

PROBLEMA CONSENSULUI ÎN CALCULUL TOLERANT LA ERORI

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

Ancușa Versavia Maria

Conducător științific: prof.univ.dr.ing Vlăduțiu Mircea
Referenți științifici: acad. dr. ing. Petrescu Mircea
prof.univ.dr.ing. Popescu Dana
prof.univ.dr.ing. Crețu Vladimir

Ziua susținerii tezei: 11 iunie 2009.

Seriile Teze de doctorat ale UPT sunt:

- | | |
|------------------------|---|
| 1. Automatică | 7. Inginerie Electronică și Telecomunicații |
| 2. Chimie | 8. Inginerie Industrială |
| 3. Energetică | 9. Inginerie Mecanică |
| 4. Ingineria Chimică | 10. Știința Calculatoarelor |
| 5. Inginerie Civilă | 11. Știința și Ingineria Materialelor |
| 6. Inginerie Electrică | |

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

Copyright © Editura Politehnica – Timișoara, 2006

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

România, 300159 Timișoara, Bd. Republicii 9,
tel. 0256 403823, fax. 0256 403221
e-mail: editura@edipol.upt.ro

Cuvânt înainte

Teza de doctorat a fost elaborată pe parcursul activității mele în cadrul Departamentului de Calculatoare al Universității „Politehnica” din Timișoara, sub conducerea Prof. Dr. Ing Mircea Vlăduțiu.

Mulțumiri deosebite se cuvin rectorului Prof. Dr. Ing. Nicolae Robu, pentru sprijinul acordat în participarea la conferințe. O parte din această teză este rezultatul colaborării cu ing. Răzvan Virgil Bogdan, căruia îi mulțumesc mai ales pentru răbdare, tact și tenacitate.

Timișoara, mai 2009

Versavia Ancușa

Părinților mei.

Ancușa, Versavia Maria

Problema consensului în calculul toleranței la erori

Teze de doctorat ale UPT, Seria X, Nr. YY, Editura Politehnica, 2009, 176 pagini, 62 figuri, 12 tabele.

ISSN:1842-7707

ISBN (10):; ISBN (13): 978-973-625-895-4

Cuvinte cheie: consens, arhitecturi din al treilea val, agenți inteligenți, toleranță la defectare

Rezumat,

Metoda atingerii consensului este una dintre metodele de bază în atingerea toleranței la defectare. Bazată pe problema generalilor bizantini, unele aspecte legate de temporalitate și de consistența rezultatului final se pot îmbunătăți bazat pe perspectiva socială. Pentru testare s-a căutat un sistem distribuit de actualitate. Drept urmare s-a născut conceptul de Intelligent Grid, o arhitectură din proximitatea val de computație care a fost emulată pe componente actuale folosind agenți inteligenți ce comunică folosind toate metodele de obținere a toleranței la defectare. Același protocoale de consens pot fi aplicate cu succes și în domeniul medicinei.

Cuprins

Lista figurilor	7
Lista tabelelor	9
Notații	11
1. Introducere	13
1.1. Motivație	13
1.2. Organizarea tezei	15
2. Consensul	17
2.1. Definiția consensului	17
2.2. Abordarea socială	18
2.3. Modele existente în știința calculatoarelor	21
2.3.1. Consensul în sisteme sincrone	22
2.3.2. Consensul în sisteme parțial sincrone	23
2.3.3. Consensul în sisteme asincrone.....	23
2.4. Modelul matematic al consensului	24
2.5. Direcții de dezvoltare	26
3. Cazul particular al consensului în știința calculatoarelor	29
3.1. Problema generalilor bizantini (BGP)	29
3.1.1. Utilizare practică și posibile beneficii	29
3.1.2. Apartenența la clasa problemelor de consens	30
3.1.3. Modelul matematic al BGP	31
3.1.4. Tipuri de mesaje în cadrul BGP	32
3.1.5. CBGP varianta cu mesaje orale	36
3.1.6. CBGP varianta cu mesaje scrise	40
3.1.7. WBG (Weak Byzantine Generals Problem)	42
3.1.8. BGP varianta cu mesaje orale	45
3.1.9. BGP varianta cu mesaje scrise	46
3.1.10. Alți algoritmi bazați pe BGP	47
3.1.11. Discuție.....	52
3.2. Licitatii	55
3.2.1. Utilizări practice ale licitațiilor	56
3.2.2. Apartenența la clasa problemelor de consens	56
3.2.3. Licitatiile prin formalism	57
3.3. Recunoașterea întregului	57
3.3.1. Utilizare practică	58
3.3.2. Apartenența la clasa problemelor de consens	58
3.3.3. Formalizarea recunoașterii întregului	59
3.4. Luarea de decizii.....	59
3.4.1. Utilizare practică	60
3.4.2. Apartenența la clasa problemelor de consens.....	60
3.4.3. Formalizarea problemelor de luare a deciziilor.....	60
3.5. Concluzie.....	60
4. Intelligent Grid	63
4.1. Motivația arhitecturii	63
4.1.1. Etapele evoluției în știința calculatoarelor	63
4.1.2. Rețele ce includ senzori	65

4.1.3. Rețele ce includ multimedia	71
4.1.4. Ambient Intelligence.....	75
4.1.5. Grid computațional.....	79
4.2. Arhitectura Intelligent Grid	82
4.3. Portabilitatea, scalabilitatea și configurabilitatea	85
4.3.1. Alegerea middleware-ului	85
4.3.2. Agenți inteligenți	87
4.3.3. Aplicație practică.....	91
4.4. Puterea, costul și dimensiunea	93
4.4.1. Implementare experimentală.....	99
4.5. Toleranța la defectare	109
4.5.1. Redundanța.....	112
4.5.2. Redundanță la nivelul zonelor de sensibilitate	112
4.5.3. Redundanță la nivel de legătură.....	113
4.5.4. Redundanță la nivel de mesaj.....	119
4.6. Consensul de tranzacții în IG.....	123
4.6.1. Parte de autentificarea a algoritmului de consens	123
4.6.2. Rularea algoritmului de consens.....	127
4.7. Discuție	132
5. O aplicație a consensului în medicină	133
5.1. Motivație	133
5.2. O descriere a problemei medicale	133
5.2.1. Determinarea riscului de osteoporoză și osteoartroză	134
5.3. Crearea unei aproximări a scorului T	135
5.3.1. Selectarea factorilor utilizați	135
5.3.2. Cuantizarea factorilor de risc	138
5.3.3. Crearea formulei de aproximare.....	139
5.4. Testarea formulei	140
5.5. Concluzii.....	141
6. Concluzii	143
6.1.1. Contribuții originale	145
6.2. Dezvoltări viitoare	149
Bibliografie.....	151
Distribuția în timp a bibliografiei	171
Anexa – lista de lucrări publicate	173

Lista figurilor

Figura 2.1. Ideea consensului	18
Figura 2.2. Procesul uman al atingerii consensului [w3co05]	20
Figura 2.3. Consensul în calculatoare [MaKa05]	21
Figura 2.4. Tranzacție bazată pe consens la baze de date distribuite	22
Figura 2.5. Paradoxul omului chel [MaKa05]	23
Figura 3.1. Comandantul este trădător [LaPS80]	34
Figura 3.2. Locotenentul 2 este trădător [LaPS80]	35
Figura 3.3. Comandantul este trădător; mesaje scrise prelucrare [LaPS80]	36
Figura 3.4. CBG, mesaje orale, locotenentul 3 trădător [LaPS80]	37
Figura 3.5. CBG, mesaje orale, comandantul e trădător [LaPS80]	38
Figura 3.6. Variația mesajelor funcție de numărul generalilor, $m=3=ct.$	53
Figura 3.7. Numărul de mesaje în algoritmi SM și FSM	54
Figura 3.8. Relația între modelele matematice	61
Figura 4.1. Evoluția claselor de calculatoare [Bell08]	64
Figura 4.2. Clasele de calculatoare și valorile de calculație	64
Figura 4.3. Piața senzorilor (mil. de \$) – estimări prelucrate din [OEMa05]	68
Figura 4.4. Arhitectura sensor-grid	69
Figura 4.5. Arhitectura Open SensorWeb [ThBu05]	70
Figura 4.6. Diversitate de semnale multimedia [Kins02]	71
Figura 4.7. Dezvoltarea economică a pieței multimedia – prelucrare din [w3em05] și [w3ve07]	72
Figura 4.8. Arhitectura mmGrid [BAKY03]	74
Figura 4.9. Evoluția care conduce la Ambient Intelligence [Chev04]	76
Figura 4.10. Ambient Intelligence [w3Ph08]	77
Figura 4.11. Posibile utilizări ale AmI [w3Ph08]	78
Figura 4.12. Estimare a pieței de grid computing [w3Su02]	80
Figura 4.13. Taxonomia sistemelor Grid [KrBM02]	80
Figura 4.14. Intelligent Grid (IG) [AnBV08a]	84
Figura 4.15. Ridicarea nivelului de abstractizare [Raba05]	85
Figura 4.16. IG modelată cu agenți [AnBV08a]	89
Figura 4.17. Timpul de rulare [AnBV08a]	92
Figura 4.18. Realizări practice de scavenging de energie [Raba04]	93
Figura 4.19. Efectul Seeback	95
Figura 4.20. Firele de oxid de zinc [Wang07]	95
Figura 4.21. Small scale design - Proiect PicoRadio [Raba05]	96
Figura 4.22. Energie per tip de instrucție [EkKM04]	97
Figura 4.23. FIPA Contract Net [BeCG07]	103
Figura 4.24. BAM aplicat pe protocolul ContractNet	104
Figura 4.25. Timpul de comunicare pentru protocolul BAM, erori prezente	106
Figura 4.26. Timpul de comunicare pentru protocolul BAM, erori absente	107
Figura 4.27. Timpul de comunicare pentru BAM și ABAM, erori absente	108
Figura 4.28. ABAM VS BAM - Zoom	109
Figura 4.29. Moduri de a studia dependabilitatea - prelucrare după [ALRL04]	110
Figura 4.30. Niveluri de fiabilitate	112
Figura 4.31. Redundanță la nivel de arie de sensibilitate [CoNS05]	113
Figura 4.32. Redundanță la nivel de legătură [AnBV08b]	114

Figura 4.33 Grad de acoperire senzori variabil, numărul de controlere fix (9) [AnBV08b]	115
Figura 4.34. Grad de acoperire senzori variabil, numărul de controlere fix (9) - Zoom[AnBV08c].....	116
Figura 4.35. Numar de controlere de variabil, numărul de erori de fix (1) [AnBV08b]	117
Figura 4.36. Numar de controlere de variabila, grad de acoperire fix (9) [AnBV08b]	118
Figura 4.37. Grad de acoperire variabil, numărul de erori de fix (1) [AnBV08c]...	119
Figura 4.38. Interacțiunea de la un client din punct de vedere. [GuSc01].....	120
Figura 4.39. Invocare-răspuns din punct de vedere al server-S1 [GuSc01]	120
Figura 4.40 Modul de calcul al semnăturii digitale [Xiao07]	123
Figura 4.41 Număr fix de controlere (9), numărul de erori de variabil [BoAV08b]	125
Figura 4.42. Număr fix de controlere (9), numărul de erori = 9 [BoAV08b]	126
Figura 4.43. Număr fix de erori (1) [BoAV08b]	127
Figura 4.44. . Rata de eroare maximă, medie și minimă VS numărul de agenți, 20 runde, adâncime de memorie = 5 [Ancu08a]	128
Figura 4.45. Timpul exprimat ms în VS numărul de agenți, 20 runde, adâncime de memorie = 5 [BoAV08a]	129
Figura 4.46. Rata maximă de eroare VS nr. de agenți pentru 50 de runde [Ancu08]	130
Figura 4.47. Timp de variație VS nr. de agenți pentru 50 de runde [BoAV08a]....	131
Figura 5.1. Evoluția osteoartrozei [SPAR06]	133
Figura 5.2. Comparția între scor T estimat și scor T măsurat DXA. [SBGA07]....	140

Lista tabelelor

Tabelul 2.1. Condiții de insolvabilitate [MaKa05]	24
Tabelul 3.1. Tipuri de licitații [BeSe98]	56
Tabelul 4.1. Cele trei generații de rețele de senzori [ChKu03]	66
Tabelul 4.2. Atributele rețelelor de senzori [ChKu03]	67
Tabelul 4.3. Compunerea grid-urilor[AnBV08a]	83
Tabelul 4.4. Privire de ansamblu asupra cerințelor de programare[AnBV08a]	86
Tabelul 4.5. Tipuri de middleware si folosirea lor[AnBV08a]	87
Tabelul 4.6. Capabilitățile agenților[HuSt99]	90
Tabelul 4.7. O comparație între sursele de putere [Raba04]	94
Tabelul 4.8. Comparație între procesoare de putere ultra-scăzută [EkkM04]	98
Tabelul 5.1. Tehnici noninvasive de măsurare a masei osoase (5- cea mai mare, 1- cea mai mică) [ABVS08]	134
Tabelul 5.2. Analiza factorilor de risc (10-maxim, 0-minim) [ABVS08]	137

Notații

ABAM	Anti-hijacking Battery Aware Measurement Protocol
AmI	Ambient Intelligence
AOP	Agent-Oriented Programming
BAM	Battery Aware Measurement Protocol
BG1	Fiecare divizie a armatei bizantine este comandată de către propriul său general.
BG2	Generalii, dintre care unii sunt trădători, comunică între ei prin mesageri.
BG3	Toți generalii loiali trebuie să se decidă asupra aceluiași plan de acțiune
BG4	Un număr mic de trădători nu poate cauza adoptarea unui plan greșit.
BGP	problema generalilor bizantini cu generali egali ca și rang
BGR3	Toți generalii loiali primesc aceleași informații pe baza cărora vor lua decizia finală.
BGR4	Informațiile trimise de către generalii loiali trebuie să fie folosite de către toți generalii loiali.
BMD	Bone Marrow Density
CBG1	Toți locotenenții trebuie să urmeze aceleași ordine.
CBG2	Dacă comandantul general este loial, atunci fiecare dintre ceilalți locotenenți loiali ascultă de ordinele trimise de acesta.
CBGP	problema generalilor bizantini cu un comandat și n-1 locotenenți
CFP	Call For Proposals
D1	Consensul reprezintă rezultatul final al uneia sau mai multe prelucrări realizate asupra unui șir de valori inițiale; rezultatul final fiind mai apoi recunoscut și însușit, la sfârșitul dezbaterilor, de către suficient de mulți participanți la prelucrări. Valorile inițiale pot fi total sau parțial diferite, apărute simultan sau succesiv.
D2	Înțelegerea: Nu există două procese care se decid asupra unor valori diferite. Validitatea: Valoarea asupra căreia s-a decis trebuie să fi fost valoarea inițială a cel puțin unui proces Terminarea: Fiecare proces corect se va decide în cele din urmă asupra unei valori, iar această decizie este irevocabilă.
DAI	Distributed Artificial Intelligence
DXA	Dual-energy X-ray Absorptiometry
IG	Intelligent Grid
IMC	Indice de Masă Corporală
m	numărul de trădători
M	factor de majoritate
MEMS	Micro-electro-mechanical systems
n	numărul de generali
OMH1	Fiecare mesaj trimis este receptat corect. Căderea liniilor de comunicații între două componente nu se distinge față de căderea uneia dintre componente. Căderea liniei mai adaugă doar încă un trădător

OMH2	Cel care primește mesajul știe cine l-a trimis. Avem deci o rețea sub forma unui graf tare conex, cu fire de comunicație ideale.
OMH3	Absența unui mesaj poate fi detectată. Acest lucru se poate realiza doar în sistemele sincrone, cu ajutorul unui mecanism de time-out.
QoS	Quality of Service
SOA	Service-Oriented Architecture
v:i:j	valoarea v semnată (cronologic) de i și j
V_f	mulțimea valorilor finale
V_i	mulțimea valorilor inițiale
WBG1	Dacă nici un general nu trădează și toți generalii doresc inițial aceeași acțiune, atunci aceasta va fi acțiunea aleasă de către generali
WBG2	Oricare doi generali loiali aleg aceeași acțiune finală.
WBGp	problema generalilor bizantini relaxată
WD2	Înțelegerea și validitatea de la D2 Terminarea „ușoară”: Unele procese corecte se vor decide în cele din urmă asupra unei valori, iar această decizie este irevocabilă
WMH1	Semnătura unui general loial nu poate fi modificată Rezultă deci că odată depus un mesaj acesta nu poate fi modificat
WMH2	Oricine poate verifica autenticitatea semnăturii
Θ (tau)	numărul maxim de apariții al oricărei valori în vectorul de ieșire, Y , al funcției ψ .
v (niu)	funcția frecvență a aparițiilor unui simbol
ψ (psi)	funcția de consens

1. Introducere

1.1. Motivație

În orice sistem cu aplicație practică, noțiunea de dependabilitate este esențială pentru a permite buna funcționare a sistemului. Dependabilitatea, sau abilitatea de a evita căderile care se află sub o anumită limită și/sau frecvență [ALRL04], poate fi abordată fie prin studierea atributelor sale (availability, reliability, etc); a tipurilor de amenințări sau prin studierea mijloacelor de a o atinge.

Metodele de atingere a dependabilității sunt: prevenirea, prezicerea, înlăturarea și tolerarea defectelor.

Prevenirea defectelor (fault prevention) se realizează de obicei în faza de dezvoltare a sistemului/produsului, în timp ce prezicerea și înlăturarea defectelor sunt activate înainte, respectiv după apariția unor probleme.

Des întâlnită în natură, dar prezentând probleme la implementare în sistemele create de om, toleranța la defectare (fault tolerance) dorește menținerea funcțională a sistemului, în prezența erorilor. Pentru a obține tolerarea erorilor într-un sistem se folosesc două tehnici majore: detectarea erorilor și recuperarea sistemului dintr-o stare prezentând erori, continuând cu aducerea lui într-o stare fără erori. Aceste tehnici se concretizează la nivel practic, prin intermediul a trei metode: redundanța, codurile detectoare și corectoare de erori și protocoale de consens.

Redundanța presupune, în mod tradițional, existența a mai multe unități funcționale identice, funcționând în paralel. Codurile detectoare și corectoare de erori sunt o modalitate de a asigura corecta transportare a informației între unități precum și posibilitatea de a reconstrucție a datelor eronate, de către unitatea destinație.

Consensul implică corelarea de date de la unități diferite pentru a obține aceeași rezoluție finală pentru o fracțiune suficientă corectei funcționări a sistemului din unitățile implicate. Algoritmii de consens pornesc de la datele anunțate de un set de componente, care pot sau nu să funcționeze corect, iar pe pe baza acestor date va trebui să se realizeze o acțiune simultan de către toate componentele care funcționează corect. Scopul acestor algoritmi nu este identificarea defectelor, ci aducerea părților care funcționează corect în aceeași stare, efectuând aceeași acțiune.

Algoritmii de consens se încadrează în cea de-a doua modalitate de obținere a toleranței la defectare: recuperarea sistemului dintr-o stare prezentând erori, cu aducerea lui într-o stare fără erori

Studiind acești algoritmi din cadrul științei calculatoarelor, am identificat anumite limitări ale acestora: mai concret, definiția din știința calculatoarelor cere ca toate datele inițiale să apară simultan, iar rezultatul final trebuie să se regăsească între datele inițiale.

În realitatea umană însă, consensul are un înțeles mult mai larg decât cel din știința calculatoarelor. Consensul uman acceptă date noi pe parcursul derulării

algoritmului de atingere a rezultatului, precum și extrapolări sau prelucrări mai complexe ale datelor.

În consecință voi încerca să formulez o nouă definiție a consensului, precum și un model matematic al acesteia.

Cel mai des, algoritmi de consens apar în cadrul sistemelor distribuite, prin urmare, pentru partea experimentală, se va alege o arhitectură distribuită pe care care să testeze algoritmi de consens.

Studiind diverse sisteme distribuite singurul lucru constant a fost evoluția lor. În consecință am studiat evoluția sistemelor de calcul și predicțiile lui Bell din 2008 pentru ceea ce reprezintă viitoarele rețele. Întreaga discuție și trecere în revistă a modului în care evoluează calculatoarele are drept scop marcarea momentului actual, drept unul în care o clasă de calculatoare este practic depășită de către alta.

Aceasta înseamnă că orice model teoretic nu este de ajuns să fie testat pe o arhitectură actuală, deoarece în acest caz durata de viață a acesteia este destul de scurtă. Ideal ar fi să testăm direct pe noua clasă de computație, dar aceasta nu este încă complet funcțională.

Tendința evolutivă este folosirea simultană a unei varietăți cât mai mari de echipamente, extinzând în același timp fiabilitatea totală a sistemului. Arhitectura cea mai apropiată de acest deziderat, Ambient Intelligence (AmI), este proiectată pentru a fi utilizată într-un mediu potrivit pentru a fi controlat; prin urmare, principala sa țintă este casa „inteligentă” [WeRA05].

Într-un astfel de mediu, pe lângă senzori, actuatori și prelucrare multimedia, un alt participant ar putea apărea: PC-ul. Acest participant nu este luat în considerare în abordarea AmI.

Pe de altă parte, astăzi, conceptul de ambient multimedia este centrat pe un PC care controlează toată rețeaua multimedia. În cazul în care PC-ul central cade, întreaga rețea este blocată – nu mai funcționează.

Pentru a rezuma: în cadrul AmI, PC-ul nu este luat în considerare, iar în rețelele multimedia de astăzi PC-ul este elementul central, iar în același timp în aproape toate tipurile de rețele existente astăzi PC-ul este prezent.

Prin urmare, am încercat să adăugăm PC-ul la AmI, creând astfel o nouă arhitectură, în care tranzacțiile sunt bazate pe protocoale de consens și în care întâlnim toate cele trei metode menționate anterior: redundanța, coduri detectoare și corectoare de erori și algoritmi de consens.

Această arhitectură se va construi începând de la alegerea middleware-ului, continuând cu protocoale de comunicație care minimizează puterea consumată, ținând cont în primul rând de toleranța la defectare.

Redundanța, fiind în practică modul cel mai comun de atingere a toleranței la defectare, a fost abordată la mai multe niveluri, fiind chiar mixată cu un protocol de consens.

În încheierea părții experimentale se prezintă o aplicație a algoritmilor de consens în domeniul medical. Aceasta constă în aplicarea unui protocol de consens în determinarea unui rezultat final din date care se știu a fi măsurate cu un grad

ridicat de eroare. Rezultatul final va fi reprezentat de gradul de risc al unei boli, iar factorii luați în considerare pentru acea boală sunt datele inițiale.

1.2. Organizarea tezei

În capitolul 2 este prezentat conceptul de consens, precum și modelul matematic dezvoltat propriu. Înainte de a putea discuta despre algoritmi de consens se studiază diverse definiții ale acestuia, analizându-se avantajele și deficiențele acestora. Compilând toate aceste definiții, autorul va prezenta o propunere de definiție a termenului de consens, fundamentată pe un formalism matematic propriu.

Capitolul 3 prezintă aplicațiile teoretice ale consensului, în care se demonstrează că modelul matematic propus include toate aplicațiile clasice. Se demonstrează atât prin raționament intuitiv cât și prin raționament matematic, că principalele variante de a atinge consensul, folosite în știința calculatoarelor, aparțin clasei problemelor de consens în conformitate atât cu definiția clasică, dar și în conformitate cu definiția propusă.

De asemenea, se analizează alte clase de probleme care, în conformitate cu definiția propusă, aparțin clasei problemelor de consens, dar care, conform definiției clasice, nu aparțin acestei clase de probleme.

Demonstrația matematică arată că definiția propusă, este viabilă și chiar mai bună decât definițiile deja existente, putând descrie o plajă mult mai largă de probleme decât definițiile deja existente

În capitolul 4 este introdusă arhitectura de sistem distribuit originală, intelligent grid, împreună cu toate experimentele realizate pe ea.

Capitolul se deschide cu motivația dezvoltării noii arhitecturi, bazată pe arhitecturile de rețele actuale și evoluția lor, precum și arhitectura corespunzătoare celui de-al treilea val pentru a vedea ce tip de arhitectură ar fi fezabil pentru testarea modelului teoretic.

Capitolul 5 prezintă o aplicație a consensului în medicină, având scopul de a determina, cu ajutorul unor date simple și a unor calcule la fel de simple, un risc de măsură a osteoartrozei și pe osteoporozei, două probleme foarte frecvente în rândul persoanelor în vârstă realizând astfel un screening al populației. Evident, că există deja proceduri de screening dar acest studiu prezintă o nouă metodă de screening, creată prin personalizarea unor factori de risc pentru o anumită regiune din țară. Studiul de validare practică al formulei a fost efectuat la pe un lot de 129 de pacienți. Metodologia implicată în crearea formulei este un algoritm de consens și poate fi aplicată la orice alte tulburări, dar trebuie să fie ajustate pe fiecare, cum ar fi de exemplu pentru tulburări gastrice.

Capitolul 6 este dedicat discuțiilor finale și concluziilor, prezentând o sinteză a contribuțiilor aduse în cadrul tezei

2. Consensul

2.1. Definiția consensului

Înainte de a putea discuta despre consens, trebuie să îl definim, pentru a putea delimita exact ceea ce vrem să studiem. Interesant de notat, definiția consensului variază de la o sursă la alta:

În [DER62] termenul de consens (provenind de la latinescul consensus) este explicat ca fiind acord de voință sau înțelegere.

Această definiție, extrem de concisă, nu precizează nimic referitor la datele inițiale, la participanții din cadrul procesului de consens, sau la modul în care se atinge înțelegerea.

Pe de altă parte, în [Laro99] același termen este definit ca fiind acordul a mai multor persoane asupra unui sau mai multor texte.

În cadrul acestei definiții ni se precizează clar natura participanților la consens ("persoane"), natura datelor inițiale ("texte"), dar nu se precizează datele finale sau gradul de acceptare a acestora de către persoanele participante în dezbateri.

În [Webs00] definiția consensului este privită pe mai multe planuri, începând de la o înțelegere generală sau rezoluție ce a fost primită de toți / majoritatea celor vizați de ea și terminând cu solidarizarea unui grup în privința sentimentelor / credinței.

Mai complexă decât definițiile anterioare, apare în această definiție noțiunea de majoritate a participanților necesară pentru a atinge consensul, dar nu se face nici o presupunere referitoare la datele pe care se bazează acest algoritm. De notat că participanții nu mai sunt restrânși la a fi persoane, natura lor nefiind precizată explicit.

Observăm că adesea termenul de consens este echivalat (total sau parțial) cu cel de acord sau cu cel de înțelegere. Studiind definițiile acestor noțiuni, am ajuns la următoarea definiție dată în [DER62] și care este interesantă prin prisma ideii de combinație, exprimată explicit, cât și prin prisma aspectului temporal.

Acordul în domeniul muzicii este definit ca fiind combinații de două, trei, sau mai multe sunete de înălțimi diferite, produse fie simultan, fie succesiv, care împreună formează o unitate organică, deosebindu-se astfel de conglomeratele de sunete cu caracter accidental.

Compilând toate aceste definiții, în prezenta lucrare voi da următoarea propunere de definiție a termenului de consens [Ancu07]:

Consensul reprezintă rezultatul final al unei sau mai multor prelucrări realizate asupra unui șir de valori inițiale; rezultatul final fiind mai apoi recunoscut și însușit, la sfârșitul dezbaterilor, de către suficient de mulți participanți la prelucrări.

Pentru simplificare, această definiție o voi nota cu D1 și cu ajutorul acestei notații mă voi referi la ea în text.

Valorile inițiale ce sunt ulterior prelucrate, pot fi total sau parțial diferite, apărute simultan sau succesiv. De obicei, aparțin aceleași clase de valori. Valorile finale pot sau nu să aparțină aceleiași clase de valori ca și cele inițiale.

Idea prezentată în D1 este ilustrată extrem de simplu și intuitiv în figura 2.1.

Pornim de la 7 persoane care doresc să aleagă o culoare. Fiecare entitate are inițial o opinie proprie. În acest caz, fiecare opinie este una dintre culorile spectrului. Persoanele dezbate alegerea culorii și rezultă un proces de combinare, care în acest caz, amestecă în proporții egale toate opiniile.

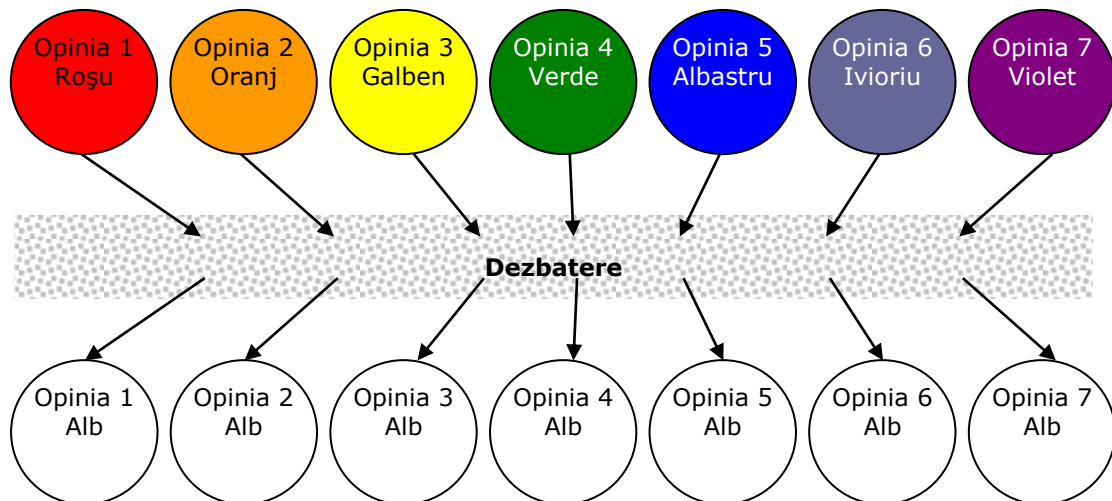


Figura 2.1. Ideea consensului

În final rezultă culoarea alb, care va deveni opinia finală pentru fiecare dintre entitățile inițiale. Aspectul temporal a fost neglijat în această discuție, pentru simplitate.

2.2. Abordarea socială

Pentru a analiza problema consensului o voi privi și prin prisma unui model deja existent și funcțional din lumea reală. Consensul este prezent la toate nivelurile de interacțiune interumană. Din punctul de vedere al psihologiei și al organizării societății umane, cel mai adesea nevoia de consens apare într-o echipă care trebuie să rezolve una sau mai multe probleme, legate de o anumită temă.

Atunci când lucrăm într-o echipă este extrem de important ca fiecare dintre membrii echipei să aibă un cuvânt de spus [w3co05]. Folosind simpla regulă a majorității 50%+1, în grupuri mici, opiniile posibil valoroase pot fi ușor excluse.

Psihologii recomandă respectarea următoarelor sugestii pentru a putea ajunge cât mai ușor la un consens benefic pentru toată lumea:

Aveți încredere în membrii grupului. Nu este o competiție.

Toată lumea pornește cu același coeficient de încredere

Fiți siguri că toată lumea a înțeles problema

Ontologie comună

Toți membrii grupului trebuie să-și aducă aportul la realizarea consensului

Un mesaj trebuie să atingă toți participanții

Fiecare membru trebuie să aibă dreptul de a nu fi de acord. Cu toate acestea, fiecare membru trebuie să fie flexibil la schimbări.

Flexibilitatea presupune capacitatea de a adopta orice rezoluție a grupului.

Nu vă legați de persoana care pune problema, ci doar de problemă.

Căderea linie de transport nu poate fi deosebită de căderea dispozitivului.

Grafic, acest proces este prezentat în Figura 2.2.

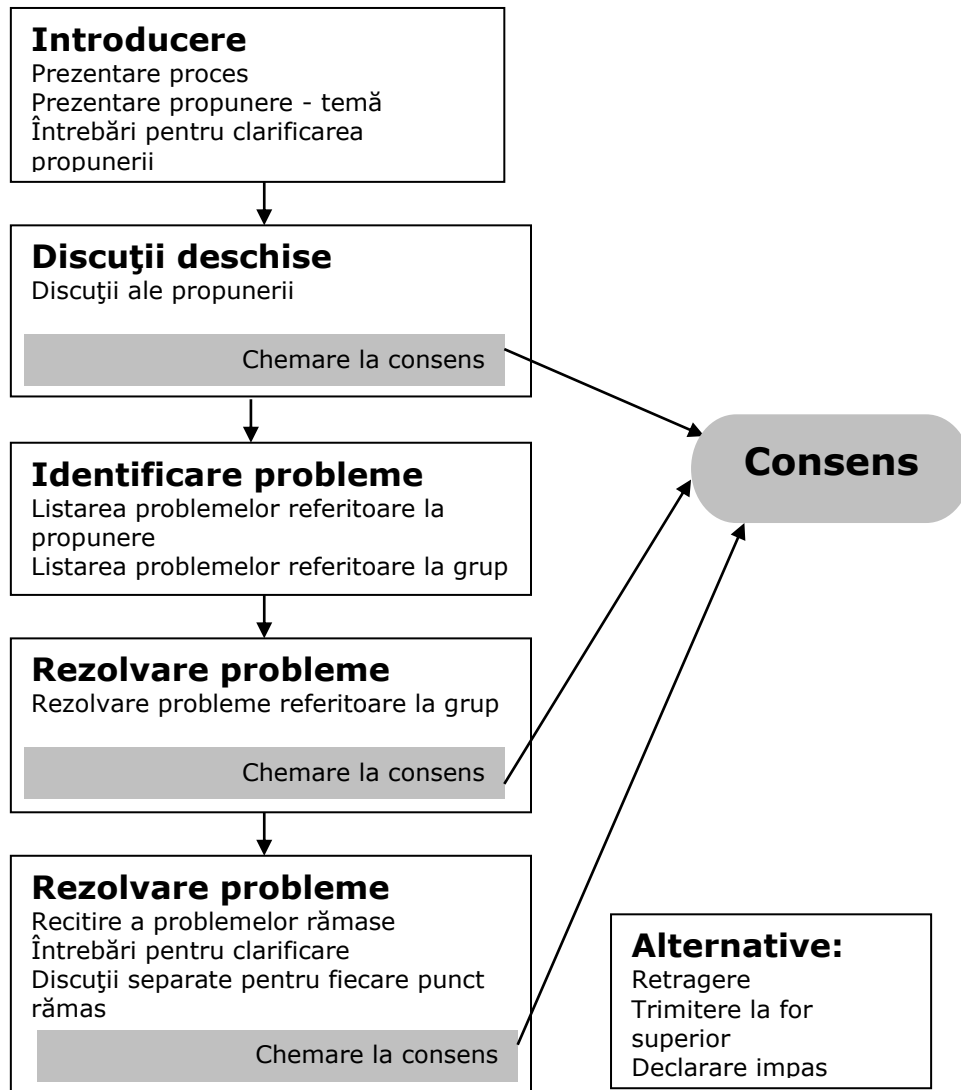


Figura 2.2. Procesul uman al atingerii consensului [w3co05]

Aș dori să remarc că pentru atingerea consensului final, este necesar, dar nu suficient, să se atingă un set de consensuri particulare. De asemenea este interesant de observat că fiecărui participant la consens i se permite să se retragă, în orice moment, iar la nivelul echipei este prevăzută imposibilitatea ajungerii la consens, dar aceasta nu implică în mod neapărat blocarea („declarare impas”) ci problema analizată poate fi redirecționată („trimitere la forul superior”).

Conform unor teorii recente [Fors05], consensul este baza funcționării societății umane. Atunci când nu se poate atinge consensul apar conflictele armate.

Un exemplu de mediu din lumea reală, în care acest algoritm se folosește cu succes în mod curent este parlamentul unei țări. Legile sunt prezentate, discutate,

sunt formulate amendamente dintre care unele sunt adoptate, iar altele nu sunt adoptate. Rezultatul final, este adoptat în urma unui vot, fiind recunoscut și însușit de către toți participanții la dezbateri. Și în cazul legilor, acestea pot fi retrase de la dezbateri, pot fi retrimise la comisiile de specialitate pentru a fi reevaluate și, foarte rar, poate fi declarat impasul.

2.3. Modele existente în știința calculatoarelor

Consensul este definit [MaKa05] în știința calculatoarelor ca fiind o problemă simetrică de ajungere la o înțelegere. Este astfel, o abstracție a unei clase de probleme în care procesele pornesc cu propriile lor opinii (posibil divergente) și trebuie să se pună de acord asupra uneia dintre ele.

Se fac următoarele presupuneri inițiale:

1. procesele trebuie să aibă suficient de multe stări inițiale (de ansamblu) care să le permită atingerea oricăreia dintre deciziile posibile.
2. se lucrează doar cu valori de 0 și 1 pentru a simplifica discuția.

Consensul din știința calculatoarelor se bazează pe ideea că avem un set de procese distribuite cu valori inițiale din $\{0,1\}$. Toate aceste procese trebuie să se decidă asupra aceleiași valori din $\{0,1\}$, bazându-se doar pe stările lor inițiale.

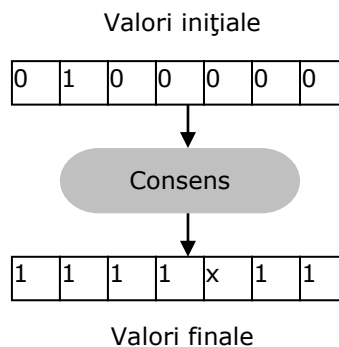


Figura 2.3. Consensul în calculatoare [MaKa05]

Condițiile (specificațiile) de corectitudine sunt:

Înțelegerea:

Nu există două procese care să se hotărească asupra unor valori diferite.

Validitatea:

Valoarea asupra căreia s-a decis, trebuie să fi fost valoarea inițială a cel puțin unui proces

Terminarea:

Fiecare proces corect se va decide în cele din urmă, irevocabil asupra unei valori..

Pentru ultima specificație, mai există o formulare, considerată „slabă, ușoară” (weak) deoarece reduce generalitatea variantei anterioare:

Terminarea „ușoară”:

Unele procese corecte se vor hotărâ în cele din urmă asupra unei valori, iar această decizie este irevocabilă

Pentru simplificare, voi nota cu D2 specificațiile de corectitudine prezentate mai sus și cu WD2 specificațiile de corectitudine incluzând terminarea „ușoară”.

În [KaDo98] se demonstrează că în cazul prezenței erorilor numai terminarea „ușoară” este posibilă.

Pentru a vizualiza mai ușor această problemă, putem să ne imaginăm o tranzacție implicând baze de date distribuite. Această tranzacție se efectuează fie în toate locațiile participante, fie în nici una dintre aceste locații.

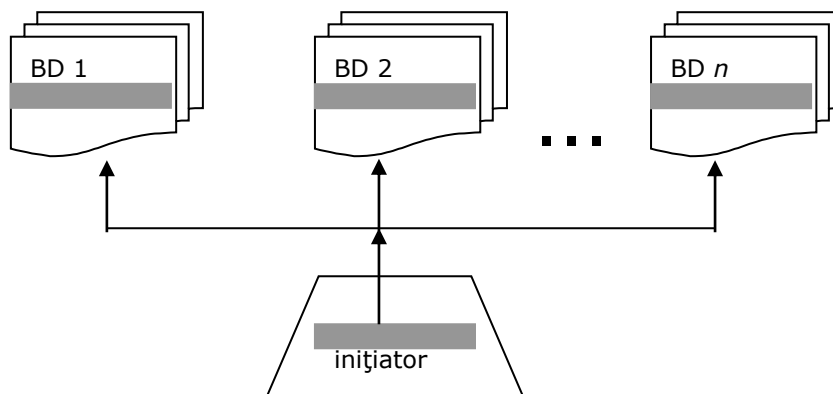


Figura 2.4. Tranzacție bazată pe consens la baze de date distribuite

Un prim proces va iniția o cerere de modificare a bazelor de date, referitoare la o anumită valoare.

Fiecare bază de date are o opinie inițială: “accept tranzacția” sau “refuz tranzacția”.

În urma procesului de consens, fiecare instanță a bazei de date va ajunge la o opinie finală irevocabilă: “nu efectuez tranzacția” sau, dimpotrivă, “încep să efectuez tranzacția”

2.3.1. Consensul în sisteme sincrone

Specific sistemelor sincrone este întârzierea constantă a mesajelor și timpul constant de execuție a proceselor. Algoritmii de consens sunt simpli, rulează în runde sincrone în care trimite mesaje, așteaptă un timp fixat să vadă ce mesaje primește și apoi procesează local aceste informații.

Este modalitatea cea mai simplă de consens și are soluție întotdeauna dacă numărul căderilor nu depășește anumite limite [LaFi84]. Din păcate, aceste sisteme nu se prea întâlnesc în lumea reală.

În știința calculatoarelor, marea majoritate a modelelor de consens ([LaPS80], [DoSt83], [Schn84],[MaSc85], [DLPS86], [Brac87], [HaMW90]) presupune existența sincronității.

2.3.2. Consensul în sisteme parțial sincrone

Un model mai apropiat de lumea reală este cel al sistemelor parțial sincrone. În acest tip de sisteme [DwLS88] există un timp global de stabilizare. Înainte de acest moment de stabilizare global se consideră că sistemul este asincron, iar după momentul de stabilizare global este sincron.

Momentul de stabilizare nu este cunoscut de la începerea algoritmului.

În acest caz, consensul are soluție dacă nu sunt mai mult de jumătate din procese defecte, dar timpul de rulare maxim rămâne nelimitat [FiLP85] deoarece procesele sunt asincrone pentru un timp nelimitat.

2.3.3. Consensul în sisteme asincrone

Fisher, Lynch și Paterson [FiLP85] au demonstrat încă din anul 1985 imposibilitatea rezolvării deterministe a problemei consensului în sisteme asincrone unde apar procese eronate.

De ce ne interesează atunci atât de mult aceste sisteme? Răspunsul îl constituie în primul rând atractivitatea sistemelor asincrone (nu trebuie să facem nici un fel de presupuneri legate de timp) precum și răspândirea acestor sisteme în natură.

Un exemplu, destul de sugestiv, îl constituie așa-numita problemă a „omului chel”. Considerând că un om este chel dacă are mai puțin de 1000 de fire de păr, se ridică următoarea problemă: dacă omul poartă o pălărie și sunt vizibile 999 de fire de păr, putem determina dacă el este chel?



Figura 2.5. Paradoxul omului chel [MaKa05]

Revenind la o notă mai serioasă, acest paradox se traduce astfel: este imposibil, la un moment dat, pentru un sistem corect să distingă două configurații, fără a cunoaște valorile inițiale ale proceselor care nu răspund. Astfel indiferent de funcționarea sa corectă, el va alege aceeași decizie în cazul ambelor configurații, rezultând conflictul și imposibilitatea atingerii consensului

Un alt exemplu al implicațiilor acestui rezultat îl constituie sistemele asincrone cu memorii partajate. Presupunem că într-o locație de memorie se află una sau mai multe variabile a căror valori trebuie a fi modificate de către mai multe procese. Cum ne putem asigura că variabila noastră va avea valoarea potrivită la momentul potrivit?

Evident, o primă soluție o constituie un consens al proceselor asupra valorii variabilei respective. Dar din [TaWo01] știm că o altă soluție referitoare la accesul zonelor de memorie partajate sunt tranzacțiile atomice - de exemplu, semafoare.

Putem compara rezultatele de imposibilitate dintre tranzacții atomice și consensul în mediul asincron:

Condiții de insolvabilitate	
Tranzacții atomice	Consens asincron
(multe) căderi ale legăturilor	e de ajuns o singură cădere a unui proces
chiar dacă sistemul e sincron	numai dacă sistemul e asincron
cauza : imposibilitatea comunicării	cauza : incertitudinea
demonstrație ușoară	demonstrație mai dificilă

Tabelul 2.1. Condiții de insolvabilitate [MaKa05]

Procesele verifică valoarea semaforului, o modifică și e foarte posibil să treacă în modul „sleep” atunci când totul s-a terminat. Toate aceste acțiuni însă se realizează într-o singură acțiune mare, indivizibilă: acțiune atomică. Atomicitatea este absolut esențială pentru a putea rezolva problemele de sincronizare și pentru a evita concurența la acces. Este evident că tranzacțiile atomice nu aparțin clasei problemelor de consens, așa cum este ea definită de D2, respectiv WD2 și nici de către D1.

2.4. Modelul matematic al consensului

Voi introduce o serie de definiții de funcții și restricții care în final se vor completa dând la iveală funcția de consens[Ancu07].

Definim o funcție, care descrie un proces real fizic, funcția ψ (psi) astfel:

$$\psi : V_i^n \times R^+ \rightarrow V_f^n \quad (a)$$

$$\psi \left(\begin{bmatrix} x_0(0) \\ x_1(0) \\ \vdots \\ x_{n-1}(0) \\ t \end{bmatrix} \right) = \begin{bmatrix} y_0 \\ y_1 \\ \vdots \\ y_{n-1} \end{bmatrix} \quad (b)$$

Ecuția 2.1

unde

V_i – mulțimea valorilor inițiale

V_f – mulțimea valorilor finale

n – numărul total de sisteme participante (la consens)

t – timpul

$x_i(0)$ – valoarea inițială pe care sistemul i o supune dezbaterii, $i \in \{0, 1, 2, \dots, n-1\}$

Ecuția 2.1b devine:

$$X^* = \begin{bmatrix} x_0(0) \\ x_1(0) \\ x_{n-1}(0) \\ t \end{bmatrix} \quad Y = \begin{bmatrix} y_0 \\ y_1 \\ y_{n-1} \end{bmatrix} \quad (a)$$

$$\Rightarrow \quad (b) \\ \psi(X^*) = Y$$

Ecuția 2.2

Fie ν (niu) o funcție denumită frecvența de apariție, care măsoară numărul de apariții a unei valori oarecare k într-un vector K aparținând mulțimii valorilor finale.

$$\nu: V_f \times V_f^n \rightarrow N$$

$$\nu(k, K) = n_k$$

Ecuția 2.3

unde

N – mulțimea numerelor naturale

Aplicăm și maximizăm această funcție asupra vectorului de ieșire al funcției ψ definite anterior în Ecuția 2.1, și notat Y :

$$\max(\nu(y_i, Y), i = \overline{0, n-1}) = \Theta$$

Ecuția 2.4

unde

Θ – reprezintă numărul maxim de apariții al oricărei valori în vectorul de ieșire, Y , al funcției ψ .

Definim funcția majoritate – *maj* – astfel:

$$maj: V_f^n \rightarrow V_f \quad (a)$$

$$maj = k_j, \text{ unde } \nu(k_j, K) \geq M \quad (b)$$

$$j \in \{0, 1, \dots, n-1\}; \quad M \geq \left\lceil \frac{n}{2} + 1 \right\rceil \quad (c)$$

Ecuția 2.5

unde

M – factor de majoritate.

Pentru majoritatea de 50% +1 este evident că $M = \lceil n/2 + 1 \rceil$. De altfel, $50\% + 1$ este pragul minim pentru care funcția majoritate este funcțională.

Dacă $M < \lceil n/2 + 1 \rceil$ pot exista două valori distincte, deci funcția majoritate nu are o singură soluție.

Există însă unele cazuri în care putem coborâ pragul aceasta nemodificând funcția majoritate.

Numim funcția ψ funcție de consens dacă:

$$\Theta \geq M$$

Ecuția 2.6

Altfel spus, funcția de consens are ca și parte a rezultatului final al său un număr de valori identice mai mare decât un anumit prag.

Concluzionând, funcția de consens se descrie cumulat astfel:

$$\psi : V_i^n \times R^+ \rightarrow V_f^n$$

$$\psi \left(\begin{bmatrix} x_0(0) \\ x_1(0) \\ \vdots \\ x_{n-1}(0) \\ t \end{bmatrix} \right) = \begin{bmatrix} y_0 \\ y_1 \\ \vdots \\ y_{n-1} \end{bmatrix}$$

$$\max(v(y_i, Y), i = \overline{0, n-1}) = \Theta$$

$$\Theta \geq M$$

Ecuția 2.7

Această funcție este oglindirea matematică a definiției D1.

2.5. Direcții de dezvoltare

Consensul este o problemă curent întâlnită în toate aspectele vieții, precum și în foarte multe din aspectele ce implică sistemele de calcul.

Această problemă a fost studiată în știința calculatoarelor deocamdată exclusiv din punctul de vedere al acestei discipline.

Teza de doctorat prezentă va încerca să realizeze o abordare multi-disciplinară, folosind modele și rezultate din lumea reală, experimente din domeniul calculatoarelor, dar bazându-se pe un formalism matematic unificator.

În următorul capitol voi prezenta principalele variante de a atinge consensul, folosite în știința calculatoarelor. Voi demonstra prin raționament intuitiv, că aceste variante aparțin clasei problemelor de consens în conformitate atât cu definiția oficială D2, dar și în conformitate cu definiția propusă D1.

De asemenea, voi analiza alte clase de probleme care, în conformitate cu definiția D1, aparțin clasei problemelor de consens, dar care, conform lui D2, nu aparțin acestei clase de probleme.

Pentru toate aceste probleme se vor prezenta utilizările practice, acestea dictând, în cele din urmă, importanța problemelor deci și nivelul de interes în rezolvarea cât mai performantă a acestora.

