așteapta o noua executie a taskului sincronizare
            Break;
        Altfel
            Așteaptă Y ms pentru o altă retransmisie
            continue;
    Altfel
        //a sosit pachet de la Nod_parinte
        nr_retransmisii = 0
        Dacă pachetul este valid
            Reține T1, T2, T3, T4
            Calculează Δ și d
            
$$\Delta = \frac{(T2 - T1) - (T4 - T3)}{2}$$

            
$$d = \frac{(T2 - T1) + (T4 - T3)}{2}$$

            Ajustează timpul_local cu T2-Δ-d
            sincronization_request = FALSE
            ultima_sincronizare = timpul_local
            Sincronized = TRUE
        Altfel Dacă pachet invalid // CRC error etc..
        Sau Dacă pachetul este not_ack
        //pachetul a ajuns la Nod_parinte dar era alterat

```

```

        Continue;
        // se reia procesul de sincronizare
    end while
end

```

În cazul task-ului de sincronizare() cererea de sincronizare este adresată de fiecare dată nodului părinte. Se folosește o variabilă, *synchronized* care arată dacă nodul este sincronizat sau nu. În cazul în care variabila este true și nodul primește o cerere de sincronizare de la nodul fiu, va furniza timpul său local. Dacă nodul nu este sincronizat, variabila *synchronized* fiind false, nodul curent va transmite el însuși o cerere de sincronizare nodului său părinte. În urma sincronizării sale, variabila va deveni true și nodul curent va putea să accepte cereri de sincronizare de la nodurile copii.

```

Task Primeste_pachete()
While (1)
    Asteapta pachet;
    Switch Tip_pachet
Case Date:
    ...
Case Sync_Request:
    Retine in T2 timpul_local
    Astepta x ms
Retine in T3 timpul_local
Timize Pachet SyncRespons
End while

```

În cazul task-ului Primeste_pachete() sunt analizate pachetele în funcție de tipul lor, iar dacă sosește un pachet de cerere de sincronizare, se generează timpii T2 și T3 necesari și împreună cu T1 primit în cerere, se transmite înapoi răspunsul la cererea de sincronizare.

