RĪGAS TEHNISKĀ UNIVERSITĀTE

Dmitrijs Rjazanovs

DEFEKTTOLERANTU ALGORITMU ANALĪZE KOOPERATĪVO MOBILO TĪKLU IZVEIDEI

Promocijas darba kopsavilkums

RTU Izdevniecība
Rīga 2025

RĪGAS TEHNISKĀ UNIVERSITĀTE
Datorzinātnes, informācijas tehnoloģijas un enerģētikas fakultāte
Fotonikas, elektronikas un elektronisko sakaru institūts

Dmitrijs Rjazanovs
Doktora studiju programmas "Telekomunikācijas" doktorants

DEFEKTTOLERANTU ALGORITMU ANALĪZE KOOPERATĪVO MOBILO TĪKLU IZVEIDEI

Promocijas darba kopsavilkums

Zinātniskie vadītāji
asociētais profesors *Dr. sc. ing.*
ALEKSANDRS IPATOVS
profesors *Dr. habil. sc. ing.*
ERNESTS PĒTERSONS

RTU Izdevniecība
Rīga 2025

Rjazanovs, D. Defekttolerantu algoritmu analīze kooperatīvo mobilo tīklu izveidei. Promocijas darba kopsavilkums. – Rīga: RTU Izdevniecība, 2025. – 46 lpp.

Publicēts saskaņā ar promocijas padomes "RTU P-08" 2025. gada 27. jūnija lēmumu, protokols Nr. 42.

Promocijas darbu atbalstījis Eiropas Sociālais fonds darbības programmas "Izaugsme un nodarbinātība" 8.2.2. specifiskā atbalsta mērķa "Stiprināt augstākās izglītības iestāžu akadēmisko personālu stratēģiskās specializācijas jomās" projekts Nr. 8.2.2.0/20/I/008 "Rīgas Tehniskās universitātes un Banku augstskolas doktorantu un akadēmiskā personāla kvalifikācijas celšana stratēģiskās specializācijas jomās".

Pētījumu atbalstījusi Rīgas Tehniskās universitātes doktorantūras grantu programma.

Vāka attēls no *www.shutterstock.com*.

https://doi.org/10.7250/9789934372261
ISBN 978-9934-37-226-1 (pdf)

PROMOCIJAS DARBS IZVIRZĪTS ZINĀTNES DOKTORA GRĀDA IEGŪŠANAI RĪGAS TEHNISKAJĀ UNIVERSITĀTĒ

Promocijas darbs zinātnes doktora (*Ph. D.*) grāda iegūšanai tiek publiski aizstāvēts 2025. gada 28. novembrī plkst. 11.00 Rīgas Tehniskās universitātes Datorzinātnes, informācijas tehnoloģijas un enerģētikas fakultātē, Āzenes ielā 12, 201. auditorijā.

OFICIĀLIE REZENZENTI

Profesors *Dr.sc.ing.* Vjačeslavs Bobrovs,
Rīgas Tehniskā universitāte

Profesors *Dr. sc. ing. Toledo Moreo Rafael*,
Kartahenas Politehniskā universitāte, Spānija

Profesors *Dr. ing. habil. Mehmet Ercan Altinsoy*,
Drēzdenes Tehniskā universitāte, Vācija

APSTIPRINĀJUMS

Apstiprinu, ka esmu izstrādājis šo promocijas darbu, kas iesniegts izskatīšanai Rīgas Tehniskajā universitātē zinātnes doktora (*Ph. D.*) grāda iegūšanai. Promocijas darbs zinātniskā grāda iegūšanai nav iesniegts nevienā citā universitātē.

Dmitrijs Rjazanovs ……. (paraksts)
Datums: ……

Promocijas darbs ir uzrakstīts angļu valodā, tajā ir ievads, piecas nodaļas, secinājumi, literatūras saraksts, 33 attēli, 12 tabulas, pieci pielikumi, kopā 101 lappuse, neieskaitot pielikumus. Literatūras sarakstā ir 61 nosaukums.

ANOTĀCIJA

Promocijas darbs pievēršas inženiertehnisko pieeju sarežģītībai defekttolerantu mobilo tīklu izstrādē. Problēma sākas ar tradicionālajām IT sistēmām, kurās, neskatoties uz programmatūras izstrādes prakses attīstības līmeni, defekttolerancei bieži netiek veltīta pietiekama uzmanība. Šī perspektīva tiek paplašināta līdz kooperatīvo mobilo tīklu risinājumu jomai. Pētījuma galvenais mērķis ir attīstīt efektīvus, defekttolerantus risinājumus kooperatīviem mobilajiem tīkliem un sniegt ieskatu defekttoleranta konsensa efektivitātē, izpētot dažādas tā formas un praktiskās ieviešanas stratēģijas.

Pētījums īpaši pievēršas izrietoši un sinhrono līdera izvēles algoritmu efektivitātes novērtēšanai kooperatīvajās bezpilota lidaparātu (BPL) patrulēšanas misijās. Tiek sniegti ieskati par izrietošo algoritma konverģences procesa raksturlielumiem, dažādu pauzu stratēģiju ietekmi uz misijas kvalitāti, kooperatīvo BPL tīklu dinamiskajām īpašībām un MI precizitātes un veiktspējas ietekmi uz misijas rezultātiem. Papildus tam konsensa problēma tiek analizēta viedo transportlīdzekļu kooperatīvas sadursmes novēršanas scenāriju kontekstā, pētot, vai Bizantijas kļūdu modelis ir nepieciešams *V2V* sakaros. Tiek izstrādāta metode, lai modelētu un mērītu plūdu konsensa veiktspēju šajā scenārijā, un rezultāti tiek analizēti. Lai sasniegtu šos mērķus, tika sagatavota un izstrādāta jauna reāllaika *Docker* balstīta simulācija. Tika izveidots rindas tīkla modelis, lai analizētu kooperatīvā BPL risinājuma dinamiskās īpašības, savukārt pielāgota *NS3* simulācija tika izstrādāta, lai novērtētu konsensu *V2V* scenārijos.

SATURS

LIETOTIE SAĪSINĀJUMI

A
ACK (*Acknowledgement*) – apliecinājums
API (*Application Programming Interface*) – lietojumprogrammu saskarne

B
BPL – bezpilota lidaparāts
BFT (*Byzantine Fault Tolerance*) – Bizantijas defekttolerance

C
C-ITS (*Cooperative Intelligent Transport Systems*) – kooperatīvas intelektuālās transporta sistēmas
CDF (*Cumulative Distribution Function*) – kumulatīvā sadalījuma funkcija

D
DT (*Delta Timeout) – Delta pauze*

E
ETSI (*European Telecommunications Standards Institute*) – Eiropas Telekomunikāciju standartu institūts

F
FOV (*Field of View*) – redzamības lauks

G
GVR – galvenie veiktspējas rādītāji
GNQN (*Gordon-Newell Queueing Network*) – Gordona–Ņūvela rindas tīkls

H
HWMP (*Hybrid Wireless Mesh Protocol*) – hibrīdais bezvadu režģa protokols

I
IBM (*International Business Machines*) – firma IBM
ID – identifikators
IoT (*Internet of Things*) – lietu internets
IP (*Internet Protocol*) – protokols IP
IS (*Infinite Server*) – bezgalīgais serveris

L
LMT – SIA "Latvijas Mobilais Telefons"

M
MVA (*Mean Value Analysis*) – vidējās vērtības analīze
MI – mākslīgais intelekts

N
NAT (*Network Address Translation*) – tīkla adrešu translēšana
NS-3 (*Network Simulator 3*) – tīkla simulators 3

O
OS – operētājsistēmā

P
PDF (*Probability Density Function*) – varbūtības blīvuma funkcija
PKI (*Public Key Infrastructure*) – publiskās atslēgas infrastruktūra
p2p (*Peer-to-Peer*) – vienādranga tīkls

Q
QN (*Queueing Network*) – rindas tīkls

S
SIM (*Subscriber Identity Module*) – lietotāja identifikācijas modulis
SSD (*Solid State Drive*) – SSD disks
STUN (*Session Traversal Utilities for NAT*) – tīkla adrešu translēšana

T
TCP (*Transmission Control Protocol*) – pārraides vadības protokols
TURN (*Traversal Using Relays around NAT*) – traversēšana izmantojot starpniekus ap NAT

U
UDP (*User Datagram Protocol*) – lietotāja datogrammu protokols
UTF (*Unicode Transformation Format*) – unikoda pārveidošanas formāta simbolu kodējums

V
V2V (*Vehicle-to-Vehicle Communication*) – V2V sakari
V2X (*Vehicle-to-Everything*) – transportlīdzekļa-jebkā sakari
VANET (*Vehicular Ad Hoc Network*) – transportlīdzekļu ekspromttīkls
VT – viedie transportlīdzekļi

W
WAN (*Wide Area Network*) – teritoriālais tīkls

PROMOCIJAS DARBA VISPĀRĒJS RAKSTUROJUMS

Promocijas darbs pēta izaicinājumu noteikt un ieviest pietiekami izturīgus defekttolerantus algoritmus kooperatīvo mobilo tīklu kontekstā. Gan defekttoleranti algoritmi, gan kooperatīvie mobilie tīkli aptver plašu lietojumu klāstu, tāpēc šis pētījums koncentrējas uz īpaši izvēlētiem scenārijiem ar mērķi iegūt vispārīgas atziņas, kas ir piemērojamas arī citiem kontekstiem. Promocijas darbs piedāvā gan teorētiskus, gan praktiskus pienesumus, galvenokārt uzsverot praktisko ieviešanu. Pirmais scenārijs aplūko kooperatīvu BPL perimetra patrulēšanu, ietverot simulāciju un līdera izvēles algoritmu. Otrais scenārijs risina konsensa panākšanas problēmu ārkārtas situācijās viedajos transportlīdzekļu tīklos. Pašreizējās nodaļas mērķis ir īsi iepazīstināt ar pētījuma problēmu, sniegt kontekstuālo informāciju un izskaidrot tās nozīmīgumu.

Tradicionāli defekttolerantu algoritmu lietojums tika apspriests un izstrādāts datubāžu klasteru kontekstā. Pastāv ierobežojumi tam, cik tālu vienu datubāzes serveri var mērogot vertikāli. Sasniedzot noteiktu mērogošanas punktu, var būt nepieciešami vairāki serveri, lai sadalītu pieprasījumu slodzi uz datubāzi. Tas rada datubāzes stāvokļa replikācijas problēmu, kas ir konsensa forma ar daudzām variācijām un ieviešanas metodēm [1]. Kopš pagājušā gadsimta 70. gadiem ir izstrādāti daudzi algoritmi un praktiskas sistēmas, lai risinātu šo problēmu. Teorija un prakse, kas balstās šajās sistēmās, ir samērā augstā līmenī, tomēr joprojām tiek veikti pētījumi šajā jomā [2].

Dalītās sistēmas ir mūsdienu tehnoloģijas pamatelements, kas atbalsta plašu lietojumprogrammu klāstu, tostarp mākoņdatošanu, autonomos transportlīdzekļus, *PageRank*, viedās elektroenerģijas tīklus, stāvokļa novērtēšanu, BPL koordināciju, vairāku pušu vadību, slodzes līdzsvarošanu un blokķēdi. Sākotnēji, balstoties agrīnos darbos datubāžu replikācijā, dalītās sistēmas ir attīstījušās, lai atbalstītu arvien sarežģītākas un decentralizētas arhitektūras. Šis progress ir ievērojami ietekmējis gan tehnoloģiju attīstību, gan sabiedrības mijiedarbību ar digitālajām sistēmām. Lai arī tās nodrošina daudzas mūsdienu autonomās un dalītās tehnoloģijas, dalītās sistēmas joprojām rada gan teorētiskus, gan praktiskus izaicinājumus [3]. Promocijas darba mērķis ir izpētīt, kā pamatkoncepcijas, piemēram, konsenss, var atšķirties pēc ieviešanas un uzvedības atkarībā no konkrētā lietojuma konteksta neatkarīgi no tā, vai tās tiek lietotas transportlīdzekļu tīklos, BPL darbībā vai bezatļaujas vienranga sistēmās.

Kooperatīvie mobilie tīkli ietver autonomus mobilos aģentus, piemēram, transportlīdzekļus, BPL vai robotus, kas mijiedarbojas un koordinējas, lai sasniegtu kopīgus mērķus bez centralizētas kontroles. Šie tīkli darbojas dinamiskā vidē ar mainīgām topoloģijām, neuzticamiem saziņas kanāliem un ierobežotiem resursiem. Efektīvai darbībai tiem ir nepieciešami decentralizēti algoritmi, kas spēj pārvaldīt kļūmes, pieņemt lēmumus kopīgi un pielāgoties reāllaika apstākļiem, radot unikālu krustpunktu starp mobilitāti, autonomiju un dalīto defekttoleranto skaitļošanu.

Autonomās viedās transportlīdzekļu sistēmas ir viens no lielajiem izaicinājumiem gan rūpniecībai, gan akadēmiskajai videi. Autonomo transportlīdzekļu darbība jau paļaujas uz dalītiem konsensa mehānismiem kritiskos uzdevumos, piemēram, kolonnas veidošanā, sadursmju novēršanā un kooperatīvu lēmumu pieņemšanā, uzsverot to nozīmi kā kooperatīvā tīkla sastāvdaļām, nevis izolētiem elementiem [4]. Šie mehānismi nodrošina reāllaika

koordināciju, satiksmes efektivitāti un drošību, tomēr tiem ir jārisina būtiski izaicinājumi. Rodas privātuma jautājumi, jo transportlīdzekļiem nepieciešama plaša datu apmaiņa, savukārt veiktspējas prasības nosaka nepieciešamību pēc izturīgas, zema latentuma apstrādes dinamiskos apstākļos. Turklāt uzticamība ir būtiska, lai novērstu sistēmas kļūmes, kas varētu radīt katastrofālas sekas. Šo izaicinājumu pārvarēšana ir būtiska, lai veicinātu drošu un mērogojamu dalīto sistēmu ieviešanu savienotajās VT ekosistēmās.

Pēdējo 10 gadu laikā personiskie elektriskie BPL ir ieguvuši plašu popularitāti visā pasaulē. Galvenais katalizators šim pieaugumam ir litija polimēru bateriju nozares attīstība [5], [6]. Sākotnēji BPL varēja uzturēties gaisā tikai dažas minūtes, savukārt pašreizējais standarts pārsniedz 20 minūtes [6]. Kooperatīvās BPL sistēmas lielā mērā paļaujas uz dalītiem konsensa mehānismiem, lai nodrošinātu kritiskas darbības, piemēram, formācijas lidošanu, koordinētas meklēšanas un glābšanas misijas, kā arī dinamisku uzdevumu sadali [7], [8]. Šie mehānismi nodrošina reāllaika koordināciju, sadursmju novēršanu un efektīvu resursu izmantošanu BPL tīklā, kas darbojas dinamiskā un neprognozējamā vidē.

Neskatoties uz defekttoleranto pamatelementu kritisko nozīmi mūsdienu IT un kibersistēmās, joprojām pastāv ievērojama plaisa starp to teorētisko izpratni un praktisko lietojumu. Pat tipiskās uzņēmumu sistēmās, kas arvien vairāk ir dalītas, bieži tiek pieļautas pamatkļūdas, kas rada zemu defekttoleranci, samazinātu sistēmas uzticamību un augstākas izstrādes un uzturēšanas izmaksas [9]. Tam ir vairāki iemesli. Viens no tiem ir tas, ka šie pamatelementi parasti tiek abstrahēti ar ietvaru un bibliotēku starpniecību, ļaujot inženieriem tos izmantot, neizprotot esošos pamatpieņēmumus un ierobežojumus. Tomēr praksē šādas abstrakcijas reti nodrošina pilnīgu, visaptverošu defekttoleranci. Izplatīts piemērs ir ziņojumu starpnieku nepareiza izmantošana. Inženieri var pieņemt, ka pietiek ar ziņojuma nosūtīšanu starpniekam pēc darbības, nepievēršot uzmanību tam, ka tīkla kļūmes var traucēt ziņojuma piegādi. Bez papildu mehānismiem, piemēram, "*transactional outbox pattern*" [10], datu konsekvenci un ziņojumu uzticamību nevar garantēt. Vēl viens faktors ir tāds, ka sistēmas bieži tiek projektētas, koncentrējoties uz "laimīgo ceļu", savukārt kļūdu scenāriji tiek ignorēti, īpaši vēlākajos izstrādes posmos, kad budžets un laiks ir ierobežots.

Darba mērķis un uzdevumi

Ņemot vērā iepriekšminētos faktus, tiek noteikts **promocijas darba mērķis** – izpētīt esošos konsensa algoritmus, novērtēt to piemērotību uzlabotos kooperatīvo mobilo tīklu scenārijos un izstrādāt simulācijas, lai novērtētu to veiktspēju sarežģītos apstākļos, nosakot mērķi tos potenciāli attīstīt reālai ieviešanai.

Lai sasniegtu šo mērķi, tika noteikti galvenie uzdevumi.

1. Veikt padziļinātu dalīto sistēmu pamatprincipu izpēti, tostarp defekttolerantos algoritmus, tīkla modeļus un konsensa teorijas robežrezultātus.
2. Izpētīt jaunākos sasniegumus dalītajās sistēmās, tostarp blokķēžu bāzes konsensa algoritmus, un novērtēt to piemērotību mobilo tīklu vidēs.
3. Pētīt un izmantot rindas teoriju kā matemātisko pamatu tīkla veiktspējas un algoritmu efektivitātes novērtēšanai mobilajos tīklos.

4. Izpētīt teorētiskos pamatus un praktiskās ieviešanas metodes BPL bāzētiem mobilajiem tīkliem, tostarp misijas plānošanu, komunikāciju un kooperatīvos scenārijus.
5. Izpētīt *V2X* komunikāciju, aptverot gan teorētiskos pamatus, gan praktiskās tehnoloģijas, ko izmanto viedajos transportlīdzekļu tīklos.
6. Novērtēt esošās simulācijas ietvarus un metodes kooperatīvo BPL misiju modelēšanai, koncentrējoties uz elastību, mērogojamību un tīkla uzticamību.
7. Izstrādāt simulācijas vidi kooperatīvām BPL misijām, kas spēj testēt dažādus kļūdu apstākļus, tostarp BPL avārijas, nestabilus tīklus un aizkavētu komunikāciju.
8. Izstrādāt metodoloģiju, lai salīdzinātu sinhronos un daļēji sinhronos konsensa algoritmus kooperatīvās BPL perimetra patrulēšanas kontekstā.
9. Izstrādāt un ieviest simulācijas modeli konsensam transportlīdzekļu tīklos (*V2V*), koncentrējoties uz defekttolerantu vienošanos avāriju apturēšanas apstākļos.
10. Veikt plašus eksperimentus, izmantojot izstrādātās simulācijas, analizēt rezultātus un formulēt secinājumus par konsensa algoritmu praktisko piemērotību reālās ieviešanas gadījumos.