3. Cazul particular al consensului în știința calculatoarelor

3.1. Problema generalilor bizantini (BGP)

Problema generalilor bizantini a apărut pentru prima dată în 1980, în articolul "Reaching Agreement in the Presence of Faults" [LaPS80]. Înainte de apariția acestui articol concepția generală era că un sistem de 3 procesoare poate tolera doar un singur procesor care funcționa incorect.

Această lucrare demonstra că erorile "bizantine" puteau să nu fie detectate de către un algoritm tradițional. Problema a fost descrisă prin următorul set de propoziții:

- BG1 Fiecare divizie a armatei bizantine este comandată de către propriul său general.
- BG2 Generalii, dintre care unii sunt trădători, comunică între ei prin mesageri.
- BG3 Toți generalii loiali trebuie să se decidă asupra aceluiași plan de acțiune
- BG4 Un număr mic de trădători nu poate cauza adoptarea unui plan greșit.

Din aceasta rezultă [LaPS80] următoarele propoziții, echivalente cu **BG3** și

BG4:

- BGR3 Toți generalii loiali primesc aceleași informații pe baza cărora vor lua decizia finală.
- BGR4 Informațiile trimise de către generalii loiali trebuie să fie folosite de către toți generalii loiali.

Problema clasică poate fi redusă la problema unui comandant general și a mai multor locotenenți [LaPS80], care se formulează astfel:

- CBG1 Toți locotenenții trebuie să urmeze aceleași ordine.
- CBG2 Dacă comandantul general este loial, atunci fiecare dintre ceilalți locotenenți loiali ascultă de ordinele trimise de acesta.

În 1982, a apărut o primă variație, a problemei generalilor bizantini, numită problema generalilor bizantini relaxată [LaFi84]. Aceasta este descrisă prin următorul set de propoziții:

- WBG1 Dacă nici un general nu trădează și toți generalii doresc inițial aceeași acțiune, atunci aceasta va fi acțiunea aleasă de către generali
- WBG2 Oricare doi generali loiali aleg aceeași acțiune finală.

Oricare dintre aceste probleme are la rândul ei două variante: cu mesaje scrise (cu semnătură nefalsificabilă) sau cu mesaje orale (fără semnătură). Și astfel începe odiseea generalilor bizantini...

3.1.1. Utilizare practică și posibile beneficii

Devine evident faptul că dacă înlocuim cuvântul "general" cu cuvântul "sistem" obținem o problemă interesantă din punctul de vedere al științei calculatoarelor. Unul dintre inițiatorii acestei probleme [Lamp] a folosit acest

paradigm pentru a soluționa accese la bazele de date [LaFi84] sau pentru a sincroniza ceasurile calculatoarelor [LaMS84, LaMS85]. Această ultimă soluție a fost sponsorizată de către NASA.

Dar nu numai NASA a folosit această problemă. Firma Boeing [Lamp] (începând din 1986) își creează avioanele folosind soluția BGP. De asemenea, firma McDonnell - Douglas a lucrat începând din anii 90 la sisteme care să tolereze erori bizantine.

BGP a fost folosită, de obicei în aplicații cu cerințe ridicate de securitate, pentru a soluționa accese la baze de date distribuite [KaDo98, MaKa05] sau la memorii partajate [GrLa04, MaKa05].

3.1.2. Apartenența la clasa problemelor de consens

În conformitate cu D1, BGP aparține clasei problemelor de consens.

Setul de valori inițiale este menționat explicit în BGR3. Faptul că pe „pe baza acelorși informații se va lua decizia finală” presupune existența unor prelucrări a acestor date. Rezultatul final este recunoscut și însușit în virtutea propoziției BG3, care formulează clar acest lucru.

Se impune o observație care se referă la ușurința cu care D1 mapează BGP. Este aproape evident și natural modul în care propozițiile ce definesc BGP se regăsesc în definiția propusă D1, dar fără a restrânge generalitatea definiției.

D2 nu se aplică, deoarece prin definiție (BG2) avem erori. Folosim atunci WD2, după cum urmează:

Înțelegerea este asigurată de presupunerea alegerii aceluiaș plan de acțiune pentru toți generalii (BG3).

Validitatea și terminarea nu sunt exprimate explicit, dar ele rezultă din context. Validitatea rezultă din presupunerea că toate planurile de acțiune sunt de fapt numai două: „atac” și „retragere”.

Terminarea este exprimată prin ideea de „decizie finală” dar, repet, nu este exprimată explicit. O altă indicație că terminarea este satisfăcută e BG4. Dar deoarece doar un număr mic de trădători nu poate împiedeca atingerea deciziei finale comune, rezultă că clauza de terminare este restrânsă.

E interesant de notat faptul că dacă planurile de atac nu sunt prestabilite, cum ar fi de exemplu, într-un caz real, în care strategiile sunt practic infinite, în care caz BGP ar putea să nu se conformeze definiției WD2, deoarece un plan de atac mai bun ar putea rezulta din combinarea a unul sau mai multe planuri de atac propuse de generali. Această opțiune de prelucrare nu este oferită de WD2, restricția fiind regăsită în însăși clauza de validitate.

Faptul că WBGP aparține clasei problemelor de consens, atât după definiția D1, cât și după definiția WD2 rezultă din includerea problemei în BGP. Condițiile particulare, specifice acestei variațiuni, adică posibilitatea inexistenței erorilor, determină acoperirea, dar numai în acest caz, a problemei și de către D2, nu doar de WD2 ca și în cazul BGP clasic.

CPGP aparține clasei problemelor de consens, atât după definiția D1, cât și după WD2 datorită includerii în BGP. Condițiile particulare, în speță faptul că există un singur inițiator de mesaje inițiale (comandantul) nu alterează cu nimic problema deoarece fiecare general are implementată o valoare inițială proprie.

3.1.3. Modelul matematic al BGP

Definiția specifică domeniului calculatoarelor, D2 este practic un caz particular al definiției propuse D1. Doresc să demonstrez matematic această afirmație, drept pentru care pornind de la funcția de consens arăt care sunt ipotezele particulare care modelează definiția D2. Fiecare dintre propozițiile ce alcătuiesc D2 determină anumite restricții.

Înțelegerea este asigurată prin următoarea condiție:

$$y_0 = y_1 = \dots = y_{n-1} = y$$

Ecuția 3.1

Prin această condiție se îndeplinește automat funcția de majoritate, deci nu mai e necesară:

$$\max(v(y_i, Y), i = \overline{0, n-1}) = \Theta = n; \quad n \geq M$$

Ecuția 3.2

Validitatea este asigurată prin următoarele condiții:

$$V_f = V_i = V \tag{a}$$

$$\exists x_i(0) = y; i = \overline{0, n-1} \tag{b}$$

Ecuția 3.3

Condiția de terminare implică, de asemenea, modificarea mulțimilor de definiție, prin restricționarea intervalului în care timpul are voie să ia valori; în speță timpul trebuie să fie finit.

$$\psi : V^n \times [0, t_{final}] \rightarrow V^n \tag{a}$$

$$t_{final} \neq \infty \tag{b}$$

Ecuția 3.4

Combinând toate aceste informații, funcția de consens obținută din particularizarea funcției de bază Ecuția 2.7 cu toate ipotezele Ecuția 3.1, Ecuția 3.2, Ecuția 3.3 și Ecuția 3.4 conform D2, este:

$$\psi : V^n \times [0, t_{final}] \rightarrow V^n$$

$$\psi \left(\begin{bmatrix} x_0(0) \\ x_1(0) \\ \vdots \\ x_{n-1}(0) \\ t \end{bmatrix} \right) = \begin{bmatrix} y \\ \vdots \\ y \end{bmatrix}$$

$$\exists x_i(0) = y; i = \overline{0, n-1}$$

$$t_{final} \neq \infty$$

Ecuția 3.5

Pentru a satisface WD2, trebuie ca nu toate valorile finale să fie identice, deci trebuie să folosim funcția majoritate. Astfel, în cadrul procesului de prelucrare, la momentul în care numărul minim de procese e atins sau dacă s-a terminat timpul, procesul se încheie și funcția trebuie să aibă un rezultat. Valorile celorlalte procese care nu au atins consensul sunt fie valorile care le trimit acestea, fie, dacă nu există, sunt reprezentate printr-un element neutru.

A se nota că aceste observații pe moment nu ne interesează deoarece țin de modul în care se face prelucrarea. Singurul lucru care ne interesează este să avem un vector de n valori la ieșirea funcției de consens.

Astfel, funcția de consens este:

$$\psi : V^n \times [0, t_{final}] \rightarrow V^n$$

$$\psi \left(\begin{bmatrix} x_0(0) \\ x_1(0) \\ \vdots \\ x_{n-1}(0) \\ t \end{bmatrix} \right) = \begin{bmatrix} y_0 \\ y_1 \\ \vdots \\ y_{n-1} \end{bmatrix}$$

$$\exists x_i(0) = y; i = \overline{0, n-1}$$

$$t_{final} \neq \infty$$

$$\max(v(y, Y), i = \overline{0, n-1}) = \Theta$$

$$\Theta \geq M$$

Ecuția 3.6

Această funcție este oglindirea matematică a definiției D2 și este provenită din funcția ce descrie D1, diferența constând în restricțiile aplicate.

3.1.4. Tipuri de mesaje în cadrul BGP

Am menționat că generalii comunică între ei prin mesageri (**BG2**). Mesajele pe care le duc acești mesageri pot fi orale sau scrise.

Mesajele orale au ca și corespondent real, un mesager uman căruia generalul său îi încredințează un mesaj vorbit și îi spune cui trebuie să-i dezvăluie conținutul mesajului.

Mesajele scrise au ca și corespondent real un mesager căruia generalul său îi încredințează un pergament scris și autentificat cu o pecete pe care doar generalul o posedă și pe care nimeni nu o poate falsifica, dar pe care orice general o recunoaște.

Evident mesagerul poate fi capturat de către inamic sau poate să trădeze (în cadrul mesajelor orale putând chiar schimba mesajul).

În știința calculatoarelor, mesajele sunt pachetele trimise într-o rețea, sau mesajele schimbate între firele de execuție, sau orice modalitate de comunicare între două sisteme, iar „pecetea” devine o semnătură digitală.

3.1.4.1 Mesaje orale

Mesajele orale implică următoarele :

- OMH1** Fiecare mesaj trimis este receptat corect. Căderea liniilor de comunicații între două componente nu se distinge față de căderea uneia dintre componente. Căderea liniei mai adaugă doar încă un trădător.
- OMH2** Cel care primește mesajul știe cine l-a trimis. Avem deci o rețea sub forma unui graf tare conex, cu fire de comunicație ideale.
- OMH3** Absența unui mesaj poate fi detectată. Acest lucru se poate realiza doar în sistemele sincrone, cu ajutorul unui mecanism de time-out.

Ca și avantaje avem lipsa necesității unei semnături nefalsificabile, precum și posibilitatea reducerii numărului de runde.

Unul din dezavantajele mesajelor orale este numărul relativ redus de trădători (mai puțin de o treime din numărul total de generali) pentru care problema are soluție corectă.

Algoritmul descris de [Gupt99] pentru acest caz presupune că:

- Se folosește o funcție majoritate care primește un set de valori și returnează valoarea majoritară dintre ele.
- Se folosesc runde în care fiecare general trimite valorile pe care le-a primit în runda anterioară tuturor generalilor care nu le primiseră înaintea sa.
- La sfârșitul unei runde, pentru oricare doi generali loiali vectorul valorilor primite este același.

Demonstrație

Prin ipoteza problemei (**BG4**) este asigurată majoritatea generalilor loiali.

La început aceștia au aceeași valoare inițială.

Aplicând funcția majoritate, în prima rundă, se va obține valoarea comună care va fi noua valoare a tuturor generalilor loiali. Acest mers al lucrurilor se repetă și în rundele următoare.

Dacă ne aflăm în cazul comandantului general cu locotenenți, dacă comandantul este loial atunci problema are soluție în 2 runde. În caz contrar [Kest95], pot fi necesare până la $m+1$ runde, unde m este numărul de trădători.

Concluzionând acest algoritm recursiv poate fi scris astfel:

Locotenenții trimit ordinele primite celorlalți locotenenți.

Comandantul ordonă astfel:

majoritate ($v(c), v(1), v(2), \dots, v(n)$)

$v(i) = \text{majoritate}(v(i), v(i)(2), v(i)(3), \dots, v(i)(n)), 1 \leq i \leq n$

$v(i)(j) = \text{majoritate}(v(i)(j), v(i)(j)(3), v(i)(j)(4), \dots)$

...

Exemplu Să considerăm cazul unui comandat și a doi locotenenți.

Dacă comandantul este trădător, la începutul primei runde el trimite atacă unui locotenent și așteaptă celuilalt. Cei doi locotenenți fac schimb de mesaje.

Privind situația din punctul de vedere al locotenentului #1, el primește următoarele mesaje:

Direct de la comandat: Comandantul spune atacă.

De la locotenentul #2: Comandantul spune așteaptă.

Astfel locotenentul #1 nu știe ce să facă.

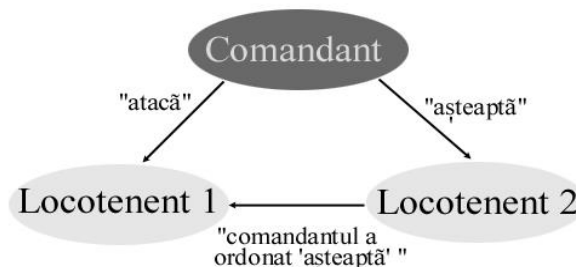


Figura 3.1. Comandantul este trădător [LaPS80]

Dacă unul dintre locotenenți este trădător, la începutul primei runde comandantul trimite atacă ambilor locotenenți. Cei doi locotenenți fac schimb de mesaje, dar locotenentul 2 trădează și trimite așteaptă.

Privind situația din punctul de vedere al locotenentului #1, el primește următoarele mesaje:

Direct de la comandat: Comandantul spune atacă.

De la locotenentul #2: Comandantul spune așteaptă.

Astfel locotenentul #1 nu știe ce să facă, din nou.

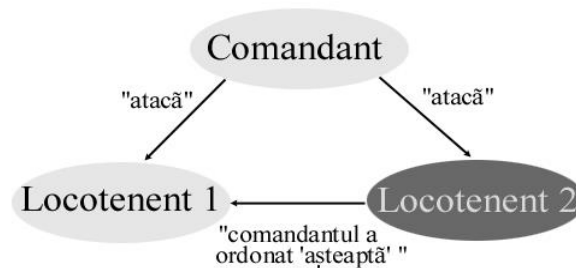


Figura 3.2. Locotenentul 2 este trădător [LaPS80]

Totuși, dacă ceea ce spune oricare dintre participanți ar putea să nu fie modificat de către următorul, atunci problema ar avea soluție. Acest lucru se rezolvă însă cu ajutorul mesajelor scrise.

3.1.4.2 Mesaje scrise

Mesajele scrise adaugă următoarele cerințe, pentru o bună funcționare a algoritmului [Huan99]:

- WMH1** Semnătura unui general loial nu poate fi modificată . Rezultă deci că odată depus un mesaj acesta nu poate fi modificat.
- WMH2** Oricine poate verifica autenticitatea semnăturii

Nu se face nici o presupunere referitoare la semnăturile generalilor trădători (în speță, este posibil chiar ca un general trădător să falsifice semnătura altui general trădător)

O schiță a algoritmului pentru $n-1$ locotenenți și un comandant general este dată de [Gupt99, Huan99]:

- Comandantul trimite ordine semnate la toți locotenenții.
- Dacă un locotenent primește un ordin de la cineva, îl pune într-un set V , dacă nu se află deja în el. De asemenea, dacă sunt mai puțin decât m semnături pe ordin, îl trimite mai departe, lăsându-și însă semnătura pe ordin.
- Toată lumea se oprește la runda $m+2$ și folosindu-se de V realizează acțiunea dorită.

Acest algoritm presupune [Gupt99] că numărul maxim de trădători este cunoscut.

Demonstrație

Dacă comandantul este loial, atunci toți locotenenții loiali primesc ordinul corect deja de la runda 1 și datorită imposibilității falsificării semnăturii nici un alt mesaj nu mai poate apărea.

Dacă comandantul nu este loial atunci la runda $m+1$ cel puțin un locotenent loial a primit ordinul până la runda m (pentru că sunt maxim m trădători). Și atunci, locotenentul loial îl va trimite tuturor. Ideea este că dacă un locotenent loial primește un ordin, toți locotenenții loiali îl vor primi în runda următoare.

În practică, mesajele sunt semnate prin tehnici criptografice.

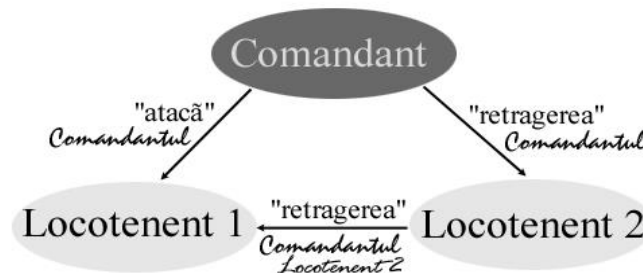


Figura 3.3. Comandantul este trădător; mesaje scrise prelucrare [LaPS80]

Exemplu:

Reconsiderând exemplul anterior, dacă comandantul trădează locotenentul #1 va primi acum următoarele mesaje:

Direct de la comandat:

Comandatul spune atacă. Semnat: Comandantul

De la locotenentul #2:

Comandantul spune așteaptă. Semnat: Comandantul, Locotenentul #2

Locotenentul #1 realizează astfel că comandantul a semnat două ordine contrarii : atacă și așteaptă, acest lucru catalogându-l drept trădător.

Dacă însă locotenentul #1 nu primește nimic de la locotenentul #2 atunci acesta este considerat trădător și ordinul comandantului rămâne valabil.

3.1.5. CBGP varianta cu mesaje orale

3.1.5.1 Algoritmul

Algoritmul CBGP presupune:

1. existența unei valori notată cu v_{def} implementată în cadrul fiecărui general (de ex: *așteaptă*)
2. existența unei funcții numită *majoritate* definită astfel:

$$majoritate(v_1, v_2, \dots, v_{n-1}) = v, \text{ dacă majoritatea valorilor variabilelor } v_i = v$$

$$majoritate(v_1, v_2, \dots, v_{n-1}) = v_{def}, \text{ dacă majoritatea valorilor variabilelor } v_i \neq v$$

Algoritmul **OM**($n, 0$) - nu există trădători

1. Comandatul trimite valoarea v tuturor locotenenților
2. Fiecare locotenent folosește valoarea trimisă de comandant, sau v_{def} în cazul în care nu a primit nici o valoare.

Algoritmul **OM**(n, m) - există m trădători, $m > 0$

1. Comandatul trimite valoarea v tuturor locotenenților
2. Pentru fiecare locotenent i
 - v_i = valoarea primită de la comandant sau v_{def} dacă nu a primit nici o valoare.
 - trimite valoarea v_i la restul de $n-2$ locotenenți folosind **OM**($n-1, m-1$)
 - Notă: în aplicarea lui **OM**($n-1, m-1$) locotenentul i este pe post de comandant
3. Pentru fiecare i și fiecare $j, j \neq i$
 - v_j = valoarea primită de locotenentul i de la locotenentul j (trimisă la pasul 2)
 - sau
 - $v_j = v_{def}$ dacă nu a primit nici o valoare
 - Locotenentul i folosește valoarea dată de funcția *majoritate* (v_1, v_2, \dots, v_{n-1})

Exemplu:

Să considerăm cazul a 4 generali, 1 trădător.

Locotenentul 3 îl considerăm trădător.

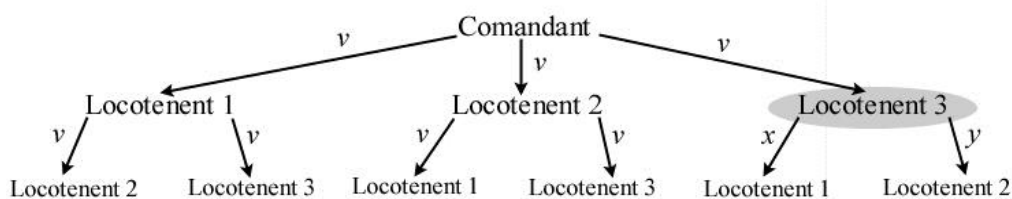


Figura 3.4.CBG, mesaje orale, locotenentul 3 trădător [LaPS80]

La sfârșitul pasului 1:

Locotenentul 1: $v_1=v$
 Locotenentul 2: $v_2=v$
 Locotenentul 3: $v_3=v$

La sfârșitul pasului 2

Fiecare locotenent a primit un set de valori și a ajuns la aceeași decizie (**CBG1**), valoarea trimisă de comandant este cea decisă de majoritate(**CBG2**).

La sfârșitul pasului 3:

Locotenentul 1: $v_1=v; v_2=v; v_3=x$
 Locotenentul 2: $v_1=v; v_2=v; v_3=y$
 Locotenentul 3: $v_1=v; v_2=v; v_3=v$

Dacă considerăm însă comandantul ca fiind trădătorul, avem situația:

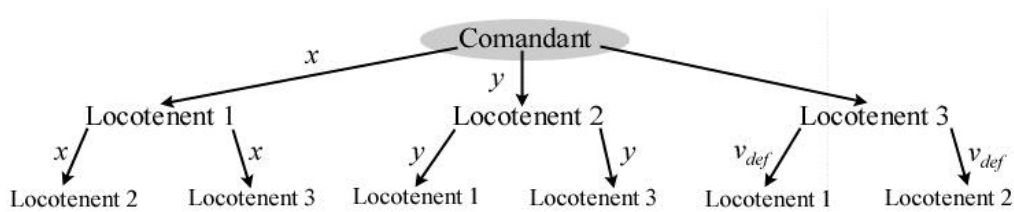


Figura 3.5.CBG, mesaje orale, comandantul e trădător [LaPS80]

La sfârșitul pasului 1:

Locotenentul 1: $v_1=x$
 Locotenentul 2: $v_2=y$
 Locotenentul 3: $v_3=v_{def}$

La sfârșitul pasului 3:

Locotenentul 1: $v_1=x$; $v_2=y$; $v_3= v_{def}$
 Locotenentul 2: $v_1=x$; $v_2=y$; $v_3= v_{def}$
 Locotenentul 3: $v_1=x$; $v_2=y$; $v_3= v_{def}$

Concluzie: Toți cei trei locotenenti loiali ajung la aceeași decizie, în ambele cazuri.

3.1.5.2 Corectitudinea algoritmului:

Lema 1

Oricare ar fi m și k , $OM(n,m)$ satisface **CBG2** dacă $n > 2k+m$ și sunt cel mult k trădători.

Demonstrație

Pentru a rezolva această demonstrație vom apela la inducția matematică

1. Din **OMH1** este evident faptul că $OM(n,0)$ funcționează dacă comandantul este loial, deci $OM(n,0)$ satisface **CBG2**.

2. Presupunem că $OM(n-1,m-1)$ satisface **CBG2** pentru $m > 0$.

Vrem să demonstrăm că $OM(n,m)$ satisface **CBG2**.

În pasul 1 al algoritmului comandantul loial trimite valoarea v la cei $n-1$ locotenenti.

În pasul 2 al algoritmului fiecare locotenent loial aplică $OM(n-1, m-1)$.

Din ipoteză avem că $n > 2k+m$, ceea ce este echivalent cu :

$$n-1 > 2k+(m-1)$$

Ecuția 3.7

Putem atunci aplica ipoteza din pasul 2 al demonstrației pe bază de inducție: $OM(n-1,m-1)$ satisface **CBG2** pentru $m > 0$. Aceasta ne indică că fiecare locotenent loial i primește $v_j=v$ de la toți locotenentii loiali j , $j \neq i$.

Din moment ce sunt cel mult k trădători și $m > 0$, inecuația (4-1) se poate rescrie sub forma următoare:

$$n-1 > 2k \Leftrightarrow k < \frac{n-1}{2}$$

Ecuția 3.8

În pasul 3 al algoritmului, atunci când apelează funcția majoritate, locotenentul i va avea $v_j=v$ de la toți locotenenții loiali j , care conform (4-2) sunt majoritari, deci va obține:

$$\text{majoritate}(v_1, v_2, \dots, v_{n-1}) = v$$

Ecuția 3.9

ceea ce satisface **CBG2**.

Teorema 2

Oricare ar fi m , **OM**(n, m) satisface **CBG1** și **CBG2** dacă sunt maxim m trădători și $n > 3m$.

Demonstrație

Pentru a rezolva această demonstrație vom apela la inducția matematică

1. Dacă $m=0$, folosind **OMH1** rezultă că **OM**($n, 0$) satisface **CBG1** și **CBG2**.
2. Presupunem că **OM**($n-1, m-1$) satisface teorema.

Vrem să demonstrăm că **OM**(n, m) satisface teorema.

Cazul 1

Comandantul este loial.

Considerând $k=m$ și aplicând **lema 1**, **OM**(n, m) satisface **CBG2**. Din moment ce **CBG1** rezultă din **CBG2** dacă comandantul este loial, problema este rezolvată.

Cazul 2

Comandantul este trădător.

Din moment ce sunt maxim m trădători, înseamnă că $m-1$ locotenenți sunt trădători.

Aplicând algoritmul, în pasul 2 se apelează **OM**($n-1, m-1$) și deoarece **OM**($n-1, m-1$) satisface **CBG1** în pasul 3 pentru fiecare j oricare doi locotenenți loial primesc aceeași valoare v_j , deci vor avea aceeași funcție *majoritate* (v_1, v_2, \dots, v_{n-1}).

Din ipotezele teoremei, mai vrem să demonstrăm:

$$n > 3m \Leftrightarrow n-1 > 3m-1$$

Ecuția 3.10

Deoarece am presupus că, că **OM**($n-1, m-1$) satisface teorema, avem:

$$n-1 > 3(m-1)$$

Ecuția 3.11

Din relațiile anterioare obținem următorul adevăr:

$$3^{m-1} > 3(m-1)$$

Ecuția 3.12

Astfel și această ultimă condiție a teoremei este satisfăcută.

3.1.5.3 Complexitatea algoritmului:

Aplicarea algoritmului $OM(n, m)$ cauzează lansarea a $n-1$ mesaje. Fiecare mesaj invocă $OM(n-1, m-1)$ care cauzează lansarea a $n-2$ mesaje, ș.a.m.d.. Sumarizând:

Etapa 1: $(n-1)$ mesaje

Etapa 2: $(n-1)(n-2)$ mesaje

...

Etapa $m+1$: $(n-1)(n-2)\dots(n-(m-1))$ mesaje

Numărul total de mesaje este de complexitate $O(n^{m+1})$.

Numărul de runde este $m+1$ pentru a ajunge la consens, în prezența a maxim m erori.

3.1.5.4 Identificarea mesajelor

O problemă ce se ivește în cadrul acestui algoritm este că mesajele pot fi ambigui din punctul de vedere al identificării runde. Soluția propusă de [LaPS82] constă în a prefixa valoarea v_i trimisă în pasul 2 al algoritmului, cu numărul i al locotenentului care o trimite. Pe măsura desfășurării algoritmului într-un moment k , algoritmul $OM(n-k, m-k)$ va fi apelat de $(n-1)(n-2)\dots(n-k)$ ori pentru a transmite o valoare având ca și prefix o secvența de k numere de locotenenți.

3.1.6. CBGP varianta cu mesaje scrise

3.1.6.1 Algoritm

Considerăm comandantul ca având numărul 0 (în semnături).

Algoritmul necesită o funcție $alegere(V)$ definită astfel:

$$alegere(\{v\}) = v$$

$$alegere(\{\}) = v_{def}$$

Trebuie menționat că fiecare locotenent i are un set V_i cu ordinele semnate corect pe care le-a primit până acum. Dacă comandantul este loial setul conține un singur element. Inițial toate seturile sunt vide.

Algoritmul $SM(n, 0)$ - nu există trădători

1. Comandatul trimite valoarea v tuturor locotenenților
2. Fiecare locotenent folosește valoarea trimisă de comandant, sau v_{def} în cazul în care nu a primit nici o valoare.

Algoritmul $SM(n, m)$ - există m trădători, $m > 0$

1. Comandatul trimite valoarea $v:0$ tuturor locotenenților
2. Pentru fiecare locotenent i

A. Dacă locotenentul primește un mesaj $v:0$ și nu a mai primit până acum un ordin, atunci locotenentul i execută următoarele acțiuni:

- A1. setează $V_i = \{v\}$
- A2. trimite mesajul $v:0:i$ tuturor locotenenților

B. Dacă locotenentul i primește un mesaj de forma $v:0:j_1:\dots:j_k$ și v nu este în V_i , atunci locotenentul i execută următoarele acțiuni:

- B1. $V_i = V_i + \{v\}$
- B2. dacă $k < m$ atunci trimite mesajul $v:0:j_1:\dots:j_k:i$ tuturor locotenenților care nu au semnat mesajul.

3. Pentru fiecare i

Dacă nu mai sunt mesaje locotenentul i folosește valoarea dată de funcția $alege(V_i)$

Este de remarcat cum în pasul 2 al prezentului algoritm, locotenentul i ignoră orice alt mesaj ce conține o valoare v aflată deja în setul de valori V_i . De asemenea, locotenentul i ignoră orice alt mesaj ce nu este semnat corespunzător.

În pasul 3 locotenentul determină că nu mai sunt mesaje dacă folosim o facilitate de tip time-out sau dacă determinăm din mesajele deja primite câte mai avem să primim.

3.1.6.2 Corectitudinea algoritmului

Teorema 3

Oricare ar fi n , algoritmul $SM(n,m)$ rezolvă problema generalilor bizantini dacă sunt maxim $m = n - 2$ trădători.

Demonstrație

Dacă comandantul este loial

Comandantul trimite ordinul $v:0$ tuturor locotenenților în pasul 1.

Toți locotenenții loiali primesc mesajul cu ordinul v în pasul 2 A și nici un alt mesaj în pasul 2 B pentru că nimeni nu poate falsifica mesajul.

Astfel, pentru toți locotenenții loiali setul $V_i = \{v\}$, ceea ce determină același rezultat al funcției $alege(V_i)$, în pasul 3.

Astfel am demonstrat **CBG2**. Cum **CBG2** este un caz particular a lui **CBG1**, pentru a demonstra **CBG1** trebuie doar să mai considerăm următorul caz:

Dacă comandantul este trădător

Pentru a demonstra **CBG1** trebuie să arătăm că oricare doi locotenenți loiali i și j iau aceeași decizie. Pentru a lua aceeași decizie, e suficient ca setul V pe care îl vor aplica funcției $alege$ să fie identic. Pentru a demonstra aceasta e suficient ca valorile din setul V să fie identice, prin urmare e suficient ca valorile ce ajung la un locotenent să ajungă și la celălalt.

Dacă i a primit ordinul v în pasul 2 A atunci, în pasul 2 B i îi trimite v lui j , care îl primește sigur (**OMH1**).

Dacă i primește ordinul v în pasul 2 B atunci

Dacă j era printre semnatarii mesajului, înseamnă că și j îl are.

Dacă j nu era printre semnatari atunci:

Dacă numărul de semnături al mesajului $k < m$ atunci i va trimite mesajul $v:0:j_1:\dots:j_k:i$ tuturor celor care nu l-au primit încă, adică și lui j .

Dacă numărul de semnături al mesajului $k = m$ atunci i nu va mai trimite mesajul, dar deoarece avem $m-1$ locotenenți trădători, cel puțin unul dintre semnatarii mesajului era loial, deci i-a trimis deja valoarea v lui j atunci când a primit-o.

3.1.6.3 Complexitatea algoritmului

Aplicarea algoritmului **SM**(n, m) cauzează lansarea a $n-1$ mesaje de către comandant. Fiecare mesaj ajuns la un locotenent cauzează lansarea a $n-2$ mesaje, ș.a.m.d.. Sumarizând:

Etapa 1: $(n-1)$ mesaje

Etapa 2: $(n-1)(n-2)$ mesaje

...

Etapa $m+1$: $(n-1)(n-2)\dots(n-(m-1))$ mesaje

Numărul total de mesaje este de complexitate **$O(n^{m+1})$** .

Numărul de runde este **$m+1$** pentru a ajunge la consens, în prezența a maxim m erori.

3.1.7. WBGP (Weak Byzantine Generals Problem)

3.1.7.1 Algoritmul

Soluția[Lamp83] aproximării **WBGP** funcționează în prezența oricărui număr de procese eronate. Trecând la limită o secvență a acestor soluții, obținem soluția exactă folosind un număr infinit de mesaje. Pentru a putea considera validă soluția aproximării presupunem că setul V al valorilor posibile este cel al numerelor reale.

Pentru fiecare număr întreg $k > 0$, definim algoritmul **AG**(k) care necesită k runde de transmiteri de mesaje. Trădătorii nu sunt constrânși în urma acest algoritmului, cu toate că facem presupunerea că ei trimit toate mesajele care ar trebui să le trimită (valoarea mesajului poate fi incorectă). Cu toate acestea, deoarece am presupus V ca fiind setul numerelor reale, fiecare valoare trimisă de trădător se va regăsi în V .

Algoritmul **AG**(k) - următoarele runde sunt folosite pentru a calcula valorile $v_i^{(r)}$, $1 \leq r \leq k$

1. Runda 1

Generalul 0 trimite valoarea sa, v , fiecărui general i .

Fiecare general i își asignează lui $v_i^{(1)}$ valoarea primită de la generalul 0.

2. Runda r , $1 \leq r < k$

Fiecare general j trimite valoarea $v_j^{(r-1)}$ tuturor generalilor i .

Fiecare general i îi asignează lui $v_i^{(r)}$ maximum celor n valori pe care le primește.

3. Fiecare general îi asignează valorii interne v_i media celor k valori obținute din rundele anterioare.

3.1.7.2 Corectitudinea algoritmului

Lema 4

Presupunem că $|v| < D, \forall v \in V$.

Fie s_r și t_r elemente din V alese astfel încât:

$$s_r \leq t_{r+1}, \forall r, 1 \leq r < k$$

Ecuția 3.13

și

$$t_r \leq s_{r+1}, \forall r, 1 \leq r < k$$

Ecuția 3.14

Atunci

$$\left| \sum_{r=1}^k s_r - \sum_{r=1}^k t_r \right| < 2D$$

Ecuția 3.15

Demonstrație:

Din Ecuția 3.13, putem scrie:

$$\begin{aligned} s_1 &\leq t_2 \\ s_2 &\leq t_3 \\ s_3 &\leq t_4 \\ &\dots \\ s_{k-2} &\leq t_{k-1} \\ s_{k-1} &\leq t_k \end{aligned}$$

Ecuția 3.16

Adunând relațiile anterioare rezultă că :

$$\sum_{r=1}^{k-1} s_r \leq \sum_{r=2}^k t_r$$

Ecuția 3.17

Adunând s_k în ambele părți ale Ecuția 3.17, obținem:

$$\sum_{r=1}^k s_r \leq s_k + \sum_{r=2}^k t_r$$

Ecuția 3.18

Adunăm t_1 de ambele părți ale inecuației și trecem suma termenilor t_r în partea stângă.

$$\sum_{r=1}^k s_r - \sum_{r=1}^k t_r \leq s_k - t_1$$

Ecuția 3.19

Din ipoteză avem $|v| < D$, $\forall v \in V$ și $s_k \in V$, $t_1 \in V$ rezultă că:

$$s_k - t_1 < 2D$$

Ecuția 3.20

Din Ecuția 3.19 și Ecuția 3.20 obținem:

$$\sum_{r=1}^k s_r - \sum_{r=1}^k t_r < 2D$$

Ecuția 3.21

Prin simetrie, ținând cont de Ecuția 3.14 se obține că:

$$\sum_{r=1}^k t_r - \sum_{r=1}^k s_r < 2D$$

Ecuția 3.22

Ecuția 3.21 și Ecuția 3.22 conduc la Ecuția 3.15.

$$\left| \sum_{r=1}^k s_r - \sum_{r=1}^k t_r \right| < 2D$$

Ecuția 3.15

Teorema 5

Dacă $|v| < D$, $\forall v \in V$, algoritmul AG(k) satisface următoarele proprietăți.

T5-1. Dacă toți generalii sunt loiali, atunci $v_i = v$, $\forall v$

T5-2. Dacă generalii i și j sunt loiali, atunci $|v_i - v_j| < 2D/k$.

Demonstrație:

Dacă toți generalii sunt loiali atunci ei trimit valorile în mod corect, deci $v_i = v$. Astfel T5-1 este îndeplinită.

Dacă generalii i și j sunt loiali, atunci ei își trimit unul-altuia, în mod corect, valorile $v_i^{(c)}$, respectiv $v_j^{(c)}$ în runda $c+1$. Datorită pasului 3 al algoritmului, $\forall r, r > c+1$, vom avea:

$$v_i^{(r)} \leq v_j^{(r+1)}$$

Ecuția 3.23

și

$$v_j^{(r)} \leq v_i^{(r+1)}$$

Ecuția 3.24

Aplicând lema 4 pentru valorile $v_i^{(r)}$, respectiv $v_j^{(r)}$, care datorită relațiilor Ecuția 3.23 și Ecuția 3.24 se supun inegalităților din ipoteza lemei Ecuția 3.13 și Ecuția 3.14 vom obține ceea ce era de demonstrat adică T5-2

$$|v_i - v_j| < 2D/k$$

Ecuția 3.25**3.1.7.3 Complexitatea algoritmului**

În runda 1 generalul 0 trimite $n-1$ mesaje.

În fiecare din rundele r , $1 \leq r \leq k$, se trimit $n(n-1)$ mesaje

Numărul total de mesaje este de complexitate **$O(n^2)$** .

Numărul de runde este **$k+1$** pentru a ajunge la consens, în prezența unui număr teoretic infinit de erori.

3.1.8. BGP varianta cu mesaje orale**3.1.8.1 Algoritmul**

Algoritmul presupune:

1. existența unei valori notată cu v_{def} implementată în cadrul fiecărui general (ex: așteaptă)
2. existența unei valori inițiale, v implementată în cadrul fiecărui general (ex: atacă)
2. existența unei funcții numită *majoritate* definită astfel:
 - $majoritate(v_1, v_2, \dots, v_{n-1}) = v$, dacă majoritatea valorilor variabilelor $v_i = v$
 - $majoritate(v_1, v_2, \dots, v_{n-1}) = v_{def}$, dacă majoritatea valorilor variabilelor $v_i \neq v$

Algoritmul **FOM**($n, 0$)

Generalul i trimite valoarea sa implicită v tuturor generalilor

Generalul i așteaptă cele $n-1$ răspunsuri și dacă timpul a expirat consideră valoarea v_{def} .

Generalul folosește funcția *majoritate* (v_1, v_2, \dots, v_{n-1}) pentru a-și afla noua valoare.

Algoritmul **FOM**(n, m) - există m trădători, $m > 0$

Fiecare general i

trimite valoarea sa implicită v tuturor generalilor

Pentru fiecare general i

trimite valoarea v_i la restul de $n-2$ generali folosind **OM**($n-1, m-1$)

Pentru fiecare i și fiecare $j, j \neq i$

v_j = valoarea primită de generalul i de la generalul j (trimisă la pasul 2)
sau v_{def} dacă nu a primit nici o valoare

Generalul i folosește valoarea dată de funcția *majoritate* (v_1, v_2, \dots, v_{n-1})

3.1.8.2 Corectitudinea algoritmului

Se observă că algoritmul **FOM** este de fapt o generalizare a algoritmului **OM.**, tot astfel cum **CBG1** și **CBG2** sunt cazuri particulare ale **BG3** și **BG4** (sau **BGR3** și **BGR4** - ele sunt echivalente[LAPS80]).

Corectitudinea algoritmului rezultă din faptul că în **FOM**($n,0$) fiecare general se comportă ca fiind comandantul din **CBGP**.

3.1.8.3 Complexitatea algoritmului

Hadzilacos și Halpern [HaHa91] au demonstrat că, în cazul algoritmului cu mesaje orale este necesar un număr mediu de $n(m+1)/4$ de mesaje. Pentru cazul cel mai defavorabil se demonstrează ca și la CBGP că:

- ♦ numărul total de mesaje este de complexitate $O(n^{m+1})$.
- ♦ numărul de runde este $m+1$ pentru a ajunge la consens, în prezența a maxim m erori.

3.1.9. BGP varianta cu mesaje scrise

3.1.9.1 Algoritmul

Algoritmul necesită o funcție *alegere*(V) definită astfel:

$$\begin{aligned} \text{alegere}(\{v\}) &= v \\ \text{alegere}(\{\}) &= v_{def} \end{aligned}$$

Trebuie menționat că fiecare general i are un set V_i cu ordinele semnate corect pe care le-a primit până acum. Inițial toate seturile sunt vide.

Algoritmul **FSM**(n, m) - există m trădători, $m > 0$

1. Generalul k trimite valoarea v : k tuturor generalilor
2. Pentru fiecare general i

A Dacă generalul primește un mesaj v : j și nu a mai primit până acum un ordin, atunci generalul i execută următoarele acțiuni:

- A1. setează $V_i = \{v\}$
- A2. trimite mesajul v : j : i tuturor generalilor

B Dacă generalul i primește un mesaj de forma v : j_1 : \dots : j_k și v nu este în V_i , atunci el execută următoarele acțiuni:

- B1. $V_i = V_i + \{v\}$

B2. dacă $k < m$ atunci trimite mesajul $v: j_1: \dots: j_k: i$ tuturor locotenenților care nu au semnat mesajul.

3. Pentru fiecare i

Dacă nu mai sunt mesaje generalul i folosește valoarea dată de funcția $alege(V_i)$

Este de remarcat cum în pasul 2 al prezentului algoritm, generalul i ignoră orice alt mesaj ce conține o valoare v aflată deja în setul de valori V_i . De asemenea el ignoră orice alt mesaj ce nu este semnat corespunzător.

În pasul 3 generalul determină că nu mai sunt mesaje dacă folosim o facilitate de tip time-out sau dacă determinăm din mesajele deja primite câte mai avem să primim.

3.1.9.2 Corectitudinea algoritmului

Se observă că algoritmul **FSM** este de fapt o generalizare a algoritmului **SM**, tot astfel cum **CBG1** și **CBG2** sunt cazuri particulare ale **BG3** și **BG4** (sau **BGR3** și **BGR4** - ele sunt echivalente [LaPS80]).

Corectitudinea algoritmului rezultă din faptul că în **FSM**, pasul 1, fiecare general se comportă ca fiind comandantul din **CBGP**.

3.1.9.3 Complexitatea algoritmului

Hadzilacos și Halpern [HaHa91] au demonstrat că, în cazul mesajelor scrise acest număr mediu scade considerabil la $(n+m-1)/2$. Pentru cazul cel mai defavorabil se demonstrează ca și la CBGP că:

- ♦ numărul total de mesaje este de complexitate $O(n^{m+1})$.
- ♦ numărul de runde este $m+1$ pentru a ajunge la consens, în prezența a maxim m erori.

3.1.10. Alți algoritmi bazați pe BGP

Algoritmii anterior prezentați sunt considerați algoritmii de consens clasici, iar de la publicarea lor în anii 80 mulți alți algoritmi au apărut, bazați pe aceștia.

3.1.10.1 Incomplete path

Ipoteze: generalii sunt doar parțiali conectați.

Lucrarea care a lansat conceptul: [LaPS80]

Descriere: algoritmul seamănă cu cel clasic, dar valorile sunt trimise de-a lungul căilor posibile, nu în varianta many-to-many. Pentru mesaje scrise, diametrul d al grafului este important.

Număr maxim de procese eronate: $k, n \geq 2k+m, m > 0$ (oral) or $n \geq k+d-1$ (scrise)

Număr total de mesaje: complexitate $O(n^2)$.

Număr de runde: $k+1$

Complexitatea algoritmului: exponențială.

3.1.10.2 Passive/active generals

Ipoteze: doar valorile netrimise deja sunt trimise mai departe.

Lucrarea care a lansat conceptul: [DoSt83]

Descriere: Un proces retransmite doar valorile pe care nu le-a transmis deja. Dacă procesul trimite două valori distincte, se oprește și emite "sender error".

Număr maxim de procese eronate: nelimitate.

Număr total de mesaje: complexitate $O(n^2)$.

Număr de runde: $m+1$

Complexitatea algoritmului: exponențială

3.1.10.3 Protocolul fazelor

Ipoteze: sistem asincron.

Lucrarea care a lansat conceptul: [AtDG84]

Descriere: Fiecare procesor trimite o notificare în care specifică în ce fază este. Fazele sunt separate prin time-out-uri care garantează că nu există două procese ce funcționează corect în două faze distincte. Algoritmul implică existența a două praguri: TRH1 și TRH2.

Număr maxim de procese eronate: m , $TRH1 \geq TRH2 + m \geq 2m + 1 - n > 4m$

Număr total de mesaje: Complexitate $O(n^2)$.

Număr de runde: $m+1$

Complexitatea algoritmului: exponențială

3.1.10.4 BGP pentru procesoare fail-stop

Ipoteze: procesoare fail-stop

Lucrarea care a lansat conceptul: [Schn84, Schn90]

Descriere: Procesoarele fail-stop se opresc dacă au o eroare. Fiecare locotenent i are un set V_i cu ordinele semnate corect pe care le-a primit până acum. Dacă comandantul este loial setul conține un singur element. Inițial toate seturile sunt vide.

Algoritmul este practic identic cu **SM**(n, m), diferența constând în definirea funcției *alege*.

Număr maxim de procese eronate: ca și SM.

Număr total de mesaje: Complexitate $O(n^2)$.

Număr de runde: $m+1$

Complexitatea algoritmului: exponențială

3.1.10.5 Inexact agreement

Ipoteze: se atinge o precizie suficient de bună a valorilor finale.

Lucrarea care a lansat conceptul: [MaSc85]

Descriere: Fiecare procesor pornește de la un set primit de valori, pe care își bazează setul de valori acceptate. Se calculează limitele acestui set, și orice valoare în afara limitelor este înlocuită cu valoarea proprie. Valoarea finală este media setului nou de valori.

Număr maxim de procese eronate: $m, n \geq 3m+1$.

Număr total de mesaje: Complexitate $O(n)$.

Număr de runde: 1

Complexitatea algoritmului: exponențială

3.1.10.6 Approximate agreement

Ipoteze: valori reale, valori într-un anumit interval de precizie, (a)sincron.

Lucrarea care a lansat conceptul: [DLPS86]

Descriere: Fiecare proces trebuie să introducă o valoare reală (și nu una binară), iar toate procesele trebuie să se decidă asupra unor valori reale care variază cu ε ($\varepsilon \rightarrow 0$) una față de alta. Dacă $\varepsilon=0$ atunci aceasta este de fapt problema generalilor bizantini. Acest algoritm funcționează prin aproximații succesive, cu viteză de convergență calculată.

Rezultatul cel mai interesant este acela că acest algoritm funcționează atât în sisteme sincrone, cât și în sisteme asincrone (ε fiind dat). Pentru sistemele sincrone, viteza de convergență depinde de raportul dintre numărul de procese eronate și numărul total de procese.

Număr maxim de procese eronate: $m, n > 3m$ (sincron) sau $n > 5m$ (asincron)

Număr total de mesaje: depinde de viteza de convergență.

Număr de runde: depinde de viteza de convergență.

Complexitatea algoritmului: exponențială

3.1.10.7 Early stopping

Ipoteze: tipul de căderă este stop.

Lucrarea care a lansat conceptul: [DLPS86]

Descriere: Dacă într-un caz particular avem $f < m$ erori, atunci ar exista o posibilitate ca algoritmul să se termine mai devreme. Se reduce generalitatea erorii.

Număr maxim de procese eronate: $f < m$

Număr total de mesaje: complexitate $O(n^2)$.

Număr de runde: $\text{Min}(m+1, f+2)$

Complexitatea algoritmului: exponențială (worst case), lineară (best case).

3.1.10.8 Abordare probabilistică

Ipoteze: numărul de trădători poate fi exprimat ca o combinația a numărului total de participanți.

Lucrarea care a lansat conceptul: [Brac87]

Descriere: dacă numărul de trădători m , are complexitate $O(n)$ atunci algoritmul are un număr de runde exponențial cu n , pe când dacă m are complexitate $O(\sqrt{n})$, atunci algoritmul are un număr de runde așteptate constant. Dacă putem transforma un sistem cu n procese și $O(n)$ procese eronate într-unul cu t procese și $O(\sqrt{t})$ procese eronate, atunci problema are soluție într-un timp constant, variabil fiind doar timpul de conversie a unui sistem în alt sistem.

Număr maxim de procese eronate: m , $m=n/(3+\epsilon)$ pentru mesaje scrise, $m=n/(2+\epsilon)$ pt mesaje orale

Număr total de mesaje: complexitate $O(n)$.