```

Task Citeste_GPS()
While (1)
    Citeste GPS
    Seteaza timpul_local = GPS_Time

End while

```

Task-ul citește_GPS() rulează doar pe nodul rădăcină al arborelui, menținând un timp cât mai exact.

3.4.4. Analiza comparativă a algoritmilor de sincronizare propuși

În cadrul algoritmilor, au fost tratate atât sincronizarea DAD-urilor, cât și componenta adaptivă a sistemului, amintită în paragraful 2.5. Aceasta s-a realizat prin soluționarea situațiilor de părăsire și revenire a nodurilor în rețea. Pentru nodurile care părăsesc și apoi revin, rețeaua își păstrează funcționalitatea.

Dacă un nod frunză părăsește rețeaua, datorită faptului că nu are copii nu vor apărea probleme în comunicare.

În algoritmi creați realizarea sincronizării rețelei se face prin sincronizarea perechilor de noduri, ca în algoritmul TPSN, din paragraful 2.3.4. Un nod realizează comunicații (handshaking) cu un nod sincronizat, așa cum este ilustrat în Fig. 2.3-8.

Algoritmii de sincronizare propuși folosesc elemente întâlnite atât la TPSN (paragraful 2.3.4) cât și la LTS (paragraful 2.3.5). Amândouă protocoalele folosesc modul de sincronizare transmițător-receptor. Aceeași metodă o folosesc și eu, fiind simplu și oferind o acuratețe optimă în raport cu resursele utilizate.

Amândouă protocoalele construiesc o ierarhie. La fel și eu, dar modul de construire al ierarhiei este diferit de cele două abordări. Diferită este și reconstrucția ierarhiei. În TPSN ierarhia se realizează la crearea rețelei și se folosește pe întreaga durată de viață a ei, furnizând o abordare rigidă și puțin fiabilă față de situația rețelei mele. LTS reconstruiește ierarhia înaintea fiecărei sincronizări, folosind excesiv resursele stațiilor.

Soluția mea are un caracter adaptiv : la crearea rețelei se construiește ierarhia. Pe parcursul funcționării ierarhia se reconstruiește, la cerere. Prin această abordare doresc să ofer flexibilitate și conservarea resurselor nodurilor.

Soluția de sincronizare propusă de mine oferă trei algoritmi. În implementare am folosit prima metodă. Metoda doi și trei sunt alternative pentru diverse sisteme pe care va putea fi implementată metoda elaborată în această teză.

În accepțiunea mea am considerat că în rețea sunt posibile mai multe trasee, frunzele arborelui putând să se conecteze între ele.

Dar există sisteme care nu permit acest lucru. Algoritmul de sincronizare numărul 3 este proiectat pentru cazuri generale, în care ierarhia este un arbore clasic, iar traseele se desfășoară de-a lungul ramurilor (branch). Pentru această situație, algoritmul 3 permite o implementare bazată pe relația dintre nodurile părinte și copil.

Algoritmul doi constituie alternativa pentru rețelele în care resursele nodurilor fiind limitate, constituie o constrângere. În acest caz construirea ierarhiei se face pe calculatorul personal, iar nodurile rețin doar un minim de informație, adică adresa centrului de sincronizare și traseul către el.

În cadrul algoritmilor sincronizarea rețelei se face la cerere, în anumite situații cum ar fi reactivarea unui centru de sincronizare din rețea. Dar în mod uzual resincronizarea rețelei se face la intervale de timp cunoscute. În cadrul cercetării, bazându-mă pe algoritmi propuși și folosind calitățile circuitelor care vor implementa soluția se va determina timpul optim la care este recomandat să se facă resincronizarea rețelei.

3.5. Nivelul rețea. Descriere algoritmi de comunicare

Nivelul rețea se ocupă de controlul funcționării, determinând modul în care pachetele sunt dirijate de la sursă la destinație, crearea traseelor, retransmisiile, cazurile de eroare sau întârzieri. Pentru aceasta în general, fiecare nod din rețea își

crează o tabelă care conține trasee sau fragmente de trasee, în cazul unei rețele voluminoase. Traseele pot fi statice, situație în care toate pachetele urmează același drum, sau în funcție de analiza rețelei și congestiile existente, pachetele trimise de la sursă la destinație pot să urmeze drumuri diferite.

Algoritmii propuși pot fi aplicați și la rețelele radio și la rețelele seriale. Din acest motiv, voi prezenta și comunicarea radio dar și pe cea serială.

Comunicarea la nivelul rețea face legătura între nivelul legătură de date și nivelul aplicație. În Fig. 3.5-1 sunt reprezentate modelul de referință OSI și comunicarea radio, particularizată pentru protocolul ZigBee.

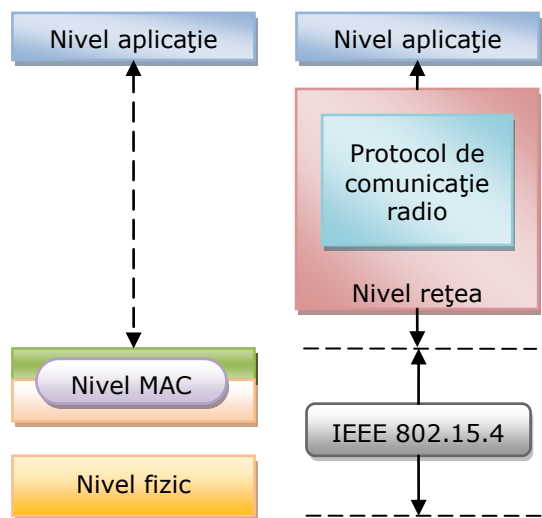


Fig. 3.5-1 Plasarea protocolului radio în cadrul modelului OSI

Alegerea protocolului ZigBee a decurs ca o consecință la cerințele algoritmilor și ale constrângerilor cercetării. Acest protocol, așa cum arată reprezentarea din Fig. 3.5-1 realizează comunicarea la nivel rețea și face legătura între nivelul aplicație și legătură de date.

Nivelul fizic și o parte din nivelul legătură de date, adică subnivelul MAC (*Medium Access Control*) din modelul OSI sunt implementate în comunicația radio din această teză folosind protocolul standard IEEE 802.15.4.

La nivel fizic se transmit și se recepționează datele prin antena radio. Se realizează comunicarea cu nivelul superior MAC.

La nivelul legătură de date, subnivelul MAC assemblează date în pachete pentru transmisie și dezassemblează pachetele sosite. Tot aici se realizează adresarea, adică se determină destinația pachetelor transmise și sursa pachetelor sosite.

Toate serviciile oferite de nivelul legătură de date superior și până la nivelul aplicație în nivelul OSI, formează în cazul radio un singur nivel reprezentat de protocolul ZigBee și denumit generic nivel rețea.

Standardul IEEE 802.15.4 implementează nivelul fizic și subnivelul MAC. Voi prezenta succint câteva caracteristici ale sale, necesare analizelor din această teză.

3.5.1. IEEE 802.15.4

Standardului IEEE802.15.4 este folosit tipic pentru aplicații repetitive care consumă puțină energie (low power). El definește trei benzi de frecvență care include 16 canale la 2,4GHz (rata maximă de transfer fiind 250 kbps), 10 canale de la 902 până la 928MHz (40 kbps) și un canal de la 868 la 870 MHz (20 kbps). Banda de 2,4GHz operează la nivel mondial (worldwide). [54]

Această bandă este folosită în comunicarea din proiectul meu, pornind de la 2,405 până la 2,480GHz. Rata de transfer maximă a datelor este teoretic de 256kps, dar datorită implementării protocolului de comunicație, în realitate se reduce aproximativ la jumătate.

În cadrul standardului IEEE802.15.4 se diferențiază 2 niveluri: fizic și subnivelul MAC, Fig. 3.5-2, și aceste niveluri oferă serviciile echivalente ale modelului OSI, prezentate în Fig. 3.5-1.

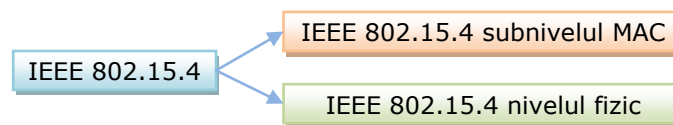


Fig. 3.5-2 Evidențierea nivelului fizic și subnivelului MAC în cadrul IEEE 802.15.4

Pentru nivelul fizic, amintesc dintre serviciile oferite de IEEE 802.15.4 doar detectarea energiei receptorului(ED), indicarea calității legăturii (LQI) sau evaluarea gradului de încărcare al unui canal(CCA), toate trei reprezentând factori în creșterea eficienței transmisiei [54,58,59].

Detectarea energiei receptorului (ED) este o metodă prin care se determină dacă există un semnal de date pe linia de comunicație. Această detecție se realizează prin recepționarea unui semnal de pe linie, măsurarea nivelului semnalului recepționat și compararea nivelurilor măsurate cu nivelurile măsurate anterior [57].

Protocolul ZigBee își alege căile de transmitere a datelor bazându-se pe legături (link) unde probabilitatea de succes este mare. Standardul IEEE802.15.4 permite la nivel fizic măsurarea calității legăturilor (LQI) dintre nodurile vecine ale unei rețele. Studiile au arătat că stabilirea rutelor folosind acest principiu a mărit valorile metricilor de fiabilitate bazate pe rutare, dintre care amintesc numărul de pachete planificate (ETX), neafectând alte metrici de performanță. În experimentele efectuate în [58], pe un sistem diferit de cel utilizat aici, rata de transmitere a datelor într-o aplicație cu trafic ridicat s-a îmbunătățit cu 400%. Și în alte rețele efectele sunt pozitive, chiar dacă nu au rezultate atât de spectaculoase.

Evaluarea gradului de încărcare al unui canal (CCA) este metoda prin care se determină dacă un canal este deschis pentru comunicare, verificându-se acest lucru înainte de începerea transmisiei. Metoda este folosită în cadrul protocolului CSMA/CA și se bazează pe principiul că transmiterea unui pachet pe canalul de

comunicație creează o intensitate a semnalului suficientă pentru a depăși un anumit prag astfel încât să demonstreze că se transmite un pachet și că nu este vorba de perturbație [59].

În comparație cu CCA, ED reprezintă o metodă nesofisticată de a determina trafic pe un canal.

Comunicația se realizează într-un interval de aproximativ 2MHz de canale de la 2,405 până la 2,480 GHz.

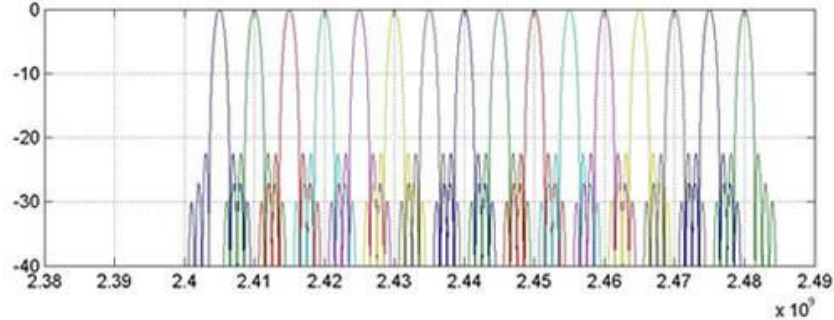


Fig. 3.5-3 Reprezentarea celor 16 canale de comunicație oferite de standardul IEEE802.15.4, figură preluată din [61]

În Fig. 3.5-3 sunt ilustrate 16 canale de comunicație și reprezentarea lor în banda de frecvență. Figura este preluată din [61].

Realizarea configurațiilor în rețea se poate face punct-la-punct, peer-to-peer sau punct-la-multipunct.

Configurația uzuală, prezentată în Fig. 3.5-4, este formată dintr-un nod central, numit coordonator, la care se conectează noduri aflate la distanță.

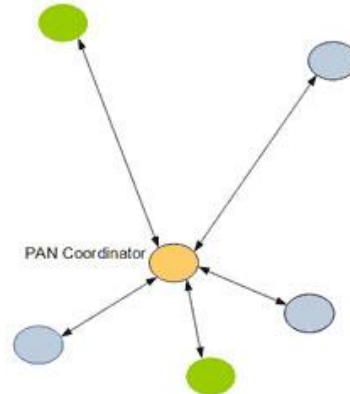


Fig. 3.5-4 Reprezentarea unei configurații uzuale folosind IEEE802.15.4

În cadrul rețelei anumite noduri permit legarea (rout-area) la ele, a altor noduri, creând în acest fel o structură complexă, cum este cea reprezentată în Fig. 3.5-5.

3.5.2. Protocolul ZigBee

Acest paragraf descrie pe scurt protocolul ZigBee, prezentarea completă găsindu-se în bibliografie la poziția [7,10,11]. Protocolul ZigBee este autoconfigurabil, și oferă multe facilități de lucru în rețea.

Legătura dintre nivelul fizic și nivelul aplicație poartă numele generic de nivel rețea și este implementat în acest proiect folosind protocolul ZigBee. Acest protocol este folosit în aplicații care necesită o rețea sigură și oferă facilități în acest sens. Am ales protocolul ZigBee pentru a face un studiu asupra felului cum reacționează în cazul unui volum mare de date.

Deasupra nivelului MAC, ZigBee implementează o stivă pe structuri asemănătoare TCP/IP care realizează operații transparente nivelului utilizator. Dintre acestea amintesc crearea unei noi rețele, desemnarea adreselor nodurilor care intră în noua rețea, găsirea traseelor de rutare și propagarea mesajelor.

Urmează detalierea câtorva dintre serviciile oferite de protocolul ZigBee, pe care le-am folosit în cercetarea și implementarea mea.

3.5.3. Topologia ZigBee

Descrierea topologiei o voi face pe două direcții: tipul rețelelor și tipul nodurilor.

Rețele ZigBee pot fi configurate, în funcție de necesități sub trei forme: stea, arbore sau mesh(plasă).

Topologia stea are un nod central și întreaga comunicație se face prin el.

Topologia arbore are un nod rădăcină de la care pornesc ramificații pe mai multe niveluri. Comunicarea se realizează în orice direcție, de sus în jos sau de jos în sus, între noduri diferite, nefiind obligatorie trecerea prin nodul rădăcină.

Topologia plasă este asemănătoare structurii arbore, având un nod rădăcină, în plus permite mai multe trasee, deoarece nodurile realizează legături cu alte noduri, nu neapărat doar cu părintele și copiii lor. Un nod care se leagă de altul poartă numele de copil, iar primul este considerat părintele lui. Comunicarea se face în orice direcție, iar în plus pot exista mai multe trasee de la un nod sursă la nodul destinație, Fig. 3.5-5.

În cercetarea mea am ales și utilizat a treia topologie, mesh-ul ZigBee. Dintre cele trei topologii la mesh comunicarea este cea mai flexibilă și fiabilă, deoarece la căderea unui nod se pot crea alte trasee alternative.

Rețele ZigBee sunt numite PAN (Personal Area Networks) și fiecare dintre ele conțin un identificator de rețea pe 16 biți, numit PAN ID. Într-un PAN există trei feluri de noduri: coordinatorul C, router-i R, și noduri terminale E. Un astfel de exemplu este ilustrat în Fig. 3.5-5

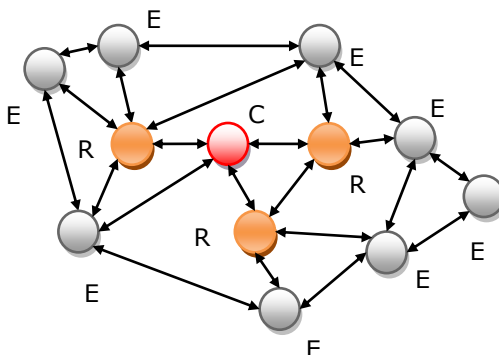


Fig. 3.5-5 Topologie ZigBee de tip Mesh

Coordonatorul C este prezent în orice PAN. El creează rețeaua stabilind PAN ID-ul și selectează canalul de comunicație. Pentru aceasta inițiază o scanare a canalelor disponibile, care pot fi maxim 16, trimițând un semnal de request spre toate nodurile din raza lui de acțiune. La sosirea mesajelor C analizează datele și în funcție de alți coordonatori își alege un canal, care să nu coincidă cu vreunul existent.

Mai multe rețele pot să funcționeze pe același canal, dar ele se evidențiază prin identificatorul de rețea, PANID, diferit.

Inițial se creează o nouă rețea care are un singur nod, coordonatorul. La ea se pot conecta router-e și stații terminale. C supraveghează comunicația, dar poate primi și trimite date ca orice alt nod. Adresa C este întotdeauna 0, fiind cunoscută.

Într-un PAN există doar un singur coordonator C.

Router-ele pot să se lege la o rețea, cu condiția ca ea să existe și să fie creată. Pentru aceasta router-ul trimite, la fel ca în situația anterioară, o cerere de scanare a tuturor nodurilor XBee. După primirea răspunsurilor, router-ul poate să se lege la orice PAN sau poate alege rețeaua, specificând PANID-ul ei. Pentru conectare se trimite un mesaj de cerere. După legare, router-ele permit altor router-e sau noduri terminale să se lege de ei. În mesh-ul din Fig. 3.5-5 există trei router-e [7].

Stațiile terminale se leagă la rețea în maniera prezentată la router-e. Ele primesc și transmit date. Spre deosebire de router-e, de un nod terminal nu se pot lega alte noduri. Nodurile terminale sunt singurele care pot intra în starea de low-power.

La intrarea unui router sau a unui nod terminal într-un mesh, pot exista mai multe noduri la care se poate lega. În această situație, de obicei este ales acela care are semnalul cel mai bun. Pentru conectare nodul trimite un mesaj de cerere spre viitorul părinte. Router-ul sau coordonatorul care primesc cererea pot accepta sau nu. Fiecare router sau coordonator au un număr fix de noduri copii pe care le poate accepta. Pentru modulele XBee Series 2 alese în acest proiect valoarea maximă este 8.

Dacă această valoare a fost atinsă, nodul respectiv va trebui să refuze noi conexiuni. Această valoare este anulată în cazul unei reinițializări (reset) a rețelei.

Numărul maxim de noduri dintr-un PAN este de 65535.

3.5.4. Adresarea

Nodurile dintr-un PAN conțin două adrese. Una permanentă pe 64 de biți și una pe 16 biți care este alocată la legarea în rețea. Dacă nodul părăsește rețeaua și apoi reintră, adresa pe 16 biți poate fi alta. Orice nod poate să ceară corespondenta pe 16 biți a altui nod, cunoscând adresa lui permanentă pe 64 de biți.

Adresa pe 64 biți este alocată de protocolul IEEE 802.15.4. Ea este permanentă și mai poartă numele de adresă extinsă sau adresă MAC.

Adresa pe 16 biți poartă numele de adresă scurtă și corespunde rețelei locale. Într-o configurație cu mai multe rețele ZigBee pot exista noduri situate în rețele diferite, dar având adresa pe 16 biți identică.

Ca o concluzie, trebuie avut în vedere că doar nodurile aparținătoare aceleiași rețele pot comunica între ele, deoarece nodurile care aparțin aceluiași PAN și folosesc același canal de comunicație.

La legarea în rețea fiecare nod primește o adresă pe 16 biți. Datele pot fi transmise sau recepționate în întreaga rețea, chiar dacă pentru acesta se traversează mai multe noduri de la sursă până la destinație. O rețea radio care permite traversarea mai multor noduri poartă numele de rețea multi-hop și se poate întinde pe o arie largă.

În cadrul unei astfel de rețele este rezolvată intern problema razei de acțiune despre care am discutat în paragraful 2.1, dar apar întârzieri de propagare la parcurgerea fiecărui "hop", de care trebuie ținut seama la algoritmul de sincronizare.

Fiecare rețea are o întârziere specifică, numită network latency. În cadrul rețelelor radio bazate pe ZigBee întârzierea rețelei este de 15ms.

În cadrul protocolului ZigBee pentru fiecare pachet transmis se primește un pachet de confirmare (acknowledge). La rețelele multi-hop, la recepția unui pachet nodul intermediar trimite confirmarea spre nodul care a trimis, dar și spre nodul sursă. Dacă confirmarea nu a ajuns la nodul intermediar, acesta retransmite pachetul. Între noduri intermediare pachetul este retransmis de maxim 4 ori.

Dacă confirmarea nu ajunge la sursă, după expirarea unui timp de așteptare (timeout), sursa retransmite pachetul în rețea. Această retransmisie globală se poate realiza de maxim 2 ori.

Pentru modulele XBee Series 2 alege numărul maxim de hopuri parcurse de la sursă până la destinație este 32 [7].

3.5.5. Metoda de rutate AODV

La transmiterea unui pachet, PAN-ul realizează trasee între sursă și destinație, care uneori parcurg mai multe noduri, numite hopuri. Procesul de descoperire a traseului și găsirea unui drum minim se bazează pe protocolul AODV(Ad-hoc On-demand Distance Vector Routing).

Acest protocol folosește tabele în care se reține nodul sursă și nodul (hopul) următor pentru ajungerea la destinație.

La transmiterea unui pachet, nodul sursă trimite o cerere spre destinație, pentru stabilirea unui traseu. Cererea poartă numele de "descoperirea drumului" (route discovery), și este propagată în toată rețeaua (broadcast). Tabelul intern al unui nod conține un număr limitat de intrări, astfel încât în rețelele de dimensiuni mari, local sunt ținute segmente de traseu, iar operația de descoperire se repetă, tabelul intern actualizându-se dinamic. La destinație ajung propuneri de traseu din toată rețeaua. Destinatarul le analizează și o reține pe cea mai avantajoasă. În final răspunde nodului sursă prezentându-i-o.

Pentru alegerea unui traseu se optează pentru calea directă, cu numărul minim de noduri. Dacă însă calitatea semnalului este slabă sau există pe traseu surse de perturbății, se va opta pentru un alt traseu, cu mai multe hopuri, dar cu semnal mai bun, în acest fel evitându-se retransmisiile [11].

3.5.6. Interogarea și setarea informației nodurilor

Protocolul ZigBee permite unui nod să afle informații despre alte noduri din rețea, cum ar fi adresa lor sau aplicațiile care rulează pe ele. Aceste informații sunt stocate în descriptorii fiecărui nod și în general se cer la intrarea unui nod nou în rețea.

Un nod poate să interogheze alt nod asupra adresei sale. Furnizând adresa extinsă pe 64 de biți poate să obțină adresa scurtă pe 16 biți sau invers.

Interogarea indică tipul nodului. Dacă nodul interogat este router sau coordinator el poate, în mod opțional, să furnizeze adresa tuturor nodurilor asociate lui. Folosind comanda NC (Number of Children) se poate determina numărul nodurilor atașate nodului curent [11].

Folosind acest mod de interogare se poate determina structura întregii rețele, pornind de la coordinator, a cărui adresă este cunoscută apriori, și interogând toate nodurile atașate lui.

Protocolul permite setarea unor valori pentru un anumit nod, pentru întreaga rețea sau pentru un nod aflat la distanță. Dintre acestea amintim modificarea dinamică a perioadei de acceptare a unor noi noduri în rețea (join). Tot la fel și timpul folosit pentru așteptarea primirii informațiilor despre alte noduri (node discover). Se poate seta, pentru fiecare transmisie în toată rețeaua (broadcast) a numărului maxim de noduri (hopuri) prin care poate trece mesajul. Un nod poate să dea informații despre spațiul de stocare disponibil (payload).

În proiectul meu am folosit serviciile oferite de protocolul ZigBee, descrise mai sus, pentru implementare.

3.5.7. Moduri de lucru oferite de ZigBee

Protocolul ZigBee oferă două moduri de operare: modul transparent și modul API.

Modul transparent are un comportament identic unei linii seriale. Datele nu sunt analizate: octeții recepționați în buffer-ul intern de recepție, Rx, sunt redirecționați spre buffer-ul de transmisie, Tx și transmiși în rețea.

Modul de comunicare API (Application Programming Interface) permite interogarea nodurilor din rețea. Datele circulă sub formă de cadre de recepție și de transmisie care descriu operații sau evenimente care au loc la nodurile din rețea. Folosind aceste tipuri de cadre se poate configura un nod aflat la distanță, se pot transmite date (broadcast, multicast, unicast) spre mai multe noduri și se primesc confirmări care dovedesc dacă cadrul a ajuns sau nu la destinație. La recepție se poate determina adresa nodului emitent [7].

În dezvoltarea comunicației am folosit modul de lucru API. Eșantioanele obținute de la senzori le-am grupat sub formă de cadre, prezentate în paragraful 3.2. Cadrele API transportă pachetele asamblate de mine în rețea. Cadrele de confirmarea primirii pachetelor sunt tot cadre API. Un cadru API o lungime maximă de 72 de octeți.

3.6. Descriere RDEA

În vederea configurării rețelei am studiat câteva rețele implementate în literatură și modul lor de configurare.

Lucrarea [76] propune un sistem complex care se desfășoară în cadrul unei confruntări armate. Dispozitivele implicate sunt vehicule, senzori, întrerupătoare (actuator), echipamente de calcul și comunicație. Sistemul lucrează în timp real, face recunoașteri, poate să planifice operații, să ceară optimizări, ducând la îndeplinire diverse sarcini. Sistemul este modelat sub forma unei ierarhii în care anumite dispozitive numite agenți, situate pe un nivel ierarhic superior, au control asupra celor de pe nivelurile inferioare[76]. Un punct de vedere original extras din această lucrare l-a constituit modelarea mediului înconjurător, care este văzută ca un mediu dinamic, în schimbare.

A doua rețea pe care o prezint este formată din 4 noduri (santinele) și citește date de la 10 senzori: acustic, seismic, magnetic, temperatură, vânt, umiditate atmosferică, ploaie, presiune, imagini și imagini infraroșu. Aplicația este implementată folosind Java Ants. Modulele folosesc un microprocesor Motorola MPC555, cu ceas intern de 40MHz, care furnizează aplicațiilor timer-e, CAN, trei porturi seriale, 32 de canale pe 10 biți de conversia A/D, 28kocteți de RAM și 384Kocteți de memorie flash[63]. Această rețea se aseamănă mult cu rețeaua creată de mine: citește senzori și transmite datele citite în rețea. Diferențele constau în faptul că RDEA folosește alte tipuri de senzori, alte dispozitive de calcul decât cele folosite de mine, iar aplicația este concepută diferit.

Următoarea rețea prezentată este folosită pentru comanda unui grup de avioane în mișcare, care sunt modelate ca noduri de rețea și sunt abstractizate într-un sistem ierarhic distribuit. În sistem funcționează concurent componente hardware și software. Avioanele sunt dotate cu senzori GPS (Global Positioning System) și INS (Inertial Navigation System). În această abordare procesoarele, memoriile, senzorii și o parte din aplicațiile aferente lor sunt considerate părți hardware și nu fac parte din proiect. Design-ul sistemului lucrează doar cu componente software. Aplicația lucrează pe un sistem de operare în timp real, RTOS. Modelarea sistemului s-a făcut folosind mediul Giotto. S-au urmărit două

cerințe: crearea unui timp general, evitându-se diferențele de fază și s-a creat o cronologie a informațiilor citite de la senzori [82].

Modelarea rețelelor prezentate, de mari dimensiuni, cu constrângeri de timp mi-au furnizat modul de abordare al RDEA din proiectul meu, care lucrează pe două niveluri ierarhice.

Infrastructura RDEA se bazează pe mesh-ul Xbee, adică pe rețeaua oferită de tehnologia ZigBee, care furnizează protocol de comunicație radio. Folosind această rețea am construit rețeaua formată din DAD-urile mele, rezultând RDEA. Nivelul inferior îl constituie protocolului de comunicație ZigBee, iar deasupra lui am dezvoltat cerințele nivelului aplicație.

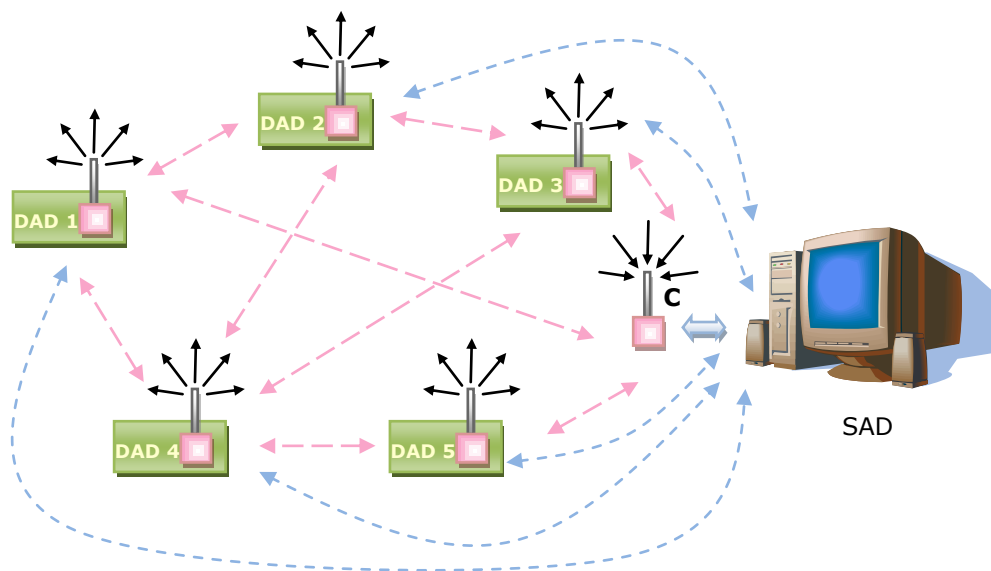


Fig. 3.6-1 RDEA și mesh-ul Xbee

Fig. 3.6-1 prezintă cele două niveluri. La bază se află rețeaua Xbee, desenată cu roz în figură (liniile punctate mai lung), peste care s-a construit rețeaua de DAD, RDEA-ul, punctată cu albastru (liniile punctate mai scurt).

În exemplul considerat, Fig. 3.6-1, RDEA conține 5 stații DAD. Fiecare DAD are atașat un modul Xbee pentru transmisia radio.

Nodurile comunică direct cu un calculator personal carer stochează datele și este numit SAD. Fiecare DAD eșantionează date și le transmite, folosind protocolul ZigBee.

La nivelul RDEA, coordonatorul rețelei Xbee, C, este transparent, el fiind perceput ca un nod obișnuit. Dacă comunicarea nu s-ar fi realizat radio, ci prin cablu, atunci acest nod poate fi puntea dintre RDEA și SAD.

RDEA este formată dintr-un număr fix de noduri. La conceperea RDEA se cunoaște numărul nodurilor. Adresele DAD sunt fixe și cunoscute. În timpul funcționării un nod se poate defecta sau poate părăsi rețeaua. Ulterior se poate integra din nou. Aceasta reprezintă o caracteristică dinamică pe care RDEA trebuie să o realizeze.

3.7. Timpul de resincronizare

Algoritmii propuși sincronizează nodurile din rețea, dar după o anumită perioadă de funcționare rețeaua se va desincroniza. În acest paragraf studiez întârzierile care apar, modul lor de tratare și calculez timpul scurs între două resincronizări ale rețelei.

Mărimea pachetului poate introduce întârzieri în comunicație. Numărul optim de eșantioane care trebuie conținut într-un cadru va fi studiat și prezentat în capitolul 5.

În cadrul comunicării radio, trebuie să se țină seama de faptul că acțiunea de descoperire a rețelei inițiată de un nod și necesară în procesul de creare al ierarhiei durează 6 secunde. Dacă nodul funcționează economic (low-power), până reintră în funcționare mai durează 30s.

Nodurile GPS au nevoie de câteva minute până stabilesc legătura cu sateliții și devin funcționali. În acel moment transmit un mesaj prin care anunță că sunt operaționali.

Aceste întârzieri trebuie considerate la crearea ierarhiei și la determinarea centrelor de sincronizare.

În paragraful 2.2, am prezentat trei mărimi utilizate în sincronizare și anume diferența (offset), distorsionarea (skew) și decalarea (drift). Distorsionarea (skew) nu afectează cercetarea noastră, deoarece DAD-urile pot avea microcontrolere cu frecvențe de lucru diferite și nu urmăresc să realizeze o sincronizare în frecvență.

Acuratețea oscilatorului de cuarț de pe placa DAD este furnizată în fișa tehnică și este de ordinul ± 20 ppm [52]. Aceasta înseamnă că la un milion de tacturi de ceas, poate apărea o eroare de maxim +20 sau -20 tacturi.

Datele tehnice [35] indică o acuratețe pentru drift având valoarea de 4ppm în primul an, iar din al doilea an de 2ppm. Din acest motiv, în implementarea cercetării am considerat neglijabilă decalarea (drift) și nu am luat-o în calcul [45] .

Doresc să determin timpul la care este necesară efectuarea unei sincronizări. Frecvența de lucru a oscilatorului de pe placă este 14,7456 MHz. Aceasta înseamnă că într-o secundă au loc 14745600 tacturi. Mai știu că eroarea cuarț-ului este de ± 20 ppm.

Astfel:

14745600 1000000 μ s

14745620X μ s

$X = 1000001,356 \mu$ s

Aceasta se interpretează prin faptul că la fiecare secundă în sistem este introdusă o abatere, o eroare de 1,356 μ s.

La 1000 sec abaterea va deveni de 1,356 ms. 1000 de secunde reprezintă 16,66 minute.

În concluzie la fiecare 10-15 minute este necesară o resincronizare pentru a păstra acuratețea rețelei la nivel de milisecundă. Această resincronizare este numită în această teză resincronizare generală.

În afară de această resincronizare generală uzuală, pot apărea cereri asincrone de sincronizare dictate de nodurile din rețea.

3.8. Comunicarea serială la nivel rețea

A doua posibilitate de comunicare în rețea este comunicarea serială.

În cadrul acestei comunicări, nodurile sunt legate prin cablu care furnizează un protocol serial. Cadrele de date prezentate în paragraful 3.2 sunt transmise în rețea fără a fi încapsulate în pachete API. Serviciile nivelului fizic și legătură de date sunt furnizate de comunicația RS-485. Deasupra lor trebuie implementat protocolul de comunicație corespunzător nivelului rețea.

DAD-urile din Fig. 3.8-1 sunt legate serial prin două fire realizând un canal de comunicație RS485 care furnizează un protocol serial. Comunicarea este half-duplex. Într-o astfel de rețea se pot conecta maxim 256 de noduri [50].

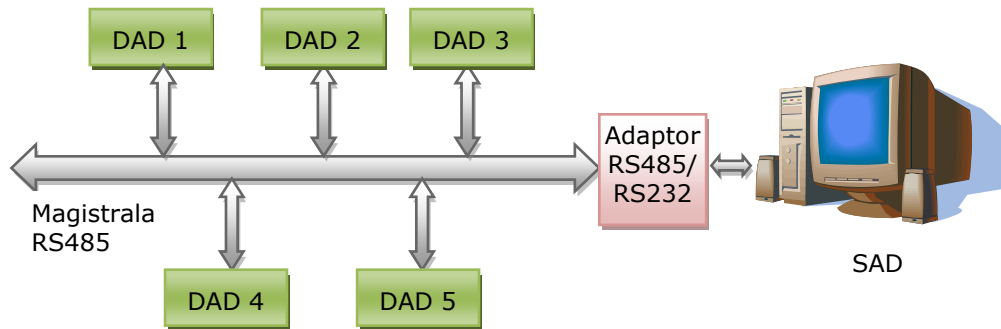


Fig. 3.8-1 Comunicarea serială în cadrul RDEA

Folosind terminologia utilizată de Tanenbaum în [1] descriu rețeaua din Fig. 3.8-1 ca o rețea cu difuzare, în care există un canal comun, la care sunt legate toate nodurile. Conectarea rețelei la calculatorul personal se face prin intermediul unui adaptor RS485-RS232.

Transmiterea datelor se face la o rată maximă de 2400 biți/s pentru noduri aflate la 1200m [50].

Toate nodurile „ascultă” canalul și primesc mesajele transmise, dar le rețin pe cele care corespund adresei lor. Doar o stație poate să transmită la un moment dat, altfel apar coliziuni și datele sunt alterate.

Principiile sunt identice cu cele întâlnite la comunicația radio, implementările fiind diferite.

Utilizatorul trebuie să furnizeze protocolul de comunicație și sincronizarea nodurilor din rețea.

Rata de transfer este relativ mică. Nu se poate transmite pe distanțe mari folosind un baud rate mare deoarece apar paraziți, perturbații, iar semnalul se pierde.

În proiect am creat rețele seriale, de mici dimensiuni, în care transmiterea datelor s-a făcut în maniera time-sharing. Cu toate că aceste rețele cablate serial funcționează, datorită ratei de transmisie reduse, nu au constituit obiectul meu de studiu și analiză cuprins în această teză, îndreptându-mi atenția spre transmiterea radio cu baud rate ridicat.

3.9. Structura generală DAD

Algoritmii prezentați până acum furnizează soluția teoretică pentru cerințele și constrângerile RDEA. Ca o consecință a metodelor a apărut necesitatea creării practice a nodurilor rețelei, a dispozitivelor de monitorizare și achiziția datelor, notate generic în această disertație DAD.

Am căutat dispozitive existente pe piață, COTS, pentru implementarea nodurilor RDEA, iar acolo unde nu au existat sau nu furnizau exigențele cerute le-am adaptat. În final a rezultat o configurație originală de DAD care poate fi aplicată și folosită pentru acest sistem.

Un DAD include un sistem cu microcontroler care are scopul achiziționării semnalelor provenite de la diferiți senzori, așa cum e reprezentat în Fig. 3.9-1. După cercetarea materialelor bibliografice legate de senzori s-au conturat soluții tehnice care constau în găsirea de metode de gestionare al unui volum mare de date și modalitățile de comunicare în rețea.

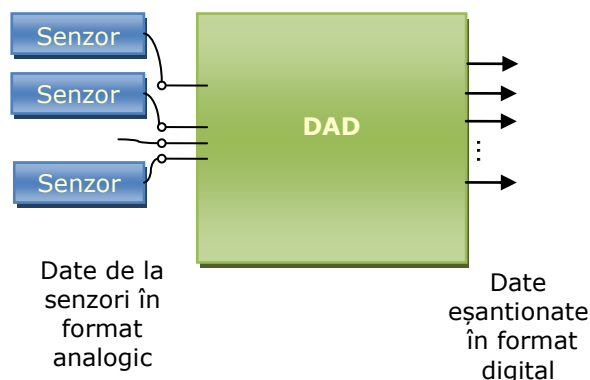


Fig. 3.9-1 Structură generală DAD

Abstractizând DAD-ul realizat pentru această teză, el poate fi privit ca o cutie neagră, Fig. 3.9-1, care citește date analogice și returnează date digitale, eșantionate. Datele analogice citite reprezintă valori ale unor mărimi fizice (temperaturi, presiuni, umidități), electrice (tensiuni, curenți) sau mecanice (poziții relative, deplasări, înclinări, deformări). DAD-ul este proiectat pentru a gestiona senzori identici sau diferiți și poate citi și înregistra date de la mai mulți senzori în același timp.

Structural, un DAD se compune din trei module distincte : achiziția, comunicația și partea centrală, vezi Fig. 3.9-2.

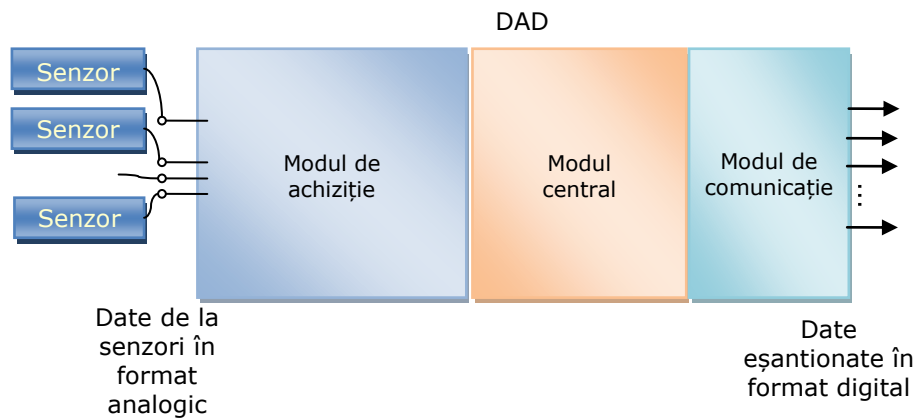


Fig. 3.9-2 Configurația DAD pe module

Partea de achiziție a semnalului preia datele de la senzori, le convertește din reprezentarea analogică în format digital.

Partea centrală conține un microcontroler ARM (Advanced RISC Machine) care gestionează celelalte module, memorează local mărimile achiziționate și sincronizează DAD-ul cu celelalte noduri din rețea.

Partea de comunicație permite comunicare prin fir sau radio.

În Fig. 3.9-3 s-au detaliat cele 3 module ale DAD-ului și structura lor.

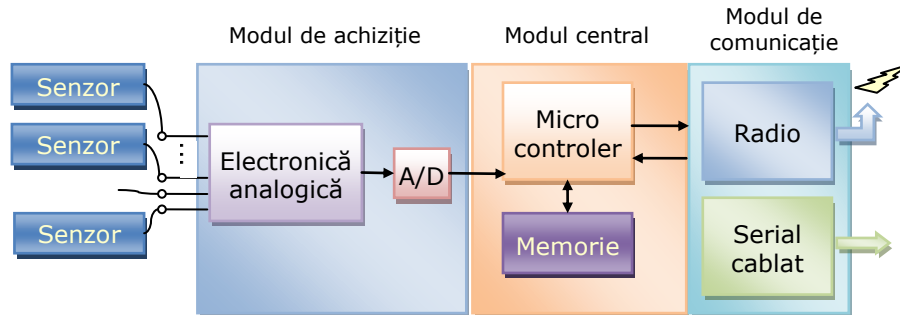


Fig. 3.9-3 Configurația generală a unui DAD

Modulul de achiziție reprezintă interfața dintre dispozitivul de achiziție și senzorii folosiți, realizat printr-un modul de electronică analogică, care prelucrează datele care provin de la senzori și le furnizează convertorului analog-digital. Un dispozitiv de achiziție citește date de la unul sau mai multi senzori, iar aceștia pot fi identici, sau diferiți. Fiecare tip de senzor, în funcție de parametrul pe care îl monitorizează, are rata lui de eșantionare. Rata de eșantionare se modifică dinamic în timpul funcționării. Datele eșantionate sunt preluate de microcontroler care realizează o analiză preliminară în urma căreia se stabilește rata de eșantionare, pe care o transmite blocului de electronică analogică.

Modulul central, cu microprocesorul și memoria internă preia datele convertite în format digital, le aplică o ștampilă de timp, le analizează, apoi le assemblează în pachete și le transmite în rețea. Tot în acest modul sunt tratate pachetele de date recepționate și confirmările.

Modulul de comunicație se ocupă exclusiv de transmiterea datelor primite de la microcontroler și recepționarea pachetelor primite din rețea.

Dintre senzorii monitorizați, îi exemplific doar pe cei de temperatură, vibrație, tensiune, curent, tensiune din trolu sau GPS. DAD-ul poate citi practic orice senzor mecanic, electric sau de altă natură. Am dat câteva exemple pentru a clarifica gradul de diversitatea al senzorilor ce pot fi monitorizați de DAD. În general fiecare senzor are un mod propriu de comunicație între el și placa de achiziție, definit de un protocol. Aplicația creată de mine rezolvă detaliile tehnice legate de protocoalele de comunicație între senzori și placa de achiziție, făcându-le transparente utilizatorului și furnizând primitive generice de comunicație la nivel utilizator.

Citirea senzorilor se face în funcție de tipul lor, pornind de la milisecunde și ajungând până la valori de minute sau uneori ore.

3.10. Schema electrică DAD

Acest paragraf descrie schema electrică a DAD, ilustrată în Fig. 3.10-1.

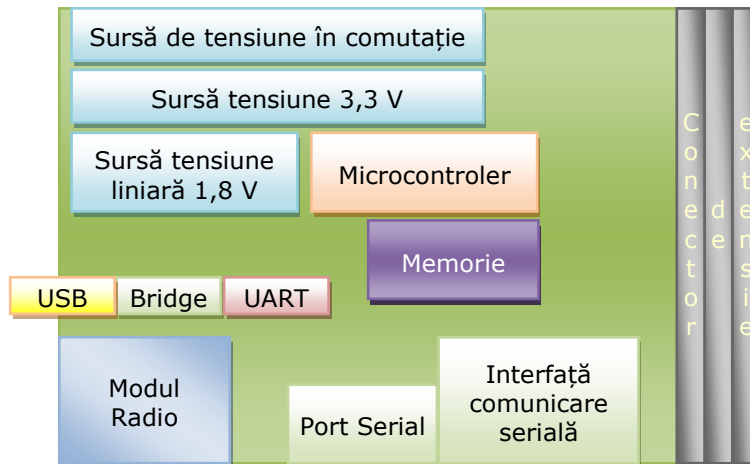


Fig. 3.10-1 Schema electrică DAD

DAD-ul a fost proiectat să fie alimentat cu o tensiune de 24Vcc, destul de des folosită în industrie. Circuitele integrate folosite în DAD necesită tensiuni de 5V, 3,3V respectiv 1,8V pentru core-ul microcontrolerului. Pentru a obține un randament bun al sursei de alimentare s-a folosit o sursă în comutație, care obține tensiunea de 5V din tensiunea de 24V. Tensiunile mici de 3,3V respectiv 1,8V s-au obținut cu regulatoare liniare de tensiune. Sursele de tensiune folosite sunt ilustrate în Fig. 3.10-1.

Programarea DAD s-a făcut folosind puntea USB-UART, reprezentată și ea în Fig. 3.10-1.

Partea centrală o constituie microcontroler-ul și memoria internă.

Pentru partea de comunicație au fost montate pe placă un modul radio și un port serial cu interfața necesară comunicării seriale.

Un DAD poate să facă parte dintr-o structură modulară. Astfel, conectorul de extensie din Fig. 3.10-1 poate fi folosit pentru a conecta mai multe plăci, cum este ilustrat în Fig. 3.10-2.

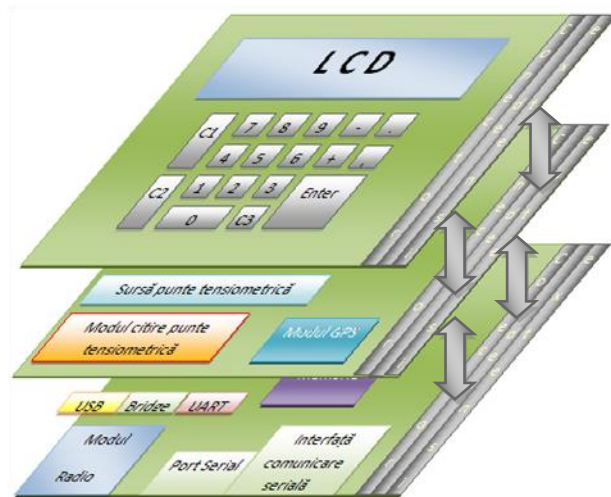


Fig. 3.10-2 Schema modulară cu mai multe plăci DAD

În această structură modulară, din Fig. 3.10-2, există mai multe plăci DAD suprapuse. Prima conține un LCD pentru vizualizarea și configurarea parametrilor. A doua placă conține un convertor performant pentru o analiză dedicată, cum ar fi o punte tensiometrică pentru măsurarea tensiunii din trolu, sau chip-uri specializate pentru măsurarea impulsurilor de la TIRO, necesare determinării vitezelor. În cazul din figură pe placă s-au amplasat o punte tensiometrică și un modul GPS. Ultima placă este identică cu cea prezentată în Fig. 3.10-1.

O primă observație o constituie utilizarea unei resurse la mai multe plăci, cum este cazul modulelor de comunicație sau al surselor de tensiune, care sunt amplasate doar pe ultima placă.

Placa 1 din structura modulară oferă o altă caracteristică a DAD-ului și anume aceea de a vizualiza și configura parametri.

Concluzia desprinsă din cele relatate este că Fig. 3.10-1 prezintă structura generală a unui dispozitiv de achiziția datelor, dar în practică se folosesc mai multe modele, diferențiindu-se în funcție de mărimea monitorizată și cerințele aplicației.

Atragem atenția că DAD nu este o componentă vandabilă într-un magazin de piese electronice. A fost proiectat special pentru scopul acestei lucrări și se află sub incidența confidențialității unui brevet de invenție [87].

Având în vedere dimensiunile mari ale sistemului, pentru o analiză completă, sunt necesare mai multe DAD-uri care să monitorizeze un număr mare de parametri. Pe sistem (excavator), se montează o rețea de dispozitive de achiziție, care realizează comunicarea radio sau prin intermediul unui protocol serial. În Fig. 1.3-1 este reprezentat excavatorul pe care s-a montat o rețea de DAD-uri.

Modulele de comunicație folosite de dispozitivele de achiziție pot forma mai multe rețele. Parametrii necesari monitorizării excavatorului, sau ai oricărui sistem, pot fi grupați iar grupele organizate în rețele separate, care comunică radio sau

serial. În sistemul de comunicație propus în acest brevet funcționează simultan mai multe rețele de comunicație, unele radio altele prin fir.

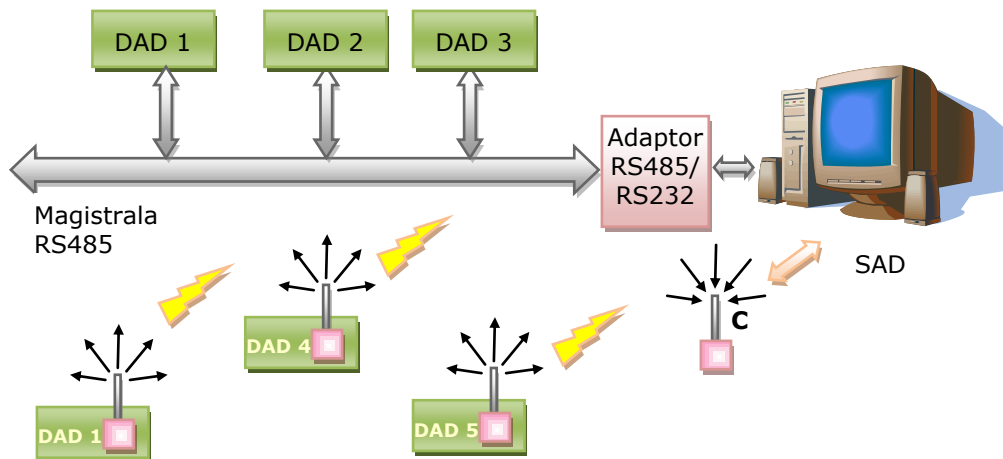


Fig. 3.10-3 DAD-uri grupate în două rețele

În Fig. 3.10-3 sunt reprezentate DAD-uri montate pe același sistem, dar grupate în două rețele diferite. Într-o rețea comunicare se face radio, iar în cea de-a doua serial.

Datele care circulă în rețele pot fi stocate separat sau împreună pe un calculator personal într-un fișier sau într-o bază de date.

O dezvoltare pe mai departe prevede posibilitatea vizualizării datelor pe o pagină de Internet.

3.11. Modul de funcționare al DAD

Senzorii eșantionează date la o rată mare, la fiecare 0.5ms. Datele achiziționate de la senzori sunt salvate într-un buffer intern circular, cum este cel prezentat în Fig. 2.7-2.

În funcție de tipul și caracteristicile parametrilor fizici monitorizați, datele din buffer sunt analizate și citite, la o rată relativ mică de eșantionare, de ordinul milisecundelor care poate ajunge până la minute sau ore, realizându-se o mediere, o valoare medie a eșantioanelor cuprinse între cele două citiri. Se urmărește eliminarea perturbațiilor.

La citire se înregistrează perechi formate din două elemente: valoare senzor și timpul la care a avut loc citirea. DAD-urile au fost proiectate pentru a se putea sincroniza între ele astfel încât toate mărimile achiziționate să poarte o ștampilă de

timp cu o origine comună. În acest fel datele achiziționate în întreaga RDEA pot fi ulterior corelate permițând studierea fenomenului care le-a provocat.

Un senzor este interogată când timpul prestabilit dintre două citiri succesive a expirat. Dacă s-a depășit o valoare de prag dată, se schimbă timpul dintre citiri, adică rata de citire între înregistrări, descrescând sau crescând considerabil. Volumul de date care poate fi înregistrat în unitatea de timp diferă de la un echipament la altul. În Fig. 2.7-2 consider rata de citire normală a datelor căsuțele colorate cu galben (gri deschis), iar o rată de citire în caz de alertă a căsuțelor colorate cu verde (gri închis). Fig. 2.7-2 prezintă un exemplu, succesiunea căsuțelor galbene și verzi nefiind schițată la scara reală.

Echipamentele au fost create atât pentru monitorizarea funcționării în condiții normale dar și pentru înregistrarea avariilor sau anomaliilor. Prin funcționare corectă înțelegem rularea în parametri stabiliți de producător: motorul să nu consume mai mult curent decât a fost proiectat, presiunea uleiului să nu crească sau temperatura în lagăre să nu depășească valorile limită. Având în vedere gabaritul utilajului dereglarea parametrilor duce la reacții de avarie în lanț. La orice ieșire din funcționarea normală, interogarea parametrilor poate să crească și de 10 ori. În acest caz încep să se acumuleze din ce în ce mai multe date în fiecare DAD. Volumul de date este dat de rata de eșantionare a fiecărui senzor. Limita maximă a datelor citite este dată de rata maximă de eșantionare. Considerând că există zeci, sute de parametri de înregistrat la care măcar a zecea parte au variații rapide, ceea ce presupune înregistrări cu timp de eșantionare mare, volumul de date generat în situații anormale este foarte mare.

În mod ideal, dacă rețeaua este bine dimensionată și protocolul de comunicație bine ales, se dorește ca transmisia tuturor datelor să se facă fără nici o pierdere. Mai mult, se dorește ca, în cazul în care apar erori în pachete, să fie retransmise și pachetele care au conținut erori.

Astfel se creează premisele analizelor din această teză și furnizarea unor soluții practice la metodele descrise mai sus.

3.12. Coexistența mai multor rețele

DAD-urile permit comunicarea radio sau serială. Pe un sistem de dimensiuni mari se poate monta un număr mare de senzori, de ordinul sutelor, dar nu toate informațiile lor trebuie fuzionate. Astfel se pot crea grupuri de senzori care împart monitorizarea pe subgrupe de urmărire. Grupurile de senzori pot fi asimilați de rețele independente de comunicație. În domeniul radio, există posibilitatea realizării mai multor rețele care să comunice în același timp, utilizând canale de frecvență diferite sau identificatori diferiți ai rețelei. Pe de altă parte, senzorii pot fi grupați și în rețele legate prin cablu. În Fig. 3.12-1 sunt ilustrate două rețele, una radio și una prin cablu care transmit date.

În acest fel s-a realizat o rețea hibridă, reprezentată în Fig. 3.12-1 .

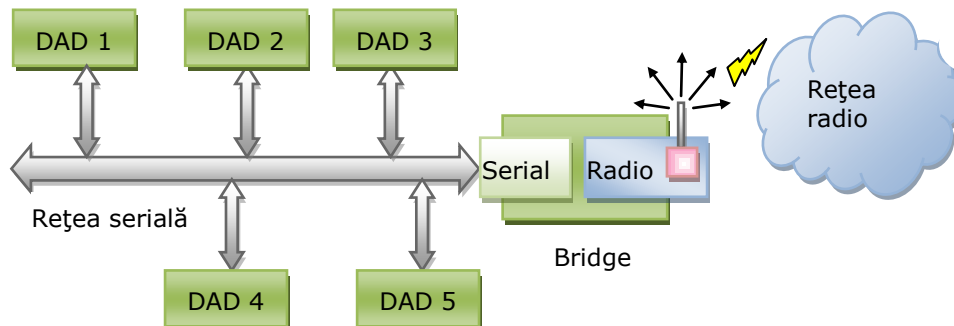


Fig. 3.12-1 Rețea hibridă formată dintr-o rețea serială și o rețea radio

Rețeaua hibridă este formată dintr-o rețea serială compusă din cinci DAD-uri, care comunică cu o rețea radio. Pentru comunicare se folosește un bridge realizat dintr-un DAD adaptat. Cu ajutorul modului radio se realizează comunicația în rețeaua radio, iar cu ajutorul modului serial se face legătura cu rețeaua serială. Prin configurare, rețeaua radio asimilează, „vede” întreaga rețea serială ca un nod radio cu mulți senzori.

3.13. Related work

Protocolul ZigBee și modulele XBee Series2 pe care le-am folosit în această teză fac subiectul multor studii prezente. Astfel se cercetează tehnici de securitate, încriptare/decriptare pentru rețele formate din module XBee.

Se analizează timpul de viață al bateriei modulelor în funcție de lucrul în regim economic (low-power) și cum variază puterea consumată la trezirea nodurilor.

Se cercetează fiabilitatea rețelelor datorită faptului că nodurile XBee se defectează mult mai repede decât nodurile unei rețele clasice. În rețelele radio de dimensiuni mari, se folosesc noduri anonime, iar defectarea lor nu implică înlocuire, sarcina lor fiind preluată de nodurile vecine.

Cercetarea rețelelor radio de dimensiuni mari urmărește gruparea nodurilor (clustering) în vederea furnizării unor informații concludente asupra evenimentelor (de exemplu riscul de avalanșă).

Alte ramuri ale cercetării le constituie studiile și elaborarea algoritmilor de rutare în rețelele radio de dimensiuni medii și mari.

Multe studii se ocupă de nodurile rețelei și de particularitățile lor. Este cazul aplicațiilor din medicină, sau din industria autovehiculelor sau avioanelor, unde în afara modulului XBee echipamentul folosit trebuie să aibă o acuratețe deosebită, fiind vorba de viața oamenilor[11].

Există aplicații multiple implementate care utilizează tehnologia ZigBee și modulele radio XBee. Dintre ele voi exemplifica câteva.

În domeniul medicinei, spitalele, clinicile sau laboratoarele au nevoie de informații rapide privind sosirea pacienților, antecedentele spitalizării sau

monitorizarea stării lor. Aceste informații ar trebui să poată fi accesate atât de la patul bolnavului, din cabinetul doctorului sau din diferite pavilioane ale spitalelor. În afară de acest aspect pur medical, tehnologia radio, bazându-se pe rețeaua Ethernet poate mări mobilitatea din spitale prin monitorizarea la distanță a echipamentelor medicale și notificarea automată a defectării unui aparat cum ar fi cel de dializă sau a unui ecograf, consultarea automată a aplicațiilor financiare pentru întocmirea notelor de plată ale spitalizațiilor, salvarea automată a evoluției bolnavului nefiind necesară introducerea manuală a rezultatelor analizelor sau observațiilor medicilor [50].

În Fig. 3.13-1 este prezentată o rețea de dispozitive radio care transmite date medicale în cabinetul unei asistente medicale. Informațiile sunt culese de la mai multe EKG-uri. Asistenta nu mai trebuie să meargă din salon în salon și își poate da seama imediat de îmbunătățirea sau înrăutățirea stării unui bolnav.



Fig. 3.13-1 Rețea ZigBee într-un spital [50]

Cercetarea privește și în față, în viitor. Se cercetează mult aplicații legate de domeniul casnic cum ar fi verificarea consumului, particularizarea preferințelor sau integrarea funcționalităților.

Verificarea consumului permite folosirea unor dispozitive care indică energia consumată din întreaga casă. Sunt folosite diferite grafice colorate sau rezultatele pot fi salvate într-un fișier pe calculator. Mai multe detalii despre aceste dispozitive se găsesc în [42].

Prin particularizarea preferințelor fac o analogie cu facilitățile oferite în industria auto. Un automobil poate reține profilurile auto ale posesorilor. Astfel sunt memorate mai multe profile. Un profil conține aranjarea scaunului, a oglinzilor, a temperaturii, a postului de radio preferat. Când un posesor se apropie de mașină, aceasta îi detectează identitatea și-i încarcă profilul, realizând automat setările descrise. În același fel la intrarea în casa viitorului aceasta detectează cine a venit și în funcție de identitatea persoanei reglează temperatura, pornește muzica, trage sau nu draperiile, aprinde lumina la intensitatea preferată. Având în vedere că în

casă pot locui mai multe persoane, se pot crea profile comune, îmbinând preferințele mai multor persoane, în mod armonios.

Prin integrarea funcționalităților înțeleg pornirea/oprirea luminilor pe bază de voce, pornirea/oprirea televizoarelor, a mașinii de spălat, a cuptorului. Un astfel de exemplu este ilustrat în Fig. 3.13-2. Toate serviciile exemplificate se pot face și la distanță. Înainte de a pleca de la servici am porni cuptorul și când ajungem acasă prăjitura este gata. Dacă la plecarea în concediu am uita luminile aprinse, am avea nevoie doar de o conexiune la Internet: accesez site-ul familiei și le pot stinge, sau pot să văd ce mai e pe acasă, sau pot să ud florile sau să hranesc peștii.

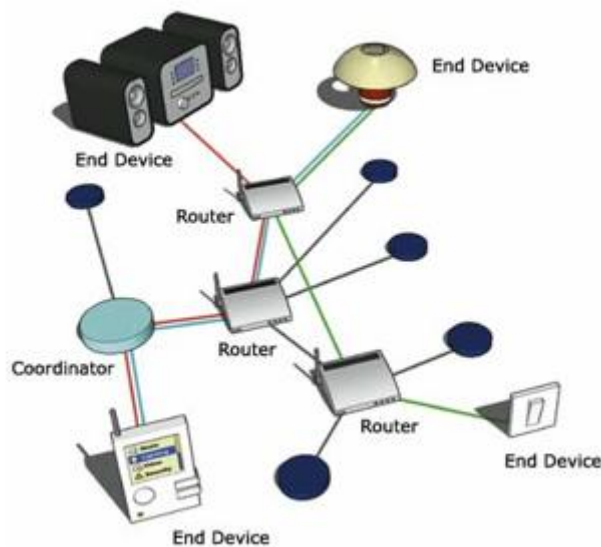


Fig. 3.13-2 Rețea ZigBee folosită în domeniul casnic[42]

Un alt scenariu îl constituie o rețea radio de pe terenul agricol al unui fermier, să spunem din Statele Unite. El are un sistem de irigații comandat electronic. La ora 4-5 dimineața senzorii citesc gradul de umezeală din pământ și umiditatea din aer. În funcție de parametri citiți comandă cantitatea de apă sistemului de irigații. Senzorii de pe teren sunt grupați într-o rețea mesh. La defectarea unora dintre ei traseele se reconfigurează iar informația globală se păstrează [51]. Aceste sisteme sunt implementate în câteva țări din lume. În Statele Unite s-a implementat o rețea de dispozitive de monitorizare, întinsă pe mari dimensiuni. Aceasta urmărește căldura degajată din pământ, umiditatea și furnizează date care determină cantitatea de irigații necesară terenului. Rețeaua proiectată este multi-hop și folosește protocolul de comunicare ZigBee. Fiecare nod al rețelei, este dotat cu o mică baterie solară. Soluția furnizată este ieftină și extrem de fiabilă [79].

Un alt proiect care este implementat și funcționează îl descriu mai departe. Proiectul utilizează cele mai recente descoperiri din cadrul dispozitivelor cu senzori multifuncționali și aplicabilitatea lor în rețele radio care fac legătura dintre internet și fenomene care apar în domeniul agricol, în cadrul fermelor. Se efectuează măsurători ale condițiilor atmosferice și climatice, combinate cu sisteme de

poziționare (GPS) și rezultatele sunt furnizate on-line. Pentru comunicare s-au folosit rețele radio care utilizează protocolul ZigBee. Cercetarea a fost aplicată în trei locații: o podgorie, în regiunea agricolă a statului Washington unde există deja două rețele radio de dimensiuni mari, una pentru vreme, iar cealaltă dedicată operațiilor agricole descrise în exemplul anterior; și în câteva orașe din China, Noua Zeelandă și Australia, care pot furniza pe Internet prognoza meteo la fiecare 60 de minute. Un nod al rețelei conține mai mulți senzori: dintre care amintesc viteza și direcția vântului, umiditatea, radiația solară, presiunea atmosferică, temperatura din aer și din sol, poluarea atmosferică (CO₂) [80]. Comunicarea se realizează folosind tot protocolul ZigBee.

Un proiect pionier de actualitate se desfășoară în pădurile din Statele Unite și constă în monitorizarea fenomenelor din mediul înconjurător folosind rețele de senzori. Cercetătorii se confruntă cu realizarea eșantionării dar și cu problematica apariției unor variații neprevăzute. Pentru realizarea acestor deziderate s-au folosit algoritmi adaptivi care răspund corect variabilității mediului cât și evenimentelor discrete raportate de senzori. Sistemul propus, NIMS (Networked Infomechanical Systems), este format atât din senzori statici, cât și mobili și urmărește impactul fenomenelor care apar și se desfășoară în atmosfera de deasupra unei păduri. Proiectul încearcă să aproximeze impactul fluxului de dioxid de carbon asupra biomasei pădurii. Algoritmul de eșantionare adaptivă propune o structură ierarhică, formată din grupe (cluster) de senzori statici. Pentru monitorizare există și câțiva senzori mobili. Fiecare grupă analizează informațiile acumulate și furnizează o valoare. Acea valoare este comparată cu valorile oferite de senzorii mobili, și după caz, se face sau nu o ajustare. [78]

Continuând în aceeași manieră, prezint proiectul "orașul însuflețit", descris în [41], unde clădirile unui oraș vor comunica între ele, fiind noduri ale unei rețele radio. Senzorii vor fi încorporați în pereți și am putea spune că în acest fel clădirile "prind viață", dând informații atât despre nevoile din interiorul clădirilor, cât și despre mediul înconjurător: calitatea aerului, temperatură, poluare.

Desigur, nu toate proiectele futuriste trebuie să fie sobre.

În proiectul prezentat în [40], cercetătorii intră în universul complex și destul de necunoscut deocamdată al plantelor, analizând nevoile lor, limbajul lor și traducându-l pentru noi, oamenii, prin metode accesibile nouă: mesaje, telefoane sau imagini.

Un alt proiect futurist este exemplificat în [44,43] și presupune existența unui dispozitiv mobil, în care se poate sufla. La suflarea în acest dispozitiv se pot realiza diferite acțiuni: se aprind luminile, se umflă un balon, pornește un ventilator. Am apreciat ingeniozitatea proiectului.

Voi considera exemplul radio din Fig. 3.13-1, cu o rețea de EKG-uri dintr-un spital. La un moment asistenta care monitorizează starea pacienților primește un apel de urgență de la pacientul P. Merge la el în salon și aparatul indică totul normal. Se întoarce în cabinet și mai primește un apel de urgență de la același pacient P. Merge din nou la salonul lui unde totul este normal. La întoarcere este se întâlnește cu sora pacientului Z din alt salon care o întreabă dacă nu a primit două apeluri de urgență de la pacientul Z. Îi răspunde că a primit, dar îl indicau pe pacientul P. Aceasta reprezintă o limitare a protocolului ZigBee.

Voi explica ce s-a întâmplat de fapt.

Așa cum am arătat în paragraful 3.5.4 modulelor radio li se alocă două feluri de adrese: una permanentă pe 64 de biți și una temporară la intrarea în rețea, pe 16 biți. Dacă un nod părăsește rețeaua și se reconectează, poate primi o adresă pe 16 biți diferită de cea inițială.

În exemplul din Fig. 3.13-1 considerăm coordonatorul C conectat direct la calculatorul personal din cabinetul asistentei. Router-ele sunt montate în tavanul saloanelor. EKG-urile conțin noduri terminale încorporate în ele. EKG-ul 1 are adresa extinsă 0x123456789AB0001 și adresa la intrarea în rețea este 0x6789, iar EKG-ul 2 are adresa extinsă 0x123456789AB0002 și adresa la intrarea în rețea este 0xABCD.

Considerăm situația în care coordonatorul C1 trebuie înlocuit.

Dacă noul coordonator C2 își alocă alt canal sau alt PAN, atunci toate nodurile trebuie să se reconecteze (join) la rețea. Dar dacă după înlocuire noul coordonator C2 își alocă același canal și PAN pe care le folosea vechiul coordonator C1 nodurile nu efectuează nici o acțiune. În acest caz, nodurile EKG1 și EKG2 continuă să funcționeze. La un moment dat o nouă stație EKG3 se conectează în rețea, având adresa extinsă 0x123456789AB0009. Fiecare coordonator are un spațiu de adrese de rețea pe care le utilizează. La intrarea nodului EKG3 în rețea, coordonatorul C2 își verifică spațiul de adrese și observă că nu a alocat nici o adresă. Atunci îi alocă nodului cu EKG3 adresa 0xABCD, care există în rețea, dar a fost alocată de coordonatorul C1.

În acest moment, în rețea există 2 noduri care au adresa extinsă diferită, iar adresa de rețea identică: 0xABCD.

Această situație poate apărea, într-o varietate de scenarii. Un nod din rețea poate primi mai multe adrese de rețea dacă i se schimbă părintele.

Concluzia este că în rețelele ZigBee adresarea se face folosind adresa de rețea pe 16 biți și este recomandată utilizarea sau corelarea și cu adresa extinsă pe 64 de biți. Acest lucru este dificil de realizat într-o rețea cu 20, 100 sau 200 de noduri.

ZigBee Alliance urmărește eliminarea acestui inconvenient și într-o versiune viitoare (release) va folosi la transmisia datelor doar adrese extinse în locul celor pe 16 biți.

Alte probleme care pot apărea sunt cele legate de transmisia radio (wireless). Pot apărea probleme în condiții meteorologice defavorabile: ploaie, viscol, ceață. Frecvența liberă de 2.4GHz coincide cu frecvența de rezonanță a apei. Pentru această frecvență lungimea de undă este de doar câțiva centimetri ($\lambda * f = 300$) iar undele sunt absorbite de ploaie. Frecvența de 2,4 GHz este supusă interferențelor de la instalațiile radar sau cuptoarele cu microunde. În plus semnalul radio poate fi reflectat și recepționat pe mai multe căi, datorită reflecțiilor. Pentru toate aceste probleme există soluții, dar acestea necesită efort și resurse suplimentare.

3.14. Analiza comparativă a algoritmilor prezențați

În acest capitol s-au pus bazele teoretice ale tezei de față, din acest motiv voi puncta contribuțiile personale ale doctorandei la acest nivel de reprezentare. Contribuțiile s-au realizat atât în domeniul software, hardware, dar și în domeniul gestionării și configurării rețelelor.

Având în vedere specificul RDEA și constrângerile pentru realizarea unei eșantionări simultane la milisecundă nu s-a putut aplica un algoritm existent, fiind necesară realizarea metodelor de comunicare prin propunerea algoritmilor din paragraful 3.3. Algoritmii propuși sunt dinamici și se aplică atât comunicării radio cât și seriale. RDEA are caracteristicile unui sistem adaptiv, iar algoritmii realizează această cerință. Metoda de rezolvare propusă în această teză face parte dintr-un brevet de invenție[87].

Structura pachetului de date prezentat în paragraful 3.2 este specifică cerințelor proiectului. Nodul unei rețele poate monitoriza mai mulți senzori identici sau diferiți. Acest lucru trebuie specificat la transmiterea datelor, pentru identificare. Din punctul de vedere al fiabilității este recomandată transmiterea mai multor eșantioane grupate într-un singur pachet.

La transmiterea eșantioanelor trebuie să se aplice o ștampilă de timp pentru realizarea cronologiei citirilor. Ștampila de timp are acuratețe la nivel de milisecundă, o cerință destul de rară în aplicațiile industriale uzuale.

Nodurile RDEA monitorizează un sistem eterogen, iar datele culese fuzionează în analiză. Din acest motiv nodurile trebuie sincronizate. În paragrafele 2.4 și 3.4.4 au fost analizați algoritmii de sincronizare existenți și s-a demonstrat de ce nu pot fi folosiți ca atare în acest sistem. Am propus trei algoritmi de sincronizare, bazați pe aceeași structură, dar deosebiți prin perspectiva implementării. Toți algoritmii realizează o sincronizare sursă-receptor prin transmiterea a două mesaje de sincronizare. Se creează o ierarhie a nodurilor rețelei la crearea ei și la cerere, de oricâte ori este necesar. Nodurile rețelei se sincronizează la un interval de timp stabilit euristic în urma testelor și analizei, numit sincronizare generală, dar sincronizarea se poate realiza și la cerere.

Primul algoritm de sincronizare determină centrele de sincronizare apriori și le memorează în cadrul informațiilor de configurare la crearea rețelei. Pentru siguranță sunt memorate 2 noduri GPS, iar în cazul în care acestea devin inoperabile, sincronizarea se face după ceasul calculatorului personal.

Al doilea algoritm creează la configurarea rețelei o ierarhie a nodurilor, care se stochează pe calculatorul personal, iar acesta transmite nodurilor informații stricte, necesare sincronizării, conservând astfel resursele nodurilor. Ierarhia se poate recrea la cerere.

Al treilea algoritm pornește de la structura arborescentă clasică, pe nodul rădăcină este instalat un GPS, care este conectat și la calculatorul personal. Sincronizarea tuturor nodurilor se face în raport cu nodul părinte.

În domeniul gestionării și configurării rețelelor am realizat practic implementarea algoritmilor și a metodelor prezentate mai sus.

Am creat rețele funcționale care transmit date radio și rețele care transmit serial.

S-au creat rețele radio și seriale care funcționează împreună.

În această teză am prezentat și o variantă hibridă de rețea, Fig. 3.12-1, în care prin configurarea creată, într-o rețea radio se transmit date de la și spre o rețea serială, care este privită ca un nod al rețelei radio.

Pentru implementarea practică a RDEA nu am găsit un dispozitiv de achiziția datelor care să corespundă cerințelor. Din acest motiv am creat DAD-ul prezentat în acest capitol.

DAD-ul a fost proiectat la un nivel de abstractizare ridicat, permitând adaptarea lui pentru eșantionarea datelor de la tipuri diferite de senzori, dintre care amintesc viteză, accelerație, temperatură, lumină, GPS, tensiune, curent. Așa cum este dimensionat, DAD-ul poate citi practic orice senzor.

DAD-ul poate fi adaptat și se poate crea o structură modulară cum este cea prezentată în Fig. 3.10-2, în care se folosesc resurse comune, cum ar fi surse de tensiune, antena radio sau comunicarea serială pentru mai multe plăci deodată. Numărul senzorilor citați în acest caz este mai mare decât în cazul unei singure plăci.

În Fig. 3.10-2 prezint un DAD modificat cu ajutorul căruia se pot vizualiza pe un LCD și configura parametri cu ajutorul unei mici tastaturi.

În paragraful 3.12 este prezentat un DAD modificat care are funcția de punte (bridge) între o rețea serială și una radio, realizând această funcție prin utilizarea a două module radio montate pe aceeași placă.

Un alt avantaj al DAD-ului îl constituie posibilitatea comunicării radio sau serială.

Am realizat abstractizarea comunicațiilor de la nivelul senzorilor și al plăcii de achiziție. Indiferent de modul de accesare al datelor: printr-o magistrală de comunicație 1-Wire, sau magistrală de tipul I2C sau SPI, sau citirea valorilor de la un convertor analog-numeric sau de la un dispozitiv rezistiv, la nivelul fiecărui DAD există primitive generale de citire a datelor care fac transparentă utilizatorului comunicarea cu diferitele tipuri de senzori.

Configurarea unui DAD cuprinde aplicația care citește simultan eșantioanele de la unul sau mai mulți senzori identici sau diferiți, formează cadre de date și le transmite în rețea. După crearea pachetelor acestea sunt transmise în rețea. Se analizează pachetele de confirmare. Pentru buna desfășurare a comunicației în rețea, DAD-urile sunt sincronizate în medie o dată la 10-15 minute, așa cum am arătat în paragraful 3.7.

Contribuțiile prezentate aici au fost desprinse în urma prezentării teoretice din capitolul 3. Pe parcursul tezei voi adăuga alte contribuții, acelea cuprinzând partea aplicativă, practică a cercetării realizate.

Tendența nu este de a crea noi protocoale, de aceasta se ocupă marile corporații care le valorifică transformându-le în standarde. O ramură a cercetării actuale se ocupă de reunirea diferitelor piese existente: module, microcontrolere, protocoale; și precum niște copii talentați crează din cărămizi lego obișnuite jucării surprinzătoare și creatoare, tot astfel pot explica originalitatea acestei teze.

De fapt aici nu am inventat nimic nou. Noutatea și originalitatea constau în adunarea laolaltă și îmbinarea, care a necesitat căutări și analize, a dispozitivelor hardware și crearea aplicațiilor care să le facă să funcționeze împreună.

Pornind de la ideea de a dimensiona o rețea de senzori și până la furnizarea unei soluții optimale am realizat multe studii. După cristalizarea și elaborarea metodei de comunicație au urmat mai multe etape. Fiecare pas a fost analizat: pornind de la comunicare: să fie prin fir sau radio, continuând cu senzorii, apoi DAD-urile, analiza protocoalelor și în final analiza rețelei prin benchmark.

3.15. Concluzii

Luând în considerație specificul aplicațiilor cărora le este destinată rețeaua, am dedicat prezentul capitol analizei metodelor de comunicare la diverse niveluri.

Astfel, la nivelul aplicației am investigat structura cadrelor de date, precum și a celor de confirmare, sincronizare și răspuns la o cerere de sincronizare. Specificul rețelei este că poate monitoriza mai mulți senzori, fapt care se impune marcant la transmisia datelor în vederea identificării acestora. De asemenea, în scopul creșterii fiabilității transmisiei, trimiterea mai multor eșantioane se impune grupată într-un singur pachet. Eșantioanelor li se aplică o ștampilă de timp creându-se cronologia citirilor. Ștampila de timp se menține la acuratețea de 1 ms, revendicată cu aplicația noastră, ceea ce este neîntâlnit în aplicațiile uzuale industriale.

Drept consecință, având în vedere specificul RDEA și constrângerile pentru realizarea unei eșantionări simultane, la milisecundă, nefiind posibilă aplicarea algoritmilor existenți, am propus algoritmi originali de comunicare care, în mod dinamic, pot fi aplicați atât comunicației radio cât și comunicației cablate de tip serial.

Menționez că soluția originară adoptată face obiectul unui brevet de invenție[87].

Caracterul eterogen al nodurilor RDEA impune o funcționare sincronizată a cărei implementare, așa cum a fost prezentată prin algoritmi descriși în capitolul anterior, nu poate fi realizată.

Drept consecință am propus trei algoritmi de sincronizare caracterizați prin faptul că realizează o sincronizare sursă – receptor prin transmiterea a două mesaje.

Caracteristic algoritmilor este faptul că se creează o ierarhie a nodurilor rețelei, iar sincronizarea are loc la un interval de timp stabilit ca optim euristic, drept consecință a analizei efectuate.

Algoritmii implementează în general aceeași soluție, dar sunt diferiți prin prisma implementării. Astfel, primul algoritm de sincronizare determină centrele de sincronizare a priori și le memorează în cadrul informației de configurare la crearea rețelei.

Cel de-al doilea realizează o ierarhie a nodurilor care este stocată pe un calculator personal, iar acesta transmite nodurilor doar informațiile necesare sincronizării, conservând resursele nodurilor.

Ultimul algoritm, cel de-al treilea, are la bază faptul că pe nodul rădăcină este instalat un sistem de poziționare globală (GPS) care este conectat la un calculator personal. Sincronizarea celorlalte noduri se face în raport cu nodul părinte.

Trecând apoi la nivelul de rețea comunicația se poate realiza, așa cum am mai menționat, prin radio sau cablat serial.

La transmisia radio se folosește metoda de rutare AODV.

În vederea configurării RDEA sunt date precizii legate de dispozitivele de achiziția datelor. Acestea sunt proiectate la un nivel de abstractizare înalt, ele permițând adaptarea pentru eșantionarea datelor de la diferite tipuri de senzori.

DAD-ul poate fi realizat pe mai multe plăci, dar uzual el folosește o singură placă.

De asemenea DAD-ul poate fi realizat și modificat sub forma unei punți între o rețea și una serială.

Există mai multe moduri de a accesa datele, iar DAD-ul face transparentă utilizatorului comunicarea între diversele tipuri de senzori.

Caracteristic DAD-urilor este faptul că citesc simultan eşantioane de la unul sau mai mulți senzori de același tip sau de tipuri diferite și formează cadre de date pe care le transmit în rețea.

Reamintim că configurarea aparte a DAD-urilor pentru a corespunde aplicației obiectiv a tezei face parte dintr-o cerere de brevet de invenție, deja menționată [87].

4. IMPLEMENTAREA OPTIMALĂ, PRIN PRISMA PERFORMANȚĂ-FIABILITATE, A METODELOR DE COMUNICARE ÎN REȚELE DE ACHIZIȚIA DATELOR, CU APLICABILITATE LA EXCAVATOARELE CU CUPE

În capitolul 3 am prezentat metodele teoretice pentru realizarea dezideratelor propuse în această teză. Acest capitol, 4, prezintă implementarea propriu zisă a metodelor prezentate în capitolul anterior, ilustrând configurarea rețelei, urmată de configurarea unui DAD.

În cadrul implementării au existat aplicații la două niveluri. Primul nivel l-a constituit nodul unei rețele, adică un DAD, iar al doilea nivel a fost protocolul de comunicație în rețea.

Implementarea unui nod, DAD, în funcție de mărimile fizice care sunt monitorizate, are partea de achiziție diferită pentru fiecare tip de senzori folosiți. Modul de comunicare al senzorilor cu microcontroler-ul diferă, fiecare grupă de senzori comunicând diferit. Aplicația realizează gestionarea achiziției și a comunicării în mod transparent utilizatorului, acesta folosind funcții și task-uri la nivel înalt, care uniformizează comunicarea.

La nivelul rețelei, bazându-mă pe protocolul radio sau serial oferit de mediul de lucru, am construit aplicația de comunicare. Deși structural, algoritmi sunt identici, la implementare a trebuit să țin seama de caracteristicile și de serviciile oferite de fiecare protocol în parte.

4.1. Analiza soluțiilor existente prin prisma parametrului vibrații ca element critic al funcționării fiabile a excavatorului

Pe excavatoarele cu cupe există rețele de senzori care implementează sisteme de comandă și control, furnizând informații directe operatorului de la pupitrul excavatorului. Aceste sisteme anunță operatorul dacă manevrează utilajul în limite normale, indicând pe un ecran mai mult sau mai puțin sofisticat limitele de deplasare. Sistemele propun diferite traiectorii sau valori ale vitezei, ale unghiurilor de excavare și a altor parametri cu diverse scopuri dintre care amintesc doar mărirea cantității de minereu sau micșorarea consumului de combustibil. Astfel de sisteme sunt propuse în invențiile [70,71], pentru excavatoare obișnuite, dar se pot aplica și la excavatoarele miniere. Scopul invențiilor este de a realiza controlul automat al excavării bazat pe o direcție prestabilită de utilizator. În plus asigură controlul cât mai precis al funcționării, poziționare optimă în spațiul de lucru și un consum minim de energie.

Sistemul de monitorizare propus de mine nu este implementat până în prezent pe un excavator cu cupe. Sisteme asemănătoare celui propus de mine se

90 4 - Implementarea optimală, prin prisma performanță-fiabilitate, a metodelor de comunicare în rețele de achiziția datelor, cu aplicabilitate la excavatoarele cu cupe găesc în domeniul medical, în industria automobilistică, la avioane (cutiile negre), la hidrocentrale sau centrale atomice. Scopul lor este de a monitoriza și de a crea un istoric al funcționării. Sistemul propus de mine monitorizează mai mulți parametri fizici, acordând o atenție specială vibrațiilor. Măsurarea vibrațiilor și analiza lor este un subiect de actualitate. Voi da câteva exemple.

Unul dintre cel mai simplu mod de a măsura vibrațiile unui motor este propus în invenția [73] care prevede un dispozitiv care monitorizează starea de funcționare a unui motor prin intermediul unui senzor și transmite datele radio pe distanțe scurte, la un dispozitiv aflat pe centura operatorului.

Invenția [72] propune o metodă elaborată pentru măsurarea vibrațiile unui dispozitiv operațional, care poate fi un excavator obișnuit. Se folosește un sistem cu microcontroler care conține un senzor de vibrație și un dispozitiv de comunicare. Senzorul de vibrație reprezintă un accelerometru pe trei axe. Datelor de la accelerometru de pe cele 3 axe, li se aplică o funcție de filtrare individuală pentru fiecare axă. Rezultatele sunt comparate cu praguri de limitare, iar rezultatele sunt transmise șoferului și afișate pe un ecran [72].

O altă invenție, prezintă o rețea de dispozitive care măsoară vibrații, realizând un sistem predictiv de mentenanță pentru un grup de centrale eoliene. Sistemul include secțiunea de control a centralelor, rețeaua de comunicare și algoritmi care implementează sistemul. Sistemul de monitorizare este realizat din accelerometre montate pe turbinele eoliene care transmit date sistemului de supraveghere și control. Deciziile de mentenanță predictivă sunt luate în urma analizelor survenite de la sistemul de supraveghere și control [74].

Scopul cercetării mele a fost realizarea cadrului de implementarea a unei soluții care să studieze vibrațiile și alți parametri pentru un excavatorul cu cupe.

În timpul cercetării, am participat la diverse proiecte de studiu, în urma cărora au rezultat câteva articole, dintre care amintesc [85] și [86].

În cadrul articolului [85] se creează o metodă de procesare a imaginilor pentru analiza vibrațiilor pentru structuri mari, cum ar fi excavatorul din Fig. 1.3-1.

Scopul cercetării a fost determinarea frecvenței proprii de oscilație a unei structuri mari, care poate fi brațului excavatorului sau o paletă de elice. Pentru aceasta se montează pe fiecare din structurile studiate o miră marcată la fiecare 50mm. Mira reprezintă pentru cercetare o țintă. Pe suprafața mirei se aleg unul sau mai multe puncte și li se rețin coordonatele. Cu ajutorul unei camere de filmat se fixează și se urmăresc punctele de pe miră. Structura (excavatorul sau elicea) este pusă în mișcare iar camera video înregistrează mișcarea punctelor în timp. Datele de la cameră sunt stocate într-un fișier cu extensia .avi.

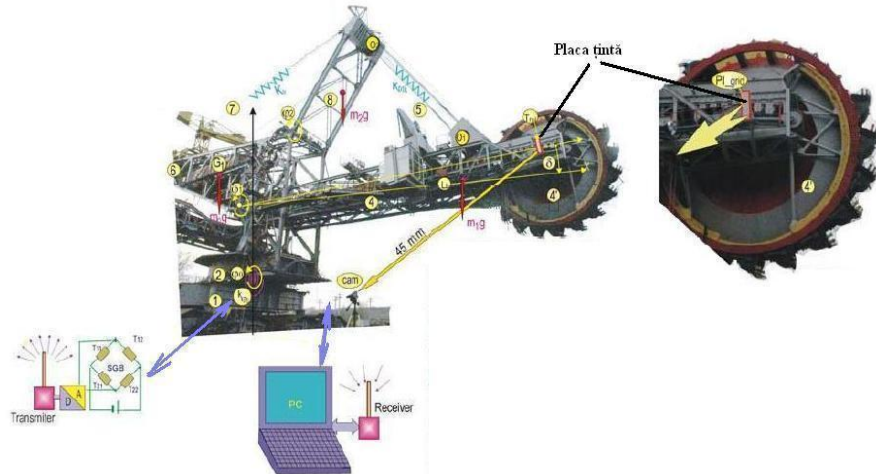


Fig. 4.1-1 Amplasarea senzorului și a unei camere de filmat pe excavator

Grupul de cercetare analizează filmul cadru cu cadru folosind un program grafic cum ar fi Corel R.A.V.E. Din analiză sunt extrase toate pozițiile punctului sau ale punctelor stabilite inițial. Valorile extrase sunt analizate. Se creează reprezentări grafice.

Metoda este eficientă și relativ ieftină față de alte soluții existente cu măsurători folosind laserul sau utilizarea unor senzori seismici, piezorezistivi sau capacitivi. În plus, nu necesită condiții favorabile de vreme, filmarea putându-se realiza chiar și cu un cer înnorat.

Această metodă permite determinarea vibrațiilor dintr-un interval de frecvență redus, între 0.3 Hz și 3 Hz; permițând observarea unor elongații la nivel de 10mm, 3 mm sau chiar sub 0.5 mm, ceea ce reprezintă o precizie optimă pentru experimentul propus.

Metoda poate fi îmbunătățită folosind 2 camere sau mai multe camere de filmat sincronizate. O cameră rămâne fixă urmărind ținta, iar celelalte urmăresc mai multe zone țintă diferite. În acest fel se poate determina mișcarea întregii structuri mastodontice. Pentru mărirea acurateței sincronismului, camera video principală poate fi înlocuită cu o punte tensiometrică, vezi Fig. 4.1-1, care transmite radio unui receptor și de la el computerului. Semnalul recepționat de la puntea tensiometrică conține atât componenta statică cât și cea dinamică a mișcării și reprezintă tensiunea mecanică exercitată asupra dispozitivului în timpul funcționării.

La acest proiect am participat cu recepționarea datelor de la puntea tensiometrică, transmisia și stocarea lor pe computer, Fig. 4.1-2.

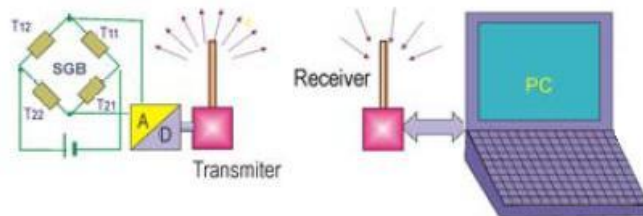


Fig. 4.1-2 Puntea tensiometrică și modulele radio

92 4 - Implementarea optimă, prin prisma performanță-fiabilitate, a metodelor de comunicare în rețele de achiziția datelor, cu aplicabilitate la excavatoarele cu cupe

Eșantionarea datelor punții tensiometrice s-a făcut la 20 ms, datele citite având rezoluția de 24 de biți (3 octeți).

În Fig. 4.1-3 detaliez electronica pentru puntea tensiometrică folosită și pentru DAD-ul utilizat.

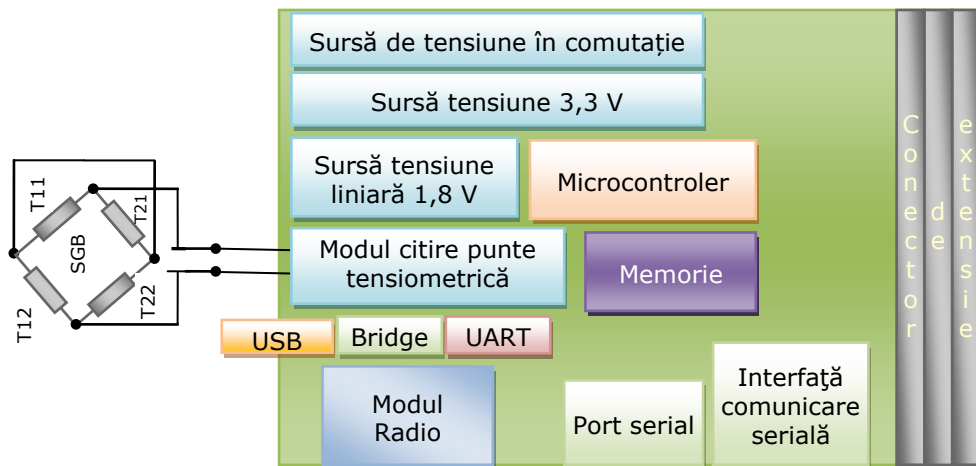


Fig. 4.1-3 Puntea tensiometrică și DAD-ul dedicat

Puntea tensiometrică este formată din 4 rezistențe T_{11} , T_{12} , T_{21} și T_{22} , având fiecare valoarea de 120 Ω . Cu ajutorul punții tensiometrice s-a măsurat atât deformarea, cât și curbarea structurii. În Fig. 4.1-4 am ilustrat modul în care a fost montată puntea tensiometrică pe structura metalică.

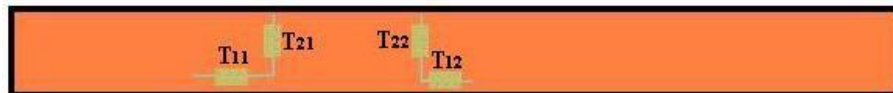


Fig. 4.1-4 Montarea punții tensiometrice pe structura metalică

Pentru transmiterea datelor am folosit module de comunicație XBee, descrise în această lucrare, în paragraful 3.5.

Receptorul radio primește datele de la puntea tensiometrică și le salvează pe computer într-un fișier text. Fișierul are un header cu următoarele câmpuri: descriere acțiune, data și ora la care a pornit eșantionarea datelor și data și ora la care s-a oprit măsurătoarea. După header, în fișier sunt trecute datele recepționate.

4.1 - Analiza soluțiilor existente prin prisma parametrului vibrații ca element critic al funcționării fiabile a excavatorului 93

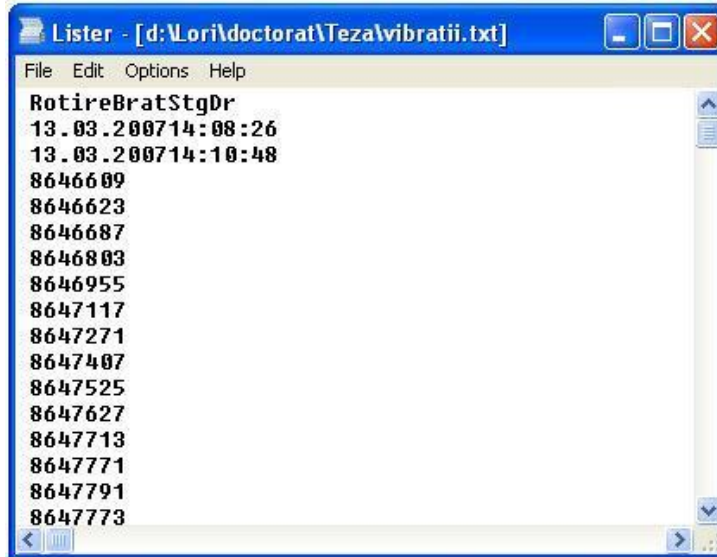


Fig. 4.1-5 Fișier cu date înregistrate de la puntea tensiometrică

În continuare valorile sunt extrase și analizate folosind aplicația MatLab.
În urma prelucrării se realizează grafice care sunt interpretate și din ele sunt extrase informații.

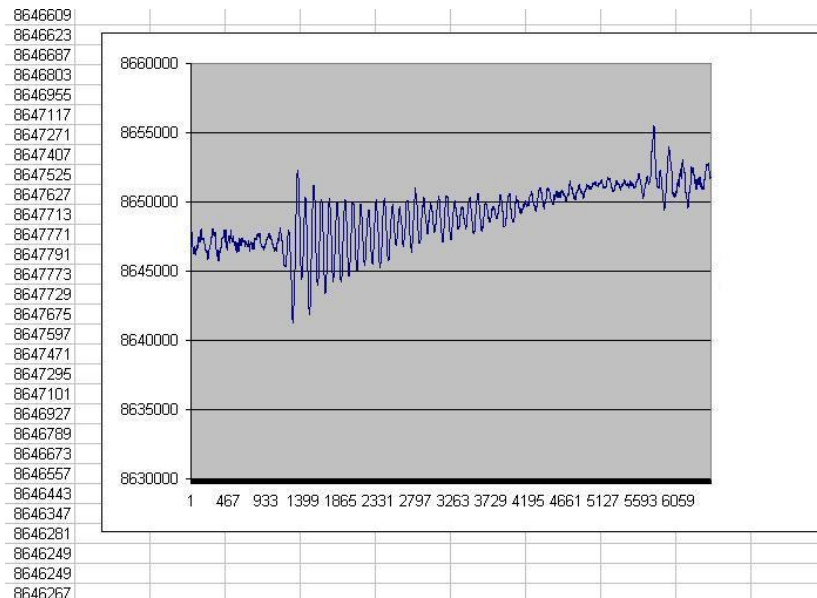


Fig. 4.1-6 Graficul mărimilor din fișierul de date

În Fig. 4.1-6 este reprezentat graficul rezultat în urma citirii datelor din fișierul ilustrat în Fig. 4.1-5.

94 4 - Implementarea optimală, prin prisma performanță-fiabilitate, a metodelor de comunicare în rețele de achiziția datelor, cu aplicabilitate la excavatoarele cu cupe

Formula (8) din articolul [85] reprezintă ecuația diferențială a mișcării excavatorului.

$$[M]\{\ddot{\varphi}\} + (c_1[K_0] + c_2[K_{pltk}] + c_3[K_{tr}])\{\dot{\varphi}\} + ([K_0] + [K_{pltk}] + [K_{tr}] + [K_G])\{\varphi\} = \{Q_{Fc}\} + \{Q_G\}$$

Ecuția diferențială a mișcării excavatorului

Ecuția diferențială a mișcării excavatorului guvernează mișcarea excavatorului, vectorul deplasărilor conținând componente modale.

Aceste componente modale se regăsesc și în semnalele și reacțiunile de pe cele trei reazime, iar una dintre aceste reacțiuni este măsurată cu sistemul propus în articolul [85] și a căror valori sunt ilustrate în Fig. 4.1-5. Forma grafică este reprezentată în Fig. 4.1-6.

Mișcările de vibrație sunt numite moduri naturale de vibrație. Asupra datelor se realizează o analiză în amplitudine care furnizează starea de încărcare și o analiză în frecvență care furnizează frecvențele proprii ale componentelor.

Din cadrul analizei se evidențiază două componente, una statică și una dinamică. Pentru echilibrarea sistemului se folosește componenta statică, care determină stabilitatea statică.

Ecuția diferențială a mișcării excavatorului, prin integrare, furnizează legile mișcării, care conțin componente ale modurilor naturale de vibrație, dintre care doar 2-3 sunt importante în analiză. Prin analiza lor se determină dacă vibrațiile sau alți parametri funcționează în limite normale. Acesta este scopul articolului.

În continuare voi prezenta câteva dintre detaliile prelucrării datelor în MatLab. $\varepsilon m1$ și $\varepsilon m2$ reprezintă medieri ale valorilor de pe grafic.

