RĪGAS TEHNISKĀ UNIVERSITĀTE

Mārtiņš Mihaeljans

ADAPTĪVO RISINĀJUMU NOVĒRTĒJUMS VIRTUALIZĒTO PROGRAMMDEFINĒTO TĪKLU VADĪBAI

Promocijas darba kopsavilkums

RTU Izdevniecība
Rīga 2025

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

Mārtiņš Mihaeljans
Doktora studiju programmas "Telekomunikācijas" doktorants

ADAPTĪVO RISINĀJUMU NOVĒRTĒJUMS VIRTUALIZĒTO PROGRAMMDEFINĒTO TĪKLU VADĪBAI

Promocijas darba kopsavilkums

Zinātniskais vadītājs
profesors *Dr. sc. ing.*
Jurģis Poriņš

Zinātniskais konsultants
Dr. sc. ing.
Andris Skrastiņš

RTU Izdevniecība
Rīga 2025

Mihaeljans, M. Adaptīvo risinājumu novērtējums virtualizēto programmdefinēto tīklu vadībai. Promocijas darba kopsavilkums. Rīga: RTU Izdevniecība, 2025. 36 lpp.

Publicēts saskaņā ar promocijas padomes "RTU P-08" 2025. gada 27. jūnija lēmumu, protokols Nr. 40.

Vāka attēls no *www.shutterstock.com.*

https://doi.org/10.7250/9789934372223
ISBN 978-9934-37-222-3 (pdf)

PROMOCIJAS DARBS IZVIRZĪTS ZINĀTNES DOKTORA GRĀDA IEGŪŠANAI RĪGAS TEHNISKAJĀ UNIVERSITĀTĒ

Promocijas darbs zinātnes doktora (*Ph. D.*) grāda iegūšanai tiek publiski aizstāvēts 2025. gada 14. novembrī plkst. 11.00 Rīgas Tehniskās universitātes Datorzinātnes, informācijas tehnoloģijas un enerģētikas fakultātē, Āzenes ielā 12, 201. auditorijā.

OFICIĀLIE RECENZENTI

Asociētais profesors *Dr. sc. ing.* Andis Supe,
Rīgas Tehniskā universitāte

Vadošais pētnieks *Dr. sc. comp.* Atis Elsts,
Elektronikas un datorzinātņu institūts, Latvija

Asociētais profesors *Ph. D.* Oskars Java,
Vidzemes Augstskola, Latvija

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.

Mārtiņš Mihaeljans ……. (paraksts)
Datums: ……

Promocijas darbs ir uzrakstīts angļu valodā, tajā ir ievads, astoņas nodaļas, secinājumi, literatūras saraksts, 40 attēli, astoņi pielikumi, kopā 134 lappuses, ieskaitot pielikumus. Literatūras sarakstā ir 68 nosaukumi.

ANOTĀCIJA

Promocijas darbs ir solis uz priekšu informācijas pasaules attīstībā. Darbs radies, izpētot vispārīgo tradicionālo tīklu principus, programmdefinēto tīklu (*SDN*) un tā iespējas, lietu internetu (*IoT*) un tā ierobežoto (*constrained*) tīkla ierīču īpašības un mūsdienīgo nolūkā balstītu tīklu (*IBN*).

Promocijas darbs ir astoņu pētījumu rezultātu apkopojums, kurā tika izstrādāti un novērtēti septiņi adaptīvi risinājumi telekomunikāciju tehnoloģijām, kā arī darba tapšanas brīdī tas kalpo kā ieskats nozarē jaunajās un aktuālajās tehnoloģijās. Apspriestās problēmas ir tīkla ceļu ģenerēšana, sadrumstalotības mazināšana *IoT* tīklos, noteiktam mērķim būvēta tīkla aprīkojuma izmantošanas samazināšana un automatizācija tīkla konfigurācijas izveidē un pārvaldībā.

Promocijas darbā aplūkotie pētījumu rezultāti ir atspoguļoti zinātniskajās publikācijās, kas pievienotas darba pilnā teksta pielikumos. Izvirzītie un novērtētie adaptīvie risinājumi ir neviendabīga servisa funkciju ceļa iekapsulēšana, reaktīva servisa ceļa atklāšana, servisa funkciju koplietošana, servisa funkcijas ceļa deklarēšana, ieteikumi *IoT* lietu specifikācijai, ieteikumi *Thread* režģtīkla blīvumam (savienojum skaitam starp iekārtām) un ieteikumi *IBN* mākslīgā intelekta uzdevumu vienkāršošanai, tādējādi veicinot *IBN* adaptāciju telekomunikāciju nozarē.

Promocijas darba hipotēzes balstītas autora veiktajos un publicētajos pētījumos. Hipotēzes ir apstiprinājušās, jo adaptīvie risinājumi sniedz atbildes uz problēmu nostādnē izklāstītajām problēmām. Adaptīvo risinājumi savietojumam ar esošo globālā datortīkla infrastruktūru var būt nepieciešama to papildu modificēšana atbilstoši vēlamajam darbības principam.

PATEICĪBAS

Vislielākais paldies manai ģimenei, kas sniedza milzīgu atbalstu, ļaujot man fokusēties uz pētījumu izstrādi un mācībām! Es domāju ne tikai studijas doktorantūrā, bet visus 10 universitātē pavadītos gadus kopā.

Paldies manam mentoram Andrim Skrastiņam, kurš ne tikai piedalījās visos aspektos saistībā ar zinātnes izveidi, bet arī sniedza pamatīgu iespaidu un labu piemēru tajā, kā novērtēt un veikt darbus pēc to būtības!

Liels paldies manam trenerim Albertam Bagojanam! Būt par daļu no RTU atlētu komandas bija viena no pozitīvajām pieredzēm vairāku mācību gadu garumā.

Mans promocijas darba vadītājs Jurģis Poriņš un viņa RTU komanda ir tie, kas pirmām kārtam padarīja šo doktorantūru iespējamu. Esmu pateicīgs par to, kā arī par visu, ar ko universitāte ir palīdzējusi!

Mārtiņš Mihaeljans

SATURS

IZMANTOTIE SAĪSINĀJUMI

AI – Artificial Intelligence – mākslīgais intelekts
CLI – Command Line Interface – komandrindas saskarne
CPU – Central Processing Unit – centrālais procesors
DC – Data Center – datu centrs
DLP – Data Leak Prevention – datu noplūdes aizsardzība
DPI – Deep Packet Inspection – padziļinātā pakešu pārbaude
FW – Firewall – ugunsmūris
GAN – Generative Adverbial Network – ģeneratīvs adverbiāls tīkls
GUI – Graphical User Interface – grafiskā lietotāja saskarne
IBN – Intent-Based Networking – nolūkā balstīta tīklošana
IMT-2020 – starptautiskais mobilo telekomunikāciju standarts 2020
IoT – the Internet of Things – lietu internets
IP – Internet Protocol – interneta protokols
IPS – Intrusion Prevention System – pretielaušanās aizsardzība
ISO – International Organization of Standardization – Starptautiskā standartizācijas organizācija
ITU-T – Starptautiskās telekomunikāciju apvienības Telekomunikāciju standartizācijas sektors
LAN – Local Area Network – lokālais tīkls
LB – Load Balancer – slodzes balansētājs
ML – Machine Learning – mašīnmācīšanās
MPLS – Multiprotocol Label Switching – vairāku protokolu etiķešu maiņa
NAT – Network Address Translation – tīkla adrešu translācija
NF – Network function – tīkla funkcija
NFV – Network Function Virtualization – tīkla funkciju virtualizācija
NLP – Natural Language Processing – dabīgās valodas apstrāde
NSH – Network Service Header – tīkla servisa virsraksts
OSI – Open System Interconnection – atvērto sistēmu starpsavienojums
PAN – Personal Area Network – personālais tīkls
PBNM – Policy-Based Network Management – politikā balstīta tīkla pārvaldība
PISA – Protocol-Independent Switch Architecture – no protokola neatkarīga komutatoru arhitektūra
PSA – Programmable Switch Architecture – programmējamo komutatoru arhitektūra
P4 – Programmable Protocol-independent Packet Processing – programmējama protokola neatkarīga pakešu apstrāde
RAM – Random Access Memory – brīvpiekļuves atmiņa
SDN – Software Defined Networking – programmdefinēta tīklošana
SF – Service Function – servisa funkcija
SFF – Service Function Forwarder – servisa funkciju pārsūtītājs
SFC – Service Function Chaining – servisa funkciju virkne
SID – Segment Identifier – segmenta identifikators
SR-MPLS – Segment Routing with MPLS data plane – segmentu maršrutēšana ar *MPLS*
TE – Threat emulation/extraction – apdraudējuma emulācija/izvilkšana
VLAN – Virtual Local Area Protocol – virtuālā lokālā tīkla protokols

IEVADS

Apvienoto Nāciju Organizācijas Ģenerālās asamblejas ilgtspējas attīstības mērķu rezolūcijā minētie cilvēka radītie resursu pārvaldes un apgādes mehānismi – enerģētikas tīkls, ūdens apgāde, ēdiena nodrošinājums, sanitārās, militārās, monetārās un veselības aprūpes sistēmas – ir saslēgti informācijas pasaules globālajā fizisku un loģisku mezglu tīklā. Tā savstarpēji savienoto sistēmu darbība un attīstība sekmē labklājību tīklam piesaistītajās ekosistēmās un starp to iemītniekiem [1].

Kopš digitālizācijas pirmsākumiem, informācijas pasaules galvenā sastāvdaļa ir pakešu komutējamais tīkls. Komunikācija, ko šis tīkls nodrošina, ļauj tā lietotājiem apmainīties ar informāciju, precēm un pakalpojumiem [2]. Adaptīvo risinājumu novērtējums virtuālizēto programmdefinēto tīklu vadībai ir balstīts vairākos pētījumos par pakešu komutējamiem tīkliem, kuru rezultāts ir septiņi adaptīvi risinājumi.

Servisa funkciju virknēšana tiek papildināta ar vairākpunktu izvietojumu (*multihomed* jeb vienam galamērķim vienlaikus pieejami vairāki savienojuma punkti). Šī funkcionalitāte dod iespēju veikt servisa funkciju koplietošanu daudzceļu datu plūsmu sinhronai apstrādei.

Ierosināta alternatīva pilnas iekapsulēšana lietošanai visos servisa funkciju ceļa posmos. Alternatīvā metode sniedz iespēju veikt pakešu iekapsulēšanu tikai tajos virknes posmos, kur tas nepieciešams.

Iepriekš iestatīta servisa funkciju ceļa politika tiek aizstāta ar mehānismu automātiskai ceļu atrašanai, lietojot noklusējuma ceļu un atkārtotu tīkla trafika klasifikāciju. Šī funkcionalitāte samazina cilvēka iesaisti datortīkla pārraudzībā, tādējādi izslēdzot tīkla darbības traucējumu rašanos iespējamību nepareizas konfigurācijas rezultātā.

Ir izstrādāts piedāvājums lietu kategorizēšanas trūkumam lietu internetā. Tas ļauj sagrupēt iekārtas pēc to fiziskām iezīmēm. Kategorizēšana piešķir iespēju viena tipa iekārtām definēt noteiktu parametru politiku (sasniedzamība, identificēšana, programmatūras atbalsts, enerģijas taupīšanas veids), tādējādi palielinot savstarpējo iekārtu savietojamību.

Veikta *Thread* režģtīkla (tuvas distances komunikācijas tehnoloģija *IoT* iekārtām maza izmēra personālajos tīklos) blīvuma (savienojumu skaita starp iekārtām) izpēte. Šī izpēte rezultējas ar ierosinājumu, kas ļauj samazināt personālā tīkla reģenerācijas cikla radīto visa režģtīkla nepieejamību.

Izstrādāts programmējamas pakešu apstrādes lietojuma modelis servisa funkciju virknēšanai, lietojot segmentu maršrutēšanu ar *MPLS* transportam. Šis modelis parāda segmentu identifikatora un *MPLS* iezīmju savstarpējo sasaisti un ierosina apzīmēt servisa ceļu ar vienotu identifikatoru.

Ierosināta mākslīgā intelekta un mašīnmācīšanās veidota nolūkā balstīta tīkla darbības principa modifikācija. Modifikācija vienkāršo mākslīgā intelekta uzdevumus, sadalot pārraudzības principus.

Promocijas darbs strukturēts četrās lielās daļās. Pirmajā daļā sniegts iesaistīto tehnoloģiju izklāsts. Otrajā – skaidroti veiktie pētījumi, skaidrojumus papildinot ar tiem atbilstošiem attēliem. Trešajā daļā apkopoti secinājumi. Ceturtajā daļā – pielikumos – pievienotas autora zinātniskās publikācijas par šo pētījumu rezultātiem. Šajā darbā apskatītie pētījumi un to rezultāti ir publicēti astoņos zinātniskajos rakstos, kas indeksēti *Scopus* datubāzē, ar mērķi uzlabot telekomunikāciju un datortīklu infrastruktūras darbību.

Idejiskā nostādne un jauninājums

Šis pētījums ir aizsācies no tādu jomu izpētes kā konvencionālā tīklošana un tās pamatprincipi, programmdefinētā tīklošana (*SDN*) un tās spēja veikt servisa funkciju virknēšanu, lietu internets (*IoT*) un tā ierobežoto tīkla iekārtu īpašības un mūsdienīgā (*state-of-the-art*) nolūkā balstītā tīklošana. Visas minētās tehnoloģijas ir ciešā sasaistē ar komunikācijas un informācijas tehnoloģiju jomu.

Pētījums balstās vairāku darbu, kas izstrādāti no 2020. gada rudenim līdz 2025. gada pavasarim, apkopojumā, tāpēc tas ne tikai parāda arvien pieaugošo izstrādes un inovāciju virzību pašā nozarē, bet arī sniedz risinājumus, kas ir lietderīgi lietojumam telekomunikāciju industrijā.

Vēl jo vairāk, promocijas darba izklāsts pārspēj komerciāli pieejamo produktu piedāvājumu, jo pētījumi ietver tehnoloģijas, kuru lietojums darba izstrādes brīdī nav plaši izplatīts. Piemēram, programmējama komutatoru arhitektūra (*PSA*). Šī tehnoloģija paredzēta datu pakešu apstrādes procesa pielāgojumam, tās lietotāja individuālajām vajadzībām. Šī darba gaitā, izmantojot *PSA*, tika izveidota un notestēta servisa funkciju virknēšana.