Număr de runde: complexitate $O(\log n)$

Complexitatea algoritmului: exponențială

3.1.10.9 Abordarea probabilistică complexă

Ipoteze: convertește și descompune sistemul în sub-sisteme mai mici

Lucrarea care a lansat conceptul: [Brac87]

Descriere: dacă considerăm $t=O(n^2)$ și fiecare din aceste m procese este un proces compus din $s=O(\log n)$ procese fizice (dintre care oricare poate fi într-unul sau mai multe procese compuse), atunci putem alege procesele compuse astfel încât cel mult \sqrt{t} sunt eronate. O rutină standard cu un număr de $m+1$ runde poate fi rulată în fiecare din aceste t procese compuse și rezultatul poate fi rulat pe un număr de t procese cu un număr de runde estimate constant, $O(\log n)$.

Număr maxim de procese eronate: m , $m=n/(3+\epsilon)$ pentru mesaje scrise, $m=n/(2+\epsilon)$ pt mesaje orale

Număr total de mesaje: complexitate $O(n^2)$.

Număr de runde: complexitate $O(\log n (\log \log n))$.

Complexitatea algoritmului: exponențială

3.1.10.10 Continual common knowledge

Ipoteze: luarea de decizii independente; terminare prematură independentă a diverșilor generali

Lucrarea care a lansat conceptul: [HaMW90]

Descriere: la un moment dat în algoritm unele procesoare pot avea destule informații pentru a se decide. Altfel spus, dacă cunoașterea comună (întâlnită la problema clasică) spune că "Toată lumea știe că toată lumea știe că ...x", prin cunoașterea în avans se spune că "Toată lumea va știi că toată lumea va știi că ...x".

Număr maxim de procese eronate: m , la fel ca și algoritmul clasic.

Număr total de mesaje: complexitate $O(n^2)$.

Număr de runde: $m+1$

Complexitatea algoritmului: exponențială

3.1.10.11 Game theory

Ipoteze: convertește generalii în agenți

Lucrarea care a lansat conceptul: [Halp02]

Descriere: Înțelegerea bizantină poate fi văzută ca un joc în care, la fiecare pas, un agent poate decide să trimită un mesaj sau să nu trimită nimic sau să atace sau să se retragă. În esență BGP poate fi văzută ca un joc între două echipe agenții trădători și agenții loiali, dar compoziția echipelor este necunoscută de către agenții loiali.

Pentru modelare imaginați-vă că "trădătorii" pot să corupă unii dintre membrii "echipei loialilor", aducându-l în tabăra "trădătorilor". Odată ce un general este "corupt" ce poate face adversarul cu el depinde de tipul de erori considerate. În cazul căderilor, adversarul decide care dintre mesajele agentului corupt va fi trimis în runda în care agentul e corupt, dar adversarul nu poate modifica mesajele. În cazul erorilor bizantine adversarul face mesajele agentului corupt, în speță el poate chiar modifica mesajele primite din tabăra loială. În cazul omisiunilor agentul se comportă conform protocolului inițial, putând însă omite să trimită un set de mesaje la un moment aleator. Acest tip de erori este specific în modelarea problemelor de comunicare locală (ex: o congestie de mesaje).

Este foarte dificil în a "recompensa" agentul trădător.

Număr maxim de procese eronate: m – la fel ca și varianta clasică.

Număr total de mesaje: complexitate $O(n^2)$.

Număr de runde: $m+1$

Complexitatea algoritmului: exponențială

3.1.10.12 Low complexity

Ipoteze: se folosește un protocol sincron pentru a atinge consensul asincron

Lucrarea care a lansat conceptul: [CNLV03]

Descriere: presupune existența componentei de tip wormhole care sunt sincrone și sigure spre deosebire de restul sistemului care este asincron și poate cădea oricum și oricând. Poate funcționa în două variante. Prima variantă urmărește doar consensul unui număr de biți fixat, iar a doua variantă consensul asupra unei mixturi de date, fiecare dată fiind de dimensiune fixă.

Număr maxim de procese eronate: m – la fel ca și varianta clasică.

Număr total de mesaje: $O(2n)$ – pentru multicasts sau $O(n(2n-m-2))$ – pentru unicasts.

Număr de runde: 2

Complexitatea algoritmului: exponențială.

3.1.11. Discuție

Cele două tipuri de mesaje cu care funcționează BGP reprezintă practic extreme: detecția performantă versus viteză. Nu se poate alege o variantă ideală atât ca și detecție cât și ca și viteză, fie și numai din cauza acestor opțiuni strict distincte.

Din prezentarea celor trei variante de BGP, se mai deduce încă o concluzie și anume că timpul necesar rulării acestui algoritm este deosebit de consistent. Prin relaxarea ipotezelor (WBGP) se poate obține un timp mai rezonabil, dar în defavoarea performanței.

Am realizat un mic studiu, bazat pe funcțiile ce descriu complexitatea algoritmilor. Am preluat formulele ce descriu aceste funcții, așa cum au rezultat ele din teorie și le-am trasat graficul (folosind TableCurver). Rezultatele sunt cel puțin interesante.

Dacă considerăm $m=3$ figura 3.6 prezintă variația numărului mediu de mesaje (funcția 1) și cea a numărului maxim de mesaje (funcția 2) pentru algoritmul cu mesaje orale.

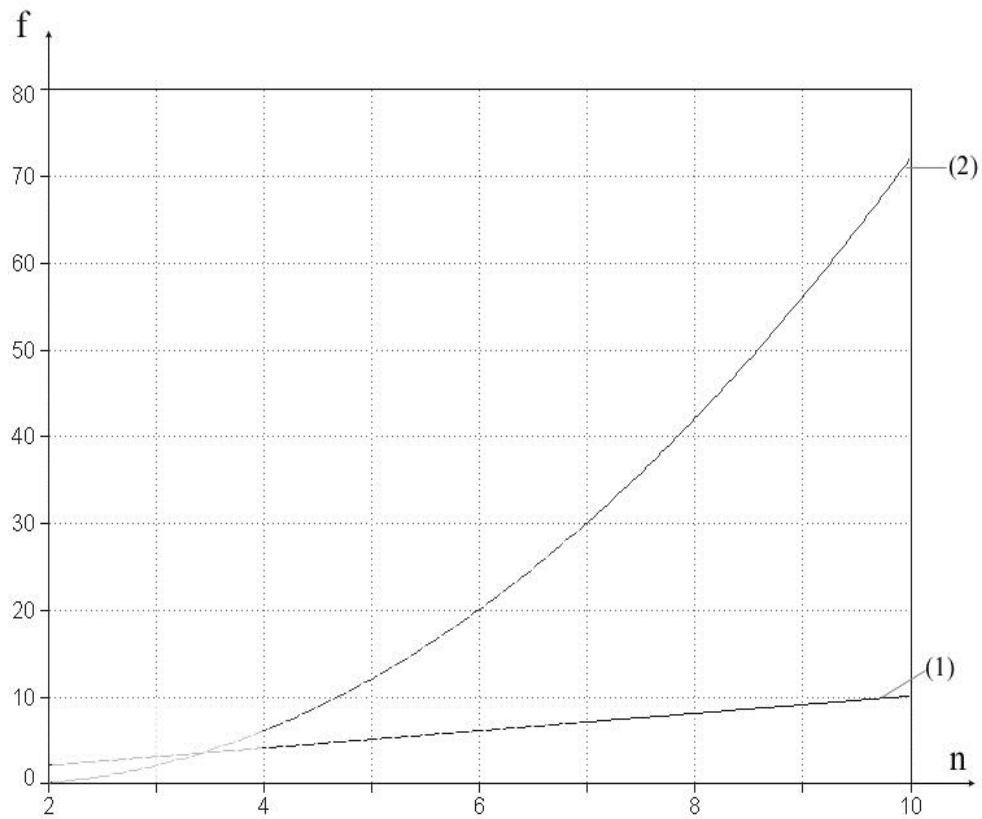


Figura 3.6. Variația mesajelor funcție de numărul generalilor, $m=3=ct.$

Interesant este de observat că pentru $n=2$ și $n=3$ generali (pentru care problema cu mesaje orale nu are rezolvare întotdeauna) valoarea medie a numărului de mesaje este mai mare decât cea maximă(!!).

De la $n=4$ generali însă lucrurile reintră în normalitate și soluția este cea corectă. Graficul este însă reprezentat cu linie întreruptă până la valoarea $n=7$ deoarece având 3 trădători problema obține rezultatul corect doar începând cu 7 generali.

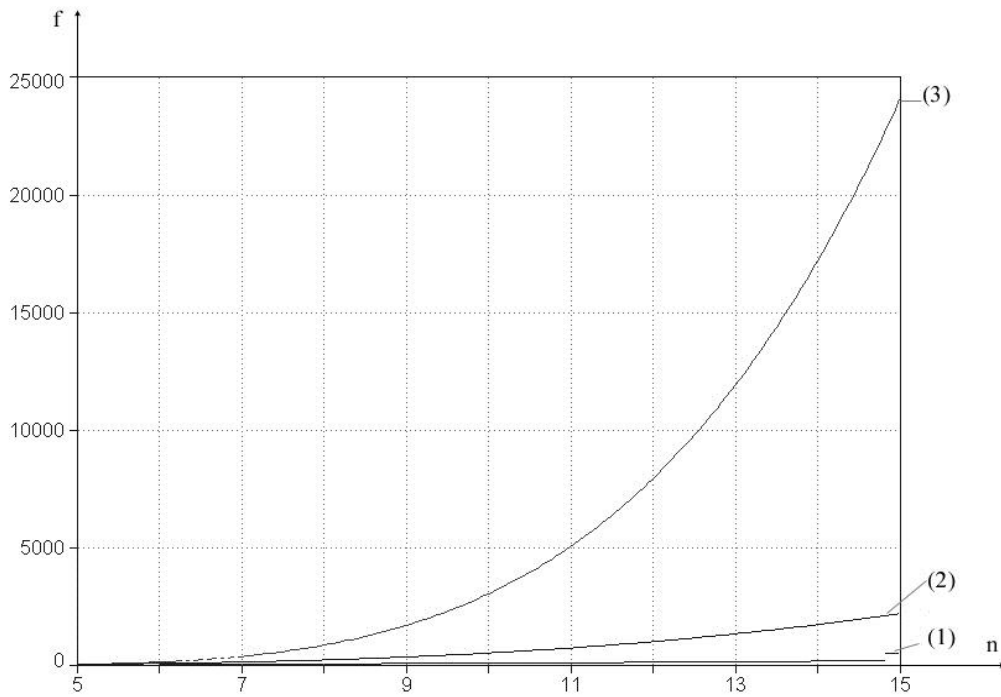


Figura 3.7. Numărul de mesaje în algoritmii SM și FSM

Pentru algoritmii **SM** și **FSM** se observă cum odată cu creșterea numărului de trădători de la 3 (funcția 1) la 4 și 5 (funcțiile 2, respectiv 3) numărul de mesaje (de pe axa Oy) crește semnificativ. Pe axa Ox avem numărul total de generali.

3.2. Licitații

Există foarte multe variante de licitații [BeSe98], comună acestora fiind idea de ofertă a unei valori, ofertă care poate fluctua, până se ajunge la o valoare care, din diverse motive, nu mai variază. Matematic, acest lucru se formalizează astfel: fiind dată o variabilă a , un set de valori posibile pentru ea $\{v \mid a, v \in V\}$ și fiind definită o relație de ordine pe mulțimea V , atunci

$$a = \text{funcție_licitație}(v_0, v_1, \dots, v_k)$$

unde:

$$v_i \in V, i=0, k$$

Următorul tabel reprezintă o sinteză a diverselor tipuri de licitații și a regulilor lor.

Tipul licitației	Scurtă descriere a regulilor
Engleze ască	Licitație continuu ascendentă. Cei care licitează văd cea mai mare sumă (posibil și restul sumelor licitate, precum și comentariile licitatorilor). Licitatorii își pot mări suma licitată de ei pe tot decursul licitației. Câștigă persoana care a licitat cel mai mult, sumă pe care o și plătește.
Primul preț Licitație ascunsă/ în plic/ închisă	Licitatorii trimit în secret valoarea pe care ar fi dispuși să o plătească. Toate mesajele se deschid simultan și cel cu valoarea cea mai ridicată este câștigător. Licitatorii nu își pot mări suma licitată de ei. Câștigă persoana care a licitat cel mai mult, sumă pe care o și plătește
Olandeză	Licitație continuu descendentă, cei care licitează văd prețul curent și trebuie să decidă dacă vor cumpăra la acest preț sau mai așteaptă până scade. Câștigă persoana care a licitat prima prețul final, sumă pe care o și plătește.
Licitație dublă continuă	Similară cu NASDAQ. Cumpărătorii și vânzătorii văd continuu piața și prețurile, fac oferte în timp real, care se achită la prețul pieții.
Licitație dublă închisă (ascunsă)	Cumpărătorii și vânzătorii trimit oferte în plic simultane. Cei care organizează licitația deschid ofertele și clarifică piața. Licitația se repetă de câteva ori pentru a oferi un preț continuu.

Vickrey	Similară cu licitația ascunsă, câștigă persoana care a licitat cel mai mult. Cu toate acestea, câștigătorul plătește doar valoarea celei de-a doua oferte (în ordine descrescătoare)
----------------	--

Tabelul 3.1. Tipuri de licitații [BeSe98]

Putem considera că o licitație este o modalitate de atingere a consensului deoarece dacă atunci când s-a hotărât câștigătorul, toți cunosc valoarea pentru care a câștigat, deci putem considera că toți s-au pus de acord asupra unei valori.

3.2.1. Utilizări practice ale licitațiilor

Se spune că prima licitație este menționată chiar în Biblie [RaNo02] (vânzarea lui Ioan de către frații săi). Cu toate acestea, primele licitații documentate arheologic sunt cele din jurul anului 500 î.Ch. din Babilon, pentru vânzarea femeilor.

Pe vremea romanilor licitațiile erau deja ceva bine reglementat [Reyn01]. Locul unde se desfășura licitația se numea „atrium auctionarium” și licitația avea următorii 4 actori, bine definiți:

Dominus – cel care vindea
 Argentarius – cel care organiza licitația
 Praeco – cel care făcea reclamă produsului și care conducea licitația
 Emptor – cel care licita cel mai mult („caveat emptor”)

De asemenea de pe vremea romanilor avem licitațiile publice (subhastare) care se realizau cu bunurile cucerite de către soldați.

Cea mai grozavă licitație din istoria omenirii a avut loc în 193 e.n. când miza era întregul Imperiu Roman, scos la licitație de garda pretoriană. A câștigat Didius Iulianus cu 6250 drahme pentru fiecare membru al gărzii.

În Asia, preoții budiști licitau posesiunile confrăților lor decedați.

În 1595, licitațiile au fost trecute în Oxford English Dictionary, iar în 1744 a fost fondată casa de licitații Sotheby's, urmată de Christie's în 1766. Interesant este faptul că de atunci ambele tipuri majore de licitații: cea engleză și cea olandeză erau prezente în cadrul aceleiași licitații.

În secolul XXI e-Bay este deja un fenomen bine cunoscut. Apărând o asemenea nevoie majoră a unei negocieri automate apare marea problemă a unei ontologii comune, posibil bazându-se pe agenți inteligenți [BeSe99].

La nivel hardware licitație descrescătoare întâlnim, de exemplu, în comunicație, atunci când emițătorul nu știe ce poate receptorul și va emite la început la frecvență maximă, scăzând-o apoi treptat, permițându-i receptorului să răspundă atunci când va putea diferenția semnalul, deci când va ajunge la valoarea maximă a frecvenței pe care o recunoaște.

3.2.2. Apartenența la clasa problemelor de consens

În conformitate cu D1 toate tipurile de licitații aparțin clasei problemelor de consens.

Valorile de licitație inițiale (aparțin aceleiași clase de valori – numerice) sunt rulate (prelucrate) într-una sau mai multe runde până se atinge o anumită

valoare (maximă/minimă). Valorile licitate la un anumit moment de timp pot apărea simultan sau succesiv, în conformitate cu D1.

Rezultatul final – valoarea de închidere a licitației – este recunoscut și însușit de către toți participanții.

Faptul că doar unul dintre participanți va beneficia de obiectul pentru care se licitează nu contravine lui D1, deoarece scopul licitației nu ne interesează. Ne interesează doar ca aceasta licitație să se încheie și ca rezultatul să fie recunoscut.

Licitația dublu continuă poate fi văzută ca și un consens în sistem parțial sincron, dar cu mai multe momente (sfârșitul zilei de tranzacționare) de sincronizare. Este, de altfel, singurul tip de licitație care prin definiție nu se încheie niciodată.

În conformitate cu D2 doar licitațiile de tipul primul preț (licitație ascunsă/ în plic/ închisă) și Vickrey aparțin clasei problemelor de consens, deoarece directiva de validitate ("Valoarea asupra căreia s-a decis trebuie să fi fost valoarea inițială a cel puțin unui proces") este încălcată de restul tipurilor de licitații (valorile de pornire diferind, de obicei, de cele de oprire și se pot modifica în decursul licitației).

3.2.3. Licitațiile prin formalism

Particularizarea licitațiilor o facem prin restrângerea funcției consens Ecuția 2.7. Avem nevoie de existența unei valori neutre, notată cu NB , de la „no bet”.

$$V_f = V_i \times V_i \quad (a)$$

$$y_i = (x_{win}, x_{i\max}) \quad (b)$$

$$\exists NB \in V_i, \forall v \in V_i \quad (c)$$

$$NB \leftarrow \overset{\text{overpowered by}}{v}$$

Ecuția 3.26

Astfel, mulțimea valorilor finale este o compunere de două ori a mulțimii valorilor inițiale. Fiecare valoare finală i este o pereche formată din valoarea care a câștigat licitația și valoarea cea mai bună (maximă) licitată de către participantul i .

Prin existența valorii NB permitem abținerea de la participarea efectivă la licitație, putând astfel modela un sistem care a căzut. Cu toate acestea, deoarece valoarea finală conține valoarea care a câștigat licitația pentru toate sistemele, nu mai avem nevoie de funcția majoritate.

Putem chiar să găsim câștigătorul licitației, prin faptul că acesta are ca și valoare finală o pereche cu două valori identice.

Fiecărui tip de licitație în parte îi corespunde o funcție ψ de consens diferită. Scopul nostru însă nu este să descriem funcționarea internă în detaliu a acestei funcții.

3.3. Recunoașterea întregului

„A fost odată ca niciodată, în India, un elefant și cinci oameni orbi, care nu ieșiseră niciodată din casă. Cei cinci se hotărâsc, într-o bună zi să facă o

plimbare și ajung la elefant. Vrând să vadă cum este, fiecare dintre ei a pus mâna pe el. Primul om a pipăit urechea și a ajuns la concluzia că elefantul seamănă foarte mult cu o frunză. Al doilea om a examinat corpul animalului și s-a gândit că acesta seamănă foarte mult cu un munte. Al treilea a studiat coada patrupedului și a cugetat că elefantul este asemănător unei frânghii. Al patrulea om a atins piciorul animalului și a comparat elefantul cu un copac. Cel de-al cincilea om a pus mâna pe trompa elefantului și a dedus că animalul acesta este pecum un șarpe. Apoi, fericiți, au plecat cu toții și depănu povestea lor despre cum era elefantul.”

Atât de frumos exprimată în această parabolă veche de 3000 de ani, clasa problemelor de recunoaștere a unui întreg, bazat pe descrieri parțiale este o problemă de actualitate în calculatoare.

Pe baza unor date inițiale provenite dintr-una sau mai multe surse se cere o predicție asupra identității obiectului măsurat, de obicei prin selectarea acestuia dintr-o plajă mai largă de obiecte. Plaja de obiecte dintre care se alege poate fi fixă sau se pot adăuga noi obiecte, de exemplu, în urma unui proces de învățare.

Evident că se poate să nu avem o potrivire perfectă, iar atunci, funcție de implementare, se poate opta între a trece acel obiect la categoria „necunoscut” sau să-l clasificăm într-una sau mai multe categorii cu care are similitudini.

3.3.1. Utilizare practică

O primă aplicație este cea de recunoaștere imagini. Recunoașterea fețelor pe baza unor trăsături primare sau descifrarea fotografiilor aeriene sunt numai câteva exemple ale posibilităților de utilizare. Recunoașterea amprentelor e o altă aplicație, deja clasică.

Dar nu numai aceste aplicații sunt interesante. Putem menționa, de asemenea, recunoașterea semnalelor primite pe canale bruiaate diferite sau recunoașterea semnăturilor virușilor sau a unor atacuri [ALRL04], pe baza unor „simptome” specifice [PRWS07] sau chiar recunoașterea serviciilor oferite de către un server [HLLC05].

3.3.2. Apartenența la clasa problemelor de consens

În conformitate cu D2, această problemă a recunoașterii întregului, nu este o problemă de consens, deoarece valoarea finală și valorile inițiale aparțin unor mulțimi distincte, încălcându-se astfel presupunerea de validitate.

În conformitate cu D1, însă, această problemă însă este o problemă de consens.

Valorile inițiale sunt valori în general, distincte, aparținând unei clase de valori folosită pentru descrierea însușirilor, comună tuturor participanților la consens.

Valoarea finală este obținută pe baza valorilor inițiale și aparține unei clase de valori folosită pentru descriere obiectelor.

Valoarea finală este mai apoi recunoscută de către cel puțin unul dintre participanți, de obicei cel care a realizat și calcularea ei. Dacă calculul acesteia este realizat distribuit, atunci toți participanții la calculul valorii finale o vor accepta în final.

Interesant este faptul că dacă calculul se petrece pe un server central, restul sub-sistemelor fiind doar furnizori de date finale, este suficient ca procesul principal să ajungă la o concluzie ca să considerăm încheiată funcția de consens. În acest caz particular "o parte suficientă dintre participanții la dezbateri", este chiar 1.

Ținând cont de toate aceste precizări, putem să enunțăm faptul că clasa problemelor de recunoaștere a întregului este inclusă în clasa problemelor de consens.

3.3.3. Formalizarea recunoașterii întregului

O abordare interesantă apare atunci când dorim să descriem un sistem în care fiecare sub-sistem înregistrează o altă variabilă, iar fiecare sub-sistem va trebui să cunoască, în cele din urmă, starea generală a întregului sistem.

Această problemă în conformitate cu D2 nu este o problemă de consens, deoarece valoarea finală nu se află în cele inițiale. D1 include însă această problemă în clasa problemelor de consens.

Timpul poate fi limitat, ca și în cadrul BGP, dar datorită [Lamp86], îl vom lăsa nelimitat.

Restricțiile aplicabile formulei (3-7) pentru a descrie această clasă de probleme sunt minime, ele referindu-se doar la mulțimea valorilor finale.

$$V_f = V_i^m \times W$$

Ecuția 3.27

unde

m – numărul total de sub-sisteme esențiale

W – o altă clasă de valori

Sub-sistemele esențiale reprezintă, în principiu, acele sisteme ale căror valori sunt luate în considerare. Numărul acestor sub-sisteme poate sau nu să fie egal cu numărul total de sisteme.

Această nuanță este datorată de faptul că unele aplicații necesită redundanță în măsurarea unor mărimi sau pentru că unele sub-sisteme sunt defecte, iar algoritmul din interiorul funcției ψ le va exclude de la prelucrare.

3.4. Luarea de decizii

Luarea deciziilor reprezintă o altă clasă de probleme cu aplicații importante și utile în practică. Se pornește de la un set de date, de obicei, de dimensiuni mari, și se dorește extragerea unei concluzii sau chiar a unei funcții de transformare a acestor date într-o concluzie.

De obicei, această problemă se rezolvă prin sisteme expert. Acestea au drept date un set de valori inițiale, unde fiecare dintre aceste valori reprezintă răspunsul la o întrebare a sistemului. Pe baza răspunsurilor date de utilizator, se poate selecta o rezolvare la problema acestuia.

3.4.1. Utilizare practică

În ultima perioadă s-au dezvoltat diverse ajutoare software care să îi asiste pe oameni în evaluarea riscurilor și oportunităților [OnHH05]. Domeniile în care acest proces s-a dezvoltat mai ales sunt cel al economiei, al medicinei și cel juridic.

3.4.2. Apartenența la clasa problemelor de consens

În conformitate cu D2, această problemă nu este o problemă de consens, deoarece valoarea finală și valorile inițiale aparțin unor mulțimi distincte, încălcându-se astfel presupunerea de validitate.

În conformitate cu D1, însă, această problemă este o problemă de consens.

Valorile inițiale care pot să nu aparțină aceleiași clase de valori conduc la o valoare finală care sigur nu aparține claselor de valori inițiale.

Rezultatul e în cele din urmă cunoscut și însușit de către fiecare dintre sistemele care au introdus valorile inițiale.

Interesant e de menționat faptul că procesarea se poate face la nivelul unui singur sistem, sau distribuit, acest lucru neafectând cu nimic valabilitatea definiției propuse.

3.4.3. Formalizarea problemelor de luare a deciziilor

Pornind de la restrângerea ipotezelor inițiale pentru aplicațiile de recunoaștere a întregului, o posibilitate interesantă apare dacă:

$$V_f = W$$

$$m = 0$$

Ecuția 3.28

În această eventualitate starea generală a sistemului poate fi descrisă suficient de precis printr-o singură valoare care nu aparține claselor de valori inițiale. Acesta este cazul problemelor rezolvabile prin sisteme expert.

Se aplică și aici observația referitoare la cazul când calculul se petrece pe un server central, restul sub-sistemelor fiind doar furnizori de date finale. Este suficient ca procesul principal să ajungă la o concluzie ca să considerăm încheiată funcția de consens. În acest caz particular "o parte suficientă dintre participanții la dezbateri", este chiar un singur participant.

3.5. Concluzie

Demonstrația matematică ne arată că definiția propusă, D1, este viabilă și chiar mai bună decât definițiile deja existente (D2, WD2).

Prin D1 putem descrie o plajă mult mai largă de probleme decât D2, iar dacă vom găsi o modalitate de rezolvare a lui D1 aceasta va putea fi aplicată tuturor claselor de probleme pe care le descrie.

Cele două tipuri de mesaje cu care funcționează BGP reprezintă practic extreme: detecția performantă versus viteză. Nu se poate alege o variantă ideală atât ca și detecție cât și ca și viteză, fie și numai din cauza acestor opțiuni strict distincte.

Din prezentarea celor trei variante de BGP, se mai deduce încă o concluzie și anume că timpul necesar rulării acestui algoritm este deosebit de consistent. Prin relaxarea ipotezelor (WBGP) se poate obține un timp mai rezonabil, dar în defavoarea performanței.

Interesant de remarcat e faptul că cele două modele matematice: cel propus și cel clasic nu se exclud, ci se completează. Modelul propus este bazat pe o abordare de tip black-box, în timp ce modelul clasic, fiind croit exact pe o sub-problemă ne descrie interiorul cutiei negre.

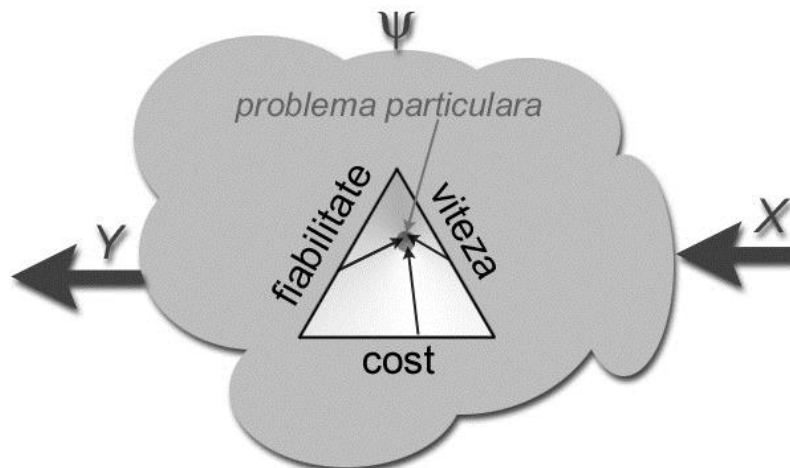


Figura 3.8. Relația între modelele matematice

Din figura de mai sus se evidențiază generalitatea modelului propus și limitarea pe un caz particular a modelului clasic. Piramida constructorului apare drept cuantificator al ipotezelor problemei particulare care se dorește a fi descrisă.

4. Intelligent Grid

4.1. Motivația arhitecturii

4.1.1. Etapele evoluției în știința calculatoarelor

Deschizând primul număr din 2008 al revistei Communications publicată de ACM, Legea lui Bell referitoare la nașterea și moartea claselor de calculatoare [Bell08] face lucrurile interesante prin prezicerea unui nou boom în următorii 2-3 ani.

Articolul începe cu o descriere a evoluției din 1950 până în 2010:

<<La începutul anilor 1950 o persoană putea merge în interiorul unui computer iar până în 2010 un singur computer (sau "cluster") cu milioane de procesoare se va fi extins la dimensiunea unei clădiri. Mai important, computerele încep să "meargă" în interiorul nostru. Aceste capete ale spectrului computației ilustrează vasta dinamică a puterii de calcul, dimensiunii, costului și a altor factori importanți pentru clasele secolului 21.>>

Noțiunea de clasă de calculatoare se referă la o serie de calculatoare, într-o marjă particulară de preț, definite prin mediul de programare și interfața utilizatorului în comunicarea cu alte sisteme pentru prelucrarea informației și alți oameni[Bell08].

Natura comună a programelor stocate pe calculator este aceea că un calculator poate fi programat să replice – emuleze – funcțiile altei clase. Astfel, o clasă poate include sau "omorâ" altă clasă.

Așa cum prezintă Bell, clasele de calculatoare mor sau sunt preluate de calculatoare cu preț mai mic care evoluează mai rapid. Această idee e ilustrată în Figura 4.1.

Un alt factor important în analiza cauzelor nașterii și morții claselor de computație este că generalitatea învinge întotdeauna specificitatea. Altfel exprimat, un calculator construit pentru o funcție specifică nu va fi preferat în fața unui calculator care poate îndeplini o multitudine de funcții.

Prețul supercomputerelor tinde să crească în timp, dar performanța per cost tinde să scadă în timp. Sub-clasa în formare va deveni într-o zi noua clasă, având aceeași performanță cu clasa curentă la un preț mai mic decât aceasta. Când diferența de preț este suficient de mare și sub-clasa are o piață suficient de largă de desfacere, sub-clasa devine noua clasă. Toate acestea se întâmplă pe fondul unei tendințe constante de a reduce prețul computerelor specializate care folosesc cele mai recente tehnologii.

Partea tehnologică evoluează însă constant, ceea ce duce la apariția unei noi generații de computere la fiecare 10 de ani.

Prima generație (1950-1960) a fost formată din computere bazate pe tuburi electronice, în timp ce cea de-a doua (1958-1970) a fost marcată de invenția tranzistorului, fiind astfel compusă din calculatoare bazate pe tranzistoare.

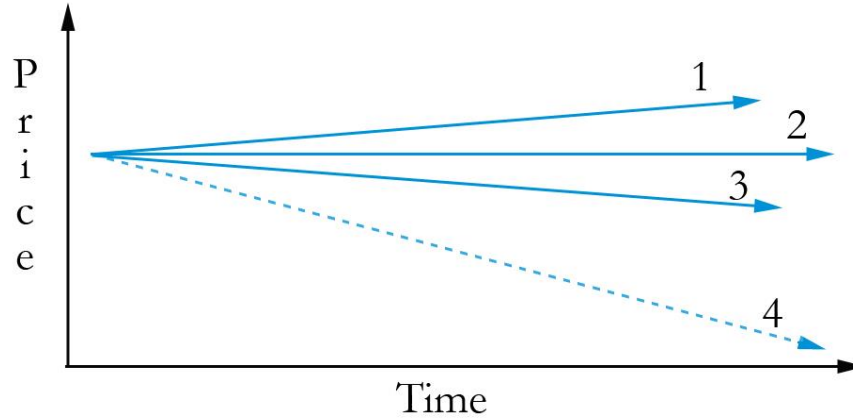


Figura 4.1. Evoluția claselor de calculatoare [Bell08]

1- Supercomputere the largest computers of the day; 2- Preț constant, creștere a performanței; 3- Formarea sub-clasei; 4- Noi calculatoare de preț "minimal" – aplicații și industrii noi

Cea de-a treia clasă (1965-1985) a fost bazată pe către circuitele integrate TTL și ECL. Circuitele integrate MOS și CMOS au inițiat a patra generație începând cu 1971.

Clasele de computere, așa cum sunt definite de către Bell, sunt grupate în valuri.

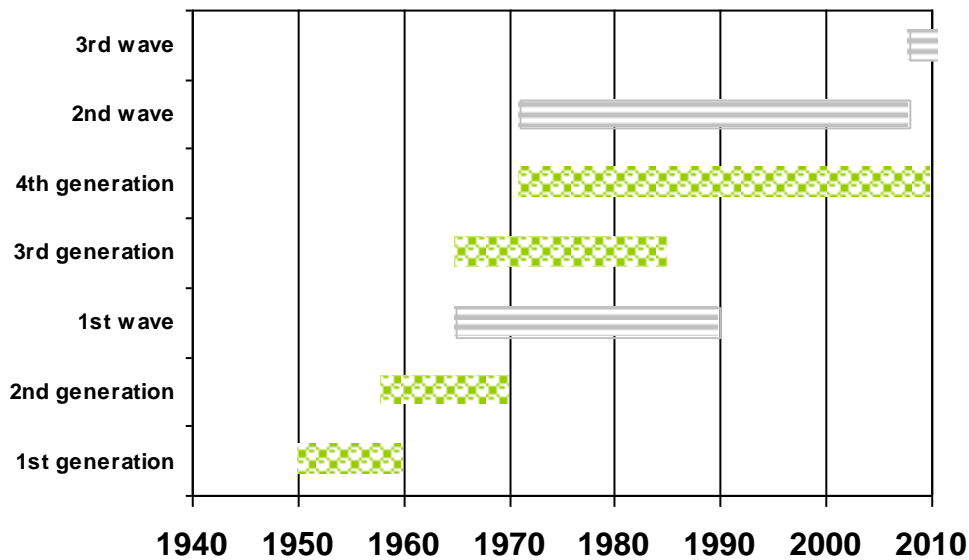


Figura 4.2. Clasele de calculatoare și valurile de computație

În primul val (1965-1990), înțelesul a fost centrat pe un singur dispozitiv, cum ar fi de exemplu mainframe-ul; în timp ce în având maximul în anii 1990, cel de-al doilea val (1971 - prezent) a fost în axat pe legătură. Calculele și accesul la informație într-un mediu distribuit au fost punctul central de interes. Cel de-al doilea val de calcul, a adus în atenție conexiunea. Inițial conexiunea s-a realizat între diferite dispozitive, la rularea acestora în colaborare, pentru a utiliza caracteristicile lor speciale. Primul pas din acest val au fost, prin urmare, rețelele. Următorul pas a fost conexiunea de diferite rețele cu alte rețele – astfel născându-se grid-ul. Scopul acestui tip de sistem a fost, inițial, mărirea puterii de calcul disponibilă utilizatorului. Aceasta a însemnat că partea cea mai importantă a devenit puterea de calcul neutilizată a dispozitivului, nu caracteristicile sale.

Cel de-al treilea val, care începe acum, se axează pe colecția de dispozitive și interfața lor reactivă. Acest val de calcul reprezintă un nou mod de interacțiune între electronice și persoanele umane [WeRA05].

Unele caracteristici interesante implicate de acest al treilea val sunt implicita toleranță la defectare (dacă o componentă nu funcționează corect, obiectivul poate fi realizat în continuare) precum și includerea de obiective și de constrângeri în interfață.

Este de așteptat ca în jurul anului 2010, al treilea val să conducă la computerul personal la nivel mondial.

Toată această discuție referitoare la modul cum evoluează calculatoarele are drept scop marcarea momentului actual drept unul în care o clasă de calculatoare practic este depășită de către alta.

Aceasta înseamnă că orice model teoretic nu este de ajuns să-l testăm pe o arhitectură actuală, deoarece în acest caz durata de viață a acesteia este destul de limitată. Ideal ar fi să-l testăm direct pe noua clasă de computație, dar aceasta nu este încă complet funcțională.

Astfel, am hotărât să studiem arhitecturile de rețele actuale și evoluția lor, precum și arhitectura corespunzătoare celui de-al treilea val pentru a vedea ce tip de arhitectură ar fi fezabil pentru testarea modelului teoretic.

4.1.2. Rețele ce includ senzori

Dacă în anii 80 rețelele au fost doar "o curiozitate academică", în anii 90 folosirea lor a fost extinsă de la universități și întreprinderi mari, la realitate zilnică a milioane de oameni. Arhitectura [Tane03] și răspândirea lor s-au schimbat de asemenea, pe tot parcursul anilor: la mijlocul anilor 90, au existat numeroase tipuri de rețele LAN și WAN, împreună cu mai multe stive de protocol. În jurul anului 2003, singurul LAN în utilizarea generalizată a fost Ethernetul, și aproape toate WAN-urile au trecut pe Internet. Prin urmare, destul de interesant, Internetul nu este o singură rețea, ci o rețea de rețele. În prezent, rețele fără fir sunt în creștere exponențială.

Definiția unei rețele de senzori [ChKu03] este: un grup de traductoare specializate, cu o infrastructură de comunicații destinate pentru a monitoriza și a înregistra condițiile de la diverse locații.

De la început este afirmat că, într-o rețea de senzori, caracteristică este capacitatea dispozitivelor interconectate de a interacționa cu mediul înconjurător

(“pentru a înregistra”) și de capacitatea de a procesa informațiile înregistrate (“a monitoriza”).

Fiecare nod – senzor este echipat cu un traductor, un microcomputer, un transceiver și sursa de energie. Traductorul generează semnale electrice pe baza fenomenelor fizice. Microcomputerul procesează datele primite de la senzor. Transceiverul primește comenzi de la un alt computer și transmite datele la acel computer. Energia pentru fiecare nod – senzor este derivată fie de la rețeaua electrică, fie de la o baterie internă.

Fiecare componentă a unui nod a evoluat în timp, în principal datorită evoluției tehnologiilor pe care sunt bazate aceste componente.

Evoluția în timp a rețelelor de senzor privită din diverse puncte de vedere (dimensiune, arhitectură, putere, implementare) este descrisă în tabelul următor.

	1980's-1990's	2000's	2010
Manufacturer	Custom contractors	Commercial: Crossbow Technology, Inc. Sensoria Corp., Ember Corp.	Dust, Inc and others to be formed
Size	Large shoe box and up	Pack of cards to small shoe box	Dust particle
Weight	Kilograms	Grams	Negligible
Node architecture	Separate sensing, processing and communication	Integrate sensing, processing and communication	Integrate sensing, processing and communication
Topology	Point-to point, star	Client server, peer to peer	Peer to peer
Power supply lifetime	Large batteries, hours/days	AA batteries, days to weeks	Solar or scavenging; months to years
Deployment	Vehicle-placed or air-drop single sensor	Hand-emplaced	Embedded, “sprinkled” left-behind

Tabelul 4.1. Cele trei generații de rețele de senzori [ChKu03]

Fiabilitatea este o problemă în rețelele ce includ senzori, din cauza multitudinii de atribute care caracterizează o asemenea rețea.

Sensors	<p>Size: small (e.g., micro-electro mechanical systems (MEMS)), large (e.g., radars, satellites)</p> <p>Number: small, large</p> <p>Type: passive (e.g., acoustic, seismic, video, IR), active (e.g., radar)</p> <p>Composition or mix: homogeneous (same types of sensors), heterogeneous (different types of sensors)</p> <p>Spatial coverage: dense, sparse</p> <p>Deployment: fixed and planned (e.g., factory networks), ad hoc (e.g., air-dropped)</p> <p>Dynamics: stationary (e.g., seismic sensors), mobile (e.g., on robot vehicles)</p>
Sensing entities of interest	<p>Extent: distributed (e.g., environmental monitoring), localized (e.g., target tracking)</p> <p>Mobility: static, dynamic</p> <p>Nature: cooperative (e.g., air traffic control), non-cooperative (e.g., military targets)</p>
Op. environment	Benign (e.g., factory floor), adverse (e.g., battlefield)
Communication	<p>Networking: wired, wireless</p> <p>Bandwidth: high, low</p>
Processing architecture	Centralized (all data sent to central site), distributed (located at sensor or other sites), hybrid
Energy availability	Constrained (e.g., in small sensors), unconstrained (e.g., in large sensors)

Tabelul 4.2. Atributele rețelelor de senzori [ChKu03]

Problemele [Chev04] implicate de rețelele de senzori sunt atât hardware cât și software ținând de platformă – gateway platform, low-power processing, smart sensor devices, redundanță – sau de rețea – topologie, rutare, probleme de lățime de bandă, de agregare și filtrare de date.

Cu toate acestea, piața de desfacere imensă și în plină expansiune a rețelelor de senzori, fac ca acest tip de rețele să fie în atenția cercetătorilor.

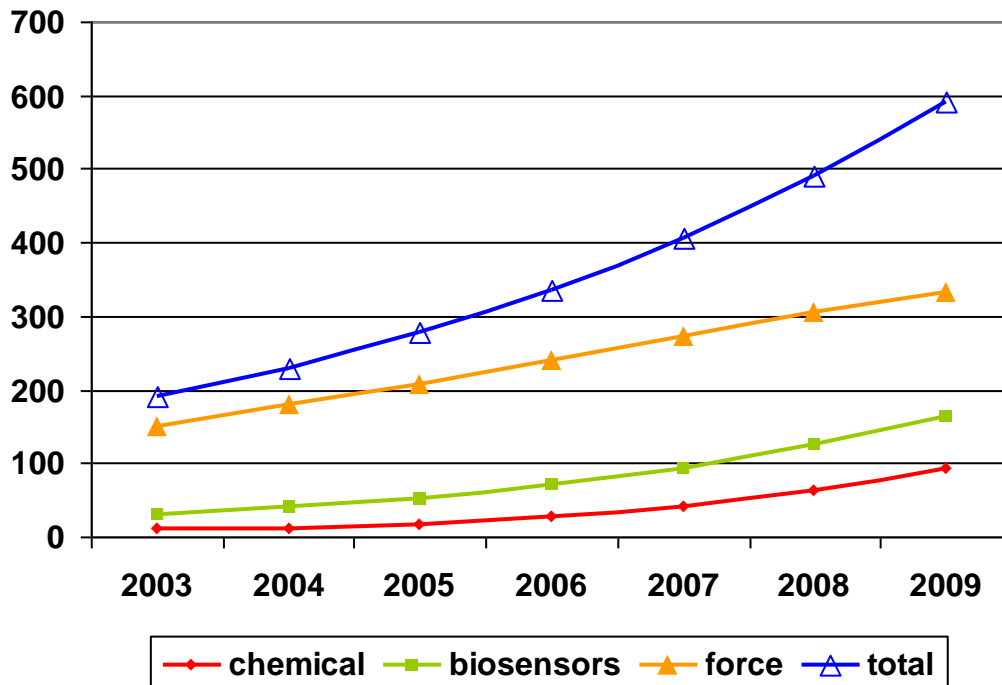


Figura 4.3. Piața senzorilor (mil. de \$) – estimări prelucrate din [OEMa05]

În paralel cu dezvoltarea rețelelor de senzori, în ultimul deceniu a existat o creștere a naturii aplicațiilor orientate pe date din cadrul grid computing. Pe măsură ce tehnologiile din spatele rețelor de senzori au evoluat, oamenii de știință au lucrat la integrarea [Chak07] de senzori și de actuatori cu sistemele de calcul de scop general.

Capacitatea de a simți a fost întotdeauna o parte integrantă a oricărui computer [Want07]; de altfel, o tastatură este doar o colecție de senzorii de presiune care preiau de la utilizator comenzi pentru sistemul de operare. De aici și până la integrarea rețelelor de senzori cu rețele de calculatoare a fost doar un pas.

Integrarea rețelelor de senzori cu rețele de calculatoare reprezintă practic adăugarea de "ochi" și "urechi" la sistemele computaționale [ThBu05]. Prelucrarea și modelarea de date în timp real combinată cu puterea de calcul permite răspunsuri și decizii aproape instantanee la scară largă. Gama standard de aplicații include monitorizarea mediului înconjurător pentru a putea fi avertizată populația la apariția unei calamități naturale sau detectarea rachetelor, precum și urmărirea și interceptarea lor.

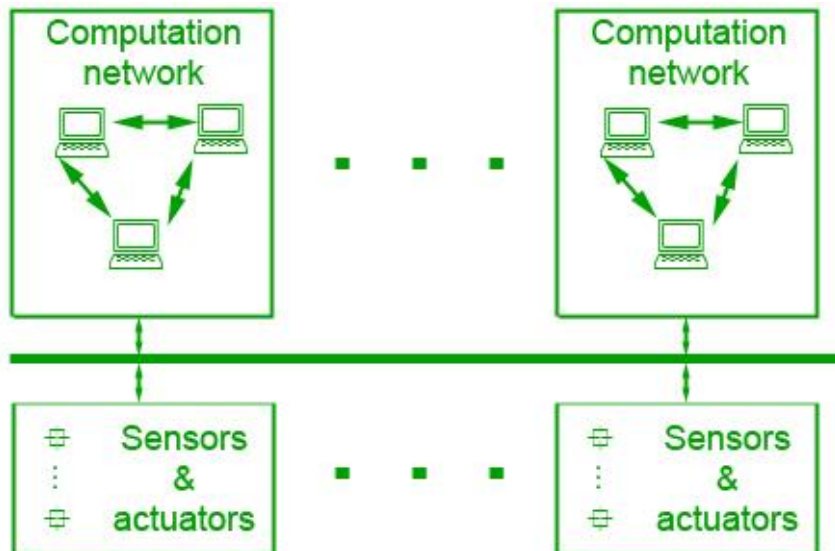


Figura 4.4. Arhitectura sensor-grid

Pentru implementarea unui sensor grid (Figura 4.4) există două tipuri principale de abordări: soluție centralizată sau soluție distribuită.

Versiunea centralizată este cea mai simplă de pus în aplicare, rețeaua va emite comenzile adecvate de acționare, dar prelucrarea datelor și luarea deciziilor nu se realizează aproape de sursa care le-a generat [ThBu05]. Drept urmare, această abordare are un slab nivel de toleranță la erori, datorată prăbușirii întregului sistem în lipsa comunicării.

Versiune distribuită permite prelucrarea datelor și luarea deciziilor la mai multe niveluri decât versiunea centralizată, eliminând astfel, principala problemă care duce la o toleranță la defectare scăzută a versiunii centralizate.

Ideea de a SOA (Service-Oriented Architecture) pentru o rețea de calcul nu este nimic nou, dar în contextul sensor grid, această abordare permite în primul rând descoperirea, accesul precum și schimbul de servicii între diverse noduri ale rețelei.

O altă facilitate oferită de SOA este partajarea aceleiași infrastructuri sensor-actuator de un număr de diferite aplicații și utilizatori.

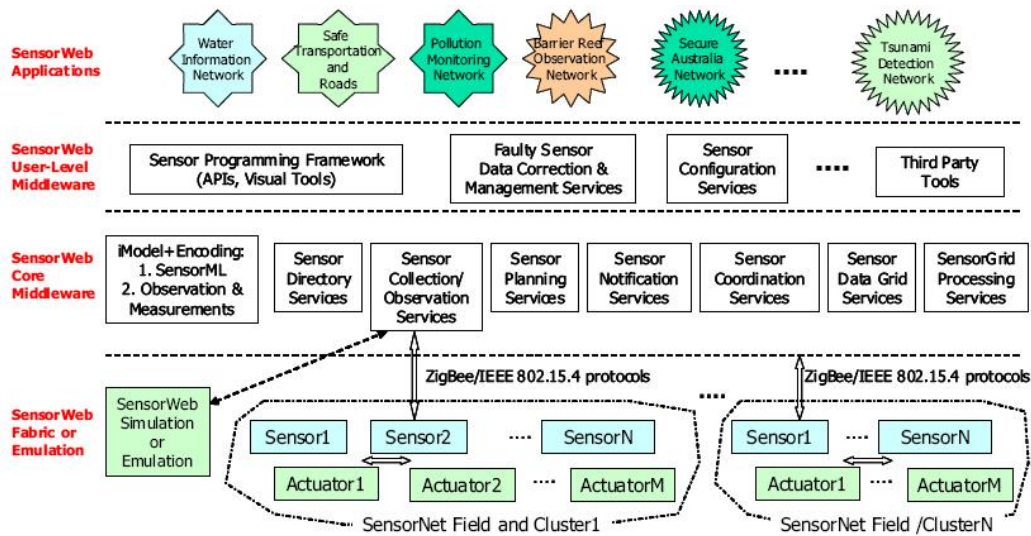


Figura 4.5. Arhitectura Open SensorWeb [ThBu05]

O arhitectura actuală interesantă, bazată pe SOA, este Open SensorWeb Architecture, prezentată pe larg în [ThBu05].

Un prim motiv care face acest arhitectura interesantă este faptul că este pusă în aplicare practic pe scară largă, de către NICTA / Universitatea Melbourne. Din punct de vedere tehnic, acest lucru este bazat pe Web Services și standardul SensorML.

O caracteristică interesantă a Open SensorWeb (Figura 4.5) este prezența middleware-ului [ThBu05]. Prezența middleware-ului permite un nivel mai ridicat de abstractizare atunci când se programează aplicațiile, astfel se tolerează ușor înlocuirea senzorilor la nivelurile inferioare.

