KAUNO TECHNOLOGIJOS UNIVERSITETAS
INFORMATIKOS FAKULTETAS
KOMPIUTERIŲ TINKLŲ KATEDRA

Tadas Janionis

Prarastų paketų pakartojimo metodo tyrimas ir modeliavimas bevieliame lokaliajame tinkle

Magistro darbas

Darbo vadovas
Lekt. dr. Dangis Rimkus

Kaunas, 2012

KAUNO TECHNOLOGIJOS UNIVERSITETAS
INFORMATIKOS FAKULTETAS
KOMPIUTERIŲ TINKLŲ KATEDRA

Tadas Janionis

Prarastų paketų pakartojimo metodo tyrimas ir modeliavimas bevieliame lokaliajame tinkle

Magistro darbas

Recenzentas

Doc. dr. E. Karčiauskas
2012-05-

Vadovas

Lekt. dr. D. Rimkus

2012-05-

Atliko

IFN-0/3 gr. stud.
Tadas Janionis
2012-05-28

Kaunas, 2012

TURINYS

LENTELĖS

PAVEIKSLAI

1. Įvadas

Augant bevielės įrangos populiarumui atsiranda vis daugiau bevielės tinklo įrangos bei bevielių įrenginių. Naudojant šią įrangą vartotojai sutaupo nemažai resursų: nebereikia tiesti ryšio kabelių, tinklo įrenginių bei tinklo topologijos modifikacija (*poreikiui esant*) tampa paprasta, nereikalaujanti didelių pastangų ir investicijų. Nors vartotojams bevieliai tinklai yra gana patogūs, tačiau ryšys tokiuose tinkluose yra gana nepatikimas ir sąlyginai lėtas, lyginant su laidiniais tinklais.

Dažniausia paplitusi tinklų topologija yra ta, jog tinklas sudarytas iš dviejų skirtuminių dalių: laidinės ir bevielės. Bevielė dalis dažniausiai pasitaiko paskutiniame arba pirmame (*priklauso nuo tinkle teikiamų/naudojamų paslaugų*) tinklo topologijos šuolyje. Paskutinis (*pirmas*) tinklo topologijos šuolis dažniausiai yra sujungimas tarp bevielių klientų ir bevielio prieigos taško, kuris laidiniu tinklu yra prijungtas prie pasaulinio tinklo. Naudojant patikimo duomenų transportavimo protokolą (*dažnu atveju – TCP protokolą*), bevielis tinklo sujungimas gali būti naudojamas patikimam duomenų perdavimui. Tačiau patikimumas gaunamas laiko sąnaudų sąskaita.

Realaus laiko duomenų, charakterizuojamų pagal pralaidumą, delsą bei bitų klaidų kiekį, pristatymas bevieliuose tinkluose yra sudėtingas uždavinys. Bevielė tinklo dalis, esanti bet kurioje tinklo topologijos vietoje, kartu su laidine tinklo dalimi sukuria butelio kaklelio efektą. Taip naudojant patikimo duomenų transportavimo protokolą bei esant nepakankamai bevielio tinklo sujungimo kokybei yra gaunamas bendras mažesnis tinklo pralaidumas, kuris gali įtakoti nepakankamą paslaugų užtikrinimą arba netgi paslaugų užtikrinimo nebuvimą (*DoS*[1]).

Šiuolaikiniai patikimo duomenų transportavimo protokolai nėra pritaikyti realaus laiko duomenų srautui: jų pagrindinis tikslas patikimai pristatyti duomenis. Daugeliu atveju duomenų pristatymo patikimumas yra daug svarbiau už duomenų pristatymo greitį, tačiau kai kurioms aplikacijoms (*balso, multimedijos ir pan.*), mažas delsos faktorius yra tiek pats svarbus kaip ir mažas duomenų klaidų skaičius. Naudojant nepatikimo duomenų transportavimo protokolus, ši problema taip pat išlieka aktuali, nes patikimumas realaus laiko duomenų transportavime yra pakankamai svarbus, kad į tai galima būtų neatsižvelgti.