Pētījumu metodika

Veicot definēto uzdevumu izpildi, tika rūpīgi analizēta dalīto algoritmu teorētiskā pareizība, veiktspēja, pieņēmumi un garantijas. Šī analīze tika apvienota ar programmatūras projektēšanas principiem, lai izstrādātu simulācijas un radītu dažādus scenārijus, kas ļāva kvantitatīvi salīdzināt dalīto sistēmu efektivitāti. Matemātiskie aprēķini tika veikti, lai atbalstītu simulācijas rezultātus, lauka eksperimenti tika veikti, lai pārbaudītu tīkla darbību reālos apstākļos.

Lai novērtētu *NAT* šķērsošanu laukā, tika izmantotas šādas tehnoloģijas: *C#* programmēšanas valoda, *Azure* mākoņa virtuālo mašīnu pakalpojumi, *Ubuntu* bāzēts klēpjdators, *Windows* bāzēts klēpjdators, *UDP* tīkla protokols un *SIM* kartes no vairākiem mobilajiem operatoriem. Šī konfigurācija ļāva veikt visaptverošu *NAT* šķērsošanas un vienranga savienojamības testēšanu dažādos tīkla apstākļos.

Kooperatīvās BPL patrulēšanas misijas simulācijas ieviešanai tika izmantots *Docker* konteinerizētai izvietošanai, savukārt simulācijas loģika tika izstrādāta, izmantojot *C#* un *.NET* 8 ar *XUnit* testēšanas ietvaru, kas nodrošināja rezultātu uzticamību un atkārtojamību. Simulācijā tīkla darbība tika modelēta, izmantojot lognormālu varbūtības sadalījumu, lai attēlotu pārraides aizkaves, parauga lieluma aprēķina formula tika izmantota, lai noteiktu minimālo simulācijas veikšanas reižu skaitu statistiski nozīmīgiem rezultātiem. Pamata sadarbības mehānismi tika veidoti, balstoties uz "izrietoši" un ideālajiem bojājuma indikatoriem, kā arī uz "izrietoši" un ideālajiem līdera indikatoriem, kas nodrošināja stabilu pamatu BPL koordinācijai.

Python kopā ar *Storybook* un *VS Code* vidi tika izmantots simulācijas rezultātu apstrādei un BPL tīkla dinamisko raksturlielumu aprēķināšanai. Matemātiskais pamats balstījās slēgtu *Gordon-Newell* rindas tīkla (*QN*) modelī, kas definēja BPL tīkla galvenās sastāvdaļas. Buzena konvolūcijas algoritms tika izmantots, lai aprēķinātu caurlaidspēju, noslodzi un citas būtiskas

QN īpašības. Turklāt tika izstrādāts pielāgots *Python* skripts, lai aprēķinātu *QN* atbildes laiku varbūtības sadalījumu, izmantojot *QN* normalizācijas konstanti.

NS-3 ietvars tika lietots, lai simulētu visi-uz-visiem" pielāgotus *UDP* un *HWMP* bāzētus apraides algoritmus *802.11s* sieta tīklā. Tika testēti dažādi savienojamības grafu diametri un topoloģijas, lai noteiktu vidējo "visi-uz-visiem" apraides aizkavi. Savāktie dati tika pakļauti polinomu regresijas analīzei, un tā rezultātā tika iegūta trešās pakāpes polinoma aizkaves funkcijas aproksimācija, kas bija atkarīga no mezglu skaita un grafu diametra. Plūdu konsensa algoritms tika izmantots, lai modelētu un aprēķinātu sagaidāmo konsensa aizkaves vērtību mezglu avāriju gadījumā.

Pētījumu rezultāti un zinātniskā novitāte

PROMOCIJAS DARBA JAUNIEGUVUMI

1. Tika izstrādāta 2D paplašināma reāllaika simulācijas platforma kooperatīvām BPL misijām, kas atbalsta vairākus vadības režīmus un ir paredzēta sistēmas izturības pārbaudei tādos apstākļos kā tīkla nestabilitāte, mezglu avārijas un aizkavēta komunikācija. Platforma ir modulāra un atkārtoti izmantojama kā pamats praktiskiem risinājumiem.
2. Tika ierosināta un lietota metode, kas ļauj kvantitatīvi novērtēt "izrietoši" algoritmu ietekmi, izmantojot GVR. Šī pieeja nodrošina tiešu algoritmu veiktspējas salīdzināšanu, balstoties izmērāmos rādītājos.
3. Ideālā līdera noteikšanas algoritma ar konservatīvu pauzi veiktspēja tika salīdzināta ar "izrietoši" ideālā līdera indikatoru kooperatīvas BPL perimetra patrulēšanas misijas kontekstā, nodrošinot datus par to uzticamību un konverģences uzvedību.
4. Tika izstrādāts slēgts *Gordon-Newell* rindas tīkla (*QN*) modelis, lai novērtētu kompromisu starp centralizētā MI pakalpojuma atbildes laiku un iebūvētā MI precizitāti kooperatīvās BPL patrulēšanas misijās. Modelis ļauj tieši salīdzināt sistēmas veiktspēju dažādās konfigurācijās.
5. *NS-3* vidē tika izstrādāts un simulēts pielāgots varbūtības *UDP* un *HWMP* balstīts "visi-uz-visiem" apraides protokols. Simulācijas rezultāti tika izmantoti, lai modelētu plūdu konsensa algoritma uzvedību komunikācijas saišu trūkuma apstākļos.

PROMOCIJAS DARBA GALVENIE SECINĀJUMI

1. Novērtējot defekttolerantu algoritmu ietekmi, ieteicams vispirms izstrādāt optimālu lēmumu pieņemšanas loģiku, jo tā būtiski ietekmē misijas GVR pat scenārijos ar plašiem bojājumiem.
2. Reāllaika simulācijai ir ārkārtīgi liela nozīme kļūdu identificēšanai, kas citādi parādītos reālas izvietošanas laikā, kad to novēršana var būt ievērojami dārgāka. Galvenais reāllaika simulācijas trūkums ir izpildes laiks. Hibrīdā pieeja (diskrētas simulācijas izmantošana kopējās veiktspējas testēšanai un reāllaika simulācija sarežģītiem

gadījumiem projekta vēlākajos posmos) var palīdzēt sasniegt maksimālu defekttoleranci un izturību.

3. Simulācijas dati rāda, ka kumulatīvais kļūdainu avāriju skaits, ko izraisa "izrietoši" līdera indikators simulācijas laikā, ievēro logaritmisku modeli. Palielinoties izvietoto BPL skaitam, laiks līdz pēdējai kļūdainai avārijas noteikšanai arī ievērojami pieaug. Ar vairāk nekā pieciem BPL un lielāku noteikšanas pauzi (*DT*) kļūdainas avāriju noteikšanas var turpināties līdz 10 minūtēm. Tomēr, pateicoties šīs uzvedības logaritmiskajam raksturam, lielākā daļa kļūdainu avāriju notiek ātri, un tikai neliela daļa no kopējām kļūmēm ietekmē GVR.
4. Kooperatīvās BPL perimetra patrulēšanas misiju simulācijās kvantitatīvi GVR rādītāji ir vērtīgi, lai salīdzinātu līdera izvēles algoritmu veiktspēju un noteiktu, vai "izrietoši" konsistences algoritmi ir piemēroti kooperatīvo BPL scenārijiem.
5. Algoritmā "Ideāls līdera indikators" avāriju noteikšanas pauzes pagarināšana no 15 sekundēm (ideāli) līdz 60 sekundēm izraisa GVR samazinājumu par 12,1 % piecu minūšu simulācijā, ja BPL avāriju līmenis ir 30 %. Algoritms "Izrietoši ideāls līdera indikators", konfigurēts ar piecu sekunžu bāzes pauzi un trīs sekunžu delta, uzrāda 4,1 % GVR zaudējumu, salīdzinot ar "Ideālu" algoritmu ar ideālu pauzi. Būtiski, ka "Izrietoši ideāls" algoritms pārspēj "Ideālo" algoritmu reālos apstākļos (60 s pauze), nodrošinot 9,7 % GVR uzlabojumu, ja avāriju līmenis ir 30%.
6. BPL uzraudzības tīklu, kas izmanto MI bāzētu draudu noteikšanu, var modelēt, izmantojot slēgtu *Gordon-Newell* rindas tīklu. Lai gan iebūvētā MI precizitātes uzlabojumi lineāri samazina laiku, kas zaudēts kļūdainai pozitīvai uzraudzībai, centrālā MI atbildes laika samazināšanai ir lielāka ietekme uz zaudētā uzraudzības laika samazināšanu, pieaugot sistēmas noslodzei.
7. Bizantijas kļūdu modelis ir plaši izplatīts avionikā un kosmosa inženierijā. Tomēr šis termins netiek pieminēts *ETSI ITS* tehniskajos ziņojumos. *ITS* literatūras apskats atklāj, ka Bizantijas defekttolerantie algoritmi, kas ierosināti *V2X* sistēmām, galvenokārt koncentrējas uz privātumu, blokķēžu arhitektūru un reputācijas mehānismiem.
8. Pielāgots, nejaušināts, *UDP* bāzēts, uzticams, "visi-uz-visiem" apraides algoritms, izmantojot *802.11s* sieta tīklu un testēts *NS-3* simulācijā, uzrāda vidējo aizkāvi 280 ms 33 mezgliem ar savienojamības grafu diametru 10. Mazākiem diametriem un mezglu skaitiem vidējās aizkaves ir zem 100 ms.
9. Matemātiskā modelēšana parāda, ka kļūdu apturošais plūdu konsensa algoritms, balstīts minētajā apraides algoritmā, teorētiski var sasniegt konsensu mazāk nekā 500 ms laikā, ja $D = 8$, $N = 11$, ar līdz pat 30 % mezglu avārijām un mazāk nekā 200 ms laikā mazākiem diametriem. Uzticamās apraides aizkaves līknes forma ir atkarīga no saziņas grafu diametra: ja $D = 2$, aizkave pieaug līdz ar mezglu skaitu, savukārt, ja $D \geq 7$, pieauguma ātrums samazinās. Lineārāka tendence tiek novērota tad, ja $D = 4$.

Promocijas darba praktiskā vērtība

1. Izstrādāta *UDP NAT* šķērsošanas programmatūra, lai izveidotu autonomus BPL tīklus mobilajos tīklos. Veikti lauka eksperimenti, lai pārbaudītu *NAT* šķērsošanu dažādu mobilo tīklu nodrošinātāju tīklos.
2. Izstrādāta un ieviesta *UDP* bāzēta defekttoleranta komunikācijas starpniekprogrammatūra kooperatīvām BPL misijām. Izveidots 2D reāllaika paplašināms simulācijas ietvars kooperatīvām BPL misijām, koncentrējoties uz defekttolerances pārbaudi.
3. Simulācijas laikā atklāts, ka "izrietoši" līdera indikatora algoritma konverģences laiks uzrāda logaritmisku saistību ar delta parametru un BPL skaitu. Identificēta saistība starp misijas GVR un BPL skaitu, izmantojot simulācijas analīzi.
4. Analizēts, kā dažādas "izrietoši" un ideālā līdera indikatora algoritmu konfigurācijas ietekmē misijas GVR, sniedzot vadlīnijas piemērotu risinājumu izvēlei kooperatīvām BPL misijām.
5. Izstrādāts matemātisks modelis, kas palīdz programmatūras un aparatūras konfigurācijas izstrādē kooperatīviem BPL tīkliem.

Zinātniskā pētījuma promocijas darba rezultātus var izmantot kooperatīvu BPL un citu mobilo mezglu misiju un pakalpojumu risinājumu izstrādei.

Promocijas darba aizstāvamās tēzes

1. Izmantojot jaunu reāllaika 2D simulāciju sadarbīgai, BPL balstītai perimetra apdraudējumu meklēšanas sistēmai ar *UDP* balstītu komunikāciju un nestabilu tīklu, kas modelēts ar lognormālām aizkavēm ($\mu = 6$, $\sigma = 1.5$), ir iespējams novērtēt līdera vēlēšanu stratēģiju efektivitāti. Eksperimenti rāda, ka "izrietoši" ideāls algoritms ar adaptīvu avārijas noteikšanas taimautu uzlabo misijas GVR par 9,7 %, salīdzinot ar statiskiem, konservatīviem taimautiem.
2. Izmantojot slēgtu *Gordon-Newell* rindu tīkla modeli kooperatīvām BPL apdraudējumu meklēšanas misijām, ir iespējams parādīt, ka attālinātā MI pakalpojuma reakcijas laika samazināšana efektīvāk samazina bezpilota lidaparātu izniekoto laiku kļūdainas klasifikācijas dēļ nekā borta MI precizitātes uzlabošana.
3. Izmantojot jaunu *NS-3* simulāciju un *UDP* balstītu varbūtisku uzticamas apraides mehānismu, eksperimenti rāda, ka kļūdu apturošais plūdu konsensa algoritms spēj sasniegt konsensu 99 % gadījumu mazāk nekā 500 ms laikā *802.11s* sieta tīklā ar astoņiem pārlēcieniem un 11 mezgliem pie līdz pat 30 % mezglu avārijām un mazāk nekā 200 ms laikā mazākiem diametriem.

Pētījuma rezultātu aprobācija

Promocijas darba galvenie rezultāti prezentēti 10 starptautiskās zinātniskās konferencēs, kā arī atspoguļoti divās publikācijās zinātniskajos žurnālos un astoņās publikācijās pilna teksta konferenču materiālos.

ZIŅOJUMI STARPTAUTISKĀS ZINĀTNISKĀS KONFERENCĒS

1. **Rjazanovs D**., Pētersons E., Ipatovs A., Juškaite L., Yeryomin R. "Byzantine Failures and Vehicular Networks". // 2021 IEEE Microwave Theory and Techniques in Wireless Communications (MTTW). Latvia, Riga, 2021. pp. 30–34.
2. **Rjazanovs D**., Petersons E. "Decentralized Byzantine Fault-Tolerant Proof of Location". // PoEM Workshops, 2020. pp. 1–10.
3. **Rjazanovs D**., Ratkuns A., Kārkliņš T., Nagla I., Ipatovs A. "Making Ping Feint to Avoid Service State Desynchronization". // 2023 Workshop on Microwave Theory and Technology in Wireless Communications (MTTW). 2023. pp. 34–38.
4. **Rjazanovs D**., Grabs E., Pētersons E., Aleksandrovs-Moisejs D., Chen T., Čulkovs D., Aleksandrovs M., Ipatovs A. "Drone Cooperation, NAT Traversal, and Performance". // 2024 Photonics & Electromagnetics Research Symposium (PIERS). 2024. pp. 1–6.
5. **Rjazanovs D**., Grabs E., Pētersons E., Lahs A., Kopats A., Kārkliņš T., Aleksandrovs-Moisejs D., Titovics J., Chen T., Čulkovs D., Aleksandrovs M., Ipatovs A. "Analysis of Leader Election Algorithms in the Context of Cooperative UAV Mission Simulation". // 2025 Photonics & Electromagnetics Research Symposium (PIERS). 2025.
6. Aleksandrovs-Moisejs D., Ipatovs A., Grabs E., **Rjazanovs D**. "Evaluation of a Long-Distance IEEE 802.11 ah Wireless Technology in Linux Using Docker Containers". // Elektronika ir Elektrotechnika, Vol. 28, No. 3, 2022, pp. 71–77.
7. Chen T., Grabs E., Petersons E., Efrosinin D., Ipatovs A., Bogdanovs N., **Rjazanovs D**. "Multiclass Live Streaming Video Quality Classification Based on Convolutional Neural Networks". // Automatic Control and Computer Sciences, Vol. 56, No. 5, 2022, pp. 455–466.
8. Aleksandrovs-Moisejs D., Ipatovs A., Grabs E., **Rjazanovs D**., Sinuks I. "Arduino-based Temperature Sensor Organization and Design". // 2023 PhotonIcs & Electromagnetics Research Symposium (PIERS). 2023. pp. 1408–1415.
9. Andrejevs D., Grabs E., Čulkovs D., Jeralovičs V., Juškaite L., Chen T., **Rjazanovs D**., Ipatovs A. "Development of Laser Communication Algorithm for Moving Objects". // 2024 Photonics & Electromagnetics Research Symposium (PIERS). 2024. pp. 1–4.
10. Klūga J., Grabs E., Chen T., Ancāns A., Stetjuha M., **Rjazanovs D**., Ipatovs A. "Motion Sensors Data Fusion for Accurate Measurement in AHRS Systems". // 2024 Photonics & Electromagnetics Research Symposium (PIERS). 2024. pp. 1–4.

Darba apjoms un struktūra

Promocijas darba apjoms ir 101 lappuse. Darbā ir piecas nodaļas, nobeigums, literatūras saraksts un pieci pielikumi.

Pirmā nodaļa sākas ar īsu ievadu dalīto sistēmu teorijā, aptverot tās vēsturi, defekttoleranci, pamatstruktūras elementus un sarežģītību. Tajā tiek aplūkota parādība, kā lietojumprogrammas un teorijas ir attīstījušās pēdējo desmitgažu laikā, dabiski veidojot mobilo tīklu balstītas dalītās sistēmas. Nodaļā uzsvērta šo sistēmu kritiskā nozīme mūsdienu pasaulē, izceļot ievērojami augstāku mobilo tīklu sarežģītību, salīdzinot ar tradicionālajām IT sistēmām, un apspriests paradokss par zemas kvalitātes defekttolerances ieviešanu standarta inženierpraksē. Nodaļā sniegti arī visi nepieciešamie pamati turpmākajām nodaļām, tostarp esošās literatūras apskats un problēmas formulējums.

Otrā nodaļa pilnībā veltīta simulācijai, kas izstrādāta šī promocijas darba galvenajiem eksperimentiem. Tā sākas ar prasību izklāstu, kas īpaši definētas šim pētījumam, aptverot gan funkcionālos, gan nefunkcionālos aspektus, kā arī izpētot iespējas atkārtoti izmantot esošās simulācijas platformas. Pēc tam tiek aplūkoti ar saziņu saistīti temati, tostarp *NAT* šķērsošana, saziņas protokoli un tīkla aizkaves simulācija. Tālāk padziļināti aplūkota sistēmas arhitektūra, koncentrējoties uz slāņošanu un modularitāti. Visbeidzot tiek izklāstīta testēšanas stratēģija, statistiskā analīze un secinājumi.

