

Soluții hardware-software IIoT pentru interoperarea și îmbunătățirea sistemelor din industria automotive în contextul Industry 4.0

Teză destinată obținerii
titlului științific de doctor

la

Universitatea Politehnica Timișoara
în domeniul Ingineria sistemelor
de către

Alexandru IOANA

Președintele comisiei: prof.univ.dr.ing.....
Conducător științific: prof.univ.dr.ing. Ioan SILEA
Referenți științifici: prof.univ.dr.
prof.univ.dr.ing.
conf.univ.dr.ing.

Ziua susținerii tezei:

Seriile Teze de doctorat ale UPT sunt:

- | | |
|---|---|
| 1. Automatică | 11. Știința și Ingineria Materialelor |
| 2. Chimie | 12. Ingineria Sistemelor |
| 3. Energetică | 13. Inginerie Energetică |
| 4. Inginerie Chimică | 14. Calculatoare și Tehnologia Informației |
| 5. Inginerie Civilă | 15. Ingineria Materialelor |
| 6. Inginerie Electrică | 16. Inginerie și Management |
| 7. Inginerie Electronică și Telecomunicații | 17. Arhitectură |
| 8. Inginerie Industrială | 18. Inginerie Civilă și Instalații |
| 9. Inginerie mecanică | 19. Inginerie Electronică, Telecomunicații și Tehnologii Informaționale |
| 10. Știința Calculatoarelor | |

Universitatea Politehnica 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, 2022

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

România, 300223 Timișoara, Bd. Vasile Pârvan 2B
Tel./fax 0256 404677
e-mail: editura@upt.ro

Cuvânt înainte

Lucrarea actuală a fost elaborată pe parcursul celor trei ani ai ciclului de studii doctorale, pe care l-am urmat în cadrul Universității Politehnica Timișoara, Departamentul de Automatică și Informatică Aplicată.

Aș dori pe aceasta cale să aduc mulțumiri familiei mele, care m-a sprijinit și m-a încurajat pe toate planurile, întotdeauna. Datorită lor am reușit să traversez cu bine toate etapele de dezvoltare și formare profesională, iar Părinții mei au fost principala sursă de inspirație pentru întreaga mea carieră.

De asemenea, aș dori să îi mulțumesc principalului meu colaborator, domnul conf.dr.ing. Adrian KORODI, pentru susținerea pe care mi-a oferit-o, începând cu anii de facultate și de masterat, culminând în perioada doctoratului. Dumnealui m-a motivat să urmez studiile doctorale și a crezut în mine, încă de când mi-a fost profesor, în anul doi de facultate.

Mulțumiri deosebite se cuvin coordonatorului științific de doctorat, domnul prof.univ.dr.ing. Ioan SILEA, pentru expertiza și ajutorul oferite pe parcursul întregii perioade, fără de care nu ar fi fost posibil acest întreg demers.

Aș dori să îmi exprim de asemenea recunoștința față de toți colaboratorii mei, echipei de cercetare din care am făcut parte, comisiei de îndrumare și întreg personalul didactic și nedidactic al Universității Politehnica Timișoara.

Ca ultime precizări, având în vedere situațiile neobișnuite din ultimii ani, pandemia globală și conflictele externe în apropierea teritoriului României, condițiile în care s-a desfășurat cercetarea doctorală au fost excepționale, iar incertitudinile au fost mereu prezente. Cu ajutorul lui Dumnezeu, am reușit să finalizez studiile de doctorat (prezentând rezultatele în această teză) și să depășesc aceste stări de tensiune. Sper ca în viitorul apropiat situația să se îmbunătățească și să trecem cu toții cu bine peste aceste evenimente nefericite.

Timișoara, septembrie 2022

Alexandru IOANA

IOANA, Alexandru

Soluții hardware-software IIoT pentru interoperarea și îmbunătățirea sistemelor din industria automotive în contextul Industry 4.0

Teze de doctorat ale UPT, Seria X, Nr. YY, Editura Politehnica, 2022, 157 pagini, 52 figuri, 12 tabele.

Cuvinte cheie: Industrial Internet of Things, Automotive, Industry 4.0, OPC UA, SOME/IP, DDS, eCAL, Gateway, comunicație Ethernet, funcționare în timp real.

Rezumat,

Teza curentă vizează dezvoltarea unor soluții inovative de interoperare și îmbunătățire a sistemelor din industria automotive, urmând principiile conceptelor Industry 4.0 și IIoT. Aceste soluții au scopul de a facilita interacțiunea dintre mai multe tehnologii majore, prezente în implementările industriale, în cadrul unor scenarii complexe de comunicare. Principalele tehnologii vizate sunt OPC UA, SOME/IP, DDS și eCAL, fiecare dintre acestea fiind răspândite în sectoare industriale specifice și oferind mecanisme particulare de mare potențial. Soluțiile obținute reprezintă aplicații Gateway asociate unor cazuri de utilizare autentice, analize și îmbunătățiri arhitecturale cu aplicabilitate ridicată, respectiv metode și rezultate cuantificabile ce evidențiază eficiența, stabilitatea și limitele sistemelor actuale de comunicație pe Ethernet. Aceste soluții sunt generate în conformitate cu cerințe industriale referitoare la funcționarea în timp real, gestionarea unui volum semnificativ de date, flexibilitate și scalabilitate. În urma studiului actual, aplicabilitatea tehnologiilor utilizate se extinde către noi scenarii și domenii industriale.

CUPRINS

Notații, abrevieri, acronime	8
Lista de tabele	10
Lista de figuri	11
1. Introducere	14
1.1. Domeniul Tezei	14
1.2. Obiectivele cercetării	17
1.3. Aplicabilitatea Practică vizată de rezultatele cercetării	18
1.4. Structura Tezei	19
2. OPC UA și SOME/IP Gateway în contextul comunicației Car-to-Infrastructure ...	21
2.1. Noțiuni generale / Stadiul actual	21
2.1.1. Contextul Comunicației Car-to Infrastructure	21
2.1.2. Protocoale de comunicare emergente	24
2.2. Metode și tehnologii utilizate	26
2.2.1. Contextul Tehnologiei TSN (Time Sensitive Networking)	26
2.2.2. OPC UA în concordanță cu TSN	27
2.2.3. Mecanismul OPC UA Publish-Subscribe: Noțiuni Introductive	28
2.2.4. OPC UA Publish-Subscribe: Abordare și Aprofundare	29
2.2.5. Contextul funcționării în timp real	30
2.3. Arhitectura Sistemului	32
2.4. Studii de caz	34
2.4.1. Studiu de caz 1	34
2.4.2. Studiu de caz 2	37
2.4.3. Studiu de caz 3	39
2.4.4. Studiu de caz 4	40
2.5. Rezultate	43
2.6. Revendicări și Concluzii	45
3. Îmbunătățirea mecanismului OPC UA Publish-Subscribe în contextul eficientizării comunicației în timp real	47
3.1. Noțiuni generale	47
3.2. Importanța și Aplicabilitatea conceptului	50
3.2.1. Mecanismul Publish-Subscribe: Design și Arhitectură	50
3.2.2. Sincronizarea de timp în contextul OPC UA și TSN	54
3.3. Studiu de caz	57
3.3.1. Arhitectura Sistemului	58
3.3.2. OPC UA Publisher	58
3.3.3. Aplicația Broker	59
3.3.4. Receptorii de tip OPC UA Subscriber	61
3.3.5. Algoritm de Sincronizare	62
3.4. Rezultate și Revendicări	64
3.5. Concluzii	68
4. Abordarea mecanismului OPC UA Publish-Subscribe în contextul transmiterii de imagini	71
4.1. Noțiuni Generale și Aplicabilitatea în Industrie	71
4.2. Metode și Tehnologii utilizate	73
4.2.1. OPC UA Publish-Subscribe în contextul Cercetării și Industriei	74

4.2.2. Conceptul transmiterii de imagini utilizând OPC UA Publish-Subscribe	75
4.3. Arhitectura și Implementare	78
4.4. Studii de caz	80
4.4.1. Studiu de caz inițial pentru evaluarea performanțelor în contextul procesării imaginilor.....	81
4.4.2. Studiu de caz 1 – Transmiterea imaginii pe Un singur canal Publish-Subscribe	81
4.4.3. Studiu de caz 2 – Transmiterea imaginii pe Patru canale Publish-Subscribe	85
4.4.4. Studiu de caz 3 – Transmiterea imaginii pe 20 de canale Publish-Subscribe	85
4.5. Rezultate	89
4.6. Concluzii	90
5. Coexistența protocoalelor DDS și OPC UA în contextul Industry 4.0 și a comunicației în timp real	92
5.1. Noțiuni generale.....	92
5.2. Metode și tehnologii utilizate	94
5.2.1. DDS în contextul IIoT	94
5.2.2. DDS – middleware în ROS2.....	96
5.2.3. OPC UA: în Cercetare și în Industrie	98
5.2.4. TSN: Evoluție, Provocări și Așteptări	98
5.3. Arhitectura Sistemului	99
5.3.1. Comparație cu alte dezvoltări relevante	101
5.4. Studii de caz și Rezultate	102
5.4.1. Studiu de caz 1	103
5.4.2. Rezultatele Studiului de caz 1.....	106
5.4.3. Studiu de caz 2	109
5.4.4. Rezultatele Studiului de caz 2.....	110
5.5. Concluzii	114
6. Tehnologii IoT și Automotive de comunicație, aplicate într-un context V2X prin intermediul unui Gateway Multi-Protocol.....	117
6.1. Noțiuni Introductive	117
6.2. Studiu de aplicabilitate în domeniul automotive	120
6.3. Instrumente și Tehnologii.....	121
6.3.1. SOME/IP.....	121
6.3.2. DDS	123
6.3.3. eCAL	125
6.3.4. Contextul de aplicabilitate vizat și lucrări conexe	125
6.4. Arhitectura Sistemului	126
6.5. Studii de caz si Rezultate	132
6.5.1. Dezvoltarea celor două studii de caz	132
6.5.2. Rezultate	135
6.6. Concluzii	142
7. Concluziile Tezei.....	144
7.1. Concluzii finale	144
7.2. Contribuții personale	145

7.3. Publicații	146
Bibliografie	150

NOTAȚII, ABREVIERI, ACRONIME

Termen	Descriere
AMQP	Advanced Message Queuing Protocol
AOI	Automated Optical Inspection
API	Application Programming Interface
ARM	Advanced RISC Machine
ASIC	Application Specific Integrated Circuit
AUTOSAR	AUTomotive Open System ARchitecture
CAN	Controlled Area Network
CBC	Cell Broadcast Center
CCTV	Closed-Circuit-Television
CPS	Cyber Physical Systems
DDS	Data Distribution Service
eCAL	enhanced Communication Abstraction Layer
ECU	Electronic Control Unit
EOL	End-of-line
FPGA	Field-Programmable Gate Array
GPS	Global-Positioning-Systems
GSM	Global System for Mobile Communication
IIoT	Industrial Internet of Things
LIN	Local Interconnect Network
LTE	Long-Term-Evolution
MES	Manufacturing Execution System
MQTT	Message Queuing Telemetry Transport
OMG	Object Management Group
OPC UA	Open Protocol Communication Unified Architecture
OpenCV	Open-Source Computer Vision Library
OSI	Open System Interconnection

OTA	Over The Air
PLC	Programmable Logic Controller
QoS	Quality of Service
REST	Representational state transfer
ROS	Robot Operating System
RTPS	Real-Time Publish-Subscribe
SCADA	Supervisory Control And Data Acquisition
SDK	Software Development Kit
SOA	Service Oriented Architecture
SOME/IP	Scalable service-Oriented MiddlewarE over Internet Protocol
TCP/UDP/IP	Transmission Control Protocol / User Datagram Protocol / Internet Protocol
TRL	Technology Readiness Levels
TSN	Time-Sensitive Networking
UADP	UA Datagram Protocol
V2I	Car-to-Infrastructure
V2V	Vehicle-to-Vehicle
V2X	Vehicle-to-Everything
YOLO	You Only Look Once

LISTA DE TABELE

Tabelul 1.	Analiza rezultatelor studiilor de caz	44
Tabelul 2.	Analiza rezultatelor studiului de caz.....	68
Tabelul 3.	Analiza rezultatelor studiilor de caz	89
Tabelul 4.	Nod de actualizare DDS – OS Virtualizat Linux – Operațiune de Publicare.	106
Tabelul 5.	Nod de control DDS – OS Nativ Linux – Operațiune de Publicare.	106
Tabelul 6.	Nod de control DDS – OS Nativ Linux – Operațiune de Recepție.	106
Tabelul 7.	Nod de diagnoză DDS – OS Virtualizat Linux – Operațiune de Recepție.	106
Tabelul 8.	Nod de actualizare OPC UA – OS Virtualizat Linux – Operațiune de Publicare.	106
Tabelul 9.	Nod de control OPC UA – OS Nativ Linux – Operațiune de Publicare. .	106
Tabelul 10.	Nod de control OPC UA – OS Nativ Linux – Operațiune de Recepție. ..	107
Tabelul 11.	Nod de diagnoză OPC UA – OS Virtualizat Linux – Operațiune de Recepție.	107
Tabelul 12.	Avantaje și dezavantaje aferente protocoalelor SOME/IP, DDS și eCAL	140

LISTA DE FIGURI

Figura 1.	Protocoalele OPC UA și VSOME/IP în relație cu UDP, TSN, Raw Ethernet, în contextul modelului OSI.....	29
Figura 2.	Arhitectura generală a conceptului VSOME/IP-OPC UA Gateway.....	32
Figura 3.	Arhitectura Hardware în raport cu entitățile dezvoltate	33
Figura 4.	Comunicația Car-to-Infrastructure raportată la studiul actual	34
Figura 5.	Secvența de transmisie în cadrul plasării tuturor entităților pe același dispozitiv	35
Figura 6.	Secvența de transmisie bidirecțională între OPC UA Server, Aplicația Gateway și VSOME/IP Client.....	36
Figura 7.	Aplicația Gateway recepționează valori valide de la un OPC UA Server extern	37
Figura 8.	Clientul OPC UA Softing recepționează valori de la același OPC UA Server extern	37
Figura 9.	Aplicația Gateway accesând trei OPC UA Servere	40
Figura 10.	Secvența configurării mecanismului OPC UA Publish-Subscribe	42
Figura 11.	Dependența între subcomponentele aplicației Gateway, OPC UA și VSOME/IP în contextul Publish-Subscribe în timp real	43
Figura 12.	Propunerea de design și servicii, aferente unui component Middleware, în cazul unui scenariu de funcționare în timp real	52
Figura 13.	Propunere de design în care toate entitățile se bazează pe o bază temporală comună	56
Figura 14.	Arhitectura sistemului	58
Figura 15.	Arhitectura internă și Interacțiunea aplicației Broker	60
Figura 16.	Cele 2 scenarii din perspectiva Receptorilor de tip Subscriber, în legătură cu validitatea și recurența mesajelor. (a) Scenariul 1: -cu recurența mare pentru mesajele valide. (b) Scenariul 2: -cu recurența mică pentru mesajele valide	63
Figura 17.	Terminalul aplicației Publisher și datele trimise către aplicația Broker	66
Figura 18.	Terminalul aplicației Broker recepționând date de la Publisher și transmițându-le receptorilor	66
Figura 19.	Terminalul aplicației Subscriber 1 recepționând datele de interes la intervalul de 3 secunde.....	67
Figura 20.	Terminalul aplicației Subscriber 2 recepționând datele de interes la intervalul de 1 secundă.....	67

Figura 21.	Testarea unității ECU la sfârșitul liniei de producție utilizând procesarea imaginilor în producția de automobile	76
Figura 22.	Etapele pentru transmisia de imagini prin mecanismul OPC UA Publish-Subscribe.....	78
Figura 23.	Arhitectura generală a sistemului.....	79
Figura 24.	Comparație între imaginea recepționată (a) și imaginea originală (b) la o recurență de transmisie de 1 ms/valoare a unui pixel. (Faza 1)	82
Figura 25.	Comparație între imaginea recepționată (a) și imaginea originală (b) la o recurență de transmisie de 5 ms/valoare a unui pixel. (Faza 1)	82
Figura 26.	Comparație între imaginea recepționată (a) și imaginea originală (b) la o recurență de transmisie de 1 ms/valoare a unui pixel. (Faza 2)	83
Figura 27.	Comparație între imaginea recepționată (a) și imaginea originală (b) la o recurență de transmisie de 4 ms/valoare a unui pixel. (Faza 2)	84
Figura 28.	Utilizarea mecanismului OPC UA Publish-Subscribe pe 20 de canale, pentru transmiterea imaginilor cu ambalarea cutiilor la sfârșitul liniei de producție: (a) Imaginea recepționată și (b) Imaginea originală	87
Figura 29.	Utilizarea mecanismului OPC UA Publish-Subscribe pe 20 de canale, pentru transmiterea imaginilor pe baza procesului AOI asupra componentelor ECU: (a) Imaginea recepționată și (b) Imaginea originală	88
Figura 30.	Vizualizarea schematică a coexistenței protocoalelor OPC UA-DDS în contextul Industry 4.0	93
Figura 31.	Arhitectura Sistemului	100
Figura 32.	Nodurile multithreading din perspectiva arhitecturală	105
Figura 33.	Rezultate procentuale bazate pe rata de succes a mecanismului data-buffering.....	108
Figura 34.	Semnalul digital generat pe baza mesajelor specifice livrate de Aplicația Gateway la o recurență de 100 ms.....	110
Figura 35.	Semnalul digital generat pe baza mesajelor specifice livrate de Aplicația Gateway la o recurență de 10 ms.....	111
Figura 36.	Semnalul digital generat pe baza mesajelor specifice livrate de Aplicația Gateway la o recurență de 5 ms	112
Figura 37.	Semnalul digital generat pe baza mesajelor specifice livrate de Aplicația Gateway la o recurență de 2 ms	113
Figura 38.	Semnalul digital generat pe baza mesajelor specifice livrate de Aplicația Gateway la o recurență de 1 ms	113
Figura 39.	Prima perspectivă de aplicabilitate industrială pentru actualul concept gateway multi-protocol.....	120
Figura 40.	A doua perspectivă de aplicabilitate industrială pentru actualul concept gateway multi-protocol.....	121
Figura 41.	Arhitectura Hardware	127
Figura 42.	Arhitectura sistemului, fiind reprezentate ambele versiuni ale aplicației gateway	129

Figura 43. Secvența de proceduri pentru toate nodurile arhitecturii	130
Figura 44. Secvența Data Buffering	131
Figura 45. Rezultatele ratei de succes a mecanismului data buffering	136
Figura 46. Semnalul digital generat pe baza evenimentului de tip heartbeat recepționat și transmis de Aplicația Gateway pentru studiul de caz 1 (deasupra) și studiul de caz 2 (sub) la recurențe de: a) 100 ms b) 10 ms c) 2 ms d) 1 ms	138
Figura 47. Interpretarea pulsului, pe baza evenimentului de tip heartbeat, în conformitate cu secvențele de recepție și de transmisie ale aplicației gateway, în studiul de caz 2	139
Figura 48. O imagine de ansamblu asupra tehnologiilor utilizate	142
Figura 49. Statisticile de accesare pentru a doua lucrare (a se vedea [123])	147
Figura 50. Statisticile de accesare pentru a treia lucrare (a se vedea [124])	148
Figura 51. Statisticile de accesare pentru a patra lucrare (a se vedea [125])	148
Figura 52. Statisticile de accesare pentru a cincea lucrare (a se vedea [126])	149

1. INTRODUCERE

1.1. Domeniul Tezei

În ultimii ani, conceptele Industrial Internet of Things și Industry 4.0 au devenit tot mai prezente atât în cadrul cercetărilor științifice, cât și în industrie. Cele două concepte se bazează pe interoperabilitatea sistemelor complexe, specifice mai multor sectoare industriale, cu scopul îmbunătățirii performanțelor actuale de interacțiune. Interacțiunea sistemelor complexe implică, de la un sector industrial la altul, o multitudine de soluții hardware și software specializate pentru procese industriale particularizate, cu diverse necesități și dependențe tehnologice. De obicei, tehnologiile tradiționale utilizate în industrie la scară largă, au fost concepute cu obiective precise și au fost validate de-a lungul timpului în scenarii bazate pe nevoile unui sector industrial. Astfel, adaptarea soluțiilor actuale către cazuri de utilizare mai ample, reprezintă o provocare și singurul mod fezabil pentru a se realiza presupune o serie de etape de dezvoltare, începând de la nivel arhitectural până la implementarea efectivă a mecanismelor de bază și validarea acestora. Pentru a eficientiza pe cât posibil acest proces granular de dezvoltare, un prim pas trebuie să vizeze identificarea cât mai exactă a cerințelor și nevoilor pentru viitoarele sisteme IIoT, iar mai apoi selectarea tehnologiilor cu cel mai mare potențial și impact în cadrul sectoarelor industriale de interes major.

Domeniul automotive este la ora actuală unul din cele mai dezvoltate domenii industriale, fiind un pilon de bază în economia globală. Raportul dintre cerere și ofertă, facilitează producția în mod constant, iar de la an la an, se investește masiv în cercetare și dezvoltare, dorindu-se o evoluție tehnologică continuă. Ramurile aferente acestui sector sunt proiectarea, fabricarea, asamblarea, testarea și vânzarea de autoturisme. Luând în considerare progresele recente, se poate afirma că domeniul automotive trece printr-o perioadă de tranziție către producția în masă de mașini electrice și autonome. Această tranziție stimulează dezvoltarea de tehnologii hardware și software capabile să susțină industria pe termen lung, conturându-se o nouă viziune ce lărgiște orizonturile explorate până în prezent. Luând în considerare conceptele de conducere autonomă, cât și evoluția preconizată a arhitecturilor și a capabilităților de interconectare și interoperare, este cel mai prielnic moment pentru plasarea în context comun a sectoarelor automotive și IIoT, urmărindu-se principiile de dezvoltare specifice Industry 4.0. Având în vedere faptul că termenul Internet of Things are o acoperire semnificativă, pentru conceperea de soluții avansate de interoperare, va fi nevoie de îmbinarea de tehnologii cu capacități diverse, specifice în general sectoarelor de automatizare industrială, laolaltă cu cele mai avansate tehnologii automotive.

Principalele concepte și tehnologii vizate sunt în general protocoalele avansate de comunicare pe Ethernet, arhitecturile ce implică multiple dispozitive hardware cu capabilități variate și paradigmele de comunicație ce facilitează distribuția de date în mod eficient și de preferat în timp real. Pentru dezvoltarea soluțiilor automotive – IIoT, trebuie urmărite aspecte referitoare la eficiență, fezabilitate, scalabilitate, și

flexibilitate. Eficiența comunicațiilor pe Ethernet stă la baza conceptului IoT și cu siguranță reprezintă o arie de cercetare unde se pot înregistra progrese semnificative cu impact industrial ridicat. Având în vedere etapele de dezvoltare ce implică interacțiunea între domenii industriale diferite, fezabilitatea soluțiilor trebuie luată în considerare din mai multe perspective, evaluându-se atent riscurile. Compatibilitatea și portabilitatea tehnologiilor implicate trebuie demonstrată în mod constant, în cazuri de utilizare cât mai variate. Ulterior, asocierea soluțiilor dezvoltate cu procese industriale concrete poate certifica fezabilitatea în context industrial, fiind îndeplinite numeroase criterii relevante (de eficiența, de stabilitate, de scalabilitate, etc.). Evaluarea scalabilității sistemelor complexe de comunicație pe Ethernet începe de la nivelul arhitectural, unde sunt de interes arhitecturile distribuite orientate pe servicii. Este necesară adaptarea și îmbunătățirea acestor arhitecturi pentru a susține funcționarea în numeroase scenarii și pentru a permite modificările în mod dinamic, pe bază de noi soluții, fără a fi necesară o redefinire la fiecare etapă de dezvoltare. În procesul de selecție a tehnologiilor cu potențial major, multitudinea de mecanisme ce facilitează funcționarea în moduri cât mai variate, oferind numeroase soluții pentru obstacolele și cerințele întâlnite, reprezintă gradul de flexibilitate al sistemului. Combinarea tehnologiilor de nivel înalt, fiecare cu mecanisme proprii de gestionare a datelor și a comunicației Ethernet, crește flexibilitatea și implicit scade probabilitatea de a întâmpina situații critice care să nu poată fi rezolvate.

În cadrul etapelor de cercetare detaliate în actuala teză, tehnologiile vizate aparțin în prezent unor sectoare industriale clar definite, însă utilizate în context comun, oferă perspective noi și rezultate precise ce confirmă progresul și obiectivele propuse. Având în vedere domeniul automatizărilor industriale, ce poate fi perceput în mod indubitabil ca un subcomponent al sectorului IoT, protocolul OPC UA (Open Platform Communications Unified Architecture) a fost utilizat în numeroase prototipuri și concepte originale, cunoscându-se amplitudinea și popularitatea sa atât în context industrial, cât și academic. Asociat protocolului OPC UA, s-au efectuat investigații și experimente în legătură cu tehnologia TSN (Time Sensitive Networking) și sistemele de operare Posix, ambele noțiuni reprezentând teme actuale, de mare interes în foarte multe dezvoltări industriale. Soluția middleware de comunicație pe Ethernet, recunoscută și validată în domeniul automotive este SOME/IP (Scalable service-Oriented MiddlewarE over IP). Fiind protocolul utilizat în majoritatea aplicațiilor disponibile la ora actuală și parte a standardului AUTOSAR (AUTomotive Open System ARchitecture), SOME/IP a fost selectat ca tehnologie de referință în cadrul diverselor etape de cercetare, urmărindu-se o autenticitate cât mai mare în raport cu industria. Adicional tehnologiilor ce pot fi asociate fără echivoc unor sectoare particulare din industrie, se mai regăsesc și tehnologii cu mare potențial, conforme atât cu sectorul automotive cât și cu domeniul IoT, ce nu sunt neapărat plasate și răspândite într-un context specific. Protocolul DDS (Data Distribution Service) și soluția middleware eCAL (enhanced Communication Abstraction Layer) se află în această ipostază, și au fost vizate în implementările aferente diverselor etape de cercetare. În ultimii ani, DDS a fost în continuă expansiune și a început să fie categorisit ca protocol IoT, fiind utilizat în scenarii foarte diversificate, oferind inclusiv conformitate cu domeniul automotive, respectiv cu standardul AUTOSAR. Soluția middleware eCAL a devenit disponibilă mai recent, nu este foarte cunoscută, iar ariile de aplicabilitate încă nu sunt stabilite. Cu toate acestea, eCAL are originile tehnologice în domeniul automotive

și reprezintă o alternativă la soluțiile existente de comunicație pe Ethernet, ce merită explorată în mod detaliat.

Sistemele complexe IoT variază considerabil în funcție de cerințele îndeplinite, iar acest set de cerințe crește în mod constant pe baza nevoilor identificate în fiecare etapă de dezvoltare. Luând în considerare gradul de inovativitate și abordările practice ce evoluează rapid în concordanță cu particularitățile fiecărui sistem, încă nu este stabilită o viziune foarte clară asupra posibilităților și limitelor tehnologice, în special în cazul interacțiunii sectoarelor automotivă și IoT. Prin urmare, este nevoie de soluții ce pot extinde aplicabilitatea tehnologiilor actuale în mod concret, însă în egală măsură este necesară definirea unor criterii clare de evaluare, pe baza cărora se pot identifica limitele soluțiilor disponibile în mediul industrial. Pe baza unor astfel de criterii, este necesar să fie definite și implementate strategii și mecanisme ce pot genera rezultate cuantificabile în legătură cu capacitățile de interoperare și comunicare în timp real.

Din perspectiva academică, direcțiile de cercetare ce se focalizează pe conceptele Industry 4.0 și IoT sunt vaste și se intersectează cu multiple arii tehnologice și industriale. Studiile emergente fac referire de multe ori la cazurile clasice de interacțiune între structuri cloud și procese particulare, plasate într-un context predictibil și utilizate la scară relativ redusă. Abordarea generală se menține la un nivel abstract, iar aplicabilitatea industrială a studiilor nu este imediată. Sectorul automotiv nu este în general perceput având o traiectorie rapidă de dezvoltare, strategiile pieței menținând constant ritmul evoluției. Cu toate acestea, tranziția către era electrificării și starea curentă a sistemelor și tehnologiilor specifice, crează premisa ideală pentru evoluția către sisteme îmbunătățite automotivă – IoT cu beneficii și capabilități substanțiale. Acest proces de dezvoltare se orientează asupra unor cazuri de utilizare specifice conceptelor de conducere autonomă, comunicațiilor de tip car-to-infrastructură și car-to-car (V2V – vehicle to vehicle), dar și asupra particularităților și îmbunătățirilor arhitecturale și funcționale, ce pot fi aduse în cadrul soluțiilor individuale deja existente în industria automotivă și în sectorul IoT.

Conceptele și soluțiile dezvoltate pe parcursul etapelor de cercetare aferente tezei actuale, prezintă într-un spectru cât mai cuprinzător avantajele și dezavantajele tuturor tehnologiilor utilizate, fiind evidențiate numeroase rezultatele relevante, obținute în baza unor criterii de interes universal. În general, soluțiile de interoperare s-au materializat sub forma unor aplicații complexe de tip gateway, inovative, ce plasează în context comun cele mai relevante tehnologii din industrie. Implementările obținute extind aplicabilitatea protocoalelor de comunicare pe Ethernet, perspectivele confirmate în urma studiului contribuie la clarificarea unor teme cu rezonanță în comunitatea academică, iar scenariile de utilizare originale și rezultatele cuantificabile, edifică noi metode și abordări inovative pentru dezvoltările ulterioare, structurate pe principiile Industry 4.0 și IIoT.

1.2. Tema și obiectivele cercetării

Analiza celor mai eficiente tehnologii de comunicație în diverse contexte de operare, îmbunătățirea arhitecturilor curente, a mecanismelor aferente protocoalelor de comunicare vizate și explorarea posibilităților de funcționare în timp real, conturează tema actualei teze, urmărindu-se în final obținerea unor rezultate cât mai elocvente în legătură cu toate aspectele menționate și dezvoltarea unor soluții concrete ce eficientizează procesele industriale și prezintă noi capacități necesare în circumstanțele Industry 4.0.

Obiectivele cercetării doctorale sunt structurate în concordanță cu nevoile industriale și cu temele de interes major din comunitatea academică. Se urmăresc principiile de interoperare asociate conceptelor Industry 4.0 și IIoT, iar soluțiile dezvoltate contribuie la îmbunătățirea sistemelor de comunicație pe Ethernet și la rafinarea arhitecturilor complexe. Conceptele dezvoltate sunt aplicate într-un context diversificat, urmărind la fiecare etapă a cercetării, asocierea cu procese industriale sofisticate. Tehnologiile vizate sunt analizate în mod amănunțit, iar implementările rezultate combină particularități tehnologice, cu scopul eficientizării capacităților de comunicare pe numeroase tipuri de dispozitive și sisteme de operare. Având în vedere multitudinea de protocoale de comunicație vizate, aplicațiile de tip gateway dezvoltate se aliniază criteriilor industriale și evidențiază perspective noi, atât din punct de vedere arhitectural cât și aplicativ. Îmbunătățirile aduse sunt prezentate sub forma unor rezultate precise, iar avantajele și dezavantajele strategiilor și metodelor utilizate, sunt descrise în mod obiectiv, în raport cu cerințele și cazurile de utilizare din sectoarele automotive și IoT.

Principalele obiective sunt următoarele :

1. Dezvoltarea unei aplicații Gateway între protocoalele OPC UA și SOME/IP, în contextul unui scenariu de comunicație de tip Car-to-Infrastructure, fiind utilizate paradigmele Server-Client, respectiv Publish-Subscribe, pentru obținerea pe cât posibil a unei funcționări în timp-real;
2. Analiza și îmbunătățirea mecanismului OPC UA Publish-Subscribe, în concordanță cu cerințele industriei automotive, prin adaptarea arhitecturii pentru constrângeri de operare în timp real și pentru sincronizarea procedurilor de pe dispozitive separate;
3. Extinderea aplicabilității protocolului OPC UA în contextul transmiterii de imagini, prin utilizarea mecanismului Publish-Subscribe;
4. Dezvoltarea unei aplicații Gateway între protocoalele DDS și OPC UA, analizându-se compatibilitatea și eficiența în cadrul unei arhitecturi ce permite utilizarea celor două tehnologii atât în context paralel cât și comun;
5. Conceperea și dezvoltarea unei soluții configurabile de tip gateway multi-protocol, ce poate interfața entități SOME/IP, DDS, eCAL complet separate, extinzând aplicabilitatea către concepte V2X și interacțiuni între structuri de comunicație specifice diferitelor domenii industriale.

Având în vedere anvergura și dificultatea obiectivelor principale, în cadrul fiecărei etape de cercetare s-au stabilit anumite obiective adiționale sau derivate. Acest set complementar de obiective clarifică metode și strategii utilizate pe parcurs,

respectiv detaliază abordări și rezultate obținute în urma numeroaselor experimente și analize.

- A. Analiza impactului tehnologiei TSN în raport cu protocoalele de comunicare pe Ethernet;
- B. Furnizarea unor concepte arhitecturale fezabile și a unui algoritm de sincronizare pentru îmbunătățirea mecanismului OPC UA Publish-Subscribe;
- C. Definirea unei strategii de comunicare UDP multi-canal pentru utilizarea mecanismului OPC UA Publish-Subscribe în contextul transmiterii de imagini;
- D. Definirea unor criterii și metode specifice, ce permit examinarea protocoalelor de comunicare de nivel înalt, în cadrul unor sisteme diversificate;
- E. Definirea unor arhitecturi eficiente în cazurile de interacțiune între multiple tehnologii automotive – IoT;
- F. Generarea unor rezultate cuantificabile ce evidențiază stabilitatea și capabilitatea de funcționare în timp real pentru soluțiile dezvoltate pe parcursul etapelor de cercetare.

1.3. Aplicabilitatea Practică vizată de rezultatele cercetării

Obiectivele actualei teze fac referire la domeniul automotive și la sectorul IoT, fiind urmărite principii de interoperabilitate de interes general. Toate soluțiile dezvoltate pe parcursul etapelor de cercetare sunt asociate unor procese industriale din sectorul automotive, aparținând ariilor de proiectare, de producție sau de fabricație a componentelor. Desigur, multe concepte dezvoltate pot fi aplicate și într-un context industrial diferit. Tehnologiile OPC UA și DDS sunt mai răspândite în sectoarele automatizărilor sau a roboților industriali, iar adaptarea structurilor existente de comunicație bazate pe aceste tehnologii, este în sine un obiectiv industrial de actualitate. Prin rezultatele obținute pe parcursul cercetării, se dorește o clarificare asupra modului în care tehnologiile vizate pot impacta industria, iar metodele și strategiile definite pe parcurs, sunt concepute în ideea utilizării la scară largă și sunt validate în cazuri de utilizare cât mai diversificate. Setul de cerințe pentru sistemele complexe automotive și IoT este într-o continuă evoluție, există multe similarități în ceea ce privește conectivitatea și interoperabilitatea. Dacă soluțiile dezvoltate respectă principiile generale referitoare la eficiență, fezabilitate, flexibilitate și scalabilitate, acestea pot fi utilizate fie într-un context oarecum comun, de exemplu conceptul de comunicație Car-to-Infrastructure, fie în mod specific unui sector individual, de exemplu îmbunătățirile arhitecturale OPC UA pot fi transpuse în sistemele deja existente și utilizate în domeniul automatizărilor sau producției.

Revenind la obiectivele stabilite pentru actuala cercetare, prin utilizarea tehnologiei SOME/IP în paralel cu OPC UA, DDS, eCAL, având în vedere relevanța protocolului în industria automotive, și direcția către conceptele de conducere autonomă și interacțiune fie cu infrastructura fie cu alte structuri separate, soluțiile dezvoltate au aplicabilitate directă în scenariile și procesele prezente în sistemele actuale. Unele soluții dezvoltate extind aplicabilitatea tehnologiilor oferind noi posibilități de utilizare, iar tranziția către era vehiculelor electrice facilitează acest

fenomen, oferind o nouă perspectivă asupra echipamentelor și resurselor hardware disponibile.

Utilizarea protocolului OPC UA pentru transmiterea de imagini are aplicabilitate directă în procesele de fabricație din sectorul automotive, însă inovativitatea conceptului oferă noi oportunități și în cadrul arhitecturilor prezente, deși adaptarea acestora la noile mecanisme Publish-Subscribe poate necesita o perioadă medie de timp.

Soluțiile și rezultatele referitoare la tehnologia DDS, se aliniază curentului de expansiune survenit în ultimii ani, există multiple implementări și utilizări DDS, iar combinarea cu alte tehnologii relevante și raportarea la sectorul automotive, edifică oportunitățile de aplicare în scenarii autentice.

Având în vedere mențiunile anterioare și rezultatele obținute pe parcursul fiecărei etape a cercetării, se poate afirma că nivelul de aplicabilitate pentru cercetarea curentă este foarte ridicat, vizează multiple sectoare industriale, iar soluțiile obținute au fost dezvoltate în permanentă concordanță cu nevoile industriei, edificând teme de mare interes din comunitatea științifică.

1.4. Structura Tezei

Pentru o percepție cât mai clară asupra informațiilor ce urmează a fi prezentate în capitolele următoare, se impune o detaliere a structurii tezei curente, cu scopul de a exista o corelare între parcurgerea în mod cronologic a etapelor de cercetare și obiectivele stabilite și detaliate anterior.

Capitolul introductiv prezintă contextul vizat și are rolul de a evidenția în mod general direcțiile de studiu, conturând o perspectivă inițială în legătură cu secțiunile următoare, mai dense din punct de vedere tehnic. Al doilea capitol descrie procesul de dezvoltare al aplicației gateway OPC UA – SOME/IP, utilizându-se toate paradigmele disponibile, în contextul unui scenariu de comunicație Car-to-Infrastructure. Capitolul al treilea prezintă îmbunătățiri aduse mecanismului OPC UA Publish-Subscribe, cu scopul eficientizării comunicației în timp real. A patra secțiune detaliază abordarea mecanismului OPC UA Publish-Subscribe în contextul transmiterii de imagini, astfel fiind extinsă aplicabilitatea protocolului. În capitolul cinci se analizează coexistența tehnologiilor DDS și OPC UA, urmărindu-se inclusiv procesul de dezvoltare al unei aplicații gateway între cele două protocoale de comunicare. Al șaselea capitol vizează un scenariu de comunicație de tipul Vehicle-to-X, prin dezvoltarea unui gateway multi-protocol, în conformitate cu numeroase cerințe industriale. Ultimul capitol prezintă în mod concis concluziile finale și oferă detalii în legătură cu lucrările științifice publicate pe parcursul acestei cercetări.

Pe baza subsecțiunilor prezentate anterior, se poate identifica corelarea între capitolele tezei și rezolvarea obiectivelor propuse, după cum urmează: Capitolul 2 tratează obiectivele 1, A, E; Capitolul 3 tratează obiectivele 2, B; Capitolul 4 tratează obiectivele 3, C; Capitolul 5 tratează obiectivele 4, D, F; Capitolul 6 tratează obiectivele 5, E, F;

20 | 1. Introducere

Din punct de vedere statistic, actuala lucrare are 157 pagini, conține 52 figuri și 12 tabele, structurate pe baza a 7 capitole.

2. OPC UA ȘI SOME/IP GATEWAY ÎN CONTEXTUL COMUNICAȚIEI CAR-TO- INFRASTRUCTURE

2.1. Noțiuni generale/ Stadiul Actual

Odată cu progresele recente în domeniul interfațării OPC-UA și cerințele în continuă creștere din domeniul automatizărilor industriale, combinate cu configurațiile din ce în ce mai complexe ale ECU-urilor din interiorul vehiculelor și serviciile asociate cu paradigmele de comunicare Car-to-Infrastructure și Car-to-Car, devine clar faptul că decalajul dintre cele două domenii trebuie analizat și completat cu soluții concrete. Acest decalaj s-a produs în principal din cauza rigidității și lipsei de transparență a părților software-hardware ale sectorului automotive precum și din cauza extinderii acestor noi tipuri cerințe din zona comunicației Car-to-Infrastructure. Principalele probleme sunt legate de protocoalele de comunicare pe Ethernet, precum și de opinii conceptuale privind rezolvarea cerințelor și adoptarea unor direcții individuale pentru soluționarea acestora. Lumea industrială se află în era Industry 4.0, iar contextul internetului industrial al lucrurilor (IIoT), facilitatorul principal pentru interfațare este protocolul OPC-UA. Pentru a satisface cerințele industriale legate de volumul mare de date, rate de transfer tot mai mari, nevoia de un număr apreciabil de noduri de comunicație în orice tip de sistem, îmbunătățirea coordonării entităților și serviciilor, protocolul OPC UA oferă o soluție fiabilă prin intermediul mecanismului Publish-Subscribe, disponibil și descris în cadrul specificațiilor tehnice. Acesta este deseori plasat în același context cu tehnologia TSN. Activitatea curentă plasează într-un context comun mecanismul OPC-UA Publish-Subscribe și mecanismul Notify-Subscribe specific protocolului (V)SOME/IP, obținând o mai bună înțelegere a relației actuale dintre nevoile industriei și tehnologiile adecvate și oferind o analiză aprofundată asupra celor mai recente paradigme dezvoltate pentru transmiterea datelor, luând în considerare atât capabilitățile de funcționare în timp real precum și cazurile de utilizare de mare interes în domeniile automatizării și automotive. Rezultatul prevăzut este obținerea unei aplicații de tip (V)SOME/IP – OPC UA Gateway care să includă caracteristicile și serviciile necesare pentru a completa decalajul menționat, în raport cu protocoalele de comunicare utilizate. Rezultatele studiilor de caz efectuate demonstrează eficiența conceptului și oferă o mai bună înțelegere a impactului dintre soluțiile în curs de dezvoltare și cerințele viitoare.

2.1.1. Contextul Comunicației Car-to Infrastructure

Abordările conceptuale de actualitate în contextul IIoT determină direcțiile industriei. În privința necesităților de comunicare pe baza paradigmatelor Car-to-Infrastructure, Intra-Car și Inter-Car communication, fiind abordate diverse strategii.

Aceste strategii se bazează pe următoarele direcții: (a) Creșterea volumelor de date, a ratelor de transfer, a fiabilității și securității protocoalelor; (b) Dezvoltarea de soluții de tip Gateway pentru cele mai bine definite protocoale de comunicare pentru a unifica domeniul automotive și domeniul infrastructurii și, de asemenea, pentru a transla datele specifice comunicației din interiorul vehiculului astfel încât să fie accesibile la diferite niveluri de control ierarhic; (c) Centralizarea datelor de infrastructură și a datelor despre vehicule pentru a oferi soluții și răspunsuri participanților la trafic. Fiecare direcție menționată este relevantă și oferă opțiuni de dezvoltare atât din punct de vedere științific cât și practic. Scopul principal al acestor abordări este obținerea unei infrastructuri inteligente și sigure capabilă să comunice direct cu vehiculele inteligente. Este necesar să fie identificat un plan comun între protocoalele de comunicare, soluțiile disponibile și aceste strategii industriale în continuă dezvoltare, îmbunătățind astfel deopotrivă infrastructura [1-3], respectiv comunicațiile mai puțin dinamice din interiorul vehiculelor [4].

În domeniul automotive, atât pe partea de producție cât și pe partea de dezvoltare hardware-software, în ceea ce privește controlul traficului, evoluțiile actuale se focalizează pe controlul centralizat și integrat al mediului dinamic folosind un model de management. Potrivit sursei [5], se fac cercetări pentru a putea elimina semafoarele din orașele inteligente, identificarea și interpretarea semnalelor de circulație devenind scopul conducerii vehiculelor autonome. Industriile și grupurile de cercetare se confruntă în prezent cu problema activă a detectării și interpretării semafoarelor. În [6], autorii menționează necesitatea algoritmilor genetici pentru sistemele din interiorul vehiculului, pentru detectarea și recunoașterea semafoarelor, inclusiv identificarea problemelor precum ocluziile parțiale și defecțiunile inerente ale LED-urilor sistemelor autonome. Modelul sistemului de control al semaforului are anumite caracteristici pentru o funcționare corectă și sigură: controlerului nu i se permite să intre într-o stare de tip deadlock din cauza combinațiilor neașteptate de acțiuni; ar trebui să poată servi toate fazele semnalului și să revină la o anumită stare inițială. După cum se menționează în [7], pentru a evita coliziunile precum și alte pericole, vehiculul se bazează pe un set de reguli și instrucțiuni stabilite de controlerul semaforului. La baza sistemelor de control al traficului se află semne, lumini și alte dispozitive care transmit indicații, avertismente sau cerințe specifice. În [7], controlerul de semafor a fost implementat utilizând un microcontroler, un FPGA (Field-Programmable Gate Array) și un ASIC (Application Specific Integrated Circuit). Procesul de conducere a autovehiculelor în lumea modernă, așa cum se menționează în [8], înseamnă atât conducere normală cât și autonomă în traficul de rețea, folosind hărți de înaltă precizie, actualizări în timp real și aplicații multimedia. Introducerea comunicației Ethernet în scenariul comunicației de tip Car-to-Infrastructure necesită o cale de acces (aplicații de tip Gateway) la tehnologiile de rețea prezente în interiorul vehiculului, astfel încât să existe o tranziție între datele specifice infrastructurii și datele specifice mașinii, așa cum se menționează în [9]. Autovehiculele din vremurile noastre au în general mai mult de 80 de ECU-uri și necesitatea de comunicare între aceste dispozitive a crescut semnificativ în ultimii ani. În arhitectura actuală a vehiculelor, aplicațiile de tip Gateway reprezintă singura cale de a asigura comunicarea între diferite rețele și tehnologii. Aplicația de tip Gateway poate fi catalogată ca unul dintre cele mai importante componente pentru sistemele distribuite, ce conțin multiple echipamente hardware și multiple tehnologii și aplicații

software, iar aceste aplicații Gateway trebuie să asigure transmiterea eficientă a datelor în rețea, îndeplinind toate constrângerile specifice sistemului.

Interesul pentru vehiculele inteligente a crescut în ultimii ani, în principiu pentru a se încerca reducerea posibilității de accidente, oferindu-se astfel servicii de siguranță pentru participanții la trafic. Automobilul inteligent este un vehicul ce oferă în primul rând siguranță, combinând tehnologia electronică, tehnologia IT și tehnologia specifică domeniului auto pentru a recunoaște starea de siguranță atât pentru spațiul interior cât și pentru spațiul exterior automobilului. Multiple sisteme încorporate sunt implementate ca parte a arhitecturii interne, cu rol specific în menținerea siguranței șoferilor și pietonilor. Paradigma de comunicare V2X (Vehicle-to-Everything) face referire atât la partea de comunicare între vehicule (V2V) cât și la partea de comunicare între vehicul și infrastructură (V2I). În esență, comunicația de tip V2I (sau Car-to-Infrastructure) semnifică schimbul de informații critice de siguranță și control între vehicule și infrastructura rutieră, având ca scop principal prevenirea și reducerea accidentelor auto, oferind o gamă largă de măsuri de siguranță și facilități ce îmbunătățesc procesul de circulație rutieră. Comunicațiile de tipul V2I se pot aplica oricărui tip de vehicul și drum, transformând echipamentele de infrastructură rudimentare în infrastructură inteligentă. Conform sursei [10], centrul de control colectează informații de siguranță, de exemplu: accidente de mașină, ambuteiaje, vreme nefavorabilă; folosind dispozitive LTE (Long-Term-Evolution), CCTV (Closed-Circuit-Television), GPS (Global-Positioning-Systems) din zonele disponibile. Dacă în zona respectivă apar incidente de trafic, acest centru de control inițiază CBC (Cell Broadcast Center) ce are ca obiectiv transmiterea informațiilor pe baza cărora sistemul de securitate trebuie să pornească. Gateway-ul se bazează pe două unități de control Multipoint (Multipoint Control Units, MCUs) separate pe aceeași placă, deoarece aplicațiile software rulează pe sistemul de operare AUTOSAR, acesta fiind optimizat pentru sistemele E/E (Electrical and Electronic systems) ale vehiculului. Prin urmare, un MCU este prezent pentru a transpune datele de pe modulul LTE, sub forma unui eveniment, iar al doilea MCU este prezent pentru a rula platforma AUTOSAR și aplicația software răspunzătoare pentru declanșarea sistemelor de securitate adecvate. Dacă s-a petrecut un accident într-o zonă în care vizibilitatea este redusă, infrastructura trimite informații concrete către vehicule. Aceste informații pot include detalii despre accident precum distanța, banda și viteza de deplasare din zona respectivă. Pe baza acestor informații, Gateway-ul declanșează sistemul de securitate din vehicul. Tehnologia Automotive trebuie să fie foarte precaută în adoptarea de noi sisteme inteligente, siguranța umană fiind prioritară. În [11] se implementează o aplicație Gateway ce rulează pe un sistem de operare embedded-Linux portat pe o placă de dezvoltare ARM (Advanced RISC Machine). Datorită numeroaselor abordări privind aplicațiile de comunicații pentru vehiculele autonome, studiile [12] analizează modele de risc pentru evaluarea dispozitivelor prezente în sistemele inteligente de transport. Studii precum [13] propun deep RL (deep Reinforcement learning) pentru virajul la stânga în cazul vehiculelor autonome conectate la intersecțiile din cadrul paradigmei V2I, procesul focalizându-se pe centralizarea datelor. Autorii din lucrarea [14] subliniază faptul că software-ul aplicativ din industria automotive devine din ce în ce mai necesar. Potrivit [14], există producători de mașini care au actualizat deja soluțiile de conectivitate wireless la cea mai recentă infrastructură industrială, atașând dispozitive cu caracteristici Wi-Fi sau

GSM (Global System for Mobile Communication) la arhitectura internă, oferind conectivitate bazată pe tehnologii 3G/LTE și local, prin puncte de acces Wi-Fi. Soluția software este reprezentată de o aplicație de diagnosticare la distanță. Distribuția UNIX utilizată constă în prezența unui sistem de operare dedicat funcționării în timp real, capabil să îndeplinească constrângerile de timp și consistență a datelor, cerințe specific domeniului automotive.

2.1.2. Protocoale de comunicare emergente

Protocolul OPC UA (Open Platform Communications Unified Architecture) este principala tehnologie de interfațare și interoperare pentru Industry 4.0 și IIoT. Prin urmare, reprezintă tehnologia potrivită pentru a susține infrastructura inteligentă. În [15] autorul prezintă stadiul industriei automotive, inclusiv în relație cu infrastructura semnalelor de trafic, confirmând faptul că OPC UA este protocolul de comunicare recomandat în contextul comunicației Car-to-Infrastructure. Bazat pe sinteza din [16], conectarea dispozitivelor electronice echipate cu capacități de comunicație de tip fieldbus în CPS (Cyber Physical Systems) implică crearea de noi modele ce combină datele livrate de la dispozitivele de bază, cu metadate pe baza cărora sunt structurate și informațiile distribuite în timp real. Soluțiile de actualitate trebuie să se bazeze pe standarde de comunicare acceptate pe scară largă, fiind de asemenea ușor de implementat chiar și în cazurile ce presupun resurse limitate. Autorii exprimă necesitatea de a converti sistemele intra-car bazate pe mesaje pe CAN (Controlled Area Network) în mesaje specifice OPC UA. Lucrarea [16] se bazează pe un server OPC UA folosit pe post de Gateway pentru a partaja informațiile disponibile într-o rețea de control bazată pe CAN. Serverul este implementat folosind SDK-ul (Software Development Kit) C# de la Unified Automation, făcând legătură între datele obținute de pe dispozitivele fizice și metadatele necesare pentru înțelegerea corectă a mesajelor specifice OPC UA. Aceste informații preluate sunt disponibile oricărui client OPC UA conform modelului de date aferent protocolului. Spațiul de adrese se bazează pe modelul client-server și pe comunicația CAN utilizată pentru adresarea mesajelor și comunicarea multicast. CAN-ul gestionează datele de pe dispozitivul fizic, nefiind însă potrivit pentru distribuirea la scară largă a datelor. Pentru un schimb eficient de informații, trebuie stabilită o conexiune singulară între client și server. Clientul poate căuta sau naviga în spațiul de adrese al serverului OPC UA, spațiu ce poate include multiple servere. Dacă o parte din informația dorită este stocată pe un server diferit, se realizează o conexiune prin referințe. Conexiunea realizată într-un mod automatizat de tip Subscription poate facilita un mod de comunicare flexibilă și eficientă, necesară în cazul anumitor scenarii complexe. Autorii lucrării [17] analizează strategiile de comunicare intra-vehicule și prezintă o perspectivă către noi protocoale ce pot facilita acest proces. OPC UA este considerat un protocol de viitor având în vedere experiențele industriei de producție, unde este predominant. Perspectiva oferită nu oferă o soluție concretă, evidențiind posibilitățile de utilizare fie în cazul schimbului de date din interiorul vehiculului, fie pentru interfațarea CAN cu dispozitive externe. Luând în considerație mențiunile din lucrarea [18], vehiculele moderne folosesc resurse ample și dispozitive multiple pentru scenariile de comunicație, fiind necesară o schemă de comunicație de bază pentru a obține

performanțe sustenabile și latență scăzută în cazul tuturor procedurilor ce stau la baza schimbului de date. Protocoalele de comunicare de bază precum CAN și FlexRay rămân relevante, însă adoptarea comunicației în mod majoritar pe Ethernet pentru generațiile viitoare de sisteme complexe, va reprezenta un pas major pentru dezvoltare. Autorii lucrării [18] afirmă că un Gateway de înaltă performanță poate gestiona cerințele de actualitate prezente în scenariile de comunicare, iar numărul mare de ECU-uri (Electronic Control Unit) implicate în procesarea datelor trebuie să conțină funcționalități stricte în timp real. Informațiile de la aceste ECU-uri ar trebui adunate sub forma unui Gateway central, iar comunicarea dintre acestea ar trebui realizată strict prin intermediul Gateway-ului. Potrivit [19], vehiculele devin din ce în ce mai inteligente, conectate ca parte din Internet. În timp ce sunt dezvoltate noi funcții, precum recunoașterea naturală a vorbirii și servicii bazate pe cloud, sistemele moștenite trebuie suplimentate cu noi dezvoltări pentru a eficientiza costurile. Comunicația bazată pe Ethernet pune la dispoziție suficiente mecanisme flexibile ce pot facilita această eficientizare de costuri fără a periclita performanțele vizate. Lucrarea [20] prezintă o generare de chei pentru comunicații securizate atât în contextul paradigmei V2V cât și V2I, fără a se face referire la straturile modelului OSI (Open System Interconnection).

Industria automotive consideră protocolul SOME/IP ca soluție standard pentru comunicațiile pe Ethernet din interiorul vehiculelor, VSOME/IP fiind implementarea open source a protocolului realizată de GENIVI. În ultimii ani, SOME/IP s-a extins ca prezentă atât pe platforma Clasic AUTOSAR cât și pe platforma Adaptive AUTOSAR, fiind prezent și validat în multiple aplicații și scenarii. Deși OPC UA reprezintă o posibilitate pentru îndeplinirea unei game largi de cerințe ale comunicației bazate pe Ethernet specifice domeniului automotive, perspectiva realistă implică interfațarea celor două protocoale pentru scenarii ce implică paradigma de comunicație Car-to infrastructure. Lucrarea [21] este considerată foarte semnificativă, întrucât abordează tehnologia OPC UA oferind o evaluare a aplicabilității în contextul TSN (Time Sensitive Networking) pentru automatizarea fabricilor. De asemenea, lucrarea [22] prezintă un studiu ce plasează tehnologia TSN și dispozitivele utilizate în diverse scenarii de comunicație, într-un context comun. Prezența tot mai clară a studiilor ce fac referire la tehnologiile OPC UA și TSN ca parte comună a unui sistem de comunicație cu capabilități de funcționare în timp real și latență scăzută, precum și interesul industriei pentru conceptele menționate anterior, confirmă importanța interacțiunii între OPC UA și domeniul automotive, respectiv protocolul SOME/IP, oferind direcții concrete de cercetare.

Considerând stadiul curent atât din punct de vedere industrial, cât și academic, studiul actual definește o serie de obiective concrete, bazate pe interacțiunea OPC UA – VSOME/IP:

- Dezvoltarea de aplicații Gateway între OPC UA și VSOME/IP, acoperind multiple scenarii industriale, oferind o analiză detaliată asupra trendurilor și tehnologiilor cu potențial ridicat.
- Implementarea paradigmei de tipul Server-Client și Publish-Subscribe pentru ambele protocoale, prezentând pașii de implementare în mod detaliat.

- Analiza și îndeplinirea cerințelor de funcționare în timp real prin sincronizarea precisă a procedurilor de recepție și transmisie de date pentru toate entitățile implicate.
- Furnizarea unei viziuni/orientări clare asupra celor mai eficiente și fezabile soluții pentru comunicația de tipul Car-to-Infrastructure.

2.2. Metode și tehnologii utilizate

2.2.1. Contextul Tehnologiei TSN (Time Sensitive Networking)

TSN reprezintă o soluție pentru un transfer controlat în timp real al mesajelor critice utilizând echipamente hardware standard. În ultimii ani, comunicarea clasică prin Ethernet a început să fie utilizată atât între controlere cât și între senzori și cloud, sub formă de aplicații cu constrângeri de timp real. Având în vedere că Ethernet-ul este parte a unei multitudini de protocoale de comunicare de nivel înalt, cu aplicații ce rulează pe diverse tipuri de componente hardware, împreună cu implicarea de rețele cu capacități diferite în fluxul de comunicații, necesitatea unei tehnologii de rețea care ar putea garanta livrarea mesajelor într-o manieră sincronizată, ca parte a infrastructurii pentru aplicațiile ce implementează diverse protocoale de comunicare bazate pe Ethernet, a fost doar o chestiune de timp. TSN se bazează pe o serie de standarde individuale, majoritatea referindu-se la nivelul legăturii de date, nivelul 2 al modelului OSI. TSN nu ar trebui clasificat ca fiind un protocol de comunicare, ci mai degrabă ca o tehnologie de bază ce poate fi folosită de protocoalele de comunicare bazate pe Ethernet, precum OPC UA, în cazurile de utilizare unde există cerințe de funcționare în timp real. Principalele caracteristici ale TSN în comparație cu infrastructurile Ethernet existente sunt reprezentate de sincronizarea procedurilor de comunicație implicate indiferent de dispozitivele gazdă, latența minimă și garanția livrării datelor în intervalul configurat. Aplicația software va transmite cerințele particulare rețelei, acestea constând în prioritatea procedurilor, intervalele de timp pentru diverse operațiuni și datele de configurare specifice, iar rețeaua se va asigura că toate serviciile implicate vor garanta livrarea mesajelor de rețea în conformitate cu nevoile aplicației. De asemenea, TSN oferă posibilitatea ca o rețea să fie convergentă, diverse tipuri de protocoale în timp real fiind capabile să comunice și să facă schimb de informații în interiorul unei singure rețele, oferind o flexibilitate greu de atins în cazul infrastructurilor tehnologice diferite. Acest fapt face ca investiția într-o astfel de infrastructură să fie profitabilă pe termen lung.

Tehnologia TSN este compusă din multiple substandarde Ethernet definite în IEEE 802.1 TSN task Group. Scopul principal al standardizării unor părți ale nivelului legăturii de date al modelului OSI, nu este încă pe deplin atins, multe upgrade-uri și modificări fiind încă posibile, însă principalele caracteristici pot fi integrate în produse și aplicații. Considerând că mai multe substandarde sunt implicate în tehnologia TSN, scopul principal fiind funcționalitatea în timp real a rețelei, este esențial să fie menționate cele mai importante dintre acestea. Pentru atingerea scopului principal, toate echipamentele conectate la aceeași rețea trebuie să aibă o referință comună asupra timpului, pe baza căreia se va realiza sincronizarea generală a entităților din interiorul rețelei. IEEE 1588-2008-Precision Time Protocol furnizează Grandmaster

clock-ul rețelei (cel mai precis clock luând în considerare toate dispozitivele participante în rețea), utilizând algoritmi din cadrul rețelei. IEEE 802.1AS-2011- Time Protocol gestionează traficul de date cu timp de livrare garantat. Pentru prioritizarea datelor critice și non-critice, standardul IEEE 802.1Q stabilește 8 clase de trafic. Datorită mecanismului de buffer al switch-urilor de Ethernet, pachetele cu prioritate scăzută pot produce întârzieri pentru date critice, astfel fiind necesare mecanisme suplimentare pentru creșterea eficienței în ceea ce privește prioritatea și programarea fluxurilor de date. Unul dintre aceste mecanisme este Credit Base Shaper (IEEE 802.1Qav). Fiind dezvoltat pentru tehnologia Audio/Video Bridging, un predecesor al tehnologiei actuale TSN, mecanismul Credit Base Shaper este responsabil de alocarea de credite pentru fluxurile de date, credite consumate în timpul transmisiei cât timp valoarea este pozitivă. Mecanismul promovează principiul best effort traffic, astfel încât atunci când o transmisie epuizează creditele de trimitere, pachetele de date sunt livrate unul după altul în funcție de accesul la resursele necesare. Dacă acest proces întârzie transmiterea pachetelor cu prioritate mai mare, valoarea creditelor de trimitere pentru acele pachete este mărită, urmând ca acestea să fie livrate imediat ce traficul o va permite. Un alt mecanism responsabil pentru creșterea eficienței transmiterii datelor este Time-Aware Scheduler (IEEE 802.1Qav). Similar mecanismului de tip Scheduler în contextul aplicațiilor cu cerințe de funcționare în timp real, Time-Aware Scheduler creează cicluri de timp discret în care clasele de trafic, în funcție de prioritatea lor, pot livra datele cu precizie la momentele de timp dorite. Datorită acestui mecanism de planificare, transmiterea și recepționarea fluxurilor de date pot fi predictibile pentru diverse tipuri de mesaje, sincronizarea transmisiilor multiple este mai ușoară, iar toate entitățile conectate la rețea sunt informate cu privire la prioritatea mesajelor, fiind clar momentul de transmitere și momentul de recepție și procesare a mesajului din perspectiva fiecărui dispozitiv conectat. Pentru aplicațiile cu cerințe de funcționare în timp real implicate în cazuri de utilizare stricte (de exemplu aplicațiile de siguranță sau de conducere autonomă), tehnologia TSN oferă flexibilitate prin adaptarea diferitelor standarde la scenarii concrete, nefiind neapărat nevoie de toate mecanismele cât timp rețeaua poate îndeplini necesitățile. Astfel rețeaua de tip TSN ar putea fi configurată în concordanță cu așteptările legate de eficiență și costuri, pentru fiecare tip de aplicații.