```

ORIGIN = 1
s := READPRN("01.txt") rotire de la z=11 la z=22
n := last(s)    n1 := 250    n2 := 250
i := 1..n - 3
                                Δt := 1/50
εj := si+3 - 8.269 × 106
m := min(ε)
εm1 := 1/n1 · ∑K=1n1 εK
m = -5233
εm2 := 1/n2 · ∑K=1n2 εn-3-K
εm1 = -329.104    εm2 = -2098.472

```

Fig. 4.1-7 Secvență din MatLab care prelucrează datele din fișierul prezentat în Fig. 4.1-5

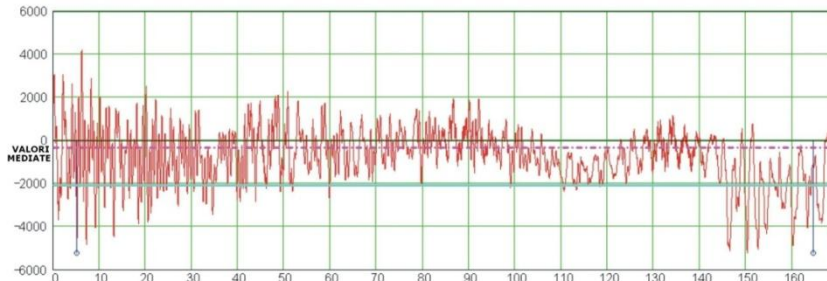


Fig. 4.1-8 Grafic din MatLab care prelucrează datele realizând o mediere a valorilor

În cadrul articolului nu sunt detaliate metodele de achiziție a datelor, cum nu sunt detaliate nici analizele din MatLab la care au fost supuse datele, și pe care l-am prezentat în Fig. 4.1-7 și Fig. 4.1-8. În articol este prezentată metoda de achiziție a datelor, fiind originală și simplă față de tehnicile folosite până acum.

Cercetarea s-a materializat în articolul „Image Processing Method in Low Frequency Modal Analysis of the Large Structures” care a fost prezentat la conferința internațională a dinamicii ingineriei (International Conference of Engineering Dynamics), în aprilie 2007 în Portugalia.

O altă activitate de cercetare la care am luat parte a avut ca scop studierea dinamicii unei macarale amplasate la o înălțime de 115 m pe un coș de termocentrală la Rovinari, jud. Gorj. Termocentrala este un ansamblu format din mai multe clădiri, având înălțimi diferite. Pe unul dintre coșurile termocentralei, la 115m, a fost amplasată o macara care ridică echipamente (ventilatoare, transformatoare) de pe pământ și le lasă la un nivel superior. Macaraua efectuează o cursă verticală de maxim 120m [86].

Scopul cercetării a fost verificarea uzurii echipamentului (brațul macaralei) și dacă echipamentul mai poate ridica sarcina maximă admisă în fișa tehnică. Pentru aceasta grupul a studiat tensiunile din structură în timpul mișcării. Cablurile macaralei sunt elastice. Având în vedere înălțimea la care se lucrează, sub acțiunea sarcinii transportate și a gravitației, cablul devine un pendul. Astfel pot apărea forțe de balans care produc efecte dinamice necontrolate care periclitează integritatea sistemului. În acest caz intervin 3-4 vibrații datorate efectului de pendul și alte 3-4 vibrații datorate efectului elastic de mișcare verticală (tiribombă).

S-au amplasat pe brațul macaralei doi senzori: o punte tensiometrică ca în Fig. 4.1-5 și un accelerometru. Puntea tensiometrică analizează tensiunea din structură, iar accelerometru determină unghiul de înclinare.

După ce au fost instalate sistemele pe macara, aceasta a început să ridice de câteva ori greutatea a căror masă era cunoscută. În timpul ridicării greutăților s-au înregistrat valorile date de cei doi senzori.

96 4 - Implementarea optimă, prin prisma performanță-fiabilitate, a metodelor de comunicare în rețele de achiziția datelor, cu aplicabilitate la excavatoarele cu cupe

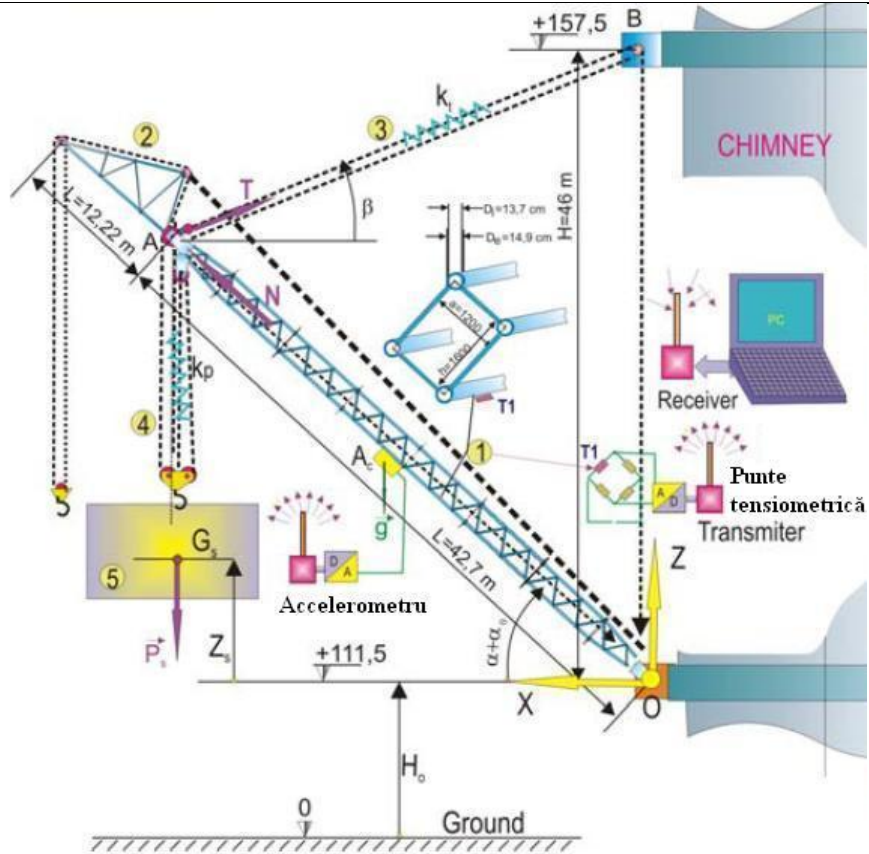


Fig. 4.1-9 Amplasarea punții tensiometrice și a senzorului de accelerație pe macara

Montarea timbrilor pe macara urmăresc vibrațiile care provoacă efecte dinamice necontrolate. Ecuațiile 2 și 3 din [86] descriu modul de deformare al barei.

$$f_{st}(u) = R \cdot \xi(1 - 2\xi^2 + \xi^3)/24$$

Ecuația de deformare a barei

$$\xi = \frac{u}{L} ; R = \frac{pL^4}{El} \cos(\alpha_0)$$

Variabilele implicate în deformare

Ecuațiile descriu fenomenul, iar cu ajutorul datelor achiziționate se verifică limitele normale ale funcționării și apariția fenomenelor nedorite. Efectele dinamice sunt însoțite de vibrații ale structurilor. Studiarea brațului macaralei se realizează în condiții vitrege, asemănătoare cu cele de la excavator. În urma analizelor se obțin

4.1 - Analiza soluțiilor existente prin prisma parametrului vibrații ca element critic al funcționării fiabile a excavatorului 97
 vibrații similare cu cele prezentate în primul articol [85], iar analiza confirmă cercetările și soluțiile prezentate la excavator.

Montarea celor două sisteme de achiziție pe macara a constituit din punctul meu de vedere o rețea radio. Senzorii citesc date la 20ms și le transmit radio la un receptor.

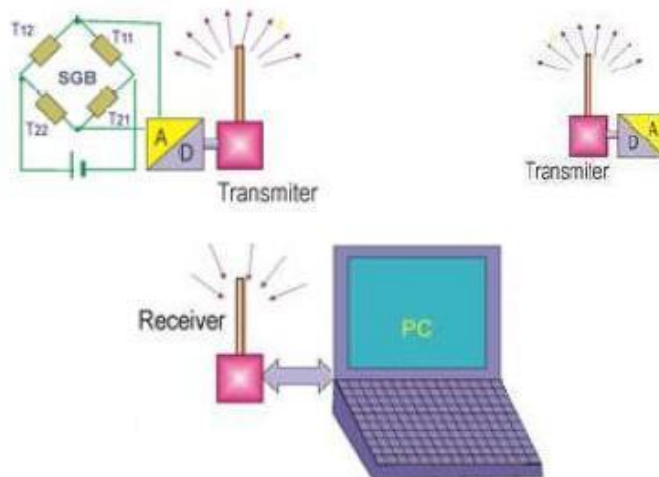


Fig. 4.1-10 Rețea radio formată din 2 transmițători și un receptor

Aici a apărut pentru prima dată necesitatea sincronizării datelor primite. Am folosit un cadru de sincronizare al dispozitivelor. Eșantionarea datelor s-a făcut la 20ms. Datele, având rezoluția de 24 de biți (3 octeți) sunt transmise receptorului, care le trimite computerului unde sunt stocate.

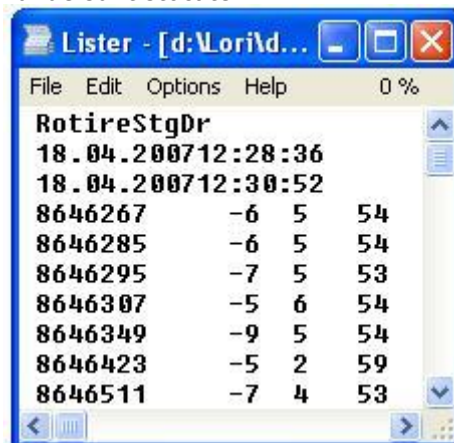


Fig. 4.1-11 Fișier cu date înregistrate de la 2 senzori

Datele din fișier au fost prelucrate cu aplicația MatLab. Contribuția mea la cercetare a fost gestionarea rețelei și transmiterea datelor. Din punctul meu de vedere valorile citite sunt transparente. Trebuia găsită o

98 4 - Implementarea optimală, prin prisma performanță-fiabilitate, a metodelor de comunicare în rețele de achiziția datelor, cu aplicabilitate la excavatoarele cu cupe metodă de sincronizare a citirii datelor folosind aceeași bază de timp. Ulterior, datele citite sunt transmise în rețea și stocate pe un SAD. Având în vedere că în rețea există doi transmițători trebuie evitate coliziunile și alterarea datelor.

Reacția pozitivă avută la cele două lucrări comunicate m-a încurajat în cercetarea mea pe mai departe. Având în acest fel confirmarea nivelului de cercetare și siguranța unei exigențe corespunzătoare, am căutat metode pentru gestionarea unei rețele de dispozitive de achiziția datelor, folosind algoritmi de comunicare și sincronizarea nodurilor. Întreaga rețea are un caracter adaptiv care trebuie redat în descrierea metodelor de comunicare.

Structura de dimensiuni mari optimă cercetării o aveam. Era excavatorul. Grupul de cercetare a furnizat datele fizice care trebuiau monitorizate și scalabilitatea valorilor, adică intervalul de valori în care trebuiau citite valorile și precizia lor.

Realizarea protocolului de comunicație, sincronizarea la milisecundă au avut ca consecință crearea unei configurații originale de sistem de achiziția datelor.

Structura dispozitivului trebuia să fie cât mai generală, dar să se aibă în vedere posibilitatea folosirii unor senzori diferiți. Astfel a apărut DAD-ul. Pentru generalitate s-a creat structura modulară cu mai multe plăci care utilizează în comun resursele.

În timpul cercetării am analizat modulele și circuitele integrate existente pe piață, oprindu-mă asupra modulul radio XBee Series2 sau XBee-PRO ZNet 2.5 OEM. În cadrul primului articol am observat caracteristicile și facilitățile oferite. În al doilea articol am creat o rețea radio, în care erau folosite două sisteme de achiziție care transmiteau date.

M-am decis asupra unei comunicări radio formată din module XBee bazate pe protocolul ZigBee încă din fazele incipiente ale cercetării, dar doream să văd impactul ideii mele asupra lumii științifice.

Metoda și sistemul realizate de mine nu au fost materializate în nici un articol publicat, grupul de cercetare propunând să încununez munca efectuată cu o serie de brevete, dintre care primul este în curs de analiză. Publicarea articolelor ar fi atins clauza de anterioritate, lucru care ar fi stopat certificarea brevetului.

4.2. Soluții potențiale de creștere a fiabilității vibrațiilor

Analizele prezentate în articolele [85] și [86] urmăresc variații singulare ale anumitor parametri. În articolul [85] se determină tensiunea din punctul de sprijin al excavatorului. Din datele citite se extrag variațiile provocate de vibrații și se corelează cu imaginile video captate. Din măsurătorile efectuate în [86] se poate realiza o analiză a vibrațiilor. În cadrul excavatorului sunt cuprinse brațe asemănătoare celui studiat la macara.

Descrierea completă a funcționării excavatorului cu cupe implică monitorizarea mai multor parametri. Calculele și analizele din paragraful 4.1 au deschis noi perspective de studiere și diagnosticare a funcționării excavatorului.

Din acest motiv am propus lărgirea analizei prin folosirea mai multor tipuri de senzori și crearea unei rețele care să realizeze o cronologie a primirii datelor. În acest fel se lărgeste viziunea de tratare a problemei. Utilizând mai mulți senzori se pot realiza interpretări elaborate cum ar fi determinarea direcției unde de vibrație de-a lungul structurii sau determinarea fenomenelor care au dus la provocarea vibrațiilor. Sunt situații în care vibrația nu este provocată de acțiuni mecanice ci de

supraîncălzirea uleiului din lagăre, informație furnizată de un senzor de temperatură.

În acest fel s-au adăugat parametri suplimentari analizei.

În Tab. 4.2-1 este prezentată lista mărimilor fizice stabilite la începutul cercetării :

Mărime fizică	Unitate măsură	Perioada eșantionare	Mărime eșantion	Tip senzor/ Senzorului
Vibrație	G	20 ms	3 octeți	LIS 302
Temperatură	°C	1 pe minut	2 octeți	PT100
Tensiune	V	50 ms	3 octeți	Dispozitiv rezistiv R
Curent	A	50 ms	3 octeți	TRAFO curent
Turație roată	Rot / min	1 pe minut	1 octeți	TIRO
Presiune	bar	20 ms	3 octeți	MLH250BSD1 4B
Tensiunea din trolu	mV / V	10/ sec sau 80/sec	1 octeți	Punte tensiometrică
Limitatori de poziție	On/Off	60 *10 ⁻³ ms	1 bit	Comutator
Patinare bandă	m / s	50 ms	4 octeți	TIRO
Deviere laterală	On/Off	1 pe minut	1 bit	Comutator
Poziție geografică	M	1 pe sec	42 = 19+6+9+4 +4	GPS

Tab. 4.2-1 Mărimile fizice măsurate și senzorii aferenți lor

Pornind de la mărimile fizice prezentate pe prima coloană în Tab. 4.2-1 am găsit tipul de senzori necesari achiziționării datelor, reprezentați pe ultima coloană. Tabelul mai conține unitățile de măsură ale mărimilor fizice, mărimea în octeți a unui eșantion și rata de eșantionare.

Pe cât sunt de diferiți senzorii, pe atât de diferită a fost montarea lor și modul de comunicație cu ei.

Astfel, senzorii de temperatură supraveghează temperatura din lagărele excavatorului. Pe excavator există multe lagăre, ele fiind implicate în toate subansamblele care efectuează rotații. La funcționarea normală temperatura nu trebuie să depășească 55°C-60°C. Depășirea pragului de 60°C poate crea avarii, dacă nu, duce chiar la distrugerea lagărelor. Lagărul are un punct unde este montată sonda de temperatură. Senzorul de temperatură se leagă la microcontrolerul dispozitivului de achiziție printr-o magistrală de comunicație de tipul 1-Wire.

Un excavator port-cupe, în funcție de tip, are între 14 și 20 de cupe. El efectuează între 4-6 rotații pe minut. Cupele sale pot fi tăietoare și/sau încărcătoare. Fiecare cupă la "mușcarea din mal" produce vibrații. Acestea se amplifică la contactul cu roci dure. Pentru înregistrarea vibrațiilor am folosit senzori de tip accelerometru. Accelerometrul comunică cu microcontrolerul dispozitivului de

100 4 - Implementarea optimală, prin prisma performanță-fiabilitate, a metodelor de comunicare în rețele de achiziția datelor, cu aplicabilitate la excavatoarele cu cupe achiziție printr-o magistrală de comunicație de tipul I2C și furnizează accelerația pe cele trei axe.

Curentul motorului de la roata cu cupe nu trebuie să depășească 76A. Monitorizarea lui se face folosind un transformator de curent (TRAFO curent). Una dintre metodele de măsurare a curentului se poate realiza montând clești de măsură pe cablurile de alimentare ale motorului. Transformatorul de curent generează o tensiune direct proporțională cu curentul. Tensiunea este citită cu un Convertor Analog Numeric pe 24 bit comandat de microcontroler.

Pe excavator există mai multe motoare. Cel de la roata cu cupe nu trebuie să aibă o tensiune mai mare de 6kV. Monitorizarea tensiunii se face folosind un dispozitiv rezistiv care face din 6000V un semnal de ordinul voltilor care este măsurat și citit de microcontroler.

Excavatorul, cum este ilustrat în Fig. 1.3-1 are mai multe trolii, necesare operațiunilor de ridicare/coborâre. Dacă sunt forțate, aceste trolii cedează, ducând la ruperea sau dărâmarea excavatorului, cu consecințe ireversibile. Din acest motiv se măsoară tensiunea din troliu folosind o punte tensiometrică. Doza tensiometrică se montează într-un punct din sistemul trolului. Puntea are 4 fire. Două de excitație și două pentru citit valorile. Puntea se leagă la modulul care citește puntea, aflat pe dispozitivul de achiziție. Modulul care citește puntea comunică cu microcontrolerul pe o magistrală SPI. În Fig. 4.1-3 este ilustrat un DAD care citește o punte tensiometrică iar în Fig. 4.1-4 este arătată aplicarea, lipirea timbrelor cu rezistențe pe structură.

Am dat câteva exemple pentru a clarifica gradul de diversitatea al senzorilor care pot fi monitorizați de dispozitivul de achiziție și modurile de comunicație între senzori și placa de achiziție, încercând să sugerez amploarea cercetării și analizelor efectuate.

4.3. Nucleul ARM

Analizând mai multe microcontrolere existente la ora actuală pe piață, dintre care amintesc tipurile: AVR (ATMEL), MSP (Texas Instruments), STM32 CORTEX-M3 (ST); am ales varianta de la NPX (fostă Philips): LPC 210x. Acest microcontroler foarte performant pentru clasa sa, este un ARM7 pe 32 de biți. Are o putere de calcul de 50 MIPS și multe resurse: 2 UART, SPI, 2 I2C, 3 timere, watchdog, PWM, RTC, AD Convertor, DA Convertor. Detaliile tehnice furnizate de microcontroler sunt prezentate în [14], aici fiind punctate doar pentru justificarea alegerii. Un alt motiv important al alegerii lui l-a constituit existența mediilor de dezvoltare prietenoase create pentru această familie de microcontrolere.

Voi realiza o scurtă descriere a plăcii realizate, DAD. Punctul de pornire l-a constituit placa iARM-2103, realizată de iMAGECraft Creations, INC în 2006. Informațiile complete despre această placă se găsesc în iARM-2103 User Manual [13]. Studiind acea placă am adaptat-o cerințelor mele, adăugând pe ea circuitele integrate și modulele de comunicație necesare realizării cercetării mele.

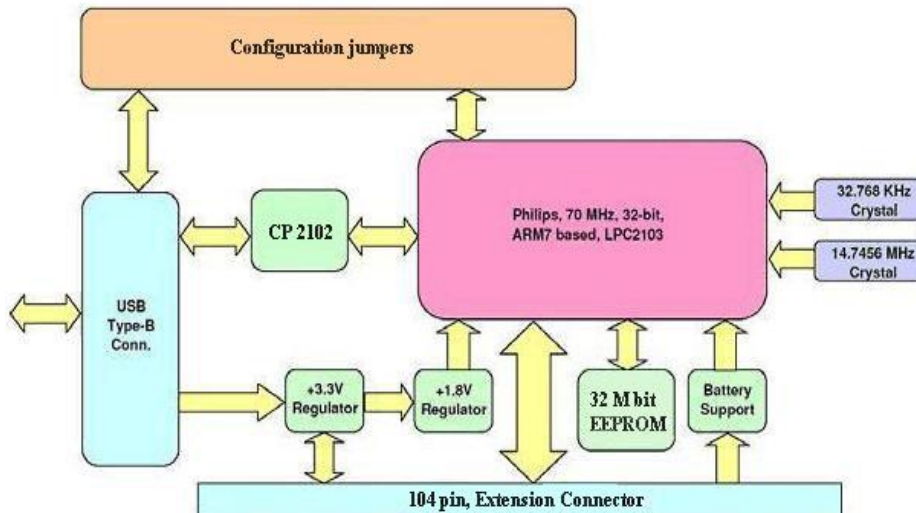


Fig. 4.3-1 Detaliere DAD cu partea de ARM, adaptată din [13]

Nucleul procesorului este de dimensiuni mici, consumă puțin și răspunde în timp real întreruperilor adresate. Se folosesc tehnici de pipeline astfel încât sistemele de procesare și memoria lucrează continuu. Uzual, în timp ce o instrucțiune este executată, succesoarea ei este decodată, iar a treia instrucțiune este adusă din memorie.

Placa rezultată, DAD-ul prezentat în această teză, este o placă electronică de dimensiuni mici, 9 cm x 9,5 cm x 0.2 cm, pe care sunt montate circuite integrate. Placa are un consum redus, de 230 mW și poate să lucreze la temperaturi de la -10° până la 70°C. Schema electrică a plăcii este reprezentată în Fig. 4.3-1.

DAD conține două cristale: de 32.768 necesar RTC și unul de 14.7456 MHz pentru ceasul sistemului care furnizează divizori preciși pentru rata de transfer (baud rate) a UART-urilor [13].

Circuitul oscilator al microcontrolerului suportă cristale externe pe placă. Quartzul folosit în cazul meu este de 14.7456 MHz și permite programarea microcontrolerului până la o frecvență maximă de 58.98 MHz deoarece se folosește PLL. PLL(Phase Locked Loop) este un dispozitiv care multiplică frecvența de ceas a procesorului în aplicații.

Astfel

$$CCLK = M_PLL * Fosc, unde$$

CCLK	Frecvența de ceas a procesorului
M_PLL	Valoarea de multiplicare a PLL
Fosc	Frecvența de ieșire a oscilatorului cristalului de quartz

Atunci $CCLK = 4 * 14.7456 \text{ MHz}$
 $CCLK = 58.9824 \text{ MHz}$

102 4 - Implementarea optimală, prin prisma performanță-fiabilitate, a metodelor de comunicare în rețele de achiziția datelor, cu aplicabilitate la excavatoarele cu cupe

În funcție de valoare multiplicatorului, M_PLL, se poate ajunge la o frecvență de lucru a procesorului de 70MHz.

Conectorul de extensie pe 104 pini, vezi Fig. 3.10-2, este utilizat pentru conectarea mai multor plăci. DAD-ul este proiectat și construit pentru a face parte dintr-o structură modulară. Astfel, conectorul de extensie din Fig. 3.10-1 poate fi folosit pentru a conecta mai multe plăci, cum este ilustrat în Fig. 3.10-2.

4.3.1. Watchdog timer

Rolul watchdog timer-ului este de a reinițializa (reset) microcontroler-ul la anumite perioade de timp, dacă acesta intră într-o buclă infinită. Există multe cazuri care pot provoca buclarea infinită, eu amintesc doar căderea pentru foarte puțin timp a alimentării (glitch) sau apariția unui zgomot pe linia de comunicație. Aceste fenomene nu resetează microcontroler-ul, dar pot să altereze valorile regiștrilor lui interni, ceea ce duce la anomalii în funcționare. În aceste cazuri intervine watchdog-ul, care reinițializează microcontroler-ul.

WDT ul este alcătuit dintr-un Timer pe 32 de biți și un divizor de frecvență cu valoare fixă. Tactul PCLK ajunge la timer după ce este divizat cu 4. Timerul se decrementează la fiecare clock primit. Valoarea minimă de la care se poate decrementa timerul este 0xFF. Încărcarea unei valori mai mici de 0xFF determină inițializarea timerului tot cu valoarea 0xFF.

Valoarea minimă a watchdog-ului timer-ului este (TPCLK x 256 x 4), iar valoarea maximă este (TPCLK x 2³² x 4), totul în multipli de (TPCLK x 4).

$$TPCLK = CCLK \times PLL = 14,745\text{MHz} \times 4 = 58,98 \text{ MHz}$$

la...1 seccorespund ... TPCLK ticuri ale clock-ului intern al microprocesorului

$$1/ TPCLK = 0,0169 \mu\text{s} \text{ corespunzătoare unui tic al clock-ului intern}$$

Considerând și pre-scaler-ul divizibil cu 4, pot concluziona că la fiecare

4 X TPCLK = 4 X 0,067 μs = 0,0678 μs se decrementează valoarea stocată în watchdog timer.

Valoarea minimă a watchdog-ului este = TPCLK x 4 x 256 = 256 X 0,0678 μs = 17,36 μs

$$\text{Valoarea minimă} = 17,36 \mu\text{s}$$

Valoarea maximă a watchdog-ului este = TPCLK x 4 x 2³² = 0,0678 X 4294967296 μs = 219198782,668 μs = 219,198782668s = 4,85 min

$$\text{Valoarea maximă} = 4,85 \text{ min}$$

Cunoscând valorile minime și maxime care se pot încarca în watchdog timer, am căutat valorile necesare aplicației mele.

O aplicație încapsulată (embedded) rulează într-o buclă infinită în cadrul căreia sunt analizate evenimentele care apar. După modelul sistemelor de operare în timp real, folosind întreruperi de ceas (timer), la intervale regulate de timp (5ms, 10ms, 20ms, etc) am apelat funcții. În cadrul acestor funcții se realizează operațiile de achiziția datelor și transmiterea pachetelor. Achiziția datelor de la senzori se face

folosind rate de eșantionare diferite. Valorile stabilite pentru watchdog timer trebuie să fie mai mari decât timpul în care se realizează operațiile. Astfel pentru vibrații, a căror eșantionare se face la fiecare 0.5ms am setat valoare watchdog timer-ului la 1,5ms. Pentru senzori, a căror date care sunt eșantionate la 20 ms valoarea watchdog timer-ului am stabilit-o la 60ms, iar pentru senzori cum ar fi temperatura, la care eșantionarea se face la secundă, am încărcat în registrul WDTV valoarea corespunzătoare unui minut.

WDT folosește în funcționare 4 regiștri:

WDMOD – WD mode register, furnizează modul de lucru al WD și starea lui. Acest registru poate fi scris și citit (R/W).

WDTC – WD Timer constant register. Este registrul în care se încarcă valoarea decrementată, de time-out. Acest registru poate fi scris și citit (R/W).

WDFEED – Wd Feed sequence register. În acest registru se scrie valoarea 0xAA, urmată de valoarea 0x55, care pornește sau reîncarcă WD timer-ul cu valoarea inițială. Este de preferat ca WD să se încarce înainte de generarea unui reset. Acest registru poate fi doar scris(WO).

WDTV – Wd Timer Value register. Acest registru ilustrează valoarea curentă a WD timer-ului. Acest registru poate fi doar citit (RO).

În funcționare, valoarea timpului se încarcă în registrul WDTC, iar watchdog-ul se pornește prin scrierea valorii 0xAA, urmată de 0x55 în registrul WDFEED. Reîncărcarea se face prin scrierea succesiunii de valori 0xAA, urmată de 0x55, în registrul WDFEED.

Dacă valorile succesive 0xAA și 0x55 nu sunt scrise și numărătorea inversă ajunge la 0 (underflow), în pointerul de instrucțiuni (Program Counter) se încarcă adresa 0x 0000 0000, exact ca și în situația unei inițializări externe (external reset).

În afară de situațiile din aplicație în care pot apărea anomalii care determină intervenția watchdog-ului, pot apărea resetări la încărcarea valorilor în registrul WDFEED. Astfel, dacă după încărcarea valorii 0xAA în registrul WDFEED intervine o întrerupere, valoarea 0x55 care trebuie încărcată la următorul tact al ceasului sistemului, nu se mai încarcă și are loc o reinițializare.

Am căutat soluții pentru rezolvarea acestei situații.

Prima soluție constă în dezactivarea întreruperilor înainte de scrierea în registrul WDFEED, urmată imediat de activarea lor.