Problēmu nostādne

Informācijas pasaulei jeb digitālajai videi ir raksturīgi būt mainīgai un neprognozējamai. Izmaiņas, kas tajā notiek, bieži rezultējas ar tīkla problēmām. Daži problēmu piemēri ir attālināti izvietotu tīkla mezglu sasaistes pārrāvumi tīkla posmu nepieejamības rezultātā, tiešraides straumēšanas savienojuma kļūmes neoptimizētas satura izkliedes dēļ, balss signāla iestrēgšana nepietiekamas sakaru kvalitātes dēļ u. c. Piemēros minētās problēmas ir tieši adresējamas, jo to risinājums rodams konkrētās komunikācijas iesaistīto tīkla mezglu analīzē. Taču ne visas problēmas ir tieši adresējamas. Promocijas darbā apskatītas tādas problēmas, kuru risinājums var tikt iegūts vairākos veidos un kas nav attiecināmas uz konkrēta tīkla mezgla darbības traucējumiem.

1. Tīkla ceļu ģenerēšanas un tajos pārraidītās informācijas koordinācijas uzlabošana, izmantojot servisa funkciju virknēšanu.
2. Lietu interneta sadrumstalotības mazināšana un aparatūras savietojamības veicināšana, izmantojot vienotus telekomunikāciju standartus.
3. Specifiskam lietojumam būvētas aparatūras izmantošanas samazināšana, lietojot pārprogrammējamu aparatūru.
4. Cilvēka iesaistes samazināšana tīkla konfigurācijas izstrādē, tā darbības uzraudzības nodrošināšanā, kā arī telemetrijas datu analīzē, izmantojot nolūkā balstītu tīklošanu.

Minētās problēmas attiecas uz pētījumu pamatā esošajām tehnoloģijām, kas atspoguļots darba pilnā teksta 1. daļā. Izvērtējot šīs problēmas, definēts pētījuma mērķis un hipotēzes.

Darba mērķis un uzdevumi

Darba mērķis ir izpētīt jau esošos risinājumus un izstrādāt jaunus adaptīvos risinājumus programmatūras definētai tīklošanai un ar to saistītajām tehnoloģijām, kas ir saderīgi ar nākotnes tīklu un *IoT* ietvaru, tai skaitā *IMT-2020* neradio aspektiem, kas definēti *ITU-T Y* rekomendācijās. Izstrādātie risinājumi paredzēti lietošanai pakešu komutējamajos tīklos – datu centros, robežu ierīcēs (*edge* jeb tādās ierīcēs, kas izvietotas attālināti no lokālā tīkla, bet ģeogrāfiski tuvāk galalietotājam nekā mākoņa resursi), kā arī piekļuves tīklos un lietu internetā. Risinājumi ir veidoti tā, lai tie būtu savietojami ar esošajām tehnoloģijām un iekļautos to rekomendētajā vai standartizētājā darbības principā. Par to liecina arī fakts, ka darba pilnā teksta literatūras avotu sarakstā ir atrodamas vairākas rekomendācijas un tehnoloģiju standarti.

Mērķa sasniegšanai definēti vairāki uzdevumi.

1. Veikt programmatūras definētās tīklošanas, saistīto tehnoloģiju un lietu interneta analīzi problēmu nostādnes definēšanai.
2. Veikt programmatūras definēto tīklu, to saistīto tehnoloģiju un lietu interneta topoloģiju eksperimentālu izpēti, emulējot to darbības principus, un sniegt rekomendācijas par noteiktajiem un novērtētajiem risinājumiem problēmu novēršanai.
3. Pamatojoties uz veikto zinātnisko izpēti un eksperimentu rezultātiem, definēt noteikto risinājumu kopu atbilstoši paredzētajiem telekomunikāciju ietvariem.
4. Novērtēt izstrādāto risinājumu attīstības scenārijus atbilstoši tehnoloģiju attīstības tendencēm, piemēram, ierīču skaita straujš pieaugums, virzība uz pakešu pārsūtīšanu vai virzība uz virtualizāciju un automatizāciju.

Uzdevumi definē kopīgu risināšanas gaitu, kas piemērojama visām šajā darbā pētītajām problēmām.

Hipotēzes

Promocijas darba izstrādes gaitā radīti septiņi adaptīvi risinājumi, kas ir pakārtoti definētajām hipotēzēm.

1. Pārraidāmās informācijas ceļa reaktīvu izveidi un koordinēšanu vairāku administratīvo domēnu ietvaros ir iespējams realizēt, lietojot servisa funkciju virknēšanu.
2. Lietu interneta sadrumstalotība ir novēršama, lietojot vienotu pārraides vidi, un ierīču savietojamība ir nodrošināma, lietojot vienotu aplikācijas slāni, ja tiek konkretizētas lietu interneta definīcijas.
3. Servisa funkciju virknēšanu ar segmentu maršrutēšanas apakšslāni var realizēt, lietojot programmējamu aparatūru, tādējādi mazinot specifiskam lietojumam būvētas aparatūras izmantošanu.
4. Datortīkla izveides un uzturēšanas darbus var automatizēt, lietojot nolūkā balstītu tīklošanu, ja mākslīgā intelekta funkcionalitāte tiek deleģēta katram no datortīklu pārvaldības veidiem atsevišķi.

Pirmā hipotēze ietver vairākus pētījumus, jo servisa funkciju virknēšana ir šajā promocijas darbā visplašāk pētītais temats. Šī tehnoloģija tiek lietota datu centros.

Arī trešā hipotēze ietver servisa funkciju virknēšanu (*SFC*), bet citādā veidā nekā pārējās. Šajā gadījumā *SFC* ir pakārtota kā papildinājums programmējamu iekārtu darbības izpētei, veicot eksperimentu ar programmējamu komutatoru arhitektūru.

Autora ieguldījums

Promocijas darbs ir astoņu pētījumu rezultātu apkopojums, kas ne tikai veicina jaunu adaptīvu risinājumu nodrošināšanu telekomunikāciju tehnoloģijās, bet arī šī raksta tapšanas brīdī kalpo kā ieskats jaunākajās un aktuālākajās tehnoloģijās pašā nozarē.

Visplašāk izpētītā ir servisa funkciju virknēšana (*SFC*). Piecos no astoņiem pētījumiem adaptīvie risinājumi ir ierosināti, lai pārvarētu *SFC* tehnoloģijas ierobežojumus. Šie risinājumi neprasa veikt izmaiņas arhitektūrā, jo tie ir definēti tādā veidā, kas ļauj atkārtoti pārdomāt jau esošo elementu lietojumu.

Darbs ietver arī daudz pētītu tehnoloģiju – lietu internetu. Tā saistošās rekomendācijas un akadēmiskie materiāli liecina par noteiktu iekārtu parametru kopu, kuras lietojuma definējuma trūkuma rezultātā rodas iekārtu savietojamības problēmas.

Zinātniskā novitāte

Veiktajos promocijas darba pētījumos ir radītas programmatūras definētā tīklošanā un saistītajās tehnoloģijās un lietu internetā lietojamas novitātes:

- divas jaunas servisa funkciju virknēšanas pakešu iekapsulēšanas metodes [3];
- jauns servisa funkciju ceļu politikas izveides paņēmiens [4], [5];
- jauna lietu interneta fizisko ierīču kategorizēšanas koncepcija esošās iekārtu savstarpējās savietojamības problēmu risināšanai [6];
- jauns servisa funkciju domēnā esošu tīkla ierīču lietojums daudzceļu datu plūsmu apkalpošanai [7];
- jauna cēloņsakarība starp *Thread* tehnoloģijas režģtīkla iekārtu savienojumu skaitu un tīkla reģenerācijas biežumu [8];
- jauns servisa funkciju ceļa apzīmēšanas paņēmiens segmentu maršrutēšanā [9] un programmas programmējamas komutatoru arhitektūras tīkla iekārtām [10];
- divi jauni nolūkā balstītas tīklošanas modeļi, kas atvieglo mākslīgā intelekta uzdevumu veikšanu, pielāgojot modeļus atbilstoši vadāmajam tīklam [11].

Darba praktiskā vērtība

Promocijas darbā nav aplūkota tikai viena galvenā problēma un definēta viena hipotēze, ko pierādīt. Tādas ir vairākas. Arī darba uzdevumi ir vadlīnijas, kas parāda veidu, kādā pētījumi tika izstrādāti. Šāds atspoguļojums ir izvēlēts, jo katram veiktajam pētījumam ir sava nozīmi un zinātniskā publikācija. Tādēļ to rezultāti, kas uzskaitīti tālāk, ir definēti kā adaptīvi risinājumi.

1. Pēc nepieciešamības lietota servisa funkciju ceļa iekapsulēšana servisa funkciju virknēšanā pretstatā esošajai iekapsulēšanai visos ceļa posmos.
2. Reaktīva servisa ceļa atrašana, izmantojot noklusējuma ceļu un datu plūsmu atkārtotu klasifikāciju, pretstatā esošajai iepriekš iestatītajai (pirms pienākošo datu plūsmu klasifikācijas notikuma definētai) manuālai ceļu politikai.
3. Servisa funkciju koplietošana tīklos starp datu centriem, nodrošinot iespēju paplašināt pieejamo servisa funkciju ceļu tīkla funkciju klāstu vai nodrošinot esošo tīkla funkciju redundanci.
4. Viena segmenta identifikatora piesaiste visa servisa funkciju ceļa definēšanai, lietojot segmentu maršrutēšanu, uz pārprogrammējamas aparatūras.
5. Rekomendācija *IoT* lietu savstarpējas savietojamības nodrošināšanai, ieviešot kategorizēšanu to definēšanā.
6. Rekomendācija *IoT Thread* režģtīkla blīvumam (savienojumu skaitam), lai izvairītos no visa tīkla reģenerācijas, nosakot minimālo nepieciešamo iekārtu starpsavienojumu skaitu, kādam jāveidojas starp maršrutēt spējīgu iekārtu un tai pakārtotu galaiekārtu.
7. Rekomendācija nolūkā balstītas tīklošanas modeļa modifikācijai, kas paredz mākslīgā intelekta uzdevumu vienkāršošanu atbilstoši vadāmajam tīklam.

Visi septiņi adaptīvie risinājumi ir pārdomāti un konstruēti saderībai ar standartizētajām un darba izstrādes brīdī izmantotajām tehnoloģijām, kas tika pētītas. It īpaši servisa funkciju virknēšanā balstītie risinājumi, jo tie neprasa izmaiņas *SFC* arhitektūras standartos.

Risinājumi, kas ir veidoti rekomendāciju formā, ietver to praktisko vērtību pašu pētījumu kontekstā. Piemēram, piektais risinājums ietver vairākus paragrāfus ar argumentiem, kas izceļ esošajos lietu interneta standartos atstātās nepilnības.

Visi ievadā minētie aspekti tiek padziļināti apskatīti nākamajās nodaļās un visaptveroši iekļauti apkopotajos secinājumos.

Darba aprobācija un publikācijas

Pētījumu rezultāti, kas apskatīti šajā darbā, ir publicēti zinātniskajos rakstos, kas pievienoti pielikumā un uzskaitīti tālāk tekstā, minot arī konferences, kurās zinātniskie darbi prezentēti.

Zinātniskās publikācijas

1. M. Mihaeljans and A. Skrastins, "Network Topology-aware Service Function Chaining in Software Defined Network", 2020 28th Telecommunications Forum (TELFOR), Belgrade, Serbia, 2020, pp. 1–4, doi: 10.1109/TELFOR51502.2020.9306554.
2. M. Mihaeljans and A. Skrastins, "Reactive Service Function Path Discovery Approach in Software Defined Network", 2021 29th Telecommunications Forum (TELFOR), Belgrade, Serbia, 2021, pp. 1–4, doi: 10.1109/TELFOR52709.2021.9653356.
3. M. Mihaeljans and A. Skrastins, "Evaluation of reactive service function path discovery in symmetrical environment", Telfor Journal, vol. 14, no. 1, pp. 2–7, 2022, doi: 10.5937/telfor2201002M.
4. M. Mihaeljans and A. Skrastins, "IoT concept and SDN fusion in consumer products: Overview", 2023 3rd International Conference on Electrical, Computer, Communications and Mechatronics Engineering (ICECCME), Tenerife, Canary Islands, Spain, 2023, pp. 1–6, doi: 10.1109/ICECCME57830.2023.10252518.
5. M. Mihaeljans and A. Skrastins, "Openthread Network Density Evaluation: Quantitative Analysis", 2023 Symposium on Internet of Things (SIoT), São Paulo, Brazil, 2023, pp. 1–5, doi: 10.1109/SIoT60039.2023.10390236.
6. M. Mihaeljans and A. Skrastins, "Efficient Multipath Service Function Chaining in Inter-Data Center Networks", 2023 31st Telecommunications Forum (TELFOR), Belgrade, Serbia, 2023, pp. 1–4, doi: 10.1109/TELFOR59449.2023.10372615.
7. M. Mihaeljans, A. Skrastins and J. Porins, "Parmounts of Intent-Based Networking", 28th International Conference ELECTRONICS 2024, Palanga, Lithuania, September, 2024.
8. M. Mihaeljans, A. Skrastins and J. Porins, "Service Function Chaining via SR-MPLS over P4: A rudimentary analysis", The International Conference on Intelligent Computing, Communication, Networking and Services (ICCNS2024), Dubrovnik, Croatia, September, 2024.

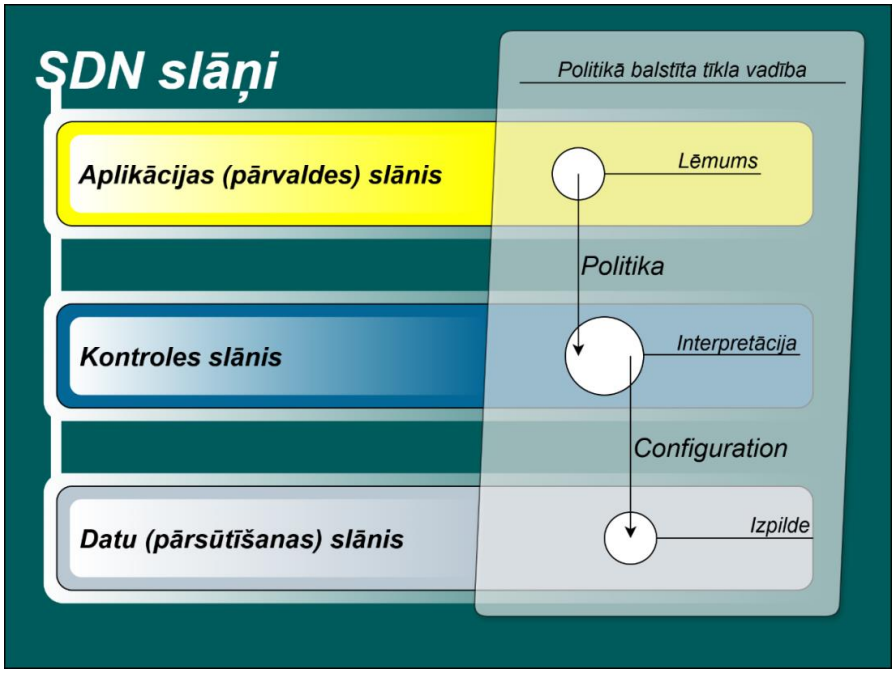
Rezultātu prezentācija konferencēs

1. 28th Telecommunications Forum (TELFOR), Belgrade, Serbia, 2020.
2. 29th Telecommunications Forum (TELFOR), Belgrade, Serbia, 2021.
3. 3rd International Conference on Electrical, Computer, Communications and Mechatronics Engineering (ICECCME), Tenerife, Canary Islands, Spain, 2023.
4. Symposium on Internet of Things (SIoT), São Paulo, Brazil, 2023.
5. 31st Telecommunications Forum (TELFOR), Belgrade, Serbia, 2023.
6. 28th International Conference ELECTRONICS 2024, Palanga, Lithuania, September, 2024.
7. The International Conference on Intelligent Computing, Communication, Networking and Services (ICCNS2024), Dubrovnik, Croatia, September, 2024.