[1] *DoS (Denial of Service angl.) – Paslaugų nutraukimas, nebuvimas*

Laikas yra ribotas resursas realaus laiko duomenų srautui. Duomenų paketo siuntimas bei kartojimas nesėkmingo paketo pristatymo atveju, tol kol paketas pasieks vartotoją, gali būti per ilgas bei nepateisinamas paslaugų gavėjui. Algoritmai realaus duomenų srauto siuntimui parenkami individualiai, priklausomai nuo duomenų srauto ypatumų.

Baigiamojo magistrinio darbo tikslas yra sumodeliuoti realaus laiko įgaliotąjį protokolą bei jį ištirti OPNET programiniu paketu. Šiame darbe modeliuosime bei tirsime bevielį tinklą, paremtą *TCP/IP įgaliojimo*[2] principu. *TCP/IP* įgaliojimo metodas yra geras mišraus tinklo, sudaryto iš laidinės ir bevielės tinklo dalių, patobulinimas, leidžiantis efektyviai išnaudoti esamus tinklo resursus, skirtus tiek realaus laiko, tiek nerealaus laiko duomenų srautams. Pažymėtina, jog šis metodas gali būti taikomas bevieliuose tinkluose kur gausu trukdžių, taip pat gali būti naudojamas prieš paslaugų nutraukimo atakas bevieliuose tinkluose, kai yra atakuojama (*viena arba keletas*), tinklo bazinė stotis (*bevielis prieigos taškas*), taip sudarant bendrą interneto kanalo apkrovimą, o kartu ir paslaugos nutraukimą paskutiniame tinklo topologijos šuolyje esantiems vartotojams.

Minėtojo principo įgyvendinimas yra skaidrus bet kokiems tinkle esantiems įrenginiams, bei aukštesnio lygio protokolams. Norimas efektas pasiekiamas naudojant vietinį duomenų persiuntimą, planavimo kontrolę bei veiksmingą esamų išteklių (*bevielis paskutinio šuolio kanalas, interneto kanalas*) panaudojimą.

Baigiamajame magistro darbe atliekami šie veiksmai: išanalizuojami dažniausiai heterogeniniuose tinkluose naudojami *PEP*[3], iš kurių išrenkamas vienintelis sąlygas tenkinantis protokolas-principas (*optimaliausiai tinkantis tiek realaus laiko srauto paslaugoms tiek paprastam duomenų persiuntimui*). Toliau detaliai išnagrinėjamas pasirinktas principas, nagrinėjamos pasirinkto principo atsparumo galimybės DoS atakoms nevienalyčiame tinkle, aptariamas principo įgyvendinimas OPNET programiniame pakete. Paskutinis darbo etapas yra įgyvendinto principo tyrimas, analizė bei išvadų formulavimas.

[2] *Proxy angl. - Įgaliojimas*

[3] *PEP (Performance Enhancing Proxy angl.) – Vykdymą gerinantis įgaliojimas*

2. TCP protokolo įgaliojimo koncepcijos apžvalga

Standartiškai TCP protokolas neturi pakankamo išpildymo realaus laiko duomenų srautui valdyti, taip pat šio protokolo veikimas yra išskirtinai optimizuotas laidiniams tinklams, kurie charakterizuojami dideliu pralaidumu, maža delsa, bei santykinai mažu bitų klaidų skaičiumi. Atsižvelgiant į TCP protokolo architektūrą matyti, kad TCP nėra atskirtas laidinei ir bevielei tinklo daliai. Tuo pačiu šis protokolas negali diferencijuoti delsų, patirtų dėl srauto spūsčių laidinėje tinklo dalyje bei dėl delsų, atsirandančių bevielėje tinklo dalyje, esant duomenų paketų praradimui. Tokiai situacijai esant, bendras sujungimo pralaidumas sumažės, nes TCP protokolas delsas, atsirandančias bevielėje tinklo dalyje, traktuos kaip srauto spūstį laidinėje tinklo dalyje.