Middleware este împărțită în două sub-niveluri: unul de bază (esențial pentru funcționarea rețelei) și unul la nivel de utilizator. Aceasta din urmă permite ca principalele funcții de agregare a datelor, de stocare și de recepție a lor să fie transparente pentru utilizator, și în același timp, facilitățile avansate, cum ar fi detectare erorilor sau corectarea datelor să fie analizate și la nevoie actualizate la orice moment de timp.

Dezavantajul acestei soluții este dat de faptul că programele trebuie să fie lansate de un utilizator, un program nu poate lansa mai multe instanțe ale sale pe alte mașini, el nu poate schimba mașina pe care lucrează, iar interacționarea cu alte platforme este limitată.

Un alt dezavantaj al acestei arhitecturi este faptul că serviciile sunt disponibile la nivelul middleware-ului. Ele pot fi accesate de diversele aplicații, dar aplicația în sine nu poate dezvolta un serviciu propriu, care să fie accesat de alte aplicații. Se poate eventual scrie un serviciu (de exemplu un protocol de consens) la nivelul middleware-ului care mai apoi va putea fi descoperit și accesat de toate aplicațiile care sunt interesate în folosirea sa.

4.1.3. Rețele ce includ multimedia

O altă categorie de rețele dedicate unui anume scop sunt rețelele ce oferă servicii specifice multimedia. Conținutul multimedia joacă un rol din ce în ce mai important în afacerile ce includ comunicații. Aceste rețele oferă servicii de tip best-effort service și ar trebui să fie capabile să îndeplinească cerințele de livrare ale aplicațiilor interactive. Calitatea serviciilor este, de asemenea de mare importanță, și aceasta ar trebui să fie adaptată pentru livrarea de conținut multimedia.

Termenul "multimedia" se referă la un spectru de clase utilizate pentru a reprezenta informații. Clasificarea datelor din cadru multimedia le împarte în trei grupuri:

1. de tip text
2. vizuale
3. sunet.

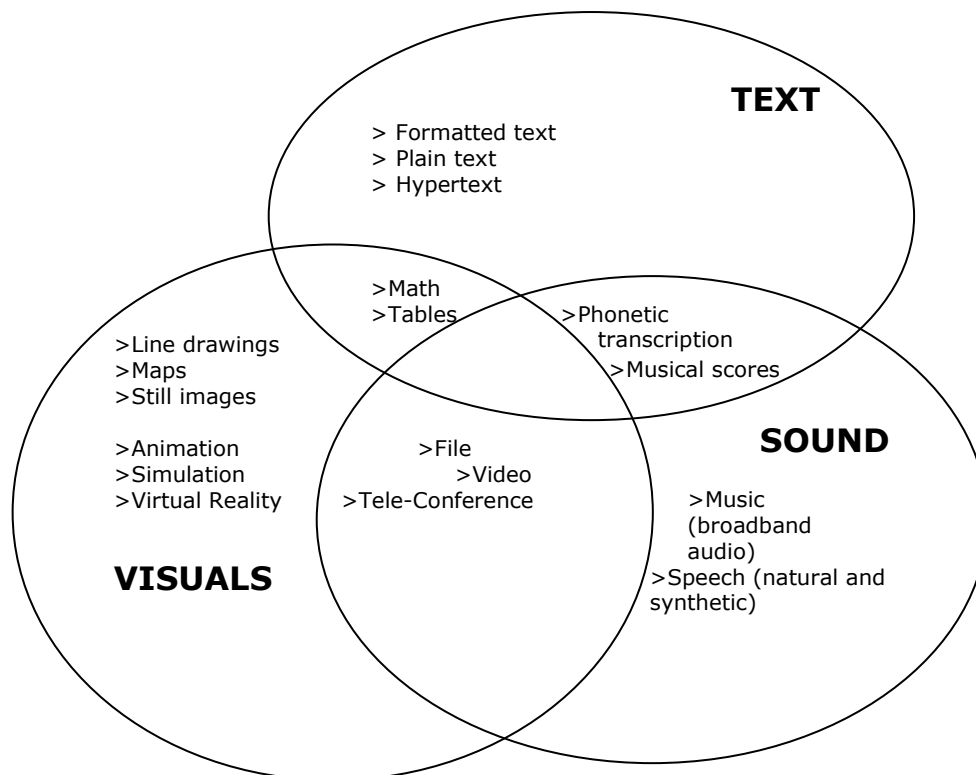


Figura 4.6. Diversitate de semnale multimedia[Kins02]

În figura anterioară, textul poate fi simplu sau formatat cu caractere de control, expresii matematice, partituri de muzică, transcrierea fonetică a vorbirii sau chiar alte reprezentări simbolice, cum ar fi hypertext.

Cea de-a doua categorie – elementele vizuale – poate include desene, hărți, imagini și fotografii, animație, simulare și tele-video-conferințe, obiecte din realitate virtuală.

Categoria sunet poate include telefonía – reprezentarea vocii – wideband audio pentru reproducerea de muzică, sunete și chiar înregistrări cum ar fi electrocardiograme sau alte semnale biomedicale.

Multimedia reprezintă una dintre piețele importante din lumea IT. Pe măsură ce apar diverse platforme de servicii multimedia, furnizori de servicii construiesc propria lor arhitectura media fără fir, mutând practic divertismentul la domiciliu clienților.

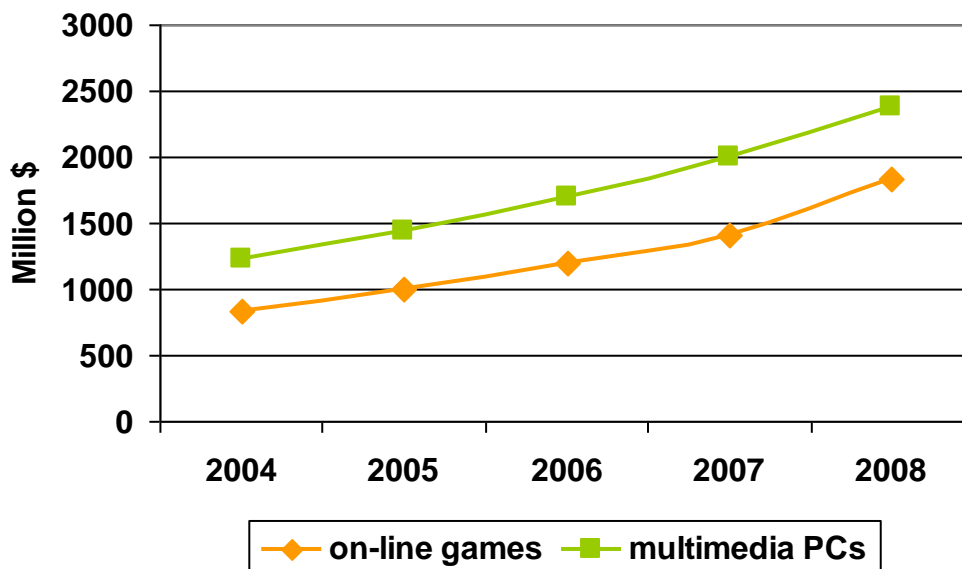


Figura 4.7. Dezvoltarea economică a pieței multimedia – prelucrare din [w3em05] și [w3ve07]

În ceea ce privește toleranța la erori a acestui tip de rețele este foarte greu de obținut o soluție care întreține QoS (Quality of Service), atât de important în cazul acestor rețele.

Clasic, atunci când nu sunt prezente erori, fiecare proces creează puncte locale de control (checkpoints), utilizând o anumită capacitate de stocare, cu un grad ridicat de stabilitate, pentru a salva cu informațiile sale de stare[OsHi01]. Când apare un eșec, acesta este punctul de recuperare. Cu toate acestea, acest punct local de recuperare pune problema de consistență a datelor în cazul unui control la nivel global.

Rețele multimedia au cerințe specifice, cum ar fi mesaje de mari dimensiuni și execuție lipsită de erori, prin urmare, prin scindarea verificării consistenței[OsHi01], coerența generală a datelor poate fi atinsă. Evident, acest tip de asigurare de toleranță la erori, presupune o capacitate suplimentară de stocare

necesară recuperării datelor și presupune de asemenea existența timpului necesar pentru această recuperare.

Pe de altă parte, în cadrul sistemelor real-time multimedia fluxurile de date multimedia constă din frame-uri video și audio, aranjate într-o ordine predefinită. Datorită stricteții constrângerilor de timp, QoS este satisfăcută doar atunci când toate secvențele video și audio sunt redade constant și în ordinea corespunzătoare într-o perioadă determinată[ZCCL06].

Din cauza limitării capacității de servicii a sistemului multimedia, QoS va fi inevitabil degradată dacă prea multe cereri sunt admise simultan în sistem. Astfel, sistemul trebuie să rezerve resursele în avans, pentru a putea admite cererile și pentru a garanta viitoarea lor performanță.

Această procedură este cunoscută ca rezervare de resurse. Într-o grid de servicii de sistem, capacitatea sistemului este foarte mult îmbunătățită, comparativ cu sisteme de servicii tradiționale, dar pot fi un număr mult mai mare de cereri potențiale.

În același timp, un sistem comercial grid permite gestionarea de diverse tipuri de cereri, cu priorități diferite. Dacă mai multe cereri cu prioritate mai mare sunt îndeplinite în mod corespunzător, sistemul poate obține profituri mai mari.

În acest mod logic, o evoluție a rețelelor multimedia se prezintă sub numele de multimedia grid.

În [BAKY03, BAKY02] este prezentat Multimedia Grid (mmGrid) ca o arhitectura creată pentru a sprijini aplicații multimedia într-un mediu de grid computing.

Ideea acestui proiect este de a oferi suport pentru aplicații din domenii precum grafică, vizualizare, streaming media și tele-imersiune și, de asemenea, de a oferi un mecanism de provizionare pentru resurse de calcul.

Pentru că o multimedia grid este concentrată pe grafică, atât vizualizare cât și flux de, iar în viitor, teleimersiune, rezervarea în avans a benzii de transmisie este o cerință cheie pentru o bună performanță. În plus este necesară capacitatea de a face rezervări în avans a unei stații de lucru pentru aplicații interactive. Aceste probleme sunt, de obicei, rezolvate prin proiectare a unui planificator (scheduler).

Într-o rețea bazată pe servicii care se confruntă cu un număr mare de cereri cu diferite servicii de profit și semnificație variată, întotdeauna va exista un trade-off între profitul sistemului și calitatea serviciului (QoS) [ZCCL06].

În astfel de sisteme, controlul admisiei joacă un rol important: sistemul trebuie să aibă o strategie pentru a face decizii de admitere de control și de rezerve de resurse pentru astfel de cereri care vin pentru a realiza un cât mai mare profit, fără încălcarea QoS a cererilor deja admise. Prin analize teoretice complete și simulări pe scară largă, a fost demonstrat[ZCCL06] că strategia cu mai multe niveluri prag este mai eficientă și mai flexibilă decât strategiile existente pentru sisteme de servicii bazate pe multimedia grid.

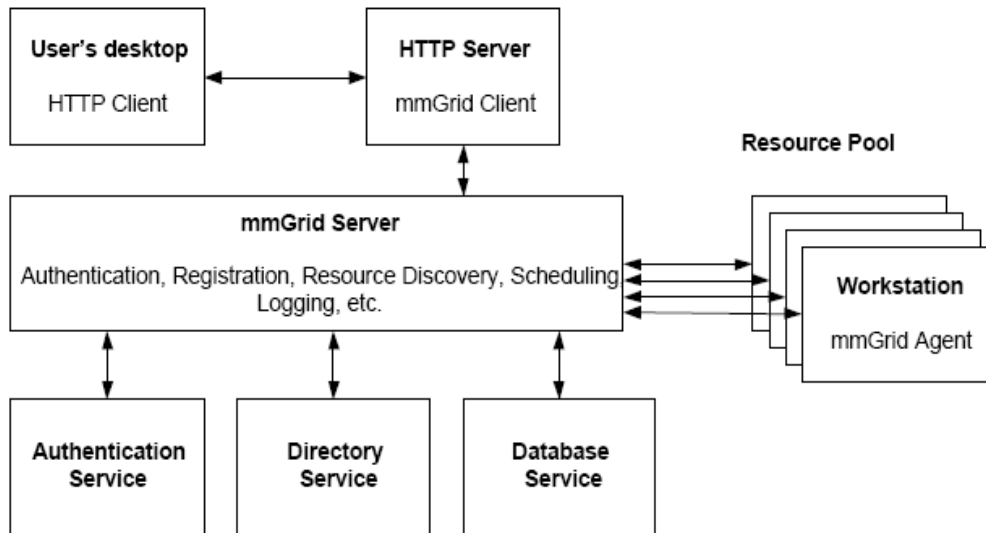


Figura 4.8. Arhitectura mmGrid [BAKY03]

Prin urmare, într-o rețea de servicii, rezervarea de resurse este foarte necesară și un mecanism de control al admisiei este sigur de a avea un impact enorm asupra eficienței și a profitului din întregul sistem.

Fiind una dintre puținele aplicații de multimedia grid efectiv realizate practic, mmGrid prezintă o arhitectură interesantă de a fi analizată.

În primul rând trebuie remarcată folosirea agenților, câte un agent pentru fiecare workstation. În această arhitectură bazată pe agenți, fiecare dreptunghi din Figura 4.8 anterioară reprezintă un calculator separat, cu toate că unele dintre serviciile ar putea rula pe același computer, cum ar fi serviciile de autentificare și directorul de servicii.

Agenții sunt poziționați pe toate resursele și răspund la un server. Acesta este un punct slab din punctul de vedere al toleranței la defectare, datorită faptului, că, dacă acest server cade, întreaga rețea s-a prăbușit.

Pe server există directorul de servicii, un serviciu tipic pentru o platformă bazată pe agenți, în timp ce serviciul de autentificare este caracteristic la grid și serviciul de bază de date este o caracteristică a rețelelor de date.

O altă caracteristică interesantă este posibilitatea de interfațare cu alte platforme software, în speță cu http, aceasta fiind o trăsătură comună între Open SensorWeb și mmGrid. Aceasta înseamnă că cele mai notabile soluții din prezent pentru grid multimedia și sensor grid au caracteristici comune.

4.1.4. Ambient Intelligence

Conform [WeRA05, MaRS06, și Raba05] există un anumit potențial în fuzionarea senzorilor și actuatorilor cu multimedia, rezultatul fiind ambianța inteligentă –ambient intelligence (AmI).

AmI reprezintă un nou mod de interacțiune între electronice și persoanele umane și care, conform legii lui Bell, aparține celui de-al treilea val de computație (care a adus în atenție colecțiile și interfața reactivă).

Numele de "ambianță inteligentă" este definit [Boek02] astfel:

- toată tehnologia este încorporată și ascunsă în fundal
- arhitectura este sensibilă, adaptativă și răspunde la prezența oamenilor și obiectelor
- susține activitățile prin asistență inteligentă, non-explicită
- păstrează securitatea, confidențialitatea și încrederea, prin utilizarea de informații numai atunci când este necesar și adecvat

Idee de bază se poate rezuma astfel: persoana umană este în centru, iar electronicele sunt invizibile în fundal.

Paradigma de "ambianță inteligentă" diferă în două moduri majore comparativ cu generațiile anterioare de calcul [WeRA05].

Mai întâi de toate, interfața cu utilizatorul a devenit reactivă, astfel spus acțiunile sale nu sunt solicitate în mod explicit, ci sunt rezultatul simplei prezențe a oamenilor sau a avatarurilor asociate lor (desigur, cu toate obiectivele și constrângerile explicite sau implicite).

În al doilea rând, sensul computației nu mai pot fi asociat cu un singur dispozitiv sau un set de dispozitive conectate, ci este situat în "colecția de dispozitive". Aceasta înseamnă că un singur eșec a unei componente nu implică imposibilitatea realizării scopului final.

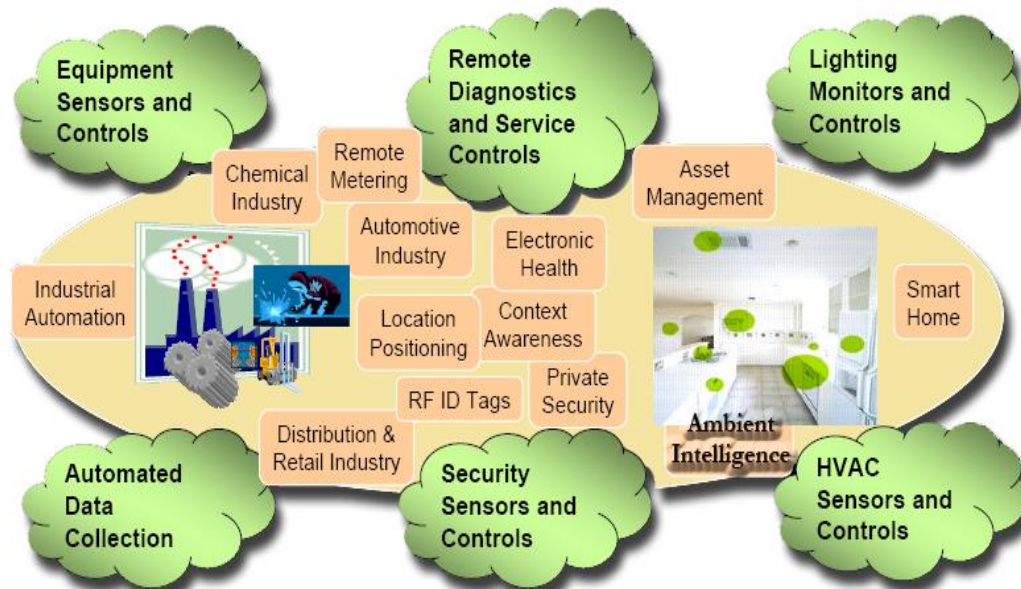


Figura 4.9. Evoluția care conduce la Ambient Intelligence [Chev04]

Ambient Intelligence articulează o nouă paradigmă pentru electronice de consum pentru dincolo anul 2010.

Conceptul care stă la baza AmI a fost dezvoltată la sfârștul anilor 90 într-o serie de ateliere de lucru interne, care au fost comandate de Philips[Aart06], și care a condus la o viziune a unei lumi complet integrate, ușor de utilizat, împrejurimi inteligente care să conducă la susținerea bunăstării, auto-exprimării și a productivității.

Această viziune s-a dezvoltat în ultimii ani într-o bază solidă pentru strategia companiei Philips numită Stil de viață și a campaniei de brand "Sense and Simplicity" care se bazează pe cele trei promisiunea făcute clienților

- proiectat in jurul tau
- ușor de experimentat
- avansat

Această descriere este un mijloc excelent de a explica AmI oamenilor nespecialiști în domeniul electronic. Realizările deosebite obținute de Phipils sunt de Active AmBi Light TV si Ambient Experience.

Fiind paradigmă atât de nouă, acest concept ridică probleme incepand cu componente de preț redus, electronică de putere ultra mică sau cu viteză superrapidă de procesare de semnal pentru o experiență vizuală satisfăcătoare.

Scopul ambianței inteligente poate varia de la sprijinirea contactului interpersonal uman, prin oferirea de informații și de orientare, ori de câte ori este necesar, de exemplu la controlul de securitate sau al sănătății umane.

Drumul spre Ambient Intelligence este plin de obstacole [MaLo06] sub formă de provocări pentru cercetare foarte eterogene, multe discipline diverse, dar totuși puternic legate. Având în vedere importanța și impactul pe care Ambient Intelligence le poate reprezenta în societate, toate problemele legate de Ambient Intelligence sunt primite cu un interes sporit din partea comunității de cercetare și un sprijin puternic industrial. În cele din urmă, agenții guvernamentale, organismele de standarde, și factorii de decizie sunt, de asemenea, un element-cheie în acest drum pentru a producerii de Ambient Intelligence viziune

Pentru a realiza aspectele legate de convergență și integrare ale AmI este nevoie de cooperare la mai multe niveluri. Cercetătorii din diverse domenii vor trebui să lucreze împreună pentru a ajunge la descoperirile necesare la nivel de tehnologic, precum și pentru a asigura acceptarea de către societate a produselor și proceselor puse la dispoziție de tehnologie.

Dintr-o perspectivă a rețelei apar concepte ca "internet al obiectelor", "rețele de senzori omniprezente" sau pur și simplu "Viitorul Internet".

Din perspectiva software-ului se pune accent pe "Societatea Informațională Centrată pe Servicii".

Pentru cercetători în micro și nano-electronică și în putere de calcul este avută în vedere spațiu de stocare și putere de calcul cvasi-infinite



Figura 4.10. Ambient Intelligence [w3Ph08]

Din punct de vedere al utilizatorilor, cu toate acestea, securitatea, confidențialitatea, interoperabilitatea și ușurința de utilizare vor fi cele mai importante puncte de pe ordinea de zi. Toate aceste puncte vor determina în final acceptarea AmI de către oameni.

În următorii ani s-a prezis apariția unor reușite deosebite tehnologice [Bus06] spre furnizarea de Ambient Intelligence (AMI), în programul nostru de zi cu zi.

O convergență între creșterea de calcul, comunicarea și difuzarea programelor, tehnologiile geospatiale bazate pe locație, rețele de senzori și miniaturizarea conduc practic către AmI.

Din punct de vedere economic, conceptul este foarte ademenitor, pentru că cererea este în creștere în toate sub-piețe de calcul, de comunicații și electronice de consum.

În discursul ținut la Prima Conferința Internațională AmI 2006, domnul Jacques Bus, Șef de Unitate, D4 - TIC pentru încredere și siguranță DG Societate Informațională, Comisia Europeană susține ideea că a AmI mare potențial pentru industria europeană de cercetare și de a lua o duce promițătoare, în multe activități antreprenoriale cu noi produse și procese inovatoare.

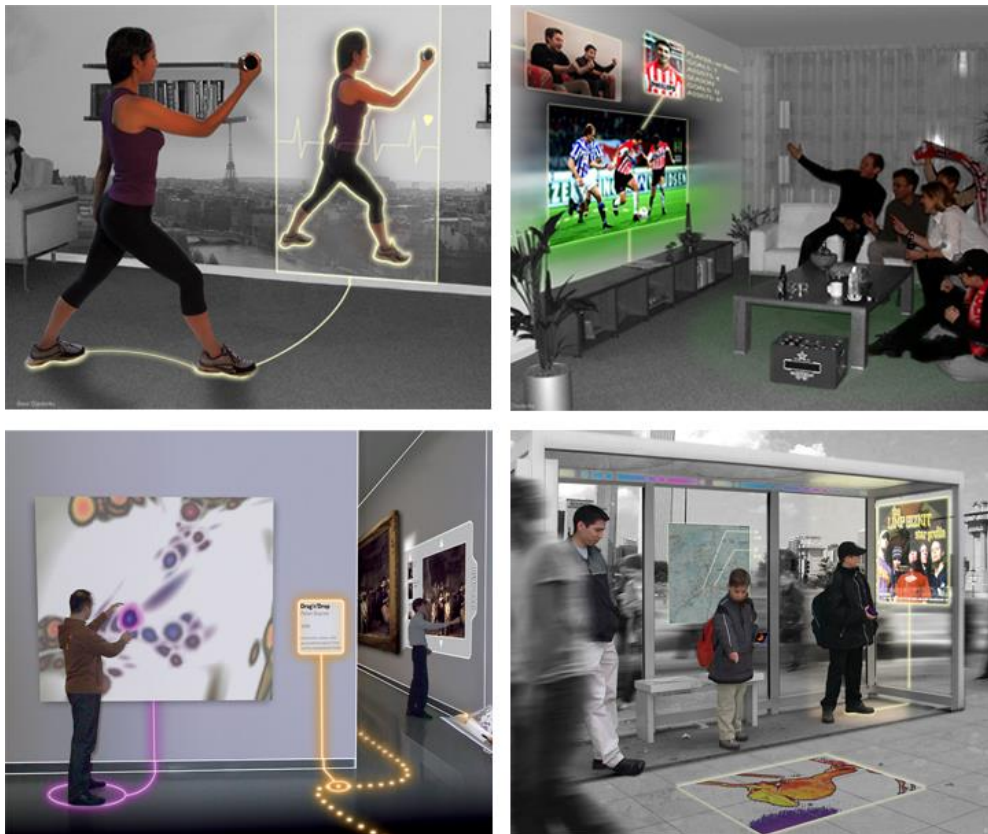


Figura 4.11. Posibile utilizări ale AmI [w3Ph08]

Al șaptelea program-cadru comunitar pentru cercetare(FP7) practic este construit[Bus06] pentru a permite dezvoltarea AmI în Europa.

În cadrul programului cadru anterior (FP6), pe lângă Philips, altă firmă: Fraunhofer Gesellschaft, a îmbrățișat conceptul și proiectul lor InHaus[w3In08] reprezintă acum o concurență pentru Philips HomeLab [Aart06, w3Ph08].

Rezumând, un nou concept, numit Ambient Intelligence este pe cale să schimbe fața lumii așa cum o cunoaștem. Una dintre principalele sale cerințe este toleranța la defectare, precizată ca atare în însăși definiția AmI. Dar diversitatea dispozitivelor implicate în cadrul AmI face ca această sarcină să fie extrem de dificilă.

Un prim punct de plecare îl constituie diversele tehnici de tolerare a defectelor din rețele de senzori și multimedia, dar cu toate acestea, se prezintă oportunitatea apariției de tehnici noi, inovatoare, datorită acestui paradigm atât de nou și inovator la rândul său.

4.1.5. Grid computațional

Acționând în alt domeniu decât AmI, conceptul de grid computațional s-a născut la mijlocul anilor 1990, când au existat numeroase proiecte de cercetare în curs de desfășurare în cadrul comunității academice care s-au axat pe distribuția calculului.

La Conferința Super Computing IEEE / ACM din 1995 de la San Diego, 11 rețele de mare viteză au fost utilizate pentru a conecta 17 site-uri cu scop de mărire a resurselor de calcul pentru o demonstrație în încercarea de a crea "metacomputerul". Această demonstrație a fost condusă de Ian Foster. Șaizeci de diferite aplicații [Fost00], originare din diferite facultăți, au fost elaborate pentru și au fost rulate această demonstrație de rețea.

Succesul acestei demonstrații a obținut atenția DARPA, în 1996, care a finanțat un proiect pentru crearea de instrumente de calcul distribuit. Proiectul a fost numit Globus și a fost condus de Ian Foster de la Argonne National Labs și Carl Kesselman de la Universitatea din California de Sud. Echipa a creat o suită de utilitare care au pus bazele grid computing în comunitățile academice și de cercetare.

La Conferința Super Computing din 1997, 80 site-uri la nivel mondial care rulează pe baza Globus Toolkit au fost conectate împreună.

Acest efort a început să fie menționat ca Grid Computing – nume creat pentru a face o analogie la rețeaua de energie electrică. [Wald02]

Începând de la sfârșitul lui 2000, grid computing a devenit un concept rentabil economic[Abba04]. Mari corporații, cum ar fi IBM, Sun Microsystems, Intel, Hewlett Packard, și chiar unele companii mai mici, cum ar fi Platform Computing, Avaki, Entropia, DataSynapse caută a crea următoarea generație de Grid Computing, care este axată pe aplicații de afaceri, mai degrabă decât de cercetare academică.

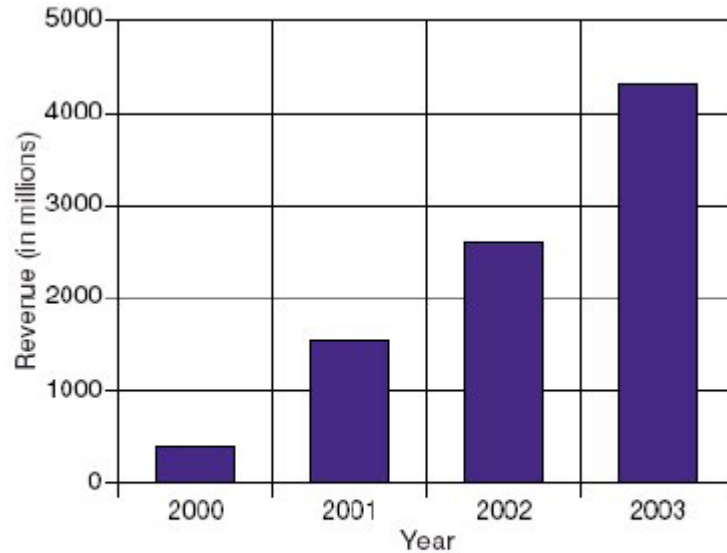


Figura 4.12. Estimare a pieței de grid computing [w3Su02]

Foarte puțin a fost publicat, cu detalii de la o aplicație comercială [Smit04], în general gridurile fiind folosite în comunitățile academice și de cercetare.

Funcțional, se poate vorbi de mai multe tipuri de grid [KrBM02]:

- Grid computațional, care sunt în principal concentrate asupra operațiunilor computațional intensive.
- Grid de date care sunt specializate în repartizarea și gestionarea de cantități mari de date distribuite.
- Grid de servicii care prestează servicii ce nu pot fi furnizate de către o singură mașină

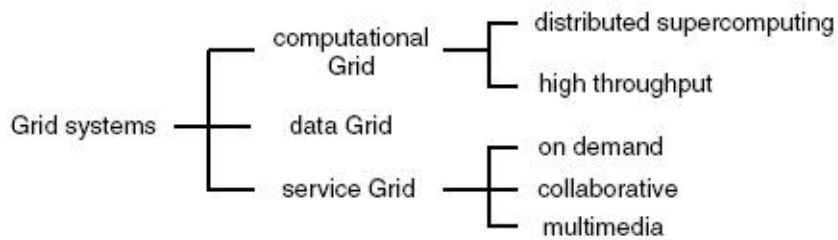


Figura 4.13. Taxonomia sistemelor Grid [KrBM02]

Din gridurile de date fac parte sensor grids care au fost deja discutate, iar din gridurile de servicii fac parte multimedia grids, de asemenea discutate anterior.

Definiția clasică a computational grid este dată în [FoKe98], ca fiind o infrastructură hardware și software, care oferă acces demn de încredere, consecvent, omniprezent și necostisitor la capacități computaționale extinse.

În această categorie de grid sistemele computaționale au capacitate totală mai mare disponibilă pentru aplicații decât capacitatea existentă pe orice mașină din sistem.

Un supercomputing grid distribuit, execută aplicația în paralel pe mai multe sisteme, pentru a-i reduce timpul de finalizare. De obicei, aplicațiile care necesită supercomputing sunt probleme mari, cum ar fi modelare de date meteo și simulare nucleare. High throughput grid crește rata de finalizare a unui flux de sarcini și sunt bine adaptate pentru cererile precum simulări Monte Carlo.

În crearea unui grid, trebuie să se îndeplinească anumite cerințe [FLYL05]:

- hardware: datorită faptului că un mediu de rețea este constituită din resurse de calcul, care, în condiții normale sunt pur și simplu computere, este considerată sursa de putere de calcul și capacitatea de stocare de date. Prin urmare, fiecare computer individual trebuie să aibă resurse suficiente de putere de calcul și capacitatea de stocare a datelor pentru a rula corect pe platformă și trebuie să mai știe cine e entitatea care preia cererile sale la grid.
- software: trebuie să existe o interoperabilitate între platformele grid bazate pe diverse resurse de calcul și software-ul trebuie să permită comunicarea între orice pereche de resurse de calcul (fie directă sau indirectă)
- resurse umane: trebuie să existe cel puțin o persoană capabilă să efectueze sarcini administrative la nivel înalt (cum ar fi: modernizarea platformei, gestionarea conturilor și aplicațiilor) și cineva, care oferă suport tehnic pentru alți utilizatori.

Am menționat anterior, orice rețea trebuie să fie susținută de o platformă. Pentru ca acest lucru să fie posibil, trebuie ca platforma să asigure mai multe caracteristici de fiabilitate[FLYL05], precum:

- Monitorizare la distanță a resurselor de calcul
- Păstrarea de informații de logare despre toate activitățile efectuate de pe platforma
- Controlul accesului la platforma
- Securizarea datelor schimbate în cadrul platformei;

Toleranța la defectare este o proprietate importantă în cadrul computational grid, pentru că nu există certitudine cu privire la resurse în principal pentru utilizarea lor în afara zonei definite, este o mare posibilitate ca o resursă utilizată este dăunătoare în vreun fel. Un prim pas pentru putea crea toleranța la erori, este de a crea un model al erorilor și al defectelor posibile. În ceea ce privește tehnicile utilizate, se consideră o bună tehnică este redundanța sarcinilor de executat, servicii de consens, programare dinamică de activități[ToXu01].

4.2. Arhitectura Intelligent Grid

Tendința evolutivă este folosirea simultană a unei varietăți cât mai mari de echipamente, extinzând în același timp fiabilitatea totală a sistemului. Arhitectura cea mai apropiată de acest deziderat, AmI, este proiectată pentru a fi utilizată într-un mediu potrivit pentru a fi controlat; prin urmare, principala sa țintă este casa „inteligentă”. [WeRA05]

Într-un astfel de mediu, pe lângă senzori, actuatori și prelucrare multimedia, un alt participant ar putea apărea: PC-ul. Acest participant nu este luat în considerare în abordarea AmI.

Pe de altă parte, astăzi conceptul de ambient multimedia este centrat pe un PC (de exemplu, Microsoft Media Center, Apple FrontRow, și HP Digital Center). Acest PC controlează toată rețeaua care asigură serviciile multimedia. În cazul în care PC-ul central cade, întreaga rețea este blocată – nu mai funcționează. Utilizatorul este de administrator de sistem, tot el realizând și configurarea, drept urmare acest concept este încă orientat pe conexiune și/sau dispozitiv.

Pentru a rezuma: în cadrul AmI, PC-ul nu este luat în considerare, iar în rețelele multimedia de astăzi PC-ul este principalul punct de susținere, în timp ce în aproape toate tipurile de rețele existente PC-ul este prezent.

Prin urmare, am încercat să adăugăm PC-ul la AmI. În această abordare, PC-ul este privit doar ca și un alt nod în rețea. Se poate muta (spațial), poate dispărea sau apărea, sau poate „dormi”. Când PC-ul este treaz și utilizatorul uman interacționează cu aceasta, el poate fi văzut ca și o entitate dublă: senzor-actuator. Partea de senzor „citește” utilizatorul și dorințele sale, iar partea de actuator interacționează cu utilizatorul prin oferirea de informații. Pe de altă parte, când PC-ul este treaz el ar putea oferi putere de calcul în rețea, prin urmare, poate fi folosit pentru a procesa operațiile. În acest mod, alte noduri din rețea pot dormi, rezultând reducerea consumului de energie.

Să presupune acum că nu adăugăm doar un singur PC. Având în vedere că în AmI nevoia de prelucrare a datelor este crescută, aceste PC-uri pot face o mare parte din prelucrarea datelor din spatele senzorilor și actuatorilor. Dar aceasta înseamnă că PC-urile ar trebui să formeze o rețea (computațională)[AnBV08a].

Numele ales de noi pentru a această arhitectura este de „intelligent grid” de la AmI („inteligentă ambientală”) și de la „grid computațional”.

Pornind de la definiția AmI, vom defini o „intelligent grid” – notată IG – astfel

- *majoritatea* tehnologiei este încorporată și ascunsă în fundal
- arhitectura este sensibilă, adaptativă și răspunde la prezența oamenilor și obiectelor
- susține activitățile prin asistență inteligentă, non-explicită
- păstrează securitatea, confidențialitatea și încrederea, prin utilizarea de informații numai atunci când este necesar și adecvat
- *poate primi în mod explicit cereri de la utilizatori pentru a acționa*

• *pentru aplicațiile în care este necesară putere computațională, caută să folosească puterea nefolosită din rețea*

Cu italice sunt diferențele între definiția AmI și cea a IG.

Se naște însă următoarea întrebare: această arhitectură aparține celui de-al doilea sau de-al treilea val de computație? Răspunsul plasează IG ca fiind o arhitectură de tranziție, până la apariția noului val de componente specifice AmI. Este o arhitectură din cel de-al treilea val de computație **emulată** pe o arhitectură aparținând celui de-al doilea val.

Aceasta ne aduce înapoi la Legea lui Bell. AmI este dedicată susținerii vieții umane, nu realizării de calcule complexe. Rețelele computaționale, pe de altă parte, nu sunt create pentru a opera cu senzori. IG, pe de altă parte, este mult mai general proiectată. Prin urmare, în conformitate cu Legea lui Bell, IG ar trebui să aibă mai mult succes decât rețelele specifice unui anumit scop.

Network \ Grid	Sensor grid	Multimedia grid	Ambient intelligence	Intelligent grid
Sensor network	X		X	X
Multimedia network		X	X	X
Computation network	X	X		X

Tabelul 4.3. Componerea grid-urilor[AnBV08a]

Este clar demonstrat că această rețea inteligentă include nu numai de calcul, multimedia, dar, de asemenea, și senzori de rețea, astfel, posibilitatea de a-și expune ambiant inteligentă calcul, practic în același timp.

Nivelurile propuse de acest model sunt ilustrate în Figura 4.14.

Piese de echipament (senzori, actuatori, controlere, PC-uri, DSPs), deși extrem de eterogene, sunt grupate toate la nivelul dispozitivelor.

Controlerele accesează periodic senzorii și bazat pe aceste valori, se poate interacționa cu mediul înconjurător, utilizând de actuatori. Aceasta înseamnă că fiecare controler poate executa activități complexe și, adeseori, pe termen lung, și poate iniția o acțiune chiar și fără un stimul explicit de la un utilizator.

Legăturile între toate aceste dispozitive, precum și conectarea la Internet sunt la nivel de conexiune, în timp ce nivelul colecțiilor include orice surse de date. Aceste surse pot fi fluxuri video, fluxuri de date sau chiar baze de date.

Fluxurile de date (de exemplu, fluxul de date al unei camere video) sunt trimise la server. Serverul poate oferi aceste date pe internet și dacă sunt resurse disponibile aceste date se poate procesa.

Dacă serverul principal este ocupat, datele se pot împărți și pot fi procesate de controlere disponibile.

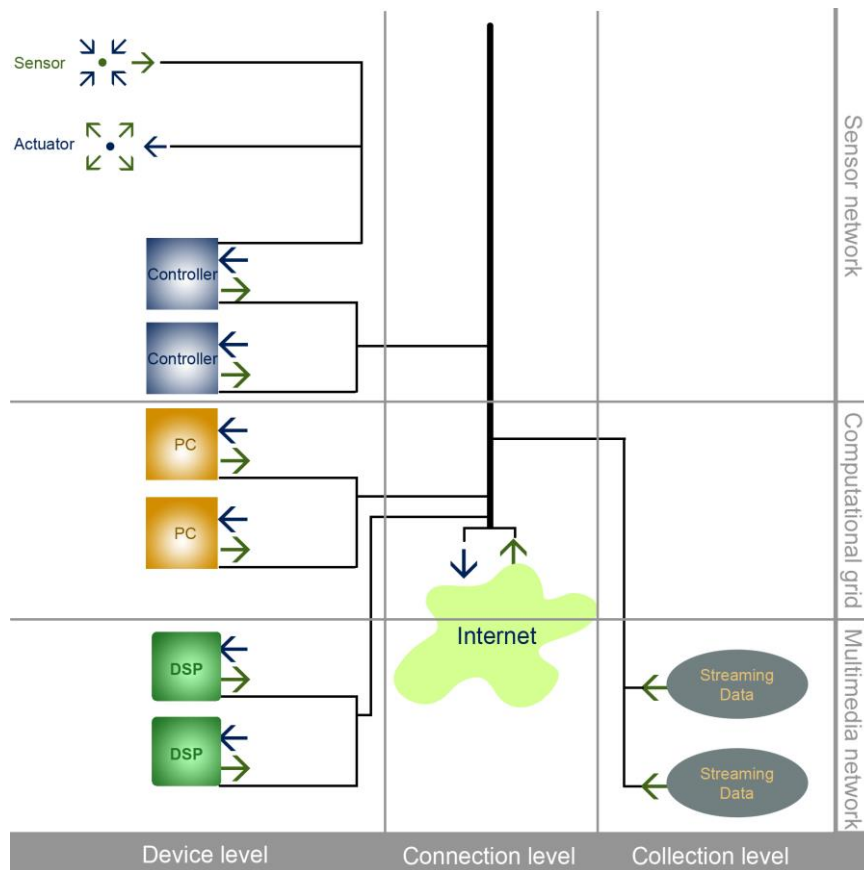


Figura 4.14 Intelligent Grid (IG) [AnBV08a]

Conform lui Rabaey, orice arhitectură trebuie să fie studiată din 4 mari puncte de vedere:

1. Portabilitate, scalabilitate și configurabilitate
2. Putere, cost și mărime
3. Fiabilitate
4. Securitate și confidențialitate

IG a fost studiată din aceste puncte de vedere și în această lucrare numai primele trei puncte de vedere sunt prezentate în detaliu.

Ultimul punct de analiză (securitate și confidențialitate) este dezvoltat pe larg de Razvan Bogdan.

Cu toate acestea, această problemă este atinsă la nivelul semnăturilor digitale, de interes fiind timpul necesar acestora, nu modalitățile specifice de obținere a semnăturilor.

4.3. Portabilitatea, scalabilitatea și configurabilitatea

Portabilitatea, scalabilitatea și configurabilitatea sunt o problemă ce implică software-ul care susține rețeaua.

În ipoteza cea mai defavorabilă, o schimbare a rețelei la nivel de hardware poate duce la schimbarea software-ului de pe toată rețeaua, ceea ce duce la necesitatea rescrierii tuturor programelor care rulează pe această rețea.

O soluție reală este considerată a fi de ridicare a nivelului de abstractizare [Raba05], a modului cum este privită întreaga arhitectură

Acest lucru este ilustrat în figura următoare.

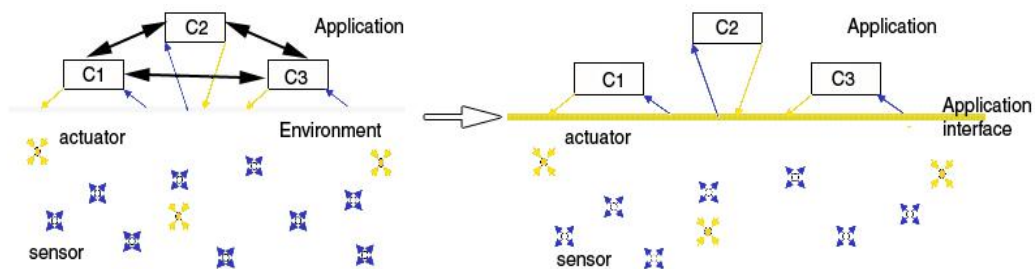


Figura 4.15. Ridicarea nivelului de abstractizare [Raba05]

Interfața aplicației acceptă servicii esențiale [Raba05], cum ar fi interogări, comenzi, sincronizarea ceasurilor pe rețea, localizare, etc și este similară cu conceptul de interfață socket pe Internet.

Aplicația de interfațare [Raba05] ne permite să:

- definim serviciile necesare pentru a exploata o rețea ce conține senzori
- nu există o relație de dependență între arhitectura rețelei și platforma hardware
- oferim posibilități de interoperabilitate și inovare

Avantajul folosirii unei aplicații de interfațare constă în existența unui singur punct pentru asigurarea interoperabilității, care realizează și relația de independență între arhitectura platformei de hardware și partea de software. Astfel se poate adăuga un nou tip de dispozitiv hardware iar acesta, odată programat în interfață, va putea fi folosit de către software-ul vechi, fără a fi nevoie să se rescrie acesta.

4.3.1. Alegerea middleware-ului

În faza de programare a oricărei rețele, mai întâi de toate este nevoie de o programare pentru abstracțiuni care simplifică tasking-ul, și pentru un middleware care acceptă astfel de abstracțiuni de programare [Raba05].

Cu toate acestea, trebuie să se ia în considerare faptul că rețeaua noastră este compusă din trei tipuri distincte de rețea, cu cerințele lor specifice.

Un rezumat al cerințelor de programare pentru fiecare tip de rețea este prezentat mai jos.

Programming requirements	Networks based on			
	or	Sens	Multi media	Com putation
Concealed issues	ware	hard and distribution	ware hard	distribution
Restricted Resources				
	Energy	X		
	computing power	X	X	X
	communication bandwidth	X	X	X
	Network Dynamics	high	um medi	Low
	Scale of Deployments	N*(100...1000)	N*(10...100)	N*(10...100)
Real-world Integration				
	Time scale	X	X	X
	Location scale	X	X	
Collection and Processing of Data				
	Preprocessing	X	X	
	Aggregating data	X	X	X
	Local processing		X	X

Tabelul 4.4. Privire de ansamblu asupra cerințelor de programare[AnBV08a]

În alegerea unui middleware trebuie să vedem la fiecare tip de rețea dacă îi va suporta toate cerințele. Există deja o serie de middleware pentru fiecare tip de rețea, și în următorul tabel vom prezenta aceste tipuri de abordări.

Type of approach	Networks based on		
	Sensors	Multimedia	Computation
Events	X		
Remote Procedure Call		X	X
Object Request Broker		X	X
Message-oriented			X
Databases	X	X	
Mobile (Intelligent) Agents	X	X	X

Tabelul 4.5. Tipuri de middleware si folosirea lor[AnBV08a]

Poate fi observat că tipul de abordare middleware ce acoperă toate tipurile de rețele care fac parte din IG este bazat pe agenți inteligenți.

Dintre toate middleware-urile orientate spre agent în folosință astăzi cel mai răspândit este JADE (Java Agent DEvelopment framework).

Jade este un sistem middleware complet distribuit [BeCG07], cu o infrastructură flexibilă care permite ușor extinderea cu module de tip add-on. Mediul facilitează dezvoltarea completă de aplicații pe bază de agenți printr-un run-time environment în care se poate defini ciclul de viață al agenților, logica lor internă, precum și o bogată suită de utilitare grafice.

Cum JADE este scris complet în Java, acesta beneficiază de imensul set de biblioteci oferite de terți și, astfel, oferă un bogat set de abstracțiuni de programare care permit dezvoltatorilor de a dezvolta ușor în JADE sisteme multi-agent.

JADE a fost inițial dezvoltat de către Departamentul de Cercetare și Dezvoltare din Telecom Italia Spa, dar acum este community project și este distribuit ca open source sub licența LGPL.

JADE este în concordanță cu specificațiile FIPA și standardele IEEE, specificațiile FIPA fiind standard IEEE începând cu iunie 2008.

4.3.2. Agenți inteligenți

Utilizarea agenților inteligenți ne permite să privim un controler / DSP / PC-ul ca și un agent care acesază capabilitățile hardware ale dispozitivului pe care rulează.

Agenții au mai multe caracteristici importante precum autonomia, capacitatea de a acționa proactiv și capacitatea de a comunica cu alți agenți. Aceste caracteristici le permit agenților să execute sarcini complexe și, adeseori, pe termen lung, sau chiar să inițieze o sarcină chiar și fără un stimul explicit de la utilizator. Capacitatea de a comunica cu alți agenți permite unui agent să interacționeze cu alți agenți, pentru a-și realiza propriul său scop.

Programare orientată pe agent (AOP) este un domeniu în care conceptele din inteligența artificială trec în domeniul sistemelor distribuite.

Aplicația în AOP este o colecție de componente numite agenți. Comunicarea între agenți este intrinsec peer to peer.

Tehnologia bazată pe agenți a fost subiectul unei ample discuții[BeCG07] în cadrul comunității științifice pentru mai mulți ani, dar doar recent, a apărut un grad semnificativ de exploatare în aplicații comerciale.

Sisteme multi-agent sunt folosite din ce în ce mai mult într-o largă varietate de aplicații, de la sisteme mici pentru asistență personală până la sisteme mari, complexe pentru aplicații industriale. Exemple de domenii industriale unde sisteme multi-agent au fost realizate și rulează cu succes includ controlul proceselor, sisteme de diagnosticare, de fabricație, de transport și logistică până chiar la sisteme pentru managementul rețelei.

Astfel, arhitectura de IG, așa cum este descrisă anterior implementând conceptul de agenți inteligenți este descrisă în Figura 4.16. IG modelată cu agenți

Folosind acest concept de agent, la nivel de dispozitiv au rămas doar senzorii și actuatorii, la nivel de legătură este prezent middleware-ul; în timp ce la nivelul colecțiilor avem agenții.

Agenții pot rula individual, pe o platformă hardware, sau pot rula mai mulți pe aceeași platforma hardware. Un lucru interesant este că, ridicând nivelul de abstractizare, aspectul fizic al platformei nu este relevant.

Singurul interes pentru a programa funcționalitatea rețelei este în realizarea agenților și, implicit, a modului în care aceștia interacționează și rezolvă problemele.