DARBA NODAĻU PĀRSKATS

1. nodaļa. Tehnoloģiju skaidrojums

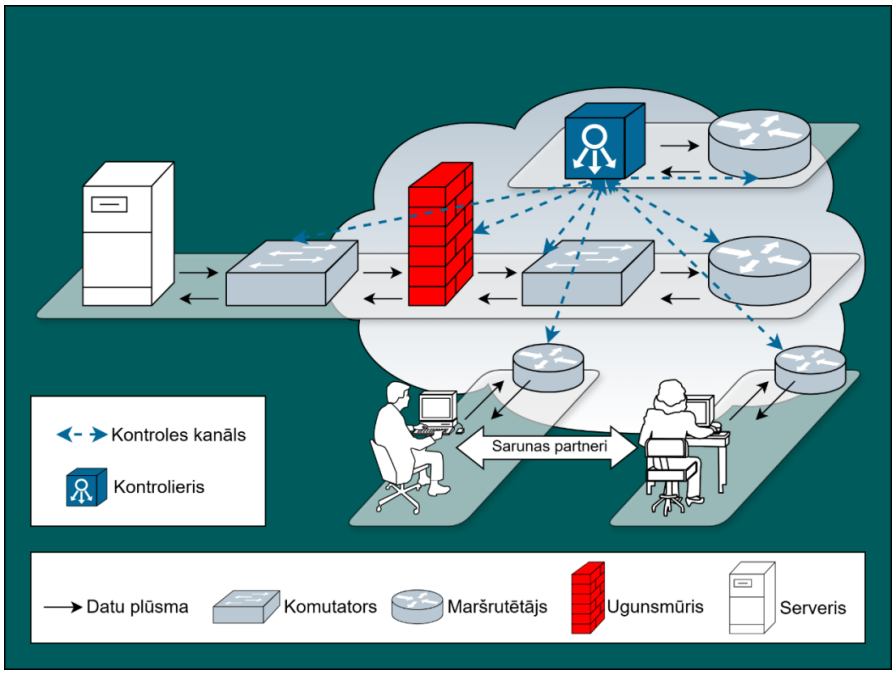
Programmdefinētais tīkls ir pretstats konvencionālajai tīklošanai. Tradicionālajās iekārtās visi komutācijas uzdevumi (pārraide, kontrole, vadība) tiek izpildīti katrā iekārtā individuāli. Turpretim programmdefinētais tīkls (*SDN*) ietver trīs slāņus, katram no tiem ir savs uzdevums – datu pārsūtīšanai, kontrolei, vadībai, kā redzams 1. attēlā.



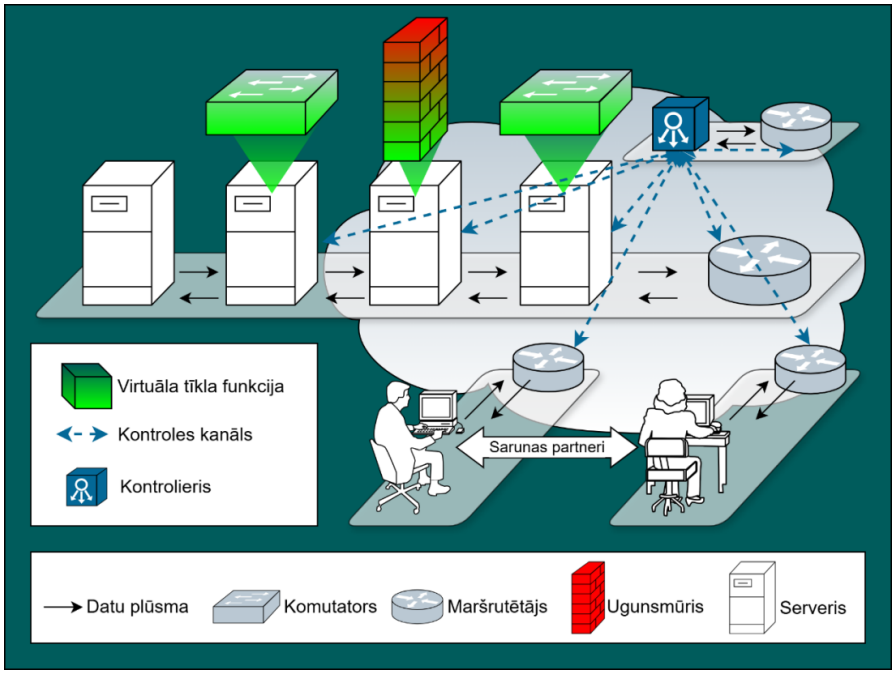
1. att. Programmdefinētās tīklošanas slāņi.

Politikā balstīta tīkla vadība (*PBNM*) ir vadības paradigma, kas atdala sistēmas uzvedības pārvaldības noteikumus no tās funkcionalitātes. Tā ļauj mazināt informācijas sistēmu uzturēšanas izmaksas, vienlaikus uzlabojot tās pielāgojamību [12].

Aplikācijas slānis nodrošina cilvēka un tīkla mijiedarbību, izmantojot grafisko lietotāja saskarni (*GUI*) vai komandrindas saskarni (*CLI*). Šis slānis ietver aplikācijas, kas nodrošina mijiedarbību, ko definē cilvēks un izpilda iekārta. Kontroles slāņa mērķis ir nodrošināt centralizētu kontroli visiem datu slāņa elementiem un viena punkta tīkla stāvokļa pārskatāmību. Var tikt lietoti viens vai vairāki kontrolieri klasterī, vai vairāki, kas ir decentralizēti un pārrauga katrs savu tīkla daļu. Datu (pārsūtīšanas) slānis lielākoties ietver vienāda tipa tīkla elementus jeb iekārtas, kas veic datu pārsūtīšanu. Tradicionālajos tīklos datu paketes tiek komutētas un maršrutētas atbilstoši starptautiskās standartizācijas organizācijas atvērto sistēmu starpsavienojuma modelim (*ISO OSI*). *SDN* datu paketes tiek pārsūtītas atbilstoši pārsūtīšanas politikai, kas definēta saderības darbību notecē (*match-action pipeline*). 2. attēlā redzams *SDN*, kur kontroles funkcija ir atdalīta no pārsūtīšanas un drošības tīkla iekārtām un nodrošināta, izmantojot attālinātu kontrolieri caur kontroles kanālam.



2. att. Programmdefinēts tīkls.



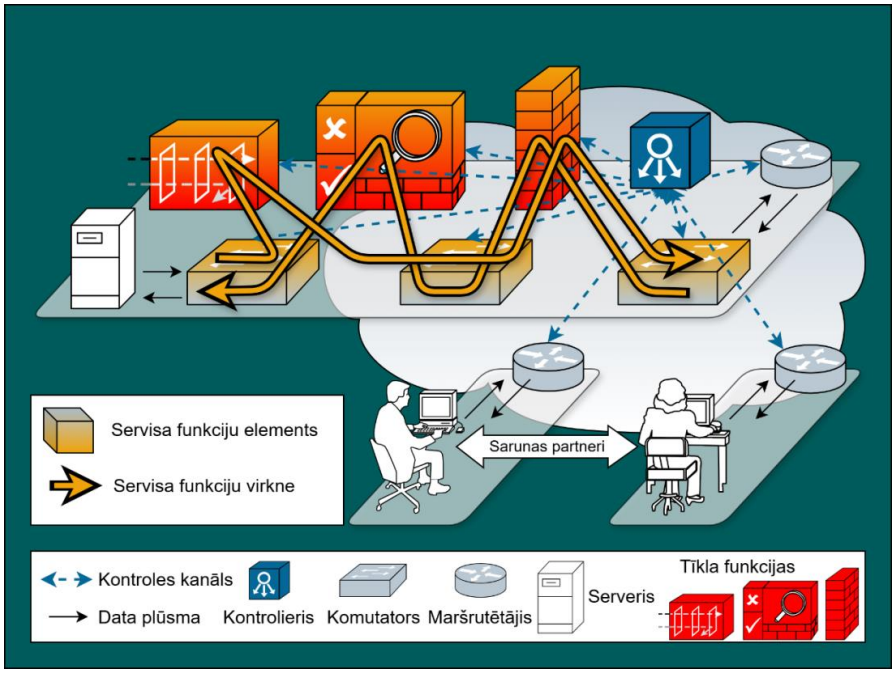
3. att. Tīkla funkciju virtualizācija.

Tīkla funkciju virtualizācija ļauj specifiskam nolūkam būvētu tīkla aparatūru aizstāt ar vispārējam nolūkam paredzētu aparatūru, ar kuras palīdzību tīkla funkcijas (*NF*) tiek emulētas. Tīkla funkcijas ir visas darbības, kas tiek veiktas ar pārraidāmajiem datiem, atskaitot pašu datu pārraidi (komutēšanu, maršrutēšanu, pārsūtīšanu). Tīkla funkciju virtualizācija (*NFV*) redzama 3. attēlā.

NF ir ugunsmūris (*FW*), tīkla adrešu translācija (*NAT*), slodzes balansētājs (*LB*), padziļinātā pakešu pārbaude (*DPI*), starpniekserveris, antivīruss, ielaušanās aizsardzība (*IPS*), datu noplūdes aizsardzība (*DLP*), apdraudējuma emulācija/izvilkšana (*TE*).

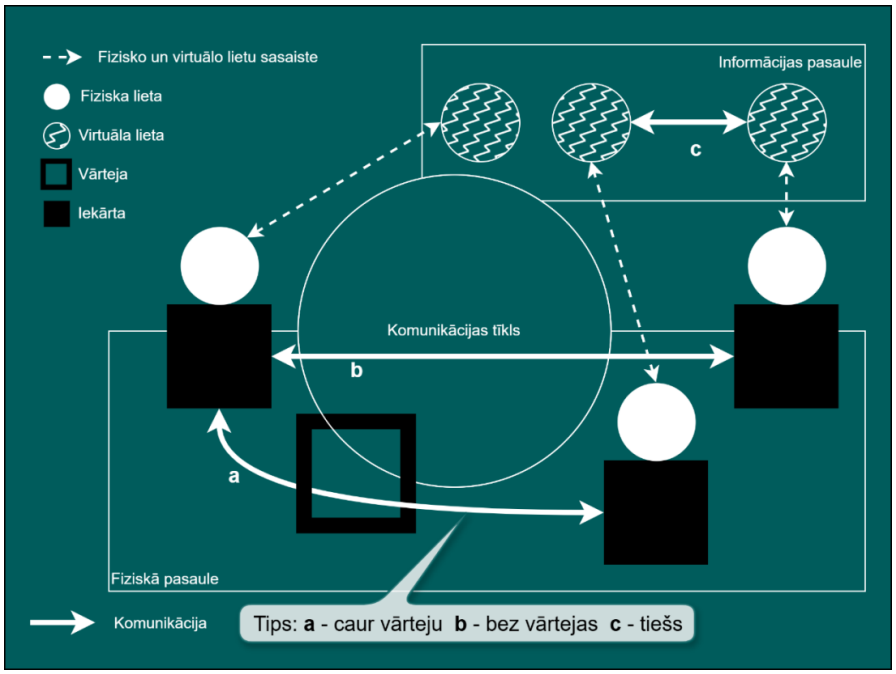
Pagrieziena punkts datu centru (*DC*) izaugsmē ir vispārējam lietojumam būvētu ierīču lietojuma aizsākums. Šīs ierīces ir paredzētas virtuālu servisu izvietojumam uz koplietošanas centrālā procesora (*CPU*), brīvpiekļuves atmiņas (*RAM*), cietā diska un citiem resursiem.

Servisa funkciju virknēšana (*SFC*) ir datu pakešu virzīšanas mehānisms cauri noteiktam servisa funkciju ceļam, izmantojot datu pakešu iekapsulēšanu. Tas ļauj mainīt pakešu apstrādes procesu, nemainot transporta topoloģiju. Pakešu iekapsulēšana ir paņēmiens, kad oriģinālajai datu paketei tiek pievienots papildu virsraksts, parasti virs vai zem *ISO OSI* otrā līmeņa virsraksta, atkarībā no izvēlētās pārraides vides. *SFC* terminoloģijā servisa funkcija ir tas pats, kas tīkla funkcija. *SFC* redzama 4. attēlā.

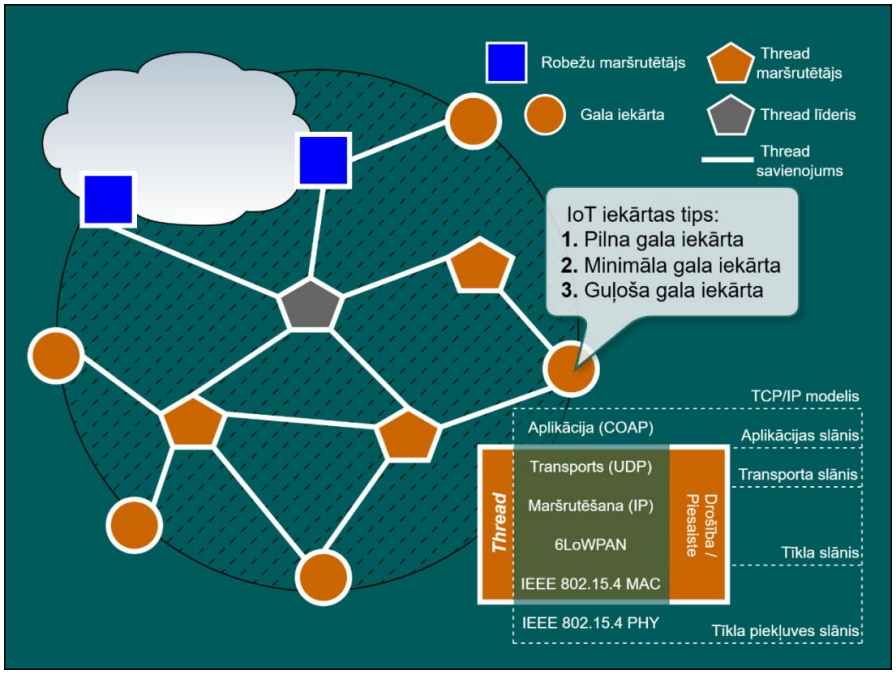


4. att. Servisa funkciju virknēšana.