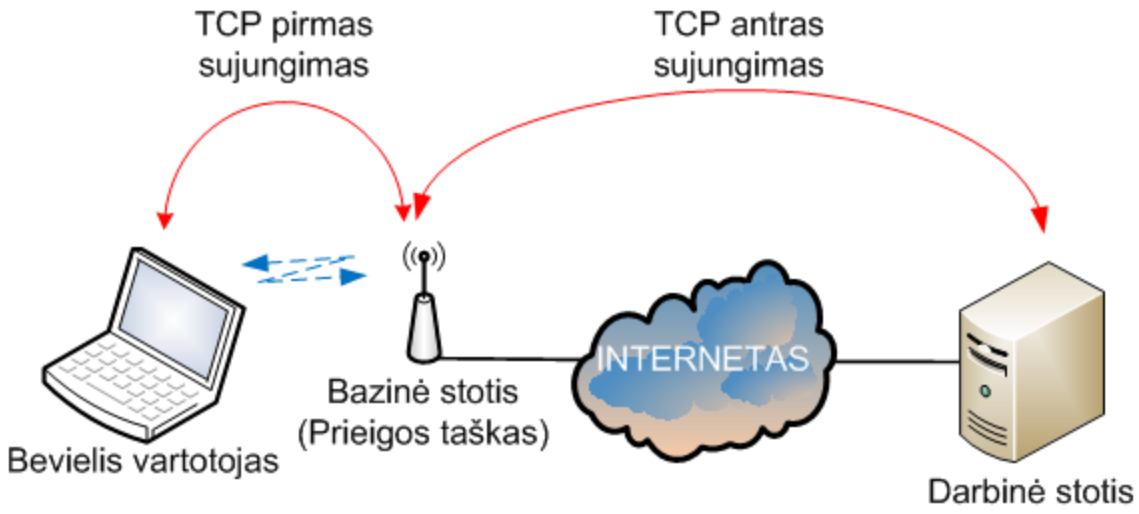
Dar viena aktuali problema, susijusi su TCP protokolo naudojimu realaus laiko duomenų paslaugoms yra tai jog TCP protokolas kanaliniame lygmenyje naudoja Ethernet protokolą, kuris turi du, skirtingus klaidų aptikimo algoritmus: laidiniam tinklui (*CSMA/CD*) bei bevieliam tinklui (*CSMA/CA*). Šių algoritmų skirtumas negarantuoja realaus laiko duomenų srauto patikimo pristatymo heterogeniniame (*laidinis – bevielis*) tinkle: nėra jokios garantijos, jog paketas tinkamu laiku bus siunčiamas bevieliu kanalu (*kanalą gali būti užimtas*), taip pat atsitiktinė laiko delsa sąlygota kolizijų laidinėje tinklo dalyje gali sukelti nepageidaujamų pasekmių.[1, 2]

2.1 Laidinio ir bevielio tinklo kombinacijos ypatumai

Kaip jau buvo minėta magistrinio darbo įvade pats TCP protokolas nėra optimizuotas bevieliams tinklams. Jei per tam tikrą laiką yra negaunamas patvirtinimas apie išsiųstą duomenų paketą, laikoma kad paketas yra prarastas dėl srauto grūsties, tuo pačiu yra sumažinama siuntimo sparta. Tačiau jei duomenų paketas yra prarandamas dėl ryšio problemų (*bevielio tinklo atveju*), paketas turi būti persiunčiamas nedelsiant, tuo pačiu turi būti nemažinama ryšio sparta. Šiame poskyryje yra pateikiama keletas minėtos problemos sprendimų. Toliau pateikti keli protokolai/principai bei problemos sprendimo būdai, heterogeninio (*laidinio – bevielio*) tinklo kombinacijai.

