MINISTERUL EDUCAȚIEI
UNIVERSITATEA TEHNICĂ
DIN CLUJ-NAPOCA

Calculatoare şi Tehnologia Informaţiei

TEZĂ DE DOCTORAT

– REZUMAT –

Contribuții la îmbunătățirea performanței în rețelele definite prin software

Student-doctorand:
Ing. Sorin Buzura

Conducător ştiinţific:
Prof. Dr. Ing. Vasile-Teodor DĂDÂRLAT

Comisia de evaluare a tezei de doctorat:

Preşedinte: Prof. Dr. Ing. **Rodica Potolea** - Universitatea Tehnică din Cluj-Napoca;
Conducător ştiinţific: Prof. Dr. Ing. **Vasile-Teodor Dădârlat** –
Universitatea Tehnică din Cluj-Napoca;
Referenţi:
- Prof. Dr. Ing. **Mariana Mocanu** - Universitatea Politehnica din Bucureşti;
- Prof. Dr. Ing. **Adrian Munteanu** - Vrije Universiteit Brussel, Belgium;
- Conf. Dr. Ing. **Piroska Haller** - Universitatea de Medicină, Farmacie, Ştiinţe şi Tehnologie „George Emil Palade", Târgu Mureş;
- Conf. Dr. Ing. **Bogdan Iancu** - Universitatea Tehnică din Cluj-Napoca.

– Cluj-Napoca –
2023

1. Introducere

Progresul tehnologic în comunicaţii a favorizat dezvoltarea rapidă a reţelelor de calculatoare la scară largă. Acest lucru este vizibil în ziua de azi sub forma internetului, care este utilizat în majoritatea activităţilor cotidiene, fie că acestea sunt de natură profesională sau personală. Dezvoltarea rapidă a internetului, datorată cererii la nivel global, s-a realizat prin instalarea multiplelor dispozitive hardware cu modele fixe (cum ar fi rutere şi comutatoare), care toate implementează aceleaşi protocoale de comunicare (fără posibilitatea de a fi uşor personalizate). Această arhitectură este cunoscută sub denumirea de arhitectura tradiţională a reţelelor, care se bazează pe omogenitatea dispozitivelor utilizate. Deşi această arhitectură este uşor scalabilă, dispozitivele nu au posibilitatea de a fi personalizate cu uşurinţă, iar prin acest lucru nu se pot deservi cerinţele curente în care diferite aplicaţii şi domenii industriale au nevoie de o gestiune personalizată. Această problemă este argumentată şi de eterogenitatea în creştere a dispozitivelor conectate în reţelele moderne. Pentru a adresa aceste probleme s-a introdus paradigma reţelelor definite prin software (terminologia în engleză: Software Defined Networking - SDN) care doreşte să definească politicile de funcţionare a unei reţele în mod configurabil şi dinamic la nivel de software.

1.1. Context

Paradigma reţelelor definite prin software reprezintă o nouă modalitate de a proiecta reţele pentru a ţine pasul cu noile descoperiri în domeniu. Această paradigmă nu adresează în mod direct problemele reţelelor tradiţionale, cum ar fi rutarea, ingineria traficului sau securitatea, dar oferă posibilitatea de a implementa soluţii inovative pentru a adresa aceste probleme [1]. Reţelele definite prin software nu depind de hardware-ul peste care sunt construite, iar complexitatea proiectării unei astfel de reţele este mutată la nivel de software. O reţea definită prin software este împărţită în planuri diferite, care sunt utilizate în funcţie de contextul de utilizare a reţelei. Împărţirea în planuri este astfel: planul de date, planul operaţional, planul de control, planul de management şi planul de aplicaţii. Planul de date gestionează pachetele transmise în reţea pe baza regulilor primite de la planul de control. Acţiunile principale pe care le pot executa dispozitivele din planul de date asupra pachetelor de reţea sunt de: transmitere, ştergere şi schimbare. Planul operaţional este teoretic delimitat, deşi în implementările practice, funcţiile acestuia fac parte din planul de date. Funcţiile principale ale acestui plan sunt de a determina starea dispozitivelor (pornite sau oprite) sau numărul de porturi disponibile de pe fiecare dispozitiv. Planul de control are rolul de a lua decizii asupra traficului din reţea trimiţând instrucţiuni dispozitivelor din planul de date referitor la acţiunile care trebuie luate cu pachetele de date. Planul de management este responsabil pentru a monitoriza şi configura modul de funcţionare a dispozitivelor de reţea, în special a dispozitivelor din planul de control. Planul de aplicaţii conţine aplicaţiile care utilizează serviciile reţelei care definesc, de asemenea, şi contextul de utilizare a reţelei [2].

1.2. Motivaţie

Provocările propuse în teza curentă se referă la îmbunătăţirea calităţii serviciilor (în engleză: Quality of Service - QoS) şi a calităţii experienţei utilizatorilor (în engleză: Quality of Experience - QoE) în reţelele definite prin software. Referitor la tematica propusă de

îmbunătăţire a serviciilor şi a performanţei în reţelele programabile, trebuie definite metricile de măsurare a performanţei în raport cu fiecare contribuţie personală. Un proces ştiinţific este caracterizat prin două trăsături: măsurabilitatea şi repetabilitatea. Astfel, pentru fiecare scenariu de testare trebuie definiţi parametrii măsuraţi, iar fiecare simulare sau experiment trebuie să aibă un comportament determinist. În reţelele tradiţionale, QoS-ul este caracterizat prin măsurarea volumului de date transmis între două noduri de reţea într-un anumit interval de timp. QoE-ul este definit ca fiind nivelul de satisfacţie al utilizatorilor faţă de serviciile reţelei [3]. Acest lucru nu este atât de uşor de pus în practică în reţelele definite prin software, unde separarea arhitecturală în diferite planuri împreună cu utilizarea mai multor protocoale (de ex. OpenFlow, LISP - Location Identifier Separation Protocol sau BGP - Border Gateway Protocol) permit diferite optimizări locale, care nu sunt uşor măsurabile la transmisia datelor între două terminale. Astfel, o provocare importantă care este adresată de această teză de doctorat este aceea de a identifica metrici de măsurare pentru fiecare simulare şi fiecare experiment în parte, care să evidenţieze îmbunătăţirile aduse.

Munca dezvoltată în această teză este semnificativă deoarece oferă un cadru de lucru care adresează diferite subiecte teoretice şi funcţionale în domeniile: ingineria traficului, dezvoltarea sistemelor colaborative, inteligenţa artificială, dar şi securitatea. Sunt propuse inovaţii tehnice în toate aceste domenii teoretice de activitate şi la toate nivelurile arhitecturale ale reţelelor programabile. Subiectele anterior menţionate înglobează majoritatea funcţiilor unei reţele, iar implementarea lor în acelaşi sistem oferă o soluţie completă care poate fi utilizată în reţelele definite prin software. În timp ce majoritatea studiilor se concentrează asupra unei singure tematici unde se folosesc aceleaşi metrici tradiţionale pentru măsurarea performanţei, această teză propune noi topologii de reţele şi repere (din engleză: „benchmarks”) noi pentru simulări şi experimente. Având topologii nou propuse care abordează subiecte interdisciplinare, metricile de măsurare trebuie, de asemenea, să fie adaptate pentru a reprezenta diferitele operaţii ale reţelei, care influenţează alte funcţionalităţi în interiorul reţelei. Un astfel de exemplu, care va fi detaliat în capitolele următoare, este în contextul securităţii unde un atac la nivel de reţea consumă din lăţimea de bandă disponibilă. Acest lucru duce la scăderea lăţimii de bandă alocată traficului de date util şi relevant, iar metrica propusă pentru a îmbunătăţi performanţa reţelei este timpul de detecţie a unui atac care are loc în reţea, care, cu cât este mai mic, cu atât reuşeşte să mărească volumul traficului de date esenţiale. Deci, nu se măsoară strict viteza de transmisie la cele două capete, ci se măsoară durata unei alte operaţii care afectează ulterior această viteză a traficului relevant.

Complexitatea acestei lucrări doctorale este dată nu doar de dificultăţile tehnice, ci şi de metodele de lucru alese, care, prin dezvoltare incrementală, au dus la rezultate pe tot parcursul muncii desfăşurate. Printre elementele tehnice care ies în evidenţă în creşterea complexităţii se numără: caracterul interdisciplinar al tezei care abordează diferite domenii cu aplicabilitate în domeniul reţelelor definite prin software; caracterul eterogen al lucrărilor efectuate care utilizează diferite tehnologii hardware, diferite tehnologii software, dar şi diferite protocoale de comunicare între dispozitivele de reţea; dezvoltarea soluţiilor care utilizează informaţii din toate nivelurile stivei de protocoale TCP/IP (Transmission Control Protocol/Internet Protocol); implementarea soluţiilor software în diferite sisteme de operare (Windows, Linux) şi în diferite limbaje de programare (C, C++, Windows Batch, Matlab, Python, Java, C#, Linux Bash, qmake); automatizarea scenariilor de testare prin programabilitate. Printre metodele de învăţare şi de lucru care au facilitat dezvoltarea soluţiilor publicate se numără metodologiile: învăţare bazată pe probleme - această metodologie presupune existenţa unei probleme care nu are soluţie şi trebuie rezolvată într-un fel; învăţare bazată pe proiect - această metodologie presupune existenţa unei teme de

proiect care trebuie dusă la finalitate; învățare bazată pe provocări - această metodologie presupune realizarea unui obiectiv despre care încă nu există cunoştinţele necesare, iar acestea trebuie întâi identificate, apoi învăţate.

1.3. Obiective

Acest subcapitol prezintă obiectivele acestei teze. Obiectivele principale adresează subiectele conceptuale şi teoretice ale tematicii abordate, iar obiectivele secundare se referă la detaliile de implementare care au fost necesare pentru a duce la finalitate obiectivele principale ale acestei teze. Fiecărui obiectiv principal îi corespunde un capitol din secţiunea de contribuţii personale. Din cele patru capitole de contribuţie personală, primele trei adresează tematici individuale, iar ultimul capitol, pe lângă contribuţia tehnică oferită, propune şi înglobarea soluţiilor obiectivelor anterioare în acelaşi mediu de lucru.

1.3.1. Obiective principale

1.3.1.1. Obiectivul principal al primului capitol de contribuţie personală este de a propune îmbunătăţirea performanţei în reţelele definite prin software îmbunătăţind caracteristicile de QoS şi QoE ale reţelei prin reducerea traficului redundant la nivel de plan de date într-o reţea wireless de senzori. (Capitol 3)

1.3.1.2. Obiectivul principal al celui de-al doilea capitol de contribuţie personală este de a propune îmbunătăţirea performanţei în reţelele definite prin software dezvoltând un plan de control distribuit şi colaborativ pentru reducerea traficului de control şi maximizând traficul de date esenţiale în reţea utilizând diferite tehnici de colaborare. (Capitol 4)

1.3.1.3. Obiectivul principal al celui de-al treilea capitol de contribuţie personală este de a propune îmbunătăţirea performanţei în reţelele definite prin software prin adresarea riscurilor de securitate care pot avea impact negativ asupra traficului esenţial de date din reţele. (Capitol 5)

1.3.1.4. Obiectivul principal al celui de-al patrulea capitol de contribuţie personală este de a propune îmbunătăţirea performanţei în reţelele definite prin software creând un cadru de lucru hibrid software şi hardware care permite testarea concomitentă a politicilor definite în obiectivele anterioare. (Capitol 6)

1.3.2. Obiective secundare

Obiectivele secundare sunt grupate pentru fiecare obiectiv principal reprezentat de un capitol separat de contribuţie personală:

1.3.2.1. Obiective secundare pentru gestionarea traficului în SDWSN (Capitol 3):

- Studiul diferitelor metode de gestionare a traficului care pot fi integrate în SDN
- Crearea unei reţele de test pe un singur dispozitiv utilizând proprietatea interfeţelor de reţea de a avea asignate mai multe adrese IP
- Identificarea şi utilizarea unui set de date reprezentativ pentru testarea implementării
- Definirea metricilor de măsurare a performanţei la nivel de plan de date în SDWSN

1.3.2.2. Obiective secundare pentru dezvoltarea planului de control distribuit şi colaborativ (Capitol 4):

- Implementarea comunicării între dispozitivele sistemelor distribuite (planul de control distribuit şi colaborativ)
- Utilizarea soluţiei OpenVSwitch şi a protocolului OpenFlow pentru a genera seturile de date în timpul testării
- Studierea şi implementarea tehnicii de fuzzing în context de QoS în SDN
- Studierea şi implementarea tehnicilor de inteligenţă artificială (AI) în SDN, mai exact tehnica de învăţare federată (în engleză: „federated learning") descentralizată
- Definirea metricilor de măsurare a performanţei la interacţiunea dintre planul de control şi planul de date din SDN

1.3.2.3. Obiective secundare pentru adresarea riscurilor de securitate în SDN (Capitol 5):

- Studierea diferitelor tipuri de atacuri care pot fi lansate la nivel de reţea
- Lansarea atacurilor la nivel de reţea utilizând Kali Linux
- Utilizarea soluţiei software OpenVSwitch bazată pe protocolul OpenFlow şi extinderea acesteia cu funcţionalitate care va rula la nivel de plan de date în SDN
- Definirea metricilor de măsurare a performanţei la nivel de plan de date în SDN

1.3.2.4. Obiective secundare pentru dezvoltarea unui cadru de lucru hibrid (Capitol 6):

- Evidenţierea caracterului eterogen şi interdisciplinar al tematicii reţelelor programabile
- Integrarea diferitelor tehnici de programare şi arhitecturi software în munca de cercetare pentru dezvoltare şi prototipizare rapidă
- Aplicarea soluţiilor în SDN la toate nivelurile arhitecturale (planurile de aplicaţie, de control şi de date)
- Implementarea cadrului de lucru folosind medii de lucru utilizate în cercetare, precum Mininet şi Mininet-WiFi
- Integrarea mai multor dispozitive hardware în reţelele create în Mininet şi Mininet-WiFi
- Dezvoltarea metodelor utilizând protocoale din toate nivelurile stivei de protocoale TCP/IP

1.4. Prezentarea cercetării din teză

Această teză de doctorat este susţinută de publicarea rezultatelor, printre care se numără 3 articole în jurnale indexate ISI (toate ca prim autor), 4 articole prezentate la conferinţe indexate ISI Proceedings (dintre care 3 ca prim autor), 6 articole prezentate la conferinţe internaţionale indexate BDI (dintre care 3 ca prim autor).

2. Studiu bibliografic raportat la obiectivele propuse

2.1. Ingineria traficului în rețelele programabile de senzori wireless

Așteptările utilizatorilor se reflectă prin creșterea numărului de funcționalități în aplicații care utilizează date oferite de o rețea wireless de senzori [4]. Comunitatea academică face eforturi continue de a oferi posibilități de a satisface cerințele utilizatorilor oferind soluții de programare dinamică a unor astfel de rețele. Scopul final al acestor eforturi este de a putea gestiona în mod cât mai autonom rețele eterogene de diferite topologii [5], [6], [7]. Paradigma utilizată pentru programabilitatea rețelelor este cea a rețelelor definite prin software. Este important de menționat în acest moment că această paradigmă nu impune un protocol specific de comunicare între planurile ei, dar protocolul OpenFlow este cel mai adesea întâlnit [8]. [9] propune un algoritm de rutare în contextul SDWSN care definește rutele de transmitere a pachetelor în funcție de nivelul de energie al fiecărui nod din rețea. [10] arată șabloanele de activitate ale unui WSN în timp ce datele sunt transmise sau rețeaua este inactivă. În literatura de specialitate există studii care folosesc diferite metode de caching de date în WSN-uri [11]. [12] utilizează o variantă a caching-ului de conținut al datelor. O altă modalitate de a îmbunătăți calitatea traficului este de a utiliza un algoritm de rutare care folosește și tehnici de echilibrare a sarcinilor, sau grupare [13]. [14] reușește să formuleze un mecanism de planificare care îmbunătățește randamentul energetic al unui SDWSN cu un procent variabil între 20-40%. [15] propune un algoritm de separare a fluxurilor de date pentru a minimiza congestia cu scopul de a îmbunătăți eficiența energetică a rețelei. [16] arată o soluție de compromis care transmite datele calculate în urma unei operații de hashing. [17] prezintă modalități de a găsi similitudini între date prin procesarea hash-urilor generate. Una dintre cele mai utilizate tehnici pentru a reduce consumul de energie în WSN-uri este gruparea senzorilor într-un cluster [18], [19]. Senzorul de coordonare este ales în funcție de nivelul de energie al senzorilor din cluster, dar diferite metode de predicție și probabilitate a nivelurilor de energie pot fi și ele alese [20]. [21] utilizează tehnica de învățare automată (din engleză: „machine learning”) pentru a decide care este senzorul de coordonare al grupului, reducând traficul transmis.

2.2. Sisteme colaborative în planul de control al rețelelor definite prin software

Îmbunătățirile în domeniul QoS-ului sunt în continuă ajustare și vor evolua pe măsură ce rețelele își vor îmbunătăți parametrii de funcționare [22]. Protocolul OpenFlow este utilizat pentru a standardiza comunicarea dintre planul de date și planul de control al unei rețele definite prin software. Prin eliminarea unor funcții de calcul de pe dispozitivele din planul de date, acestea își vor mări volumul de date pe care îl pot comuta, ducând la îmbunătățirea caracteristicilor de QoS a rețelei. Acest lucru este demonstrat în [23], care susține, de asemenea, și că unul dintre cei mai importanți parametri din regulile OpenFlow care poate fi utilizat în sistemul colaborativ este durata de valabilitate a regulii OpenFlow, adică timpul până la ștergerea regulii în momentul în care acesta expiră. [24] implementează un cadru de lucru în grupurile de noduri (cluster) ale unui plan de date cu scopul de a oferi îmbunătățiri pentru QoS. [25] validează utilizarea protocolului OpenFlow în rețelele definite prin software cu scopul final de a asigura un QoS adaptabil și dinamic. [26] validează utilizarea unui plan de control distribuit în special în contextul rețelelor wireless de senzori definite prin software.