Trešā nodaļa veltīta līdera izvēles algoritmu problēmai kooperatīvas BPL perimetra patrulēšanas kontekstā. Konkrēti, tiek analizēti un salīdzināti divi algoritmi: ideāls līdera indikators un "izrietoši" līdera indikators. Galvenais mērķis ir kvantitatīvi novērtēt "izrietoši" algoritma izmantošanas ietekmi uz misijas veiktspēju. Tiek veikti dažādi eksperimenti, izmantojot iepriekšējā nodaļā aprakstīto simulācijas platformu. Rezultāti tiek prezentēti, uzsverot "izrietoši" algoritma konverģences laika ietekmi un identificējot, kurš algoritms konkrētos scenārijos darbojas labāk.

Ceturtā nodaļa koncentrējas uz konkrētu kooperatīvas BPL patrulēšanas aspektu, proti, hibrīdas MI draudu noteikšanas pieejas izmantošanu un to, kā katras sastāvdaļas uzlabojumi ietekmē misijas GVR. Tiek prezentēts ierosinātais slēgtais *Gordon-Newell* rindas tīkla (*QN*) modelis, detalizēti aprakstot sistēmas sastāvdaļas, parametrus un pieeju atbildes laika varbūtības sadalījuma aprēķināšanai.

Piektajā nodaļā risinātas viedo transportlīdzekļu ārkārtas konsensa problēmas, analizējot, vai Bizantijas kļūdu modelis ir piemērojams *V2X* scenārijos. Nodaļa sākas ar pieejamās literatūras pārskatu par Bizantijas modeļa lietojumu gan *V2X*, gan citās jomās. Otrajā daļā tiek aprēķināts kļūdu apturošā plūdu konsensa izpildes laiks ārkārtas sadursmju novēršanas *V2V* scenārijos. Tajā sniegts detalizēts *NS-3* simulācijas konfigurācijas apraksts, kas ietver "visi-uz-visiem" *802.11s* sieta tīklu un *UDP* bāzētu varbūtības uzticamu apraidi. Apraides izpildes laika rezultāti tiek aproksimēti un izmantoti konsensa izpildes laika modelēšanai.

Promocijas darba noslēgumā apkopoti galvenie darba secinājumi un definēti turpmākās pētniecības virzieni. Pielikumi ietver konferenču, publikāciju un projektu sarakstus, eksperimentālo mērījumu rezultātus un simulācijas kodus.

PROMOCIJAS DARBA SATURS

Pirmā nodaļa

Nodaļas mērķis ir īsi iepazīstināt ar pētījuma problēmu, sniegt kontekstuālu informāciju par defekttoleranci un mobilajiem tīkliem un izskaidrot tās nozīmīgumu.

1.1. apakšnodaļā apkopotas dalīto sistēmu teorijas klasifikācijas. Inženieri pašsaprotami koncentrējas uz galveno lietošanas gadījumu, risinot mērķus un pamatfunkcionalitātes, kas sistēmai ir jānodrošina. Kad galvenā funkcionalitāte ir ieviesta, bieži vien paliek maz laika vai vēlmes apsvērt negatīvos scenārijus un īpašos gadījumus. Neraugoties uz plašajām teorētiskajām zināšanām par kļūmēm, tendence ignorēt negatīvos scenārijus joprojām pastāv, ko uzskatāmi apliecina "izkliedētās skaitļošanas maldi" nemainīgā aktualitāte [11]. Šie maldi demonstrē nemainīgo kļūdu režīmu nenovērtēšanu dalītajās sistēmās.

Saskaņā ar [12] klasifikāciju pastāv piecas kļūdu klases: avārijas kļūdas (*Crash Faults*); izlaiduma kļūdas (*Omission Faults*); avārijas ar atkopšanos (*Crash with Recovery*); noklausīšanās (*Eavesdropping*); Bizantijas kļūdas (*Byzantine Faults*). Pētījums koncentrējas uz avārijas kļūdām un Bizantijas kļūdām. Pamatojoties uz to, tiek iegūtas šādas algoritmu klases: apturoša kļūme (*fail-stop*); kļūme bez atteices (*fail-silent*); kļūme ar maldinošu informāciju (*fail-noisy*); kļūme ar atkopšanos (*fail-recovery*); kļūme ar patvaļīgu darbību (*fail-arbitrary*); randomizēti algoritmi. Pētījumā tika izmantotas apturošas kļūmes, kļūmes ar maldinošu informāciju un randomizētie algoritmi.

1.2. apakšnodaļa apskatīti līdera izvēles un konsensa algoritmi esošajā literatūrā. Līdera izvēle ir dalīto sistēmu pamatproblēma, kas saistīta ar viena izcila procesa (vai mezgla) izvēli starp izkliedētiem un potenciāli identiskiem procesiem [13]. Lai līdera izvēles protokoli būtu efektīvi, tiem ir jānodrošina, ka visi pareizie procesi vienojas par vienu un to pašu līderi un ka šis līderis ir unikāls un aktīvs. Klasiskā problēmas formulējumā ir trīs vēlamās īpašības: unikalitāte (tikai viens līderis vienlaikus); vienošanās (visi pareizie procesi vienojas par līderi); pabeigšana (līderis galu galā tiek izvēlēts) [14]. Lai padarītu līdera izvēli iespējamu daļēji sinhronās sistēmās [15], pētnieki ieviesuši bojājuma indikatorus, kas darbojas kā abstrakti moduļi, nodrošinot (iespējami neuzticamu) informāciju par procesu kļūmēm. Divas labi zināmas bojājuma indikatoru klases ir ideāls bojājuma indikators (P) un izrietošais ideāls bojājuma indikators ($\Diamond P$) [12]. Tiek pētītas divas līdera izvēles stratēģijas:

1) līdera izvēles algoritms, kas balstīts ideālu bojājuma indikatorā (P), kas nodrošina precīzu procesu avāriju noteikšanu;
2) līdera izvēles algoritms, kas balstīts izrietoši ideālā bojājuma indikatorā ($\Diamond P$), kas sākotnēji var vērst uzmanību uz pareiziem procesiem, bet laika gaitā stabilizējas.

Visus līdera izvēles algoritmus var iedalīt divās galvenajās kategorijās: tie, kas pieņem sinhronu tīklu; tie, kas darbojas daļēji sinhronā tīklā. Katrā kategorijā ir dažādas ieviešanas. Līdera izvēle ir cēlusies no tīkla sastāvdaļām, piemēram, *IBM* marķiergredzena (*IBM Token Ring*) [16], kur datori bija savienoti fiziskā gredzena vai zvaigznes topoloģijā. Klasisks piemērs ir *Bully* algoritms [17], kas pieņem vispārēju topoloģiju un sinhronu tīklu, paļaujoties uz statiskām pauzēm, lai noteiktu neaktīvu serveri. Cits piemērs datubāzu klasteros ir *Raft*

algoritms [18], kur līderis tiek izvēlēts, lai nodrošinātu efektīvu stāvokļa mašīnas replikāciju (pilnīgas secības apraide). Atšķirībā no *Bully* algoritma *Raft* pieņem daļēji sinhronu tīklu, taču tam ir nepieciešams, lai vairāk nekā puse mezglu būtu tiešsaistē, lai algoritms darbotos pareizi.

Darba [19] autors piedāvāja *Raft* balstītu līdera izvēles un klasterizācijas shēmu, kas ir izturīga pret BPL kļūmēm. Līdzīgi – darba [20] autori piedāvā simulācijā balstītu pētījumu par balsošanas metodi līdera izvēlei BPL kopās ar vadītāja-sekotāja konfigurāciju ierobežotas saziņas apstākļos. *Mousavi et al.* [21] risināja koalīciju veidošanu BPL tīklos, izmantojot daudzkritēriju optimizācijas algoritmu, kas balstīts kvantu evolūcijas tehnikās. Viņu darbs koncentrējas uz izmaksu, uzticamības un uzticamības optimizēšanu, piešķirot BPL uzdevumu koalīcijām. *Ganesan et al.* [22] ierosināja *BOLD* algoritmu, kas apvieno daļiņu kopas optimizāciju un zirnekļpērtiķu heiristiku, lai izvēlētos energoefektīvus klasteru vadītājus dinamiskos BPL tīklos. *Wang et al.* [23] ieviesa drošu *UAV* atbalstītu klastera vadītāja izvēles mehānismu, kas paredzēts, lai aizsargātu bezvadu sensoru tīklus no apdraudētajiem mezgliem.

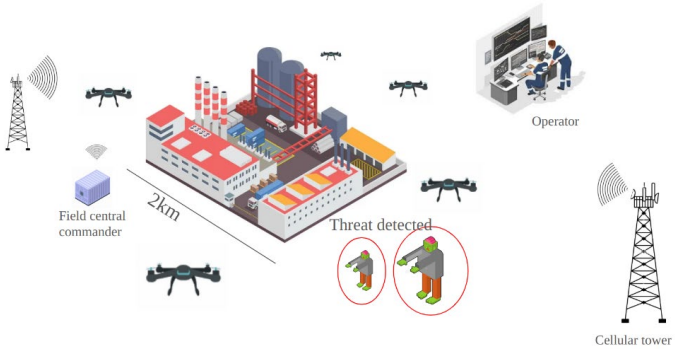
Kopumā iepriekšējos darbus par līdera izvēli BPL sistēmās var iedalīt trīs plašās kategorijās: (1) optimizācijā balstīta līdera izvēle, kas koncentrējas uz enerģijas vai resursu efektivitāti; (2) drošas vai uzraudzītas shēmas, kas pieņem centralizētu kontroli vai ideālu novērošanu; (3) konsensā balstīti modeļi, kas apraksta korektumu, bet tiek novērtēti tikai ideālos apstākļos.

Šajā apakšnodaļā tiek apspriesti esošie konsensa algoritmi un to īpašības. Konsenss ir dalītās skaitļošanas pamatproblēma, kur grupai procesu ir jāvienojas par vienu vērtību, neskatoties uz kļūdām un komunikācijas aizkavēm. Formāli konsensa algoritmam ir jāatbilst šādām īpašībām [14]: (1) vienošanās (*agreement*) – nevienam no pareizi funkcionējošiem procesiem nevajadzētu pieņemt atšķirīgu lēmumu; (2) derīgums (*validity*) – ja visi pareizi funkcionējošie procesi piedāvā vienu un to pašu vērtību, tai ir jābūt pieņemtajai vērtībai; (3) pabeigšana (*Termination*) – katram pareizi funkcionējošam procesam laika gaitā ir jāpieņem lēmums.

Transportlīdzekļu tīklos, īpaši *V2V* scenārijos, konsenss tiek izmantots drošības kritiskai koordinācijai, piemēram, sadursmju novēršanai vai kooperatīvai bremzēšanai. Šie scenāriji nosaka stingras reāllaika prasības, parasti zem 500 ms, un tiem ir jāiztur komunikācijas aizkaves un neuzticamas saites [4]. Šis pētījums koncentrējas uz avāriju apturošu konsensu sinhronās sistēmās, kur ziņojumu aizkaves ir ierobežotas un kļūmes tiek droši noteiktas, kā to pieprasa plūdu konsensa algoritms [12].

Otrā nodaļa

Otrās nodaļas mērķis ir visaptveroši prezentēt kooperatīvās BPL misijas funkcionalitātes izstrādi un tās atbilstošās reāllaika simulācijas ietvaru. Turklāt funkcionalitāte tika izstrādāta ar atkārtotu izmantojamību prātā, ļaujot to integrēt reālās BPL programmatūrās.



1. attēls. Kooperatīva BPL patrulēšanas risinājuma koncepcija.

2.1. apakšdaļā aprakstītas simulācijas prasības, sniegts funkcionālais apraksts, simulācijas veidi un tehnoloģiskais nodrošinājums. Galvenais mērķis ir izstrādāt paplašināmu platformu, lai eksperimentētu ar dažādiem kooperatīviem BPL scenārijiem. Simulācijas galvenais uzsvars ir nodrošināt ietvaru sistēmas defekttolerances testēšanai, lai atvieglotu tās izvietošanu reālās infrastruktūrās.

Šajā apakšnodaļā aprakstīts sistēmas lietojuma scenārijs un pamatfunkcionalitāte (1. attēls). Scenārijā tiek pieņemts, ka pastāv lielas infrastruktūras, kas parasti atrodas piepilsētas vai attālos rajonos. Šādas infrastruktūras var ietvert rūpnīcas, elektrostacijas, lidostas, ostas, stratēģisko infrastruktūru, militāros vai citus augstas vērtības objektus. Tipiska perimetra garums svārstās no 1 km līdz 20 km. Galvenais mērķis ir reāllaikā atklāt potenciālus draudus, kas parādās gar perimetru. Pamatideja ir tāda, ka BPL floti (iespējams, līdz 50 vienībām) var izvietot, lai autonomi sadalītu uzraudzības teritoriju un nepārtraukti uzraudzītu potenciālos draudus. Lai gan specifiskais aparatūras un infrastruktūras nodrošinājums, kas nepieciešams šādas misijas atbalstam, pārsniedz šī pētījuma robežas, tiek definēti vairāki vispārīgi pieņēmumi [24], [25]. Katrs BPL ir aprīkots ar iebūvētu MI, kas spēj reāllaikā veikt objektu klasifikāciju. Turklāt tiek pieņemts, ka ir pieejama centralizēta apstrādes vienība, kas nodrošina uzlabotu MI bāzētu analīzi, ļaujot ar augstāku pārliecību noteikt un klasificēt draudus.

Darba pamatprincips paredz, ka uzticamībai un drošībai jābūt iestrādātai sistēmas arhitektūrā jau no paša sākuma, izmantojot defekttolerantu dizainu. Lai gan turpmāk šajā apakšnodaļā tiks apspriestas dažādas arhitektūras pieejas, šī pētījuma uzmanības centrā ir līdera balstīta arhitektūra. Šajā modelī viens BPL tiek izvēlēts kā līderis starp visām iesaistītajām vienībām. Līderis ir atbildīgs par atrašanās vietas datu vākšanu no sekojošajiem BPL, notikumu ziņojumu apstrādi (piemēram, draudu noteikšana vai to izzušana), optimālās misijas stratēģijas aprēķināšanu, tās sadalīšanu konkrētos uzdevumos un šo uzdevumu nodošanu multiraidē atbilstošajiem BPL izpildei. Piemēram, līderim ir jāsadala perimetrs segmentos un jāpiešķir katrs segments piemērotākajam BPL patrulēšanai. Līdera balstītā arhitektūra nodrošina līdzsvaru starp defekttoleranci un misijas efektivitāti. Tomēr, tāpat kā jebkuram BPL, arī pašam līderim var rasties kļūme. Šādos gadījumos atlikušie BPL ir spiesti izvēlēt jaunu līderi. Izaicinājums ir nodrošināt, lai visi BPL vienotos par vienu un to pašu līderi, jo pretējā gadījumā misija var kļūt fragmentēta vai neefektīva. Tas veido klasisku iepriekš aprakstīto konsensa

problēmu; tāpat kā visiem dalītajiem algoritmiem, kas tiek izmantoti šajā kontekstā, arī līdera izvēles mehānismam ir jābūt defekttolerantam.

Turpmākajā nodaļā tiek analizēta izvēle starp deterministisko vai reāllaika simulāciju. Deterministiskajai pieejai ir trīs galvenās priekšrocības. Pirmkārt, tā garantē reproducējamību, veicot simulāciju ar to pašu sākotnējo vērtību, vienmēr tiek iegūta tāda pati darbību un rezultātu secība. Otrkārt, simulāciju var izpildīt ievērojami ātrāk nekā reāllaikā, un tās ātrumu ierobežo tikai pieejamie aparatūras resursi. Treškārt, atkļūdošana ir ievērojami vienkāršota. Tomēr šāda veida simulācijai ir arī vairāki ierobežojumi. Simulācijas soļu granularitāte ir jādefinē manuāli, un tās integrācija ar reālām tīkla sastāvdaļām nav iespējama, jo simulācija pēc savas būtības ir deterministiska.

Otrais simulācijas veids ir reāllaika sistēma ar reālu saziņas steku. Šajā režīmā programma darbojas tā, it kā reāli BPL būtu izvietoti laukā, uztverot reālistisku paralēlismu un tīkla uzvedību. Šī pieeja ļauj simulēt patiesu paralēlismu, palīdzot atklāt problēmas, kas rastos reālās izvietošanas laikā, tādējādi ievērojami samazinot izstrādes un kvalitātes nodrošināšanas izmaksas. Tika izvēlēta reāllaika *Docker* balstīta un *.NET 8* (*C#*) simulācijas pieeja, jo tā ļauj visaptveroši testēt defekttolerantu starpprogrammatūru un atkārtoti izmantot programmas moduļus reālai izvietošanai (1. tabula).

1. tabula

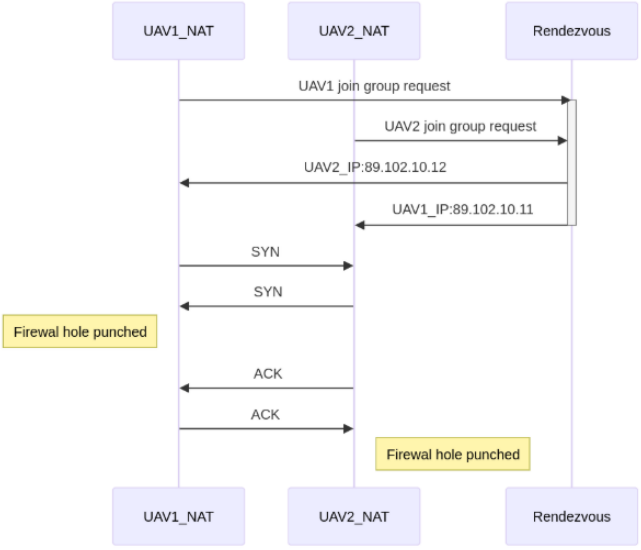
Simulācijas ietvaru salīdzinājums

Funkcija/Prasība	*OMNeT++*	*NS-3*	*SimPy*	*Gazebo*	*AirSim*	*Custom C# & Docker*
Reāls UDP protokola atbalsts	ar spraudni	Nē	Nē	Nē	Nē	Jā
Nedeterministiska vienlaicība	Nē	Nē	Nē	Nē	Nē	Jā
Mērogojamība	Augsts	Augsts	Vidēji	Vidēji	Zems	Augsts
Pielāgojamība	Augsts	Augsts	Augsts	Augsts	Vidēji	Ļoti augsts
Atkārtojamība	Jā	Jā	Jā	Jā	Jā	Daļēji
Viegla integrācija ar aparatūru	Nē	Nē	Nē	Jā	Nē	Jā

2.2. apakšnodaļā tiek apspriestas problēmas, kas saistītas ar savienojuma izveidi starp BPL. Kooperatīvo BPL koordinācijai ir pieejamas vairākas saziņas protokolu iespējas. Plānotajā reālās pasaules izvietošanā par galveno saziņas veidu ir izvēlēta mobilā komunikācija vairāku praktisku iemeslu dēļ. Pirmkārt, mobilie tīkli nodrošina plašu pārklājumu, tostarp daudzās lauku un nomaļās teritorijās. Otrkārt, tie piedāvā gatavu infrastruktūru ar augstu caurlaidspēju, novēršot speciālu saziņas sistēmu nepieciešamību.