```
VICIntEnClr=0x... // Dezactivarea întreruperilor în VIC
```

```
WDFEED=0xAA; // Reîncărcare watchdog
```

```
WDFEED=0x55;
```

```
VICIntEnable=0x.. // Activarea întreruperilor în VIC
```

Dar folosind această soluție pot apărea întreruperi false (spurious).

Voi explica ce se întâmplă în aplicația care rulează în timp real.

Pas1 Controler-ul vectorilor de întrerupere (VIC) detectează o cerere de întrerupere (IRQ request) și trimite semnalul nucleului microcontrolerului.

Pas2 Nucleul microprocesorului blochează starea întreruperii.

Pas3 Procesarea continuă pentru încă câteva cicluri de ceas, având încă resurse în pipeline.

Pas4 Se lansează în execuție rutina de tratare a întreruperii.

104 4 - Implementarea optimală, prin prisma performanță-fiabilitate, a metodelor de comunicare în rețele de achiziția datelor, cu aplicabilitate la excavatoarele cu cupe

Dacă la pasul 3 starea VIC se modifică, situație provocată de dezactivarea întreruperilor, cum am prezentat în exemplu, atunci pot apărea în anumite cazuri, nu neapărat de fiecare dată, întreruperi false.

Dacă starea VIC se modifică înainte de încărcarea adresei rutinei de întrerupere din VIC, atunci PC-ul va furniza o adresă predefinită din registrul implicit al adreselor VIC (VIC Default Vector Address Register). Dar la reset, acest registru este inițializat cu 0 și sistemul se va comporta exact ca și în cazul unui reset.

În această situație, m-am gândit la o a doua soluție. Implementarea procesului de feed (reîncărcare watchdog timer) prin scrierea valorilor succesive 0xAA, urmată de 0x55 am realizat-o în cadrul unei rutine de tratare a întreruperii. Microcontroler-ul ales în aplicație nu permite ca în timpul tratării unei întreruperi să intervină o altă întrerupere și m-am folosit de acest fapt.

În cazul apariției unui reset DAD-ul va trebui reîncărcat cu datele de configurare. La pornirea microcontroler-ului, registrul de reset (RSIR-Reset Source Identification Register) indică dacă reinițializarea a fost produsă de watchdog. Pentru reîncărcarea valorilor de configurare ale microcontroler-ului am folosit un sector al memoriei SPI, sector care este alimentat extern de o baterie. După reinițializare, se reîncarcă din flash datele de configurare obținute la pornirea în condiții normale. După reinițializare, DAD-ul trebuie resincronizat cu centrul de sincronizare, care conține un sistem de poziționare globală (GPS). Așa cum am arătat în algoritmiile prezentați, nu este necesară resincronizarea întregii rețele, ci doar a nodului care s-a reinițializat. În acest fel am oferit o soluție fiabilă și performantă.

4.4. Programarea microcontrolerului

Fiecare DAD este echipat cu un microcontroler LPC2103 70 Mhz ARM al firmei NPX (fondat de Philips). Informațiile complete despre microcontrolerul LPC2103 se găsesc în manualul de utilizare [14].

LPC2103 este un microcontroler RISC (Reduced Instruction Set Computing) de mare performanță și consum mic, pe 32 de biți având la bază un ARM7TDMI-S [15].

Alimentarea lui se poate face prin USB sau direct pe placă. Microcontrolerul rulează pe placă la o frecvență de 60MHz ceea ce furnizează o putere de calcul de peste 50MIPS (Milion of Instructions Per Second).

Microprocesorul dispune de 32 KB Flash și 8 KB SRAM [13] [14].

Are 48 de pini dintre care 32 sunt GPIO (General Purpose Input/Output) programabili.

Microcontrolerul LPC2103 se folosește într-o gamă largă de aplicații, dintre care amintesc [14] aplicații de uz general, sisteme medicale, control industrial, control de acces, puncte de vânzare, porți (gateway) de comunicare și programe încapsulate (embedded software).

Un avantaj îl constituie perifericele și caracteristicile oferite:

Convertor AD cu 8 canale, pe 10 biți

2 timer/counter/PWM (Pulse Width Modulation) pe 32 biți

- 2 timer/counter/PWM pe 16 biți
- Watchdog timer
- RTC (Real Time Clock)
- Interfețe seriale: 2 magistrale I2C, 2 UART, SPI, SSP
- Permite lucrul cu întreruperi
- Are moduri de power down și wake-up
- Pini GPIO toleranți la 5V

Fiecare periferic poate fi activat sau dezactivat individual, dar este posibilă și funcționarea simultană a tuturor. Comunicarea cu perifericele se poate face serial cât și prin intermediul întreruperilor.

Microcontrolerul LPC2103 dispune de un port, numit PORT0, care conține 32 de pini, acestea fiind linii rapide de comunicație. Toți cei 32 de pini pot fi reconfigurați, folosindu-se atât ca porturi de uz general, cât și ca linii dedicate. De exemplu un anumit pin al portului, P0.11, poate fi intrare sau ieșire GPIO, linie de transmisie pentru UART și intrare de captură pentru timer.

Portul GPIO(General Purpose Input/Output) este controlat cu ajutorul unor regiștrii pe 32 de biți: IOPIN, IOSET, IODIR, IOCLR.

Valoarea portului poate fi modificată folosind o singură instrucțiune. Există regiștri care setează (IOSET) și resetează (IOCLR) porturile la nivel de bit. Într-o singură instrucțiune pot fi setați sau resetați oricâți biți ai unui port.

Folosirea regiștrilor standard realizează compatibilitatea cu versiunile mai vechi de microcontroler[14]. Acești regiștri poartă denumirea de GPIO „încet” sau GPIO standard și sunt adresabili la nivel de cuvânt (32 biți).

GPIO a fost îmbunătățit. Mai există un set de regiștri: FIODIR, FIOMASK, FIOPIN, FIOSET și FIOCLR care formează așa zisul GPIO “rapid”.

Cu ajutorul regiștilor mască (FIOMASK) se trece de la modul de utilizare standard la modul rapid și se pot accesa și modifica anumite grupuri de biți ai porturilor, lăsându-i pe ceilalți nemodificați. Toți acești regiștrii rapizi sunt adresabili la nivel de octet (8 biți) și jumătate de cuvânt (16 biți), permițând acces mai ușor și mai rapid la pini portului fizic. Folosind GPIO rapid accesul la pini fizici este de 3,5 ori mai rapid decât în cazul GPIO-ului standard [14].

Deoarece regiștrii de GPIO rapid și standard controlează aceiași pini fizici, lucrul cu ei este mutual exclusiv și independent. Modificările care au loc în setul de GPIO rapid nu vor fi observate în regiștrii GPIO standard.

În aplicația mea am folosit portul GPIO standard pentru inițializarea sistemului și a perifericelor cu care am lucrat. GPIO-ul rapid l-am folosit la programarea dispozitivelor LCD și la comunicarea pe UART1.

Toți pini GPIO sunt toleranți la 5V. Acesta înseamnă că la depășirea valorii tensiunii la peste 3,3V pini sunt protejați și nu se strică, decât la tensiuni de peste 5V.

Abordarea și programarea embedeed este diferită de tehnicile de programare uzuale. În primul rând se ține seama de caracteristicile tehnice ale circuitelor integrate și sunt implicate configurații ale pinilor sau ale dispozitivelor de lucru. Voi exemplifica aceasta printr-o funcție de configurare a UART, care stabilește valorile comunicării prezentate în 2.7.3.

```
// funcția de configurare UART0 și UART1
```

106 4 - Implementarea optimală, prin prisma performanță-fiabilitate, a metodelor de comunicare în rețele de achiziția datelor, cu aplicabilitate la excavatoarele cu cupe

```
int Serial_Open(int uart,
                int baud, int databits,
                int parity, int stopbits)
{
    dbits = databits;
    par = parity;
    stop_bits = stop_bits;
    if(uart == COM_UART0)
    {
        init_UART0(baud);
        return TRUE;
    }
    else if(uart == COM_UART1)
    {
        init_UART1(baud);
        return TRUE;
    }
    return FALSE;
}

// funcția de inițializare UART0
void init_UART0(int baud)
{
    unsigned int divisor = GetPclk() / (16 * baud);
    int dummy=0;

    tx0buf.in = 0;    //resetare buffer transmisie
    tx0buf.out = 0;
    tx0_restart = 1;

    rx0buf.in = 0;    //resetare buffer recepție
    rx0buf.out = 0;

    PINSEL0_bit.P0_0 = 0x1;
    PINSEL0_bit.P0_1 = 0x1;

    //activare întreruperi
    VICVectAddr1 = (unsigned long)uart0_irq;
    VICVectCntl1_bit.NUMBER = VIC_UART0;
    VICVectCntl1_bit.ENABLED = 1;
    VICIntEnable_bit.UART0 = 1;

    U0LCR = 0x03;      // 8 bits, no Parity, 1 Stop bit
    U0IER = 0;         // Disable UART1 Interrupts
    U0LCR |= 0x80;     // Set DLAB
    U0DLL = divisor & 0xFF;
    U0DLM = (divisor >> 8) & 0xFF;
    U0LCR &= ~0x80;   // Clear DLAB
    dummy = U0IIR;    //Read IrqID-Required to Get Interrupts Started
    U0IER = 3;
}
```

Programul principal rulează continuu într-o buclă, iar acțiunile sunt realizate folosind anumite task-uri care sunt apelate la intervale constante de timp, 5ms, 10ms, ori prin intermediul întreruperilor.

```

void main (void)
{
    initializare_sistem();
    initSysTime();

    // initializez UART1
    Serial_Open(COM_UART1, 115200, 8, PARITY_NONE, STOPBITS_ONE );

    //initializez ceasul sistemului
    RTC_InitHardware();

    ...

    tid_uart_read = Task_AllocateTask(ReadUart0Task, 1,FALSE);
    Task_Start(tid_uart_read);

    i2cInit();           //initializare I2C
    init_senzor_Vibratii();
    tid_senzor_read = Task_AllocateTask(
        Sensor_ReadVibrationsTasks,
        SAMPLING_RATE, TRUE);
    Task_Start(tid_senzor_read);

    tid_uart_write = Task_AllocateTask(
        SendPacketsTask,
        SAMPLES_TO_SEND*SAMPLING_RATE, FALSE);
    Task_Start(tid_uart_write);

    ...