[27] explică diferite arhitecturi de proiectare pentru planul de control în rețelele definite prin software. În domeniul rețelisticii există multe tehnici utilizate pentru a îmbunătăți QoS-ul oferit de rețelele de calculatoare: algoritmi RED (Random Early Detection), învățare automată [28], [29], diferite metode de clasificare a traficului [30], [31], [32] etc. [33] prezintă diferite moduri în care tehnici bazate pe învățare sunt integrate în sisteme distribuite, cum este contextul tezei curente de doctorat. Definită recent de Google în 2017 [34], învățarea federată a crescut de atunci, fiind utilizată în industrie pentru diverse cazuri de utilizare în domenii legate de QoS [35]. [36] prezintă diferite aplicații ale utilizării tehnicii de învățare federată, dar o caracteristică generală este faptul că procesul implică un model global centralizat care este clonat și distribuit fiecărui nod participant. [37] propune o nouă soluție în care ideea principală este de a comunica doar semnul fiecărei actualizări a gradientului, în loc de valoarea efectivă. [38] folosește comprimarea ternară rară pentru a rezolva problema datelor non IDD. Algoritmul este inspirat din tehnica de sparsificare top-k, comunicând doar o mică parte a gradientelor cele mai mari din model. [39] este un studiu care aplică tehnica de învățare federată pentru îmbunătățirea rutării pachetelor într-o rețea definită prin software. [40] utilizează învățarea federată pentru a îmbunătăți sistemele de monitorizare a traficului pe baza datelor colectate de acestea. [41] adresează problema clasificării traficului de date prin propunerea unui protocol de clasificare a traficului. Într-un comutator care implementează protocolul OpenFlow, ori de câte ori un pachet nu se potrivește cu nicio înregistrare a fluxului, transmiterea pachetului este oprită [42] până când dispozitivul de control asociat răspunde cu durata de valabilitate pentru fluxul respectiv de date. [43] abordează problema ratării tabelei și încearcă să o reducă prin implementarea pe comutatoarele OpenFlow a funcționalității de a nu șterge înregistrările fluxului când le expiră durata de valabilitate, ci mai degrabă să le marcheze ca inactive pentru o perioadă de timp în cazul în care aceeași regulă ar fi necesară în viitorul imediat. [44] abordează din nou problema reducerii numărului de ratări ale tabelei, dar în acest caz lucrarea se folosește de tamponare și cache-uire pentru o serie de pachete aparținând aceluiași flux, trimițând astfel o singură cerere packet_in pentru seria de pachete aparținând aceluiași flux de date. Pentru sistemul care utiliează tehnica de fuzzing din teza curentă de doctorat, simulările utilizează seturi de date care conțin valori similare cu cele prezentate în [45] pentru a măsura performanța sistemului.

2.3. Securitatea în rețelele definite prin software

Împreună cu oportunitățile pozitive introduse în domeniul rețelelor definite prin software au crescut și riscurile de securitate în astfel de rețele, iar în consecință, multe din eforturile din domeniul cercetării sunt îndreptate către detectarea și mitigarea atacurilor la nivel de rețea [46]. Atacurile bazate pe tehnica de ARP spoofing permit unui actor malițios să își asume identitatea unui alt dispozitiv de rețea generând cereri ARP (prin pachete de tip „gratuitous ARP"), dar care par să origineze de la dispozitivul destinat comunicării reale din rețea [47]. Este important de menționat că atacul de tip ARP spoofing este doar atacul care creează contextul de desfășurare pentru alte atacuri, cum ar fi: MitM (din engleză: „Man in the Middle"). Prin urmare, acesta este de fapt atacul care trebuie detectat și mitigat, nu atacurile care sunt dezvoltate în urmarea acestuia [48]. [49] scoate în evidență trei mari categorii de soluții împotriva atacurilor de tip ARP spoofing: soluții bazate pe șabloane de trafic; soluții bazate pe diagrame de flux; soluții bazate pe analizarea mapărilor dintre adresele IP și adresele MAC. [50] este o aplicație care analizează traficul de date și care monitorizează toate cererile ARP care circulă prin rețea, iar în momentul în care o cerere conține mapări neconcordante cu mapările deja existente în baza de date a aplicației de monitorizare, acest

cadru de date este blocat. [51] prezintă o arhitectură de rețea îmbunătățită, care previne generarea atacurilor de tip ARP MitM. [52] se bazează pe mapările stocate pe celelalte dispozitive din rețea. Mecanismul de funcționare a protocolului OpenFlow de comunicare se derulează în mai mulți pași conform [53], iar complexitatea și durata acestui întreg proces se dorește a fi evitată.Graficele de flux se bazează pe interceptarea și interpretarea pachetelelor OpenFlow din rețea. [54] generează în mod dinamic grafice de flux pentru traficul OpenFlow între planul de control și planul de date al unei rețele definite prin software. [55] nu operează în contextul rețelelor definite prin software, dar oferă o abordare pentru adresarea atacurilor de tip ARP spoofing prin analizarea traficului de rețea utilizând librăria JPCAP pentru monitorizarea traficului de date. O situație similară se regăsește în [56], unde un dispozitiv de control este extins cu o funcționalitate care notează adresele MAC malițioase într-o listă neagră (în engleză: „blacklist"), iar această listă este transferată comutatoarelor prin protocolul OpenFlow. [57] proiectează un algoritm compus din mai mulți pași pentru a detecta atacurile în timp real. [58] susține că eficiența computațională este îmbunătățită atunci când dispozitivul de control nu este implicat în luarea deciziilor. [59] descrie implementarea unor metode pentru a respinge atacurile de tip ARP spoofing. Soluția este implementată ca o extensie pentru soluția de control Open Floodlight [60]. [61] afirmă că demonstrează o soluție pentru aceeași problemă, fiind implementată în planul de date, dar care necesită soluția Ryu [62] de control pentru a defini regulile inițiale care sunt transmise comutatoarelor din planul de date. Dintr-o perspectivă tehnică, [63] propune un algoritm de detecție implementat tot în Bash, dar rulează doar pe dispozitive terminale, iar atacul poate să fie detectat doar dacă este efectuat înspre mașina curentă. [64] consideră impactul unor astfel de soluții din punctul de vedere al QoS-ului.

2.4. Cadre de lucru pentru simulări și experimente

Mininet [65] este un emulator pentru rețele cablate Ethernet, dar există și extensii ale acestuia, precum Mininet-Wifi [66] și Mininet-Optical [67], care implementează protocoalele de nivel 2 din stiva de protocoale ISO/OSI, specifice tehnologiilor de transmisie fără fir și respectiv specifice tehnologiilor de transmisie optică. Un exemplu pentru utilizarea protocolului OpenFlow este studiul din [68], unde se explică utilizarea soluției POX SDN controller [69] pentru definirea regulilor de funcționare în rețeaua definită prin software. [70] prezintă o extensie a simulatorului Cooja sub numele de WSDP (Weather & Soil Data Provider). Această extensie reprezintă de fapt un plugin pentru simulatorul Cooja [71] care este capabil să genereze în timp real date geografice pentru senzorii creați în interiorul simulatorului. [72] prezintă o abordare complexă pentru a adresa problema congestiei în rețele. [73] prezintă un cadru de lucru pentru simularea procesului de generare a rutelor în funcție de diferite categorii de trafic într-o rețea wireless de senzori 6LoWPAN. [74] descrie modul de gestionare a rețelelor wireless de senzori prin paradigma SDWSN unde consumul de energie este redus prin utilizarea planului de control pentru gestiunea rețelei. Un alt studiu care adresează problema rutării dinamice este prezentat în [75] unde se folosește un mediu real de lucru pentru a testa diferiți algoritmi de rutare cu scopul de a mări durata de viață a rețelei. [76] utilizează simulatorul OMNeT++ [77] pentru a implementa algoritmi de îmbunătățire a ratei de transfer în cazul apariției congestiei la nivel de ruter. Studiul prezentat în [78] explorează diferite tehnologii utilizate în experimente de performanță și stabilitate a rețelelor definite prin software. Un alt studiu care măsoară performanța soluțiilor de control POX și Ryu în rețele create în Mininet este [79]. Diferite topologii de rețele liniare, rețele de tip arbore sau rețele de centre de date sunt utilizate în simulări. [80] utilizează un algoritm de echilibrare a efortului (din engleză: „load balancing"), care se bazează pe monitorizarea

traficului din reţeaua de senzori utilizată. Studiul prezentat în [81] utilizează Net2Plan [82], un utilitar de gestiune a funcţiilor unei reţele, împreună cu dispozitivul de control specific pentru SDN, OpenDaylight, care este bazat pe limbajul de programare Java. [83] creează un cadru de lucru pentru a aborda problema eficienţei energetice şi a securităţii în reţele wireless de senzori din domeniul IoT. [84] utilizează Mininet-WiFi împreună cu soluţia Ryu de control pentru a măsura performanţa unui algoritm de rutare bazat pe căi multiple având scopul final de îmbunătăţire a regulilor de QoS. Pentru procesarea programatică în timp real a pachetelor care circulă prin reţea se pot utiliza soluţii software PCAP (Packet CAPture), printre care se numără următoarele: Npcap [85], SharpPcap [86], Pcap4J [87] sau jNetPcap [88]. Aceste soluţii sunt de obicei utilizate sub formă de librărie statică sau dinamică de către programul principal. [89] prezintă un sistem de monitorizare şi gestiune a QoS-ului la nivel de client utilizând librăria SharpPcap. [90] prezintă provocările oferite de un mediu IoT la scară largă. [91] prezintă o soluţie software care compară timpul de execuţie al unor operaţii efectuate la nivel de reţea. [92] adresează un subiect emergent al vehiculelor aeriene fără pilot (drone), care beneficiază de control autonom în cazul în care aceste dispozitive pierd conexiunea la reţea. Studiul din [93] prezintă cinci cadre de lucru SDN de scară largă utilizate în reţele de tip campus sau în reţele principale (din engleză: „backbone networks”). Studiul prezentat în [94] adresează problema reţelelor de scară largă unde este nevoie de mai multe dispozitive de control pentru a deservi cerinţele reţelei. [95] discută diferite probleme legate de amplasarea dispozitivelor de control SDN. [96] descrie o soluţie software de simulare a unui sistem de gestionare a datelor în context de „cloud computing. [97] analizează transmisiile asincrone între nodurile unei reţele cu scopul final de a îmbunătăţi performanţa generală a reţelei şi de a reduce consumul total de energie. [98] clasifică diferite abordări pentru gestionarea aplicaţiilor şi operaţiilor de lucru în sisteme construite la scară largă. [99] analizează problema securităţii în reţelele definite prin software. [100] propune soluţia IoTSim-SDWAN, un simulator capabil să modeleze, să simuleze şi să evalueze soluţii noi în sisteme SD-WAN şi în centre de date construite peste arhitectura SDN. [101] propune un mecanism de instalare a dispozitivelor IoT la capetele reţelei utilizând metodologia oferită de SDN. [102] demonstrează utilizarea a două arhitecturi pentru proiectarea planului de control şi anume, o arhitectură liniară şi una ierarhică. Conform [103], printre metodele tradiţionale utilizate în contextul actualizărilor software, se numără următoarele: actualizarea individuală în mod manual pentru fiecare senzor; rularea unei imagini statice de sistem de operare conectată static la nivel de nod; utilizarea sistemelor de operare dinamice. Conform [104], pentru actualizarea unui WSN trebuie considerate trei elemente strâns legate de gestionarea nivelului de energie în WSN-uri şi legate de capacitatea computaţională a tuturor dispozitivelor: mediul de execuţie la nivelul fiecărui senzor; protocolul de transmisie de date în reţea; algoritmul utilizat pentru optimizarea procesului de actualizare software. Pe lângă aceste elemente tradiţionale, reţelele moderne impun câteva elemente adiţionale, conform [105] şi anume: actualizarea neintrusivă de la distanţă (din engleză: „Over the Air”); reducerea impactului asupra performanţei WSN-ului; limitarea comunicării între dispozitive şi constrângerea comunicării doar la traficul esenţial; propagarea rapidă a actualizărilor software în reţea; implementarea abilităţii de a scala reţelele de scară largă. [106] adresează această problemă din punctul de vedere al securităţii utilizând un sistem de blockchain în mecanismul de transport a datelor. [107] adresează această problemă tot din punctul de vedere al securităţii şi implementează un sistem propriu de alocare şi distribuire a cheilor de autentificare pentru procesul de actualizare software. [108] discută diferite standarde IoT industriale, precum Zigbee [109], WirelessHART [110], ISA.100 [111] şi modul în care acestea operează la diferite niveluri din stiva de protocoale ISO/OSI.

3. Îmbunătățirea QoS-ului prin ingineria traficului

3.1. Propunerea sistemului

Obiectivul de cercetare din acest capitol este de a implementa un sistem în care o componentă de control suficient de bine informată poate gestiona o rețea wireless de senzori și poate determina dacă senzorii trebuie să transmită sau nu date. În plus, dispozitivul de control poate decide, de asemenea, dacă să transmită datele în mod proactiv pentru fiecare senzor, reducând astfel volumul traficului de date în planul de date constituit dintr-un WSN. Această propunere de sistem este prezentată în figura 3.1.

Pentru ca dispozitivul de control să realizeze funcţiile propuse, acesta implementează două componente. În primul rând, implementează o componentă de învăţare de bază care învaţă comportamentul pentru fiecare senzor, adică frecvenţa de transmitere a datelor pentru fiecare senzor, pentru a determina un model de transmisie. Şi, în al doilea rând, implementează un mecanism de transmitere a datelor bazat pe modelele de intervale de timp stabilite anterior. Este important de remarcat faptul că componenta de control nu colectează nivelurile de energie de la senzori pentru a îmbunătăţi eficienţa energetică, ci se bazează exclusiv pe frecvenţa de transmitere a datelor de la fiecare senzor.

Mediul de testare prezentat în figura 3.1 a fost construit prin implementarea la nivel de software a unei componente de control şi a unei componente generice şi reutilizabile de senzori. Componenta de control şi mai multe instanţe de senzori au fost implementate în diferite maşini virtuale pe diferite calculatoare, într-o reţea locală fără fir. Pentru testele iniţiale, este posibilă, de asemenea, configurarea mai multor dispozitive de reţea în aceeaşi maşină prin utilizarea interfeţelor de reţea virtuale, însă, pentru măsurătorile finale se impune o comunicare fără fir între dispozitivele de reţea. Acest lucru este obligatoriu pentru a apropia mediul de un WSN real.

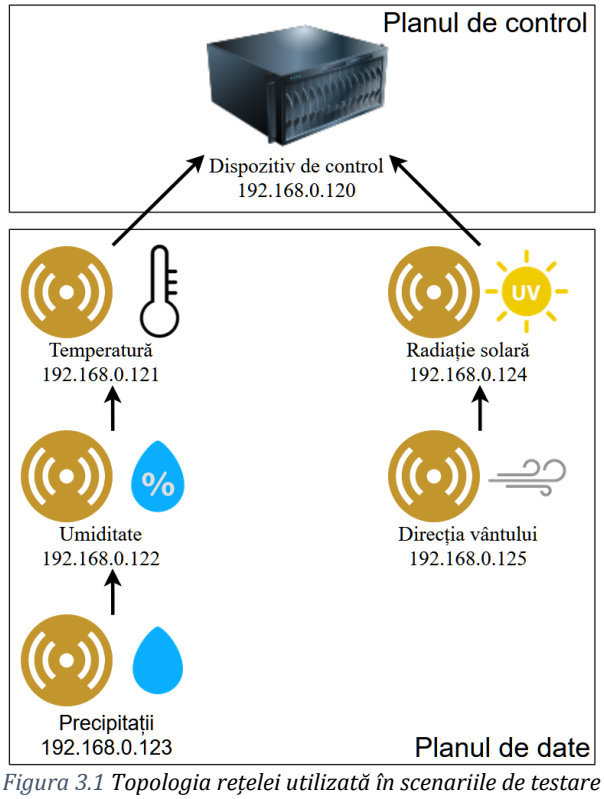

Figura 3.1 Topologia reţelei utilizată în scenariile de testare

Fiecare senzor implementează funcţionalităţi pentru a genera date de senzor şi pentru a acţiona ca un router în conformitate cu conexiunile ilustrate între dispozitivele de reţea. Spre deosebire de utilizarea unor emulatoare precum Mininet, în care toate dispozitivele de reţea sunt rulate de pe un singur calculator, această abordare are avantajul că există un interval de timp mai realist între transmiterea şi sosirea pachetelor. În plus, fiecare senzor poate fi configurat cu uşurinţă pentru a genera date aleatorii sau pentru a genera date deja stocate dintr-o sursă de referinţă la orice interval. Reţeaua suportă protocolul IP, iar adresele IP statice au fost configurate pe dispozitivele de reţea în conformitate cu topologia reţelei. Senzorii au fost implementaţi cu module pentru a genera date de la senzori, a genera pachete HELLO şi capacităţi de rutare pentru a redirecţiona pachetele către o destinaţie. În ceea ce priveşte utilizarea pachetelor HELLO, acestea nu sunt incluse în modelul matematic al propunerii actuale. Acest lucru porneşte de la ipoteza că utilizarea mesajelor HELLO este deja un mecanism implementat de standardele de transmisie de date care se regăsesc la nivelul de legătură de date din stiva de protocoale TCP/IP, verificând dacă nodurile wireless sunt sau nu active. Standardul wireless şi literatura de specialitate garantează că, în timpul perioadei fără contenţie, mai multe informaţii pot fi combinate într-un singur pachet, cum ar fi datele şi confirmările [112]. Dimensiunea minimă a unui pachet TCP este de 20 de octeţi [113] şi, dacă se adaugă la dimensiunea sarcinii utile de date. Pachetele de date ale senzorilor de pe dispozitivele de reţea au fost generate din date reale ale senzorilor recuperate din surse de date online [114] pe o perioadă de patru săptămâni. Pachetele HELLO au fost concepute pentru a fi generate la o frecvenţă mult mai mare decât pachetele de date ale senzorilor. Componentele au fost implementate în C++ cu ajutorul cadrului Qt [115] pentru facilităţile pe care le aduce la comunicarea în reţea şi la dezvoltarea de software în general.