Lai iesāktu misiju, operatoram ir jākonfigurē katrs BPL ar unikālu identifikatoru un saistītu *SIM* karti. Tomēr viens no galvenajiem mobilo tīklu ierobežojumiem ir tas, ka tie pēc noklusējuma neatbalsta tiešu starpierīču (*P2P*) saziņu, jo klienta ierīcēm nav publisko *IP* adrešu. Lai gan daudzos pētījumos mobilais tīkls tiek uzskatīts par galveno vai papildu saziņas kanālu, tikai daži pievēršas izaicinājumam nodrošināt *P2P* saziņu šajā kontekstā [26], [27], [28]. Mobilā infrastruktūra nav paredzēta, lai katra ierīce darbotos kā serveris. Neskatoties uz šo ierobežojumu, daudzas viedtālruņu lietotnes, tostarp ziņojumapmaiņas platformas, veiksmīgi nodrošina *P2P* saziņu. Tas tiek panākts, izmantojot *NAT* šķērsošanas tehnikas, kas ļauj ierīcēm aiz privātām *IP* adresēm izveidot tiešus saziņas kanālus (2. attēls).

Pastāv dažādi *NAT* veidi: (1) pilnais konusveida *NAT* (*Full Cone NAT*); (2) ierobežotais konusveida *NAT* (*Restricted Cone NAT*); (3) porta ierobežotais konusveida *NAT* (*Port-Restricted Cone NAT*); (4) simetriskais *NAT* (*Symmetric NAT*). Pēdējais veids ir ierobežojošākais un sarežģītākais. Tas ģenerē unikālu kartējumu katrai iekšējā avota un ārējā galamērķa adreses un porta kombinācijai. Ja abām ierīcēm ir simetriskais *NAT*, vienīgais veids, kā izveidot vienranga (*P2P*) savienojumu, ir izmantot starpniekserveri (*relay server*) [29]. Visos pārējos *NAT* veidos var izmantot *UDP* "caurumu izsišanas" (*UDP hole punching*) tehniku.



2. attēls. *UDP NAT* šķērsošanas algoritma secības diagramma.

Simulācijas ietvaros tika ieviests īpašs "tikšanās punkta" (*Rendezvous*) serveris. Tā mērķis ir ļaut BPL izveidot savstarpēju *P2P* savienojumu. Serveris īsteno *UDP* "caurumu izsišanas" metodi un ir gatavs izvietošanai reālajā vidē.

Papildus tika veikti daži lauka eksperimenti, lai pārbaudītu, kā dažādi mobilo tīklu nodrošinātāji konfigurē savas *NAT* ierīces. **1. eksperiments** ietvēra konfigurāciju ar vienu LMT mobilo klientu un vienu *Azure* mākoņa serveri ar publisku *IP* adresi. Šī konfigurācija bija veiksmīga, apstiprinot, ka izejošā *UDP* satiksmes plūsma no LMT klienta var sasniegt publisko

serveri un saņemt atbildes un *NAT* kartējumi saglabājās pietiekami ilgi, lai nodrošinātu stabilu saziņu. **2. eksperimentā** tika pārbaudīta tiešā saziņa starp diviem LMT mobilo klientiem, kas atrodas fiziski tuvu viens otram. Neskatoties uz to, ka abām ierīcēm operators piešķīra atšķirīgas publiskās *IP* adreses, mēģinājums nebija veiksmīgs. Šis rezultāts norāda, ka LMT lietotās *NAT* vai ugunsmūra politikas ierobežo vienādranga *UDP* savienojamību, iespējams, simetriskā *NAT* uzvedības vai operatora ieviesto papildu drošības mehānismu dēļ. **3. eksperimentā** tika savienots viens LMT klients ar vienu BITE mobilo klientu. Līdzīgos apstākļos, kādi bija iepriekšējā eksperimentā, *UDP* "caurumu izsišana" bija veiksmīga. Tas liecina, ka vismaz dažas mobilo tīklu nodrošinātāju kombinācijas atbalsta tiešu *UDP* saziņu, visticamāk, sakarā ar atšķirībām *NAT* konfigurācijā vai ugunsmūra politikās starp pakalpojumu sniedzējiem.

Nodrošinot alternatīvu *NAT* šķērsošanai, daži mobilo tīklu operatori piedāvā *SIM* kartes ar publiski maršrutējamām *IP* adresēm par papildu maksu. Šī pieeja vienkāršo vienādranga saziņu un ļauj izmantot arī *PVP*, kas var būt vēlamāks scenārijos, kad nepieciešama uzticama piegāde un savienojuma stāvokļa pārvaldība. Turpmākā nodaļa ir veltīta reālistiska tīkla uzvedības modelēšanai un ieviešanai – pakešu aizkavēm un zudumiem. Visa komunikācija starp BPL simulācijas laikā tika veikta, izmantojot atsevišķu pielāgotu *UDP* starpniekservera konteineru. BPL izmanto īpašas galvenes, kas norāda starpniekserverim sākotnējo *UDP* paketes galamērķi. Pats starpniekserveris izmanto konfigurējamu lognormālu sadalījumu, lai atlasītu nejaušas paketes aizkaves (kas ietver secības izmaiņas) un nejaušus zudumus. Starpniekservera caurlaidspēja tika izmērīta un sasniedza 350 paketes sekundē ar vidējo izmēru 250 baiti.

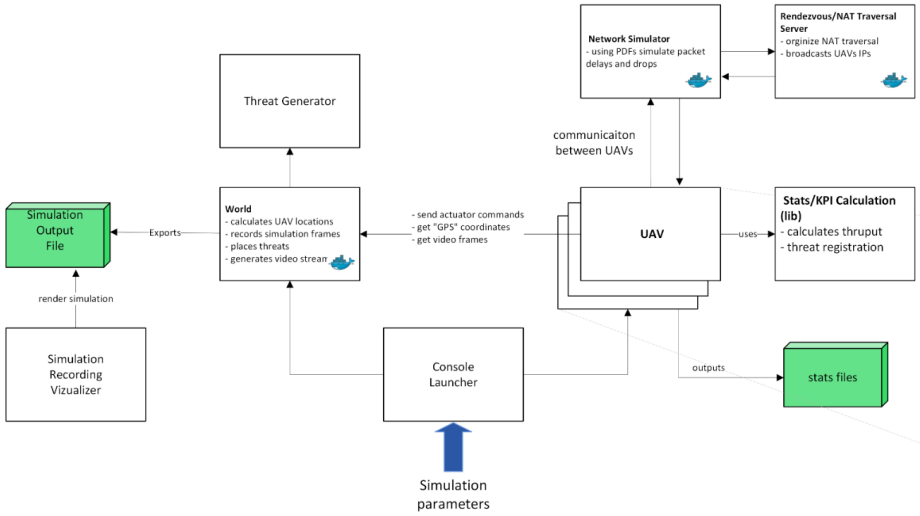
```
Implements:
    EventuallyPerfectFailureDetector, instance ◇P.

Uses:
    PerfectPointToPointLinks, instance pl.

upon event ⟨ ◇P, Init ⟩ do
    alive := Π;
    suspected := ∅;
    delay := Δ;
    starttimer(delay);

upon event ⟨ Timeout ⟩ do
    if alive ∩ suspected ≠ ∅ then
        delay := delay + Δ;
    forall p ∈ Π do
        if (p ∉ alive) ∧ (p ∉ suspected) then
            suspected := suspected ∪ {p};
            trigger ⟨ ◇P, Suspect | p ⟩;
        else if (p ∈ alive) ∧ (p ∈ suspected) then
            suspected := suspected \ {p};
            trigger ⟨ ◇P, Restore | p ⟩;
        trigger ⟨ pl, Send | p, [HeartbeatRequest] ⟩;
    alive := ∅;
    starttimer(delay);
```

3. attēls. Izrietoši ideāls kļūdu indikators – taimauta palielināšanas algoritms [12].

2.3. apakšnodaļā aprakstīta simulācijas arhitektūra. Galvenās sastāvdaļas redzamas 4. attēls. Platformas kodolā atrodas *World* modulis, kas simulē 2D virtuālo vidi. Šis modulis ir atbildīgs par BPL pozīciju atjaunināšanu, draudu ievadīšanu un simulācijas stāvokļa ierakstīšanu kadru pa kadram. *World* modulis ģenerē arī vienkāršotu videostraumi, kur katrs kadrs tiek attēlots, izmantojot *UTF-16* rakstzīmju režģi, kas ļauj viegli vizualizēt simulāciju tieši konsolē. Šī pati 2D simboliskā reprezentācija tiek izmantota, lai emulētu BPL redzi, kur

katrs BPL uztver savu apkārtni caur lokalizētu logu, kas tiek attēlots kā *UTF-16* simbolu matrica.

Visi simulācijas rezultāti tiek ierakstīti simulācijas izvades failā, ko vēlāk var vizualizēt, izmantojot simulācijas ierakstu vizualizatoru. Šis rīks atkārto simulāciju, izmantojot to pašu *UTF-16* konsolē balstīto attēlošanas dzini, kas ļauj efektīvi pārskatīt eksperimenta rezultātus bez nepieciešamības atkārtoti veikt pilnu simulāciju.



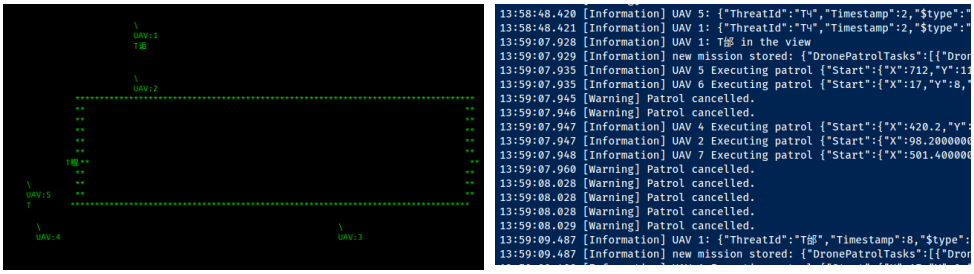
4. attēls. Simulācijas platformas arhitektūra.

Lai novērtētu BPL uzvedības un koordinācijas stratēģiju efektivitāti, simulācija seko vairākiem galvenajiem veiktspējas rādītājiem (GVR). Šie rādītāji novērtē, cik labi tiek atklāti, uzraudzīti un kontrolēti draudi, kā arī BPL saziņas efektivitāti.

- *T. reg.* (%) – draudu reģistrācijas rādītājs. Šis rādītājs atspoguļo procentuālo daļu draudu, ko BPL veiksmīgi reģistrējuši simulācijas laikā. Drauds tiek uzskatīts par reģistrētu, ja to patrulēšanas laikā atklāj kāds BPL. Vērtība tiek aprēķināta kā reģistrēto draudu skaits, dalīts ar kopējo ģenerēto draudu skaitu.
- *T. lifetime mon.* (%) – draudu uzraudzības laiks. Šis GVR mēra to draudu dzīves ilguma procentuālo daļu, kuru laikā draudi tika aktīvi uzraudzīti ar BPL. Katram draudam ir noteikts dzīves ilgums, un jebkurš laiks, kad to uzrauga BPL, tiek ieskaitīts šajā rādītājā. Tas atspoguļo nepārtrauktas uzraudzības efektivitāti.
- *T. contr.* (%) – draudu kontroles rādītājs. Kad drauds ir reģistrēts, sistēma var piešķirt divus BPL, lai to vienlaikus uzraudzītu; šo stāvokli dēvēt par kontroli. Šis rādītājs atspoguļo to draudu procentuālo daļu, kas jebkurā simulācijas brīdī sasniedza šo kontrolēto stāvokli.
- *T. lifetime contr.* (%) – kontrolētais dzīves ilgums. Šis GVR mēra procentuālo daļu no kopējā draudu dzīves ilguma, kurā visi draudi atradās kontrolē, tas ir, katram draudam

bija piešķirti vismaz divi BPL. Tas norāda, cik konsekventi sistēma uzturēja pilnu pārklājumu pār aktīvajiem draudiem.

5. attēlā redzams simulācijas piemērs ar nelielu perimetru.

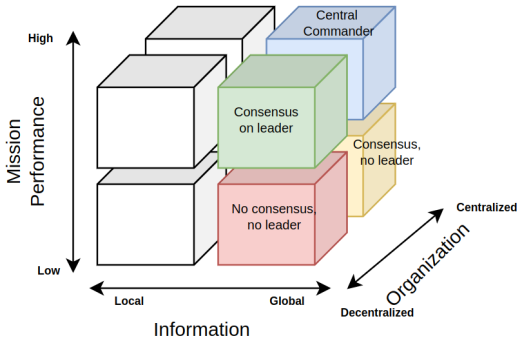


5. attēls. Simulācijas vizualizatora piemērs (pa kreisi) un reālā laika ieraksti (pa labi).

Prezentētā simulācijas platforma nodrošina reālistisku BPL koordinācijas stratēģiju testēšanu, veidojot tiltu starp simulāciju un reālo izvietošanu. Atšķirībā no *NS-3*, kas ir piemērots protokolu testēšanai, šī platforma ļauj vienmērīgi pāriet no simulācijas uz fiziskajiem BPL, nodrošinot konsekventu izstrādes procesu. Galvenie secinājumi ietver būtisko vienlaicības problēmu ietekmi un reāllaika izpildes nozīmi, lai atklātu laika kļūdas, un reāllaika simulācijas ierobežojumus tās ilguma dēļ. *Docker* nodrošināja tīkla reālismu, nevis atmiņas izolāciju, un fiksētas sākotnējās vērtības uzlaboja testa atkārtojamību, taču nevarēja garantēt pilnīgu determinismu. Platforma efektīvi modelē aizkaves un pakešu zudumus, padarot to par spēcīgu rīku *UAV* sistēmu testēšanai un izstrādei.

Trešā nodaļa

Šajā nodaļā prezentēti pētījuma rezultāti par to, kā laika gaitā atbilstoša līdera izvēle ietekmē kooperatīvo misiju veiktspēju. **3.1. apakšnodaļā** formulēta problēma, aprakstīts simulācijā izmantotais misijas kontroles režīms un izskaidroti pētījumā izmantotie defekttolerantie pamatelementi. Viens no galvenajiem mērķiem ir novērtēt, vai laika gaitā atbilstoša līdera izvēle var efektīvi darboties kooperatīvos scenārijos, piemēram, perimetra patrulēšanā. Galvenais izaicinājums ir izprast, kā pagaidu līderības nekonsekvence ietekmē misijas veiktspēju. Lai to izpētītu, misijas efektivitāte tiek novērtēta, izmantojot kvantitatīvus rādītājus (GVR), kas ļauj analizēt, kā nekonsekvence ietekmē kopējo misijas kvalitāti.



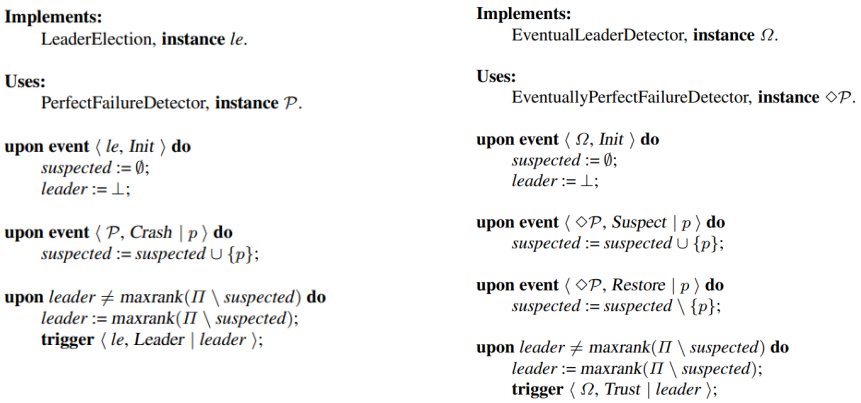
6. attēls. Kooperatīvās kontroles režīmu klasifikācija.

6. attēlā redzami visi BPL kontroles režīmi, kas ieviesti simulācijā. Katrs no režīmiem balstās dažādos saziņas un defekttolerances pamatelementos. Šajā pētījumā tiks izmantots līderu konsensa režīms (vai dinamiskais līderis), jo tas nodrošina vislabāko līdzsvaru starp veiktspēju un defekttoleranci [30], [31].

Šajā pētījumā tiek izmantoti divi bojājumu indikatora veidi: ideāls bojājuma indikators (P); izrietoši ideāls bojājuma indikators ($\Diamond P$), abi definēti 1. nodaļā. Jāatzīmē, ka termins "ideāls" nenozīmē, ka avārijas noteikšana ir sinhronizēta visos mezglos. Praksē, ja avārijas noteikšanas pauze ir T, tad laika starpība starp jebkuru divu mezglu procesu X kļūmes noteikšanu būs mazāka par T, un sliktākajā gadījumā noteikšanas laiks būs ierobežots ar $2T$. Tas notiek tāpēc, ka mezgls var atbildēt ar apstiprinājumu jau signālziņojuma intervāla sākumā, radot situāciju, kurā aizrit gandrīz $2T$, pirms tā neesamība tiek atklāta.

Lai gan pārbaudes biežuma palielināšana var uzlabot noteikšanas laiku, tas notiek uz lielākas tīkla satiksmes plūsmas rēķina, kas potenciāli var ietekmēt kopējo sistēmas veiktspēju. Šajā pētījumā tiek apsvērti tieši divi līdera izvēles algoritmi, katrs atvasināti no viena no iepriekš aprakstītajiem bojājumu indikatoriem (7. attēls). Vienkāršības labad algoritms, kas balstās ideālā bojājuma indikatorā (P), tiek dēvēts par "ideālu" līdera izvēles algoritmu, bet tas, kas balstās izrietoši ideālā bojājuma indikatorā ($\Diamond P$), tiek saukts par "izrietoši" algoritmu (7. attēls). Par "izrietošo" algoritmu ir svarīgi saprast, kā darbojas pamata dinamiskais taimauts (3. un 9. attēls).