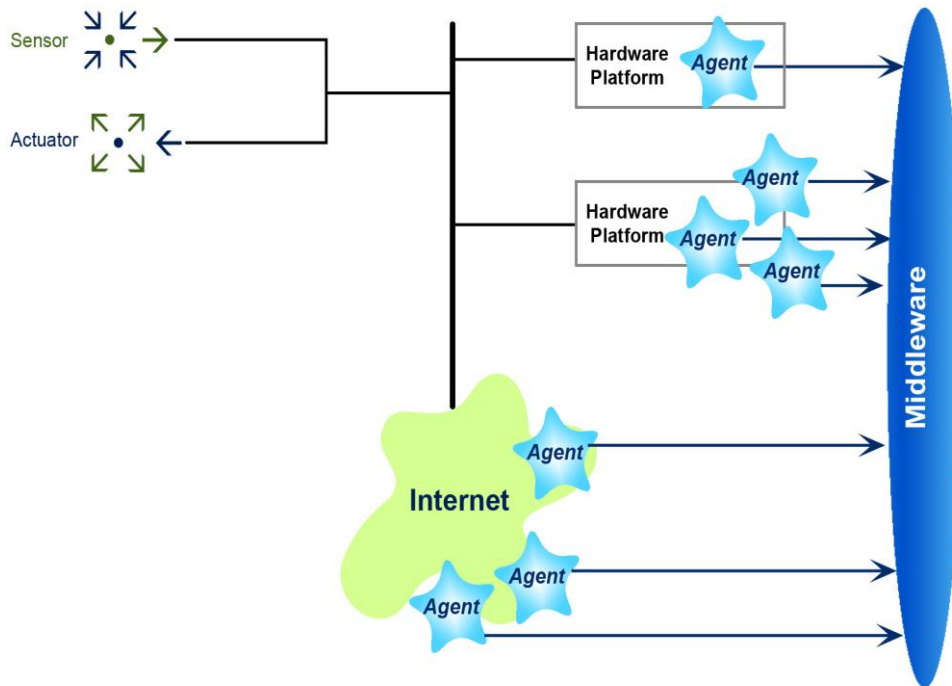


Figura 4.16. IG modelată cu agenți[AnBV08a]

Platforma hardware pe care este situat un agent pot fi schimbată în orice moment, datorită conceptului de mobilitate inter-platformă al agenților. Aceasta implică migrarea codului agenților de pe o platformă hardware pe alta, putând astfel, de exemplu, echilibra încărcarea rețelei.

Un alt factor interesant este faptul că agenții sunt situați, la nivelul de logic, pe platforme de agenți (platforme software) și agenții pot migra între de platforme software. Aceasta poate duce la acceptarea simultană a mai multor platforme software și chiar portabilizarea funcțiilor unui agent indiferent de platforma pe care rulează

Sistemele multiagent sunt considerate a fi cel mai bun mod de a caracteriza sau de a proiecta sistemele de calcul distribuite[HuST99]. Prelucrarea informației este prezentă în aproape fiecare aspect al vieții noastre zilnice. Procesoare de date sunt, în ziua de azi, încorporate în toate aspectele legate de mediul nostru.

De exemplu în bucătărie începând cu cuptorul cu microunde, prăjitorul de cafea sau chiar partea de curent electric prezintă procesoare embedded.

Numărul mare de procesoare și numărul practic infinit de moduri în care acestea interacționează, fac ca sistemele de calcul distribuite să fie paradigma dominantă de azi. Când, datorită evoluției, controlerele devin destul de inteligente pentru a fi considerate agenți, atunci devine convenabil să se gândească la ele în termeni antropomorfici[HuST99].

De exemplu, „prăjitorul de pâine prăjită știe când se face pâinea”, și „ibricul de cafea știe când este gata de cafeaua”. Când aceste sisteme sunt interconectate, astfel încât să poată interacționa, atunci ele ar trebui de asemenea să știe că cafeaua și pâinea prăjită și ar trebui să fie gata în aproximativ în același timp pentru micul dejun. Mergând mai departe, pentru a ști când este momentul pentru mic dejun, această informație trebuie obținută și trimisă agenților prăjitor de pâine și ibric de cafea. În aceste condiții, bucătăria devine mai mult decât o simplă colecție de procesoare – un simplu sistem de calcul distribuit – ea devine un sistem multiagent. Soluțiile centralizate sunt în general mai eficiente dar avantajul unei soluții distribuite rezidă din algoritmi de calcul care nu pot exista într-o soluție centralizată sau chiar de rezolvarea problemei prin procesul de distribuire în sine. Există, de asemenea, momente în care o abordare centralizată este imposibilă, deoarece sistemele și datele aparțin unor organizații independente care doresc să-și păstreze informațiile private și sigure pentru motive competitiv.

Agenții pot opera și exista în medii diverse, computaționale sau fizice. Deși există situații în care un agent poate funcționa util de la sine, creșterea interconectării și dezvoltarea rețelelor de calculatoare face o astfel de situație rară, făcând ca de obicei, un agent să interacționeze cu alți agenți. Pentru a putea studia modul de interacționare se poate considera că interacțiunile sunt de două feluri: trimitere (send) și recepție (receive) a doar două tipuri de informații: afirmații (assertions) și interogări (queries).

O clasificare a agenților așa cum este ea prezentată în [HuSt99], parte a [Weiss99] ne definește patru mari clase de agenți.

	Basic Agent	Passive Agent	Active Agent	Peer Agent
Receive assertions	*	*	*	*
Receive queries		*		*
Send assertions		*	*	*
Send queries			*	*

Tabelul 4.6. Capabilitățile agenților [HuSt99]

Protocoale de comunicare permit agenților să schimbe și să înțeleagă mesajele. Protocoale de interacțiune permite agenților să aibă conversații, care, pentru considera că sunt schimburi de mesaje structurate.

Ansamboul acestui sistem se regăsește sub denumirea de inteligență artificială distribuită (DAI) iar aceste sisteme de informare sunt vaste și au următoarele proprietăți [Weiss99]:

- pot fi distribuite din punct de vedere geografic

- pot avea multe componente
- pot avea un conținut foarte mare (de date / concepte)
- pot avea un larg domeniu de aplicare
- pot fi distribuite și eterogene (de obicei).

În acord cu această abordare, agenții trebuie să fie distribuiți și încorporați în întreaga sistem.

Pentru IG, agenții inteligenți ar putea funcționa ca și resurse de informare activă [HuSt99], "wrappers" care încadrează componente convenționale. Aceștia ar avea informații despre resursele locale, pe care rulează, și pot să coopereze între ei pentru a oferi acces la nivel global.

4.3.3. Aplicație practică

În faza de simulare am vrut să studiem cum se comportă JADE în comparație cu alt middleware. Pentru că partea interesantă în IG este adăugarea puterii de calcul la AmI testul nostru constă în compararea unui middleware tipic computațional, MPI cu JADE.

Pentru a face acest lucru, am modelat o simplă problemă care ne așteptăm să fie extrem de comună în cadrul IG (toy-benchmark). Există un număr de noduri computaționale care procesează de un număr de factori (S). Modul de prelucrare este foarte simplu (de complexitate polinomială).

Numărul de noduri și numărul de intrări poate fi variat și deoarece timpul de răspuns al întregii rețele este ceea ce ne interesează, vom măsura viteza aplicației de pe ambele middleware-uri.

În această privință, aceeași problemă a fost pusă în aplicare cu agenți JADE care au lansat mesaj CFP (apel pentru propuneri) de fiecare dată când au nevoie de o valoare de la un senzor.

Agentul la care e senzorul conectat direct răspunde, cu o valoare la propunere și în cazul în care valoarea este acceptată de agent se confirmă că le-a primit valoarea. Programul a fost executat cu un agent de pe fiecare nod computațional, astfel încât comparația cu MPI ar fi corectă.

Rezultatele comparative între MPI și JADE, pentru un număr constant de intrări sunt prezentate mai jos.

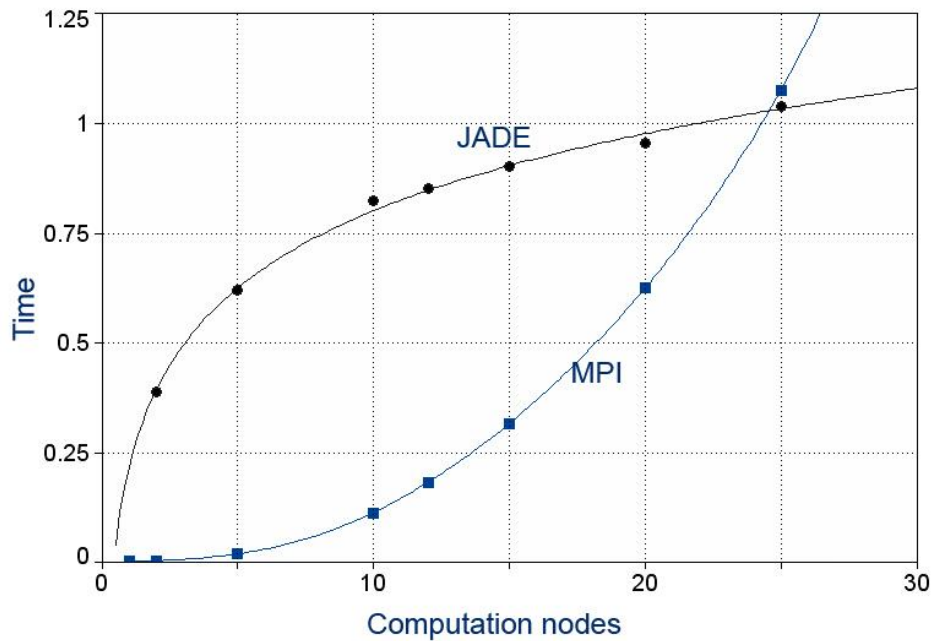


Figura 4.17. Timpii de rulare[AnBV08a]

Se poate observa că, pentru câteva noduri de calcul, timp de execuție global JADE este în mod clar mai mare decât cel obținut de MPI. La 25 noduri, overhead-ul comunicației pe varianta cu MPI o încetințește, astfel încât JADE este mai rapid.

Deoarece rulează pe Java, ne-am așteptat ca JADE să fie mai lent decât MPI, rezultatul practic confirmând acest lucru, dar fără diferențe mari de timp între cele două variante, iar portabilitatea codului și dependabilitatea timpului de rulare de scalabilitatea problemei, ne face să alegem totuși JADE.

4.4. Puterea, costul și dimensiunea

Discuția despre puterea, mărimea și costul componentelor este complicată din cauza de naturii eterogene a modelului de IG.

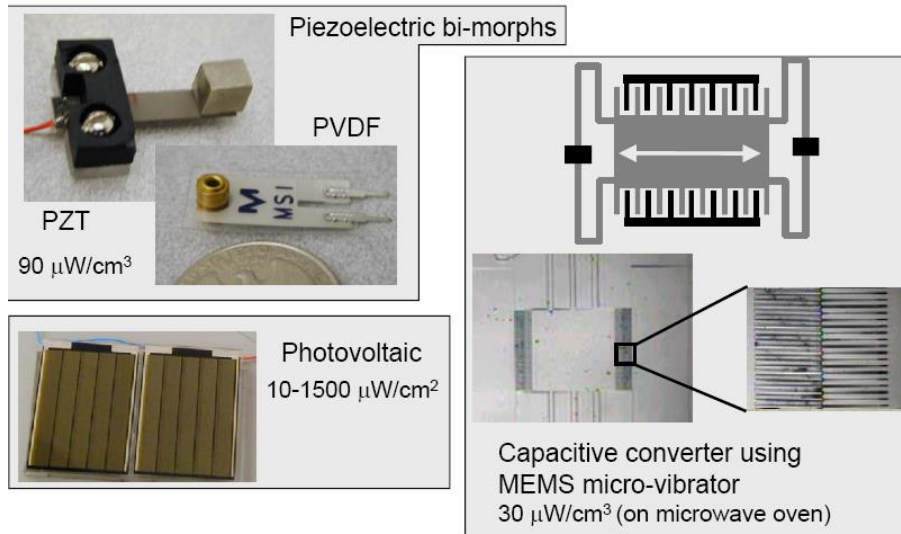


Figura 4.18. Realizări practice de scavenging de energie [Raba04]

Pentru a evalua puterea, mărimea și costul pentru senzorii, actuatorii și controlerile DSP care fac parte din IG, trebuie să discutăm despre „disappearing electronics”.

În prezent senzori la scară mică [Raba04] pentru achiziții de date sunt:

- complet integrați cu dimensiunea lor mai mică de 1 cm^3
- minimizarea puterii / energiei de disipare prin limitarea disipării puterii la $100 \mu\text{W}$
- oferirea posibilității de scavenging de energie
- sprijin pentru viteza redusă a datelor ($< 100 \text{ kbit / sec}$)
- formarea de rețele ad-Hoc auto-configurabile și robuste

Principala problemă în „disappearing electronics” este că aceste dispozitive de putere super-mică ar trebui să nu aibă nevoie de baterii sau sursă de tensiune externă, bazându-se în loc pe microgeneratoare.

Prin urmare, este nevoie de dispozitive care pot „recolta” energie, numite dispozitive de scavenging a energiei. În esență, aceste dispozitive pot produce propria lor de energie electrică din surse prezente în mediul ambiant.

Energia gratuită provine de la soare, din vibrații, din diferențe de temperatură și presiune, precum și din energia umană (energia solară, evident, fiind cea mai cunoscută și cea mai dezvoltată din punct de vedere tehnologic).

Power Source	P/cm³ (μW/cm³)	E/cm³ (J/cm³)	P/cm³/yr (μW/cm³/Yr)
Primary Battery	-	2880	90
Secondary Battery	-	1080	34
Micro-Fuel Cell	-	3500	110
Ultra-capacitor	-	50-100	1.6-3.2
Heat engine	-	3346	106
Radioactive (Ni63)	0.52	1640	0.52
Solar (outside)	15000	-	-
Solar (inside)	10	-	-
Temperature	40	-	-
Human Power	330	-	-
Air Flow	380	-	-
Pressure Variation	17	-	-
Vibrations		-	-

Tabelul 4.7. O comparație între sursele de putere [Raba04]

Multe soluții [WBDG05] au apărut în ultimii ani, deși în ultima vreme ideea de recoltare de energie [RBSS07] este abordată sistematic.

Toate acestea sunt încurajate de rezultatele din industrie [Swan07], care estimează că, în 2012, costul energiei solare fotovoltaice va fi de \$1.50/Watt care este prețul energiei de la rețeaua electrică.

Dezvoltarea de nanogeneratoare este înfloritoare. Două domenii principale sunt deschise cercetării agresive: nanoscale thermoelectric energy harvesting [VWSP07] și Nano-Piezotronics.

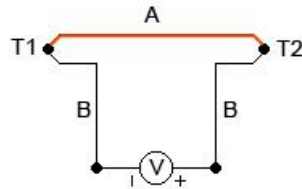


Figura 4.19. Efectul Seebeck

Prima abordare se bazează pe un fenomen numit efect Seebeck, care permite transformarea variațiilor de temperatură direct în energie electrică, pe baza unui film subțire termoelectric. Acesta este dezvoltat în cadrul unui proiect sponsorizat de DARPA.

Pe măsură ce chip-ul se încălzește datorită funcționării, pompa de căldură transformă această variație de temperatură direct în energie electrică [VWSP07], estimându-se că s-ar putea produce suficientă putere pentru a acționa un ventilator pe chip-ul care generează căldura.

Deoarece căldura produsă de computer este în întregime irosită, această soluție pare deosebit de elegantă pentru soluționarea problemelor de supraîncălzire a computerelor.

Privit în ansamblu [YeMH07] domeniul MEMS (micro-electro-mechanical systems) este o altă direcție tehnologică interesantă, iar în acest domeniu cea mai promițătoare abordare recentă este convertirea la scală nanometrică a energiei mecanice în energie electrică prin nano-fire de oxid de zinc [Wang07] care crează o descărcare electrică când sunt îndoite.

O serie de electrozi, într-o aranjare în zig-zag, este suspendată peste aceste fire minuscule și atunci când se produc vibrații, firele trec ca o perie peste electrozi, trimițând un flux de energie electrică.



Figura 4.20. Firele de oxid de zinc [Wang07]

În [RBSS07] sunt prezentați chiar senzori de presiune fără baterii aflați în cauciucuri inteligente cu traductoare care transformă puterea de accelerare a

impactului în energie pentru senzori. O astfel de soluție ar putea face cauciucurile să auto-măsoare toate aspectele legate de performanță în timp real.

O idee esențială în obținerea echipamentelor cu senzori care funcționează cu energie preț redusă și sunt de mici dimensiuni este considerată a fi simplitatea ceea ce conduce de obicei, la soluții fără fir [Raba04].

Aceasta lasă PC-urile pentru a fi analizate din punctul de vedere al puterii, de costului și al mărimii.

PC-urile este prognozat să dispară deja de ceva timp [Hall98], și fiecare companie majoră (de exemplu, IBM – cu conceptul său BladeCenter și stațiile de lucru – sau Microsoft [Gate03]) se pregătește pentru aceasta. Computerul viitorului poate avea doar o interfață cu utilizatorul, principalele sarcini de calcul fiind situate pe un server, probabil partajate cu alți utilizatori.

În afara scavenging pentru energie; există avansuri în arhitecturile procesoarelor pentru a maximiza regimul de așteptare (stand-by) și pentru a avea procesoare de putere ultra-reduce cu regimul de așteptare la rândul lui de putere redusă.

Invizibilitate tehnicii de calcul va fi ajutată de noi tehnologii precum ar fi chip-uri MEMS ieftine, de dimensiuni mici care sunt capabile de stocare a terabytes de date.

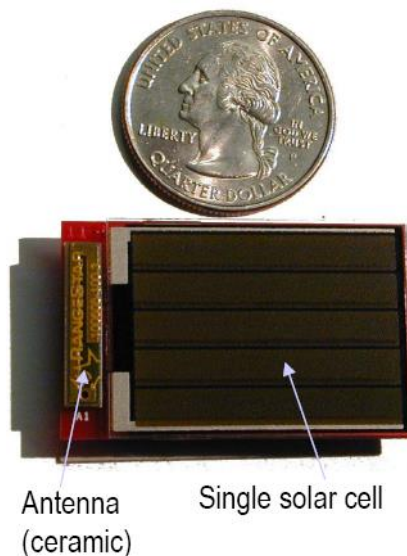


Figura 4.21. Small scale design - Proiect PicoRadio [Raba05]

Scădere costurilor electronicelor va face ușor pentru producătorii de PC de a include capabilități inteligente și de conectivitate și chiar în cele mai comune dispozitive. În cele din urmă, puterea de calcul în sine ar putea deveni aproape prea ieftină pentru a merita să fie măsurată.

Toate acestea vor duce la o schimbare fundamentală [Gate03] în modul în care vom percepe computere. Folosirea unui computer va deveni la fel ca și folosirea

de energie electrică, ca atunci când aprindem lumina. Computere, ca și energia electrică, vor juca un rol în aproape tot ceea ce faceți. Ele vor fi tot în jurul nostru, esențiale în aproape fiecare parte a vieții noastre, fără a le vedea efectiv – ele au „dispărut”.

Să luăm drept exemplu un procesor de energie redusă – SNAP / LE [EkKM04], pentru rețele de senzori asincrone – optimizat pentru operațiunile de monitorizare a datelor în rețelele de senzor.

Evoluțiile recente în auto-alimentare pe baza de MEMS pentru dispozitive de comunicare radiofrecvență ne va duce la auto-alimentare în întregime pentru acest procesor [EkKM04].

În loc de a minimiza energia convențională utilizată a unui microprocesor, în [EkKM04] a fost proiectat un nou microprocesor asincron care conține suport hardware pentru operațiunile care au loc frecvent în rețele de senzor; scopul fiind de a maximiza durata de viață a unei rețele.

SNAP / LE este condus pe evenimente, cu tranziții foarte scurte între perioade de activitate și inactivitate.

Rezultatul îl reprezintă un procesor cu valoare maximă de aproximativ 300 pJ/instrucție și valoare minimă de 100 pJ/instrucție.

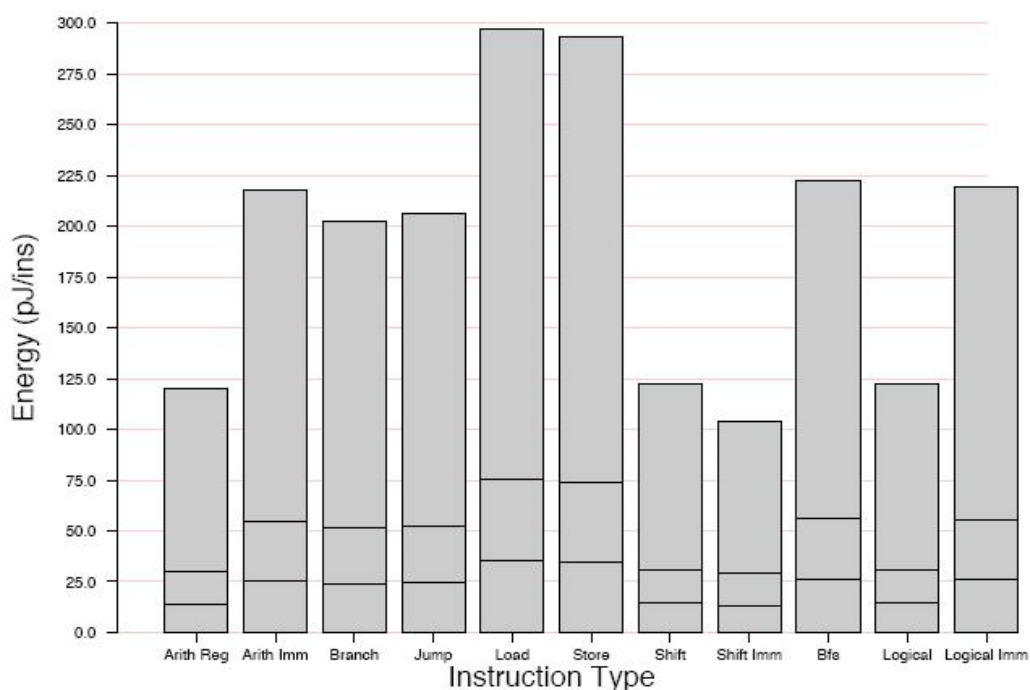


Figura 4.22. Energie per tip de instrucție [EkKM04]

Utilizarea de circuite asincrone dă rezultate bune în managementul fin al puterii, pentru că în circuitele care nu trebuie să efectueze o anumită operațiune, nu au nici o activitate de comutare în așteptarea unui eveniment.

Utilizarea circuitelor asincrone necesită, de asemenea, eliminarea pericolelor sau rateurilor apărute la comutări noi în procesor, eliminând astfel o altă sursă de consum de energie

Processor	Clocked	Speed (MIPS)	Datapath (bits)	Memory	Voltage (Volts)	E/ins (pJ)
Atmel Mega 128 L	Y	4	8	4-8K	3	1500
Intel X Scale	Y	200-400	32	16-32M	1.3-1.65	890- 1028
Dynamic Voltage Scaled Microproc essor	Y	7-84	32	16K	1.8-3.8	540- 5600
CoolRISC	Y	1	8	22K	2.4	720
Lutonium	N	200	8	8K	1.8	500
Aspro-216	N	24-140	16	64K	1-2.5	1000- 3000
SNAP/LE	N	28	16	8K	0.6	24
SNAP/LE	N	240	16	8K	1.8	218

**Tabelul 4.8. Comparație între procesoare de putere ultra-scăzută
[EKKM04]**

O altă facilitate a procesorului SNAP/LE este suportul hardware pentru executarea evenimentelor care elimină nevoia de a rula un sistem de operare. Neavând un sistem de operare nu numai că se reduce numărul de instrucțiuni statice și dinamice, dar de asemenea, se permite de a simplifica proiectarea de SNAP / LE, care nu are nevoie de a suporta excepții precise și excepții de memorie virtuală.

Design-ul final al acestui procesor seamănă cu acela al unui microcontroler, utilizându-se extrem de puțină energie per operațiune.

4.4.1. Implementare experimentală

În următorul paragraf, termenul nod se referă la un agent care acționează pe un nod și accesează toate capacitățile aceluși nod.

Ipotezele în care vom lucra sunt următoarele:

- Un nod știe care sunt nodurile dintr-o anumită zonă geografică. Modul în care acest lucru se realizează nu este important acum, el putându-se obține prin diverse mijloace (fie prin directorul de servicii, sau prin intermediul programării anterioare)
- Fiecare nod poate măsura (cu precizie echitabilă) propria lor capacitate de energie.
- Orice nod poate fi defect, și încă va transmite date (apărând astfel trimiterea de informații eronate)[LaPS80].
- Orice nod se poate recupera dintr-o stare eronată, în orice moment[Haye02]
- Pentru a măsura o valoare externă, un nod necesită mai multă energie decât pentru a măsura starea internă a bateriei sale.

Protocol propus în continuare l-am denumit Battery Aware Measurement (BAM) Protocol și poate, în conformitate cu ipoteza de securitate de relaxată, mări durata de viață a nodurilor și păstra lățimea de banda a rețelei.

Acest protocol poate fi util atunci când nu vrem să se trezească toate nodurile astfel încât acestea să poată afla toate o anumită valoare.

Într-un mediu de securitate mai ridicată, toate nodurile măsoară valoarea și o transmit apoi la nodul central care poate primi un set de valori identice. În această situație, nodurile utilizează resurse prețioase de energie, atunci când este de ajuns ca doar un singur nod să răspundă la cererea inițială. În același timp, lățimea de banda a rețelei pot fi limitată, din cauza acestei comunicări.

Protocolul BAM poate fi descris prin următoarele etape:

- Pasul 1: Un nod oarecare C vrea să afle o valoare care poate fi măsurată de n noduri, de aceea, C trimite o cerere pentru a afla de la cele n noduri statusul lor energetic
- Pasul 2: Nodurile răspund (într-un interval de timp – T), cu valoarea respectivă. Nodurile care răspund după ce timerul intern pentru T s-a scurs, sunt ignorate. Ar trebui să existe cel puțin un răspuns pentru a putea continua la pasul următor. Dacă nu sunt răspunsuri primite, protocolul raportează eroare.
- Pasul 3: Nodul cu cea mai multă energie este rugat apoi a măsura valoarea. În caz că există mai multe răspunsuri cu aceeași valoare pentru energia rămasă, unul din noduri este ales aleator.
- Pasul 4: Nodul selectat măsoară și trimite valoarea măsurată la C sau un mesaj de eșec, în cazul în care măsurarea nu s-a putut efectua.
- Pasul 5: C își stabilește valoarea internă la valoarea primită.

Să considerăm, pentru simplitate, că toate nodurile au aceleași valori de energie necesare pentru a măsura o valoare – V și au nevoie de aceeași cantitate de energie pentru a măsura starea bateriei sale interne – B .

Din ipoteză:

$$V > B$$

Ecuția 4.1.

Acest lucru poate fi transformată în egalitate dacă vom defini ε drept valoarea care face diferența dintre energia necesară pentru a măsura valoarea și suma de energii necesare pentru a trimite un mesaj și pentru a măsura starea bateriei interne.

$$\varepsilon = V - B$$

Ecuția 4.2

Trebuie să se ia în considerare energia necesară pentru a primi un mesaj – R și energie pentru a trimite un mesaj – M , cu scopul de a defini complet energia necesară, pentru un singur nod care nu măsoară valoarea, pentru a finaliza protocolul (E_p).

$$E_p = R + B + M$$

Ecuția 4.3.

Prin urmare, pentru $n-1$ din nodurile implicate în protocolul BAM totalul de energie necesară este E_m .

Cu toate acestea, nodul singular care măsoară valoarea are un consum de energie diferit: E_m .

$$E_m = R + B + M + V + M$$

Ecuția 4.4

Astfel se poate deduce totalul de energie cerut de Protocolul de BAM – E_{BAM} , având în vedere că toate nodurile răspund în limita de timp (nici un mesaj nu este pierdut).

$$\begin{aligned} E_{BAM} &= E_m + (n-1) \times E_p \leftrightarrow \\ E_{BAM} &= R + B + M + V + M + (n-1) \times (R + B + M) \leftrightarrow \\ E_{BAM} &= n \times (R + B + M) + B + \varepsilon + M \leftrightarrow \\ E_{BAM} &= (n+1) \times (R + B + M) + \varepsilon - R \end{aligned}$$

Ecuția 4.5.

Dacă protocolul BAM nu ar fi pus în aplicare, fiecare nod ar măsura valoarea lor, fără raportare de stare a bateriei – prin urmare, utilizând energia E :

$$E = n \times (R + V + M)$$

Ecuția 4.6

Dacă vom înlocui conținutul din Ecuția 4.2 în ecuația precedentă, vom obține:

$$E = n \times (R + B + \varepsilon + M) \leftrightarrow$$

$$E = n \times (R + B + M) + n \times \varepsilon$$

Ecuția 4.7.

Pentru a determina când se justifică o utilizare a protocolului BAM, trebuie să îl comparăm cu protocolul normal.

Acest lucru poate fi realizat folosind Legea lui Amdahl [PaHe02].

$$\begin{aligned} \frac{E_{BAM}}{E} &= \frac{(n+1) \times (R + B + M) + \varepsilon - R}{n \times (R + B + M) + n \times \varepsilon} \leftrightarrow \\ \frac{E_{BAM}}{E} &= 1 + \frac{R + B + M - (n-1) \times \varepsilon - R}{n \times (R + B + M) + n \times \varepsilon} \leftrightarrow \\ \frac{E_{BAM}}{E} &= 1 + \frac{B + M - (n-1) \times \varepsilon}{n \times (R + B + M) + n \times \varepsilon} \end{aligned}$$

Ecuția 4.8.

Aceasta înseamnă că protocolul BAM este mai bun, din punct de vedere energetic decât protocolul clasic, atât timp cât:

$$\frac{B + M - (n-1) \times \varepsilon}{n \times (R + B + M) + n \times \varepsilon} > 0$$

Ecuția 4.9.

Numitorul din această fracție este întotdeauna pozitiv, pentru că este o sumă de energii, prin urmare, ecuația anterioară poate fi redusă la:

$$\begin{aligned} B + M - (n-1) \times \varepsilon &> 0 \leftrightarrow \\ B + M &> (n-1) \times \varepsilon \leftrightarrow n \times B + M > (n-1) \times V \end{aligned}$$

Ecuția 4.10

Valorile implicate de această ecuație sunt dependente de dispozitiv. În cazul în care dispozitivele din rețea respectă această condiție, atunci acest protocol poate rula mai bine din punct de vedere energetic decât protocoalele clasice.

Un alt avantaj al acestui protocol este că având mai multe noduri evaluate precum nodul C, se poate stoca statutul energetic de de alte noduri și pe acesta se poate baza un studiu pe rețea, se pot trimite rapoarte de activitate, etc

Un dezavantaj al acestui protocol este lățimea de bandă suplimentară cerută de comunicare. Dacă protocolul normal necesită două mesaje de la și la fiecare nod, protocolul BAM adaugă în total alte două mesaje (cel al valorii măsurate de la nodul „câștigător” și avizul de câștigare). Cu toate acestea, acest overhead de numai 2 mesaje poate fi considerat a fi acceptabil, având în vedere câștigul energetic general.

Un alt dezavantaj al protocolului BAM este că un nod rău-intenționat poate pretinde de a avea mereu energie maximă, astfel încât să poată trimite valori false în rețea.

De asemenea, în cazul în care nodul selectat este afectat de un eșec, după selecție, acest nodul central nu poate afla valoarea. O soluție este a se adăuga un alt timer pentru această parte a protocolului.

O soluție simplă pentru a îmbunătăți protocolul BAM este obținută prin înlocuirea pașilor de la 3 la 5 din protocolul anterior, cu pașii de la 3* la 5*, după cum urmează, rezultând astfel crearea unui protocol mai tolerant la erori intitulat Anti-hijacking Battery Aware Measurement (ABAM) Protocol.

Pasul 1: Un nod oarecare C vrea să afle o valoare care poate fi măsurată de n noduri, de aceea, C trimite o cerere pentru a afla de la cele n noduri statusul lor energetic

Pasul 2: Nodurile răspund (într-un interval de timp – T), cu valoarea respectivă. Nodurile care răspund după ce timerul intern pentru T s-a scurs, sunt ignorate. Ar trebui să existe cel puțin un răspuns pentru a putea continua la pasul următor. Dacă nu sunt răspunsuri primite, protocolul rapoartează eroare.

Pasul 3*: Primele trei noduri cu cea mai multă energie sunt rugate a măsura valoarea. În caz că există mai multe răspunsuri cu aceeași valoare pentru energia rămasă, cele trei noduri este alese aleator.

Pasul 4*: Nodurile selectate măsoară și trimit valoarea măsurată la C sau un mesaj de eșec, în cazul în care măsurarea nu s-a putut efectua.

Pasul 5*: C așteaptă același timp T – ca și pentru pentru mesajele de la punctul 2. Apoi C compară valorile și dacă acestea sunt toate la fel totul este în ordine, dacă două dintre ele sunt la fel, atunci această valoare e aleasă. Dacă toate cele trei valori sunt diferite, atunci C repornește protocol, dar acum va folosi un protocol clasic (de consens).

Repetarea raționamentului utilizat mai înainte ne va estima energia pentru protocolul ABAM:

$$E_{ABAM} = 3 \times E_m + (n-3) \times E_p \leftrightarrow$$

$$E_{ABAM} = (n+3) \times (R+B+M) + 3 \times (\varepsilon - R)$$

Ecuția 4.11.

Prin urmare, condiția pentru care protocolul ABAM este mai bun decât protocolul clasic este de:

$$\frac{E_{ABAM}}{E} = \frac{(n+3) \times (R+B+M) + 3 \times (\varepsilon - R)}{n \times (R+B+M) + n \times \varepsilon} \Leftrightarrow$$

$$\frac{E_{BAM}}{E} = 1 + \frac{3 \times (R+B+M) - (n-3) \times \varepsilon - 3 \times R}{n \times (R+B+M) + n \times \varepsilon} \Leftrightarrow$$

$$\frac{E_{BAM}}{E} = 1 + \frac{3 \times (B+M) - (n-3) \times \varepsilon}{n \times (R+B+M) + n \times \varepsilon} \Leftrightarrow$$

$$3 \times (B+M) - (n-3) \times \varepsilon > 0 \Leftrightarrow$$

$$3 \times (B+M) > (n-3) \times \varepsilon \Leftrightarrow 3 \times M + n \times B > (n-3) \times V$$

Ecuția 4.12

O problemă interesantă apare atunci când, de exemplu, eroarea de măsurare internă din fiecare nod oferă valori diferite pentru aceeași variabilă.

O soluție este să se ia în considerare valorile primite de la nodul C ca fiind egal, dacă acestea variază într-un interval limitat.

Rezultatul final, poate fi valoarea medie a valorilor cu erori de măsură. Acest lucru ar putea corecta chiar și parțial aceste erori de măsurare.

În scopul punerii în aplicare a acestui protocol, ne putem folosi de protocolul FIPA Contract-Net care este prezentat în Figura 4.23.

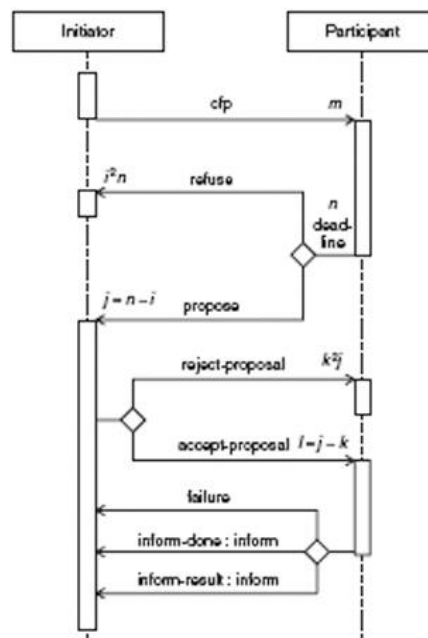


Figura 4.23. FIPA Contract Net [BeCG07]

Deși acest protocol este o directivă în JADE, putem vedea că doar anumite rute sunt folosite de BAM și ABAM de protocol, așa cum se vede în Figura 4.24.

Când răspund după directiva de CFP (apel pentru propuneri), fiecare dintre noduri trebuie să-și propună ca valoare statusul bateriei.

Partea din protocol care implică refuzul sau respingerea acestei valori nu este niciodată folosită.

Este de așteptat ca rezultatul să fie o directivă inform - result, cu valoarea cerută în solicitarea de propuneri (CFP).

Inform - done nu este, folosită niciodată pentru că scopul protocolului este de a găsi o valoare.

O opțiune interesant este incapacitatea de răspuns, care ar putea fi folosită în scopul de a semnal de un eșec.

În implementare, pentru motive de simplitate, această opțiune nu a fost folosită.

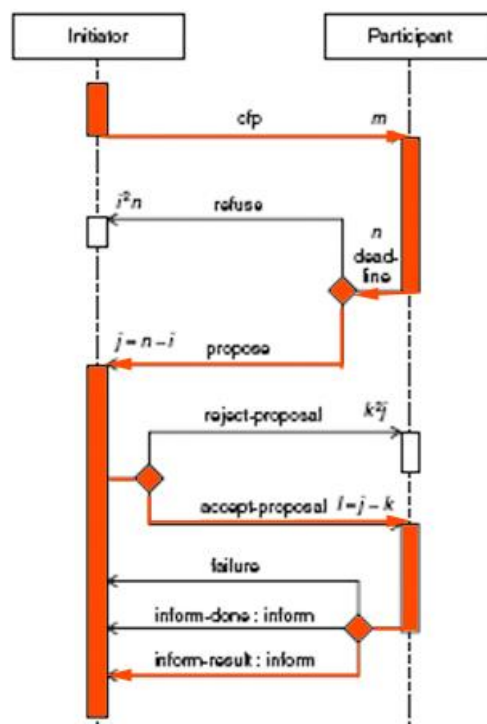


Figura 4.24. BAM aplicat pe protocolul ContractNet

Putem acum determina teoretic timpul de execuție a protocolului BAM.

Pentru a face asta, vom defini următoarele variabile:

$t_m(n)$ – timpul necesar pentru o comunicare de tip 1- n

$t_m(1)$ – timpul necesar pentru interschimbarea între două noduri a unui mesaj

t_B – timpul pentru a măsura starea bateriei

t_V – timpul pentru a măsura valoarea

$t_{max}(i)$ - timpul necesar pentru a alege câștigătorul (câștigătorii)

T – perioada maximă pentru care nodul C așteaptă un răspuns (evident $T > t_B$)

Astfel, tot timp de execuție pentru BAM este:

$$T = t_m(n) + T + t_{max}(1) + t_m(1) + t_v + t_m(1)$$

Ecuția 4.13.

Dacă se dorește a fi stocate valorile primite, de exemplu pentru un history log, atunci la timpul general trebuie să adăugați timpul de stocare.

După cum putem vedea, dacă durează mai mult timp decât timerul T intern comunicarea cu toate cele n noduri, unele mesaje se pot pierde.

Cu toate acestea, o ameliorare se poate realiza dacă în loc de așteptarea mereu până la expirarea timerului T , dacă sunt primite toate cele n mesaje de la cel n noduri, atunci protocol se mută la următorul pas. În acest caz, timpul total de execuție este:

$$T = t_m(n) + \min(t_b + t_m(n), T) + t_{max}(1) + t_m(1) + t_v + t_m(1)$$

Ecuția 4.14.

Pentru protocolul ABAM trebuie să fie adăugat timpul necesar pentru compararea celor 3 mesaje.

$$T = t_m(n) + T + t_{max}(3) + t_m(3) + \min(t_v + t_m(3), T) + t_{comparison}$$

Ecuția 4.15.

Pentru protocoale ABAM și BAM, numărul de noduri influențează timpul numai în prima fază a schimbului de mesaje, în a doua parte, numărul de noduri este constant (1 sau 3).

Protocoale au fost testate pentru a vedea global timp de execuție. Pentru a cuantiza diferența între timpul de prelucrare și cel al comunicării, ele au fost măsurate diferit.

Pentru a arăta utilitatea intervalului de timp – T , am injectat erori la un număr aleator (dar mai puțin de o treime din numărul total) din noduri și atunci când toate valorile sunt primite, contorul de timp T este oprit și se începe următorul pas. Acest lucru duce la faptul că, în unele tururi de scrutin, în care s-a primit toate mesajele, timpul a fost mai scurtă decât T .

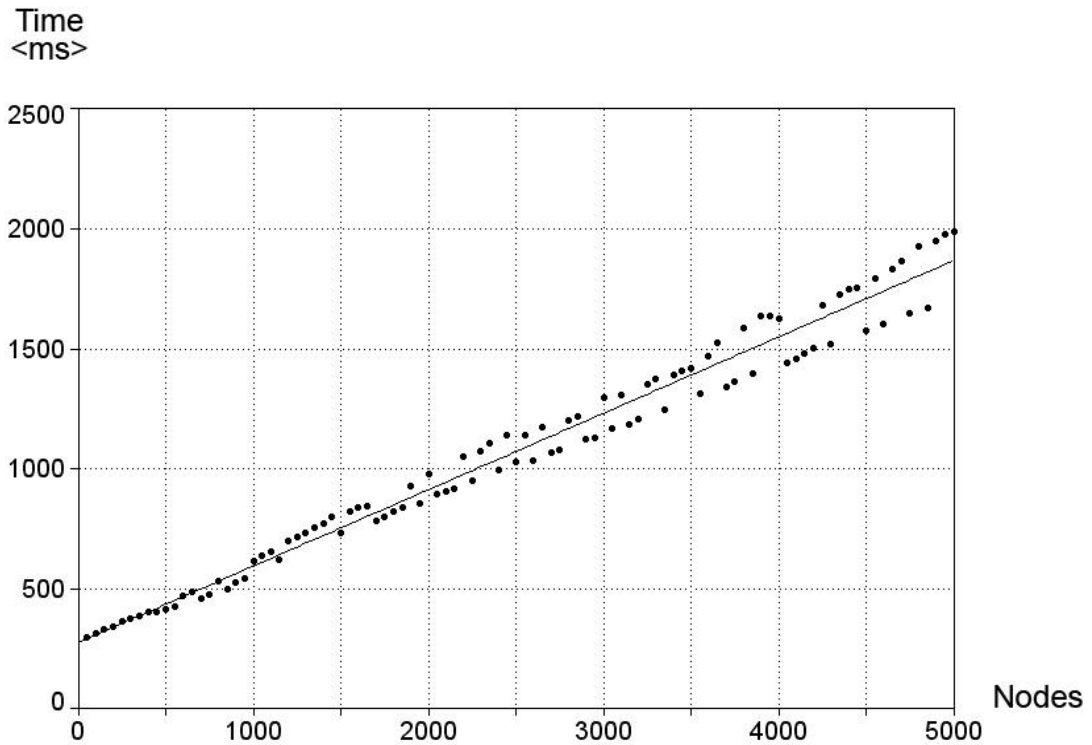


Figura 4.25. Timpul de comunicarea pentru protocolul BAM, erori prezente

Putem considera timpul necesar pentru a măsura bateria și valoarea sunt de interes neglijabil, datorită faptului că acestea sunt constante și dependente de dispozitiv.

Ele apar foarte clar, la un număr mic de noduri, în cazul în care overheadul de comunicație este încă nesemnificativ. Aceste rezultate au fost obținute cu TableCurve. Eroarea medie pătratică pentru funcția care se încadrează în este $r^2 = 0.97695$, iar funcția, așa cum este de așteptat, este liniară:

$$\begin{aligned} y &= a + bx; \\ a &= 271.31455; \\ b &= 0.31818 \end{aligned}$$

Ecuția 4.16. Funcția de timp pentru BAM

Dacă nu am considerat nici un mesaj eronat și pierdut, astfel, limita T nemaifiind necesară rezultatele sunt prezentate în Figura 4.26.

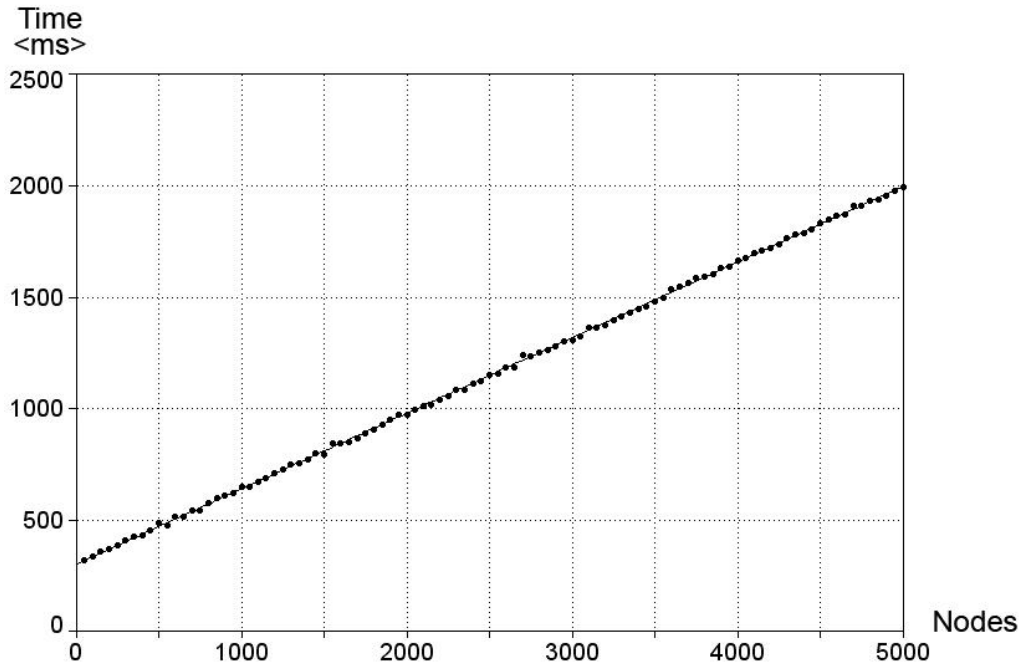


Figura 4.26. Timpul de comunicarea pentru protocolul BAM, erori absente

Protocolul ABAM nu arată o diferență notabilă față de protocolul BAM.

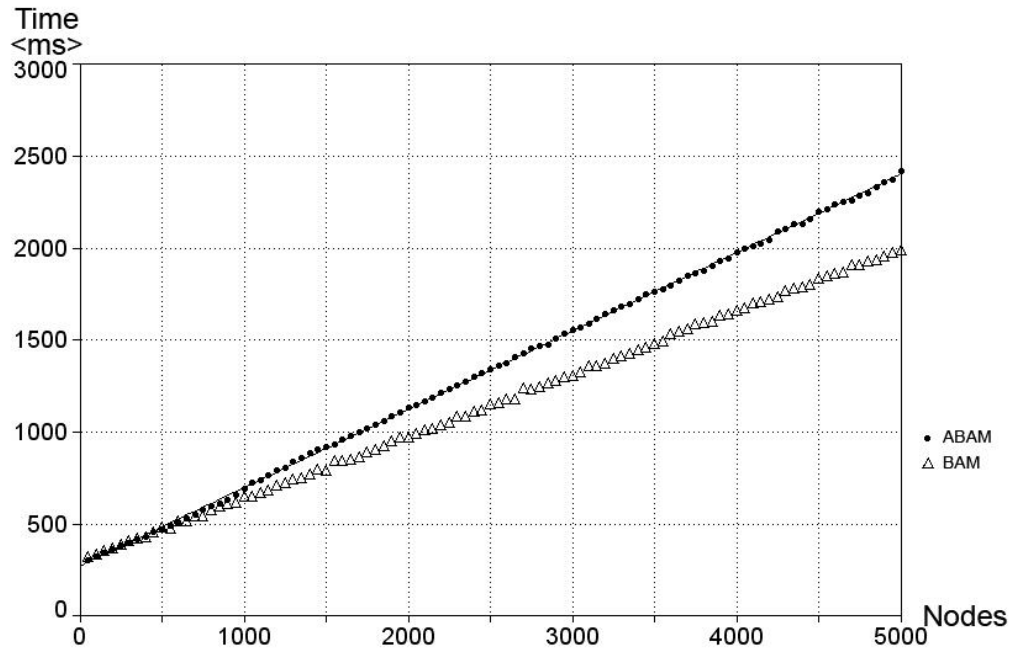


Figura 4.27. Timpul de comunicarea pentru BAM și ABAM, erori absente

Pentru un număr mai mic de noduri (mai puțin de 1000), diferența între BAM și ABAM aproape nu poate fi luată în considerare, după cum se arată într-un detaliu în Figura 4.28.