    while(1)
    {
        for(i=0;i<Task_GetTasksNumber();i++)
        {
            if(Task_IsDue(i))
            {
                Task_Run(i);
            }
        }
        Task_Run(tid_uart_read);
    }
}

```

Funcția **Serial_Open** inițializează UART0 sau UART1. Parametrii de intrare sunt `uart`, care face diferența dintre cele două dispozitive, rata de transmisie, baud rate, numărul de biți folosiți pentru codificarea unui caracter, 8 în cazul nostru, numărul de biți de paritate folosiți, 0 în cazul nostru și numărul de biți de stop folosiți, 1 în cazul nostru. În cadrul funcției sunt apelate funcții de inițializare specifice fiecărui dispozitiv.

Am detaliat funcția de inițializare pentru UART0, care resetează buffer-ele de transmisie (tx) și recepție (rx); pregătește UART0 să recepționeze date, validează și permite lucrul cu întreruperi.

Apelarea funcției **Serial_Open** se face în programul principal. În primul rând se inițializează sistemul, apoi UART0, se validează folosirea ceasului în timp real (RTC) al sistemului, se realizează configurarea unui senzor de vibrații. Se stabilește rata de eșantionare.

108 4 - Implementarea optimală, prin prisma performanță-fiabilitate, a metodelor de comunicare în rețele de achiziția datelor, cu aplicabilitate la excavatoarele cu cupe

Urmează apelarea diverselor taskuri dintre care amintesc citirea portului serial, citirea datelor de la senzor și transmiterea datelor în rețea.

Urmează bucla principală care tratează toate task-urile existente.

4.5. Programarea nodurilor DAD

Pentru implementarea cercetării am folosit mai multe aplicații. Proiectul pentru microcontrolere l-am realizat folosind aplicația Keil μ Vision 3V3.12a. În urma compilării proiectului a rezultat un fișier binar, cu extensia hex.

Aplicația Keil μ Vision reprezintă un mediu integrat care permite crearea de proiecte cu particularitățile microprocesorului ales prin generarea fișierului de configurare inițială, startup.s. De asemenea permite editarea aplicației cu indentări și evidențierea cuvintelor cheie, fișiere de ajutor (help) și generarea codului sursă cu posibilitatea efectuării de debug.

Compilerul C atașat mediului Keil permite toate tipurile de ANSI C, iar programele l-am scris utilizând limbajul C.

În cadrul sesiunii de depanare (debug) pot fi folosite mai multe puncte de întrerupere (breakpoint). Sunt accesibile ferestre cu harta memoriei, a stivei utilizate sau a perifericelor.

În Fig. 4.5-1 este reprezentată o captură de ecran din timpul depanării (debug), și este prezentat meniul cu perifericele care pot fi interogate sau configurate:

Blocul de control al sistemului cu acceleratorul de memorie (MAM), PLL(Phase Locked Loop), divizorul VPB, sau întreruperile externe

Controlerul vectorilor de întrerupere pentru Watchdog, Timer1, Timer0, UART0, UART1, PWC, I2C, SPI, PLL Lock, RTC, External interrupt 0,1,2

blocul de conectare al celor 32 de pini de uz general, P0.0-P0.31;

Registrii de intrare/ieșire GPIO

UART-urile

Interfața I2C

Interfața SPI

Timere

PWM

Ceasul de timp real RTC

Watchdog

Operațiunea de depanare (debug) permite simularea programului scris în microcontroler.

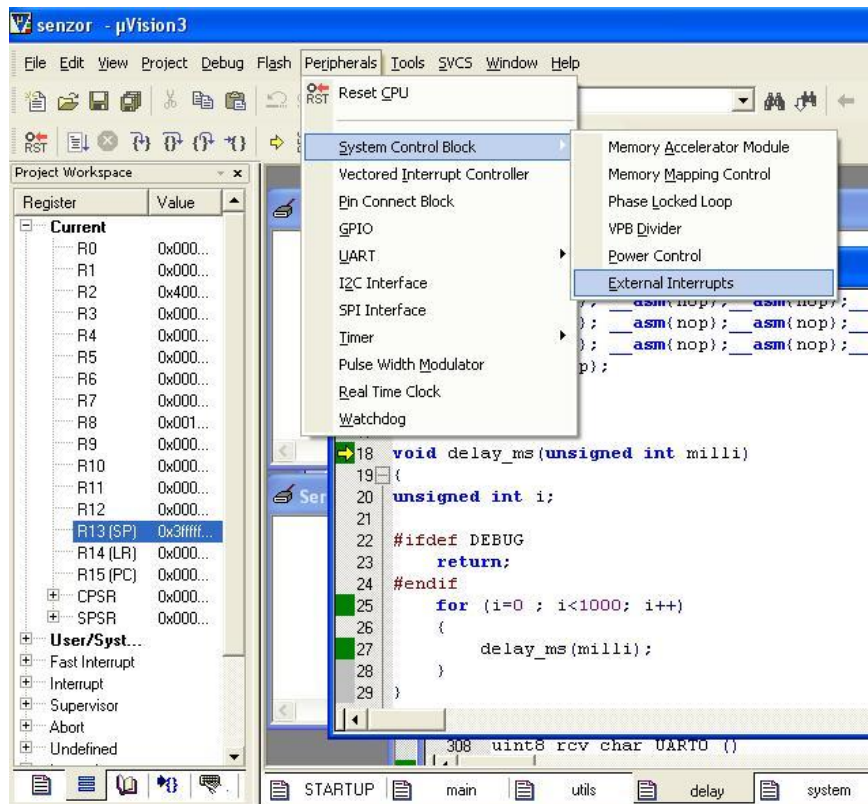


Fig. 4.5-1 Captură de ecran din timpul depanării în Keil

Încărcarea fișierului sursă hexa în microcontroler s-a realizat cu aplicația Flash Magic versiunea 3.59.227.

Verificarea configurației DAD-urilor și testarea porturilor de comunicare s-a realizat cu aplicația X-CTU.

Programul de analiză al protocolului, benchmark-ul și materializarea valorilor în formă grafică l-am realizat folosind aplicația Microsoft Visual Studio 2005, cu limbajul de programare C#. Sistemul de operare folosit a fost Windows XP.

Graficele analizelor au fost realizate în Microsoft Excel.

În Fig. 4.5-2 este prezentată o captură a aplicației Flash Magic care încarcă un fișier binar, numit „DAD12.hex”. Cu ajutorul acestei aplicații se configurează tipul procesorului, portul prin care se încarcă fișierul .hex în microcontroler-ul DAD-ului, frecvența de lucru a oscilatorului și baud rate la care se comunică cu microprocesorul.

110 4 - Implementarea optimală, prin prisma performanță-fiabilitate, a metodelor de comunicare în rețele de achiziția datelor, cu aplicabilitate la excavatoarele cu cupe

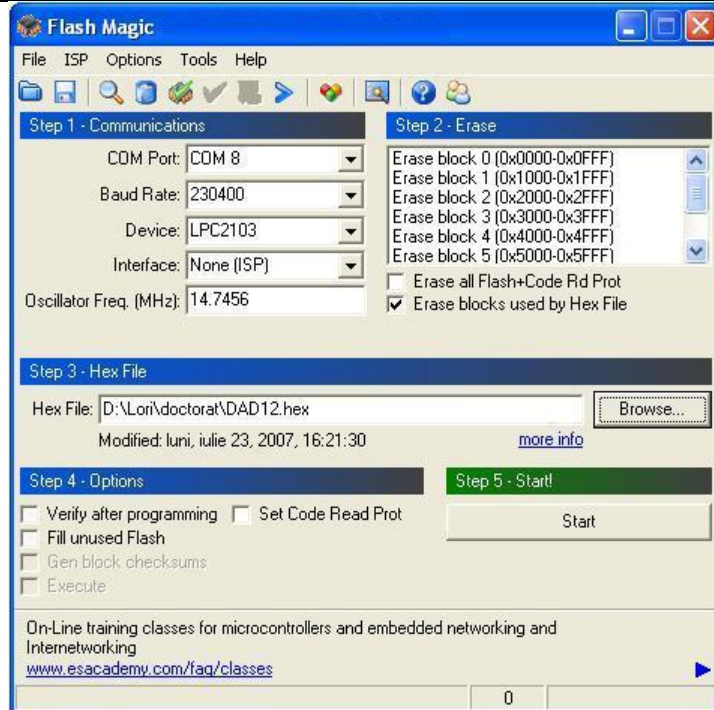


Fig. 4.5-2 Captura cu flash magic folosit la încărcarea fișierului .hex în microcontroler

În ajutorul proiectului meu am găsit pe Internet programe simple care arată și demonstrează caracteristicile nucleului iARM, cum ar fi: configurarea microcontrolerul, comunicația pe serială UART, configurarea timer-elor, ceasul în timp real (RTC), convertorul AD sau lucrul cu întreruperi. Pornind de la ele am construit proiectele aplicației mele, tratând fiecare tip de senzori și comunicația de la fiecare DAD în parte. Acolo unde nu au existat aplicații le-am construit bazându-mă pe datele și configurațiile din manualele tehnice ale dispozitivelor.

O altă direcție a constituit-o configurarea și realizarea programelor de comunicație radio, bazate pe modulul XBee.

Pe fiecare placă a DAD-ului se găsesc jumperi care sunt folosiți la programarea plăcii, pentru configurarea modulelor de comunicație, la transmisie sau recepție.

Astfel se face legătura între calculatorul personal și microcontroler pentru programarea microcontrolerului și încărcarea aplicației realizate în Keil sau între calculatorul personal și modulul de comunicație radio pentru a-l programa.

Modul de funcționare normal al nodurilor radio folosește comunicarea microcontroler-XBee la transmiterea cadrelor și XBee-microcontroler la recepție. Coordonatorul C realizează uzual comunicarea la nivel Xbee-calculator personal.

Au existat mai multe tipuri de comunicație:

Microcontroler - XBee – utilizată în mod normal pentru comunicarea radio a tuturor nodurilor din rețea, în afara coordonatorului C.

Calculator personal - XBee folosit pentru programarea XBee-ului prin comenzi AT sau folosind aplicația X-CTU și definirea baud-rate-ului, și a altor parametri de configurare. Comunicarea calculator personal-XBee este folosită în funcționare de coordonatorul C al rețelei radio.

Calculator personal - microcontroler – comunicare utilizată pentru programarea router-elor și nodurilor finale radio și pentru depanarea programelor.

În Fig. 4.5-3 și Fig. 4.5-4 este detaliată comunicarea serială.

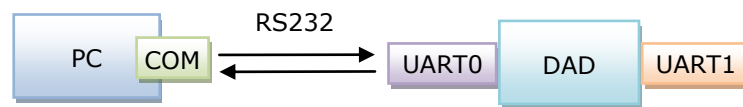


Fig. 4.5-3 Configurarea comunicării seriale

Pentru programarea microcontroler-ului se folosește comunicarea calculator personal-UART0. Tot această metodă este folosită și pentru depanare. După ce sunt programate individual toate DAD-urile care fac parte din rețeaua serială, are loc conectarea, ilustrată în fig. 4.3-5.

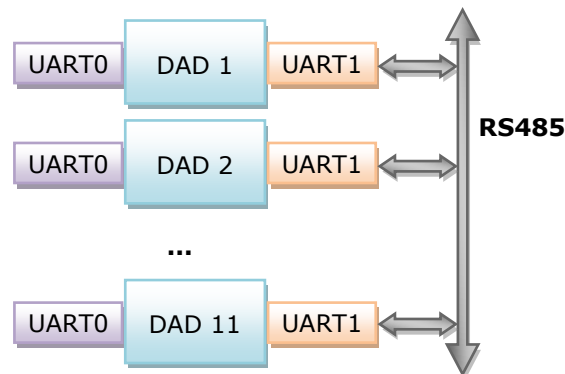


Fig. 4.5-4 Comunicarea serială

Inițial calculatorul personal comunică prin intermediul unui COM cu UART0 al microcontroler-ului folosind comunicarea RS232, așa cum este ilustrat în Fig. 4.5-3, iar după programare, dispozitivele comunică serial între ele folosind protocolul RS485 prin intermediul UART1 al microprocesorului, Fig. 4.5-4.

4.6. Programarea RDEA în cazul comunicării radio

Montarea DAD-urilor s-a realizat foarte aproape de senzorii existenți pe utilaj pentru ca semnalul analogic captat să fie supus unei perturbații minime. În acest caz varianta optimă este folosirea unui modul radio. O soluție clasică, cu derularea

112 4 - Implementarea optimală, prin prisma performanță-fiabilitate, a metodelor de comunicare în rețele de achiziția datelor, cu aplicabilitate la excavatoarele cu cupe de cablu ar fi fost grea, dacă nu chiar imposibilă, deoarece utilajele monitorizate au părți care realizează mișcări relative.

Aplicațiile XBee realizate până în prezent propun o comunicare low-rate între dispozitive care lucrează o mare parte din timp în regim economic (low-power).

În cercetarea mea studiez comportamentul rețelei în cazul unui volum mare de date. Achiziția datelor senzorilor se face la milisecundă. RDEA realizează sincronizarea dispozitivelor tot la milisecundă.

Teza de față implementează cerințele corespunzătoare nivelului aplicație, ținând cont de specificul echipamentelor și urmărind sincronizarea lor pentru a funcționa ca un sistem eterogen. Realizarea unor structuri stufoase la nivel aplicație poate afecta viteza. Pe toată perioada cercetării am urmărit constrângerea de minimizare și conservare a resurselor de calcul și memorie ale nodurilor.

Dacă până nu demult rețelele radio erau recomandate pentru aplicații industriale care nu necesitau o rată mare de transfer și nici o calitate a serviciului superioară, noile generații de module radio și protocoale precum ZigBee egalează și chiar întrec performanțele rețelelor clasice cu fir (wired) furnizând aceiași parametri de fiabilitate la o performanță superioară.

Anul 2006 aduce primele module direct conectabile în rețea, fabricate de diverși producători. Dintre acestea am ales, cum am explicat deja, modulul radio XBee Series2 sau XBee-PRO ZNet 2.5 OEM, care folosesc protocolul ZigBee bazat pe standardul IEEE802.15.4.

Alegerea am făcut-o având în vedere că dispozitivele sunt dedicate monitorizării și controlului, timpul de viață al bateriei este lung în comparație cu alte dispozitive. Rata de transfer, deși este mai mică decât a unor sisteme prezentate în Tab. 4.6-1, per ansamblu, pentru aplicația mea este acceptabilă și suficientă.

În tabelul 4.6.1-1 sunt ilustrate câteva aspecte care m-au ajutat în luarea deciziei. În plus față de ele prețul modului radio și dimensiunile sale reduse au fost atractive.

	802.15.4 ZigBee	GSM/GPRS	802.11 Wi-Fi	802.11.1 Bluetooth
Aplicații	Monitorizare și control	Date și voce pe arii întinse	Internet de mare viteză	Conectare echipamente
Timp viață baterie (zile)	100-1000	1-7	1-7	1-7
Lățime bandă	250kbps	>128k	11Mbps	720kbps
Raza de acțiune (m)	100+	km	50-100	10-100
Dimensiune rețea	65000	-	32	7

Tab. 4.6-1 Comparație între dispozitive și protocoale actuale [54,60]

Avantajele oferite de ZigBee la timpul de viață al bateriei, numărul de noduri din rețea și raza de acțiune sunt net superioare protocoalelor Wi-Fi și Bluetooth.

Modulele XBee-Series 2 și 3 au o rază de acțiune de la 300m până la 1200m în câmp deschis și până la 100m în clădiri, în oraș. Ele consumă 60mW [61]. Ele utilizează banda de frecvență de 2,4 GHz și folosesc 16 canale DSSS(Direct Sequence Spread Spectrum) [46].

DSSS are o capacitate mai mare față de alte tehnologii utilizate în rețelele radio de dimensiuni mari (WLAN) cum ar fi FHSS(Frequency Hopping Spread Spectrum). Lățimea de bandă a mesajului transmis este mai mare decât lățimea mesajului original, deoarece la transmitere se adaugă încă un semnal (Spreading Code). [47,48,49]

Cu ajutorul modulelor XBee se crează rețele flexibile ad-hoc cu legături între noduri de tipul punct-la-punct, peer-to-peer, punct-la-multi-punct sau multidrop și permit utilizarea a peste 65,000 adrese de rețea.

Dimensiunea stivei și rata de transmisie a datelor sunt mai scăzute în cazul protocolului ZigBee dată fiind natura aplicațiilor de monitorizare și control pe care le prestează, în comparație cu protocolul Wi-Fi care furnizează servicii video, web sau mesagerie.

4.6.1. Cadrele API

Pentru comunicarea în rețea am ales modul de operare API, care permite configurarea și interogarea nodurilor. Mai mult, sunt transmise coduri de eroare în cazul unor anomalii.

Datele, comenzile și răspunsurile sunt asamblate în cadre API.

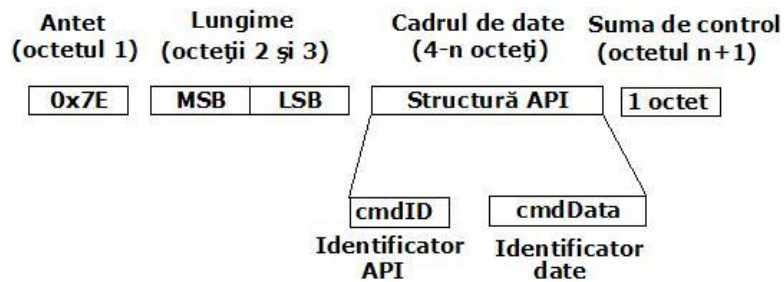


Fig. 4.6-1 Structura unui cadru API

Structura generală a unui cadru API este ilustrată în Fig. 4.6-1 și conține:

- caracter de start, pe 1 octet, cu valoare fixă 0x7E
- lungimea comenzii API din cadrul pachetului. Este reprezentat pe 2 octeți.
- comanda API
- un octet pentru suma de control (checksum)

Partea de recepție în mod API ignoră orice caracter primit care este diferit de 0x7E. La apariția acestui caracter, consideră că a primit un indicator de început de cadru API și transferă datele în buffer-ul de recepție. Dacă cadrul nu este recepționat corect sau dacă suma de control este alterată, modulul radio XBee va răspunde cu un cadru de eroare, indicând și felul acesteia.

Comanda API este formată, vezi Fig. 4.6-1 din:

- identificatorul API, sau identificatorul comenzii API
- comanda API

114 4 - Implementarea optimală, prin prisma performanță-fiabilitate, a metodelor de comunicare în rețele de achiziția datelor, cu aplicabilitate la excavatoarele cu cupe

Identificatorul reprezintă tipul mesajului API care va fi încorporat în cadru. În implementare am folosit șase tipuri de comenzi API. Acestea sunt enumerate în Tab. 4.6-2.

Identificatorul comenzii API	Valoare
Comanda AT	0x08
Comanda AT cu parametri	0x09
Răspuns la comanda AT	0x88
Pachet de transmisie date	0x10
Starea transmisie	0x8B
Pachet primire date	0x90

Tab. 4.6-2 Comenzile API utilizate

Comanda AT cu parametri sau fără este folosită pentru configurarea modulului radio XBee: baud rate, viteză, adresă, nivel de energie.

Răspunsul comenzii AT poate furniza versiunea softului scris în modul sau informații despre nodul respectiv sau despre nodurile din rețea (Node Discover).

Există aplicația X-CTU pe care am utilizat-o pentru configurarea modulului radio XBee. X-CTU scrie firmware-ul și o parte din valorile setate de mine în microcontroler-ul modemului XBee.

În Fig. 4.6-2 este reprezentată captura de ecran cu utilizarea aplicației X-CTU.

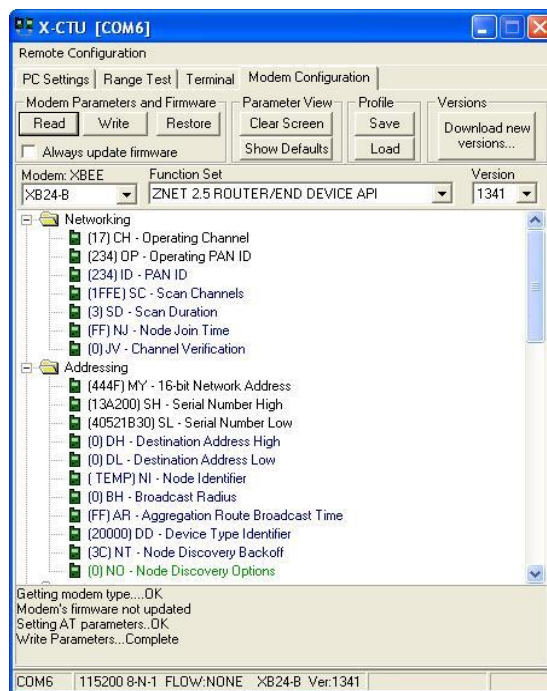


Fig. 4.6-2 Configurarea unui XBee folosind aplicația X_CTU

Avantajul X-CTU este că oferă o interfață prietenoasă de lucru. Dar, nu toți parametri folosiți în proiect au putut fi setați cu această aplicație, deoarece X-CTU

configurează doar o parte comună, uzuală a parametrilor. Pentru restul parametrilor am folosit comenzile AT parametrizate sau nu.

Pachetul de transmisie date este folosit pentru transmiterea pachetelor de date în rețea.

Starea transmisiei, reprezintă pachetul de confirmare primit de la nodul receptor, în care se indică dacă pachetul de date a ajuns, dacă a ajuns cu succes, sau este alterat, corupt și necesită retransmiterea.

Pachetul primire date, este pachetul ajuns la receptor care conține datele trimise de nodul sursă, emitent.

Formatul câtorva comezi API folosite în proiect este detaliat în anexa1 .

4.6.2. Configurare RDEA

DAD-urile sunt introduse offline în rețea. Înainte de introducere sunt programate.

În cazul comunicației radio coordonatorul C, router-ele și nodurile finale sunt programate firmware, cu ajutorul X_CTU, fiecare pentru rolul lor.

Se face legătura între calculatorul personal și microcontroler pentru programarea microcontrolerului și încărcarea aplicației realizate în Keil sau între calculatorul personal și modulul de comunicație radio pentru a-l programa.

Pentru comunicarea serial cablată DAD-urile sunt programate cu algoritmul simplificat al modulului de transmisie radio.

După legarea tuturor modulelor urmează rularea algoritmului de sincronizare, pentru fiecare rețea creată.

Într-o rețea radio defectarea unui nod sau părăsirea rețelei se desfășoară fără afectarea funcționării. La fel, la reintrarea nodului în rețea, aceasta se reconfigurează păstrându-și funcționalitatea. Protocolul ZigBee realizează comunicația în rețea, furnizează confirmări, rezolvă situațiile de coliziune.

Resursele unui nod sunt limitate. Din acest motiv se propun algoritmi de alocare dinamică a memoriei și se propun tehnici pentru colectarea reziduurilor (garbage collection), care furnizează capacitate suplimentară de calcul [69].

4.7. Baza de date

În cadrul aplicațiilor prezentate în articolele [85] și [86] pentru stocarea datelor s-au folosit fișiere, cu header-e, în care se salvau datele pentru fiecare senzor în parte, așa cum am arătat în Fig. 4.1-5 și Fig. 4.1-11.

Dar în proiectul realizat în această teză numărul senzorilor ajunge până la 386, ceea ce necesită o centralizare a datelor. Pentru aceasta am creat o bază de date necesară stocării eșantioanelor.

În Fig. 4.7-1 sunt reprezentate câteva tabele din cadrul bazei de date a proiectului: XBeeModem, Boards, SensorData, Sensor și SampleDefinition.

116 4 - Implementarea optimală, prin prisma performanță-fiabilitate, a metodelor de comunicare în rețele de achiziția datelor, cu aplicabilitate la excavatoarele cu cupe

Tabela XBeeModem cuprinde informații necesare protocolului de comunicație radio:

XBeeModemID - identificatorul modului radio de pe DAD,

DiscoverTime - durata de timp cât un nod așteaptă răspunsul de la nodurile din rețea, după ce a transmis comanda „Node Discover” pentru aflarea de informații despre toate nodurilor din rețea

SL+SH formează adresa extinsă pe 64 de biți a nodului din rețea. SL - Destination address Low, SH- Destination address High

My - adresa de rețea pe 16 biți

NI - Node Identifier. Reprezintă un șir de caractere ce caracterizează nodul din rețea. Folosind identificatorul nodului (NI) descriu centrele de sincronizare cu GPS. În identificatorul nodului lor apare cuvântul cheie „GPS”.

PNA - reprezintă identificatorul rețelei. Este necesar deoarece în proiect pot funcționa mai multe rețele radio simultan, iar acest câmp stabilește din ce rețea face parte nodul.

Type - reprezintă tipul nodului din rețea: Coordonator C, Router R sau Dispozitiv Final E.

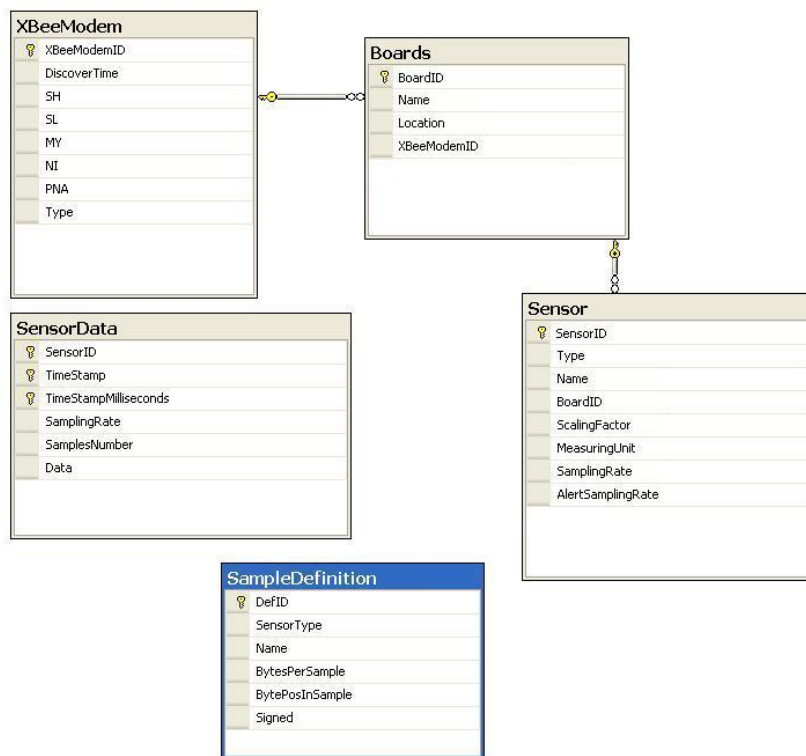


Fig. 4.7-1 Tabele din cadrul bazei de date a proiectului

Tabela Boards furnizează informații despre placa de achiziția datelor, despre DAD:

BoardID - identificatorul plăcii DAD,

Name - numele plăcii

Location – reprezintă locul unde este montat pe excavator
XBeeModemID - identificatorul modului radio de pe DAD, câmp care face legătura cu fișierul XBeeModem

Tabela SensorData furnizează informații despre eșantioanele de date, cuprinzând informații pentru crearea ștampilei de timp:

SensorId - identificatorul senzorului
TimeStamp – Timpul citirii: ore, minute, secunde
TimeStampMilliseconds – Timpul citirii, valoarea milisecundelor
SamplingRate – rata de eșantionare
SamplesNumbers – numărul de eșantioane
Data – data calendaristică la care s-a făcut citirea

Tabela Sensor furnizează informații despre senzorul care este monitorizat și de la care se achiziționează date:

SensorId - identificatorul senzorului
Type – Tipul senzorului. În Tab. 4.2-1 și Tab. 5.4-1 sunt prezentați senzorii cu care se lucrează.

Name – Numele senzorului. Pot exista mai mulți senzori de același tip.
BoardID – DAD-ul care achiziționează date de la senzor
ScalingFactor – valorile citite de la senzori sunt transformate în mărimi interpretabile fizic, în limite de funcționare. De ex. temperatura, presiunea. Valoarea acestui câmp este folosită pentru realizarea corelării.

MeasuringUnit – unitatea de măsură a parametrului fizic monitorizat. În tabelul 4.2 sunt prezentați senzorii și unitatea de măsură aferentă lor.

SamplingRate – rata de eșantionare în regim normal
AlertSamplingRate – rata de eșantionare în regim de alertă

Tabela SamplingDefinition furnizează informații despre eșantioanele citite:

DefID – Definition ID, folosit pentru identificarea setului de date
SensorType - tipul senzorului
Name – identificatorul, numele senzorului
BytePerSample – numărul de octeți corespunzător fiecărui eșantion. Valorile sunt enumerate în Tab. 4.2-1.
BytePosInSample – folosit pentru poziționarea în cadrul eșantioanelor pe mai mulți octeți
Signed – indică dacă valoarea datelor este cu semn sau fără

În Fig. 4.7-2 este ilustrat un fișier de achiziție neprelucrat, care conține eșantioanele de la un senzor. Tipul senzorului este 7. Rata de eșantionare se realizează la 100ms. La fiecare citire se atașează ștampila de timp cu data, ora, minut, secundă și milisekunde. Citirea poate să se facă la mai multe eșantioane deodată, lucru care apare specificat prin numărul de eșantioane citite.

118 4 - Implementarea optimală, prin prisma performanță-fiabilitate, a metodelor de comunicare în rețele de achiziția datelor, cu aplicabilitate la excavatoarele cu cupe

SensorID	TimeStamp	TimeStampMills...	SamplingRate	SamplesNumber	Data
7	24.05.2008 00:01:19	700	100	2	<Binary data>
7	24.05.2008 00:01:19	900	100	1	<Binary data>
7	24.05.2008 00:01:20	0	100	3	<Binary data>
7	24.05.2008 00:01:20	300	100	2	<Binary data>
7	24.05.2008 00:01:20	500	100	2	<Binary data>
7	24.05.2008 00:01:20	700	100	2	<Binary data>
7	24.05.2008 00:01:20	900	100	1	<Binary data>
7	24.05.2008 00:01:21	0	100	3	<Binary data>
7	24.05.2008 00:01:21	300	100	2	<Binary data>
7	24.05.2008 00:01:21	500	100	2	<Binary data>
7	24.05.2008 00:01:25	700	100	2	<Binary data>

Fig. 4.7-2 Datele din fișierul de stocare, înainte de introducerea în baza de date

În cadrul bazei de date există mai multe fișiere care conțin informații brute, cum sunt cele prezentate în Fig. 4.7-2.

În Fig. 4.7-3 sunt prezentate datele care au fost preluate din fișiere și au fost introduse în baza de date. Datele din Fig. 4.7-3 sunt obținute prin interogări (queries).

SensorID	TimeStamp	Value
1	7	24.05.2008 00:01:20.000 -6
2	7	24.05.2008 00:01:20.100 -6
3	7	24.05.2008 00:01:20.200 -7
4	7	24.05.2008 00:01:20.300 -6
5	7	24.05.2008 00:01:20.400 -5
6	7	24.05.2008 00:01:20.500 -9

Fig. 4.7-3 Afișarea datelor din baza de date folosind interogări (queries)

4.8. Concluzii

În vederea implementării optimale, prin prisma performanță-fiabilitate, a metodelor de comunicare în rețele de achiziția datelor, cu aplicabilitate la excavatoarele cu cupe, am efectuat o analiză a vibrațiilor pentru structuri mari, așa cum sunt ele întâlnite în aplicațiile industriale.

În acest sens, am elaborat două lucrări științifice în care urmăresc să pun în relief factorii care contribuie la mișcarea excavatorului, factori sintetizați într-o ecuație diferențială a mișcării acestuia.

Am arătat că vibrațiile pot fi analizate și printr-o secvență de prelucrare a datelor elaborată în Matlab și am criticat această soluție prin faptul că relațiile nu pot lua în considerație toți factorii reali care contribuie la vibrații, fapt pentru care prin cumul defavorabil de erori, analiza poate fi imprecisă și uneori poate duce la interpretări necorespunzătoare.

De asemenea am încercat să găseșc analogii pentru a descrie cât mai real vibrațiile produse de excavatorul cu cupe. În acest sens, culegând date prin puncte

tensiometrică și senzori de accelerație am încercat să asimilez problematica specifică excavatoarelor cu cea mai simplă în tratare, specifică brațului macaralei.

Am conștientizat că prin prisma vibrațiilor, element de importanță capitală, specific excavatoarelor cu cupă, soluția cea mai bună o reprezintă cea propusă în lucrare și bazată pe utilizarea unei rețele de dispozitive de achiziția datelor a cărei configurație optimală, prin prisma performanței și fiabilității am tratat-o în capitolele anterioare ale lucrării.

În continuare lucrarea cuprinde configurația hardware de tip ARM cuprinzând ca element de creștere al fiabilității un watchdog timer (WDT), al cărui rol funcțional este descris detaliat în lucrare.

De asemenea, sunt exemplificate părți din programarea microcontroller-ului cu detalii funcționale prin capturi de ecrane a etapelor de proiectare și de depanare în vederea punerii în funcțiune.

Teza cuprinde de asemenea programarea RDEA, configurarea cadrelor de date, precum și în final o bază de date necesară aplicației cu un număr foarte mare (până la 386) de senzori.

Toate elementele descrise poartă un profund caracter original având în vedere, aplicația particulară pentru care a fost proiectată, ale căror caracteristici au fost cuprinse, așa cum menționăm în propunerea de brevet de invenție [87].

5. EVALUAREA INDICILOR DE PERFORMANȚĂ ȘI FIABILITATE PENTRU REȚELE DE DISPOZITIVE DE ACHIZIȚIA DATELOR APLICATE LA EXCAVATOARELE CU CUPE

Din descrierea realizată până acum se observă că punctul sensibil al cercetării, locul unde pot apărea congestii, îl constituie canalul de transmisie, deoarece chiar dacă capacitatea de stocare și transmisie a unui DAD este suficient de mare, la un moment dat, atât radio cât și serial, doar un singur DAD trebuie să transmită.

Din acest motiv am căutat diverse soluții. Voi prezenta câteva analize efectuate pe parcursul cercetării, bazându-mă pe introducerea metricilor prezentate în capitolul 2.

5.1. Benchmark

Pentru realizarea analizelor am creat un benchmark. Benchmark-ul reprezintă o aplicație sau un nivel de calitate care determină performanțele unui obiect care este măsurat prin comparație cu alte obiecte sau produse similare.

Benchmark-ul poate fi folosit pentru analiza unui computer personal (PC), indicând resursele hard și soft ale acestuia, capacitatea de umplere a hard disk-urilor, viteza sau capacitatea unor periferice. Benchmark-urile sunt folosite pentru a compara performanțele computerelor sau componentelor, cum ar fi plăcile video sau hard disk-urile. Pentru a realiza aceste comparații, aplicațiile de benchmark rulează teste și evaluează rezultatele. Benchmark-ul poate să analizeze un site de Internet indicând volumul traficului cât și volumul unor site-uri similare, necesar comparațiilor.

Cercetarea mea pentru RDEA are particularități specifice și realizarea ei necesită soluții particulare. Pentru analiza performanțelor și compararea diferitelor configurații am fost nevoită să creez un benchmark propriu care inițiază proceduri de analiză ale metricilor de performanță și fiabilitate descrise în paragrafele 2.7 și 2.8.

În urma testării diferitelor abordări am elaborat concluziile bazate atât pe demonstrarea teoretică, cât și pe validări și testări euristice furnizând în final o soluție optimală din perspectiva performanță-fiabilitate.

În Fig. 5.1-1 este prezentată fereastra principală a benchmark-ului, în timpul transmisiei a 6 DAD-uri.

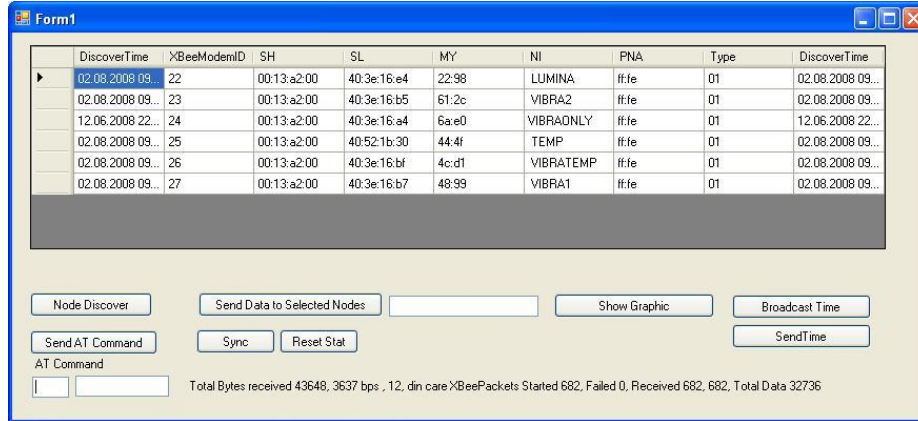


Fig. 5.1-1 Captură cu benchmark-ul folosind 6 DAD-uri

Benchmark-ul este un analizor de date colectate, care furnizează informații la nivel de rețea, dar și la nivel de nod al rețelei. La nivel de rețea urmărește volumul de date transmis în rețea, pachetele pierdute, timpii de transmisie, timeout-ul sau numărul pachetelor retransmise.

Cu ajutorul benchmark-ului se pot extrage informații despre fiecare nod, DAD din RDEA. În Fig. 5.1-2 este prezentat dialogul care realizează interogarea nodurilor. Se poate alege DAD-ul, nodul din rețea care este analizat.



Fig. 5.1-2 Interogare senzori

Combo box-ul ilustrat în Fig. 5.1-3 ilustrează posibilitatea alegerii unui DAD dintre nodurile active în rețea la momentul efectuării înregistrării. Astfel, în exemplul din Fig. 5.1-3, la momentul efectuării înregistrării existau 3 DAD-uri active în rețea: PC_11, PC_12 și PC_13. Pentru analiză a fost ales nodul, DAD-ul PC_11.

122 5 - Evaluarea indicilor de performanță și fiabilitate pentru rețele de dispozitive de achiziția datelor aplicate la excavatoarele cu cupe



Fig. 5.1-3 Alegere DAD analizat

Se poate stabili perioada de timp care interesează. Pentru aceasta, în Fig. 5.1-2 sunt ilustrate combo box-uri care permit stabilirea datei.

Un DAD poate monitoriza mai mulți senzori, cum este ilustrat în Fig. 5.1-2. Se pot alege toți senzorii, dar și selectiv, doar cei care interesează. DAD-ul ales, având identificatorul PC_11 monitorizează doi senzori, unul de temperatură și unul de vibrație a cozii excavatorului. Pentru analiză am ales senzorul de vibrație. Puteau fi alese ambele.

La un senzor pot exista mai multe câmpuri de descrierea funcționării sale. În Fig. 5.1-4, pentru descrierea vibrației se folosesc coordonatele pe axele X,Y și Z, dar și un parametru indicând starea senzorului.

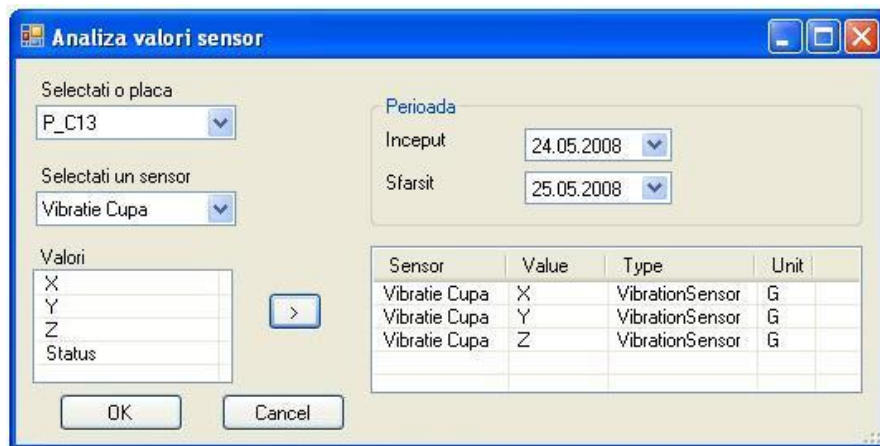


Fig. 5.1-4 Alegere parametri senzor

În analiza propusă de mine am considerat doar valorile vibrației pe cele 3 coordonate.

După stabilirea informațiilor legate de nodul rețelei, data calendaristică la care a avut loc înregistrarea, parametrii senzorului sau senzorilor care sunt analizați, informația poate apărea atât sub forma numerică, cât și sub forma grafică.

Fig. 5.1-5 prezintă graficul vibrației pe cele trei coordonate X, Y și Z. Pe grafic se observă o activitate concentrată.

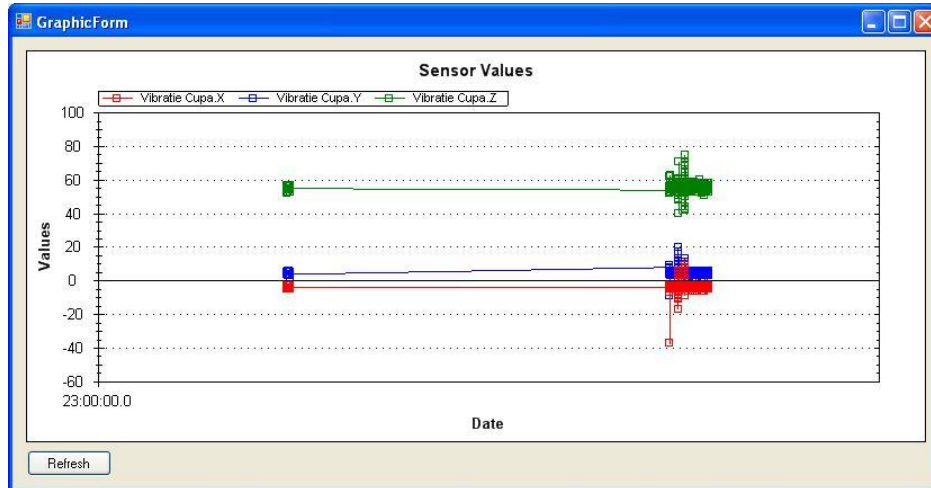


Fig. 5.1-5 Vizualizare parametri senzori

Această porțiune este selectată și se furnizează o examinare mărită a zonei cu perturbații.

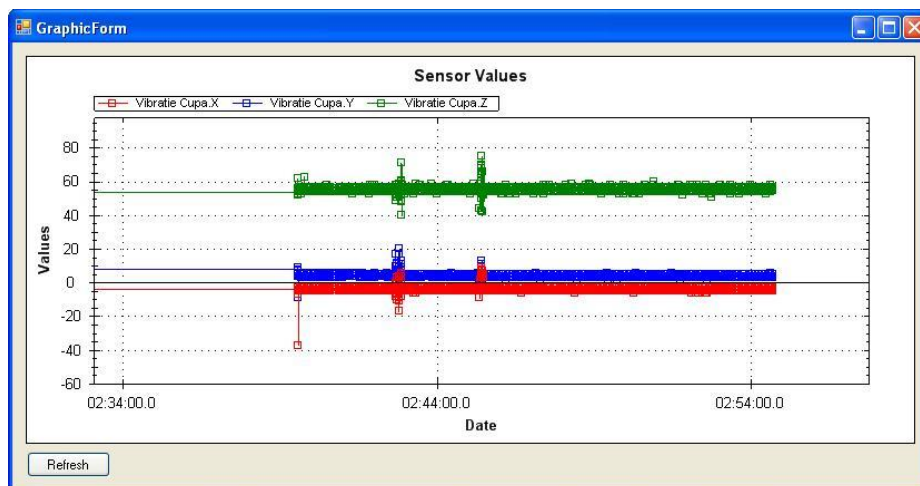


Fig. 5.1-6 Vizualizare zonă perturbații

Fig. 5.1-6 detaliează la o scară mai mare zona de perturbație apărută în Fig. 5.1-5. La fel și Fig. 5.1-7. Se pot observa cele două zone de perturbații din Fig. 5.1-6 care se regăsesc în Fig. 5.1-7.

124 5 - Evaluarea indicilor de performanță și fiabilitate pentru rețele de dispozitive de achiziția datelor aplicate la excavatoarele cu cupe

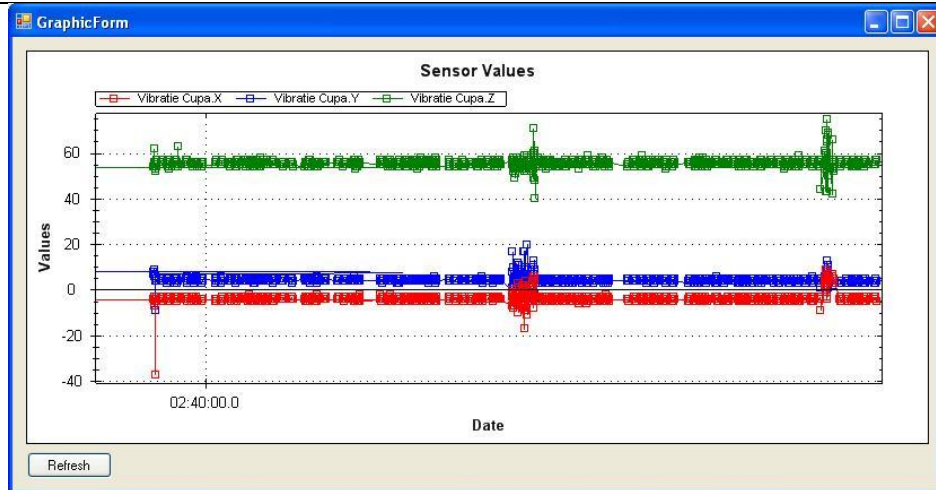


Fig. 5.1-7 Detaliere zonă perturbății

În Fig. 5.1-8 detalierea merge până la nivel valoric. Pe axa X este reprezentată ora la care a avut loc evenimentul, iar pe axa Y valorile vibrațiilor pentru cele trei axe X,Y și Z. Din motive estetice nu am trecut pe grafic valorile vibrațiilor, dar ele se folosesc în analize.

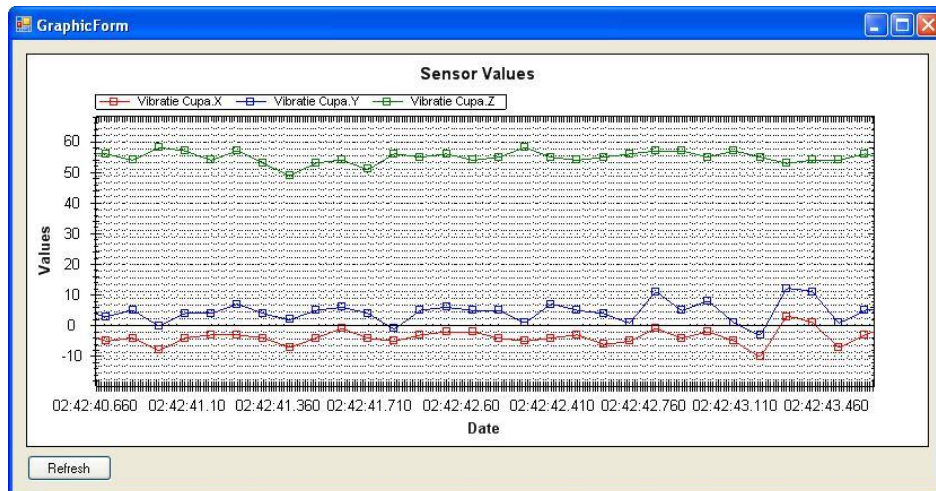


Fig. 5.1-8 Detalierea zonă perturbății la nivel valoric

Pentru furnizarea unui raport optim performanță-fiabilitate am creat mai multe configurații, iar benchmark-ul a fost folosit pentru analiza metricilor propuse în paragrafele 2.7 și 2.8. În continuare voi prezenta câteva dintre aceste analize.

5.2. Analiza întinderii RDEA

Conform documentației [7] am specificat că pentru modulele XBee Series 2 numărul maxim de hopuri parcurse de la sursă până la destinație este 32.

Pentru determinarea întinderii rețelei se va folosi formula 5.1

$$\text{Raza_intinderii} = \text{Distanța_maximă} \times \text{Număr_maxim_de_hopuri} \quad (5.1)$$

În tabelul Tab. 4.6-1 rata de acțiune a ZigBee în mediul urban, în clădiri este 100m, iar în câmp deschis ajunge până la 1,2 km.

Astfel aplicând (5.1) vom avea două raze ale întinderii rețelei, una în mediu urban, clădiri, iar cealaltă în câmp deschis.

$$\text{Raza_intindere_mediu_urban_clădiri} = 100\text{m} \times 32 \text{ noduri} = 3200 \text{ m}$$

Întinderea maximă garantată a rețelei în mediul urban, în interiorul clădirilor, unde apar perturbații este de 3,2 km.

$$\text{Raza_intindere_câmp_deschis} = 1200\text{m} \times 32 \text{ noduri} = 38400 \text{ m}$$

Întinderea maximă garantată a rețelei în câmp deschis, cum este cazul carierelor de lignit de suprafață este de 38,4 km.

5.3. Montarea DAD și a senzorilor pe excavator

În Tab. 5.3-1 sunt prezentate locații în care au fost montate DAD-uri pe excavator.

Numărul DAD	Componentele DAD	Locul de montare pe excavator
1	Temperatură lagăre reductor și cuplaj Temperatură ulei Temperatură pinion	V1
2	Tensiune motor	V1
3	Presiune ulei ungere	V1
4	Vibrație	V1
5	Sistem poziție GPS	V1
6	Turație roată cupe	V1
7	Tensiunea din troliu roată cupe	V2
8	Vibrație	V2
9	Limitatori poziție stânga Limitatori poziție dreapta	V3
10	Stare frâne	V3
11	Limitatori direcție	V4
12	Temperatură lagăr Temperatură tambur acționare roată	V5
13	Patinare bandă	V5
14	Deviere laterală bandă Suprasarcină acționare	V5

Tab. 5.3-1 Locul de amplasare al senzorilor pe excavator

126 5 - Evaluarea indicilor de performanță și fiabilitate pentru rețele de dispozitive de achiziția datelor aplicate la excavatoarele cu cupe

Locurile de montare pe excavator V1, .., V5 sunt punctate în fig.1.3.

Tab. 4.2-1 urmărește să arate tipurile de senzori folosite în exploatare. În tabel nu este trecut numărul senzorilor, aceștia pot fi dublați sau triplați în locația unde sunt instalați.

5.4. Analiza timpului de eșantionare pentru fiecare parametru fizic

Acest paragraf tratează cei mai importanți dintre senzori, prezentați în Tab. 5.4-1. În funcție de modulul de achiziție ales, se ilustrează datele tehnice necesare în cadrul aplicației. Acestea sunt: frecvența de citire a datelor, numărul de valori citite și dimensiunea în octeți a valorilor.

Folosind aceste date se poate determina numărul de citiri efectuate într-un minut și cantitatea de date, exprimată în octeți acumulată timp de un minut.

$$\begin{aligned} \text{Dimensiunea totală octeți pe sec} &= \\ &= \text{nr_valori_la_o_citire} * \text{nr_octeți_o_valoare} * \text{nr_citiri/sec} \quad (5.4-1) \end{aligned}$$