Svarīgākā ideālā algoritma īpašība, kas nodrošina to, ka pēc līdera izvēlēšanās tiek garantēts, ka visi iepriekš aizdomīgie līderi patiešām ir avarējuši. Lai gan līdera izvēle nenotiek sinhroni, jo ziņojumu piegāde ir asinhrona, P stiprās garantijas nodrošina, ka neviens pareizi funkcionējošs process vienlaikus neuzskatīs par aktīviem dažādus līderus. Tas efektīvi novērš kļūdaini dubultas vadības (*split-brain*) scenārijus ideālā algoritmā.



7. attēls. "Ideālā" līdera izvēles algoritms (pa kreisi) un "izrietošā" līdera izvēles algoritms (pa labi).

8. attēlā apkopotas katra algoritma stiprās puses un ierobežojumi. Zilās krāsas apmale izceļ šī pētījuma fokusu: novērtēt, kā laika gaitā atbilstošais modelis ietekmē misijas GVR un vai pastāv iespēja, ka tas pārspēj ideālo modeli apstākļos, kad ilgāka pauze (pielāgota novērotajām aizkavēm) samazina kļūdainu avāriju noteikšanu..

Mazāks taimauts	Lielāks taimauts	Dinamisks taimauts
Ātra atbilde	Vienmēr viens līderis	Ātrāka atbilde
Vairāki līderi	Lēmumu aizkavēšanās	Galu galā viens līderis
Dubulta pārklāšanās		Pagaidu nekonsekvence

8. attēls. Dažādu paužu stratēģiju plusi un mīnusi.

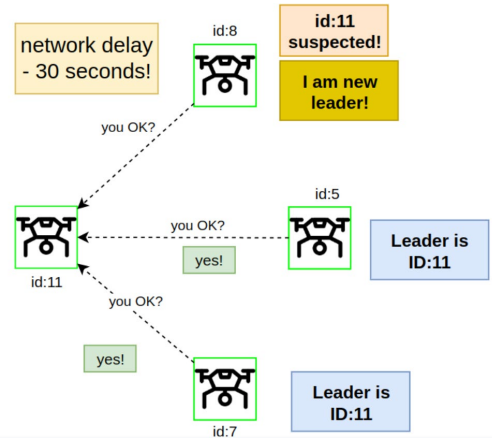
3.2. apakšnodaļā aprakstīti etalonieksperimenti. Simulācijas vide ietver plašu parametru klāstu, piemēram, perimetra formu un izmēru, BPL lidojuma augstumu (kas nosaka uzraudzības diametru), BPL ātrumu, draudu parādīšanās ātrumu, BPL skaitu un ar tīklu saistītos iestatījumus. Tomēr, ņemot vērā reāllaika simulācijas, kur trīs minūšu simulācija prasa trīs minūtes reālas izpildes laikam, nav iespējams saprātīgā laika periodā izpētīt visas iespējamo parametru kombinācijas. Tādēļ šis pētījums koncentrējas tikai uz izvēlētā parametru apakškopas mainīšanu: tīkla konfigurāciju, pauzes iestatījumiem un BPL avāriju līmeņiem. Pārējie parametri tiek saglabāti nemainīgi visos eksperimentos, un to vērtības ir apkopotas 2. tabulā.

2. tabula

Simulācijas pamatparametri

Parametra nosaukums	Vienība	Vērtība
Perimetra augstums	Metri	500
Perimetra platums	Metri	100
BPL video diametrs	Metri	30
BPL ātrums	Metri/sek	12
BPL skaits	BLP	10
Vienas simulācijas ilgums	Minūtes	3
Draudu biežums	Draudi/min	5
Draudu vidējais mūžs	Sekundes	30
Simulācijas laiks	Sekundes	300
Delta pauze	ms	3000

Uzraudzības (video) diametrs ir fiksēts 30 metru apmērā, kas aprēķināts, izmantojot 2. formulu, kur redzamības lauks (*FOV*) ir 80°, BPL lidojuma augstums *h* ir 20 metri. BPL ātrums 12 m/s atbilst vidējam lidojuma ātrumam *DJI Matrice 300 RTK*, padarot to par reālistisku parametru izmantošanai reālās izvietošanas situācijās.



9. attēls. Izrietošā līdera izvēles algoritma izpildes piemērs. Daļējas sinhronitātes dēļ mezgli īslaicīgi nevienojas par to, kurš ir līderis.

Lai novērtētu nepieciešamo simulācijas izpildes reižu skaitu, kas nodrošinātu uzticamu vidējo vērtību galvenajiem veiktspējas rādītājiem (GVR), tika izmantota šāda funkcija:

$$n = \left(\frac{z \cdot \sigma}{E}\right)^2, \quad (1)$$

kur: Z – Z-punktu skaits (*Z-score*), kas atspoguļo uzticamības līmeni (piemēram, 1,96 par uzticamības līmeni); n – minimālais nepieciešamo simulāciju skaits; σ – datu kopas standarta novirze; E – pieļaujamā kļūdas robeža.

Ja uzticamības līmenis ir 95 %, ņemot vērā novēroto standarta novirzi $\sigma = 3{,}82$ un noteikto kļūdas robežu 1 (E), tika noteikts, ka nepieciešamas 56 simulācijas, lai nodrošinātu uzticamu vidējo vērtību. Ņemot vērā simulācijas ilgumu (piecas minūtes), tas ļauj 12 stundu testēšanas logā novērtēt aptuveni divas atšķirīgas parametru konfigurācijas scenārijus.

$$h = \frac{D}{2 \times \tan\left(\frac{F}{2}\right)}, \tag{2}$$

kur F – kameras redzamības lauks, grādi;
h – bezpilota lidaparāta lidojuma augstums, m;
D – videokamerā iemūžinātie zemes metri, m.

Lai simulētu nestabilu bezvadu tīkla vidi, ziņojumu kavējumu modelēšanai tika izvēlēts lognormāls sadalījums. Atšķirībā no standarta normālā sadalījuma lognormāls sadalījums pieļauj gadījuma rakstura ilgākus kavējumus, labāk atspoguļojot reālās pasaules bezvadu sakaru neregularitāti. Sadalījums ir konfigurēts ar parametriem $\mu = 6$ un $\sigma = 1{,}5$, ieviešot piegādes laika mainīgumu. Turklāt, lai vēl vairāk imitētu neuzticamus tīkla apstākļus, tiek piemērots 1 % *UDP* pakešu atmešanas ātrums.

Vienkāršotajā izpildē mezgls A atzīmētu mezglu B kā neveiksmīgu, ja *ACK* netiktu saņemts X pauzes limita laikā, kas potenciāli varētu izraisīt kļūdainu vai retu, bet pamatotu kļūdu atklāšanu paketes zudumu aizkavēšanās noviržu dēļ. Lai to mazinātu, tika ieviests atkārtošanas mehānisms: ja laika posmā X netiek saņemts *ACK*, mezgls A nosūta signālu atkārtoti līdz n reizēm. Katrs atkārtojums tiek modelēts kā neatkarīgs paraugs no tā paša lognormālā kavēšanās sadalījuma, un tas ir neatkarīgi pakļauts paketes zudumam.

Varbūtība, ka viens atkārtošanas mēģinājums neizdosies – vai nu paketes atmešanas, vai aizkavēšanās dēļ, kas pārsniedz X, – ir noteikta 3. vienādojumā. Lognormālās kavēšanās kumulatīvā sadalījuma funkcija ir 4. vienādojums. Pieņemot neatkarību, varbūtību, ka visi n atkārtotu mēģinājumu neizdosies, raksturo 5. vienādojums.

$$P_{\text{fail}}(X) = p_{\text{drop}} + (1 - p_{\text{drop}}) \cdot \mathbb{P}(D > X) \tag{3}$$

$$\mathbb{P}(D \le X) = \Phi\left(\frac{\ln(X) - \mu}{\sigma}\right) \quad \Rightarrow \quad \mathbb{P}(D > X) = 1 - \Phi\left(\frac{\ln(X) - \mu}{\sigma}\right) \tag{4}$$

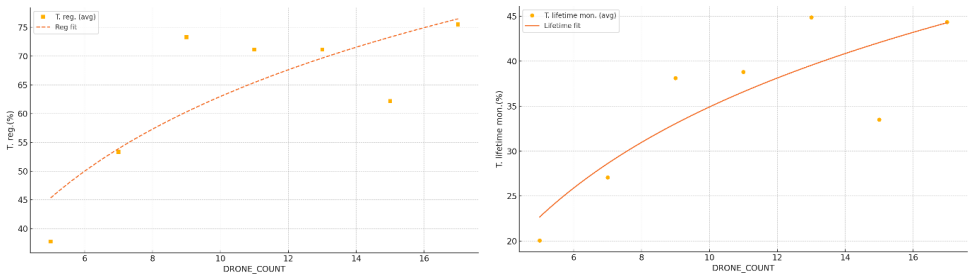
$$P_{\text{total fail}}(X, n) = [p_{\text{drop}} + (1 - p_{\text{drop}}) \cdot \mathbb{P}(D > X)]^n, \tag{5}$$

kur Φ ir *KSF*. Teorētiskais modelis rāda, ka četri atkārtojumi ar kavēšanās slieksni $D = 3500$ ms nodrošina 99,999 % varbūtību, ka netiks konstatēta kļūdaina avārija. Vairāk nekā 200 simulācijas braucienu, katrs piecas minūtes garš, apstiprina to, ka, izmantojot ideālu 15 sekunžu pārtraukumu un atkārtotu mēģinājumu intervālu 3500 ms, nav novērots neviens kļūdains avārijas atklāšanas gadījums.

Eksperimentu rezultāti parādīja, ka gan **darbības laika uzraudzības** GVR, gan draudu reģistrācijas laika (*t. reg.*) GVR ir logaritmiski atkarībā no bezpilota lidaparātu skaita (10. attēls). Ņemot vērā 1200 metru perimetru, 10 BPL izvietošana nozīmē, ka katrs BPL ir atbildīgs

par aptuveni 100 metru uzraudzību. BPL skaita palielināšana virs 10 nodrošina tikai minimālus veiktspējas uzlabojumus, bet būtiski palielina tīkla trafiku.

Tika veikts ideāls tīkla eksperiments ar pakešu zudumu likmi 0 un bez aizkavēm. Pēc tam tika piemērots iepriekšējā apakšnodaļā aprakstītais degradētais tīkla modelis. Kā redzams 3. tabulā, neuzticami tīkla apstākļi var izraisīt GVR samazinājumu līdz pat 7,2 %.



10. attēls. Aptuvenā funkcija, kas balstīta simulācijas datos par GVR atkarību no dronu skaita (pa kreisi – t reģ., pa labi – T darb. uzr.).

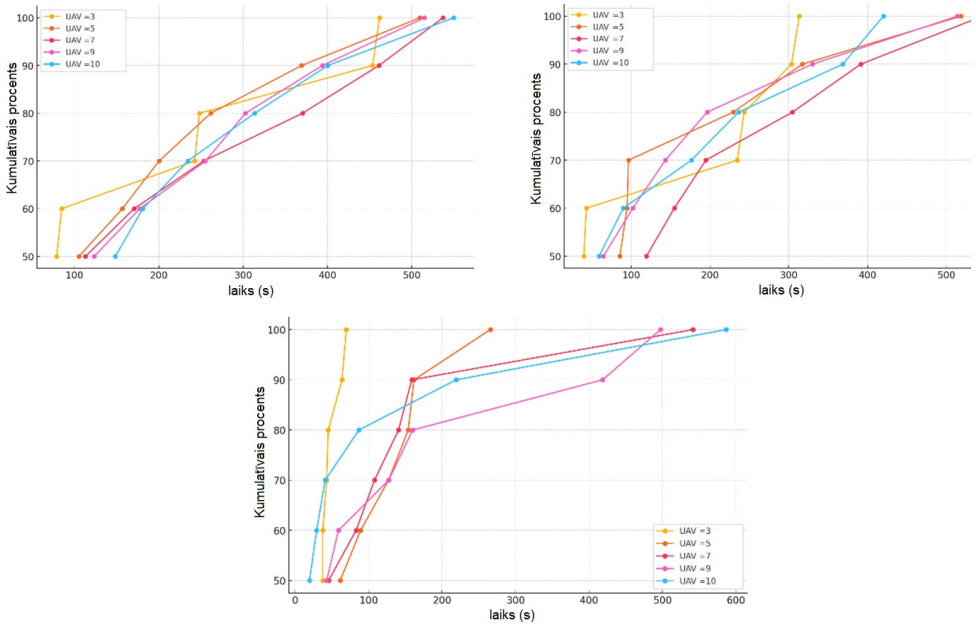
3. tabula

"Ideālā" tīkla un bāzes līnijas parametru salīdzinājums ("sliktais" tīkls)

GVR	**"Ideals" tīkls**	**"Slikts" tīkls**
T. reg. (%)	71	71
T. contr. (%)	52	46
T. lifetime mon. (%)	64	60
T. lifetime contr. (%)	47	40
Summa	**234**	**217**

3.3. apakšnodaļā tiek analizēts kļūdaini dubultas vadības (*split-brain*) stāvoklis – tiek pētīti apstākļi, kuros tas var rasties, tā paredzamais ilgums, pamatojoties uz sistēmas parametriem, un ietekme uz misijas GVR.

Ne visi kļūdainās avārijas noteikšanas gadījumi tieši izraisa dubultas vadības stāvokli. Piemēram, ja kļūdaina avārija tiek noteikta starp diviem BPL ar zema līmeņa *ID*, no kuriem neviens nav pašreizējais līderis, sistēmas funkcionalitāte netiek ietekmēta. Abi BPL turpinās saņemt komandas no pašreizējā līdera un sūtīt tam notikumu atjauninājumus. Tomēr problēmas rodas, ja kļūdaina avārija tiek noteikta starp BPL ar *ID*, kas ir vienāds ar pašreizējā līdera *ID* – 1, un pašreizējo līderi.



11. attēls. Kumulatīvais kļūdainu avāriju skaits simulācijas laikā atkarībā no delta pauzes (*DT*) = 1 s (pa kreisi), 2 s (pa labi), 3 s (centrā).

Šo eksperimentu mērķis bija noteikt brīdi, pēc kura pauzes pielāgojumi vairs nenotiek, t. i., visi BPL ir stabilizējušies un kļūdainas avāriju noteikšanas vairs neturpinās. Kā bija gaidāms, tas ir cieši saistīts ar tīkla modeli. Eksperimenti tika veikti, izmantojot iepriekš aprakstīto degradēto tīkla modeli. Rezultāti parāda, ka delta pauzes (*DT*) parametrs būtiski ietekmē konverģences laiku (11. attēls). Eksperimenti pierādīja, ka pat ar lielu *DT* vērtību (trīs sekundes) un BPL grupu lielumu virs pieciem kļūdaini pozitīvi gadījumi var saglabāties līdz pat 10 minūtēm. Neskatoties uz to, lielākā daļa kļūdainu avāriju noteikšanas notiek pirmajās trīs minūtēs.

3.4. apakšnodaļā tiek salīdzināti "izrietošā" un "ideālā" algoritma GVR. Pēdējā eksperimentā tiek novērtēts, kā visas trīs konfigurācijas darbojas dažādos avāriju līmeņos un degradētos tīkla apstākļos (4. tabula). Ja avāriju līmenis ir 30 %, "izrietošais" algoritms uzrāda tikai 2 % GVR zaudējumu, salīdzinot ar ideālo pauzi. Jāatzīmē, ka dažās simulācijās pauze bija īsāka par ideālajām 15 sekundēm. Šāda uzvedība ir sagaidāma un var daļēji izskaidrot minimālo atšķirību, kas novērota starp abiem algoritmiem. Ja avāriju līmenis ir 50 %, tiek novērota lielākā atšķirība – "izrietošais" algoritms uzrāda 3,7 % zaudējumu, salīdzinot ar ideālo pauzi. Ja avāriju līmenis ir 70 %, atšķirība kļūst nebūtiska, novirze starp "izrietoši" un ideālo konfigurāciju ir mazāka par 1 %.

Sākotnējais šī pētījuma mērķis bija atbildēt uz jautājumu, vai izrietošā līdera izvēle var pārspēt ideālo līdera izvēli sagaidāmo BPL avāriju kontekstā. Šis pētījums pierādīja, ka laika gaitā atbilstoši sadalītos algoritmus – konkrēti, "izrietoša" līdera izvēle, – var efektīvi izmantot kooperatīvās BPL misijās, piemēram, perimetra patrulēšanā. Lai gan *UAV* pamatkontroles

loģikai ir lielāka ietekme uz misijas panākumiem nekā neregulārām nekonsekvencēm, "izrietoša" pieeja parādīja augstu izturību avārijām pakļautā vidē un noteiktos apstākļos pat pārspēja perfekto līdera izvēles stratēģiju. Galvenie atklājumi parādīja, ka misijas veiktspēju ietekmē tīkla nestabilitāte, BPL skaits un konverģences laiks nelineārā veidā. Papildus tika pierādīts, ka sistēmas izturību var vēl vairāk uzlabot, iekļaujot loģiku nekonsekvenču pārvaldīšanai, piemēram, vairāku līderu gadījumā. Kopumā "izrietošie" algoritmi piedāvā praktisku, defekttolerantu alternatīvu ideālistiskiem koordinācijas modeļiem, jo īpaši reālās vides scenārijos, kur avārijas un tīkla traucējumi ir neizbēgami.

Ceturtā nodaļa

Patrulēšanas misijā tiek izmantoti divi MI veidi – iebūvētais BPL MI un jaudīgāks, uz zemes izvietots centrālais MI mezgls. Šis pētījums koncentrējas uz draudu noteikšanas aspektu patrulēšanas misijā. Šajā nodaļā BPL patrulēšanas misija tiek modelēta kā *Gordon-Newell* rindas tīkls, lai analizētu sistēmas dinamiskās īpašības, identificētu vājās vietas, modelētu, kā MI veiktspēja ietekmē misijas GVR, un salīdzinātu centrālā MI mezgla uzlabojumu efektu ar iebūvētā MI uzlabojumiem.