3.2. Rezultate practice

Contextul simulat a fost reprezentat de diferenţele climatice din diferite regiuni geografice. Trebuie menţionat aici că sistemul funcţionează şi în alte contexte, dar reţeaua de test a fost aleasă cu subiect de testare a stării vremii, deoarece există multe surse de date uşor disponibile care pot fi utilizate pentru a efectua teste. Datele meteorologice istorice au fost preluate de pe site-ul web al sursei de date la care se face referire în [114] pentru diferite locaţii: Reykjavik (Islanda), Ouarzazate (Maroc), Budapesta (Ungaria). Motivul pentru care au fost alese aceste trei locaţii se datorează climatului diferit pe care acestea îl prezintă. Reykjavik are un climat ploios oceanic, Ouarzazate are un climat deşertic, în timp ce Budapesta are un climat temperat-continental. Calcularea câştigurilor pentru diferitele scenarii de durată de viaţă a reţelei arată că sistemul propus prezintă un comportament coerent în fiecare scenariu. În scenariul cu durată de viaţă mare, câştigul total (adică procentul de pachete optimizate) reprezintă 25,1% din toate pachetele. În scenariul de durată medie de viaţă, câştigul total reprezintă 25,71% din numărul total de pachete. Şi, în cele din urmă, în scenariul cu durată de viaţă mică, câştigul reprezintă 25,05% din totalul pachetelor. Acest lucru duce la concluzia că sistemul economiseşte aproximativ 25% din traficul de reţea generat. Majoritatea valorilor de date salvate care nu sunt transmise sunt valorile implicite ale senzorilor, care sunt generate atunci când senzorii nu au nimic de raportat. Mica diferenţă între scenarii depinde exclusiv de natura datelor senzorilor din setul de date selectat. Literatura de specialitate descrie un procent de 5% ca fiind critic pentru QoS-ul unui sistem [116].

3.3. Discuţii şi concluzii

Sistemul se dovedeşte a fi eficient în eliminarea traficului duplicat în nivelul WSN prin câţiva algoritmi simpli care sunt implementaţi atât la nivelul dispozitivului de control, cât şi la nivelul senzorilor. Aceştia combină concepte precum învăţarea (la nivelul dispozitivului de control), conştientizarea conţinutului (la nivelul senzorilor) şi memoria cache. Sistemul este eficient chiar şi în medii care prezintă pierderi de pachete, fapt demonstrat de scenariile de testare anterioare în care au apărut erori de transmisie. Pragul la care sistemul devine mai puţin eficient este atunci când rata de transmisie a pachetelor ajunge la 15%, ceea ce este considerabil mai mare decât pragul definit în literatura de specialitate pentru un QoS de calitate inferioară în sistemele de date fără fir. Resursele suplimentare de memorie necesare sistemului sunt neglijabile în nivelul WSN. Dispozitivul de control are nevoie de mai multă memorie pentru a stoca în memoria cache datele de la fiecare senzor din topologie; cu toate acestea, dispozitivul de control nu aparţine unui mediu cu resurse limitate şi poate fi mai uşor de actualizat pentru a susţine implementarea. În plus faţă de reducerea consumului de energie în nivelul de WSN, această propunere de sistem prezintă un beneficiu suplimentar şi anume reducerea congestiei în cozile de aşteptare ale senzorilor datorită reducerii traficului de reţea, accelerând astfel procesarea pachetelor.

Considerând scenariile de testare, care utilizează date din trei locaţii reale, se poate afirma că sistemul este mai adecvat pentru implementarea selectiva într-o locaţie decât în toate locaţiile. Acest lucru se datorează valorilor reale ale datelor, care pot fi mai stabile în anumite părţi ale planetei. Cu toate acestea, acest lucru nu este în niciun caz o regulă în toate scenariile posibile, deoarece datele de testare pot fi diferite şi în funcţie de perioada selectată. În orice caz, o concluzie definitivă este faptul că sistemul aduce optimizări în orice mediu în care este instalat. De asemenea, sistemul reuşeşte să mărească durata de viaţă a reţelei deoarece sunt transmise mai puţine pachete de către nivelul WSN, îmbunătăţeşte QoS-ul prin reducerea volumului de trafic în interiorul nivelului de WSN şi îmbunătăţeşte QoE-ul prin transmiterea proactivă a datelor stocate în memoria cache către planul de aplicaţii, în loc să aştepte ca aceleaşi date să fie primite de la senzori. Există unele limitări şi situaţii în care QoE-ul nu este atins, dar acestea nu sunt considerate a fi situaţii critice.

În concluzie, obiectivul principal definit în subcapitolul 1.3.1.1. a fost îndeplinit îmbunătăţind performanţa reţelei prin ingineria traficului, iar efectul dorit de a mări durata de viaţă a reţelei a fost realizat cu succes. Obiectivele secundare definite în subcapitol 1.3.2.1. au fost şi ele îndeplinite în totalitate prin următoarele aspecte: traficul a fost manipulat prin tehnicile de caching de date şi broadcast adaptiv de date; sistemul a fost implementat utilizând comunicare cu socket-uri între mai multe adrese IP configurare pe interfaţa locală a calculatorului folosit pentru simulări; setul de date reale a fost preluat de pe site-ul Meteoblue care conţine date istorice accesibile pentru validarea soluţiei; metrica pentru verificarea performanţei a fost aleasă ca fiind durata de viaţă a reţelei.

3.4. Diseminarea rezultatelor

Rezultatele obţinute în acest capitol au fost diseminate în următoarea publicaţie:

[117] Buzura, S.; Iancu, B.; Dadarlat, V.; Peculea, A.; Cebuc, E. Optimizations for Energy Efficiency in Software-Defined Wireless Sensor Networks. Sensors 2020, 20, 4779. https://doi.org/10.3390/s20174779 IF: 3.576 Q1

4. Dezvoltarea sistemelor colaborative pentru îmbunătățirea caracteristicilor de QoS în rețelele definite prin software

4.1. Propunerea sistemului

În această etapă au fost definite şi implementate două medii de lucru. Primul mediu de lucru s-a bazat pe simularea comunicării între dispozitivele de reţea şi simularea traficului de reţea pe protocoalele TCP, UDP, ICMP (Internet Control Message Protocol). Această abordare a fost utilă pentru validarea algoritmilor utilizaţi înainte de aplicarea lor în context de trafic de date real. Al doilea mediu de lucru s-a bazat pe generarea traficului de reţea între mai multe dispozitive care rulează pe acelaşi calculator. Acest lucru s-a realizat prin utilizarea mai multor adrese IP locale (adresele de loopback: 127.0.0.1 - 127.255.255.255 sau mai multe adrese IP configurate pe aceeaşi interfaţă de reţea) care permit proceselor din sistemul de operare să utilizeze o adresă unică de nivel 3. Această abordare este utilă pentru a genera scenarii de test realiste având la bază trafic de reţea real. Scenariile de testare realiste sunt necesare pentru a genera rezultate concludente. Au fost considerate trei topologii ale reţelelor propuse. Acestea sunt: o reţea cu plan de control nedistribuit; o reţea cu plan de control distribuit şi necolaborativ; o reţea cu plan de control distribuit şi colaborativ. Prima topologie a avut mai mult rol de control pentru a testa implementarea continuă a algoritmilor utilizaţi. De asemenea, scenariul nedistribuit şi cel distribuit necolaborativ au avut şi rolul de a valida îmbunătăţirile sistemului distribuit şi colaborativ propus la final.

4.1.2. Propunerea sistemului distribuit de fuzzing

După cum s-a menţionat anterior, obiectivul sistemului este să fie autoadaptiv şi să ajusteze în mod automat regulile de comutare a traficului în funcţie de numărul de reguli ratate. Numărul de reguli ratate înseamnă de fapt numărul de pachete pentru care comutatorul nu are o regulă de comutare activă şi este forţat să ceară instrucţiuni dispozitivului de control despre cum să acţioneze asupra pachetului respectiv. Modelul fuzzy a fost ales ca modalitatea dispozitivelor de control de a lua decizii în ceea ce priveşte modificarea regulilor de comutare a pachetelor, în timp ce metoda distribuită de utilizare a mai multor componente de control permite luarea mai multor decizii simultan, găsindu-se astfel setul optim de reguli mai rapid decât atunci când se utilizează o singură componentă de control. Comunicarea dintre dispozitivele de control se realizează cu ajutorul mesajelor de anunţare de tip broadcast (transmise în întreg planul de control) prin intermediul cărora dispozitivele de control îşi partajează configuraţiile şi statisticile pentru fiecare configuraţie, mai exact, un tabel cu fiecare valoare de timeout care trebuie utilizată pentru fiecare tip de trafic şi contorul de ratări pentru fiecare tip de trafic. Algoritmul este implementat pe dispozitivele de control ale reţelei şi presupune că mediul este un sistem SDWSN complet reactiv, în care comutatoarele din planul de date ale componentei SDN nu conţin reguli predefinite la iniţializare pentru comutarea pachetelor.

4.1.3. Propunerea sistemului de învăţare federată descentralizată

În primul rând, fiecare dispozitiv de control aşteaptă primirea unui pachet. În momentul primirii pachetului, dispozitivul de control verifică sursa pachetului dacă a fost generat de un alt dispozitiv de control sau de comutatorul conectat direct. În cazul în care sursa de proveniență a pachetului este un alt dispozitiv de control, acest pachet poate conține

doar noi date de antrenament care vor fi stocate pentru reantrenare. Un dispozitiv de control pornește o antrenare proprie în momentul în care partajează date celorlalte dispozitive de controlÎn cazul în care sursa de proveniență a pachetului este comutatorul atașat, acest pachet poate să fie de două tipuri. Primul tip de pachet este un pachet OpenFlow packet_in, iar al doilea este un pachet de răspuns la solicitarea dispozitivului de control pentru a cere datele de antrenament de la comutator. În cazul în care pachetul este de tip OpenFlow packet_in, dispozitivul de control incrementează contorul de erori pentru tipul de trafic interogat de comunicator și răspunde comutatorului cu un pachet de tip packet_flow_mod care conține valoarea de timeout (pentru a informa comutatorul cat timp poate comuta pachete cu acel protocol) care va fi stocată pe comutator până la expirarea acesteia. Valoarea de timeout cu care răspunde dispozitivul de control este calculată la fiecare trei minute prin interogarea librăriei locale de învățare automată. În cazul în care pachetul reprezintă răspunsul primit pentru cererea datelor de antrenament, aceste date urmează să fie procesate în vederea componentei de învățare automată.

4.2. Rezultate practice

Pentru obținerea rezultatelor practice au fost efectuate simulări în topologiile descrise în secțiunea 4.1. Cele trei topologii au fost testate în scenariul de simulare pentru tehnica de fuzzing, iar în cazul testării scenariului de învățare federată au fost utilizate doar topologiile cu mai multe dispozitive de control. Motivul pentru care toate cele trei topologii au fost folosite doar în scenariul de fuzzing este pentru că prima topologie a fost utilă în etapele de dezvoltare, dar nu și pentru testul final. Scopul final al acestor rezultate practice este de a putea observa comportamentul celor două tehnici utilizate în sisteme distribuite.

4.2.1. Rezultate obținute cu tehnica de fuzzing

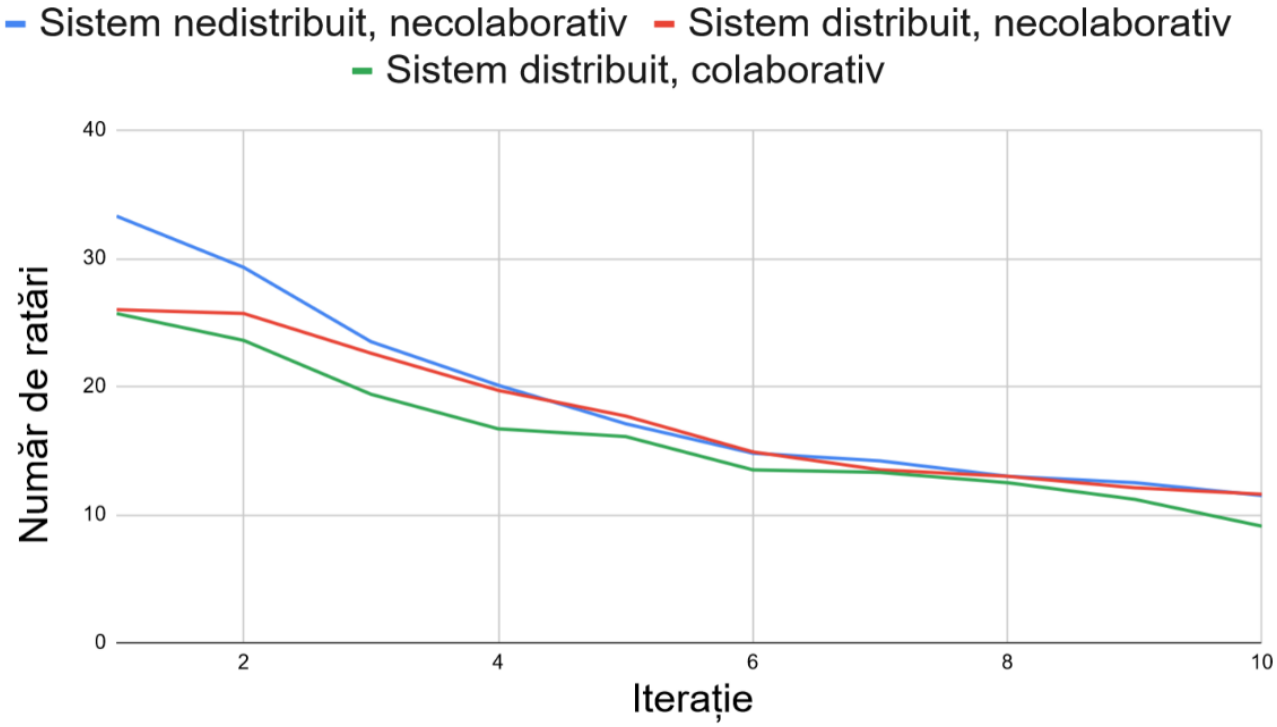

Figura 4.1 Rezultate obținute pentru utilizarea tehnicii de fuzzing

În primul scenariu de testare, topologia nedistribuită şi necolaborativă, dispozitivul de control conţine o singură tabelă de rată de erori şi o utilizează pentru fiecare comutator conectat. După fiecare iteraţie, dispozitivul de control efectuează operaţia de fuzzing pe valorile de timeout curente cu o variaţie aleatoare de maximum zece secunde. Cel de-al doilea scenariu de testare, topologia distribuită şi necolaborativa, implementează acelaşi algoritm pe dispozitivele de control, cu diferenţa că există câte un comutator conectat la fiecare dispozitiv de control, însă numărul de pachete trimise este identic. Dispozitivele de control implementează acelaşi algoritm, dar, deoarece numărul lor este multiplicat cu trei, va exista mai mult fuzzing şi, prin urmare, vor exista mai multe şanse de a genera un set de reguli mai favorabile. Al treilea scenariu de testare, topologia distribuită şi colaborativă, prezintă o extindere a scenariului precedent, în care dispozitivele de control partajează informaţii despre seturile lor actuale de reguli de comutare şi despre numărul de ratări generate de fiecare regulă pentru fiecare tip de trafic. Principala diferenţă faţă de scenariul anterior este faptul că există mai puţine şanse de a genera rezultate temporar negative, deoarece dispozitivele de control au posibilitatea de a-şi anula setul de reguli curent dacă un alt dispozitiv de control a partajat un set de reguli cu performanţă mai bună, evitându-se astfel utilizarea unor seturi de reguli insuficient de bune. Figura 4.1 prezintă rezultatele obţinute utilizând tehnica de fuzzing.

4.2.2. Rezultate obţinute cu tehnica de învăţare federată

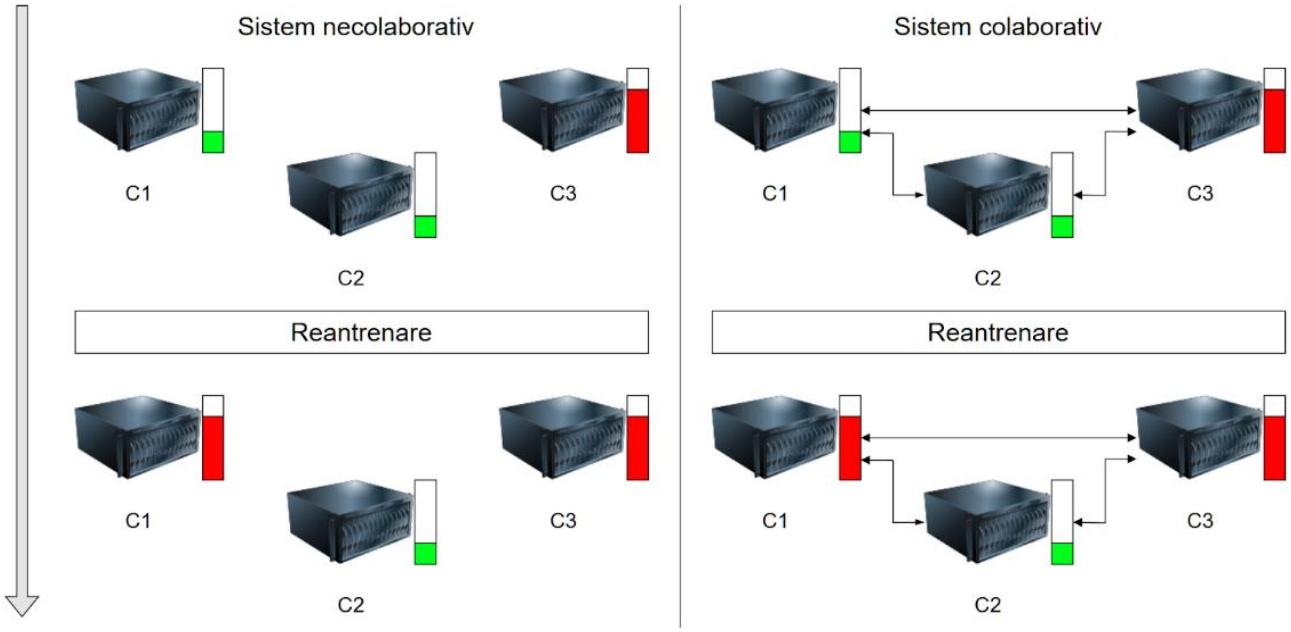

Figura 4.2 Scenariul de testare propus pentru utilizarea tehnicii de învăţare federată descentralizată

Figura 4.2 prezintă scenariul de testare care a fost considerat pentru simulările care au folosit tehnica de învăţare federată descentralizată. Au fost considerate două sisteme distribuite cu trei dispozitive de control, unul necolaborativ şi celălalt colaborativ. Figura indică prin bara de încărcare verde sau roşie nivelul de congestie întâlnit de către fiecare dispozitiv de control, unde culoarea verde înseamnă congestie redusă, iar bara roşie înseamnă congestie ridicată. Este important de explicat faptul că aceasta este o ilustraţie simplificată, congestia nu este întâlnită pe dispozitivul de control ci pe comutator, iar această congestie se reflectă pe dispozitivul de control printr-un număr mai mare de pachete OpenFlow packet_in primite de la comutatorul conectat. Deci, bara de încărcare verde sau roşie înseamnă la nivel de dispozitiv de control numărul de cereri de tip OpenFlow packet_in care sunt în creştere pe

măsură ce comutatorul ataşat primeşte trafic de reţea pentru comutare. Săgeata gri indică faptul că există o perioada de tranziţie între stările întâlnite în simulare, iar cele trei stări diferite sunt următoarele: dispozitivele de control întâmpină trafic de reţea, C1 şi C2 nu prezintă congestie, C3 prezintă congestie; are loc o reantrenare după o anumită perioada de timp; în a treia stare, după ce are loc reantrenarea pe toate dispozitivele de control, C1 începe să prezinte congestie faţă de prima stare în care nu se întamplă acest lucru.