2.2 Išskaidytasis TCP sujungimas naudojant I-TCP principą

Vienas iš keleto būdų kaip spręsti paskutinio šuolio bevielio ryšio problemą yra Indirect-TCP (*I-TCP*) principo naudojimas. Naudojant šį metodą TCP sujungimas yra išskiriamas į dvi atskiras dalis. Pirmoji dalis – laidinė: tarp fiksuoto taško bei bazinės stoties (*prieigos taško*), antroji bevielė: tarp bazinės stoties bei mobilių vartotojų. Žemiau pateiktame paveiksle pavaizduotas išskaidyto TCP sujungimo supaprastinta schema.



1 pav. Išskaidyto TCP sujungimo supaprastinta schema

Pirmoji sujungimo dalis naudoja standartinę TCP protokolo architektūrą, o antroji sujungimo dalis naudoja optimizuotą bevielio ryšio sujungimui protokolo architektūrą. I-TCP veikimas yra grindžiamas modifikuotu TCP protokolo veiklos algoritmu bevielėje tinklo dalyje. Laidinėje tinklo dalyje TCP protokolo veikimas yra nepakitęs[3].

Taip pat išskaidant sujungimą i dvi atskiras dalis, galima išskirti srauto spūstis laidinėje dalyje bei klaidas ir vėlinimus bevielėje tinklo dalyje. Šis sprendimas pagerina TCP protokolo naudojimą duotuoju atveju, tačiau pažeidžia ryšio tęstinumo[4] principą: paketo patvirtinimai

[4] End-to-end angl. - Tęstinis

siuntėjui gali ateiti anksčiau, nei gavėjas gaus duomenis, o tai gali įtakoti bazinės stoties užkimšimą, kai darbinė stotis nuolat gaudama netikrus patvirtinimus apie gautus paketus, siųs sekančius pagal eilę paketus. Taip pat dar vienas trūkumas priskiriamas šiam protokolui-principui yra TCP dėklo apkrova: kiekvienas paketas pereina TCP dėklą keturis kartus (*vieną pas siuntėją, du – bazinėje stotyje/bevieliame prieigos taške, o ketvirtą – gavėjo pusėje*). Į šią TCP dėklo perkrovą įeina duomenų replikavimas bazinėje stotyje iš įeinančio sujungimo į išeinantį sujungimą.[4]

2.3 Kanalinio lygmens retransliavimai

Kitas būdas, kaip užtikrinti patikimą duomenų siuntimą bevieliame tinkle yra duomenų retransliavimas kanaliniame lygmenyje. Bevielio ryšio kokybė gali būti pakankamai pagerinta nepriklausomai nuo aukštesnio lygmens protokolo. Tačiau jei naudojamas aukštesnio lygmens protokolas kuria ryšio tęstinumo principą (*šiuo atveju nagrinėjamas TCP protokolas*), pakartotinis retransliavimas gali sumažinti ryšio kokybę: TCP retransliavimo algoritmas gali „konkuruoti“ su kanalinio lygmens retransliavimo algoritmu, taip mažinant protokolų veikimo optimalumą. Vienintelis būdas to išvengti apjungti konkuruojančius algoritmus, taip kad tie algoritmai turėtu kuo mažesnę neigiamą įtaką duomenų siuntimui. Tai gali būti pasiekta protokolams keičiantis informacija apie laiko išsekimo vertes bei retransliavimo logiką. Toks būdas nėra galimas jei norima naudoti tik vieną protokolą bei norima neapkrauti sistemų.[5]

2.4 Greitas pakartotinis duomenų persiuntimas

Žinoma, jog TCP protokolas gali generuoti greitus patvirtinimus (*dublikuotus patvirtinimus*[5]) kai segmentai yra gaunami ne iš eilės. Šie dublikuoti patvirtinimai neturėtų būti vėlinami. Šių patvirtinimų tikslas yra informuoti kitą sujungimo pusę, kad atvykęs segmentas buvo gautas ne iš eilės, ir nurodo, kokios eilės segmentas yra laukiamas duotuoju momentu.