Utilizând facilitățile oferite de TableCurve, putem constata că funcția care se potrivește cu aceste puncte, $r^2 = 0.9997$, este tot una liniară, care are următoarea structură:

$$y = a + bx$$

$$a = 265.93818$$

$$b = 0.426$$

Ecuția 4.17. Funcția de timp pentru ABAM

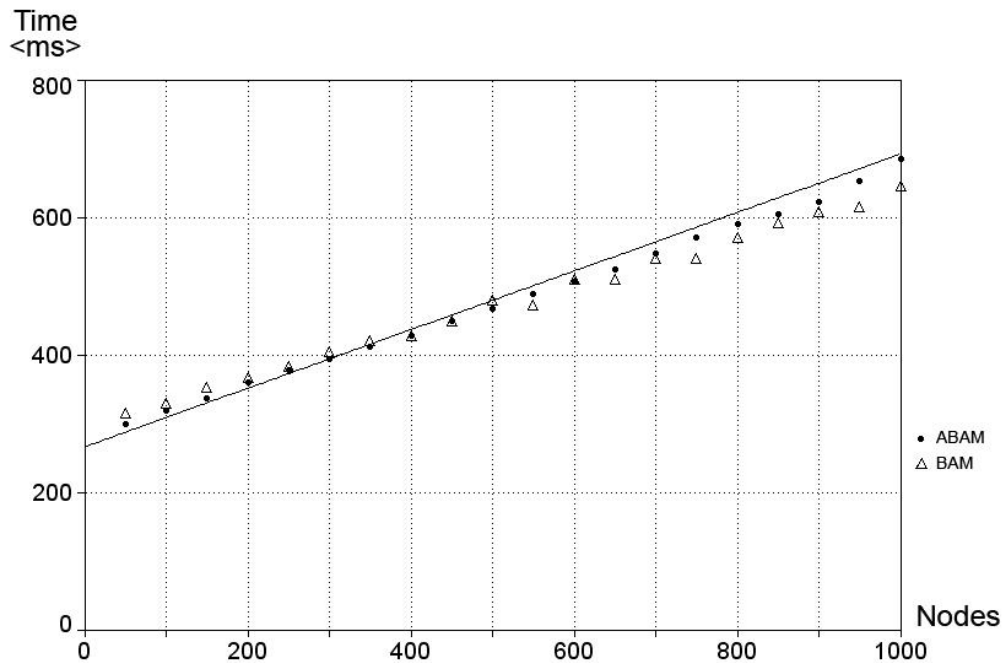


Figura 4.28. ABAM VS BAM - Zoom

În concluzie, cele două protocoale prezentate, de complexitate liniară sunt analizate. Complexitatea lor liniară, precum și ușurința de a le implementa ca parte din protocoalele standard FIPA face din aceste protocoale o alegere bună atunci când se caută să se mențină durata de viață a bateriei dintr-o vastă rețea de noduri, menținând în același timp un grad de toleranță la defectare, ceea ce apropie modelul IG de AmI.

4.5. Toleranța la defectare

În orice sistem real, noțiunea de dependabilitate este crucială pentru a permite folosirea sistemului pe scară largă. Dependabilitatea, sau abilitatea de a evita căderile care se află sub o anumită limită și/sau frecvență [ALRL04], poate fi abordată fie prin studierea atributelor sale (availability, reliability, etc); a tipurilor de amenințări sau prin studierea mijloacelor de a o atinge (vezi Figura 4.29).

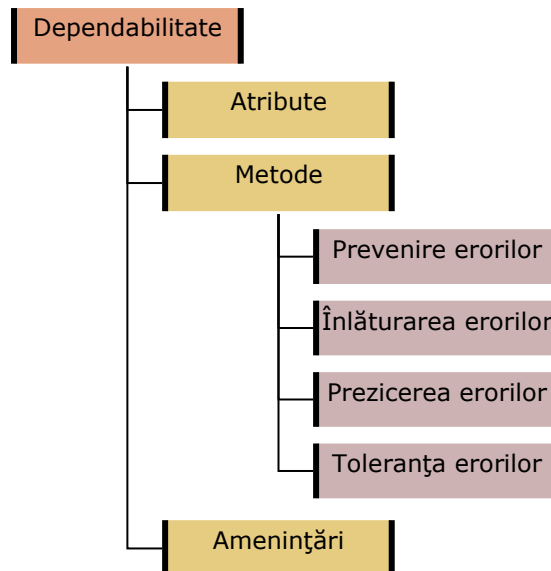


Figura 4.29. Moduri de a studia dependabilitatea - prelucrare după [ALRL04]

Prevenirea erorilor (fault prevention) implică prevenirea apariției sau introducerii erorilor și se realizează de obicei în faza de dezvoltare a sistemului/produsului.

Înlăturarea erorilor (fault removal) dorește reducerea numărului și a severității erorilor [ALRL04]. Poate fi realizată în timpul dezvoltării sistemului (prin verificare, urmată de diagnoză și corecție) sau în timpul utilizării sistemului (prin întreținere corectivă sau preventivă).

Prezicerea erorilor (fault forecasting) este estimarea numărului, incidenței și consecințelor erorilor. Această prezicere se realizează pe baza unei evaluări calitative și cantitative a comportamentului sistemului în ceea ce privește apariția și activarea erorilor.

Toleranța erorilor (fault tolerance) dorește menținerea funcțională a sistemului, în prezența erorilor. Pentru a obține tolerarea erorilor într-un sistem se folosesc două tehnici majore: detectarea erorilor și recuperarea sistemului dintr-o stare prezentând erori și aducerea lui într-o stare fără erori.

Fiabilitate [ALRL04] este definită ca „continuitatea serviciului corectă” și este considerată a fi o altă problemă majoră, datorită faptului că unreliability este inerentă în cadrul conceptului de „disappearing electronics” [WeRA05].

Acest lucru este cauzat primar de faptul că nodurile pot apărea în mod neașteptat, se pot muta, pot dispărea sau își pot termina rezervele de energie (temporar sau nu). Toate aceste probleme sunt exacerbate de constrângerile referitoare la cost, mărime și puterea dispozitivelor.

Fiabilitate poate fi atins prin intermediul a trei metode: redundanța, codurile detectoare și corectoare de erori și protocoale de consens. Fiecare dintre aceste trei metode acoperă o parte de fiabilitate (vezi Figura 4.30).

Redundanța presupune, în mod tradițional, existența a mai multe unități funcționale identice, funcționând în paralel.

Așa cum va fi arătat, redundanța poate apărea și la alte niveluri. Ideea din spatele redundanței este că dacă unele unități nu pot funcționa corect, încă mai există alte unități în locul celor nefuncționând corect, care pot sprijini funcționarea corectă globală a sistemului.

Codurile detectoare și corectoare de erori sunt o modalitate de a asigura corecta transportare de informații între unități – acest lucru înseamnă protejarea informațiilor împotriva unui număr de probleme posibile (de exemplu: atacuri, zgomot, etc) și posibilitatea de a reconstrucție a informațiilor dintr-o unitate afectată de aceste probleme.

Consensul indică corelarea de date diferite pentru a obține același rezultat final pe o fracțiune suficientă de participanți. Tradiționalii algoritmi de consens se asigură că datele sunt suficient de reprodus în întreaga rețea pentru a le permite participanților de a obține același rezultat final.

Rezumând, codurile detectoare și corectoare de erori sunt preocupate cu tranziția informațiilor în condiții de siguranță, redundanța este de ajutor cu asigurarea a destul de multe unități funcționale, pentru a permite sistemului să opereze sub un anumit număr de eșecuri; iar consensul este atent la o modul în care are loc schimbul de informații care se face între unitățile funcționale, pentru a obține rezultatul final.

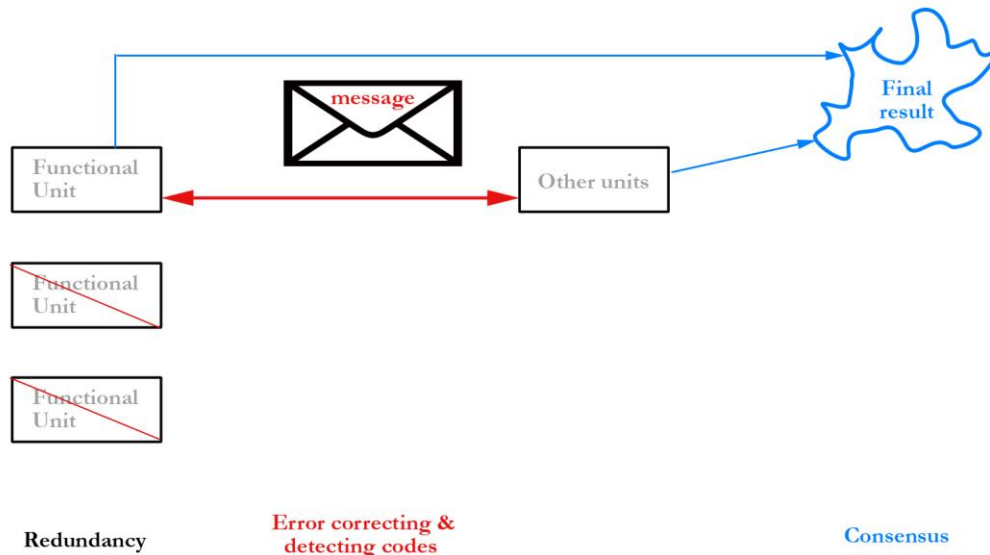


Figura 4.30. Niveluri de fiabilitate

Revenind la legea lui Bell, redundanța este folosită mai ales la nivel de dispozitiv, codurile detectoare și corectoare de erori sunt utilizate la nivel de conexiune și consensul este regăsit la nivelul de colecții (de exemplu, de mesaje).

4.5.1. Redundanța

Soluția utilizată în cadrul AmI pentru a obține fiabilitatea este redundanța [WeRA05, Raba05]; de aceea, ne așteptăm ca acesta să fie de asemenea, o soluție în acest tip de grd (IG).

Redundanța și aleatorizarea sunt considerate a fi mijloacele pentru a crea robustețe, deoarece:

- Elemente și legăturile dintre ele sunt supuse erorilor
- Mediul este predispus la schimbarea rapidă a condițiilor

În afara de aceasta, având în vedere energia necesită de regimul de așteptare și cel de funcționare, folosind redundanță, uneori am putea dezactiva elemente din cadrul rețelei.

4.5.2. Redundanță la nivelul zonelor de sensibilitate

În timp ce potențialul de aplicare a rețelelor conținând senzori este nelimitat, construcția unor astfel de rețele este deosebit de dificilă. Una din principalele provocări este de a menține timpul de viață al rețelei, precum și zone suficient de sensibile. Deși senzori au un cost redus de putere, în general, întreaga lor aprovizionare cu energie este pentru moment foarte dificil de realizat.[CGVC06]

Mai întâi de toate, o densitate mare de senzori irosește o gramadă de energie, dar în același timp, oferă oportunități de proiectare a protocoalelor eficiente energetic. O abordare larg folosită pentru reducerea consumului de energie în

rețelele ce includ senzori este aceea de a dezactiva senzorii redundanți prin planificarea funcționării alternative a acestora[CoNS05].

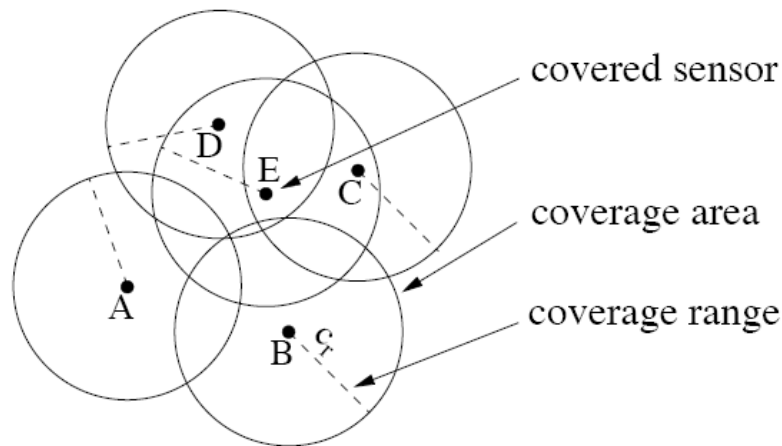


Figura 4.31. Redundanță la nivel de arie de sensibilitate [CoNS05]

Reducerea costurilor de energie, fără introducerea de complexitate în alte părți ale sistemului este încă o problemă grea. Chiar dacă dezactivarea senzorilor este o strategie frecvent utilizată de reducere a costurilor de energie, această acțiune ar putea genera puncte nemăsurate și, în consecință, ar reduce raza de acoperire a rețelei.

Pentru un anumit domeniu de implementare, punctele oarbe sunt considerate a fi de regiuni care nu pot fi monitorizate de către orice senzori [CoNS05].

O rețea ce include senzori oferă arie de acoperire maximă atunci când toți senzorii sunt alimentați și funcționali. Cu toate acestea, dacă toți senzorii funcționează tot timpul, este posibil a se irosi o mulțime de energie și, astfel, se va reduce durata de viață a nodurilor și implicit a rețelei.

Pentru că cele mai multe aplicații ar putea să nu necesită acoperire maximă, este important a oferi algoritmi [CGVC06] pentru dezactivarea senzorilor fără a degrada acoperire globală într-un mod semnificativ statistic.

4.5.3. Redundanță la nivel de legătură

După cum este descris în detaliu în literatura, o eroare apărută pe mediul de transport nu pot fi separată de eșecul componentelor care comunică prin acel mediu de transport [LaPS80]. Dar dacă aceeași componentă este legată redundant la mai multe componente putem depăși această problemă. Astfel, nu numai că putem determina unde este eroarea pe link sau pe componente, putând mări de asemenea, capacităților de tolerare la defectare.

Costul, implicat de cabluri suplimentare poate fi justificat în anumite aplicații, în care eroarea pe mediu de transmisie are o probabilitate mult mai mare decât probabilitatea de defectare a componentei în sine.

Acest aspect de redundanță a fost implementat pe IG, prezentată anterior.

Ipotezele în care redundanță la nivel de link-uri a fost implementată țin de cunoașterea poziționării și conectării controlerelor și senzorilor. Sensorii din aceeași zonă sunt conectați la un controler, zonele supravegheate de controlere se suprapun în anumite părți. Un exemplu este prezentat în figura următoare.

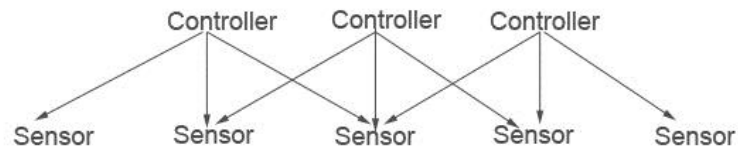


Figura 4.32. Redundanță la nivel de legătură[AnBV08b]

Acest aranjament lasă senzorii din zonele marginale să aibă un grad de acoperire mai scăzut decât cei din mijloc.

O posibilă soluție ar fi o arhitectură de tip ring care este posibilă numai în anumite situații. Oricum ne vom concentra pe partea de mijloc, în care este atinsă acoperirea/ redundanța maximă a senzorilor.

Așa cum am justificat anterior, am folosit JADE.

Pe fiecare controler un agent trimite solicitări, din timp în timp, la toți ceilalți agenți pentru a-și citi valorile de la senzorii la care sunt conectați. Acei agenți îi trimit acele valori și primul agent se va folosi de toate aceste valori pentru a face o acțiune.

Pentru aceste experimente, tipul de acțiune a fost simbolic: stabilirea valorii la o variabilă [LaPS80].

Fire de transmitere între senzori și controller-au fost au fost considerate a fi fail-stop, altfel spus, dacă se produce o eroare, linia nu trimite nici o valoare.

Erorile au fost injectate astfel încât să fie cât mai concentrate, altfel spus să afecteze cât mai multe linii de senzor posibil pentru un controler.

Numărul maxim de erori în care cel puțin o valoare de senzor este primit de către orice controler este de:

$$\text{Numărul maxim de erori} = \text{senzori complet acoperți} * (\text{acoperire}-1)$$

Ecuția 4.18. Numărul maxim de erori

Există trei variabile independente implicate de această arhitectură:

- Număr de controlere
- Numărul de erori
- Redundanța / acoperirea senzorilor

Metrica de performanță utilizată pentru a evalua aceste variabile a fost timpul de execuție.

Dezavantajul pentru aceasta tehnică ne apare atunci când se ia în considerare utilizarea de rețele fără fir. Nefiind fire, nu putem aplica acest nivel de redundanță.

Figura 4.33 prezintă modul de variație în timp folosind un grad de acoperire senzori variabil și menținând numărul de controlere fix.

Din acest grafic rezultă că numărul de erori injectate are prea puțin un efect global asupra timpului de execuție, diferența dintre 2 și 7 erori variind între 1 și 0,2 ms.

Funcția care descrie cel mai bine această variație este

$$y = (a + b \ln x / x^2)^{-1}$$

Ecuția 4.19

Putem afirma aceasta deoarece eroarea medie pătratică pentru această funcție a avut valoarea maximă de 0.9992 și valoarea minimă de 0.997 pentru un număr de erori variind de la 0 la 7.

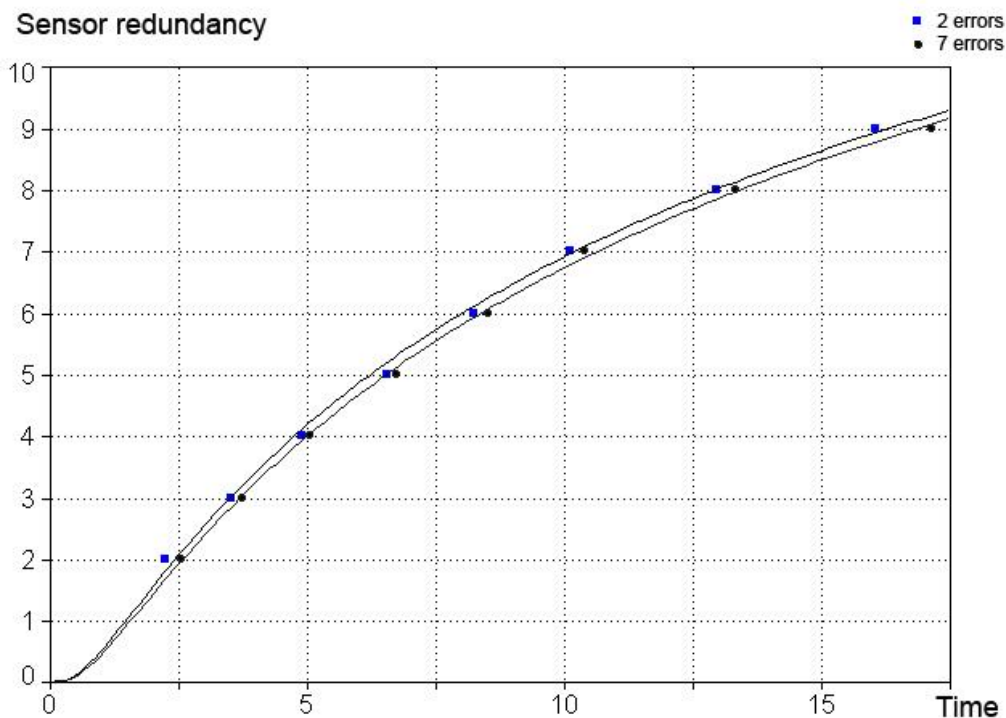


Figura 4.33 Grad de acoperire senzori variabil, numărul de controlere fix (9) [AnBV08b]

Figura 4.34 a fost concepută cu scopul de a afișa o parte din prima figura folosind de asemenea mai multe erori în trasarea acestui grafic.

Din nou se observă efectul aproape neglijabil al erorilor asupra timpului global de execuție.

Concluzia care se naște este că timpul global este influențat mai mult de alți factori (precum redundanța senzorilor) și nu de numărul de erori prezente. Aceasta implică că nu va apărea o degradare semnificativă de performanță în IG atunci când sunt multe erori față de momentele fără erori.

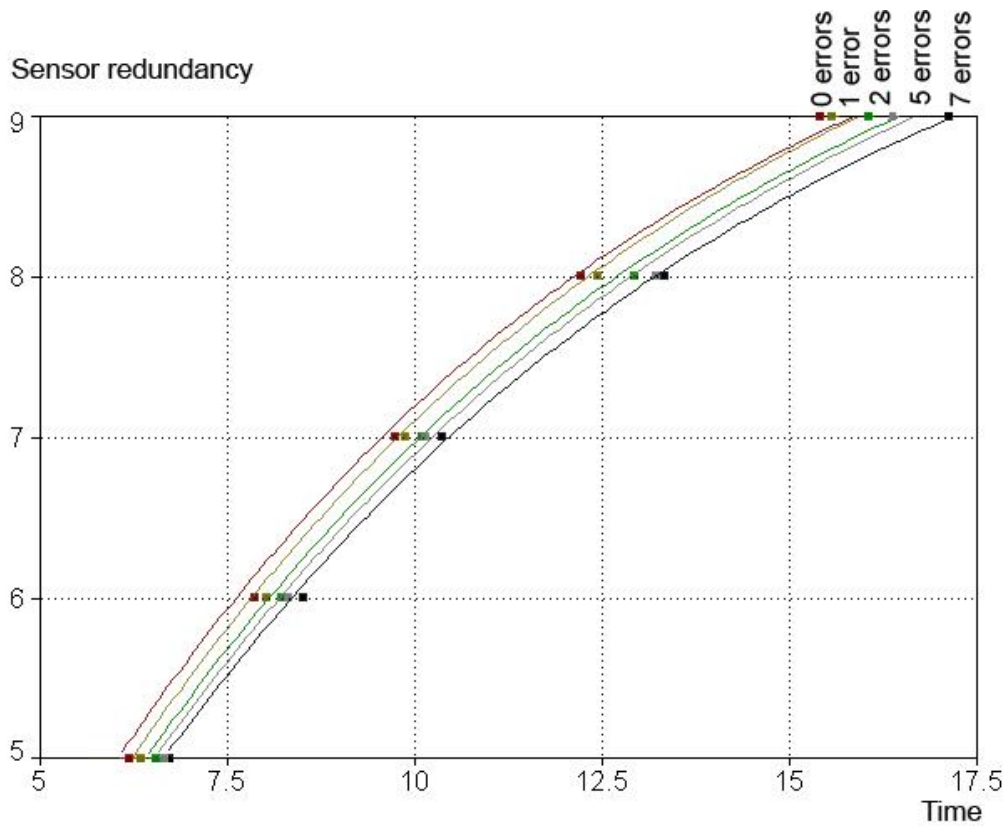


Figura 4.34. Grad de acoperire senzori variabil, numărul de controlere fix (9) - Zoom[AnBV08c]

În Figura 4.35 avem rezultatele obținute prin menținerea unui număr fix de erori (1), variind în timp numărul total de controlere.

Efectele extinderii redundanței senzorilor sunt evidente, timpul crescând proporțional cu acesta.

Astfel, pentru același număr de controlere diferența între două niveluri consecutive de redundanță este de aproximativ 1 ms.

Este de menționat aici că funcția care descrie cel mai bine toate aceste variante este:

$$y=a+bx^2$$

Ecuția 4.20

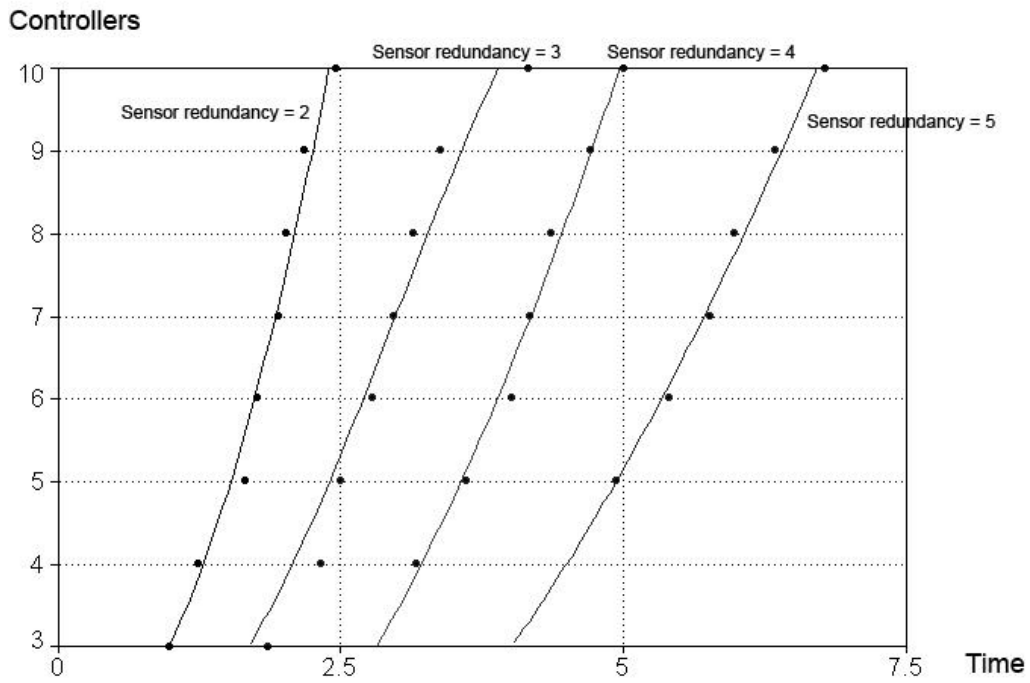


Figura 4.35. Numar de controlere de variabil, numărul de erori de fix (1) [AnBV08b]

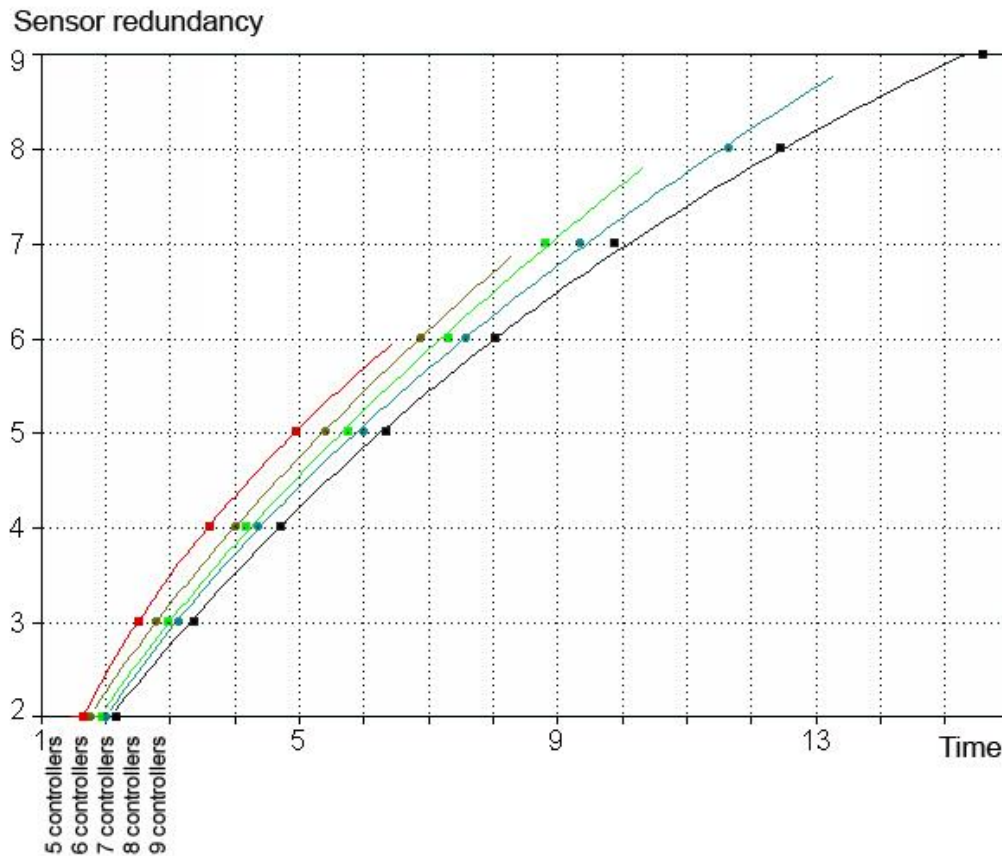


Figura 4.36. Numar de controlere de variabila, grad de acoperire fix (9) [AnBV08b]

Un rezultat interesant este prezentat în Figura 4.36 descriind funcția de timp, pentru un număr total de controlere variabil, gradul de acoperire fiind fix.

Primele trei funcții, reprezentând cazul în care sunt prezente 0,1 și 2 erori, au aceeași alură, având o tendință de a limitare superioară.

Însă ultima funcție, semnificând situația în care sunt 3 linii căzute, are un aspect diferit la această scară, diferența constând în limitarea scării de reprezentare a acesteia.

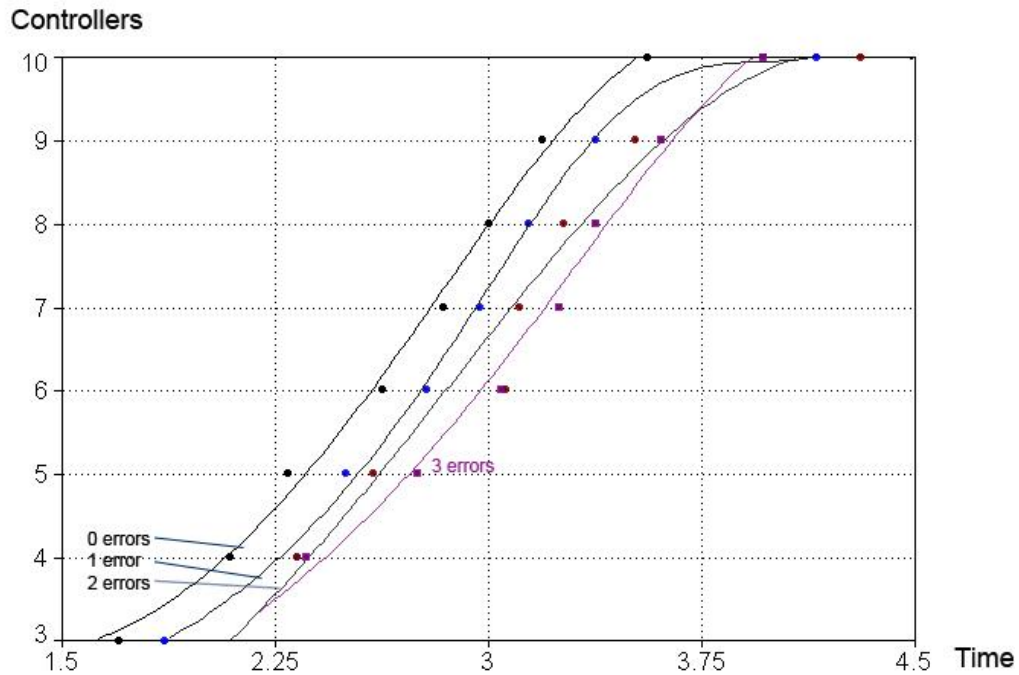


Figura 4.37. Grad de acoperire variabil, numărul de erori de fix (1)
[AnBV08c]

Ultima figură din acest paragraf (Figura 4.37) arată modul în care variază timpul, atunci când numărul total de controlare este un parametru, pe axa Oy fiind gradul de acoperire.

Funcția care descrie cel mai bine această variație este funcția de putere (r^2 măsurat variind între 0.993 și 0.9999).

$$y = a + bx^c$$

Ecuția 4.21

Unele din grafice sunt incomplete, în partea superioară de acoperire, pentru că nu poate fi obținută acea valoare a gradului de acoperire pentru acel număr particular de controlare, ceea ce este exprimat în formula pentru numărul maxim de erori (Ecuția 4.18) în care cel puțin o valoare de senzor este primit de către orice controler.

4.5.4. Redundanță la nivel de mesaj

Din anul 1985, Schneider [Schn85] a sugerat utilizarea consensului ca o paradigmă centrală pentru programare distribuită fiabilă. Relativ recent, în [GuSc01] s-a realizat o implementare a acestei teorii, prin crearea unui mod sistematic și

generic de a transforma diferite probleme de acord în consens, rezultând astfel crearea unui cadru unificat de a dezvolta protocoale tolerante la erori de acord într-un mod corect și eficient.

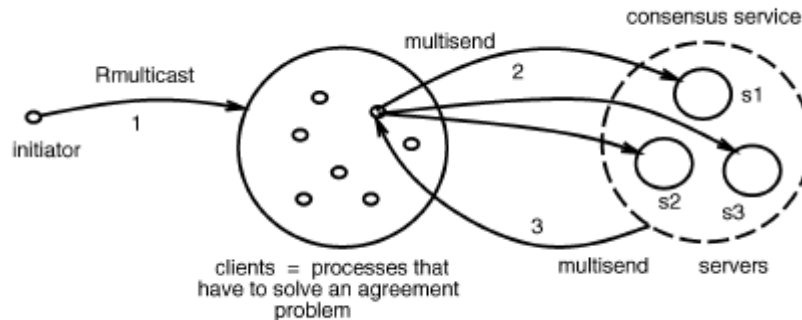


Figura 4.38. Interacțiunea de la un client din punct de vedere. [GuSc01]

În general aceste servicii sunt utilizate pentru a construi aplicații distribuite sau de a pune în aplicare servicii distribuite de nivel mai ridicat precum: servere de fișiere, servere de timp, serverele de nume, servere de autentificare, etc. Cu toate acestea, au fost foarte puține propuneri de servicii dedicate special construcției de protocoale tolerante la erori, cum ar fi protocoale de acord atomic, etc. De obicei, aceste protocoale sunt luate în considerare separat și nu se bazează pe o infrastructură comună.

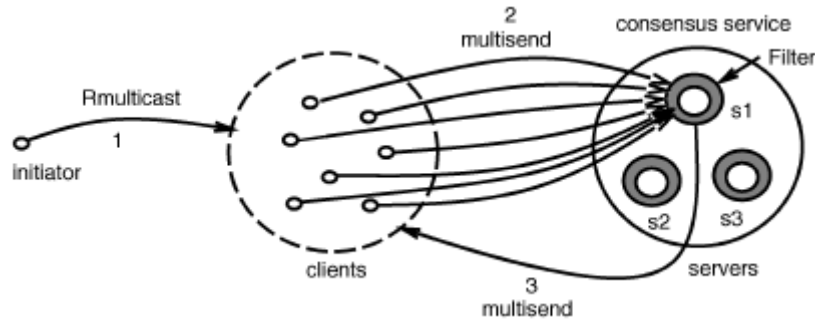


Figura 4.39. Invocare-răspuns din punct de vedere al server-S1 [GuSc01]

O excepție notabilă este un grup de servicii [GuSc01], care a fost folosit pentru a pune în aplicare diferite protocoale de difuzare. Un grup de probleme (rezolvate de către serviciul de membru) este doar un exemplu de la un acord cu o problemă care apare în sisteme distribuite. De fapt, toate problemele de acord (angajament atomic, în total, pentru difuzare, de grup de membru) sunt legate de problema abstractă a consensului și, de aceea, sunt supuse, în sistemele asincrone la rezultatul de imposibilitate Fischer-Lynch-Paterson [GuSc01].

Protocoale de acord fault-tolerant sunt construite folosind o interacțiune client-server, reprezentând cazul în care clienții sunt procese care trebuie să rezolve o problemă de acord și servere realizează punerea în aplicare a serviciilor de consens.

Acest serviciu este accesat printr-un filtru generic de consens, personalizat pentru fiecare problemă specifică de acord. Printr-o reducere sistematică la consens, diferite protocoale de acord tolerant la erori, cum ar fi angajament nonblocking atomic, total order multicast se pot rezolva [GuSc01].

În plus față de modularitate, această abordare permite implementări eficiente de protocoale de acord precise în ceea ce privește caracterizarea și ipotezele care stau la baza lor proprietăților de liveness și siguranță.

Toată ideea din această abordare constă în redundanță de mesaje. Prin utilizarea acestui tip de redundanță și a redundanței la routare se măresc șansele de a beneficia de valorile corecte, chiar și în posibilitatea de a atacatori multipli. În scopul de a determina mesajul de răspuns corect, este folosit protocolul de consens.

Rezumând, redundanța zonelor de sensibilitate presupune cunoașterea dispozitivului, redundanța link-ului presupune cunoașterea conexiunilor iar acest tip de redundanță a mesajelor – protocol pe bază de consens – implică colecția (de mesaje). Se aduce astfel, Legea lui Bell [Bell08] încă o dată în atenția noastră

Transferând conceptele prezentate [GuSc01] pentru IG, această problemă poate fi definită pe un set de mesaje schimbate de către agenți dintr-un sistem distribuit.

Fiecare agent începe cu o valoare inițială și agenții vor trebui să se decidă, pe o anumită valoare comună.

Un exemplu este următorul scenariu: inițiatorul lansează o cerere pentru un alt agent. Cererea este trimisă la toți agenții din lista inițiatorului, agenți care-l vor ajuta să trimită acest mesaj la agentul la care trebuia să ajungă.

De la inițiator va ajunge un set de mesaje (unele dintre ele pot fi distruse/eronate). Receptorul folosește un tip de algoritm BGP pentru a afla care este mesajul corect. După aceasta, receptorul se va conforma cererii inițiatorului (de exemplu printr-o prelucrare a datelor). Apoi, receptorul trimite răspunsul său la toți agenții pe la care a trecut spre el mesajul de la inițiator (aceasta poate include inițiatorul propriu-zis).

Inițiatorul ajunge la un set de răspunsuri de la receptor și de după utilizarea BGP pentru a afla care este răspunsul corect inițiatorul poate utiliza răspunsul corect. Prin urmare, în termeni de formulei de consens acest algoritm poate fi descris ca:

$$\psi : V^n \times [0, t_{final}] \rightarrow V^n$$

$$t_{final} \neq \infty$$

$$\psi \left(\begin{bmatrix} x_0(0) \\ x_1(0) \\ \vdots \\ x_{n-1}(0) \\ t \end{bmatrix} \right) = \begin{bmatrix} y_0 \\ y_1 \\ \vdots \\ y_{n-1} \end{bmatrix}$$

$$\exists f(x_i(0)) = y; i = \overline{0, n-1}$$

$$\max(v(y, Y)) = \Theta$$

$$\Theta \geq M$$

Ecuția 4.22 [Ancu08b]

Singura diferență de la modul în care este descrisă BGP este prezența funcției f , funcție care descrie:

1. rularea BGP cu privire la valorile inițiale în scopul de a determina corect valoarea inițială
2. prelucrarea efectuată la receptor peste valoarea inițială
3. rularea BGP pe setul final de valori în scopul de a determina corect valoarea finală.

Despre prelucrarea locală nu putem face ipoteze, deoarece nu știm în ce va consta.

În ceea ce privește algoritmul BGP, problema a fost intens studiată. Există mai multe tipuri de BGP (tradițional, limitat ca și ipoteză, ne-tradițional) și fiecare din aceste tipuri are două moduri de punere în aplicare a mesajului de transfer: mesaje orale și scrise mesaje. Ultima presupune existența unei semnături digitale pe fiecare mesaj.

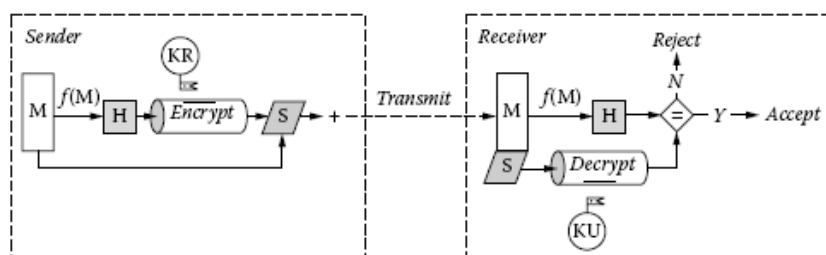
Având în vedere că metrica folosită pentru evaluare este timpul, apariția semnăturilor de mesaje presupune adăugarea la timpul cerut de mesajele orale a timpului necesar pentru codificare / decodificare a mesajelor

4.6. Consensul de tranzacții în IG

Vom aplica ideea prezentată anterior implementând astfel un protocol bazat pe consens pentru orice tip de tranzacții în cadrul IG.

Cum acest protocol este bazat pe BGP, avem două opțiuni de schimb al mesajelor: scrise sau orale.

Mesajele scrise implică, așa cum a mai fost precizat, existența unei



semnături digitale, complexitatea acestora influențând în mod clar timpul de rulare al algoritmului.

Drept urmare, în studiul acesta, vom trata partea de autentificare mai întâi, pentru a vedea posibila sa influență asupra timpului de rulare, continuând apoi cu algoritmul propriu-zis.

4.6.1. Parte de autentificarea a algoritmului de consens

Să studiem efectul semnăturilor digitale [Xiao07] pe baza de criptare asimetrică (criptare cu chei publice) care pot fi folosite pentru a semna digital un mesaj, M , și în acest fel va autentifica expeditorului.

Figura 4.40 Modul de calcul al semnăturii digitale [Xiao07]

Pentru a putea semna digital un semnal M , acesta trece printr-o funcție unidirecțională de hash $F(M)$ care produce valoare de hash H .

În termeni de paradigma proprietar - consumator, o posibilă cale de atac ar fi de a juca rolul unui consumator. Acesta este cazul unui atac Sybil.

O soluție la această problemă este găsită prin adoptarea unui broker.

În termenii modelului propus, un broker este un agent special cu rol de pază al traficului de date pe rețea.

Broker-ul este găsit printr-un proces de negociere între proprietar și consumator pe o lista de brokeri obținute de la directorul de servicii. Procesul de negociere este de fapt tot un proces de consens.

O semnătură digitală este creată de broker, următorii pași fiind efectuați:

- Pasul 1: Broker-ul creează o cheie de criptare asimetrică, pereche care e adusă la cunoștința consumatorilor și proprietarului.
- Pasul 2: Broker-ul efectuează o scanare a agentului mobil, pregătit pentru a fi relocat (cod, date și, dacă este necesar,

variabilele de stare). Dacă scanarea nu are succes tranzacția este anulată.

Pasul 3: Broker-ul creează codul de agent serializat pentru a fi relocalat

Pasul 4: Broker-ul generează un cod de hash H din codul serializat.

Pasul 5: Broker-ul produce o semnătură digitală S de criptare cu hash-ul H.

Odată ce acest proces este încheiat, broker-ul trimite înapoi la consumator codul serializat și semnătura, deoarece consumatorul trebuie să le transmită la agenți, și nu broker-ul.

Proprietarul apoi efectuează următorii pași:

Pasul 1: Decriptează semnătura S cu cheia publică obținând astfel codul de recuperare al hash H.

Pasul 2: Proprietarul creează un nou cod de hash H1. Dacă $H = H1$ atunci nodul poate accepta agentul de lansat cu încredere că aceasta a fost inspectată de către broker. Altfel, procedura este terminată.

Acest tip de algoritm detectează aproape toate atacurile care încercă să distrugă integritatea datelor, dar nu poate face nimic împotriva atacurilor [BoAV09] care încercă să blocheze sau să limiteze de lățime de bandă de rețea.

Comparativ cu lansarea simplă de un agent de la un agent la un alt recipient, se adaugă numai trei mesaje și timpul de calcul necesar pentru două cereri de hash (unul de la broker, unul de la consumator).

O problemă importantă este cea de credibilitate a broker-ului. În cazul în care un atacator este broker-ul, întreaga rețea este compromisă.

În implementarea noastră, am folosit o soluție simplă: broker-ul este PC-ul din rețea și există un singur broker în întreaga rețea.

Așa cum am menționat anterior, platforma pe care a fost testat de algoritm rula JADE și era deja implementată redundanța la nivel de legături.

Codificarea hash a fost realizat în două tehnici:

- un algoritm inactiv- care doar returnează valoarea și este util pentru a măsura experimental complexitatea algoritmului propus exclusiv de complexitatea algoritmului de hash.

- un algoritm de complexitate de tip liniar, cu scopul de a evalua impactul de algoritmi de hash în ansamblu pentru timp de rulare.

Nu se presupune pierderea mesajelor.

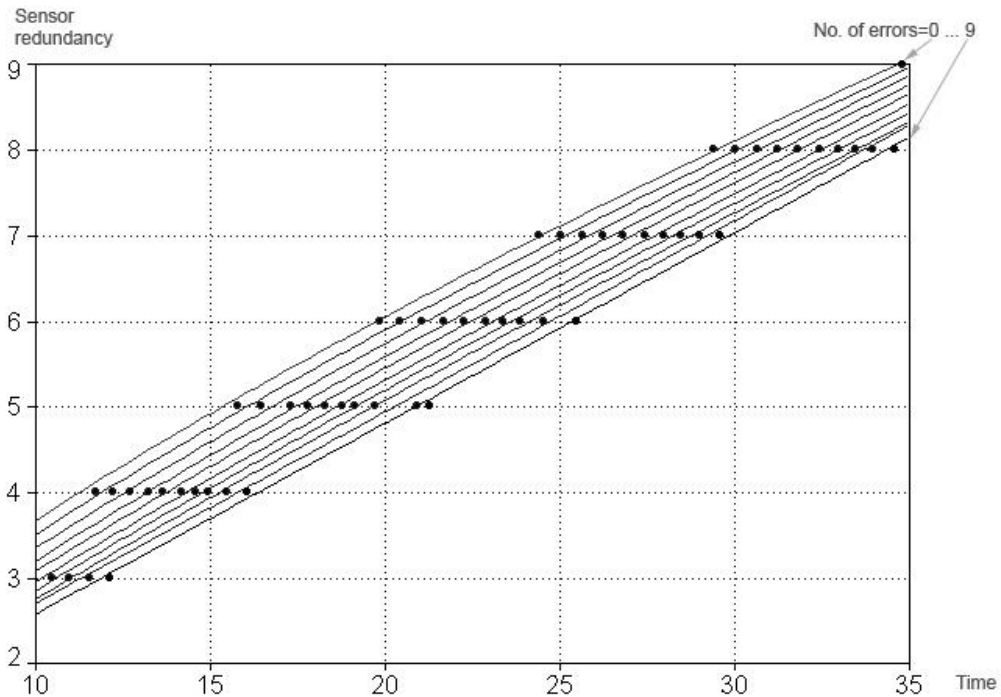


Figura 4.41 Număr fix de controlare (9), numărul de erori de variabil [BoAV08b]

În Figura 4.41 vom vedea variația de timp cu privire la redundanța senzorilor. Funcția obținută de către această variație este

$$\ln y = a + b \ln x$$

Ecuția 4.23

În Figura 4.42 este o comparație între timpul de rulare a unui hash-dummy și timpul de rulare a unui hash de complexitate liniară, în cazul în care există un număr fixat de erori.

Tipul general de funcție care descrie această variantă nu se schimbă.

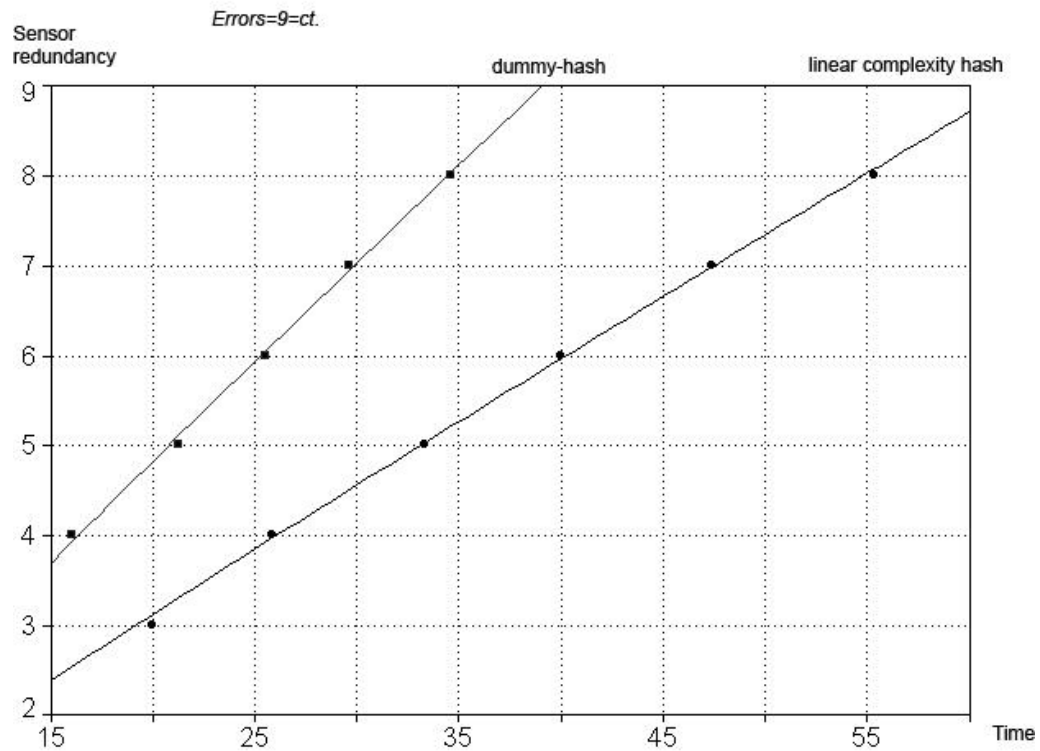


Figura 4.42. Număr fix de controlere (9), numărul de erori = 9 [BoAV08b]

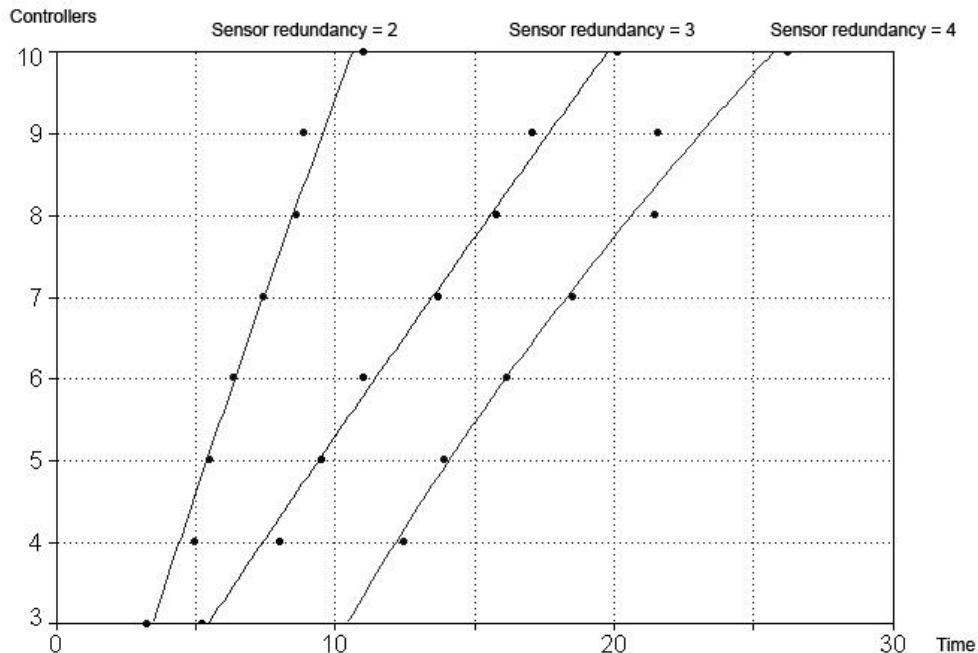
Errors=1=ct

Figura 4.43. Număr fix de erori (1) [BoAV08b]

În ultima figură vom vedea variația de timp, atunci când redundanța senzorilor și numărul de controler variază și avem injectată o singură eroare. Funcția care descrie cel mai bine acest lucru este în funcție de putere.