Figura 4.3 prezintă în formă de grafic numărul de ratări la intervale de 10 minute întâlnit în simularea efectuată. Simularea întreagă a durat 2h şi 30 minute. Antrenarea a avut loc în fiecare sistem aproximativ la jumătatea timpului de rulare. Momentul antrenarii depinde de numărul de seturi de date colectate pentru antrenare, deci aceasta operaţie nu are tot timpul loc la intervale regulate. În cazul sistemului necolaborativ, dispozitivele de control C1, C2 şi C3 s-au reantrenat la minutele 1h 40m, 1h 50m şi respectiv 1h 30m. Din grafic se poate observa cum în momentul trecerii în starea a treia, unde apare congestia pe dispozitivul de control C1, are loc o creştere în numărul de ratări. Acest lucru se datorează faptului că mai mult trafic de reţea va genera inevitabil mai multe ratări prin lipsa perioadelor în care nu există trafic de reţea. Se poate observa că în cazul sistemului colaborativ există tot timpul o performanţă mai bună prin faptul că datele de antrenament sunt partajate anterior apariţiei congestiei. La intrarea în starea a treia, dispozitivul de control C1 va genera un număr aproximativ identic de ratări ca dispozitivul de control C3, deoarece dispozitivul de control C1 a reuşit deja să se antreneze pe baza experienţei avute de dispozitivul de control C3. Acest lucru nu se întâmplă în cazul sistemului necolaborativ, în acest caz dispozitivul de control C1 va avea la începutul celei de-a treia stări un număr de ratări aproximativ egal cu numărul de ratări întâlnit de pe dispozitivul de control C3 la începutul primei stări.

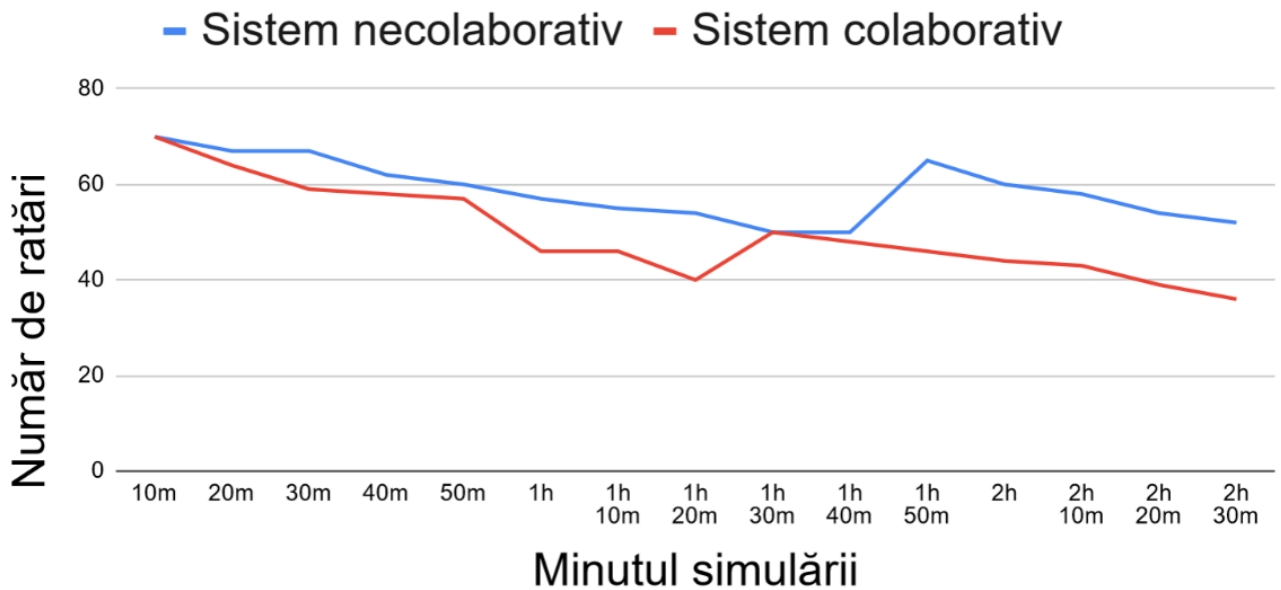

Figura 4.3 Rezultate obţinute pentru utilizarea tehnicii de învăţare federată descentralizată

4.3. Discuţii şi concluzii

Sistemul de fuzzing propus prezintă o îmbunătăţire generală a numărului mediu de ratări pe comutatoarele din planul de date atunci când se utilizează sistemul distribuit şi colaborativ. Tehnica colaborativă este, de asemenea, cea care generează cea mai rapidă scădere a numărului de ratări după finalizarea primei iteraţii. Dacă va fi explorat în continuare acest mecanism bazat pe delegare distribuită de luare a deciziilor, acest mecanism poate deveni o soluţie puternică în mediile cu conţinut dinamic (cum ar fi mediile de tip SDWSN)

pentru a reduce efortul de gestionare manuală a rețelei. Automatizarea definirii regulilor QoS îmbunătățește experiența utilizatorilor și, având în vedere tehnologiile și capacitățile actuale de inteligență artificială, acest domeniu poate fi mult îmbunătățit.

În continuarea sistemului de fuzzing a fost implementat un sistem care utilizează tehnica de învățare federată descentralizată și care rezolvă aceeași problemă a scăderii numărului de ratări pe măsură ce este întâmpinat trafic de rețea diferit de către comutatoarele din planul de date. Scenariul simulat în sistemul de învățare federată descentralizată este mai complex și utilizează trafic de date real, spre deosebire de scenariul de fuzzing, unde a fost simulat traficul de rețea. Simulările demonstrează eficacitatea abordării distribuite și colaborative. Aceasta oferă cea mai bună performanță în ceea ce privește reducerea pachetelor de comunicare OpenFlow între dispozitivele de control și comutatoare. În plus, autoadaptarea este demonstrată în fiecare scenariu, dar se dovedește a avea cea mai bună performanță în scenariile colaborative.

În concluzie, QoS-ul într-un astfel de sistem distribuit și colaborativ este îmbunătățit pentru că, dacă sunt mai puține interacțiuni cu pachete OpenFlow, atunci timpul de așteptare de a lua decizia prin protocolul OpenFlow este înlocuit cu timp petrecut de către comutator transmițând traficul de date în rețea. Astfel, viteza de date efective este crescută și scade timpul petrecut transferând pachete de control între planul de control și planul de date care, în plus, blochează transmisia datelor efective. Astfel, crește viteza de date la nivel de comutator, deci performanța rețelei este îmbunătățită. Prin urmare, a fost îndeplinit obiectivul principal propus în 1.3.1.2. De asemenea, implementarea complexă a sistemului de test a impus diferite niveluri de implementare, dar prin reușita implementării atât a sistemului de fuzzing cât și a sistemului de învățare federată descentralizată, au fost îndeplinite și obiectivele secundare propuse în 1.3.2.2.

4.4. Diseminarea rezultatelor

Rezultatele obținute în acest capitol au fost diseminate în următoarea publicație:

[118] S. Buzura, V. Dadarlat, B. Iancu, A. Peculea, E. Cebuc and R. Kovacs, "Self-adaptive Fuzzy QoS Algorithm for a Distributed Control Plane with Application in SDWSN", 2020 IEEE International Conference on Automation, Quality and Testing, Robotics (AQTR), Cluj-Napoca, Romania, 2020, pp. 1-6, doi: 10.1109/AQTR49680.2020.9129922

5. Îmbunătățirea QoS-ului prin adresarea riscurilor de securitate

5.1. Propunerea sistemului

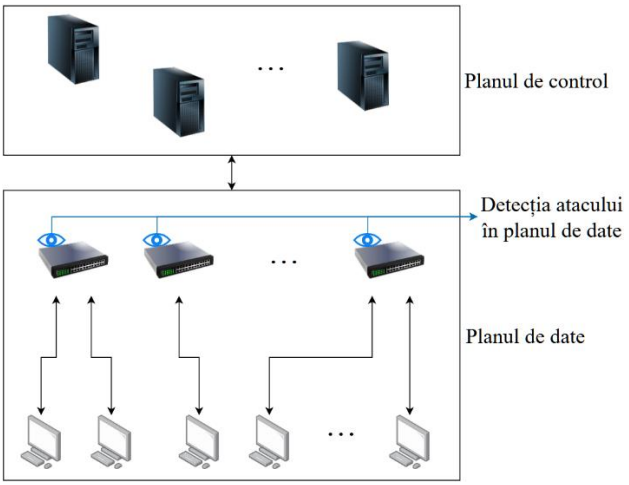

Figura 5.1 Implementarea propusă în arhitectura SDN. Este prezentată execuția soluției pe comutatoarele din planul de date

Având în vedere arhitectura SDN, este important de remarcat faptul că soluția actuală operează în planul de date și este implementată ca o extensie pentru OpenVSwitch. Figura 5.1 prezintă grafic locația implementării raport la topologia unui SDN. Acest subcapitol prezintă sistemul care a fost implementat pentru a demonstra conceptele și abordarea propusă. Figura 5.2 prezintă componentele soluției care se comportă într-un mod similar cu arhitectura de Linux pipeline, în care informațiile sunt transmise de la un proces la altul prin intermediul unui pipe sau al unui fișier. Este vorba de un flux secvențial de operațiuni, iar fiecare componentă generează date de ieșire care sunt considerate date de intrare pentru următoarea componentă. Utilizând această abordare, fiecare componentă este ușor de înlocuit sau de îmbunătățit, singura restricție fiind faptul că trebuie să se păstreze constrângerile de format pentru datele de intrare și de ieșire.

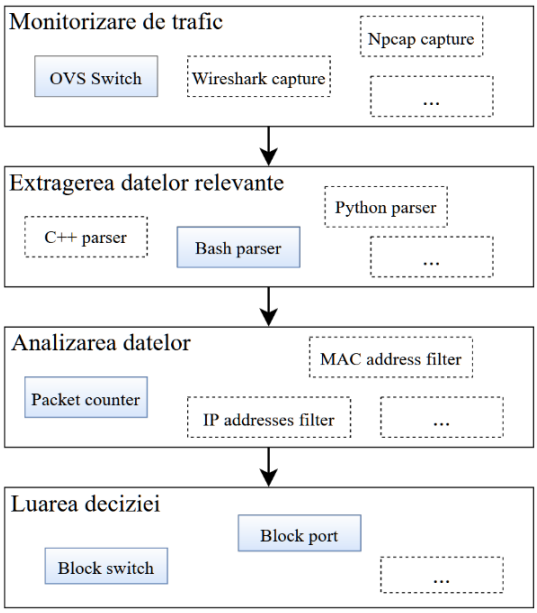

Figura 5.2 Secvența de operații necesară detectării și mitigării atacurilor. Sunt evidențiate componentele utilizate și posibile alternative de implementare

O altă caracteristică importantă a acestei soluții este faptul că secvența de operații este executată într-o buclă cu interval de timp configurabil. Parametrul intervalului de timp este stocat într-un fișier de configurare citit la inițializarea rulării. Astfel, soluția este dinamică și oferă parametrizare la rulare, și nu la compilare. Acest lucru este util în cazul în care comutatorul trebuie să fie configurat fără a recompila nicio componentă. Fișierul de configurare conține, de asemenea, un parametru pentru nivelul de severitate transmis componentei de luare a deciziilor. Nivelul de severitate poate fi utilizat diferit în funcție de nivelul de criticalitate al rețelei curente. Decizii diferite pot fi declanșate în cazul detectării unui atac în funcție de domeniul de activitate unde este utilizată rețeaua.

5.2. Rezultate practice

Un lucru important care trebuie scos în evidență este faptul că durata și timpul de mitigare reprezintă, de fapt, timpul necesar soluției pentru a executa întreaga buclă. În consecință, durata de detecție și mitigare nu poate fi mai scurtă decât execuția soluției. Prin urmare, timpul minim de detecție și mitigare al atacului este dat de timpul de execuție a soluției. După cum arată experimentele, durata variază în funcție de dimensiunea rețelei și de nivelul de congestie. În cazul scenariilor cu trafic redus, durata medie de detecție și mitigare este aproximativ identică. Cu toate acestea, odată cu creșterea numărului de dispozitive de rețea și a traficului, timpul de calcul crește exponențial, după cum se arată în figura 5.3.

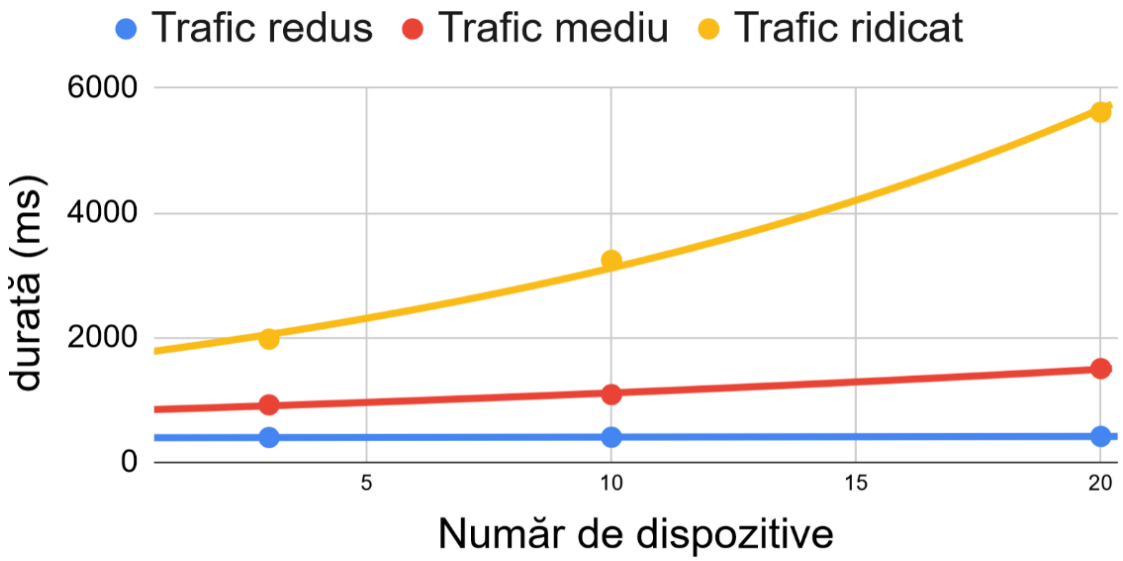

Figura 5.3 Durata medie de detecție și mitigare a atacurilor în rețele de diferite dimensiuni având diferite niveluri de congestie. Figura arată creșterea exponențială a duratei pe măsură ce crește numărul de dispozitive din rețea, dar și pe măsură ce crește nivelul de congestie

Acest comportament este cauzat de faptul că traficul scăzut determină un număr mai mic de intrări de flux OpenVSwitch, intrări care trebuie generate pe comutator, ceea ce înseamnă că următoarele etape din secvență nu primesc atât de multe date de intrare pentru procesare și nu generează un timp de procesare suplimentar semnificativ. Pe măsură ce se generează mai mult trafic, cresc și resursele necesare hardware de memorie și de calcul. Pentru scenariile cu volum de trafic mediu, o creștere a duratei de detecție și mitigare este mai vizibilă pe măsură ce topologia rețelei capătă complexitate. Comparând rețeaua cu zece dispozitive cu rețeaua cu trei dispozitive, timpul de procesare crește cu 17,8%. Comparând rețeaua cu douăzeci de dispozitive cu rețeaua cu zece dispozitive, durata de procesare crește cu 38,3%. Prin urmare, o constatare importantă care se deduce este că, pe măsură ce în rețea

sunt mai multe dispozitive, timpul de calcul de procesare creşte în mod exponenţial. În cazul scenariilor cu volum de trafic ridicat, creşterea duratei este din nou vizibilă, la fel ca în cazul precedent unde este un volum mediu de trafic. Comparând reţeaua de zece dispozitive cu reţeaua de trei dispozitive, timpul de procesare creşte cu 63,9%. Comparând reţeaua de douăzeci de dispozitive cu reţeaua de zece dispozitive, durata de procesare creşte cu 73,3%. Din nou, este vizibilă o creştere neliniară, ceea ce arată că, cu cât reţeaua este mai mare, cu atât soluţia devine mai lentă.