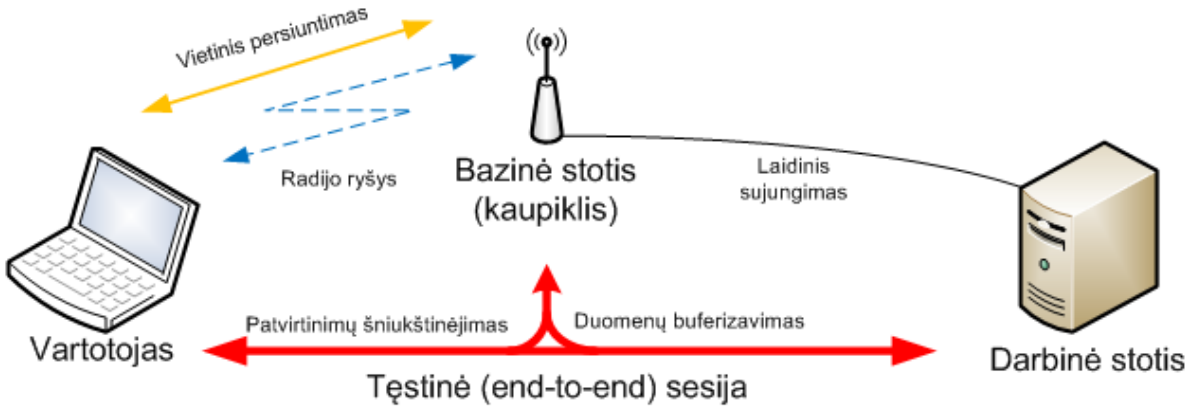
[5] Duplicate Acknowledgments angl. – Dublikuoti Patvirtinimai

Kadangi TCP protokolas nežino ar dublikuotas patvirtinimas yra atsiradęs dėl prarasto segmento, ar dėl segmentų eiliškumo, jis laukia keleto dublikuotų patvirtinimų. Jei gaunami du dublikuoti patvirtinimai, daroma prielaida, kad tai įvyko dėl netinkamo segmentų eiliškumo. Jei yra aptinkama trys ir daugiau dublikuoti patvirtinimai, laikoma, kad segmentas buvo prarastas. TCP protokolas persiunčia segmentus, kurie, kaip yra manoma, yra dingę, nelaukiant kol išseks persiuntimo laikas.[4]

Greitas pakartotinis duomenų persiuntimas užtikrina patikima duomenų siuntimą. Tačiau šis sprendimas turi ir neigiamą pusę. Pagrindinis šio sprendimo trūkumas yra tai, kad šis metodas labiau yra tinkamas valdyti judriųjų klientų perdengimui nuo vienos bazinės stoties iki kitos, o ne valdyti klaidas bevielėje tinklo dalyje.

2.5 Šniukštinėjimo[6] protokolas

Šniukštinėjimo protokolas slepia klaidų tikimybę bevieliame tinkle, paskutiniame šuolyje, skaidriai kaupiant duomenis bevieliame prieigos taške. Tai įgalina šniukštinėjimo protokolą naudoti vietinį duomenų persiuntimą bevieliu sujungimu. Palyginti su paprastu ryšio tęstinumo principu tai yra daug greitesnis duomenų persiuntimo būdas.[4]



2 pav. Šniukštinėjimo protokolo veikimo principinė schema

[6] Snoop angl. – Šniukštinėti, šniukštinėjimo

Šiuo atveju tęstinė TCP sesija išlieka, tačiau dublikuoti patvirtinimai bus filtruojami bazinėje stotyje. Šį veiksmą atliks bazinėje stotyje esantis podėlis[7]. Taip pat šniukštinėjimo protokolas siųs patvirtinimus siuntėjui tiktai kai gaus patvirtinimą iš judriojo kliento. Taip yra dėl to, kad šniukštinėjimo protokolas turi TCP protokolo logiką, kaupia patvirtinimus tiek iš siuntėjo tiek iš gavėjo.