Lietu internets ir tīkls, kurā ikdienas objekti tiek sajūgti ar digitālo vidi. 5. attēlā lietu interneta (*IoT*) iekārta attēlota kā melnā kaste, jo tā var būt jebkas – no mājsaimniecības iekārtas līdz industriālajam vai dabas objektam. Literatūrā *IoT* kontekstā ir sastopami divi viens otru aizvietojoši jēdzieni – fiziskā un informācijas pasaule pret reālo un digitālo pasauli. 5. attēls ilustrē arī to, kā *IoT* iekārtas var komunicēt cita ar citu un kāds ir to novietojums tīklā.



5. att. Lietu internets.



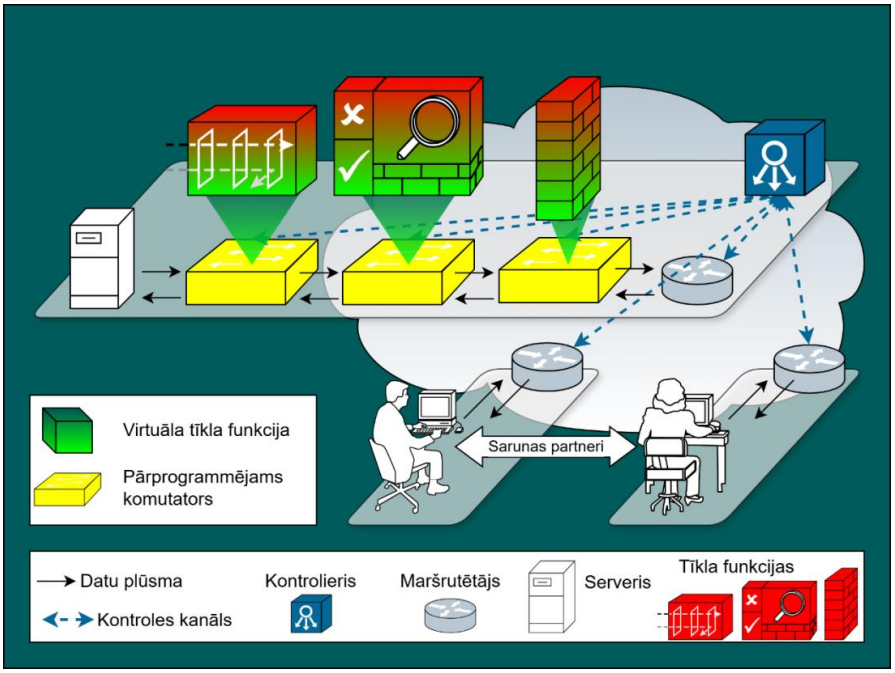
6. att. *Thread* tīkls.

***Thread* režģtīkls un *Matter* protokols** ļauj *IoT* lietām lokāli mijiedarboties. *IoT* iekārtas, kas veidotas kā plaša patēriņā preces (paredzētas lietošanai mājsaimniecībās lokāli), saskaras ar trīs lielām problēmām – savstarpējo nesavietojamību fiziskajā datu slānī, savstarpējo nesavietojamību aplikācijas slānī un tām piesaistītā vadības elementa izvietojumu mākonī. Iekārtu ražotāji identificēja šīs problēmas un, lai tās risinātu, izveidoja darba grupas, kas izstrādāja *Thread* tehnoloģiju un *Matter* protokolu.

Matter lieto kopīgu aplikācijas slāni un datu modeli, kas nodrošina aplikācijas slāņa savietojamību starp ierīcēm, ļaujot tām komunicēt cita ar citu virs vairākām *IP* tīkla topoloģijām. Kopš izveides *Matter* darbojas *Wi-Fi*, *Thread* un *Ethernet* tīkla slāņos, un iekārtu piesaistei tīklam *Matter* lieto *Bluetooth Low Energy* [13].

Matter uzskata tīklu par koplietojamu resursu (tīkla mezgli un to funkcijas ir brīvi pieejami vadībai no vairākiem avotiem). Tas neveic pieņēmumus par tīkla elementu piederību vai piekļuves rezervāciju. Rezultātā ir iespējams veidot vairākus *Matter* tīkla pārklājumus (*overlay*) virs vienas un tās paša ierīču kopas *IP* tīklā [14]. Respektīvi, ierīču funkcijas var tikt pakārtotas tā, lai to vadību vienlaikus varētu nodrošināt vairāki avoti.

Personīgais tīkls (*PAN*), kas izveidots ar *Thread*, un *Thread* novietojums *TCP/IP* modelī redzams 6. attēlā. Šajā attēlā *Thread* tīkls ir savienots ar mākoni caur robežas maršrutētāju. Lai arī *Thread* ir *IP* balstīts, tas ir paredzēts lietojumam lokālajā tīklā.



7. att. Programmējama no protokola neatkarīga pakešu apstrāde.

Programmējama no protokola neatkarīga pakešu apstrāde (*P4*) ir koncepts, kas ļauj veikt tīkla ierīču pārprogrammēšanas funkciju. *P4* valoda tiek lietota iekārtās, kuru arhitektūra atbilst no protokola neatkarīgai komutatoru arhitektūrai (*PISA*). *SDN* populārā *OpenFlow* protokolu atbalstošā arhitektūra tika ieviesta, lai programmētu

kontroles slāni, bet saderības darbību notece palika no tīkla protokoliem atkarīga. Savukārt *P4* ir veidots, lai programmētu datu slāni [15].

7. attēlā tīkla funkcijas ir virtuāli realizētas un palaistas uz pārprogrammējamiem komutatoriem. Šos komutatorus arī var kontrolēt, izmantojot attāli izvietotu kontrolieri. Tīkla funkciju izstrāde uz *PISA* arhitektūras ir izaicinājums telekomunikāciju inženierijā, un vienlaikus tā ir arī nepieciešamība, kas sekmē *SDN* tehnoloģijas attīstību.

Segmentu maršrutēšana no jauna ir izraisījusi industrijas interesi, pateicoties tam, ka, lietojot šo tehnoloģiju, ir iespējams veikt servisa funkciju virknēšanu, izmantojot *MPLS* vai *IPv6* kā transportu, tādējādi nojaucot robežās starp tradicionālajām tīkla iekārtām un tām, kas darbojas pēc *SDN* principiem.

Segmentu maršrutēšana ar *MPLS* ir pakešu maršrutēšanas metode, kas apvieno avota maršrutēšanu ar vairāku protokolu etiķešu maiņas (*MPLS*) iekapsulēšanu. Segmentu maršrutēšanā avots vai pirmais maršrutētājs ceļā uzliek iekapsulēšanu uz oriģinālas paketes. Šī iekapsulēšana tiek lietota tranzīta mezglos lēmumu pieņemšanai par tālāku paketes virzību.

Segmentu maršrutēšana dod iespēju pakalpojumu sniedzējiem veikt tīkla sadalīšanu daļās *IP/MPLS* transporta tīklā. Tīkls izkliedētā un automatizētā veidā var koplietot infrastruktūras resursus vairākām virtuālu servisu daļām. Piemēram, viena tīkla daļa ir optimizēta zemu izmaksu transportam, otra daļa – zema latentuma transportam, trešā orķestrē atsaistītus servisus u. c. [16].

Nolūkā balstīta tīklošana (*IBN*) ļauj tā lietotājiem veikt pieprasījumus jeb nolūkus, no kuriem var tikt ģenerēta tīkla politika. Lietotājs tikai sniedz informāciju par to, ko ir nepieciešams sasniegt, atstājot jautājumu, kā to izdarīt, mākslīgā intelekta (*AI*) ziņā. Nolūka piemēri:

- atdalīt piepilsētas ofisu no galvenās ēkas, lai nodrošinātos pret datu noplūdēm;
- limitēt iespēju lietot pieejamo joslas platumu multimediju straumēšanai.

IBN ir tīkla pārvaldības tehnoloģija, kas paredzēta tam, lai mazinātu cilvēka iesaisti datortīkla izveidē un uzturēšanā, lietojot mākslīgo intelektu un mašīnmācīšanos (*ML*). Sākot no tādiem uzdevumiem kā nolūku profilēšana, izmantojot dabiskās valodas apstrādi (*NLP*), līdz pat tīkla pieejamības prognozēšanai, izmantojot lielos datu un ģeneratīvos adverbiālos tīklus (*GAN*) [17]–[21].

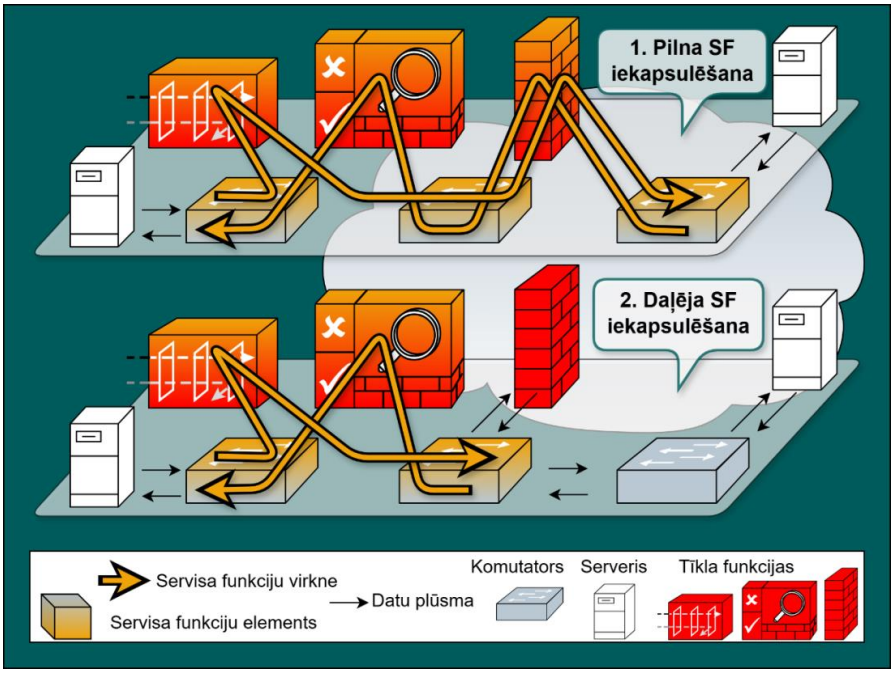
Lai veiktu izmaiņas tīkla politikās, tīkla iekārtām ir jāspēj interpretēt piemērojamās politikas. Ne visām iekārtām tīklā ir obligāti jāspēj veikt tiešu tīkla politiku interpretēšanu. Nespējīgas iekārtas var izmantot citu iekārtu palīdzību [22]. Respektīvi, *IBN* struktūrai ir jābūt skaidri definētai un jāļauj veikt tīkla politiku interpretāciju iestatāmā konfigurācijā.

2. nodaļa. Tīkla topoloģiju zinoša *SF* virknēšana

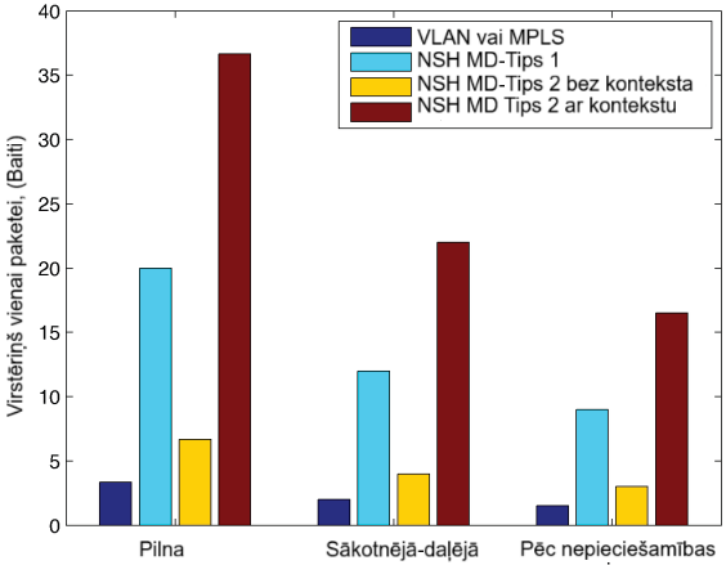
Tīkla funkciju virtualizācijas (*NFV*) lietojumiem un pakalpojuma sniedzēja tradicionālo tīklu iekārtām jāveido simbioze, vienlaikus saglabājot savietojamību ar jau esošo tīkla pārvaldību un resursu orķestrēšanu. *NFV* arhitektūrai ir jānodrošina iespēja migrācijai no slēgta tipa (*proprietary*) fizisko tīkla ierīču lietošanas uz virtuālo un atvērtos standartos balstītām ierīcēm. Citiem vārdiem sakot, *NFV* ir jāspēj strādāt hibrīda režīmā starp tradicionālajām fiziskajām ierīcēm un virtuālajām tīkla ierīcēm [23].

Lai atbildētu uz nepieciešamību veikt orķestrēšanu daudzveidīgā vidē, šajā promocijas darbā ir izstrādāta un novērtēta tīkla topoloģiju zinoša servisa funkciju virknēšana. Tās pamatprincips ir noņemt iekapsulēšanu visur, kur tā nav nepieciešama.

Servisa funkciju virknēšanas (*SFC*) oriģinālā dizaina modifikācija nav nepieciešama. Atšķirība starp pilnu un daļēju *SF* iekapsulēšanu redzama 8. attēlā, kur *SF* ceļš ir daļēji iekapsulēts, atstājot abiem virzieniem kopīgo tīkla funkciju bez pakešu iekapsulēšanas.



8. att. Daļēja *SF* iekapsulēšana.



9. att. Iekapsulēšanas lietojuma paņēmienu salīdzinājums.