4.1. apakšnodaļā tiek uzsvērta BPL redzamības vienkāršošanas problēma simulācijā. Draudi simulācijā tika modelēti kā 2D *UTF-16* simboli, un to noteikšanas process bija salīdzinoši vienkāršs – 30 × 30 rakstzīmju masīvs tika nodots BPL draudu noteikšanas modulim, un, ja tika atrasts draudu simbols, drauds tika uzskatīts par atklātu. Lai gan šī pieeja slikti atbilst reālos apstākļos sastopamajiem nosacījumiem, tā bija pietiekama šī pētījuma mērķiem. Reālajā vidē BPL būtu aprīkoti ar videokamerām un patrulējot uztvertu daudzus objektus. Lielākā daļa šo objektu – gan statiski, gan dinamiski – nebūtu reāli draudi. Lai nodrošinātu automātisku draudu noteikšanu, var izmantot dziļās mācīšanās modeļus. Šādu modeļu specifika pārsniedz šī pētījuma robežas. Tā vietā, pamatojoties uz esošo literatūru šajā jomā [24], [25], tiks pieņemti pieņēmumi par MI precizitāti, aparatūras prasībām un veiktspēju. MI pakalpojums šajā pētījumā tiks uztverts kā melnā kaste.

4. tabula

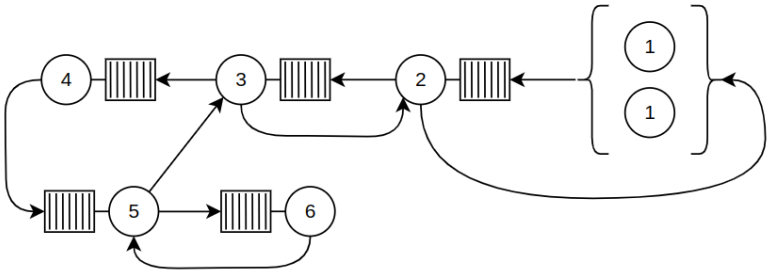
QN modeļa parametri

Mezgls	**Nosaukums**	$E[S]$ **(s)**	V_i	D_i
1	Laika intervāls starp BPL pieprasījumiem ($E[Z]$)	20	1	20 (3 obj/min)
2	Mobilo sakaru pārraide ($E[S_{\text{CellularTransmission}}]$)	0,03	1	0,03
3	Centrālā komandiera API ($E[S_{\text{CommanderRequest}}]$)	0,2	1	0,2
4	Centrālā MI ($E[S_{\text{CentralAI}}]$)	1,5	1	1,5
5	*WAN* pārraide ($E[S_{\text{WANTransmission}}]$)	0,3	0,001	0,0003
6	Cilvēka operatora analīze ($E[S_{\text{OperatorProcessing}}]$)	30	0,001	0,03

Šī pētījuma mērķis ir analizēt sistēmas struktūru, lai identificētu tās galvenos komponentus un iespējamos veiktspējas ierobežojumus. Otrkārt, tas cenšas noteikt sistēmas caurlaidspējas

teorētiskās robežas un analizēt atbildes laika sadalījumu, izmantojot rindas teoriju. Visbeidzot, pētījums skata, kuri sistēmas aspekti būtu jāoptimizē, piemēram, iebūvētā MI precizitāte vai centrālā MI atbildes laiks, lai sasniegtu iespējami labākos misijas GVR.

4.2. apakšnodaļā tiek definēts matemātiskais modelis. Katru sistēmas sastāvdaļu var attēlot kā rindu, un visu sistēmu var modelēt kā rindas tīklu. Šis tīkls apstrādā vienu uzdevuma veidu – pieprasījumu no BPL, lai noteiktu, vai konstatētais objekts ir reāls drauds. Pašreizējā scenārijā BPL skaits ir fiksēts, un tie darbojas kā lietotāji, kas mijiedarbojas ar koplietotu pakalpojumu, izsniedzot pieprasījumus ar vidējiem intervāliem. Tāpēc sistēma vislabāk tiek attēlota kā slēgts rindas tīkls, kā redzams 12. attēlā. *Gordon-Newell* rindas tīkli (*GNQN*), nosaukti to izgudrotāju vārdā, pārstāv slēgtu Markova rindas tīklu klasi, kur visiem apkalpošanas laikiem ir negatīvs eksponenciālais sadalījums. Šajā modelī tika strādāts ar *M*/*M*/1 rindām, kas ir savstarpēji saistītas, pamatojoties uz iepriekš noteiktām maršrutēšanas varbūtībām. Vidējais apkalpošanas laiks rindā *i* ir dots ar šādu izteiksmi $\mathbb{E}[S_i] = 1/\mu_i$.



12. attēls. Slēgts *Gordon-Newell QN* modelis kooperatīvam BPL tīklam.

Visi rindas tīkla (*QN*) parametri ir apkopoti 4. tabulā.

Lai aprēķinātu dažādus šī moduļa interesējošos rādītājus, tika izmantots Buzena konvolūcijas algoritms. Buzena konvolūcijas algoritms ir efektīva metode normalizācijas konstantes *G*(*K*) aprēķināšanai slēgtos rindas tīklos ar ierobežotu klientu skaitu *K*. Galvenā ideja ir rekursīvi aprēķināt *G*(*K*), izmantojot dinamiskās programmēšanas pieeju, izvairoties no tiešas summēšanas pār visiem iespējamajiem stāvokļiem. Ņemot vērā apkalpošanas ātrumus D_i katrai rindai *i* un klientu skaitu *K*, normalizācijas konstante tiek aprēķināta šādi:

$$G(M,K) = \sum_{\mathbf{n}\in\mathcal{I}(M,K)} \prod_{i=1}^{M} D_i^{n_i} \tag{6}$$

kur:

- *G*(*M*,*K*) – normalizācijas konstante slēgtam rindas tīklam ar *M* rindām un *K* kopējo klientu skaitu;
- $\mathcal{I}(M,K)$ apzīmē visu iespējamo stāvokļu vektoru kopu *n*;
- D_i pārstāv apkalpošanas pieprasījumu rindā *i*;
- n_i – klientu skaits rindā *i* stāvoklī *n*;
- produkts $\prod_{i=1}^{M} D_i^{n_i}$ novērtē katra stāvokļa *n* devumu normalizācijas konstantei.

Buzens pārformulēja šo summāciju kā iteratīvu atkārtošanās sakarību:

$$G(M,K) = G(M-1,K) + D_M\, G(M,K-1) \tag{7}$$

Lai iesāktu šo rekursiju, nepieciešamas sākuma vērtības. Ja tīklā ir tikai viena rinda, pēc definīcijas $G(1,k) = D_1^k, \quad \text{for all } k \in \mathbb{N}$. Ir tikai viens veids, kā sadalīt 0 klientus starp *M* rindām, tāpēc *G*(*M*,0) = 1 [32].

Īpašs gadījums rodas, analizējot rindas tīklu (*QN*), kas ietver vienu bezgalīgu serveru staciju. Šādā tīklā ar *K* klientiem bezgalīgo serveru staciju var uzskatīt par *K* serveru staciju (1. mezgls). Jāpieņem, ka tīklā ir tikai viena bezgalīga servera stacija; ja pastāv vairākas, tās var apvienot vienā ekvivalentā stacijā. Galvenā izmaiņa aprēķinu shēmā ietver sākotnējās inicializācijas pielāgošanu, lai ņemtu vērā bezgalīgā servera mezgla esamību. Stacionārā izkliede klientu skaitam šajā tīklā, ņemot vērā kopējo *K* klientu skaitu, ir sniegta darbā [32]:

$$\Pr\{N = n\} = \frac{1}{G(M,K)} \frac{D_1^{n_1}}{n_1!} \prod_{i=2}^{M} D_i^{n_i} \tag{8}$$

$$G(m,0) = 1, \quad \text{for } m = 1,\ldots,M, \tag{9}$$

$$G(1,k) = \frac{D_1^k}{k!}, \quad \text{for } k = 0,\ldots,K. \tag{10}$$

Vienīgā neatbilstība rindas tīklā parādās aprēķinu shēmas inicializācijā, savukārt pārējie aprēķini paliek nemainīgi (10).

Lai salīdzinātu rezultātus, kas iegūti ar Buzena algoritmu, var izmantot vidējās vērtības analīzes pieeju [32]. Bezgalīgo serveru mezgliem gaidīšanas laika nav, tāpēc atbildes laiks ir vienāds ar apkalpošanas laiku:

$$\begin{gathered} E[R_j(K)] = E[S_j], \quad E[\hat{R}_j(K)] = D_j \\ \mathbb{E}[\hat{R}_i(K)] = (\mathbb{E}[N_i(K-1)] + 1)\, D_i \\ E[N_i(K)] = X(K) E[\hat{R}_i(K)] \\ \sum_{i=1}^{M} E[N_i(K)] = K, \quad X(K) = \frac{K}{E[\hat{R}(K)]} \end{gathered} \tag{11}$$

Sagaidāmais atbildes laiks, klientu skaits un caurlaidspēja (11(11). Šie vienādojumi paplašina vidējās vērtības analīzi (*MVA*) uz bezgalīgo serveru (*IS*) mezgliem, saglabājot analītisko konsekvenci.

5. tabula

QN sastāvdaļu reakcijas laiks un caurlaidspēja pēc K

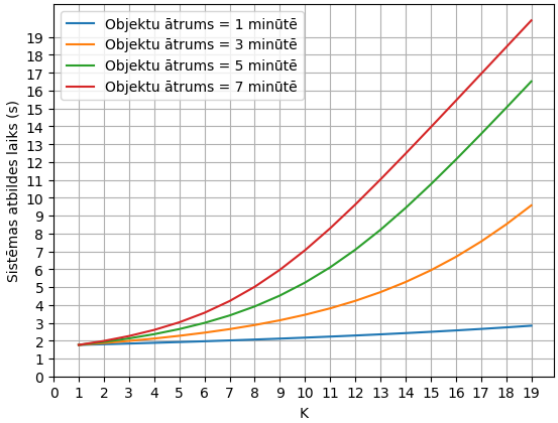
K	$E[R_1(K)]$	$E[R_2(K)]$	$E[R_3(K)]$	$E[R_4(K)]$	$E[R_5(K)]$	$E[R_6(K)]$	$X(K)$
1	20	0,03	0,20	1,50	0,00	0,03	0,05
4	20	0,03	0,21	1,85	0,00	0,03	0,18
7	20	0,03	0,21	2,37	0,00	0,03	0,31
10	20	0,03	0,22	3,17	0,00	0,03	0,43

5. tabulas turpinājums

13	20	0,03	0,22	4,43	0,00	0,03	0,53
16	20	0,03	0,23	6,40	0,00	0,03	0,60
19	20	0,03	0,23	9,28	0,00	0,03	0,64
22	20	0,03	0,23	13,02	0,00	0,03	0,66
25	20	0,03	0,23	17,27	0,00	0,03	0,67
28	20	0,03	0,23	21,72	0,00	0,03	0,67

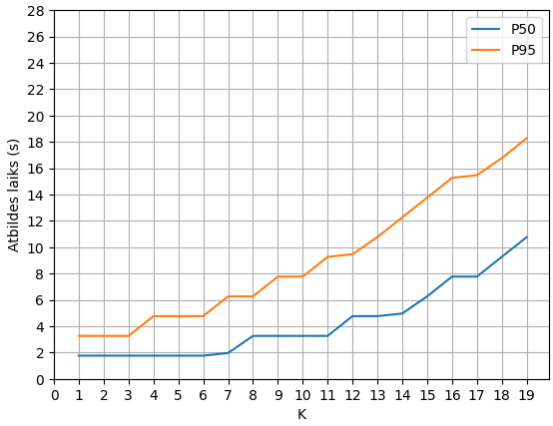
4.3. apakšnodaļā tiek apspriesti galvenie veiktspējas rādītāji, atbildes laika sadalījuma aprēķins un tas, kā kļūdainas MI klasifikācijas ietekmē misijas GVR.

5. tabulā apkopotas veiktspējas vērtības, kas aprēķinātas, izmantojot Buzena algoritmu, dažādiem BPL skaitiem (K) un objektu ātrumam 3/min. Visinteresantākā sastāvdaļa ir sistēmas vājā vieta, kas ir sastāvdaļa ar lielāko apkalpošanas laiku – 4. mezgls (centrālais MI). Sistēmas maksimālā caurlaidspēja tika sasniegta ap 28 BPL. Svarīgākās vērtības ir sistēmas atbildes laiks, noslodze un vidējais mezglu skaits rindā. Atsaucei, ja $K = 10$ BPL, vājākais posms ir 4. mezgls ar noslodzi 64 %, vidējo rindu 1,35 un atbildes laiku 3,17 s. 13. attēlā redzams, cik ātri atbildes laiks pieaug atkarībā no BPL skaita un dažādiem vidējiem pieprasījumu intervāliem.



13. attēls. Sistēmas reakcijas laiks atkarībā no K (bezpilota lidaparātu skaita). Funkciju ģimene pēc identificēto objektu ātruma.

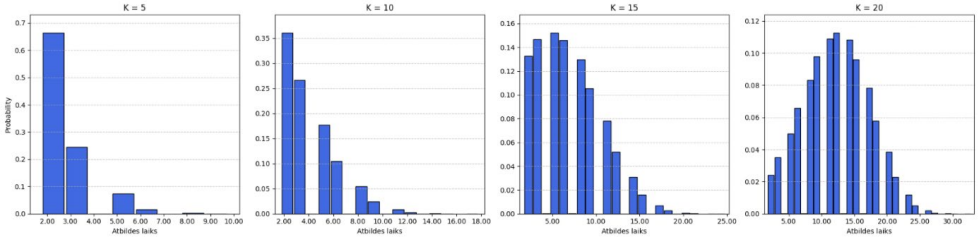
Veiktspējas novērtējums rāda, ka, palielinoties BPL skaitam (K), sistēmas reakcijas laiks pieaug nelineāri, īpaši pārsniedzot $K = 10$. Centrālā AI (4. mezgls) ir sistēmas šaurs kakls, sasniedzot gandrīz pilnu noslodzi (līdz 99,6 %), ja UAV skaits ir augsts. Tas izraisa strauju reakcijas laika un rindas izmēra pieaugumu.



14. attēls. Sistēmas reakcijas laiks atkarībā no *K* pēc 90 procentilēm un 95 procentilēm.

Bieži vien vidējais reakcijas laiks nav pietiekams rādītājs sistēmas nefunkcionālajām prasībām, jo tas neatspoguļo lietotāju uztverto kvalitāti. Tā vietā ir informatīvāk novērtēt laika diapazonu, kurā notiek 95 % reakciju. Ņemt vērā to, ka ir pieejams gan sagaidāmais reakcijas laiks katrai sistēmas stāvoklim, gan atbilstošās stāvokļu varbūtības, ir iespējams izveidot sistēmas reakcijas laika varbūtības sadalījuma funkciju (14. un 15. attēls). Tas nodrošina reālistiskāku sistēmas veiktspējas attēlojumu. Tā rezultātā, ja *K* vērtības ir zemākas, sadalījuma grafiki var neizskatīties reprezentatīvi. Lai uzlabotu interpretāciju, tika grupēti (sadalīti intervālos) daudzi zemas varbūtības reakcijas laiki 1 sekundes intervālos, kas rada informatīvākus un vieglāk lasāmus grafikus.

Šīs nodaļas mērķis ir tālāk analizēt iespējamos iebūvētā MI uzvedības scenārijus, modelēt šos scenārijus, iekļaut tos plašākā rindas tīkla (*QN*) struktūrā un novērtēt, kā tie ietekmē kopējo GVR.



15. attēls. Aprēķinātais atbildes laika varbūtības sadalījums.

BPL ir aprīkots ar pirmā slāņa MI programmatūru, kas spēj noteikt neparastus objektus savā redzamības laukā. Tika pieņemts, ka šī programmatūra darbojas gan precīzi, gan ātri. Nākamais uzdevums ir klasificēt katru atklāto objektu vienā no divām kategorijām – drauds vai nav drauds. Piemēram, BPL var novērot putnu lidojumā; šim objektam būtu pareizi jābūt identificētam kā nedraudīgam. Ja tas kļūdaini tiek klasificēts kā BPL, tas rada kļūdaini pozitīvu gadījumu un tiek izšķērdēts laiks putna izsekošanā, nevis patrulēšanas turpināšanā un reālu draudu identificēšanā.

Require: Response time PMF PMF, onboard accuracy A_0, threat ratio r, threat rate λ, simulation time T_{sim}
Ensure: Array of lost time fractions L
Set number of samples $S \leftarrow 5000$
$N \leftarrow \lfloor \lambda \cdot T_{\text{sim}} \rfloor$
$n_{\text{real}} \leftarrow \lfloor N \cdot r \rfloor$
$n_{\text{nonthreat}} \leftarrow \lfloor N \cdot (1 - r) \rfloor$
$n_{\text{fp}} \leftarrow \lfloor (1 - A_0) \cdot n_{\text{nonthreat}} \rfloor$
$n_{\text{fn}} \leftarrow \lfloor (1 - A_0) \cdot n_{\text{real}} \rfloor$
Extract (R, P) from PMF, where R is array of response times and P their probabilities
Normalize P: $P \leftarrow P / \sum P$
Sample $n_{\text{fp}} \times S$ values from R using P: FP_samples
$\text{wasted}_{\text{fp}} \leftarrow \sum$ FP_samples along each column
Sample $n_{\text{fn}} \times S$ values from R using P: FN_samples
$\text{wasted}_{\text{fn}} \leftarrow \sum$ FN_samples along each column
$\text{wasted}_{\text{total}} \leftarrow \text{wasted}_{\text{fp}} + \text{wasted}_{\text{fn}}$
$L \leftarrow \text{clip}(1 - \text{wasted}_{\text{total}} / T_{\text{sim}}, 0, 1)$
return L

16. attēls. Algoritms aprēķina izšķērdētā laika procentuālo daļu no kopējā simulācijas laika. Izšķērdētais laiks ir laiks, ko bezpilota lidaparāts pavadījis, sekojot nepareizam objektam, kas kļūdaini klasificēts.

Iespējamie iznākumi, kad BPL atklāj objektu, ir šādi.