Šniukštinėjimo protokolas gali užtikrinti patikimą tęstinį ryšį, taip pat turi logiką, kuri leidžia atskirti ne iš eilės atėjusius duomenų segmentus, taip pasiekiant efektyvų TCP protokolo veikimą. Be to šis sprendimas neįtakoja galinių sujungimo taškų: TCP protokolo modifikacija yra reikalinga tik bazinėje stotyje, įgyvendinant kaupiklio logiką.[6]

2.6 TCP/IP įgaliojimo protokolas

Kaip jau minėta anksčiau buvusiuose darbo poskyriuose TCP protokolas buvo kuriamas laidiniams tinklams. Pritaikant šį protokolą nevienalyčiams tinklams susiduriama su keliomis (*gana rimtomis*) problemomis.

Efektyvų TCP protokolo veikimą laidiniuose tinkluose, galima būtų paaiškinti taip: TCP protokolas traktuoja laiko išsekimo vertes kaip klaidas, atsiradusias dėl srauto spūsčių. Šis protokolas neįvertina galimybės, jog klaidos gali atsirasti dėl paketų pametimo bevielėje nevienalyčio tinklo dalyje. Esant dideliam bitų klaidų skaičiui (*BER*[8]) bevielio tinklo dalyje, traktuojama, kad yra susidariusi srauto spūstis (*kalbant ne tik kaip apie bevielį tinklą, bet kaip apie heterogeninį tinklą*). Tokiu atveju siuntėjas mažina siuntimo langą, taip automatiškai yra mažinamas bendras tinklo pralaidumas. Jei duomenys yra pamesti tinkle, siuntėjas nedelsiant turi persiųsti trūkstamus duomenų segmentus, taip užtikrinant nenutrūkstamą tęstinį ryšį su duomenų gavėju. Atliekant pakartotinį duomenų persiuntimą neretai padidinamas duomenų perdavimo rodiklis. Taip neefektyviai yra apkraunamas tęstinis kanalas tarp siuntėjo ir gavėjo,

[7] Cache angl. - Podėlis

[8] BER (Bit Error Rate) angl. – Bitų klaidų lygis

iki paskutinio šuolio nevienalyčiame tinkle. Kadangi TCP protokolas neatskiria BER sukeliančio šaltinio, jis negali imtis atitinkamų veiksmų šalint šių klaidų padariniams.[7]

Patikimumą TCP protokolas užtikrina nuolatos siųsdamas paketus tol, kol šie pasiekia gavėją. Šis duomenų siuntimo būdas realaus laiko duomenų srautui nėra galimas. Tol kol TCP sesijos gavėjas neatsiunčia patvirtinimo apie priimtus paketus, tol siuntėjas neturi galimybės išmesti paketų: kartoja nuolatinį duomenų siuntimą. Tačiau yra keletas būdų kaip paketas gali būti pašalinamas iš tinklo. Vienas iš šių būdų yra išmesti paketą, kai baigiasi jo galiojimo laikas (*TTL*[9]) tinkle. Tokiu atveju šis paketas tampa nebesvarbus duomenų gavėjui. Kitas būdas pašalinti paketą iš tinklo – nustatyti kad paketas būtų pašalintas, kai išsenka siuntėjo RTO[10] laikas. Šiuo būdu yra pasiekiama, tai, jog į tinklą nebepateks pasikartojantys paketai, taip pat tinklas nebus perpildytas, tinklo pralaidumas nemažės.[7]

Realaus laiko įgaliotasis protokolas turi dvi atskiras funkcijas: turi galimybę valdyti realaus laiko duomenų persiuntimą, taip pat jis veikia kaip tarpinis paslaugų tiekėjas, galintis maskuoti bevielio ryšio, esančiame paskutiniame tinklo šuolyje, klaidas. Realaus laiko semantika Realaus laiko įgaliotajame protokole leidžia siųsti duomenis naudojant standartinę TCP semantiką. Vienintelis standartinio TCP protokolo pakeitimas būtų galinių laiko verčių pridėjimas. Galinės laiko vertės įrašomos siuntėjo TCP antraštėje. Šią reikšmę galima būtų nustatyti TCP protokolo antraštės TTL lauke.