În ceea ce priveşte atacul MitM distribuit propus, detecţia şi mitigarea acestuia necesită resurse de calcul mai ridicate decât în cazul atacului efectuat de pe un singur dispozitiv. Detaliile prezentate arată cazul în care falsificarea ARP este împărţită doar între două dispozitive; cu toate acestea, acest lucru se poate scala la mai mulţi atacatori care acţionează coordonat simultan. Deoarece detecţia trebuie să verifice toate dispozitivele de reţea disponibile în tupluri de câte K argumente, complexitatea soluţiei creşte la O(nˆK), n la puterea lui K, unde n este numărul de dispozitive din reţea şi K este numărul de atacatori coordonaţi. Desigur, se pot efectua mai multe optimizări, cum ar fi marcarea unor dispozitive ca fiind de încredere, astfel încât acestea să nu fie inspectate, reducând astfel complexitatea, dar acest lucru iese din sfera de aplicare a studiului actual.

5.3. Discuţii şi concluzii

În cadrul etapelor ulterioare de dezvoltare, pot fi realizate mai multe experimente prin înlocuirea componentelor în arhitectura de tip plugin şi prin studierea noilor componente care au performanţă mărită. Performanţa sistemului poate fi extinsă prin monitorizarea doar a unor anumite dispozitive pe baza adreselor acestora, reducând astfel timpul de calcul, deoarece vor fi inspectate mai puţine componente. O altă idee de dezvoltare ulterioară privind nivelul fizic al reţelelor este de a efectua teste în reţele programabile eterogene în care dispozitivele WiFi se conectează la un comutator prin intermediul unor puncte de acces fără fir. Soluţia propusă suportă deja această configuraţie, deoarece reţelele WiFi sunt procesate în mod identic cu reţelele Ethernet cablate (au acelaşi antet de nivel 2), dar ar putea fi realizat un studiu mai aprofundat în ceea ce priveşte latenţa conversiilor de semnal din nivelul fizic (de la unde radio la semnale electrice), iar un punct de acces ar permite conectarea mai multor dispozitive într-o anumită locaţie. Acest studiu ar oferi mai multe informaţii cu privire la posibilităţile de implementare a acestei soluţii într-un mediu hardware real. O altă idee de dezvoltare ulterioară este de a extinde acest concept la un sistem distribuit. Se poate crea un mediu de colaborare în care diferite switch-uri pot împărtăşi date privind adresele MAC cunoscute ale atacatorilor, astfel încât un atacator să poată fi găsit mai uşor în diferite locaţii ale reţelei. Un mediu colaborativ poate integra tehnici de inteligenţă artificială, cum ar fi învăţarea federată, pentru a detecta atacurile pe baza unor şabloane de trafic mai complexe. Traficul de reţea între dispozitivele Mininet a fost generat cu ajutorul utilitarului iPerf [119].

Dezvoltarea ulterioară a soluţiei actuale poate depăşi ideile de complexitate locală prin software şi prin încercări de a creşte performanţa unui singur dispozitiv de reţea. Prin urmare, studiile viitoare pot fi incluse în planurile arhitecturii SDN prin utilizarea virtualizării funcţiilor de reţea prin intermediul hipervizoarelor de reţea. Studiul prezentat în [120] prezintă CoVisor, un hipervizor de reţea care permite coexistenţa mai multor dispozitive de control pe acelaşi dispozitiv şi permite partajarea celor mai bune caracteristici ale acestora pentru optimizarea funcţionalităţii reţelei. Libera este, de asemenea, un hipervizor de reţea care are capacitatea de a crea o infrastructură de reţea dedicată pentru gestionarea unui trafic de reţea specific [121]. Astfel de hipervizoare precum CoVisor şi Libera ar putea fi utilizate împreună pentru a crea un ecosistem virtualizat care să permită aplicarea unor politici şi

caracteristici diferite în diferite infrastructuri de rețea. Un caz de utilizare mai concret legat de soluția prezentată în acest articol privind securitatea, ar fi implementarea unor strategii diferite de mitigare a atacurilor în diferite infrastructuri de rețea. Testarea și măsurarea simultană a performanței pot avea loc în infrastructurile de rețea create separat, ceea ce duce la o metodă mai rapidă de a decide care este cea mai bună metodă de tratare a unui atac specific. Cu toate acestea, un dezavantaj pe care hipervizoarele îl pot avea în contextul securității este că funcționalitatea este mutată în nivelurile intermediare ale rețelei, dar atacurile sunt întotdeauna identificate de preferință cât mai aproape de sursa atacului. Astfel, pentru cazurile de utilizare critice, este mai bine să se opereze la marginea rețelei și nu în nivelurile superioare SDN după ce informațiile au fost deja transmise prin multe dispozitive intermediare. Într-un astfel de scenariu, utilizarea unui hipervizor de rețea este discutabilă și trebuie efectuate analize comparative și teste adecvate înainte de a implementa o soluție de infrastructură de rețea.

Un alt concept care poate fi utilizat împreună cu ecosistemele virtualizate și programabile este cel al circuitelor integrate programabile pentru aplicații specifice (din engleză: “application-specific integrated circuit”), în care funcțiile de la nivelul software al aplicației sunt mutate la nivelul de hardware al circuitului. Astfel, în teorie, cazurile de utilizare care sunt implementate în nivelurile superioare ale sistemului de operare pot fi mutate în hardware. Utilizată împreună cu avantajele oferite de un hipervizor de rețea, programabilitatea dinamică poate fi configurată în diferite locații de rețea.

Rețelele definite prin software au avantajul de a fi programabile și permit dezvoltarea numeroaselor cazuri particulare de utilizare a rețelelor. În acest sens, soluția actuală s-a concentrat pe crearea unei arhitecturi software extensibile care poate detecta și mitiga atacurile bazate pe tehnica de ARP spoofing în rețele Ethernet create în Mininet. Soluția este construită pe arhitectura de Linux pipeline și pe un sistem de plugin-uri, în care este obligatorie o secvență de operații. Secvența de operații conține diferite module pentru achiziția de date, extragerea datelor relevante, analiza datelor și luarea deciziilor. Fiecare dintre componente este înlocuibilă cu ajutorul unui fișier de configurare și, în funcție de cazurile de utilizare a rețelei și de nivelurile de severitate dorite, pot fi întreprinse diferite acțiuni la detectarea atacului. Faptul că componentele sunt înlocuibile permite ca sistemul să fie ușor de extins și configurabil.

În concluzie, obiectivul principal definit în 1.3.1.3. a fost îndeplinit îmbunătățind performanța rețelei prin detecția și mitigarea riscurilor de securitate în rețelele definite prin software, iar efectul dorit de a mări viteza traficului esențial de date a fost realizat cu succes. Obiectivele secundare definite în 1.3.2.3. au fost și ele de asemenea îndeplinite în totalitate prin următoarele aspecte: traficul malițios al atacurilor bazate pe tehnica de ARP spoofing a fost analizat în detaliu în urma efectuării atacurilor MitM bazate pe tehnica de ARP spoofing utilizând utilitarul Ettercap din KaliLinux; soluția OpenVSwitch a fost extinsă cu posibilitatea de detecție și mitigare a atacurilor bazate pe tehnica de ARP spoofing; metrica pentru verificarea performanței a fost aleasă ca fiind viteza de transfer în planul de date a datelor esențiale.

5.4. Diseminarea rezultatelor

Rezultatele obținute în acest capitol au fost diseminate în următoarea publicație:

[122] S. Buzura, M. Lehene, B. Iancu, V. Dadarlat "An Extendable Software Architecture for Mitigating ARP Spoofing-Based Attacks in SDN Data Plane Layer", Electronics 2022, 11, 1965. doi: https://doi.org/10.3390/electronics11131965 IF 2.69 Q3

6. Dezvoltarea unui cadru de lucru pentru simularea rețelelor definite prin software

6.1. Cadrul de lucru în contextul obiectivelor propuse

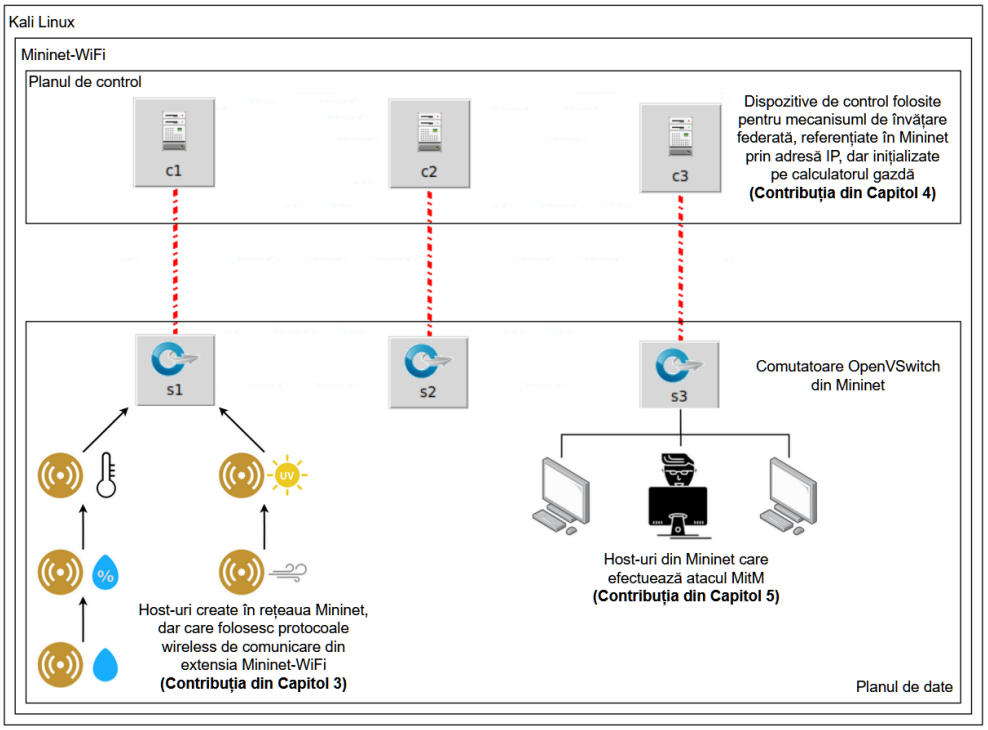

Figura 6.1 Cadrul de lucru utilizat în contextul obiectivelor propuse în această teză de doctorat

Pentru finalitatea acestei teze de doctorat se impune și înglobarea în aceeași rețea a soluțiilor obiectivelor principale propuse pentru finalitatea tezei. Astfel, figura 6.1 prezintă modul în care se pot îmbina toate aceste soluții într-o singură topologie de testare. Se identifică următoarele componente: un sistem de operare Kali Linux care este necesar pentru efectuarea atacului MitM; emulatorul Mininet-WiFi instalat în cadrul mașinii virtuale care folosește sistemul de operare Kali Linux, este important de amintit faptul că Mininet-WiFi conține atât extensiile de protocoale wireless cât și funcționalitatea de bază oferită de Mininet; dispozitivele de control sunt adăugate în rețeaua Mininet și sunt referențiate prin adresă IP, acest lucru înseamnă că nu se impune rularea soluțiilor de control pe același sistem de operare unde rulează instanța de Mininet, iar în acest caz acestea vor rula de pe calculatorul gazdă care folosește un sistem de operare Windows de unde se poate folosi librăria de învățare automată exportată din Matlab; nodurile wireless care compun o rețea de senzori sunt atașate la comutatorul s1, acestea pot fi coordonate să trimită sau să nu trimită date în funcție de soluția de optimizare bazată pe cache-uirea datelor. Cu toate acestea spuse, cadrul de lucru este validat pentru a fi folosit în contexte de simulare cu tehnologii eterogene și interdisciplinare și oferă o soluție completă în posibilitățile de studiu al comportamentului rețelelor programabile adresând atât tematici de QoS cât și de securitate, oferind numeroase posibilități de a studia contribuțiile propuse de îmbunătățire a performanței rețelelor definite prin software.

6.2. Mediul de testare

În acest moment este important de reiterat faptul că obiectivul principal al acestui capitol este de a oferi detalii arhitecturale și de implementare privind modul de construire a unui mediu hibrid având componente de simulare software și hardware. Activitatea experimentală se limitează la proiectarea pentru portabilitate, deoarece acest context de utilizare este singurul experiment care poate furniza valori de măsurare precise și reutilizabile pentru literatura de specialitate. Acest lucru se datorează faptului că pachetele OpenFlow care sunt utilizate în acest experiment sunt transmise pe tipuri de rețele LAN și WAN care sunt utilizate în mod obișnuit în rețelele moderne. Proiectele pentru scalabilitate, calcule paralele și livrare de actualizări software depind în mare măsură de resursele hardware ale mașinii Linux care rulează Mininet și Mininet-WiFi, care sunt subiective din perspectiva lucrării actuale. Pentru a simula ideile de proiectare de mai sus, componentele necesare sunt prezentate în figura 6.2. Configurația constă dintr-un PC cu sistem de operare Linux, pe care au fost instalate Mininet și Mininet-WiFi, componente hardware, care au fost atașate direct la PC prin interfețe USB (Ubertooth One și senzorul Libelium Waspmote) și o interfață Ethernet (placa Toradex i.MX6). Avantajul abordării propuse mai sus pentru duplicarea traficului și citirea simultană a mai multor date de la mai multe dispozitive hardware este posibilitatea de a o extinde la simulări la scară largă, în care se poate utiliza doar un singur senzor de fiecare tip. Astfel, costul simulării și timpul de implementare scad drastic.

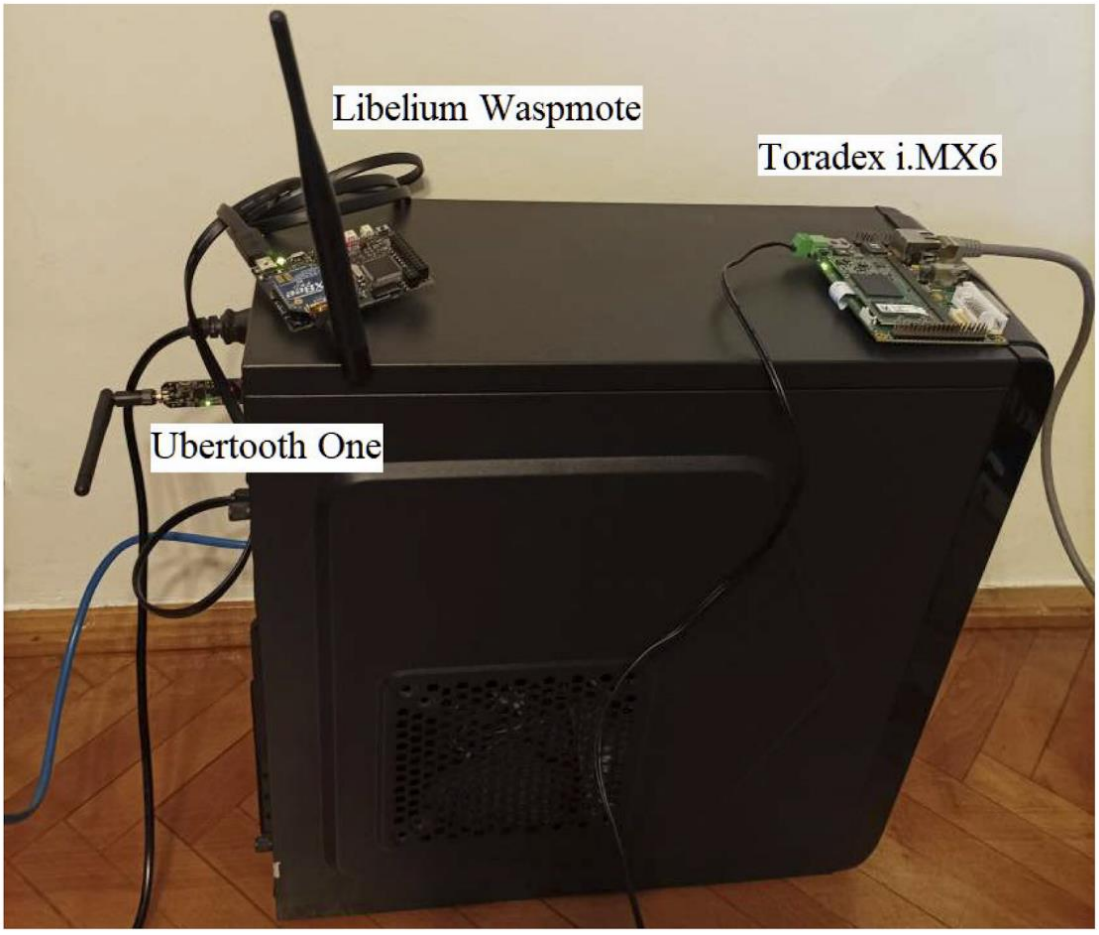

Figura 6.2 Componentele utilizate pentru crearea cadrului de lucru hibrid software și hardware pentru simulările propuse

6.3. Discuţii şi concluzii

Rezultatul muncii de cercetare dezvoltată în acest capitol a creat un cadru de lucru hibrid de simulare software şi hardware pentru testarea reţelelor. Au fost utilizate patru idei principale de proiectare şi anume: proiectarea pentru scalabilitate, proiectarea pentru calcule paralele, proiectarea pentru portabilitate şi proiectarea pentru actualizarea software-ului pe nodurile unei reţele wireless de senzori. Aceste idei de proiectare oferă posibilitatea testării sistemului în diferite cazuri de utilizare. Contextele de utilizare prezentate în lucrare sunt: colectarea de date de la senzori dintr-un WSN, securitatea reţelei, testarea comparativă a diferitelor locaţii de plasare a dispozitivelor de control SDN şi transmiterea de actualizări software într-un mediu SDN. Sistemul de testare SDN care a fost implementat conţine contribuţii în planul de date şi în planul de control. Scopul final al utilizării paradigmei SDN este de a aduce beneficii planului aplicaţiilor, care sunt visibile în urma îmbunătăţirilor efectuate în planurile inferioare SDN. Pentru validarea abordării de simulare hibridă software hardware propusă, au fost efectuate măsurători în scenariul conceput pentru portabilitate. Scopul a fost acela de a studia efectul poziţionării unui dispozitiv de control SDN în diferite locaţii din LAN-ul care conţine planul de date, dar şi într-o locaţie diferită din internet. Rezultatele au arătat că impactul transferului de date între comutatorul planului de date şi dispozitivul de control SDN este neglijabil atunci când dispozitivul de control SDN se află în aceeaşi reţea LAN (indiferent de numărul de comutatoare intermediare), dar transferul de date este semnificativ mai lent atunci când dispozitivul de control SDN se află într-o reţea de nivel 3 diferită.