2.2.2. OPC UA în concordanță cu TSN

Odată cu dezvoltarea IIoT în ultimii ani, nevoia schimbului de date la un ritm mai ridicat, cu multipli participanți la procesul de comunicare, a generat scenarii mai complexe și cerințe mai exigente pentru aplicațiile din domeniu. Era de așteptat ca protocoalele de comunicare de nivel înalt să extindă posibilitățile și să se adapteze și să interacționeze cu alte diferite tehnologii, chiar și la nivelele de bază ale modelului OSI. Procesele de producție, infrastructura existentă și liniile de producție vor trebui de asemenea să evolueze, obiectivul IIoT fiind creșterea eficienței proceselor existente, vizând flexibilitatea, profitabilitatea și întreținerea continuă a viitoarelor produse dezvoltate. Extinderea OPC UA ajunge până la nivelul echipamentelor de teren, al interacțiunii cu arhitecturile cloud, al comunicației între controlere și al integrării cu diferite entități de tip broker de mesaje, oferind soluții accesibile pentru

aplicațiile la scară largă ce implică un volum mare de transmitere de date, algoritmi variați specifici IoT, comportament în timp real, latență scăzută, sincronizare între dispozitive multiple. Astfel, interacțiunea între OPC UA și o alta tehnologie capabilă să îndeplinească astfel de cerințe, precum TSN, reprezintă următorul pas în domeniul automatizărilor.

Utilizarea OPC UA în concordanță cu TSN extinde posibilitățile tehnologiei actuale prin modificarea modelului de automatizare, cu ierarhie între nivelul de întreprindere, management al proceselor și control, într-un nou model în care conectivitatea în interiorul rețelei va fi disponibilă la o scară mai mare, între diferiți participanți, multiple procese de producție fiind astfel realizate și îmbunătățite cu costuri mai mici și flexibilitate adițională. Capacitățile OPC UA și îmbunătățirea infrastructurii de comunicație prin implementarea de standarde suplimentare la nivelele de bază ale modelului OSI (în cazul TSN în principal se face referire la Data Link Layer), reprezintă o abordare complexă, cu aplicabilitate industrială semnificativă.

2.2.3. Mecanismul OPC UA Publish-Subscribe : Noțiuni Introductive

Pentru ca OPC UA să evolueze și să îmbunătățească automatizarea în contextul Industry 4.0, pentru a obține funcționalitatea în timp real și pentru integrarea facilă a echipamentelor hardware provenite de la orice furnizor, îndeplinind cerințele de comunicație rapidă și integrare în infrastructuri cloud, OPC Foundation a conceput mecanismul Publish-Subscribe ca partea a 14-a a specificațiilor OPC UA [24]. Când se face referire la comunicația între două controlere, nevoile sunt reprezentate de latența scăzută într-un design al transmisiei de tip one-to-one. Când se face referire la interacțiunea cu structuri de tip cloud, designul transmisiei devine de tip one-to-many sau many-to-many, fiind nevoie de o rețea cu capacitate de comunicație mare. În momentul combinării la scară largă a diverselor tipuri de tipare de comunicație și dispozitive cu caracteristici variate, mecanismul Publish-Subscribe se dovedește a fi potrivit din punct de vedere al flexibilității, performanței, iar în funcție de tipul de transport selectat, mecanismul este permisiv în integrarea dispozitivelor cu capacități diferite, în același context. Conform [24], mecanismul Publish-Subscribe poate utiliza protocoale de transport diferite în funcție de cum sunt structurate datele livrate și în funcție de capacitatea rețelei. Pentru transmisii frecvente și volum redus de informații, transportul UDP cu mesaje codificate binar reprezintă o soluție bună în transmisiile de tip one-to-one și one-to-many. Pentru integrarea cu structurile cloud și aplicațiile de analiză, sunt preferate protocoalele standard de mesagerie precum MQTT și AMQP. Pe baza flexibilității și integrării facile a dispozitivelor, cerințele de funcționare în timp real necesită a fi rezolvate, iar tehnologia TSN oferă posibilitatea de a atinge o calitate maximă a serviciilor în ceea ce privește sincronizarea, viteza, predictibilitatea și lățimea de bandă, prin implementarea standardelor specifice nivelului Data Link Layer. Astfel, capabilitățile OPC UA se extind din punct de vedere al scalabilității și eficacității. În contextul relației dintre conceptele, specificațiile și protocoalele abordate în acest studiu Figura 1 prezintă corelația dintre nivelele modelului OSI în cazul OPC UA și VSOME/IP.

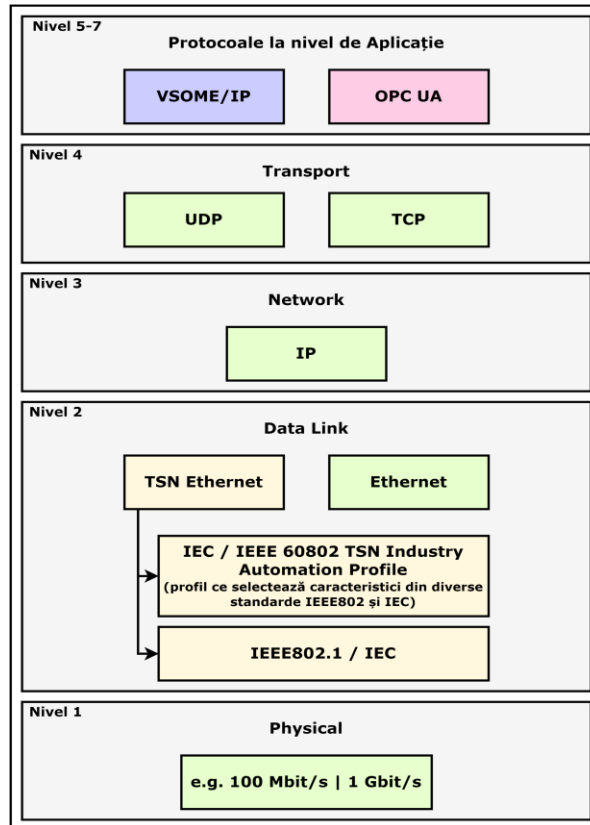


Figura 1. Protocoalele OPC UA și VSOME/IP în relație cu UDP, TSN, Raw Ethernet, în contextul modelului OSI

2.2.4. OPC UA Publish-Subscribe: Abordare și Aprofundare

Pentru implementarea unor entități de tip Publisher și Subscriber, primul pas este crearea unei conexiuni pentru cele 2 elemente. Conform [24] (vezi detalii în capitolul 6), mecanismul Publish-Subscribe definește diferiți parametri de configurare pentru diferite componente. Pe baza acestei configurații Publisher-ul și Subscriber-ul vor dobândi un comportament specific. Conexiunea Publish-Subscribe se ocupă de detalii precum protocolul de transport utilizat în schimbul de date. Alte subcomponente ale mecanismului Publish-Subscribe sunt de tipul WriterGroup și ReaderGroup, fiecare conexiune având posibilitatea de a avea multiple subcomponente de acest fel. Subcomponentele WriterGroup și ReaderGroup conțin detalii necesare pentru crearea de DataSetMessages. La rândul lor, DataSetMessages aparțin altor subcomponente numite DataSetWriter în cazul WriterGroup, respective DataSetReader în cazul ReaderGroup. DataSetWriter este entitatea responsabilă cu crearea de DataSetMessages. Pe baza tipului de informații ce necesită a fi stocat, un DataSetMessage poate avea mai multe câmpuri. După crearea subcomponentului DataSetWriter, acesta este legat de un singur subcomponent de tipul

PublishedDataSet. Un PublishedDataSet poate fi perceput ca materializarea tuturor parametrilor necesari ce descriu conținutul unui mesaj. Cu toate acestea, PublishedDataSet poate conține mai multe subcomponente de tipul DataSetWriter, relația dintre ele nefiind echivalentă.

Într-un context în care Publisher-ul trimite date, iar Subscriber-ul dorește să le acceseze, subcomponentul DataSetReader va face parte din aplicația de tip Subscriber. Subscriber-ul recepționează mesaje de rețea împreună cu detalii specifice protocolului, bazate pe tipul de transport utilizat și pe particularități specifice din punct de vedere al rețelei. Pentru obținerea payload-ului, a datelor dorite, mesajele de rețea recepționate necesită să fie filtrate pe baza configurațiilor utilizate de către Publisher înaintea expedierii mesajului. Această responsabilitate este atribuită subcomponentului DataSetReader, care extrage informațiile utile din mesajul de rețea, iar la acel moment payload-ul poate fi accesat printr-un proces de browsing (răsfoire) a tuturor câmpurilor mesajului. Noțiunea de mesaje de rețea nu trebuie confundată cu DataSetMessage, cel din urma însemnând payload-ul. Mesajele de rețea conțin într-adevăr payload-ul sub forma noțiunii DataSetMessage, furnizat de către Publisher, însă sunt specifice ca noțiune legăturii dintre Publisher și Subscriber și pot fi diferite în ceea ce privește codificarea și protocolul de transport utilizat. Subscriber-ul recepționează mesaje de rețea, iar prin intermediul filtrelor definite și prin configurația subcomponentului DataSetReader, datele dorite sunt extrase. În cazul protocolelor de transport MQTT (Message Queuing Telemetry Transport) și AMQP (Advanced Message Queuing Protocol), comunicarea se realizează printr-o entitate intermediară denumită broker, iar în cazul utilizării UDP, comunicația se realizează printr-o adresă multicast. SDK (Software Development Kit) open62541 [25] a fost utilizat în acest studio pentru implementarea tuturor entităților OPC UA, iar versiunea Genivi a protocolului SOME/IP (cunoscută ca VSOME/IP) a fost utilizată pentru implementarea tuturor entităților VSOME/IP.

2.2.5. Contextul funcționării în timp real

Conceptul de timp real este utilizat des în diferite domenii tehnice pentru a descrie în general execuția instrucțiunilor/operațiilor la un anumit moment de timp, pe baza unei anumite necesități. Totuși, conceptul de timp real trebuie descris în profunzime mai ales în contextul protocolelor de comunicare și al sincronizării schimburilor de date. Sincronizarea operațiilor și recurența sarcinilor fac parte dintr-un sistem în timp real, iar aplicațiile specifice controlerelor implementează aceste concepte, însă în contextul protocolelor de comunicare, termenul ar trebui să implice și capacități specifice. Una dintre aceste capacități vizează calitatea serviciului (QoS – Quality of Service) și posibilitatea de a asigura utilizatorul că intervalele de timp sunt gestionate eficient, procedurile de transmisie și recepție oferind garanția de a se executa în intervalele de timp dorite, făcându-se referire inclusiv la posibilitățile protocolului folosit. Așadar trebuie plasate în același context atât specificații ce se raportează la modul în care procedurile trebuie implementate cât și specificații clare în legătură cu intervalele de timp în care este necesar ca procedurile să fie executate. În baza acestui context larg ce se focalizează pe partea de specificații, utilizatorul trebuie să fie asigurat că mecanismele de monitorizare ale

aspectelor descrise în specificații sunt active și capabile să reacționeze și să supravegheze permanent toate procedurile aflate în derulare. O a doua capabilitate ce merită menționată în contextul funcționării în timp real, face referire la predictibilitatea sistemului și identificarea cazurilor de eșec. Pe baza specificațiilor, pentru ca un protocol de comunicare să poată fi categorisit ca fiind orientat spre funcționarea în timp real, acesta trebuie să identifice cazurile de eșec atât din perspectiva timpului de execuție al sarcinilor, cât și din perspectiva volumului de date distribuit, ținând cont atât de interacțiunea între nivelele tehnologice diferite (de exemplu, la care nivel al modelului OSI poate apărea un eventual impediment), cât și de particularitățile protocolelor implicate (de exemplu, în cazul OPC UA, designul Publish-Subscribe facilitează funcționarea în timp real cât timp nu se folosește un protocol de transport ce implică un broker). A treia capabilitate importantă în cazul unui protocol cu potențial de funcționare în timp real, ar trebui să implice mecanisme de siguranță pentru diferite scenarii și cazuri de eșec. Orice sistem poate eșua în cele din urmă, chiar dacă așteptările în acest sens sunt minime. Însă atunci când o defecțiune poate să apară, în contextul comportamentului predictibil al procesului vizat, utilizatorul trebuie să aibă acces la mecanisme de siguranță răspunzătoare pentru: reducerea daunelor, salvarea stării sistemului din momentul precedent eșecului, identificarea și raportarea posibilelor cauze ale eșecului, pașii necesari pentru ameliorarea și evitarea situațiilor de eșec.

În industrie și producție, însușirea acestor capabilități pentru sistemele utilizate se transpune în menținerea costului de întreținere și îmbunătățire la un nivel scăzut, timpul de nefuncționare va scădea semnificativ, evoluția arhitecturilor existente se va produce cu o mai bună conștientizare, calculându-se mai precis riscurile și costurile. În domeniul automotive, cerințele de funcționare în timp real sunt determinante pentru majoritatea operațiunilor, impactând diverse nivele ale standardului AUTOSAR. Pe baza unor astfel de cerințe, aplicații de toate felurile rulează pe ECU-uri asigurând o sincronizare de înaltă precizie între sarcinile sistemului și dispozitivele hardware implicate. Sunt specifice domeniului: comunicația între controlere pe diverse protocoale și magistrale, interacțiunile între componente cu funcționalități diferite, procesarea datelor provenite de la o varietate de senzori și echipamente hardware cu diverse capabilități. Strategiile și soluțiile adoptate în ceea ce privește structurarea cerințelor de funcționare în timp real au atins un nivel detaliat în domeniul automotive existând de-a lungul timpului diverse stadii de evoluție atât în privința procesului de standardizare, cât și în privința dezvoltării hardware. Platforma Adaptive care se focusează (printre altele) pe identificarea de soluții pentru procesul de conducere autonomă, cazurile de utilizare ce implică: nivele de siguranță, procese de tipul Over The Air Updates, transmisii de date la eficiență înaltă pe Ethernet, a facilitat evoluția cerințelor ce vizează comportamentul predictibil și sincronizarea dispozitivelor în arhitecturile orientate pe servicii. Conform sursei [26] cerințele privind detectarea defecțiunilor sunt descrise cu posibilități de identificare a erorilor și resincronizare a bazelor de timp, confirmând necesitatea mecanismelor de siguranță pentru diverse cazuri de eșec în contextul aplicațiilor Adaptive. Pe lângă importanța detectării anomaliilor, sunt necesare și subsisteme dinamice responsabile pentru măsuri concrete în cazurile de desincronizare. Pe baza aceluiași document, importanța definirii referinței asupra timpului este explicată în contextul sarcinilor de execuție și sincronizare, la nivel de component software cât și la nivel de dispozitiv

ECU. În implementarea studiului actual, sincronizarea operațiunilor de transmitere și receptive de date aparținând aplicației Gateway s-a realizat ținând cont de principiile menționate anterior existând o focalizare atât pe prioritizarea unei sarcini de execuție în funcție de altă sarcină (de exemplu transmiterea se va executa mereu după recepție), cât și pe sincronizarea sarcinilor de execuție între dispozitivele utilizate, fiecare entitate având o referință proprie asupra timpului (fiecare entitate raportându-se la propriul Clock). Scopul final al actualei abordări a fost îndeplinirea cerințelor funcționării în timp real pe baza principiilor industriale, pe cât posibil.

Mecanismul OPC UA Publish-Subscribe a fost conceput în contextul necesităților de funcționare în timp real, punându-se accent pe comunicarea între echipamente cu resurse limitate și interacțiunea cu arhitecturi cloud. Capacitățile de publicare rapidă ale mecanismului au fost confirmate în studii precum [27]. În studiul curent, procesul de recepționare a datelor de către aplicația Gateway a urmărit respectarea unui comportament în timp real la nivel de aplicație, fiind folosit în cadrul unui schimb periodic de date cu un OPC UA Publisher ce transmite ciclic mesajele de rețea, obiectivul final al aplicației fiind intermedierea comunicației între o entitate OPC UA (de tip Publisher) și o entitate VSOME/IP (de tip Subscriber), fără a se pierde nici un ciclu de transmisie.

2.3. Arhitectura Sistemului

Considerând conceptul inovator de interoperabilitate între două protocoale complexe de nivel înalt și obiectivul dezvoltării unei aplicații Gateway performante, neexistând alte implementări ce pun în același context tehnologiile OPC UA și VSOME/IP, actuala cercetare a fost structurată din punct de vedere practic sub forma a 4 studii de caz. Fiecare studiu de caz conține particularități din punct de vedere al paradigmatelor folosite, al scenariilor vizate și al componentelor hardware și software necesare. Având în vedere aceste diferențele semnificative, a fost definită o arhitectură generală a conceptului de interfațare pe baza aplicației de tip Gateway ce poate fi observată în Figura 2.

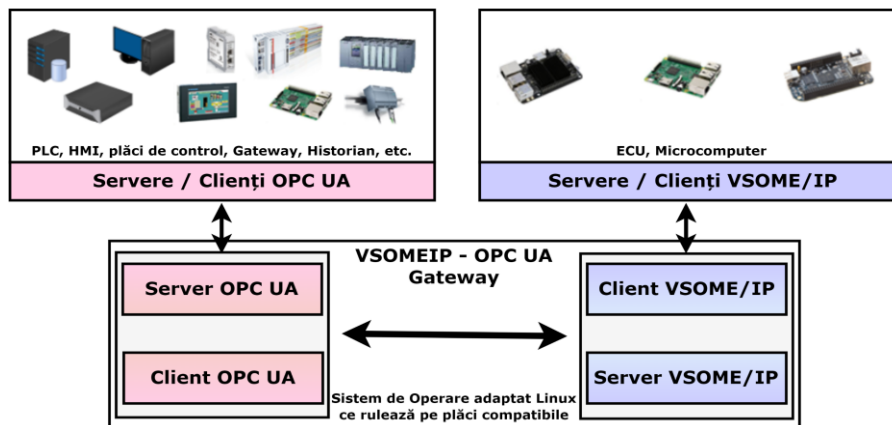


Figura 2: Arhitectura generală a conceptului VSOME/IP-OPC UA Gateway

Deși diferențele între cazurile de studiu presupun modificări în legătură cu entitățile dezvoltate, se poate defini o structură arhitecturală de bază a sistemului, validă în toate aceste cazuri cu mici ajustări în funcție de scenariul vizat. Această arhitectură se bazează pe 5 entități cu roluri diferite, după cum urmează:

1. Un Server/Publisher OPC UA ce stochează datele necesare pentru a fi transmise și reprezintă punctul de pornire al schimbului de date.
2. Un Client OPC UA (în cazul în care entitatea 1 este de tip Server), ce modifică datele stocate cu o recurență configurabilă.
3. Aplicația Gateway – având un subcomponent OPC UA Client/Subscriber responsabil de recepționarea datelor provenite de la furnizorul OPC UA.
4. Aplicația Gateway – având un subcomponent sub forma unui serviciu VSOME/IP cu rolul de Server/Notifier, responsabil pentru trimiterea datelor mai departe către receptorul final.
5. Un Client/Subscriber VSOME/IP cu rolul de receptor final al datelor.

Obiectivul principal al prototipului dezvoltat este de a face posibilă comunicarea OTA (Over The Air) de la furnizorul OPC UA prin intermediul aplicației Gateway, către receptorul VSOME/IP, atât în contextul paradigmei de tip Server-Client (în cazul OPC UA) respectiv Request-Response (în cazul VSOME/IP) cât și în contextul paradigmei orientate către funcționarea în timp real Publish-Subscribe (în cazul OPC UA) respectiv Notify-Subscribe (în cazul VSOME/IP). Fiecare entitate dezvoltată poate rula pe un dispozitiv bazat pe Linux, conectat la rețea. Deși funcționarea entităților a fost verificată și sub forma unor procese ce rulează și schimbau date în cadrul aceluiași dispozitiv, în general au fost utilizate 3 dispozitive diferite fiecare având fiecare un rol separat, după cum se poate observa în Figura 3.

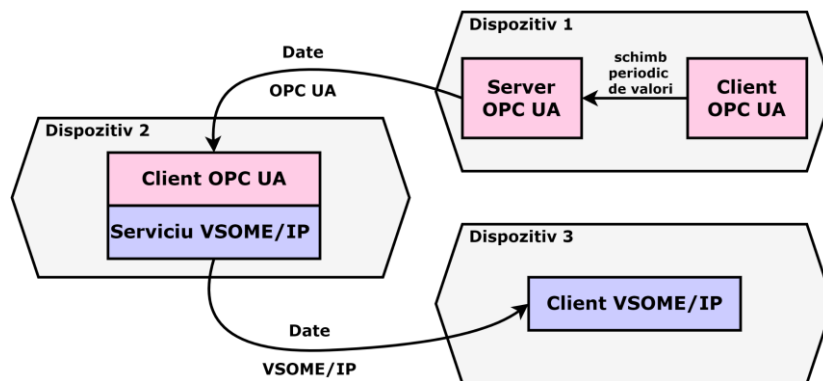


Figura 3. Arhitectura Hardware în raport cu entitățile dezvoltate

În cazul utilizării mecanismului OPC UA Publish-Subscribe, cele 2 entități prezente pe primul dispozitiv sunt înlocuite de o singură entitate de tipul OPC UA Publisher, iar celelalte entități sunt înlocuite de varianta cu rol echivalent specifică paradigmei utilizate.

2.4. Studii de caz

Deși lucrarea actuală extinde în mare măsură ideea de gateway, prin intermediul a numeroase concepte și dezvoltări variate în special în cazul protocolului OPC UA, scenariul principal urmărit în toate studiile de caz, cu implementările aferente, se bazează pe faptul că VSOME/IP este tehnologia de bază folosită pentru comunicația pe Ethernet în interiorul vehiculului, iar OPC UA este soluția selectată pentru a susține infrastructura inteligentă (în cazul de față un semafor). Această comunicație de tipul Car-to-Infrastructure este evidențiată în Figura 4, obiectivul fiind dezvoltarea unui sistem rapid de comunicație, ce ar putea realiza schimbul de date la diverse intervale de timp.

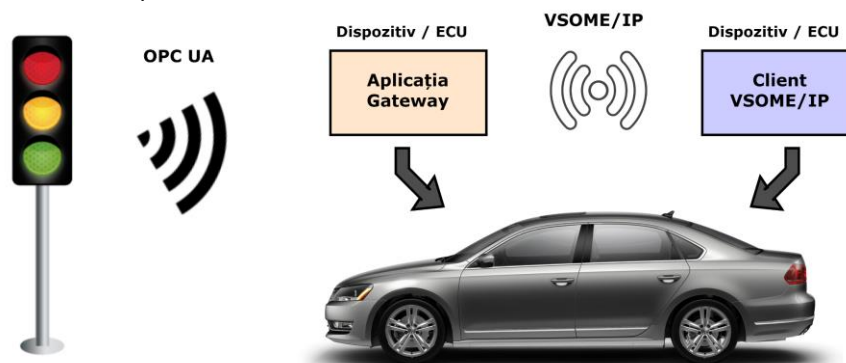


Figura 4. Comunicația Car-to-Infrastructure raportată la studiul actual

2.4.1. Studiu de caz 1

Primul studiu de caz reprezintă confirmarea compatibilității celor două protocoale utilizate, precum și să evidențieze capabilitățile de interacționare cu structuri industriale deja validate. Urmărind designul arhitectural și adoptând paradigmele clasice de tipul Server/Client respectiv Request-Response, entitatea OPC UA este considerată furnizorul datelor de interes, fiind asociată unui proces industrial (de exemplu un semafor inteligent). Aplicația Gateway facilitează recepționarea datelor prin subcomponentul de tip OPC UA Client și le transmite subcomponentului VSOME/IP Server pentru pregătirea livrării către receptorul final VSOME/IP Client. Această transmisie este structurată sub forma a 5 pași, după cum urmează:

- Pasul 1. Serverul OPC UA transmite datele (mesajul "0xaa") către aplicația Gateway.
- Pasul 2. Aplicația Gateway transferă mesajul mai departe către receptorul final.
- Pasul 3. Clientul VSOME/IP recepționează datele de interes și pregătește un mesaj de confirmare.
- Pasul 4. Clientul VSOME/IP trimite mesajul de confirmare (mesajul "0xcc") către aplicația Gateway.

Pasul 5. Aplicația Gateway recepționează mesajul de confirmare și îl expediază către Serverul OPC UA.

În cazul acestui parcurs inițial, toate entitățile au fost rulate și testate pe același dispozitiv (o placă Raspberry Pi 3 model B / o mașină virtuală cu sistem de operare Linux), așa cum se poate observa în Figura 5. Secvența de transmisie descrisă poate fi urmărită în Figura 6, fiind confirmată în mod practic compatibilitatea între OPC UA și VSOME/IP obținând o comunicare bidirecțională între entitățile implicate.

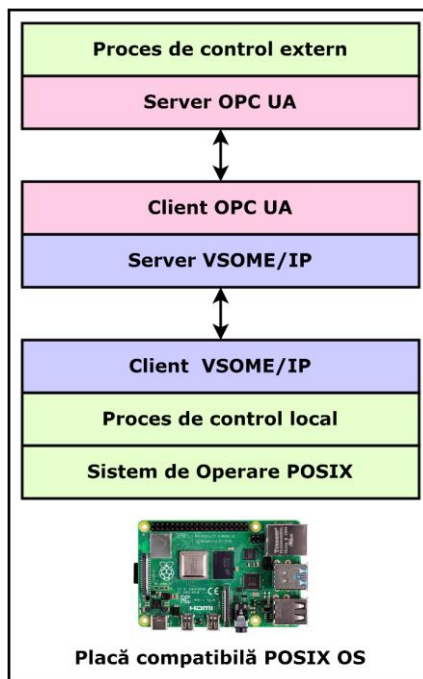


Figura 5. Secvența de transmisie în cadrul plasării tuturor entităților pe același dispozitiv

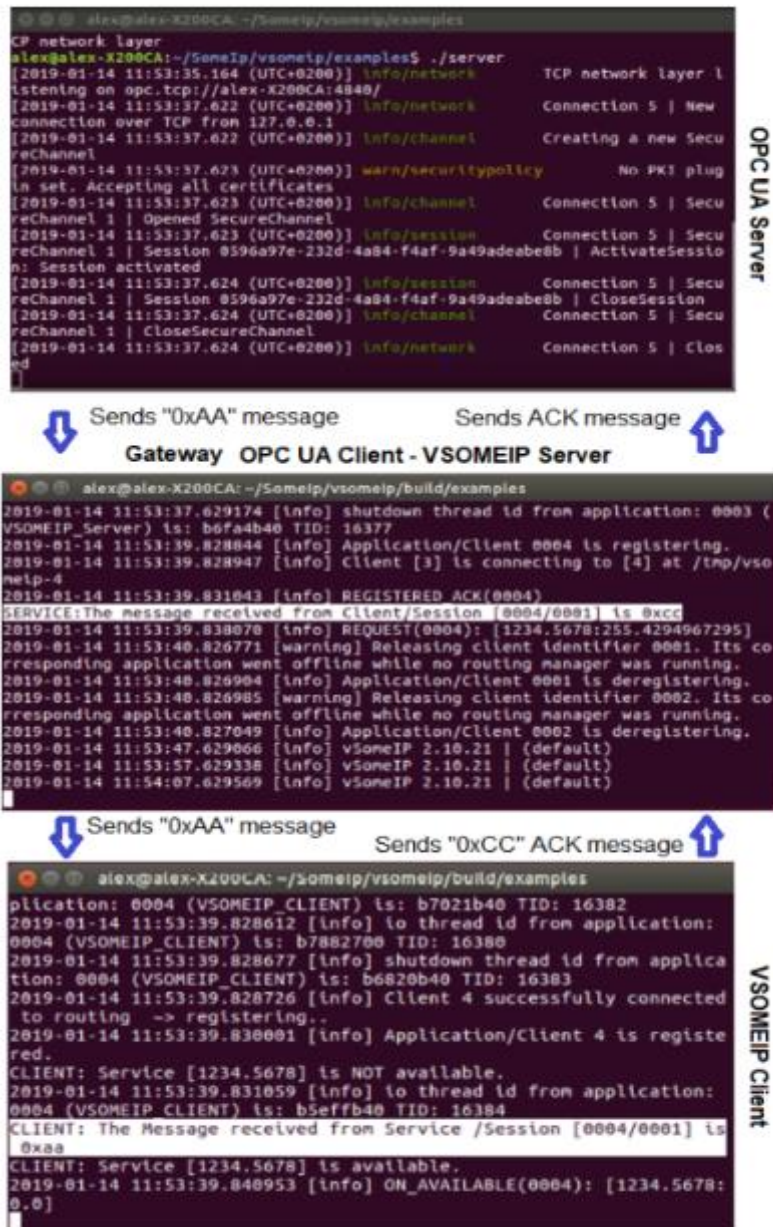


Figura 6. Secvența de transmisie bidirecțională între OPC UA Server, Aplicația Gateway și VSOME/IP Client

A doua fază a primului studiu de caz constituie interacțiunea dintre aplicația Gateway și structuri externe OPC UA validate în scenariile industriale. Prin urmare, scenariul luat în considerare presupune conectarea la un Server OPC UA dintr-o instalație funcțională, asociat unui Server SCADA WinCC Professional v13. În acest fel

au fost accesate două etichete (tag-uri) corespunzătoare unor valori ale energiei electrice din interiorul unei centrale. Aplicația Gateway prin intermediul subcomponentului OPC UA Client reușește să acceseze valorile etichetelor din server, recepționând trei seturi de valori valide la distanțe scurte de timp (fiind expediate apoi către receptorul final), după cum se poate observa în Figura 7.

```
[2019-01-14 22:00:34.596 (UTC+0200)] info/userland 1.We receive from node DB111_MCC_FB1_SA1_ENERGIE value:17.275760
[2019-01-14 22:00:34.818 (UTC+0200)] info/userland 2.We receive from node DB1_HMI_ACREL[1].PEAKS value:55.599998
[2019-01-14 22:00:43.179 (UTC+0200)] info/userland 1.We receive from node DB111_MCC_FB1_SA1_ENERGIE value:17.275778
[2019-01-14 22:00:43.376 (UTC+0200)] info/userland 2.We receive from node DB1_HMI_ACREL[1].PEAKS value:55.599998
[2019-01-14 22:01:39.858 (UTC+0200)] info/userland 1.We receive from node DB111_MCC_FB1_SA1_ENERGIE value:17.275896
[2019-01-14 22:01:40.061 (UTC+0200)] info/userland 2.We receive from node DB1_HMI_ACREL[1].PEAKS value:55.599998
```

Figura 7. Aplicația Gateway recepționează valori valide de la un OPC UA Server extern

Pentru confirmarea eficienței interfațării dintre aplicația dezvoltată și structuri industriale OPC UA, a fost utilizat în paralel un OPC UA Client validat în context industrial (de la compania Softing), în scopul comparării valorilor obținute. Clientul OPC UA Softing accesează cele două etichete menționate în aceeași zi, la intervale de timp apropiate, rezultatele fiind prezentate în Figura 8. După cum se observă, valorile obținute de aplicația Gateway sunt corecte, una dintre acestea fiind ușor mai mică, așa cum era de așteptat în cazul diferenței de aproximativ o ora între cele două citiri.

State	Display Name	Node Id	Data Type	Value	Server Timestamp	Source Timestamp
🟢	...B111_MCC_FB1_SA1_ENERGIE	ns=1;s=1 DB111_MCC_FB1_SA1_ENERGIE	Double	17.28651312...	23:27:46.186	23:27:45.855
🟢	...sDB1_HMI_ACREL[1].PEAKS	ns=1;s=1 DB1_HMI_ACREL[1].PEAKS	Double	55.59999847...	23:24:16.786	19:48:05.120

Previous timestamp		
Name	Value	Type
DB111_MCC_FB1_SA1_ENERGIE	17.28580510560	Double
Value	17.28580510560	Double
StatusCode	Good	StatusCode

Value		
Name	Value	Type
DB1_HMI_ACREL[1].PEAKS	55.599998474121	Double
Value	55.599998474121	Double
StatusCode	Good	StatusCode

Figura 8. Clientul OPC UA Softing recepționează valori de la același OPC UA Server extern

Versiunea dezvoltată în primul studiu de caz îndeplinește obiectivele propuse, oferind confirmările necesare despre compatibilitatea, eficiența și flexibilitatea tehnologiilor folosite. Toate modulele implementate sunt funcționale, soluția fiind extinsă și îmbunătățită în următoarele studii de caz.

2.4.2. Studiu de caz 2

Plecând de la o abordare mai detaliată asupra scenariului de comunicație Car-to-Infrastructure, studiul de caz 2 se focalizează pe corelarea datelor trimise, cu culorile și coordonatele unui semafor inteligent, structurând mai concret mesajele distribuite și automatizând procesul de comunicare păstrând paradigma OPC UA Client-Server și adoptând paradigma Notify-Subscribe specifică VSOME/IP.

Serverul OPC UA conține trei noduri interne, unul dintre acestea fiind alocat pentru stocarea unui cod corespondent culorii semaforului, iar celelalte două noduri fiind alocate pentru valorile de latitudine și longitudine. Codul culorii este schimbat recurent de un Client OPC UA ce funcționează în mod automat pe baza unui interval de timp configurabil, rulând pe același dispozitiv cu Serverul OPC UA. (Vezi Figura 3). Această funcționalitate asemănătoare cu modul de funcționare al unui semafor real, poate fi testată prin achiziția datelor corespondente culorii, la intervale de timp diferite, constatându-se astfel o alternare autentică între codurile de culoare stocate pe Serverul OPC UA. Prezența coordonatelor presupusului semafor oferă o posibilă distincție și localizare în scenariul unei intersecții cu mai multe semafoare, unde ar fi nevoie de identificarea din timp a semaforului în cazul unui proces de conducere autonomă. Conexiunea între entitățile OPC UA se realizează prin adresele IP ale dispozitivelor conectate în aceeași rețea, asociate unor Hostname-uri, sau prin adresele IP ale dispozitivelor de tip Router (care mai apoi livrează datele prin portul configurat către entitățile receptoare) în cazul plasării receptorilor în rețele diferite față de furnizori. Această caracteristică a sistemului ar permite funcționarea în cazul plasării autovehiculului și infrastructurii în rețele diferite, acest lucru fiind foarte probabil într-un scenariu real.

Din punct de vedere al implementării Serverului OPC UA, primul pas constă în definirea atributelor necesare pentru stocarea datelor folosind tipul de dată `UA_VariableAttributes`. După ce se configurează câte un set de parametrii (descriere, tip de dată, nivel de acces) pentru fiecare atribut, o valoare hexazecimală corespondentă tipului de informație stocat (culoare, longitudine, latitudine) este asignată folosind metoda `UA_Variant_setScalar`. Odată ce atributele sunt configurate, iar datele au fost asignate, următorul pas presupune declararea, configurarea și introducerea nodurilor în modelul informațional folosind metoda `UA_Server_addVariableNode`. În continuare, serverul poate rula, iar informația devine disponibilă pentru clienții interesați.

Următorul element al lanțului de comunicație este aplicația Gateway ce recepționează datele de la Serverul OPC UA și le publică către VSOME/IP Subscriber. Similar cu cazul de studiu 1 (respectiv [23]), structura aplicației Gateway este bazată pe două subcomponente cu roluri diferite. Un subcomponent de tip OPC UA Client responsabil de recepționarea datelor de la server și un alt subcomponent, de tip VSOME/IP Notifier ce transmite datele la recurențe de timp configurabile într-un mod automat către receptorul final. Detaliile de rețea necesare aplicației pentru facilitarea transmisiunii între OPC UA și VSOME/IP sunt depozitate într-un fișier de configurare de tip json, fișier ce este pasat către aplicație prin intermediul unor variabile de mediu specifice protocolului VSOME/IP, similar metodologiei industriale. Implementarea clientului OPC UA începe prin inițializarea unei conexiuni cu serverul. Odată ce conexiunea este realizată, se creează o cerere de navigare (răsfoire), pe baza căreia se obține conținutul de interes de la furnizor, utilizându-se `UA_Client_Service_browse`. Procesele de recepție și transmisie ale subcomponentelor aplicației sunt sincronizate pe baza unui timer comun, astfel fiind obținut un comportament predictibil în ceea ce privește gestionarea datelor. Existând această sincronizare, transmisiile ciclice efectuate de către aplicația Gateway pentru VSOME/IP Subscriber conțin mereu date valide, asigurând eficiența maximă chiar și în cazul funcționării în timp real.

Mecanismul browse request – browse response (`UA_BrowseRequest` and `UA_BrowseResponse`) facilitează filtrarea informației din diferite noduri, în funcție de tipul nodului și de structura datelor. Această abilitate permite viitoarelor arhitecturi

de servere OPC UA complexe să stocheze volume mari de date fără a exista pericolul amestecării informațiilor, oferind un grad ridicat de flexibilitate. În ciuda acestui fapt, mecanismul Publish-Subscribe reprezintă o soluție superioară atât din punct de vedere al filtrării datelor cât și al integrării în cazuri de utilizare cu cerințe restrictive de timp real.

În perspectiva entităților OPC UA Client, datele de interes sunt accesibile atât subcomponentului aplicației Gateway cât și clientului responsabil de modificarea valorii asociate culorii semaforului, prin accesări individuale la diverse momente de timp. În cazul mecanismului Publish-Subscribe, informația de interes este furnizată ciclic, procesul de filtrare fiind efectuat la nivelul entității Subscriber, interogarea rețelei producându-se în mod automat, pe baza unei recurențe preconfigurate. În cazul curent, intervalul de timp la care se produce schimbarea valorii asociate culorii semaforului este identic cu recurența de publicare către receptorul final, astfel încât, după fiecare secvență de transmitere completă, valorile recepționate să fie distincte.

Ultimul element al lanțului de comunicație este VSOME/IP Subscriber. Acesta este abonat la datele publicate de către aplicația Gateway, fiind notificat la fiecare schimbare de culoare din interiorul serverului. Acest ultim receptor constituie destinația finală pentru întregul schimb, fiind nevoie de datele recepționate pentru procesele locale de control a diverse module din interiorul vehiculului. Raportat la un scenariu real, valorile latitudinii și longitudinii ar clarifica originea informației, eliminând riscul recepționării de date de la un semafor diferit.

Întregul proces de distribuire a datelor a fost testat la multiple recurențe de timp, între 10 secunde și 1 milisecundă, sistemul reușind să transmită informația cu acuratețe de 100%. Totuși, sustenabilitatea în timp a transmisiilor foarte rapide poate deveni problematică în lipsa tehnologiei TSN, rețeaua nefiind capabilă să garanteze livrarea pachetelor. Factorii ce se pot dovedi nefavorabili în cazul intervalelor rapide de timp sunt: sistemul de operare, rețeaua, volumul de date, resursele hardware. Comportamentul sistemului rămâne neschimbat în cazul plasării OPC UA Server-ului într-o rețea diferită, viteza de transfer și acuratețea rămân de asemenea identice.

2.4.3. Studiu de caz 3

Al treilea studiu de caz implementat pentru actualul concept se focalizează pe scenariul de interacțiune între aplicația Gateway și trei servere OPC UA diferite, datele fiind accesate concomitent (vezi Figura 9). Acest scenariu are scopul de a evidenția potențialul sistemului din punct de vedere al flexibilității și eficacității, oferind posibilitatea integrării într-o arhitectură mult mai amplă. Obiectivele ce aparțin de viteza de transmitere completă și acuratețea datelor livrate rămân valabile, utilizându-se aceleași metode de testare și aceleași intervale de timp ca în studiul de caz anterior. Testarea a fost realizată cu succes atât în cadrul unei singure rețele, cât și în cadrul poziționării dispozitivelor în rețele diferite. Performanțele obținute sunt identice cu cele anterioare confirmând implementarea robustă a tuturor entităților implicate, cât și a arhitecturii definite. Un potențial dezavantaj în cazul actual este reprezentat de conexiunile multiple între aplicație și servere, latența sistemului crescând direct proporțional cu numărul de servere.

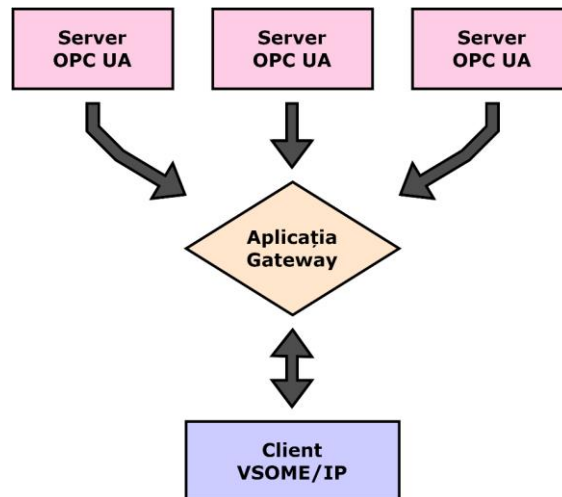


Figura 9. Aplicația Gateway accesând trei OPC UA Servere

Conexiunea dintre servere și clienți necesită un scurt interval de timp, însă în cazul unei secvențe de conexiuni, intervalul de timp crește în funcție de numărul conexiunilor, fapt ce poate reprezenta o latență pentru transmisiile ciclice către VSOME/IP Subscriber, acestea având o recurență fixă. În configurația actuală, pentru a evita potențiale desincronizări în acest caz, în momentul configurării recurenței ciclice, trebuie luat în considerare un interval adițional de timp, variabil în funcție de numărul de conexiuni respectiv de servere. De asemenea, procesul de navigare (răsfoire, browse) în spațiul de adrese al serverelor până la identificarea datelor necesare poate produce întâzieri în funcție de complexitatea structurii interne a fiecărui server. Toate aceste potențiale cauze se adaugă factorilor nefavorabili menționați în studiul de caz anterior, aplicabilitatea într-o situație cu cerințe de funcționare în timp real fiind incertă.

Studiul de caz 3 a evidențiat fezabilitatea conceptului din punct de vedere al flexibilității și eficienței, însă au fost identificate și situații adiționale de risc în privința îndeplinirii nevoilor de comportament predictibil, în intervale exacte de timp real. Numărul conexiunilor, structura internă a serverelor și timpul necesar autentificării dintre aplicația Gateway și furnizorii datelor, evidențiază limitările paradigmatelor de tipul Server-Client în ceea ce privește sistemele rapide de comunicații, fiind necesară o perfecționare a capacităților de funcționare în timp real.

2.4.4. Studiu de caz 4

Implementarea studiului de caz 4 se bazează pe adoptarea paradigmei OPC UA Publish-Subscribe împreună cu paradigma similară VSOME/IP Notify-Subscribe, infrastructura inteligentă (în cazul de față semaforul) fiind reprezentată de un element de tipul OPC UA Publisher. Acest Publisher comunică cu aplicația Gateway printr-un subcomponent complementar de tip OPC UA Subscriber, respectând structura aplicației bazată pe două subcomponente. Odată cu trecerea la paradigme orientate către distribuirea de date în mod automat la recurențe de timp configurabile, sistemul

evoluează semnificativ spre îndeplinirea necesităților de funcționare în timp real, la nivel de aplicație. Integrarea celor două paradigme se face mai ușor decât în studiile de caz 2 și 3, nefiind nevoie de o entitate adițională pentru adaptarea la momentul potrivit a valorilor stocate de furnizor.

Pentru studiul actual, principala provocare a fost reprezentată de implementarea entităților OPC UA Publisher și Subscriber, capabile să comunice prin intermediul unei adrese multicast, distribuind între ele un DataSet alcătuit pe baza datelor de interes. Odată ce componentele Publisher-ului sunt configurate corespunzător, informația devine accesibilă pentru aplicația Gateway prin subcomponentul OPC UA Subscriber, ce mai apoi pregătește datele pentru livrare pe VSOME/IP către receptorul final. Receptorul VSOME/IP Subscriber așteaptă în mod constant să fie notificat de aplicația Gateway în legătură cu datele actualizate. Rata de livrare mare, acuratețea și multiplele filtrări ale datelor la diferite stadii ale transmisiei, reprezintă mecanisme cu potențial mare pentru un caz de utilizare cu cerințe de funcționare în timp real, specific domeniului automotive și IIoT. De exemplu, în cazul unei aplicații ce procesează imagini, utilizarea unei soluții ce permite procesarea și transmiterea informațiilor provenite de la o camera din trafic (aparținând conceptului de infrastructură inteligentă), către autovehicul, prin intermediul a multiple protocoale ce oferă mecanisme de sincronizare ar reprezenta un avantaj major din punct de vedere al performanței și fiabilității. Aplicația Gateway constituie o soluție ce utilizează cele mai eficiente metode disponibile de transmisie și recepție de date pe Ethernet, fiind conformă cu necesitățile de funcționare în timp real la nivel de aplicație. Soluția actuală rulează pe clasic Ethernet, însă ar trebui să fie compatibilă cu tehnologia TSN, existând posibilitatea îmbunătățirii performanței, garantând cicluri mai precise și mai rapide de transmisie, devenind un sistem în timp real atât la nivel de aplicație cât și la nivel de rețea.

Primul pas de implementare a fost crearea și configurarea tuturor entităților descrise anterior. Conexiunea Publish-Subscribe a fost efectuată pe baza specificațiilor [24], pentru ambele părți implicate. Odată ce parametrii necesari au fost configurați, conexiunea este atașată serverului prin metoda `UA_Server_addPubSubConnection`. Conexiunea Publish-Subscribe este configurată atât pentru Publisher cât și pentru Subscriber (subcomponent al aplicației Gateway), doar la începutul aplicației. Fiind o noțiune deterministică în procesul de comunicație, odată ce parametrii sunt configurați, doar în cazul necesității reconfigurării conexiunea ar trebui modificată.

Elementul de tip `WriterGroup` aparținând de Publisher, este creat și atașat conexiunii prin intermediul metodei `UA_PubSubConnection_addWriterGroup`. Conform specificațiilor OPC UA [24] (capitolul 5.2 DataSet), elementul DataSet poate conține variabile, evenimente, marcaje de timp, și poate fi configurat să conțină mai multe câmpuri de informații. În contextul actual, payload-ul de date va fi constituit din variabilele corespondente semaforului, la nivelul DataSet-ului publicat.

La nivelul protocolului de transport, UDP este opțiunea preferată în acest caz. Subscriber-ul va recepționa mesaje de rețea și va aplica filtre pentru extragerea datelor relevante. Odată ce datele sunt extrase, funcționalitatea serviciului VSOME/IP al aplicației Gateway le va prelua, pregătind mesajul ce urmează a fi trimis către VSOME/IP Subscriber pentru etapa finală de procesare a datelor.

42 | 2. OPC UA și SOME/IP Gateway în contextul comunicației Car-to-Infrastructure

Din perspectiva subcomponentului OPC UA Subscriber, odată ce conexiunea Publish-Subscribe este finalizată, aplicația Gateway va sincroniza procedura de recepție a mesajelor OPC UA cu procedura de transmisie a mesajelor VSOME/IP. În Figura 10, secvența necesară configurării tuturor elementelor esențiale paradigmei OPC UA Publish-Subscribe, poate fi urmărită, laolaltă cu toți pașii precedenți și ulterioari transmisiunii.

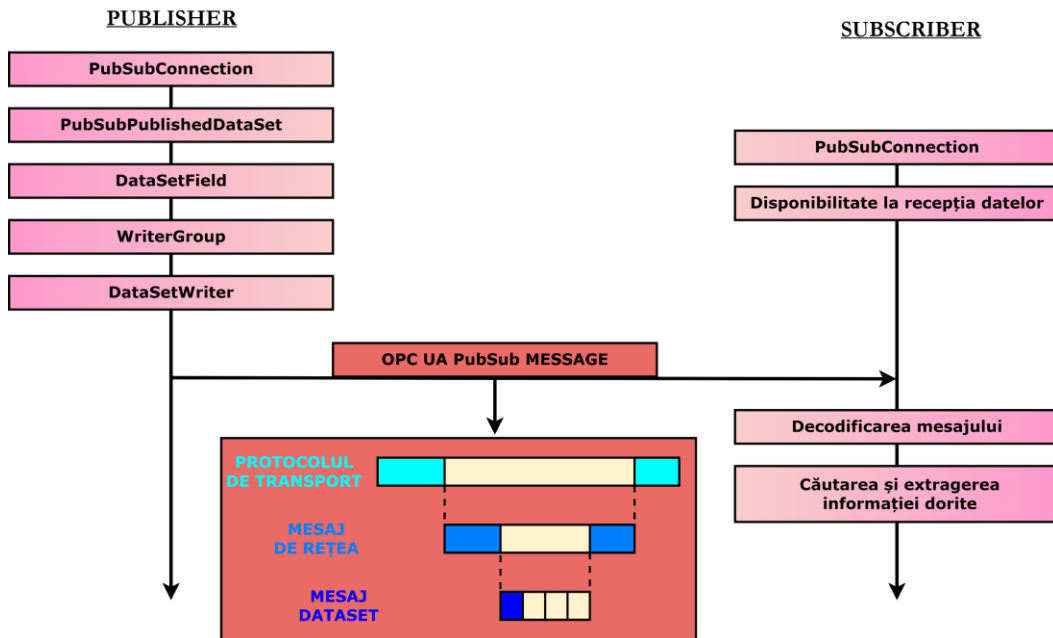


Figura 10. Secvența configurării mecanismului OPC UA Publish-Subscribe

În cazul utilizării protocoalelor de comunicare ce implementează paradigme de tipul Publish-Subscribe, este nevoie de o corelație între intervalele de publicare și de recepție, împreună cu o coordonare a celor două subcomponente ale aplicației Gateway (a se vedea Figura 11). Procedura de notificare (publicare) specifică subcomponentului VSOME/IP Notifier este dependentă de mesajul recepționat de către OPC UA Subscriber. În consecință, nici o transmisie către receptorul final nu este inițiată înainte de recepționarea datelor, abia după finalizarea procedurii de recepție fiind posibilă începerea transmisiei finale. După transmisie, secvența de execuție revine la procedura de recepție într-o perioadă de timp mai mică sau egală cu intervalul de publicare al furnizorului.

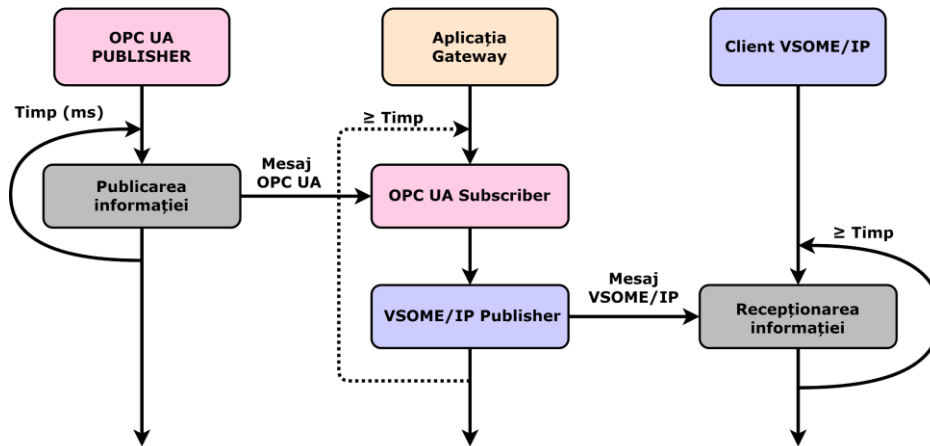


Figura 11. Dependența între subcomponentele aplicației Gateway, OPC UA și VSOME/IP în contextul Publish-Subscribe în timp real

2.5. Rezultate

Pentru fiecare studiu de caz, pe lângă focalizarea pe implementarea tuturor entităților necesare, au fost luate în considerare cerințe concrete în ceea ce privește funcționarea în intervale de timp bine determinate (1 s, 500 ms, 100 ms, 10 ms, 1 ms) și eficiența transmisiei complete (mai exact abilitatea de transmitere și recepție a mesajului nealterat în intervalul de timp vizat). Cel mai rapid interval de timp, vizat în toate studiile de caz a fost de 1 milisecundă. Eficiența așteptată și obținută inclusiv pe parcursul acestui interval minim de timp a fost de 100% din punct de vedere al integrității datelor, sistemul reușind transmisii complete (de la un capăt la altul) la toate intervalele configurate, cu asumarea faptului că sustenabilitatea acestor transmisii rapide scade odată cu micșorarea intervalelor de timp vizate. În cazul în care datele nu ar fi ajuns niciodată nealterate în intervalul de timp dorit, acest interval ar fi fost majorat, nefiind considerat fezabil pentru sistemul actual. Pe baza tuturor criteriilor stabilite, toate studiile de caz s-au dovedit a fi un succes. Prototipul implementat este conceput atât pe baza situațiilor în care schimbul de date se efectuează între dispozitive conectate la aceeași rețea, cât și pe baza situațiilor în care aplicația Gateway și receptorul final VSOME/IP sunt plasate într-o rețea diferită față de furnizorul OPC UA.

În Tabelul 1, rezultatele studiilor de caz pot fi observate și comparate pe baza tuturor criteriilor luate în considerare pe parcursul procesului de dezvoltare a tuturor versiunilor aplicației Gateway și a entităților aferente. Criteriile cu privire la eficiență și capabilități de funcționare în timp real, au fost aplicate pentru toate studiile de caz, însă considerând paradigmele folosite, doar studiul de caz 4 se poate clasa ca o soluție complet orientată spre îndeplinirea cerințelor de funcționare în timp real. Prin această abordare de a utiliza doar mecanisme de tipul Publish-Subscribe, s-a urmărit apropierea de un scenariu industrial cât mai autentic ce ar viza interfațarea Car-to-Infrastructure. Din perspectiva protocolului OPC UA, soluția obținută este conformă cu specificațiile mecanismului Publish-Subscribe, orientat către situațiile cu constrângeri de timp, iar prin adăugarea sincronizării procedurilor de transmisie și recepție pentru subcomponentele aplicației Gateway, se poate afirma că soluția a obținut comportament în timp real la nivel de aplicație.

44 | 2. OPC UA și SOME/IP Gateway în contextul comunicației Car-to-Infrastructură

Tabelul 1. Analiza rezultatelor studiilor de caz

Studiu de caz	Mecanism	Avantaje	Dezavantaje	Eficiența	Recurența de timp
1 & 2	OPC UA Server-Client	Dificultate moderată la implementare	Nu este orientat către funcționarea în timp real	100%	>=1 ms (aplicabil doar în studiul de caz 1)
3	OPC UA Server-Client	Dificultate moderată la implementare	Nu este orientat către funcționarea în timp real, ar putea produce latența suplimentară	100%	>=1 ms
4	OPC UA Publish-Subscribe	<ul style="list-style-type: none"> • Este orientat către funcționarea în timp real • Rapid, sigur și apropiat de tehnologiile folosite în domeniul Automotive • Potențial de a obține transmisii mai rapide cu tehnologia TSN 	<ul style="list-style-type: none"> • Complexitate mare la implementare • Dificultate mare pentru sincronizarea proceselor de pe dispozitive diferite, fără tehnologia TSN 	100%	>=1 ms

Implementarea mecanismului OPC UA Publish-Subscribe în studiul de caz 4 a reprezentat un pas mai complex în comparație cu implementările din studiile de caz 1,2 și 3, același lucru fiind valabil și din perspectiva integrării în context comun cu mecanismul VSOME/IP Notify-Subscribe. Pentru îndeplinirea obiectivelor ce vizează funcționarea în timp real, necesitatea implementării și configurării multitudinii de elemente specifice mecanismelor Publish-Subscribe, și necesitatea sincronizării operațiunilor de recepție și transmisie de date pentru ambele protocoale, reprezintă argumente concrete pentru o dificultate semnificativ mai mare de implementare și testare. Pentru testarea celor patru studii de caz, au fost create și configurate multiple entități adiționale (Servere, Clienți, etc.), fapt ce a presupus o dificultate mai mare pentru testare și extragerea rezultatelor, fiind efectuate numeroase scenarii de testare.

Soluția implementată s-a dovedit a fi eficientă, asigurând distribuirea datelor la diverse intervale de timp, într-un mod eficient și flexibil, urmărind principiile și cerințele industriale. Este de așteptat ca viteza de transmisie să crească în contextul TSN,

paradigma Publish-Subscribe reprezentând un mecanism cheie în selectarea protocoalelor de comunicare pe Ethernet de nivel înalt, pentru dezvoltarea aplicațiilor cu constrângeri de timp și cu cerințe de scalabilitate și flexibilitate hardware.

2.6. Revendicări și Concluzii

Odată cu interacțiunea dintre mecanismul OPC UA Publish-Subscribe și tehnologia TSN în contextul IIoT, noi capacități devin accesibile în domeniul automatizării, noi scenarii devenind posibile. Este de așteptat ca viitoare implementări și soluții să se bazeze pe cele mai recente specificații OPC UA, extinzând posibilitățile de funcționare în timp real și facilitând interacțiuni între diferite arhitecturi industriale. Prin soluția prezentată, s-a obținut o înțelegere aprofundată asupra nevoilor industriale și asupra procesului de selecție a tehnologiilor potrivite pentru îndeplinirea acestor nevoi, fiind dezvoltate noi strategii aplicabile la scară largă. În perspectiva sincronizării schimbului de mesaje pe Ethernet, sunt utilizate mecanisme de actualitate în moduri similare cu abordările industriei pe aceeași temă, soluția fiind bazată pe concepte de înalt interes atât în sectorul automotive cât și în sectorul automatizărilor.

Studiul actual extinde zona de cunoștințe despre interfațarea protocoalelor OPC UA și VSOME/IP în contextul comunicației Car-to-Infrastructure aplicând mecanisme definite recent și integrând soluțiile în cazuri de utilizare de mare interes atât pentru comunitatea științifică cât și pentru industrie. Plecând de la primul studiu de caz (a se vedea [23]), cercetarea actuală a adus îmbunătățiri semnificative conceptului, acoperind o gamă largă de situații posibile. Diferențele dintre studiul de caz inițial și cel final merită menționate cu scopul de a evidenția varietatea de concepte utilizate, evoluția arhitecturii și îmbunătățirea capacităților de funcționare.

- A. Primul studiu de caz poate fi categorisit ca un proof-of-concept, neexistând implementări practice comune pentru OPC UA și VSOME/IP până la acel moment, soluția finală este mai apropiată de cerințele unui caz real prezent în domeniul Automotive și IIoT.
- B. În primul studiu de caz au fost utilizate paradigmele Server-Client pentru OPC UA, respectiv Request-Response pentru VSOME/IP, fiind mai potrivite și mai cunoscute pentru stadiul inițial al conceptului. În ultimul studiu de caz, paradigmele inițiale au fost înlocuite de mecanisme de tipul Publish-Subscribe, implementările fiind inovative pentru comunitatea științifică.
- C. În cazul de studiu inițial transmisia între entități ce rulează pe dispozitive complet separate nu a reprezentat un obiectiv. Soluțiile următoare au explorat posibilitățile de funcționare atât pe dispozitive separate cât și pe rețele separate, confirmând flexibilitatea conceptului atât din punct de vedere software cât și hardware. De asemenea, au fost definite arhitecturi și configurații variate, fiecare având relevanță din punct de vedere industrial.
- D. Primul studiu de caz nu a luat în considerare capacitățile de funcționare în timp real. Soluția finală reușește să funcționeze în timp real la nivel de aplicație, clarificând posibilitățile actuale, și oferă direcții concrete de îmbunătățire, analizând contextul interacțiunii cu tehnologia TSN.

Procesul de interfațare a paradigmelor OPC UA Publish-Subscribe și VSOME/IP Notify-Subscribe reprezintă o soluție valoroasă în contextul comunicației rapide și

46 | 2. OPC UA și SOME/IP Gateway în contextul comunicației Car-to-Infrastructure

eficiente pe Ethernet și are aplicabilitate substanțială considerând viitoarele direcții Industry 4.0. Utilizarea extinsă a ambelor protocoale în domeniile aferente confirmă direcțiile de cercetare urmate ca fiind de mare interes. Compatibilitatea conceptului dezvoltat cu arhitecturile prezente în mediul industrial, certifică flexibilitatea și scalabilitatea, oferind alternative atât din punct de vedere al echipamentelor ce pot fi utilizate, cât și al infrastructurii de implementare. Toate obiectivele acestei etape de cercetare au fost îndeplinite cu succes, soluțiile și analizele obținute contribuie decisiv la formarea unei viziuni clare asupra posibilităților de interoperare atât în contextul Car-to-Infrastructure cât și în cazul viitoarelor aplicații IoT.

3. ÎMBUNĂTĂȚIREA MECANISMULUI OPC UA PUBLISH-SUBSCRIBE ÎN CONTEXTUL EFICIENTIZĂRII COMUNICAȚIEI ÎN TIMP REAL

3.1. Noțiuni generale

Protocoloalele de comunicare evoluează în mod constant, cerințele de interfațare și interoperabilitate reprezentând fundația conceptelor Industry 4.0 și Industrial Internet of Things (IIoT). Protocolul OPC UA este o tehnologie majoră, a fost adoptat masiv în industrie, iar din perspectiva comunității academice, se fac eforturi pentru îmbunătățirea capabilităților sale, astfel încât să poată satisface nevoile în continuă expansiune pentru diverse domenii industriale. Chestiunile ce trebuie soluționate fac referire la cele mai recente specificații și la contextul de funcționare în timp real, fapt ce ar putea extinde aplicabilitatea protocolului aducând beneficii consistente din punct de vedere al vitezei de transmisie, volumului de date, securității sau eficientizării resurselor hardware. Studiul actual oferă o analiză conceptuală pentru îmbunătățirea interfațării prin mecanismul OPC UA Publish-Subscribe, vizând constrângerile de funcționare în timp real și alocarea rolurilor în cazul arhitecturilor distribuite, luând în considerare strategii de interfațare bine fundamentate din sectorul automotive. Această analiză conceptuală este materializată într-o soluție ce duce comunicația OPC UA Publish-Subscribe pe UDP la următorul nivel, fiind dezvoltat un mecanism de sincronizare și o aplicație multithreading broker în scopul de a se obține o reacție în timp real a sistemului, a crește eficiența entităților implicate și pentru a se facilita transmiterea unor volume mai mari de date fără impact asupra rețelei și serviciilor prezente.