Concluzia care se desprinde din aceste experimente este că, în esență, algoritmul de semnare al mesajelor este de complexitate liniară la care se adaugă complexitatea algoritmului de hash și redundanța senzorilor. Numărul de erori nu influențează semnificativ timpul de rulare.

4.6.2. Rularea algoritmului de consens

În [GuSc01] consensul este considerat ca o cutie neagră și detectoare de erori sunt utilizate în afara acestei casete de consens.

Prin urmare, pe de arhitectura descris anterior, am pus în aplicare protocolul de consens între agenți.

Fiecare agent a avut o „memorie” – buffer a mesajelor schimbate anterior și a trebuit să își aleagă în final valoarea sa internă.

Fiecare agent ar putea eșua.

Algoritmul a rulat în runde, pentru a putea determina cu mai multă acuratețe efectele bufferului de mesaje.

Am implementat și conceptul de „uitare” prin limitarea bufferului de memorie.

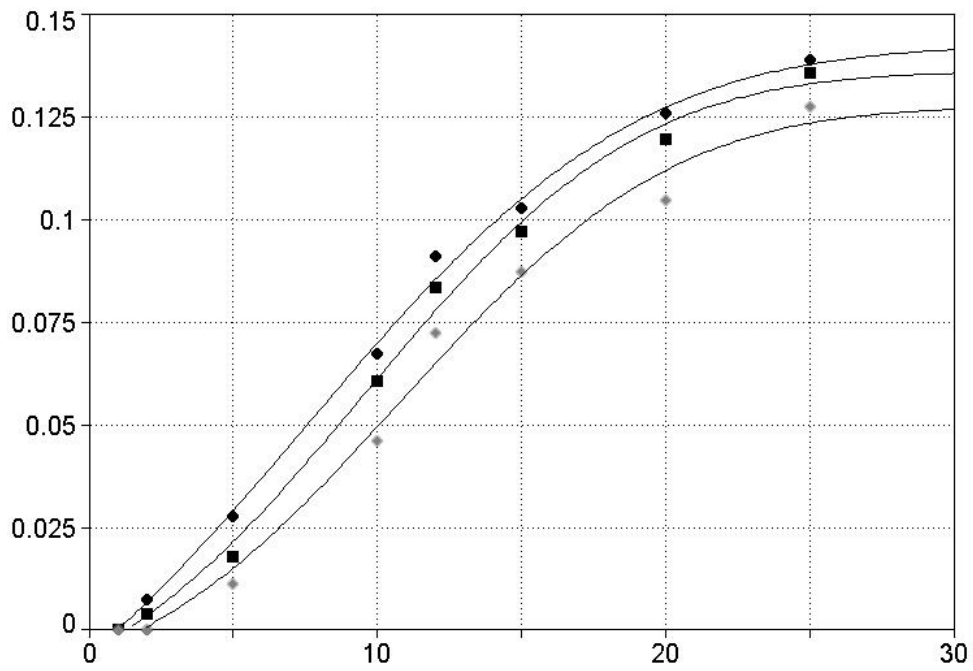


Figura 4.44. . Rata de eroare maximă, medie și minimă VS numărul de agenți, 20 runde, adâncime de memorie = 5 [Ancu08a]

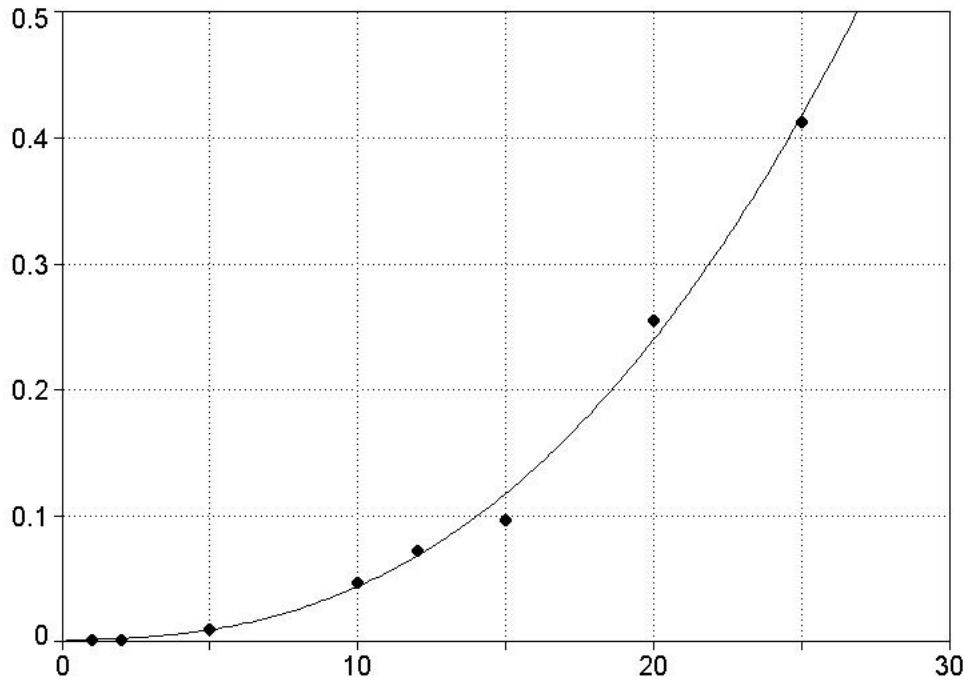


Figura 4.45. Timpul exprimat ms în VS numărul de agenți, 20 runde, adâncime de memorie = 5 [BoAV08a]

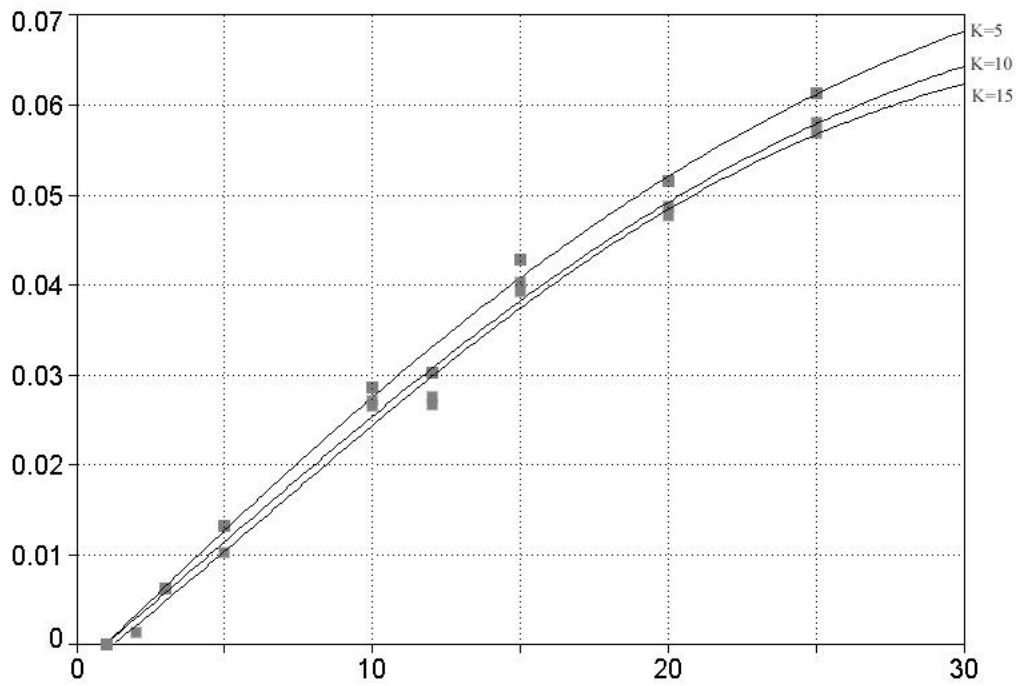


Figura 4.46. Rata maximă de eroare VS nr. de agenți pentru 50 de runde [Ancu08]

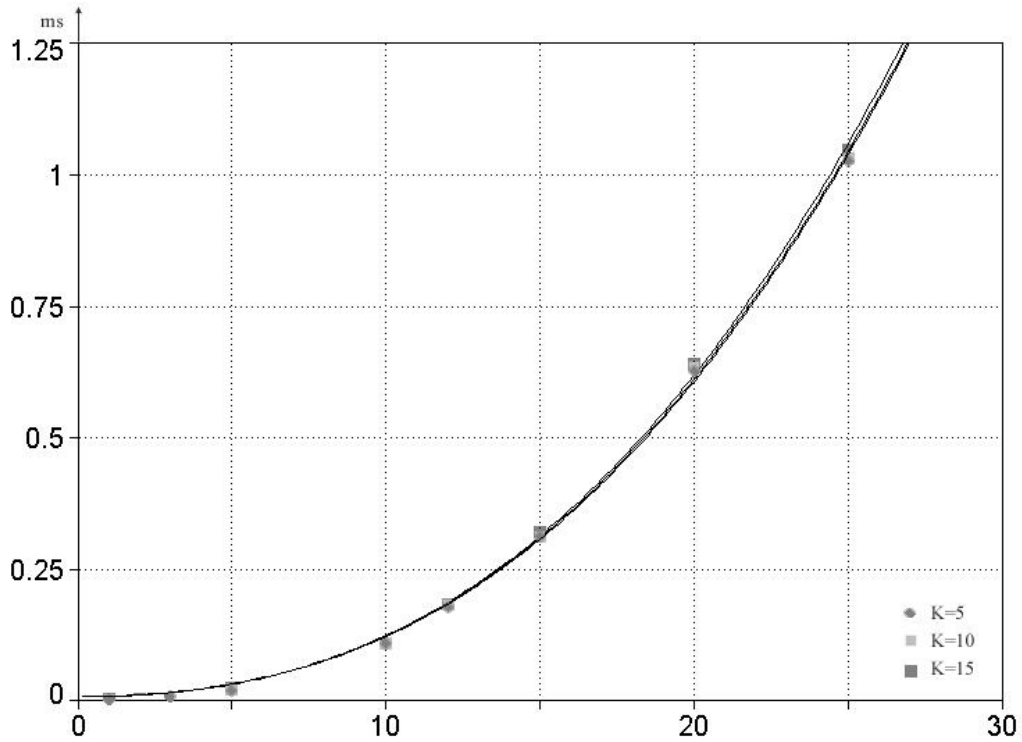


Figura 4.47. Timp de variație VS nr. de agenți pentru 50 de runde [BoAV08a]

Așa cum era de așteptat, timpul este exponențial, lucru obișnuit pentru toate protocoalele de consens.

$$y^{0.5} = a + bx \ln x$$

Ecuția 4.24

Dimensiunea bufferului nu influențează aproape de loc timpul de ansamblu, dar acesta influențează semnificativ rata de eroare. Cu toate acestea, rata de eroare nu s-a îmbunătățit proporțional cu creșterea în mărime a bufferului.

4.7. Discuție

Problema generalilor bizantini este extrem de amplă, cu multiple moduri de abordare. Un singur lucru rămâne comun tuturor acestor variante: necesitatea particularizării lor pentru cazul practic în care se aplică.

Variantele de program realizate pun în evidență invers proporționalitatea dintre generalitate, siguranță în exploatare pentru un număr cât mai mare de cazuri și viteza de execuție, respectiv folosirea resurselor.

S-a propus o soluție originală care are potențial și poate fi explorată din multe aspecte.

Un punct interesant este că, după lansarea celui problemei generalilor bizantini (în 1980) a urmat o serie de protocoale (până în 1986), apoi o pauză până la sfârșitul anilor 90, când problema s-a relansat, declanșată de dezvoltarea de agenți inteligenți, ceea ce face soluția noastră extrem de interesantă și actuală.

5. O aplicație a consensului în medicină

5.1. Motivație

În prezent, procesul de îmbătrânire a populației în Europa, din cauza expansiunii duratei de viață, crează modificări notabile în structura populației, de exemplu: în 2020, 1 persoană din 5 va avea peste 60 ani și 1 persoană din 14 va avea peste 70 ani. De aceea, tulburările care afectează în special persoanele în vârstă sunt extrem de interesante. Cele mai de temut probleme ale senescentei [Merc00] presupun spitalizare și / sau imobilizare la pat fiind, de asemenea, probleme care presupun o escaladarea în cheltuielile de sănătate.

În acest studiu, ne-am axat pe osteoartroză și pe osteoporoză, două probleme foarte frecvente în rândul persoanelor în vârstă. Acest lucru poate duce la tulburări de circulație și fracturi, având ca și rezultat, spitalizare și / sau imobilizare la pat. Orice ajutor pentru a determina persoanele cu risc ridicat, rezultă într-o posibilitate de a evita sau cel puțin de a aștepta problemele și spitalizarea în viitor.

Scopul nostru a fost de a determina, cu ajutorul unor date simple și a unor calcule la fel de simple, un risc de măsură a acestor două boli realizând astfel un screening al populației. Evident, că există deja proceduri de screening dar acest studiu prezintă o nouă metodă de screening, creată prin personalizarea unor factori de risc pentru o anumită regiune din țară.

5.2. O descriere a problemei medicale

Tulburările articulațiilor și ale țesuturilor adiacente, sunt clasificate în trei grupe: tulburări inflamatorii (poliartrita reumatoidă), tulburări degenerative (osteoartroză) și osteoporoză.

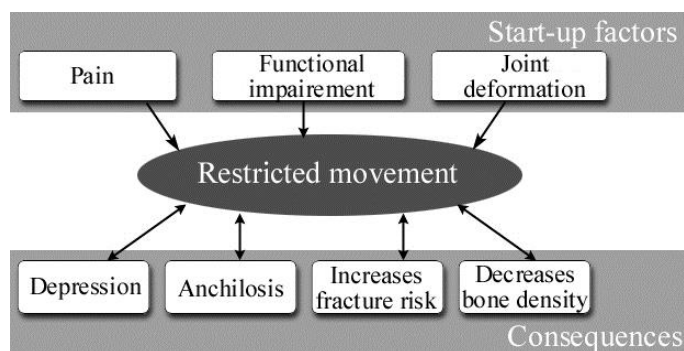


Figura 5.1. Evoluția osteoartrozei [SPAR06]

Poliartrita reumatoidă (PR) este o boală cronică multisistem de cauza necunoscută, prezentă la 1% în populația generală. Printre consecințele poliartritei reumotoide sunt modificările de integritate, frecvente din cauza deteriorării cartilajului și a oaselor precum și, de asemenea, atrofie musculară și stare generală de deteriorare.

Tulburările degenerative, cunoscute ca osteoartroză, sunt: spondiloza, artroza șoldului și gonartroza. 80% din aceste tulburări se găsesc la persoanele în vârstă.

Deprecierea posibilităților de mișcare este cea mai importantă consecință a tulburărilor degenerative, bazată pe un cerc vicios, după cum este descris în Figura 5.1. Evoluția osteoartrozei.

Osteoporoza este definită ca fiind reducerea masei osoase (sau a densității masei osoase), sau prezența unei fracturi de fragilitate [KBFH04].

Pierderea țesutului osos are ca și rezultate deteriorarea arhitecturii scheletului osos care crește riscul de fractură. În țările europene există peste 40 de milioane de cazuri de osteoporoză care reprezintă 10 - 12% din totalul populației.

5.2.1. Determinarea riscului de osteoporoză și osteoartroză

Etiologia osteoporozei și osteoartrozei conține două clase de factori: factorii care generează anomalii biomecanice (activități anormale, anomalii statice, activități normale ale țesuturilor anormale) și factori de risc (genetici, legați de stilul de viață, tulburări endocrine, medicamente, sindrom de malabsorbție (Ca), imobilizare și climatici).

Pentru estimarea densității osoase există mai multe tehnici de noninvasive: dual-energy x-ray absorptiometry (DXA), single-energy x-ray absorptiometry (SXA), quantitative computed tomography (CT), imagistică prin rezonanță magnetică (MRI) și ultrasunete. O scurtă comparație între ele este prezentat mai jos.

Technique	Accuracy	Radiation	Usage	Observations
SXA	2	2	1	1D, important variations
DXA	3	3	5	standard, 2D, Z-score, T-score
CT	4	4	3	expensive, less reproducible
MRI	5	5	1	very expensive, shows 3 mm lesions
ultrasound	1	1	4	low-cost, screening method, mobility,

Tabelul 5.1. Tehnici noninvasive de măsurare a masei osoase (5-cea mai mare, 1-cea mai mică) [ABVS08]

Valoarea BMD obținută din DXA variază cu înălțimea, producătorul de instrumente de măsurare sau cu locația măsurătorii. În consecință, a devenit o practică standard de a se raporta rezultatele la valori "normale", folosind un T-scor, care compara rezultatele cu cele individuale, într-o populație tânără din aceeași grupă de sex și rasă cu cea a pacientului.

Scorul Z compară rezultatele individuale cu cele ale unei populații de aceeași vârstă care este, de asemenea, asemănătoare ca și sex și rasă. [KBFH04]

Astfel, femeie de 60 de ani cu un scor Z de -1 (deviație standard cu 1 mai jos decât valoarea medie pentru grupa de vârstă) are un scor T de -2,5 (deviație standard cu 2,5 mai jos decât valoarea medie pentru grupa de tineri de control)

Așa cum este definită de către Organizația Mondială a Sănătății, orice valoare de scor T mai mare de -1.2 reprezintă valori normale ale densității minerale osoase, în cazul în care scor T este în intervalul [-1.2, -2.5), pacientul are osteopenie, și în cazul în care scorul T este mai mic de -2.5 avem un caz de osteoporoză.

Scorul T reprezintă o măsură suficient de bună a densității osoase, datorita largii distribuții și utilizării, precum și datorită unei suficient de bune precizii de măsurare. Prin urmare, vom folosi scorul T ca o măsură de risc de osteoporoză și osteoartroză.

5.3. Crearea unei aproximări a scorului T

Scopul nostru este de a utiliza diferiți factori, pentru a ajunge la o valoare finală (de scor T). În alte cuvinte, acesta este un algoritm de consens în definiția extinsă (D1). Se adună informații din diferite surse, se decide importanța lor și se trage o concluzie, în acest caz este scorul T.

5.3.1. Selectarea factorilor utilizați

În primul rând am trebuit să ne decidem asupra factorilor necesari pentru a estima scorul T. Selecția de factori ar trebui să fie larg răspândită în regiunea noastră, ușor de măsurat și să fie aibă influențe semnificative asupra acestor boli. De asemenea, o altă cerință a fost că numărul total de factori selectați să fie relativ mic (≤ 10).

Din start beneficiem de faptul că populația în județul nostru deja prezintă doi factori de risc: rasă caucaziană (99%) și clima blândă (100%).

Restul de factori au fost supuși la o analiză după cum se arată în tabelul următor. Inactivitatea fiind un factor foarte subiectiv a fost împărțită în alte 3 factori (derivați de la start-up factorii prezentați în Figura 5.1): nivel de dureri, nivel de mobilitate și fracturi anterioare.

Coeficientul de frecvență (a factorului de risc, raportat la populația totală) și a coeficientului de influență (gradul în care acest factor de risc poate influența modificarea de scor T) au fost bazate pe diferite studii regionale [EdTB06], [w3HS07], [MaTD06] și [RoCP99].

Factor	Measurement				Frequency	Observations
	unit	difficulty	accuracy	influence		
sex	female/ male	0	10	9	10	
age	years	0	9	9	10	rounding of the months
body mass index (BMI)	numeric value	5	10	7	10	correlation of height and weight
alcohol	l (liter)/ week	3	9	6	8	calculate alcohol percentage of drink
coffee	cups a day	0	8	8	6	different size, concentrations, etc
calcium intake	mg	9	4	9	10	difficulties in assessing
smoking	cigarettes a day	1	9	7	7	must considerate passive smokers
pregnancy	numeric value	0	10	10	4	0 & 4+ kids - high risk
pain level	numeric value	2	2	10	10	highly subjective
mobility	numeric value	2	2	10	10	same as above
previous fractures	numeric value	1	10	7	9	antecedents; limited mobility
menopause start-age	years	1	9	10	4	rounding of the months
estrogen deficiency	ng/dL	10	9	10	3	different periods of the cycle
steroid usage	ng/dL	10	9	8	1	same as above
hyperparathyroidism	μU/mL	10	10	7	2	same as above

antacids	pg/dL	9	9	5	4	different absorption rate, concentrations, treatment duration
heparin	mg/day	8	9	5	3	same as above
gluco-corticoids	mg/day	9	9	6	4	same as above
anticonvulsants	mg/day	9	9	5	2	same as above
immunosuppressants	mg/day	9	9	7	1	same as above

Tabelul 5.2. Analiza factorilor de risc (10-maxim, 0-minim) [ABVS08]

Toți acești factori au fost testați în toate combinațiile posibile în grupuri de 10, cu scopul de a maximiza următoarea funcție de consens[SpAM04]:

$$F = \sum_{i=1}^{10} (k_0 d_i + k_1 a_i + k_2 p_i + k_3 r_i)$$

Ecuția 5.1.

în cazul în care:

d_i - dificultate în evaluarea factor de risc i (coloana 3 din Tabelul 5.2)

a_i - precizie în măsurarea de factor de risc i (coloana 4)

p_i - de influență a factor de risc i (coloana 5)

r_i - frecvență de factor de risc i (coloana 6)

k_i - coeficientul de importanță atribuit

i - orice factor de risc din Tabelul 5.2.

Am maximizat frecvența și ușurința în măsurare nefiind atât de concentrați pe precizie, având următorul raport între coeficienți.

$$k_0 > k_3 > k_1 > k_2$$

Ecuția 5.2

Pentru că toate valorile din Tabelul 5.2 care participă la acest calcul sunt din același interval $[0, 10]$, toți coeficienții trebuie, de asemenea, să aparțină unui interval comun, pe care l-am considerat a fi tot $[0, 10]$

Bazat pe toate acestea am ales primii 10 factori. Singurul factor non-liniară, în top 10 a fost existența sarcinii, și ne-am decis, din motive de simplitate, a-l înlocui cu următorul de pe listă.

O altă înlocuire a avut loc în faza de testare, IMC, care a fost înlocuit de înălțime și greutate. Singur, nici unul din aceste două elemente nu sunt considerate un factor de risc separat, doar fiind corelați prezintă această proprietate. Motivul din spatele acestei decizii este că multe persoane nu știu ce este IMC, dar nu au nici o problemă în a-și identifica înălțimea și greutatea.

Selecția de factori finală est: sex, vârstă, înălțime, greutate, fumat, alcool, cafea, nivel de durere, nivelul de mobilitate, fracturi anterioare, vârstă începere menopauză.

5.3.2. Cuantizarea factorilor de risc

Toți factorii de risc au trebuit să fie exprimați în cifre și, de preferință, într-un mod simplu și ușor de înțeles.

Factorul sex a fost codificat binar: 0 - de sex masculin, 1 - de sex feminin. Acest factor va fi reprezentat acum ca S.

Factorul vârstă a fost exprimat în ani. Pentru a fi calculat s-a normalizat [TaDu00]: $\text{vârsta}/100$, simbol VN.

Nivelul de durere va fi exprimat ca valoare numerică între 0 - nici o durere și 10 - dureri enorme. Acest factor va fi prea normalizat $\text{pain_level}/10$ și va fi simbolizat prin DN.

Nivelul de mobilitate va fi exprimat ca valoare numerică între 10 - complet mobil și 0 - imobil. Acest factor va fi de asemenea normalizat $(10\text{-mobility_level}) / 10$ și simbolul său va fi MN.

Faptul că nivelul de durere și cel de mobilitate sunt valori în același interval, dar cu semnificații diferite (0 având un sens pozitiv, pentru dureri, și negativ, în sensul de mobilitate), poate părea un pic confuz, dar testarea arată că acest mod de notație este găsit de oameni a fi intuitiv, undeva de-a lungul termenilor: "mobilitate zero" sau "zero durere".

Factorul IMC va fi calculat din înălțime și greutate, în mod obișnuit: $(\text{înălțime în cm-greutate în kg}) / \text{greutatea în kg}$.

Vârsta de start a menopauzei va fi, de asemenea, normalizată, dar într-un mod diferit: $(\text{vârsta curentă} - \text{vârsta intrare menopauză}) / 10$ și va fi menționat în continuare ca EN.

Factorul alcool (A) va fi exprimată în litri de alcool pur consumat în timpul unei săptămâni, și nu vor fi modificate în nici un fel.

Un alt factor nemodificate este numărul de fracturi anterioare, în continuare desemnate ca P.

Aportul de cofeină a fost rezolvată prin împărțirea numărului de cești de cafea pe zi la 5 și numirea acestui parametru HN.

Numărul de țigări pe zi a trebuit a fi de asemenea normalizat și, prin urmare, a fost împărțit la 30, devenind CN.

O situație interesantă apare în cazul nefumătorilor. Dacă nefumătorul este un fumător pasiv, el trebuie să numere țigările inhalate zilnic. Acest număr este apoi înmulțit cu 3 dacă pacientul este de sex masculin și 4 în cazul în care pacientului este de sex feminin [IARC03].

5.3.3. Crearea formulei de aproximare

Am pornit de la următoarele:

$$- T\text{-score} = c_s \times S + c_v \times VN + c_d \times DN + c_m/MN + c_b \times BMI + c_e \times EN + c_a \times A + c_p \times P + c_c \times CN + c_h \times HN$$

Ecuția 5.3

unde c_* reprezintă un coeficient de importanță.

Algoritmul a început cu valoarea coeficientului proporțională cu suma din coloanele 3 - 5 din Tabelul 5.2, care să corespundă factor de risc de care aparține coeficientul.

Am folosit o funcție, similară cu funcția de la ecuația 5.1, pentru a crea coeficienții:

$$C_i = n_0 d_i + n_1 a_i + n_2 p_i$$

Ecuția 5.4.

în cazul în care:

d_i - dificultate în evaluarea factor de risc i (coloana 3 din Tabelul 5.2)

a_i - precizie în măsurarea de factor de risc i (coloana 4)

p_i - de influență a factor de risc i (coloana 5)

n_i - coeficient de importanță atribuit

i - orice factor de risc selectat

Principala diferență a fost că acum accentul a fost pe influență și precizie.

$$n_2 > n_1 > n_0$$

Ecuția 5.5 . Relația dintre coeficienți

Inițializarea este importantă, deoarece se reduce timpul de calcul și pentru fiind deja bazat pe statistici, factorii sunt corect echilibrați [FHMV03] de la început.

În scopul de a testa și de a se adapta coeficienții, am folosit câteva cazuri tipice (30) și programul a ajustat valorile coeficienților pentru a se potrivi acestor cazuri.

Formula obținută este:

$$- \text{estimated } T\text{-score} = 0.276 \times S + 0.318 \times VN + 0.297 \times DN + 0.259 \times MN + 0.342 \times P + 0.264 \times CN + 0.323 \times HN + 0.164 \times BMI + 0.148 \times BMI A + 0.58 \times EN$$

Ecuția 5.6

Această formulă este consistentă cu scopul nostru inițial: ușor de măsurat și calculat.

5.4. Testarea formulei

Studiul de validare practică al formulei a fost efectuat la Spitalul Universitar CF Timisoara, pe un lot de 129 de pacienți. Toți pacienții au fost diagnosticați, prin mijloace clinice și examenele paraclinice complexe, orientate pe sistemul musculo-scheletal cu osteoartrita și osteoporoză. Osteoporoza a fost găsit la 73 de pacienți, 67 femei și 6 bărbați. Vârsta lor a variat între 61 și 86 de ani, cu o medie de 65.2 ani. Restul de 56 (34 femei, 22 bărbați) pacienți au avut boală degenerativă localizată. Toți pacienții au fost supuși la DXA și au fost invitați să completeze un formular care conține datele necesare pentru estimarea noastră. Din aceste date a fost calculat scorul T estimat.

O comparație, în valoare absolută, între scor T DXA și scor T estimat (calculat) este descrisă în Figura 5.2. Pe axa valoare (Oy) este eroarea relativă scor T estimat, comparativ cu valoarea DXA. Pe celalată axă se exprimă în procente numărul de pacienți, număr care este prezentat, de asemenea, în fiecare categorie. Pacienții de sex masculin, sunt prezentați pe dreapta, pentru a-i diferenția pe deplin de pacienții de sex feminin.

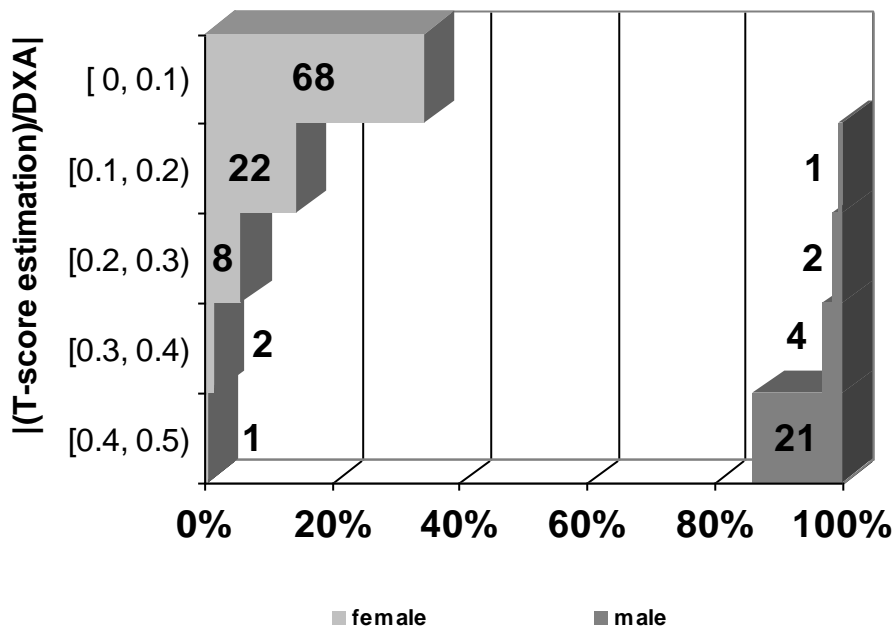


Figura 5.2. Comparația între scor T estimat și scor T măsurat DXA. [SBGA07]

Rezultatele arată o bună apropiere a scorului T estimat, în special pentru pacienții de sex feminin. 11 din pacienții de sex feminin, au fost în afara intervalului

principal de precizie pentru că au suferit de probleme relativ rare, precum sindrom de malabsorbție sau utilizare pe termen lung de cortizon sau de heparină.

Pacienții de sex masculin, totuși a avut mult mai multe erori decât pacienții de sex feminin, în mod evident, datorită faptului că factorul menopauză nu se putea aplica.

Grupare însă a punctajului pentru sex masculin, crează posibilitatea de a propune o altă versiune a formulă într-o variantă non-lineară.

5.5. Concluzii

Estimarea scorului T prezentată aici este o modalitate simplă și rapidă de a determina necesitatea de examinare la medicul specialist, ușurința în calcul făcând-o o procedură de screening eficientă în principal pentru pacienții de sex feminin.

Metodologia implicată în crearea formulei poate fi aplicat la orice alte tulburări, dar trebuie să fie ajustate pe fiecare, cum ar fi de exemplu pentru tulburări gastrice[SBGA06]

În viitor dorim să complicăm această formulă, prin adăugarea de elemente non-lineare și de factori mai greu de estimat. Noua formula are scopul de a fi un instrument competitiv de diagnostic pentru medici.

6. Concluzii

În cadrul diverselor modalități de obținere a toleranței la defectare a unui sistem una dintre modalitățile considerate a fi complexe este folosirea unor algoritmi care pornind de la datele anunțate de un set de componente să selecteze un set de date corect pe bază căruia să realizeze o acțiune. Acești algoritmi se numesc algoritmi de consens.

Studiind acești algoritmi din cadrul științei calculatoarelor, am identificat anumite limitări ale acestora: mai concret, definiția din știința calculatoarelor cere ca toate datele inițiale să apară simultan, iar rezultatul final trebuie să se regăsească între datele inițiale.

În realitatea umană însă, consensul are un înțeles mai larg decât cel din știința calculatoarelor. Consensul uman acceptă noi date, extrapolări și prelucrări mai complexe ale datelor.

În consecință am formulat o nouă definiție a consensului, pe care am modelat-o matematic și care am demonstrat că include toate problemele considerate clasice de consens din știința calculatoarelor.

Consensul este o problemă curentă în toate aspectele vieții, precum și în foarte multe din aspectele ce implică sistemele de calcul. Această problemă a fost studiată în știința calculatoarelor deocamdată exclusiv din punctul de vedere al acestei discipline. În această lucrare s-a propus o nouă definiție a termenului de consens:

Consensul reprezintă rezultatul final al uneia sau mai multe prelucrări realizate asupra unui șir de valori inițiale; rezultatul final fiind mai apoi recunoscut și însușit, la sfârșitul dezbaterilor, de către suficient de mulți participanți la prelucrări.

Această definiție a fost mai apoi formalizată matematic, demonstrându-se că include toate definițiile existente în literatură și totodată extinde clasa problemelor de consens.

Unul dintre avantajele apărute prin formalismul introdus este că orice modalitate de rezolvare a problemei, va putea fi aplicată tuturor sub-claselor de probleme pe care le descrie.

Aplicarea teoriei ergodice în domeniul strict al consensului este o idee nouă, teoria ergodică fiind până acum aplicată în știința calculatoarelor doar în domeniul rețelelor [FINa89] sau al calculului cuantic [MaSu01]. Formalismul matematic propus facilitează aplicarea acestei teorii pentru funcția de consens.

Cele trei variante clasice ale problemei generalilor bizantini: CBGP – problema generalilor bizantini cu un comandant suprem, WBGP – problema generalilor bizantini cu aproximare care folosește valori numerice reale, precum și clasa BGP – problema generalilor bizantini egali în rang s-au prezentat pe larg în capitolul 4. Se studiază algoritmi corespunzători acestor probleme prin prisma celor două tipuri de mesaje posibile: mesaje orale și mesaje scrise și se prezintă demonstrațiile matematice clasice detaliate pentru toate problemele expuse.

Cele două tipuri de mesaje cu care funcționează BGP reprezintă practic extreme: detecția performantă versus viteză. Nu se poate alege o variantă ideală atât ca și detecție cât și ca și viteză, fie și numai din cauza acestor opțiuni strict distincte.

Din prezentarea celor trei variante de BGP, se mai deduce încă o concluzie și anume că timpul necesar rulării acestui algoritm este deosebit de consistent. Prin relaxarea ipotezelor (WBGP) se poate obține un timp mai rezonabil, dar în defavoarea performanței.

Trebuie remarcat faptul că cele două modele matematice: cel propus și cel clasic nu se exclud, ci se completează. Modelul propus este bazat pe o abordare de tip black-box, în timp ce modelul clasic, fiind croit exact pe o sub-problemă ne descrie interiorul cutiei negre.

Din Figura 3.8. Relația între modelele matematice se evidențiază generalitatea modelului propus și limitarea pe un caz particular a modelului clasic. Piramida constructorului apare drept cuantificator al ipotezelor problemei particulare care se dorește a fi descrisă.

Pentru partea experimentală, am căutat o arhitectură de sistem distribuit pe care să testez algoritmi de consens. Studiind diverse rețele singurul lucru constant a fost evoluția lor. În consecință am studiat evoluția sistemelor de calcul și predicțiile lui Bell din 2008 pentru ceea ce va veni. Toată această discuție referitoare la modul cum evoluează calculatoarele are drept scop marcarea momentului actual drept unul în care o clasă de calculatoare practic este depășită de către alta.

Aceasta înseamnă că orice model teoretic nu este de ajuns să-l testăm pe o arhitectură actuală, deoarece în acest caz durata de viață a acestuia este destul de limitată. Ideal ar fi să-l testăm direct pe noua clasă de computație, dar aceasta nu este încă complet funcțională.

Tendința evolutivă este folosirea simultană a unei varietăți cât mai mari de echipamente, extinzând în același timp fiabilitatea totală a sistemului. Arhitectura cea mai apropiată de acest deziderat, AmI, este proiectată pentru a fi utilizată într-un mediu potrivit pentru a fi controlat; prin urmare, principala sa țintă este casa „inteligentă”. [WeRA05]

Într-un astfel de mediu, pe lângă senzori, actuatori și prelucrare multimedia, un alt participant ar putea apărea: PC-ul. Acest participant nu este luat în considerare în abordarea AmI.

Pe de altă parte, astăzi conceptul de ambient multimedia este centrat pe un PC care controlează toată rețeaua multimedia. În cazul în care PC-ul central cade, întreaga rețea este blocată – nu mai funcționează.

Pentru a rezuma: în cadrul AmI, PC-ul nu este luat în considerare, iar în rețelele multimedia de astăzi PC-ul este principalul punct de susținere, în timp ce în aproape toate tipurile de rețele existente astăzi PC-ul este prezent.

Prin urmare, am încercat să adăugăm PC-ul la AmI.

Astfel, am identificat în cadrul treptelor de computație o nișă liberă – care are ca și avantaj generalitatea, dezavantajul constând în aspectul de arhitectură de tranziție. Arhitectura aceasta am botezat-o „intelligent grid”.

Fiind o arhitectură atât de generală, am implementat diversele sale aspecte ținând spre o arhitectură tolerantă la defectare, bazată pe protocoale de consens. Neputând implementa fizic protocoale pe această arhitectură bazată pe AmI, deoarece AmI nu există încă decât câteva sub forma unor prototipuri limitate, am încercat să emulăm pe o rețea existentă arhitectura hibridă propusă.

S-a construit această arhitectură începând de la alegerea middleware-ului, continuând cu protocoale de comunicație care minimizează puterea consumată, dar țin cont de toleranța la defectare.

În implementarea toleranței la defectare a acestei arhitecturi s-au folosit toate cele trei mari metode de atingere a acesteia: redundanța, semnături digitale și consensul. Redundanța, fiind în practică modul cel mai comun de atingere a toleranței la defectare, a fost abordată la mai multe niveluri, iar la nivelul colecției de mesaje s-a completat cu protocolul de consens.

O altă aplicație experimentală constă în aplicarea unui protocol de consens în determinarea unui rezultat final din date care se știe a avea erori. Această aplicație a fost testată în domeniul medical în care toate datele sunt cunoscute a fi măsurate cu eroare, rezultatul algoritmului fiind practic o formulă de screening. Metodologia implicată în crearea formulei poate fi aplicat la orice alte tulburări, dar trebuie să fie ajustate pe fiecare, cum ar fi de exemplu pentru tulburări gastrice.

6.1.1. Contribuții originale

Această teză aduce următoarele noutăți:

I. În cadrul algoritmilor de consens

I. 1. Definiția consensului

Datorată neclarităților și inconsistenței definițiilor existente (de exemplu, în definiția inițială din știința calculatoarelor se cere ca toate datele inițiale să apară simultan, iar rezultatul final trebuie să se regăsească între datele inițiale, în timp ce alte definiții nu cer aceasta) am propus o definiție care acoperă toate punctele necesare pentru a declara consensul.

I. 2. Modelul matematic asociat consensului

Consensul nu a fost formalizat în știința calculatoarelor. Modelul obținut de mine evidențiază clar datele inițiale, rezultatul final, aspectul temporal, prelucrarea dintre datele inițiale și finale și condiția de rezolvare.

Cu acest model se pot descrie toate problemele din această clasă într-un mod unitar, se pot determina clar incluziunile de sub-clase de probleme și deosebirile acestora.

Orice algoritm de consens poate fi înlocuit cu altul, dacă rămân aceleași restricții asupra funcției ce descrie algoritmul, prin urmare permițând o plajă largă de algoritmi de consens.

I.3. Demonstrația extinderii problemei consensului dincolo de problema generalilor bizantini

Conform modelului și definiției propuse, problema generalilor bizantini nu este singura problemă de consens, acest lucru permițând folosirea unor algoritmi de tip licitație, mai rapizi față de algoritmi de tip generali bizantini.

Toate aceste contribuții se reflectă în lucrările:

V. Ancusa - Extending the Consensus Problem - International Joint Conferences on Computer, Information, and Systems Sciences, and Engineering (CIS2E 07), Bridgeport, USA, 2007 - *cotată ISI*

V. Ancusa - Extending the Consensus Problem - Innovations and Advanced Techniques in Systems, Computing Sciences and Software Engineering, pag.511-514, Springer Science-Business Media B.V. ISBN 978-1-4020-8734-9, 2008

V. Ancusa - Message redundancy in sensor networks implemented with intelligent agents - IEEE International Workshop on Robot, and Sensors Environments (ROSE 2008) 2008 - *cotată IEEE*

I. 4. Aplicația în medicină a protocoalelor de consens

Pentru obținerea unei formule de screening pentru o anumită maladie, se aplică de mai multe ori algoritmi de consens: în primul rând pentru determinarea factorilor luați în considerare, iar apoi prin rularea pe un set de cazuri "școală" fiecare factor generând un coeficient de încredere. Acești coeficienți sunt coeficienții formulei de screening dorite.

Această contribuție se reflectă în lucrările:

V. Ancusa, R. Bogdan, M. Vladutiu, L. Susan - A Customized Population Screening Method for Osteoporosis and Osteoarthritis -8th International Conference on Technical Informatics (CONTI2008), proc. vol.1, pp.171-174, Timisoara, 2008

L. Susan, A. Pacurari, **V. Ancusa**, I. Romosan, V. Vacariu, C. Dascau - Changes of Life Quality in Elderly People Suffering from Osteoarthritis, Gerontology Today , The International Conference on Gerontology: Arad sept. 2006, pp. 304-308, ISBN 978-973-8437-96-8 , 2007

L. Susan, C. Banciu, S. Gotia, **V. Ancusa** A Link between the Helicobacter Pylori and Gastric Disorders at Elder Persons, Falk Symposium, Berlin, 2006

II. În cadrul arhitecturilor de sisteme distribuite:

II. 1. Arhitectura Intelligent Grid

Tendința evolutivă este folosirea simultană a unei varietăți cât mai mari de echipamente, extinzând în același timp fiabilitatea totală a sistemului. Apărând din necesitatea emulării unei arhitecturi din al treilea val, un hibrid între AmI și computational grid, arhitectura Intelligent Grid a fost, în cadrul conferințelor în care am prezentat diverse aspecte legate de ea, considerată extrem de inovativă.

IG este o arhitectură de tranziție, emulată pe o rețea din al doilea val de computație, implementarea ei fizică așteptând apariția noului val de componente tehnologice din al treilea val. AmI este dedicată susținerii vieții umane, nu realizării

de calcule complexe. Rețelele computaționale, pe de altă parte, nu sunt create pentru a opera cu senzori. IG, este mult mai general proiectată. Prin urmare, în conformitate cu Legea lui Bell, IG ar trebui să aibă mai mult succes decât rețele specifice unui anumit scop.

Această contribuție se reflectă în lucrările:

V. Ancusa, R. Bogdan, M. Vladutiu- Discussing the Intelligent Agent Approach in Non-traditional Grids -_The 5th International Conference on Cybernetics and Information Technologies, Systems and Applications: CITSA 2008, Orlando, Florida, SUA, 2008

V. Ancusa, R. Bogdan, M. Vladutiu - Redundancy at Link Level for Non-traditional Grids Implemented with Intelligent Agents - 4th International Conference on Networked Computing and Advanced Information Management(NCM2008), Coreea, 2008 - *cotată ISI, IEEE,ACM*

V. Ancusa, R. Bogdan, M. Vladutiu - Discussing Redundancy Issues in Intelligent Agent-Based Non-traditional Grids - 12th International Conference on Knowledge-Based and Intelligent Information & Engineering Systems, KES 2008, Part II, LNAI 5178, pp. 297-305 2008 - *cotată ISI*

V. Ancusa - Message redundancy in sensor networks implemented with intelligent agents - IEEE International Workshop on Roboti, and Sensors Environments (ROSE 2008) 2008 - *cotată IEEE*

R. Bogdan, **V. Ancusa**, M. Vladutiu - Fault Tolerance Issues in Non-traditional Grids Implemented with Intelligent Agents, 2008 International Conference on Computer and Electrical Engineering (ICCEE 2008), Thailanda, 2008 - *cotată ISI, IEEE,ACM*

R. Bogdan, **V. Ancusa**, M. Vladutiu - Performance Metrics for Information Security in Intelligent Grid, International Conference on Machine Learning and Computing (ICMLC 2009), Australia, 2009- *cotată ISI, IEEE*

R. Bogdan, **V. Ancusa**, M. Vladutiu - Possible Threats in an Intelligent Sensor Grid, 8th International Conference on Technical Informatics (CONTI2008), proc. vol.2, pp.51-56, Timisoara, 2008

II.2. Alegerea middleware-ului pentru această arhitectură

Necesitatea alegerii unui middleware competitiv pentru această arhitectură este crucială pentru a rezolva toate problemele de portabilitate, scalabilitate și configurabilitate. Studiind diversele rețele care sunt incluse în IG, poate fi observat că tipul de abordare middleware ce acoperă toate tipurile de rețele care fac parte din IG este bazat pe agenți inteligenți.

Această contribuție se reflectă în lucrarea

V. Ancusa, R. Bogdan, M. Vladutiu- Discussing the Intelligent Agent Approach in Non-traditional Grids -_The 5th International Conference on Cybernetics and Information Technologies, Systems and Applications: CITSA 2008, Orlando, Florida, SUA, 2008

II.3. Aplicarea redundanței la nivel de legătură în IG

După cum este descris în detaliu în literatura, o eroare apărută pe mediul de transport nu pot fi separată de eșecul componentelor care comunică prin acel mediu de transport [LaPS80]. Dar dacă aceeași componentă este legată redundant la mai multe componente putem depăși această problemă. Astfel, nu numai că putem determina unde este eroarea pe link sau pe componente, putând capabilitățile de tolerare la defectare.

Costul, implicat de cabluri suplimentare, poate fi justificat în anumite aplicații, în care eroarea pe mediu de transmisie are o probabilitate mult mai mare decât probabilitatea de defectare a componentei în sine.

Această contribuție se reflectă în lucrările:

V. Ancusa, R. Bogdan, M. Vladutiu - Redundancy at Link Level for Non-traditional Grids Implemented with Intelligent Agents - 4th International Conference on Networked Computing and Advanced Information Management(NCM2008), Coreia, 2008 - *cotată ISI, IEEE, ACM*

V. Ancusa, R. Bogdan, M. Vladutiu - Discussing Redundancy Issues in Intelligent Agent-Based Non-traditional Grids - 12th International Conference on Knowledge-Based and Intelligent Information & Engineering Systems, KES 2008, Part II, LNAI 5178, pp. 297-305 2008 - *cotată ISI*

II.4. Transmiterea tuturor mesajelor folosind redundanța mesajelor și algoritmi de consens cu/fără semnături digitale

Din anul 1985, Schneider [Schn85] a sugerat utilizarea consensului ca o paradigmă centrală pentru programare distribuită fiabilă. Relativ recent, în [GuSc01] s-a realizat o implementare a acestei teorii, prin crearea unui mod sistematic și generic de a transforma diferite probleme de acord în consens, rezultând astfel crearea unui cadru unificat de a dezvolta protocoale tolerante la erori de acord într-un mod corect și eficient.

Transferând conceptele din [GuSc01] la IG, aceasată problemă poate fi definită pe un set de mesaje schimbate de către agenți dintr-un sistem distribuit. Fiecare agent începe cu o valoare inițială și agenții vor trebui să se decidă, pe o anumită valoare comună.

Un exemplu este următorul scenariu: inițiatorul lansează o cerere pentru un alt agent. Cererea este trimisă la toți agenții din lista inițiatorului, agenți care-l vor ajuta să trimitetă acest mesaj la agentul la care trebuia să ajungă.