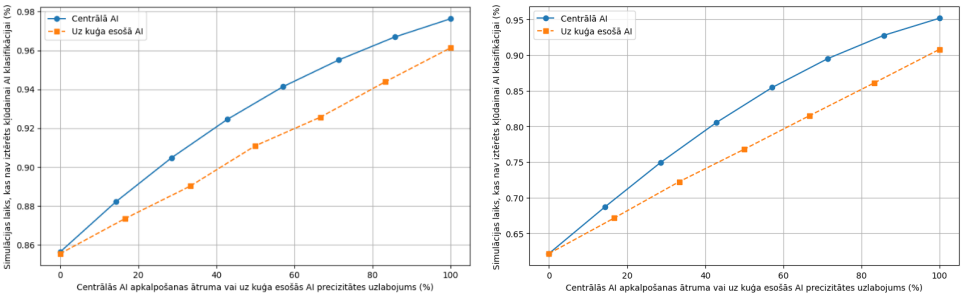
- Reāls drauds:
  - pareizi izseko draudu → iegūts GVR (īstais pozitīvais);
  - nespēj izsekot draudu → zaudēts GVR (īstais negatīvais).
- Nav drauds:
  - pareizi ignorē objektu → nav ietekmes uz GVR (īstais negatīvais);
  - kļūdaini izseko objektu → zaudēts GVR (kļūdaini pozitīvs).

Izmantojot šo klasifikācijas struktūru, var izveidot divas formulas, lai kvantitatīvi noteiktu kļūdaini negatīvo un kļūdaini pozitīvo gadījumu līmeni.

$$P(FalsePositive) = (1 - A_0) * (1 - r)$$
$$P(FalseNegative) = (1 - A_0) * r \quad (12)$$

kur: A_0 – iebūvētā MI precizitāte; r – reālo draudu īpatsvars starp visiem objektiem.

Promocijas darba mērķis ir novērtēt kopējā simulācijas laika daļu, kas tiek zaudēta iebūvētā MI kļūdainu klasifikāciju dēļ. Pakāpeniski uzlabojot vai nu iebūvētā MI precizitāti, vai centrālā AI veiktspēju, var ģenerēt grafikus, kas parāda zaudētā laika daļu kā funkciju no katra uzlabojuma. Šie grafiki (17. attēls) palīdzēs noteikt, vai ieguldījumiem jādod priekšroka iebūvētā MI precizitātes uzlabošanā, vai centrālā MI atbildes laika samazināšanā. Tālāk redzamajos grafikos ir salīdzinātas abas stratēģijas. Pieaugot objektu noteikšanas ātrumam, plaisa starp abām līknēm kļūst izteiktāka.



17. attēls. Veiktspējas salīdzinājums starp iebūvētā mākslīgā intelekta precizitāti un centrālā mākslīgā intelekta apstrādes laiku. Objekta biežums ir 3/min, izmantojums 78 % (pa kreisi), un 5/min un 95 % (pa labi).

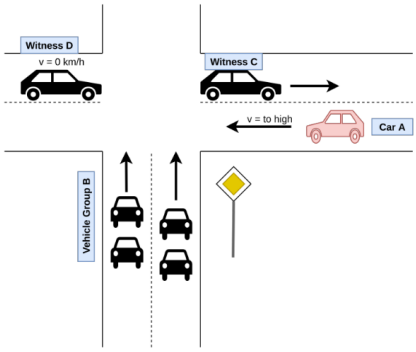
Šajā nodaļā tika ieviests veiktspējas modelēšanas ietvars kooperatīvām BPL draudu noteikšanas misijām, izmantojot slēgtu *Gordon-Newell* rindas tīklu (*QN*). Modelis atspoguļo hibrīdās MI arhitektūras lēmumu pieņemšanas plūsmas dinamisko uzvedību, kurā objektu klasifikācijā piedalās gan iebūvētais, gan centrālais MI, un retos gadījumos tiek piesaistīts cilvēka operators. Lai modelētu iebūvētā MI klasifikācijas uzvedību, tika izstrādāts lineārs modelis, lai novērtētu kļūdainu noteikumu skaitu. Šis modelis tika izmantots, lai aprēķinātu vidējo gaidīšanas laiku, ko BPL izšķērdē, gaidot pakalpojuma atbildi, kas koriģē tā darbību.

Tika konstatēts, ka, uzlabojot iebūvētā MI precizitāti, zaudētais laiks samazinās lineāri. Savukārt, uzlabojot centrālā AI apkalpošanas ātrumu, laika samazinājums ir vēl lielāks un seko logaritmiskam raksturlielumam, pat tad, ja centrālā MI noslodze ir ievērojami zem 100 %. Jo augstāka ir servera noslodze, jo lielāka ir veiktspējas uzlabojumu ietekme uz zaudētā laika samazināšanu.

Piektā nodaļa

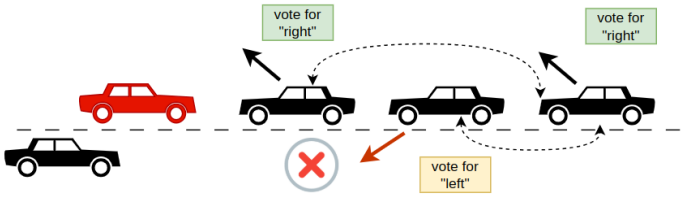
Šajā nodaļā tiek aprakstīta pētītā konsensa tēma transportlīdzekļu tīklos. Sākotnēji tiek apskatīts Bizantijas kļūdu modelis, analizējot tā prasības un pārskatot esošās ieviešanas pieejas. Diskusija pēc tam tiek pārorientēta uz praktiskāku avāriju apturošo konsensa modeli, uzsverot biežākos lietošanas gadījumus un veiktspējas izaicinājumus. Nobeigumā tiek piedāvāta metode plūdu balstīta konsensa veiktspējas modelēšanai dažādās tīkla topoloģijās, izmantojot *NS-3* simulācijas, ko papildina analītiskas tuvinājuma metodes.

5.1. apakšnodaļā aprakstīti kooperatīvie viedo transportlīdzekļu lietošanas gadījumi, kuros var izmantot konsensu. 18. attēls ilustrē kooperatīvu sadursmes novēršanu krustojumā. Lai novērstu potenciālu negadījumu, tiek apsvērti trīs teorētiski pieejas varianti, ko inteliģentie transportlīdzekļi varētu izmantot: (1) katrs transportlīdzeklis patstāvīgi pieņem lēmumu par to, kā reaģēt uz situāciju; (2) katrs transportlīdzeklis pieņem lēmumu, savukārt pēc tam notiek balsošanas process; grupā tiek izvēlēta un īstenota stratēģija ar visvairāk balsīm; (3) tiek izpildīts konsensa algoritms, kas ļauj visiem grupas transportlīdzekļiem vienoties par vienotu, koordinētu stratēģiju.



18. attēls. Avārijas sadursmes novēršanas vienprātības izmantošanas piemēri *V2V* sistēmā.

Otrais scenārijs redzams attēlā. Šajā gadījumā transportlīdzekļu kolonna brauc pa automaģistrāli, kad pēkšņi priekšā tiek konstatēts ātri braucošs transportlīdzeklis. Lai gan var būt vairākas stratēģijas, kā izvairīties no sadursmes, problēmas būtība paliek tāda pati kā pirmajā scenārijā – transportlīdzekļiem ir jāpieņem koordinēts lēmums, balstoties daļējā novērojumu kopumā. Galvenā atšķirība ir transportlīdzekļu topoloģijā, proti, kā tie ir izvietoti un savienoti, un tā ir šī pētījuma galvenā pētniecības interese.



19. attēls. Vienprātības panākšana kolonnā par izvairīšanos no avārijas ārkārtas situācijā.

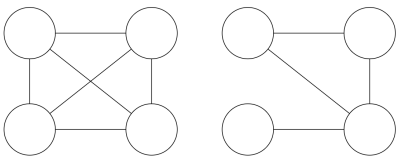
5.2. apakšnodaļā tiek analizēts, vai Bizantijas avāriju kļūmes ir nozīmīgas transportlīdzekļu scenārijos. Pastāv izplatīts maldīgs uzskats, ka Bizantijas kļūmes ir tikai teorētiskas un nav vērts tās apsvērt reālās sistēmās [33]. Pēdējos gados ir strauji attīstījušās decentralizētas arhitektūras, tostarp *IoT* tīkli un *VANET* [34]. Tomēr ierobežotā uzmanība, kas tiek pievērsta *BFT V2V* sistēmās, joprojām ir spekulatīva. Viens no iespējamiem iemesliem ir plaši izplatītais uzskats, ka pietiek ar šifrēšanu, digitālajiem parakstiem un *PKI*, lai aizsargātu sistēmu [35]. Piemēram, *C-ITS* standarti plaši koncentrējas uz *PKI* konfigurācijām, taču vispār neapskata defekttoleranci [4].

Pretstatā tam tādās nozarēs kā aviācija ne tikai tiek atzīta *BFT* nozīmība, bet arī aktīvi tiek pētīti un integrēti *BFT* risinājumi sistēmās [36]. Kā norādīts darbā [33], viens no iemesliem, kāpēc *BFT* bieži tiek ignorēta tradicionālajā sistēmu izstrādē, ir pieņēmums "ja es to neesmu redzējis, tas nepastāv".

Darbā [37] autori apgalvo, ka *VANET* uzticamību var uzlabot, piemērojot *BFT* algoritmus lēmumu pieņemšanas procesiem *adhoc* tīklos. Turklāt viņi iesaka izmantot klasteru topoloģijas kā stratēģiju sastrēgumu problēmu risināšanai. Darbā [38] autori piedāvā *VANET* protokolu,

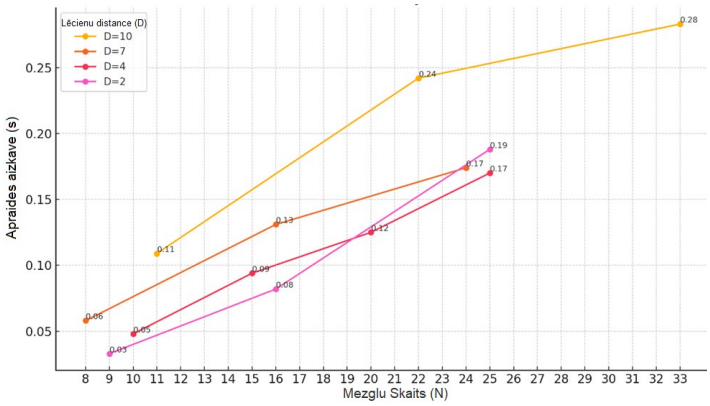
kas prioritizē lietotāju privātumu un ievieš atlīdzībā balstītu notikumu ziņojumu izplatīšanu. Viņi izceļ divus galvenos izaicinājumus: "Pirmkārt, ir grūti pārsūtīt uzticamus paziņojumus, neatklājot lietotāju identitātes. Otrkārt, lietotājiem parasti trūkst motivācijas pārsūtīt paziņojumus." Piedāvātais risinājums ir privātumu saglabājošs paziņojumu protokols, kas izmanto sliekšņa parakstus, lai aizsargātu lietotāju identitātes. Darbā [39] autori piedāvā jaunu atbilstības pierādījuma (*Proof-of-Eligibility*) mehānismu, lai panāktu konsensu *VANET* tīklos. Galvenā ideja ir ierobežot dalību konsensā ar mezglu apakškopu, kas tieši iesaistīti konkrētā notikumā. Šī pieeja ir paredzēta, lai novērstu mezglu ar īsu saziņas diapazonu spēju apdraudēt konsensa procesu, palielinot iespēju pārsniegt viena trešdaļas ļaunprātīgo dalībnieku robežu, kā noteikts Bizantijas kļūdu modelī [40]. Darbā [41] autori pievēršas problēmai, kas saistīta ar viltus datu injicēšanu transportlīdzekļu tīkla notikumu ziņojumos. Lai gan ir ierosināti daudzi mehānismi, lai uzlabotu *VANET* drošību, daudzi joprojām ir neaizsargāti pret nepatiesas informācijas ievadi, kas var tikt izmantota citu vadītāju uzvedības manipulēšanai. Lai to mazinātu, autori piedāvā algoritmu, kas balstīts attiecināmības pierādījuma (*Proof of Relevance*, *PoR*) koncepcijā, kuras mērķis ir pierādīt, ka ziņojošais mezgls patiešām ir saistīts ar ziņoto notikumu.

5.3. apakšnodaļā tiek pievērsta uzmanība dažiem svarīgiem jautājumiem, kas saistīti ar kļūmes apturēšanas konsensa izmantošanu *V2V* saziņā ārkārtas situācijās. Pirmkārt, vai kļūmes apturēšanas konsensa algoritms var darboties efektīvi steidzamos *V2V* scenārijos, kuros transportlīdzekļiem ir ātri jāvienojas par rīcību. Otrkārt, kā transportlīdzekļu skaits tīklā ietekmē algoritma darbības ātrumu un efektivitāti. Treškārt, kā tīkla topoloģija ietekmē algoritma veiktspēju.



20. attēls. Pilnais grafs (pa kreisi) un savienotais grafs (pa labi). Malas nozīmē, ka ir ierobežots uzticams savienojums, bet trūkstošā mala nozīmē, ka savienojuma nav vispār vai piegāde aizkavējas.

Plūdu konsensa algoritms [12] (*Flooding consensus*) nodrošina šādas garantijas: (1) pabeigšana (*Termination*) – katrs pareizi funkcionējošs process pieņem kādu vērtību; (2) derīgums (*Validity*) – ja process pieņem vērtību v, tad v ir jābūt kāda procesa piedāvātai vērtībai; (3) integritāte (*Integrity*) – neviens process nepieņem lēmumu vairāk nekā vienu reizi; (4) vienošanās (*Agreement*) – nevienam no pareizi funkcionējošiem procesiem nevajadzētu pieņemt atšķirīgas vērtības. Algoritms darbojas, pieņemot, ka ir pieejams perfekts bojājumu indikators. Katrā kārtā algoritms rada $O(N^2)$ ziņojumu sarežģītību. Turklāt, kad process pieņem lēmumu, tas izsūta $O(N^2)$ *DECIDED* ziņojumus. Katrai papildu kārtai, ko izraisa avārija, notiek $O(N^2)$ ziņojumu apmaiņa. Līdz ar to sliktākajā gadījumā kopējais ziņojumu skaits var pieaugt līdz $O(N^3)$ [12].

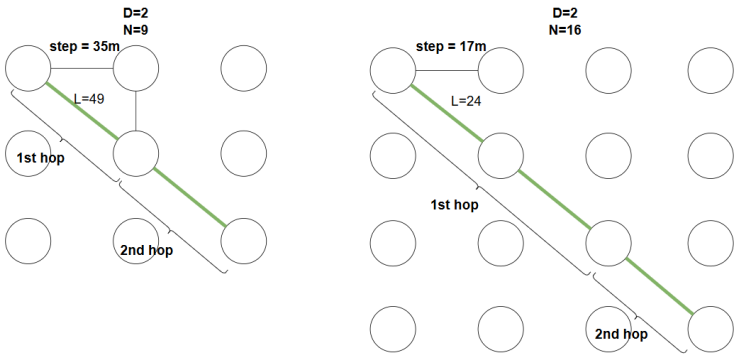


21. attēls. Vidējā "visi-visiem" raidīšanas aizkavē, kas apkopota no simulācijas datiem.

Šajā modelī tika simulēts konsenss starp *V2V* mezgliem, attēlojot tīklu kā savienotu grafiku, kur katrs mezgls pārraida ziņojumus, kas tiek izplatīti caur režģi. Tiek iekļautas tikai tās saites, kuru aizkave ir zem iepriekš noteikta reāllaika sliekšņa; saites, kuru aizkave pārsniedz šo slieksni, tiek uzskatītas par neesošām. Šāda pieeja modelē reāllaika saziņas ierobežojumus (20. attēls).

Lai novērtētu pārraides izplatīšanās veiktspēju bezvadu režģa tīklos, tika izstrādāta pielāgota simulācija, izmantojot *NS-3* tīkla simulatoru. Simulācija modelē režģa bāzētu tīkla topoloģiju un ievieš pielāgotu plūdu algoritmu, lai noteiktu laiku, kas nepieciešams pilnīgai ziņojuma izplatīšanai.

Tā kā *UDP* nesniedz piegādes garantiju, tiek ieviesta redundance, izmantojot nejaušas atkārtotas pārraides. Katrs mezgls ieplāno konfigurējamu raidījumu skaitu, katru no tiem atdalot ar nejaušu aizkavi no 1 ms līdz lietotāja noteiktajam maksimālajam laikam. Šīs aizkaves palīdz samazināt traucējumus. attēlā redzami konfigurāciju piemēri, kas atbilst izvēlētajām mezglu skaita un grafu diametra kombinācijām.



22. attēls. Piemēri, kā konfigurēt dažādas *N* un *D* kombinācijas, izmantojot mainīgo soli. Sakaru diapazons ir konstants (50 m).

Simulācijā tiek izmantots *UDP* protokols bez piegādes apstiprinājumiem. Ņemot vērā ārkārtas konsensa stingrās laika prasības – parasti pilnīgai vienošanās sasniegšanai jānotiek zem 500 ms – simulācijas mērķis ir identificēt apraides konfigurācijas, kas statistiski garantē piegādi. Šajā modelī, kad mezgli izpilda konsensu, katrs mezgls vairākas reizes (piemēram, trīs reizes) izplata savu ziņojumu, starplaikos ievērojot nejaušas aizkaves. Paredzams, ka ar lielu varbūtību ziņojums veiksmīgi tiks piegādāts visiem pārējiem mezgliem. Ja mezgls nesaņem ziņojumu šajā periodā, tas tiek uzskatīts par avarējušu.

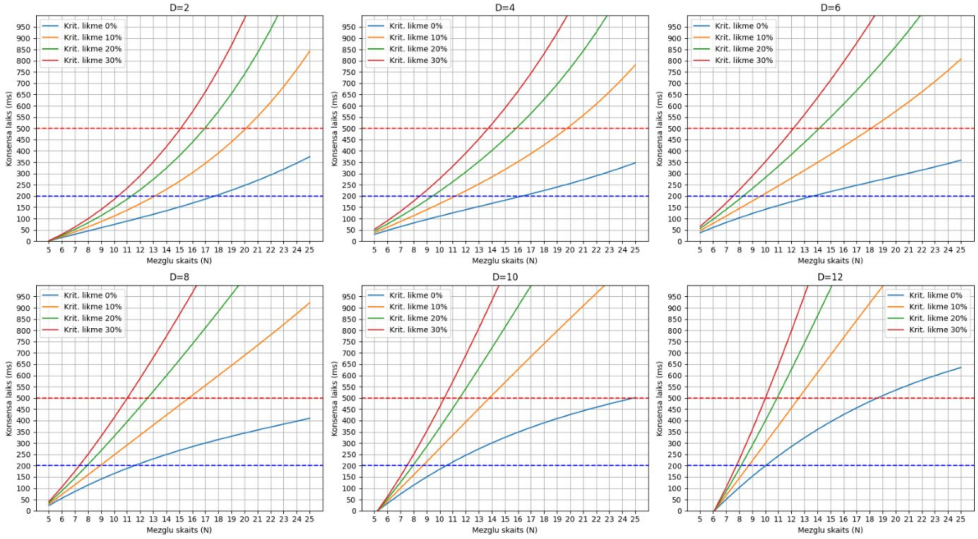
6. tabulā apkopoti statistiski uzticami "visi-visiem" vidējie apraides laiki dažādiem savienojamības grafiku diametriem un mezglu skaitiem. Atbilstošie neapstrādātie dati redzami 21. attēlā. Topoloģijas konfigurācijas elastības ierobežojumu dēļ katram diametram varēja pārbaudīt tikai trīs līdz četrus atšķirīgus mezglu skaitus. Iegūtā "visu-mezglu" apraides aizkaves analītiskā aptuvenā vērtība, pamatojoties uz simulācijas datiem, tiek izteikta ar trešās pakāpes polinomu.