Tipul senzor	Frecv. de citire a datelor	Nr. valori la o citire	Dimensiunea în octeți a datelor unei citiri	Număr de citiri pe minut	Dimensiunea totală octeți pe minut
Temp.	1 min	1	2	1	2
Vibrație	20 ms	3	1	60 * 50	9000
Presiune	20 ms	1	2	60 * 50	6000
Tensiune, Curent	50 ms	3	2	60 * 20	7200
GPS	1 sec	5	42=19+6+9+4+4	60	2520

Tab. 5.4-1 Tipul senzorilor, rata de citire și dimensiunea eșantioanelor

$$\begin{aligned} \text{Dimensiunea totală octeți pe minut} &= \\ &= \text{nr_valori_la_o_citire} * \text{nr_octeți_o_valoare} * \text{nr_citiri/min} \quad (5.4-2) \end{aligned}$$

$$\begin{aligned} \text{Temperatura} &= 1 * 2 * 1 \\ \text{Vibrație} &= 3 * 1 * 60 * 50 = 9000 \\ \text{Lumina} &= 1 * 2 * 60 * 50 = 6000 \\ \text{Tensiune} &= 3 * 2 * 60 * 20 = 7200 \\ \text{GPS} &= 42 * 60 = 2520 \end{aligned}$$

Voi considera cazul vibrației, care este eșantionată la fiecare 20ms. La o citire se citesc 3 valori, coordonatele pe axele X, Y și Z. Fiecare valoare are un octet.

Urmărind Tab. 5.4-1 observăm că mărimea cea mai solicitantă este vibrația. Un eșantion corespunzător vibrației conține 3 octeți. În regim normal eșantioanele sunt citite la 20ms, iar în cazul unei avarii, frecvența de eșantionare crește la 2ms.

5.5. Mărimea optimă a pachetului de date

Pentru o rețea cu un singur nod, transmit 16k octeți de date și variez mărimea pachetului de date, la 16 octeți, 32 octeți, 48 octeți, 56 octeți și 64 octeți. Analizez timpii de transmisie, și numărul retransmisilor.

Dimensiune date = 16k octeți
 Dimensiune pachet = 16 octeți
 Nr. pachete coresp. cant. de date = 1022 pachete
 Nr. retransmisii = 0 pachete
 În total s-au transmis = 1022 pachete
 Timp transmisie = 29,860 secunde
 Transmisie medie pachet = 29msec
 Număr de plăci din rețea = 1

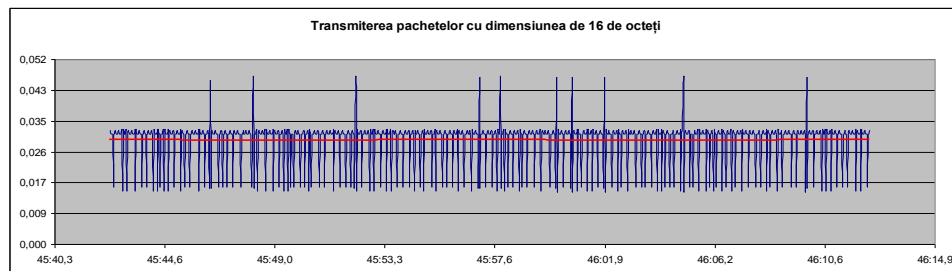


Fig. 5.5-1 Timpii de transmisie a 16k octeți de date în pachete de 16 octeți

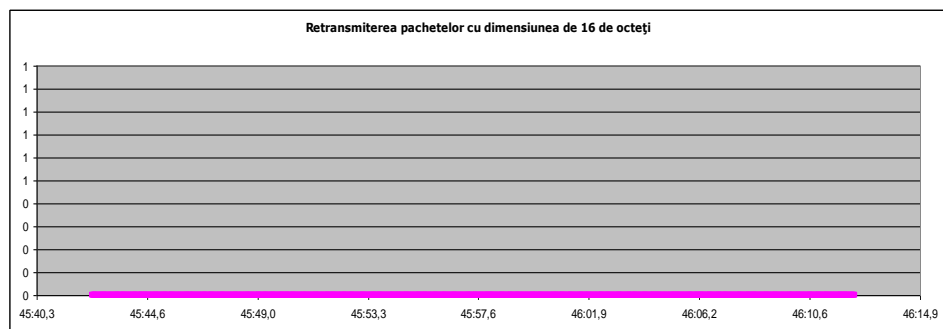


Fig. 5.5-2 Retransmisia pachetelor în cazul ilustrat Fig. 5.5-1

Experimentul ilustrat în Fig. 5.5-1 și Fig. 5.5-2 este reprezentat în Tab. 5.5-1 pe coloana 1 și transmite 16k octeți de date, în pachete având dimensiunea de 16 octeți. Experimentele au continuat, păstrând constantă cantitatea de date transmisă, 16k octeți și variind dimensiunea pachetului. Rezultatele sunt trecute în Tab. 5.5-1.

Dimensiune date	16octeți	16octeți	16octeți	16octeți	16octeți
Dimensiune pachet	16 octeți	32 octeți	48 octeți	56 octeți	64 octeți
Număr pachete coresp. cant. de date	1022 pachete	512 pachete	341 pachete	292 pachete	256 pachete
Nr. retransmisii	0 pachete	0 pachete	0 pachete	42 pachete	226 pachet
Nr. total pachete	1022 pachete	512 pachete	341 pachete	334 pachete	482 pachete
Timp transmisie	29.860 s	17,203 s	12,828 s	16,062 s	35,031 s
Transmisie medie pachet	29 msec	34 msec	38 msec	48 msec	73 msec
Număr de noduri din rețea	1	1	1	1	1

Tab. 5.5-1 Analiza dimensiunii optime a pachetului

Extrăgând din cadrul Tab. 5.5-1, valorile timpului cât a durat transmisia în funcție de dimensiunea pachetului obțin rezultatele ilustrate în Tab. 5.5-2.

Dimensiune pachetului de date	Timp necesar transiterii
16	29,860
32	17,203
48	12,828
56	16,062
64	35,031

Tab. 5.5-2 Analiza dimensiune pachet în funcție de timpul de transmisie

Rezultatele din Tab. 5.5-2, sunt ilustrate în graficul prezentat în Fig. 5.5-3.

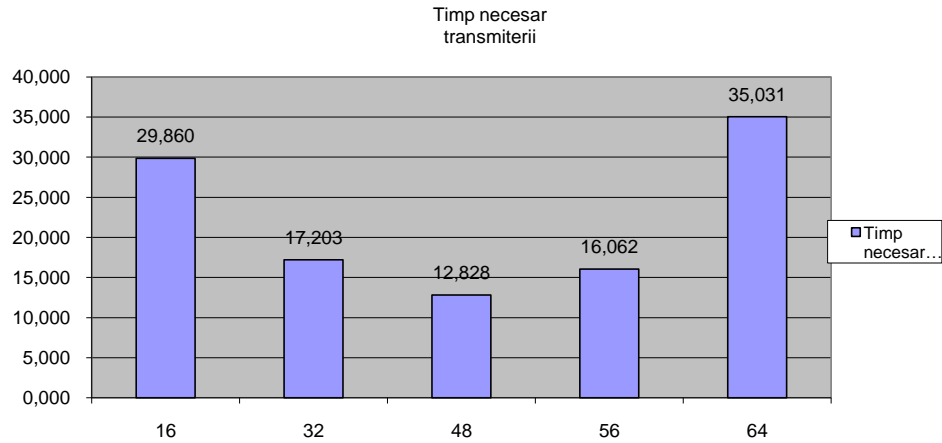


Fig. 5.5-3 Analiza dimensiune pachet în funcție de timpul de transmisie

În urma testelor am observat că la transmiterea a 16k octeți de date, pachetele cu mărimea de 48 de octeți ajung cel mai rapid la destinație, în 12,828ms.

În Fig. 5.5-4 și Fig. 5.5-5 este ilustrat experimentul cu pachetele de date având lungimea de 48 de octeți.

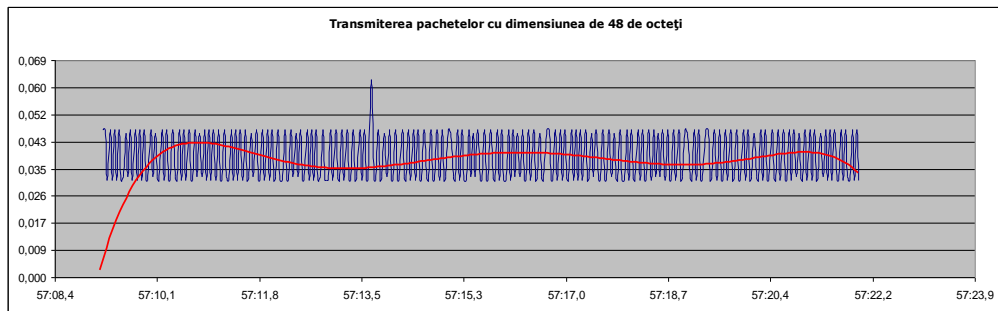


Fig. 5.5-4 Timpii de transmisie a 16k octeți de date în pachete de 48 octeți

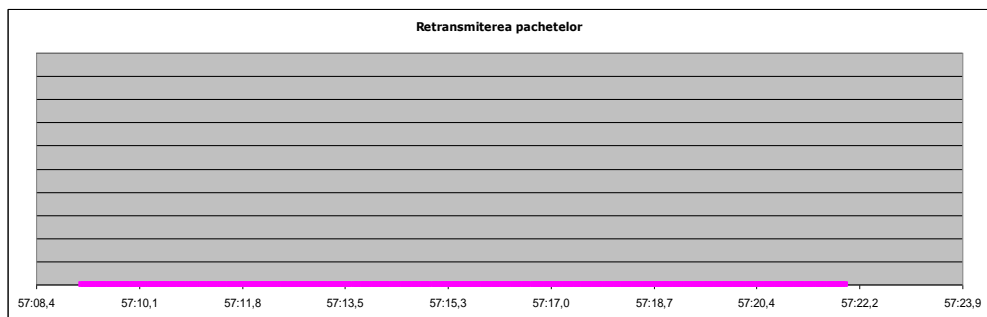


Fig. 5.5-5 Retransmisia pachetelor în cazul ilustrat în Fig. 5.5-4

5.6. Variația timpului de transmisie în funcție de volumul de date transportat

În cadrul următorului experiment se transmit 16 kocteți de date, folosind pachete de aceeași dimensiune, de 48 de octeți. Dar transmiterea se face de la noduri diferite din rețea.

Dimensiune date	16k octeți	16k octeți	16 kocteți	16 kocteți	16 kocteți
Dimensiune pachet	48 octeți	48 octeți	48 octeți	48 octeți	48 octeți
Număr pachete coresp. cant. de date	341 pachete	341 pachete	341 pachet	341 pachete	341 pachete
Nr. retransmisii	4 pachete	0 pachete	2 pachete	0 pachete	6 pachete
Nr. total pachete	345 pachete	341 pachete	343 pachete	341 pachete	347 pachete
Timp transmisie	9,766 s	12,828 s	9,594 s	12,937 s	10,0 s
Transmisie medie pachet	28 msec	38 msec	28 msec	38 msec	29 msec
Număr de noduri din rețea	1 COM3	1 COM4	1 COM5	1 COM6	1 COM8

Tab. 5.6-1 Analiza timpilor de transmisie de la diferite noduri din rețea

Se urmărește comportamentul diferitelor noduri din rețea, care deși sunt proiectate identic din punctul de vedere hardware, iar condițiile de efectuare a testelor au fost identice, rezultatele sunt diferite. În urma acestor experimente, în analize se folosesc valori medii ale timpilor obținuți.

Fig. 5.6-1 reprezintă experimentul cu timpul cel mai bun din prima coloană a Tab. 5.6-1 și ilustrează timpii de transmisie ai pachetelor.

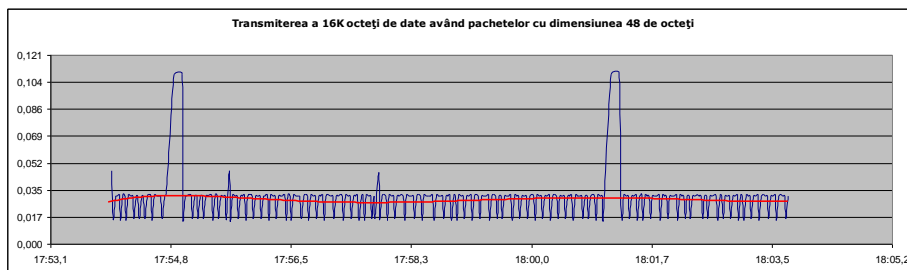


Fig. 5.6-1 Timpul total de transmisie obținut 9,766s

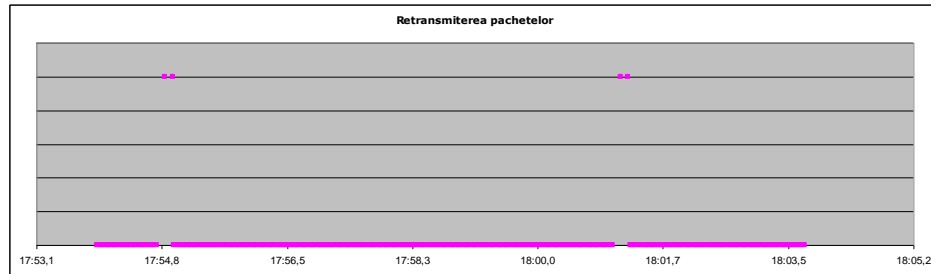


Fig. 5.6-2 Retransmisia pachetelor în cazul experimentului din Fig. 5.6-1

În Fig. 5.6-2 sunt reprezentate cazurile de retransmisie a pachetelor. Chiar dacă experimentul a trimis 16k octeți de date în 9,766s, totuși au existat 4 pachete care au fost retransmise, sau un singur pachet a fost retransmis de 4 ori.

În următoarea serie de experimente am analizat timpii de transmise dintr-o rețea formată din 3 noduri, în care fiecare nod transmite câte 16 octeți.

Dimensiune date	16K octeți	16k octeți	16k octeți	16k octeți	16k octeți	16 octeți
Dimensiune pachet	48 octeți	48 octeți	48 octeți	48 octeți	48 octeți	48 octeți
Număr pachete coresp. Cant. De date	341 pachete	341 pachete	341 pachete	341 pachete	341 pachete	341 pachete
Nr. retransmisii	17 pachete	15 pachete	2 pachete	0 pachete	0 pachete	2 pachete
Nr. total pachete	358 pachete	356 pachete	343 pachete	341 pachete	341 pachete	343 pachete
Timp transmisie	14,5s	13,797 s	12,687 s	12,250 s	12,453 s	12,156 s
Transmisie medie pachet	41 msec	39 msec	37 msec	36 msec	37 msec	35 msec
Numele stațiilor	COM3	COM4	COM8	COM3	COM4	COM5
Număr de noduri din rețea	3	3	3	3	3	3
Ordinea pornirii	1	2	3	3	2	1

Tab. 5.6-2 Analiza timpilor de transmisie într-o rețea cu 3 noduri

Fig. 5.6-3 reprezintă experimentul din prima coloană a Tab. 5.6-2. Sunt reprezentați timpii de transmisie ai pachetelor.

132 5 - Evaluarea indicilor de performanță și fiabilitate pentru rețele de dispozitive de achiziția datelor aplicate la excavatoarele cu cupe

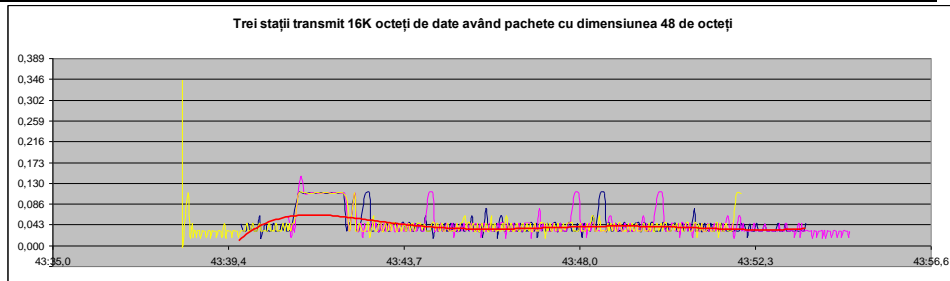


Fig. 5.6-3 Timpii de transmisie pentru o rețea cu 3 noduri

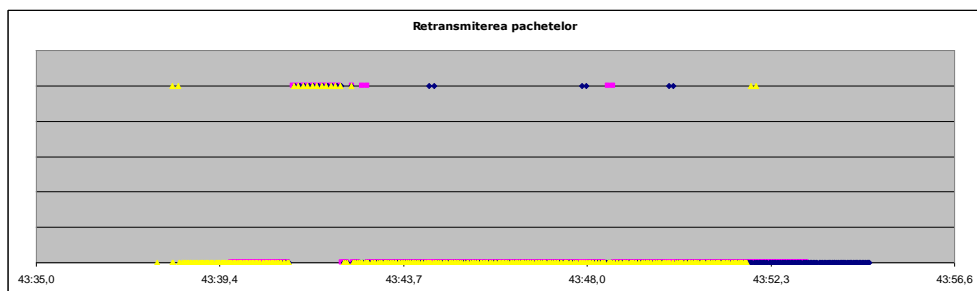


Fig. 5.6-4 Retransmisiile pachetelor în cazul experimentului din Fig. 5.6-3

În cazul acestui experiment, timpii au ieșit mai mari decât la experimentul din coloana a doua a Tab. 5.6-2. Acest lucru se datorează și faptului că numărul de retransmisii al pachetelor a fost substanțial mai mare decât în cazul celui de-al doilea experiment. Fig. 5.6-4 prezintă retransmisiile pachetelor.

În Fig. 5.6-3 și Fig. 5.6-5 se observă că la un moment doar un singur nod transmite în rețea. Dacă mai încearcă un alt nod să transmită apar situațiile de coliziune, care sunt reprezentate prin retransmiterea pachetelor, ilustrate în fig. 5.6-6 și 5.6-8.

Următorul experiment reprezintă o rețea formată din 5 noduri, în care fiecare nod transmite câte 16 octeți de date. Fig. 5.6-5 ilustrează timpii de transmisie pentru cele 5 noduri ale rețelei, desenate cu 5 culori diferite, iar în Fig. 5.6-6 sunt reprezentate retransmisiile pachetelor.

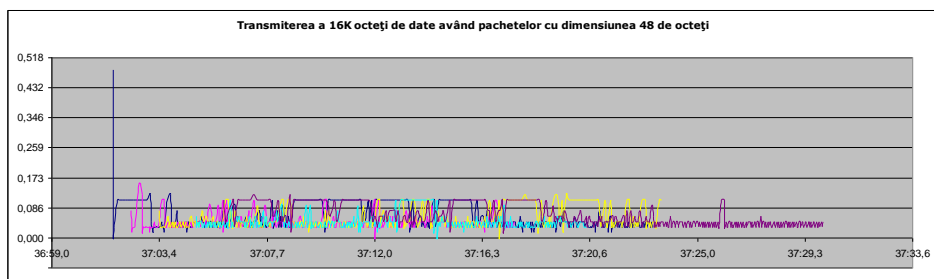


Fig. 5.6-5 Timpii de transmisie pentru o rețea cu 5 noduri

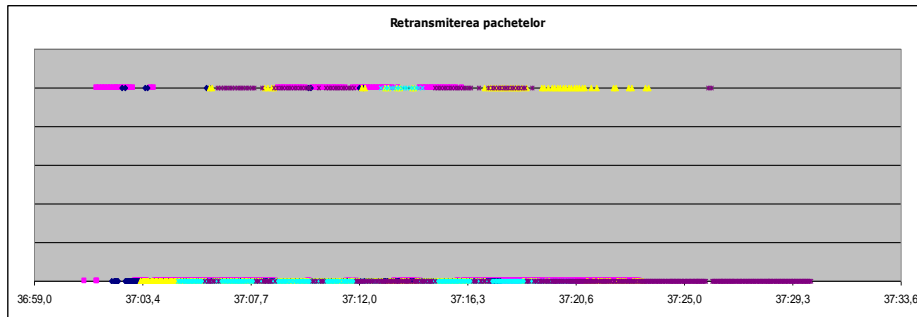


Fig. 5.6-6 Retransmitema pachetelor în cazul experimentului din Fig. 5.6-5

Fig. 5.6-6 trasează retransmitema pachetelor în cazul experimentului din Fig. 5.6-5.

Tab. 5.6-3 prezintă rezultatele experimentului.

Dimensiune date	16k octeți	16k octeți	16k octeți	16k octeți	16k octeți
Dimensiune pachet	48 octeți	48 octeți	48 octeți	48 octeți	48 octeți
Număr pachete coresp. cant. de date	341 pachete	341 pachete	341 pachete	341 pachete	341 pachete
Nr. retransmitemii	71 pachete	10 pachete	53 pachete	15 pachete	73 pachete
Nr. total pachete	412 pachete	351 pachete	394 pachete	356 pachete	414 pachete
Timp transmisie	22,14s	15,484 s	20,25 s	15,735 s	24,25 s
Transmisie medie pachet	54 msec	44 msec	51 msec	44 msec	59 msec
Numele stațiilor	COM3	COM4	COM5	COM6	COM8
Număr de noduri din rețea	5	5	5	5	5

Tab. 5.6-3 Analiza timpilor de transmisie într-o rețea cu 5 noduri

Propun următoarea analiză. Având o rețea cu n noduri vreau să studiez cum este mai performantă transmisia: fiecare nod să-și transmită serial datele sau să transmită toate nodurile în paralel, lăsând protocolul să rezolve transmisia. Pentru particularizare am considerat o rețea cu 5 noduri ($n=5$).

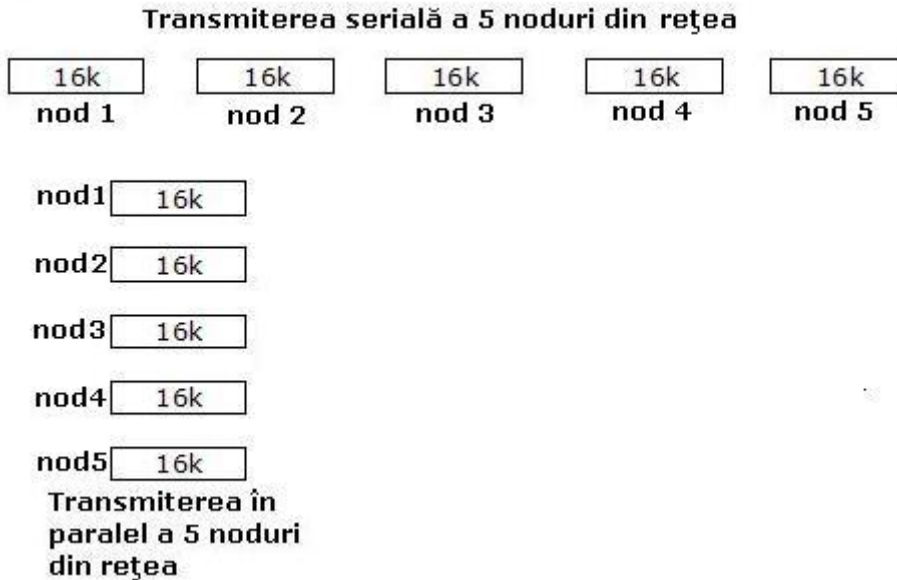


Fig. 5.6-7 Premisele analizei pentru o rețea cu n=5 noduri

Pe baza experimentelor efectuate până acum vreau să realizez o analiză care să indice dacă într-o rețea cu n noduri timpul total de transmisie al celor n noduri care transmit serial, pe rând date este mai bun decât timpul de transmisie a n noduri în paralel.

În Tab. 5.6-4 sunt reprezentați timpii de transmisie pentru 5 plăci care transmit pe rând, serial și pentru 5 plăci care transmit paralel. Cantitate de date transmisă crește de la 16kocțeți până la 8Mocțeți.

kB	Timp transmisie	
	5 placi serial	5 placi paralel
16	55,13	19,57
32	110,25	39,14
64	220,50	78,29
128	441,00	156,57
256	882,00	313,15
512	1764,00	626,30
1024	3528,00	1252,60

Tab. 5.6-4 Timp de transmisie comparativi

În urma analizei comparative rezultă că transmiterea în rețea, în paralel a mai multor noduri oferă un timp superior transmiției seriale.

Acest rezultat este ilustrat și în Fig. 5.6-8.

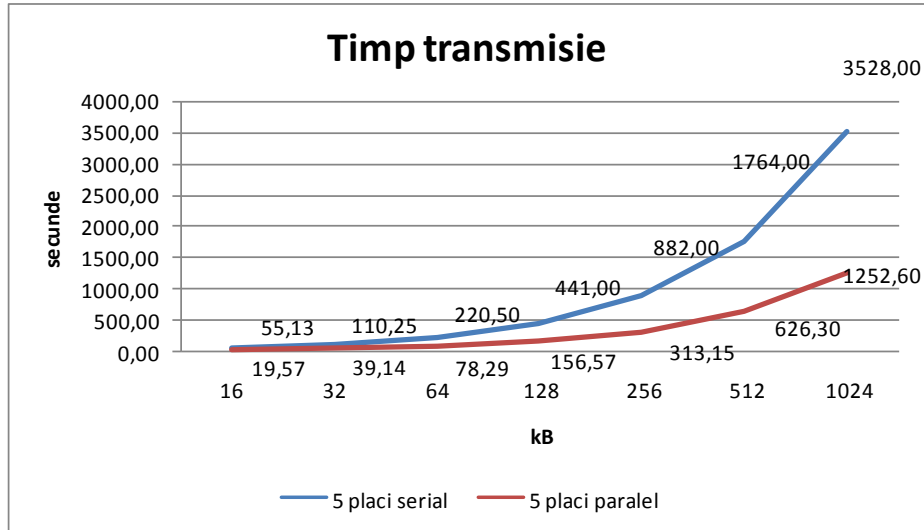


Fig. 5.6-8 Variația timpilor de transmisie

Graficul de sus (având culoarea albastră) reprezintă variația timpului de transmisie pentru cinci plăci care transmit serial, iar graficul de jos (roșu) reprezintă variația timpilor în cazul transmiterii în rețea în paralel.

5.7. Dimensiunea buffer-ului intern al unui nod de rețea

Fiecare DAD conține un buffer intern care stochează datele. Mărimea buffer-ului intern constituie una dintre variabilele rețelei. În cazul în care viteza de transmisie a eșantioanelor este mai mică decât generarea acestora, valorile noi vor fi suprascrise peste cele vechi, existente. Din acest motiv, este necesară dimensionarea buffer-ului intern al unui DAD și adăugarea, după caz, a unei memorii locale de stocare.

În urma testelor efectuate în mod repetat am ajuns la concluzia că un DAD transmite un pachet având dimensiunea de 48 caractere, în medie în 16-17 milisecunde. Tot în urma testelor am observat că 3 stații care transmit simultan un pachet de 48 de caractere au nevoie în medie de 40ms, iar 5 DAD-uri de 50ms.

1 DAD1 pachet 48 caractere date17 ms
3 DAD1 pachet 48 caractere date40 ms
5 DAD1 pachet 48 caractere date50 ms

Tab. 5.7-1 Timpii reali de transmitere ai unui pachet de date

Într-un pachet de 48 de caractere, există aproximativ 36 de octeți de date, restul fiind folosiți pentru header. Structura detaliată a pachetului de date este descrisă în paragraful 3.2.

Voi considera cazul vibrațiilor deoarece acesta este cel mai sensibil.

Într-un pachet de 48 de octeți, există aproximativ 12 eșantioane, adică 12X3 octeți. Aproximând, putem spune că un pachet de date transmite 10 eșantioane o dată.

Datele experimentale, arată că pentru a transmite 10 eșantioane sunt necesare 50ms, deoarece într-o rețea de 5 noduri un pachet de date are nevoie în medie de 50ms pentru a ajunge la destinație.

Eșantionarea vibrațiilor în mod normal se face la 20ms și în mod de alertă la 2ms.

La fiecare 20 ms este interogată un senzor și se obține un eșantion de date.

În 10 minute trebuiesc citite 30000 de eșantioane, la rata de 20ms.

$10 \times 60 \times 50 \text{ eșantioane/sec} = 30\ 000 \text{ eșantioane la o eșantionare de } 20\text{ms}$

Știind că 1 pachet cu 10 eșantioane e transmis în 50ms, 30000 de eșantioane vor fi transmise în 150 secunde (150 000ms).

Cele 30000 de eșantioane sunt grupate în 300 de pachete care sunt transmise în aproximativ 2,5 minute.

Voi trata cazul eșantionării la 2ms. Cercetătorii din domeniul mecanic au spus că la producerea unei perturbații de alertă, aceasta poate dura de la câteva secunde până la maxim 10 minute, timp în care citirea datelor se va face la 2ms.

Astfel în 10 minute se strâng 300 000 eșantioane

$10 \times 60 \times 500 \text{ eșantioane/sec} = 300\ 000 \text{ eșantioane}$

Cele 300 000 de eșantioane vor fi transmise în 25 de minute(1500 s).

Dar un eșantion conține 3 octeți de date, conform Tab. 5.4-1.

$300\ 000 \text{ eșantioane} \times 3 \text{ octeți} = 900\ 000 \text{ octeți pentru vibrație.}$

Pentru transmiterea a 900000 octeți = 879 Kocteți sunt necesare 25 minute.

Cantitățile de date rezultate în urma eșantionării, colectate de la un senzor de vibrație, în funcție de rata de citire, le-am trecut în Tab. 5.7-2.

Rata de eșantionare	Timp de eșantionare	Număr eșantioane	Număr octeți	Buffer intern date
20 ms	10 minute	30000	90000	88 octeți
2ms	10 minute	300000	900000	850Kocteți

Tab. 5.7-2 Cantitatea de date rezultată în urma eșantionării

În cazul eșantionării la 20ms, 90000 de octeți de date, citiți în 10 minute, pot fi transmiși în aproximativ 3 secunde. În acest caz nu este practic nevoie de un buffer intern.

În cazul eșantionării la 2ms, 300 000 de eșantioane, citiți în 10 minute, vor fi transmiși în 25 minute. În acest caz este nevoie de un suport de stocare.

Mediul de stocare va trebui să conțină cel puțin 900 000 octeți. Putem considera suficientă o memorie de 1Moctet.

Atrag atenția că rețeaua de 5 noduri care a fost dimensionată aici permite ca toate nodurile să aibă senzor de vibrație. Cu alte cuvinte, în acest paragraf s-a tratat cazul cel mai defavorabil: worst case scenario.

Am concluzionat că în cel mai defavorabil caz, va fi nevoie de o memorie de 1M octet de date la fiecare stație. Dar, o dată cu citirea valorilor este necesară alăturarea unei ștampile de timp, folosită la analiză.

Uzual, stocarea va trebui să conțină valoarea citită și ștampila de timp atașată ei, așa cum sunt stocate în fișiere și ilustrate în Fig. 5.7-1.

TimeStamp	TimeStampMillis...	SamplingRate	SamplesNumber
24.05.2008 00:01:19	700	100	2
24.05.2008 00:01:19	900	100	1
24.05.2008 00:01:20	0	100	3
24.05.2008 00:01:20	300	100	2
24.05.2008 00:01:20	500	100	2

Fig. 5.7-1 Ștampila de timp atașată datelor

Atașând fiecărui eșantion ștampila de timp se va se vor folosi cel puțin 15 octeți de date, 6 pentru data calendaristică și 9 pentru ora.

Urmărind furnizarea unei soluții performante, dar și fiabile am realizat următoarea îmbunătățire: în primul cadru transmit header-ul, urmat de date, ca în Tab. 5.7-3.

Header-ul folosit conține ștampila de timp formată din dată, oră, minute, milisecunde. Sunt specificate și rata de citire a eșantioanelor și numărul de eșantioane din cadru.

Header:
Data, Ora, Milisec, Rata de eșantionare, Samples no.
Valoare 1
Valoare 2
Valoare n

Tab. 5.7-3 Îmbunătățire pentru mărirea performanței

În acest fel se știe data și ora precisă când a fost citit eșantionul și se mai cunoaște și rata de eșantionare. Având în vedere că nodurile RDEA lucrează sincronizat, această îmbunătățire poate fi aplicată. Header-ul conține 17 octeți, ceea ce poate fi considerat neglijabil având în vedere că este transmis doar o singură dată. Următoarele cadre vor conține doar numărul de eșantioane și eșantioanele propriu-zise. Pe parcursul funcționării, dacă rata de citire a datelor se modifică, se va genera din nou un cadru având header-ul prezentat în Tab. 5.7-3.

138 5 - Evaluarea indicilor de performanță și fiabilitate pentru rețele de dispozitive de achiziția datelor aplicate la excavatoarele cu cupe

În continuare vom efectua o analiză din punctul de vedere al fiabilității. Am analizat situațiile de alertă în care senzorii produc o cantitate mare de date, care trebuie stocate. Am calculat că în acele cazuri este necesară o memorie având capacitatea de 850k octeți. Aproximând propunem o memorie de capacitatea 1M octet pentru fiecare nod din rețea care produce un volum mare de date.

În continuare voi analiza în cât timp se va transmite în rețea informația din memoria de 850k.

Consider că DAD-ul eșantionează datele la 20ms. Intervine un eveniment, eșantionarea continuă să se facă la 2ms, timp de 10 minute. În timpul celor 10 minute, DAD-ul transmite în rețea, dar datele sunt stocate în memorie. După acest interval, DAD-ul revine la rata de eșantionare de 20ms, continuând să emită în rețea.

Mă interesează în cât timp vor fi transmise datele din memorie în rețea și care este durata de timp minimă la care se pot succeda perioadele de alertă, astfel încât să nu se piardă date.

Știu că 10 eșantioane sunt transmise în 100ms. La 850k octeți corespund 300000 eșantioane, care vor fi transmise în 25 minute.

Considerând o abordare simplificată, dar reală, pot asuma că nu toate nodurile din rețea monitorizează vibrații.

Cu toate acestea, unui nod care a generat un volum mare de date îi sunt necesare 30 de minute pentru a transmite datele în rețea.

Este recomandabil pentru a nu se pierde datele să apară câte o situație defavorabilă perturbatoare la cel mult 30 de minute. Dar aceste fenomene pot fi verificate doar în practică și depind foarte mult de condițiile din carieră, condiții care pot să varieze de la un excavator la altul. În această situație, în funcție de ceea ce se petrece pe teren, recomand pentru cazurile în care perturbațiile se produc mai repede de 30 de minute să se folosească o memorie mai mare, de 2M sau 4M octeți.

5.8. Decizii luate în cadrul proiectului pentru mărirea performanței

Am eliminat o parte din ștampila de timp, lăsând doar secunda și milisecundele, iar în schimb am introdus un mecanism de sincronizare al DAD-urilor în rețea.

Din motive de performanță, știind că nodurile RDEA sunt sincronizate, se cunoaște data, ora și minutul, iar în cadrul ștampilei de timp, TimeStamp, se transmit doar secunde și milisecunde.

5.9. Determinarea ratei de succes SR (succes ratio)

În paragraful 2.8.1 s-a definit metrica SR. Pe baza testelor doresc să determin rata de succes în cazul RDEA.

Pentru determinarea unui procent cât mai real am realizat mai multe testări și am furnizat valori medii pentru SR.

Consider cazul cel mai defavorabil, al existenței în rețea a senzorilor de vibrație. Voi considera pe rând o rețea care conține 2 senzori cu vibrații, 3 sau cinci. Voi considera cazul defavorabil, de alertă, în care senzorii sunt consultați la fiecare 2ms, generându-se o cantitate mare de date. Din acest motiv, în cele trei experimente se va transmite 1M octet de date, fiecare pachet având mărimea de 48 de octeți.

În cazul unei rețele care are 2 senzori de vibrație, rezultatele din Tab. 5.9-1 arată o rată de succes în medie de 99%, rata pachetelor pierdute este sub 1%, iar timpul mediu de transmisie a 1M octet de date este de aproximativ 12 minute.

Placa	Pierdute	Primate	Transmise	SR	PR	Timp
1	169	21676	21845	99,23	0,77	11:40,6
2	208	21637	21845	99,05	0,95	12:21,9
Total	377	43313	43690	99,14	0,86	12:01,2

Tab. 5.9-1 Rata de succes pentru două noduri cu vibrații

În cazul unei rețele care are 3 senzori de vibrație, rezultatele din Tab. 5.9-2 arată scăderea ratei de succes la o medie de 89%, rata pachetelor pierdute crește spre 6%, iar timpul mediu de transmisie a 1M octet de date este de aproximativ 16 minute.

Placa	Pierdute	Primate	Transmise	SR	PR	Timp
1	465	21380	21845	97,87	2,13	13:03,5
2	1568	17962	21845	82,22	7,18	20:11,3
3	1850	19131	21909	87,32	8,44	15:37,4
Total	3883	58473	65599	89,14	5,92	16:17,4

Tab. 5.9-2 Rata de succes pentru trei noduri cu vibrații

În cadrul experimentului cu 5 noduri care produc o mare cantitate de date am realizat 3 scenarii legate de retransmiterea pachetelor: în primul caz pachetele pierdute nu sunt retransmise, în al doilea caz pachetele sunt retransmise de maxim 2 ori, iar în al treilea caz retransmisia pachetelor pierdute se realizează de maxim 5 ori.

Placa	Success Rate		
	Fara Retry	2 Retry	5 Retry
1	87,21	93,75	96,53
2	85,70	93,69	95,66
3	85,91	93,96	95,74
4	85,93	93,21	94,54

140 5 - Evaluarea indicilor de performanță și fiabilitate pentru rețele de dispozitive de achiziția datelor aplicate la excavatoarele cu cupe

5	86,40	92,96	96,75
Medie	86,23	93,51	95,84

Tab. 5.9-3 Rata de succes pentru cinci noduri cu vibrații

Rezultatele experimentului sunt ilustrate în Tab. 5.9-3, indicând cea mai bună rată de succes, de aproximativ 95,8%, pentru scenariul 3 în cadrul căruia pachetele pierdute sunt retransmise de maxim 5 ori.

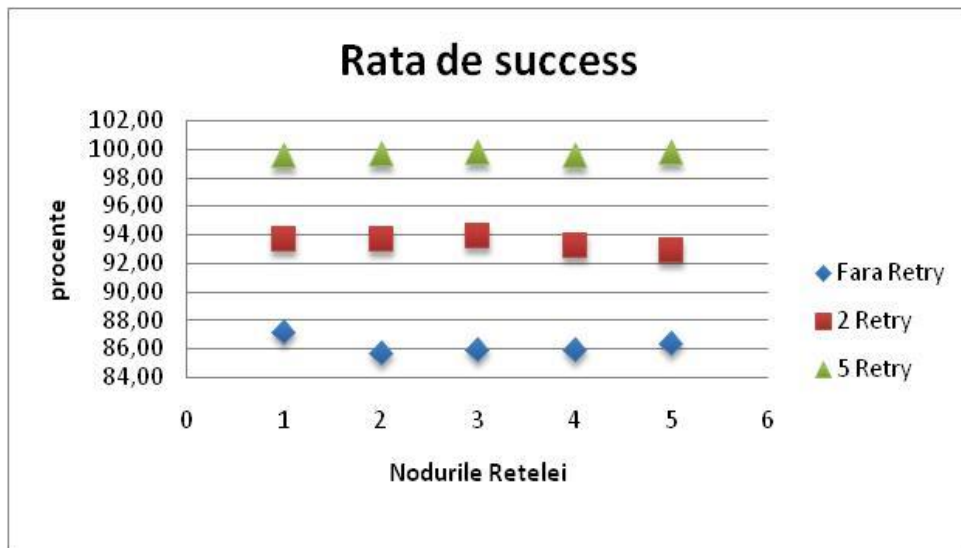


Fig. 5.9-1 Rata de succes pentru cinci noduri cu vibrații

5.10. Determinarea numărului optim de retransmisii în cadrul RDEA

Experimentul anterior, realizat cu RDEA formată din 5 noduri care transmit o mare cantitate de date, este analizat în Tab. 5.10-1.

Placa	Rata pachetelor pierdute		
	Fara Retry	2 Retry	5 retry
1	12,83	6,25	3,47
2	14,35	6,31	4,34
3	14,18	6,04	4,26
4	15,28	6,79	5,46
5	14,01	7,04	3,25
Medie	14,13	6,49	4,16

Tab. 5.10-1 Rata pachetelor pierdute pentru cinci noduri cu vibrații

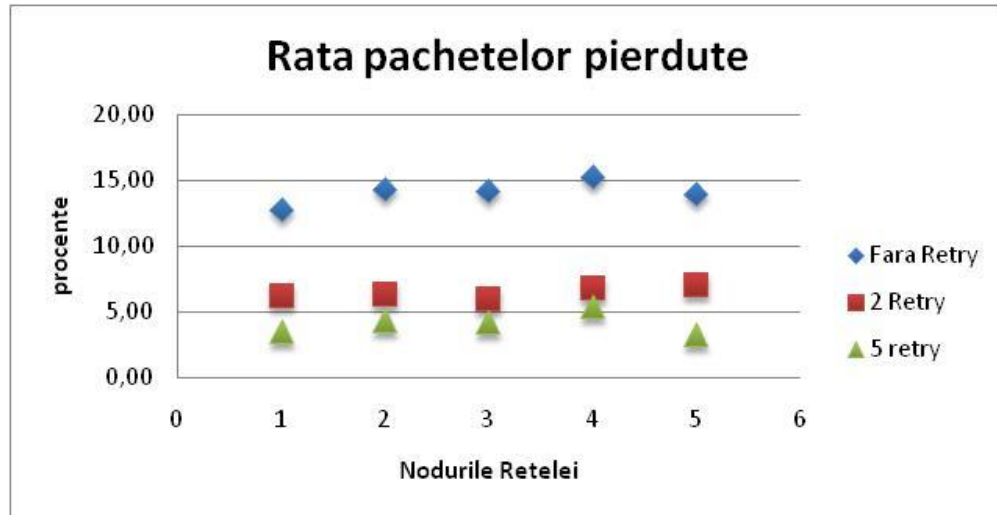


Fig. 5.10-1 Rata pachetelor pierdute pentru cinci noduri cu vibrații

Rata pierderii pachetelor este invers proporțională cu rata de succes, astfel încât scenariul cu 5 retransmisii oferă cea mai bună rată de pierdere a datelor, de 4%, față de rata de 6,5% în cazul a două retransmisii sau 14% în situația în care nu se realizează nici o retransmisie.

5.11. Analiza timpului de transmisie în cadrul RDEA

Placa	Timp		
	Fara Retry	2 Retry	5 Retry
1	17:39,6	20:08,1	33:26,3
2	18:05,5	20:05,7	33:26,3
3	18:03,4	20:02,4	32:08,4
4	18:03,3	19:58,4	24:25,9
5	17:53,5	19:58,2	32:18,2

Tab. 5.11-1 Timpul transmiterii pentru cinci noduri cu vibrații

În Tab. 5.11-1 este analizat timpul de transmisie din cadrul experimentului prezentat în paragrafele 5.9 și 5.10. Scenariul 3 din experimentul anterior permitea retransmiterea unui pachet pierdut de maxim 5 ori. Acest scenariu oferea o rată de succes în medie de 95,8%, iar rata pachetelor pierdute era de 4%. Retransmiterea unui pachet care nu ajunge la destinație de 5 ori implică un timp suplimentar și acest lucru este ilustrat prin valoarea medie de 31 de minute de transmisie.

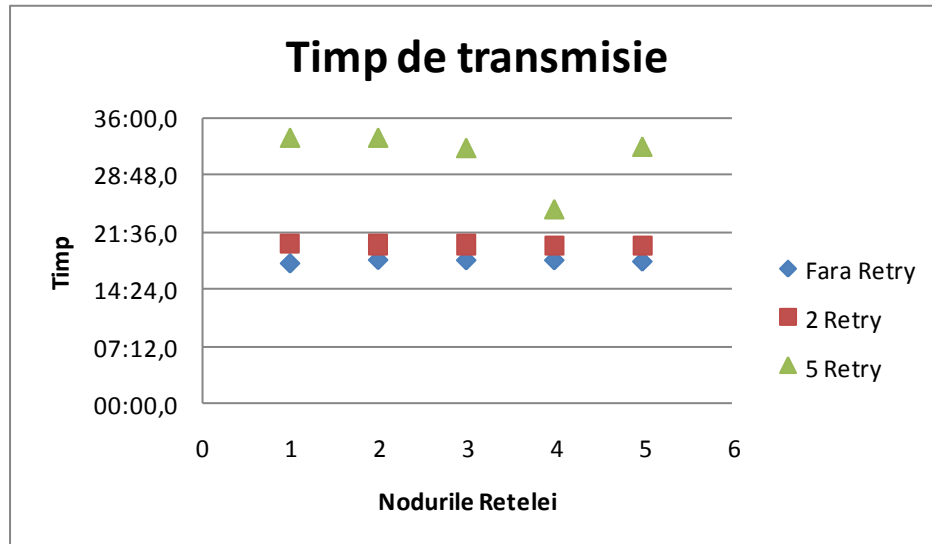


Fig. 5.11-1 Timpul transmiterii pentru cinci noduri cu vibrații

O observație interesantă o prezintă timpul mediu de 20 minute realizat în cazul comunicării cu 2 retransmisii față de media de 17 minute la comunicarea cu pierderea pachetelor, unde nu se realizează nici o retransmisie.

5.12. Analiza vitezei de transmisie în cadrul RDEA

Placa	Viteza bytes/sec		
	Fara Retry	2 Retry	5 Retry
1	862,64	834,12	522,71
2	828,25	835,82	522,71
3	831,73	839,56	543,86
4	832,00	842,24	715,25
5	844,36	842,56	541,05

Tab. 5.12-1 Viteza transmiterii pentru cinci noduri cu vibrații

În Tab. 5.12-1 sunt reprezentate vitezele de transmisie a datelor utile în rețea. Utilizarea celor 5 retransmisii pentru pachetele pierdute, deși realizează o rată de succes ridicată este deficitară în cazul vitezei.

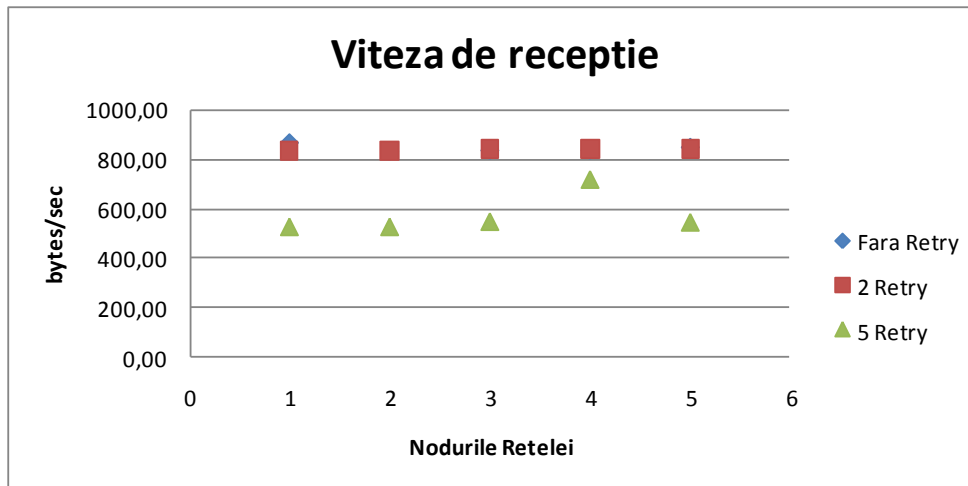


Fig. 5.12-1 Viteza transmiterii pentru cinci noduri cu vibrații

5.13. Decizii luate în cadrul proiectului pentru mărirea fiabilității

Analizele realizate în acest capitol furnizează valori optime pentru mărirea pachetului de date și dimensiunea buffer-ului intern al unui nod al rețelei. Multe experimente au fost repetate folosind noduri diferite sau scenarii de intrare sau de părăsire a RDEA, pentru a obține valori medii necesare analizelor.

S-a realizat un experiment în condiții defavorabile cu o rețea de 5 senzori care transmit un volum mare de date. Acești senzori pot măsura vibrații. În cadrul experimentului s-au realizat trei scenarii. În cadrul primului scenariu pachetele pierdute au fost neglijate, în al doilea scenariu pachetele pierdute au fost retransmise de 2 ori, iar în al treilea scenariu pachetele sunt retransmise de 5 ori. În urma analizelor din paragrafele 5.9 - 5.12 se pot desprinde mai multe concluzii. Astfel, dacă se dorește o RDEA care să aibă o rată de succes ridicată, de 95%-96%, ceea ce înseamnă că din datele transmise să se piardă cât mai puțin pachete se recomandă un număr ridicat de retransmisii a pachetelor pierdute, cum s-a realizat în scenariul 3. Dacă se acceptă o rată de transmisie moderată, de 93%, în schimb se urmărește transmisia rapidă a datelor atunci soluția optimă o constituie un RDEA care realizează 2 retransmisii. În funcție de tipul senzorilor se decide asupra rigurozității ratei de succes.

În afară de analizele realizate de mine, am utilizat caracteristicile puse la dispoziție de protocolul utilizat. În cazul comunicării radio, protocolul ZigBee [11] oferă câteva caracteristici de fiabilitate dintre care amintesc tratarea pachetelor corupte sau pierdute și funcționarea într-un mediu protejat.

ZigBee utilizează protocolul de transmisie CSMA/CA (Carrier Sense, Multiple Access with Collision Avoidance), care "ascultă canalul" înainte de a trimite datele. Înainte de transmiterea datelor, nodul care dorește să emită verifică dacă se transmit date. Dacă canalul este liber, adică nu se transmit date, nodul care dorește să transmită date trimite un mesaj de atenționare celorlalte stații să nu emită și

începe să transmită. Dacă altă stație transmite și canalul este ocupat, stația așteaptă o perioadă de timp aleatoriu înainte de a încerca să transmită din nou, apoi verifică din nou canalul și are posibilitatea să transmită sau să reia așteptarea un alt interval aleatoriu. Aplicând aceste perioade aleatorii scade posibilitatea apariției coliziunilor, deoarece stațiile încearcă să transmită date la intervale diferite.

Algoritmul CSMA/CA este folosit în LAN-uri radio și rezolvă problema razei de acțiune prezentată în Fig. 2.1-3.

Protocolul ZigBee folosește în implementare confirmările, pentru a se asigura dacă mesajul transmis a ajuns la destinație. La sosirea unui mesaj receptorul transmite emitentului o confirmare. Dacă confirmarea nu ajunge într-un anumit interval de timp, emitentul va retransmite mesajul inițial.

În topologia mesh (plasă) rețeaua reține drumul de la sursă la destinație. Dacă acest drum devine impracticabil prin căderea unuia sau mai multor noduri, rețeaua caută și descoperă dinamic alte trasee pentru transmiterea mesajului.

Coordonatorul C și router-ele R își pot configura perioada în care permit intrarea noilor noduri în rețea (join). Intrarea noilor noduri în rețea (join) se poate face oricând, sau pot să existe perioade când sunt permise intrările, în funcție de încărcarea și de traficul nodului respectiv .

Mai multe rețele ZigBee funcționează fără probleme în aceeași frecvență de bandă neinterferând una cu cealaltă. Mai mult, două rețele ZigBee pot funcționa în aceeași vecinătate, pe același canal, diferența fiind dată de identificatorul rețelei, PANId-ul. O altă măsură de fiabilitate oferită de protocolul ZigBee o constituie operarea fără perturbații în vecinătatea altor rețele radio bazate pe aceleași standarde, cum ar fi Wi-Fi and Bluetooth[54]. Din acest punct de vedere, ZigBee este o rețea sigură, prevenind și eliminând nu doar perturbațiile de la alte rețele apropiate, dar și atacurile și intruziunile ostile. Pentru aceasta folosește un algoritm de criptare pe 128 de biți bazat pe AES (Advanced Encryption Standard).

Chiar și în prezența unui număr mare de interferențe, protocolul ZigBee comunică eficient. Acest lucru poate fi explicat prin alegerea căilor de transmisie care să ocolească zonele cele mai afectate de interferențe[53]. Această situație este exemplificată în Fig. 5.13-1, unde calea de rutare aleasă „ocoloște” sursa de interferență.

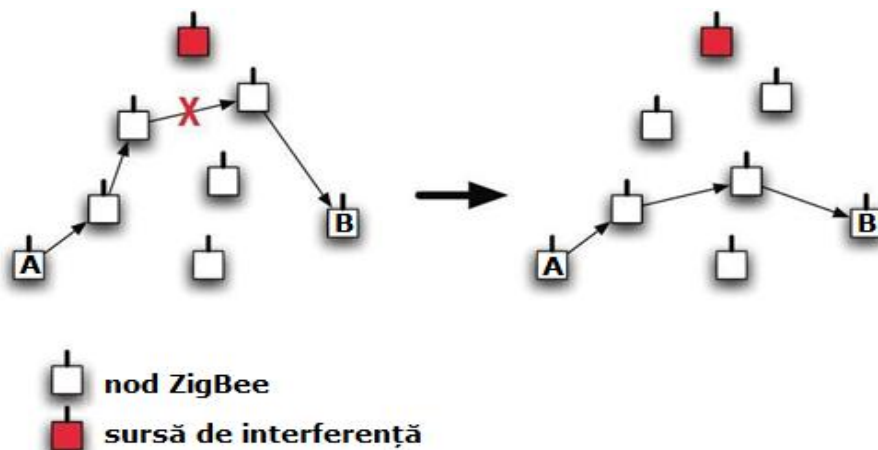


Fig. 5.13-1 Alegerea rutei în cazul vecinătății cu o sursă de interferențe [53]

O altă metodă de securitate o reprezintă eliminarea mesajelor expirate. În acest fel sunt excluse din rețea atacurile mesajelor retransmise. O astfel de situație o poate reprezenta înregistrarea unei comenzi de deschidere a unui garaj, pentru ca mai târziu comanda să fie retransmisă și să se încalce proprietatea. Această prevenție se realizează prin adăugarea unui contor mesajului transmis pentru a se verifica la receptor dacă acel mesaj a fost sau nu deja primit.

Un alt serviciu disponibil în ZigBee, oferit de IEEE802.15.4, este posibilitatea unui nod de a-și alege „partenerii de conversație”. Se creează o listă cu nodurile agreate, iar orice mesaj de la alte noduri nu este acceptat.

O propunere pentru mărirea performanței RDEA constă în folosirea mai multor canale de transmisie radio și crearea mai multor rețele. Voi realiza o analiză a topologiei rețelei și voi propune două topologii.

5.13.1. Analiza topologiei rețelei

În paragraful 5.7 am analizat cazul cel mai defavorabil al unei rețele formate din 5 DAD-uri, pe toate fiind montați senzori de vibrație.

În realitate nu toți senzorii eșantionează o cantitate așa de mare de date. În Tab. 5.4-1 sunt enumerați senzorii folosiți în RDEA și timpii de eșantionare la funcționare.

Știind acest lucru și bazându-mă pe proprietățile mesh-ului XBee ofer două soluții:

Prima soluție constă în crearea a 2 rețele, ilustrate în Fig. 5.13-2. RDEA conceput în dizertație se comportă exact cum a fost descris până acum, în paragraful 3.6, modificările fiind transparente utilizatorului.

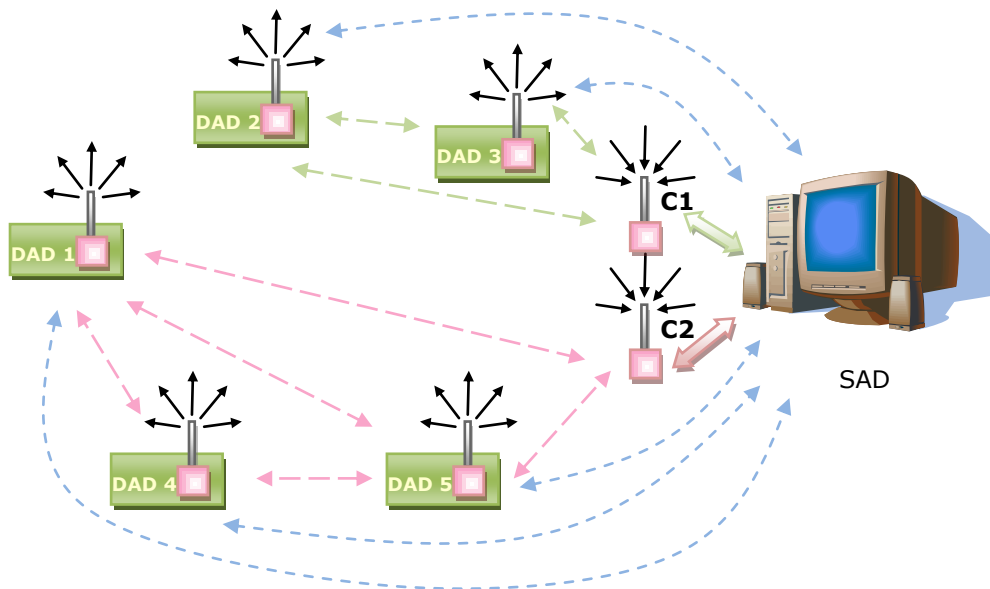


Fig. 5.13-2 RDEA formată din 2 mesh-uri XBee

146 5 - Evaluarea indicilor de performanță și fiabilitate pentru rețele de dispozitive de achiziția datelor aplicate la excavatoarele cu cupe

Dar, intern sunt realizate două rețele bazate pe tehnologia XBee. Acest lucru este posibil deoarece XBee permite comunicarea pe 16 canale. Cele 2 rețele comunică pe canale de frecvență diferite. În acest caz este necesară construirea unui coordonator pentru fiecare rețea XBee existentă. În cazul reprezentat în Fig. 5.13-2 există coordonatorul C1, care gestionează rețeau formată din nodurile DAD3 și DAD2. Comunicarea în a doua rețea cuprinde nodurile DAD1, DAD4 și DAD5, fiind gestionată de coordonatorul C2.

În acest fel se pot forma tot atâtea rețele independente câte canale de comunicație XBee există. Protocolul este identic, doar că se lucrează pe canale diferite.

O stație poate părăsi rețeaua sa, poate schimba canalul de comunicație și se poate alătura alteia.

O rețea creată nu pot fi dizolvată dinamic. Coordonatorii, nu pot să-și modifice opțiunea, să devină noduri obișnuite, iar un mesh-ul XBee nu permite existența mai multor coordonatori pe același canal. Pentru aceasta coordonatorul trebuie reconfigurat prin metode firmware. În acest caz dacă toate nodurile unui mesh XBee vor părăsi rețeau, coordonatorul va rămâne ținând rețeau activă și pregătită pentru orice nod care dorește să se alătore ei.

5.13.2. Prima propunere de topologie

Având în vedere caracteristicile prezentate până acum, propun soluția realizării a două rețele sau mai multe rețele: una formată din toți senzorii de vibrație, iar cealaltă formată din toți ceilalți senzori. Prima rețea are noduri puține dar traficul este mare.

Cea de-a doua rețea are noduri multe, dar traficul este aerisit.

Această soluție am ales-o deoarece trebuie ținut cont de sincronizarea senzorilor de vibrație. Acuratețea lor determină direcția de propagare a undei de perturbație. Din acest motiv am ales sa facă parte toți din aceeași rețea.

Este posibil ca cerința de DAD-uri din rețea să crească și în acest caz, în locul celei de-a doua rețele se pot forma mai multe.

În concluzie RDEA va fi formată din două sau mai multe rețele XBee, fiecare lucrând pe canale separate. În cadrul fiecărei rețele nodurile sunt sincronizate, dar rețelele între ele funcționează desincronizat.