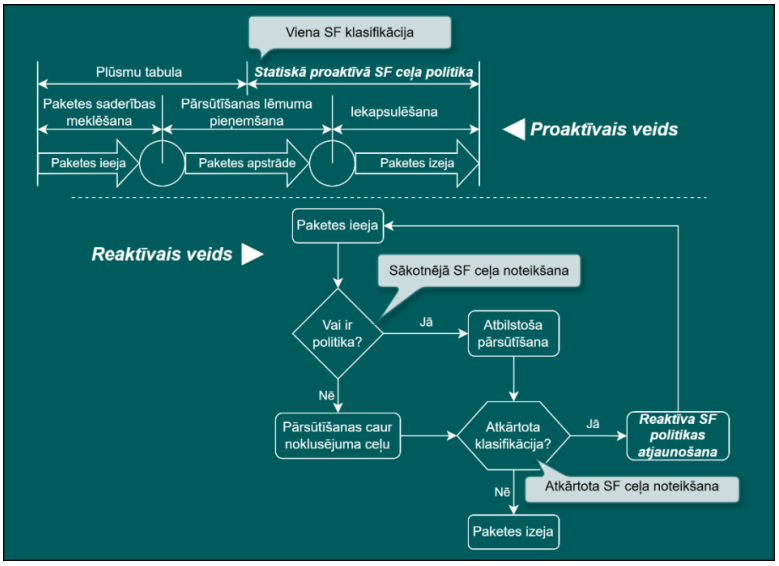
9. attēlā redzams servisa funkciju domēna emulācijas rezultāts. Grafiks ir tieši atkarīgs no emulētās tīkla topoloģijas (divu līmeņu zvaigznes topoloģija ar trīs servisa funkcijām). Citos gadījumos *SF* skaits un izvēlētais topoloģijas veids var būt mainīgs atkarībā no tīkla izmēra un citiem faktoriem. Šis fakts tomēr neizslēdz to, ka rezultāti parāda viennozīmīgu iespēju samazināt datu pakešu pārraides virstēriņu. Runa ir par virstēriņu, kas radies, uzliekot papildu virsraksta lauku datu paketēm, kad tās šķērso servisa funkciju domēnu. Virstēriņš šajā darbā izstrādātajiem un novērtētajiem daļējas iekapsulēšanas paņēmieniem (sākotnējā daļējā un pēc nepieciešamības) ir mazāks nekā pilnai iekapsulēšanai (visos *SF* ceļā posmos). Virstēriņa samazinājums (atšķirība starp pilnu un daļēju iekapsulēšanu emulētajā topoloģijā) ir līdz 50 % **(38 baiti virstēriņa vienai paketei pret 16 baitiem, kā redzams 9. attēlā *NSH MD* 2. tipam ar konteksta iekapsulēšanu).** Tas raksturīgs visiem emulētajiem iekapsulēšanas protokoliem – virtuālajam lokālajam tīkla protokolam (*VLAN*), vairāku protokolu etiķešu maiņas protokolam (*MPLS*), tīkla servisa virsraksta protokolam (*NSH*).

3. nodaļa. Reaktīva *SF* ceļa atrašana

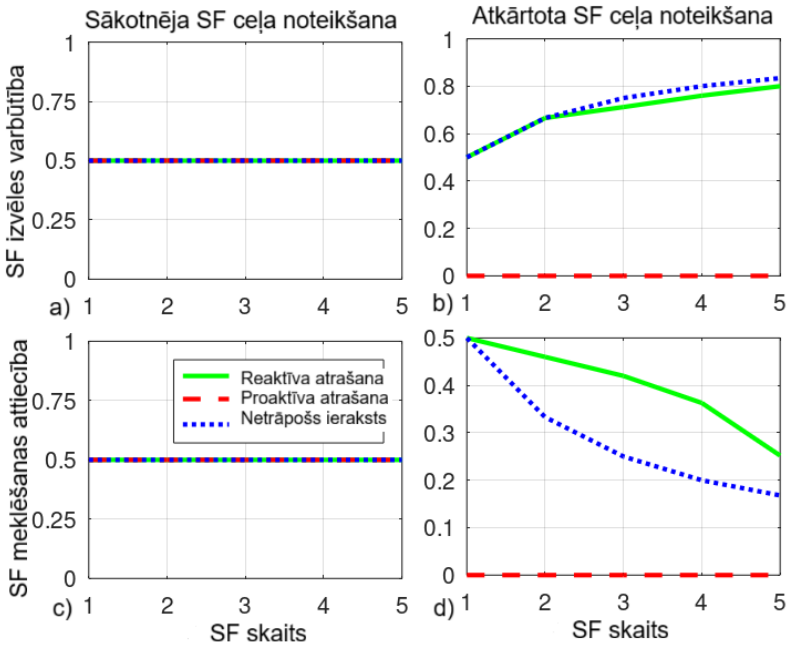
10. attēlā redzami divi datu pakešu klasifikācijas veidi. Proaktīvais veids ir lietots servisa funkciju virknēšanas arhitektūrā pēc noklusējuma, savukārt reaktīvais veids ir šajā darbā izstrādāts un novērtēts risinājums.

Proaktīvais veids veic pakešu klasificēšanu, ja ir izveidota administratora manuāli iestatīta, iepriekšdefinēta statiska ceļa politikas noteikumu bāze pienākošā tīkla trafikam. Šis veids ir izmantojams, ja iepriekš ir zināms, kāds būs pienākošais trafiks.

Šajā darbā izstrādātais reaktīvais veids ļauj sevi modificēt, esot darbībā. Respektīvi, pienākošais tīkla trafiks iegūst sākotnēju iekapsulēšanu, bet iegūtā iekapsulēšana var tikt atjaunināta, lietojot atkārtotu klasifikāciju, ko aizsāk servisa funkcija (*SF*), ja tā nespēj veikt pakešu apstrādi. Reaktīvs veids ļauj veikt *SF* ceļu grupēšanu hierarhijās. Iepriekš servisa funkciju virknes varēja izpildīt, izmantojot vienu no pieejamajiem *SF* ceļiem, savukārt hierarhijas ļauj grupēt vairākus *SF* ceļus vienas virknes īstenošanai.



10. att. Reaktīva *SF* ceļa atrašana.



11. att. Reaktīvā un proaktīvā veida salīdzinājums:
a) varbūtība izvēlēties datu plūsmai atbilstošu *SF* sākotnējā klasifikācijā;
b) varbūtība izvēlēties datu plūsmai atbilstošu *SF* atkārtotā klasifikācijā;
c) *SF* saderības meklēšanas attiecība sākotnējai klasifikācijai;
d) *SF* saderības meklēšanas attiecība atkārtotai klasifikācijai.

11. attēlā redzama reaktīvā un proaktīvā *SF* ceļa atrašanas veidu atšķirība. Grafiki ir tieši saistīti ar emulēto tīkla topoloģiju un servisa funkciju domēna raksturojošiem nosacījumiem. Emulācijas vidē katrs *SF* ceļš ietvēra vienu servisa funkciju. Tika izmantota viena parametra klasifikācija. Citos gadījumos *SF* ceļš var ietvert vairākas *SF*, kā arī klasifikāciju var lietot atkārtoti, ne pēc sākotnējā klasifikācijā lietotā parametra. Taču, atsaucoties uz emulācijas rezultātiem, tie parāda viennozīmīgu reaktīva veida iespēju uzlabot *SF* ceļa atrašanu atkārtotas klasifikācijas gadījumā.

11. attēla a) un b) grafikos attēlotā varbūtība ir iespējamība, ar kādu klasifikācijas rezultātā tiks atrasts pienākošajam datu plūsmām atbilstošais *SF* ceļš. 11. attēla c) un d) grafiki parāda attiecību starp visām iespējamajām saderībām pret atbilstošajām saderībām.

Ņemot vērā nosacījumus, 11. attēla a) un c) grafikos redzamā 50 % varbūtība un saderības atrašanas attiecība ir vienāda reaktīvā un proaktīvā gadījumā, jo ir tikai viena iespēja veikt klasifikāciju (klasifikācija *SF* virknes sākumā). Turpretim b) un d) grafikos proaktīvs veids nevar izpildīt atkārtotu klasifikāciju, tādēļ tā vērtība ir 0. Reaktīva *SF* ceļa atrašanas veids, lietojot atkārtotu klasifikāciju, spēj atrast atbilstošu *SF* ceļu. Reaktīvā veida varbūtības līkne b) grafikā nekad nepārsniegs vērtību 1, un tās tuvošanās maksimālajai vērtībai skaidrojama ar faktu, ka iespējamība neatrast atbilstošu *SF* ceļu samazinās, palielinoties atkārtotas klasifikācijas gadījumiem. Diemžēl d) grafikā redzama apgriezta tendence, jo atbilstošu *SF* ceļu attiecība pret visiem iespējamajiem samazinās, pieaugot atkārtoto klasifikāciju skaitam.

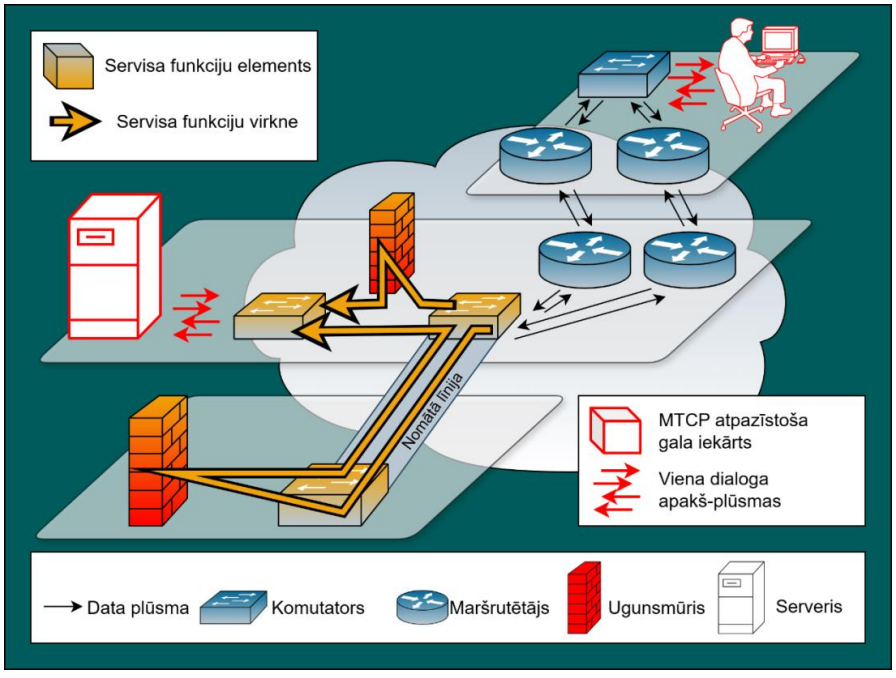
4. nodaļa. *SF* koplietošana tīklos starp datu centriem

Ne vienmēr tīkla resursus ekskluzīvi lieto viens pakalpojumu sniedzējs. Jau šobrīd uzņēmumi tiek izvietoti daudzdomēnu infrastruktūrās. Virtuālo tīkla funkciju princips ļauj palielināt pielāgojamību resursu koplietošanā un samazināt pārvaldības izmaksas. Pakalpojumu sniedzējs var piedāvāt infrastruktūras un aplikāciju kopumu kā platformu, kurā uzņēmumi var izvietot savus tīkla lietojumus. Ar šo platformu uzņēmumi var izstrādāt paši savus tīkla servisu pielāgojumus, kas piemēroti to biznesa nepieciešamībām [24].

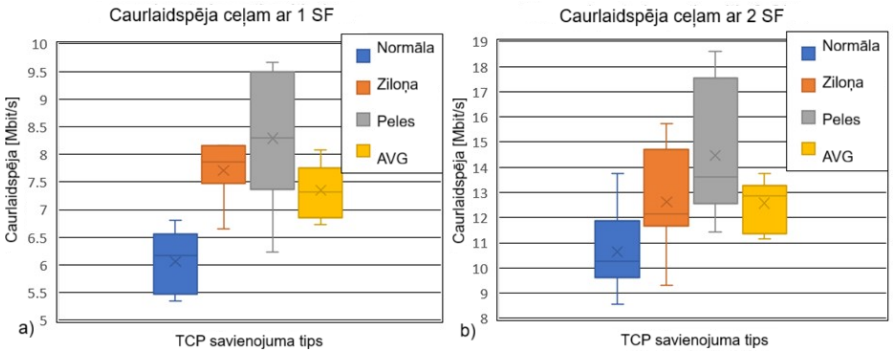
12. attēlā redzama *SF* koplietošana starpdatu centru tīklos. Komunikācija notiek starp trīs attālām lokācijām. No tām divas ir vairāksavienojumu (*multihomed* jeb vienam galamērķim vienlaikus pieejami vairāki savienojuma punkti), savukārt trešā ir ar tiešu savienojumu ar otro, izmantojot nomāto līniju.

Šajā darbā izstrādātā un novērtētā *SF* koplietošana starp kaimiņu datu centriem optimizētu tīkla aparatūras resursu lietojumu (dīkstāvē esoša tīkla aparatūra vienā datu centrā varētu apkalpot otram datu centram adresēto tīkla trafika radīto noslodzi) un ļautu veikt sistēmu dalītu izvietošanu (servisa funkciju virkni var veidot, apvienojot tīkla funkcijas, kas ir pieejamas vienā datu centrā, ar citām tīkla funkcijām, kas ir pieejamas otrā datu centrā). Pīķa noslodzes un citus neparedzamus tīkla notikumus var pārraudzīt bez liekas resursu rezervācijas kā garantijas.

Daudzceļu *TCP* (*MPTCP – Multipath TCP*) ir protokola variācija, kas ļauj vairākas datu pakešu plūsmas piesaistīt vienam dialogam, virzot tās caur dažādiem maršrutiem. *MPTCP* datu plūsma, kas pieder vienam dialogam, tiek sadalīta apakšplūsmās un adresēta vairākām *IP* adresēm, bet tiek virzīta cauri tam pašam *SF* ceļam saņemošajā galā, izslēdzot sadalīšanas ieguvumus. *SF* koplietošana ir šīs problēmas risinājums.



12. att. *SF* koplietošana.



13. att. *MPTCP* plūsmu caurlaidspējas salīdzinājums:
a) vairākas viena dialoga plūsmas tiek virzītas caur vienu *SF*;
b) vairākas viena dialoga plūsmas tiek virzītas caur divām *SF* paralēli.

13. attēlā redzams tīkla topoloģijas emulācijas rezultāts, kur a) grafikā ir gadījums, kad viss dialogs tiek virzīts uz vienu *SF* viena datu centra ietvaros, savukārt b) grafiks parāda izstrādātā un novērtētā risinājuma rezultātu, kad dialoga plūsmas tiek virzītas uz divām *SF* paralēli, lietojot savienojumu starp datu centriem (piemēram, nomāto līniju).

Rezultāti ir tieši atkarīgi no topoloģijas, kur vienā gadījumā tika izmantota viena *SF*, otrā – divu *SF* ceļi, un katrs no tiem ietvēra vienu *SF*. Citos gadījumos *SF* ceļi var ietvert vairākas *SF* vai arī var izmantot vairāk nekā divus SF ceļus. Taču, atsaucoties uz emulēto topoloģiju, rezultāti parāda viennozīmīgu caurlaidspējas pieaugumu visu plūsmu veidiem *SF* koplietošanas risinājumā. Pieaugums (starp vienas *SF* lietošanu un divu paralēlu *SF* lietošanu) ir aptuveni līdz 60 % **(8 Mbit/s caurlaidspējas pret 14 Mbit/s *AVG* (vidējā starp visām plūsmām))**. Abos grafikos novērtētas dažāda veida plūsmas, kas kategorizētas pēc plūsmu garuma, katram mērījumam atvēlot 60 sekundes (normāla – 1 dialogs 10 sekundes, ziloņa – 1 dialogs 60 sekundes, peles – 5 dialogi paralēli, katrs 0,2 sekundes, *AVG* – visu plūsmu vidējā vērtība). Lielākā caurlaidspēja novērojama peles izmēra plūsmām.