De la inițiator va ajunge un set de mesaje (unele dintre ele pot fi distruse/eronate). Receptorul folosește un tip de algoritm BGP pentru a afla care este mesajul corect. După aceasta, receptorul să va conforma cererii inițiatorului (de exemplu printr-o prelucrare a datelor). Apoi, receptorul trimite răspunsul său la toți agenții pe la care a trecut spre el mesajul de la inițiator (aceasta poate include inițiatorul propriu-zis).

Inițiatorul ajunge la un set de răspunsuri de la receptor și de după utilizarea BGP pentru a afla care este răspunsul, inițiatorul poate utiliza răspunsul corect. Prin urmare acest algoritm poate fi descris în termenii de formulei de consens. Singura

diferență de la modul în care este descrisă BGP este prezența unei funcții care descrie:

1. rularea BGP cu privire la valorile inițiale în scopul de a determina corect valoarea inițială
2. prelucrarea efectuată la receptor peste valoarea inițială
3. rularea BGP pe setul final de valori în scopul de a determina corect valoarea finală.

Despre prelucrarea locală nu putem face ipoteze, deoarece nu știm în ce va consta.

În ceea ce privește algoritmul BGP, problema a fost intens studiată. Există mai multe tipuri de BGP (tradițional, cu ipoteze limitate, ne-tradițional) și fiecare din aceste tipuri are două moduri de punere în aplicare a mesajului de transfer: mesaje orale și scrise mesaje. Ultima presupune existența unei semnături digitale pe fiecare mesaj.

Apariția semnăturii digitale afectează timpul proporțional cu complexitatea semnăturii. Ținând cont că timpul de rulare al BGP este exponențial, o semnătură cu complexitate liniară nu afectează semnificativ acest timp de rulare.

Această contribuție se reflectă în lucrările:

- V. Ancusa** - Message redundancy in sensor networks implemented with intelligent agents - IEEE International Workshop on Roboti, and Sensors Environments (ROSE 2008) 2008 - *cotată IEEE*
- R. Bogdan, **V. Ancusa**, M. Vladutiu - Fault Tolerance Issues in Non-traditional Grids Implemented with Intelligent Agents, 2008 International Conference on Computer and Electrical Engineering (ICCEE 2008), Thailanda, 2008 - *cotată ISI, IEEE, ACM*
- R. Bogdan, **V. Ancusa**, M. Vladutiu - Performance Metrics for Information Security in Intelligent Grid, International Conference on Machine Learning and Computing (ICMLC 2009), Australia, 2009- *cotată ISI, IEEE*
- R. Bogdan, **V. Ancusa**, M. Vladutiu - Possible Threats in an Intelligent Sensor Grid, 8th International Conference on Technical Informatics (CONTI2008), proc. vol.2, pp.51-56, Timisoara, 2008

6.2. Dezvoltări viitoare

Împreună cu Răzvan Bogdan, doresc să implementez fizic arhitectura IG, aplicând-o în cadrul unui mediu extrem de important a fi controlat: o casă de bătrâni. Este evidentă necesitatea toleranței la defectare a sub-sistemelor implicate, precum și posibilitatea redirectării puterii de calcul nefolosite în proiecte publice de gen SETI@home.

Bibliografie

No.	Id.	Authors	Title	Publisher	Year
1.	[Aart06]	Emile Aarts	<u>Ambient Intelligence and Open Innovation</u>	The First International Conference on Ambient Intelligence Developments	2006
2.	[Abba04]	Ahmar Abbas	<u>Grid Computing: A Practical Guide to Technology and Applications</u>	ISBN:1584502762 Charles River Media	2004
3.	[ABVS08]	V.Ancusa , R. Bogdan, M.Vladutiu, L.Susan	<u>A Customized Population Screening Method for Osteoporosis and Osteoarthritis</u>	8 th Interantional Conference on technical Informatics (CONTI2008), proc. vol.1, pp.171-174, Timisoara	2008
4.	[AgCh99]	A.Aggarwal, A.K. Chandra	<u>Communication Complexity of PRAMs</u>	Proc. 15th Int. Colloquium on Automata, Languages and Programming , pp. 1- 18	1999
5.	[Alls00]	Glen B. Alleman	<u>Fault-Tolerant System Reliability In The Presence Of Imperfect Diagnostic Coverage</u>	Niwot Colorado	2000
6.	[ALRL04]	Algirdas Avizienis, Jean-Claude Laprie, Brian Randell, Carl E. Landwehr	<u>Basic Concepts and Taxonomy of Dependable and Secure Computing</u>	IEEE Transactions on Dependable and Secure Computing 1(1): 11-33	2004
7.	[AnBV08a]	V. Ancusa , R. Bogdan, M. Vladutiu	<u>Discussing the Intelligent Agent Approach in Non-traditional Grids</u>	The 5th International Conference on Cybernetics and Information Technologies, Systems and Applications: CITSA 2008, Orlando, Florida	2008
8.	[AnBV08b]	V. Ancusa , R. Bogdan, M.	<u>Discussing Redundancy Issues in Intelligent</u>	12th International	2008

		Vladutiu	<u>Agent-Based Non-traditional Grids</u>	Conference on Knowledge-Based and Intelligent Information & Engineering Systems, KES 2008, Part II, LNAI 5178, pp. 297-305	
9.	[AnBV08c]	V. Ancusa, R. Bogdan, M. Vladutiu	<u>Redundancy at Link Level for Non-traditional Grids Implemented with Intelligent Agents</u>	4th International Conference on Networked Computing and Advanced Information Management (NC M2008), Corea	2008
10.	[Ancu07]	V. Ancusa	<u>Extending the Consensus Problem</u>	International Joint Conferences on Computer, Information, and Systems Sciences, and Engineering (CIS2E 07), Bridgeport, USA,	2007
11.	[Ancu08a]	V. Ancusa	<u>Extending the Consensus Problem</u>	Innovations and Advanced Techniques in Systems, Computing Sciences and Software Engineering, pag.511-514, Springer Science+Business Media B.V. ISBN 978-1-4020-8734-9	2008
12.	[Ancu08b]	V. Ancusa	<u>Message redundancy in sensor networks implemented with intelligent agents</u>	IEEE International Workshop on Robotic and Sensors Environments (ROSE 2008)	2008
13.	[AtDG84]	Chagit Attiya, Danny Dolev, Joseph Gil	<u>Asynchronous Byzantine Consensus</u>	Journal of the Association for Computing Machinery	1984

- | | | | | | |
|-----|----------|--|--|---|------|
| 14. | [BAKY02] | Sujoy Basu,
Sameer Adhikari,
Raj Kumar, Yong
Yan, Roland
Hochmuth, Bruce
E. Blaho | <u>mmGrid: Distributed
Resource Management
Infrastructure for
Multimedia Applications</u> | Mobile and Media Systems
Laboratory
HP Laboratories
Palo Alto HPL-
2002-350 | 2002 |
| 15. | [BAKY03] | Sujoy Basu,
Sameer Adhikari,
Raj Kumar, Yong
Yan, Roland
Hochmuth, Bruce
E. Blaho | <u>mmGrid: Distributed
Resource Management
Infrastructure for
Multimedia Applications</u> | 17th International
Parallel and
Distributed
Processing
Symposium,
Nice, France | 2003 |
| 16. | [BCFV07] | A. Bondavalli, A.
Ceccarelli, L. Falai,
M. Vadursi | <u>Foundations of
Measurement Theory
Applied to the
Evaluation of
Dependability
Attributes</u> | 37th Annual
IEEE/IFIP
International
Conference on
Dependable
Systems and
Networks
(DSN'07), June,
pp. 522-533 | 2007 |
| 17. | [BeCG07] | Fabio Bellifemine,
Giovanni Caire,
Dominic
Greenwood | <u>Developing multi-agent
systems with JADE</u> | John Wiley& Sons
Ltd ISBN: 978-0-
470-05747-6
(HB) | 2007 |
| 18. | [Bell08] | Gordon Bell | <u>Bell's Law for the Birth
and Death of Computer
Classes</u> | Communications
of the ACM, Vol
51, No. 1, pp 86-
94 | 2008 |
| 19. | [BeSe98] | Carrie Beam and
Arie Segev | <u>Auctions on the
Internet: A Field Study</u> | Fisher Center for
Information
Technology &
Management,
Walter A. Haas
School of
Business,
University of
California,
Berkeley,
http://haas.berkeley.edu/~citm/neo-proj.html | 1998 |
| 20. | [BeSe99] | Carrie Beam and
Arie Segev | <u>Automated
Negotiations: A Survey
of the State of the Art</u> | Fisher Center for
Information
Technology &
Management,
Walter A. Haas
School of
Business, | 1999 |

- | | | | | | |
|-----|-----------|---|--|--|------|
| | | | | University of California, Berkeley,
http://haas.berkeley.edu/~citm/neo-proj.html | |
| 21. | [Bici06] | E. Bici | <u>Consensus Ontology Generation in a Socially Interacting MultiAgent System</u> | 30th Annual International Computer Software and Applications Conference (COMPSAC'06), September, pp. 279-284 | 2006 |
| 22. | [BoAV08a] | R. Bogdan, V. Ancusa , M. Vladutiu | <u>Possible Threats in an Intelligent Sensor Grid</u> | 8 th Interantional Conference on technical Informatics (CONTI2008), proc. vol.2, pp.51-56, Timisoara | 2008 |
| 23. | [BoAV08b] | R. Bogdan, V. Ancusa , M. Vladutiu | <u>Fault Tolerance Issues in Non-traditional Grids Implemented with Intelligent Agents</u> | 2008 International Conference on Computer and Electrical Engineering (ICCEE 2008), Thailanda | 2008 |
| 24. | [BoAV09] | R. Bogdan, V. Ancusa , M. Vladutiu | <u>Performance Metrics for Information Security in Intelligent Grid</u> | International Conference on Machine Learning and Computing (ICMLC 2009), Australia | 2009 |
| 25. | [Boek02] | Fred Boekhorst | <u>Ambient Intelligence, the Next Paradigm for Consumer Electronics: How will it Affect Silicon?</u> | ISSCC 2002 IEEE International Solid-State Circuits Conference, vol.1, pp. 28-31 | 2002 |
| 26. | [BoGS00] | P. Bonnet, J. E. Gehrke, and P. Seshadri | <u>Querying the PhysicalWorld</u> | IEEE Personal Communications, 7(5):10-15 | 2000 |
| 27. | [BoHS03] | A. Boulis, C.C. Han, and M. B. Srivastava | <u>Design and Implementation of a Framework for Programmable and Efficient Sensor</u> | MobiSys 2003, San Franscisco, USA | 2003 |

- | | | <u>Networks</u> | | | |
|-----|----------|--|--|---|------|
| 28. | [Brac87] | G. Bracha | <u>An O(log n) Expected Rounds Randomized Byzantine Generals Protocol</u> | Journal of the ACM Vol. 34 No. 4, October, pag. 910-920 | 1987 |
| 29. | [Bus06] | Jacques Bus | <u>Which cooperation / competition issues for AmI stakeholders</u> | The First International Conference on Ambient Intelligence Developments | 2006 |
| 30. | [CFLV03] | Miguel Correia, Nuno Ferreira Neves, Lau Cheuk Lung, Paulo Verissimo | <u>Low - Complexity Byzantine-Resilient Consensus</u> | http://www.di.fc.ul.pt/tech-reports | 2003 |
| 31. | [CGVC06] | Bogdan Carbunar, Ananth Grama, Jan Vitek, Octavian Carbunar | <u>Coverage Preserving Redundancy Elimination in Sensor Networks</u> | ACM Transactions on Sensor Networks, Volume 2, Issue 1 | 2006 |
| 32. | [Chak07] | Anirban Chakrabarti | <u>Grid Computing Security</u> | Springer-Verlag, ISBN 978-3-540-44492-3 | 2007 |
| 33. | [Chev04] | Pierre R. Chevillat | <u>Research in Sensor Networks, Platforms, and Wireless Technology</u> | IBM Zurich Research Laboratory | 2004 |
| 34. | [ChKu03] | Chee-Yee Chong and Srikanta P. Kumar | <u>Sensor Networks: Evolution, Opportunities, and Challenges</u> | Proceedings of the IEEE, vol. 91, no. 8 | 2003 |
| 35. | [ChPe03] | Haowen Chan and Adrian Perrig | <u>Security and Privacy in Sensor Networks</u> | IEEE Computer, Vol. 36, No. 10, pp. 103-105 | 2003 |
| 36. | [CoNS05] | Alexandru Coman Mario A. Nascimento Jorg Sandera | <u>Exploiting Redundancy in Sensor Networks for Energy Efficient Processing of Spatiotemporal Region Queries</u> | ACM Proceedings of CIKM'05, Bremen, Germany | 2005 |
| 37. | [Davi01] | A. Davies | <u>The internet computing market</u> | Parabon Computation, white paper | 2001 |
| 38. | [Davi04] | Antony Davies | <u>Computational intermediation and the evolution of computation as a commodity</u> | Applied Economics, vol. 36, pp. 1131-1142 | 2004 |

- | | | | | | |
|-----|----------|---|--|---|------|
| 39. | [DER62] | | <u>Dictionar enciclopedic român</u> | Ed. Politica, vol I
A-C | 1962 |
| 40. | [DLPS86] | D. Dolev, N. Lynch, S. Pinter, E. Stark, W. Weihl | <u>Reaching Approximate Agreement in the Presence of Faults</u> | Journal of the ACM Vol.33 No. 3, July, pag. 499-516 | 1986 |
| 41. | [Dole81] | Dani Dolev | <u>The Byzantine Generals Strike Again</u> | Journal of Algorithms Vol. 3, Nr. 1, pag. 14-30 | 1981 |
| 42. | [DoSt83] | D. Dolev, H. Strong | <u>Authenticated Algorithms for Byzantine Agreement</u> | SIAM Journal of Computer, Vol. 12 Nr. 4, , pag. 656-666 | 1983 |
| 43. | [DwLS88] | C. Dwork, N. Lynch, L. Stockmeyer | <u>Consensus in the presence of partial synchrony</u> | Journal of the ACM, Vol. 35 Nr. 2, pag. 288-323 | 1988 |
| 44. | [EdTB06] | B. Edit, F. Tizedes, G. Bako | <u>Incidence of metabolic syndrome among elderly people</u> | International Conference on Gerontology, Arad, Romania | 2006 |
| 45. | [EkKM04] | Virantha Ekanayake, Clinton Kelly, IV, and Rajit Manohar | <u>An Ultra Low-Power Processor for Sensor Networks</u> | ACM Proceedings of the ASPLOS'04, | 2004 |
| 46. | [ErSc06] | Ekwall, R.; Schiper, A. | <u>Solving Atomic Broadcast with Indirect Consensus</u> | Dependable Systems and Networks, 2006. DSN 2006. International Conference on 25-28 June 2006
Page(s):156 - 165 | 2006 |
| 47. | [FHMV03] | R. Fagin, J. Y. Halpern, Y. Moses, M. Y. Vardi | <u>Reasoning About Knowledge</u> | The MIT Press | 2003 |
| 48. | [FiLP85] | M. Fischer, N. Lynch, M. Peterson | <u>Impossibility of Distributed Consensus with one Faulty Process</u> | Journal of the ACM, Vol. 32, No. 2, | 1985 |
| 49. | [FINa89] | G. Florin, S. Natkin | <u>Necessary and Sufficient Ergodicity Condition for Open Synchronized Queueing Networks</u> | IEEE Transactions on Software Engineering, pp. 367-380 | 1989 |
| 50. | [FLYL05] | Luis Ferreira, Fabiano Lucchese, Tomoari Yasuda, Chin Yau Lee, Carlos Alexandre | <u>Grid Computing in Research and Education s</u> | ibm.com/redbook | 2005 |

- | | | | | | |
|-----|----------|--|---|---|------|
| | | Queiroz, Elton
Minetto, Antonio
Mungioli | | | |
| 51. | [FoIa03] | Ian Foster and
Adriana Iamnitchi | <u>On Death, Taxes, and
the Convergence of
Peer-to-Peer and Grid
Computing</u> | Peer-to-Peer
Systems II,
Lecture Notes in
Computer
Science, Springer
Berlin, pp. 118-
128 | 2003 |
| 52. | [FoKe98] | I. Foster and C.
Kesselman | <u>The Grid: Blueprint for
a Future Computing
Infrastructure</u> | Morgan
Kaufmann
Publishers | 1998 |
| 53. | [Fors05] | Greg Forster | John Locke's Politics of
Moral Consensus | Cambridge
University Press
ISBN
0521842182 | 2005 |
| 54. | [Fost00] | Ian Foster | <u>Internet Computing and
the Emerging Grid</u> | Nature 12/2000 | 2000 |
| 55. | [Fost02] | Ian Foster | <u>What is the Grid? A
Three Point Checklist</u> | Argonne National
Laboratory &
University of
Chicago | 2002 |
| 56. | [FrMR05] | R. Friedman , A.
Mostefaoui , M.
Raynal | <u>Simple and Efficient
Oracle-Based
Consensus Protocols for
Asynchronous
Byzantine Systems</u> | IEEE Transactions
on Dependable
and Secure
Computing,
January, pp. 46-
56 | 2005 |
| 57. | [Gate03] | Bill Gates | <u>The Disappearing
Computer</u> | http://www.micro
soft.com /
presspass
/ofnote/11-02
worldin2003.msp
x | 2003 |
| 58. | [GrLa04] | Jim Gray, Leslie
Lamport | <u>Consensus on
Transaction Commit</u> | Microsoft
Research | 2004 |
| 59. | [GuRa04] | Guerraoui, R.;
Raynal, M. | <u>The information
structure of indulgent
consensus</u> | IEEE Transactions
on Computers
Volume 53,
Issue 4, April
2004
Page(s):453 -
466 | 2004 |
| 60. | [GuSc01] | Rachid Guerraoui,
Andre Schiper | <u>The Generic Consensus
Service</u> | IEEE Transactions
On Software
Engineering, vol.
27, no. 1,
January | 2001 |

- | | | | | | |
|-----|----------|--|---|--|------|
| 61. | [Gust05] | G. Gustafson | <u>Fixed Time</u> | http://www.scl.a2005meslab.gov/Publications/FixedTime/FixedTime.html | 2005 |
| 62. | [HACH07] | Herrera-Viedma, E.; Alonso, S.; Chiclana, F.; Herrera, F.; | <u>A Consensus Model for Group Decision Making With Incomplete Fuzzy Preference Relations</u> | IEEE Transactions on Fuzzy Systems, Volume 15, Issue 5, Oct. 2007
Page(s):863 - 877 | 2007 |
| 63. | [HaHa91] | V. Hadzilacos, J. Halpern | <u>Message-Optimal Protocols for Byzantine Agreement</u> | ACM Symposium on Principles of Distributed Computing, 309-323 | 1991 |
| 64. | [Hall98] | Brian Halla | <u>How the PC Will Disappear</u> | IEEE Computer, Vol. 31, No. 12, pp. 134-136 | 1998 |
| 65. | [Halp02] | Joseph Y. Halpern | <u>A Computer Scientist Looks at Game Theory</u> | Journal of Economic Literature, January 18 | 2002 |
| 66. | [HaMW90] | J. Halpern, Y. Moses, O. Waarts | <u>A Characterization of Eventual Byzantine Agreement</u> | ACM Symposium on Principles of Distributed Computing | 1990 |
| 67. | [HaTo94] | Vassos Hadzilacos and Sam Toueg | <u>A modular approach to fault tolerant broadcasts and related problems</u> | Technical Report # 94-1425, Dept of Computer Science, University of Toronto and Dept of Computer Science, Cornell University, May, http://www.cs.utoronto.ca/~vassos/research/publications/HT94/paper.ps.gz | 1994 |
| 68. | [Haye98] | John P. Hayes | <u>Computer Architecture and Organization</u> | Mc-Graw Hill Book Company | 1998 |
| 69. | [HLLC05] | C.-L. Huang , C.-C. Lo , Y. Li , K.-M. Chao , J.-Y. Chung , Y. Huang | <u>Service Discovery through Multi-Agent Consensus</u> | IEEE International Workshop on Service-Oriented System Engineering (SOSE'05), October, pp. 37- | 2005 |

- | | | | |
|-----|----------|--|--|
| | | | 44 |
| 70. | [HuPJ02] | Y.C. Hu, A. Perrig,
D.B. Johnson | <u>Wormhole detection in wireless ad hoc networks</u>
Tech. Rep. TR01- 2002
384, Department
of Computer
Science, Rice
University |
| 71. | [HuSt99] | Michael N. Huhns,
Larry M. Stephens | <u>Multiagent Systems and Societies of Agents</u>
ISBN 0-262- 1999
23203-0, pp.79-
120 |
| 72. | [IARC03] | International
Agency For
Research On
Cancer | <u>Tobacco Smoke and Involuntary Smoking</u>
IARC Monographs 2003
on the Evaluation
of Carcinogenic
Risks to Humans.
Lyon, France, vol.
83 |
| 73. | [IzMa06] | Izumi, T.;
Masuzawa, T. | <u>Condition adaptation in synchronous consensus</u>
IEEE Transactions 2006
on Computers
Volume 55,
Issue 7, July
2006
Page(s):843 -
853 |
| 74. | [KaDo98] | I. Keidar, D. Dolev | <u>Increasing the resilience of a distributed and replicated database systems</u>
Journal of 1998
Computer and
System Science
special issue with
selected papers
from ACM
SIGACT –
SIGMOD
Symposium on
Principles of
Database
Systems (PODS),
1995, 57 (3):
309-324,
December |
| 75. | [KaMo09] | Kar, S.; Moura,
J.M.F | <u>Distributed Consensus Algorithms in Sensor Networks With Imperfect Communication: Link Failures and Channel Noise</u>
IEEE Transactions 2009
on Signal
Processing
Volume 57,
Issue 1, Jan.
2009
Page(s):355 -
369 |
| 76. | [KaWa03] | C. Karlof, D.
Wagner | <u>Secure routing in sensor networks: Attacks and countermeasures</u>
Ad Hoc Networks, 2003
vol 1, issues 2– 3
(Special Issue on
Sensor Network
Applications and |

- | | | | | |
|-----|----------|---|--|---|
| | | | Protocols), pp. 293-315, Elsevier | |
| 77. | [KBFH04] | D.L. Kasper, E. Braunwald, A. Fauci, S. Hauser, D. Longo, J. L. Jameson | <u>Harrison's Principles of Internal Medicine</u> | 16th Edition. McGraw-Hill Professional; July 2004 |
| 78. | [KBSK04] | Shashank Khanvilkar, Faisal Bashir, Dan Schonfeld, and Ashfaq Khokhar | <u>Multimedia Networks and Communication</u> | Multimedia Networks and Communication, 'The Electrical Engineering Handbook', ed. Wai Chen. Academic Press. ISBN: 0-12-170960-4. 2004 |
| 79. | [Kest95] | Lawrence Kesteloot | <u>Fault-Tolerant Distributed Consensus</u> | January 1995 |
| 80. | [Kins02] | W. Kinsner | <u>Compression and its metrics for multimedia</u> | Proceedings of First IEEE International Conference on Cognitive Informatic, pp: 107-121.s 2002 |
| 81. | [KiWL07] | J.H. Kim, M. S. Waterman, L. M. Li | <u>Accuracy Assessment of Diploid Consensus Sequences</u> | IEEE/ACM Transactions on Computational Biology and Bioinformatics, January, pp. 88-97 2007 |
| 82. | [KQPH07] | X. Kai-ping, W. Qi, H. Pei-lin, L. Han-cheng, H. Ming-xiu | <u>A Conceptual Incentive Mechanism of P2P File Sharing Systems Based on Game Theory</u> | 2007 IFIP International Conference on Network and Parallel Computing Workshops (NPC 2007), September, pp. 77-82 2007 |
| 83. | [KrBM02] | Klaus Krauter Rajkumar Buyya and Muthucumaran Maheswaran | <u>A taxonomy and survey of grid resource management systems for distributed computing</u> | Software - Practice And Experience 32:135-164 2002 |
| 84. | [KuMo05] | M. Kulkarni, B. M. E. Moret | <u>Consensus Methods Using Phylogenetic Databases</u> | IEEE Computational Systems 2005 |

				Bioinformatics Conference - Workshops (CSBW'05), August, pp. 61- 62	
85.	[LaFi84]	Leslie Lamport, Michael Fischer	<u>Byzantine Generals and Transaction Commit Protocols</u>	unpublished, 28April 1982, minor revision: 25 April1984	1984
86.	[Lamp83]	Leslie Lamport	<u>The Weak Byzantine Generals Problem</u>	Journal of the Assoclatton for Computing Machinery, Vol 30, No 3, July, pag 668-676	1983
87.	[Lamp86]	Leslie Lamport	<u>Buridan's Principle</u>	http://research. microsoft.com/us ers/lamport/pubs /pubs.html	1986
88.	[LaMS84]	Leslie Lamport, P. M. Melliar-Smith	<u>Byzantine Clock Synchronization</u>	Journal of the Association for Computing Machinery, Vol. 31, No. 2,	1984
89.	[LaMS85]	Leslie Lamport, P. M. Melliar-Smith	<u>Synchronizing Clocks in the Presence of Faults</u>	Journal of the Association for Computing Machinery, Vol. 32, No. I	1985
90.	[LaPS80]	Leslie Lamport, Marshall Pease, Robert Shostak	<u>Reaching Agreement in the Presence of Faults</u>	Journal of the Association for Computing Machinery, Vol. 27 No. 2	1980
91.	[LaPS82]	Leslie Lamport, Marshall Pease, Robert Shostak	<u>The Byzantine Generals Problem</u>	ACM Trans. on Programing Languages and Systems, Vol. 4 No. 3, pag. 382- 401	1982
92.	[Laro99]		<u>Pettite Larousse</u>	Librarie Larousse, 1999 Paris	
93.	[LeCu02]	P. Levis and D. Culler.	<u>Mate: A Tiny Virtual Machine for Sensor Networks</u>	ASPLOS X, San Jose, USA	2002
94.	[LISS03]	S. Li, S. H. Son, and J. A. Stankovic	<u>Event Detection Services Using Data Service Middleware in Distributed Sensor</u>	IPSN 2003, Palo Alto, USA	2003

- | | | <u>Networks</u> | |
|------|----------|---|---|
| 95. | [LuPP99] | Fabrizio Luccio,
Linda Pagli,
Geppino Pucci | <u>Three non Conventional
Paradigms of Parallel
Computation</u>
Padeborn
Technical Report
1999 |
| 96. | [MaAl06] | J. P. Martin, L.
Alvisi | <u>Fast Byzantine
Consensus</u>
IEEE Transactions
on Dependable
and Secure
Computing, pp.
202-215
2006 |
| 97. | [MaKa05] | Jeff Magee,
Christos
Karamanolis | <u>Distributed Algorithms
Course</u>
Cambridge,
http://www.doc.i
c.ac.uk/~ctk/Cou
rses/DistrAlg
2005 |
| 98. | [MaLo06] | Antonio Maña,
Volkmar Lotz (ed) | <u>Developing Ambient
Intelligence:
Proceedings of the First
International
Conference on Ambient
Intelligence
Developments
(AmID'06)</u>
Springer, ISBN-
13 : 978-2-287-
47469-9
2006 |
| 99. | [MaRS06] | Marculescu, R.;
Rabaey, J.;
Sangiovanni-
Vincentelli, A | <u>Is "Network" the Next
"Big Idea" in Design?</u>
Design,
Automation and
Test in Europe,
2006. DATE '06.
Proceedings,
Volume 1,
Page(s):1 - 3
2006 |
| 100. | [MaSu01] | J.J. Maier , F.J.
Sullivan | <u>Study of an ergodicity
pitfall in multitrajectory
simulation</u>
December 2001
Winter Simulation
Conference
(WSC'01) -
Volume 2, pp.
797-804
2001 |
| 101. | [MaTD06] | D. Mandasescu, M.
Trisca, A.M.
Dascalita | <u>Screening of
osteoporosis in a small
Romanian town</u>
International
Conference on
Gerontology,
Arad, Romania,
Sept. 2006
2006 |
| 102. | [Merc00] | Various authors | <u>The Merck Manual of
Geriatrics</u>
3rd Ed., Merck
&Co. Inc.
2000 |
| 103. | [MFHH02] | S. R. Madden, M.
J. Franklin, J. M.
Hellerstein, and
W. Hong | <u>TAG: a Tiny
Aggregation Service for
Ad-Hoc Sensor
Networks</u>
OSDI 2002,
Boston, USA
2002 |
| 104. | [NeCV05] | N. F. Neves , M.
Correia , P.
Verissimo | <u>Solving Vector
Consensus with a
Wormhole</u>
IEEE Transactions
on Parallel and
Distributed
Systems,
December 2005,
pp. 1120-1131
2005 |

- | | | | | | |
|------|----------|--|---|---|------|
| 105. | [OEMa05] | OE Magazine | <u>Optoelectronic Markets</u> | http://oemagazin 2005
e.com/
fromTheMagazine
/
jan05/busspot.html | 2005 |
| 106. | [OIFM07] | Olfati-Saber, R.;
Fax, J.A.; Murray,
R.M. | <u>Consensus and
Cooperation in
Networked Multi-Agent
Systems</u> | Proceedings of
the IEEE
Volume 95,
Issue 1, Jan.
2007
Page(s):215 -
233 | 2007 |
| 107. | [OnHH05] | K. Ono , M. Harao
, K. Hirata | <u>Multi-Agent Based
Modeling and
Simulation of
Consensus Formations
in Arguments</u> | Third
International
Conference on
Information
Technology and
Applications
(ICITA'05)
Volume 1, July
2005, pp. 264-
267 | 2005 |
| 108. | [OsHi01] | Shinji Osada and
Hiroaki Higaki | <u>Fault-Tolerant
Multimedia
Communication
Networks with QoS-
based Checkpoint
Protocol</u> | IEEE, pp.485-488 | 2001 |
| 109. | [PaDN07] | A. Patel , J.
Degesys , R.
Nagpal | <u>Desynchronization: The
Theory of Self-
Organizing Algorithms
for Round-Robin
Scheduling</u> | First International
Conference on
Self-Adaptive and
Self-Organizing
Systems (SASO
2007), July 2007,
pp. 87-96 | 2007 |
| 110. | [PaHe02] | David A.
Patterson, John L.
Hennesy | <u>Computer Architecture
- A Quantitative
Approach</u> | 3rd edition,
Morgan
Kaufmann
Publishers | 2002 |
| 111. | [PaMa07] | Luis Palafox, J.
Antonio Garcia
Macias | <u>Exploiting Redundancy
for Secure Data
Dissemination in
Wireless Sensor
Networks</u> | Computacion y
Sistemas vol.11
no.2 pp. 129-142 | 2007 |
| 112. | [PRWS07] | S. Pai , T. Roosta ,
S. Wicker , S.
Sastry | <u>Using Social Network
Theory Towards
Development of
Wireless Ad Hoc
Network Trust</u> | 21st International
Conference on
Advanced
Information
Networking and | 2007 |

			Applications Workshops (AINAW'07), May 2007, pp. 443-450	
113.	[Raba04]	Jan Rabaey	<u>Reliability Despite Ultra-low Power – The Ultimate Quest in AWSN</u>	presentation at Stanford University, 2004
114.	[Raba05]	Jan M. Rabaey	<u>Wireless Sensor and Consumer Multimedia Networks – A Story of Converging Trajectories?</u>	presentation, CCNC, Las Vegas 2005
115.	[RaNo02]	Prof. Sheizaf Rafaeli and Avi Noy	<u>Online Auctions and Computer Mediated Communication (CMC) Research</u>	Graduate School Of Business, The University of Haifa 2002
116.	[RBSS07]	J. Rabaey, F. Burghardt, D. Steingart; M. Seeman; P. Wright	<u>Energy Harvesting - A Systems Perspective</u>	Electron Devices Meeting, 2007. IEDM 2007. IEEE International, Page(s):363 - 366 2007
117.	[Reyn01]	Kate Reynolds	<u>Going, Going, Gone! A Survey of Auction Types</u>	Agorics Inc 2001
118.	[Robe00]	L. G. Roberts.	<u>Internet growth trends</u>	IEEE Computer, No. 1 January, pp. 52-58 2000
119.	[RoCP99]	Royal College of Physicians	<u>Osteoporosis: clinical guidelines for prevention and treatment</u>	London: RCP 1999
120.	[Rome04]	Kay Romer	<u>Programming Paradigms and Middleware for Sensor Networks</u>	GI/ITG Fachgespräch Sensornetze, Karlsruhe. 2004
121.	[RoSS06]	T. Roosta, S. Shieh, S. Sastry	<u>Taxonomy of Security Attacks on Sensor Networks</u>	First IEEE International Conference on System Integration and Reliability Improvements. Hanoi, Vietnam 2006
122.	[SBGA06]	L. Susan, C. Banciu, S. Gotia, V. Ancusa	<u>A Link between the Helicobacter Pylori and Gastric Disorders at Elder Persons</u>	Falk Symposium, 2006 Berlin

- | | | | | | |
|------|-----------|---|---|--|------|
| 123. | [Schn84] | Fred B. Schneider | <u>Byzantine generals in action: Implementing fail-stop processors</u> | Association for Computing Machinery TOCS 2,2 (May) 1984, 145-154 | 1984 |
| 124. | [Schn85] | Fred B. Schneider | <u>Paradigms for distributed programs. Distributed Systems. Methods and Tools for Specification</u> | LNCS, Vol. 190., 1985, Springer-Verlag, New York, pag.343-430 | 1985 |
| 125. | [Schn90] | Fred B. Schneider | <u>Implementing Fault-Tolerant Services Using the State Machine Approach: A Tutorial</u> | ACM Computing Surveys, Vol. 22, No. 4 | 1990 |
| 126. | [Schn96] | B. Schneier | <u>Applied Cryptography. Protocols, Algorithms, and Source Code in C</u> | 2nd edition, Wiley | 1996 |
| 127. | [Shar69] | William F. Sharpe | <u>The Economics of Computers</u> | The RAND Corporation | 1969 |
| 128. | [ShSJ01] | C. C. Shen, C. Srisathapornphat, C. Jaikaeo | <u>Sensor Information Networking Architecture and Applications</u> | IEEE Personal Communications, 8(4):52-59 | 2001 |
| 129. | [Smit04] | Roger Smith | <u>Grid Computing: A Brief Technology Analysis</u> | CTOnet.org | 2004 |
| 130. | [SpAM04] | D. J Spiegelhalter, K. R. Abrams, J. P. Myles | <u>Bayesian Approaches to Clinical Trials and Health-Care Evaluation</u> | John Wiley & Sons | 2004 |
| 131. | [SPAR 06] | L. Susan, A. Pacurari, V. Ancusa , I. Romosan, V. Vacariu, C. Dascau | <u>Changes of Life Quality in Elderly People Suffering from Osteoarthritis</u> | International Conference on Gerontology, Arad | 2006 |
| 132. | [SPAR07] | L. Susan, A. Pacurari, V. Ancusa , I. Romosan, V. Vacariu, C. Dascau | <u>Changes of Life Quality in Elderly People Suffering from Osteoarthritis</u> | Gerontology Today , The International Conference on Gerontology: Arad sept. 2006, pp. 304-308, ISBN 978-973-8437-96-8 | 2007 |
| 133. | [SPLR04] | Jeff Shneidman, Peter Pietzuch, Jonathan Ledlie, Mema Roussopoulos, Margo Seltzer, Matt Welsh | <u>Hourglass: An Infrastructure for Connecting Sensor Networks and Applications</u> | Harvard University, Harvard Technical Report TR2104
http://hourglass.eecs.harvard.edu | 2004 |

- | | | | | | |
|------|----------|--|--|--|------|
| 134. | [StMS85] | Stephen. R. Mahaney, Fred B. Schneider | <u>Inexact Agreement: Accuracy, Precision and Graceful Degradation</u> | Proc. Fourth Annual SIGACT-SIGOPS Symposium on Principles of Distributed Computing (Minaki, Ontario, Canada, August 1985), ACM, New York, pag. 237-249 | 1985 |
| 135. | [Swan07] | R. Swanson | <u>Future Developments in Silicon Solar Cells</u> | Electron Devices Meeting, 2007. IEDM 2007. IEEE International, Page(s):359 - 362 | 2007 |
| 136. | [SZZC07] | M. Shao, S. Zhu, W. Zhang, and G. Cao | <u>pDCS: Security and Privacy Support for Data-Centric Sensor Networks</u> | 26th Annual IEEE Conference on Computer Communications, IEEE INFOCOM | 2007 |
| 137. | [TaDu00] | A.C. Tamhane, D.D. Dunlo | <u>Statistics and Data Analysis from Elementary to Intermediate</u> | Prentice Hall | 2000 |
| 138. | [Tane03] | Andrew S. Tanenbaum | <u>Computer Networks, Fourth Edition</u> | Prentice Hall; ISBN: 0-13-066102-3 | 2003 |
| 139. | [TaWo01] | Andrew.S. Tanenbaum, Albert S. Woodhull | <u>Operating Systems, Design and Implementation</u> | Pearson Education Inc., 2nd Edition | 2001 |
| 140. | [ThBu05] | Chen-Khong Tham and Rajkumar Buyya | <u>SensorGrid: Integrating Sensor Networks and Grid Computing</u> | CSI Communications, pp 24- 29 | 2005 |
| 141. | [ToXu01] | Paul Townend and Jie Xu | <u>Fault Tolerance within a Grid Environment</u> | e-Demand project @ University of Durham, UK
http://www.dur.ac.uk/e-demand | 2001 |
| 142. | [VWSP07] | R.Venkatasubramanian, Cynthia Watkins, David Stokes, John Posthill, Chris Caylor | <u>Energy Harvesting for Electronics with Thermoelectric Devices using Nanoscale Materials</u> | Electron Devices Meeting, 2007. IEDM 2007. IEEE International, Page(s):367 - 370 | 2007 |
| 143. | [w3co05] | | <u>On Conflict and Consensus</u> | http://consensus.net , accesat in 2005 | 2005 |

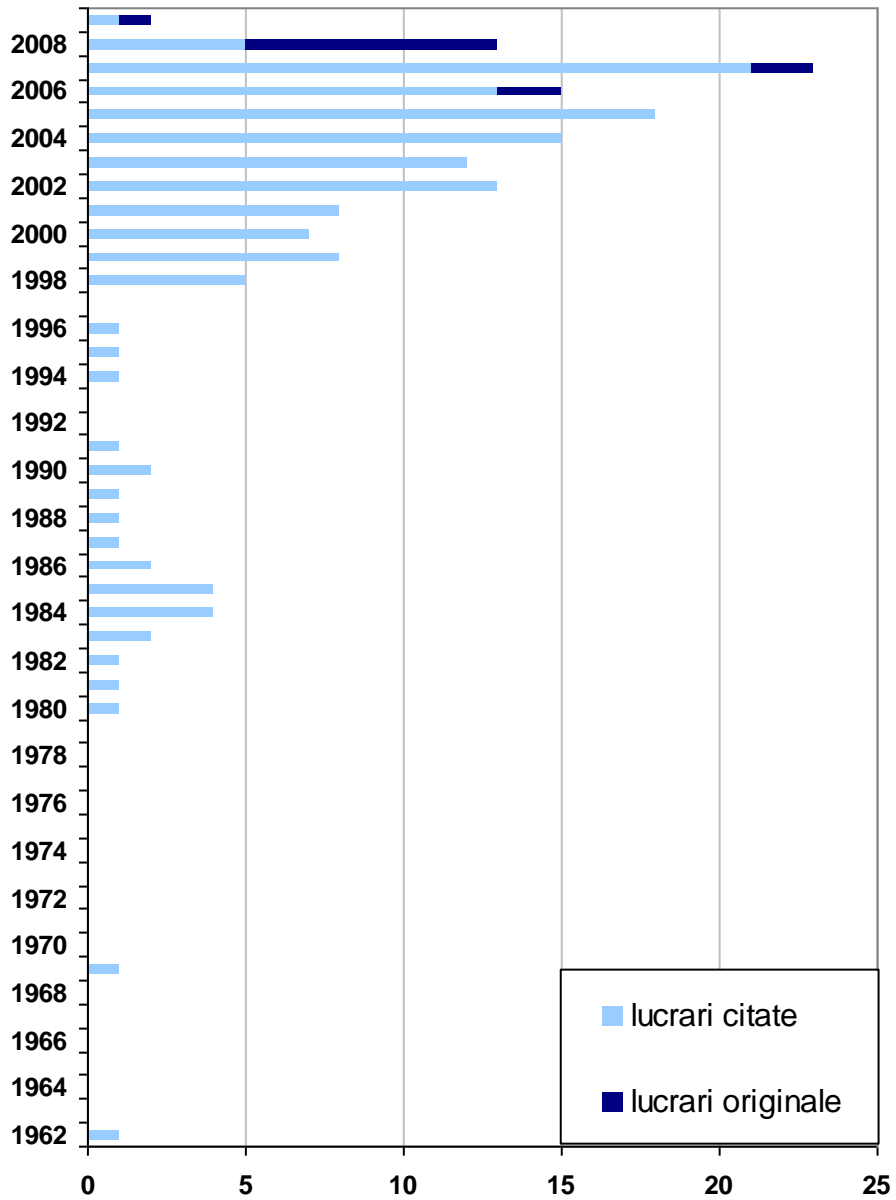
- | | | | | | |
|------|----------|-----------------------------|---|---|------|
| 144. | [w3em05] | ems now | <u>IDC's Asia/Pacific top 10 consumer market predictions for 2005</u> | www.emsnow.co
m | 2005 |
| 145. | [w3Gr05] | Grid today | <u>Platform CEO: Chinese Grid Market to Parallel Economic Growth</u> | www.gridtoday.c
om | 2005 |
| 146. | [w3hp99] | | <u>Cluster Computing for CFD. Evaluating Performance of FLUENT on Hewlett-Packard Linux, HP-UX, and Windows NT Clusters</u> | http://www.hp.co
m/go/fluenthpclu
sters; | 1999 |
| 147. | [w3HS07] | | <u>HSTAT: Guide to Clinical Preventive Services, 3rd Edition: Recommendations and Systematic Evidence Reviews, Guide to Community Preventive Services</u> | http://www.nlm.
nih.gov/index.ht
ml
last accessed:
august 2007 | 2007 |
| 148. | [w3In08] | InHaus | <u>Home >Downloads > Publications</u> | www.inhaus-
zentrum.de/site_
en | 2008 |
| 149. | [w3La08] | Leslie Lamport | <u>Coments on published works</u> | http://research.
microsoft.com/us
ers/lamport/pubs
/pubs.html | 2008 |
| 150. | [w3Ph08] | Philips | <u>Converting dreams into real experience</u> | www.philips.com/
global/index.html | 2008 |
| 151. | [w3PI04] | Platform Computing web site | <u>The Economics of the grid: Grid computing as a strategic IT investment priority</u> | http://www.platf
orm.com/pdfs/se
rvices/profession
al/Platform_Econ
omics_of_Grid.pd
f | 2004 |
| 152. | [w3Su02] | Sun Microsystems web site | <u>Grid Engine White Paper</u> | http://www.sun.c
om/aboutsun/me
dia/presskits/wor
kstations/sc2001/
sunpowers.pdf | 2002 |
| 153. | [w3ve07] | verisign | <u>Top 10 Predictions for the Global Wireless Market</u> | www.verisign.co
m | 2007 |
| 154. | [Wald02] | Mitchel M.Waldrop | <u>Grid Computing</u> | Technology
Review 3/2002 | 2002 |
| 155. | [Wang07] | Zhong Lin Wang | <u>From Nanogenerators to Nano-Piezotronics</u> | Electron Devices
Meeting, 2007.
IEDM 2007. IEEE
International,
Page(s):371- 374 | 2007 |

168 Bibliografie

- | | | | | | |
|------|----------|---|--|--|------|
| 156. | [Want07] | Roy Want | <u>Sensor-Driven Computing Comes of Age</u> | IEEE Pervasive Computing, pp.4-6, April-June 2007 | 2007 |
| 157. | [WBDG05] | Weber, W.; Braun, C.; Dienstuhl, J.; Glaser, R.; Gsottberger, Y.; Knoll, B.; Lauterbach, C.; Leitner, D.; Shi, M.X.; Schnell, M.; Savio, D.; Stromberg, G.; Verbeck, M. | <u>Disappearing electronics and the return of the physical world</u> | VLSI Technology, 2005. (VLSI-TSA-Tech). 2005 IEEE VLSI-TSA International Symposium on Volume , Issue , 25-27, Page(s): 45 - 48 | 2005 |
| 158. | [Webs00] | | <u>Webster's Third New International Dictionary of the English Language</u> | | 2000 |
| 159. | [Weis99] | Gerhard Weiss | <u>Multiagent Systems: A Modern Approach to Distributed Modern Approach to Artificial Intelligence</u> | The MIT Press ISBN 0-262-23203-0 | 1999 |
| 160. | [WeRA05] | W. Weber, J.M. Rabaey and E. Aerts | <u>Ambient Intelligence</u> | Springer-Verlag, ISBN: 978-3-540-23867-6 | 2005 |
| 161. | [Xiao07] | Y. Xiao | <u>Security in distributed, grid, mobile, and pervasive computing</u> | Auerbach Publications | 2007 |
| 162. | [YeMH07] | Eric M.Yeatman, Paul D. Mitcheson, Andrew S.Holmes | <u>Micro-Engineered Devices for Motion Energy Harvesting</u> | Electron Devices Meeting, 2007. IEDM 2007. IEEE International, Page(s):375 - 378 | 2007 |
| 163. | [ZCCL06] | Yang Zhang, Jiannong Cao, Xiaolin Chen, Sanglu Lu, Li Xie | <u>Threshold-based admission control for a multimedia Grid: analysis and performance evaluation</u> | Concurrency And Computation: Practice And Experience, no. 18, pp. 1747-1758; Wiley and Sons | 2006 |
| 164. | [Ziel07] | P. Zielinski | <u>Automatic Verification and Discovery of Byzantine Consensus Protocols</u> | 37th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN'07), June 2007, pp. 72-81 | 2007 |

-
165. [ZiWa08] Feng Xiao; Long Wang Asynchronous Consensus in Continuous-Time Multi-Agent Systems With Switching Topology and Time-Varying Delays IEEE Transactions 2008 on Automatic Control, Vol. 53, Issue 8, Page(s):1804 - 1816

Distribuția în timp a bibliografiei



Anexa – lista de lucrări publicate

V. Ancusa

Extending the Consensus Problem

International Joint Conferences on Computer, Information, and Systems Sciences, and Engineering (CIS2E 07), Bridgeport, USA, 2007

V. Ancusa

Extending the Consensus Problem

Innovations and Advanced Techniques in Systems, Computing Sciences and Software Engineering, pag.511-514, Springer Science+Business Media B.V. ISBN 978-1-4020-8734-9
2008

V. Ancusa

Message redundancy in sensor networks implemented with intelligent agents

IEEE International Workshop on RObotic and Sensors Environments (ROSE 2008)
2008

V. Ancusa, R. Bogdan, M. Vladutiu, L. Susan

A Customized Population Screening Method for Osteoporosis and Osteoarthritis

8th International Conference on technical Informatics (CONTI2008), proc. vol.1, pp.171-174, Timisoara
2008

V. Ancusa, R. Bogdan, M. Vladutiu

Discussing the Intelligent Agent Approach in Non-traditional Grids

The 5th International Conference on Cybernetics and Information Technologies, Systems and Applications: CITSA 2008, Orlando, Florida
2008

V. Ancusa, R. Bogdan, M. Vladutiu

Discussing Redundancy Issues in Intelligent Agent-Based Non-traditional Grids

12th International Conference on Knowledge-Based and Intelligent Information & Engineering Systems, KES 2008, Part II, LNAI 5178, pp. 297–305
2008

V. Ancusa, R. Bogdan, M. Vladutiu

Redundancy at Link Level for Non-traditional Grids Implemented with Intelligent Agents

4th International Conference on Networked Computing and Advanced Information Management(NCM2008), Corea
2008

R. Bogdan, **V. Ancusa**, M. Vladutiu

Possible Threats in an Intelligent Sensor Grid

8th Interantional Conference on technical Informatics (CONTI2008), proc. vol.2,
pp.51-56, Timisoara
2008

R. Bogdan, **V. Ancusa**, M. Vladutiu
Fault Tolerance Issues in Non-traditional Grids Implemented with Intelligent Agents
2008 International Conference on Computer and Electrical Engineering (ICCEE
2008), Thailanda
2008

R. Bogdan, **V. Ancusa**, M. Vladutiu - Performance Metrics for Information Security
in Intelligent Grid, International Conference on Machine Learning and Computing
(ICMLC 2009), Australia
2009

L. Susan, A. Pacurari, **V. Ancusa**, I. Romosan, V. Vacariu, C. Dascau
Changes of Life Quality in Elderly People Suffering from Osteoarthritis
Gerontology Today , The International Conference on Gerontology: Arad sept. 2006,
pp. 304-308,ISBN 978-973-8437-96-8
2007

L. Susan, A. Pacurari, **V. Ancusa**, I. Romosan, V. Vacariu, C. Dascau
Changes of Life Quality in Elderly People Suffering from Osteoarthritis
International Conference on Gerontology, Arad
2006

L. Susan, C. Banciu, S. Gotia, **V. Ancusa**
A Link between the Helicobacter Pylori and Gastric Disorders at Elder Persons
Falk Symposium, Berlin
2006