Principiile Industry 4.0 și IIoT ghidează productivitatea, calitatea, eficiența și siguranța, îmbunătățind accesibilitatea și competitivitatea [28,29]. Structurile, metodele, obiectivele și constrângerile au rolul de a spori eficiența, rezultatele făcându-se simțite aproape instant, prin sisteme masive capabile să comunice și să conlucreze [30,31]. Colectarea de date, identificarea dependențelor și a tiparelor, definirea de concepte noi și testarea acestora pe scară largă precum și îmbunătățirea capacităților de reacție a sistemelor inteligente reprezintă viitoarele direcții necesare pentru evoluția tehnologică [32,33]. Multe dintre aceste posibile îmbunătățiri se bazează pe capacitatea de interfațare a protocolului OPC UA [3,34].

OPC UA a fost implementat pentru prima oară la nivel SCADA (Supervisory Control and Data Acquisition), în medii software specifice SCADA și sub formă de servere centralizate OPC UA, însă tendința a fost spre folosirea avantajelor protocolului și la alte niveluri. Tehnologia a intrat la nivelul PLC utilizând paradigma Server-Client, iar companiile au extins funcționalitățile către cât mai multe caracteristici și servicii posibile. De-a lungul vremii au fost efectuate cercetări pentru extinderea și îmbunătățirea capabilităților OPC UA, pentru a satisface cerințele în creștere ale industriilor specifice, la diverse niveluri ierarhice. Au fost efectuate studii pentru a se încerca aducerea gradului de aplicabilitate până la dispozitivele de teren

48 | 3. Îmbunătățirea mecanismului OPC UA Publish-Subscribe în contextul eficientizării comunicației în timp real

[22,35], iar alte cercetări s-au concentrat pe integrarea tehnologiei OPC UA cu structuri cloud [36].

Primul răspuns la problemele OPCUA care ambiționează comunitățile academice și industriale, legate de funcționarea în timp real și volumele mai mari de date, a venit cu setul de specificații 14 ale OPC UA [24], specificații ce se bazează pe mecanismul Publish-Subscribe. În lucrarea [21], autorii evaluează aplicabilitatea OPC UA Publish-Subscribe în automatizarea fabricilor, în timp ce cercetarea [27] stabilește perspectiva aplicării în cadrul SDK-ului open62541 [25], care este în continuă dezvoltare. Cu toate că au fost întreprinse eforturi de către comunitatea de cercetare, trebuie clarificate aspecte legate de mecanismul OPC UA Publish-Subscribe, în special în ceea ce privește contextul în timp real, aplicabilitatea arhitecturii și îmbunătățirile ce pot fi aduse pentru diferite scenarii industriale. Lucrurile fiind departe de a fi rezolvate, sunt inițiate studii, pentru a accelera evoluțiile în vederea îndeplinirii cerințelor generale, axate pe protocoale de nivel inferior, precum MQTT (de exemplu, gateway-ul OPC UA-MQTT prezentat în [37], sau combinația dintre Sparkplug B și MQTT din [38]).

Protocoalele de comunicare care utilizează paradigma Publish-Subscribe sunt utilizate pe scară largă în aplicații cu cerințe în timp real. Chiar dacă cazurile de utilizare pot să difere de la o industrie la alta, pe lângă concepții diferite, implementări particulare ale paradigmei, precum și protocoale și standarde diferite utilizate pentru realizarea mecanismului, există și practici adoptate în comun care s-au dovedit a fi decisive în asigurarea unor performanțe ridicate pentru aplicațiile în timp real dezvoltate în ultimii ani. Având în vedere că mecanismul OPC UA Publish-Subscribe a fost definit recent, sunt așteptate îmbunătățiri și modificări în viitorul apropiat, care ar trebui să aibă la bază principiile fundamentale observate în aplicațiile deja validate în scenarii din mai multe industrii.

Unul dintre protocoalele de comunicare care implementează mecanismul Publish-Subscribe în domeniul automotive este Scalable Service-Oriented Middleware over IP (SOME/IP). Standardul Automotive Open System Architecture (AUTOSAR) descrie anumite cerințe privind protocolul SOME/IP pentru aplicațiile din domeniul automotive. În [39], procesul de notificare dintre Publisher și Subscriber este menționat prin definirea mecanismului responsabil de informarea Subscriberului cu privire la valorile modificate sau la apariția unor evenimente specifice, lăsând, de asemenea, posibilitatea că Subscriberul să solicite modificări ale valorilor sau evenimentelor sau să verifice starea variabilelor și evenimentelor prin metode desemnate, permițând implementarea mai multor strategii din punct de vedere arhitectural. Standardul împarte responsabilitățile între instanța SOME/IP utilizată pentru transportul valorilor modificate și instanța SOME/IP Service Discovery, care este responsabilă de procesele de recepție și publicare. Procesul de notificare este descris în termeni de strategii posibile pentru diferite scenarii, cu posibilități de notificare ciclică, de notificare în cazul modificărilor și de notificare condiționată pentru Subscriber, ceea ce oferă o cale versatilă de implementare a aplicațiilor cu capacități specifice de publicare-subscriere. De asemenea, mesajul de notificare SOME/IP trebuie să aibă anumite particularități, dintre care merită menționată transmiterea lungimii conținutului serializat, informație care se poate dovedi utilă pentru verificări și filtrări de la un capăt la altul. La nivel de protocol, pe lângă utilizarea UDP, protocolul SOME/IP utilizează și TCP pentru gestionarea congestiei rețelei, a mesajelor pierdute,

a erorilor de bit și a altor posibile defecțiuni care pot apărea în timpul transmisiunilor. OPC UA Publish-Subscribe se axează pe faptul că abonații interoghează în mod constant rețeaua pentru mesaje și, ulterior, filtrează mesajele primite pentru a identifica dacă au fost furnizate informațiile dorite. Cu un serviciu sau un modul de notificare dedicat, similar cu cel utilizat în SOME/IP, OPC UA ar putea extinde modularitatea conceptului Publish-Subscribe. De asemenea, acesta ar putea asigura o decuplare completă în ceea ce privește entitățile de tip Publishers și Subscribers, precum și între responsabilitățile prezente în sistem (cine trimite informațiile, cine notifică entitățile interesate, cine este responsabil pentru securitate, cine gestionează baza de timp a sistemului, cine primește informațiile). Un set de servicii sau module dedicate care ar putea fi utilizate pe post de entități separate sau ca parte a entităților definite în prezent (Publisher, broker, Subscriber), este necesar pentru a asigura o calitate ridicată a serviciilor și ar putea constitui instrumentele necesare pentru dezvoltarea de aplicații la scară largă și de arhitecturi robuste cu capacități ridicate de interacțiune între ele. Acest middleware între punctul de trimitere și punctul de primire trebuie să asigure toate mecanismele de care un scenariu ar putea avea nevoie, într-un mod abstract, facilitând scalabilitatea arhitecturii și făcând-o accesibilă tuturor utilizatorilor în funcție de nevoile acestora. În cazul protocolului de transport utilizat, este posibil ca folosirea de către OPC UA a protocolului UDP să nu fie suficientă în viitor în contextul aplicațiilor în timp real. Traectoria observabilă este îmbinarea dintre UDP și TCP la nivelul de transport, pentru protocoalele de comunicare de cel mai înalt nivel ce implementează paradigma Publish-Subscribe.

În afară de domeniul automotive, implementarea mecanismului Publish-Subscribe poate fi întâlnită în aplicații cu constrângeri de timp real și obiective similare în ceea ce privește conectivitatea, scalabilitatea și performanța ridicată. DDS (Data Distribution Service) este utilizat pe scară largă în diverse domenii, cum ar fi aeronautică, industria medicală, industria energetică etc. (de exemplu, [40,41]). Dezvoltat cu obiective de control în timp real, DDS oferă schimburi tolerante la erori, în timp real, cu latențe mici și posibilități de filtrare performante, asigurând o modularitate ridicată, o proiectare decuplabilă a entităților de Publish-Subscribe și posibilități de operare în timp real, ceea ce permite implementarea mai multor strategii de sincronizare, sporind robustețea și scalabilitatea pentru o gamă largă de aplicații. Interacțiunea dintre DDS și TSN pentru posibilități sporite de funcționare deterministică în timp, este așteptată să fie definită în mod standardizat. Fiind utilizate pe scară largă, unele dintre practicile aplicative specifice DDS se pot dovedi utile pentru conceptele mai recente ale OPC UA care vizează capacitățile de funcționare în timp real într-o rețea. Adoptarea tehnologiei TSN ca opțiune standardizată ar trebui, de asemenea, să fie așteptată în viitorul apropiat pentru OPC UA. Cu toate acestea, înainte de a se ajunge în acest punct, principiile de proiectare ale mecanismului Publish-Subscribe ar putea suferi modificări. Serviciile specifice care ar putea consolida legătura dintre TSN și OPC UA ar trebui să fie implementate ca o etapă prealabilă, îmbogățind conceptul actual de Publish-Subscribe și extinzând capacitățile standardului OPC UA în contextul IIoT. Strategia cheie pentru acest scenariu constă în observarea nevoilor și a soluțiilor pentru probleme în rândul mai multor industrii și stabilirea unor obiective pentru creșterea calității serviciilor (QoS) pe toate zonele posibile.

50 | 3. Îmbunătățirea mecanismului OPC UA Publish-Subscribe în contextul eficientizării comunicației în timp real

Obiectivele etapei actuale de cercetare sunt următoarele:

- Furnizarea unei analize a interfațării OPC UA utilizând paradigma Publish-Subscribe, în ceea ce privește constrângerile în timp real și distribuția rolurilor între entități, luând în considerare starea actuală a dezvoltărilor și a strategiilor de interfațare bine fundamentate din sectorul automotive;
- Conceperea și implementarea unei soluții ce vizează mecanismul OPC UA Publish-Subscribe prin UDP, axată pe un algoritm de sincronizare și pe o aplicație broker multithreading care să faciliteze reacții în timp real și o eficiență mai mare, pentru a extinde QoS și aplicabilitatea mecanismului de interfațare. Soluția prevede reducerea magnitudinii cuplării slabe între Subscriber și Publisher, a dificultății de a trimite volume mai mari de date pentru diverși abonați la viteze mari, minimalizând sarcina aplicată rețelei și serviciilor în ceea ce privește interogarea și filtrarea. Disponibilitatea și siguranța trebuie să ghideze această abordare, întrucât soluția trebuie să gestioneze apariția defectelor, concentrându-se pe detectare, toleranța și recuperare.

3.2. Importanța și Aplicabilitatea conceptului

3.2.1. Mecanismul Publish-Subscribe: Design și Arhitectură

Mecanismul Publish-Subscribe este un model care se bazează pe schimbul de informații în mod eficient, între 2 sau mai multe entități, având ca principiu de bază faptul că Publisher-ul va publica în mod constant o informație sau un eveniment și că Subscriber-ul va fi notificat atunci când valorile s-au modificat sau când a avut loc un eveniment. Rolurile Publisher-ului și Subscriber-ului și toate operațiile intermediare în afară de trimiterea și primirea datelor, pot fi și se recomandă să fie repartizate între mai multe entități (noduri) dintr-un sistem. Asocierea clasică dintre rolurile de server și publisher, respectiv rolurile de client și subscriber trebuie să fie desconsiderată, iar accentul trebuie pus pe informațiile care sunt publicate, cu posibilitatea ca mai multe entități să publice aceleași informații (de exemplu, un server și un alt server de rezervă, sau topicuri împărțite între servere). Un alt accent trebuie să fie pus pe recepționarea informațiilor în intervalul de timp preconizat, de către abonați. Considerând o viziune mai abstractă asupra sistemului, legătura dintre entitățile implicate este mai puțin importantă decât legătura dintre informație și țintă. Sistemul ar trebui să funcționeze cu mai puține detalii luate în considerare (de exemplu, numărul de abonați ai unui Publisher), iar toate entitățile implicate ar trebui să își împartă rolurile în avantajul unei distribuiri eficiente a informațiilor. În [42], este sugerată ideea unui mecanism intermediar, fără cunoștințe inter-obiecte prezente într-un sistem. Acest mecanism ar trebui să preia o parte din rolurile care pot fi îndeplinite fără context în ceea ce privește semnificația datelor, dar mai ales să servească drept intermediar între entitățile Publisher și entitățile Subscriber, middleware ce ar putea oferi un avantaj semnificativ pentru sistemele la scară largă. Unul dintre avantajele menționate de către autor este că entitățile responsabile cu transmiterea și primirea informațiilor se pot concentra doar pe acele operațiuni, menținând o utilizare redusă a capacităților lor de calcul (acest lucru s-ar putea

traduce prin hardware mai ieftin pentru unele noduri) și oferind o utilizare mai redusă a rețelei (acest lucru s-ar putea traduce prin capacități de răspuns mai bune între participanții din cadrul rețelei). Mesajul va fi distribuit într-o manieră mai previzibilă, oferind o viziune abstractă în ceea ce privește relațiile dintre participanții implicați de tipul Publisher și Subscriber.

În [42], sunt discutate problemele centralizării în ceea ce privește operarea senzitivă în sistemele complexe bazate pe conținutul informațional. În afară de menționarea diferitelor tipuri și subtipuri de evenimente pentru o mai bună manipulare a abonaților, serverele responsabile de evenimente unice sunt prezentate ca fiind dezavantaje în sistemele care încapsulează numeroase dependențe bazate pe un singur nod al sistemului. Astfel de entități care servesc rolul de furnizori, dar și rolul de servere de evenimente, ce țin evidența tuturor evenimentelor transmise și facilitează decuplarea dintre Publisher și Subscriber, ar trebui să fie considerate ca fiind riscuri ridicate în orice sistem implementat la scară largă. De exemplu, în cazul OPC UA, o fabrică care implementează un singur server de publicare pentru interfațarea cu o arhitectură cloud, poate suferi pierderi mari dacă serverul se prăbușește, neexistând responsabilități de rezervă desemnate către un alt server. În lipsa unui mecanism de siguranță implementat special pentru evitarea oricărui impact critic în sistem, se pot pierde date importante și se poate ajunge la o perioadă mare de nefuncționare, sau la o posibilă defecțiune totală a sistemului, dacă doar un server conține informații privind conectarea entităților (relații între multiple noduri de tip Publisher și Subscriber). Pentru a avea responsabilitățile distribuite în sistem, trebuie să existe entități middleware desemnate să fie responsabile pentru notificarea abonaților atunci când este necesar și, de asemenea, să servească drept entități de legătură între Publisher și Subscriber. Totodată, trebuie implementate măsuri de siguranță pentru a monitoriza și a reacționa în caz de defecțiuni ale nodurilor cu rol de publicare sau cu rol de server de evenimente. Numai în acest mod se poate ajunge la independență completă între conceptele de Publisher și Subscriber.

În specificații [24], rolurile de Publisher și Subscriber sunt descrise ca fiind ușor cuplate, fără a fi impactat schimbul de date de numărul de Subscriberi abonați la o singură entitate Publisher. Principala relație dintre furnizor și receptor este înțelegerea comună a seturilor de date specifice (DataSets) implicate și a detaliilor de publicare a mesajelor. Principala relație dintre ele este înțelegerea comună a seturilor de date specifice implicate și a detaliilor de publicare a mesajelor. Middleware-ul orientat asupra mesajelor este descris ca o conexiune la o adresă multicast în cazul mesajelor UDP sau ca un broker în cazul mesajelor MQTT sau AMQP. În cazul în care se utilizează UDP la nivel de transport, Publisher-ul va fi cel care trimite informații la adresa multicast. În cazul în care informațiile sunt complexe și sunt destinate unor abonați la momente diferite în timp, fără ca rolul de notificare să fie bine definit, abonații vor asculta toate mesajele care ajung la middleware (adresa multicast) și vor filtra informațiile pe baza DataSetMetaData care, de asemenea, trebuie să fie transmisă de același Publisher fiecărui Subscriber înainte de punctul de filtrare a mesajelor din rețea. Documentul [24] se referă la DataSetMetaData ca fiind informația utilizată la nivelul abonatului pentru filtrarea mesajului de rețea primit, fiind descrise opțiunile pentru obținerea Metadata de la Publisher. Una dintre metodele descrise este că DataSetMetaData să fie expediat ca mesaj de rețea înainte ca conținutul DataSetMessages să fie modificat. În cazul în care există un furnizor pentru mai mulți receptori, această strategie ar putea adăuga complexitate la sistem și ar putea crea dificultăți în ceea ce privește constrângerile de timp între publicarea și primirea mesajului. În cazul în care se adaugă criptarea pentru securitate în

52 | 3. Îmbunătățirea mecanismului OPC UA Publish-Subscribe în contextul eficientizării comunicației în timp real

context, o nouă entitate numită server de chei de securitate va fi responsabilă cu administrarea cheilor de securitate necesare entităților deja implicate. Într-un sistem robust de mari dimensiuni, cu mulți receptori implicați în relația cu un furnizor, presupunând că cheile de criptare sunt primite corect de fiecare abonat, iar aceștia ascultă toate mesajele transmise, decriptarea ar putea fi necesară ca un prim filtru în decodificarea mesajului. Executând această etapă și alte operațiuni de filtrare până când Subscriber-ul va identifica dacă mesajul este cel așteptat sau nu, s-ar putea genera un efort de calcul constant, fără niciun beneficiu imediat asupra receptorilor. Acest factor va influența atât îndeplinirea cerințelor de funcționare în timp real, cât și cazul transmiterii mesajelor multiple de către Publisher în mod dinamic (presupunând că și DataSetMetaData trebuie să fie trimis în mod constant), unde decriptarea, filtrarea și extragerea datelor trebuie efectuate de un număr necunoscut de abonați. Rețeaua însăși ar putea deveni incapabilă să gestioneze prea multe schimburi în anumite intervale de timp. O entitatea middleware între un furnizor și receptori, ar trebui să dețină informațiile privind relațiile dintre aceștia, iar în acest caz, ar putea, de asemenea, să preia distribuția DataSetMetaData și alte roluri adiționale, dacă este necesar (a se vedea Figura 12).

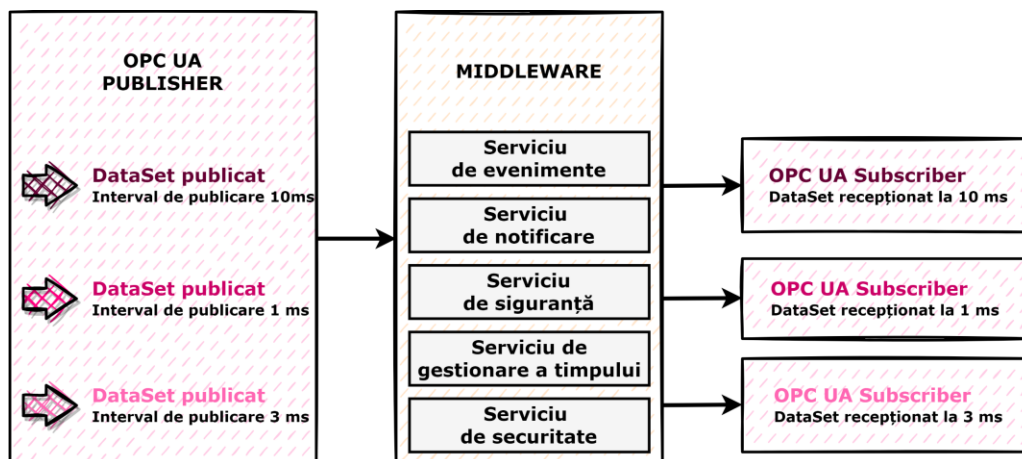


Figura 12. Propunerea de design și servicii, aferente unui component Middleware, în cazul unui scenariu de funcționare în timp real

Ar trebui puse în aplicare măsuri de siguranță pentru acest tip de situații și, în mod constant, entitatea care conține relațiile dintre furnizori și receptori ar trebui să verifice capacitățile entităților implicate pe baza acestor informații. În modelul ideal, în care entitățile Publisher și Subscriber sunt decuplate unul de celălalt, accentul ar trebui să fie pus doar pe trimiterea și primirea de informații. În special în cazul sistemelor în timp real, mecanisme de siguranță și garanții privind timpul de livrare, timpul de decriptare și timpul de filtrare ar trebui luate în considerare. Acestea nu ar putea fi implementate fără ca relația dintre toate entitățile implicate să fie stocată în mod constant într-un nod de sistemului. O posibilitate pentru un astfel de component cu rol de stocare, în cadrul mecanismului OPC UA Publish-Subscribe, ar putea fi reprezentată de următoarele dezvoltări ale PubSub Directory, conform mențiunilor din specificații.

O altă strategie sugerată în [42] detaliază noțiunea de serviciu de evenimente ca locație pentru relațiile dintre furnizori și receptori, și descrie un scenariu în care furnizorul împărtășește o anumită informație sau un anumit tip de eveniment, primind informațiile referitoare la abonații interesați de la serviciul de evenimente. În acest mod, furnizorul poate informa direct abonații cu privire la evenimente, însă această metodă nu este la fel de eficientă precum în cazul unei entități intermediare între furnizor și receptor. În plus, prin aplicarea acestei strategii, furnizorul și receptorul nu mai sunt decuplați. Conform mențiunilor autorului, comunicația de tipul many-to-many ar trebui luată în considerare când se discută despre arhitectura unui sistem robust și eficient, în care principiile mecanismului Publish-Subscribe sunt implementate corect.

În OPC UA, procesul de filtrare a mesajului de rețea din partea receptorului, creează multiple etape intermediare între punctul de recepție a mesajului și accesul la datele de interes. Aceste etape ar putea presupune mai mult efort de calcul decât este necesar și ar putea produce întârzieri problematice pentru îndeplinirea necesităților de funcționare în timp real. Implementarea unui serviciu plasat între Publisher și Subscriber, capabil să realizeze o parte din filtrarea mesajelor, ar produce un decuplaj mai accentuat. Astfel, componentul de tip Subscriber ar putea să se concentreze asupra tipului de informații pe care le vizează și ar putea, probabil, să ofere o anumită siguranță împotriva atacurilor de tip "man-în-the-middle" care ar putea apărea în cazul unei independențe totale între entitățile editorilor și abonaților.

În cazul în care middleware-ul este reprezentat de un broker în OPC UA, abordarea de proiectare a sistemului în ceea ce privește decuplarea furnizorilor și a receptorilor este mai apropiată de principii, dar în acest caz entitatea dintre furnizori și receptori nu leagă aplicații de pe același nivel al modelului OSI. Entitatea Publisher se va conecta într-adevăr la broker și va fi decuplată de receptori, datele vor fi primite și stocate de broker, dar, în funcție de protocolul de transport utilizat de Publisher, brokerul va fi reprezentat ca o entitate de recepție MQTT sau AMQP, deci un receptor de nivel inferior comparativ cu o aplicație OPC UA Publisher. Având în vedere forma mesajului (format MQTT/AMQP), receptorii brokerului vor fi reprezentați de alte aplicații MQTT/AMQP, care vor citi datele dorite fără a fi nevoie să interacționeze cu straturile superioare nivelului de transport (în care este implementat OPC UA) din modelul OSI. Ca atare, în acest caz opțiunea Publish/Subscribe nu este implementată în totalitate, ci mai degrabă este un schimb între un OPC UA Publisher și o entitate destinatară (entitatea broker) care extrage informațiile și le partajează cu alte aplicații receptoare de tipul MQTT/AMQP. Cazul de utilizare al acestui design este mai degrabă interacțiunea cu arhitecturi cloud și cu entități cu diferite niveluri de înțelegere a modului OPC UA de transmitere a datelor, cerințele de timp real în acest caz putând fi dificil de implementat. Caracteristicile și serviciile de siguranță care ar putea garanta transmiterea în anumite intervale de timp sau detectarea pierderii mesajelor, s-ar putea de asemenea dovedi mai greu de implementat în acest tip de design. Conform [24], se va realiza securizarea transportului între toate entitățile implicate (în acest caz Publisher, broker și Subscriber), deși nivelul de înțelegere al protocolului OPC UA între Publisher și Subscriber diferă. Cu toate acestea, securizarea specifică protocolului, de la un capăt la altul între Publisher și Subscriber mai sus de nivelul de transport, nu este realizabilă. Într-un caz clasic în care controlerile comunică cu interfețele om-mașină (HMI) și în care este necesară decuplarea entităților, utilizarea unui design cu un broker (MQTT/AMQP) ar putea crea dificultăți în asigurarea faptului că transmisia se face într-un ciclu de timp fix, așa cum ar trebui să se întâmple. Deși acest design este potrivit pentru schimburi multiple de date între dispozitive de cost

54 | 3. Îmbunătățirea mecanismului OPC UA Publish-Subscribe în contextul eficientizării comunicației în timp real

redus cu software de complexitate redusă, asigurarea calității serviciilor s-ar putea dovedi mai laborioasă decât ar trebui să fie.

În [43], un mecanism Publish/Subscribe este descris într-un design cu un middleware (middleware ROS), în care furnizorul nu trebuie să fie la curent cu identitatea sau numărul receptorilor. Fiecare astfel de detaliu va fi gestionat de middleware, oferind o viziune abstractă asupra furnizorului și a receptorilor. Acest lucru va permite înlocuirea entităților Publisher și Subscriber în timp real, dacă este necesar. Având această concepție, middleware-ul este implementat ca o veritabilă legătură între aplicațiile de tip Publisher și Subscriber la același nivel, cu aceeași înțelegere în privința tehnologiile utilizate. Această implementare diferă de designul cu broker OPC UA (specific pentru transportul MQTT/AMQP), unde aplicațiile Publisher și Subscriber pot avea o înțelegere diferită a tehnologiei OPC UA.

În domeniul automotive, în cazul protocolului SOME/IP, modelul Publish Subscribe este responsabil pentru schimbul de evenimente între entitățile implicate. Receptorul nu se abonează direct la un anumit eveniment, ci la un grup de evenimente, similar cu abonarea la un set de date publicat în cazul OPC UA, însă abonarea se face prin intermediul serviciului SOME/IP Service Discovery. Principala responsabilitate a serviciului (Service Discovery) este de a transmite disponibilitatea serviciilor pentru orice participant și de a controla secvența de trimitere a oricărui mesaj specific unui eveniment. Disponibilitatea acestui serviciu permite entităților de tip Subscriber să recepționeze doar mesajele de interes. Abordarea referitoare la procesul de descoperire este utilizată și în alte protocoale care implementează mecanismul Publish-Subscriber. De exemplu, DDS utilizează descoperirea topic-urilor, într-un sistem ce nu se bazează ca aplicația să găsească datele de interes, ci mai degrabă să se focalizeze pe procesul de recepție, îmbunătățind în acest fel designul decuplat al sistemului și oferind posibilități de implementare a unor mecanisme de siguranță în vederea detectării mesajelor pierdute.

3.2.2. Sincronizarea de timp în contextul OPC UA și TSN

În contextul TSN și al mecanismului OPC UA Publish Subscribe, accentul în IIoT se îndreaptă rapid către funcționalitatea în timp real de la nivelul rețelei până la nivelul aplicației. Obiectivul OPC UA Publish-Subscribe este de a face următorul pas în furnizarea de soluții pentru aplicații la scară largă care pot face schimb de informații în timp real și care ar putea fi ușor de integrat în arhitecturile existente. Tehnologia TSN reprezintă soluții pentru nivelul de legătură de date (Data Link Layer) al modelului OSI, soluții ce asigură precizia timpului în transmiterea mesajelor în cadrul unei rețele, utilizând o serie de standarde pentru sincronizarea clock-urilor și definirea referințelor temporale pentru toți participanții la rețea. Având în vedere că ambele tehnologii sunt dezvoltate progresiv în vederea îndeplinirii cerințelor de funcționare în timp real, ar trebui să se ia în considerare o nouă abordare orientată către examinarea altor tehnologii și infrastructuri validate în îndeplinirea cerințelor de timp real, în domenii precum cel automotive, aeronautic, militar și în diverse alte industrii. Trebuie evidențiat modul în care sunt îndeplinite cerințele de funcționare timp real și impactul asupra altor subiecte, cum ar fi securitatea, siguranța, calitatea serviciilor, scalabilitatea și costurile. Compararea cu contextul actual din automatică în legătură cu nevoile și soluțiile disponibile, ar trebui să îmbunătățească în mod constant evoluția tehnologiei, înțelegându-se mai bine ce obstacole trebuie depășite în sectorul IIoT.

În domeniul automotive au fost dezvoltate aplicații în timp real cu complexități diferite. În sisteme care interacționează la diferite niveluri în interiorul unui automobil. Standardul AUTOSAR oferă module specifice care definesc cerințele privind gestionarea și sincronizarea timpului. Unul dintre acestea este de modulul de sincronizare a timpului. Făcând parte atât din platforma clasică, cât și din platforma Adaptive, implementarea modulului de sincronizare a timpului, controlează baza de timp utilizată pentru sincronizarea între diferite noduri și poate fi descrisă că o încapsulare a protocolelor de timp care facilitează coordonarea între procesele ce rulează pe diferite unități de control electronic (ECU), prin stabilirea unei înțelegeri comune asupra timpului. Versiunea Adaptive pentru procesul de sincronizare a timpului, este reprezentată de un singur modul (SWS TimeSync), și include de asemenea componentul TSP (Time Synchronization Provider), component ce în cazul platformei clasice este împărțit în alte 3 module, fiecare fiind specific protocolului. Ținând cont de obiectivele privind funcționarea în timp real pentru aplicațiile OPC UA, este logic să analizăm unele dintre cerințele utilizate pentru obținerea unui comportament în timp real în standardul AUTOSAR, începând cu modulul de sincronizare a timpului și, de asemenea, să analizăm impactul pe care îl are o astfel de cerință asupra diferitelor module și tehnologii.

Conform [44], starea de sincronizare ar trebui să fie disponibilă prin intermediul interfețelor pentru monitorizarea și detectarea componentelor al căror comportament ar putea fi influențat de posibile desincronizări în cazul procesului de recepție a informațiilor. Alte cerințe se referă la existența unei înțelegeri comune a timpului în toate entitățile implicate, la posibilitatea de a partaja această bază de timp prin Ethernet și la un proces de sincronizare realizat în cadrul bazei de timp definite, ce trebuie să aibă loc între toate componentele software și ECU implicate, pentru a asigura o procesare precisă a datelor senzorilor din surse multiple. În cazul tehnologiei OPC UA, pentru realizarea unei sincronizări stricte în timp real, este necesară utilizarea tehnologiei TSN specifică pentru nivelul de legătură de date (Data Link layer) al modelului OSI.

Unul dintre standardele implementate de TSN este legat de sincronizarea participanților pe baza unui grandmaster clock, după cum se menționează în [45]. Implementarea TSN ar trebui să garanteze o anumită livrare în intervale de timp bine stabilite, a mesajelor din rețea și ar trebui să integreze sistemul Publish-Subscribe al OPC UA într-un context de funcționare în timp real, vizând schimbul de date între controlere. Cu toate acestea, în momentul de față, este greu de prezis dacă adoptarea tehnologiei TSN va fi suficientă pentru aplicațiile OPC UA la scară largă, cu cerințe de funcționare timp real, utilizându-se specificațiile actuale ale mecanismului Publish-Subscribe. Monitorizarea sincronizării între entitățile implicate, în acest caz între OPC UA Publishers și Subscribers, este un alt subiect important. Ținând cont de faptul că, în infrastructurile de mari dimensiuni, cu aplicații multiple care vor necesita cerințe de sincronizare diferite, fără interfețe dedicate care să permită monitorizarea fiecărei componente implicate, gestionarea bazelor de timp și a interacțiunilor dintre aplicațiile în timp real ar deveni copleșitoare până la urmă. Serviciile dedicate pentru gestionarea timpului și măsurile de siguranță care ar putea proteja utilizatorul împotriva desincronizării operațiilor, pot îmbunătăți popularitatea și scalabilitatea oricărui protocol de comunicare. Acest tip de mecanisme ar putea reprezenta o soluție pentru aplicațiile în timp real, în special în contextul IIoT, în care interacțiunea va fi necesară nu numai între componente, ci și între arhitecturi existente.

Alte cerințe importante prezentate în [44] menționează notificările în cazul în care se depășesc perioade de timp pentru operațiile ce așteaptă să primească

56 | 3. Îmbunătățirea mecanismului OPC UA Publish-Subscribe în contextul eficientizării comunicației în timp real

informații la anumite intervale fixe, cu scopul evitării stărilor de așteptare. Aceste notificări sunt prezentate în document cu scopul evitării pe cât posibil a procedurilor de interogarea a rețelei. În contextul evitării interogării inutile a rețelei, mecanismul actual OPC UA Publish-Subscribe nu pare să partajeze cu servicii dedicate, intervalele de timp folosite în procesul de publicare, așadar receptorii interoghează rețeaua în mod constant în cazul mesajelor pe UDP distribuite de Publisher. Viitoarele servicii care ar putea crește eficiența aplicațiilor de tip Subscriber și ar putea diminua efortul de calcul necesar pentru o astfel de operațiune, prin repartizarea responsabilităților către alte entități (de exemplu, un serviciu de notificare care va informa abonatul atunci când este necesar, permițându-i acestuia să se concentreze doar pe primirea informațiilor), ar putea fi dezvoltate cu obiective precise, cum ar fi prevenirea interogării inutile a rețelei.

Posibilitatea de transmisiuni declanșate, este menționată în cazurile în care nu mai este necesară așteptarea următorului ciclu de timp, iar operațiunile pot avea loc chiar în acel moment, permițând posibile creșteri ale eficienței în unele cazuri și potențiale resincronizări. Transmisia ar putea fi declanșată similar în cazul OPC UA, însă momentul de declanșare este specific aplicației, pe baza capacităților hardware sau a capacităților sistemului de operare. Luând în considerare garanțiile temporale pe care TSN le-ar putea oferi, referințele temporale trebuie să fie clar definite între nivelul de legătură de date (Data Link Layer) specific TSN și nivelele OPC UA în contextul modelului OSI, pentru a asigura, în primul rând, o sincronizare a operațiunilor din interiorul entității (Publisher sau Subscriber) pe baza aceleiași baze temporale (a se vedea Figura 13).

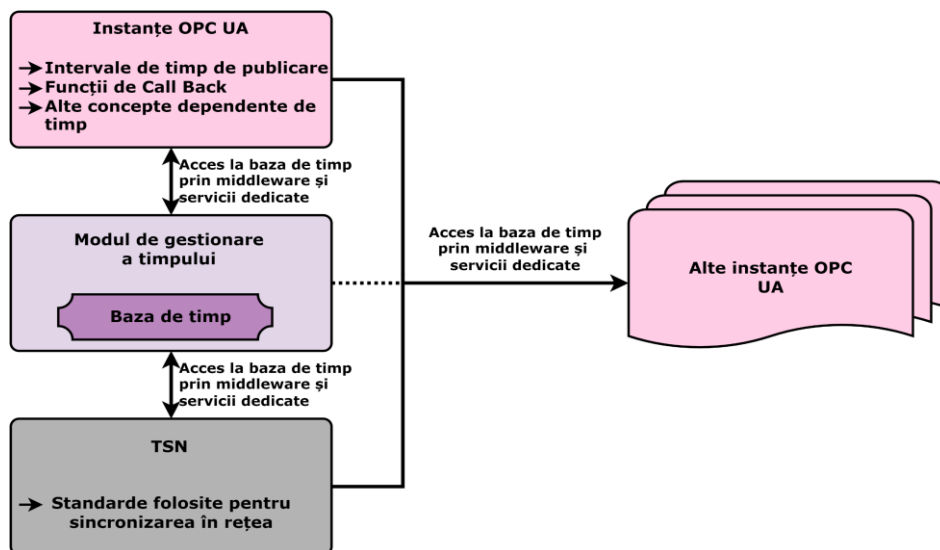


Figura 13. Propunere de design în care toate entitățile se bazează pe o bază temporală comună

În cazul standardului AUTOSAR (și precizat de asemenea în documentul [44]), trebuie să existe modalități de accesare a bazei de timp, alături de interfețe care să furnizeze informații privind desincronizarea, abaterile de timp, întârzierile și alte detalii utile de care aplicațiile ar putea beneficia într-un scenariu real. Orice detaliu referitor la conceptul de timp, chiar și în cazul temporizatoarelor specifice sistemului

de operare, ar putea fi folosit în procesele de sincronizare internă și cu alte entități. Prin urmare, trebuie să existe mecanisme dedicate, care să lege tehnologiile deterministice în timp, de preferință într-un mod standardizat.

Actualul determinism temporal OPC UA se bazează pe conceptul de publicare-subscriere, iar TSN poate oferi capacități îmbunătățite pentru îndeplinirea cerințelor de timp real. Cu toate acestea, percepția timpului, la nivel intern în entitățile implicate în schimbul de date prin rețea și la nivel extern prin diferite mijloace și strategii de sincronizare, trebuie să fie dezvoltată în continuare în scopul standardizării, prin servicii bine definite și soluții middleware, așa cum se observă în cazul altor tehnologii deterministe cu obiective similare. În cazul îmbunătățirii QoS, în fiecare concept care implică operațiuni bazate pe timp, noțiunea de timp în sine ar trebui să fie prezentă și la nivelurile superioare ale modelului OSI (nivelurile OPC UA). Serviciile și modulele desemnate care gestionează și partajează accesul la baza de timp trebuie să fie dezvoltate în concordanță cu etapele viitoare pentru creșterea performanțelor și a posibilităților aplicațiilor, module ce ar putea lua în considerare interacțiunea cu tehnologia TSN care oferă, de asemenea, capacități în timp real. De la intervalele de publicare, până la funcțiile de tip call-back care utilizează o bază de timp pentru operațiuni, trebuie asigurată uniformitatea noțiunii de timp. Modulele și serviciile desemnate, alături de adoptarea altor standarde și tehnologii utile pentru garantarea sincronizării timpului și a latenței, reprezintă următorul pas pentru a aduce standardul OPC UA mai aproape de depășirea provocărilor prezente în contextul IIoT și Industry 4.0.

Pentru a obține un răspuns în timp real între aplicații cu baze temporale diferite, este necesară o sincronizare. Pentru studiul de caz, detaliat în secțiunea următoare, în care o aplicația de tip broker transmite informații la un interval de timp prestabilit, scopul este ca entitatea Subscriber să fie sincronizată cu brokerul, reușind să identifice momentul potrivit pentru a citi mesajul din rețea. Pentru această situație a fost elaborat un algoritm de sincronizare, descris în detaliu de asemenea în secțiunea următoare, alături de alte operații necesare pentru a identifica doar mesajele de interes în cazul fiecărui receptor, în orice moment de timp, astfel evitându-se pe cât posibil interogarea rețelei.

3.3. Studiu de caz

Mecanismul OPC UA Publish-Subscribe trebuie analizat în mod constant în contextul cerințelor de funcționare în timp real, pentru diverse scenarii prezente în industrie. Odată cu avansarea rapidă a IIoT și cu evoluția și extinderea constantă a stivei de tehnologii implicate în procesul de schimb de date între multiple componente de interes, observarea constantă în ceea ce privește soluțiile disponibile și diverse comparații cu nevoi și provocări specifice mai multor domenii, poate produce îmbunătățiri semnificative pentru dezvoltarea viitoare a aplicațiilor la scară largă și pentru identificarea celei mai bune soluții în orice caz de utilizare. În acest capitol se descrie implementarea unei aplicații multithreading broker în contextul mecanismului OPC UA Publish-Subscribe, și se analizează principiile de funcționare și îmbunătățirile arhitecturale obținute. Aplicația folosește UDP ca protocol de transport și se focalizează atât pe îndeplinirea constrângerilor de funcționare în timp real cât și pe o mai bună distribuire a rolurilor în interiorul arhitecturii definite.

3.3.1. Arhitectura Sistemului

Scopul aplicației de tip broker este de a asigura interfațarea între Publisher și Subscriberi într-un mod eficient, menținând implementarea aproape de specificații [24]. Având mai multe entități prezente, abordarea studiului de caz presupune ca fiecare entitate să ruleze pe un dispozitiv separat, cu sistem de operare bazat pe Linux, după cum se vede în Figura 14.

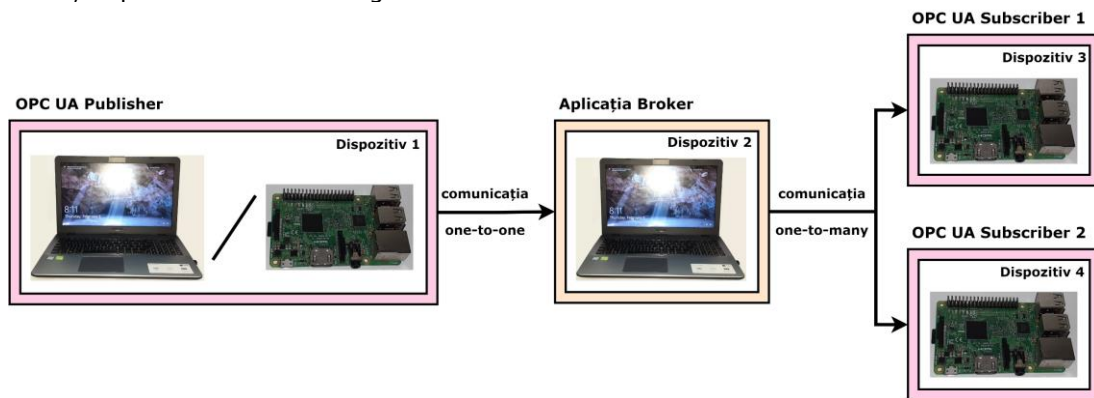


Figura 14. Arhitectura sistemului

3.3.2. OPC UA Publisher

Entitatea Publisher este responsabilă pentru furnizarea de date fără a avea cunoștințe despre destinatarii și consumatorii finali ai informațiilor. La fel ca în [45], implementarea actuală urmează aceeași secvență în ceea ce privește implementarea componentelor de configurare OPC UA Publish-Subscribe. Publisher-ul trebuie mai întâi să fie configurat în mod corespunzător și să inițializeze conexiunea Publish-Subscribe folosind metoda `UA_Server_addPubSubConnection`. În acest caz, entitatea Publisher urmărește să transmită informații către o singură entitate, aplicația Broker, prin urmare aplicația va avea, de asemenea, o conexiune inițializată și configurată pentru a recepționa date doar de la Publisher. Al doilea pas este crearea componentei `WriterGroup` (a se vedea [24,45]), ce conține parametrii responsabili pentru crearea mesajului de rețea. Un parametru foarte important în cazul constrângerilor de funcționare în timp real, este intervalul de publicare, parte din tipul de dată `UA_WriterGroupConfig`. Ținând cont de faptul că se dorește o comunicație de tipul one-to-one între Publisher și aplicația Broker, datele ce se transmit pot fi asamblate într-o manieră mai abstractă, păstrând simplitate din punct de vedere al implementării. Este de așteptat ca unul dintre scopurile de a avea o aplicație Broker în acest caz, să fie reprezentat de menținerea aplicației Publisher mai abstractă din perspectiva componentelor de configurare. Payload-ul este reprezentat printr-un număr hexazecimal (de exemplu, `0xDC`), fără să se știe cum vor fi împărțite datele între entitățile Subscriber, sau câți astfel de receptori țintă există. Pentru o mai bună înțelegere a structurii mesajului, ca exemplu ipotetic se poate presupune că, fiecare 2 biți ai numărului pot reprezenta o valoare colectată de un senzor, iar furnizorul rulează pe un dispozitiv de teren ce colectează aceste date. Numărul va fi reprezentat

de un DataSetMessage (a se vedea [24,45]) cu un singur câmp, menținând astfel structura payload-ului cât mai simplă posibil. Intervalul de timp de publicare este stabilit în mod corespunzător în funcție de nevoile entității Publisher, însă, în funcție de scenariul vizat, intervalul de timp ar trebui să fie stabilit aproape de nevoile consumatorului final. Ultimul detaliu în dezvoltarea entității Publisher este conexiunea cu aplicația broker. Acesta se așteaptă să distribuie date doar către aplicația broker, astfel încât nu este necesar să se utilizeze o adresă multicast pentru conexiune, IP-ul dispozitivului care conține aplicația broker și un port disponibil fiind suficiente pentru a se conecta și a publica datele.

3.3.3. Aplicația Broker

Într-un design cu o aplicație broker, se urmărește îmbunătățirea eficienței mecanismului de publicare-subscriere. Pe lângă filtrarea și direcționarea numai a informațiilor utile către fiecare receptor, creșterea eficienței poate fi realizată prin obținerea sincronizării între entitățile implicate. Implementarea actuală urmărește toate aspectele menționate mai sus și urmărește îndeplinirea fiecărui obiectiv într-o manieră apropiată de cerințele industriei.

Aplicația Broker este formată din 2 componente, fiecare cu roluri diferite. Rolurile principale sunt primirea datelor de la OPC UA Publisher și publicarea ulterioară a acestor date în intervalul de timp dorit (către fiecare abonat). Obiectivele principale ale Broker-ului sunt de a prelua unele dintre rolurile implicate în designul Publish-Subscribe, de a îndeplini cerințele de funcționare în timp real prin stabilirea intervalelor de publicare în funcție de nevoile fiecărui receptor și de a executa majoritatea proceselor de filtrare în acest punct specific al schimbului de date. Având de gestionat 2 operațiuni diferite (primirea și transmiterea ulterioară a datelor) cu constrângeri de timp diferite, alături de alte sarcini specifice, s-a ales o abordare multithreading pentru o performanță și o sincronizare mai bune, între procedurile aplicației broker. Deși SDK-ul [25] nu oferă multithreading, faptul că se gestionează operațiuni complet diferite la nivelul aplicației Broker, a făcut ca această abordare să fie fezabilă în cazul de față. Arhitectura internă a aplicației Broker este împărțită în 2 componente, fiecare dintre ele rulând pe un thread diferit (a se vedea Figura 15).

60 | 3. Îmbunătățirea mecanismului OPC UA Publish-Subscribe în contextul eficientizării comunicației în timp real

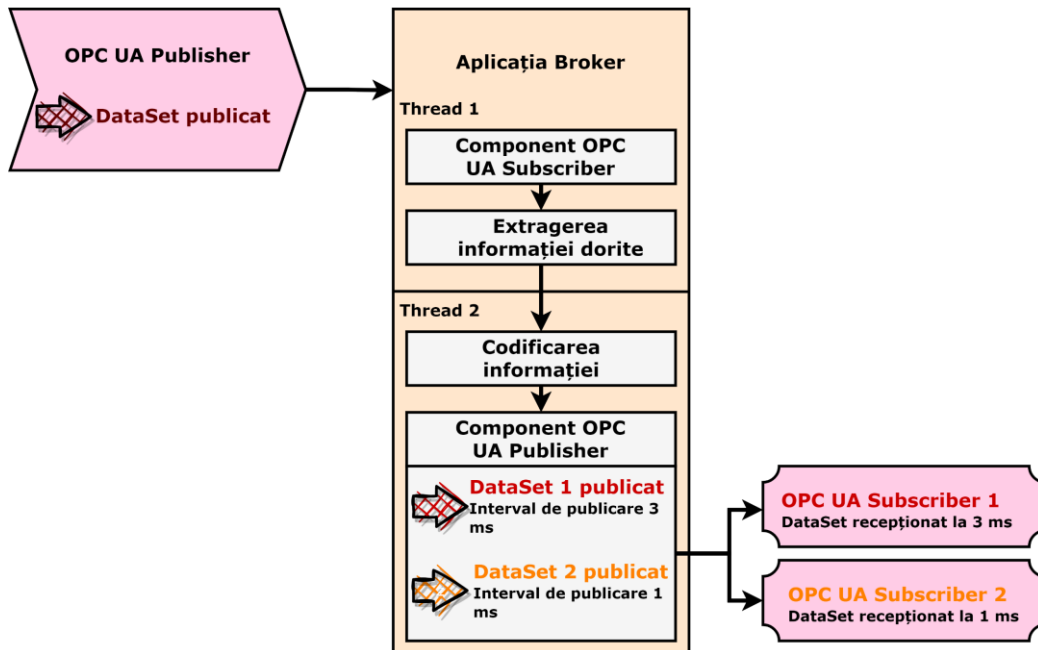


Figura 15. Arhitectura internă și Interacțiunea aplicației Broker

Conceptul din spatele aplicației este realizat pe baza unor presupuneri în legătură cu detaliile de funcționare în timp real, datele de interes și ID-ul unic, caracteristici care ar trebui să fie furnizate de către Subscriberi. Viitoarele iterații ale aplicației Broker ar putea implementa o a treia componentă responsabilă pentru un prim schimb între Broker și receptorii interesați, pentru detaliile de configurare, prin intermediul unei transmisiuni unice inițiale, folosind paradigma clasică server-client. Alte opțiuni fiabile pentru furnizarea către aplicația Broker a detaliilor particulare specifice receptorilor, ar consta fie într-o etapă de configurare bazată pe un fișier de configurare Json în timpul execuției, fie printr-o comunicare periodică către un server dedicat ce conține astfel de detalii specifice participanților. Pentru studiul de caz actual, aceste detalii de configurare sunt hard-codate la nivelul aplicației Broker.

Primul thread este dedicat componentei Subscriber (parte a aplicației Broker) ce comunică cu furnizorul de date de tip Publisher. Pe lângă procedura de recepție, această componentă va executa, de asemenea, extragerea datelor dorite în variabile țintă specifice, pe baza preferințelor fiecărui Subscriber. Primul pas în execuția aplicației Broker este inițializarea unei conexiuni PubSubConnection cu furnizorul principal pe o adresă unicast (comunicare unu la unu). După ce conexiunea este inițializată, componenta începe să asculte în rețea mesajele de interes. Implementarea procesului de subscriere se face într-un mod clasic, prin interogarea a tot ceea ce circulă prin rețea cu o recurență cât mai mică posibil, pentru a garanta că niciunul dintre mesajele de tip DataSetMessages importante nu se pierde, prin parcurgerea tuturor câmpurilor conținute în DataSetMessages și prin filtrarea pe baza tipurilor de date.

Al doilea thread este dedicat componentei Publisher (parte a aplicației Broker). Pe baza numărului de receptori și a intervalelor de timp la care aceștia se

așteaptă să primească datele, componenta inițializează o instanță de publicare OPC UA, având mai multe subcomponente de tipul `WriterGroups`, fiecare cu un interval de timp de publicare diferit, bazat pe nevoile tuturor abonaților. Datele sunt stocate după extragere, înainte de a fi transmise mai departe, permițând accesul la istoric, dacă este necesar (mecanism denumit `data buffering`, esențial din punct de vedere al disponibilității datelor ce circulă între multiple entități). Înainte ca informațiile să fie atribuite unui `DataSet`, este necesară o operațiune de codificare cu un anumit ID unic, specific fiecărui receptor. În cazul acestui studiu de caz, ID-urile au fost definite sub forma unui număr hexazecimal (0xF corespundent datelor de interes pentru Subscriber 1, respectiv 0xA corespundent datelor de interes pentru Subscriber 2). Funcția de codificare decalează informația extrasă cu 4 biți și atribuie ID-ul pe primii 4 biți mai puțin semnificativi din numărul obținut după decalare. Prin urmare, payload-ul 0xDC (menționat în secțiunea anterioară, OPC UA Publisher) destinat receptorilor finali, devine:

- 0xCF pentru Subscriber 1, unde 0xC reprezintă datele de interes pentru Subscriber 1 și 0xF ID-ul,
- 0xDA pentru Subscriber 2, unde 0xD reprezintă datele de interes pentru Subscriber 2 și 0xA ID-ul.

După codificare, noile payload-uri sunt încapsulate în `DataSetMessages` și publicate la diferite intervale de timp de către unul dintre subcomponentele `WriterGroups`. Transmiterea se face pe o adresă multicast pentru accesul comun al abonaților (comunicare de la unul la mai mulți), în conformitate cu [24].

3.3.4. Receptorii de tip OPC UA Subscriber

Pentru acest studiu de caz, au fost dezvoltate 2 entități Subscriber ca și consumatori ai payload-ului inițial, furnizat de OPC UA Publisher, fiecare cu așteptări diferite în ceea ce privește datele vizate și comportamentul în timp real. Pentru exemplificarea scenariului actual, intervalele de timp la care receptorii așteaptă datele, sunt de 1 secundă pentru Subscriber 1 și de 3 secunde pentru Subscriber 2. Conceptul este testat și funcțional și cu intervale de timp sub 10 milisecunde (la 3 milisecunde comportamentul încă este similar cu cel din cazul intervalelor de timp vizate).

Fiecare Subscriber este implementat în mod similar, primii pași fiind, în ambele aplicații, inițializarea `PubSubConnection` și stabilirea comunicării cu aplicația Broker pe adresa multicast selectată. Știind că aplicația Broker va trimite numai informațiile de interes, au fost evitate orice operații inutile de filtrare specifice mesajului de rețea OPC UA. De asemenea, s-a evitat operațiunea de navigare (`browse`) utilizată de obicei pentru filtrarea conținutului recepționat, fiecare Subscriber știind că aplicația Broker va trimite numai datele specifice și numai la momentul dorit (în cazul ideal). Funcția de decodificare bazată pe ID se face pentru abilitatea receptorului de a clasifica mesajele recepționate ca fiind valide și nevalide cu scopul sincronizării, scenariu detaliat în secțiunea următoare.

Obiectivele principale ale aplicației Broker au fost ca receptorul să evite interogarea continuă a rețelei pentru conținut și să folosească intervalul de timp de publicare al conceptului de Publisher pentru acest lucru, precum și să evite cât mai multe operațiuni de filtrare, având garanția că, datorită sincronizării, vor fi primite numai informații utile. Prin realizarea acestor obiective, eficiența aplicației Subscriber este sporită, ceea ce reprezintă un avantaj, în special dacă scenariul vizat este

62 | 3. Îmbunătățirea mecanismului OPC UA Publish-Subscribe în contextul eficientizării comunicației în timp real

asocierea aplicației Subscriber la un dispozitiv de teren cu putere de calcul și resurse reduse. Ținând cont de faptul că cerințele de funcționare în timp real reprezintă un aspect important al conceptului OPC UA Publish-Subscribe, implementarea din perspectiva aplicațiilor Subscriber a urmat același aspect, instrucțiunile privind primirea datelor fiind executate numai la intervale de timp egale cu cerințele prestabilite. Pentru exemplificarea intervalului de timp dintre apelurile instrucțiunilor de recepție, în secțiunea următoare se va folosi termenul Delay. Starea ideală de funcționare pentru Subscriber va fi ca Delay-ul să fie egal cu cerințele de așteptare prestabilite. Cu toate acestea, au fost dezvoltate anumite mecanisme în eventualitatea ieșirii din cazul ideal de funcționare (desincronizare). Toate aceste mecanisme și detalii adiționale referitoare la algoritmul de sincronizare dezvoltat pentru acest design (ce conține o aplicație OPC UA Broker), sunt descrise în secțiunea următoare.

3.3.5. Algoritmul de Sincronizare

Cu aplicația Broker în funcțiune și cu receptorii care așteaptă mesajele corecte la momentul de timp potrivit, pentru cazul ideal de funcționare, toate configurațiile sunt corecte. Cu toate acestea, există scenarii în care doar configurarea corectă a detaliilor de sincronizare atât pentru Broker, cât și pentru receptori, nu este suficientă pentru a realiza sincronizarea în timp real între entități care nu au o bază temporală comună și nu fac schimb constant de referințe temporale prin intermediul unui mecanism de notificare. În aceste cazuri, provocarea constă în sincronizarea dinamică a entităților separate, știind intervalele aproximative ale operațiunilor de transmisie și recepție. Pentru exemplificarea conceptului, se va utiliza termenul de polling în următoarea parte a acestei secțiuni pentru a descrie desincronizarea entităților Subscriber de aplicația Broker. Termenul polling s-a definit pe baza a 2 situații specifice stării receptorilor:

- A. Receptorul primește informații pe care funcția de recepție OPC UA le clasifică drept date invalide. (care nu au forma unui mesaj de rețea sau operațiunea nu a fost executată corespunzător din motive specifice rețelei), ceea ce înseamnă că nu există nicio modalitate de a ști când va ajunge mesajul de interes și că reexecutarea funcției de recepție cât mai rapid (fiind în acest moment desincronizare față de aplicația Broker/ stare de polling), va furniza în cele din urmă mesajul dorit.
- B. Receptorul primește un mesaj de rețea, dar are un tip diferit de cel dorit și definit de protocolul OPC UA, ceea ce înseamnă din nou că nu există nicio modalitate de a ști când va sosi mesajul de tipul dorit, iar reexecutarea funcției de recepție cât mai rapid (fiind în acest moment desincronizare față de aplicația Broker/ stare de polling), va furniza în cele din urmă mesajul dorit.

Ambele situații specifice stării de polling definite, pot fi evitate doar prin obținerea sincronizării între momentul trimiterii și momentul primirii datelor dorite.

Din configurarea aplicației Broker pentru a expedia mesajele desemnate către diferiți Subscriberi pe baza unui ID (a se vedea secțiunea anterioară), și având două intervale de livrare diferite pentru mesaje, pe o adresă multicast, din perspectiva

abstractă a receptorului putem identifica 2 scenarii diferite bazate pe validitatea mesajelor (receptorul va clasifica mesajul codificat cu ID-ul său ca fiind un mesaj valid și va clasifica toate celelalte mesaje, în ciuda numărului de receptori implicați, ca fiind mesaje invalide), și pe baza recurenței mesajelor (în scop exemplificativ, recurențele alese au fost de 100 de milisecunde pentru mesajele valide și 1000 de milisecunde pentru mesajele invalide și viceversa) (vezi Figura 16).

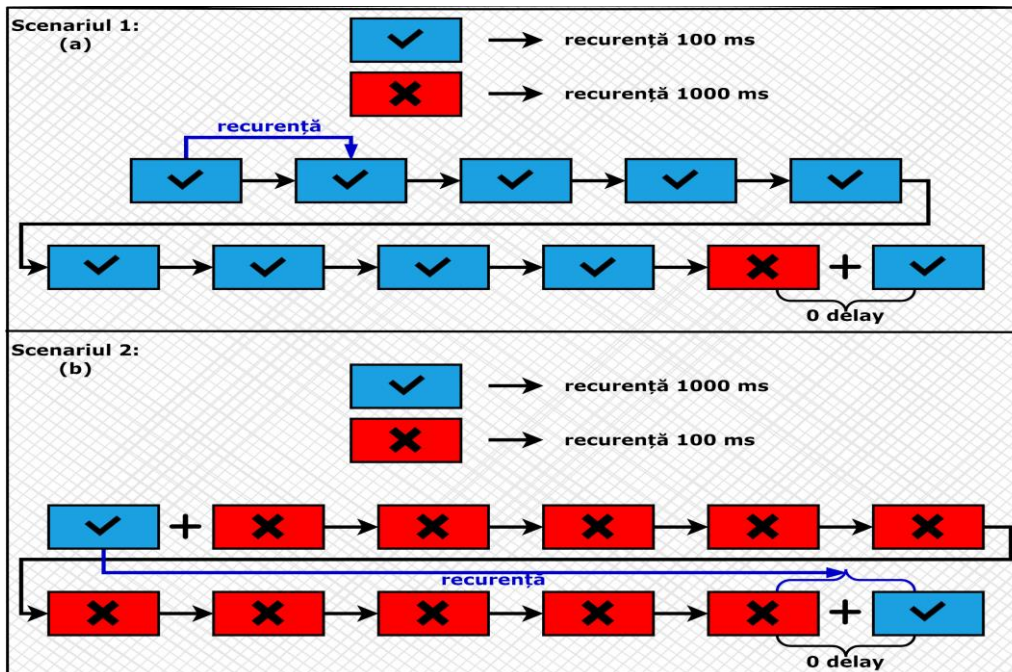


Figura 16. Cele 2 scenarii din perspectiva Receptorilor de tip Subscriber, în legătură cu validitatea și recurența mesajelor. (a) Scenariul 1: -cu recurența mare pentru mesajele valide. (b) Scenariul 2: -cu recurența mică pentru mesajele valide

În scenariul 1: Dacă a fost primit un mesaj invalid sau dacă este detectată o stare de polling, intervalul Delay devine 0 (deci instrucțiunile de recepție se vor executa cât mai repede posibil) până când va avea loc un eveniment de sincronizare (în cazul algoritmului, evenimentul de sincronizare înseamnă primirea unui mesaj valid). Dacă a avut loc un eveniment de sincronizare și a fost primit un mesaj valid, intervalul Delay devine din nou egal cu intervalul de timp de recepție dorit, specific receptorului. Din acest moment, există siguranța că aplicația Broker trimite mesajul folosind același interval de timp, astfel încât este normal să nu se execute operațiunea de recepție până la următorul ciclu, iar dacă mesajul a sosit, nu mai este necesară starea de polling. Din acest moment, dacă rețeaua este stabilă, receptorul ar trebui să fie sincronizat cu aplicația Broker. Singurul caz în care este posibilă desincronizarea, este atunci când intervalele de timp de livrare pentru cei 2 receptori se vor intersecta, iar aplicația Broker va furniza mesajele foarte rapid unul după altul, receptorul nefiind sigur care dintre ele va fi furnizat primul. În acest caz, instrucțiunea

de recepție este executată și dacă mesajul care sosește primul este invalid, intervalul Delay redevine 0 și instrucțiunea se repetă cât mai repede posibil până când mesajul valid este detectat, astfel nefiind pierdut din cauza suprapunerii intervalelor. Și de această dată, apariția mesajului valid este interpretată ca un eveniment de sincronizare și până la următoarea intersectare a intervalelor de timp de livrare pentru aplicația Broker, receptorul este sincronizat. Chiar dacă desincronizarea apare în mod neprevăzut, mecanismul de modificare dinamică a intervalului Delay, are capacitatea de a resincroniza receptorul cu aplicația Broker. Între mesajele valide va exista întotdeauna recurența dorită, interogarea rețelei (starea de polling) este evitată din momentul în care se face prima sincronizare (ceea ce crește eficiența), iar probabilitatea de a primi mesaje invalide este luată în considerare, desincronizarea fiind remediată cât mai repede posibil.

În scenariul 2: Dacă a fost primit un mesaj invalid sau dacă este detectată o stare de polling, intervalul Delay devine 0 (deci instrucțiunile de recepție se vor executa cât mai repede posibil) până când va avea loc un eveniment de sincronizare (în cazul algoritmului, evenimentul de sincronizare înseamnă primirea unui mesaj valid). Comportamentul este același ca în scenariul 1, cu excepția faptului că, la fiecare ciclu, va avea loc un moment de intersectare între intervalele de timp de livrare pentru aplicația Broker, însă mecanismul de modificare dinamică a intervalului Delay în cazul mesajelor invalide va resincroniza brokerul și receptorul. Între mesajele valide va exista întotdeauna recurența dorită, se evită interogarea rețelei (starea de polling) din momentul în care se face prima sincronizare (ceea ce crește eficiența), iar operațiunea de resincronizare va avea loc la fiecare ciclu fără a produce pierderi de mesaje.