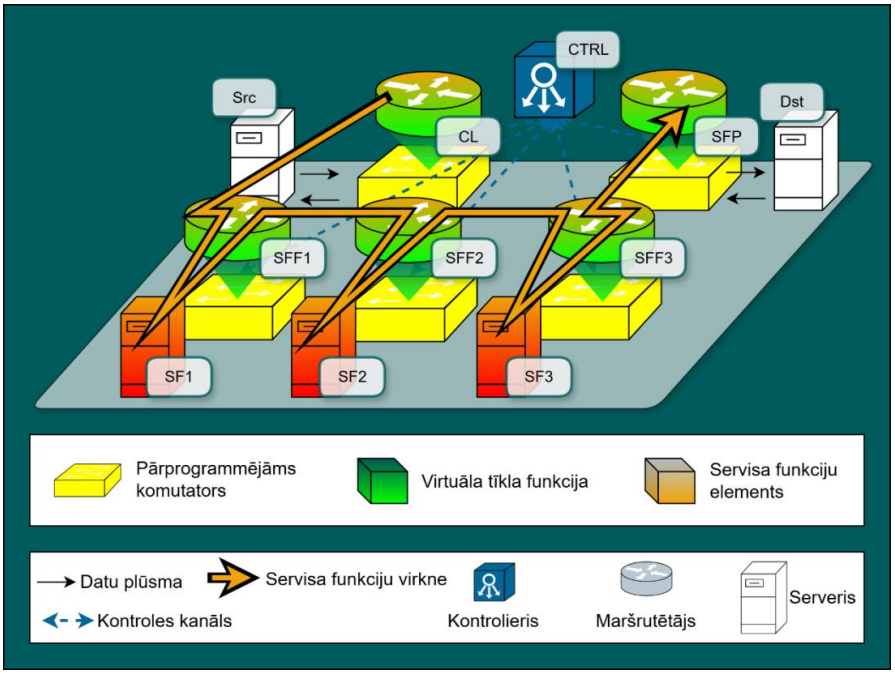
5. nodaļa. *SR-MPLS* virs pārprogrammējama datu slāņa

Segmentu maršrutēšanas (*SR*) domēns ir tīkla mezglu kopa, kas piedalās avota bāzētā maršrutēšanas modelī. Šie mezgli var būt izvietoti vienā fiziskā infrastruktūrā, vai arī tie var būt izvietoti attālināti. Ja tiek lietotas vairākas maršrutēšanas instances, tad *SR* domēns visbiežāk iekļauj visas protokolu instances tīklā. Lai gan daži lietojumi mēdz sadalīt tīklu vairākos *SR* domēnos, no kuriem katrs ietver vienu vai vairākas maršrutēšanas instances, ir paredzēts, ka visi tīkla mezgli *SR* domēnā tiek pārvaldīti ar vienu un to pašu administratīvo elementu [25].

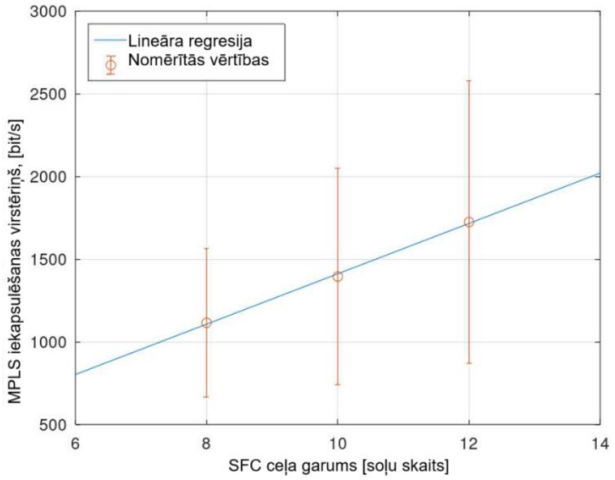
Šajā darbā ir izstrādāts un novērtēts *SR-MPLS* uz *P4* komutatoriem. Lai to realizētu, bija nepieciešams izveidot programmatūru, ko ir iespējams atkārtoti palaist uz vairākām tīkla iekārtām. Servisa funkciju virknēšanas (*SFC*) domēns tika izveidots, lai topoloģijā emulētu dažādus segmentu maršrutēšanas ceļus. Tas ļauj novērtēt *MPLS* protokola lietojuma radīto virstēriņu segmentu maršrutēšanā.

Emulētajā topoloģijā *P4* komutatori darbojās kā servisa funkciju pārsūtītāji (*SFFs*) un ietvēra loģiku, kas emulēja segmentu maršrutēšanu ar *MPLS* (*SR-MPLS*) kā transportu servisa funkciju virknēšanas (*SFC*) izveidei. Emulētā topoloģija redzama 14. attēlā. Klasifikators (*CL*) pievieno *MPLS* iezīmes kā *SFC* iekapsulēšanu paketēm,

kas saņemtas no avota (*Src*). Servisa funkciju starpnieks (*SFP*) noņem pēdējo iezīmi domēna izejā, pirms pakete tiek pārraidīta uz galamērķi (*Dst*). Šis pētījums noslēdzas ar rekomendāciju aprakstīt servisa funkciju virkni, izmantojot vienu segmenta identifikatoru (*SID*) kā vienu *MPLS* iezīmi vairāku iezīmju lietošanas vietā.



14. att. *SFC* ar *SR-MPLS* uz *P4*.



15. att. Iekapsulēšanas virstēriņa atkarība no *SF* ceļa garuma.

MPLS iekapsulēšanas radītā virstēriņa atkarība bitos sekundē no servisa funkciju ceļa posmu skaita redzama 15. attēlā. Rezultāti ir tieši atkarīgi no emulētās tīkla topoloģijas (divu līmeņu zvaigznes topoloģija ar trīs servisa funkcijām). Citos gadījumos var atšķirties topoloģijas veids, kā arī *SF* skaits topoloģijā u. c. Taču, atsaucoties uz emulācijas rezultātiem, tie parāda viennozīmīgu virstēriņa pieaugumu, palielinoties ceļa posmu skaitam.

Runa ir par virstēriņu, ko rada *MPLS* iekapsulēšana, kas tiek uzlikta oriģinālajām datu plūsmām, kamēr tās šķērso servisa funkciju domēnu. Novērotais pieaugums (starp *SF* ceļu ar astoņiem posmiem un *SF* ceļu ar 12 posmiem) ir aptuveni 40 % **(11 000 bit/s pret 17 000 bit/s virstēriņu;** 15. att.**).** Šī pieauguma izcelsmei ir divi cēloņi – tiek lietota *MPLS* iezīmju grēda *SF* ceļa aprakstīšanai, tiek lietota tikai sākotnēja klasifikācija *SF* domēna ieejā. Šajā darbā ir ierosināts abus iepriekš minētos cēloņus likvidēt, lietojot vienu segmenta identifikatoru (*MPLS* iezīmi), lai aprakstītu visu *SF* ceļu.

Šajā darbā tika izstrādāta un novērtēta arī programmatūra servisa funkciju virknēšanai, izmantojot *SR-MPLS* uz *P4* komutatoriem. Izstrādātā programmatūra ir saderīga ar *PISA* arhitektūras atvasinājumu *bmv1*, ko izmanto *Mininet* emulators. Programmatūra ir pieejama repozitorijā [10].

6. nodaļa. Lietu interneta izpēte

Lietu internets (*IoT*) ir pastāvīgi augošs tīkla ierīču režģtīkls, kur iekārtas ir tieši vai netieši piesaistītas internetam. Katra no iekārtām var būt autonoms risinājums vai integrācija jau eksistējošā objektā. *IoT* koncepcija nepilnīgi definē *IoT* lietas, ļaujot iekārtu ražotājiem variēt ar definēto īpašību piemērošanu. *IoT* režģtīklu par vertikāli sadrumstalotu sistēmu padara pieejamā protokolu steka sarežģītība, datu formātu daudzveidība un daudz dažādo saziņas procedūru lietojums [26].

Šajā darbā ierosināta *IoT* lietu segmentēšana pēc to piederības apkārtējai videi / saistītajai ekosistēmai. Segmentēšana redzama 16. attēlā. Lietas tiktu segmentētas trīs grupās.

1. Savrupa lieta – *IoT* funkcionalitāte ir lietas pastāvēšanas būtība.
2. Papildinājuma lieta – *IoT* ir papildinājums lietai ar citu galveno funkcionalitāti.
3. Modulāra lieta – *IoT* funkcionalitāte eksistē tikai tad, ja vairākas atsevišķas lietas savstarpēji sadarbojas.

16. att. *IoT* lietu segmentēšana.

17. attēlā redzamā klasifikācija ir šajā darbā ierosinātā klasifikācija pēc fiziskā izmēra. Attēls ataino lietas izmērus attiecībā pret pieaugušu cilvēku. Izmēri varētu variēt attiecībā pret bērniem fiksētas lietas gadījumā, savukārt pārnēsājamām un valkājamajām lietām vajadzētu palikt ērti lietojamām visiem cilvēkiem.

17. att. *IoT* lietas, klasificētas pēc izmēra.

Šajā darbā arī ir identificēti 11 *IoT* tīkla atribūti jeb īpašības:

- tīkla ierīcēm ir savstarpēja savienojamība;
- tīkls ir neviendabīgs;
- tīkls ir nepastāvīgs;
- tīkls ir liela mēroga (*IoT* lietu pievienošana publiskajam internetam citu autoru darbos tiek apspriesta kā problēma to skaita dēļ);
- tīkla ierīces ir jāvar identificēt;
- tīkls ir sevi regulējošs (tā darbībai ir jāatjaunojas atsevišķu tīkla mezglu nesasniedzamības gadījumā);
- tīkla ierīču atrašanās vietai ir jābūt nosakāmai;
- tīkls ir drošs;
- tīklā tiek ievērots indivīda privātums;
- tīkla ierīces ir viegli lietojamas, izmantojot *PnP* funkcionalitāti (*PnP – Plug and play* jeb pievieno un lieto);
- tīkls un tā ierīces ir vadāmi (*manageable*).

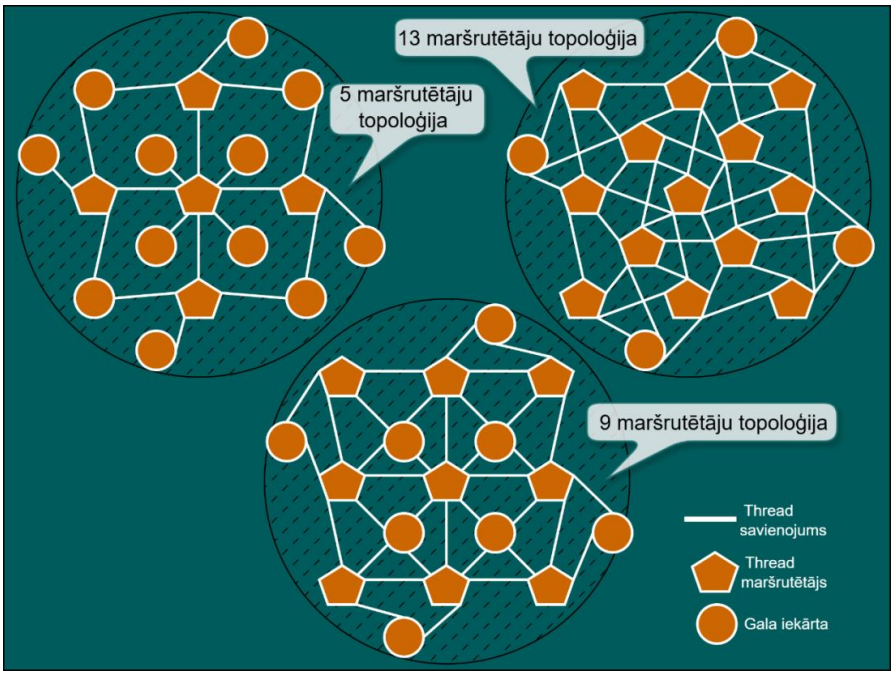
Atribūtus var lietot daļēji, izraisot tīkla sadrumstalotību un lietu nespēju savstarpēji sadarboties.

7. nodaļa. *Thread* režģtīkla blīvuma apsvērumi

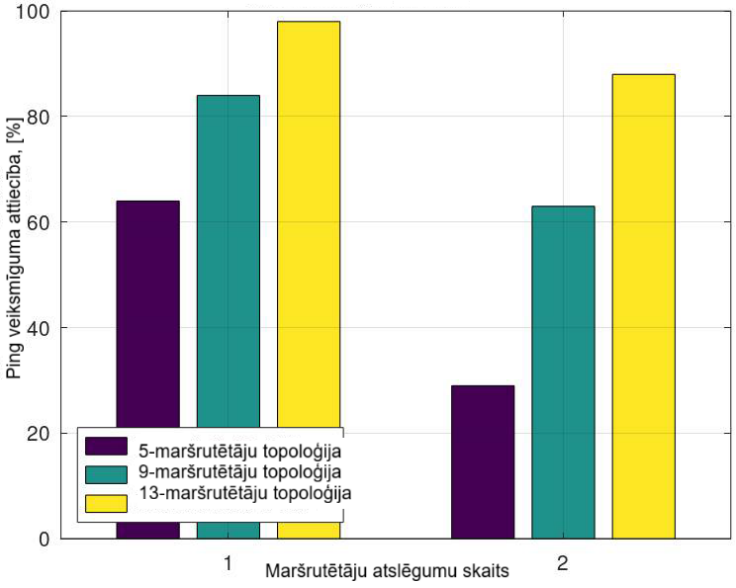
Promocijas darbā novērtētas *Thread* tehnoloģijas, emulējot tīkla topoloģiju. Apskatīts *Thread* režģtīkls, mainot tā blīvumu (savienojumu skaitu, kāds veidojas starp ierīcēm) un nosakot ietekmi, kādu savienojuma pārtraukumi rada uz katru no topoloģijām. Topoloģijas redzamas 18. attēlā. Šī darba mērķis ir mazināt personālā lokāla tīkla (*PAN*) reģenerāciju biežumu, kas notiek, mainoties *Thread* režģtīkla savienojumiem.