Realaus laiko įgaliotasis protokolas valdo duomenų podėlį, taip kad duomenys šiame podėlyje yra rikiuojami reikiama tvarka. Taip yra paslepiamos problemos iškylančios nevienalyčio tinklo paskutiniame – bevieliame šuolyje. Šis realaus laiko protokolas yra arčiausiai problemų ištakų (*paskutinis šuolis*), todėl įgyvendinat šį protokolą, galima valdyti klaidas bei įgyvendinti greitą paskutiniame šuolyje atsirandančių klaidų korekciją. Kadangi realaus laiko įgaliotasis protokolas ne tik greitai aptinka klaidas, tačiau jas ir ištaiso, vartotojai gali net nežinoti apie tinkle atsirandančias klaidas bei paketų pametimus.

Iš pirmo žvilgsnio šios dvi Realaus įgaliotojo protokolo funkcijos atrodo nesuderinamos: įgaliotoji protokolo dalis užtikrina, jog kiekvienas prarastas duomenų paketas būtų iškart

[9] TTL (Time to Live angl.) – Paketo gyvavimo tinkle laikinė vertė

[10] RTO (Recovery Time Objective) angl. – Reikalaujamas atstatymo laikas (šnekant apie nutrūkusios paslaugos teikimą)

persiunčiamas, tuo pačiu realaus laiko funkcija uždraus bei naikins paketus tinkle, kai šie pasieks kritinę ribą. Gavėjui yra svarbi tik ta informacija kuri dar nėra pasiekusi kritinės ribos. Realaus laiko savybės užtikrina, jog visi duomenys laikomi kaupiklyje yra aktualūs siuntėjui.

Norėdami pasinaudoti realaus laiko duomenų savybėmis, kurias teikia Realaus laiko įgaliotasis protokolas, tiek siuntėjas tiek gavėjas turi kontroliuoti sujungimą. Siuntėjo pusėje, realaus laiko aplikacijos turi vesti dialogą su TCP protokolu, taip, kad šis protokolas gautų žymes apie kritines ribas ir jas naudotu siunčiamiems duomenis formuoti. Aplikacija taip pat turi atsižvelgti į perdavimo greitį, kuriuo TCP protokolas siunčia duomenis.

Gavėjo pusėje aplikacija nuolatos skaito duomenis iš TCP buferių. Jei aplikacija aptinka trūkį nuolatiniame duomenų sraute, šį trūkį aplenkia, toliau tęsdama skaitymą. Tokių būdų matome, kad kritinės ribos reikalingos tam, kad duomenys atvyktų anksčiau, nei jie būtų imti vartoti gavėjo pusėje esančios aplikacijos.

Realaus laiko įgaliotasis protokolas turi keletą privalumų[7]:

1) Lokalizuoti pakeitimai

Realaus laiko įgaliotajam protokolui nėra reikalingi pakeitimai tiek siuntėjo tiek gavėjo pusėje. Pakeitimas yra reikalingas bazinės stoties pusėje, bevielio ryšio sudarymui.

2) Realaus laiko suvokimas

Realaus laiko įgaliotasis protokolas buvo projektuojamas taip, kad turėtų galimybę atsižvelgti į realaus laiko srauto informaciją (pvz. paketų gyvavimo kritines ribas). Paketai gali būti naikinami iš tinklo, nesvarbu koks tvarkaraštis/laikmatis yra naudojamas.