În ambele scenarii, algoritmul de sincronizare implementat se dovedește a fi eficient, iar testarea implementării a confirmat faptul că, comportamentul vizat este prezent atât la aplicația Broker cât și la cele două entități Subscriber.

În acest stadiu, scopul codificării datelor la nivelul aplicației Broker poate fi cu adevărat observat. În cazul adoptării unei soluții universale de tip Broker pentru transmisiunile UDP între OPC UA Publishers și Subscribers, procesul de codificare a datelor de interes se poate dovedi decisiv pentru procesul de sincronizare.

În cazul în care aplicația Broker este închisă, indiferent de motiv, conform algoritmului de sincronizare, receptorii vor intra în starea de polling și vor executa procedura de recepție a datelor până când va avea loc un eveniment de sincronizare (în cazul algoritmului, evenimentul de sincronizare înseamnă primirea unui mesaj valid), iar după ce se va realiza sincronizarea, dacă rețeaua este stabilă, starea de polling nu ar trebui să fie atinsă din nou.

3.4. Rezultate și Revendicări

Implementarea aplicației Broker alături de entități de tip Publisher și Subscriber, a fost realizată cu succes, dovedind că acest concept poate fi funcțional. Au fost atinse obiectivele majore din punct de vedere arhitectural, oferind avantaje concrete pentru comunicația OPC UA.

La momentul implementării studiului de caz anterior, SDK-ul [25] ce a fost utilizat, nu a oferit o versiune finală de API. Din punctul de vedere al autorului, soluția actuală pentru operațiunea de recepție la nivel de Subscriber este conformă cu specificațiile [24]. Cu toate acestea, cu un API final încă în curs de dezvoltare, chiar

dacă autorul nu a identificat nici un caz concret, există posibilități ca la nivel inferior din punct de vedere software (sub nivelul software utilizat pentru dezvoltarea conceptului Subscribe) să existe interogări de rețea sau utilizări de buffere care stochează toate mesajele de rețea ce trec prin adresa de multicast, care ar putea avea loc independent de nivelele de aplicație ale SDK-ului utilizat de către autor. Acesta este motivul pentru care starea de polling a fost definită de autor așa cum se explică în secțiunea anterioară, la nivelul aplicației specifice utilizării soluțiilor SDK actuale. Un obiectiv major a fost evitarea stării de polling (evitarea executării mai multor instrucțiuni decât este necesar, mai des decât este necesar, lucru imposibil fără sincronizare) astfel încât receptorul să primească doar mesajele de interes, aceasta fiind o îmbunătățire adusă conceptului de recepție, față de cum este descris în specificații [24]: "Abonații trebuie să fie pregătiți să primească mesaje pe care nu le înțeleg sau care sunt irelevante. Fiecare mesaj de rețea furnizează date necriptate în antetul mesajului de rețea pentru a sprijini identificarea și filtrarea furnizorilor, a mesajelor DataSetMessages, a claselor DataSetClasses sau a altor conținuturi relevante ale mesajelor". De asemenea, implementarea curentă diferă de cea din capitolul anterior (ce folosește design-ul din specificații), prin aplicația Broker și prin algoritmul de sincronizare ca părți arhitecturale și funcționale ale conceptului OPC UA Publish-Subscribe pe UDP, acestea reprezentând o alternativă proprie cu beneficii concrete.

În urma fazei de implementare, se pot observa o serie de rezultate și revendicări, din mai multe perspective:

1. Entitatea Publisher este implementată într-un mod mai abstract decât în cazul obișnuit, transmițând informații pentru 2 receptori cu preferințe și așteptări temporale diferite, utilizând doar o configurație de bază prin intermediul unui singur Dataset publicat. Acest lucru poate fi văzut ca un avantaj pentru transmiterea unor cantități mai mari de date cu mai puțin efort.
2. Aplicația Broker folosește multithreading, deci este rapidă, iar cele 2 subcomponente funcționează independent. Este viabilă pentru operațiunile de complexitate ridicată în cazul aplicațiilor la scară largă.
3. Aplicația Broker stochează cantități de date ce trec de la furnizor către receptori. Stocarea de date (procesul data buffering) poate fi de interes în multe cazuri ce implică protocolul OPC UA.
4. Aplicația Broker poate publica informațiile (ultimele date viabile stocate) dacă este necesar, chiar și în cazul în care furnizorul se închide, servind drept server de rezervă. În acest fel, unele dintre rolurile entităților implicate sunt preluate de alte entități (în acest caz, rolul de publicare), ceea ce face ca sistemul să fie mai bine pregătit în caz de defecțiune. De asemenea, pot fi implementate măsuri de siguranță la nivelul aplicației Broker pentru a informa consumatorii de date cu privire la o funcționare defectuoasă a furnizorului.
5. Aplicația Broker furnizează date la intervalele de timp dorite, iar prin utilizarea strategiei de codificare și a algoritmului de sincronizare, receptorii primesc datele la o rată stabilă (nu mai repede decât este necesar) într-un mod eficient.
6. Receptorii sunt total decuplați de sursa de informații, fără a cunoaște niciun detaliu de la furnizorul principal. Acest lucru îmbunătățește conceptul de Publish-Subscribe față de cum este descris în [24]: "Publisher-ii și Subscriber-ii sunt slab cuplați. Adesea, ei nici măcar nu se cunosc între ei. Relația lor

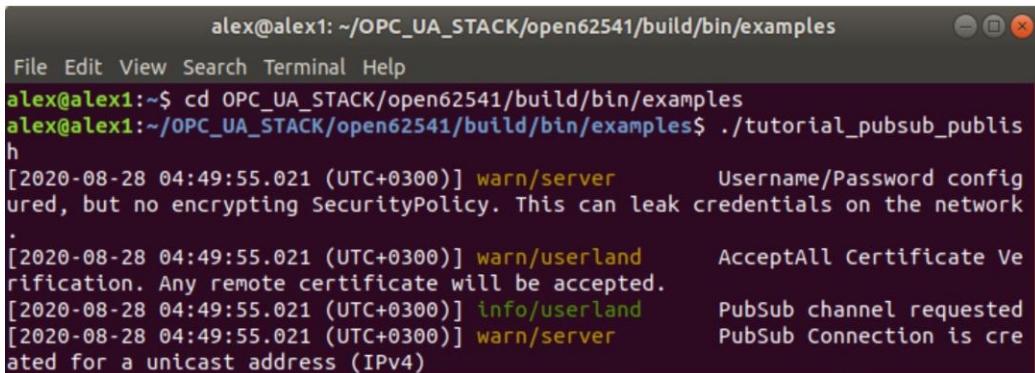
66 | 3. Îmbunătățirea mecanismului OPC UA Publish-Subscribe în contextul eficientizării comunicației în timp real

principală este înțelegerea comună a unor tipuri specifice de date (DataSet-uri), și caracteristicile de publicare ale mesajelor ce includ aceste date".

7. Algoritmul de sincronizare permite receptorilor să evite, pe cât posibil, starea de interogare a rețelei (starea de polling) și operațiunile de filtrare de diferite tipuri. Acest lucru se manifestă printr-o eficiență sporită și o diminuare a efortului de calcul și a utilizării resurselor pentru dispozitivele gazdă pe care rulează entitățile Subscriber.
8. Algoritmul de sincronizare asigură receptorii că datele vor fi distribuite la momentul potrivit, evitând, în cazuri normale, starea de interogare a rețelei (starea de polling), asigurând, de asemenea, posibilități de resincronizare. Acest lucru conferă importanță modului în care sunt publicate datele (la un anumit moment de timp), astfel încât, prin sincronizarea aplicației Broker și a entității Subscriber, cerințele de funcționare în timp real sunt îndeplinite, iar soluția poate fi destinată scenariilor de comunicație între dispozitive cu resurse limitate.
9. Implementarea actuală este diferită de brokerul folosit pentru protocoalele de transport AMQP și MQTT, descris în specificații [24]. În designul actual cu aplicație de tip Broker, toate entitățile implicate au o înțelegere comună a tipurilor de date OPC UA, toate aceste componente ale lanțului de transmisie fiind aplicații OPC UA.

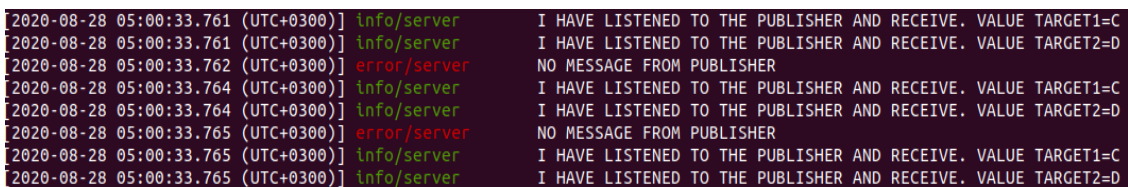
Pe baza implementării arhitecturii descrise în Figurile 14 și 15, rezultatele menționate mai sus pot fi observate în Figurile 17-20, fiecare figură ilustrând terminalul unei entități implicate în procesul de comunicație: Publisher, aplicația Broker, Subscriber 1 și Subscriber 2.

```
UA_UInt64 publishValue =0xDC; //INFORMATION 0xDC; D -target Subs2 // C -target Subs1
```



```
alex@alex1: ~/OPC_UA_STACK/open62541/build/bin/examples
File Edit View Search Terminal Help
alex@alex1:~$ cd OPC_UA_STACK/open62541/build/bin/examples
alex@alex1:~/OPC_UA_STACK/open62541/build/bin/examples$ ./tutorial_pubsub_publish
[2020-08-28 04:49:55.021 (UTC+0300)] warn/server Username/Password configured, but no encrypting SecurityPolicy. This can leak credentials on the network
[2020-08-28 04:49:55.021 (UTC+0300)] warn/userland AcceptAll Certificate Verification. Any remote certificate will be accepted.
[2020-08-28 04:49:55.021 (UTC+0300)] info/userland PubSub channel requested
[2020-08-28 04:49:55.021 (UTC+0300)] warn/server PubSub Connection is created for a unicast address (IPv4)
```

Figura 17. Terminalul aplicației Publisher și datele trimise către aplicația Broker



```
[2020-08-28 05:00:33.761 (UTC+0300)] info/server I HAVE LISTENED TO THE PUBLISHER AND RECEIVE. VALUE TARGET1=C
[2020-08-28 05:00:33.761 (UTC+0300)] info/server I HAVE LISTENED TO THE PUBLISHER AND RECEIVE. VALUE TARGET2=D
[2020-08-28 05:00:33.762 (UTC+0300)] error/server NO MESSAGE FROM PUBLISHER
[2020-08-28 05:00:33.764 (UTC+0300)] info/server I HAVE LISTENED TO THE PUBLISHER AND RECEIVE. VALUE TARGET1=C
[2020-08-28 05:00:33.764 (UTC+0300)] info/server I HAVE LISTENED TO THE PUBLISHER AND RECEIVE. VALUE TARGET2=D
[2020-08-28 05:00:33.765 (UTC+0300)] error/server NO MESSAGE FROM PUBLISHER
[2020-08-28 05:00:33.765 (UTC+0300)] info/server I HAVE LISTENED TO THE PUBLISHER AND RECEIVE. VALUE TARGET1=C
[2020-08-28 05:00:33.765 (UTC+0300)] info/server I HAVE LISTENED TO THE PUBLISHER AND RECEIVE. VALUE TARGET2=D
```

Figura 18. Terminalul aplicației Broker recepționând date de la Publisher și transmitându-le receptorilor

68 | 3. Îmbunătățirea mecanismului OPC UA Publish-Subscribe în contextul eficientizării comunicației în timp real

Tabel 2: Analiza rezultatelor studiului de caz

Entitate	Avantaje	Dezavantaje	Realizări
OPC UA Publisher	<ul style="list-style-type: none"> dificultate moderată în implementare configurație ușoară pentru toate entitățile Subscriber ce au necesități diferite decuplaj total față de receptori 		<ul style="list-style-type: none"> o modalitate ușoară de a trimite cantități mari de date pentru mai multe entități Subscriber cu necesități diferite
Broker App	<ul style="list-style-type: none"> capabilități multithreading capabilități de funcționare în timp real 	<ul style="list-style-type: none"> complexitate mare în implementare este necesar un prim pas pentru obținerea cerințelor și ID-urile de la entitățile Subscriber (în implementarea actuală, informația este hard-codată) 	<ul style="list-style-type: none"> comportament în timp real și sincronizare cu entitățile Subscriber Stocare de date (data buffering) rol de furnizor de rezervă capabilități de siguranță în cazul opririi furnizorului
OPC UA Subscribers	<ul style="list-style-type: none"> dificultate mică/moderată în implementare decuplaj total față de furnizorul de date capabilități de sincronizare bazate pe algoritmul de sincronizare descris 	<ul style="list-style-type: none"> este necesar un prim pas pentru transmiterea cerințelor și ID-urile (în implementarea actuală, informația este hard-codată) 	<ul style="list-style-type: none"> comportament în timp real și sincronizare cu aplicația Broker se evită procesul de interogare a rețelei se evită filtrarea în scopul identificării mesajelor de interes

3.5. Concluzii

În paragrafele următoare sunt discutate provocările acestei etape de cercetare și contextul de aplicabilitate al conceptului implementat.

Un prim obstacol în faza de implementare a aplicației Broker a fost acela de a identifica care sunt punctele forte și punctele slabe ale designului actual și cum pot

fi aplicate punctele forte pentru a îmbunătăți în continuare comportamentul în direcția unei reactivități în timp real și a unei eficiențe sporite a entităților de tip Publisher și Subscriber.

Una dintre provocările majore ale studiului actual a fost identificarea mecanismului de sincronizare între Subscriber și aplicația Broker. Pentru atingerea acestui obiectiv, au fost luate în considerare numeroase scenarii în legătură cu diferitele stări posibile ale receptorului în toate momentele de timp, în ce condiții receptorul ar trebui să înceapă să scadă și să se mărească intervalul Delay, cum poate fi evitată pierderea mesajelor chiar dacă aplicația Broker și receptorul nu împart aceeași bază temporală. O altă mare provocare a fost reprezentată de găsirea unei metode de codificare și decodificare, care să fie rapidă și simplă pentru a economisi efort și timp atât pentru aplicația Broker, cât și pentru receptor, și care să ofere o modalitate clară de a distinge mesajul dorit în momentele de timp în care componenta Publisher a aplicației Broker va trimite mai multe mesaje simultan. S-a urmărit ca receptorul Subscriber să fie menținut cât mai simplu posibil, o entitate care doar primește și consumă mesajele relevante din rețea, doar la momentul potrivit, astfel încât filtrarea bazată pe diversele subcomponente specifice entității Publisher să fie evitată, menținând un design complet decuplat și păstrând aceeași funcționalitate, cu o cantitate mai mică de informații necesare din partea oricărei entități implicate.

Alte provocări au fost determinate de identificarea arhitecturii potrivite pentru aplicația Broker, astfel încât ambele componente să funcționeze independent și eficient, și de asemenea de implementarea furnizorului și a receptorilor cu un design cât mai simplu, menținând toate operațiunile importante la nivelul aplicației Broker. Testarea conceptului a fost dificilă, fiind necesară utilizarea unei versiuni incrementale de software pentru toate entitățile, evoluția fiecărei versiuni influențând modul în care reacționează celelalte. Având mai multe dispozitive implicate, testarea rezultatelor specifice diferitelor versiuni ale fiecărui software s-a dovedit a fi o sarcină dificilă și a fost implementat cod de testare suplimentar în diverse etape ale fazei de dezvoltare.

Abordarea conceptuală actuală are diverse contexte de aplicabilitate directă. În cazul controlerului parametric senzorial care expune datele senzoriale colectate sau al controlerelor I/O utilizate pentru a colecta date de la dispozitivele de teren, pentru a comunica cu PLC-uri sau cu PLC-uri redundante, conceptul propus s-ar potrivi și ar avantaja mecanismul de Publish-Subscribe OPC UA din perspectiva comunicației Ethernet în rețea.

Sincronizarea și viteza sunt esențiale în cazul cobots (roboți colaborativi), în industria prelucrătoare, aceștia trebuie să se alinieze la cerințele Industry 4.0 în ceea ce privește protocolul de comunicare utilizat. Din perspectiva stadiului industrial, la diferite companii care utilizează roboți, provocările integrării roboților în procesul de producție constă în implementarea interfațării. De obicei, în interiorul celulelor de producție cobots, cea mai bună dezvoltare funcțională întâlnită în cazul cerințelor de viteză a fost eliminarea oricărei interfețe industriale și implementarea unei interfețe simple bazate pe șiruri de caractere pentru a comunica cu structura locală de monitorizare din interiorul celulei. Dezvoltarea emblematică [46] pentru roboții universalii ar putea fi îmbunătățită cu ajutorul conceptului actual.

Industria prelucrătoare în general, dar și alte industrii ar putea beneficia de utilizarea conceptului actual. Luând în considerare un exemplu de testare și ambalare la sfârșitul liniei de fabricație a automobilelor, unde o mașină realizează inspecția optică automatizată pentru pinii ECU [47], după inserarea robotizată a pinilor pe plăci, fixarea și etichetarea carcasei și înainte de transportul și ambalarea în cutii și inspecția optică finală de numărare a plăcilor din cutii, pentru a fi trimise la companiile client,

70 | 3. Îmbunătățirea mecanismului OPC UA Publish-Subscribe în contextul eficientizării comunicației în timp real

structura de inspecție optică automată (AOI) trebuie să finalizeze întreaga procesare a imaginii în câteva secunde și trebuie să comunice cu PLC-ul de pe același dispozitiv, care fixează carcasa, rotește structura și trimite ECU pentru ambalare sau pentru eliminarea pieselor. În această situație, inspecția optică ar trebui să trimită date către PLC-ul de pe dispozitiv, către sistemele de control din upstream și din downstream, la sistemele de execuție a producției (MES), pentru a se obține un proces de funcționare corect, coordonat, sincronizat și sigur. Soluția prezentată ar contribui în mod semnificativ la automatizarea completă a producției și testării la sfârșitul liniei de fabricație și ar reduce influența MES în funcționarea întregului proces.

Implementarea aplicației Broker alături de algoritmul de sincronizare, prezintă capacitățile protocolului OPC UA în cazuri de utilizare cu cerințe de funcționare în timp real și oferă abordări care îmbunătățesc mecanismul OPC UA Publish-Subscribe în direcția unei eficiențe sporite și a unui răspuns în timp real pentru toate entitățile implicate în schimbul de date, extinzând zona posibilităților de dezvoltare pentru aplicațiile specifice contextului IIoT.

4. ABORDAREA MECANISMULUI OPC UA PUBLISH-SUBSCRIBE ÎN CONTEXTUL TRANSMITERII DE IMAGINI

Protocolul Open Platform Communication Unified Architecture (OPC UA) este un facilitator cheie pentru Industry 4.0 și al Internetului industrial al lucrurilor (IIoT). OPC UA este deja validat de către industrie și se așteaptă ca prezența sa să ajungă în tot mai multe domenii, aplicații și niveluri ierarhice. Progresele din cadrul celor mai recente specificații oferă posibilitatea de a extinde capacitățile și aplicabilitatea protocolului, vizând performanțe mai bune în ceea ce privește volumele de date, viteza de transmisie, disponibilitate și securitate. Continuând cercetările anterioare care se concentrează pe mecanismul Publish-Subscribe și pe constrângerile de funcționare în timp real, studiul actual își propune să ia în considerare volume de date mai mari, să abordeze comunicarea bazată pe protocolul de transport UDP (User Datagram Protocol) utilizând mai multe canale și să analizeze robustețea mecanismului dezvoltat în contextul transmiterii de date pe termen lung. În consecință, etapa de cercetare curentă propune extinderea aplicabilității OPC UA în contextul transmiterii de imagini. Deși foarte necesară, transmiterea imaginilor după procesare este în prezent în afara domeniului de aplicabilitate al OPC UA sau al altor protocoale industriale tradiționale, fiind considerată o fracțiune separată în mediul industrial. Conceptul și dezvoltările actuale sunt aplicate luând în considerare atât procesul de fabricație industrială la sfârșitul liniei de producție în sectorul automotive, cât și comunicarea între autovehicule și infrastructură. Fără constrângeri hardware speciale, rezultatele obținute se dovedesc a fi apreciable, deschizând diverse perspective de viitor pentru transmiterea de imagini cu ajutorul OPC UA.

4.1. Noțiuni generale și Aplicabilitate în Industrie

Industria de prelucrare evoluează continuu către o perspectivă completă a Industry 4.0. Această evoluție se desfășoară în două direcții - cercetare și implementare practică pe linia de producție. Cercetarea își urmează tendința firească, cu abordări teoretice și practice, concentrându-se în principal pe acțiuni inovatoare în ritm alert, în cadrul IIoT. Dezvoltarea industrială practică este întotdeauna încetinită de echipamentele, soluțiile și procedurile tradiționale, dar, în același timp, este întotdeauna împinsă înainte de noile entități și servicii bazate pe necesități, dezvoltate rapid, care, cel puțin pe termen scurt, răspund nevoilor de evoluție. Aceste noi entități și servicii nu sunt întotdeauna cele mai bune opțiuni sau cele mai complete și adesea nu au o viziune de ansamblu în ceea ce privește statutul cercetării. Cu toate acestea, ele reprezintă îmbunătățiri clare în ceea ce privește statutul actual în industria de producție. Aceste exemple sunt larg răspândite în întreaga industrie și cronologie, iar evoluțiile în ritm alert din ultimii ani evidențiază tot mai multe situații de acest fel. Nevoia de integrare a datelor și lipsa unei imagini complete a IIoT facilitează adesea o tradiție a protocoalelor locale care sunt integrate utilizând serverele centralizatoare

72 | 4. Abordarea mecanismului OPC UA Publish-Subscribe în contextul transmiterii de imagini

Open Platform Communication (OPC) fără a lua în considerare interfațarea directă cu Open Platform Communication Unified Architecture (OPC UA) sau cu wrapperele OPC UA locale [48,49]. Nevoia de alinierea roboților colaborativi (cobots) la cerințele funcționale, conduce adesea la o dezvoltare rapidă de soluții în cadrul companiilor, soluții ce sunt complet lipsite de protocol industrial, robustețe, compatibilitate și securitate (de exemplu, strategia de comunicare particulară de bază dezvoltată în cadrul unor echipe mici de cercetare și dezvoltare din cadrul companiilor). Această abordare nu ia în considerare posibilitatea OPC UA în cadrul cobots [46]. Strategiile de colectare a datelor la nivel de tehnologie operațională sunt, de obicei, implementate ca soluții clasice de tip istoric (historian) cu diverse soluții supraestimate, dar cu capacitate mică, fără a lua în considerare posibilitatea unui istoric proactiv (proactive historian, de exemplu, [50]). Integrarea datelor de proces la nivelul tehnologiei informației este privită, în cea mai mare parte, ca un algoritm rudimentar repetitiv de export-import de fișiere sau, în cel mai bun caz, ca un transfer de stare reprezentatională (REST), în timp ce un protocol industrial evoluat ar oferi beneficii clare. Pentru cele mai bune soluții, cercetarea trebuie să se îndrepte mai clar și mai rapid către niveluri mai ridicate de pregătire tehnologică (TRL), respectiv trebuie să se consolideze legătura cu industria.

Prelucrarea imaginilor în industrie a apărut din necesitate și a fost adoptată fără a lua în considerare o legătură completă a rezultatelor cu protocoalele de comunicare industriale și cu procesul de producție. Starea generală actuală a comunicării între entități este reprezentată de un sistem de execuție a fabricației (MES) bazat pe executarea procedurilor de tip cerere-aprobare, iar stocarea și transportul imaginilor se realizează fără un protocol industrial. Cu toate acestea, prelucrarea imaginilor în sine este strâns legată de liniile de producție. În [53], cercetarea a prezentat o diagnoză a sistemului hidraulic de pompă axială utilizând conversia semnalelor în imagini prin intermediul transformării continue wavelet, respectiv extragerea caracteristicii din imaginile transformate timp-frecvență. În [54], autorii au abordat o soluție de procesare a imaginilor și învățare aprofundată pentru a detecta deformarea pe banda de contact a pantografului vehiculelor feroviare. În [47], autorii au prezentat o soluție low-cost de procesare a imaginilor bazată pe Open-Source Computer Vision Library (OpenCV), utilizată pentru a detecta defectele de fabricare a pieselor auto, în special defecte precum pini defecti, lipsă sau în plus, fisuri ale plăcii și ale unității de control electronic (ECU). Majoritatea studiilor consideră procesul industrial și transmiterea de imagini ca fracțiuni separate. În domeniul realității augmentate, studiul [55] a propus Node-RED și Message Queuing Telemetry Transport (MQTT) pentru a comunica cu dispozitivele mecatronice, dar aspectele legate de imagine se concentrează asupra dispozitivului mobil în cadrul aplicației iOS.

Pe lângă procesul de producție propriu-zis, industria automotive se concentrează pe noile concepte precum condusul autonom și îmbunătățit. Aceste concepte sunt strâns legate de procesarea de imagini, comunicația car-to-infrastructură, comunicația car-to-car și dezvoltarea de proceduri de siguranță. S-au înregistrat progrese mari în legătură cu tehnicile de procesare a imaginilor ca aspect esențial al conducerii autonome [56]. În plus, prelucrarea imaginilor este studiată în mod eficient pentru a îmbunătăți conducerea și siguranța, și pentru a detecta daunele (fisurile) din infrastructură, așa cum este descris în [57]. Detectarea accidentelor pe baza imaginilor din infrastructură a fost luată în considerare în [58]. În acest sens, dispozitivele inteligente de pe marginea drumului preiau și procesează imaginile din trafic folosind YOLO-CA (un model de învățare aprofundată YOLO - You Only Look

Once - pentru accidente auto), iar rezultatele sunt prevăzute a fi trimise către un sistem central în scopuri de salvare și semnalizare elementară pentru mașinile din trafic. În [59], autorii au prezentat o estimare a lungimii cozii de așteptare bazată pe procesarea imaginilor în contextul comunicației car-to-Infrastructură. Rezultatele au fost prezentate folosind doar simulări. Evident, comunicația car-to-Infrastructură, în contextul interoperabilității orizontale și verticale, necesită un protocol de comunicare industrial, iar lucrări precum [45] și [23] au detaliat comunicația de tipul car-to-Infrastructură bazată pe OPC UA.

Abordarea conceptuală și aplicabilitatea strategiei de comunicare UDP pe mai multe canale, alături de rezultatele și observațiile multiplelor experimente cu mecanismul Publish-Subscribe, în ceea ce privește capacitatea și stabilitatea, se poate dovedi utilă în viitoarele integrări ale protocolului OPC UA în diferite arhitecturi.

Lucrări precum [51] analizează conformitatea standardelor industriale prin intermediul OPC UA. Cele mai recente specificații [24] permit trecerea la nivelul următor cu protocolul OPC UA, prin abordarea mecanismului Publish-Subscribe. În acest fel, capacitățile protocolului pot fi extinse și se pot obține performanțe mai bune în ceea ce privește viteza de transmisie, volumele de date, disponibilitatea și securitatea. Mecanismul Publish-Subscribe OPC UA este dezvoltat cu succes în [45,52], permițând extinderea aplicabilității protocolului. Cu toate acestea, cercetarea trebuie să evolueze și să ofere o soluție OPC UA Publish-Subscribe bazată pe transmisia de imagini, care ar permite integrarea completă a protocolului în industrie. În plus, soluția OPC UA Publish-Subscribe din cercetările anterioare trebuie să își confirme robustețea în contextul transmiterii de date pe termen lung.

În contextul menționat mai sus, obiectivele actualei etape de cercetare sunt următoarele:

- Continuarea cercetărilor anterioare [45,52] bazate pe mecanismul OPC UA Publish-Subscribe și funcționarea în timp real, considerând analiza transmisiunilor de volume de date mai mari, respectiv analiza pe o durată lungă de timp a unei singure transmisii.
- Abordarea strategiei de comunicare UDP pe mai multe canale în contextul OPC UA Publish-Subscribe urmărind reducerea duratelor de transmisie.
- Extinderea aplicabilității protocolului OPC UA în contextul transmiterii de imagini.
- Aplicarea soluției implementate, fără constrângeri din punct de vedere hardware, în studii de caz referitoare atât la producția industrială din domeniul automotive, cât și la comunicația Car-to-Infrastructură.

4.2. Metode și Tehnologii utilizate

Secțiunea curentă se concentrează pe aplicabilitatea protocolului OPC UA și a mecanismului Publish-Subscribe, în industrie, luând în considerare diverse tipuri de cerințe industriale. De asemenea, se urmărește identificarea de posibilele avantaje obținute în urma integrării conceptului cu alte tehnologii în diverse cazuri de utilizare. Autorul propune o aplicație pentru transmiterea de imagini care utilizează mecanismul OPC UA Publish-Subscribe în cadrul comunicațiilor multicanal, în timp real.

4.2.1. OPC UA Publish-Subscribe în contextul Cercetării și Industriei

OPC UA este un protocol adoptat și validat în industrie, fiind utilizat la nivel global, oferind o fiabilitate și o interoperabilitate ridicată pentru diverse tipuri de sisteme. Odată cu noile specificații [24], paradigma Publish-Subscribe a devenit accesibilă în cadrul OPC UA, extinzând capacitățile protocolului și domeniile de aplicabilitate. Un set mai complex de cerințe a putut să fie atins, scenarii ce vizează comunicarea între dispozitive cu resurse limitate, precum și alte operațiuni cu constrângeri de timp au fost realizate. Literatura de specialitate prezintă câteva studii de cercetare consistente cu privire la conceptul de OPC UA Publish-Subscribe. Autorii din [45] au prezentat o cercetare care a analizat cerințele de timp real din diferite perspective în contextul mecanismului de publicare-subscriere OPC UA, iar conceptul care vizează multiple domenii a fost implementat cu succes. Alte studii au vizat explorarea mecanismului de Publish-Subscribe OPC UA în comparație cu paradigme similare din multiple protocoale de comunicare utilizate în alte domenii, iar îmbunătățiri și perspective noi au fost sugerate pentru cazuri de utilizare industrială [52]. În ceea ce privește dispozitivele hardware utilizate în cercetare și în industrie, pentru implementarea și testarea aplicațiilor bazate pe mecanismul de publicare-subscriere OPC UA, se poate observa utilizarea a diverse tipuri de dispozitive. În [60], mai multe dispozitive Xilinx au fost utilizate pentru rularea unui Publisher și a unui Subscriber care schimbă mesaje UADP (UA Datagram Protocol) între ele, efectuându-se o analiză a timpului de execuție a operațiunilor. În [61], pentru schimbul de informații, entitățile OPC UA au fost implementate pe Raspberry PI Model 3B+, utilizându-se diverse stive open source, și fiind efectuate măsurători de eficiență. În [52] au fost implementate mai multe entități de tipul Publisher și Subscriber, fiind testate atât pe dispozitive cu resurse mari cât și pe dispozitive încorporate, cu obiectivul de a realiza sincronizarea între operațiunile de transmisie și recepție ale protocolului OPC UA. Conform lucrării [45], tehnologia TSN (time-sensitive networking) reprezintă o soluție pentru schimbul de informații critice pe Ethernet, din punct de vedere al funcționării în timp real, fiind utilizată atât pentru comunicarea între senzori și cloud, cât și pentru comunicarea între controlere.

TSN este specific nivelului de legătură de date (Data Link layer), iar prin intermediul unui set de standarde particulare, oferă garanții de timp pentru anumite operațiuni din rețea. După cum se menționează în [62] și [63], OPC UA și TSN sunt prevăzute colaborând și la nivelul dispozitivelor de teren, cu resurse limitate. Studiul [64] a pus accentul pe problema funcționării în timp real în context industrial, și pe aplicabilitatea protocolului asupra dispozitivelor de teren, oferind doar simulări. În cazul mecanismului Publish-Subscribe OPC UA, integrarea tehnologiei TSN reprezintă următorul pas în obținerea unui comportament în timp real, a unei eficiențe sporite și a îmbunătățirii calității serviciilor. Au fost efectuate anumite studii cu privire la interacționarea dintre OPC UA și TSN, iar rezultatele dovedesc că sincronizarea în timp, timpii de latență mici și flexibilitatea pot fi obținute prin adoptarea de standarde diferite față de mecanismul bine-cunoscut care este prezent în industrie [60,65].

Având în vedere că specificațiile [24] sunt încă recente, există direcții care pot fi explorate în scopul creșterii eficienței și fiabilității conceptului și pot fi dezvoltate noi aplicații cu caracteristici speciale pentru a rezolva provocări specifice industriei. Odată cu creșterea numărului dispozitivelor de rețea, prezente în tot mai multe aplicații specifice IIoT, mulțimea de operațiuni ce trebuie efectuate este de asemenea

în creștere și, după cum se afirmă în [66], gestionarea și extragerea datelor din surse multiple și distribuite devine decisivă. Toate entitățile implicate în scenarii de comunicații, ce sunt prezente în astfel de arhitecturi complexe, trebuie să evolueze astfel încât performanțele să răspundă așteptărilor în materie de fiabilitate, robustețe și eficiență. Mecanisme precum OPC UA Publish-Subscribe sporesc capacitățile sistemelor complexe, însă, în aplicațiile industriale sau în cercetare, există o proiecție către un set mai variat de funcționalități și cerințe pentru orice caz de utilizare în contextul IIoT. În plus, toate tehnologiile implicate trebuie analizate în multiple scenarii. Este logic să ne gândim la noi scopuri și cazuri de utilizare pentru a rezolva sau îmbunătăți soluțiile existente pentru anumite servicii. În acest context, autorul a ales un scenariu fezabil pentru a analiza capacitățile mecanismului Publish-Subscribe din diferite perspective, și a fost dezvoltată o aplicație de transmitere a imaginilor bazată pe protocolul OPC UA.

4.2.2. Conceptul transmiterii de imagini utilizând mecanismul OPC UA Publish-Subscribe

Transmiterea și prelucrarea de imagini sunt concepte din ce în ce mai prezente în aplicațiile IIoT. Cu capacități în creștere pe Ethernet, se așteaptă ca aceste concepte să devină parte a proceselor existente în fabricile inteligente, în industria de producție și în alte domenii precum cel automotive sau aeronautic. Cu OPC UA ca verigă cheie în interoperabilitate și având deja integrate mai multe tehnologii la diferite niveluri OSI, noi caracteristici, cum ar fi transmiterea de imagini, s-ar putea dovedi utile în anumite scenarii.

Prelucrarea imaginilor are un rol important în dezvoltarea aplicațiilor de siguranță în industria automotive și în cazul sistemelor inteligente de trafic. Având în vedere că, în prezent, conceptul de conducere autonomă prezintă un mare interes, majoritatea producătorilor de automobile implementează module bazate pe procesarea imaginilor. Capacitățile de comunicare între autovehicule și în interiorul acestora, alături de comunicarea între vehicule și infrastructură, evoluează într-un ritm alert. Atât în cazul infrastructurii inteligente, cât și în cazul automobilelor inteligente, detectarea pietonilor și schimbul rapid de informații prin Ethernet sunt concepte vitale. Au fost realizate studii pentru integrarea OPC UA ca protocol cu potențial pentru infrastructuri inteligente, în context comun cu alte protocoale de comunicare specifice domeniului automotive, fiind dezvoltate soluții care ar putea ruta informațiile prin mecanisme de publicare-subscriere, inclusiv OPC UA Publish-Subscribe [45,23]. Potențialul de extindere al aplicabilității OPC UA este de așteptat să determine pe viitor utilizarea protocolului inclusiv în module de siguranță ce se bazează pe procesarea și transmiterea imaginilor.

În industria producției de automobile, prelucrarea imaginilor este utilizată în mod frecvent. Conform exemplificării din studiul [47], în timpul fabricării plăcilor ECU, inspecția optică automată (AOI) se realizează la sfârșitul liniei de testare și ambalare. Referitor strict la procesele de testare și ambalare la sfârșitul liniei de producție, zeci de mii de produse sunt testate zilnic într-o companie, folosind procesarea imaginilor pentru detecția a diverse defecte, iar procesul de ambalare este, de asemenea, asigurat folosind procesarea imaginilor.

Rezultatele soluțiilor de prelucrare a imaginilor includ și transmiterea de imagini. Deoarece rezultatele bazate pe biți/etichete sunt integrate în comunicarea MES, de obicei fără niciun protocol industrial, imaginile încheiate și raportate local

76 | 4. Abordarea mecanismului OPC UA Publish-Subscribe în contextul transmiterii de imagini

sunt transferate din când în când într-un mod rudimentar între diverse entități. Un proces de sfârșit de linie de producție este prezentat în Figura 21, în care prelucrarea imaginilor este utilizată la sfârșitul liniei de testare și ambalare în producția de piese auto.

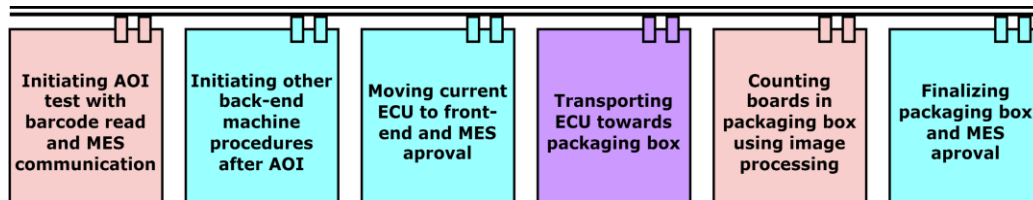


Figura 21. Testarea unității ECU la sfârșitul liniei de producție utilizând procesarea imaginilor în producția de automobile

După ce pini sunt introduși, iar ECU este plasat în poziția corespunzătoare, ECU-urile sunt testate cu ajutorul AOI pentru a detecta defecte la pini (de exemplu, îndoșiți, plasați greșit, lipsă sau pini în plus), la cleme (plasare necorespunzătoare sau cleme rupte) sau fisuri ale plăcii. Soluția AOI comunică cu MES, solicitând aprobarea pentru a începe testarea și transmiterea unui rezultat atât în cazul în care placa este defectă cât și în cazul în care placa a trecut inspecția. După efectuarea AOI, cea mai bună soluție în concordanță cu conceptele Industry 4.0 ar fi integrarea protocolului OPC UA pentru transportul imaginii, ca parte a soluției finale de procesare a imaginilor. Cu toate acestea, OPC UA ar trebui să asigure de asemenea cerințele de funcționare în timp real, viteză de transmisie și stabilitate în cazul unui volum mare de date. Pe baza cercetărilor [45] și [52], protocolul ar putea asigura o parte din cerințele de funcționare în timp real prin utilizarea mecanismului Publish-Subscribe. Scopul conceptului actual este de a crește nivelul utilizării mecanismului OPC UA Publish-Subscribe comparativ cu studiile de caz anterioare, astfel încât să se fie fezabil pe termen mai lung, cu volume mai mari de date, oferind transmisii cât mai rapide și eficiente. Astfel de cerințe sunt absolut necesare pentru transmisiunile de imagini în contextul unei integrări complete în procesul industrial. În urma unei asemenea integrări, se poate obține o interoperabilitate verticală și orizontală completă prin OPC UA. Așa cum este descris în Figura 21, plăcile sunt transportate ulterior și introduse în cutiile de ambalare. Numărarea ECU-urilor din interiorul cutiilor de ambalare, se realizează, de asemenea, cu ajutorul procesării imaginilor. După ce se ajunge la capacitatea maximă a unei cutii, înainte de a o închide și de a o trimite clientului, se stochează o imagine finală cu toate plăcile din interior. Aceeași soluție ce asigură interoperabilitatea OPC UA completă, ar fi necesară și în această situație, în care imaginile sunt, de obicei, mai mari, dar și intervalul de timp dintre transmisiile de imagini este mai mare.

Unul dintre cazurile de utilizare care necesită testat din punct de vedere al performanțelor, în raport cu aplicația OPC UA de transmitere a imaginilor, este eficiența procesării unei imagini și a extragerii obiectelor de mare interes de către o rețea neuronală și publicarea situației obiectelor către un Subscriber în timp real. Acest tip de caz de utilizare poate clarifica anumite limite de timp de livrare a unui set de informațiile minime și necesare, ce ar putea fi obținute pe baza unei imagini.

Pentru operațiunile de procesare a imaginilor ale conceptului actual, a fost utilizat modelul YOLO v3 [67]. YOLO implică o singură rețea neuronală convoluțională profundă (numită DarkNet). Pentru a reduce semnificativ orice detecție falsă, YOLO a

fost antrenat pe setul de date COCO, iar acest model a putut detecta obiecte aferente a 80 de clase diferite.

Pentru diferite operații specifice imaginii vizate pentru transmitere, Open Source Computer Vision Library (OpenCV) a fost utilizată - populară în industrie ca o soluție specializată în domeniul vederii computerizate în timp real. OpenCV este cross-platform, poate fi utilizată gratuit în cadrul licenței open-source Apache2 și este compatibilă cu Torch/PyTorch, TensorFlow și Caffe software. Domeniile de aplicare pentru OpenCV se focalizează pe recunoașterea facială, interacțiunea om-calculator, urmărirea mișcării, robotica mobilă și segmentarea imaginilor.

Aplicația de transmitere a imaginilor este formată din două instanțe diferite, corespondente unui Publisher OPC UA și a unui Subscriber OPC UA, și se axează pe trimiterea unei imagini între dispozitive separate prin intermediul mecanismului OPC UA Publish-Subscribe. Există trei versiuni ale aplicației, fiecare dintre ele vizează un studiu de caz diferit, obținând diverse performanțe de transmisie. Obiectivul principal constă în a trimite o imagine prin publicarea fiecărui pixel și reconstruirea imaginii la partea receptorului după ce entitatea Subscriber a obținut și stocat toți pixelii de la Publisher. Viteza transmiterii integrale și calitatea imaginii primite sunt principalii parametri care dovedesc stabilitatea, eficiența și fezabilitatea conceptului. Aplicația este testată în diferite scenarii și sunt trimise diverse imagini, fiecare cu particularități proprii în ceea ce privește rezoluția. Pentru toate cele trei versiuni ale aplicației, intervalul de publicare este setat la valori diferite, pentru a obține intervale de timp diferite pentru livrare și pentru a analiza modul în care viteza de transmisie ar putea influența calitatea imaginii. Entitatea Publisher și entitatea Subscriber sunt configurate pentru a schimba mesaje de tipul DataSetMessages (conform specificațiilor [24]), fiecare dintre acestea conținând valoarea unui pixel din formatul de imagine pe octeți, iar valorile recepționate sunt stocate fiecare sub forma unui număr întreg pe 8 biți cu un spectru de valori cuprins între 0 și 255. Mesajele de tipul DataSetMessages sunt create de către componenta DataSetWriter a Publisherului. Protocolul de transport utilizat pentru transmisia Publish-Subscribe este UDP. Conceptul se bazează pe patru etape principale pentru transmiterea completă a imaginii și poate fi observat în Figura 22. Durata transmisiunilor și a altor operațiuni specifice fiecărui scenariu au fost măsurate cu ajutorul unor funcții de testare special dezvoltate care utilizează temporizatoare Linux. Din perspectiva rețelei, conceptul a fost testat în rețele cu capacități diferite (2,4 GHz și 5 GHz). Având în vedere că mecanismul OPC UA Publish-Subscribe este orientat spre funcționarea în timp real, dar nu este capabil să ofere garanții de timp fără integrarea tehnologiei TSN, în cazul transmisiunilor rapide pe perioade lungi de timp, poate apărea o desincronizare între operațiunile de transmitere și de recepție ce rulează pe cele două dispozitive.

78 | 4. Abordarea mecanismului OPC UA Publish-Subscribe în contextul transmiterii de imagini

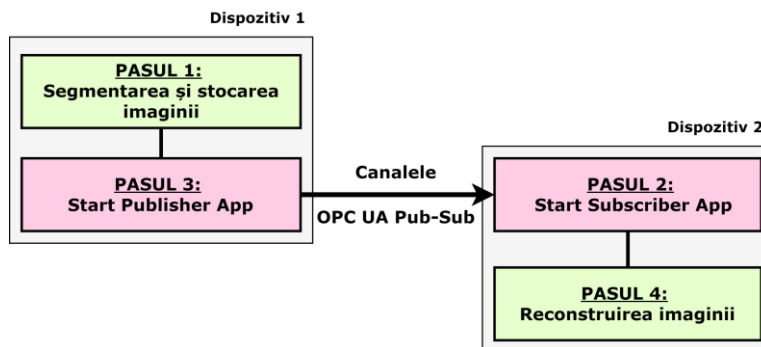


Figura 22. Etapele pentru transmisia de imagini prin mecanismul OPC UA Publish-Subscribe

Mecanismul Publish-Subscribe se axează în primul rând pe scenarii de comunicație de tip one-to-many, având însă capacitatea de a realiza cerințe de funcționare în timp real. În cazurile în care desincronizarea și latența se doresc a fi evitate, mai multe canale Publish-Subscribe complet separate pot reprezenta o soluție scalabilă pentru o serie de operațiuni industriale, care anterior nu puteau fi rezolvate într-un mod fezabil prin paradigma clasică server-client. În cazul canalelor multiple de transmisie, este importantă comunicarea eficientă între entități și au fost efectuate studii în direcția unor algoritmi care pot îmbunătăți schimbul de informații [68]. Cu toate acestea, în implementarea actuală, au fost dezvoltate canale independente pentru a minimiza timpul de transmisie, toate canalele transmițând la o recurență ridicată, obținându-se un comportament previzibil bazat pe volumul de date ce se dorește a fi transmis (valorile pixelilor unei anumite imagini). Securitatea în contextul IIoT reprezintă un subiect important în comunitatea științifică [69,70], iar din acest punct de vedere, OPC UA oferă mecanisme, la diferite niveluri ale modelului OSI, care au asigurat de-a lungul timpului securitatea și creșterea interoperabilității. Așadar, abordarea strategiei multi-canal în cazul protocolului OPC UA nu prezintă riscuri sporite de securitate, nici măcar în contextul transmiterii de imagini.

Pentru implementarea entităților OPC UA, autorul a luat în considerare mai multe stive OPC UA open-source disponibile, implementate în diferite limbaje de programare. Pe baza fiabilității confirmate în implementările anterioare (validate inclusiv în comunitatea științifică, a se vedea [45,52,27]) și pe baza studiilor privind capacitățile și eficiența stivei OPC UA (a se vedea [61]), a fost selectat SDK-ul open62541 [25] pentru implementarea aplicațiilor Publisher și Subscriber, de pe ambele dispozitive utilizate.

4.3. Arhitectura și Implementare

Aplicația OPC UA de transmitere a imaginilor este compusă din două părți - una cu rol de Publisher OPC UA, specifică dispozitivului care dorește să transmită imaginea, și una cu rol de Subscriber OPC UA, specifică dispozitivului destinat să recepționeze imaginea. Adicional celor două părți diferite, etapele suplimentare necesare înainte și după transmiterea propriuzisa a imaginii, cum ar fi segmentarea imaginii vizate și reconstrucția pe dispozitivul receptor, sunt de asemenea importante și stabilesc operațiuni diferite care trebuie să fie executate urmând o ordine specifică.

După cum s-a precizat în secțiunea anterioară (Figura 22), principalele patru etape ale aplicației sunt segmentarea și stocarea imaginii, pornirea aplicației Publisher, pornirea aplicației Subscriber și reconstrucția imaginii. Există diferite versiuni ale aplicației, asociate cu diferite studii de caz, fiecare cu configurații specifice. Prima etapă privind segmentarea și stocarea imaginii constă în împărțirea imaginii în părți aproximativ egale, fiecare parte având același număr de pixeli. Ulterior, se creează câte un buffer diferit pentru fiecare parte, accesibil de către entitatea Publisher în etapa a 3-a a procesului. Segmentarea imaginii se face pe baza numărului de canale Publish-Subscribe utilizate pentru transmisie, respectiv pe baza dimensiunii imaginii care urmează a fi transmisă. Etapa 4, reconstrucția imaginii în partea entității Subscriber, este un proces invers celui din etapa 1. Receptorul primește în același timp diverse părți ale imaginii pe diferite canale Publish-Subscribe și le stochează în propriile sale buffere. După ce transmisia este efectuată, aceste buffere sunt utilizate pentru a construi un fișier de pixeli specific imaginii, în care fiecare parte recepționată este plasată în ordinea corectă. În cazul în care transmisia a fost efectuată corect pentru toate canalele implicate și dacă valorile pixelilor au fost plasate în ordinea potrivită, imaginea poate fi reconstruită pe dispozitivul de recepție cu aceleași caracteristici ca și cea originală. Arhitectura sistemului este prezentată în Figura 23, împreună cu toate operațiile care au loc în timpul executării celor patru etape principale ale aplicației OPC UA de transmitere a imaginilor.

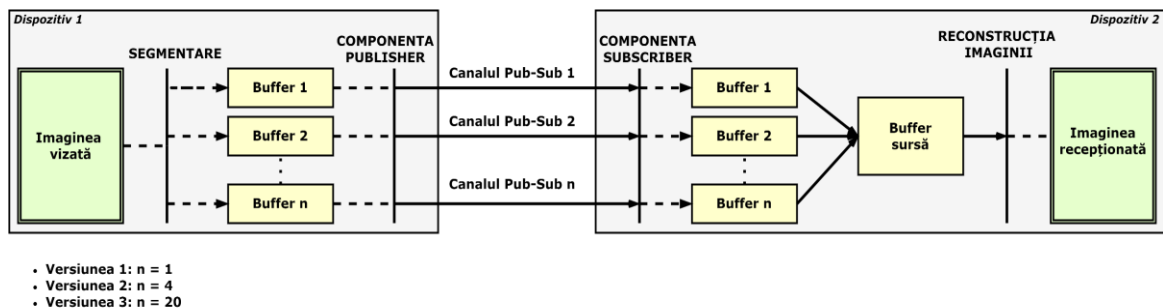


Figura 23. Arhitectura generală a sistemului

Procesul de publicare și procesul de recepție sunt sincronizate pentru a transmite și a recepționa valorile pixelilor la anumite intervale de timp. După cum se menționează în [52], în timp ce transmisia se face prin UDP, fără tehnologia TSN, dispozitivele nu sunt conectate la o bază de timp comună. Există două dispozitive care partajează informații, dar fiecare dintre ele este gestionat pe baza propriei referințe asupra timpului, având definite diferite intervale de timp pentru procedurile necesare. Acest scenariu ar putea afecta comportamentul dezirabil al aplicației. Când furnizorul livrează date la o recurență sub 10 milisecunde, fără opțiunea de a pierde unele dintre valori, fiecare valoare reprezentând un pixel trimis o singură dată, și fără un mecanism de prevenție în cazul pierderilor de mesaje, impactul asupra rezultatului final este influențat de robustețea metodelor folosite și de stabilitatea și capacitatea rețelei. Cu toate acestea, chiar dacă câțiva pixeli nu ar putea altera în mod decisiv calitatea imaginii recepționate, având în vedere că mai multe părți ale imaginii sunt transmise simultan prin diferite canale Publish-Subscribe și că, ulterior, conținutul bufferelor formează un singur fișier pentru pixeli, a fost necesară dezvoltarea unui

80 | 4. Abordarea mecanismului OPC UA Publish-Subscribe în contextul transmiterii de imagini

mecanism de siguranță pentru acest tip de cazuri. Mecanismul este specific doar părții Subscriber a aplicației, iar în funcție de rezoluția imaginii în cauză, fiecare buffer specific fiecărui canal Publish-Subscribe este testat și se contabilizează fiecare operațiune de recepție. La sfârșitul transmisiunii, dacă numărul de valori prezente în fiecare buffer nu corespunde cu numărul așteptat de valori corespondente pixelilor, mecanismul completează cu valori suplimentare până când bufferul este plin. Avantajele acestei abordări constă în faptul că imaginea poate fi reconstruită chiar dacă nu a fost transmisă în întregime. Mai mult, pe baza calității imaginii reconstruite, observatorul poate identifica pe ce canale Publish-Subscribe transmisia a întâmpinat desincronizări sau pierderi, precum și când au avut loc acestea, având în vedere valoarea aproximativă a intervalului de timp preconizat pentru transmisie.

Entitățile Publisher și Subscriber ale aplicației sunt implementate în conformitate cu specificațiile [24], iar pentru fiecare canal Publish-Subscribe, componentele specifice mecanismului sunt prezente pe ambele dispozitive. După cum se precizează în [45], entitatea Publisher este responsabilă pentru crearea mesajelor de tipul DataSetMessages ce sunt trimise către Subscriber la un interval de timp configurat la nivelul componentei WriterGroup. Conținutul DataSetMessage este valoarea unui pixel extrasă din bufferele de stocare, în urma procesului de segmentare a imaginii. După trimiterea unei valori către Subscriber, se creează un nou DataSetMessage și se trimite valoarea următoare din buffer. Așadar, este esențial ca receptorul să primească toate mesajele pentru a obține o calitate a imaginii reconstruite, identică cu cea a imaginii originale. Receptorul primește periodic mesajele și extrage conținutul din singurul câmp prezent în DataSetMessage, menținând structura mesajelor cât mai simplă posibil. Chiar și în cazul canalelor multiple, cu mai mulți furnizori și receptori implicați în transmisie, designul vizat este o comunicare unu-la-unu, astfel încât entitatea Subscriber să poată fi implementată în mod similar cu modul descris în [52], evitându-se o parte din etapele de filtrare și obținând un timp de execuție mai rapid, necesar în cazul transmisiilor de mare viteză.

Diferitele versiuni ale aplicației OPC UA de transmitere a imaginilor, utilizează un număr diferit de canale Publish-Subscribe și de buffere de stocare pentru ambele dispozitive, vizând astfel următoarele aspecte pentru observare și analiză:

- robustețea mecanismului OPC UA Publish-Subscribe;
- viteza de transmisie în cazul imaginilor cu rezoluții diferite;
- calitatea imaginilor recepționate și reconstruite în comparație cu originalul.

Alte obiective majore includ următoarele:

- examinarea comportamentului în contextul unor rețele cu capacități diferite;
- realizarea unei soluții stabile care să poată furniza imagini identice în perioade de timp ce ar putea confirma utilitatea conceptului în industrie;
- angrenarea conceptului în scenarii autentice și valide, bazate pe industrie, cu o idee clară în privința avantajelor și dezavantajelor mecanismului OPC UA Publish-Subscribe.

4.4. Studii de caz

Mecanismul OPC UA Publish-Subscribe poate produce aplicații complexe cu capacități de funcționare în timp real și poate extinde utilizarea actuală a protocolului

către alte funcționalități sau alte domenii. Pe baza unor căutări amănunțite, autorul nu a identificat alte studii privind transmiterea de imagini prin OPC UA, acesta fiind deocamdată singurul studiu ce pune în context comun tehnologia OPC UA și transmiterea de imagini. Pe baza popularității protocolului în industrie și a evoluției tehnologiilor și mecanismelor integrate, aceste implementări cu obiective concrete, îndeplinite, pot rezolva sau îmbunătăți scenariile actuale și pot reprezenta un pas înainte către interconectivitate în contextul IIoT.

4.4.1. Studiu de caz inițial pentru evaluarea performanțelor în contextul procesării imaginilor

Pentru o analiză inițială privind utilizarea mecanismului de publicare-subscriere OPC UA în contextul prelucrării imaginilor, cazul de utilizare ales a fost detectarea și transmiterea de către un Publisher a mesajelor, pe baza prezenței obiectelor de interes într-o imagine, către un Subscriber localizat pe un alt dispozitiv. Scopul acestui studiu de caz a fost de a integra mecanismul într-un scenariu des întâlnit în industrie și de a determina un interval de timp în care operațiunea poate fi finalizată. YOLO a fost utilizat pentru detectarea a opt obiecte de interes în cadrul unei imagini și au fost trimise opt mesaje care semnalează prezența sau absența obiectelor către un al doilea dispozitiv.

Finalizarea operațiunii complete s-a făcut în 2.11 secunde; majoritatea duratei a fost consumată pentru detectarea caracteristicilor din cadrul imaginii. Intervalul de publicare pentru mesajele dintre entitatea Publisher și entitatea Subscriber a fost de 1 milisecundă pentru un mesaj (8 milisekunde pentru toate mesajele). Timpul de execuție pentru întreaga operațiune reprezintă o țintă orientativă pentru următoarele studii de caz, în care Subscriber-ul va primi întreaga imagine (nu doar mesajul care descrie prezența obiectelor în interiorul imaginii), astfel informația transmisă este sub o formă mai complexă.

4.4.2. Studiu de caz 1 - Transmiterea imaginii pe Un Singur canal Publish-Subscribe

Transmiterea integrală a unei imagini prin OPC UA Publish-Subscribe a fost menită să determine fezabilitatea conceptului și să stabilească relația dintre viteza de transmisie și calitatea imaginii recepționate. Într-un scenariu inițial de tip worst-case, care conturează prima fază a actualului studiului de caz, folosind doar un singur canal Publish-Subscribe, o imagine color de dimensiune medie spre mare a fost transmisă. Scenariul a utilizat o conexiune Ethernet clasică (fără tehnologia TSN), pentru a determina eventualele impedimente și desincronizări posibile ce pot apărea în timpul transmisiei, și pentru a identifica cauzele și posibilele soluții pentru astfel de obstacole. În cazul Figurilor 24 și 25, a fost utilizată o conexiune Wi-Fi pentru transmiterea unei imagini, aceasta conexiune nefiind favorabilă din punct de vedere al stabilității.

82 | 4. Abordarea mecanismului OPC UA Publish-Subscribe în contextul transmiterii de imagini

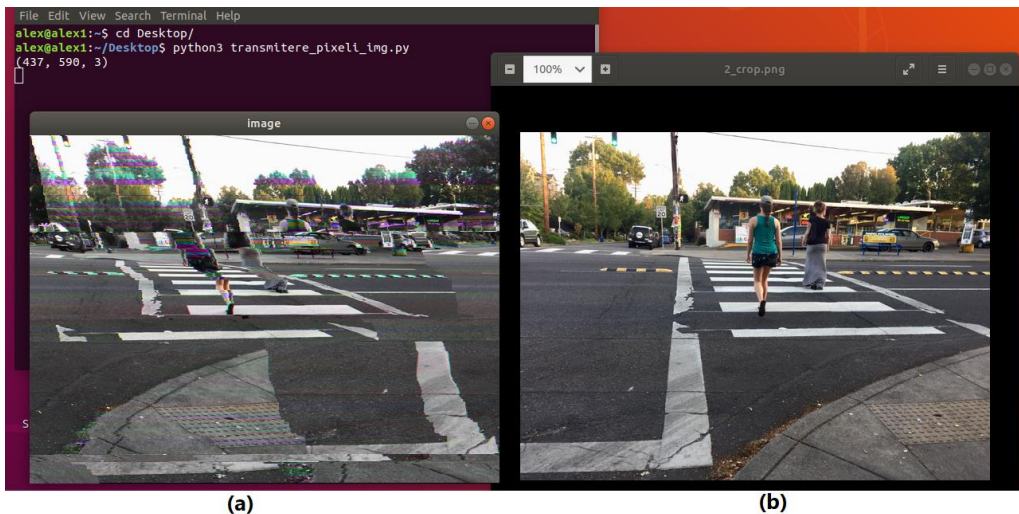


Figura 24. Comparație între imaginea recepționată (a) și imaginea originală (b) la o recurență de transmisie de 1 ms/valoare a unui pixel. (Faza 1)

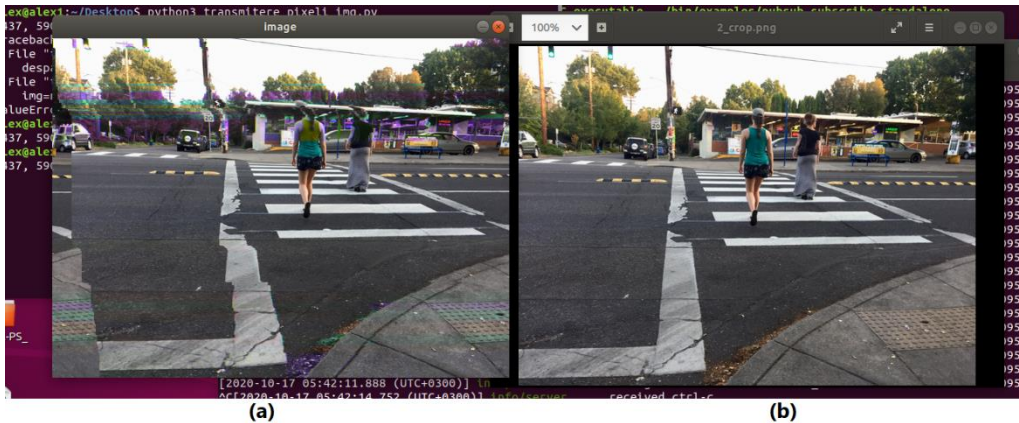


Figura 25. Comparație între imaginea recepționată (a) și imaginea originală (b) la o recurență de transmisie de 5 ms/valoare a unui pixel. (Faza 1)

Intervalele de publicare în cazul transmisiunilor au fost configurate între 1 milisecundă (1 ms/valoare a unui pixel) și 5 milisekunde (5 ms/valoare a unui pixel) din motive de stabilitate, cu scopul principal de a obține o imagine cât mai apropiată de versiunea originală. În Figurile 24 și 25, se poate observa calitatea imaginilor recepționate în comparație cu imaginea originală.

Fiind trimise 773,490 de valori corespondente pixelilor, între Publisher și Subscriber la diferite recurențe, rezultatul demonstrează că implementarea este capabilă să facă față unor transmisiuni de lungă durată (aproximativ 12.9 minute la recurența de 1 ms/valoare a unui pixel și 64.5 minute la recurența de 5 ms/valoare a unui pixel) pentru a livra imaginea către cel de-al doilea dispozitiv, validând robustețea mecanismului Publish-Subscribe pe UDP. Rezultatele pot fi clasificate ca fiind tot mai bune odată cu creșterea intervalului de timp de publicare. Explicația este

că, în condițiile în care valorile sunt trimise cu o viteză mai mică, orice desincronizare între cele două dispozitive implicate, determinată de gestionarea timpului sau de instabilitatea rețelei, va produce mai puține daune calității imaginii recepționate. Pe parcursul perioadei de desincronizare, cu mai puține valori trimise într-un interval de timp fix, se vor pierde mai puține valori, prin urmare, la 5 ms/valoare publicată, imaginea recepționată va deveni vizibil mai apropiată din punct de vedere calitativ de cea originală. Cu toate acestea, chiar dacă calitatea imaginii recepționate se îmbunătățește, timpul de transmisie este departe de orice limită fezabilă pentru industrie, așadar testele suplimentare în cadrul acestei faze a actualului studiu de caz, nu sunt utile în acest moment.