No emulācijām izriet šāds scenārijs. *Thread* tīkla pārrāvuma gadījumā noklīdusī galaiekārta sākotnēji meklē maršrutētspējīgu iekārtu, kas to pievienos atpakaļ *PAN*, bet, ja tāda netiek atrasts, tā izveido jaunu *PAN* sev. Kad rodas iespēja sākotnējajai ierīču kopai apvienoties, tad tas vairs nav iespējams, jo eksistē divi *PAN*. Lai atgrieztos pie viena *PAN*, ir jānotiek reģenerācijas procesam visās tīkla iekārtās. Tas nozīmē, ka pat

tām iekārtām, ko savienojuma pārtraukums nav skāris, ir nepieciešams izbeigt esošo stabilo savienojumu, lai pieslēgtos jaunajam *PAN*.



18. att. *Thread* režģtīkla blīvuma variācijas.



19. att. *Thread* tīkla sasniedzamība.

Izmērītā emulēto *Thread* tīkla topoloģiju sasniedzamība redzama 19. attēlā. Lai arī tika emulēti tikai trīs topoloģiju veidi *Thread* režģtīkla darbības principa dēļ, ar to pietiek, lai veikto emulācijas novērojumu rezultātus piemērotu citiem gadījumiem, kur ierīču skaits ir lielāks.

Sasniedzamība tika mērīta, novērojot ietekmi, kādu rada maršrutētāju atslēgšana no režģtīkla. Pat divu centrālo maršrutētāju atslēgšanas gadījumā 13 maršrutētāju topoloģija ir spējīga turpināt uzturēt savienojumu starp raidošo un uztverošo tīkla mezglu, lietojot rezerves ceļus. Eksperimenta gaita ir šāda:

- tiek raidīti *Ping* ziņojumi no vienas galaiekārtas uz otru;
- tiek veikta pakešu tveršana (tverot gan *Ping* abus virzienus, gan *Thread* komutācijas nodrošināšanas ziņojumus);
- tiek izslēgti divi centrālie maršrutētāji;
- eksperiments tiek veikts konkrētu laika periodu (10 minūtes), kurā izslēgtie maršrutētāji tiek atstāti izslēgti.

Eksperimenta rezultāts 13 maršrutētāju topoloģijas gadījumā ir aptuveni 84 % veiksmīga *Ping* datu pakešu piegāde.

Šī darba rezultātā ir izstrādāta rekomendācija – novietot *Thread* tīkla iekārtas tādā attālumā, kas ļauj veidoties vismaz diviem rezerves savienojumiem. Šis blīvums ļauj izvairīties no personālā tīkla reģenerācijām savienojuma pārtraukumu gadījumos (ja kāda no maršrutētspējīgajām iekārtām atslēdzas no tīkla), jo galaiekārtām ir pietiekams skaits rezerves ceļu.

8. nodaļa. Nolūkā balstītas tīklošanas izpēte

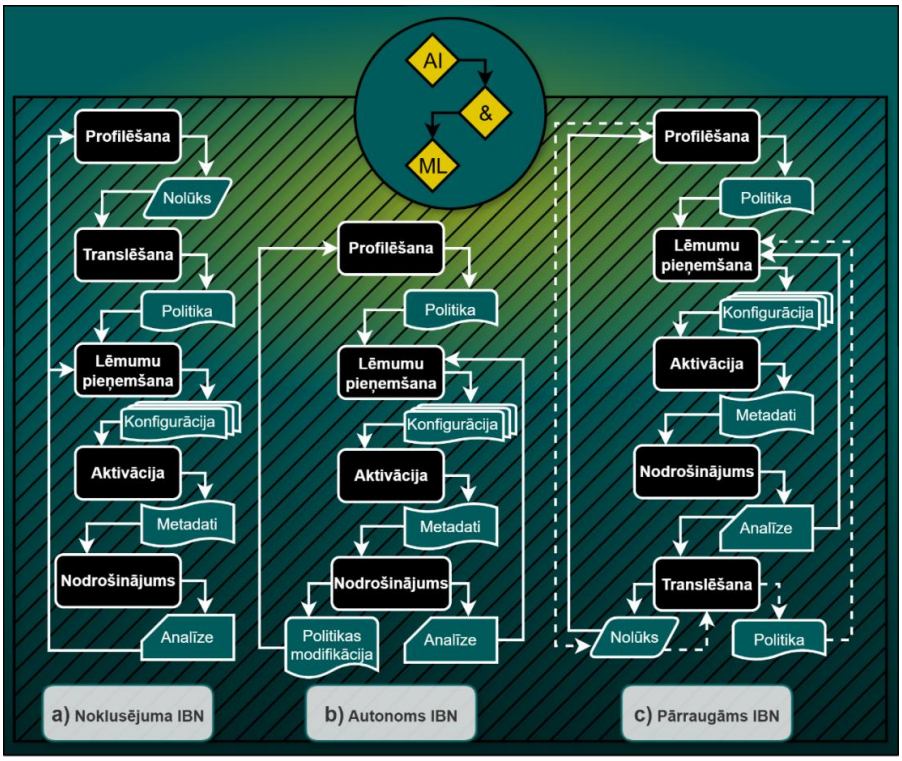
Ir noteikts, ka lietotāju pakalpojuma pieprasījumiem ir (vai vajadzētu būt) no lietotās tehnoloģijas neatkarīgiem. Tas nozīmē, ka lietotāji nezina tehnoloģijas, ko izmanto pakalpojumu sniedzējs. Tādēļ lietotāji nevar veikt pieprasījumu specifiskas tehnoloģijas lietojumam, un vieņiem ir ierobežotas iespējas pielāgot tikai piegādāto pakalpojumu [27].

Šādu darbības principu atbalsta nolūkā balstīta tīklošana (*IBN*). Tās galvenais mērķis ir piesaistīt mākslīgo intelektu tīkla tehnisko jautājum risināšanā (konfigurācijas izveidē, savienojuma starp tīkla mezgliem uzturēšanā, drošības nodrošināšanā u. c.).

Lai paātrinātu nolūkā balstītas tīklošanas ieviešanu tīklu kontroles pārvaldībai, šajā darbā ir atklāts, ka nepieciešams veikt izmaiņas modeļa koncepcijā. Veiktais pētījums identificē esošā modeļa struktūru un ierosina veidot divus jaunus tās atvasinājumus, kas redzami 20. attēlā.

20. a) attēlā redzamajam eksistējošajam noklusējuma *IBN* modelim nav definēta tīkla vadības veida. Šī darba izstrādes gaitā definētie tā atvasinājumi ir šādi:

1) autonoms *IBN* modelis b) – paredzēts lietojumam autonomos tīklos, kuru izveide un uzturēšana ir process, kur tīkla vadībai nav nepieciešama cilvēka uzraudzība. Šādi tīkli ir orientēti uz savienojuma nodrošināšanu un tā efektivitātes monitoringu. Izmaiņas tīklā tiek veiktas, ja nepieciešams uzlabot pārraides parametrus vai pievienot vai noņemt tīkla mezglus;
2) pārraugāma (*supervised*) *IBN* modelis c) – paredzēts lietojumam tīklos, kuru uzvedība tiek definēta un modificēta atbilstoši uzrauga vēlmēm. Šādi tīkli ir orientēti uz maksas pakalpojumiem vai paaugstinātu drošību un tiek modificēti atkarībā no auditācijas rezultātiem par procesiem, kas ir (vai var būt) nesaistīti ar tīkla darbības efektivitāti.



20. att. *IBN* modeļi:
a) noklusējuma *IBN* modelis;
b) *IBN* modelis autonomiem tīkliem;
c) *IBN* modelis pārraugāmiem tīkliem.

Šajā darbā ir izpētīti *IBN* jau eksistējošie lietojumi un ierosinātās standartizācijas mēģinājumi. Pats galvenais ir noteikti *IBN* struktūras elementi (profilēšanas bloks, translēšanas bloks, lēmumu pieņemšanas bloks, aktivācija bloks, nodrošinājuma bloks) un to savstarpējā saistība. No šiem atklājumiem ir secināts, ka pārlieku sarežģītais tīkla pārvaldības uzdevums ierobežo *IBN* adaptāciju.

SECINĀJUMI

Promocijas darbā apkopoti astoņi izstrādāti savstarpēji saistīti pētījumi. Visi pētījumi veikti pakešu komutējamajos tīklos šādu telekomunikāciju tehnoloģiju pilnveidei:

- programmatūras definēta tīklošana (*SDN*);
- tīkla funkciju virtualizācija (*NFV*);
- servisa funkciju virknēšana (*SFC*);
- programmējama no protokola neatkarīga pakešu apstrāde (*P4*);
- nolūkā balstīta tīklošana (*IBN*);
- lietu internets (*IoT*).

Pēc veikto pētījumu rezultātu apkopošanas ir secināti vairāki būtiski aspekti pētīto tehnoloģiju darbības principos, kas liecina par tehnoloģiju nepilnībām, to iespējamo optimizāciju un iespējām lietot jaunus risinājumus pilnīgai nepilnību novēršanai.

Secināts, ka servisa funkciju virknēšanas (*SFC*) domēnam pēc noklusējuma ir vairākas nepilnības – tas nedefinē iekapsulēšanas lietojuma nosacījumus, tas funkcionē, tikai pateicoties iepriekš iestatītai manuālai servisa funkciju (*SF*) ceļu politikai, tam nav definēta funkcionalitāte vairākpunktu (*multihomed*) izvietojumam. Visām nepilnībām ir atrasti risinājumi – tīkla topoloģiju atpazīstoša *SFC* iekapsulēšana, reaktīva *SF* ceļu politikas izveide, *SF* koplietošana tīklos starp datu centriem.

Secināts, ka lietu internets (*IoT*) pēc noklusējuma definē 11 dažādas īpašības jeb atribūtus (sasniedzamība, drošums, identifikācija, nepārtrauktība u. c.), nenosakot to sasaisti ar fizisko iekārtu parametriem (izmērs, lokācijas nemainīgums, datu pārraides veids u. c.), radot sadrumstalotību iekārtu savstarpējā sasaistē. Daļēji šo problēmu risina *Thread* un *Matter* tehnoloģijas, kas paredzētas iespējai centralizēt iekārtu vadību maza izmēra lokālajā tīklā, bet pats *Thread* režģtīkls cieš no savienojuma pārrāvumiem, ko aizsāk tā tīkla reģenerācijas process. Ir atrasti minēto nepilnību risinājumi – rekomendācija *IoT* iekārtu kategorizēšanai un rekomendācija *Thread* tīkla blīvumam.

Secināts, ka programmējama komutatoru arhitektūra (*PSA*) sniedz iespēju virtuāli realizēt tīkla funkcijas, taču nepieciešams izstrādāt programmas, kas definē vēlamo funkcionalitāti. Uz *PSA* izveidojot *SFC* funkcionalitāti, secināts arī tas, ka *SF* ceļi, kas realizēti, izmantojot segmentu maršrutēšanas ar *MPLS* iekapsulēšanu (*SR-MPLS*), ir atkarīgi no iekapsulēšanas iezīmju skaita. Izstrādāta programmatūra *SFC* izveidei uz *PSA* ar *SR-MPLS* iekapsulēšanu un, izvērtējot eksperimentu rezultātus, sniegta rekomendācija *SF* ceļu deklarēšanai lietot vienu segmenta identifikatoru.

Secināts, ka nolūkā balstīts tīklošanas (*IBN*) modelis ietver vairākus pamatblokus – profilēšanas, translēšanas, lēmumu pieņemšanas, aktivācija un nodrošinājuma bloku, no kuriem katrs realizēts ar mākslīgā intelekta (*AI*) iesaisti. Nodrošinājuma bloks, kas ir atbildīgs par tīkla darbības prognozēšanu, nevar tikt realizēts, jo esošā mākslīgā intelekta funkcionalitāte nespēj šo uzdevumu veikt. Lai atvieglotu *AI* uzdoto prognozēšanas uzdevumu, definēti divi alternatīvi *IBN* modeļi.

Darba mērķis (izpētīt esošos risinājumus un izstrādāt jaunus adaptīvos risinājumus telekomunikāciju tehnoloģijām) un no tā izrietošie uzdevumi rezultējās ar septiņu adaptīvo risinājumu izstrādi un novērtējumu, attiecinot tos uz četrām telekomunikāciju problēmām.

Sasniegtie rezultāti, kas aprakstīti secinājumos, tālāk sakārtoti pēc hipotēzēm.

1. **Pārraidāmās informācijas ceļa reaktīvu izveidi un koordinēšanu vairāku administratīvo domēnu ietvaros ir iespējams realizēt, lietojot servisa funkciju virknēšanu.** Hipotēze izvirzīta no problēmas – tīkla ceļu ģenerēšanas un tajos pārraidītās informācijas koordinācijas uzlabošana, izmantojot servisa funkciju virknēšanu. Piedāvātie risinājumi ir šādi.

1.1. **Pēc nepieciešamības veikta servisa funkciju ceļa iekapsulēšana servisa funkciju virknēšanā.**
Promocijas darbā izstrādātas divas tīkla datu pakešu iekapsulēšanas metodes, kas sniedz iespēju veikt iekapsulēšanu tikai nepieciešamajos servisa virknes posmos pretēji iekapsulēšanas lietojumam visos virknes posmos. Tās samazina virstēriņu (emulētajā tīkla topoloģijā ir sasniegts 50 % virstēriņa samazinājums, kas ir 38 baitu virstēriņš vienai paketei pret 16 baitiem, kā redzams 9. attēlā, salīdzinot ar pilnu iekapsulēšanu), nodrošina vairākpunktu (*multihomed*) izvietojumu, kā arī izslēdz servisa ceļa ierobežojumus (posmu skaita atkarība no virsraksta, SF ceļa virzība tikai viena administratīva domēna ietvaros).
(Darba pilnajā tekstā – risinājums izvirzīts pētījuma 1. pielikumā. Tas izskaidrots 2. nodaļā. Atbilstošo tehnoloģiju skaidrojums dots 1.2. un 1.3. nodaļā.)

1.2. **Reaktīva servisa ceļa atrašana, izmantojot noklusējuma ceļu un datu plūsmu atkārtotu klasifikāciju.**
Promocijas darbā izstrādātais risinājums ir kombinācija no noklusējuma ceļa un atkārtotas tīkla datu pakešu klasifikācijas lietošanas pretēji iepriekš definētai politikai. Tas samazina cilvēka iesaisti servisa funkciju ceļu izveidē. **Autora piedāvātais risinājums spēj veikt atkārtotu *SF* ceļa atklāšanu viena kritērija politikas virzīta tīkla trafikam, bet esošais risinājums to nespēj.**
(Darba pilnajā tekstā – risinājums izvirzīts pētījuma 2. un 3. pielikumā. Tas izskaidrots 3. nodaļā. Atbilstošo tehnoloģiju skaidrojums dots 1.1. un 1.3. nodaļā.)