Noutatea principală propusă pentru a introduce citirea în timp real a datelor de la senzori este limitată la un număr de senzori egal cu numărul de interfeţe disponibile pe calculator, astfel se deduce care este numărul exact de dispozitive hardware care se pot conecta la Mininet. Printre aceste interfeţe se numără următoarele: USB, seriale, Bluetooth etc. Numărul de interfeţe disponibile poate fi mărit într-o oarecare măsură prin adăugarea de adaptoare externe sau hub-uri USB care permit conectarea mai multor dispozitive. Cu toate acestea, acest număr rămâne limitat, iar citirea datelor de pe interfeţe multiple consumă şi alte resurse hardware (puterea de procesare a procesorului şi memoria disponibilă) care sunt necesare pentru calculele experimentale. În ceea ce priveşte măsurătorile efectuate asupra pachetelor OpenFlow, acestea au fost efectuate numai asupra pachetului de tip OpenFlow packet_in, care are o dimensiune cuprinsă între aproximativ 600 şi 1400 de octeţi. Pachetul OpenFlow packet_in conţine cadrul Ethernet pentru care se face solicitarea, la care se adaugă în plus câţiva octeţi suplimentari care sunt specifici protocolului OpenFlow. Cu o dimensiune redusă a pachetului, timpii de transmisie vor avea valori aproximativ similare. Pentru a crea o evaluare completă, întreg setul OpenFlow de tipuri de pachete ar putea fi măsurat pentru transmiterea de date. Interpretând rezultatele prezentate, variaţia duratei de transfer este mică, iar acest lucru se datorează faptului că OpenFlow este un protocol de nivel aplicaţie care rulează peste TCP. Stabilirea unei conexiuni ocupă o parte considerabilă din acest timp de transmisie cu handshake-ul TCP în trei paşi.

Având în vedere gama largă de aplicabilitate a domeniului SDN şi mediile de scară largă, crearea unui mediu de testare real este o sarcină complexă şi costisitoare. Pentru a adresa această limitare, a fost propus un cadru de lucru hibrid de simulare software şi hardware. Prin integrarea datelor simulate cu date de senzori în timp real (de la dispozitive hardware) într-un emulator Mininet, au fost generate şi validate simulări realiste în diferite contexte reale de utilizare. În plus, noi seturi de date pot fi generate cu uşurinţă pentru scenarii specifice prin salvarea datelor capturate de la senzori în timp real. Seturile de date şi

scenariile nou-generate pot fi uşor de reprodus în alte proiecte de cercetare, contribuind astfel la literatura de specialitate.

În concluzie, obiectivul principal definit în 1.3.1.4. a fost îndeplinit îmbunătăţind performanţa reţelelor definite prin software, permiţând crearea de scenarii de simulare complete care pot testa concomitent diferite funcţionalităţi utilizând şi dispozitive hardware. Obiectivele secundare definite în 1.3.2.4. au fost şi ele de asemenea îndeplinite în totalitate prin următoarele aspecte: eterogenitatea reţelelor definite prin software este scoasă în evidenţă prin cadrul de lucru hibrid care îmbină diferite soluţii software cu diferite dispozitive hardware; au fost utilizate diferite concepte de programabilitate în diferite limbaje de programare pentru crearea cadrului de lucru; reţelele programabile au fost adresate din toate planurile arhitecturale; diferite reţele create în Mininet şi Mininet-WiFi au fost extinse cu date citite în timp real de la dispozitive hardware; în dezvoltarea cadrului de lucru au fost utilizate protocoale din toate nivelurile stivei de protocoale TCP/IP, oferind astfel o soluţie completă pentru cadrul de lucru creat.

6.4. Diseminarea rezultatelor

Munca de cercetare efectuată în acest capitol a fost diseminată în următoarele publicaţii:

[123] S. Buzura, A. Peculea, B. Iancu, E. Cebuc, V. Dadarlat, R. Kovacs "A Hybrid Software and Hardware SDN Simulation Testbed", Sensors 2023, 23, 490. https://doi.org/10.3390/s23010490 IF: 3.847 Q2

[124] S. Buzura, V. Lazar, B. Iancu, A. Peculea and V. Dadarlat, "Using Software-Defined Networking Technology for Delivering Software Updates to Wireless Sensor Networks," 2021 20th RoEduNet Conference: Networking in Education and Research (RoEduNet), Iasi, Romania, 2021, pp. 1-6, doi: 10.1109/RoEduNet54112.2021.9637720.

[125] S. Buzura, V. Dadarlat, A. Peculea, H. Bertrand and R. Chevalier, "Simulation Framework for 6LoWPAN Networks Using Mininet-WiFi," 2022 IEEE International Conference on Automation, Quality and Testing, Robotics (AQTR), Cluj-Napoca, Romania, 2022, pp. 1-5, doi: 10.1109/AQTR55203.2022.9802017.

[126] S. Buzura, A. Suciu, E. Cebuc, B. Iancu and V. Dadarlat, "Development Framework for Simulating Routing Behavior in Software-Defined Wireless Networks," 2022 21st RoEduNet Conference: Networking in Education and Research (RoEduNet), Sovata, Romania, 2022, pp. 1-6, doi: 10.1109/RoEduNet57163.2022.9921101.

7. Concluzii

7.1. Concluzii generale

Această teză de doctorat oferă un cadru de lucru în domeniile de ingineria traficului, dezvoltarea sistemelor colaborative, inteligența artificială și securitatea, cu aplicabilitate în rețelele definite prin software. Contribuțiile personale au fost implementate la toate nivelurile rețelelor programabile, cu scopul final de a îmbunătăți QoS-ul oferit de un astfel de sistem. Cadrul de lucru rezultat oferă o soluție completă pentru efectuarea studiilor de cercetare în rețelele definite prin software utilizând domeniile menționate anterior.

Domeniul rețelelor de calculatoare este un domeniu interdisciplinar unde diferite noțiuni legate de programabilitate, administrare și securitate adesea se suprapun în cercetare. Din această cauză, se impune o înțelegere teoretică foarte bună a acestor domenii, dar este necesară și experiență practică relevantă pentru a putea dezvolta soluții și pentru a efectua studii de cercetare în domeniul rețelelor de calculatoare. Din acest motiv, complexitatea domeniului și, în consecință a tezei curente, nu are doar o caracteristică tehnică de implementare locală a funcționalităților propuse ci are și o caracteristică complexă de integrare a unor concepte interdisciplinare cu rol de a găsi compatibilitatea între acestea.

În ceea ce privește modul în care au fost atinse rezultatele prezentate în aceasta teză de doctorat, este important de menționat faptul că unele dintre temele de cercetare s-au derulat pe o perioadă îndelungată, chiar pe mai mulți ani, până la reușirea diseminării rezultatelor. Procesele utilizate au fost incrementale cu diferiți colaboratori (coautorii articolelor publicate), iar modul de lucru abordat a fost publicat în [127] la o conferință cu tematică educativă. Continuând această tematică educativă de pe parcursul stagiului doctoral, îndrumătorul de laborator pentru materia de Rețele de calculatoare (în limba engleză de predare) a fost și el actualizat cu subiecte studiate și experimentate practic pe parcursul acestei teze de doctorat. Astfel, pentru îndrumătorul referențiat în [128] au fost adăugate două lucrări de laborator complet noi despre programare cu socket-uri și atacuri de tip ARP spoofing cu rol de a-i pregăti pe studenți atât cu noțiuni generale tehnice, dar și pentru diverse subiecte cercetare.

7.2. Originalitatea și contribuțiile inovative ale tezei

Capitolul 3 cu tematica de inginerie a traficului a propus o contribuție pentru îmbunătățirea performanței în rețele de tip SDWSN. În acest context, îmbunătățirea performanței se referă la îmbunătățirea eficienței energetice și la creșterea duratei de viață a rețelei. Eficiența energetică într-un SDWSN a fost îmbunătățită prin utilizarea tehnicilor de analizare a conținutului datelor transmise și a transmiterii adaptive de tip broadcast a datelor. Eficiența energetică este îmbunătățită prin reducerea numărului de pachete transmise în planul de date al rețelei definite prin software. Tehnica de analizare a conținutului a fost implementată la nivelul senzorilor programabili individuali în planul de date, unde resursele hardware sunt limitate pe dispozitivele de rețea. În soluția finală, transmiterea adaptivă a datelor este mutată în planul de control și înlocuiește transmiterea datelor din planul de date. Sistemul implementat mărește durata de viață a rețelei, deoarece numărul de pachete din nivelul WSN este redus, astfel îmbunătățind QoS-ul prin reducerea volumului de trafic și evitarea congestiei. QoE-ul este îmbunătățit prin transmiterea proactivă a datelor din planul

de control către planul de aplicaţii. Astfel, obiectivul principal propus în 1.3.1.1. a fost îndeplinit. De asemenea, obiectivele secundare propuse în 1.3.2.1. au fost şi ele îndeplinite.

Capitolul 4 cu tematica de implementare a unui plan de control distribuit şi colaborativ a propus o contribuţie pentru definirea mai eficientă a regulilor de comutare OpenFlow. În acest context, îmbunătăţirea performanţei se referă la reducerea traficului de control din reţea, astfel favorizând traficul esenţial de date şi, în consecinţă, mărind viteza de transfer a datelor esenţiale. Problema propusă a fost rezolvată în două moduri, utilizând tehnica de fuzzing şi utilizând tehnica de învăţare federată descentralizată. Ambele tehnici au un rezultat pozitiv reducând traficul de control din reţea, dar se poate argumenta un avantaj în favoarea tehnicii de învăţare federată şi anume, faptul că regulile generate se mulează mai bine pe durata reală a traficului, iar aceste reguli nu raman active în perioade fără trafic. Astfel, obiectivul principal propus în 1.3.1.2. a fost îndeplinit. De asemenea, obiectivele secundare propuse în 1.3.2.2. au fost şi ele îndeplinite.

Capitolul 5 cu tematica de securitate propune contribuţii pentru îmbunătăţirea QoS-ului într-un SDN prin eliminarea traficului maliţios din reţea, astfel mărind traficul de date relevante transmis. Performanţa este îmbunătăţită astfel prin mărirea vitezei de transmisie a datelor esenţiale. A fost implementată o soluţie pentru detectarea şi mitigarea atacurilor de tip ARP spoofing în planul de date al unui SDN. Arhitectura software utilizată este capabilă să acomodeze diferite medii de funcţionare (bazate pe OpenVSwitch, npcap, etc.) şi poate integra mai multe tehnologii software pentru procesarea datelor. Implementarea soluţiei este făcută strict în planul de date al unui SDN fără interacţiune cu planul de control. Soluţia implementată a fost evaluată în reţele de dimensiuni diferite şi cu diferite niveluri de congestie, iar evaluarea prezintă şi impactul pe care soluţia îl are asupra volumului de date transmis în reţea în scenarii cu şi fără atacul MitM în derulare. În final este conceptualizat şi un nou atac distribuit de tip ARP spoofing şi este făcută o analiză comportamentală a acţiunilor atacatorului. Este oferită şi o soluţie pentru acest nou tip de atac, oferind astfel o noutate în raport cu stadiul actual al cunoaşterii. Astfel, obiectivul principal propus în 1.3.1.3. a fost îndeplinit. De asemenea, obiectivele secundare propuse în 1.3.2.3. au fost şi ele îndeplinite.

Capitolul 6 dedicat creării unui cadru de lucru care înglobează toate contribuţiile anterioare propune contribuţii software de implementare pentru simularea reţelelor definite prin software. Sunt utilizate diverse limbaje de programare şi tehnici de programare cu socket-uri nativ în sisteme de operare Windows şi Linux, dar şi în reţele create cu ajutorul emulatoarelor Mininet şi Mininet-WiFi. Sunt explorate diferite contexte reale de utilizare a reţelelor, cum ar fi: simularea reţelelor de scară largă; actualizarea software-ului de pe senzori instalaţi în producţie fără a avea acces cablat la aceştia; simularea tehnicilor de rutare în SDWSN; simulări pentru testarea diferiţilor algoritmi în paralel pe aceleaşi date; scenarii de testare cu dispozitive portabile. Sunt efectuate teste în reţele cu diferite constrângeri, precum: reţele cablate IPv4; reţele necablate IPv6/6LowPan; reţele care utilizează servicii oferite de sistemul distribuit IPFS; reţele care extind protocolul OpenFlow şi soluţii software care folosesc acest protocol, soluţia Pox. În final este propus şi un mediu hibrid de simulare care utilizează reţele construite în Mininet şi Mininet-WiFi, care sunt extinse cu date primite în timp real de la senzori conectaţi fizic la calculatorul gazdă prin diferite interfeţe (USB, serial). Toate aceste contribuţii dovedesc flexibilitatea şi eterogenitatea paradigmei SDN. Mediile de simulare software dezvoltate pot fi extinse în viitor cu uşurinţă pentru a continua studiile în domeniul reţelelor programabile. De asemenea, aceste contribuţii oferă şi mai multe detalii legate de implementare, care adesea lipsesc în lucrările ştiinţifice din literatura de specialitate, dar sunt necesare pentru reproducerea studiilor şi validarea acestora. Astfel,

obiectivul principal propus în 1.3.1.4. a fost îndeplinit. De asemenea, obiectivele secundare propuse în 1.3.2.4. au fost şi ele îndeplinite.

În concluzie, prin cercetările efectuate, obiectivele propuse au fost îndeplinite în totalitate.

7.3. Dezvoltări ulterioare

Reţelele definite prin software au câştigat popularitate în special în reţele de centre de date sau reţele de campus. Un context mai nou este utilizarea paradigmei de SDN în reţele WAN, creând topologia de tip SD-WAN. O posibilă dezvoltare ulterioară este crearea unui cadru de lucru bazat pe emulatorul Mininet-Optical care, pe lângă caracteristicile unui SDN, include şi un simulator al proceselor fizice de transmitere a datelor prin fibra optică. Diferite studii se pot face cu diferiţi parametrii atât la nivelurile superioare din stiva de protocoale TCP/IP (nivelurile de reţea, de internet şi aplicaţie), cât şi la nivelul inferior de acces la mediu. Astfel, se pot crea diferite topologii de testare care pot deservi diferite studii de cercetare în diferite domenii de aplicabilitate.

O altă posibilă dezvoltare ulterioară este şi în domeniul securităţii, şi anume verificarea susceptibilităţii dispozitivelor dintr-un SDN la diferite atacuri, cum ar fi atacul Nethammer [129], care este de fapt atacul Rowhammer cauzat de un volum mare de trafic de date primit din reţea. Acest atac presupune în primul rând o soluţie software care citeşte datele primite de pe interfaţa de reţea în aceeaşi zonă de memorie (identificată de acelaşi pointer). Iar în al doilea rând, dispozitivul hardware pe care rulează această soluţie software trebuie să utilizeze un anumit tip de memorie RAM susceptibilă la acest atac (o memorie care nu izolează complet interferenţele electromagnetice între tranzistori). Studiul ulterior ar presupune investigarea diferitelor soluţii SDN şi efectuarea unor teste de stres şi penetrare pe acestea pentru a cauza apariţia atacului. Soluţii SDN care oferă acces la codul sursă pot fi mai profund analizate şi înţelese pentru modul în care acestea citesc datele primite din reţea. În cazul găsirii unor astfel de vulnerabilităţi se pot propune soluţii pentru remedierea acestora, lucru care va duce la îmbunătăţirea calităţii oferite de soluţiile existente. Pentru o mai bună înţelegere a acestor vulnerabilităţi se pot studia şi aplicaţiile practice din [130] şi [131] care detaliază procesele de la nivel de hardware.