A doua fază a studiului de caz actual se axează pe un scenariu industrial mai probabil, unde imaginea trimisă are rezoluție și mărime mai mică, fiind de asemenea într-un format alb-negru, pentru o micșorare a numărului de valori corespondente pixelilor, ce necesită să fie livrate. Scopul principal este de a obține un timp de livrare mai relevant din punct de vedere industrial, pentru transmisia completă către dispozitivul receptor. Imaginea selectată conține 46,225 de valori corespondente pixelilor, ce necesită a fi livrate cu ajutorul aceleiași implementări a mecanismului Publish-Subscribe, precum în faza anterioară a studiului de caz. Rezultatele pot fi observate în Figurile 26 și 27.



Figurile 26. Comparație între imaginea recepționată (a) și imaginea originală (b) la o recurență de transmisie de 1 ms/valoare a unui pixel. (Faza 2)



Figurile 27. Comparație între imaginea recepționată (a) și imaginea originală (b) la o recurență de transmisie de 4 ms/valoare a unui pixel. (Faza 2)

Pentru cea de-a doua fază a studiului de caz, o imagine identică a fost recepționată utilizându-se un interval de publicare de 4 milisecunde. Cu un număr mai mic de valori ce trebuie livrate, un interval de publicare mai rapid a determinat ca probabilitatea de desincronizare să fie mai mică, în condițiile unei recurențe ce nu producea o imagine identică în faza inițială a studiului de caz. Se poate observa că în Figura 26, desincronizarea a apărut în al treilea sfert al imaginii, iar alterarea imaginii recepționate este mai mică în comparație cu imaginea primită în faza 1 la aceeași recurență (Figura 24). Același efect a făcut posibilă obținerea unei imagini identice la o viteză mai mare decât în faza inițială, folosind exact aceeași implementare. Având în vedere imaginea originală utilizată în faza 2 a studiului de caz, timpul de transmisie a fost îmbunătățit semnificativ - aproximativ 47 de secunde pentru o transmisie integrală la un interval de publicare de 1 milisecundă (1 ms/valoare pixel), și aproximativ 3.12 minute la un interval de publicare de 4 milisecunde (4 ms/valoare pixel). Faza 2 a studiului de caz a demonstrat că mecanismul OPC UA Publish-Subscribe oferă posibilitatea transmiterii de imagini în perioade de timp fixe și, chiar și cu probleme de stabilitate în cazul transmisiunilor de lungă durată, comunicarea dintre Publisher și Subscriber nu este întreruptă, inclusiv în cazul fluxurilor mari de date publicate. În contextul imaginilor industriale (cu dimensiuni, culori și rezoluție moderate), conceptul poate reprezenta o soluție cu performanțe ce pot fi îmbunătățite.

4.4.3. Studiu de caz 2 - Transmiterea imaginii pe Patru canale Publish-Subscribe

Având garanția că mecanismul OPC UA Publish-Subscribe este capabil de a transmite o imagine de la un dispozitiv la altul, fără nicio alterare a acesteia, îmbunătățirile care pot fi aduse vizează timpul de livrare al transmisiei integrale. Printre factorii care pot îmbunătăți durata de transmisie, pe lângă imaginea vizată, se află și strategia transmisiei multi-canal, ce va livra diferite segmente ale imaginii în paralel, scăzând numărul de valori corespondente pixelilor, necesare a fi trimise pe fiecare canal individual. Astfel, scade probabilitatea de desincronizare și crește probabilitatea de a recepționa o imagine identică la nivelul receptorului, utilizând recurențe cât mai mari. Pentru ca acest proces să aibă succes, au fost implementate patru canale Publish-Subscribe diferite, care funcționează independent. Cele patru entități de publicare schimbă informații cu cele patru entități receptoare implementate pe al doilea dispozitiv, utilizând două adrese multicast și patru porturi diferite, corespondente fiecărui canal individual de comunicare. Intervalele de publicare sunt configurate să fie identice, astfel încât duratele celor patru transmisiuni să fie aproximativ egale.

Rezultatele experimentului au confirmat concluziile inițiale ale studiului de caz 1. Imaginea recepționată pe al doilea dispozitiv a fost identică cu imaginea originală (la fel ca în Figura 27) la un interval de publicare de 1 milisecundă (1 milisecundă/valoare de pixel pentru toate cele patru canale), obținându-se o transmisie completă în aproximativ 12 secunde.

Pe lângă confirmarea comportamentelor deja observate, studiul de caz actual oferă o soluție stabilă, capabilă să transmită o imagine de la un dispozitiv la altul prin mecanismul OPC UA Publish-Subscribe, dovedind faptul că strategia de transmisie multi-canal poate îmbunătăți performanțele aplicației, crescând stabilitatea chiar și în cazul transmisiilor la rate ridicate și oferind o metodă scalabilă de implementare, cu potențial de a atinge constrângerile de timp necesare în industrie pentru transmiterea de imagini.

4.4.4. Studiu de caz 3 - Transmiterea imaginii pe 20 de canale Publish-Subscribe

Studiul de caz actual are ca scop obținerea unui timp de transmisie fezabil, demonstrând scalabilitatea strategiei de transmisie multicanal prin mecanismul de publicare-subscriere OPC UA. De asemenea, se urmărește integrarea conceptului în scenarii prezente în industrie, utilizând imagini relevante din cadrul proceselor industriale. Din perspectiva rețelei utilizate, testarea impactului generat de capacitatea unei rețele în contextul implementării curente, reprezintă un punct de interes.

Studiul de caz 3 face referire la trei scenarii în contextul OPC UA Publish-Subscribe de transmitere a imaginilor, după cum urmează:

- primul scenariu constă în comunicația între autovehicule și infrastructură, menționată anterior;
- cel de-al doilea scenariu analizează transmiterea imaginii cu starea cutiei ambalate complet în procesul de testare la sfârșitul liniei de producție în domeniul automotive;

86 | 4. Abordarea mecanismului OPC UA Publish-Subscribe în contextul transmiterii de imagini

- cel de-al treilea scenariu se focalizează asupra liniei de producție a componentelor ECU, mai exact pe transmiterea rezultatelor inspecției optice automate (AOI) în legătură cu posibile defecțiuni.

Obiectivul principal, în ceea ce privește durata de livrare a imaginii integrale, a fost de a reuși transmisia după cum urmează:

- sub 3 secunde pentru transmisia integrală în cazul scenariului de comunicare car-to-infrastructură, utilizându-se imaginea din studiile de caz 1 și 2.
- sub 12 secunde pentru imaginea cutiilor ambalate complet, la sfârșitul liniei de producție;
- sub 3 secunde pentru o imagine ce reprezintă un rezultat negativ al procesului de detecție a defecțiunilor ECU în urma AOI.

Scenariul studiilor de caz anterioare, în care a fost luat în considerare contextul comunicației car-to-infrastructură, este extins în cadrul implementării studiului de caz 3. Rezultatele versiunii actuale a aplicației, care utilizează 20 de canale separate, au furnizat un timp de livrare de aproximativ 2.4 secunde pentru aceleași imagini utilizate în studiile de caz 1 și 2 (Figurile 26 și 27), cu un interval de publicare de 1 milisecundă (1 ms/valoare pixel). Imaginea obținută a fost identică cu cea originală (aceleși rezultat precum cel din în Figura 27). Se confirmă astfel scalabilitatea strategiei de transmisie multi-canal, permițând configurații multiple pentru diverse scenarii din cadrul dezvoltărilor viitoare ale aplicației.

Conceptul este testat în continuare, folosind scenarii bazate pe procesele de producție industrială, unde procesarea imaginilor este deja integrată în proces, iar aplicabilitatea poate fi extinsă în mod considerabil. Procesul de inspecție optică automată (AOI) a componentelor ECU, descris anterior, care face parte din procesul de testare a produsului final, va fi împărțit în două scenarii. Cele două scenarii reprezintă două etape separate din cadrul testării EOL (end-of-line, adică la sfârșitul liniei de producție) în care se utilizează AOI.

Primul dintre aceste scenarii bazate pe procese industriale, se focalizează pe cutiile ambalate la sfârșitul liniei de producție, ce urmează a fi livrate către client. După numărarea continuă a componentelor ECU ce sunt introduse în cutii, atunci când limita de numărare pentru cutia selectată este atinsă, MES este notificat că ambalarea este finalizată și imaginea finală este stocată local (a se vedea Figura 28b). Luând în considerare procesul, timpii specifici de execuție în linia de producție sunt următorii: un nou ECU este introdus în cutie după minimum 12 secunde; detectarea unei noi cutii de către sistemul AOI este de aproximativ 6 secunde; și o umplere completă, în funcție de cutie și de ECU, este mai mare de 2.5 minute.

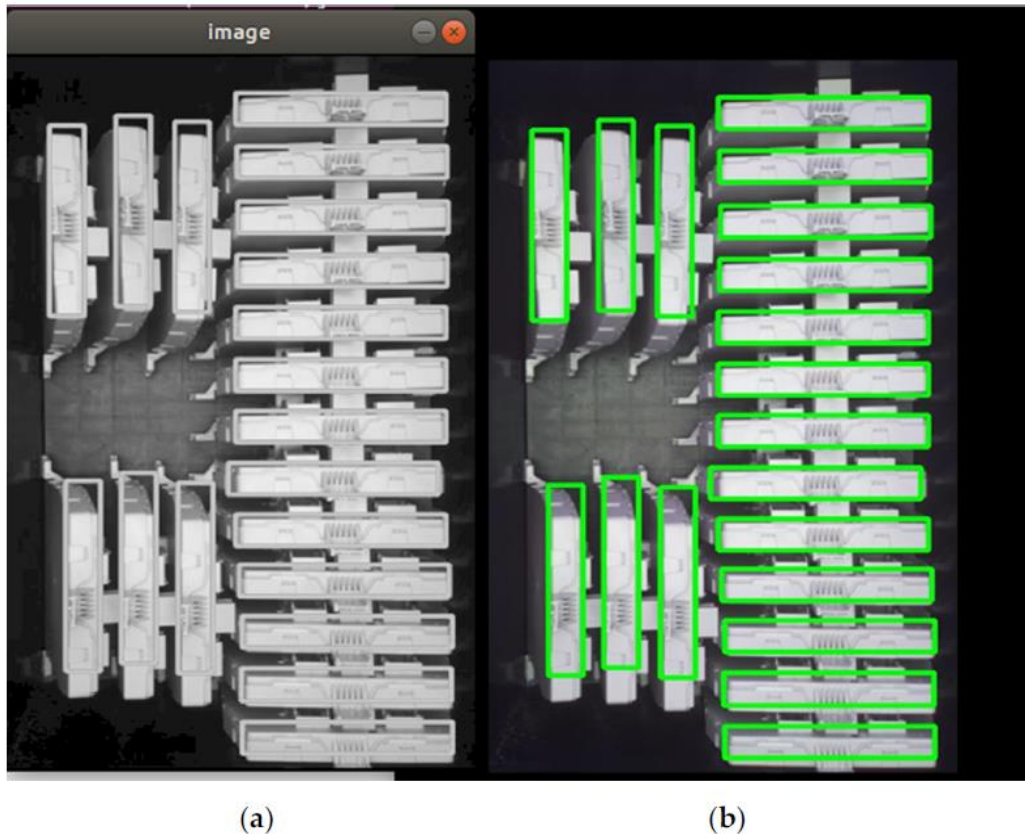


Figura 28. Utilizarea mecanismului OPC UA Publish-Subscribe pe 20 de canale, pentru transmiterea imaginilor cu ambalarea cutiilor la sfârșitul liniei de producție: (a) Imaginea recepționată și (b) Imaginea originală.

În urma implementării strategiei de publicare multi-canal pe OPC UA, rezultatele obținute sunt încurajatoare. Imaginea prezentă în Figura 28b este convertită în format alb-negru și transmisă fără a fi alterată, către destinație. Imaginea recepționată este prezentată în Figura 28a.

Pentru transmisia completă a imaginii reprezentate în Figura 28, a fost obținut un timp de 7.85 secunde. Prin urmare, soluția ar permite din punct de vedere al duratei chiar și transmiterea unei imagini pentru fiecare placă introdusă în cutie. Astfel obiectivele principale ale studiului de caz actual au fost îndeplinite cu succes, pentru primele două scenarii propuse.

Scenariul final din cadrul studiului de caz 3 este orientat către o etapă anterioară a procesului de testare la sfârșitul liniei de producție. Inspecția optică automată (AOI) a componentelor ECU are rolul de a detecta posibile defectele la nivelul pinilor, sau la nivelul suprafeței (a se vedea studiul [47]). Rezultatele din [47] prezintă detecția defectelor bazată pe procesarea imaginilor în aproximativ 6.5 secunde. Bitul de stare (defectă/funcțională) rezultat este transmis către MES în limita de timp necesară. Având în vedere durata tuturor celorlalte proceduri ce au loc în linia de producție, trecerea de la o placă la alta în procesul de inspecție optică automată

88 | 4. Abordarea mecanismului OPC UA Publish-Subscribe în contextul transmiterii de imagini

(AOI), ar permite o durată de aproximativ 12 secunde pentru a transmite imaginile rezultate în cadrul fiecărui ciclu de testare a plăcilor. Numai defectele (rezultate negative ale procesului de detecție) sunt stocate și ar necesita să fie transmise, respectiv, aceste rezultate nu reprezintă de obicei mai mult de 1-2 imagini, deoarece o imagine stocată poate conține mai multe defecte ale pinilor (de exemplu, imaginea din Figura 29 ilustrează un conector care poate conține mai mult de 30 de defecte la nivelul pinilor).

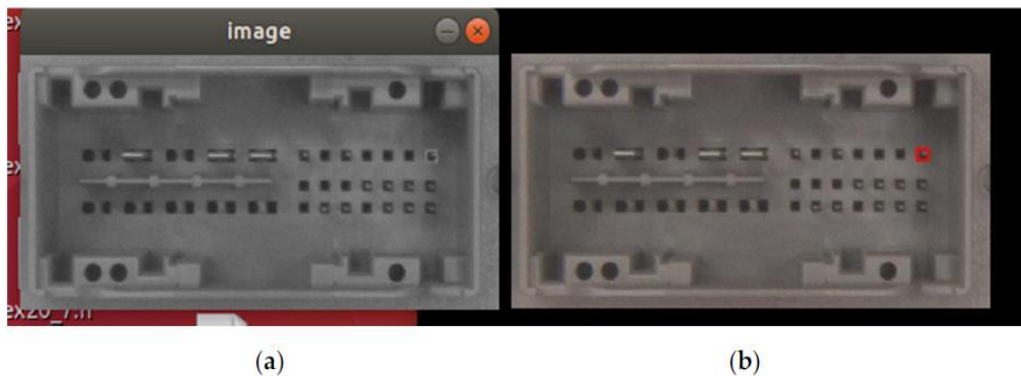


Figura 29. Utilizarea mecanismului OPC UA Publish-Subscribe pe 20 de canale, pentru transmiterea imaginilor pe baza procesului AOI asupra componentelor ECU: (a) Imaginea recepționată și (b) Imaginea originală.

În urma implementării mecanismului Publish-Subscribe pentru acest ultim scenariu, se poate afirma faptul că rezultatele obținute reprezintă un succes. Imaginea originală prezentată în Figura 29b este convertită în format alb-negru și transmisă integral către destinație, fără a fi alterată. Imaginea recepționată poate fi observată în Figura 29a.

Durata transmisiei integrale a imaginii prezentate în Figura 29 este de 2.52 secunde. Așadar, considerând durata disponibilă de 12 secunde, pentru transmiterea imaginilor rezultate în cadrul fiecărui ciclu de testare, aproximativ patru imagini pot fi transmise utilizându-se soluția OPC UA Publish-Subscribe pe 20 de canale.

Conceptul a fost testat în diferite rețele de 2,4 GHz și 5 GHz. Probabilitatea de desincronizare a operațiunilor de transmisie și recepție, este prezentă în ambele tipuri de rețele, cât timp cele două dispozitive ce comunică prin UDP nu sunt sincronizate la nivelul legăturii de date (Data Link Layer) specific modelului OSI, prin standarde specializate precum în cazul tehnologiei TSN. În orice caz, această probabilitate nu poate fi calculată în mod exact, existând mulți factori ce pot contribui la desincronizare. Totuși în cazul rețelei de 5GHz, eficiența a fost de 100% utilizând versiunea actuală a aplicației. E de așteptat ca în scenarii industriale, problemele de rețea, referitoare la stabilitate, să fie monitorizate și gestionate cu ajutorul unor instrumente suplimentare, asigurând o calitate ridicată pentru toate serviciile implicate. Studiile viitoare ar putea arăta o imagine mai completă a impactului dintre aplicațiile OPC UA Publish-Subscribe și rețelele industriale, fiind implicate diverse tipuri de dispozitive, pe măsură ce mecanismul OPC UA Publish-Subscribe se va răspândi în domeniul automatizărilor.

4.5. Rezultate

Implementarea aplicației OPC UA Publish-Subscribe pentru transmiterea imaginilor a fost realizată cu succes. Fiecare studiu de caz a generat concluzii importante în legătură cu stabilitatea și eficiența mecanismului Publish-Subscribe, în concordanță cu factorii industriali vizați. Performanța aplicației în scenarii autentice, prezente în industrie, a demonstrat potențialul de utilizare într-o zonă inaccesibilă până în prezent pentru protocolul OPC UA.

Capacitățile de funcționare în timp real ale aplicațiilor OPC UA bazate pe mecanismul Publish-Subscribe sunt analizate din diferite perspective în capitolele anterioare (a se vedea [45,52]), cu diverse rezultate și realizări, și considerente asupra tehnologiei TSN. După cum se menționează în [71], conceptul OPC UA Publish-Subscribe implică și utilizarea extensiei TSN pentru Ethernet, pentru a obține garanții asupra intervalelor de timp real, pentru toate procesele specifice mecanismului Publish-Subscribe și pentru a obține un comportament determinist în ceea ce privește transportul mesajelor de tipul DataSetMessages. Pentru obținerea unui comportament deplin în timp real între diferite dispozitive, pentru evitarea evenimentelor de desincronizare și pentru atingerea potențialului maxim al mecanismului OPC UA Publish-Subscribe, tehnologia TSN reprezintă o soluție adecvată, chiar dacă au fost dezvoltate metode de sincronizare suplimentare pentru anumite scenarii (a se vedea [52]). Pentru cazul actual, obstacolele care pot apărea la transmisiunile de lungă durată și care pot modifica rezultatul final al aplicației, sunt relevante pentru înțelegerea necesității TSN, iar îmbunătățirile viitoare, din punct de vedere al performanței, trebuie să fie preconizate viitoare integrări între OPC UA și TSN.

În tabelul 3, poate fi observată comparația rezultatelor relevante, ale tuturor studiilor de caz, cu avantaje și dezavantaje pentru fiecare versiune a aplicației, luând în considerare contextul specific fiecărui caz de utilizare.

tabelul 3. Analiza rezultatelor studiilor de caz

Studiu de caz	Numărul de Canale Publish-Subscribe	Intervalul de timp necesar pentru o transmisie integrală a unei imagini identice cu originalul	Factori ce pot produce instabilitate	Concluzii
1	1	64.5 minute pentru Faza 1 (interval de publicare de 5 ms/valoare pixel)	<ul style="list-style-type: none"> volum mare de date ce trebuie trimise de un singur canal duratele lungi de transmisie cresc probabilitatea de desincronizare între dispozitive 	nefezabil pentru procesele industriale
		3.12 minute pentru Faza 1 (interval de publicare de 4 ms/valoare pixel)		

90 | 4. Abordarea mecanismului OPC UA Publish-Subscribe în contextul transmiterii de imagini

2	4	12 secunde (interval de publicare de 1 ms/valoare pixel)	<ul style="list-style-type: none">volum mai mic de date ce trebuie transmis de fiecare canal individualnu este garantată o probabilitate mică de desincronizare între dispozitive	performanțe îmbunătățite, dar departe de rezultatul dorit
3	20	2.4 secunde (interval de publicare de 1 ms/valoare pixel)	<ul style="list-style-type: none">volum adecvat de date ce trebuie transmis de fiecare canal individualeste garantată o probabilitate foarte mică de desincronizare între dispozitive	fezabil în scenarii industriale, pentru procese specifice

Odată cu transmiterea cu succes, a imaginilor prin mecanismul OPC UA Publish-Subscribe, aplicabilitatea protocolului OPC UA se extinde către noi funcționalități și confirmă fezabilitatea ridicată a paradigmei Publish-Subscribe pentru viitoare dezvoltări, oferind posibilitatea de a satisface cerințele industriale. Performanțele actuale și comportamentul în timp real pot fi îmbunătățite în contextul tehnologiei TSN. Acest concept oferă modalități de implementare pentru aplicații cu capacități mari de schimb de date, la viteze mari, între diferite dispozitive, prin Ethernet, asigurând astfel obiectivele indispensabile în contextul IIoT și Industry 4.0.

4.6. Concluzii

Principalele concluzii ale acestei etape de cercetare demonstrează că protocolul OPC UA poate oferi soluții pentru transmiterea de imagini în scenarii desfășurate în timp real și poate atinge performanțe fezabile utilizând concepte robuste și scalabile. Mecanismul Publish-Subscribe este fiabil pentru mai multe scenarii complexe, iar prin dezvoltarea actuală, transmisiile de date pe termen lung, sunt analizate în diferite circumstanțe. Aplicabilitatea strategiilor și arhitecturii curente, pentru transmisiunile multi-canal, permite o viteză mai mare de transmisie a imaginii, fiind posibilă astfel extinderea către domeniul industrial de fabricație auto și către scenarii de comunicație de tip car-to-infrastructure, unde protocolul OPC UA poate aduce îmbunătățiri semnificative.

Toate obiectivele propuse în această etapă de cercetare au fost atinse. În ceea ce privește constrângerile de funcționare în timp real pentru volume mari de date, mecanismul OPC UA Publish-Subscribe s-a dovedit a fi fiabil, chiar dacă riscurile de

desincronizare sunt prezente și pentru intervalele rapide de publicare, pe perioade lungi de timp, fără adoptarea tehnologiei TSN. Strategia de comunicare multi-canal pe UDP a permis asigurarea unor intervale de timp de transmisie fezabile în scenarii industriale, extinzând aplicabilitatea protocolului. Soluția implementată oferă posibilitatea transmiterii de imagini prin OPC UA în studii de caz cu un impact semnificativ asupra unor cazuri de utilizare reale, cu cerințe complexe, oferind noi abordări pentru implementarea mecanismului Publish-Subscribe într-un context inovativ.

Principalele provocări pentru această fază a studiului curent, sunt legate de multitudinea de scenarii ample ce au fost vizate pentru a se obține detalii și observații relevante cu privire la factorii care pot crește sau diminua performanța aplicației. Pentru fiecare scenariu și pentru fiecare versiune a aplicației a fost implementat cod de testare suplimentar pentru toate cele patru etape ale transmisiunii, concomitent cu dezvoltarea unor funcții speciale cu rol de monitorizare a duratelor fiecărei etape de transmisie. Identificarea criteriilor decisive pentru analiza fiecărui studiu de caz, s-a dovedit a fi o sarcină complexă, din cauza aspectului inovator al conceptului, iar comparația cu situații reale din industrie, adaugă complexitate atât din punct de vedere al implementării cât și al performanțelor așteptate. Strategia de comunicare multi-canal prin UDP reprezintă un scenariu scalabil de comunicație one-to-one între entitățile Publisher și Subscriber ce împart același canal. Principala provocare pentru acest tip de strategie este reprezentată de utilizarea de resurse pentru toate canalele individuale dezvoltate, însă cu toate acestea, prin adoptarea strategiei se produc progrese semnificative în ceea ce privește timpul de transmisie și stabilitatea aplicației.

Chiar și în cazul rezultatelor de succes ale aplicației OPC UA Publish-Subscribe de transmitere a imaginilor, analiza direcțiilor viitoare de dezvoltare poate duce la creșterea performanțelor și a stabilității. Cei mai importanți pași ce pot fi realizați în cadrul studiilor viitoare constă în dezvoltarea unor strategii diferite în legătură cu formatul imaginii și manipularea pixelilor, pentru obținerea unor modalități diferite de alcătuire a datelor distribuite, respectiv dezvoltarea de strategii privind structura și secvențele de transmitere a mesajelor de tipul DataSetMessages. De asemenea, adoptarea tehnologiei TSN poate genera o viziune mai clară asupra comportamentului predictibil, în timp real, în cazul utilizării protocolului OPC UA. Din punct de vedere hardware, studiile privind capacitățile diferitelor dispozitive industriale compatibile cu mecanismul OPC UA Publish-Subscribe, ar putea crește fezabilitatea conceptului în structurile industriale.

Abordarea mecanismului OPC UA Publish-Subscribe în contextul transmiterii de imagini, extinde posibilitățile de utilizare ale arhitecturilor OPC UA bine stabilite, și confirmă eficiența transmisiunilor multi-canal, luând în considerare intervalele de timp de livrare specifice scenariilor industriale din diferite domenii, astfel furnizând un următor pas spre interoperabilitate în sectorul IIoT.

5. COEXISTENȚA PROTOCOALELOR DDS ȘI OPC UA ÎN CONTEXTUL INDUSTRY 4.0 ȘI A COMUNICAȚIEI ÎN TIMP REAL

Continuând evoluția către Industry 4.0, protocoalele de comunicare industrială reprezintă un subiect de interes major, deoarece schimbul de date în timp real între mai multe dispozitive constituie pilonul scenariilor IIoT (Industrial Internet of Things). Deși protocoalele tradiționale sunt încă persistente în industrie, tranziția a fost inițiată de protocolul cheie de facilitare a conceptului Industry 4.0, Open Platform Communication Unified Architecture (OPC UA). OPC UA trebuie să atingă nivelul preconizat de aplicabilitate și, prin urmare, trebuie să ia în considerare coexistența cu alte aplicații emergente orientate spre funcționarea în timp real în liniile de producție. Protocolul DDS (Data Distribution Service) va fi cu siguranță prezent în viitoarele arhitecturi din anumite sectoare ca cel al roboților industriali, co-boților și al unităților compacte. Capitolul curent propune o soluție de evaluare a coexistenței și funcționării în timp real în cazul protocoalelor OPC UA și DDS, atât în paralel cât și într-un context ce implică o aplicație gateway. Scopul acestui studiu este de a demonstra fezabilitatea și compatibilitatea între cele două protocoale, alături de o definiție generală a criteriilor de evaluare și a așteptărilor, din punct de vedere arhitectural. Se identifică astfel avantajele și dezavantajele în cazul utilizării protocoalelor în context comun, conturând o viziune cuprinzătoare asupra posibilităților și soluțiilor disponibile la ora actuală. Arhitectura definită pentru actualul studiu, este menită să corespundă atât cu scenariul de comparație a performanțelor, cât și cu scenariul de interacțiune, prin intermediul unei aplicații de tip gateway. Având în vedere tendințele industriale, soluția dezvoltată este aplicată folosind infrastructuri non-ideale, pentru a oferi autenticitate și aplicabilitate rapidă în liniile de producție.

5.1. Noțiuni generale

Fiind vizate îmbunătățiri în ceea ce privește disponibilitatea, siguranța, productivitatea și reducerea costurilor, industria de producție se îndreaptă spre Industry 4.0, în consecință și spre IIoT, cele două concepte fiind strâns legate. Toate posibilele îmbunătățiri în contextul Industry 4.0 și IIoT se bazează pe capacitatea de a se colecta și de a se realiza schimburi de date între dispozitive [72,73,29], și, prin urmare, pe concepte precum interfațarea, interoperabilitate, interoperare și conectivitate.

Protocoalele de comunicare reprezintă implicit o temă de cercetare foarte importantă în actuala revoluție industrială. Prima provocare reprezentată de protocoalele tradiționale a fost depășită prin tranziția către principalul facilitator al Industry 4.0, și anume OPC UA [49]. Lunga dispersie tehnologică și evoluția lentă a sistemelor noi de automatizare, precum și progresul limitat în domeniul cercetării și aplicării protocoalelor moderne, determină încă o prezență generalizată a sistemelor tradiționale. Cu toate acestea, cerințele industriale determină cercetarea a diverse tipuri de soluții de împachetare, pentru a crește performanța [31]. În viitorul apropiat,

luând în considerare necesități tot mai mari pentru proceduri de comunicare bazate pe Ethernet între diferite tipuri de dispozitive, este de așteptat ca arhitecturile majore, stabilite prin intermediul celor mai eficiente și mai populare tehnologii, să interacționeze între ele, la scară largă. Unul dintre obiectivele esențiale pentru obținerea unor rezultate rapide, este identificarea soluțiilor potrivite și demonstrarea compatibilității acestora în circumstanțe autentice.

Specificațiile recente ale OPC UA [24], permit protocolului să evolueze spre cerințele industriale de funcționare în timp real. Deși s-au făcut progrese semnificative [52,71], protocolul încă trebuie să își atingă aplicabilitatea preconizată. În acest context, cercetarea trebuie să progreseze în legătură cu mecanismul OPC UA Publish-Subscribe, focalizându-se semnificativ pe cerințele de funcționare în timp real. Cu toate acestea, liniile de producție din industria de prelucrare includ domenii precum roboții industriali, coboții (roboți colaborativi), unitățile compacte, unde un alt sistem emergent, cu capacități de funcționare în timp real, va fi cu siguranță prezent în arhitecturile viitoare. Protocolul DDS (Data Distribution Service) este definit de OMG (Object Management Group) și a devenit în ultimii ani prezent în numeroase implementări industriale, fiind perceput ca o soluție eficientă pentru schimbul de date pe Ethernet. În acest caz, analiza de coexistență a soluțiilor OPC UA și DDS bazate pe paradigma Publish-Subscribe, este esențială și este nevoie să fie orientată către cerințele industriale de funcționare în timp real. Etapa actuală de cercetare propune un instrument de analiză și o metodologie de evaluare a capacității de funcționare în timp real pentru protocoalele de comunicare OPC UA și DDS, atât în context de funcționare paralelă cât și comună, prin intermediul unei aplicații de tip gateway (Figura 30).

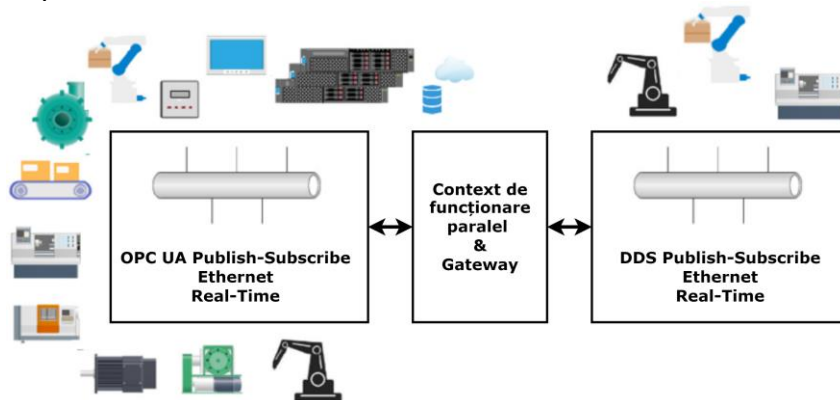


Figura 30. Vizualizarea schematică a coexistenței protocoalelor OPC UA-DDS în contextul Industry 4.0.

Există un alt aspect important care trebuie analizat în profunzime, referitor la comportamentul în timp real pentru DDS și OPC UA. Mediul industrial are nevoie de o aplicabilitate tehnologică rapidă a noilor concepte. Liniile de producție actuale conțin în principal soluții de tip OPC UA client-server și protocoale tradiționale. Se dorește tranziția către protocoale de tip Publish-Subscribe orientate spre funcționarea în timp

94 | 5. Coexistența protocoalelor DDS și OPC UA în contextul Industry 4.0 și a comunicației în timp real

real, iar în acest context infrastructura este, de asemenea, o problemă critică (a se vedea [74]). Studiile [27,60] se focalizează pe echipamente TSN (Time-Sensitive-Network), dar industria are nevoie de abordări mai rapide în ceea ce privește aplicabilitatea constrângerilor de funcționare în timp real, din perspectiva sistemelor de operare și a echipamentelor utilizate. Nu s-a realizat încă o tranziție completă către TSN din cauza progresului foarte lent al adoptării TSN în cadrul utilizării industriale.

Considerând situația actuală, această etapă de cercetare propune următoarele obiective:

- Definirea unor criterii specifice, ce permit examinarea protocoalelor DDS și OPC UA într-un sistem imperfect, astfel luând în considerare multe provocări prezente în industrie.
- Analizarea comportamentului în timp real pentru DDS și OPC UA, implementând toate mecanismele necesare pentru acest proces.
- Definirea unei arhitecturi adecvate pentru utilizarea DDS și OPC UA în paralel, oferind însă și opțiunea de interacțiune între cele două protocoale de comunicare.
- Implementarea unei aplicații de tip gateway DDS-OPC UA.

Următoarea secțiune abordează aspecte de interes, în legătură cu tehnologiile DDS și OPC UA în cadrul IIoT și Industry 4.0, alături de provocări privind infrastructura orientată spre funcționarea în timp real.

5.2. Metode și Tehnologii utilizate

În acest subcapitol, protocoalele DDS și OPC UA vor fi descrise din punct de vedere industrial cât și din perspectiva cercetării, luând în considerare diferite contexte și posibilități de aplicare. Fiecare dintre cele patru secțiuni aferente, oferă o analiză aprofundată și observații în legătură cu cele mai importante mecanisme și tehnologii cu rol important în cadrul sistemelor deterministice de comunicare. Din punct de vedere al implementării, a fost utilizat eProsima Fast DDS [75] open-source SDK pentru toate entitățile DDS, respectiv open62541 SDK [25] pentru toate entitățile OPC UA implicate.

5.2.1. DDS în contextul IIOT

DDS vizează comportamentul în timp real în contextul comunicației pe Ethernet, implementând o paradigmă de tipul Publish-Subscribe și oferind, de asemenea, o paradigmă de tipul Request-Reply pentru trimiterea de comenzi ce așteaptă un răspuns sau pentru accesarea serviciilor individuale, asigurând o eficiență ridicată pentru scenarii de comunicație de tipul one-to-many, one-to-one și many-to-many. Disponibilitatea a numeroase mecanisme ce pot fi utilizate pentru schimbul de date, face din DDS o soluție flexibilă și scalabilă atunci când se ia în considerare evoluția constantă a cerințelor specifice domeniului IIoT. În cadrul mecanismului publish-subscribe, schimbul unidirecțional se realizează pe baza politicilor QoS definite în cadrul arhitecturii, punând la dispoziție un set divers de opțiuni din punct de vedere al implementării. DDS oferă un model data-centric, care stabilește conceptul de spațiu global de date la care pot avea acces toate entitățile, un spațiu în care informațiile

sunt propagate odată ce rolul fiecărei entități este stabilit. Designul paradigmei publish-subscribe se bazează pe Topic-uri. Entitățile receptoare își exprimă interesul în legătură cu un Topic și astfel se realizează o legătură cu entitatea furnizoare. Cu un design bazat pe Topic-uri, în arhitecturile complexe, schimbul de date este simplificat, permițând ca doar datele de interes să ajungă la fiecare receptor, sporind eficiența în cazul constrângerilor de funcționare în timp real și oferind o modalitate mai abstractă de clasificare și încapsulare a datelor. Acest design asigură scalabilitatea protocolului în cazurile de utilizare unde arhitectura sistemului conține multe entități participante, iar multiple tipuri de date sunt publicate la diverse recurențe. Entitățile pot fi grupate sub formă de domenii, având spații virtuale izolate unde furnizorii și receptorii pot realiza schimbul de date specifice, îmbunătățind astfel nivelul de flexibilitate al protocolului și sporind volumul și complexitatea datelor distribuite, fiind posibilă dezvoltarea unor arhitecturi mai elaborate.

Ca protocol de legătură, se folosește protocolul RTPS (Real-Time Publish-Subscribe), ce asigură interoperabilitatea în rețelele standard, punând accent pe cerințele de funcționare în timp real. La nivel de transport, RTPS poate utiliza TCP/UDP/IP (Transmission Control Protocol / User Datagram Protocol / Internet Protocol) oferind acces la toate caracteristicile specifice DDS, și asigurând portabilitatea și compatibilitatea cu versiuni DDS diferite. Similar cu DDS, aceleași principii privind asocierea entităților la un domeniu, sunt aplicate și în cazul RTPS, iar subcomponentele și caracteristicile DDS sunt mapate la RTPS. Acest lucru oferă o modalitate abstractă de abordare a detaliilor specifice sistemului, din perspectiva arhitecturii și a definirii rolurilor pentru toate entitățile implicate, de la nivel "de fir" până la nivel de aplicație. De asemenea, este menținut același nivel de flexibilitate în legătură cu toate mecanismele necesare pentru realizarea cerințelor de funcționare în timp real, cât și pentru identificarea dintre furnizori și receptori în cadrul unui domeniu.

Pentru un protocol de comunicare capabil să asigure funcționarea în timp real, se impune ca apelurile de funcții să fie executate într-un context unde de timpii de execuție și timpii de așteptare pot fi configurați și monitorizați. Conform acestui principiu, pe sistemele de operare bazate pe Linux, DDS pune la dispoziție un interval configurabil de timp de blocare pentru funcțiile care concurează pentru resurse. În situațiile în care intervalele de timp de blocare sunt depășite, există metode dedicate care prioritizează cele mai importante operații, confirmând astfel eficiența protocolului DDS în cazul aplicațiilor cu cerințe stricte.

Pe lângă structurarea API-ului sub forma a doua nivele, RTPS și cel specific conceptelor DDS mai abstracte, o altă caracteristică importantă a protocolului DDS este reprezentată de mecanismele de descoperire ale participanților. Descoperirea simplă se bazează pe standardul RTPS și permite compatibilitatea cu alte implementări DDS. Descoperirea statică presupune cunoașterea tuturor subcomponentelor de tipul DataWriters și Data Readers, alături de tipurile de date și Topic-urile pe baza cărora este structurată arhitectura, astfel entitățile fiind capabile să se identifice reciproc prin IP-uri și porturi specificate. Mecanismul Serverului de descoperire, constituie o altă soluție flexibilă, în care furnizorul informațiilor de metatrafic este reprezentat de una dintre entitățile existente, aparținând domeniului.

Odată cu evoluția complexității în cazul arhitecturilor distribuite, sistemele vizează utilizarea mai multor protocole și dispozitive cu diferite tipuri de cerințe. Astfel devine necesară dezvoltarea unor mecanisme de siguranță la diferite niveluri pentru prevenirea defecțiunilor de sistem și asigurarea recuperării rapide în cazul anomaliilor. Aceste tipuri de mecanisme pot contribui decisiv la eficiența tehnologiilor

utilizate în scenarii industriale. Pentru protocoalele de comunicare care asigură interoperabilitatea între mai multe dispozitive, ce transmit sau recepționează volume mari de date, recuperarea rapidă a procedurilor de comunicare, în urma unei anomalii, poate reprezenta diferența dintre o tehnologie eficientă în situații critice și o tehnologie inadecvată. În cazul DDS, un mecanism pentru acest tip de scenariu este reprezentat sub forma serviciului de persistență. Serviciul de persistență permite stocarea detaliilor legate de context alături de ultima modificare notificată a valorii datelor, în cadrul procesului de recuperare al unei stări anterioare a sistemului. Având această funcționalitate disponibilă, sistemul este capabil să restabilească, cât mai repede posibil, starea comunicației, de dinaintea unei opriri nedorite. Astfel, entitățile implicate, responsabile de operațiunile de transmitere sau de recepționare, pot acționa corespunzător situației, oferind o imagine clară asupra datelor ce au sosit înainte de evenimentul de oprire, respectiv asupra datelor ce trebuie să fie livrate la momentul de după reluarea activității. Abilitățile de acest fel reprezintă un avantaj pentru toate tehnologiile IIoT și oferă, din perspectiva dezvoltării, o viziune mai largă și posibilități de control sporite asupra sistemului. În cazul DDS, serviciul de persistență sporește robustețea și oferă o calitate ridicată a serviciilor, confirmând faptul că tehnologia este potrivită pentru o mare varietate de cazuri de utilizare cu o gamă largă de cerințe.

În domeniul automotive, DDS face parte din platforma Adaptive AUTOSAR, alături de alte protocoale de comunicare prin Ethernet, mai răspândite, cum este cazul protocolului SOME/IP. SOME/IP este utilizat în mod universal pentru scenarii de comunicație în interiorul vehiculului, iar DDS este considerat a fi utilizat în principal pentru procesele de fabricație din domeniul automotive, fiind implementat în linii de asamblare și de producție pentru roboții industriali. De asemenea, considerând sistemele integrate și dispozitivele cu resurse limitate, Micro-XRCE-DDS este disponibil ca soluție middleware ce asigură QoS și suport multi-platformă, cu un consum redus de resurse.

Există disponibile în comunitatea științifică mai multe studii ce fac referire la implementări DDS pentru o varietate de domenii și cazuri de utilizare. În lucrarea [76], autorii menționează avantajele paradigmei Publish-Subscribe, alături de adoptarea RTPS ca protocol de legătură pentru asigurarea interoperabilității între dezvoltări DDS provenite de la furnizori diferiți. Soluția concepută este utilizată pentru schimburi de date între vehiculele subacvatice, indicând fezabilitatea DDS în medii extreme. În lucrarea [77], au fost analizate provocările privind configurația QoS pentru sistemele în timp real, fiind evidențiată de asemenea dificultatea implementării aplicațiilor cu interoperabilitate ridicată, din cauza lipsei standardelor DDS. În consecință, înțelegerea modelelor dinamice DDS și aplicarea acestora la capacitate maximă reprezintă o sarcină complexă. Autorii oferă, de asemenea, un tool de dezvoltare ce presupune reutilizarea software-ului pe baza a diferite implementări DDS, util în arhitecturi cu un număr mare de dispozitive IoT eterogene.

5.2.2. DDS – middleware în ROS2

The Robot Operating System 2 (ROS 2) este un software framework open-source, ce constă în algoritmi, drivere și tool-uri pentru construirea de aplicații complexe pentru roboți, compatibile cu diverse echipamente hardware. Acesta oferă un sistem de comunicare adaptat pentru cazurile de utilizare IIoT și mecanisme de

vizualizare și simulare, utile în definirea arhitecturilor și în monitorizarea evoluțiilor sistemelor complexe, la un cost redus. Cu un întreg ecosistem disponibil pentru dezvoltare, ROS 2 câștigă popularitate atât în sectorul de cercetare, cât și în cel industrial (a se vedea [78,79]). Integrarea altor tehnologii, concepute pentru detectarea obstacolelor, procesarea imaginilor, precum și gestionarea dependențelor între componente, este accesibilă. Acest lucru a dus la adoptarea rapidă a ROS2 ca soluție dezirabilă pentru implementarea unor aplicații complexe pentru arhitecturi distribuite specializate în controlul roboților și odometrie. Ca middleware de comunicare, DDS a fost validat ca o soluție adecvată pentru ROS 2 [80], oferind posibilități de comunicare decentralizată între componente și extinderea capacităților în vederea îndeplinirii cerințelor stricte de funcționare în timp real prin intermediul paradigmei Publish-Subscribe.

Ca parte a ROS 2, modul de publicare implicit utilizat de DDS este modul asincron, ceea ce înseamnă că datele sunt copiate într-o coadă de așteptare, iar controlul asupra operațiunii de publicare va fi preluat de un alt thread, în fundal. Acest thread va fi responsabil de consumul datelor din coada de așteptare, lăsând thread-ul principal disponibil chiar înainte ca datele să fie expediate. Deși se oferă opțiuni în legătură cu procesul de publicare, acest mod este destinat nodurilor care nu au așteptări rapide și nu sunt responsabile de operațiuni critice în materie de timp. Cel de-al doilea mod disponibil este modul de publicare sincronă, în care controlul asupra operațiunii de publicare este deținut de thread-ul principal. Nu sunt executate alte funcții înainte de expedierea datelor, datorită anumitor mecanisme existente care împiedică alte părți ale aplicației să interfereze cu thread-ul principal. Avantajele modului sincron constă într-un control mai bun asupra intervalelor de timp necesare în care se preconizează că se vor expedia datele, respectiv asupra intervalelor de timp în care volume mai mari de date sunt livrate cu o latență minimă către nodurile de sistem responsabile de evenimente critice. Configurația celor două moduri se poate realiza cu ușurință, confirmând astfel nivelul de flexibilitate acordat de DDS în cazul sistemelor complexe cu cerințe diverse.

Includerea ROS 2 în proiecte academice, se află într-o fază incipientă. Principalul obstacol este reprezentat de dificultatea migrării de la ROS 1, din cauza multiplelor cerințe de actualizare tehnologică. Conform studiului [81], o consecință directă a acestui fapt este reducerea capacității de utilizare a sistemelor și dezvoltărilor existente. Cu toate acestea, se așteaptă ca ROS 2 să înlocuiască, în general, versiunea anterioară, datorită noilor caracteristici ce sunt adaptate la cerințele prezente în scenariile sectorului IoT. DDS poate fi perceput ca un contribuitor principal la avansarea și extinderea fezabilității din perspectiva comunicației pe Ethernet. În studiul [82], importanța DDS ca parte a ROS2 este detaliată, împreună cu importanța a diverse dezvoltări adiționale DDS, pentru atingerea obiectivelor majore ale ROS2 în ceea ce privește: cooperarea roboților industriali, utilizarea unor sisteme integrate mai mici cu resurse limitate, constrângerile de funcționare timp real ce sunt prezente în numeroase cazuri de utilizare industrială, precum și comunicarea prin intermediul rețelelor instabile. Autorii propun o soluție pentru a conecta în mod dinamic o implementare DDS adecvată, compatibilă cu situații specifice, ca middleware la ROS2. Se așteaptă ca DDS să adapteze și să îmbunătățească ROS2, generând o soluție compatibilă cu arhitecturi complexe de control industrial. În acest fel, protocolul DDS este validat din punct de vedere al eficienței, iar potențialul mare poate extinde aplicabilitatea și către alte domenii.

5.2.3. OPC UA : în Cercetare și în Industrie

Protocolul OPC UA este prezent în sectorul automatizărilor industriale ca principală tehnologie, capabilă să ofere o mare varietate de funcționalități la toate nivelurile modelului OSI (Open Systems Interconnection), în conformitate cu cerințele industriale în creștere. OPC UA pune la dispoziția utilizatorului paradigma clasică Server/Client, având o aplicabilitate dominantă în acest moment în multiple sectoare, cum ar fi industria apei, industria de producție, industria energetică și altele. Mai recent, pe baza noilor specificații, a devenit disponibilă și paradigma Publish-Subscribe, ca răspuns la cerințele tot mai mari de funcționare în timp real existente în cadrul progreselor către Industry 4.0. Din perspectiva domeniului de cercetare, OPC UA reprezintă un subiect major atât pentru mediul academic cât și pentru cel industrial. Au fost efectuate numeroase studii [83,64] focalizate pe diverse funcționalități ale protocolului OPC UA, iar rezultatele au confirmat mereu eficiența tehnologiei în contextul industrial. Odată cu apariția noilor specificații [31], a fost definit mecanismul Publish-Subscribe, iar studiile mai recente au avut ca temă principală comparația între noile capacități ale protocolului și cerințele industriale în continua expansiune.

În lucrările [23,45], OPC UA a fost implementat în contextul comunicație de tipul car-to-infrastructure, demonstrând fezabilitatea protocolului în cazurile de utilizare în care interacționează sectoare diferite. În lucrarea [84], autorii au implementat un prototip de integrare a tehnologiilor OPC UA și TSN în standardul AUTOSAR Adaptive. În lucrarea [52], mecanismul OPC UA Publish-Subscribe este analizat în detaliu și comparat cu alte mecanisme similare specifice altor protocoale de comunicare, fiind sugerate îmbunătățiri arhitecturale pentru cazuri particulare de utilizare industrială. În dezvoltarea [85], aplicabilitatea protocolului OPC UA a fost extinsă către scenarii de transmitere a imaginilor, utilizându-se mai multe canale UDP Publish-Subscribe de transmisie. Multitudinea de scenarii în care OPC UA a fost implementat cu rezultate semnificative, indică nivelul ridicat de robustețe și flexibilitate, iar astfel protocolul s-a propagat ca tehnologie de referință în sectorul industrial. Odată cu adăugarea mecanismului Publish-Subscribe, capacitățile sporite i-au extins relevanța și în alte domenii. În contextul necesităților de funcționare în timp real, asociate conceptului Industry 4.0, OPC UA oferă o calitate ridicată a serviciilor. Este de așteptat ca OPC UA să își continue expansiunea și treptat să se adapteze la orice situații de funcționare din cadrul IIoT.

5.2.4. TSN: Evoluție, Provocări și Așteptări

Când se iau în considerare cele mai promițătoare protocoale de comunicare prin Ethernet, pentru arhitecturi distribuite, trebuie menționată și tehnologia TSN, deoarece aceasta joacă un rol important în toate cazurile de utilizare unde sunt necesare garanții asupra timpilor de expediere a mesajelor în rețea. Constituind un set de standarde și mecanisme care pot fi configurate pentru atingerea mai multor obiective, adoptarea TSN poate genera numeroase provocări pentru orice sistem de comunicare. Unul dintre obstacolele principale este reprezentat de numărul mare de interdependențe pe care le implică utilizarea TSN, fiind necesare numeroase ajustări pentru toate dispozitivele și aplicațiile participante, ceea ce presupune un efort semnificativ din punct de vedere al dezvoltării. Un alt obstacol este reprezentat de

numeroasele posibilități de configurare, disponibile pentru fiecare standard în parte. Astfel, stabilirea configurației potrivite devine un proces complex ce poate genera probleme de scalabilitate. O schemă standard de configurare este un pas necesar în viitorul apropiat, însă este o sarcină dificilă, dacă se iau în considerare toți parametrii ce ar putea influența o astfel de procedură (numărul de dispozitive, capacitățile fiecărui dispozitiv, setul cel mai eficient de mecanisme necesare pentru un anumit scenariu, cerințele de funcționare în timp real, limitările de securitate și siguranță etc.). Abordarea cea mai eficientă, adoptată pe moment în industrie, presupune mai întâi o identificare corectă a mecanismelor TSN necesare pentru cazul de utilizare dorit și, ulterior, se realizează configurarea fiecărui mecanism, optimizând sistemul în mod treptat.

Deși sunt cunoscute beneficiile considerabile oferite de TSN, se observă un progres lent atât în plan academic cât și în plan industrial, în legătură cu adoptarea TSN pentru sistemele de comunicație pe Ethernet. Principalul motiv pentru acest fapt este reprezentat de cantitatea limitată de dispozitive de pe piață, capabile să suporte tehnologia TSN. Cu toate acestea, companiile de hardware specializate încep să se axeze pe dezvoltarea de dispozitive adecvate pentru cazuri de utilizare IIoT, cu cerințe de funcționare în timp real, ce implementează standarde TSN specifice la nivelul legăturii de date (Data Link Layer) al modelului OSI. Un exemplu relevant poate fi observat în domeniul automotive, unde scenariile de comunicație pe Ethernet între controlere devin tot mai frecvente, iar echipamentele hardware trebuie să asigure o funcționare sincronizată pentru operațiunile critice de transmisie și recepție de date.

Îmbunătățirea infrastructurilor de comunicare pe Ethernet reprezintă următorul pas pentru trecerea la sisteme complexe IoT în industrie, iar tehnologia TSN poate facilita acest proces de evoluție, oferind garanții asupra timpilor de livrare și recepție de date la nivel de rețea. Deși există numeroase provocări, acest subiect rămâne de mare interes pe viitor atât din punct de vedere științific cât și industrial.

5.3. Arhitectura Sistemului

Etapa actuală de cercetare se axează pe două protocoale de comunicare majore, DDS și OPC UA, specializate în comunicarea pe Ethernet, urmărindu-se funcționarea în context comun prin scenarii diversificate. Aceste tehnologii sunt adaptate la o gamă largă de aplicații, sunt capabile să satisfacă cerințele de sincronizare și de funcționare în timp real, fiind adecvate pentru sistemele complexe și scalabile. DDS și OPC UA oferă un set de mecanisme de mare potențial pentru scenariile industriale unde se urmărește progresul către următorul nivel de conectivitate. Chiar dacă există similitudini din punct de vedere al capacităților, al mecanismelor oferite, și al tehnologiilor componente, se poate observa o diferențiere importantă la nivel de aplicabilitate. OPC UA este mai mult prezent în automatizările industriale și în fabrici, asigurând o comunicare robustă între diverse tipuri de echipamente, iar DDS este mai prezent în procedurile de control și cooperare multi-robot, prin integrarea sa ca middleware în ROS2. Odată cu evoluția preconizată către IIoT și Industry 4.0, luând în considerare expansiunea rapidă către mai multe sectoare industriale, probabilitatea ca tehnologiile majore precum OPC UA și DDS să se intersecteze și să coopereze devine destul de mare.

Pentru a crea un scenariu autentic și pentru a aplica abordări multiple care să răspundă preocupărilor actuale din industrie (interacțiunea între multiple tehnologii în cadrul aceleiași arhitecturi, controlul unor procese industriale complexe, funcționarea

100 | 5. Coexistența protocoalelor DDS și OPC UA în contextul Industry 4.0 și a comunicației în timp real

sistemului în timp real), a fost definită o arhitectură simetrică cu mai multe noduri asigurate celor două protocoale. Astfel, există trei noduri de comunicare pentru protocolul DDS și alte trei noduri pentru protocolul OPC UA. Fiecare nod are un rol specific și rulează pe un dispozitiv diferit, iar fiecare tip de dispozitiv execută aplicațiile în cadrul unui sistem de operare nativ sau virtualizat bazat pe Linux.

Arhitectura definită a fost concepută pentru a respecta atât scenariile de comparare a performanțelor, cât și scenariile de interacțiune prin intermediul unei aplicații gateway, cu scopul de a demonstra compatibilitatea și fezabilitatea dintre DDS și OPC UA și de a oferi o definiție generală a criteriilor și așteptărilor din punct de vedere arhitectural. Atunci când se adoptă sau se selectează o tehnologie pentru anumite cazuri de utilizare, în special în cazul tehnologiilor de ultimă generație, procesul de dezvoltare este încetinit de numeroasele incertitudini generate de provocări nedocumentate. Pentru a se evita aplicarea strategiei clasice de încercare-eroare, comunitatea științifică trebuie să furnizeze orientări, criterii de performanță, perspective arhitecturale și să evidențieze avantajele și dezavantajele într-o manieră neutră, conturând o viziune cuprinzătoare ce detaliază toate posibilitățile de dezvoltare. Arhitectura sistemului actual poate fi observată în Figura 31.

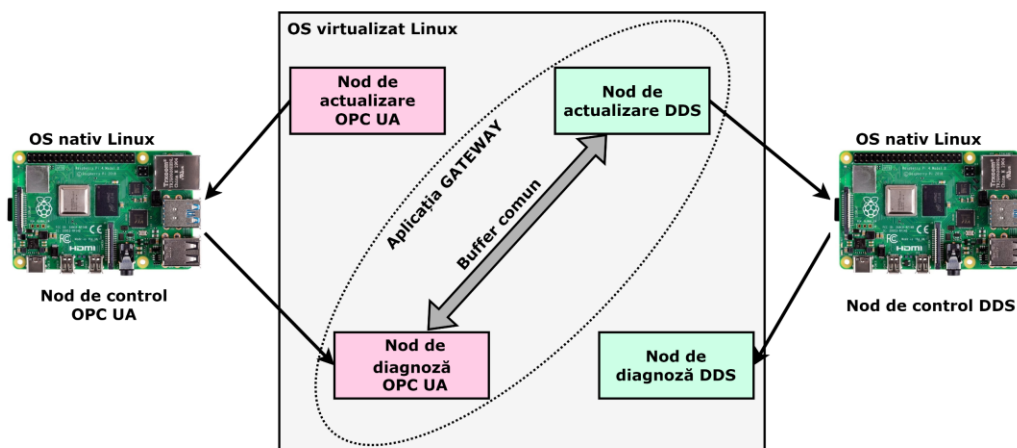


Figura 31. Arhitectura Sistemului

Nodurile de control OPC UA și DDS sunt definite în scopul simulării controlului, și sunt concepute pentru a fi principalii consumatori de date. Aceste noduri de control rulează pe dispozitive Raspberry Pi 4 cu un sistem de operare nativ bazat pe Linux. Acestea trebuie să fie percepute ca fiind segmentul principal de control al arhitecturii, într-un scenariu real (de exemplu: controlul unui dispozitiv PLC, nodul de control al unui robot sau al unei entități de control în linia de producție). Ambele noduri sunt proiectate cu două subcomponente de comunicare:

- subcomponentă de tip Subscriber ce primește informații actualizate de la nodul de actualizare, la diferite intervale de timp;
- o subcomponentă Publisher ce trimite informațiile către un nod de diagnostică, pentru eventuale operațiuni de diagnostică sau de siguranță specifice unui anumit proces industrial.

Nodurile de actualizare sunt principalii furnizori și distribuitori de date. Acestea oferă posibilități complexe de configurare, fiind accesibile numai de către nodurile de control.

Nodurile de diagnoza recepționează date numai de la nodurile de control și pot efectua operațiuni de verificare pentru datele distribuite sau pot să funcționeze ca o componentă gateway, permițând interacțiunea dintre OPC UA și DDS pentru scenarii de control complexe (de exemplu: trimiterea de date de la un dispozitiv PLC controlat de nodul de control OPC UA, către un motor acționat de nodul de control DDS). Atunci când sistemul este configurat pentru cazuri de utilizare de tip gateway, entitățile complementare corespondente celor două protocoale (nodul de diagnoza OPC UA - nodul de actualizare DDS și viceversa) fac schimb de date prin intermediul unui buffer specializat la care au acces comun. Această arhitectură a sistemului oferă flexibilitate sporită, permite monitorizarea fiecărei operațiuni conform cerințelor de funcționare în timp real și gestionează multiple situații de control, fiind necesare eforturi minime de configurare.

5.3.1. Comparație cu alte dezvoltări relevante

Au fost realizate și alte studii ce plasează tehnologiile actuale într-un context comun, fiecare cu abordări, obiective și realizări semnificativ diferite. În lucrarea [86], autorii au evaluat și comparat performanțele OPC UA și DDS alături de alte protocoale de comunicare de interes, prezentând durată completă de transmisie-recepție în funcție de dimensiunea pachetelor în diferite scenarii de stres.

Arhitectura conceptului actual conține mai multe entități și rute implicate, și elaborează noduri specializate cu scopuri diferite, implementând aplicații multithreading pentru toate nodurile. De asemenea, se urmărește reacția sistemului de operare în scenarii complexe, până la identificarea limitelor temporale de transmisie și recepție. Cu alte cuvinte, dezvoltarea curentă nu evaluează doar protocoale implicate, ci mai degrabă evaluează interacțiunea, eficiența și capacitatea de utilizare în scenarii IIoT cu precondiții specifice preconizate, cum ar fi dispozitive multiple, sisteme de operare cu diferite particularități, rețele neideale și cerințe diferite de funcționare în timp real. Astfel de evaluări orientate pe cazuri concrete de utilizare și pe procese industriale, se pot dovedi utile atunci când se ia în considerare adoptarea unei tehnologii pentru un anumit domeniu. Comunitatea științifică are potențialul de a fi principalul facilitator al evoluției implementărilor industriale actuale, arhitecturilor, standardelor și cerințelor industriale, spre Industry 4.0 și IIoT.

O altă diferență semnificativă față de studiul [86] este reprezentată de criteriile de evaluare stabilite pentru conceptul actual. Primul criteriu implică:

- Testarea individuală a capacității de reacție în timp real a fiecărei operațiuni (publicare, recepție) la nivel de dispozitiv;
- Comparația pe baza rezultatelor ideale.

Arhitectura repartizată în cadrul mai multor dispozitive și sistemele de operare prezente pe fiecare nod, sunt factori cheie în evaluare. Un obiectiv principal în acest stadiu este reprezentat de rezultate procentuale obținute, ce oferă posibilități de estimare în condiții industriale reale, unde fiecare operațiune de la fiecare nivel al arhitecturii, are așteptări concrete de funcționare în timp real. Un al doilea obiectiv este legat de similaritățile și diferențele dintre OPC UA și DDS în scenarii arhitecturale

identice, cu rezultate procentuale asemănătoare pentru diversele intervale de timp. Se urmărește validarea compatibilității pentru situațiile de utilizare în context comun, în cadrul viitoarelor arhitecturi industriale cu precondiții variate, departe de cazurile ideale. Aceste precondiții foarte variate sunt parte a sistemelor complexe de comunicare, și sunt prezente în toate arhitecturile ce utilizează mai multe tehnologii, fiecare cu propriile dependențe și cerințe, reprezentând o cauză majoră a progresului lent în cadrul IIoT. Cel de-al doilea criteriu implică stocarea tuturor datelor recepționate pentru o perioadă de timp (data buffering) și compararea conținutului și cantității acestora cu setul ideal de date așteptat în acea perioadă prestabilită de timp. Obiectivul nu constă doar în supravegherea operațiunii executate (publicare, recepționare), ci și în validarea datelor primite, luând în considerare faptul că stabilitatea rețelei joacă un rol important în rezultatul final.

În lucrarea [87], autorii au convertit mesaje MQTT și AMQP generate de un Publisher OPC UA, în mesaje DDS ce sunt expediate între un o furnizor și un receptor DDS. Diferențele majore prezente în implementarea actuală sunt legate de comportamentul în timp real și de perspectivele arhitecturale. MQTT și AMQP sunt protocoale de transport opționale pentru OPC UA și ambele funcționează în cadrul unui design bazat pe un broker. Protocolul de transport utilizat pentru dezvoltarea actuală este UADP (UA Datagram Protocol), un protocol de transport definit de către Fundația OPC pentru cazurile de utilizare ce necesită comunicare UDP și posibilități multicast IP. Soluția de transport UADP oferă o conexiune directă între entitățile OPC UA Publisher și Subscriber, prin intermediul infrastructurii de rețea, oferind posibilități de sincronizare între dispozitivele gazdă. Diferit de designul bazat pe broker, această metodă de conectare directă, necesită o înțelegere completă a codului OPC UA (la toate nivelurile modelului OSI) din partea tuturor entităților, așa cum este necesar și în cazul protocolului DDS. Aplicația de tip broker în sine este considerată software suplimentar, iar funcționalitatea sa (distribuirea de mesaje) este considerată în afara domeniului de aplicare al specificațiilor (conform [24]), fiind o aplicație de comunicare MQTT/AMQP ca pentru orice alt furnizor și consumator MQTT/AMQP (fără a fi necesară înțelegerea codului OPC UA). Designul bazat pe broker este în general specific scenariilor cu latență ridicată (de obicei interacțiunea cu arhitecturi Cloud), iar în cazul comunicării OPC UA, entitățile Publisher și Subscriber sunt doar clienți de mesagerie [24]. În cazul aplicației actuale de tip gateway, toate entitățile și componentele implicate utilizează o strategie de conectare directă care asigură un nivel ridicat de determinism temporal pentru toate procedurile, unde toate componentele dezvoltate au o înțelegere completă a codurilor specifice DDS și OPC UA. Astfel se execută schimburi complete de informații între entități Publisher și Subscriber OPC UA (nu între entități specifice nivelului de transport MQTT/AMQP). Prin această abordare se utilizează modul de expediere a mesajelor definit de OPC, asigurându-se latență redusă și securitate suplimentară, fiind folosită infrastructura de rețea pentru comunicarea bazată pe Ethernet.