1.3. **Servisa funkciju koplietošana tīklos starp datu centriem.**
Promocijas darbā izstrādātais risinājums ļauj virzīt vienam daudzceļu *TCP* (*MPTCP*) dialogam piederošās apakšplūsmas caur atsevišķām servisa funkcijām (*SF*) pretēji vienas *SF* lietojumam. *SF* koplietošanu var piemērot tīklos starp datu centriem, šādi paaugstinot kopējo savienojuma caurlaidspēju (emulētajā tīkla topoloģijā izdevās sasniegt 60 % caurlaidspējas pieaugumu).
(Darba pilnajā tekstā – risinājums izvirzīts pētījuma 6. pielikumā. Tas izskaidrots 4. nodaļā. Atbilstošo tehnoloģiju skaidrojums dots 1.2. un 1.3. nodaļā.)

2. **Lietu interneta sadrumstalotība ir novēršama, lietojot vienotu pārraides vidi, un ierīču savietojamība ir nodrošināma, lietojot vienotu aplikācijas slāni, ja tiek konkretizētas lietu interneta definīcijas.** Hipotēze izvirzīta, ņemot vērā problēmu – lietu interneta sadrumstalotības mazināšana un aparatūras savietojamības veicināšana, izmantojot vienotus telekomunikāciju standartus. Piedāvātie risinājumi ir šādi.

2.1. **Rekomendācija lietot specializētu *IoT* lietu kategorizēšanu to definēšanai.**
Promocijas darbā izstrādāta rekomendācija nepieciešamībai pēc *IoT* lietu kategorizēšanas atbilstoši iekārtu fiziskajam izmēram **(valkājama, pārnēsājama, stacionāra)** un *IoT* funkcionalitātes lomai fiziskās iekārtas galvenā uzdevuma veikšanā **(savrupa, papildinājuma, modulāra)**, kā arī tās

savienojuma parametriem **(kādu fizisko signālu var pārraidīt)**. Tās ļauj pārvarēt sadrumstalotību saistībā ar iekārtu savstarpējo nesaderību, nosaka lietu novietojumu globālajā datu tīmeklī un uzlabo lietu atbilstību *IoT* ietvara atribūtiem jeb īpašībām.
(Darba pilnajā tekstā – risinājums izvirzīts pētījuma 4. pielikumā. Tas izskaidrots 6. nodaļā. Atbilstošo tehnoloģiju skaidrojums dots 1.4. nodaļā.)

2.2. **Rekomendācija *Thread* režģtīkla blīvumam (savienojumu skaitam).**
Promocijas darbā izstrādāta rekomendācija, kas norāda, ka nepieciešams novietot *Thread* tīkla iekārtas tādā attālumā, kas ļauj veidoties vismaz diviem rezerves savienojumiem. Šis blīvums samazina personālā tīkla reģenerāciju biežumu savienojuma pārtraukumu gadījumos (emulētajā tīkla topoloģijā izvēlētais blīvums nodrošina 84 % sakaru nepārtrauktību divu centrālo maršrutētāju darbības pārtraukumu gadījumā, mērot pakešu zudumu noteiktā laika intervālā). **Nav noteikta rekomendētā savienojumu skaita, atskaitot šajā darbā piedāvāto.**
(Darba pilnajā tekstā – risinājums izvirzīts pētījuma 5. pielikumā. Tas izskaidrots 7. nodaļā. Atbilstošo tehnoloģiju skaidrojums dots 1.5. nodaļā.)

3. **Servisa funkciju virknēšanu ar segmentu maršrutēšanas apakšslāni var realizēt, lietojot pārprogrammējamu aparatūru, tādējādi mazinot specifiskam lietojumam būvētas aparatūras izmantošanu.** Hipotēze izvirzīta, ņemot vērā problēmu – samazināt specifiskam lietojumam būvētas vai slēgta tipa (*proprietary*) aparatūras izmantošanu, lietojot pārprogrammējamu aparatūru. Piedāvātais risinājums – **servisa funkciju ceļa definēšana ar tam paredzētu segmenta identifikatoru, lietojot segmentu maršrutēšanu uz *P4*, kā arī autora izveidoto programmatūr var lietot kā alternatīvu slēgta tipa (*proprietary*) risinājumiem ar līdzīgu funkcionalitāti.**
No promocijas darbā veiktās tīkla topoloģijas emulācijas izriet secinājums, ka *SF* ceļa apzīmēšanu nepieciešams veikt, izmantojot vienu *MPLS* iezīmi visam *SF* ceļam, ja transports ir segmentu maršrutēšana. Tas dod iespēju izvairīties no pieaugoša virstēriņa garākiem *SF* ceļiem (emulācijās novērots 40 % virstēriņa pieaugums, salīdzinot *SF* ceļu ar astoņiem posmiem un *SF* ceļu ar 12 posmiem, tas ir, 11 000 bit/s pret 17 000 bit/s virstēriņa; 15. att.), kur tiek lietotas vairākas iezīmes (viena iezīme katram tīkla segmentam).
(Darba pilnajā tekstā – risinājums izvirzīts pētījuma 8. pielikumā. Tas izskaidrots 5. nodaļā. Atbilstošo tehnoloģiju skaidrojums dots 1.6. un 1.7. nodaļā.)

4. **Datortīkla izveides un uzturēšanas darbus var automatizēt, lietojot nolūkā balstītu tīklošanu, ja mākslīgā intelekta funkcionalitāte tiek deleģēta katram no datortīklu pārvaldības veidiem atsevišķi.** Hipotēze izvirzīta, ņemot vērā problēmu – cilvēka iesaistes samazināšana tīkla konfigurācijas izstrādē, tā darbības uzraudzības nodrošināšanā, kā arī telemetrijas datu analīzē, izmantojot nolūkā balstītu tīklošanu. Izvirzītais risinājums – **rekomendācija nolūkā balstītas tīklošanas adaptācijas virzībai.**
Promocijas darbā ierosināta esošā *IBN* modeļa sadalīšana divos apakšmodeļos – viens autonomiem tīkliem, otrs pārraugāmiem (*suppervised*) tīkliem. Dalījums atvieglo nodrošinājuma bloka (darbību kopums, kas, paļaujoties uz mākslīgā intelekta funkcionalitāti, sniedz tīkla analīzi un turpmākās darbības prognozi) uzdevumu izpildi. **7. pielikumā izpētītās references (citu autoru**

darbi) viennozīmīgi liecina par nepieciešamību atvieglot nodrošinājuma bloka uzdevumus. Piedāvātais risinājums vēl ir jātestē.
(Darba pilnajā tekstā – risinājums izvirzīts pētījuma 7. pielikumā. Tas izskaidrots 8. nodaļā. Atbilstošo tehnoloģiju skaidrojums dots 1.8. nodaļā.)

Izstrādātie un novērtētie adaptīvie risinājumi ir vairāku pabeigtu pētījumu rezultāts. Promocijas darbā definētās hipotēzes ir apstiprinātas, veicot vairāku telekomunikāciju tehnoloģiju izpēti un eksperimentālu problēmu risinājumu pārbaudi emulācijas vidēs. Promocijas darbs un tā rezultāti var kalpot kā pamats arī turpmāku pētījumu izstrādei.

LITERATŪRAS AVOTI

Minētie literatūras avoti atbilst lietojumam promocijas darba kopsavilkumā.

[1] United Nations General Assembly, Transforming our world: the 2030 Agenda for Sustainable Development, A/RES/70/1, October 21, 2015. [Online]. Available: https://sustainabledevelopment.un.org/post2015/transformingourworld, last viewed May 15, 2025.

[2] National Physical Laboratory of the United Kingdom, "Packet Switching: The First Steps on the Road to the Information Society". [Online]. Available: https://www.npl.co.uk/getattachment/about-us/History/Famous-faces/Donald-Davies/UK-role-in-Packet-Switching-(1).pdf.aspx, last viewed May 15, 2025.

[3] Mihaeljans, M., Skrastiņš, A. Network Topology-aware Service Function Chaining in Software Defined Network. In: 2020 28th Telecommunications Forum (TELFOR 2020): Proceedings, Serbia, Belgrade, 24–25 November, 2020. Piscataway: IEEE, 2020, pp. 1–4. ISBN 978-1-6654-0500-3. e-ISBN 978-1-6654-0499-0. Available from: doi: 10.1109/TELFOR51502.2020.9306554.

[4] Mihaeljans, M., Skrastiņš, A. Reactive Service Function Path Discovery Approach in Software Defined Network. In: 2021 29th Telecommunications Forum (TELFOR 2021): Proceedings, Serbia, Belgrade, 24–24 November, 2021. Piscataway: IEEE, 2021, pp. 29–32. ISBN 978-1-6654-2586-5. e-ISBN 978-1-6654-2585-8. Available from: doi: 10.1109/TELFOR52709.2021.9653356.

[5] Mihaeljans, M., Skrastiņš, A. Evaluation of Reactive Service Function Path Discovery in Symmetrical Environment. Telfor Journal, 2022, Vol. 14, No. 1, pp. 2–7. ISSN 1821-3251. e-ISSN 2334-9905. Available from: doi: 10.5937/telfor2201002M.

[6] Mihaeljans, M., Skrastiņš, A. IoT Concept and SDN Fusion in Consumer Products: Overview. In: 2023 3rd International Conference on Electrical, Computer, Communications and Mechatronics Engineering (ICECCME 2023), Spain, Tenerife, 19–21 July, 2023. Piscataway: IEEE, 2023, pp. 1–6. ISBN 979-8-3503-2298-9. e-ISBN 979-8-3503-2297-2. Available from: doi: 10.1109/ICECCME57830.2023.10252518.

[7] Mihaeljans, M., Skrastiņš, A. Efficient Multipath Service Function Chaining in Inter-Data Center Networks. In: 2023 31st Telecommunications Forum (TELFOR 2023): Proceedings, Serbia, Belgrade, 21–22 November, 2023. Piscataway: IEEE, 2023, pp. 1–4. ISBN 979-8-3503-0312-4. e-ISBN 979-8-3503-0313-1. Available from: doi: 10.1109/TELFOR59449.2023.10372615.

[8] Mihaeljans, M., Skrastiņš, A. Openthread Network Density Evaluation: Quantitative Analysis. In: 2023 Symposium on Internet of Things (SIoT 2023): Proceedings, Brazil, São Paulo, 25–27 October, 2023. Piscataway: IEEE, 2023, pp. 1–5. ISBN 979-8-3503-2944-5. e-ISBN 979-8-3503-2943-8. Available from: doi: 10.1109/SIoT60039.2023.10390236.

[9] Mihaeljans, M., Skrastiņš, A., Poriņš, J. Service Function Chaining via SR-MPLS over P4: A Rudimentary Analysis. In: 2024 International Conference on Intelligent Computing, Communication, Networking and Services (ICCNS 2024): Proceedings, Croatia, Dubrovnik, 24–27 September, 2024. Piscataway: IEEE, 2024, pp. 1–8. ISBN 979-8-3503-5470-6. e-ISBN 979-8-3503-5469-0. Available from: doi: 10.1109/ICCNS62192.2024.10776149.

[10] M. Mihaeljans and A. Skrastins, P4 program code used for SFC via SR-MPLS on PSA. [Online]. Available: https://github.com/MartinsMihaeljans/SFC-on-P4/tree/main, last viewed June 2024.

[11] Mihaeljans, M., Skrastiņš, A., Poriņš, J. Paramounts of Intent-Based Networking: Overview. Elektronika ir elektrotechnika = Electronics and Electrical Engineering, 2024, Vol. 30, No. 6, pp. 53–59. ISSN 1392-1215. e-ISSN 2029-5731. Available from: doi: 10.5755/j02.eie.38680.
[12] R. Boutaba and I. Aib, "Policy-Based Management: A Historical Perspective", Journal of Network and Systems Management, vol. 15, no. 4, pp. 447–480, Dec. 2007.
[13] Connectivity Standards Alliance, Matter Overview, p. 2, 2024.
[14] Connectivity Standards Alliance, Matter Specification R1.3, Document 23-27349, p. 49, Apr. 17, 2024.
[15] Z. Liu, P. Cui, Y. Hu, Y. Dong, K. Tang and L. Xue, "A programmable data plane that supports definable computing", 2021 International Conference on Advanced Computing and Endogenous Security, Nanjing, China, 2022.
[16] Z. Ali, C. Filsfils, P. Camarillo, D. Voyer, S. Matsushima, R. Rokui, A. Dhamija, and P. Maheshwari, Building Blocks for Network Slice Realization in Segment Routing Network, Internet-Draft, TEAS Working Group, p. 4, Sep. 7, 2022.
[17] A. Clemm, L. Ciavaglia, and L. Z. Granville, "Intent-Based Networking – Concepts and Definitions," IRTF RFC 9315, 2022, Available from: doi: 10.17487/RFC9315.
[18] C. Li et al., "Intent Classification draft-li-nmrg-intent-classification-00", IETF Network Working Group, 2019.
[19] S. Hares, "Intent-Based Nemo Overview draft-hares-ibnemo-overview-01", IETF Internet-Draft, 2016.
[20] A. Leivadeas and M. Falkner, "A Survey on Intent-Based Networking", in IEEE Communications Surveys & Tutorials, vol. 25, no. 1, pp. 625–655, Firstquarter 2023. Available from: doi: 10.1109/COMST.2022.3215919.
[21] A. Leivadeas and M. Falkner, "Autonomous Network Assurance in Intent Based Networking: Vision and Challenges", 2023 32nd International Conference on Computer Communications and Networks (ICCCN), Honolulu, HI, USA, 2023, pp. 1–10. Available from: doi: 10.1109/ICCCN58024.2023.10230112.
[22] M. Stevens, W. Weiss, H. Mahon, B. Moore, J. Strassner, G. Waters, A. Westerinen, and J. Wheeler, Policy Framework, Internet-Draft, Expires Mar. 2000, p. 18, Sep. 13, 1999.
[23] ETSI, Network Functions Virtualisation – Introductory White Paper Issue 1, p. 11, 2013.
[24] ETSI, Network Functions Virtualisation (NFV); Use Cases, ETSI GS NFV 001 V1.1.1, p. 21, Oct. 2013.
[25] C. Filsfils, S. Previdi, L. Ginsberg, B. Decraene, S. Litkowski, and R. Shakir, Segment Routing Architecture, RFC 8402, p. 4, Jul. 2018.
[26] W. Rafique, L. Qi, I. Yaqoob and M. Imran, R. Rasool, W. Dou "Complementing IoT Services Through Software Defined Networking and Edge Computing: A Comprehensive Survey" in Communications Surveys & Tutorials Vol. 22, No. 3, 2020.
[27] Q. Wu, W. Liu, and A. Farrel, Service Models Explained, RFC 8309, p. 9, Internet Engineering Task Force (IETF), Jan. 2018.

Mārtiņš Mihaeljans dzimis 1994. gadā. Rīgas Tehniskajā universitātē ieguvis bakalaura grādu elektrozinātnē (2018) un maģistra grādu telekomunikācijās (2020). Zinātniskās intereses saistītas ar pakešu komutējamiem datortīkliem.