REFERINŢE

[1] L. Peterson, C. Cascone, B. O'Connor, T. Vachuska, and B. Davie, Software-Defined Networks: A Systems Approach. Accesibil la: https://sdn.systemsapproach.org/index.html (accesat la data de 01.03.2020)

[2] Standardul rețelelor definite prin software specificat în RFC 7426 „Software-Defined Networking (SDN): Layers and Architecture Terminology„. Accesibil la: https://datatracker.ietf.org/doc/html/rfc7426 (accesat la data de 01.03.2020)

[3] Nam, H.; Kim, K.-H.; Kim, J.Y.; Schulzrinne, H. Towards QoE-aware video streaming using SDN. In Proceedings of the 2014 IEEE Global Communications Conference, Austin, TX, USA, 8–12 December 2014

[4] Ezdiani, S.; Acharyya, I.S.; Sivakumar, S.; Al-Anbuky, A. Wireless Sensor Network Softwarization: Towards WSN Adaptive QoS. IEEE Internet Things J. 2017, 4, 1517–1527

[5] Benzekki, K.; El Fergougui, A.; Elbelrhiti Elalaoui, A. Software-defined networking (SDN): A survey. Secur. Commun. Netw. 2016, 9, 5803–5833

[6] Misra, S.; Bera, S.; Achuthananda, M.P.; Pal, S.K.; Obaidat, M.S. Situation-Aware Protocol Switching in Software-Defined Wireless Sensor Network Systems. IEEE Syst. J. 2017, 12, 1–8

[7] Kadel, R.; Ahmed, K.; Nepal, A. Adaptive error control code implementation framework for software defined wireless sensor network (SDWSN). In Proceedings of the 2017 27th International Telecommunication Networks and Applications Conference (ITNAC), Melbourne, VIC, Australia, 22–24 November 2017

[8] McKeown, N.; Anderson, T.; Balakrishnan, H.; Parulkar, G.; Peterson, L.; Rexford, J.; Shenker, S.; Turner, J. OpenFlow: Enabling innovation in campus networks. Comput. Commun. Rev. 2008, 38, 69–74

[9] Xiang, W.; Wang, N.; Zhou, Y. An Energy-Efficient Routing Algorithm for Software-Defined Wireless Sensor Networks. IEEE Sens. J. 2016, 16, 7393–7400

[10] Alam, M.M.; Berder, O.; Menard, D.; Sentieys, O. Traffic-aware adaptive wake-up-interval for preamble sampling MAC protocols of WSN. In Proceedings of the Third International Workshop on Cross Layer Design, Rennes, France, 30 November–1 December 2011

[11] Liu, D.; Chen, B.; Yang, C.; Molisch, A.F. Caching at the wireless edge: Design aspects, challenges, and future directions. IEEE Commun. Mag. 2016, 54, 22–28

[12] Lei, F.; Cai, J.; Dai, Q.; Zhao, H. Deep Learning Based Proactive Caching for Effective WSN-Enabled Vision Applications. Complexity 2019, 1–12

[13] Hu, W.; Qiu, Z.; Wang, H.; Yan, L. A Real-time scheduling algorithm for on-demand wireless XML data broadcasting. J. Netw. Comput. Appl. 2016, 68, 151–163

[14] Wei, Y.; Ma, X.; Yang, N.; Chen, Y. Energy-Saving Traffic Scheduling in Hybrid Software Defined Wireless Rechargeable Sensor Networks. Sensors 2017, 17, 2126

[15] Li, G.; Guo, S.; Yang, Y.; Yang, Y. Traffic Load Minimization in Software Defined Wireless Sensor Networks. IEEE Internet Things J. 2018, 5, 1370–1378

[16] Modares, H.; Salleh, R.; Moravejosharieh, A. Overview of Security Issues in Wireless Sensor Networks. In Proceedings of the Third International Conference on Computational Intelligence, Modelling & Simulation, Langkawi, Malaysia, 20–22 September 2011

[17] Sadowski, C.; Levin, G. Simhash: Hash-Based Similarity Detection; Technical Report, Google; University of California: Santa Cruz, CA, USA, 2007

[18] Wang, Z.; Zhang, M.; Gao, X.; Wang, W.; Li, X. A clustering WSN routing protocol based on node energy and multipath. Clust. Comput. 2017, 22

[19] Abbasi, A.A.; Younis, M. A survey on clustering algorithms for wireless sensor networks. Comput. Commun. 2007, 30, 2826–2841

[20] Xu, L.; Collier, R.; O'Hare, G. A Survey of Clustering Techniques in WSNs and Consideration of the Challenges of Applying Such to 5G IoT Scenarios. IEEE Internet Things J. 2017, 4, 1229–1249

[21] Townsend, L. Wireless Sensor Network Clustering with Machine Learning. Ph.D. Thesis, Nova Southeastern University, Fort Lauderdale, FL, USA, 2018

[22] M. Karakus, and A. Durresi, "Quality of Service (QoS) in Software Defined Networking (SDN): A survey", Journal of Network and Computer Applications. 80, 2016, 10.1016/j.jnca.2016.12.019

[23] Adam Zarek, OpenFlow Timeouts Demystified, University of Toronto, Toronto, Ontario, Canada, 2012

[24] V. M. Vishnu, P. Manjunath, "SeC-SDWSN: Secure cluster-based SDWSN environment for QoS guaranteed routing in three-tier architecture", International Journal of Communication Systems, e4020, 2019, doi:10.1002/dac.4020

[25] Manzanares-Lopez, P.; Malgosa-Sanahuja, J.; Muñoz-Gea, J.P. A Software-Defined Networking Framework to Provide Dynamic QoS Management in IEEE 802.11 Networks. Sensors 2018, 18, 2247

[26] B. T. de Oliveira and C. B. Margi, "Distributed control plane architecture for software-defined Wireless Sensor Networks," 2016 IEEE International Symposium on Consumer Electronics (ISCE), Sao Paulo, 2016, pp. 85-86.doi: 10.1109/ISCE.2016.7797384

[27] Oktian, Y. E., Lee, S., Lee, H., & Lam, J. (2017). Distributed SDN controller system: A survey on design choice. Computer Networks, 121, 100–111. doi:10.1016/j.comnet.2017.04.038

[28] S. T. V. Pasca, S. S. P. Kodali and K. Kataoka, “AMPS: Application aware multipath flow routing using machine learning in SDN”, Twenty-third National Conference on Communications (NCC), Chennai, 2017, pp. 1-6. doi: 10.1109/NCC.2017.8077095

[29] J. Liu and Q. Xu, "Machine Learning in Software Defined Network," 2019 IEEE 3rd Information Technology, Networking, Electronic and Automation Control Conference (ITNEC), Chengdu, China, 2019, pp. 1114-1120. doi: 10.1109/ITNEC.2019.8729331

[30] R. Thupae, B. Isong, N. Gasela and A. M. Abu-Mahfouz, "Machine Learning Techniques for Traffic Identification and Classification in SDWSN: A Survey," IECON 2018 - 44th Annual Conference of the IEEE Industrial Electronics Society, Washington, DC, 2018, pp. 4645-4650. doi: 10.1109/IECON.2018.8591178

[31] B. Letswamotse, K. Modieginyane and R. Malekian, “SDN Based QoS Provision in WSN Technologies”, arXiv 2017, arXiv:abs/1702.08164

[32] P. Wang, S. Lin and M. Luo, "cs," 2016 IEEE International Conference on Services Computing (SCC), San Francisco, CA, 2016, pp. 760-765. doi: 10.1109/SCC.2016.133

[33] J. Verbraeken, M. Wolting, J. Katzy, J. Kloppenburg, T. Verbelen, and J. S. Rellermeyer, “A survey on distributed machine learning,” arXiv preprint arXiv:1912.09789, 2019

[34] H. B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, “Communication-efficient learning of deep networks from decentralized data” 2016, arXiv:1602.05629. Accesibil la: https://arxiv.org/abs/1602.05629 (accesat la data de 01.06.2020)

[35] Zhang Y., Zhang X., Li X. (2021) Towards Efficient and Privacy-Preserving Service QoS Prediction with Federated Learning. In: Gao H., Wang X., Iqbal M., Yin Y., Yin J., Gu N. (eds) Collaborative Computing: Networking, Applications and Worksharing. CollaborateCom 2020. Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering, vol 350. Springer, Cham

[36] Q. Yang, Y. Liu, T. Chen, and Y. Tong. Federated machine learning: Concept and applications. ACM TIST, 10(2):12:1–12:19, 2019

[37] Jeremy Bernstein, Yu-Xiang Wang, Kamyar Azizzadenesheli, and Animashree Anandkumar. signSGD: Compressed optimisation for non-convex problems. In International Conference on Machine Learning, pages 560–569, 2018

[38] F. Sattler, S. Wiedemann, K. -R. Müller and W. Samek, "Robust and Communication-Efficient Federated Learning From Non-i.i.d. Data," in IEEE Transactions on Neural Networks and Learning Systems, vol. 31, no. 9, pp. 3400-3413, Sept. 2020, doi: 10.1109/TNNLS.2019.2944481

[39] A. Sacco, F. Esposito and G. Marchetto, "A Federated Learning Approach to Routing in Challenged SDN-Enabled Edge Networks," 2020 6th IEEE Conference on Network

Softwarization (NetSoft), Ghent, Belgium, 2020, pp. 150-154, doi:10.1109/NetSoft48620.2020.9165506

[40] Xu, C.; Mao, Y. An Improved Traffic Congestion Monitoring System Based on Federated Learning. Information 2020, 11, 365

[41] Mun, H.; Lee, Y. Internet Traffic Classification with Federated Learning. Electronics 2021, 10, 27

[42] Protocolul OpenFlow. Specificații disponibile la: https://opennetworking.org/wp-content/uploads/2014/10/openflow-switch-v1.5.1.pdf (accesat la data de 01.06.2020)

[43] E. Kim, S. Lee, Y. Choi, M. Shin and H. Kim, "A flow entry management scheme for reducing controller overhead," 16th International Conference on Advanced Communication Technology, Pyeongchang, 2014, pp. 754-757. doi: 10.1109/ICACT.2014.6779063

[44] A. V. Atli, M. S. Uluderya, S. Tatlicioglu, B. Gorkemli and A. M. Balci, "Protecting SDN controller with per-flow buffering inside OpenFlow switches," 2017 IEEE International Black Sea Conference on Communications and Networking (BlackSeaCom), Istanbul, 2017, pp. 1-5, doi: 10.1109/BlackSeaCom.2017.8277662

[45] H. I. Kobo, G. P. Hancke and A. M. Abu-Mahfouz, "Towards a distributed control system for software defined Wireless Sensor Networks," IECON 2017 - 43rd Annual Conference of the IEEE Industrial Electronics Society, Beijing, 2017, pp. 6125-6130. Doi: 10.1109/IECON.2017.8217064

[46] Huang, S.; Griffioen, J. Network Hypervisors: Managing the Emerging SDN Chaos. In Proceedings of the 2013 22nd International Conference on Computer Communication and Networks (ICCCN), Nassau, Bahamas, 30 July–2 August 2013; IEEE: Piscataway, NJ, USA, 2013; pp. 1–7

[47] Chica, J.C.; Imbachi, J.C.; Botero Vega, J.F. Security in SDN: A Comprehensive Survey. J. Netw. Comput. Appl. 2020, 159, 102595

[48] Mallik, A.; Ahsan, A.; Shahadat, M.M.Z.; Tsou, J.-C. Man-In-The-Middle-Attack: Understanding in Simple Words. Int. J. Data Netw. Sci. 2019, 3, 77–92

[49] Shah, Z.; Cosgrove, S. Mitigating ARP Cache Poisoning Attack in Software-Defined Networking (SDN): A Survey. Electronics 2019, 8, 1095

[50] Nehra, A.; Tripathi, M.; Gaur, M.S. FICUR: Employing SDN Programmability to Secure ARP. In Proceedings of the 2017 IEEE 7th Annual Computing and Communication Workshop and Conference (CCWC), Las Vegas, NV, USA, 9–11 January 2017; pp. 1–8

[51] Furukawa, M.; Kuroda, K.; Ogawa, T.; Miyaho, N. Highly secure communication service architecture using SDN switch. In Proceedings of the 2015 10th Asia-Pacific Symposium on Information and Telecommunication Technologies (APSITT), Colombo, Sri Lanka, 22–25 September 2015; IEEE: Piscataway, NJ, USA, 2015; pp. 1–3

[52] Solomon, N. Mitigating Layer 2 Attacks: Re-Thinking the Division of Labor. Master's Thesis, School of Computer Science, The Interdisciplinary Center, Reichman University, Herzliya, Israel, 2015

[53] Al-Somaidai, M.B. Survey of Software Components to Emulate OpenFlow Protocol as an SDN Implementation. Am. J. Softw. Eng. Appl. 2014, 3, 74

[54] Dhawan, M.; Poddar, R.; Mahajan, K.; Mann, V. SPHINX: Detecting Security Attacks in Software-Defined Networks. In Proceedings of the Network and Distributed System Security Symposium, San Diego, CA, USA, 8–11 February 2015; IEEE: Piscataway, NJ, USA, 2015; pp. 8–11

[55] Hareesh, I.; Prasanna, S.; Vijayalakshmi, M.; Shalinie, S.M. Anomaly detection system based on analysis of packet header and payload histograms. In Proceedings of the 2011 International Conference on Recent Trends in Information Technology (ICRTIT), Chennai, India, 3–5 June 2011; IEEE: Piscataway, NJ, USA, 2011; pp. 412–416

[56] Girdler, T.; Vassilakis, V.G. Implementing an Intrusion Detection and Prevention System Using Software-Defined Networking: Defending against ARP Spoofing Attacks and Blacklisted MAC Addresses. Comput. Electr. Eng. 2021, 90, 106990

[57] Munther, M.N.; Hashim, F.; Abdul Latiff, N.A.; Alezabi, K.A.; Liew, J.T. Scalable and Secure SDN Based Ethernet Architecture by Suppressing Broadcast Traffic. Egypt. Inform. J. 2021, 23, 113–126

[58] Shaghaghi, A.; Kaafar, M.A.; Buyya, R.; Jha, S. Software-defined network (SDN) data plane security: Issues, solutions, and future directions. In Handbook of Computer Networks and Cyber Security; Gupta, B., Perez, G., Agrawal, D., Gupta, D., Eds.; Springer: Cham, Switzerland, 2020; pp. 341–387.

[59] Rangisetti, A.K.; Dwivedi, R.; Singh, P. Denial of ARP Spoofing in SDN and NFV Enabled Cloud-Fog-Edge Platforms. Clust. Comput. 2021, 24, 3147–3172

[60] Soluția software Open Floodlight ca dispozitiv de control în rețelele definite prin software. Accesibil la: https://floodlight.atlassian.net/wiki/spaces/floodlightcontroller/overview (accesat la data de 17.05.2021)

[61] Lin, T.-Y.; Wu, J.-P.; Hung, P.-H.; Shao, C.-H.; Wang, Y.-T.; Cai, Y.-Z.; Tsai, M.-H. Mitigating SYN flooding Attack and ARP Spoofing in SDN Data Plane. In Proceedings of the 2020 21st Asia-Pacific Network Operations and Management Symposium (APNOMS), Daegu, Korea, 22–25 September 2020; pp. 114–119

[62] Soluția software Ryu ca dispozitiv de control în rețelele definite prin software. Accesibil la: https://ryu-sdn.org/ (accesat la data de 17.05.2021)

[63] Amin, A.A.M.M.; Mahamud, M.S. An Alternative Approach of Mitigating ARP Based Man-in-the-Middle Attack Using Client Site Bash Script. In Proceedings of the 2019 6th

International Conference on Electrical and Electronics Engineering (ICEEE), Istanbul, Turkey, 16–17 April 2019; IEEE: Piscataway, NJ, USA, 2019; pp. 112–115

[64] Yang, G.; Shin, C.; Yoo, Y.; Yoo, C. A Case for SDN-based Network Virtualization. In Proceedings of the 29th International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems (MASCOTS), Houston, TX, USA, 3–5 November 2021; IEEE: Piscataway, NJ, USA, 2021; pp. 1–8

[65] Emulatorul de rețele Mininet. Accesibil la: http://mininet.org (accesat la data de 01.03.2020)

[66] Fontes, R.R.; Afzal, S.; Brito, S.H.B.; Santos, M.A.S.; Rothenberg, C.E. Mininet-WiFi: Emulating Software-defined Wireless Networks. In Proceedings of the 2015 11th International Conference on Network and Service Management (CNSM), Barcelona, Spain, 9–13 November 2015; pp. 384–389

[67] Emulatorul de rețele Mininet-Optical. Accesibil la: https://mininet-optical.org/ (accesat la data de 11.06.2022)

[68] R. Nagarathna and S. M. Shalinie, "SLAMHHA: A supervised learning approach to mitigate host location hijacking attack on SDN controllers," 2017 Fourth International Conference on Signal Processing, Communication and Networking (ICSCN), 2017, pp. 1-7, doi: 10.1109/ICSCN.2017.8085680

[69] Soluția software POX ca dispozitiv de control în rețelele definite prin software. Accesibil la: https://noxrepo.github.io/pox-doc/html (accesat la data de 17.05.2021)

[70] A. Bumb, B. Iancu and E. Cebuc, "Extending Cooja simulator with real weather and soil data," 2018 17th RoEduNet Conference: Networking in Education and Research (RoEduNet), 2018, pp. 1-5, doi: 10.1109/ROEDUNET.2018.8514130

[71] Simulatorul de rețele Cooja. Accesibil la: https://github.com/contiki-os/contiki/wiki/An-Introduction-to-Cooja (accesat la data de 17.05.2021)

[72] H. A. A. Al-Kashoash, M. Hafeez and A. H. Kemp, "Congestion Control for 6LoWPAN Networks: A Game Theoretic Framework," in IEEE Internet of Things Journal, vol. 4, no. 3, pp. 760-771, June 2017, doi: 10.1109/JIOT.2017.2666269

[73] A. HAKA, R. VASILEV, V. ALEKSIEVA and H. VALCHANOV, "Simulation Framework for Building of 6LoWPAN Network," 2019 16th Conference on Electrical Machines, Drives and Power Systems (ELMA), 2019, pp. 1-5, doi: 10.1109/ELMA.2019.8771633

[74] F. F. J. Lasso, K. Clarke and A. Nirmalathas, "A software-defined networking framework for IoT based on 6LoWPAN," 2018 Wireless Telecommunications Symposium (WTS), 2018, pp. 1-7, doi: 10.1109/WTS.2018.8363938

[75] Younus, M.U.; Islam, S.u.; Kim, S.W. Proposition and Real-Time Implementation of an Energy-Aware Routing Protocol for a Soft-ware Defined Wireless Sensor Network. Sensors 2019, 19, 2739

[76] S. Buzura, V. Dadarlat, A. Peculea, B. Iancu and E. Cebuc, "Simulations framework for network congestion avoidance algorithms using the OMNeT++ IDE," 2013 11th RoEduNet International Conference, 2013, pp. 1-8, doi: 10.1109/RoEduNet.2013.6511758

[77] Simulatorul de rețele OMNeT++. Accesibil la: https://omnetpp.org/ (accesat la data de 17.05.2021)

[78] Lunagariya, D.; Goswami, B. A Comparative Performance Analysis of Stellar SDN Controllers using Emulators. In Proceedings of the 2021 International Conference on Advances in Electrical, Computing, Communication and Sustainable Technologies (ICAECT), Bhilai, India, 19–20 February 2021; pp. 1–9

[79] Kazi, N.M.; Suralkar, S.R.; Bhadade, U.S. Evaluating the Performance of POX and RYU SDN Controllers Using Mininet. In Communications in Computer and Information Science Book Series (CCIS, Volume 1483); Venugopal, K.R., Shenoy, P.D., Buyya, R., Patnaik, L.M., Iyengar, S.S., Eds.; Springer: Cham, Switzerland, 2021; Volume 1483, pp. 181–191

[80] Cui, X.; Huang, X.; Ma, Y.; Meng, Q. A Load Balancing Routing Mechanism Based on SDWSN in Smart City. Electronics 2019, 8, 273

[81] J. Izquierdo-Zaragoza, A. Fernandez-Gambin, J. Pedreno-Manresa and P. Pavon-Marino, ”Leveraging Net2Plan planning tool for network orchestration in OpenDaylight,” 2014 International Conference on Smart Communications in Network Technologies (SaCoNeT), 2014, pp. 1-6, doi: 10.1109/SaCoNeT.2014.6867770

[82] Utilitarul Net2Plan. Accesibil la: http://www.net2plan.com/ (accesat la data de 06.03.2022)