5.4. Studii de caz și Rezultate

Prezenta cercetare se focalizează pe două studii de caz. Primul vizează observații asupra modului în care fiecare protocol de comunicare reacționează în funcție de sistemul de operare pentru operațiile executate la diferite recurențe temporale. Se analizează cât de eficient se comportă tehnologiile atunci când rulează

pe sisteme de operare nespecializate (de uz general), fără mecanisme dedicate funcționării în timp real, utilizându-se, de asemenea, o rețea ce nu oferă garanții de timp pentru transmiterea pachetelor. Al doilea studiu de caz propune o soluție gateway pentru scenarii în care interacțiunea între cele două protocoale este necesară. Comutarea între cele două cazuri de utilizare poate fi realizată cu efort minim la faza de configurare și fără modificări arhitecturale.

5.4.1. Studiu de caz 1

Dezvoltarea de aplicații de comunicare ce implică dispozitive multiple, trebuie să asigure o eficiență sporită și o bună gestionare a resurselor pe toate dispozitivele. În studiul de caz actual, pentru ambele protocoale a fost utilizată paradigma Publish-Subscribe, vizând un comportament în timp real la nivelul aplicației, pentru toate nodurile.

Inițializarea comunicării DDS se face printr-o serie de pași similari la nivelul tuturor dispozitivelor. În primul rând, atât în cazul furnizorului cât și al receptorului, un participant este inițializat cu informații specifice domeniului. Un participant este o entitate individuală (cu rol fie de Publisher, fie de Subscriber) și poate participa la schimbul de date în interiorul domeniului. După ce participantul este configurat, acesta trebuie adăugat la domeniu. Al doilea pas este legat de rolul fiecărui participant. Rolul este selectat prin tipuri de date predefinite, specifice fie unui Publisher, fie unui Subscriber, care au nevoie de o anumită configurație bazată pe unele atribute particulare. După ce configurația este finalizată, rolul este atribuit participantului. Rolul în sine ar putea fi perceput ca o subcomponentă a participantului, din perspectivă software. Al treilea pas conține multiple configurații axate pe rețea, pe topic-urile disponibile sau pe operații de corelare între Publisher și Subscriber, în funcție de cazul de utilizare.

Schimbul de date OPC UA este inițializat într-un mod similar cu implementările din capitolele anterioare, respectiv din studiile [52,45]. Pentru dezvoltarea entității Publisher, se inițializează conexiunea de tip Publish-Subscribe, după care se definește DataSet-ul și informația de interes se atribuie câmpului de date din structura mesajului de rețea. Ultimul pas este configurarea subcomponentelor Writer Group și Data Set Writer. Pentru entitatea Subscriber, după ce conexiunea de tip Publish-Subscribe este inițializată, se efectuează decodificarea, filtrarea și extragerea datelor pe baza mesajelor OPC UA recepționate recurent.

Obiectivul principal al studiului de caz actual este reprezentat de analiza comportamentului protocoalelor DDS și OPC UA într-o arhitectură similară, cu dispozitive și sisteme de operare similare, utilizate, în scenarii identice de funcționare în timp real. Pentru a observa mai bine comportamentul și pentru a cuantifica mai precis răspunsurile tehnologiilor la solicitările de funcționare în timp real, autorii au definit două criterii majore de evaluare. Criteriile nu au neapărat menirea de a afirma în urma analizei, că o tehnologie este mai bună decât cealaltă, ci mai degrabă de a stabili că răspunsurile nu sunt extrem de diferite, confirmând faptul că interacțiunea dintre OPC UA și DDS este posibilă și poate aduce valoare arhitecturilor de tip cross-domain, unde se preconizează utilizarea ambelor protocoale în viitor. În cazul sistemelor de operare Linux de uz general, ce nu au mecanisme adiționale dedicate pentru funcționarea în timp real și de asemenea, în cazul în care nu se utilizează tehnologii precum TSN, performanța în scenarii de funcționare în timp real este de așteptat să scadă pentru recurențele sub 10 milisecunde între procedurile de

transmisie sau de recepție. Ideea studiului de caz actual este de a prezenta într-o manieră cuantificabilă declinul de la recurențele de 10 milisecunde, la cele de 5, 2 și 1 milisecundă pentru ambele protocoale. Primul criteriu este reprezentat de apelurile de funcții pe care aplicația le poate executa în momentul în care intervalele de timp de execuție devin mai scurte. De exemplu, în cazul în care aplicația a stabilit un interval publicare de 5 milisecunde, rezultatul va compara numărul real de apeluri de funcții ce s-au produs în 10 milisecunde (un interval de timp în care execuția este încă garantată de sistemul de operare) cu rezultatele așteptate (pentru 5 milisecunde, rezultatul așteptat al apelurilor operațiunii de publicare este 2, în 10 milisecunde). Rezultatele vor fi măsurate cu ajutorul unui scheduler propriu, dezvoltat pentru acest studiu de caz, ce va rula pe un thread diferit de cel al operațiunilor de comunicare. Rezultatul nu este influențat de intervalul de timp stabilit pentru procedurile de publicare sau de recepție. Acuratețea scheduler-ului este verificată prin generarea unui semnal recurent la un interval prestabilit de timp, semnalul fiind măsurat de un osciloscop, astfel fiind garantată precizia măsurărilor efectuate. În rarele cazuri în care au apărut desincronizări, rezultatele nu au fost luate în considerare, iar testul a fost reluat. Acest component scheduler rulează pe toate dispozitivele ca un thread individual. Funcțiile de verificare concepute, care compară apelurile efective pentru procedurile de comunicare cu rezultatele așteptate, creează un jurnal detaliat pentru fiecare iterație a unui scenariu, oferind posibilitatea de a identifica atât desincronizările rețelei, cât și cele ale scheduler-ului, pentru întreaga arhitectură. Având mai multe noduri pe dispozitive diferite, toate aplicațiile corespondente nodurilor sunt multithreading, iar numărul de thread-uri este mai mare în cazul nodurilor de control ce au atribuite atât roluri de Subscriber, cât și de Publisher. O perspectivă mai bună asupra fiecărei aplicații multithreading asociate fiecărui nod, poate fi observată în Figura 32.

Cu o componentă scheduler prevăzută pentru comportamentul în timp real și funcții multiple de verificare pentru toate intervalele de timp prestabilite (10 ms, 5 ms, 2 ms, 1 ms), primul criteriu al studiului de caz este acoperit, iar rezultatele pot fi cuantificate sub forma de procente, pentru fiecare nod al sistemului.

Cel de-al doilea criteriu constă în analiza achiziției de date pe fiecare nod receptor, utilizând aceleași intervale de timp definite anterior. Un mecanism propriu de data-buffering a fost conceput pentru a stoca toate informațiile recepționate la nivelul fiecărei entități cu rol de Subscriber. La sfârșitul procesului, mecanismul de data-buffering generează un fișier de tip jurnal, în care sunt înregistrate toate valorile primite, iar acestea pot fi observate și comparate cu numărul de valori așteptate. Se obțin rezultate numerice procentuale, care permit identificarea desincronizărilor de rețea, sau ale dispozitivului. De asemenea, există posibilitatea utilizării mecanismului pentru implementarea conceptelor de siguranță, în scenarii elaborate. Diversitatea sistemelor de operare prezente pe fiecare dispozitiv permite o comparație între proceduri similare. Nodurile de diagnostică funcționează ca ultima entitate consumatoare din arhitectură, și permit o mai bună monitorizare a rezultatului final, în urma distribuirii datelor între trei noduri diferite. Acest mecanism de stocare a datelor, utilizează buffere configurabile, oferind un grad ridicat de flexibilitate indiferent de tipul de date utilizat. Întregul mecanism este utilizat pentru achiziția de date, de către thread-ul responsabil cu procedurile de comunicație, iar sursa de timp este obținută de la thread-ul pe care rulează componenta scheduler. Din acest punct de vedere, poate fi perceput ca un mecanism multithreading pentru toate nodurile care dețin rolul de abonat.

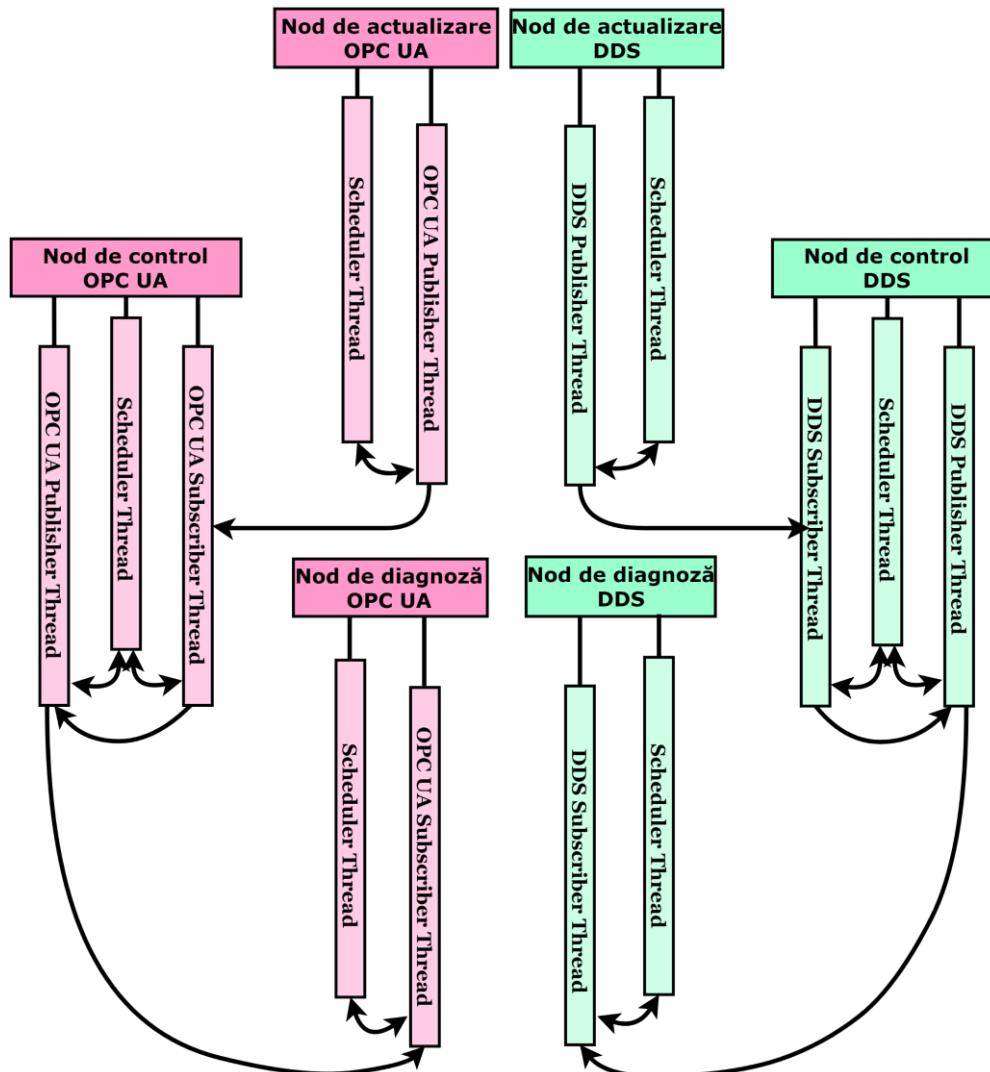


Figura 32. Nodurile multithreading din perspectiva arhitecturală

Având rezultate din ambele perspective, pe baza verificării apelurilor de funcții și pe baza mecanismului de data-buffering, este posibilă o monitorizare aprofundată a reacției protocoalelor OPC UA și DDS, în cadrul unei arhitecturi cu mai multe noduri de comunicație. Particularitățile și configurația sistemului asigură un context apropiat de starea generală actuală pentru adoptarea tehnologiilor vizate în scenariile IIoT. Sunt luate în considerare multiple provocări din domeniul industrial, evidențiindu-se numeroase aspecte pozitive și negative care au impact asupra progreselor Industry 4.0.

5.4.2. Rezultatele Studiului de caz 1

Implementarea studiului de caz 1, dezvoltarea mecanismelor de supraveghere specifice și dezvoltarea infrastructurii de comunicație pe toate dispozitivele, au fost realizate cu succes. Se confirmă astfel fezabilitatea arhitecturii și se obțin rezultate cuantificabile în legătură cu funcționarea în timp real, pentru toate nodurile, în conformitate cu cele două criterii menționate mai sus. Un număr considerabil de teste au fost efectuate la nivelul tuturor nodurilor, cu scopul de a minimiza relativitatea comportamentului, ca urmare a particularităților tehnologice, imprevizibilității sistemelor de operare în situații critice și posibilității de desincronizare a dispozitivelor în rețea.

Rezultatele procesului de verificare a apelurilor de funcții, pot fi observate în tabelele 4-7 pentru DDS și în tabelele 8-11 pentru OPC UA, conform primului criteriu al studiului de caz. Tabelele prezintă ratele de succes pentru fiecare recurență de timp.

Tabel 4. Nod de actualizare DDS – OS Virtualizat Linux – Operațiuni de Publicare.

Operațiuni de Publicare – Verificarea Execuțiilor Recurente			
10 ms	5ms	2ms	1ms
≈100%	≈90%	≈74%	≈64%
Număr TOTAL de teste: 2790			

Tabel 5. Nod de control DDS – OS Nativ Linux – Operațiuni de Publicare.

Operațiuni de Publicare – Verificarea Execuțiilor Recurente			
10 ms	5ms	2ms	1ms
≈100%	≈93%	≈84.6%	≈77%
Număr TOTAL de teste: 2865			

Tabel 6. Nod de control DDS – OS Nativ Linux – Operațiuni de Recepție.

Operațiuni de Recepție – Verificarea Execuțiilor Recurente			
10 ms	5ms	2ms	1ms
≈100%	≈85%	≈65%	≈48.5%
Număr TOTAL de teste: 2805			

Tabel 7. Nod de diagnoză DDS – OS Virtualizat Linux – Operațiuni de Recepție.

Operațiuni de Recepție – Verificarea Execuțiilor Recurente			
10 ms	5ms	2ms	1ms
≈100%	≈85%	≈65%	≈47%
Număr TOTAL de teste: 3015			

Tabel 8. Nod de actualizare OPC UA – OS Virtualizat Linux – Operațiuni de Publicare.

Operațiuni de Publicare – Verificarea Execuțiilor Recurente			
10 ms	5ms	2ms	1ms
≈100%	≈95%	≈81.2%	≈56%
Număr TOTAL de teste: 2685			

Tabel 9. Nod de control OPC UA – OS Nativ Linux – Operațiuni de Publicare.

Operațiuni de Publicare – Verificarea Execuțiilor Recurente			
10 ms	5ms	2ms	1ms
≈100%	≈100%	≈87%	≈56%
Număr TOTAL de teste: 2970			

Tabel 10. Nod de control OPC UA – OS Nativ Linux – Operațiune de Recepție.

Operațiune de Recepție – Verificarea Execuțiilor Recurente			
10 ms	5ms	2ms	1ms
≈100%	≈100%	≈91%	≈85%
Număr TOTAL de teste: 3015			

Tabel 11. Nod de diagnoză OPC UA – OS Virtualizat Linux – Operațiune de Recepție.

Operațiune de Recepție – Verificarea Execuțiilor Recurente			
10 ms	5ms	2ms	1ms
≈100%	≈87.5%	≈77%	≈64%
Număr TOTAL de teste: 3015			

Din perspectiva DDS, așa cum se poate observa în tabelele 4-7, implementarea oferă asigurarea că schimbul de date poate fi mai rapid de 10 milisecunde, pe ambele tipuri de sisteme de operare, și chiar și în cazul recurenței de 5 milisecunde, datele sunt distribuite în mod eficient, cu o posibilitate de eșec neglijabilă, dar existentă. Pe baza cifrelor, pentru intervalele de timp între 10 milisecunde și 5 milisecunde, discuția ar trebui să se axeze, pe cât de tolerant poate fi sistemul într-un schimb de date de tipul Publish-Subscribe, în care este de așteptat ca volumele și vitezele datelor distribuite să varieze. Din perspectiva sistemului de operare, experimentul a demonstrat că sistemele de operare native bazate pe Linux oferă o eficiență sporită și reprezintă o opțiune mai stabilă, cu probabilități mai mici de întârzieri și desincronizări, diferența fiind cuantificată. Din perspectiva rolurilor, procedura de tip Publish a fost executată mai bine decât procedura de tip Subscribe, pe ambele variante de sisteme de operare, în actualele condiții nefavorabile. Totuși, acest lucru s-ar putea dovedi neimportant în cazurile ideale în care, pentru ambele roluri, schimbul de date este gestionat prin mecanisme suplimentare. Pentru intervale de timp mai mici de 5 milisecunde, probabilitatea de eșec nu poate fi neglijată, ceea ce face ca realizarea obiectivelor de funcționare în timp real să nu fie posibilă în sistemul actual. Pentru intervalele de 1 milisecundă și 2 milisecunde, cifrele sunt foarte apropiate de la un tip de sistem de operare la altul, ceea ce confirmă faptul că, din punct de vedere al funcționării în timp real, există prea multe improbabilități pentru a face schimb eficient de date la astfel de recurențe, fără mecanisme speciale.

Implementarea OPC UA confirmă, de asemenea, că schimbul de date a fost realizat cu succes, între toate nodurile. Rezultatele sunt mai bune în cazul dispozitivelor cu sistem de operare nativ bazat pe Linux, obținând încă 100% din execuțiile funcțiilor și la recurența de 5 milisecunde. Pentru intervale sub 5 milisecunde, chiar și în cazul dispozitivelor care funcționează cu un sistem de operare nativ, probabilitatea de eșec este ridicată și confirmă necesitatea unor tehnologii precum TSN și alte mecanisme, care să garanteze schimbul de date la recurențele sub 5 milisecunde. Diferența față de comportamentul protocolului DDS în aceleași circumstanțe nu este semnificativă, și nu contribuie decisiv la creșterea eficienței în contextul actual. Din perspectiva rolului, procedura de Subscribe are rezultate mai bune decât cea de Publish, la recurențe sub 5 milisecunde. Cu toate acestea, implementările precum [52], propun o abordare diferită, menită să crească eficiența operațiunilor de Publish și Subscribe, însă un astfel de design are impact asupra întregii arhitecturi. Pe baza rezultatelor procesului de verificare a apelurilor de funcții în configurații identice, ambele protocoale de comunicare au comportamente apropiate în scenarii de funcționare în timp real, fiind capabile să facă schimb de date chiar și fără integrarea în arhitectură a unor mecanisme specializate. Atât DDS, cât și

108 | 5. Coexistența protocoalelor DDS și OPC UA în contextul Industry 4.0 și a comunicației în timp real

OPC UA pot fi considerate soluții capabile și flexibile pentru dezvoltarea de aplicațiile IIoT.

Cel de-al doilea criteriu al studiului de caz 1 se axează pe mecanismul de data-buffering, utilizat pentru a evidenția modul în care sistemul de operare și desincronizarea dispozitivului influențează datele recepționate. Pe baza acestui mecanism de stocare a datelor, se generează rezultate statistice la diferite intervale de timp. Rezultatele pot fi observate în Figura 33.

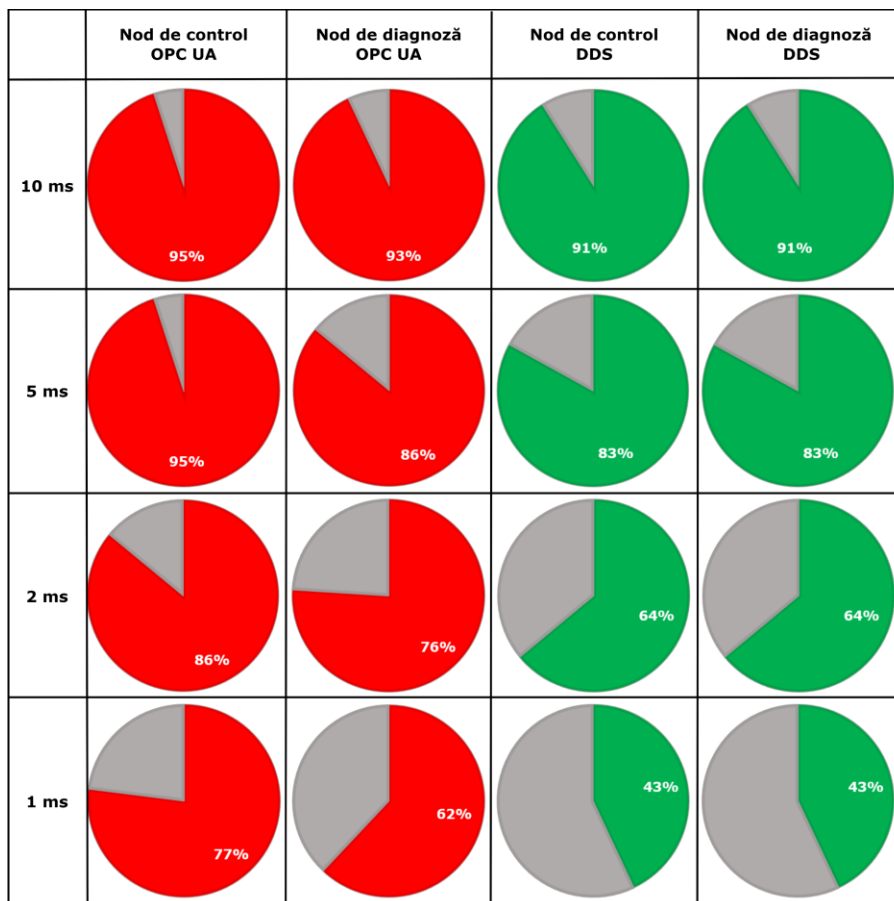


Figura 33. Rezultate procentuale bazate pe rata de succes a mecanismului data-buffering

Rezultatele mecanismului data-buffering la nivelul entităților Subscriber, confirmă rezultatele procesului de verificare a apelurilor de funcții. Cifrele sunt proporționale pentru ambele criterii, cu valori procentuale mai mici, conform așteptărilor, pentru procesul de stocare a datelor, validând astfel impactul suplimentar pe care stabilitatea rețelei îl are asupra procedurilor, față de impactul deja existent generat de limitările funcționării în timp real ale sistemului de operare. Deși la intervale mai mici de 5 milisekunde, diferențele dintre protocoale par să se accentueze ușor, există încă incertitudini cu privire la cât de mult pot fi afectate

interacțiunea și schimbul de date dintre protocoale, iar în acest caz, un scenariu de tip gateway este de mare interes.

5.4.3. Studiu de caz 2

Pentru studiul de caz 2, obiectivul principal a fost implementarea unei aplicații gateway în cadrul arhitecturii actuale, oferind posibilitatea de a confirma fezabilitatea DDS și OPC UA în scenariile IIoT ce implică o gamă largă de echipamente. Existența unei aplicații de tip gateway pentru operațiunile de control și schimb de date între domenii separate, prin intermediul tehnologiilor Ethernet, confirmă capacitățile înalte ale celor două protocoale de comunicație și extinde posibilitățile de realizare a unor progrese viitoare. Impactul protocoalelor Ethernet în cazurile de utilizare în timp real este dificil de estimat. Chiar dacă există numeroase soluții disponibile și conforme cu cerințele de funcționare în timp real, interpretarea generală nu trebuie să se concentreze pe rapiditatea cu care un protocol de comunicație este capabil să livreze date, ci mai degrabă pe cât de stabilă este livrarea, la ce intervale de timp poate eșua schimbul de date și care sunt factorii care pot influența performanța protocolului ales. Dezvoltarea actuală sugerează o exemplificare detaliată a modului în care impactul descris mai sus poate fi observat și înțeles, chiar și în scenarii în care datele sunt livrate la recurențe rapide.

Contextul industrial al aplicației gateway OPC UA-DDS este evidențiat folosind arhitectura actuală și propunând pentru nodurile de control OPC UA și DDS simularea de procese industriale specifice domeniilor de automatizare și control al roboților. În cazul unui nod de control OPC UA, informațiile sosesc de la nodul de actualizare OPC UA și se efectuează operații logice pentru a obține două tipuri diferite payload, simulând intrările care ar putea fi preluate de la un dispozitiv PLC. Cele două tipuri de mesaje sunt trimise alternativ către aplicația gateway, care acționează ca Subscriber OPC UA pentru nodul de control OPC UA și ca Publisher DDS pentru nodul de control DDS. Odată ce mesajul este recepționat de nodul de control DDS, se generează un semnal digital în funcție de tipul mesajului primit (corespunzător stărilor high sau low), simulând controlul unui actuator ce necesită impulsuri recurente rapide. Semnalul digital generat este monitorizat cu ajutorul unui osciloscop. Astfel, pentru fiecare interval de timp stabilit pentru schimbul de date între noduri, semnalul este măsurat, oferind posibilitatea de a confirma durata și frecvența pulsurilor, ceea ce înseamnă că întreaga propagare a datelor prin arhitectură poate fi monitorizată. Desincronizările preconizate și problemele de stabilitate care pot apărea pentru intervalele sub 10 milisecunde, pot fi vizualizate cu ușurință, iar degradarea calitativă a semnalului generat poate fi observată în mod concret. Acest demers oferă o perspectivă sugestivă asupra necesității unor tehnologii suplimentare, capabile să garanteze sincronizarea între dispozitive și stabilitatea rețelei în cazurile de utilizare IIoT. Aplicația gateway facilitează propagarea datelor între mai multe noduri, dispozitive, protocoale și sisteme de operare distincte, fără a fi nevoie de o configurație diferită pentru software-ul deja implementat. Soluția confirmă faptul că atât OPC UA, cât și DDS sunt compatibile și fezabile pentru astfel de scenarii, chiar și în sisteme neideale. Acest tip de experiment sporește potențialul tehnologiilor, către aplicații mai complexe și scalabile, pentru mai multe domenii industriale.

5.4.4. Rezultatele Studiului de caz 2

Implementarea studiului de caz 2 a fost un succes. Aplicația gateway facilitează schimbul de date între nodurile de control la recurențe configurabile, iar rezultatele sunt observabile și măsurabile, evidențiind impactul desincronizărilor dispozitivelor și al instabilității rețelei. Rezultatele actuale sunt în concordanță cu cele din studiul de caz 1, care au estimat o scădere a eficienței pentru livrarea datelor, cu fiecare transmisie realizată la un ciclu mai rapid. Evoluția propagării datelor prin arhitectura cu noduri multiple este vizibilă prin semnalul digital generat de nodul de control DDS, pe baza mesajelor specifice primite de la nodul de control OPC UA. Rezultatele studiului de caz 2 pot fi observate în Figurile 34-38 pentru diversele recurențe utilizate.

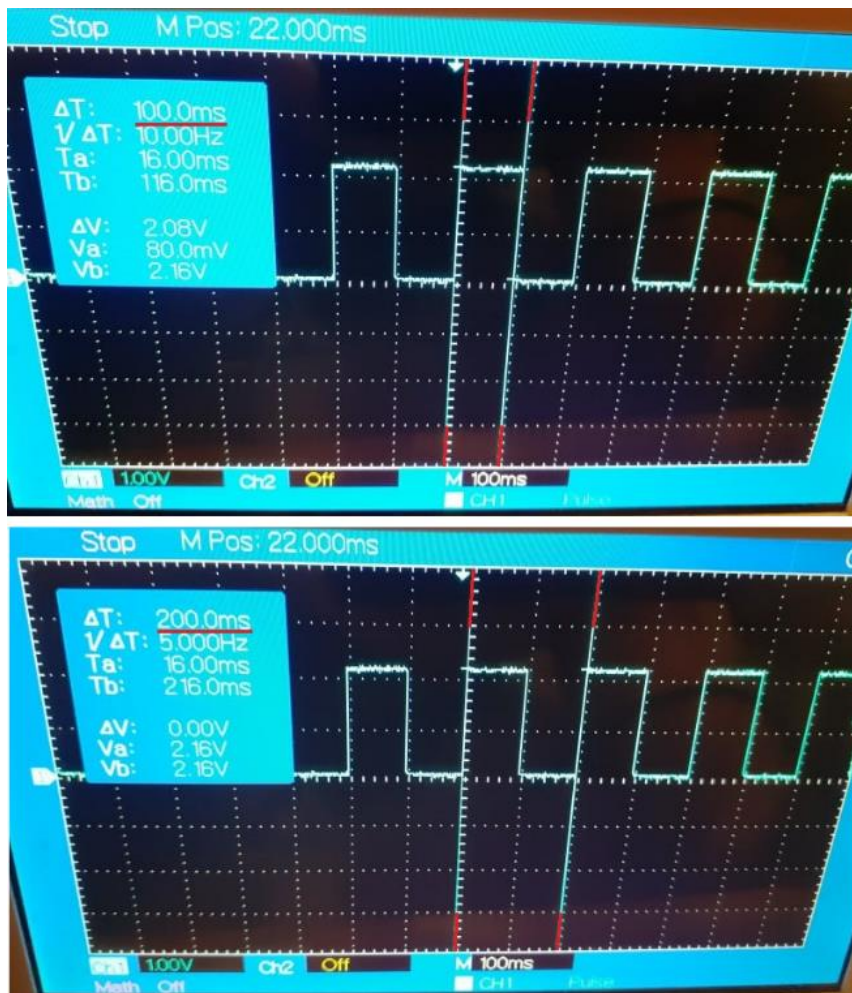


Figura 34. Semnalul digital generat pe baza mesajelor specifice livrate de Aplicația Gateway la o recurență de 100 ms

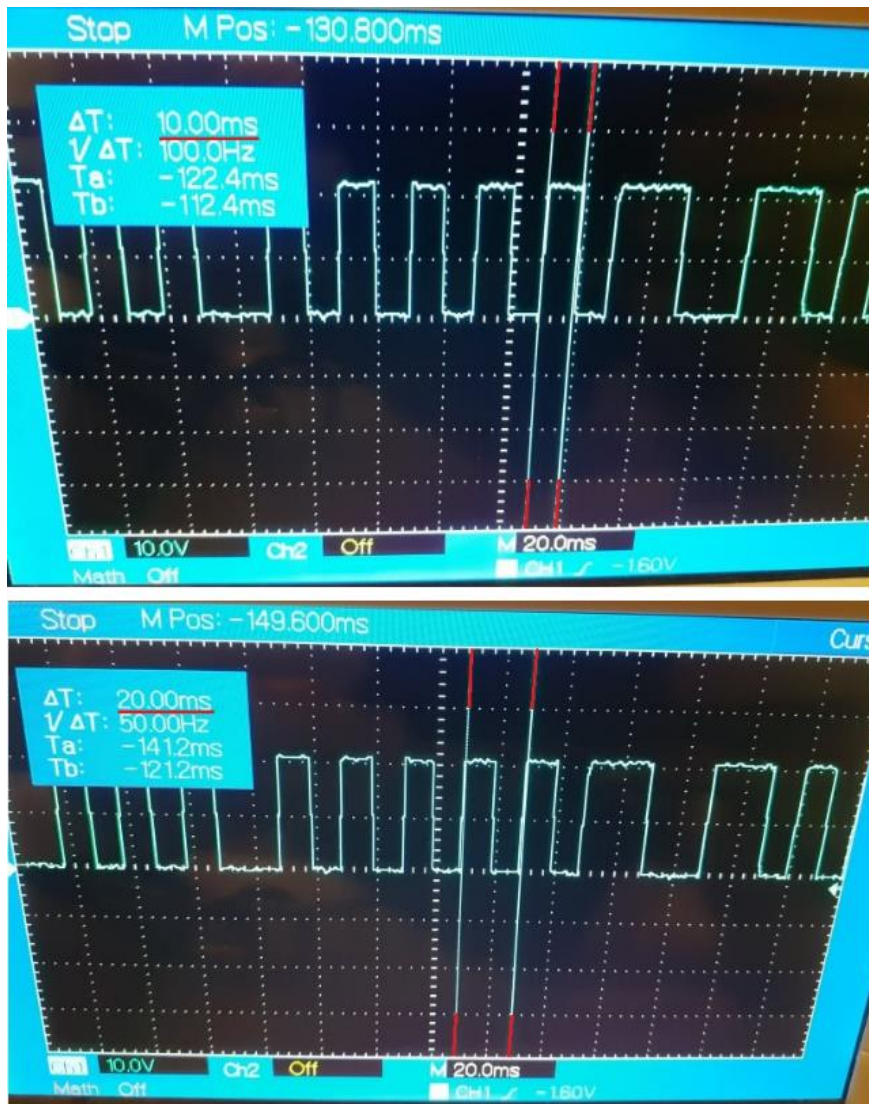


Figura 35. Semnalul digital generat pe baza mesajelor specifice livrate de Aplicația Gateway la o recurență de 10 ms

112 | 5. Coexistența protocoalelor DDS și OPC UA în contextul Industry 4.0 și a comunicației în timp real



Figura 36. Semnalul digital generat pe baza mesajelor specifice livrate de Aplicația Gateway la o recurență de 5 ms



Figura 37. Semnalul digital generat pe baza mesajelor specifice livrate de Aplicația Gateway la o recurență de 2 ms



Figura 38. Semnalul digital generat pe baza mesajelor specifice livrate de Aplicația Gateway la o recurență de 1 ms

114 | 5. Coexistența protocoalelor DDS și OPC UA în contextul Industry 4.0 și a comunicației în timp real

În Figura 34 (100 ms), semnalul este generat sub forma și frecvența așteptate, ceea ce înseamnă că datele au sosit în mod corespunzător. Rețeaua este suficient de stabilă pentru a evita desincronizările între transmisiuni, iar datele sunt livrate cu o eficiență ridicată. Potențialii factori nefavorabili nu au influențat procesul.

În Figura 35 (10 ms), durata semnalului este corectă, iar ciclurile tind să aibă regularitatea dorită. Cu toate acestea, pot fi detectate ușoare desincronizări în momente aleatorii, ceea ce înseamnă că unele mesaje nu sunt recepționate la timp de către nodul de control, cauza principală fiind incapacitatea rețelei de a garanta livrări în timp real în acest caz.

În Figurile 36-38, pentru toate intervalele sub 10 milisecunde, rezultatele observabile dovedesc capacitatea protocoalelor și a aplicației gateway de a furniza date la intervalele dorite. Dezavantajele unor astfel de schimburi rapide sunt reprezentate de factori suplimentari, pe lângă stabilitatea rețelei, factori precum răspunsul întârziat al sistemelor de operare ale mai multor noduri, deoarece semnalul generat este rezultatul schimburilor dintre mai multe noduri și fiecare întârziere din partea oricărui nod implicat poate perturba livrarea mesajelor de interes. Cu fiecare întârziere suplimentară, crește riscul unor livrări inexacte, pulsurile semnalului devenind astfel neregulate din cauza vitezei mari de transmisie. Fiabilitatea protocoalelor de comunicare în sistemul actual scade în concordanță cu rezultatele studiului de caz 1.

5.5. Concluzii

Cu arhitectura și conceptul propus, respectiv cu studiile de caz, implementate în mod performant, și pe baza rezultatelor obținute, au fost atinse următoarele obiective ale studiului actual:

- Cele două criterii definite pentru examinarea comportamentului DDS și OPC UA au furnizat o perspectivă complexă față de capacitățile protocoalelor selectate, într-un sistem care ia în considerare provocările actuale specifice comunicării între mai multe dispozitive prin intermediul Ethernet. Potențialul criteriilor actuale se poate extinde la dezvoltări viitoare care abordează etape specifice de îmbunătățire sau poate fi adaptat la mai multe sisteme și tehnologii specifice cu obiective similare;
- Mecanismul implementat și folosit pentru a analiza comportamentul în timp real al DDS și OPC UA a confirmat un nivel ridicat de eficiență, iar rezultatele cuantificabile obținute extind percepția actuală cu privire la tehnologiile vizate către noi domenii, industriale cât și de cercetare;
- Arhitectura definită s-a dovedit a fi fiabilă atât pentru utilizarea comună, cât și paralelă a protocoalelor, oferind nivelul dorit de flexibilitate și scalabilitate. Diversitatea factorilor industriali care defavorizează răspunsurile ideale din partea DDS și OPC UA adaugă autenticitate experimentului și permite adoptarea unor considerente arhitecturale similare pentru o gamă largă de aplicații;
- Dezvoltarea aplicației de tip gateway OPC UA - DDS extinde aplicabilitatea protocoalelor către scenarii între domenii industriale diferite, reconfirmă fezabilitatea și calitatea ridicată a serviciilor pentru ambele tehnologii și, în

contextul actual, oferă un punct de vedere practic în ceea ce privește conformitatea cu cerințele de funcționare în timp real.

Abordarea generală Industry 4.0 pune accentul pe volumele mari de date transmise în timp real între diferite tipuri de dispozitive, vizând interacțiunea dintre sisteme mari cu capacități și așteptări diferite. Tehnologiile se extind rapid, iar complexitatea cerințelor crește în mod proporțional. Studiile și dezvoltările actuale trebuie să stabilească direcții clare în legătură cu practicile de implementare și de evaluare, luând în considerare obstacolele actuale și anticipând viitoarele provocări inevitabile. Aplicabilitatea protocoalelor de comunicare trebuie să se extindă în mod continuu, luând în considerare avantajele și dezavantajele oferite de fiecare dintre acestea, iar sectorul de cercetare este esențial pentru a găsi modalități de creștere a eficienței și pentru a defini pași expliciți pentru depășirea factorilor nefavorabili care sunt prezenți în cazurile de utilizare IIoT.

Capacitățile protocoalelor de comunicare pot fi îmbunătățite semnificativ în sistemele care asigură o mai mare capabilitate de răspuns în timp real. Cu toate acestea, pentru astfel de situații, adoptarea de tehnologii software și hardware suplimentare și utilizarea de sisteme de operare specializate este obligatorie, ceea ce sporește complexitatea stabilirii unui set de circumstanțe infrastructurale (echipamente hardware, implementări de componente software multiple, metode complexe de integrare a software-ului, nevoi ridicate de configurare, posibilități de testare automată) pentru maximizarea eficienței tehnologice și a rentabilității.

Relevanța strategiei și a concluziilor implică evidențierea viitoarelor posibilități de îmbunătățiri și noi perspective în ceea ce privește conceptele prezentate, precizând provocările depășite pe parcursul elaborării actualei etape de cercetare. Acest studiu oferă o perspectivă detaliată asupra preocupărilor actuale din domeniile industriale și academice, în încercarea de a progresa către principiile Industry 4.0. Se pot lua în considerare etape viitoare pentru maximizarea eficienței DDS și OPC UA în contextul IIoT și pentru definirea unor metode suplimentare de analiză și dezvoltare a aplicațiilor complexe cu cerințe diversificate. Un prim pas în acest proces ar putea fi adoptarea de sisteme de operare Posix, adaptate la situațiile critice de funcționare în timp real, pentru toate dispozitivele implicate, alături de integrarea infrastructurii de analiză deja implementată. Un al doilea pas ar putea implica adoptarea de software și mecanisme hardware capabile să depășească problemele de instabilitate a rețelei, specifice abordării TSN. După includerea modulară a fiecărei îmbunătățiri suplimentare, pentru fiecare etapă a tranziției către un sistem ideal, rezultatele obținute trebuie să fie comparate și sintetizate, pentru o perspectivă mai bine conturată asupra modului în care fiecare factor favorabil, influențează arhitectura și performanța tuturor componentelor tehnologice. Având în vedere că potențialul protocoalelor de comunicare a fost demonstrat și este încă în expansiune atât la nivel industrial cât și la nivel de cercetare, experimentele viitoare pot contribui considerabil la evoluția către Industry 4.0.

Din punct de vedere al implementării, principalele provocări ale etapei de cercetare curente sunt legate de proiectarea multithreading pentru fiecare aplicație, prezentă pe mai multe dispozitive. Având pe fiecare dispozitiv mai multe thread-uri care gestionează proceduri separate, configurațiile specifice protocolului și sincronizarea thread-urilor au trebuit să fie realizate cu atenție, luând în considerare scenariile de funcționare în timp real, care se doreau a fi evaluate. Elaborarea mecanismelor corespunzătoare criteriilor aplicate în studiul de caz 1 a reprezentat o

116 | 5. Coexistența protocoalelor DDS și OPC UA în contextul Industry 4.0 și a comunicației în timp real

etapă dificilă, vizând un grad ridicat de aplicabilitate și reutilizare. În privința aplicației gateway, principalul impediment a fost integrarea diferitelor componente software specifice celor două protocoale, o situație comună, prezentă și în contextul industrial în cazul tehnologiilor de ultimă generație. Din perspectiva evaluării, configurarea și instanțierea tuturor entităților pentru diferitele etape ale procesului, a necesitat un efort considerabil, la care s-a adăugat numărul mare de teste necesare pentru generarea rezultatelor. Din punct de vedere al conceptualizării, identificarea criteriilor relevante, decisive, ce pot fi aplicate și la scară largă, a reprezentat o provocare majoră. De asemenea, stabilirea unei arhitecturi adecvate pentru toate studiile de caz, capabilă să abordeze circumstanțe diverse, a reprezentat o etapă dificilă a studiului curent.

Aplicabilitatea studiului actual vizează sectorul industrial, oferind mecanisme și rezultate concrete cu privire la coexistența protocoalelor OPC UA și DDS, analizându-se funcționarea în timp real și aplicabilitatea rapidă, într-o infrastructură nefavorabilă. În mod adițional, studiul descrie o strategie metodologică de analiză a coexistenței protocoalelor de comunicare, în contextul funcționării în timp real, strategie ce poate fi utilizată ulterior de comunitatea științifică.

6. TEHNOLOGII IOT ȘI AUTOMOTIVE DE COMUNICAȚIE, APLICATE ÎNTR-UN CONTEXT V2X PRIN INTERMEDIUL UNUI GATEWAY MULTI-PROTOCOL

Abordarea arhitecturală pentru sistemele complexe de comunicații trebuie să se adapteze rapid și să ia în considerare setul tot mai mare de cerințe pentru fiecare domeniu industrial. Domeniul automotive evoluează către era electrificării, cu transformări tehnologice masive ce se realizează la toate nivelurile arhitecturale, hardware și software. Tradiționala utilizarea exclusiv a microcontrolerelor este transformată prin adoptarea de microprocesoare cu funcționalități extinse, iar astfel se remodelează structura de dezvoltare. Deși sunt disponibile noi capacități hardware, iar protocoalele de comunicare pe Ethernet pot contribui la o nouă gamă de cazuri de utilizare pentru comunicațiile din interiorul autovehiculului sau pentru comunicațiile între vehicule (V2X), implicațiile utilizării mai multor protocoale care acoperă diferite tipuri de cerințe, în aceeași arhitectură, nu sunt pe deplin identificate. Importanța stabilirii unor așteptări clare pentru sistemele de comunicații inteligente, luând în considerare diverși factori tehnologici și arhitecturali, este decisivă pentru îmbunătățirile viitoare. Actuala etapă de cercetare se axează pe examinarea compatibilității și a capacităților de reacție în timp real, într-o arhitectură diversificată orientată spre servicii, pentru principalele tehnologii de comunicare automotive IoT bazate pe Ethernet. Analiza de fezabilitate se materializează într-o soluție de tip gateway multi-protocol ce facilitează schimbul de date între entități cu origini tehnologice diferite. Fiind cel mai important middleware pentru scenariile de comunicare intra-vehicul, SOME/IP (Scalable Service-Oriented Middleware over IP) este protocolul relevant în domeniul automotive. Odată cu adoptarea recentă în numeroase aplicații industriale și având conformitate cu standardele din domeniul automotive, DDS (Data Distribution Service) este utilizat ca tehnologie majoră, ce combină deja aplicabilitatea din domeniul automotive cu cea din sectorul IoT. eCAL (enhanced Communication Abstraction Layer) se adaugă la acest ansamblu ca o soluție alternativă pentru viitoarele scenarii de comunicare. Luând în considerare tendințele industriale, abordarea inovatoare de a combina diferite tehnologii de comunicare, pentru a satisface cerințe și cazuri de utilizare multiple, va extinde nivelul de aplicabilitate către sisteme mai complexe și mai performante.

6.1. Noțiuni Introductive

Extinderea către conceptele Industrial Internet of Things și Industry 4.0, conturează o nouă viziune în legătură cu capacitățile de comunicare ale sistemelor industriale actuale [29]. Obiectivele privind fezabilitatea și flexibilitatea pentru fiecare perfecționare tehnologică, se mențin și chiar sunt completate cu cerințe și exigențe suplimentare. Tranziția către următoarea fază este lentă, dar constantă. În ultimii ani, progresul tehnologiilor de comunicare bazate pe Ethernet a produs un impact semnificativ asupra metodelor de dezvoltare deja cunoscute, oferind noi posibilități și

118 | 6. Tehnologii IoT și Automotive de comunicație, aplicate într-un context V2X prin intermediul unui Gateway Multi-Protocol

identificând noi nevoi și provocări pe parcurs. Transmisia unui volum imens de date, în arhitecturile distribuite ce conțin diverse tipuri de dispozitive și entități, la viteze mari, într-un mod determinist și automat, poate fi considerată ca fiind principalul pas necesar în evoluția către sisteme industriale complexe interconectate. Deși la prima vedere ar putea părea diferit, conceptul de interconectivitate este relevant și aplicabil în toate domeniile industriale. Pe măsură ce capacitatea de interfațare a unor sisteme tot mai complexe va crește, indiferent dacă acest concept se aplică în domeniul producției, în domeniul rețelelor electrice, al stațiilor de apă industriale, al infrastructurilor cloud sau în domeniul automotive, performanțele vor crește și vor contribui la îmbunătățirea productivității, calității și eficienței [88,89]. Impactul unei astfel de tranziții se va resimți și prin apariția unor noi cazuri și strategii de afaceri, precum și prin reducerea costurilor administrative pentru mai multe procese industriale.

Protocoloalele de comunicare bazate pe Ethernet sunt tot mai prezente în sistemele industriale și sunt conforme cu normele de interoperabilitate și flexibilitate, oferind soluții multiple pentru scenariile IoT. Cu toate acestea, în cazul comportamentului în timp real pentru dispozitivele de teren implicate în procesele industriale, așteptările nu sunt clar definite, în principal din cauza multitudinii de factori care pot influența performanța și fezabilitatea structurilor de comunicare bazate pe Ethernet. Din punct de vedere academic, este nevoie de studii practice referitoare la arhitecturi și prototipuri, care să abordeze preocupările actuale, studii ce pot oferi un set concret de așteptări și un punct de plecare fezabil pentru viitoarele aplicații industriale.

De asemenea, domeniul automotive a început deja o fază de transformare, de la vehiculele pe bază de combustie către era electrificării. O astfel de tranziție semnificativă va implica schimbări masive la nivel arhitectural, hardware și software și poate, de asemenea, să stabilească fundamentul tehnologic de bază pentru integrarea completă a domeniului automotive în IoT/IIoT. Cu viziuni în expansiune continuă, precum conducerea autonomă [90], vehiculele definite prin software, alături de perfecționarea și îmbunătățirea standardului AUTOSAR, în anii următori așteptările sunt mari pentru principalii jucători de pe piață, iar capacitățile de comunicare bazate pe Ethernet vor avea un rol considerabil în atingerea obiectivelor.

Deși tranziția industrială a început și sunt luate în considerare noi abordări, trebuie să rămână clar că multe tehnologii deja consacrate, specifice domeniului automotive, vor persista și vor contribui la progresele viitoare. Transformarea în sine, ar trebui interpretată ca o strategie îmbogățită, care duce mai departe mai multe protocoale de comunicare, precum CAN, CAN-FD, LIN, FlexRay și altele, acestea având în continuare o largă răspândire și scopuri multiple. Rămâne de văzut dacă aria de aplicabilitate a acestor tehnologii va rămâne aceeași sau dacă se vor produce schimbări pe parcurs. În studiul [91], autorii menționează faptul că "FlexRay, Control Area Network (CAN), Local Interconnect Network (LIN), CAN-FD, Automotive Ethernet sunt principalele protocoale de comunicare automotive utilizate în arhitecturile actuale ale vehiculelor electrice și electronice (arhitectura E/E)". Din perspectiva AUTOSAR, percepția ar trebui să fie similară, chiar dacă platforma Adaptive AUTOSAR este în continuă creștere și are un impact mai accentuat decât în trecut. Importanța adoptării unui standard industrial este legată de ideea că toate conceptele disponibile și validate vor rămâne în uz, iar evoluția standardului va dicta ritmul progresului. Pe termen scurt, alte industrii pot beneficia mai repede de noile capacități tehnologice și acesta este principalul motiv pentru care alte industrii au o direcție mai bine definită către

IIoT și Industry 4.0 (de exemplu, Open Platform Communication Unified Architecture - OPC UA este un factor cheie [52] pentru Industry 4.0 și chiar și Data Distribution Service - DDS în conformitate cu AUTOSAR câștigă aplicabilitate în industrie [92]). Cu toate acestea, pe termen lung, stabilitatea și sustenabilitatea generate de standardul AUTOSAR, oferă un potențial uriaș pentru sectorul automotive în viitor și ar putea face diferența în privința progreselor ulterioare.

Starea actuală a domeniului automotive și soluțiile de comunicare disponibile și conforme, facilitează o nouă direcție de cercetare. Această direcție are în vedere adoptarea mai multor protocoale de comunicare bazate pe Ethernet, pentru a obține sisteme îmbunătățite care să poată extinde aplicabilitatea la scenariile de comunicație de tipul V2X și, de asemenea, să crească eficiența procedurilor de comunicare în interiorul autovehiculului. Din punct de vedere academic, o astfel de abordare are potențialul de a extinde cunoștințele despre comunicațiile Ethernet, menținând obiectivele de înaltă performanță și funcționare în timp real, importante pentru domeniul automotive, contribuind de asemenea la alinierea cu noile arhitecturi preconizate, specifice erei de electrificare.

Prezența din ce în ce mai influentă a microprocesoarelor în arhitecturile vehiculelor, alături de platforma Adaptive AUTOSAR și sistemele de operare POSIX, oferă oportunitatea de a defini noi concepte și, din perspectiva comunicațiilor, de a analiza noile tehnologii într-un context industrial. Astfel, se vor evidenția în mod concret avantajele și dezavantajele. Etapa actuală de cercetare oferă perspective cu privire la protocoalele de comunicare de ultimă generație conforme cu cerințele din domeniul automotive și, de asemenea, urmărește să extindă percepția către IoT/IIoT, fiind dezvoltată o soluție de tip gateway bazată pe tehnologiile relevante de comunicare pe Ethernet conforme cu standardul AUTOSAR: Scalable service-Oriented MiddlewarE over IP (SOME/IP) și Data Distribution Service (DDS), în același context cu o altă soluție middleware emergentă, Enhanced Communication Abstraction Layer (eCAL).

În contextul menționat mai sus, studiul actual are următoarele obiective:

- Analiza compatibilității tehnologiilor de comunicare selectate, combinând strategii și viziuni cu potențial ridicat și evidențiind avantajele și dezavantajele din perspective multiple, oferind o mai bună înțelegere a modului în care fiecare protocol poate fi utilizat în contextul potrivit;
- Definirea unei arhitecturi adecvate, aliniată la cele mai recente principii arhitecturale Adaptive, care să includă elementele necesare pentru o aplicabilitate sporită în domeniul automotive și IIoT, și care să abordeze provocările specifice tehnologiilor Ethernet;
- Conceperea și dezvoltarea unei soluții configurabile de tip gateway multi-protocol, care poate interfața entități SOME/IP, DDS, eCAL complet separate, extinzând aplicabilitatea către concepte V2X și interacțiuni între structuri de comunicare specifice diferitelor domenii industriale;
- Explorarea capacităților de funcționare în timp real, ale soluției de tip gateway multi-protocol, examinând impactul posibil al viitoarelor îmbunătățiri realizabile și stabilind un set clar de așteptări cu privire la cerințele industriale în evoluție.

6.2. Studiu de aplicabilitate în domeniul automotive

În ultimii ani, comunicarea bazată pe Ethernet în interiorul vehiculelor a devenit mai prezentă, iar tranziția actuală către era electrificării va spori și mai mult potențialul. Este de așteptat ca viitoarele arhitecturi să conțină noduri de comunicare bazate pe diferite protocoale și soluții middleware, cu cerințe și capacități variate. Având în vedere importanța standardului AUTOSAR, se poate presupune că unele dintre principalele provocări se vor referi la integrarea tehnologiilor și conceptelor conforme și neconforme cu standardul, într-o manieră modulară, menținând principiile și așteptările SOA (Service Oriented Architecture). Prezența diferitelor tipuri de dispozitive, tehnologii și conformități va încetini semnificativ procesul de dezvoltare, așadar orice concept de cercetare fezabil, trebuie să ofere o imagine de ansamblu clară asupra aplicabilității în industrie. Aplicația gateway dezvoltată, utilizează SOME/IP, DDS și eCAL ca tehnologii de comunicare, asigurând schimbul de date între nodurile independente ale sistemului. Având în vedere contextul industrial privind soluțiile middleware vizate, există două perspective diferite în legătură cu modul în care acest concept poate fi aplicat cu succes. Prima perspectivă arată că diferite noduri SOME/IP, DDS și eCAL pot interacționa prin utilizarea aplicației gateway ca nod central, pentru datele propagate de la microcontrolere, la microprocesoare ce pot supraveghea și gestiona date din surse și arii de funcționare multiple. În acest context, conectivitatea și interoperabilitatea între diferite protocoale, dispozitive și domenii funcționale, sunt asigurate, oferindu-se o soluție clară pentru o comunicare eficientă în interiorul vehiculului. Această primă perspectivă de aplicabilitate poate fi observată în Figura 39.

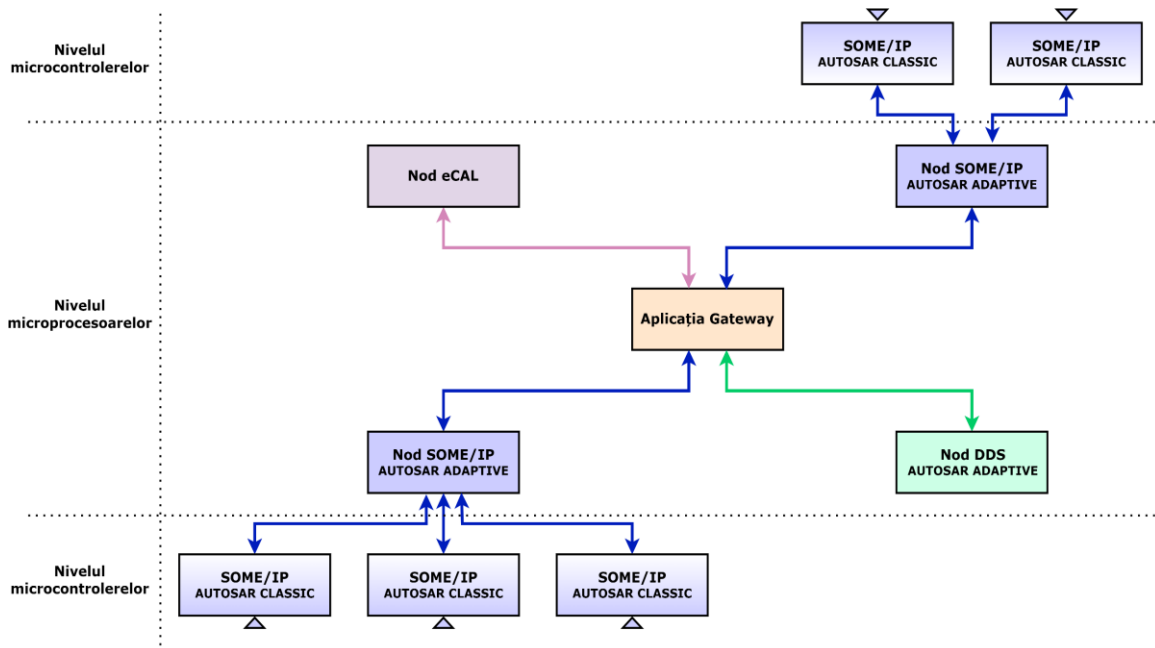


Figura 39. Prima perspectivă de aplicabilitate industrială pentru actualul concept gateway multi-protocol

A doua perspectivă de aplicabilitate se bazează pe plasarea aplicației de tip gateway direct între entitățile clasice, adaptive și cele care nu sunt conforme cu standardul AUTOSAR. Facilitând schimbul de date între microcontrolerele axate pe domenii funcționale individuale și nodurile DDS și eCAL, strategia aplicației gateway multi-protocol poate depăși dezavantajele tehnologice legate de standard. De exemplu, DDS fiind disponibil pentru platforma Adaptive și eCAL nefiind susținut de standard, ar putea totuși să comunice cu microcontrolerele SOME/IP responsabile de procedurile în timp real. Acest lucru permite nodurilor DDS și eCAL să acceseze informații necesare într-un anumit scenariu IoT sau V2X, sau să supravegheze activitatea pe zone funcționale specifice, fără suport pe platforma clasică. Această a doua perspectivă de aplicabilitate poate fi observată în Figura 40.

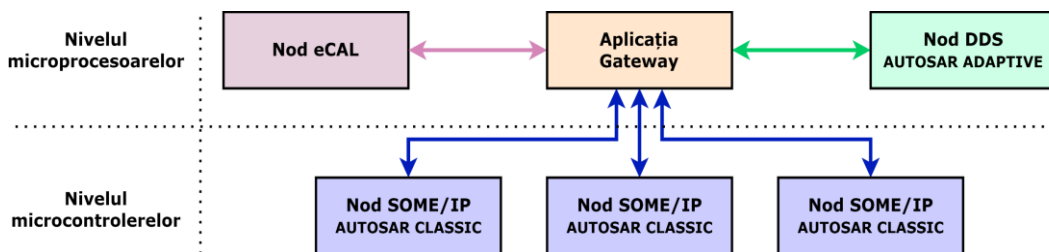


Figura 40. A doua perspectivă de aplicabilitate industrială pentru actualul concept gateway multi-protocol

6.3. Instrumente și Tehnologii

Secțiunea curentă se focalizează pe detalii tehnice privind tehnologiile middleware de comunicare SOME/IP, DDS și eCAL, în raport cu standarde și abordări specifice, într-un context relevant, de actualitate.

6.3.1. SOME/IP

În ultimii ani, complexitatea și diversitatea cerințelor (de integrare a unor noi echipamente, servicii și noi funcționalități) au crescut semnificativ în domeniul automotive. Această ascensiune se aseamănă, într-o anumită măsură, cu evoluții similare din diverse industrii, cum ar fi producția, robotica industrială, uzinele industriale, telematica și altele. Deși obiectivele care au stat la baza acestui proces au avut o origine diferită, mai orientată către un domeniu particular, rezultatul a pregătit tehnologiile de vârf pentru un set complex de scenarii de utilizare. Necesitatea de a utiliza camere de luat vederi, senzori de mare complexitate, machine learning, mecanisme de securitate avansate, necesitatea de conectare la componente non-AUTOSAR, integrarea echipamentelor de tip end-user și alte proceduri, au influențat extinderea arhitecturilor orientate pe servicii (SOA) în majoritatea ariilor din domeniul automotive. Arhitecturile orientate pe servicii asigură schimbul de seturi mari și complexe de date între furnizori și consumatori, combinând paradigmele client-server și publish-subscribe într-o manieră decuplată și abstractă. Pe lângă flexibilitatea sporită, faptul că informațiile sunt distribuite sub formă de servicii specializate, permite o proiectare arhitecturală îmbunătățită, combinându-se diferite

122 | 6. Tehnologii IoT și Automotive de comunicație, aplicate într-un context V2X prin intermediul unui Gateway Multi-Protocol

tipuri de hardware și tehnologii, în concordanță atât cu partea clasică, cât și cu cea adaptivă a standardului.

SOME/IP este conceput pentru a fi în conformitate cu standardul AUTOSAR și oferă toate mecanismele necesare pentru a se adapta la conceptul SOA. Acesta este integrat atât în partea clasică, cât și în cea adaptivă, asigurând o calitate ridicată a serviciilor (QoS) pentru scenariile de comunicare. Din punct de vedere al securității, în lucrarea [93] autorii concluzionează că lipsește securitatea încorporată în protocol, deoarece acesta a fost conceput pentru a funcționa pe protocoale de transport clasice, și propun un cadru de securizare a middleware-ului SOME/IP. O soluție care menține capacitățile de rețea ale protocolului este, de asemenea, oferită în studiul [94]. Lucrarea [95] prezintă o analiză de securitate, evaluând posibilitățile de atac asupra SOME/IP într-un design de tip Publish/Subscribe propunând extensii de securitate. În contextul extinderii utilizării SOME/IP, securitatea este abordată și în alte lucrări (axate pe securitatea rețelelor bazate pe Ethernet în interiorul vehiculului [96] sau pe provocări generale [97]). Deși există studii emergente pe tema securității pentru SOME/IP, subiectul este de mare interes. În prezent, abordările derivă în principal din perspectiva automotive, pe baza cazurilor de utilizare actuale. Cu toate acestea, în viitor, având în vedere expansiunea rapidă și potențialul protocolului, ar putea fi necesare noi măsuri de securitate pentru scenarii mai diverse.