5.13.3. A doua propunere de topologie

A doua soluție de topologie a pornit de la posibilitatea apariției unor congestii în prima rețea, care conține doar DAD-uri cu senzori de vibrație. Cum am spus anterior, în 5.13.2 am ales această soluție din necesitatea păstrării sincronizării nodurilor.

Am presupus că la crearea mai multor rețele XBee fiecare va funcționa pe un canal unic de comunicație, și fiecare va avea centru ei separat de sincronizare, astfel încât vor funcționa desincronizat între ele.

Știm că fiecare rețea are un centru de sincronizare și își sincronizează propriile noduri.

Propunerea ar fi găsirea unei metode de a sincroniza două sau mai multe rețele între ele. Acest lucru se poate realiza prin folosirea unor GPS-uri ca centre de sincronizare în fiecare rețea.

Știm că nodurile cu GPS furnizează un timp exact, și având aceste noduri ca puncte de sincronizare în rețele putem să admitem că rețelele vor funcționa sincronizat, atâta timp cât vor avea fiecare un nod cu GPS activ.

Acesta îmi permite schimbarea abordării. Având mai multe rețele sincronizate între ele pot distribui nodurile cu date stufoase uniform între rețele.

Mai mult, dacă inițial am considerat un număr maxim de 50 de DAD-uri în rețea, prin folosirea a 16 rețele paralele sincronizate numărul de dispozitive se poate mări la 200-300 de DAD-uri.

Pentru un RDEA cu 300 de DAD-uri, fiecare rețea va avea în medie 18-20 de noduri.

5.14. Concluzii

În vederea demonstrării funcționării corecte a configurației RDEA, am elaborat programe benchmark pentru evaluarea caracteristicilor de performanță și fiabilitate. Pentru analiza performanțelor și compararea diferitelor configurații am creat o aplicație constând dintr-un analizor al datelor colectate, care furnizează date atât nivelului rețea cât și la nivelul nod al rețelei. La nivelul rețea se urmărește volumul de date transmis în rețea, numărul de pachete de date pierdute, timpul de transmisie, precum și timeout-ul, respectiv numărul pachetelor netransmise.

Rezultatele sunt prezentate sub forma unor capturi cu vizualizarea parametrilor de senzor și a zonelor de perturbații.

În continuare am analizat RDEA-urile, obiectul tezei, sub aspectul dimensiunii atât prin prisma distanței dintre noduri, cât și prin prisma numărului acestora. Am investigat mărimea optimă a pachetului de date calculând pentru fiecare dimensiune indicatorii de performanță mai semnificativi, sens în care au fost elaborate grafice și tabele din care să rezulte proiectarea optimă a dimensiunii corespunzătoare atât a rețelelor, cât și a pachetelor de date.

Au fost elaborate experimente pentru determinarea variației timpului de transmisie în funcție de volumul de date transportat, din care rezultă recomandări pentru configurarea optimă a rețelelor de dispozitive de achiziția datelor. Au mai fost efectuate experimente cu referire la determinarea buffer-ului intern corespunzător unui nod de rețea. Partea experimentală mai cuprinde și evaluări cantitative, corespunzătoare unor scenarii semnificative de funcționare pentru metricile de fiabilitate a căror introducere selectivă a fost efectuată în capitolul 2.

Capitolul se încheie prin recomandări în cadrul unui rețetar de proiectare al rețelelor analizate în scopul obținerii unei fiabilități cât mai bune pe baza rezultatelor experimentale, caracterizate prin valorile metricilor obținute practic. În acest sens sunt propuse două topologii originale pentru rețele optimale destinate aplicațiilor industriale la excavatoarele cu cupe.

6. CONCLUZII

6.1. Aspecte generale legate de cercetarea efectuată

Scopul tezei constă în studierea unei RDEA prin prisma raportului performanță-fiabilitate. Punctul de plecare al cercetării l-a constituit definirea RDEA, al metodelor și al algoritmilor care să realizeze cerințele.

Dintre cerințe amintesc că RDEA este o rețea de dimensiuni mici spre medii, dar fiecare nod din rețea poate să producă un volum mare de date. Comunicarea în rețea poate să fie atât radio cât și serială. Datele primite de la nodurile rețelei fuzionează în analiză, din acest motiv a fost necesară găsirea unei metode specifice de sincronizare a nodurilor RDEA. Eșantionarea datelor se face la milisecundă, acuratețe care a fost luată în considerare la crearea algoritmilor de comunicare.

Implementarea a oferit prioritate comunicării radio, deoarece comunicarea serială fără zgomote lucrează la viteze mici, de 2400biți/sec.

Având în vedere particularitatea RDEA, am studiat soluții pentru îmbunătățirea performanței și a fiabilității comunicației.

În partea teoretică a tezei am urmărit crearea algoritmilor prin prisma performanței și a fiabilității.

Pentru demonstrarea rezultatelor teoretice, am creat o aplicație care să aibă ca obiect RDEA și ca scop validarea rezultatelor.

Cercetarea s-a desfășurat atât pentru găsirea unei soluții și materializarea unei idee de cercetare, cât și pentru aplicarea ei practică și găsirea de metode pentru îmbunătățirea funcționării rețelei.

În cercetare am analizat atât topologia rețelei cât și metricile de fiabilitate și performanță.

Partea teoretică a cercetării am validat-o prin implementarea unei rețele de dimensiuni reduse.

Pentru creșterea fiabilității și a performanțelor RDEA proiectate, după analiza teoretică, am creat aplicații de tipul benchmark pentru a compara diferite variante de funcționare.

În cadrul algoritmilor de comunicație, prezentați în paragraful 3.3, am prevăzut perioade de așteptarea pachetelor (timeout). Am analizat datele din

Tab. 5.7-1, în care într-o rețea cu 5 noduri cu vibrație, pachetele se transmit în medie la 50ms, și am stabilit perioada optimală de timeout la 80 ms.

În paragraful 5.5 am demonstrat că transmisia de la mai multe noduri în același timp în rețea oferă o performanță mai bună decât transmisia serială a fiecărui nod pe rând.

Pentru RDEA existent timpii cei mai buni de transmisie i-am obținut pentru pachetele de date care au lungimea de 48 de octeți. La transmiterea pachetelor am creat un header special cu ștampila de timp pentru primul pachet, după care în pachetele următoare sunt transmise doar eșantioane de date.

În urma testelor efectuate am demonstrat că aplicația lucrează la parametri maximi, limitarea transmiterii pachetelor fiind dată doar de întârzierea rețelei de 15ms. Rezultatele obținute de mine au fost prevăzute și de alți cercetători care au implementat rețele radio folosind protocolul ZigBee [45]. Ei au transmis pachete cu rata (baud rate) de 9600kbp, obținând timpi de 40ms la trimiterea pachetelor în

rețea. În articolul lor, ei estimează că la viteze mari se pot transmite pachete sub limita atinsă de ei de 40ms [45]. Eu am reușit acest lucru, folosind rata de transmisie (baud rate) de 19200kbps, transmisia fiind limitată doar de întârzierea rețelei.

Având în vedere cerințele de fiabilitate cerute de RDEA, ca o consecință, a decurs folosirea protocolului ZigBee, descris în paragraful 3.5.2. Alegerea protocolului a rezolvat problema razei de acțiune din cadrul rețelelor radio, a pierderii pachetelor, a pachetelor alterate, cu erori, a zgomotelor introduse de două rețele radio care funcționează împreună, sau a altor rețele radio sau dispozitive care se află în vecinătatea sistemului. Protocolul ZigBee oferă și servicii de securitate pentru atacurile nedorite îndreptate împotriva rețelelor radio.

În urma analizelor realizate în capitolul 5 soluția propusă de mine constă în crearea mai multor RDEA din DAD-urile montate pe excavator. Dacă există posibilitatea sincronizării lor globale, prin existența unor noduri cu GPS în fiecare rețea, atunci optimal ar fi existența a max 2-3 noduri cu trafic ridicat într-o rețea.

Dacă acest lucru nu este posibil atunci propun să se creeze rețele care cuprind senzorii care monitorizează același fenomen. Dacă într-o astfel de rețea numărul nodurilor cu trafic ridicat este mare, în experimente am făcut analize pe o rețea cu 5 noduri care transmite vibrații. Numărul recomandat de retransmisii al pachetelor este cuprins între 2 și 5.

O altă soluție, în cazul în care nu sunt disponibile multe GPS-uri, se realizează prin montarea nodului cu GPS pe SAD, iar sincronizarea tuturor rețelelor implicate în comunicație se face cu calculatorul personal.

Cercetarea mea de până acum a furnizat mai multe soluții de configurarea rețelelor de dispozitive pentru achiziția datelor cu aplicabilitate la excavatoarele cu cupe. O atenție specială am acordat-o părții radio de transmitere, având în vedere rata de transmisie superioară celei din comunicația cablată serială. La comunicarea serială am folosit metoda „time sharing”. O direcție viitoare de cercetare o poate constitui realizarea unui protocol serial special construit, în care să se regăsească și să se implementeze cerințele de performanță și fiabilitate implementate pentru rețelele radio: retransmisia datelor, datorată alterării sau pierderii pachetelor, tehnici pentru tratarea apariției zgomotelor, analize privind mărimea pachetului de date, care în acest caz nu este limitată superior la 72 de octeți ca în cazul pachetelor API. Se pot realiza analize pentru determinarea numărului optim de retransmisii din rețea.

O altă direcție de cercetare o constituie dezvoltarea caracteristicilor adaptive ale RDEA spre metode de predicție ale comportamentului viitor al sistemului. O posibilă implementare poate folosi rețele neuronale, care, bazate pe datele citite de la senzori prin metode de învățare furnizează tipare, care permit previziunea funcționării viitoare a sistemului.

Configurațiile rețelelor radio prezentate în această dizertație transmit datele unui calculator personal. Acolo informațiile se stochează pe disc, într-o bază de date, după care ulterior are loc procesarea lor. Pornind de aici se pot dezvolta două direcții viitoare de cercetare.

Prima direcție constă în valorificarea memoriei și a capacității de calcul a fiecărui nod din RDEA, astfel încât fiecare DAD să conțină un motor de procesare și stocare a unei cantități de date, iar mai departe, în rețea să se transmită doar rezultate sau informații importante, procesate deja. Această dezvoltare implică

folosirea unor resurse suplimentare la fiecare DAD. În momentul de față, la nivelul fiecărui nod se realizează o analiză preliminară a datelor în urma căreia se stabilește rata de citire a eşantioanelor și se mediază valorile mai multor eşantioane, realizându-se distincția între date corecte și perturbatii. Voi extinde partea de analiză care se realizează la nivelului fiecărui nod din rețea.

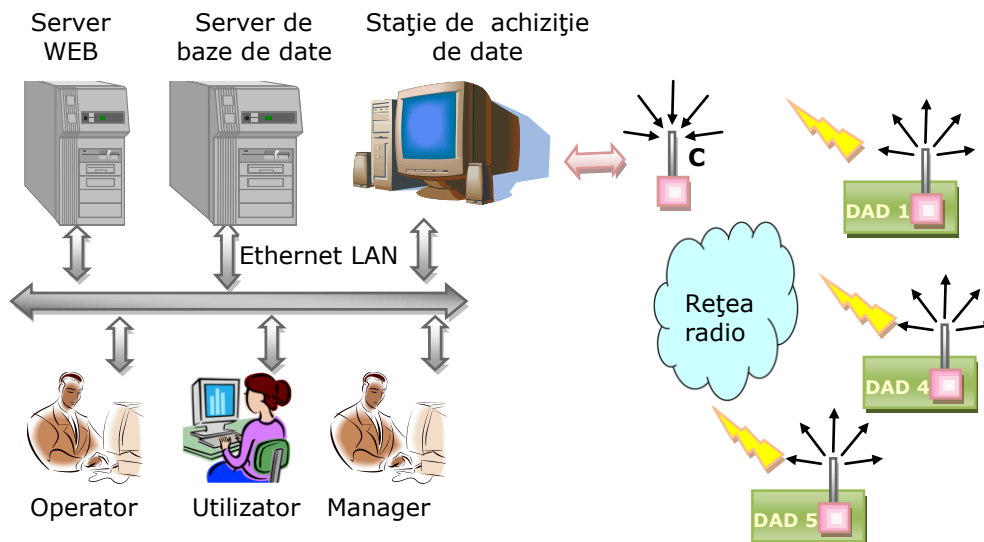


Fig. 6.1-1 Comunicarea dintre baza de date și aplicația web

A doua direcție de cercetare viitoare constă în crearea unor mecanisme de informare online asupra anumitor parametri, și vizualizarea lor pe o pagină de Internet. Se va crea o aplicație web care va consulta baza de date descrisă în teză și va actualiza un site dedicat excavatorului, furnizând informații cum ar fi poziția excavatorului în carieră, viteza de lucru sau cantitatea de cărbune extrasă. Serverul bazei de date existente și serverul de web pot fi găzduite pe același computer sau pe calculatoare diferite, cum sunt reprezentate în Fig. 6.1-1. În acest caz, calculatorul care găzduiește aplicația web (hosting) trebuie să aibă acces la baza de date a excavatorului. Comunicarea realizată în acest caz este ilustrată în Fig. 6.1-1. Se pot crea și metode de siguranță sporită. Pentru a proteja baza de date a excavatorului pe calculatorul care găzduiește aplicația web se face o copie a bazei de date și în acest fel un atac extern va distruge copia și nu baza originală. Această soluție trebuie să aibă în vedere actualizarea periodică a datelor.

În septembrie 2008 firma Digi a lansat pe piață protocolul DigiMesh [55], o alternativă cu facilități puternice a protocolului ZigBee. Voi descrie câteva avantaje ale folosirii lui.

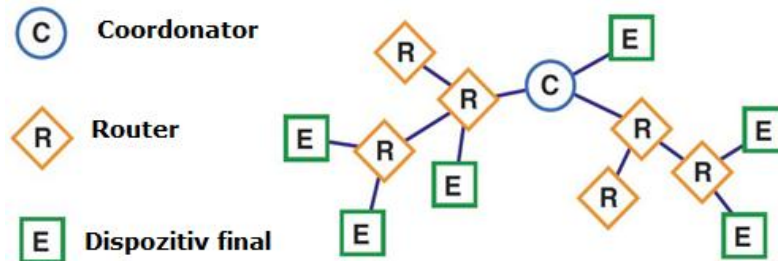


Fig. 6.1-2 Topologia ZigBee clasică

În rețeaua de tip mesh clasică, ilustrată în Fig. 6.1-2 există trei tipuri diferite de noduri: coordonator, router și nod terminal, fiecare având caracteristicile lui.

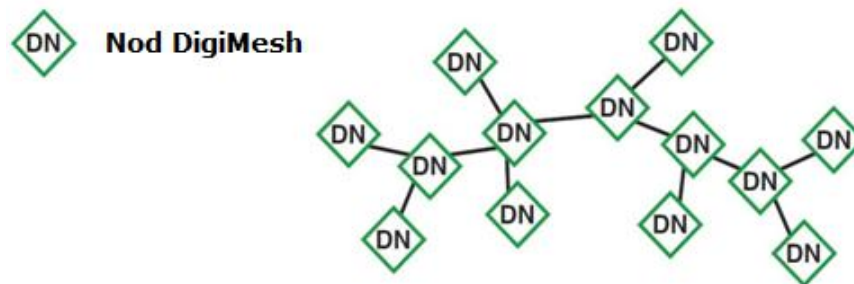


Fig. 6.1-3 Noua topologie DigiMesh

Noua topologie DigiMesh, prezentată în Fig. 6.1-3, propune toate nodurile de același fel. De aici decurg mai multe avantaje, voi prezenta pe scurt doar două. Până acum defectarea coordonatorului C determina colapsul rețelei radio. Acum acest lucru nu se mai întâmplă. Rețeaua funcționează fără a avea nevoie de un coordonator. Adresarea se face folosind doar adresele pe 64 de biți, eliminând situațiile de duplicat de adresă de rețea pe 16 biți relatate în teză, în paragraful 3.13.

Protocolul DigiMesh reprezintă o îmbunătățire a protocolului ZigBee și folosirea lui pe viitor consider că reprezintă o decizie bună și necesară.

Metoda de comunicație și sistemul realizate și prezentate în această lucrare se pot aplica oricărei rețele de dispozitive care monitorizează și transmit date: pornind de la păsări migratoare, senzori de ploaie, vânt, umiditate folosiți în agricultură până la dispozitive mobile de monitorizarea și avertizare casnice, medicale sau industriale.

Având în vedere larga aplicabilitate a acestei soluții am hotărât să o protejăm și să o valorificăm printr-un brevet de invenție [87].

Trebuie să adaug că în mod uzual în cadrul aplicațiilor nu sunt necesare cerințe atât de stricte ca cele prezentate și exploatate în această teză. Culegerea datelor nu trebuie să meargă până la nivel de milisecundă cum s-a practicat în această lucrare. Citirea datelor la nivel de secundă, sau chiar minut sunt cerințe

suficiente pentru majoritatea aplicațiilor. Astfel, în majoritatea cazurilor voi furniza o variantă simplificată a ceea ce am prezentat aici.

6.2. Contribuții personale

Prezenta teză aparține categoriei de lucrări cu un profund caracter aplicativ. Plecând de la o problemă practică concretă constând în funcționarea frecvent necorespunzătoare a excavatoarelor cu cupe, obiectiv industrial de mare importanță, în industria mineritului și nu numai, în lucrare s-a urmărit ca pe baza criticii soluțiilor tehnice existente să se justifice o soluție complet nouă, fundamentată printr-o propunere de invenție.

Soluția se bazează pe aplicarea unei rețele de dispozitive de achiziția datelor a cărei configurare optimală prin prisma performanței și a fiabilității este cercetată în lucrare.

Contribuțiile pot fi grupate în următoarele categorii:

1. Contribuții de factură teoretică
 - 1.1 Analiza vibrațiilor mecanice care apar în funcționarea excavatoarelor cu cupe și modul în care pot fi studiate, sens în care am participat în colective de cercetare mixte împreună cu specialiști în mecanică, fiind coautoare la două lucrări științifice publicate.
 - 1.2 Am efectuat analize comparative între topologii de rețele de calculatoare.
 - 1.3 Am efectuat analize comparative între protocoale de comunicație în rețea.
 - 1.4 Am efectuat analize comparative între protocoale de sincronizare în timp a funcționării rețelelor.
 - 1.5 Selecția metricilor de performanță și fiabilitate importante prin prisma aplicării rețelei de dispozitive de achiziție a datelor pentru excavatoare cu cupe a constituit altă direcție a cercetării.
2. Contribuții de factură aplicativă
 - 2.1 Propunerea justificată pe baza unui brevet de invenție a unei soluții de urmărire a vibrațiilor prin intermediul unei rețele de așa zise dispozitive de achiziția datelor.
 - 2.2 Investigarea configurării unei structuri hibride implicând comunicarea radio între nodurile rețelei și comunicarea de tip cablat serială în cadrul unui nod al rețelei.
 - 2.3 Elaborarea a trei algoritmi de sincronizare în timp a nodurilor rețelei.
 - 2.4 Elaborarea unui algoritm de comunicare între nodurile rețelei.
 - 2.5 Elaborarea unor aplicații benchmark de evaluare a metricilor de performanță și fiabilitate.
 - 2.6 Elaborarea structurii dispozitivelor de achiziția datelor în calitate de noduri eterogene ale rețelei de DAD-uri.
3. Contribuții de factură experimentală
 - 3.1 Proiectarea optimală a elementelor de structură hardware bazate pe componente moderne cum ar fi microcontroller-ul ARM sau Watchdog Timer-ul.
 - 3.2 Proiectarea optimală croită pe aplicație a programării microcontroller-ului ARM.
 - 3.3 Proiectarea optimală a programelor de aplicație.

- 3.4 Elaborarea de scenarii de experimentare edificatoare pentru metricile de performanță.
- 3.5 Elaborarea de scenarii de experimentare edificatoare pentru metricile de fiabilitate.
- 3.6 Elaborarea de scenarii de topologii originale pentru configurarea rețelelor.

Bibliografie

1. Tanenbaum A., - „Rețele de calculatoare”, ediția a IV-a 2004, Byblos
2. Martin Neil, Norman Fenton, „*Software Metrics: Roadmap*”, ICSE 2000, 8 June, RADAR (Risk Assessment & Decision Analysis Research)
3. project IST-2001-34820- Adaptive Real-Time Systems for Quality of Service Management (<http://www.artist-embedded.org/Roadmaps>)
4. Raghavendra C.S., Sivalingam Krishna M., Taieb Znati, „Wireless sensor networks”, Springer 2004
5. Paxson, V., Almes, G., Mahdavi, J. and M. Mathis, "Framework for IP Performance Metrics", RFC 2330, February 1998.
6. Nalini L. Joseph, Uma G.V., „Reliability Based Routing in Wireless Sensor Networks”, IJCSNS International Journal of Computer Science and Network Security, VOL.6 No.12, December 2006
7. XBee ZNet 2.5/ XBee-Pro ZNet 2.5 OEM RF Modules, Product manual v1.x.4x, 2008
8. D.C. Schmidt, „Adaptive and Reflective Middleware for Distributed Real-Time and Embedded Systems”, in Proceedings of the First International Workshop, EMSOFT2001, Tahoe City, USA, pages 361-372, October 2001
9. Draves R., Padhye J., Zill B., „Comparison of Routing Metrics for Static Multi-Hop Wireless Networks”, ACM SIGCOMM (Special Interest Group on Data Communication) 2004
10. Jacquet J.-P.; Abran A., „From Software Metrics to Software Measurement Methods: A Process Model”, in Third International Symposium and Forum on Software Engineering Standards (ISESS'97), Walnut Creek, CA, IEEE Computer Society, 1997, pp. 128-135
11. http://www.jennic.com/elearning/zigbee/files/content_frame.htm
12. ***http://www.st.com/stonline/products/families/sensors/motion_sensors/accelerometers/related_info/lis302x.htm
13. ***iARM-2103 User Manual
14. ***UM10161 LPC2101/02/03 User Manual
15. ***<http://www.keil.com/dd/chip/3974.htm>
16. Couto P., Valadas R., „Performance of Packet Reservation Frame-based MAC Protocols for Wireless ATM Networks”, CSNDSP 2000 (Communication Systems, Networks And Digital Signal Processing)
17. Robinson J., Swaminathan R., Knightly E. W., „Assessment of Urban-Scale Wireless Networks with a Small Number of Measurements”, MobiCom September 2008
18. De Couto D.S.J., Aguayo D., Bicket J., Morris R., „A High Throughput Path Metric for MultiHop Wireless Routing”, MobiCom, September 2008
19. Maroti M., Kusy B., Simon G., Ledeczi A., „The Flooding Time Synchronization Protocol”, Vanderbilt University, 2004
20. Sivrikaya F., Yener B., "Time Synchronization in Sensor Networks: A Survey", IEEE Network Magazine - special issue on "Ad Hoc

- Networking: Data Communications & Topology Control", Vol. 18, Issue 4, pp. 45--50, July/August 2004
21. Saumitra M. Das, Himabindu P., Papagiannaki K. , Hu Y. C., "Studying Wireless Routing Link Metric Dynamics", IMC'07 (Internet Measurement Conference), October 2007
 22. Radunović B., Le Boudec J.-Y., "Rate Performance Objectives of Multi-hop Wireless Networks", IEEE INFOCOM 2004
 23. Cena G., Bertolotti I. C., Valenzano A., Zunino C., "Evaluation of Response Times in Industrial WLANs", IEEE Transactions on Industrial Informatics, vol. 3, no. 3, August 2007
 24. Bonivento A., Fischione C., Necchi L., Pianegiani F., Sangiovanni-Vincentelli A., "System Level Design for Clustered Wireless Sensor Networks, IEEE Transactions on Industrial Informatics", vol. 3, no. 3, August 2007
 25. Nalini L. J., Uma G.V., "Reliability Based Routing in Wireless Sensor Networks", IJCSNS International Journal of Computer Science and Network Security, VOL.6 No.12, December 2006
 26. Chiang M. W., Zilic Z., Radecka K., Chenard J.-S., "Architectures of Increased Availability Wireless Sensor Network Nodes", ITC (International Test Conference) 2004
 27. Joseph P. Loyall, "Emerging Trends in Adaptive Middleware and Its Application to Distributed Real-Time Embedded Systems", EMSOFT2003, LNCS 2855, pages 20-34, 2003
 28. Ganeriwal S., Kumar R., Srivastava M. B., "Timing-sync protocol for sensor networks," ACM Conference on Embedded Networked Sensor Systems (*SENSYS 2003*)
 29. PalChaudhuri S., Kumar Saha A., Johnson David B., "Adaptive Clock Synchronization in Sensor Networks", ipsn2004
 30. Li Q., Rus D., "Global Clock Synchronization in Sensor Networks", IEEE Infocom, Hong Kong, March 2004
 31. Solis R., Borkar V., Kumar P.R., "A New Distributed Time Synchronization Protocol for Multihop Wireless Networks." Proceedings of the 45th IEEE Conference on Decision and Control, pp. 2734--2739, San Diego, Dec. 13-15, 2006.
 32. Weber G., Brusilovsky P., "ELM-ART: An Adaptive Versatile System for Web-based Instruction", International Journal of Artificial Intelligence in Education (2001), 12, 351-384
 33. Sharma K., „Designing Knowledge Based Systems as Complex Adaptive Systems”, to Appear in the Proceedings of The First Conference on Artificial General Intelligence, Memphis, Tennessee, March 2008
 34. Elson J., Estrin D., "Time Synchronization for Wireless Sensor Networks", Proceedings of the 2001 International Parallel and Distributed Processing Symposium (IPDPS), Workshop on Parallel and Distributed Computing Issues in Wireless and Mobile Computing, San Francisco, California, USA. April 2001.
 35. Elson J., Estrin D., "Fine-Grained Network Time Synchronization using Reference Broadcasts", In Proceedings of the Fifth Symposium on Operating Systems Design and Implementation (OSDI 2002), Boston, MA. December 2002. UCLA Technical Report 020008. (Supercedes earlier revisions from May and February 2002.).

36. Greunen J.V., Rabaey J., "Lightweight Time Synchronization for Sensor Networks", *Proceedings of the 2nd ACM International Conference on Wireless Sensor Networks and Applications(WSNA)*, San Diego, CA, September 2003.
37. Atomichron: "The Atomic Clock from Concept to Commercial" Product, FORMAN P., *Proceedings of the IEEE* Publication Date: July 1985 .
38. Nelson R. A., McCarthy D. D., Malys S., Levine J., Guinot B., Fliegel H. F., Beard R. L., Bartholomew T. R., "The leap second: its history and possible future", *Metrologia*, Volume 38, Number 6, 31 January 2002 , pp. 509-529(21)
39. <http://www.atomicclockrugbymf.co.uk/Atomic-clocks.htm>
40. <http://www.botanicalls.com/schematic.html>
41. <http://www.thelivingcity.net/>
42. <http://www.diykyoto.com/wattson/how-wattson-works>
43. <http://youtube.com/watch?v=kcYoMOSFIss>
44. http://youtube.com/watch?v=Tv_jRydPhjk
45. Desmet A., Naghdy F., Ros M., " Realising WACNet Through A Zigbee-Based Architecture", *IEEE Sensors 2007*
46. "Frequency Hopping vs. Direct Sequence Spread SpectrumTechniques", white paper, RayLink
47. "Frequency Hopping vs. Direct Sequence Spread SpectrumTechniques", white paper, RayLink
48. "Frequency Hopping Spread Spectrum (FHSS) vs. Direct Sequence Spread Spectrum (DSSS) In Broadband Wireless Access (BWA) and Wireless LAN (WLAN)", SCHWARTZ S. M., ver.7 - December 31, 2001
49. Carl A., "A Comparison of Frequency Hopping and Direct Sequence Spread Spectrum Modulation for IEEE 802.11 Applications at 2.4 GHz" Harris Semiconductor, Palm Bay, Florida Nov. 1997.
50. RS485_white_paper.pdf
51. Dantu K., Rahimi M., Shah H., Babel S., Dhariwal A., Sukhatme G.S., " Robomote: enabling mobility in sensor networks", *Information Processing in Sensor Networks*, 2005. IPSN 2005. Fourth International Symposium on Volume , Issue , 15 April 2005 Page(s): 404 - 409
52. <http://grupos.unican.es/dyvci/ruizrg/postscript/Componentes/SLUA051.pdf>
53. ZigBee and Wireless Frequency Coexistence, June 2007
54. An Paper on ZigBee, EDA 390 - Computer Communication and Distributed Systems, May 2006
55. Wireless Mesh Networking ZigBee versus DigiMesh, 2008
56. <http://www.merl.com/projects/zigbeenwk/>
57. Jung S.-Y., Park D.-J., "Design of preamble signal for synchronization with UWB non-coherent energy detection receiver, Ultra-Wideband", 2005. ICU 2005. 2005 IEEE International Conference on Volume , Issue , 5-8 Sept. 2005 Page(s): 464 - 468
58. Wang Y., Martonosi M., Peh L., "A New Scheme on Link Quality Prediction and its Applications to MetricBased Routing", *SenSys'05*, November 2-4, 2005, San Diego, California, USA. ACM 159593054X/05/0011

59. Engstrom J. Collin, Gray C., Nelakuditi S., "Clear Channel Assessment in Wireless Sensor Networks", National Science Foundation award #0649105, 2007
60. <http://www.makingsenseofspace.com/arduino-wireless-workshop/802154-zigbee.html>
61. <http://www.adhocelectronics.com/XBee-PRO-OEM-RF-Modules-2-4GHz-60mW-802-15-4>
62. <http://francis.courtois.free.fr/jc1/serial/Basics/BitFormat.html>
63. Bowles J. B., Stephens L. M., Huhns M. N., Tobaben R. W., Yekollu S., Kolpuru H., "Autonomous Networked Tactical Sentries," in *Proceedings Compiler and Architecture Support for Embedded Systems (CASES)*, Washington, DC, October 2, 1999
64. Sztipanovits J., Karsai G., "Embedded Software: Challenges and Opportunities" ,EMSOFT 2001 (Embedded Software), First International Workshop
65. Alexander P., Kong C., "Heterogeneous Modeling Support for Embedded System Design", EMSOFT 2001 (Embedded Software), First International Workshop
66. Benveniste A., "Some Synchronization Issues When Designing Embedded Systems from Components", EMSOFT 2001 (Embedded Software), First International Workshop
67. Alur R., Esposito J., Fierro R., Hur Y., Ivančić F., Kumar V., Lee I., Mishra P., Pappas G., Sokolsky O., "Hierarchical Hybrid Modeling of Embedded System", EMSOFT 2001 (Embedded Software), First International Workshop
68. Caspi P., "Embedded Control: From Asynchrony to Synchrony and Back", EMSOFT 2001 (Embedded Software), First International Workshop
69. Donahue S.M., Hampton M.P., Deters M., Nye J.M., Cytron R., Kavi K.M., "Storage Allocation for Real-Time", Embedded Systems, EMSOFT 2001 (Embedded Software), First International Workshop
70. Qinghua H., "Electromechanically controlled excavator and method for controlling the electromechanically controlled excavator", EP1835079, 2007-09-19
71. Unnat P., Sabri C., Richard G. I., Sohel A., Chenyao C., "System for controlling working machine has controller that receives switch, lever, sensor signals, determines tool position, controls tool to automatically carry out operating function", DE10028606, 2001-02-22
72. Lincoln D., "Method Of Measuring Vibration On A Device", US2008000301, 2008-01-03
73. Richard P.W. , Thomas N.E., " Belt pack accessory for machine condition monitoring", US2007200722, 2007-08-30
74. Derisaduromo J. I. L., "Monitor and treatment equipment for wind turbine and predictable maintenance system for wind power plant", CN1882892, 2006-12-20
75. Wirth N., "Embedded Systems and Real-Time Programming", EMSOFT 2001 (Embedded Software), First International Workshop
76. Simsek T., Varaiya P., "Design of Autonomous, Distributed Systems", EMSOFT 2001 (Embedded Software), First International Workshop

77. Meguerdichian S., Koushanfar F., Qu G., Potkonjak M., "Exposure In Wireless Ad-Hoc Sensor Networks", MOBICOM01
78. Batalin M.A , Rahimi M., Yu Y. ,Liu D., Kansal A., Sukhatme G.S., KaiserW.J., HansenM., Pottie G.J, Srivastava M., EstrinD., , "Call and Response: Experiments in Sampling the Environment", In *ACM SenSys'04* pp. 25-38, November 3-5, 2004, Baltimore, Maryland
79. Valente A., Morais R., Serodio C., Mestre P., Pinto S., Cabral M., "A ZigBee Sensor Element for Distributed Monitoring of Soil Parameters in Environmental Monitoring", *Sensors*, 2007 IEEE Volume , Issue , 28-31 Oct. 2007 Page(s):135 – 138
80. SHANMUGANTHAN S., GHOBAKHLOU A., SALLIS Ph., "Sensors for modeling the effects of climate change on grapevine growth and wine quality", OIV 2007 (International Organisation of Vine and Wine)
81. Henzinger Th.A., Horowitz B., Kirsch Ch. M., "Giotto: A Time-Triggered Language for Embedded Programming", EMSOFT 2001 (Embedded Software), First International Workshop
82. Koo T.-J., Liebman J., Ma C., Sastry Sh., "Hierarchical Approach for Design of Multi-vehicle Multi-modal Embedded Software", EMSOFT 2001 (Embedded Software), First International Workshop
83. Schmidt D. C., "Adaptive and Reflective Middleware for Distributed Real-Time and Embedded Systems", EMSOFT 2001 (Embedded Software), First International Workshop
84. Sifakis J., "Modeling Real-Time Systems-Challenges and Work Directions", EMSOFT 2001 (Embedded Software), First International Workshop
85. Cioara T.G., Nicolae I., Cireş I., Timar L., Image Processing Method in Low Frequency Modal Analysis of the Large Structures, International Conference of Engineering Dynamics, Portugalia, april 2007, www.dem.ist.utl.pt/~iced2007/pages/Book_of_Abstracts.pdf
86. Cioara T.G., Cireş I., Nicolae I., Dumitru C., Tîrdea A., Timar L., Dynamic Study Of A Special Crane Serving A Power Plant Tall Chimney, Society for Experimental Mechanics 2007, IMAC-XXV
87. Cerere de brevet de invenție cu titlul "Sistem și metodă pentru determinarea regimului de exploatare al excavatorului cu cupe", înregistrat la OSIM cu nr. A/00701 din data 11.09.2008

Anexa 1

Detaliez formatul cadrelor API folosite

Raspuns la comanda ATND

Frame

0x7E 0x00 0x05 0x08 0x01 0x4E 0x44 0x64

Where

0x00 05 = length

0x09 = AT Response Command API frame type

0x01 = Frame ID (set to non-zero value)

0x4E44 = AT command ('ND')

0x00 = OK

0x63 = Checksum

The checksum is calculated as $[0xFF - (0x09 + 0x01 + 0x4E + 0x44 + 0x00)]$

Transmit Status

Frame

0x7E 0x00 0x07 0x8B 0x01 0x4E 0x44 0x00 0x00 0x00 0xE1

Where

0x00 07 = length

0x8B = API frame type - Transmit Status

0x01 = Frame ID (set to non-zero value) - starea tramisie frame-ului cu ID

0x01

0x4E44 = Adresa pe 16 bit a nodului destinatie

0x00 = Numarul de retry-uri

0x00 = Starea tramsisiei - Success

0x00 = Stare descoperire rută - No Discovery over head

0xE1 = Checksum

The checksum is calculated as $[0xFF - (0x8B + 0x01 + 0x4E + 0x44 + 0x00 + 0x00 + 0x00)]$

Receive Packet

Primirea unui packet de date cu 5 sample-uri de la placa 2 tip senzor 1 (vibratie)

Frame

0x7E 0x00 0x18 0x90 0x10 0x01 0x00 0x13 0xA2 0x00 0x40 0x0A 0x4E
0x44 0x01 0x01 0x02 0x01 0x45 0x0A 0x05 0xE2 0x3B 0x14 0xE2 0x3B 0x12
0xE2 0x3B 0x09 0xE3 0x30 0x10 0xD2 0x3A 0x12 0xE1

161

Where

0x00 18 = length
0x90 = API frame type - Receive packet
0x0013A200 40014011 = Adresa pe 64 bit a nodului sursa
0x4E44 = Adresa pe 16 bit a nodului sursa
0x01 = packet acknowledged
0x010201450A05E23B14 E23B12 E23B09 E33010 D23A12 = datele

unde

0x01 - FrameType - Data frame
0x02 - Board ID
0x01 - Sensor Type
0x45 - Time Stamp - ss.sss
0x0A - Sampling Rate - 10 ms
0x05 - Sample Number in packet - 5
0xE23B14 0xE23B12 0xE23B09 0xE33010 0xD23A12 - 5 samples of

three bytes

0x67 = Checksum

The checksum is calculated as [0xFF - SUM(0x90 0x10 0x01 0x00 0x13
0xA2 0x00 0x40 0x0A 0x4E 0x44 0x01 0x01 0x02 0x01 0x45 0x0A 0x05 0xE2 0x3B
0x14 0xE2 0x3B 0x12 0xE2 0x3B 0x09 0xE3 0x30 0x10 0xD2 0x3A 0x12)]

ANEXA 2

IMAGE PROCESSING METHOD IN LOW FREQUENCY MODAL ANALYSIS OF THE LARGE STRUCTURES

Cioara, T.G., Nicolae, I., Cireş, I., Timar, L.

University Politehnica of Timișoara

Bv. Mihai Viteazul nr. 1 RO-300222 TIMIȘOARA, ROMANIA

E-mail cioara@mec.utt.ro

SUMMARY: Vibration measurement of the large structures of which modal components are located in the very low frequency range 0.3 to 3 Hz it is a difficult task. Using a seismic accelerometer, piezoresistive or capacitive type, is difficult to separate the two motion components, translation and rotation, presented in the recorded signal. The laser measurement requires special expensive instrumentation which are not justified for sporadic need. The paper presents a simple method which allows ease to obtain the absolute low motions of a structure using a digital image processing technology. On the structure is placed a panel with a grid of points and with a customary cheap camcorder placed in a fixed position is recorded the motion of the focalized grid. Then by image processing frame by frame it is obtained the frame (absolute) position of the grid points and actual lows of the structure in the area focused. Using more one camcorder it can be obtain the real motion of all machinery structure. The method was successful verified in the laboratory on wind turbine blade and in the field on a surface mining huge structure machinery.