[83] Haseeb, K.; Ud Din, I.; Almogren, A.; Islam, N. An Energy Efficient and Secure IoT-Based WSN Framework: An Application to Smart Agriculture. Sensors 2020, 20, 2081

[84] M. S. Hossen, M. H. Rahman, M. Al-Mustanjid, M. A. Shakil Nobin and M. A. Habib, ”Enhancing Quality of Service in SDN based on Multi-path Routing Optimization with DFS,” 2019 International Conference on Sustainable Technologies for Industry 4.0 (STI), 2019, pp. 1-5, doi: 10.1109/STI47673.2019.9068057

[85] Librăria software Npcap pentru limbajele de programare C şi C++. Accesibilă la: https://npcap.com/ (accesat la data de 06.03.2022)

[86] Librăria software SharpPcap pentru limbajul de programare C#. Accesibilă la: https://www.nuget.org/packages/SharpPcap (accesat la data de 06.03.2022)

[87] Librăria software Pcap4J pentru limbajul de programare Java. Accesibilă la: https://www.pcap4j.org/ (accesat la data de 06.03.2022)

[88] Librăria software jNetPcap pentru limbajul de programare Java. Accesibilă la: https://www.jnetpcap.com/ (accesat la data de 06.03.2022)

[89] S. B. Bose and S. S. Sujatha, ”Fuzzy Logic based QoS Management and Monitoring System for Cloud Computing”, 2020 3rd International Conference on Intelligent Sustainable Systems (ICISS), 2020, pp. 920-924, doi: 10.1109/ICISS49785.2020.9315874

[90] Bosmans, S.; Mercelis, S.; Denil, J.; Hellinckx, P. Testing IoT Systems Using a Hybrid Simulation Based Testing Approach. Computing 2018, 101, 857–872

[91] Ulbricht, M.; Acevedo, J.; Krdoyan, S.; Fitzek, F.H.P. Emulation vs. Reality: Hardware/Software Co-Design in Emulated and Real Time-sensitive Networks. In Proceedings of the European Wireless 2021 26th European Wireless Conference, Verona, Italy, 10–12 November 2021; pp. 1–7

[92] Tang, J.; Chen, X.; Zhu, X.; Zhu, F. Dynamic Reallocation Model of Multiple Unmanned Aerial Vehicle Tasks in Emergent Adjustment Scenarios. IEEE Trans. Aerosp. Electron. Syst. 2022, 1–43

[93] Huang, T.; Yu, F.R.; Zhang, C.; Liu, J.; Zhang, J.; Liu, Y. A Survey on Large-Scale Software Defined Networking (SDN) Testbeds: Approaches and Challenges. IEEE Commun. Surv. Tutor. 2017, 19, 891–917

[94] Zhao, Z.; Wu, B. Scalable SDN Architecture with Distributed Placement of Controllers for WAN. Concurr. Comput. Pract. Exp. 2017, 29, e4030

[95] Das, T.; Sridharan, V.; Gurusamy, M. A Survey on Controller Placement in SDN. IEEE Commun. Surv. Tutor. 2020, 22, 472–503

[96] Alwasel, K.; Calheiros, R.N.; Garg, S.; Buyya, R.; Pathan, M.; Georgakopoulos, D.; Ranjan, R. BigDataSDNSim: A Simulator for Analyzing Big Data Applications in Software-Defined Cloud Data Centers. Softw. Pract. Exp. 2020, 51, 893–920

[97] Alomari, A.; Subramaniam, S.K.; Samian, N.; Latip, R.; Zukarnain, Z. Resource Management in SDN-Based Cloud and SDN-Based Fog Computing: Taxonomy Study. Symmetry 2021, 13, 734

[98] Gonzalez, N.M.; Carvalho, T.C.M.D.B.; Miers, C.C. Cloud Resource Management: Towards Efficient Execution of Large-Scale Scientific Applications and Workflows on Complex Infrastructures. J. Cloud Comput. 2017, 6, 1–20

[99] Hegazy, A.; El-Aasser, M. Network Security Challenges and Countermeasures in SDN Environments. In Proceedings of the 2021 Eighth International Conference on Software Defined Systems (SDS), Gandia, Spain, 6–9 December 2021; pp. 1–8

[100] Alwasel, K.; Jha, D.N.; Hernandez, E.; Puthal, D.; Barika, M.; Varghese, B.; Garg, S.K.; James, P.; Zomaya, A.; Morgan, G.; et al. IoTSim-SDWAN: A Simulation Framework for Interconnecting Distributed Datacenters over Software-Defined Wide Area Network (SD-WAN). J. Parallel Distrib. Comput. 2020, 143, 17–35

[101] Uddin, M.; Mukherjee, S.; Chang, H.; Lakshman, T.V. SDN-Based Service Automation for IoT. In Proceedings of the 2017 IEEE 25th International Conference on Network Protocols (ICNP), Toronto, ON, Canada, 10–13 October 2017; pp. 1–10

[102] H. I. Kobo A. M. Abu-Mahfouz and G. P. Hancke "Fragmentation-Based Distributed Control System for Software-Defined Wireless Sensor Networks" IEEE Transactions on Industrial Informatics vol. 15 no. 2 pp. 901-910 Feb. 2019

[103] W. Munawar M. H. Alizai O. Landsiedel and K. Wehrle "Dynamic TinyOS: Modular and Transparent Incremental Code-Updates for Sensor Networks" 2010 IEEE International Conference on Communications pp. 1-6 2010

[104] C. Han R. Kumar R. Shea and M. Srivastava "Sensor network software update management: a survey" International Journal of Network Management vol. 15 no. 4 pp. 283-294 July 2005

[105] S. Brown and C. Sreenan "Software Updating in Wireless Sensor Networks: A Survey and Lacunae“ Journal of Sensor and Actuator Networks vol. 2 no. 4 pp. 717-760 Nov. 2013

[106] J. Lee "Patch Transporter: Incentivized Decentralized Software Patch System for WSN and IoT Environments" Sensors vol. 18 no. 574 2018

[107] S. Agrawal R. Roman M.L. Das A. Mathuria and J. Lopez "A Novel Key Update Protocol in Mobile Sensor Networks" Notes in Computer Science vol. 7671 2012

[108] P. Radmand A. Talevski S. Petersen and S. Carlsen "Comparison of industrial WSN standards" 4th IEEE International Conference on Digital Ecosystems and Technologies pp. 632-637 2010]

[109] Protocolul Zigbee. Specificații accesibile la: https://web.archive.org/web/20130627172453/http://www.zigbee.org/Specifications/ZigBee/FAQ.aspx (accesat la data de 17.05.2021)

[110] Tehnologia WirelessHART. Specificații accesibile la: https://www.fieldcommgroup.org/sites/default/files/imce_files/technology/documents/WirelessHART_system_eng_guide.pdf (accesat la data de 17.05.2021)

[111] Standardul ISA.100. Specificații accesibile la: https://www.isa.org/standards-and-publications/isa-standards/isa-standards-committees/isa100 (accesat la data de 17.05.2021)

[112] Gast, M.S. 802.11 Wireless Networks: The Definitive Guide; O’Reilly: Chicago, IL, USA, 2002; pp. 218–221

[113] Protocolul TCP specificat în RFC 793. Accesibil la: https://tools.ietf.org/html/rfc793 (accesat la data de 01.06.2020)

[114] Date istorice Meteoblue. Accesibile la: https://www.meteoblue.com/en/historyplus (accesat la data de 01.06.2020)

[115] Cadrul de lucru Qt bazat pe limbajul de programare C++. Accesibil la: https://www.qt.io/ (accesat la data de 15.01.2020)

[116] Mansfield, K.C.; Antonakos, J.L. Computer Networking from LANs to WANs: Hardware, Software, and Security; Cengage Learning: Boston, MA, USA, 2010; p. 501

[117] S. Buzura, B. Iancu, V. Dadarlat, A. Peculea, E. Cebuc "Optimizations for Energy Efficiency in Software-Defined Wireless Sensor Networks", Sensors 2020, 20, 4779. doi: https://doi.org/10.3390/s20174779

[118] S. Buzura, V. Dadarlat, B. Iancu, A. Peculea, E. Cebuc and R. Kovacs, "Self-adaptive Fuzzy QoS Algorithm for a Distributed Control Plane with Application in SDWSN," 2020 IEEE International Conference on Automation, Quality and Testing, Robotics (AQTR), Cluj-Napoca, Romania, 2020, pp. 1-6, doi: 10.1109/AQTR49680.2020.9129922

[119] Utilitarul iPerf. Accesibil la: https://iperf.fr/ (accesat la data de 17.05.2021)

[120] Jin, X.; Gossels, J.; Rexford, J.; Walker, D. CoVisor: A compositional hypervisor for software-defined networks. In Proceedings of the 12th USENIX Conference on Networked Systems Design and Implementation (NSDI'15), Oakland, CA, USA, 4–6 May 2015; USENIX-The Advanced Computing Systems Association: Berkley, CA, USA, 2015; pp. 87–101

[121] Yang, G.; Yu, B.; Jin, H.; Yoo, C. Libera for Programmable Network Virtualization. IEEE Commun. Mag. 2020, 58, 38–44

[122] S. Buzura, M. Lehene, B. Iancu, V. Dadarlat "An Extendable Software Architecture for Mitigating ARP Spoofing-Based Attacks in SDN Data Plane Layer", Electronics 2022, 11, 1965. doi: https://doi.org/10.3390/electronics11131965

[123] S. Buzura, A. Peculea, B. Iancu, E. Cebuc, V. Dadarlat, R. Kovacs "A Hybrid Software and Hardware SDN Simulation Testbed", Sensors 2023, 23, 490. https://doi.org/10.3390/s23010490

[124] S. Buzura, V. Lazar, B. Iancu, A. Peculea and V. Dadarlat, "Using Software-Defined Networking Technology for Delivering Software Updates to Wireless Sensor Networks," 2021 20th RoEduNet Conference: Networking in Education and Research (RoEduNet), Iasi, Romania, 2021, pp. 1-6, doi: 10.1109/RoEduNet54112.2021.9637720

[125] S. Buzura, V. Dadarlat, A. Peculea, H. Bertrand and R. Chevalier, "Simulation Framework for 6LoWPAN Networks Using Mininet-WiFi," 2022 IEEE International Conference on Automation, Quality and Testing, Robotics (AQTR), Cluj-Napoca, Romania, 2022, pp. 1-5, doi: 10.1109/AQTR55203.2022.9802017.

[126] S. Buzura, A. Suciu, E. Cebuc, B. Iancu and V. Dadarlat, "Development Framework for Simulating Routing Behavior in Software-Defined Wireless Networks," 2022 21st RoEduNet Conference: Networking in Education and Research (RoEduNet), Sovata, Romania, 2022, pp. 1-6, doi: 10.1109/RoEduNet57163.2022.9921101.

[127] S. Buzura, B. Iancu and V. Dadarlat, "Creating Educational and Research Tools for QoS-Focused Software-Defined Networking Projects", 2022 IEEE Global Engineering Education Conference (EDUCON), 2022, pp. 1179-1182, doi: 10.1109/EDUCON52537.2022.9766749.

[128] A. Peculea, B. Iancu, S. Buzura, V. Ratiu, coordonatori: V. Dadarlat, E. Cebuc, Computer networks. Practical activities, Ed. U.T. PRESS, 978-606-737-633-3, 2023

[129] M. Lipp et al., "Nethammer: Inducing Rowhammer Faults through Network Requests," 2020 IEEE European Symposium on Security and Privacy Workshops (EuroS&PW), Genoa, Italy, 2020, pp. 710-719, doi: 10.1109/EuroSPW51379.2020.00102.

[130] A. Peculea, B. Iancu, V. Dadarlat, S. Buzura, Circuite analogice și numerice. Aplicatii practice, Ed. U.T. PRESS, 978-606-737-458-2, 2020

[131] A. Peculea, B. Iancu, V. Dadarlat, S. Buzura, Analog and Digital Circuits. Practical Applications, Ed. U.T. PRESS, 978-606-737-459-9, 2020

LISTĂ DE PUBLICAŢII

Articole publicate în reviste indexate ISI:

S. Buzura, B. Iancu, V. Dadarlat, A. Peculea, E. Cebuc "Optimizations for Energy Efficiency in Software-Defined Wireless Sensor Networks", Sensors 2020, 20, 4779. doi: https://doi.org/10.3390/s20174779 **IF: 3.576 Q1**

S. Buzura, A. Peculea, B. Iancu, E. Cebuc, V. Dadarlat, R. Kovacs "A Hybrid Software and Hardware SDN Simulation Testbed", Sensors 2023, 23, 490. https://doi.org/10.3390/s23010490 **IF: 3.847 Q2**

S. Buzura, M. Lehene, B. Iancu, V. Dadarlat "An Extendable Software Architecture for Mitigating ARP Spoofing-Based Attacks in SDN Data Plane Layer", Electronics 2022, 11, 1965. doi: https://doi.org/10.3390/electronics11131965 **IF: 2.69 Q3**

Articole prezentate la conferinţe indexate ISI Proceedings:

S. Buzura, V. Dadarlat, A. Peculea, H. Bertrand and R. Chevalier, "Simulation Framework for 6LoWPAN Networks Using Mininet-WiFi", 2022 IEEE International Conference on Automation, Quality and Testing, Robotics (AQTR), 2022, pp. 1-5, doi: 10.1109/AQTR55203.2022.9802017.

S. Buzura, B. Iancu and V. Dadarlat, "Creating Educational and Research Tools for QoS-Focused Software-Defined Networking Projects", 2022 IEEE Global Engineering Education Conference (EDUCON), 2022, pp. 1179-1182, doi: 10.1109/EDUCON52537.2022.9766749.

S. Buzura, V. Dadarlat, B. Iancu, A. Peculea, E. Cebuc and R. Kovacs, "Self-adaptive Fuzzy QoS Algorithm for a Distributed Control Plane with Application in SDWSN", 2020 IEEE International Conference on Automation, Quality and Testing, Robotics (AQTR), 2020, pp. 1-6, doi: 10.1109/AQTR49680.2020.9129922.

R. A. Kovacs, B. Iancu, V. T. Dadarlat, E. Cebuc and **S. Buzura**, "Extending K-cover genetic algorithm for efficient energy consumption in WSNs", 2019 18th RoEduNet Conference: Networking in Education and Research (RoEduNet), 2019, pp. 1-6, doi: 10.1109/ROEDUNET.2019.8909581.

Articole prezentate la conferinţe indexate BDI:

S. Buzura, A. Suciu, E. Cebuc, B. Iancu and V. Dadarlat, "Development Framework for Simulating Routing Behavior in Software-Defined Wireless Networks", 2022 21st RoEduNet Conference: Networking in Education and Research (RoEduNet), 2022, pp. 1-6, doi: 10.1109/RoEduNet57163.2022.9921101.

Sauchea, **S. Buzura**, A. Peculea, E. Cebuc and V. Dadarlat, "Open Source Network Management System Based on SharpPcap for QoS and Security Policies", 2022 21st

RoEduNet Conference: Networking in Education and Research (RoEduNet), 2022, pp. 1-5, doi: 10.1109/RoEduNet57163.2022.9921027.

R. Kovacs, B. Iancu, V. Dadarlat, **S. Buzura**, A. Peculea and E. Cebuc, "A collaborative game theory approach for determining the feasibility of a shared AS blockchain infrastructure", 2021 20th RoEduNet Conference: Networking in Education and Research (RoEduNet), 2021, pp. 1-6, doi: 10.1109/RoEduNet54112.2021.9637711.

S. Buzura, V. Lazar, B. Iancu, A. Peculea and V. Dadarlat, "Using Software-Defined Networking Technology for Delivering Software Updates to Wireless Sensor Networks", 2021 20th RoEduNet Conference: - Networking in Education and Research (RoEduNet), 2021, pp. 1-6, doi: 10.1109/RoEduNet54112.2021.9637720.

V. Lazar, **S. Buzura**, B. Iancu and V. Dadarlat, "Anomaly Detection in Software Defined Wireless Sensor Networks Using Recurrent Neural Networks", 2021 IEEE 17th International Conference on Intelligent Computer Communication and Processing (ICCP), 2021, pp. 19-24, doi: 10.1109/ICCP53602.2021.9733669.

S. Buzura, B. Iancu, V. Dadarlat, "A System Proposal for Collecting Medical Data from Heterogeneous Sensors Using Software-Defined Networks", 7th International Conference on Advancements of Medicine and Health Care through Technology, Springer International Publishing, MediTech, 2020, pp. 282-289, doi: https://doi.org/10.1007/978-3-030-93564-1_32.

Articol prezentat la o conferință indexată ISI Proceedings înaintea stagiului de doctorat:

S. Buzura, V. Dadarlat, A. Peculea, B. Iancu and E. Cebuc, "Simulations framework for network congestion avoidance algorithms using the OMNeT++ IDE", 2013 11th RoEduNet International Conference, 2013, pp. 1-8, doi: 10.1109/RoEduNet.2013.6511758.

Îndrumătoare de laborator publicate în cadrul grupului de cercetare CNP (Computer Networks and Protocols)

A. Peculea, B. Iancu, **S. Buzura**, V. Rațiu, coordonatori: V. Dadarlat, E. Cebuc, Computer networks. Practical activities, Ed. U.T. PRESS, 978-606-737-633-3, 2023. Disponibil la: https://biblioteca.utcluj.ro/files/carti-online-cu-coperta/633-3.pdf (accesat la data de: 19.05.2023)

A. Peculea, B. Iancu, V. Dadarlat, **S. Buzura**, Circuite analogice și numerice. Aplicatii practice, Ed. U.T. PRESS, 978-606-737-458-2, 2020. Disponibil la: https://biblioteca.utcluj.ro/files/carti-online-cu-coperta/458-2.pdf (accesat la data de: 19.05.2023)

A. Peculea, B. Iancu, V. Dadarlat, **S. Buzura**, Analog and Digital Circuits. Practical Applications, Ed. U.T. PRESS, 978-606-737-459-9, 2020. Disponibil la: https://biblioteca.utcluj.ro/files/carti-online-cu-coperta/459-9.pdf (accesat la data de: 19.05.2023)