SOME/IP face parte din API-ul de gestionare a comunicațiilor în cadrul AUTOSAR Adaptive, ce poate fi considerat un strat mai abstract care se ocupă de conceptele de comunicare pentru toate tehnologiile middleware acceptate în standard. În conformitate cu principiile SOA, gestionarea comunicațiilor clasifică serviciile în 3 categorii distincte: metode, evenimente și câmpuri. Metodele pot fi reprezentate ca servicii bazate pe designul clasic client-server, cu proceduri de cerere și răspuns între entitățile implicate. Evenimentele se bazează pe designul publish-subscribe, datele fiind livrate fie ciclic, fie sporadic între furnizorii și consumatorii specializați. Câmpurile pot fi descrise ca o combinație între metode și evenimente, oferind, pe lângă livrările ciclice, opțiunea de a se accesa datele prin intermediul metodelor get și set [98-100]. Cu toate că majoritatea protocoalelor bazate pe Ethernet suportă paradigma client-server și paradigma publish-subscribe orientată spre funcționarea în timp real, în scenariile industriale complexe, se așteaptă ca o combinație a celor două modele să fie de fapt cea mai eficientă soluție. Gestionarea serviciilor într-o manieră abstractă, indiferent de protocolul de comunicare utilizat pentru transmisie, poate fi considerată un mecanism puternic, în special pentru arhitecturi și cerințe diverse. În lucrarea [101], autorii explică tendințele actuale în arhitecturile E/E din domeniul automotive, punând accentul pe prezența serverelor centralizate pentru actuatorii inteligenți și procedurile de control în timp real. Una dintre soluțiile practice evidențiate, pentru simplificarea arhitecturii, este unificarea subdomeniilor corelate din punct de vedere funcțional (de exemplu, Powertrain și Chassis). Odată cu apariția computerelor de înaltă performanță (ca servere), cazurile de utilizare gestionabile sunt numeroase. Deși SOME/IP satisface necesitățile privind consumul de resurse, compatibilitatea și scalabilitatea (a se vedea [39]), cerințele vor evolua proporțional cu capacitățile hardware. Lucrările [102,103] abordează SOME/IP în contextul interacțiunii cu soluții locale de comunicare, existente în interiorul vehiculului. În lucrarea [104], autorii abordează toleranța la erori și redundanța folosind SOME/IP middleware ca necesitate pentru cerințele actuale în materie de siguranță în domeniul automotive, menționând că standardul Adaptive AUTOSAR nu specifică cerințe privind toleranța la erori.

Datorită fiabilității ridicate și aplicabilității în domeniul automotive, SOME/IP a fost considerat un bun candidat și pentru scenariile de comunicație de tip vehicle-to-infrastructure și au fost realizate pe această temă, studii recente cu caracter practic. În studiile [23,45], a fost dezvoltată o aplicație gateway între protocolul SOME/IP și OPC-UA, vizând capacitățile de funcționare în timp real, în scenarii IoT. Rezultatele au demonstrat compatibilitatea protocoalelor în arhitecturi și design-uri multiple. Având în vedere că OPC-UA este una dintre cele mai utilizate tehnologii, trebuie luate în considerare viitoarele posibile interacțiuni cu tehnologiile automotive, într-un scenariu bazat pe standardul AUTOSAR. În lucrarea [84], se definește o arhitectură pentru legătura dintre platforma adaptive și OPC-UA, iar prototipul validează capacitățile OPC-UA ca soluție middleware pentru comunicarea orientată pe servicii în AUTOSAR. Validarea tehnologiilor specifice sectorului automotive, în scenarii industriale și IoT, și a protocoalelor industriale dedicate, în scenarii conforme cu AUTOSAR, sugerează că următorul pas pentru o interconectivitate sporită se va baza pe interacțiuni eficiente între protocoalele de comunicare cele mai capabile și mai utilizate, din diverse sectoare industriale.

În prezent, SOME/IP este utilizat la scară largă de către toți actorii importanți de pe piața auto, fiind validat în multiple proiecte și arhitecturi [105]. Deși satisface așteptările pentru moment, creșterea rapidă a cerințelor, complexitatea viitoarelor topologii de rețea și alte soluții disponibile, ar trebui să extindă viziunea către noduri mixte bazate pe diferite tehnologii de comunicare în cadrul aceleiași arhitecturi. În astfel de cazuri, legătura dintre nodurile cu origini tehnologice diferite este reprezentată de aplicații gateway avansate, ce facilitează interacțiunile.

6.3.2. DDS

Urmând modelul industrial privind tranziția către Industry 4.0, este clar că următorii pași se vor focaliza puternic pe comunicarea în timp real și pe perfecționări arhitecturale pentru capacități îmbunătățite, vizând, de asemenea, reducerea costurilor în fiecare etapă de dezvoltare. Un impact semnificativ, derivat din transformarea rapidă a sistemelor industriale actuale, se resimte în evoluția cerințelor generale. Datorită unui astfel de context dinamic, comunicațiile bazate pe Ethernet au devenit coloana vertebrală a sistemelor inteligente, iar interacțiunile inter-domenii, între protocoalele consacrate, constituie tema principală a studiilor emergente.

Middleware-ul DDS oferă un design publish-subscribe, orientat către aplicații deterministe în timp, și un design clasic de cerere-răspuns pentru apeluri de proceduri la distanță, permițând o distribuție eficientă a datelor între participanți. Pe lângă performanțe ridicate și predictibilitate, DDS oferă un set de mecanisme destinate îndeplinirii diverselor tipuri de obiective, asigurând o utilizare eficientă a resurselor. Pentru a separa complet zonele funcționale în cadrul unei arhitecturi, sistemul poate fi împărțit în mai multe domenii, fiecare dintre acestea ocupându-se doar de date specifice și de consumatorii și furnizorii interesați. Entitățile care se alătură sunt denumite participanți la domeniu, iar schimbul de date se realizează într-un spațiu virtual desemnat în cadrul sistemului. Acest tip de mecanism sporește flexibilitatea pe care o prezintă DDS, permițând o proiectare ușor de gestionat chiar și pentru un sistem de mari dimensiuni. Un alt beneficiu derivat din existența unor spații virtuale separate, pentru diferite funcționalități, este capacitatea de a adăuga sau de a elimina funcționalități într-o arhitectură complexă într-un mod granular. Întrucât unele dintre provocările viitoarelor aplicații inter-domenii sunt reprezentate de numărul mare de

124 | 6. Tehnologii IoT și Automotive de comunicație, aplicate într-un context V2X prin intermediul unui Gateway Multi-Protocol

dependințe specifice domeniului și de complexitatea crescută a software-ului, un avantaj legat de gestionarea modulară, poate fi decisiv atunci când se ia în considerare o tehnologie de comunicare.

Informațiile transmise între entitățile Publisher și Subscriber sunt structurate sub formă de topic-uri, iar fiecare consumator interesat se poate abona la unul sau la mai multe topic-uri. Fiecare topic poate avea mai multe instanțe, iar prin intermediul subcomponentelor de tipul data Readers și data Writers, informațiile sunt actualizate și distribuite în mod eficient. În general, topic-urile se bazează pe tipuri de date complexe, ce pot fi definite cu ajutorul mecanismelor integrate disponibile, permițând o structurare clară a payload-ului înainte de a fi furnizat.

Având în vedere implementările disponibile de la diferiți furnizori, compatibilitatea și portabilitatea DDS sunt asigurate prin utilizarea protocolului Real-Time-Publish-Subscribe (RTPS) ca protocol de legătură. La nivel de transport, se utilizează TCP sau UDP, subcomponentele și caracteristicile DDS fiind mapate pe protocolul RTPS. Avantajul de a accesa toate mecanismele DDS fără a avea de-a face cu constrângeri și dependențe privind stiva tehnologică furnizată, este decisiv pentru sistemele dinamice cu diferite tipuri de cerințe. Din punct de vedere al securității, în cadrul SOA, obiectivele generale sunt clare: autentificare și autorizare pentru fiecare participant, integritate și confidențialitate pentru datele distribuite, mecanisme de suport criptografic, autentificarea originii mesajelor. În cazul DDS, este disponibilă o colecție de facilități de securitate, alături de posibilitatea de a extinde capacitățile folosind interfețe plugin. Cu toate acestea, analize recente ale DDS și ale implementărilor disponibile, au scos la iveală un set de vulnerabilități exploatabile. Industrial Control System Advisory a abordat această problemă, evaluând riscurile și oferind o notificare timpurie de vulnerabilităților, identificând măsuri de atenuare de bază (a se vedea [106]). Problemele de securitate ale DDS sunt abordate și în studiul [107].

Similar cu SOME/IP, DDS este, de asemenea, asociat cu clusterul de gestionare a comunicațiilor pe platforma adaptive, asigurând conformitatea cu standardul și cu domeniul automotive. Deși este compatibil ca middleware de comunicare în cadrul standardului, DDS nu este utilizat pe scară largă de majoritatea companiilor importante de pe piață. Abia în ultimii ani, a început să atragă mai multă atenție, deoarece nevoile industriei s-au extins la cazuri de utilizare ce implică interoperabilitate majoră. Protocolul este abordat pentru vehicule autonome [108,109], vehicule aeriene [110] și alte sisteme distribuite cu multiple procesoare în timp real [111]. Deoarece protocolul este deja susținut de platforma adaptive, pentru o creștere fezabilă a aplicabilității în domeniul automotive, trebuie stabilite obiective concrete pentru susținerea DDS în cadrul platformei clasice. Chiar dacă se observă o prezență mai frecventă a aplicațiilor adaptive în procesul de dezvoltare actual, pentru o integrare realistă în viitoarele arhitecturi orientate pe servicii, tehnologiile de comunicare trebuie să fie susținute atât de platforma clasică, cât și de cea adaptive. Ca parte a API-ului de gestionare a comunicațiilor, abordarea conceptuală a metodelor, evenimentelor și câmpurilor este aplicată și în cazul DDS, oferind aceeași strategie standardizată pentru implementarea aplicațiilor adaptive, indiferent de middleware-ul utilizat.

Din punct de vedere academic, având în vedere că IoT este un subiect de mare interes pentru cercetători și pentru industrie, DDS reprezintă o tehnologie cu un potențial ridicat [112], iar studiile se axează pe identificarea celor mai performante protocoale de comunicare care pot satisface setul tot mai mare de cerințe. Cu DDS ca

middleware de comunicare pentru ROS2, un framework open source destinat controlului avansat al roboților industriali, mai multe studii au evaluat capacitățile și comportamentul în timp real al protocolului [79,80,113,114]. În studiile [86,87,115], DDS este utilizat alături de alte protocoale specifice IoT, dovedindu-se a fi o soluție flexibilă și scalabilă în conformitate cu standardele industriale. În lucrarea [116], autorii definesc o arhitectură adecvată pentru coexistența și comunicarea prin gateway între DDS și OPC-UA, analizând capacitatea de reacție în timp real a protocoalelor pe dispozitive de teren.

În prezent, DDS este utilizat în cercetare și în industrie, fiind validat într-o gamă largă de scenarii. Conformitatea cu platforma adaptive AUTOSAR permite noi strategii viabile pentru viitoarele produse automotiv, punându-se accent pe interconectivitate și interoperabilitate, conform evoluției tehnologice generate de tranziția către era electrificării. În cazul coexistenței cu alte tehnologii de comunicare, fiecare entitate DDS făcând parte dintr-un spațiu virtual separat, accesibil doar participanților săi, dificultățile legate de topologia rețelei pot fi depășite cu mai puțin efort, ceea ce face ca sistemul să fie mai flexibil și mai scalabil. Îndeplinind criteriile industriale pentru numeroase cazuri de utilizare și având în vedere ritmul rapid în ceea ce privește evoluția SOA și a cerințelor, în paralel cu contextul IIoT, DDS poate reprezenta o soluție fezabilă pentru sisteme complexe cu tehnologii și cerințe mixte.

6.3.3. eCAL

Soluțiile de transfer de date cu latență redusă sunt necesare în contextul scenariilor de comunicare de nivel superior. Cantitatea enormă de date, procedurile complexe de configurare și incompatibilitățile hardware și ale sistemelor de operare, constituie impedimente majore în procesul de dezvoltare de astăzi. Pe lângă conceptele fundamentale de comunicare oferite, capacitatea de a urmări și de a vizualiza distribuția mesajelor de prioritate ridicată, prin intermediul unei soluții middleware, poate fi benefică în arhitecturi complexe cu un număr mare de noduri. ECAL este o soluție open source cu origini în sectorul automotive, concepută pentru comunicații de înaltă performanță între diferite noduri, pe baza unei configurații minime. Protocolul implementează paradigmele publish-subscribe și client-server, urmând aceleași principii și mecanisme agreeate, în conformitate cu cerințele industriei. Având în vedere importanța cazurilor de utilizare ce presupun controlul roboților industriali în contextul IoT, există o implementare disponibilă eCAL ROS2, ce reprezintă un nivel suplimentar, cu acces la instrumente de monitorizare. Din punct de vedere al compatibilității, există versiuni eCAL stabile care rulează pe Linux, QNX și Windows, suportând platformele Intel și Arm. Fiind o soluție rapidă, fiabilă și scalabilă, ce asigură nevoile conceptuale ale structurilor de comunicare actuale, eCAL poate fi considerat un protocol viabil într-o arhitectură orientată pe servicii. Datorită compatibilității înalte, utilizării limitate a resurselor și configurării ușoare, protocolul eCAL poate fi perceput ca o tehnologie cu potențial ridicat atât pentru sectorul automotive, cât și pentru IoT, și ar trebui examinat în diferite circumstanțe pentru a determina domeniul de aplicabilitate cu cel mai mare impact.

6.3.4. Contextul de aplicabilitate vizat și lucrări conexe

Contextul vizat pentru utilizarea ideală a gateway-ului multi-protocol, este un scenariu de comunicație V2X. Studiul actual nu se axează doar pe tehnologiile

aferente sectorului automotive, ci și pe un middleware IoT performant. Conceptul de comunicație V2X reprezintă un subiect de mare interes în ultimii ani, studiile propun în mod constant noi strategii, extinzând și perfecționând conceptul într-un ritm rapid. În lucrarea [117], autorii evidențiază carențele în cercetare, în legătură cu conceptul de conducere autonomă, sugerând că viitoarele vehicule inteligente se vor baza în mare măsură pe datele de măsurare de la alte vehicule participante în cadrul unui model de comunicație V2V. În studiul [118] se propune o arhitectură cu trei straturi pentru vehicular - IoT, stratul intermediar fiind rețeaua de dispozitive inteligente interconectate, gateway-uri și servere necesare pentru a procesa și furniza datele. Una dintre problemele nesoluționate în cazul vehicular - IoT, evidențiate de autori, este "eterogenitatea comunicațiilor", aceștia descriind cât de dificilă este interoperarea "diferitelor entități care utilizează diferite protocoale de comunicare și generează diferite tipuri de date". Conceptul curent gateway multi-protocol, abordează schimbul eficient de date între entități bazate pe diferite protocoale de comunicare, iar arhitectura definită se aliniaza la principiile descrise de comunitatea științifică, contribuind la depășirea problemelor actuale recunoscute. În lucrarea [119], se anticipează prezența unei cantități semnificative de date în timp real, achiziționate de la senzori și actuatori pentru vehiculele autonome, date care "trebuie procesate și evaluate pentru a primi decizii în timp util în beneficiul utilizatorilor". Un obiectiv major al studiului actual, vizează capacitatea de reacție în timp real a sistemului, permițând o înțelegere aprofundată asupra capacităților efective. Luând în considerare preocupările comunității academice și dinamica industriei, conceptul implementat combină principii automotive și IoT într-un mod inovator, confirmând aplicabilitatea într-un context de comunicație de tipul V2X.

Ținând cont de toate particularitățile tehnice ale SOME/IP, DDS și eCAL, un scenariu de tip gateway pentru o strategie mixtă automotive - IoT, are potențialul de a evidenția modul practic prin care tehnologiile avansate pot conlucra pentru a defini următorii pași din procesul general de dezvoltare. A fost utilizată implementarea open-source GENIVI (a se vedea [120]) pentru dezvoltarea tuturor entităților SOME/IP din sistemul actual. Pentru implementarea componentelor DDS, a fost utilizat kitul de dezvoltare software (SDK) open source eProsima Fast DDS (a se vedea [75]), alături de middleware-ul eCAL (a se vedea [121]).

6.4. Arhitectura Sistemului

Având în vedere că protocoalele de comunicare de ultimă generație sunt deja stabilite în domeniile aferente, un obstacol important pentru obținerea interoperabilității în viitor, va consta în interfațarea tehnologiilor și arhitecturilor majore. Fiecare dintre acestea conține diferite tipuri de resurse hardware și tratează diferite tipuri de cerințe. Pentru studiul curent, bazat pe un caz de utilizare mixt automotive - IoT, obiectivul inițial a vizat definirea unei arhitecturi adecvate cu noduri mixte SOME/IP, DDS și eCAL ce ar putea comunica în cadrul aceleiași rețele prin intermediul unei aplicații gateway, permițând monitorizarea datelor expediate în timp real.

Din punct de vedere hardware, fiind un sistem cu tehnologii specifice automotive, abordarea arhitecturală a fost orientată spre utilizarea microprocesoarelor cu sisteme de operare native POSIX, pentru majoritatea nodurilor. Pentru o interacțiune simulată cu un nod IoT de supraveghere, s-a avut în vedere

utilizarea unui sistem de operare virtualizat Linux, pe un computer de uz general. Nodurile SOME/IP și DDS pot fi considerate ca fiind infrastructura de comunicare intra-vehicul, în conformitate cu standardul AUTOSAR. Gateway-ul facilitează transferul de date între nodurile SOME/IP și DDS, asigurând în același timp interacțiunea cu nodul de supraveghere eCAL, care este amplasat într-o locație separată. Fiecare entitate rulează pe un dispozitiv separat, iar datele transmise sunt structurate sub forma unui eveniment ciclic de tip heartbeat. Arhitectura hardware poate fi observată în Figura 41.

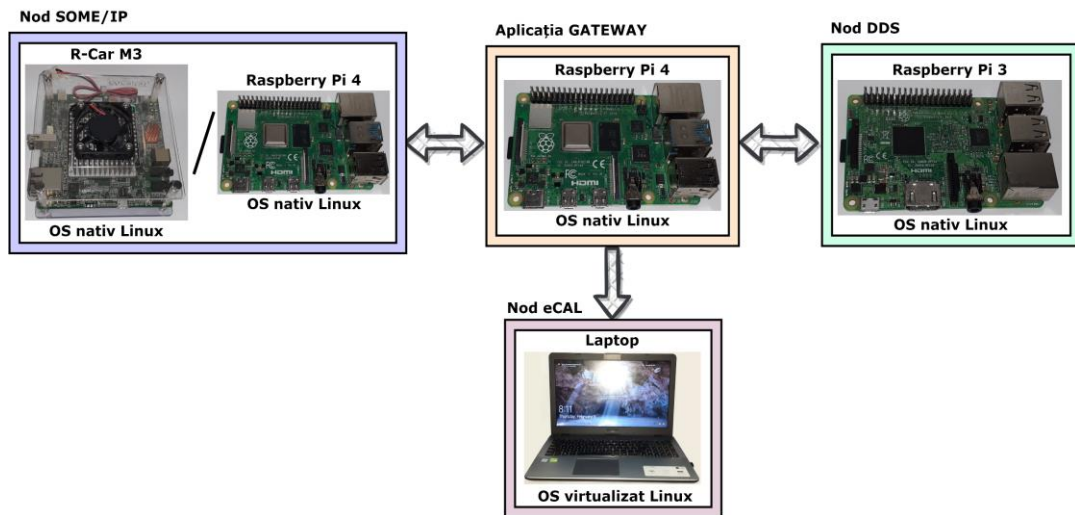


Figura 41. Arhitectura Hardware

Având în vedere că protocolul este utilizat pe scară largă în domeniul automotive, pentru entitățile SOME/IP, alegerea s-a îndreptat către procesoare validate de industrie și, de asemenea, accesibile la un preț mai mic sub forma unor platforme de uz general, pentru dezvoltare. Pe baza acestor considerente, entitățile SOME/IP au utilizat două tipuri de dispozitive:

- Primul dispozitiv utilizat a fost R-Car M3e Automotive System-on-Chip (SoC) de la Renesas, cu două procesoare Arm Cortex-A57, patru procesoare Arm Cortex-A53 și un procesor Arm Cortex-R7 Dual Lockstep. R-CAR M3e este descris ca fiind ideal pentru sistemele de calcul de nivel mediu din domeniul automotive și este utilizat în arii precum infotainment, servere de tip gateway, aplicații integrate în cockpit și altele. Pe SoC-ul R-Car M3e a fost utilizat un sistem de operare Linux cu versiunea de kernel: 5.10, compatibil cu implementarea SOME/IP open-source, utilizată în faza de dezvoltare;
- Cel de-al doilea dispozitiv utilizat pentru entitățile SOME/IP, a fost Raspberry Pi 4 (R-Pi 4) bazat pe un SoC Broadcom BCM2711 cu procesor quad-core ARM Cortex-A72 pe 64 de biți, utilizat la scară largă în diverse tipuri de aplicații, inclusiv în scenarii IoT de comunicare. Pe R-Pi 4 a fost utilizat un sistem de operare Linux cu versiunea de kernel: 5.10.63-V7l+, configurația finală oferind un echilibru cost redus / performanță ridicată.

Middleware-ul DDS, confirmat ca fiind o soluție industrială capabilă, atât în domeniul automotive cât și în sectorul IoT/IIoT, oferind flexibilitate și performanță

128 | 6. Tehnologii IoT și Automotive de comunicație, aplicate într-un context V2X prin intermediul unui Gateway Multi-Protocol

ridicată chiar și cu resurse limitate, a fost potrivit pentru utilizarea pe un dispozitiv dintr-o generație mai veche de R-Pi. R-Pi 3 utilizează un SoC Broadcom BCM2837 cu procesor quad-core ARM Cortex-A53 pe 64 de biți și un sistem de operare Raspbian cu versiunea de kernel: 5.10.17-V7+, fiind potrivit pentru aplicațiile DDS în contextul actual.

Pentru aplicația gateway, care reprezintă partea centrală a arhitecturii, dispozitivul selectat a fost un R-Pi 4. Principala motivație pentru dispozitivul selectat a fost compatibilitatea SOME/IP, DDS și eCAL în cooperare, fiecare dintre acestea necesitând un set special de dependențe. Ubuntu 21.10 cu versiunea de kernel: 5.13.0-1011-raspi a fost utilizat pe dispozitivul gateway R-Pi 4, configurația reprezentând legătura perfectă între infrastructura automotive simulată, ce încorporează SOME/IP și DDS, și nodul eCAL orientat mai mult spre o entitate "cloud" care recepționează date specifice domeniului automotive. Nodul eCAL rulează pe un sistem de operare Linux virtualizat, plasat într-o locație separată, corelându-se cu scenariul V2X simulat. Acesta achiziționează același eveniment de tip heartbeat, ce este distribuit de către gateway către toți consumatorii interesați.

Prezența diferitelor tehnologii de comunicare de nivel înalt în cadrul aceleiași arhitecturi orientate pe servicii, poate spori capacitățile și poate oferi modalități accesibile de adaptare ușoară a arhitecturii, într-o manieră modulară. Deși, în general, abordarea din industrie este aceea de a adopta un protocol principal pentru toate procedurile aferente, în cadrul unei arhitecturi, nu trebuie ignorate beneficiile pe care le aduce existența unor seturi diferite de mecanisme pentru o gamă largă de cazuri de utilizare, în special în contextul IoT. Abordarea arhitecturală actuală de a avea SOME/IP ca soluție standard automotive de comunicare, bazată pe Ethernet, ce interacționează cu middleware-ul DDS, care este conform cu AUTOSAR și este, de asemenea, compatibil cu alte implementări DDS în context industrial, extinde posibilitățile de interoperabilitate, proporțional cu adoptarea de hardware de înaltă performanță și cu adoptarea platformei adaptive. Adăugând la acest ansamblu un protocol mai simplu, dar capabil, dezvoltat pe baza considerentelor din industria automotive, precum eCAL, punerea în aplicare a unor cazuri autentice de utilizare V2X devine fezabilă. Unul dintre principalele obiective stabilite în procesul de definire a arhitecturii, a vizat perspectiva realistă a unei astfel de arhitecturi orientate pe servicii. Prezența diferitelor tipuri de dispozitive hardware și a sistemelor de operare native și simulate, respectă tendințele industriale din sectoarele IoT și automotive, evidențiind eforturile de dezvoltare și realizările necesare pentru viitoarele sisteme de comunicații.

Soluția gateway concepută distribuie date între alte 3 entități, la intervale de timp configurabile. Pe lângă dezvoltarea entităților de tip Publisher și Subscriber specifice protocoalelor SOME/IP, DDS și eCAL, gateway-ul trebuie să angreneze toți participanții și să gestioneze distribuirea eficientă a datelor de interes, luând în considerare posibilele desincronizări care pot apărea din cauza instabilității rețelei sau a reacției lente a sistemului de operare, la cerințele de funcționare în timp real. În funcție de entitatea ce are rolul de furnizor al evenimentului de tip heartbeat, au fost definite două versiuni ale aplicației gateway. Fiecare versiune conține subcomponentele necesare specifice fiecărui middleware, în conformitate cu rolurile atribuite în proiectare, iar fiecare subcomponentă primește și transmite datele pe un thread separat. Având mai multe thread-uri asociate protocoalelor în mod individual, este posibilă o mai bună observare a transmisiei recurente și a posibilelor întârzieri sau disfuncționalități de conectivitate, datele sunt primite și livrate într-un mod

eficient, capacitatea de reacție a aplicației fiind maximizată. Arhitectura sistemului pentru ambele versiuni gateway este prezentată în Figura 42.

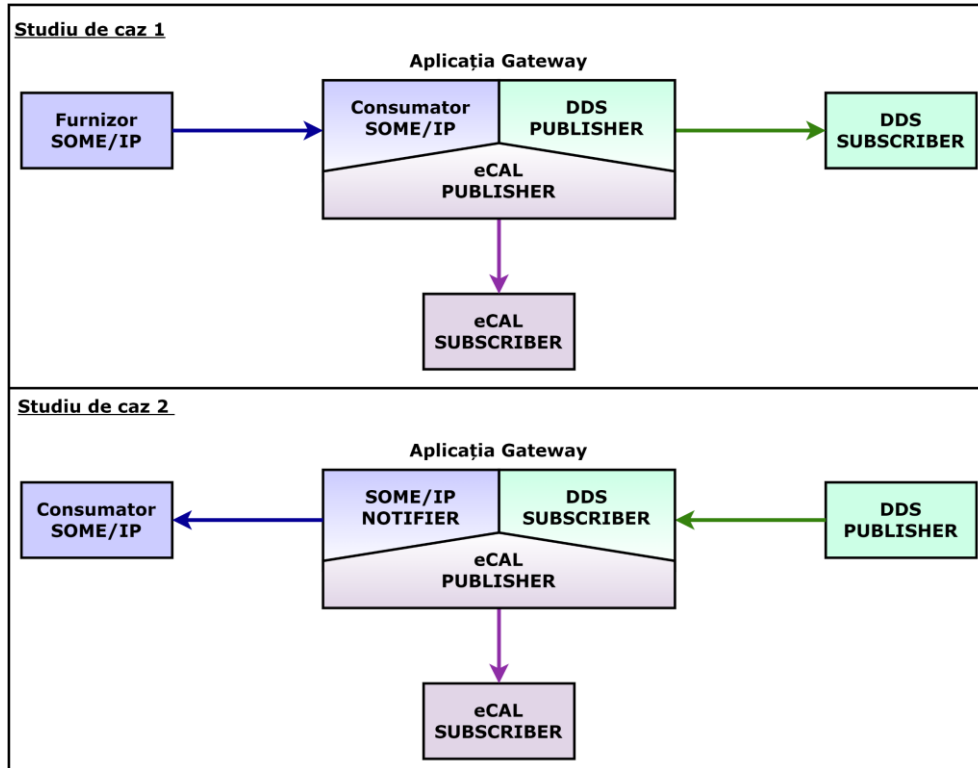


Figura 42. Arhitectura sistemului, fiind reprezentate ambele versiuni ale aplicației gateway

Conceptul și dezvoltarea studiilor de caz vizate, permit o observație clară cu privire la secvența de proceduri în cazul SOA și al nodurilor cu origini tehnologice diferite. Procedurile și interacțiunea dintre acestea, sunt sintetizate în cadrul abordării curente și pot fi observate în Figura 43.

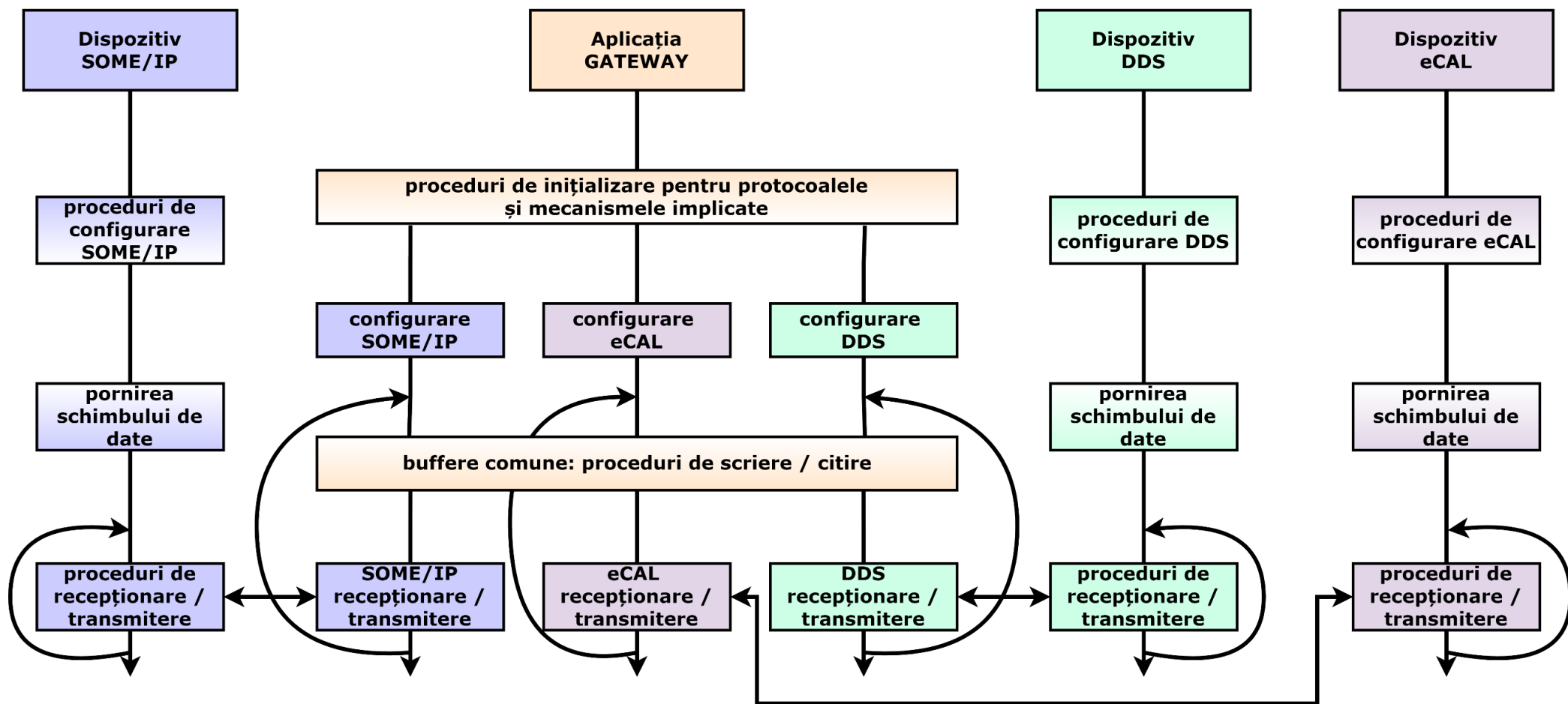


Figura 43. Secvența de proceduri pentru toate nodurile arhitecturii

Obiectivele studiilor de caz 1 și 2 care vizează fiabilitatea și eficiența conceptului sunt analizate prin conceperea și dezvoltarea unui mecanism de stocare a datelor (data buffering) și a unui mecanism de generare a semnalelor aferente evenimentului distribuit, de tip heartbeat. Fiecare mecanism implementat oferă rezultate clare în ceea ce privește comportamentul infrastructurii de comunicații, atunci când se iau în considerare cerințele de funcționare în timp real și impactul pe care rețeaua îl poate avea asupra livrării de mesaje la recurențe rapide. Capabilitatea de data buffering, este considerată o caracteristică importantă în sistemele de comunicații industriale IoT. Pentru scenariile actuale, acest procedeu de stocare a datelor este important pentru o mai bună înțelegere a obstacolelor ce pot apărea în arhitecturile complexe orientate pe servicii. Procesul de funcționare al mecanismului data buffering conceput, este detaliat din punct de vedere arhitectural în Figura 44.

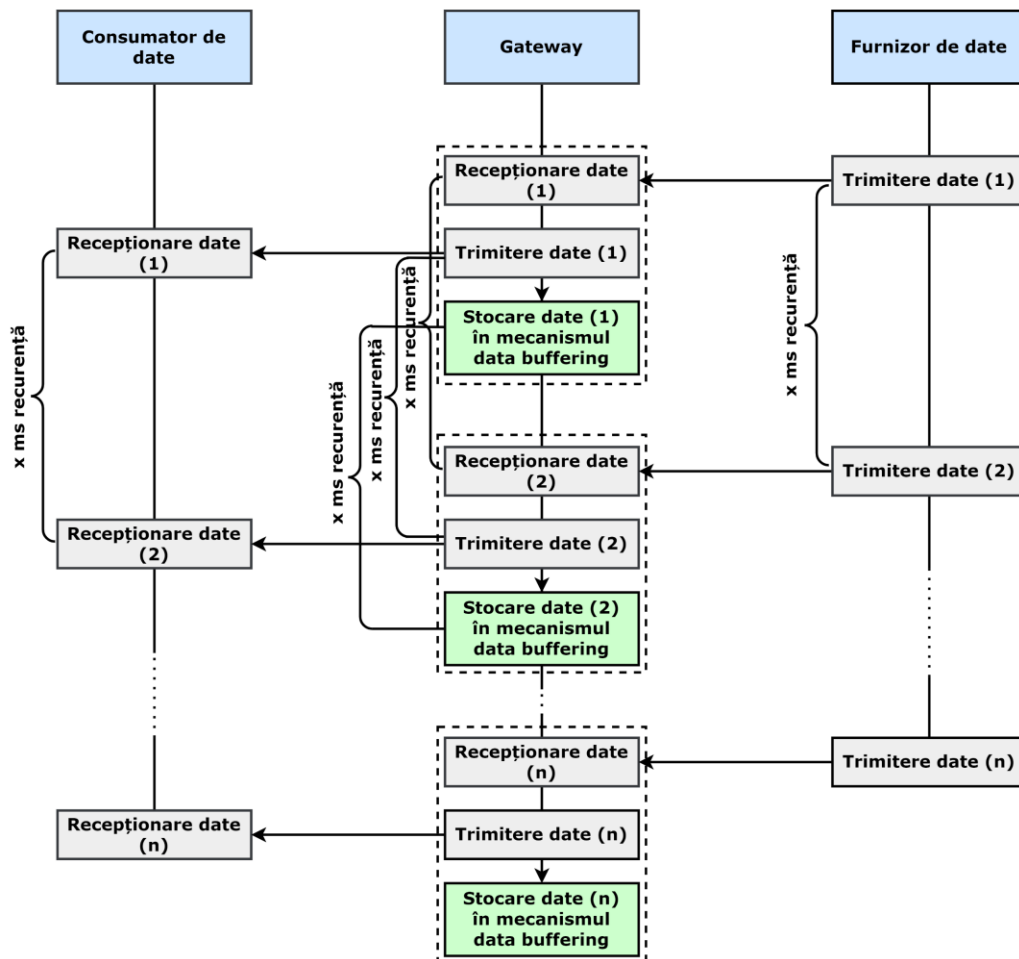


Figura 44. Secvența Data Buffering

Arhitectura definită este fezabilă pentru expedierea datelor într-o manieră deterministă în timp, combinând dispozitive, strategii și tehnologii în conformitate cu scenariile industriale. Interacțiunea reușită a celor trei protocoale utilizate, extinde aplicabilitatea către un sistem mai complex și mai capabil, care se bazează pe principiile IoT, dar păstrează esența tehnologică a domeniului automotive.

6.5. Studii de caz și Rezultate

Secțiunea curentă oferă informații cu privire la comunicarea în timp real, evaluând compatibilitatea protocoalelor în arhitectura definită, prin aplicarea unor metode și criterii specifice scenariilor industriale. Pe baza celor două versiuni ale aplicației gateway, există două studii de caz diferite, fiecare prezentând rezultate și observații detaliate cu privire la capacitățile sistemului. Ambele studii de caz simulează un caz de utilizare specific comunicației V2X, în care SOME/IP și DDS constituie infrastructura de comunicații din interiorul autovehiculului. În timp ce gateway-ul translatează mesajele pentru toți receptorii, se realizează de asemenea o conexiune cu un nod eCAL extern. Configurarea aplicației gateway, de la un studiu de caz la altul, se realizează cu efort minim și nu sunt necesare modificări arhitecturale din punct de vedere hardware, deși fiecărui nod îi sunt atribuite modificări funcționale și roluri diferite.

6.5.1. Dezvoltarea celor două studii de caz

Obiectivele principale ale implementărilor actuale au vizat o eficiență ridicată (distribuirea mesajelor în mod stabil la intervale de timp sub 100 de milisecunde), în timp ce datele sunt livrate ciclic către consumatori, asigurând fiabilitatea tuturor protocoalelor chiar și în cazul recurențelor rapide. Pentru studiul de caz 1, aplicația gateway este configurată pentru a facilita comunicarea între un furnizor SOME/IP și consumatorii DDS și eCAL, fiecare nod fiind specific unui dispozitiv separat. Conform descrierii din secțiunea dedicată arhitecturii, pentru furnizorul SOME/IP au fost utilizate 2 dispozitive diferite, validând soluția pe hardware specific automotive și IoT, fiind obținute comportamente similare. Din punct de vedere software, portabilitatea entității SOME/IP confirmă potențialul tehnologiei, chiar dacă a fost utilizată cu o variantă open source a protocolului și nu s-a aplicat contextul AUTOSAR. Comunicarea SOME/IP între furnizor și consumator se bazează pe paradigma Notify-Subscribe, în conformitate cu necesitățile scenariului actual, în care mesajul este structurat sub forma unui eveniment de tip heartbeat. Destinatarul mesajului este aplicația gateway, respectiv subcomponenta de tip Subscriber SOME/IP, care permite ulterior translatarea mesajului pentru celelalte noduri interesate. Configurațiile de rețea, alături de alte configurații specifice protocolului, se realizează prin utilizarea de fișiere json ce sunt transmise aplicațiilor în timpul execuției, prin utilizarea variabilelor de mediu specifice protocolului. Această abordare asigură o separație între logica aplicației și particularitățile de configurare, ce sunt necesare într-o anumită măsură

pentru orice protocol de comunicare bazat pe Ethernet, menținând o aliniere la abordarea industrială standard, în care fișierele json și arxml sunt utilizate în mod frecvent în scopuri de configurare. Implementarea entităților SOME/IP în versiunea actuală, asigură furnizarea de date într-un mod fiabil, iar interacțiunea cu celelalte protocoale nu scade performanța.

Aplicația gateway reprezintă nodul central al sistemului, comunicând prin trei protocoale diferite cu celelalte noduri. Aceasta este împărțită în trei subcomponente diferite, fiecare dintre ele facilitând livrarea de date către nodul complementar. În acest design, subcomponentele aplicației gateway sunt reprezentate de un Subscriber SOME/IP, un Publisher DDS și un Publisher eCAL. Datele sunt recepționate de subcomponenta Subscriber SOME/IP, după care sunt pasate către subcomponenta Publisher DDS și pregătite pentru a fi expediate către nodul consumator DDS. Aceeași procedură se execută și în cazul subcomponentei eCAL, pentru toate mesajele recepționate. Pentru o eficiență sporită, fiecare subcomponentă rulează pe un thread diferit, permițând gateway-ului să recepționeze și să transmită mesajele cu frecvența dorită, existând posibilitatea de a identifica între ce entități ar putea apărea o întârziere. Din punct de vedere al configurației, subcomponenta DDS este inițializată ca participant la un domeniu și i se atribuie rolul de Publisher. Nodul consumator DDS este, de asemenea, inițializat ca participant la același domeniu, dar i se atribuie rolul de Subscriber. Mesajul livrat este structurat sub forma unui topic și se efectuează operații de corelare între entitățile de tip Reader și de Writer specifice protocolului, pentru a se identifica furnizorul de date și consumatorul. Subcomponenta de tip Publisher DDS și nodul consumator DDS împart același domeniu, fiind o infrastructură de comunicare complet separată, ce nu este disponibilă pentru niciun alt nod. Astfel, comunicarea se menține sigură și fiabilă, fără a fi nevoie să se țină cont de celelalte noduri și de particularitățile lor de rețea. Subcomponenta eCAL a aplicației gateway, reprezintă furnizorul semnalului de tip heartbeat, din afara infrastructurii simulate de comunicații în interiorul autovehiculului, evidențiată prin prezența soluțiilor middleware SOME/IP și DDS. Protocolul eCAL poate fi perceput ca fiind soluția adecvată pentru comunicațiile V2X, datorită efortului minim de configurare necesar, dar și datorită flexibilității deosebite, fiind dezvoltat ca o modalitate simplă de transmitere a mesajelor în rețea. La fel ca în cazul SOME/IP și DDS, protocolul eCAL implementează paradigma Publish-Subscribe, permițând subcomponentei eCAL din cadrul aplicației gateway, să aibă rolul de Publisher, furnizând datele către nodul consumator eCAL. Nodurile consumatoare DDS și eCAL pot fi percepute ca fiind destinațiile finale ale datelor, ambele interacționând independent cu aplicația gateway și recepționând semnalul de tip heartbeat la frecvența configurată. Având în vedere că nodul eCAL este considerat o entitate orientată către cloud, este de așteptat ca dispozitivul cu care structura simulată de comunicare intra-vehiculară interacționează, prin intermediul aplicației gateway, să fie plasat într-o locație diferită față de restul nodurilor. Pentru a exista o corelare cât mai mare cu principiile comunicației V2X, pentru studiul actual, dispozitivul receptor eCAL a fost complet separat de celelalte dispozitive, singura legătură cu acesta fiind subcomponenta specifică a aplicației gateway. Cu toate acestea, distanța nu a întârziat procesul de recepție nici măcar pentru cicluri de transmisie foarte rapide, iar nodul consumator eCAL a reușit să recepționeze semnalul fără probleme de conectivitate.

134 | 6. Tehnologii IoT și Automotive de comunicație, aplicate într-un context V2X prin intermediul unui Gateway Multi-Protocol

Dincolo de implementarea aplicației gateway și a entităților adiționale, evaluarea funcțională a arhitecturii proiectate trebuie să respecte așteptările industriale pentru sisteme automotiv și IoT, utilizând nodul central gateway ca o punte de legătură între diferite tehnologii de comunicare, folosite în domenii multiple. Pe baza acestor așteptări, au fost luate în considerare două criterii și pentru fiecare dintre acestea au fost dezvoltate mecanisme suplimentare, oferind o perspectivă mai largă asupra capacităților de comunicare în viitoarele cazuri complexe de utilizare. Primul criteriu luat în considerare a fost eficiența proceselor de recepție și de transmisie, prin intermediul operațiunilor de tip data buffering pentru nodul gateway. În condițiile în care nu sunt disponibile asigurări în privința timpului de livrare pentru comunicațiile Ethernet și în situația posibilelor desincronizări de rețea între dispozitive, mecanismul de stocare a datelor oferă un rezultat cuantificabil în legătură cu informațiile procesate de un nod individual al sistemului. În comparație cu comportamentul ideal preconizat, rezultatele obținute pot determina compatibilitatea între diferite tehnologii de comunicare în cadrul aceleiași arhitecturi. Acest aspect poate contribui la posibile îmbunătățiri ale componentelor software din cadrul sistemului sau la o posibilă reevaluare a sistemelor de operare și a dispozitivelor utilizate. Cel de-al doilea criteriu a vizat stabilitatea sistemului în cazul procedurilor de recepție și transmitere a evenimentelor la diferite intervale de timp, obiectivul principal fiind estimarea nivelului de fezabilitate având în vedere cerințele de funcționare în timp real. Pentru a observa cu exactitate modul în care întârzierile de rețea și desincronizările dispozitivelor afectează sistemul la nivelul nodului central gateway, evenimentul heartbeat trebuia observat în diferite scenarii de funcționare în timp real. A fost implementat un mecanism suplimentar, sub forma unei subcomponente software a aplicației gateway, care generează un semnal digital precis, bazat pe evenimentul de tip heartbeat recepționat și distribuit. Având posibilitatea de a observa modul în care este alterat evenimentul heartbeat în cazul transmisiunilor rapide, predictibilitatea sistemului crește, oferind o imagine clară asupra modului în care datele distribuite pot fi afectate în cazul transmisiunilor Ethernet în arhitecturile complexe.

Cele două criterii selectate pot fi aplicate la o scară mai largă, dacă este necesar, fiind relevante în contextul SOA atât pentru scenariile din domeniul automotive, cât și pentru cele din domeniul IoT. Identificarea limitelor sistemului pentru procedurile de comunicare, este unul dintre cele mai importante obiective care trebuie atinse pentru tranziția către IoT. Mai ales în cazul dispozitivelor integrate, orientate mai mult către cerințe de funcționare în timp real, provocările sunt diverse, nefiind legate doar de protocoalele de comunicare, ci și de particularitățile arhitecturale, capacitățile hardware și ale sistemului de operare. Pentru implementarea actuală, rezultatele obținute pe baza criteriilor și metodelor menționate au oferit concluzii relevante privind compatibilitatea și fezabilitatea protocoalelor utilizate, validând, de asemenea, arhitectura definită.

Diferențele dintre cele două studii de caz sunt reprezentate de designul arhitectural și de distribuția rolurilor între nodurile sistemului. În cel de-al doilea studiu de caz, furnizorul de informații este nodul Publisher DDS ce transmite datele ciclic către aplicația gateway. Subcomponentele aplicației gateway sunt, de asemenea, modificate în conformitate cu designul curent, fiecare dintre ele

recepționând sau transmițând evenimentul heartbeat către nodurile interesate. Datele furnizate de entitatea publisher DDS sunt recepționate de o subcomponentă subscriber DDS și apoi sunt distribuite către notificatorul SOME/IP și către subcomponenta publisher eCAL. Notificatorul SOME/IP transmite datele către nodul consumator SOME/IP, iar subcomponenta publisher eCAL transmite evenimentul către nodul consumator eCAL. Ambele noduri consumatoare sunt destinațiile finale ale evenimentului generat. Structura simulată de comunicație în interiorul vehicului rămâne aceeași, compusă din entități SOME/IP și DDS, dar și din aplicația gateway. Doar rolurile nodurilor diferă, dispozitivele hardware interacționând în același mod. Principiile comunicației V2X sunt aplicate și în studiul de caz 2, nodul consumator eCAL simulează interacțiunea structurilor de comunicație intra-vehicul cu exteriorul și, similar studiului de caz 1, este plasat într-o locație diferită, schimbând date doar cu subcomponenta complementară a aplicației gateway. Aplicația gateway păstrează abordarea inițială, de a atribui fiecare procedură de comunicare unui thread separat, oferind un design software decuplat și optimizat, potrivit pentru cazurile de utilizare în care mai multe tehnologii interacționează în mod independent.

Un obiectiv suplimentar pentru studiul de caz 2 se axează pe validarea rezultatelor studiului de caz 1 într-un context ușor diferit, demonstrând că obiectivele atinse și concluzia privind compatibilitatea și fezabilitatea protocoalelor implicate rămân aceleași chiar și în cazul modificării nodurilor sistemului. Unul dintre scopurile arhitecturii definite este acela de a permite configurații diferite fără a fi necesară modificarea întregului sistem. Acest lucru contribuie la principiile industriale privind integrarea modulară și modificarea componentelor software într-un mod adecvat, menținând aceleași așteptări în privința eficienței și scalabilității. Mecanismul data buffering, împreună cu subcomponenta suplimentară responsabilă de generarea unui semnal digital bazat pe evenimentul de tip heartbeat, sunt aplicate și pentru studiul de caz 2, așteptându-se un comportament similar cu cel din studiul de caz 1.

Aplicația gateway rămâne nodul central al sistemului, implicând mai multe subcomponente ce realizează un schimb de date cu celelalte noduri, prin trei tehnologii de comunicare diferite, într-un mod eficient și rapid. Aplicarea conceptului în cadrul celor două alternative arhitecturale, confirmă faptul că strategia de combinare a tehnologiilor de comunicații de ultimă generație, poate oferi soluții scalabile, în măsură să îndeplinească diverse cerințe industriale. Rezultatele studiului de caz 2 arată cât de relevante pot fi criteriile și metodele aplicate. Pentru sistemul actual, acestea oferă o percepție clară asupra posibilelor limitări ce pot apărea în scenariile complexe, specifice unor sectoare industriale diferite.

6.5.2. Rezultate

Implementarea aplicației gateway, cu toate subcomponentele necesare, a fost realizată cu succes. Activitatea fiecărui nod a fost testată separat, ținându-se cont de capacitățile fiecărui protocol. Astfel, se asigură o interoperabilitate ridicată chiar și în scenarii mai complexe. Designul multithreading al aplicației gateway, s-a dovedit eficient și fezabil în contextul a trei tehnologii separate, ce folosesc tiparul publish-subscribe. În materie de stabilitate și eficiență, rezultatele ratei de succes a mecanismului data buffering, sunt detaliate în Figura 45.

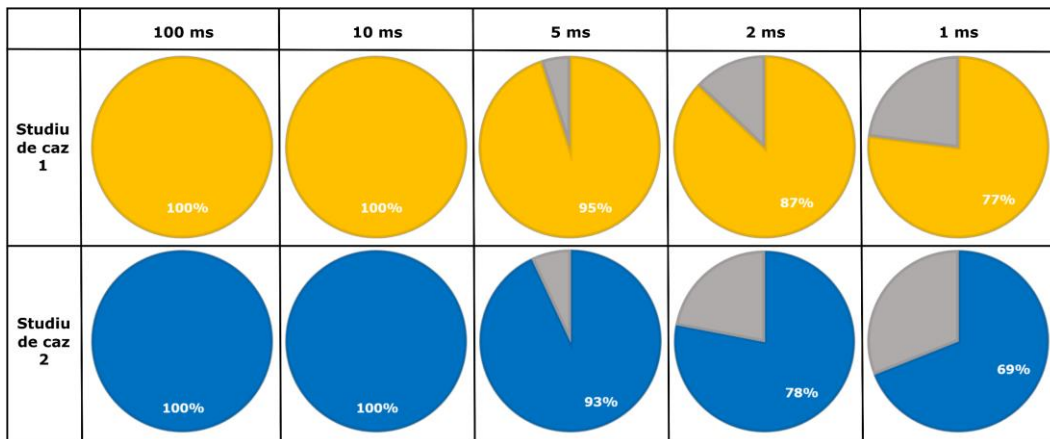


Figura 45. Rezultatele ratei de succes a mecanismului data buffering

Rezultatele obținute prin monitorizarea și achiziția datelor pentru diverse perioade de timp, iar mai apoi comparate cu setul ideal de date așteptat, confirmă compatibilitatea conceptului cu versiuni diferite ale aplicației gateway și cu roluri diferite atribuite nodurilor din cadrul arhitecturii. Subcomponenta data buffering și subcomponenta generatoare de semnal, au oferit o perspectivă clară asupra eficienței și fiabilității sistemului. Rezultatele se focalizează pe comportamentul infrastructurii de comunicare, ținând cont de cerințele de funcționare în timp real și de impactul pe care rețeaua îl poate avea asupra livrării mesajelor la recurențe rapide. Așa cum era de așteptat, eficiența procedurilor de recepție și de transmitere scade în cazul unor recurențe ridicate, o parte din mesaje fiind pierdute din cauza întârzierilor provocate de rețea. Intervalele de timp selectate pentru monitorizare au evidențiat regresul de la o transmisie aproape sigură a evenimentului, la cicluri de transmisie de 1 milisecundă, considerate o performanță rezonabilă pentru sistemul actual. Un alt factor care contribuie la transmisii mai puțin precise, este incapacitatea sistemelor de operare de uz general, de a gestiona operațiuni sub 10 milisecunde și de a asigura capabilitatea de reacție necesară pentru cerințele stricte de funcționare în timp real.

În comparație cu rezultatele studiului de caz 1, eficiența procesului de stocare a datelor a scăzut nesemnificativ în studiul de caz 2, pentru recurențele sub 10 milisecunde, menținând un comportament similar pentru ambele versiuni ale aplicației gateway. Este important să se identifice limitele tehnologice și arhitecturale ale sistemelor inteligente și să se evalueze în mod corespunzător cazurile de utilizare și cerințele realizabile. Capacitățile rețelei, utilizarea sistemelor de operare native pentru dispozitivele cu rol de furnizare sau cu roluri mixte, particularitățile de implementare și designul multithreading pot avea un anumit impact asupra rezultatelor. Totuși, obținerea unei evaluări standard este improbabilă, în cazul arhitecturilor orientate pe servicii, ce combină mai multe tehnologii și echipamente hardware.

Cea de-a doua metodă implementată pentru vizualizarea modului în care evenimentul de tip heartbeat, poate fi alterat în diverse condiții, este reprezentată de mecanismul de generare a semnalului, din cadrul aplicației gateway. Rezultatele înregistrate sunt în concordanță cu rezultatele ratei de succes a mecanismului data

buffering, confirmând fiabilitatea sistemului atunci când există cerințe de funcționare în timp real. Acest tip de monitorizare practică, pentru cazuri complexe de utilizare a comunicației pe Ethernet, poate identifica limite concrete pentru un anumit sistem, oferind posibilitatea de a îmbunătăți aspectele nefavorabile. Comportamentul sistemului a fost monitorizat la diferite recurențe, pe baza semnalului digital, generat conform evenimentului de tip heartbeat. Rezultatele pot fi observate în Figura 46.

Rezultatele demonstrează că sistemul este capabil să furnizeze datele în intervalul de timp dorit, pentru toate recurențele configurate. Cu toate acestea, așa cum era de așteptat, sustenabilitatea scade în mod proporțional cu creșterea ratei de transmisie. Având în vedere complexitatea arhitecturii, comportamentul corespunde așteptărilor realiste. Sunt utilizate dispozitive și sisteme de operare diferite, iar cele 3 protocoale de comunicare au particularități tehnice majore. Pentru intervalele sub 10 milisecunde, semnalul neregulat, generat pe baza evenimentului de tip heartbeat, este ușor de observat, prezentând rar pulsații precise.

Atunci când se discută despre eficiența unui sistem de comunicații bazat pe Ethernet, în lipsa unor standarde și protocoale dedicate care să asigure că dispozitivele sunt sincronizate pe o bază de timp comună și că fiecare dintre ele poate furniza datele în conformitate cu cerințele stricte de funcționare în timp real, nu există garanții că transmisia va fi efectuată în mod constant în intervalul de timp dorit. Rezultatele curente dovedesc eficiența comunicării bazate pe Ethernet, între mai multe dispozitive de teren, evidențiază limitele unui sistem complex și validează compatibilitatea protocoalelor și a arhitecturii, prin aplicarea principiilor industriale din domeniile automotive și IoT. Interacțiunea dintre diverse tehnologii și arhitecturi, deja consacrate în sectoare industriale diferite, va fi inevitabilă, ceea ce înseamnă că multe provocări vor deriva din cauze de compatibilitate și eficiență. Identificarea protocoalelor de comunicare și arhitecturilor adecvate, împreună cu limitările și direcțiile de îmbunătățire în vederea extinderii aplicabilității conceptelor industriale actuale, va produce aplicații fiabile și scalabile pentru viitoarele cazuri de utilizare ce vizează interoperabilitatea industrială.

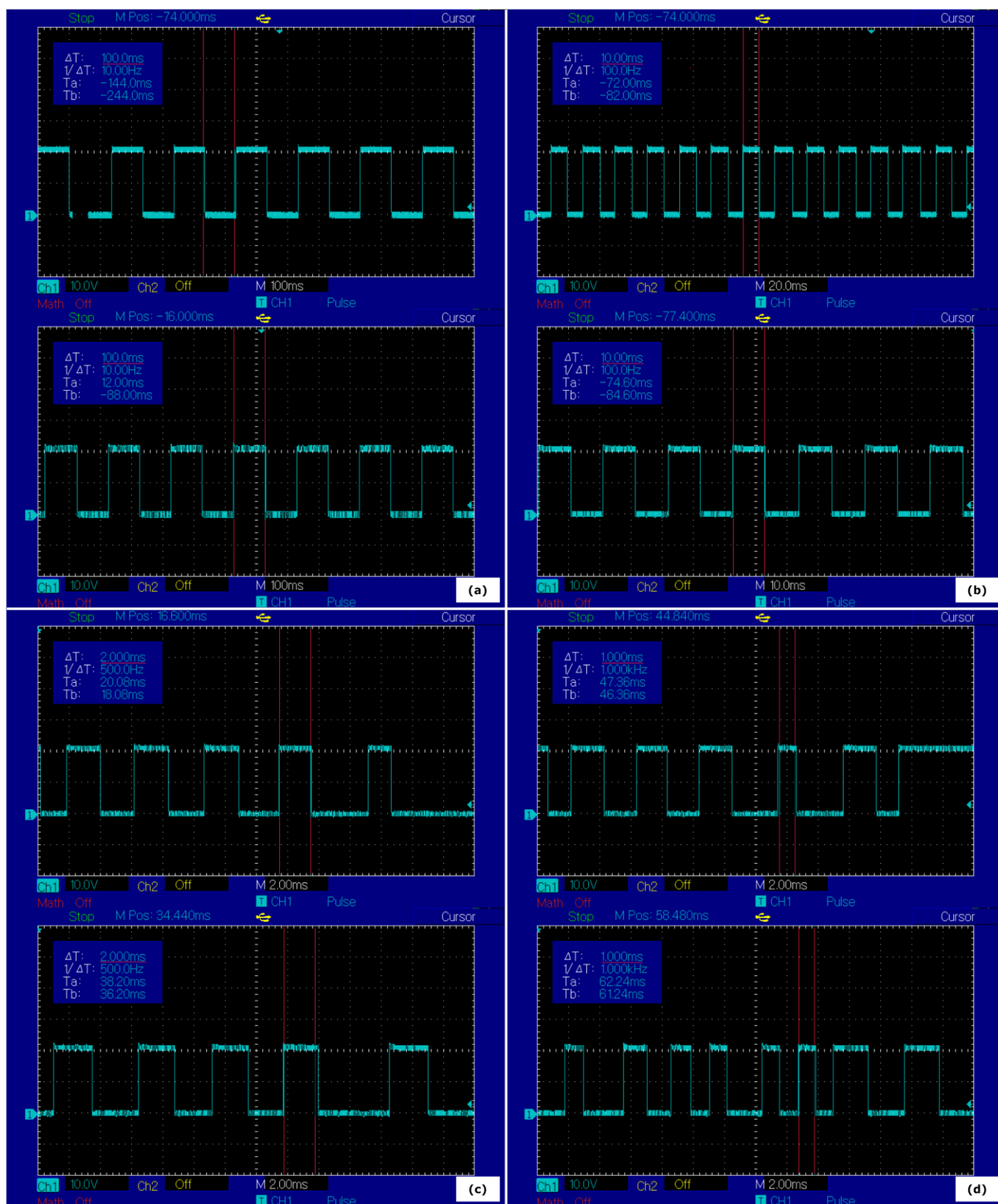


Figura 46. Semnalul digital generat pe baza evenimentului de tip heartbeat recepționat și transmis de Aplicația Gateway pentru studiul de caz 1 (deasupra) și studiul de caz 2 (sub) la recurențe de: a) 100 ms b) 10 ms c) 2 ms d) 1 ms

În ciuda scăderii semnificative a eficienței în condițiile menționate (pulsuri neregulate, în consecință transmisii cu stabilitate redusă la intervale sub 10 ms), interacțiunea dintre SOME/IP, DDS și eCAL nu poate fi considerată un factor care să contribuie la acest fapt, în cazul implementării actuale. Rutina de generare a semnalului, recunoaște starea semnalului de tip heartbeat, recepționat și transmis de toate cele 3 subcomponente ale aplicației gateway, și generează pulsul corespunzător cu o precizie ridicată. Pentru o mai bună monitorizare a procesului, a fost dezvoltată o rutină adițională, pentru interpretarea pulsului de tip heartbeat, pentru toate thread-urile aplicației gateway, luând în considerare toate secvențele de recepție și transmisie. Astfel, devine mai ușor de urmărit, dacă thread-urile sunt sincronizate și interacționează corect. În Figura 47, se poate observa pulsul interpretat pe baza evenimentului de tip heartbeat, în funcție de secvențele de recepție și de transmisie ale aplicației, pentru iterații succesive. Rezultatul relevă o gestionare corectă a datelor transmise între thread-urile separate, specifice fiecărui protocol.

```

DDS Message INFORMATION 0 RECEIVED
VSOMEIP Send event number 0
ECAL Sending message: Heart-Beat SIGNAL 0

DDS Message INFORMATION 1 RECEIVED
VSOMEIP Send event number 1
ECAL Sending message: Heart-Beat SIGNAL 1

DDS Message INFORMATION 0 RECEIVED
VSOMEIP Send event number 0
ECAL Sending message: Heart-Beat SIGNAL 0

DDS Message INFORMATION 1 RECEIVED
VSOMEIP Send event number 1
ECAL Sending message: Heart-Beat SIGNAL 1

DDS Message INFORMATION 0 RECEIVED
VSOMEIP Send event number 0
ECAL Sending message: Heart-Beat SIGNAL 0

DDS Message INFORMATION 1 RECEIVED
VSOMEIP Send event number 1
ECAL Sending message: Heart-Beat SIGNAL 1

DDS Message INFORMATION 0 RECEIVED
VSOMEIP Send event number 0
ECAL Sending message: Heart-Beat SIGNAL 0

DDS Message INFORMATION 1 RECEIVED
VSOMEIP Send event number 1
ECAL Sending message: Heart-Beat SIGNAL 1

DDS Message INFORMATION 0 RECEIVED
VSOMEIP Send event number 0
ECAL Sending message: Heart-Beat SIGNAL 0

DDS Message INFORMATION 1 RECEIVED
VSOMEIP Send event number 1
ECAL Sending message: Heart-Beat SIGNAL 1

DDS Message INFORMATION 0 RECEIVED
VSOMEIP Send event number 0
ECAL Sending message: Heart-Beat SIGNAL 0

DDS Message INFORMATION 1 RECEIVED
VSOMEIP Send event number 1
ECAL Sending message: Heart-Beat SIGNAL 1

DDS Message INFORMATION 0 RECEIVED
VSOMEIP Send event number 0
ECAL Sending message: Heart-Beat SIGNAL 0

DDS Message INFORMATION 1 RECEIVED
VSOMEIP Send event number 1
^C2021-11-24 10:21:19.861017 [Info] STOP OFFER(1277): [1234.5678:0.0] (true)
ECAL Sending message: Heart-Beat SIGNAL 1

```

Figura 47. Interpretarea pulsului, pe baza evenimentului de tip heartbeat, în conformitate cu secvențele de recepție și de transmisie ale aplicației gateway, în studiul de caz 2.