6. tabula

Visi-visiem vidējais apraides pabeigšanas laiks dažādiem mezglu skaitiem un diametriem

***D* = 10**		***D* = 7**		***D* = 4**		***D* = 2**	
N	***T* (ms)**	***N***	***T* (ms)**	***N***	***T* (ms)**	***N***	***T* (ms)**
11	109	**8**	58	**10**	48	**9**	33
N/A	N/A	**16**	131	**15**	94	**16**	82
22	242	N/A	N/A	**20**	125	N/A	N/A
33	283	**24**	174	**25**	170	**25**	188

Izmantojot analītisko funkciju, kas modelē aizkavi vienam saziņas solim (t. i., "visi-visiem" apraide, kas ietver $O(N^2)$ ziņojumus), var novērtēt kopējo konsensa aizkavi kā sistēmas avāriju līmeņa funkciju. Tas ļauj ģenerēt aizkaves novērtējumus plašā scenāriju diapazonā – no zema diametra topoloģijām līdz augsta avāriju līmeņa apstākļiem (23. attēls).



23. attēls. Aptuvenais konsensa laiks atkarībā no mezglu skaita un grafika diametra.

Pētījuma rezultāti pierāda, ka ir iespējams ieviest statistiski uzticamu "visi-visiem" apraidi *V2V* režģa tīklos, izmantojot *NS-3* simulāciju, *HWMP*, *UDP*, nejaušus atkārtojumus un savienotu grafiku topoloģijas. Simulācijas parādīja, ka līdz pat 33 mezgliem ar savienojamības grafika diametru 10 uzticama apraide var tikt nodrošināta mazāk nekā 300 ms laikā. Lietojot "visi-visiem" aizkaves funkcijas analītisko tuvinājumu, kļūst iespējams ekstrapolēt sagaidāmo aizkavi sinhronam plūdu balstītam konsensam. Pat sarežģītos apstākļos, piemēram, kolonnā ar diametru 10, deviņiem mezgliem un 30 % avāriju līmeni, konsensu joprojām var sasniegt mazāk nekā 350 ms laikā. Uzticamas apraides aizkaves līknes forma ir atkarīga no saziņas grafika diametra: ja $D = 2$, aizkave pieaug pozitīvi līdz ar mezglu skaitu. Ja $D \geq 7$, pieaugums sāk samazināties. Ja $D = 4$, tiek novērota izteiktāka lineāra tendence. Piemēram, ja $N = 9$ un 20 % avāriju līmeņa konsensa aizkave ir 120 ms, ja $D = 2$ un pieaug līdz 250 ms, ja $D = 7$, kas atspoguļo 108 % pieaugumu.

PROMOCIJAS DARBA GALVENIE REZULTĀTI

Promocijas darba izstrādes laikā tika formulēti šādi galvenie secinājumi.

1. Novērtējot defekttolerantu algoritmu ietekmi, ieteicams vispirms izstrādāt optimālu lēmumu pieņemšanas loģiku, jo tā būtiski ietekmē misijas GVR pat scenārijos ar plašiem bojājumiem.
2. Reāllaika simulācija ir nenovērtējama kļūdu identificēšanai, kas citādi parādītos reālas izvietošanas laikā, kad to novēršana var būt ievērojami dārgāka. Galvenais reāllaika simulācijas trūkums ir izpildes laiks. Hibrīdā pieeja – diskrētas simulācijas izmantošana kopējās veiktspējas testēšanai un reāllaika simulācija sarežģītiem gadījumiem projekta vēlākajos posmos – var palīdzēt sasniegt maksimālu defekttoleranci un izturību.
3. Simulācijas dati rāda, ka kumulatīvais kļūdainu avāriju noteikumu skaits, ko izraisa "izrietošais" līdera indikators simulācijas laikā, ievēro logaritmisku modeli. Palielinoties izvietoto BPL skaitam, laiks līdz pēdējai kļūdainai avārijas noteikšanai arī ievērojami pieaug. Ar vairāk nekā pieciem BPL un lielāku noteikšanas pauzi (*DT*) kļūdainas avāriju noteikšanas var turpināties līdz 10 minūtēm. Tomēr, pateicoties šīs uzvedības logaritmiskajam raksturam, lielākā daļa kļūdainu avāriju notiek ātri, un tikai neliela daļa no kopējām kļūmēm ietekmē GVR.
4. Kooperatīvās BPL perimetra patrulēšanas misiju simulācijās kvantitatīvi GVR ir vērtīgi, lai salīdzinātu līdera izvēles algoritmu veiktspēju un noteiktu, vai izrietoša konsistences algoritmi ir piemēroti kooperatīvo BPL scenārijiem.
5. Algoritmā "ideālā līdera indikators" avāriju noteikšanas pauzes pagarināšana no 15 sekundēm (ideāli) līdz 60 sekundēm izraisa GVR samazinājumu par 12,1 % piecu minūšu simulācijā, ja GVR avāriju līmenis ir 30 %. Algoritms "izrietošais ideālā līdera indikators", konfigurēts ar piecu sekunžu bāzes pauzi un trīs sekunžu delta, uzrāda 4,1 % GVR zaudējumu, salīdzinot ar "ideālo" algoritmu ar ideālu pauzi. Būtiski, ka "izrietoši ideālais" algoritms pārspēj "ideālo" algoritmu reālos apstākļos (60 s pauze), nodrošinot 9,7 % GVR uzlabojumu, ja avāriju līmenis ir 30 %.
6. BPL uzraudzības tīklu, kas izmanto MI bāzētu draudu noteikšanu, var modelēt, izmantojot slēgtu *Gordon-Newell* rindas tīklu. Kamēr iebūvētā MI precizitātes uzlabojumi lineāri samazina laiku, kas zaudēts kļūdainai pozitīvai uzraudzībai, centrālā MI atbildes laika samazināšanai ir lielāka ietekme uz zaudētā uzraudzības laika samazināšanu, pieaugot sistēmas noslodzei.
7. Bizantijas kļūdu modelis ir plaši izplatīts avionikā un kosmosa inženierijā. Tomēr šis termins netiek pieminēts *ETSI ITS* tehniskajos ziņojumos. *ITS* literatūras apskats atklāj, ka Bizantijas defekttolerantie algoritmi, kas ierosināti *V2X* sistēmām, galvenokārt koncentrējas uz privātumu, blokķēžu arhitektūrām un reputācijas mehānismiem.
8. Pielāgots nejaušināts, *UDP* bāzēts uzticams "visi-visiem" apraides algoritms, izmantojot *802.11s* sieta tīklu un testēts *NS-3* simulācijā, uzrāda vidējo aizkavi 280 ms 33 mezgliem ar savienojamības grafu diametru $D = 10$. Mazākiem diametriem un mezglu skaitiem vidējās aizkaves ir zem 100 ms.

9. Matemātiskā modelēšana parāda, ka kļūmes apturošais plūdu konsensa algoritms, balstīts minētajā apraides algoritmā, teorētiski var sasniegt konsensu mazāk nekā 500 ms laikā, ja $D = 8$, $N = 11$, ar līdz pat 30 % mezglu avārijām un mazāk nekā 200 ms laikā mazākiem diametriem. Uzticamās apraides aizkaves līknes forma ir atkarīga no saziņas grafiku diametra: ja $D = 2$, aizkave pieaug līdz ar mezglu skaitu, bet, ja $D \geq 7$, pieauguma ātrums samazinās. Vairāk lineāra tendence tiek novērota, ja $D = 4$.

Promocijas darbā ir apkopoti pabeigtā pētījuma rezultāti un noteikti **iespējamie nākotnes pētījumu virzieni**.

- Izstrādātās programmatūras testēšana perimetra patrulēšanas pētījumam, izmantojot reālus BPL, mobilo tīklu un *NAT* šķērsošanas sistēmu, kas arī tika izstrādāta pētījuma gaitā.
- Simulācijas iespēju paplašināšana, piemēram, nodrošinot simulāciju, kas nenotiek reāllaikā, ieviešot sarežģītākas koordinācijas sistēmas un nodrošinot reālistiskāku tīkla simulāciju.

BIBLIOGRĀFIJA

[1] L. Lamport, «Paxos Made Simple,» *ACM SIGACT News (Distributed Computing Column) 32, 4 (Whole Number 121, December 2001),* p. 51–58, 2001.

[2] H. Ng, S. Haridi un P. Carbone, «Omni-Paxos X: Breaking the Barriers of Partial Connectivity,» %1 *Proceedings of the Eighteenth European Conference on Computer Systems*, 2023.

[3] P. Kumari un P. Kaur, «A Survey of Fault Tolerance in Cloud Computing,» *Journal of King Saud University - Computer and Information Sciences,* sēj. 33, nr. 10, p. 1159–1176, 2021.

[4] E. European Telecommunications Standards Institute, «Intelligent Transport Systems (ITS); Security; Threat, Vulnerability and Risk Analysis (TVRA),» ETSI, Sophia Antipolis Cedex, 2017.

[5] D. C. Gandolfo, L. R. Salinas, M. E. Serrano un J. M. Toibero, «Energy Evaluation of Low-Level Control in UAVs Powered by Lithium Polymer Battery,» *ISA Transactions,* sēj. 71, p. 563–572, 2017.

[6] S. Jung, Y. Jo un Y.-J. Kim, «Flight Time Estimation for Continuous Surveillance Missions Using a Multirotor UAV,» *Energies,* sēj. 12, nr. 5, p. 867, 2019.

[7] J. Hu, H. Niu, J. Carrasco, B. Lennox un F. Arvin, «Fault-Tolerant Cooperative Navigation of Networked UAV Swarms for Forest Fire Monitoring,» *Aerospace Science and Technology,* sēj. 123, p. 107494, 2022.

[8] D. Yu, H. Wu, Y. Sun, L. Zhang un M. Imran, «Adaptive protocol of raft in wireless network,» *Ad Hoc Networks,* sēj. 154, p. 103377, 2024.

[9] M. Saadoon, S. H. Ab Hamid, H. Sofian, H. H. Altarturi, Z. H. Azizul un N. N. Mohd Daud, «Fault Tolerance in Big Data Storage and Processing Systems: A Review on Challenges and Solutions,» *Ain Shams Engineering Journal,* sēj. 13, nr. 2, p. 101538, 2021.

[10] Microsoft, «Transactional Outbox pattern with Azure Cosmos DB,» 2021. [Tiešsaiste]. Available: https://learn.microsoft.com/en-us/azure/architecture/best-practices/transactional-outbox-cosmos.

[11] P. Deutsch, *Falacies of distributed computing.*

[12] C. Cachin, R. Guerraoui un L. Rodrigues, Introduction to Reliable and Secure Distributed Programming, Berlin: Springer, 2011.

[13] M. Herlihy un N. Shavit, The Art of Multiprocessor Programming, San Francisco: Morgan Kaufmann, 2008.

[14] N. A. Lynch, Distributed Algorithms, San Francisco: Morgan Kaufmann, 1996.

[15] L. Lamport, «The Part-Time Parliament,» %1 *Concurrency: the Works of Leslie Lamport*, 2019, p. 277–317.

[16] N. C. Strole, «The IBM Token-Ring Network—A Functional Overview,» *IEEE Network,* sēj. 1, nr. 1, p. 23–30, 1987.

[17] H. Garcia-Molina, «Elections in a distributed computing system,» *IEEE Transactions on Computers,* sēj. 31, nr. 01, pp. 48-59, 1982.

[18] D. Ongaro un J. Ousterhout, «The Raft Consensus Algorithm,» *Lecture Notes CS,* sēj. 190, p. 2022, 2015.

[19] T. Feng, W. Fan, J. Tang un W. Zeng, «Consensus-based robust clustering and leader election algorithm for homogeneous UAV clusters,» %1 *Journal of Physics: Conference Series*, 2019.

[20] Y. Zou, L. Yang, G. Jing, R. Zhang, Z. Xie, H. Li un D. Yu, «A survey of fault tolerant consensus in wireless networks,» *High-Confidence Computing,* sēj. 4, nr. 2, p. 100202, 2024.

[21] S. Mousavi, F. Afghah, J. D. Ashdown un K. Turck, «Leader-follower based coalition formation in large-scale UAV networks, a quantum evolutionary approach,» %1 *IEEE INFOCOM 2018-IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, 2018.

[22] R. Ganesan, X. M. Raajini, A. Nayyar, P. Sanjeevikumar, E. Hossain un A. H. Ertas, «BOLD: Bio-inspired optimized leader election for multiple drones,» *Sensors (Basel, Switzerland),* sēj. 20, nr. 11, p. 3134, 2020.

[23] G. Wang, B.-S. Lee, J. Y. Ahn un G. Cho, «A UAV-Aided Cluster Head Election Framework and Applying Such to Security-Driven Cluster Head Election Schemes: A Survey,» *Security and Communication Networks,* sēj. 2018, nr. 1, p. 6475927, 2018.

[24] H. C. Baykara, E. Bıyık, G. Gül, D. Onural, A. S. Öztürk un I. Yıldız, «Real-time detection, tracking and classification of multiple moving objects in UAV videos,» %1 *2017 IEEE 29th International Conference on Tools with Artificial Intelligence (ICTAI)*, 2017.

[25] F. Cagatay Akyon, E. Akagunduz, S. Onur Altinuc un A. Temizel, «Sequence Models for Drone vs Bird Classification,» *arXiv e-prints,* pp. arXiv-2207, 2022.

[26] M. Mozaffari, W. Saad, M. Bennis, Y.-H. Nam un M. Debbah, «A Tutorial on UAVs for Wireless Networks: Applications, Challenges, and Open Problems,» *IEEE Communications Surveys Tutorials,* sēj. 21, nr. 3, p. 2334–2360, 2019.

[27] I. Jawhar, N. Mohamed, J. Al-Jaroodi, D. P. Agrawal un S. Zhang, «Communication and Networking of UAV-based Systems: Classification and Associated Architectures,» *Journal of Network and Computer Applications,* sēj. 84, p. 93–108, 2017.

[28] L. Gupta, R. Jain un G. Vaszkun, «Survey of important issues in uav communication,» *IEEE Communications Surveys Tutorials,* 2016.

[29] G. Halkes un J. Pouwelse, «UDP NAT and Firewall Puncturing in the Wild,» %1 *NETWORKING 2011*, Berlin, Heidelberg, Springer Berlin Heidelberg, 2011, p. 1–12.

[30] T. Shima un S. Rasmussen, UAV cooperative decision and control: challenges and practical approaches, SIAM, 2009.

[31] N. Hao, H. Yi, C. Tian, H. Yao un F. He, «A Distributed-Centralized Dynamic Task Allocation Algorithm for UAVs Tracking Moving Targets,» %1 *2021 40th Chinese Control Conference (CCC)*, 2021.

[32] B. R. Haverkort, Performance of Computer Communication Systems: A Model-Based Approach, Chichester: John Wiley and Sons, 1998.

[33] K. Driscoll, B. Hall, M. Paulitsch, P. Zumsteg un H. Sivencrona, «The Real Byzantine Generals,» %1 *23rd Digital Avionics Systems Conference (DASC)*, 2004.

[34] R. G. Engoulou, M. Bellaïche, S. Pierre un A. Quintero, «VANET Security Surveys,» *Computer Communications,* sēj. 44, pp. 1-13, 2014.

[35] V. Lozupone, «Analyze Encryption and Public Key Infrastructure (PKI),» *International Journal of Information Management,* sēj. 38, nr. 1, pp. 42-44, 2018.

[36] Y. Yeh, «Safety Critical Avionics for the 777 Primary Flight Controls System,» %1 *20th Digital Avionics Systems Conference (DASC)*, 2001.

[37] S.-C. Wang, Y.-J. Lin un K.-Q. Yan, «Reaching Byzantine Agreement Underlying VANET,» *KSII Transactions on Internet and Information Systems,* sēj. 13, nr. 7, p. 3351–3368, 2019.

[38] L. Li, J. Liu, L. Cheng, S. Qiu, W. Wang, X. Zhang un Z. Zhang, «CreditCoin: A Privacy-Preserving Blockchain-Based Incentive Announcement Network for Communications of Smart Vehicles,» *IEEE Transactions on Intelligent Transportation Systems,* sēj. 19, nr. 7, p. 2204–2220, 2018.

[39] H. Liu, C.-W. Lin, E. Kang, S. Shiraishi un D. M. Blough, «A Byzantine-Tolerant Distributed Consensus Algorithm for Connected Vehicles Using Proof-of-Eligibility,» %1 *22nd International ACM Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems (MSWiM)*, 2019.

[40] L. Lamport, M. Pease un R. Shostak, «The Byzantine generals problem».

[41] Z. Cao, J. Kong, U. Lee, M. Gerla un Z. Chen, «Proof-of-Relevance: Filtering False Data via Authentic Consensus in Vehicle Ad-hoc Networks,» %1 *IEEE INFOCOM Workshops*, 2008.

[42] M. Podhradský, J. Bone, A. Jensen un C. Coopmans, «Small Low-Cost Unmanned Aerial Vehicle Lithium-Polymer Battery Monitoring System,» %1 *ASME 2013 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, 2013.

Dmitrijs Rjazanovs dzimis 1994. gadā Rīgā. Ieguvis maģistra grādu (2018) Rīgas Tehniskās universitātes (RTU) studiju programmā "Intelektuālās robotizētās sistēmas". Profesionāli programmēšanā strādā kopš 2015. gada. Patlaban ir SIA "*C. T. Co*" programmatūras arhitekts un RTU lektors, docē kursus "Kriptogrāfija" un "Reālā laika e-komercija". Zinātniskās intereses – dalītās sistēmas, konsensa algoritmi, programmatūras inženierija.