KEYWORDS: excavator, structure vibration, image processing.

1 INTRODUCTION

The practically of interest for the large machines structures are the first few, two or three modal components, which have significant influence in the equilibrium stability, as in case of a huge mining excavator. Basically, a excavator structure is consisting from; a carriage platform **0** (on caterpillar, Figure 1), a tripod **1** which supports, by a thrust ball bearing **2**, the machinery rotating upper structure. The rotating upper structure, around vertical axis **OZ**, is consisting of a tower **3**, supporting the arm **4** (with bucket wheel **4'**), by the pulley tackle **5** and bucket chain **8**, and the balancing arm **6**, which is supported by the tyrants **7**. The arm **4** yaw motion is carried by a pulley tackle **9**.

By numerous experimental studies of the Vibration Testing and Research Laboratory (VTRL) we observed that the general motions of the excavator structure in its vertical plane can be practically approximate by three motion lows: $\varphi_0(t)$, the rotation of the rotating upper structure around a horizontal axis 0-0, in the plane of trust bearing, which is one of the most stressed and deformable part of the excavator; $\varphi_1(t)$, the yaw motion of the arm around the horizontal axis 1-1, and $\varphi_2(t)$ the rocking motion of the bucket chain 8 around the horizontal axis 2-2.[1],[2]

The both rotation axes 1-1 and 2-2 are cinematically well defined, so the instant positions of arm 4 and bucket chain 8, at a moment t , given by the angles $\varphi_1(t)$ and $\varphi_2(t)$, depend of the pulley tackle controlled cinematic position and of the elastic deformation of its cables, and of tyrants 7elastic deformations.

With this assumptions the dynamic model of the excavator structure in the vertical plane motion can be written by Lagrange equations

$$\frac{d}{dt} \left(\frac{\partial E_c}{\partial \dot{\varphi}} \right) - \frac{\partial E_c}{\partial \varphi} + \frac{\partial E_p}{\partial \varphi} + \frac{\partial E_{ed}}{\partial \varphi} + \frac{\partial E_{ds}}{\partial \varphi} = \dot{Q} \quad (1)$$

where

$$\dot{\varphi} = \dot{\varphi}_0 + \dot{\varphi}_1 + \dot{\varphi}_2 \quad (2)$$

The equations (1) were developed and numerically resolved for an excavator structure in [3] and [4], here being presented only a short overview of the developed particularities

So, E_c the kinetic energy which, in case of mining machine as variable structure, depends of the both, velocity, explicit by the column vector $\dot{\varphi}$, and by the position coordinates column vector φ , implicit by sinus and cosines of $\varphi_0(t)$, $\varphi_1(t)$ and $\varphi_2(t)$.

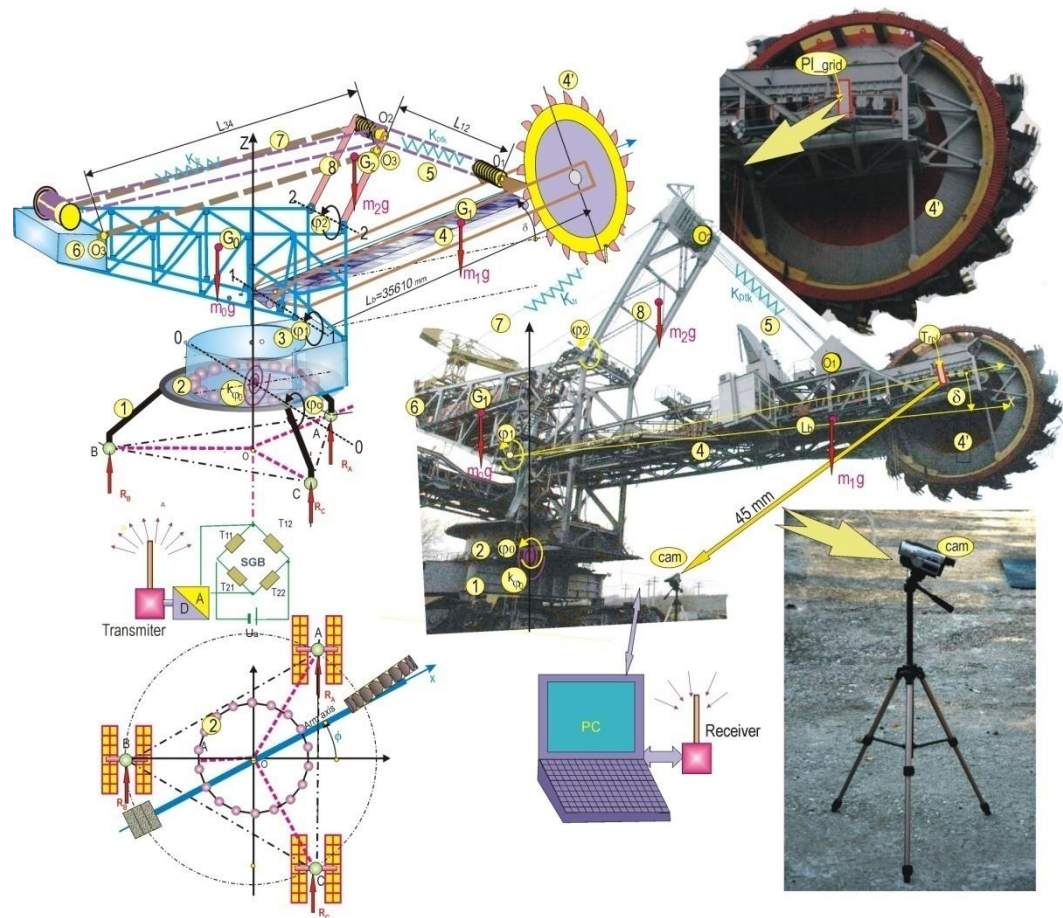


Figure 1 Illustration excavator mechanisms and experimental setup

The potential energy E_p depends of the vertical positions z_0 , z_1 and z_2 of the mass centers G_0 , G_1 and G_2

$$E_p = -m_0 z_0(\varphi_0) - m_2 z_2(\varphi_0, \varphi_1) - m_3 z_3(\varphi_0, \varphi_1, \varphi_2) \quad (3)$$

the angular coordinates $\varphi_0(t)$, and $\varphi_2(t)$ including elastic deformations only, and $\varphi_1(t)$ both elastic deformation and cinematic component controlled by pulley tackle mechanism.

Considering, the elastic deformations in the area of thrust bearing 2, of angular stiffness k_{φ_0} , the deformations of pulley tackle 5 wiring cables, of stiffness k_{ptk} , and the deformations of the tyrants 7, of stiffness k_{tr} , the deformation energy is a summation of three components

$$E_{ed} = \frac{1}{2} k_{\varphi_0} \varphi_0^2 + \frac{1}{2} k_{ptk} \left(\overline{O_1 O_2(t)} - \overline{O_1 O_2(0)} \right)^2 + \frac{1}{2} k_{tr} \left(\overline{O_3 O_4(t)} - \overline{O_3 O_4(0)} \right)^2 \quad (4)$$

where $\overline{O_1 O_2(0)}$ is the length of undeformed unloaded wiring cables and $\overline{O_3 O_4(0)}$ the undeformed unloaded length of the tyrants.

The distance $\overline{O_1 O_2(t)}$ in deformed variable in time state is of the form

$$\overline{O_1 O_2(t)} = \sqrt{\left(x_1 - x_2 \right)^2 + \left(z_1 - z_2 \right)^2} \quad (5)$$

the coordinates x_1 and z_1 and x_2 and z_2 of the end points O_1 and O_2 of the pulley tackle cable wiring, depending, by geometrical conditions, of φ_1 and φ_2 .

For a given position $\varphi_{10} = \delta$ at which the operation of yaw motion arm, controlled by pulley tackle, is stopped, on the whole excavator occurring free decay vibration motions (see Figures 4,5). For vibration motions study the above equations can be linearized replaced φ_1 by $\delta + \varphi_1$ where new φ_1 is a small variation, of elongation below 5° .

The same consideration is for bucket chain 8 of angular motion φ_2 , which is replaced by $\varphi_{20} + \varphi_2$, φ_{20} being the geometrical position of the bucket chain, fixed at $\varphi_{20} = 60^\circ$, φ_2 becoming of the small variation as φ_{20} , and φ_0 being also of small elongation.

Developing (5) in the Taylor series, around geometric length $\overline{O_1 O_2(0)}$, and taking in consideration only first order terms it is obtained the linear form,

$$\overline{O_1 O_2(t)} - \overline{O_1 O_2(0)} \approx e_0 \varphi_0 + e_1 \varphi_1 + e_2 \varphi_2 \quad (6)$$

were

$$e_i = \left(\frac{\partial \overline{O_1 O_2(t)}}{\partial \varphi_i} \right)_{t=0} \quad i = 0, 1, 2 \quad (7)$$

The same approximation is done for the tyrant length variation $\overline{O_3 O_4(t)} - \overline{O_3 O_4(0)}$.

The generalized forces column vector $\{Q\}$, contains two components, one, $\{Q_{Fc}\}$ of the cutting force of bucket wheel 4' during excavation, and second $\{Q_G\}$, of static loads $m_0 g$, $m_1 g$ and $m_2 g$, occurring from potential energy (3) series developed.

Finally, passing over a large number of developed forms, [3], an approximate dynamic model of excavator structure motion in vertical plane, can be written of the form

$$[M] \ddot{\varphi} + \left([K_0] + c_2 [k_{ptk}] + c_3 [k_{tr}] \right) \varphi = \{Q_{Fc}\} + \{Q_G\} \quad (8)$$

$[M]$, being the inertia matrix, deriving from kinetic energy.

The elasticity of the excavator ensemble structure is presents in the equation (8) by three matrices; $[K_0]$ of thrust stiffness, $[k_{ptk}]$ of pulley tackle stiffness, $[k_{tr}]$ of tyrants stiffness, and $[K_G]$, is a matrix deriving from potential energy (3). The damping of excavator structure can be considered a proportional

one the values of constants c_1, c_2 and c_3 , are fractions of proportional damping developed in thrust bearing, pulley tackle and tyrants, and these terms occurring from dissipation energy E_{ds} .

In view to simulate different critical excitations, dangerous for the integrity of the excavator structure, it is necessary to identified, by a hybrid, experiment and design data, the system of equations (8). Very important, for this identification task, is to have determined more accurate the lows motion of the structure including cinematic displacement and those which occur from elastic deformations.

In the case of using seismic sensors for motion measurement, as a capacitive or strain gage accelerometers, it is very hard to be separate the two component of the sensor output signal, the component of acceleration along accelerometer sensitivity axis and component of the gravity acceleration g projection on the same axis. More suitable is a laser measurement system which is a expensive equipment to be sporadic used, as for the special measurements as in presented cases.

Below is presented an alternative method which respond to the both desiderates, easy to be used and inexpensive.

2 CAMCORDER IMAGE PROCESSING METHOD FOR VERTICAL MOTION DETERMINATION

In view to determine the absolute vertical motion of any point on the excavator or other large structure it was placed in the structure area of interest a target rectangular panel T_{rpl} (Figures.1and 2) on which was drawn along its surface a matrix grid of points spanned with $d=50$ mm between. The points are marked by square dark spots of 5mm.

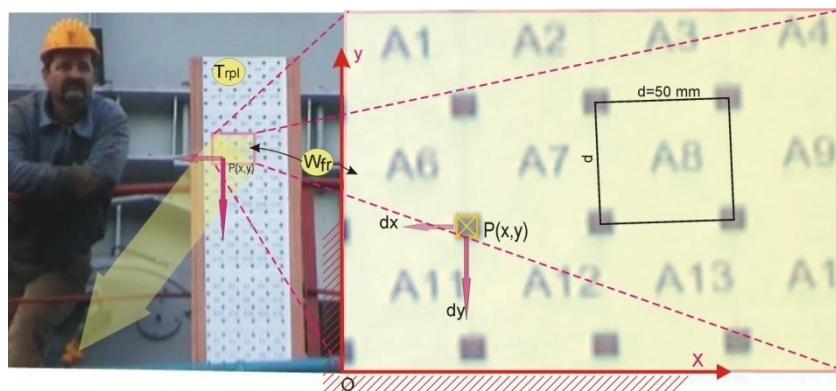


Figure 2 Illustration of target plate with the matrix grid of points

The camcorder **cam** was placed, by a fixed tripod, on the ground and its zoomed window frame w_{fr} was focused in the area of target panel. The camcorder being fixed on the ground the frame w_{fr} remain also fixed in the space, in the same vertical plane, covering a given space of panel grind. So, the absolute position of a

given spot point P of the grind is given by the coordinates x and y , reference frame Oxy , considered as fixed in space, connected to windows frame w_{fr} .

The digital images recorded as a move in an AVI extension file, for example, can be imported by a general used graphical software, as Corel R.A.V.E console (in this paper). In Figure 3 are illustrated the technology of the coordinates determination. By the cursor, an image, recorded at time $t_i=i\Delta t$, is displayed on the screen, i being the number of the image frame and Δt the time increment between two consecutive images. With a point frame Pframe, (yellow frame) free moved on the image by mouse, it is superposed exactly over the contour of a dark square spot of the grid connected on the structure. The coordinate x_{s1} and y_{s1} of the center P_1 of the spot, reference frame $Sx_s y_s$, can be reading by Corel screen ruler (Figure 3) and numerically displayed on the *Property bar: Group*.

For the next image frame the point P_1 changes its position and new values of coordinate x_{s1} and y_{s1} can determined and written in a file.

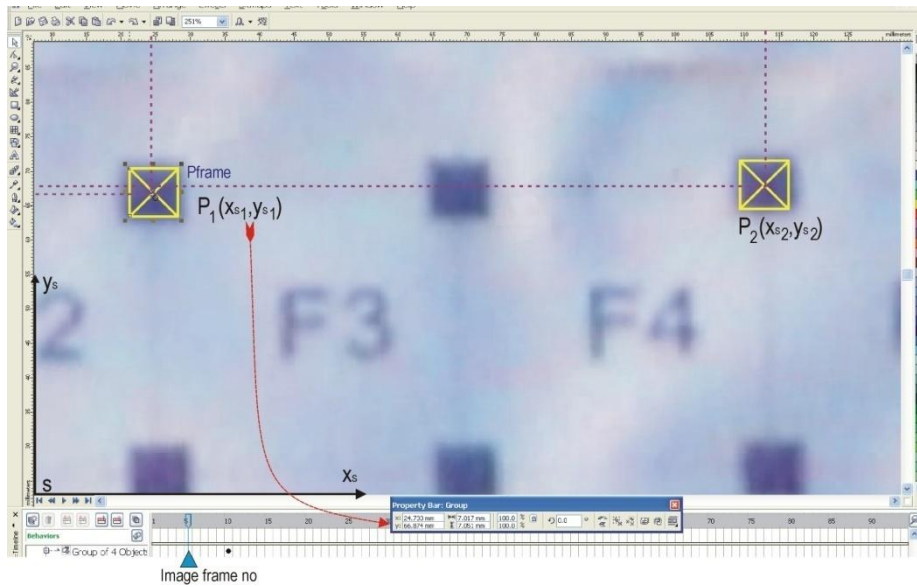


Figure 3 Illustration of point position measurement using Corel R.A.V.E.

To get the actual values of the point P_1 , coordinates, x_1 and y_1 , it is takes in consideration that grid span is known, in this case $d=50$ mm, so, taking the coordinates x_{s2} and y_{s2} of a second point P_2 , (in the Figure 3, two spans distance to point P_1) can calculate a calibration constant

$$k = \frac{2d}{\sqrt{(x_{s1} - x_{s2})^2 + (y_{s1} - y_{s2})^2}} \left[\frac{\text{mm}}{\text{mm}_{\text{screen}}} \right] \quad (9)$$

Now processing image by image can be obtained the screen coordinate values of two or more points, which define the vertical plan motion of the structure area where the target grid plate was attached.

For large displacements when one of the points quantified tends to leaves the screen windows than other point which just entering in the screen windows is quantified. The coordinates of the point which is leaving the screen can be determinate easy, by taking in consideration the span between the point processed and new one.

On the grid panel are, also, printed the field number, (in the figure 3 appearing the fields F2, F3 and F4) which are helpful in image processing.

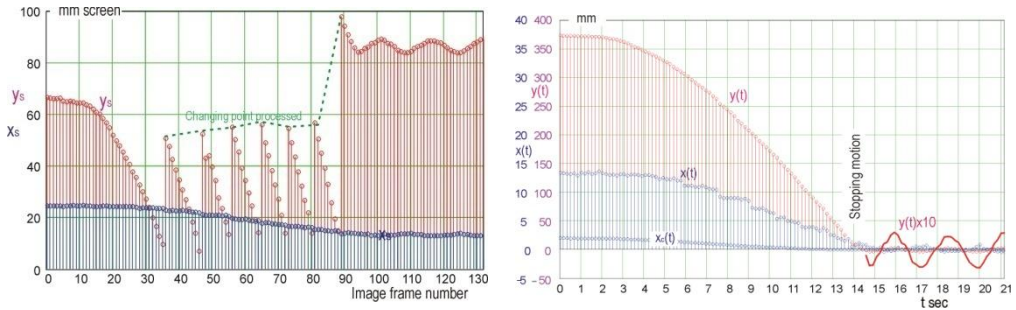
For a general motion of a large structure can be used two or more synchronized camcorders. One, with a fixed target area, and the others with the sequential changed target areas, covering all structure. More practically for this synchronism is to replace the video signal of first camcorder with a signal of an accelerometer placed on the structure in the same place, or more suitable being a signal of a strain gage montage, as that used in this work,; a signal of the reaction force R_A in bearing A (Figure 1), measured by a full strain gages bridge montage applied on the bearing A beam, which shares the bearing load to the two caterpillars. This signal is more suitable because it contents, as in video signal, the both components static and dynamic, necessary for structure equilibrium stability determination.

3 APLICATIONS AND CONCLUSION

3.1 Excavator arm motion recording

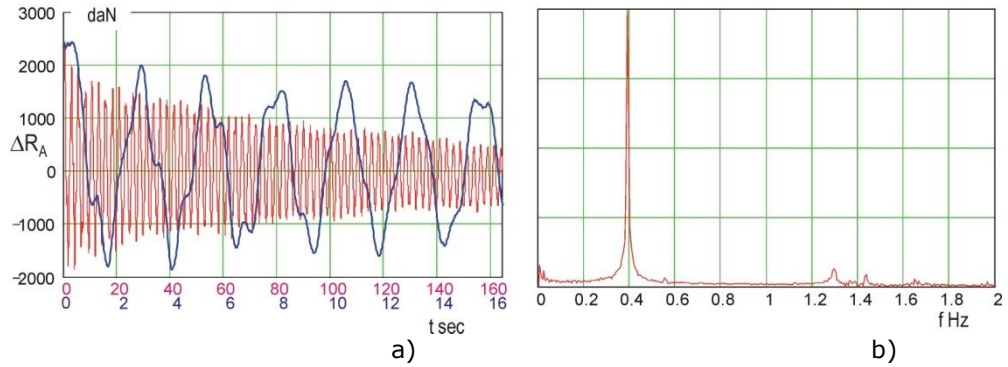
First verify of the above presented method was applied to an excavator structure, being determined the motion in the vertical plane of the area of the bucket wheel 4' (Figure 1) The arms of the excavator starts down from the horizontal position and after a vertical displacement of about 370 mm the motion was stopped, occurring free damping decay vibration.

In Figure 4a are illustrated the diagrams of the two horizontal and vertical motions by screen coordinates x_s and y_s respectively. It can be observed that along the sequential processing it was changed seven times the processed point, the length of the displacement overcome many times the screen windows dimension. The all sequences connected lead to the actual motion lows plotted in the Figure 4b, which find out a smooth start of the motion and the same smooth action of the break, the dynamic effect being of low level, resulting the vertical vibration $y(t)$ under 3 mm elongations (se the 10 times amplified $y(t)$ plot during 14.5 to 20 seconds time record).



a) sequential image processing b) time history of actual displacement

Figure 4. Illustration of horizontal and vertical motion of the excavator arm



a) b)
Figure 5. Time history, a), and its FFT spectrum, b), of the ΔR_A reaction in bearing A variation during breaking motion of the excavator arm

The same time evolution is for the signal of reaction in bearing A variation ΔR_A (Figure 5a), the main components being of first mod at frequency 0.3893 Hz , also pregnant in the signals $x(t)$ and $y(t)$ (Figure 4a). Applying the modal parameter [1] it was found two modal components at 1.1955 and 1.435 Hz .

The same processing technology was applied to arm motion of a store yielding machine with bucket wheel (Figure 6b) where was processed two point P_1 and P_2 . In Figure 6a are presented the motion time histories of the two points P_1 and P_2 ; $y_{P1}(t)$ and $y_{P2}(t)$ vertical motions and $x_{P1}(t)$ and $x_{P2}(t)$ horizontal motions. It can be notices that motion lows of two points are the same denoting the main motions being of translation. The record sequence presents a machine arm in down motion of 80 mm, with a final break down, following by free decay vibration motions, inertial excited, occurring by the first modal component of 0.51 Hz frequency, the first mode being pregnant presented and in the reaction force R_A signal, synchronized with the image recording.

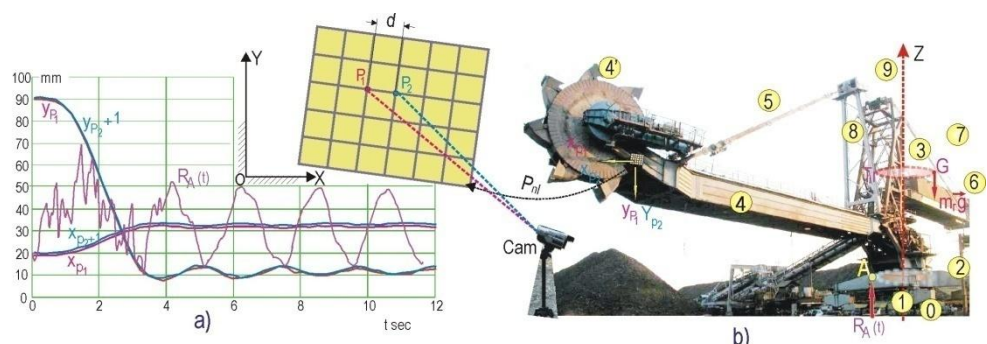


Figure 6 Illustration of displacement measurement in two points;
 a) time history of displacements b) store yielding machine with bucket wheel.

3.2 Large blade of a wind turbine modal testing

Another verify of the method was in Vibration Testing and Research Laboratory (VTRLoF University of Timișoara) on a wind turbine blade (1 Figure 7a), in modal testing on a dedicated stand 2, [4]. The panel grid 3 was placed on the free end of the blade (14m, 1500 kg) horizontal clamped in a massive body of the stand and the camcorder was placed at 11 m distance. It was recorded a free decay vibration in the vertical plane start elongation on the vertical axis being of 10 mm. By image processing it was obtained the time history of both motions (Figures 7b). $y(t)$ by vertical axis, of which spectrum, (figure 7c), find out the all modal components in the frequency domain 0 12 Hz, instead of being main excited only first mode of frequency 2.35 Hz. Even the horizontal motion $x(t)$ was obtained, instead of being under 0.5 mm elongation level (representing 10 times amplified in Figure 7b).

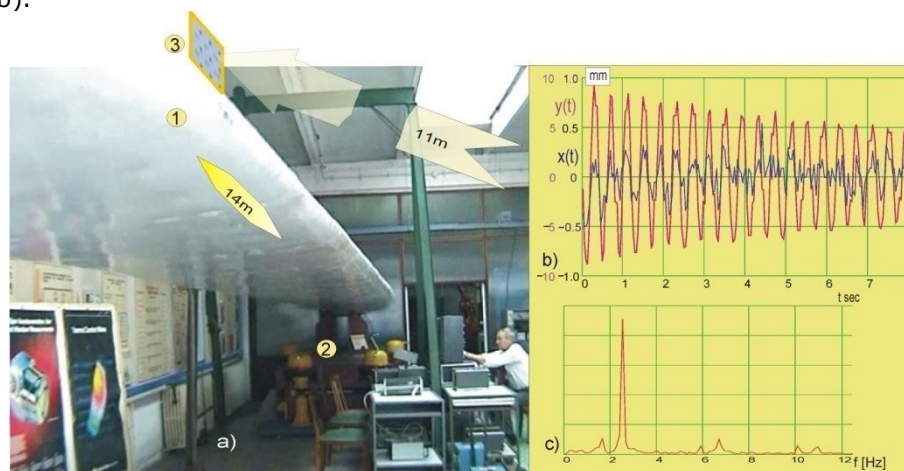


Figure 7 Illustration of displacement measurement in two points.

Finally conclusions:

- a useful and cheap method for large displacement measurement using only a commercial camcorder, as Sony DCR-TRV120E used for this work;
- accuracy of the measurement and sensitivity are good, sufficient for the applications presented above, and can be improved using the latest commercial camcorders;
- no special climatic conditions are necessary, the clouded weather in the field applications being presented
- both components, cinematic and deformation displacements are determined, and, very important for the mining structures, they can be separated

REFERENCES

- [1]. Cioara T.Gh. "On a Curve-Fitting Algorithm for Modal Parameter Estimation," Proceedings of the 6th International Modal Analysis Conference, Kissimmee, Florida, February 1988
- [2]. Cioara T.Gh. "Vibration testing of structures" Lectures (in Romanian unpublished) Univ. Politehnica Timișoara
- [3].VTRL Research Reports 295/07.31.2005-2006.
- [4] Patent Ro 102140

ANEXA 3

DYNAMIC STUDY OF A SPECIAL CRANE SERVING A POWER PLANT TALL CHIMNEY

Titus Gh. Cioara, Ph.D.
Professor, Head of
Vibration Testing &
Research Laboratory
University
"Politehnica" Timișoara
e-mail
cioara@mec.upt.ro

Ioan Cireș, PhD
Director R&D ROMINEX SA
Timișoara
e-mail:
proiectare@yahoo.com

Ionel Nicolae, Ph.D, student
Vibration Testing & Research
Laboratory
University "Politehnica"
Timișoara
e-mail
nicolae@mec.upt.ro

Dumitru . Cristea,
Director R&D
Energetical
Complex Turceni Targul
Jiu;

Amalia Jîrdea PhD student
Vibration Testing &
Research Laboratory
University "Politehnica"
Timișoara
e-mail
amaliatardea@mec.utt.ro

Laurentia Timar,
Ph.D, student
Computer Science
University "Politehnica"
Timișoara
e-mail
timar_laurentia@yahoo.com

ABSTRACT.

The paper presents the dynamical study of a special crane serving a power plant tall chimney, of 180m high. The crane consisting from a pivoting latticed boom placed at +115 m on the chimney, lifts parts and equipments along of vertical 120 m course.

In this case, the hoist ropes cables of pulley tackle being very longs, and of low stiffness, in the lifting time occur different dynamics effects, as hinged load vertical and pendular vibration motions. Consequently, the position of load hinged load is sometime difficult to be controlled.

By experiment, using strain gage mountings along the lattice boom, it was obtained important data which in a hybrid corroboration with the design data was elaborated an actual dynamic model, as base for different extremely dangerous excitation simulation: accidental losing the load, or a strong suddenly action of breaker

NOMENCLATURE

α_0, α angular position and motion of pivoting lattice boom
q first mode bending coordinate of pivoting lattice boom
 γ pendular coordinate of the load

1 CONSIDERATION ON CRANE DYNAMICS

Constructively the crane, Figure 1, is composed from a pivoting latticed boom, 1, placed at +115 m on the platform of a tall chimney. The pivoting latticed boom, 1 can be considered as special one, from the fact that the load 5 is raised from the ground to platform (more of 120m) without to be guided. The path of the load is closed to vertical wall of the power plant building so in the pendulum motion the load eventual can takes contact with the wall the, the contact being dangerous, the contact load being under control. For local manipulations of the low loads are is using a supplementary pivoting lattice boom 3

For a approximated discrete dynamic model the pivoting lattice boom, 1, (Figure 2) is considered as an equivalent elastic bending beam, which is hanged, at the pivoting position angle α_0 by a pulley tackle 3, of equivalent stiffness k_t . The rising of the load, 5, is serving by a pulley tackle, 4, of the equivalent stiffness k_t .

According to experimental data observed, the instant position of a section S_u , along the beam at the pivoting angle $\alpha + \alpha_0$ can be expressed, reference frame OXZ, by the coordinates

$$\begin{aligned} x_u &= u \cos(\alpha + \alpha_0) - f(u)q(t) \sin(\alpha + \alpha_0) \\ z_u &= u \sin(\alpha + \alpha_0) + f(u)q(t) \cos(\alpha + \alpha_0) \end{aligned} \quad (1)$$

where u is the coordinate of section S_u in the undeformed state, and $f(u)$ is the elastic bending deformation of the beam by first natural mode of law $q(t)$. α_0 expressing the geometric position and α the variation under dynamic loads

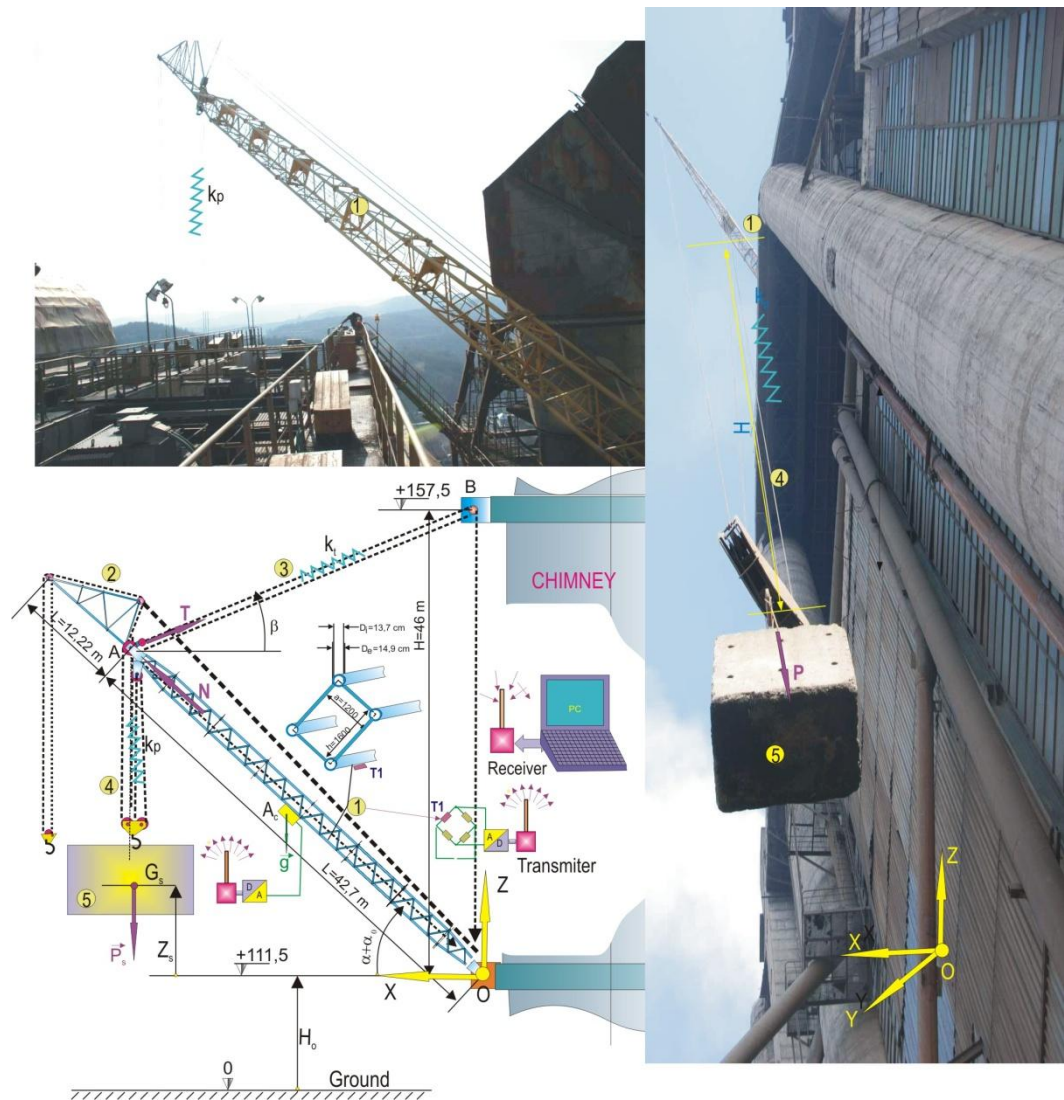


Figure 1 Illustration the special crane which serves the equipments of the tall chimney

The first mod shape of a the structure can be approximate by static deformation distribution, which in the case of the our beam, with the weighting load p uniform distribution along the beam of length $L=42.7$ m is of form

$$f_{st}(\xi) = R \cdot \xi \left(-2\xi^2 + \xi^3 \right) / 24 \quad (2)$$

with the notations

$$\xi = \frac{u}{L}; \quad R = \frac{\rho L^4}{EI} \cos(\alpha_0)$$

(3)

E, and I are the Yung modulus, and, respectively, the geometric inertial moment in transversal section of the beam. For the equivalent section of the beam the inertial moment

$$I = 4 \left(\frac{h}{2} \right)^2 A; \quad A = \frac{\pi}{4} (D_e^2 - D_i^2)$$

(4)

The kinetic energy of the equivalent beam is of integral form

$$E_{c_b} = \frac{1}{2} \int_0^L \rho (\dot{u}^2 + \dot{y}_u^2) du$$

(5)

where, $\rho = p/g$ (9.81 m/s²) is the mass on unit length of the beam.

Taking into account a modal distribution along the length L of the beam expression of kinetic energy (5) become of form

$$E_{c_b} = \frac{1}{2} J \dot{\alpha}^2 + \frac{1}{2} S_q \dot{q}^2 + S_{\alpha q} \dot{\alpha} \dot{q}$$

(6)

where

$$J = \int_0^L u^2 \rho du; \quad S_q = \int_0^L \rho (f(u))^2 du; \quad S_{\alpha q} = \int_0^L \rho f(u) u du$$

(7)

of which values can be calculate from design data.

The second kinetic energy of the dynamic system is of the load considered as a punctual mass in its gravity center G_s , of the coordinates

$$x_s = L \cos \alpha + (w_0 + w) \sin \gamma$$

$$z_s = L \sin \alpha - (w_0 + w) \cos \gamma$$

(8)

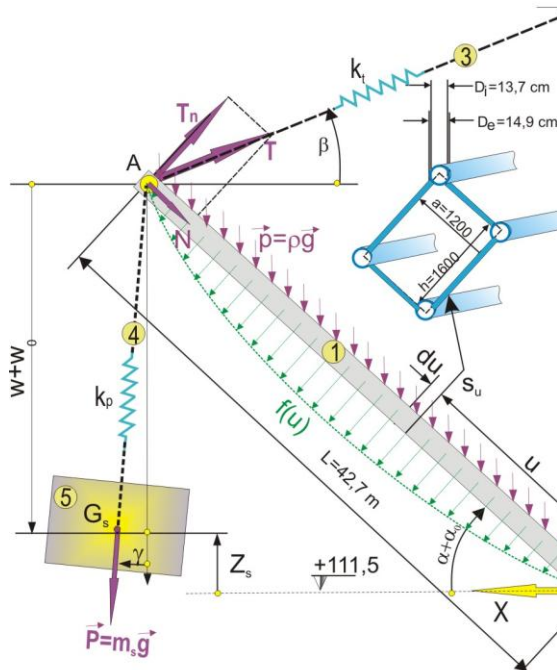


Figure 2 Dynamic schematic model of the crane

so that, the kinetic energy of the load is of the form

$$E_{c_s} = \frac{1}{2} m_s (\dot{x}_s^2 + \dot{z}_s^2) \quad (9)$$

which, by substitution (8) in (9) lead to the final form

$$E_{c_s} = \frac{1}{2} m_s \dot{\alpha}^2 + \frac{1}{2} m_s \dot{w}^2 + \frac{1}{2} m_s (w_0 \dot{\gamma})^2 + m_s L \dot{\alpha} \dot{w} \cos(\alpha_0 - \gamma) - m_s L \dot{\alpha} w_0 \dot{\gamma} \sin(\alpha_0 + \gamma) + m_s w_0 \dot{w} \dot{\gamma} \sin 2\gamma \quad (10)$$

Deformation energy of the dynamic system is composed from three components;

1. bending energy of the equivalent beam, 1,

$$E_{d_b} = \frac{1}{2} k_b q^2 \quad (11)$$

where

$$k_b = EI \int_0^l \left(\frac{\partial^2 (u)}{\partial u^2} \right)^2 du \quad (12)$$

1. Deformation energy of the cables of pulley tackle 3,

The stiffness k_t of the wiring cables of pulley tackle is of the form,

$$k_t = n_c \frac{E_{ct} I_{ct}}{AB} \quad (13)$$

where E_{ct} and I_{ct} are the Young modulus and inertial moment of a single wiring cable of number n_c

The deformation energy of pulley tackle is of the form

$$E_{d_{pt1}} = \frac{1}{2} k_t (AB)_t^2 - (AB)_0^2 \quad (14)$$

where $(AB)_t$ is the instant distance, at time t , between the two points A and B and $(AB)_0$ the distance with the system unloaded at angular position α_0 of the beam.

From geometric conditions

$$(AB)_0 = \sqrt{(\cos \alpha_0)^2 + (L \sin \alpha_0 - H)^2} \quad (15)$$

The instant position can be considered as a small variation α ($\alpha < 5^\circ$, $\sin \alpha \approx \alpha$; $\cos \alpha \approx 1$) which is added to α_0 resulting

$$(AB)_t - (AB)_0 = R_\alpha \alpha \quad (16)$$

where

$$R_\alpha = \frac{-LH \cos \alpha_0}{\sqrt{(L \cos \alpha_0)^2 + (L \sin \alpha_0 - H)^2}} \quad (17)$$

Finally from (14)

$$E_{d_{pt1}} = \frac{1}{2} k_t R_\alpha^2 \alpha^2 \quad (18)$$

2. Deformation energy of the pulley tackle 4 used for load rising

The stiffness k_p of the wiring cables of pulley tackle is of the form,

$$k_p = n_c \frac{E_{ct} I_{ct}}{w_0} \quad (19)$$

where E_{ct} and I_{ct} are the Young modulus and inertial moment of a single wiring cable of number n_c , and w_0 the

The deformation energy of pulley tackle is of the form

$$E_{d_{pt2}} = \frac{1}{2} k_p w^2 \quad (20)$$

Motions being studied in the vertical plane, in the energetic balancing must be take I consideration the potential energy

For the deformable beam is of the form

$$E_{p_b} = g m_b \frac{L}{2} \sin(\alpha + \alpha_0) + \rho g q c \cos(\alpha + \alpha_0) F; \quad \left(F = \int_0^L f(u) du \right) \quad (21)$$

and, for load

$$E_{p_s} = m_s g z_s = m_s g (L \sin(\alpha + \alpha_0) - w_0 + w (1 - \cos \gamma)) \quad (22)$$

The above developed forms of energies are used to build the system of differential equations, governing the motions of the crane, defined by the column vector

$$\mathbf{q}(t) = \begin{Bmatrix} \alpha(t) \\ q(t) \\ w(t) \\ \gamma(t) \end{Bmatrix} \quad (23)$$

and using the Lagrange method

$$\frac{d}{dt} \left(\frac{\partial E_c}{\partial \dot{\mathbf{q}}(t)} \right) - \frac{\partial E_c}{\partial \mathbf{q}(t)} + \frac{\partial E_d}{\partial \mathbf{q}(t)} + \frac{\partial E_p}{\partial \mathbf{q}(t)} = \mathbf{Q} \quad (24)$$

occurring the discrete system of equations

$$\mathbf{M}^{-1} \mathbf{K}^{-1} \mathbf{q} \quad (25)$$

were

$$\mathbf{M}^{-1} = \begin{bmatrix} J+m_s L^2 & S_{\alpha q} & 0 & -m_s L w_0 \sin \alpha_0 \\ S_{\alpha q} & 0 & 0 & 0 \\ 0 & 0 & m_s & 0 \\ -m_s L w_0 \sin \alpha_0 & 0 & 0 & m_s \end{bmatrix};$$

$$\mathbf{K}^{-1} = \begin{bmatrix} k_t R_\alpha^2 - g m_b \frac{L}{2} \sin \alpha_0 + g m_s L \cos \alpha_0 & 0 & 0 & 0 \\ 0 & k_b & 0 & 0 \\ 0 & 0 & k_p & 0 \\ 0 & 0 & 0 & m_s g w_0 \end{bmatrix} \quad (26)$$

2 EXPERIMENTAL RESULTS AND CONCLUSIONS

In view to get information on the modal components excited in the working time of the crane it was used two sensors (Figure 1) a strain gage sensor T_1 applied on one, of the four, longerons, of the pivoting latticed boom, 1 in the middle distance, at $L/2$.

At the same distance on the longeron was placed an accelerometer A_c with its sensitivity axis normal to longitudinal axis of lattice boom. As accelerometer was used ANALOG DEVICES ADXL103, of which measuring range take in the frequency zero, so that the accelerometer measure the component of static and quasi static inclination. The both signals, of the strain gage and of the accelerometer are wireless connected to computer.

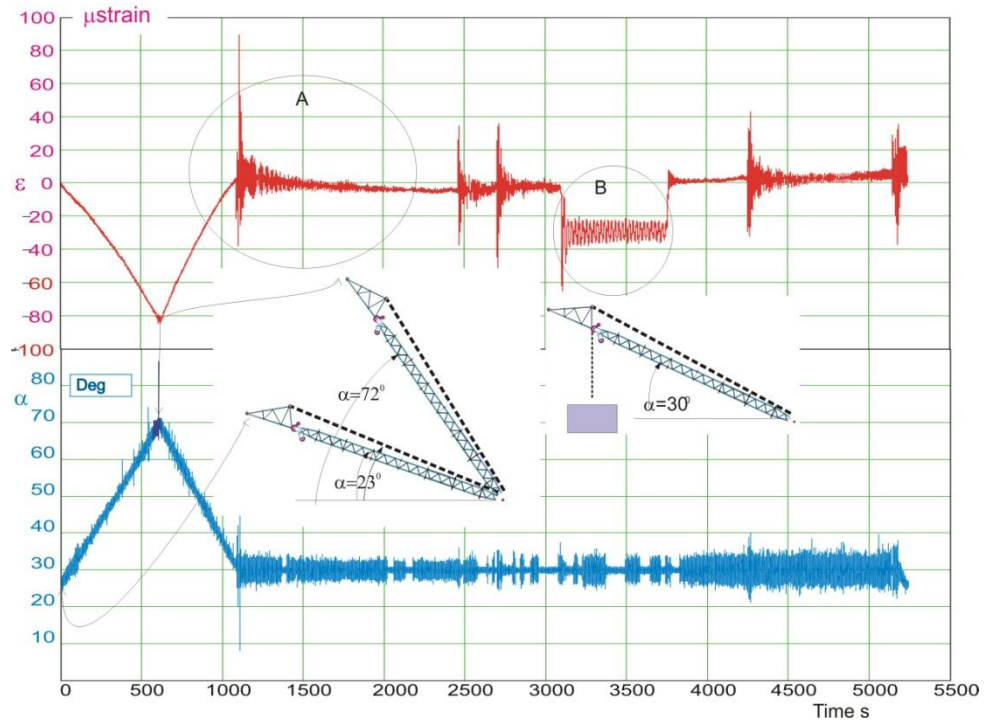


Figure 3 Time histories of the strain and angular position $\alpha=23^{\circ}$ to 70° .

In the Figure 3 are presented the time histories of the two signals. First operation is a very slowly tilting form angular position $\alpha=23^{\circ}$ to 70° , during 600 s, without the load m_s to be hanged. At this, near the static operation, results the tension T in the cables of the pivoting pulley tackle 4.

$$T = \rho g \frac{L}{2} \frac{\cos \alpha_0}{\sin(\alpha_0 + \beta)}$$

(27)

The two components of tension T ,; T_n , normal to the axis of the beam, and N axial

$$T_n = T \sin(\alpha_0 + \beta); \quad N = T \cos(\alpha_0 + \beta);$$

(28)

The bending momentum in the middle section of the beam, at $L/2$, is of the form

$$M_i = T_n \frac{L}{2} - \rho g \frac{L^2}{4} \cos \alpha_0$$

(29)

and, the axial force in the same section is of form,

$$N_m = N + \rho g \frac{L}{2} \sin \alpha_0 \tag{30}$$

The strain ε on the longeron where the strain gage is placed can be calculated with the form,

$$\varepsilon = \frac{1}{E} \left(\frac{M_{lh}}{2l} - \frac{N_m}{4A} \right)$$

which is in accordance with experiment, the maximum of compression strain is reached at $-80 \mu\text{strain}$, relative to position α_0 of the beam.

The dynamic effect is more intensive at the end of operation, when the

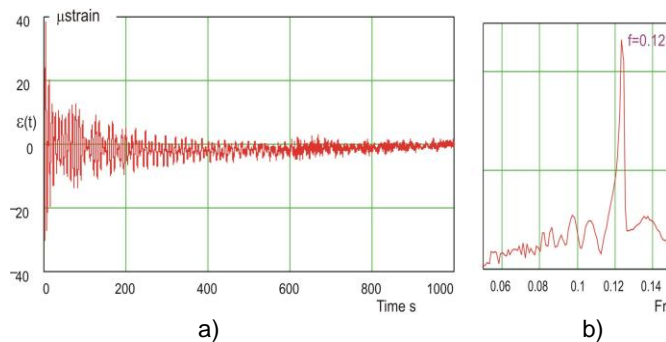


Figure 4 Time history a), and its spectrum, b), of the strain during the transient break to a given position of pivoting lattice boom.

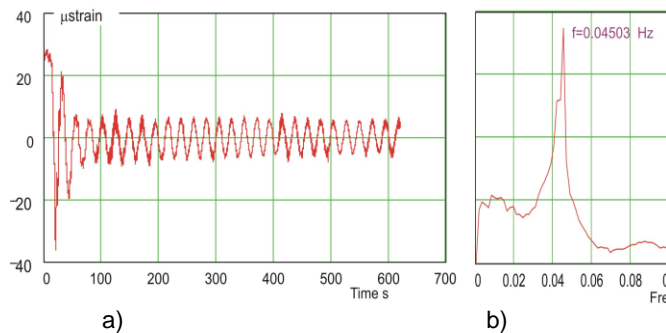


Figure 5 Time history a), and its spectrum, b), of the strain during rising the load

rocking motion of the beam is stopped at position angle $\alpha_0 \approx 30^\circ$, (medallion A Figure 3), occurring dynamic transient strain of magnitude $40 \mu\text{strain}$ (Figure 4), containing a modal component of frequency, $f=0.12378 \text{ Hz}$, estimated by [2].

In the medallion B of the Figure 3, and detailed in Figure 5 are presented the time history of the dynamic strain, and its spectrum, developed during of rising the load of $m_s=6000 \text{ kg}$.

From spectrum of the strain time history results two peaks, one of the frequency $f=0.045 \text{ Hz}$ and other having the same frequency as in spectrum b) Figure 4. of 0.124 Hz . The value of 0.045 Hz correspond with the pendulum of load hinged by a long cable of $w_0=110 \text{ m}$, resulting, from $f=(1/2\pi) \sqrt{g/w_0} \approx 0.045 \text{ Hz}$

The second value of 0.124 Hz , corresponds to rocking, α , vibration motion, this frequency component belonging to both situations,

The paper presented only of frame of the complete works [3] which is in progress having as target a new design for the pivoting latticed boom, improving its performances, especially the speeds of the operations. Another target is to design a

new pivoting lattice boom more light, from the experimental works resulting small loaded of boom structure.

REFERENCES

[1]. Cioara T.Gh. "Experimental Technique in Engineering. Transducers and Sensors Ed. Politehnica Timișoara 1999 ISBN 973-9389-33-3(in Romanian)

[2] Cioara T.Gh On a Curve-Fitting Algorithm for Modal Parameter Estimation," *Proceedings of the 6th International Modal Analysis Conference*, Kissimmee, Florida, February 1988

[3] *** Optimal design of a pivoting latticed boom. Research Contract no. 120/2007-2008, University 'Politehnica' of Timișoara in cooperation with ROMINEX S.A. Timisora.

Cuvânt înainte

Teza de doctorat a fost elaborată pe parcursul mai multor ani de cercetare. În această perioadă am fost profesoară de Informatică la Liceul Nikolaus Lenau, apoi am colaborat cu firma Rominex, iar din 2001 lucrez la firma Siemens Automotive, actuală Continental.

Nefiind parte a unui colectiv de cercetare dedicat exclusiv studiului, teza am realizat-o în timpul meu liber, cele mai importante acumulări fiind în perioadele de creștere ale ultimilor doi copii.

Teza are un grad accentuat de aplicabilitate. Metodele și sistemul propuse în această teză sunt funcționale și pot fi aplicate în mai multe domenii. Din acest motiv cercetarea mea am protejat-o prin înaintarea unei cereri de brevet de invenție.

Trebuie să-l amintesc pe conducătorul meu de doctorat, prof.dr.ing. **Mircea Vlăduțiu** care m-a susținut în toți acești ani și care după analiza cercetării m-a îndemnat să înaintez brevetul de invenție.

În timpul căutărilor mele am găsit sprijin în colectivul de cercetare de la firma **Rominex**, sub îndrumarea domnului prof. dr. ing. **Cioară Titus**, care mi-a înlesnit participarea la redactarea a două articole la care sunt coautoare. Îi mulțumesc încă o dată pe această cale!

În aceste clipe gândul meu se îndreaptă spre domnul rector, prof. dr. ing. **Ioan Cartis** care m-a încurajat în toți acești ani, sfătuindu-mă să nu renunț și explicându-mi că pot să fiu un cercetător de excepție dar și o mamă bună și devotată în același timp.

Mulțumiri deosebite se cuvin domnului director **Teofil Ghirasim**, care m-a susținut și m-a propus să fac parte din colectivul de cercetare de la firma domniei sale. Le mulțumesc și angajaților și partenerilor săi!

În toți acești ani am fost susținută de minunata mea familie. Le mulțumesc tuturor și nu știu dacă le voi putea răsplăti vreodată pentru răbdarea și bucuria cu care au fost alături de mine! Le amintesc pe **Marioara**, soacra mea și pe **Cornelia**, actuala mea mamă, prietene de suflet.

Mulțumesc sorei mele **Aurora**, soțului meu **Lucian** și celor trei băieți: **Andrei**, **Gabriel** și **Paul**!

Îi amintesc și le mulțumesc **prietenilor** și **prietenelor** mele, care m-au ajutat și s-au rugat pentru mine în toți acești ani!

Îi mulțumesc cu pioșenie bunului **Dumnezeu** și **Maicii Sale** care mi-au dat multă putere și nu m-au părașit nicio clipă!

Timișoara, decembrie 2008

Laurenția Timar