Strategia multithreading, pentru gestionarea procedurilor de comunicare, sporește eficiența sistemului și permite o separație clară între particularitățile tehnologice din cadrul aceleiași aplicații software. Arhitectura sistemului a fost validată pe baza unor rezultate similare (rezultatele procentuale generate de

140 | 6. Tehnologii IoT și Automotive de comunicație, aplicate într-un context V2X prin intermediul unui Gateway Multi-Protocol

mecanismul data-bufering se aliniază cu analiza calitativă ce vizează stabilitatea transmisiunilor) și satisfăcătoare, în ambele studii de caz, oferind o abordare flexibilă și scalabilă, ce permite gestionarea mai multor dispozitive și noduri, cu roluri și origini tehnologice diferite.

Cu toate obiectivele propuse, îndeplinite, este necesară o examinare mai atentă cu privire la cele mai importante concluzii, alături de o analiză a tehnologiilor implicate și a importanței conceptului pentru industrie. Aplicația gateway multi-protocol a reușit să interfațeze cu succes entitățile SOME/IP, DDS și eCAL, luând în considerare criteriile de eficiență și fiabilitate și adaptând conceptul la un scenariu de comunicație V2X. Arhitectura definită a fost adecvată, pentru interacțiunea dintre cele 3 tehnologii de comunicare diferite, în ambele studii de caz, fiecare dintre acestea fiind bazat pe o versiune diferită a aplicației gateway și având particularități majore. Rezultatele referitoare la comportamentul în timp real, au oferit o mai bună înțelegere asupra provocărilor privind comunicarea Ethernet în contextul IoT, iar metodele și strategiile utilizate pentru generarea rezultatelor pot fi aplicate la o scară mai largă, oferind soluții clare de îmbunătățire. A fost demonstrată compatibilitatea dintre SOME/IP, DDS și eCAL, oferind o imagine de ansamblu asupra perspectivelor, în diferite domenii industriale. S-au identificat anumite avantaje și dezavantaje aferente fiecărui protocol, după cum se arată în tabelul 12.

tabelul 12. Avantaje și dezavantaje aferente protocoalelor SOME/IP, DDS și eCAL

Protocol	Avantaje	Dezavantaje
SOME/IP	<ul style="list-style-type: none"> În conformitate cu standardul AUTOSAR Validat în multiple cazuri de utilizare automotive Prezent atât în platforma clasică cât și în cea adaptive Sigur și eficient 	<ul style="list-style-type: none"> Proces de configurare complex
DDS	<ul style="list-style-type: none"> În conformitate cu standardul AUTOSAR Oferă mecanisme multiple ce asigură flexibilitate și scalabilitate Prezent în platforma adaptive Validat în multiple aplicații și cazuri de utilizare IoT 	<ul style="list-style-type: none"> Nu este răspândit în domeniul automotive în ciuda faptului că este conform cu AUTOSAR
eCAL	<ul style="list-style-type: none"> Eficient, intuitiv și ușor de folosit în scenarii de comunicație Ethernet Proces de configurare ușor Potențial ridicat pentru cazuri de utilizare industriale 	<ul style="list-style-type: none"> Deocamdată nu este în conformitate cu standardul AUTOSAR Nu este foarte cunoscut Nu este aplicat la întregul potențial în zone tehnice explicite

Pe baza studiului curent și a contextului industrial, este indubitabil că SOME/IP reprezintă o soluție fiabilă și eficientă pentru comunicarea Ethernet, fiind susținut de standardul AUTOSAR și prezent pe scară largă în aplicațiile din domeniul automotive. Fiind complex și adecvat pentru mai multe scenarii, procesul de configurare necesită timp și efort, în special în cazul arhitecturilor complexe orientate pe servicii, în comparație cu un protocol IoT. DDS devine din ce în ce mai prezent în sistemele industriale IoT, iar mecanismele sale oferă flexibilitate și scalabilitate sporită. Conformitatea cu standardul AUTOSAR îl face potrivit și pentru aplicațiile din domeniul automotive și orice posibile interacțiuni cu middleware-ul SOME/IP, într-un context legat de sectorul automotive, reprezintă un subiect de cercetare de mare interes. Middleware-ul eCAL nu este foarte cunoscut, însă oferă o alternativă ușor de utilizat și intuitivă, pentru transmiterea mesajelor pe Ethernet. Potențialul său nu este pe deplin determinat, așadar interacțiunea cu alte soluții middleware va contribui la o mai bună înțelegere a modului în care poate fi aplicat.

După analiza protocoalelor de comunicare vizate, este important să se examineze conceptul actual într-un posibil context industrial. Pe lângă conformitatea cu standardul AUTOSAR deja menționată ca factor favorizant, tranziția aplicațiilor SOME/IP și DDS către industrie, ar trebui să fie fezabilă sub forma unor aplicații adaptive, compatibile cu microprocesoarele actuale, ce sunt desemnate ca fiind master-controlere și zone-controlere pentru diferite arii funcționale din interiorul vehiculului. Fiind parte a platformei adaptive, aceste aplicații ar trebui să respecte principiul de a fi hardware-agnostice și compatibile cu sistemele de operare POSIX. Arhitectura definită asigură o astfel de diversitate pentru dispozitivele și sistemele de operare utilizate. Pentru provocarea de a gestiona un număr mare de dependențe impuse de protocoalele vizate, sunt disponibile soluții specifice scenariilor industriale. Utilizarea sistemelor de operare POSIX particularizate, este o practică obișnuită care a devenit tot mai populară în ultimii ani, sistemele de operare fiind adaptate pentru a funcționa pe dispozitive hardware specifice. Un astfel de sistem de operare poate fi configurat pentru a satisface, necesitățile specifice pentru majoritatea protocoalelor bazate pe Ethernet, asigurând compatibilitatea între acestea și permițând tranziția, de la un prototip care utilizează biblioteci open-source, la un produs industrial. O soluție demnă de menționat, pentru adaptarea sistemelor de operare POSIX, este proiectul Yocto (a se vedea [122]). Deși SOME/IP și DDS ar putea fi stabilite ca aplicații adaptive într-un scenariu industrial autentic, prezența eCAL și DDS permite o percepție mai largă pentru conceptul actual, atunci când se examinează potențialul industrial. În prezent, considerat mai mult un protocol IoT, DDS generează posibilități reale de a viza scenarii de comunicație V2X, în momentul în care prototipurile funcționale prezintă principalele implicații și stabilesc așteptări realiste. Rezultatele cuantificabile privind stabilitatea, fiabilitatea, eficiența și comportamentul în timp real, devin mai valoroase în acest caz, permițând o mai bună înțelegere în legătură cu direcțiile curente și viitoare. În Figura 48 este prezentată o descriere concisă a modului în care tehnologiile analizate și conceptul actual ar putea fi adaptate la industrie.

142 | 6. Tehnologii IoT și Automotive de comunicație, aplicate într-un context V2X prin intermediul unui Gateway Multi-Protocol

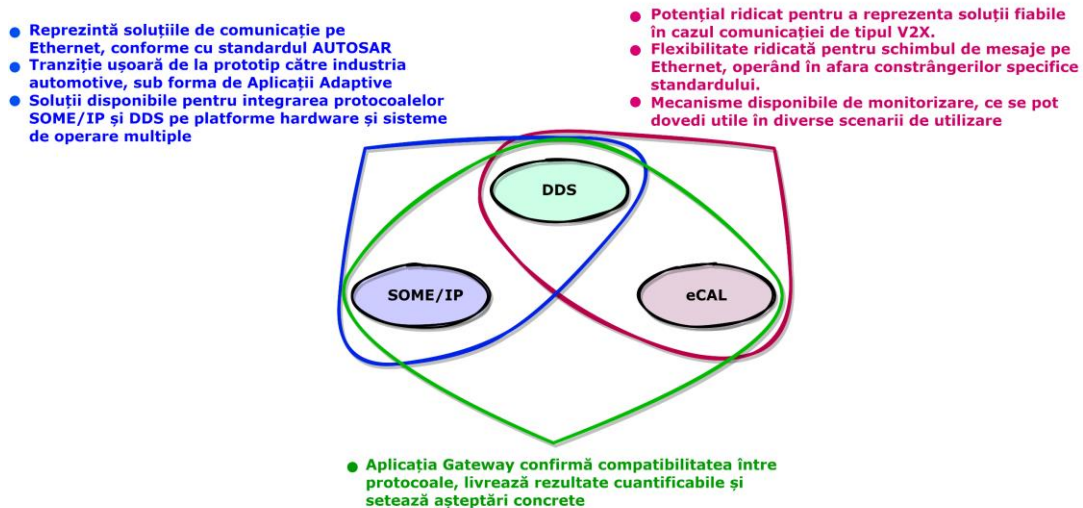


Figura 48. O imagine de ansamblu asupra tehnologiilor utilizate

6.6. Concluzii

Structurile de comunicație bazate pe Ethernet reprezintă fundamentul conceptului, eficiența și fiabilitatea în orice circumstanțe fiind obiective majore pentru orice soluție pe termen lung. Protocoalele de comunicare validate, oferă multiple mecanisme desemnate pentru a satisface nevoile industriei și sunt bine stabilite în anumite sectoare industriale. Progresele în desfășurare, din zonele IoT, oferă direcții clare de dezvoltare și provocări care trebuie abordate cât mai curând pentru a obține efectul scontat. Domeniul automotive este, de asemenea, în tranziție către era electrificării, fiind necesare schimbări cu impact din punct de vedere arhitectural și al dezvoltării, care să îmbunătățească strategiile actuale de business. Tehnologiile de comunicare bazate pe Ethernet sunt esențiale pentru stadiul actual al industriei, iar impactul lor este de așteptat să crească în următorii ani. Cu SOME/IP consacrat în domeniul automotive și utilizat pe scară largă atât în platforma clasică, cât și în cea adaptive, și cu DDS adoptat în implementările industriale și IoT, fiind în conformitate cu standardul AUTOSAR, interacțiunea dintre aceste protocoale majore este foarte probabilă în viitor, având în vedere multitudinea de arhitecturi validate deja în principalele domenii industriale. Evoluția sistemelor actuale nu va fi sustenabilă prin redefinirea arhitecturilor pentru a se conforma unei singure tehnologii de comunicare, ci mai degrabă printr-un proces de îmbunătățire care ar putea facilita interoperabilitatea între principalele sisteme industriale. Un astfel de proces se poate baza pe componente gateway eficiente și fiabile, capabile să satisfacă o gamă largă de cerințe și să asigure schimbul de date între entități independente cu diverse particularități software și hardware. Tehnologiile emergente care oferă avantaje

distincte și o calitate consistentă a serviciilor, nu trebuie ignorate, ci trebuie integrate într-un context comun cu standarde și protocoale certificate, și evaluate în funcție de criteriile industriale.

Etapa curentă de cercetare, avut ca scop combinarea a trei protocoale de comunicare, fezabile pentru scenarii automotivă și IoT, concepând o arhitectură care să asigure un grad ridicat de flexibilitate și scalabilitate. Rezultatele obținute pun în evidență capacitățile de funcționare în timp real ale sistemului, semnalând limitele în ceea ce privește eficiența și stabilitatea. Provocările întâmpinate în faza de dezvoltare au fost generate de integrarea tehnologiilor vizate, în aceeași arhitectură și în aplicația gateway, multiple dependențe fiind greu de rezolvat. Pe lângă efortul de a concepe și a dezvolta soluția, împreună cu toate entitățile complementare, a fost proiectat și implementat un cod de testare suplimentar. Au fost efectuate proceduri de testare individuale pentru fiecare protocol înainte de a obține rezultate detaliate. Arhitectura definită a trebuit să anticipeze nevoile sistemului și să minimizeze efortul de configurare și testare, urmând principiile industriale autentice. Aplicația gateway multi-protocol, este utilizată într-un context de comunicație V2X, vizând aspecte specifice automotivă și IoT, de mare interes pentru comunitățile industriale și academice.

Deși toate obiectivele propuse au fost atinse, există direcții de perfecționare, care au potențialul de a crește performanța sistemului. Adoptarea tehnologiei Time Sensitive Networking ar putea asigura un comportament sincronizat pentru toate dispozitivele implicate, garantând livrarea datelor în conformitate cu recurențele configurate. Utilizarea unor sisteme de operare cu caracteristici specializate pentru cerințe stricte de funcționare în timp real, ar trebui să îmbunătățească capacitatea de gestionare a operațiunilor critice, chiar și în contextul unei arhitecturi complexe orientată pe servicii. Metodele și strategiile aplicate în ambele studii de caz, permit adăugarea pas cu pas a oricăror soluții de îmbunătățire, rezultatele rămânând cuantificabile și relevante.

Aplicația gateway multi-protocol, ca parte a unei arhitecturi orientate pe servicii, compusă din mai multe tehnologii, dispozitive și sisteme de operare, expune capacitățile de comunicare prin Ethernet, luând în considerare obiective și provocări industriale. Conceptul este relevant pentru domeniile automotivă și IoT, gestionând tendințe și strategii de mare importanță, atât din perspectivă industrială, cât și academică. Fazele de tranziție ale industriei creează oportunități de dezvoltare masivă, iar capacitățile de comunicare pe Ethernet, vor reprezenta fundația viitoarelor sisteme de înaltă performanță.

7. CONCLUZIILE TEZEI

7.1. Concluzii finale

Abordarea principiilor Industry 4.0 în contextul soluțiilor automotive – IoT pentru interoperare s-a dovedit eficientă, producând numeroase rezultate de impact. Pe parcursul studiului s-au adus în discuție teme de interes specifice mai multor domenii industriale, iar dezvoltările realizate și perspectivele generate în urma analizelor și experimentelor efectuate, au extins aplicabilitatea tehnologiilor vizate și au oferit soluții fezabile, în concordanță cu cerințele industriale.

De-a lungul etapelor de cercetare, pe lângă dezvoltarea soluțiilor software, mult efort a fost investit în analiza arhitecturală a sistemelor industriale și în definirea unor arhitecturi scalabile și eficiente pentru a asigura interoperabilitatea a multiple tehnologii și echipamente hardware cu diverse capabilități. Acest demers pleacă de la prezumția că în viitorul apropiat, pentru creșterea interoperabilității sistemelor complexe, arhitecturi specializate și validate vor trebui să interacționeze, menținând standardele de eficacitate și flexibilitate, iar concomitent, vor putea răspunde cerințelor specifice altor sectoare industriale. Acest principiu evidențiază nevoia de a trata setul de cerințe și particularități arhitecturale într-un context comun, indiferent de sectorul industrial și de tehnologiile aferente acestuia. Conceptele și arhitecturile definite pe parcursul cercetării curente, adoptă principii atât din sectorul automotive, în special în cazul criteriilor de funcționare în timp real și de sincronizare a procedurilor de la nivelul tuturor dispozitivelor implicate, cât și din sectorul automatizărilor unde mecanismele ce oferă flexibilitate sunt dezirabile, punându-se mare preț pe interacțiunea între numeroase entități separate și pe transmiterea unui volum considerabil de date. Îmbunătățirile aduse și validarea acestora în scenarii practice, reprezintă un progres direct aplicabil în procesele de dezvoltare pentru sistemele de comunicație industriale.

De asemenea, și în cazul soluțiilor software obținute s-a urmărit o aplicabilitate directă în industrie, existând la fiecare etapă a studiului o descriere concisă a pașilor de implementare și o analiză minuțioasă asupra avantajelor și dezavantajelor observate în urma experimentelor. Pentru a se facilita această aplicabilitate industrială directă, echipamentele hardware utilizate, cât și sistemele de operare, au fost selectate în baza criteriilor și trendurilor de actualitate. Portabilitatea și reproductibilitatea au fost asigurate în general prin utilizarea implementărilor și sistemelor de bază open-source, fiind menținută o conformitate cu metodologia industrială curentă.

Luând în considerare diversitatea dezvoltărilor și cazurilor de utilizare vizate, o importanță aparte a avut identificarea limitelor de funcționare pentru sistemele complexe implementate, în special din perspectiva schimbului de date în timp real pentru tehnologiile de comunicație pe Ethernet. Criteriile și metodele utilizate pentru acest tip de evaluare a capacităților de comunicare, s-au dovedit adecvate, iar rezultatele generate au oferit o imagine de ansamblu cuprinzătoare, fapt ce reprezintă un progres semnificativ în contextul IIoT.

Din punct de vedere al protocoalelor de comunicare, au fost utilizate patru protocoale distincte de nivel înalt pe parcursul studiului, acestea fiind OPC UA, SOME/IP, DDS și eCAL. Numeroasele tipuri și versiuni de aplicații gateway dezvoltate, au contribuit la extinderea aplicabilității acestor tehnologii și a mecanismelor aferente, au validat compatibilitatea acestora atât din punct de vedere funcțional cât și din punct de vedere al cerințelor și obiectivelor îndeplinite și astfel, s-a conturat o perspectivă mai cuprinzătoare în legătură cu metodele disponibile de interacțiune între sectoare industriale separate.

Contribuția științifică adusă prin intermediul acestui studiu doctoral, extinde concepția despre interoperabilitatea sistemelor complexe în cadrul domeniilor automotive și IoT, iar acest fapt este confirmat de seria de lucrări științifice publicate pe parcursul celor trei ani, la jurnale de specialitate de mare prestigiu, indexate Web of Science. Eforturile și rezultatele acestui studiu, contribuie la tranziția către principiile Industry 4.0, pentru arhitecturile și soluțiile de interoperare ce vizează domeniul automotive, în concordanță cu strategii și metode specifice IIoT. Perspectivele obținute oferă noi oportunități de cercetare și dezvoltare pentru sporirea performanțelor sistemelor complexe de comunicație pe Ethernet.

7.2. Contribuții personale

Pe baza obiectivelor stabilite pentru actualul parcurs doctoral, în această secțiune se enumeră o serie de contribuții personale ale autorului, înregistrate pe parcursul tuturor etapelor de cercetare, după cum urmează :

- Efectuarea unei cercetări amănunțite în legătură cu domeniul și tema actualei teze, fiind identificate direcții clare de dezvoltare, relevante pentru multiple sectoare industriale.
- Implementarea practică a unei aplicații gateway structurată pe mai multe versiuni, între OPC UA și SOME/IP, în contextul unui scenariu de comunicație de tipul Car-to-Infrastructure. (Capitolul 2)
- Analiza tehnologiei TSN, identificând avantajele și potențialul impact, în cazul implementărilor din cadrul fiecărei etape de cercetare.
- Îmbunătățirea mecanismului OPC UA Publish-Subscribe din punct de vedere arhitectural, în contextul funcționării în timp real, oferind o perspectivă amănunțită în legătură cu majoritatea implementărilor disponibile, bazate pe paradigma Publish-Subscribe. (Capitolul 3)
- Extinderea aplicabilității protocolului OPC UA prin utilizarea mecanismului Publish-Subscribe pentru transmiterea de imagini, analizând capacitățile de gestionare a unui volum semnificativ de date. (Capitolul 4)
- Implementarea unei aplicații DDS - OPC UA gateway, analizând coexistența celor două tehnologii în context comun și paralel, în cadrul unei arhitecturi definite conform principiilor industriale. (Capitolul 5)
- Definirea unor criterii și metode elocvente și scalabile pentru identificarea limitelor de funcționare în timp real, pentru sistemele complexe de comunicație pe Ethernet. (Capitolele 5 și 6)

- Dezvoltarea unei soluții gateway multi-protocol, ce facilitează interoperabilitatea între SOME/IP, DDS și eCAL, în contextul unui scenariu de comunicare V2X, analizând și combinând tehnologii specifice domeniilor automotive și IoT. (Capitolul 6)
- Definirea a numeroase arhitecturi complexe ce includ tehnologii și dispozitive diverse, în concordanță cu principiile și cerințele industriale.
- Obținerea unor rezultate cuantificabile ce prezintă capabilitățile actuale ale sistemelor de comunicare și interoperare, atât din punct de vedere al eficienței, cât și al stabilității, sporind astfel răspândirea utilizării tehnologiilor aferente către noi situații și procese industriale. (Capitolele 5 și 6)

Desigur, pe lângă obiectivele îndeplinite și contribuțiile aduse până în momentul de față, fiecare etapă a cercetării oferă oportunități adiționale de dezvoltare, în multe cazuri neexistând alternative la implementările curente. Având în vedere focalizarea asupra domeniului automotive, în mod constant trebuie să existe o raportare la standardul AUTOSAR și la implicațiile utilizării tehnologiilor suplimentare. În consecință, pentru adoptarea soluțiilor, conformitatea tehnologiilor utilizate cu standardul nu este singurul aspect important, ci și definirea unor metodologii și procese de lucru specifice. Din punct de vedere funcțional, tehnologia TSN reprezintă un viitor pas major în procesul de dezvoltare al soluțiilor de interoperabilitate, iar potențialele beneficii nu se rezumă la un singur sector industrial, ci se aliniază la principiile Industry 4.0. Din punct de vedere al dispozitivelor hardware și al sistemelor de operare utilizate, studiul actual a oferit o perspectivă diversificată, autentică din punct de vedere industrial, însă există în continuare nevoie de studii relevante ce analizează capacitățile de operare în timp real în contextul comunicației Ethernet, astfel încât utilizarea tehnologiilor complexe să fie posibilă inclusiv la nivelul dispozitivelor de teren cu resurse limitate, și astfel capabilitățile de interconectare să devină sporite atât în domeniul automotive cât și în alte sectoare adiacente. Actualul demers științific contribuie la generarea unor soluții de interoperare eficiente, clarificând aspecte de interes în comunitatea academică, reprezentând însă doar unul din multitudinea de pași necesari pentru tranziția către principiile Industry 4.0 și IIoT.

7.3. Publicații

Ultimul subcapitol este rezervat prezentării setului de lucrări științifice publicate, sau în curs de publicare, bazate pe etapele cercetării curente. Având o importanță semnificativă atât din perspectiva diseminării rezultatelor obținute, cât și pentru validarea procesului minuțios de cercetare întreprins în cei trei ani, articolele publicate au fost recepționate cu un nivel de interes peste așteptări de către comunitatea academică. Colaborarea de succes, cu întreaga echipă de coordonare a făcut posibilă îndeplinirea tuturor obiectivelor în materie de publicații, astfel o ultimă serie de mulțumiri se cuvin tuturor colaboratorilor și coautorilor acestor lucrări.

Din punct de vedere statistic, pe parcursul cercetării doctorale au fost elaborate în total 6 lucrări științifice, toate indexate Web of Science, lucrarea inițială fiind prezentată în cadrul unei reputeate conferințe. Cele cinci lucrări ce au urmat, au fost publicate în jurnalul Sensors, plasat în categoria Q1 la momentul trimiterii către

publicare. Factorul de impact cumulat în urma publicării articolelor științifice (calculat pe baza factorului de impact curent al jurnalului) este 19.235 ($5 * 3.847$). Detalii suplimentare pot fi observate pe baza enumerării lucrărilor, în ordine cronologică în funcție de data publicării, după cum urmează:

1. A. Ioana and A. Korodi, "VSOMEIP - OPC UA Gateway Solution for the Automotive Industry," *2019 IEEE International Conference on Engineering, Technology and Innovation (ICE/ITMC)*, 2019, pp. 1-6, doi: 10.1109/ICE.2019.8792619.
2. Ioana, A.; Korodi, A. OPC UA Publish-Subscribe and VSOME/IP Notify-Subscribe Based Gateway Application in the Context of Car to Infrastructure Communication. *Sensors* 2020, *20*, 4624. <https://doi.org/10.3390/s20164624>. Statisticile de accesare pot fi observate în graficul de mai jos (Figura 49), preluat de pe site-ul jurnalului *Sensors* la data de 13 Iunie 2022.

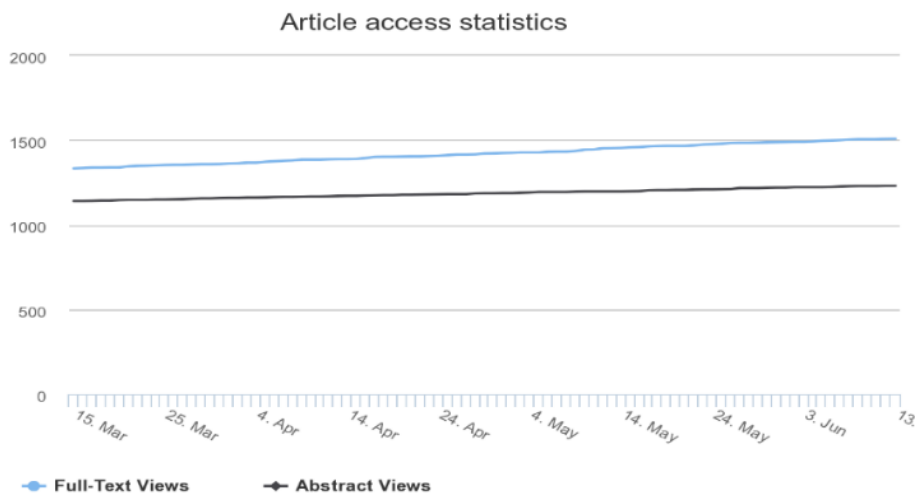


Figura 49. Statisticile de accesare pentru a doua lucrare (a se vedea [123])

3. Ioana, A.; Korodi, A. Improving OPC UA Publish-Subscribe Mechanism over UDP with Synchronization Algorithm and Multithreading Broker Application. *Sensors* 2020, *20*, 5591. <https://doi.org/10.3390/s20195591>. Statisticile de accesare pot fi observate în graficul de mai jos (Figura 50), preluat de pe site-ul jurnalului *Sensors* la data de 13 Iunie 2022.

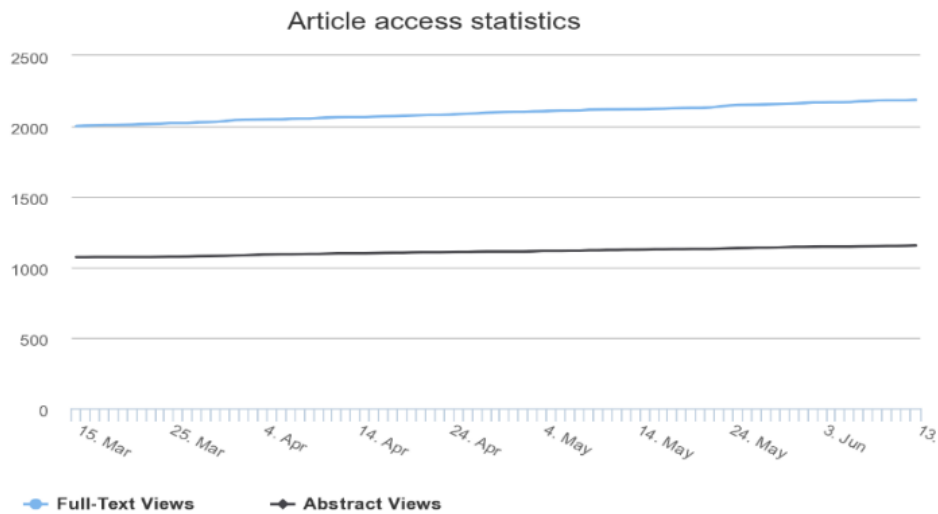


Figura 50. Statisticile de accesare pentru a treia lucrare (a se vedea [124])

- Ioana, A.; Burlacu, C.; Korodi, A. Approaching OPC UA Publish-Subscribe in the Context of UDP-Based Multi-Channel Communication and Image Transmission. *Sensors* 2021, *21*, 1296. <https://doi.org/10.3390/s21041296>. Statisticile de accesare pot fi observate în graficul de mai jos (Figura 51), preluat de pe site-ul jurnalului *Sensors* la data de 13 Iunie 2022.

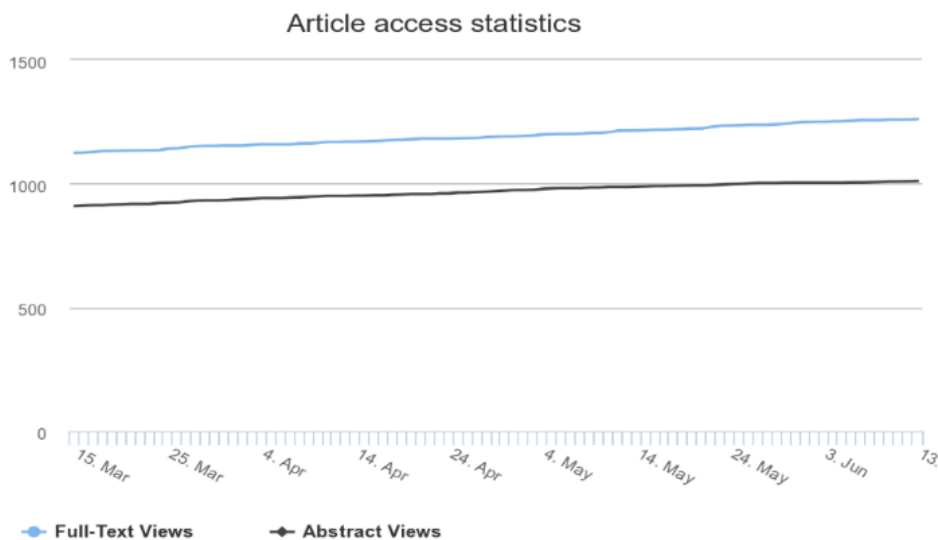


Figura 51. Statisticile de accesare pentru a patra lucrare (a se vedea [125])

5. Ioana, A.; Korodi, A. DDS and OPC UA Protocol Coexistence Solution in Real-Time and Industry 4.0 Context Using Non-Ideal Infrastructure. *Sensors* 2021, *21*, 7760. <https://doi.org/10.3390/s21227760>. Statisticile de accesare pot fi observate în graficul de mai jos (Figura 52), preluat de pe site-ul jurnalului *Sensors* la data de 13 Iunie 2022.

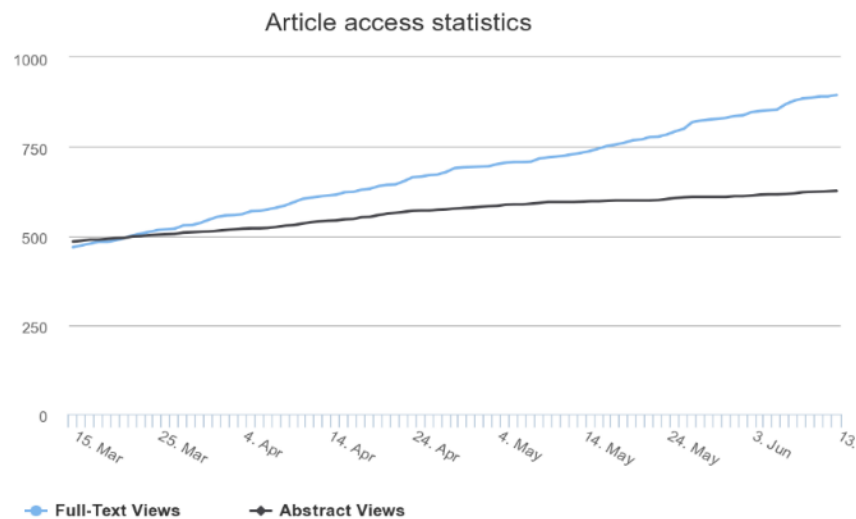


Figura 52. Statisticile de accesare pentru a cincea lucrare (a se vedea [126])

6. Ioana, A.; Korodi, A.; Silea, I. Automotive IoT Ethernet-Based Communication Technologies Applied in a V2X Context via a Multi-Protocol Gateway. *Sensors* 2022, *22*, 6382. <https://doi.org/10.3390/s22176382>

În concluzie, în urma tuturor obiectivelor îndeplinite și a rezultatelor obținute, actuala teză se încheie, oferind soluții de interoperare eficiente pentru sectoarele automotive - IoT, dar și perspective și direcții viitoare de cercetare, contribuind la tranziția sistemelor complexe de comunicație pe Ethernet, către principiile Industry 4.0 și IIoT.

BIBLIOGRAFIE

- [1] Cavalieri, S.; Salafia, M.G. Insights into mapping solutions based on opc ua information model applied to the industry 4.0 asset administration shell. *Computers* 2020, 9, 28. [CrossRef]
- [2] Gutierrez-Guerrero, J.M.; Holgado-Terriza, J.A. Automatic configuration of OPC UA for industrial internet of things environments. *Electronics* 2019, 8, 600. [CrossRef]
- [3] González, I.; Calderón, A.J.; Figueiredo, J.; Sousa, J.M.C. A literature survey on open platform communications (OPC) applied to advanced industrial environments. *Electronics* 2019, 8, 510. [CrossRef]
- [4] Lee, T.-Y.; Lin, I.-A.; Wang, J.-J.; Tsai, J.-T. A reliability scheduling algorithm for the static segment of flexray on vehicle networks. *Sensors* 2018, 18, 3783. [CrossRef] [PubMed]
- [5] Osigbemeh, M.; Onuu, M.U.; Asaolu, O.S. Design and development of an improved traffic light control system using hybrid lighting system. *J. Traffic Transp. Eng.* 2017, 4, 88–95. [CrossRef]
- [6] Chiang, C.-C.; Ho, M.-C.; Liao, H.-S.; Pratama, A.K. Detecting and recognizing traffic lights by genetic approximate ellipse detection and spatial texture layouts. *Int. J. Innov. Comput. Inf. Control* 2017, 7, 6919–6934.
- [7] Nath, S.; Pal, C.; Kandar, D. Design of an FPGA Based Intelligence Traffic Light Controller with VHDL. In *Proceedings of the International Conference on Radar, Communication and Computing (ICRCC), Tiruvannamalai, India, 21–22 December 2012*; pp. 92–97. [CrossRef]
- [8] Fazliu, Z.L.; Malandrino, F.; Chiasserini, C. mmWave in Vehicular Networks: Leveraging Traffic Signals for Beam Design. In *Proceedings of the IEEE 20th International Symposium on "A World of Wireless, Mobile and Multimedia Networks" (WoWMoM), Washington, DC, USA, 10–12 June 2019*; pp. 1–6. [CrossRef]
- [9] Zinner, H.J.; Noebauer, T.; Gallner, J.S.; Waas, T. Application and realization of gateways between conventional automotive and IP/Ethernet-based networks. In *Proceedings of the 48th ACM/EDAC/IEEE Design Automation Conference (DAC), New York, NY, USA, 5–9 June 2011*; pp. 1–6.
- [10] Sung, K.; Lee, J.; Shin, J. Study of CAN-to-3GPP LTE gateway architecture for automotive safety in V21 environment. In *Proceedings of the IEEE International Conference on Advanced Communications Technology, Kota Kinabalu, Malaesya, 1–3 July 2015*.
- [11] Li, H.; Chen, L.; Chang, W.; Tang, J.; Li, K.S. Design and development of an extensible multi-protocol automotive gateway. In *Proceedings of the IEEE International Conference on Consumer Electronics-Taiwan (ICCE-TW), Nantou, Taiwan, 27–29 May 2016*; pp. 1–2. [CrossRef]
- [12] Inkoom, S.; Sobanjo, J.; Chicken, E. Competing risks models for the assessment of intelligent transportation systems devices: A case study for connected and autonomous vehicle applications. *Infrastructures* 2020, 5, 30. [CrossRef]
- [13] Chen, J.; Xue, Z.; Fan, D. Deep reinforcement learning based left-turn connected and automated vehicle control at signalized intersection in vehicle-to-infrastructure environment. *Information* 2020, 11, 77. [CrossRef]

- [14] Darlosan, R.; Marghescu, I.; Lica, C. Intra-car communications and diagnosis solutions. In Proceedings of the International Conference on Communications (COMM), Bucharest, Romania, 9–11 June 2016; pp. 341–346. [CrossRef]
- [15] Swamy, N. Evaluation of OPC-UA Technology in a Car-2x Communication towards an Industry 4.0 Driven Automotive Domain. Master's Thesis, Technische Universität Chemnitz, Chemnitz, Germany, 2017.
- [16] Cupek, R.; Ziebinski, A.; Drewniak, M. An OPC UA server as a gateway that shares CAN network data and engineering knowledge. In Proceedings of the IEEE International Conference on Industrial Technology (ICIT), Toronto, ON, Canada, 22–25 March 2017; pp. 1424–1429. [CrossRef]
- [17] Neumann, A.; Mytych, M.J.; Wesemann, D.; Wisniewski, L.; Jasperneite, J. Approaches for in-vehicle communication—An analysis and outlook. In Communications in Computer and Information Science; Springer: Berlin/Heidelberg, Germany, 2017; Volume 718.
- [18] Shreejith, S.; Mundhenk, P.; Ettner, A.; Suhaib, A.; Steinhorst, S.; Lukasiewicz, M.; Chakraborty, S. VEGa: A high performance vehicular ethernet gateway on hybrid FPGA. *IEEE Trans. Comput.* 2017, 66, 1790–1803. [CrossRef]
- [19] Bello, L.L.; Mariani, R.; Mubeen, S.; Saponara, S. Recent advances and trends in on-board embedded and networked automotive systems. *IEEE Trans. Ind. Inform.* 2019, 15, 1038–1051. [CrossRef]
- [20] Han, B.; Peng, S.; Wu, C.; Wang, X.; Wang, B. LoRa-based physical layer key generation for secure v2v/v2i communications. *Sensors* 2020, 20, 682. [CrossRef] [PubMed]
- [21] Eckhardt, A.; Müller, S.; Leurs, L. An evaluation of the applicability of OPC UA Publish Subscribe on factory automation use cases. In Proceedings of the IEEE 23rd International Conference on Emerging Technologies and Factory Automation (ETFA), Torino, Italy, 4–7 September 2018; pp. 1071–1074.
- [22] Gogolev, A.; Mendoza, F.; Braun, R. TSN-Enabled OPC UA in Field Devices. In Proceedings of the IEEE 23rd International Conference on Emerging Technologies and Factory Automation (ETFA), Torino, Italy, 4–7 September 2018; pp. 297–303.
- [23] Ioana, A.; Korodi, A. VSOMEIP-OPC UA Gateway Solution for the Automotive Industry. In Proceedings of the IEEE International Conference on Engineering, Technology and Innovation (ICE/ITMC), Valbonne Sophia-Antipolis, France, 17–19 June 2019; pp. 1–6. [CrossRef]
- [24] OPC 10000-14-UA Specification Part 14 PubSub; OPC Foundation: Scottsdale, AR, USA, 1 April 2018.
- [25] Open62541 SDK. Available online: <https://github.com/open62541/open62541>
- [26] AUTOSAR. Requirements on Time Synchronization; AUTOSAR: Munich, Germany, 2019.
- [27] Pfrommer, J.; Ebner, A.; Ravikumar, S.; Karunakaran, B. Open Source OPC UA PubSub Over TSN for Realtime Industrial Communication. In Proceedings of the IEEE 23rd International Conference on Emerging Technologies and Factory Automation (ETFA), Torino, Italy, 4–7 September 2018; pp. 1087–1090. [CrossRef]
- [28] Minchala, L.I.; Peralta, J.; Mata-Quevedo, P.; Rojas, J. An Approach to Industrial Automation Based on Low-Cost Embedded Platforms and Open Software. *Appl. Sci.* 2020, 10, 4696. [CrossRef]

- [29] Nicolae, A.; Korodi, A.; Silea, I. An Overview of Industry 4.0 Development Directions in the Industrial Internet of Things Context. *Rom. J. Inf. Sci. Technol.* 2019, 22, 183–201.
- [30] Paszkiewicz, A.; Bolanowski, M.; Budzik, G.; Przeszłowski, Ł.; Oleksy, M. Process of Creating an Integrated Design and Manufacturing Environment as Part of the Structure of Industry 4.0. *Processes* 2020, 8, 1019. [CrossRef]
- [31] Korodi, A.; Radu, M.A.; Crisan, R. Non-Invasive Control Solution inside Higher-Level OPC UA based Wrapper for Optimizing Groups of Wastewater Systems. In *Proceedings of the IEEE 23rd International Conference on Emerging Technologies and Factory Automation (ETFA), Torino, Italy, 4–7 September 2018.* [CrossRef]
- [32] Nota, G.; Nota, F.D.; Peluso, D.; Toro Lazo, A. Energy Efficiency in Industry 4.0: The Case of Batch Production Processes. *Sustainability* 2020, 12, 6631. [CrossRef]
- [33] Korodi, A.; Crisan, R.; Nicolae, A.; Silea, I. Industrial Internet of Things and Fog Computing to Reduce Energy Consumption in Drinking Water Facilities. *Processes* 2020, 8, 282. [CrossRef]
- [34] Ye, X.; Hong, S.H. An AutomationML/OPC UA-based Industry 4.0 Solution for a Manufacturing System. In *Proceedings of the 2018 IEEE 23rd International Conference on Emerging Technologies and Factory Automation (ETFA), Torino, Italy, 4–7 September 2018; pp. 543–550.* [CrossRef]
- [35] Gogolev, A.; Braun, R.; Bauer, P. TSN Traffic Shaping for OPC UA Field Devices. In *Proceedings of the 2019 IEEE 17th International Conference on Industrial Informatics (INDIN), Helsinki, Finland, 23–25 July 2019; pp. 951–956.* [CrossRef]
- [36] Haskamp, H.; Orth, F.; Wermann, J.; Colombo, A.W. Implementing an OPC UA interface for legacy PLC-based automation systems using the Azure cloud: An ICPS-architecture with a retrofitted RFID system. In *Proceedings of the 2018 IEEE Industrial Cyber-Physical Systems (ICPS), St. Petersburg, Russia, 15–18 May 2018; pp. 115–121.* [CrossRef]
- [37] Peniak, P.; Bubenikova, E.; Spalek, J. Model of Integration Gateway for Communication of OPC/MQTT Devices. In *Proceedings of the 2020 Cybernetics & Informatics (K&I), Velke Karlovice, Czech Republic, 29 January–1 February 2020; pp. 1–5.* [CrossRef]
- [38] MQTT Sparkplug/Tahu. Available online: <https://www.cirrus-link.com/mqtt-sparkplug-tahu> (accessed on 2 September 2020).
- [39] AUTOSAR. SOME/IP Protocol Specification; Release 19–11; AUTOSAR: Munich, Germany, 2019.
- [40] Vidal, I.; Bellavista, P.; Sanchez-Aguero, V.; Garcia-Reinoso, J.; Valera, F.; Nogales, B.; Azcorra, A. Enabling Multi-Mission Interoperable UAS Using Data-Centric Communications. *Sensors* 2018, 18, 3421. [CrossRef] [PubMed]
- [41] Youssef, T.A.; Esfahani, M.M.; Mohammed, O. Data-Centric Communication Framework for Multicast IEC 61850 Routable GOOSE Messages over the WAN in Modern Power Systems. *Appl. Sci.* 2020, 10, 848. [CrossRef]
- [42] Tarkoma, S. *Publish/Subscribe Systems: Design and Principles*, 1st ed.; Wiley: Hoboken, NJ, USA, 2012.
- [43] Newman, W.S. *A Systematic Approach to Learning Robot Programming with ROS*, 1st ed.; Chapman and Hall/CRC: Boca Raton, FL, USA, 2017
- [44] Requirements on Time Synchronization. Available online: https://www.autosar.org/fileadmin/user_upload/standards/foundation/19-11/AUTOSAR_RS_TimeSync.pdf (accessed on 17 August 2020).

- [45] Ioana, A.; Korodi, A. OPC UA Publish-Subscribe and VSOME/IP Notify-Subscribe Based Gateway Application in the Context of Car to Infrastructure Communication. *Sensors* 2020, 20, 4624. [CrossRef] [PubMed]
- [46] OPC UA For Universal Robots. Available online: <https://rocketfarm.no/software-products/opc-ua/> (accessed on 2 September 2020)
- [47] Korodi, A.; Anitei, D.; Boitor, A.; Silea, I. Image-Processing-Based Low-Cost Fault Detection Solution for End-of-Line ECUs in Automotive Manufacturing. *Sensors* 2020, 20, 3520. [CrossRef] [PubMed]
- [48] Toc, S.I.; Korodi, A. Modbus-OPC UA Wrapper using Node-RED and IoT-2040 with application in the water industry. In Proceedings of the 16th IEEE International Symposium on Intelligent Systems and Informatics (SISY), Subotica, Serbia, 13–15 September 2018
- [49] Korodi, A.; Silea, I. Achieving Interoperability Using Low-Cost Middleware OPC UA Wrapping Structure. Case Study in the Water Industry. In Proceedings of the 15th IEEE International Conference on Industrial Informatics (INDIN), Emden, Germany, 24–26 July 2017; pp. 1223–1228.
- [50] Nicolae, A.; Korodi, A.; Silea, I. Identifying Data Dependencies as First Step to Obtain a Proactive Historian: Test Scenario in the Water Industry 4.0. *Water* 2019, 11, 1144. [CrossRef]
- [51] Mathias, S.G.; Schmied, S.; Grossmann, D.; Müller, R.K.; Mroß, B. A Compliance Testing Structure for Implementation of Industry Standards through OPC UA. In Proceedings of the 2020 25th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA), Vienna, Austria, 8–11 September 2020; pp. 1091–1094.
- [52] Ioana, A.; Korodi, A. Improving OPC UA Publish-Subscribe Mechanism over UDP with Synchronization Algorithm and Multithreading Broker Application. *Sensors* 2020, 20, 5591. [CrossRef] [PubMed]
- [53] Tang, S.; Zhu, Y.; Yuan, S.; Li, G. Intelligent Diagnosis towards Hydraulic Axial Piston Pump Using a Novel Integrated CNN Model. *Sensors* 2020, 20, 7152. [CrossRef] [PubMed]
- [54] Na, K.-M.; Lee, K.; Shin, S.-K.; Kim, H. Detecting Deformation on Pantograph Contact Strip of Railway Vehicle on Image Processing and Deep Learning. *Appl. Sci.* 2020, 10, 8509. [CrossRef]
- [55] Stark, E.; Kučera, E.; Haffner, O.; Drahoš, P.; Leskovský, R. Using Augmented Reality and Internet of Things for Control and Monitoring of Mechatronic Devices. *Electronics* 2020, 9, 1272. [CrossRef]
- [56] Yahiaoui, L.; Horgan, J.; Deegan, B.; Yogamani, S.; Hughes, C.; Denny, P. Overview and Empirical Analysis of ISP Parameter Tuning for Visual Perception in Autonomous Driving. *J. Imaging* 2019, 5, 78. [CrossRef]
- [57] Kang, S.; Chun, C.; Shim, S.; Ryu, S.; Baek, J. Real Time Image Processing System for Detecting Infrastructure Damage: Crack. In Proceedings of the 2019 IEEE International Conference on Consumer Electronics (ICCE), Las Vegas, NV, USA, 11–13 January 2019; pp. 1–3. [CrossRef]
- [58] Tian, D.; Zhang, C.; Duan, X.; Wang, X. An Automatic Car Accident Detection Method Based on Cooperative Vehicle Infrastructure Systems. *IEEE Access* 2019, 7, 127453–127463. [CrossRef]
- [59] Ahn, S.; Choi, J. Utilization of V2X Communications for Vehicle Queue Length Estimation. In Proceedings of the 2018 International Conference on Information and Communication Technology Convergence (ICTC), Jeju, Korea, 17–19 October 2018; pp. 645–648. [CrossRef]

- [60] Eckhardt, A.; Müller, S. Analysis of the Round Trip Time of OPC UA and TSN based Peer-to-Peer Communication. In Proceedings of the 2019 24th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA), Zaragoza, Spain, 10–13 September 2019; pp. 161–167.
- [61] Cenedese, A.; Frodella, M.; Tramarin, F.; Vitturi, A. Comparative assessment of different OPC UA open-source stacks for embedded systems. In Proceedings of the 2019 24th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA), Zaragoza, Spain, 10–13 September 2019; pp. 1127–1134.
- [62] Initiative: Field Level Communications (FLC) OPC Foundation Extends OPC UA Including TSN Down to Field Level; OPC Foundation: Scottsdale, AR, USA, 2019.
- [63] OPC UA for Programmable Logic Controllers Based on IEC61131-3; OPC Foundation: Scottsdale, AR, USA, 2010.
- [64] Panda, S.K.; Majumder, M.; Wisniewski, L.; Jasperneite, J. Real-time Industrial Communication by using OPC UA Field Level Communication. In Proceedings of the 2020 25th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA), Vienna, Austria, 8–11 September 2020; pp. 1143–1146.
- [65] Li, Y.; Jiang, J.; Lee, C.; Hong, S.H. Practical Implementation of an OPC UA TSN Communication Architecture for a Manufacturing System. *IEEE Access* 2020, 8, 200100–200111. [CrossRef]
- [66] Iatrou, C.P.; Ketznel, L.; Graube, M.; Häfner, M.; Urbas, L. Design classification of aggregating systems in intelligent information system architectures. In Proceedings of the 2020 25th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA), Vienna, Austria, 8–11 September 2020; pp. 745–752
- [67] Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You Only Look Once: Unified, Real-Time Object Detection. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 779–788.
- [68] Jiang, S.; Cao, J.; Wu, H.; Yang, Y. Fairness-based Packing of Industrial IoT Data in Permissioned Blockchains. *IEEE Trans. Ind. Inform.* 2020. [CrossRef]
- [69] Tidrea, A.; Korodi, A.; Silea, I. Cryptographic Considerations for Automation and SCADA Systems Using Trusted Platform Modules. *Sensors* 2019, 19, 4191. [CrossRef] [PubMed]
- [70] Leng, J.; Ye, S.; Zhou, M.; Zhao, J.L.; Liu, Q.; Guo, W.; Cao, W.; Fu, L. Blockchain-Secured Smart Manufacturing in Industry 4.0: A Survey. *IEEE Trans. Syst. Man Cybern Syst.* 2021, 51, 237–252. [CrossRef]
- [71] Eymüller, C.; Hanke, J.; Hoffmann, A.; Kugelmann, M.; Reif, W. Real-time capable OPC-UA Programs over TSN for distributed industrial control. In Proceedings of the 2020 25th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA), Vienna, Austria, 8–11 September 2020; pp. 278–285.
- [72] Sisinni, E.; Saifullah, A.; Han, S.; Jennehag, U.; Gidlund, M. Industrial Internet of Things: Challenges, Opportunities, and Directions. *IEEE Trans. Ind. Inform.* 2018, 99, 4724–4734. [CrossRef]
- [73] Beier, G.; Niehoff, S.; Xue, B. More Sustainability in Industry through Industrial Internet of Things? *Appl. Sci.* 2018, 8, 219. [CrossRef]

- [74] Chegini, H.; Naha, R.K.; Mahanti, A.; Thulasiraman, P. Process Automation in an IoT-Fog-Cloud Ecosystem: A Survey and Taxonomy. *IoT* 2021, 2, 92–118. [CrossRef]
- [75] DDS SDK. Available online: <https://raw.githubusercontent.com/eProsima/Fast-DDS/master/fastrtps.repos> (accessed on 22 June 2021).
- [76] Rodríguez-Molina, J.; Bilbao, S.; Martínez, B.; Frasher, M.; Cürüklü, B. An Optimized, Data Distribution Service-Based Solution for Reliable Data Exchange Among Autonomous Underwater Vehicles. *Sensors* 2017, 17, 1802. [CrossRef]
- [77] Kumar, M.; Singh, A.K. FDDS: An Integrated Conceptual FDDS Framework for DDS Based Middleware. In Proceedings of the 2019 International Conference on Communication and Electronics Systems (ICCES), Coimbatore, India, 17–19 July 2019; pp. 1952–1956.
- [78] Barcis, M.; Barcis, A.; Hellwagner, H. Information Distribution in Multi-Robot Systems: Utility-Based Evaluation Model. *Sensors* 2020, 20, 710. [CrossRef] [PubMed]
- [79] Thulasiraman, P.; Chen, Z.; Allen, B.; Bingham, B. Evaluation of the Robot Operating System 2 in Lossy Unmanned Networks. In Proceedings of the 2020 IEEE International Systems Conference (SysCon), Montreal, QC, Canada, 24–27 August 2020; pp. 1–8.
- [80] Fernandez, J.; Allen, B.; Thulasiraman, P.; Bingham, B. Performance Study of the Robot Operating System 2 with QoS and Cyber Security Settings. In Proceedings of the 2020 IEEE International Systems Conference (SysCon), Montreal, QC, Canada, 24–27 August 2020; pp. 1–6.
- [81] Coronado, E.; Venture, G. Towards IoT-Aided Human-Robot Interaction Using NEP and ROS: A Platform-Independent, Accessible and Distributed Approach. *Sensors* 2020, 20, 1500. [CrossRef]
- [82] Morita, R.; Matsubara, K. Dynamic Binding a Proper DDS Implementation for Optimizing Inter-Node Communication in ROS2. In Proceedings of the 2018 IEEE 24th International Conference on Embedded and Real-Time Computing Systems and Applications (RTCSA), Hakodate, Japan, 28–31 August 2018; pp. 246–247.
- [83] Cavalieri, S. A Proposal to Improve Interoperability in the Industry 4.0 Based on the Open Platform Communications Unified Architecture Standard. *Computers* 2021, 10, 70. [CrossRef]
- [84] Arestova, A.; Martin, M.; Hielscher, K.-S.J.; German, R. A Service-Oriented Real-Time Communication Scheme for AUTOSAR Adaptive Using OPC UA and Time-Sensitive Networking. *Sensors* 2021, 21, 2337. [CrossRef] [PubMed]
- [85] Ioana, A.; Burlacu, C.; Korodi, A. Approaching OPC UA Publish-Subscribe in the Context of UDP-Based Multi-Channel Communication and Image Transmission. *Sensors* 2021, 21, 1296. [CrossRef] [PubMed]
- [86] Profanter, S.; Tekat, A.; Dorofeev, K.; Rickert, M.; Knoll, A. OPC UA versus ROS, DDS, and MQTT: Performance Evaluation of Industry 4.0 Protocols. In Proceedings of the 2019 IEEE International Conference on Industrial Technology (ICIT), Melbourne, VIC, Australia, 13–15 February 2019; pp. 955–962.
- [87] Sim, W.; Song, B.; Shin, J.; Kim, T. Data Distribution Service Converter Based on the Open Platform Communications Unified Architecture Publish-Subscribe Protocol. *Electronics* 2021, 10, 2524. [CrossRef]
- [88] H. Huang, et al., "Real-Time Fault Detection for IIoT Facilities Using GBRBM-Based DNN", *IEEE Internet of Things J.* 2020, 7, 5713–5722
- [89] A. Nicolae, A. Korodi, I. Silea, "Complete Automation of an Energy Consumption Reduction Strategy from a Water Treatment and Distribution Facility, Inside an

- Industrial Internet of Things-Compliant Proactive Historian Application", *Sensors* 2021, 21, 2569.
- [90] Xu, W.; Hu, J.; et.al. Keypoint-Aware Single-Stage 3D Object Detector for Autonomous Driving. *Sensors* 2022, 22, 1451
- [91] Kalirajan, V.; Mader, R.; Kastner, S. Exploration of Real Time Behavior of Event Chains by Simulation and Measurements of "In-Vehicle Networks". *Symp. on Int. Automotive Technology*, Sep. 22, 2021
- [92] Ho, M.H.; Yen, H.C. et.al. Implementation of DDS Cloud Platform for Real-time Data Acquisition of Sensors. *2021 Int. Symp. on Intelligent Signal Processing and Communication Systems (ISPACS)*, 2021.
- [93] M. Iorio, M. Reineri, et.al., "Securing SOME/IP for In-Vehicle Service Protection," *IEEE T. on Vehicular Tech.*, 69/11, pp. 13450-13466, 2020.
- [94] M. Iorio, A. Buttiglieri, et.al., "Protecting In-Vehicle Services: Security Enabled SOME/IP Middleware" in *IEEE Vehicular Technology Magazine*, vol. 15, no. 3, pp. 77-85, Sept. 2020.
- [95] D. Zelle, T. Lauser, et.al. Analyzing and Securing SOME/IP Automotive Services with Formal and Practical Methods. In *The 16th Int. Conf. on Availability, Reliability and Security (ARES 2021)*. Association for Computing Machinery, New York, USA, Article 8, 1-20, 2021
- [96] Ma, B.; Yang, S.; Zuo, Z.; Zou, B.; Cao, Y.; Yan, X.; Zhou, S.; Li, J. An Authentication and Secure Communication Scheme for In-Vehicle Networks Based on SOME/IP. *Sensors* 2022, 22, 647
- [97] T. Gehrman and P. Duplys, "Intrusion Detection for SOME/IP: Challenges and Opportunities," *2020 23rd Euromicro Conference on Digital System Design (DSD)*, 2020, pp. 583-587
- [98] AUTOSAR. Specification of Communication Management; Release 19-03; AUTOSAR: Munich, Germany, 2019
- [99] AUTOSAR. Explanation of Adaptive Platform Design; Release 19-03; AUTOSAR: Munich, Germany, 2019.
- [100] AUTOSAR. Explanation of ara::com API; Release 17-03; AUTOSAR: Munich, Germany
- [101] R. Mader, G. Winkler, et.al., "The Car's Electronic Architecture in Motion: The Coming Transformation". In *the 42nd International Vienna Motor Symposium* 2021
- [102] A. Elbahnihy, M. Safar and M. W. El-Kharashi, "Hardware-accelerated SOME/IP-based Serialization for AUTOSAR Platforms," *15th Design & Technology of Integrated Systems in Nanoscale Era (DTIS)*, 2020
- [103] Zuo, Z.; Yang, S.; Ma, B.; Zou, B.; Cao, Y.; Li, Q.; Zhou, S.; Li, J. Design of a CANFD to SOME/IP Gateway Considering Security for In-Vehicle Networks. *Sensors* 2021, 21, 7917.
- [104] A. Bhat, S. Samii and R. R. Rajkumar, "Fault-Tolerance Support for Adaptive AUTOSAR Platforms using SOME/IP," *IEEE 26th Int. C. on Embedded and Real-Time Comp. Sys. & App. (RTCSA)*, 2020, pp. 1-6.
- [105] T. Häckel, A. Schmidt, et.al., "Strategies for Integrating Control Flows in Software-Defined In-Vehicle Networks and Their Impact on Network Security," *2020 IEEE Vehicular Networking C. (VNC)*, 2020, pp. 1-8.
- [106] <https://www.cisa.gov/uscert/ics/advisories/icsa-21-315-02> (acc. 22 Jan. 2022)
- [107] H. Kim, D. -K. Kim and A. Alaerjan, "ABAC-Based Security Model for DDS," in *IEEE Transactions on Dependable and Secure Computing*, doi: 10.1109/TDSC.2021.3085475

- [108] D. -S. Cho, S. Yun, et.al., "Autonomous Driving System Verification Framework with FMI Co-Simulation based on OMG DDS," 2020 IEEE Int. C. on Consumer Electronics (ICCE), 2020, pp. 1-6
- [109] T. Bijlsma et al., "A Distributed Safety Mechanism using Middleware and Hypervisors for Autonomous Vehicles," Design, Automation & Test in Europe Conference & Exhibition (DATE), 2020, pp. 1175-1180
- [110] S. El-Ferik, B. Almadani and S. M. Elkhider, "Formation Control of Multi Unmanned Aerial Vehicle Systems Based on DDS Middleware," in IEEE Access, vol. 8, pp. 44211-44218, 2020
- [111] S. Dehnavi, D. Goswami, et.al., "Modeling, implementation, and analysis of XRCE-DDS applications in distributed multi-processor realtime embedded systems," Design, Automation & Test in Europe C. & Exhibition (DATE), 2021, pp. 1148-1151
- [112] Jienan, D.; Xiangning, C.; Shuai, C. Overview of Application Layer Protocol of Internet of Things. 2021 IEEE 6th Int. Conf. on Computer and Communication Systems (ICCCS), 2021, pp. 922-926
- [113] Jeong, S.; Ga, T.; et.al. Behavior tree driven multi-mobile robots via data distribution service (DDS). 21st Int. C. on Control, Automation and Systems (ICCAS), 2021, pp. 1633-1638
- [114] Sudhakaran, S.; Mageshkumar, V.; et.al. Enabling QoS for Collaborative Robotics Applications with Wireless TSN. IEEE Int. C. on Comm. Workshops (ICC Workshops), 2021, pp. 1-6
- [115] Endeley, R.; Fleming, T.; et.al. A Smart Gateway Enabling OPC UA and DDS Interoperability. IEEE SmartWorld, Ubiq. Intel. & Comp., Adv. & Trusted Comp., Scalable Comp. & Comm., Cloud & Big Data Comp., Internet of People and Smart City Innovation, 2019, pp. 88-93.
- [116] A. Ioana, A. Korodi, "DDS and OPC UA Protocol Coexistence Solution in Real-Time and Industry 4.0 Context Using Non-Ideal Infrastructure". Sensors 2021, 21, 7760.
- [117] Chan, T.K.; Chin, C.S. Review of Autonomous Intelligent Vehicles for Urban Driving and Parking. Electronics 2021, 10, 1021.
- [118] Twahirwa, E.; Rwigema, J.; Datta, R. Design and Deployment of Vehicular Internet of Things for Smart City Applications. Sustainability 2022, 14, 176.
- [119] A.K. Tyagi, S.U. Aswathy, Autonomous Intelligent Vehicles (AIV): Research statements, open issues, challenges and road for future, Int. J. of Intelligent Networks, vol. 2, 2021, pp. 83-102, ISSN 2666-6030.
- [120] VSOME/IP SDK – COVESA: Available online: <https://github.com/COVESA/vsomeip> (22 Nov. 2021)
- [121] eCAL SDK. Available online: <https://github.com/continental/ecal> (22 Nov. 2021)
- [122] Yocto Project. Available online: <https://www.yoctoproject.org> (22 Nov. 2021)
- [123] <https://www.mdpi.com/1424-8220/20/16/4624>
- [124] <https://www.mdpi.com/1424-8220/20/19/5591>
- [125] <https://www.mdpi.com/1424-8220/21/4/1296>
- [126] <https://www.mdpi.com/1424-8220/21/22/7760>