3) Srautas, Galia ir Laikas

Daugeliu atveju siauras srauto praėjimas nevienalyčiame tinkle susidaro dėl paskutiniame šuolyje esančios bevielio ryšio linijos. Dėl šios priežasties yra svarbu užtikrinti, jog visi duomenys, siunčiami bevieliu ryšiu būtų aktualūs gavėjui. Bet kokie nereikalingi/pakartotiniai duomenų persiuntimai ar paketai, kurių galiojimo laikas tinkle yra pasibaigęs, neturėtų būti siunčiami šia beviele ryšio linija. Realaus laiko įgaliotasis protokolas gali persiųsti paketus, kurie kaip manoma yra dingę dėl bevielio ryšio linijoje atsiradusių trukdžių, taip pat gali užtikrinti realaus laiko paslaugų srauto efektyvų veikimą, nepasiekiant kritinių verčių būdingų šiam srautui.

4) Tęstinis ryšys

Esant ne kritinėms sistemos veikimo sąlygoms, Realaus laiko įgaliotasis protokolas palaiko tęstinio ryšio principą. Tačiau yra dvi išimtys, kai šis protokolas gali nevykdyti tęstinio ryšio principo: kai yra priimtas paketas, kurio kritinė riba tinkle yra išsekusi; kai yra tvarkomos srauto pliūpsnio[11] klaidos.

5) Skaidrumo principas

Visas protokolo veikimas yra skaidrus galiniams ryšio vartotojams. Iš esmės protokolas paslepia bevielio ryšio tinklo dalį, padarant tokį vaizdą, jog yra veikiama lėtame vienalyčiame tinke, kuriame paketai bet kuriuo laiko momentu gali būti perstatinėjami. Tik tada kai bevielis ryšys yra nutrūkęs, arba kai yra dideli trukdžiai, galiniai ryšio mazgai yra informuojami apie nepavykusią ryšio sesiją.

6) Persiuntimų planavimas

Srautai, naudojantys ta patį kanalą iš esmės konkuruoja dėl kanalo kai šis tampa prieinamas. Šie abu konkuruojantys srautai gali turėti skirtingą svarbą vartotojui. Užtikrinti reikiamą srautų planavimą naudojamas EDF[12] planuotojas. Jo pagrindinė funkcija išskirstyti srautus pagal jų kritines vertes, ir gavus rezultatus sužymėti srautų prioritetus. Tai yra patogus būdas srautų paskirstymams bei srautų prioritetizavimams tinkle.

Realaus laiko įgaliotojo paslaugų protokolo trūkumai[7]:

1) Reikalauja prieigos prie TCP lygio duomenų

Siekiant tinkamai valdyti paketų persiuntimą bei norint ar paketų gyvavimo laikas tinkle nėra pasibaigęs Realaus laiko įgaliotajam protokolui reikalingas priėjimas prie TCP protokolo antraštės. Toks atvejis yra negalimas kai yra naudojamas šifruoto tęstinio ryšio kanalas (šifruojamas ne tik naudingoji informacija[13], bet ir antraštės).

2) Dalinis patikimumas

Kadangi Realaus laiko įgaliotasis protokolas leidžia efektyviai tinklu siųsti realaus laiko srautą bei turi kritinių verčiu kontrolės mechanizmą, tai šis protokolas naikina paketus

[11] Burst angl. – Srauto pliūpsnis, impulsyvusis srautas

[12] EDF (Earliest Deadline First angl.) – Mažesnę kritinę vertę turintis srautas aptarnaujamas pirmesnis

[13] Payload angl. – Naudingoji informacija

tinkle, kurių kritinė vertė viršiją nustatyta TCP protokolo antraštėje. Realaus laiko įgaliotasis protokolas neturi galimybės užtikrinti absoliutaus patikimumo kanalo realaus laiko duomenų srautui. Ne realaus laiko duomenų srautui nėra kritinių verčių, todėl duomenys niekada nebus naikinami – šio tipo duomenų srautui užtikrinamas labai patikimas ryšys.

2.7 Išvados

Visi, šiame skyriuje paminėti protokolai, turi TCP protokolo korekcijas (*sugebėjimas valdyti būdingus bevielio ryšio nuostolius nevienalyčiame tinkle; sugebėjimas palaikyti realaus laiko duomenų persiuntimo paslaugas*), kurios vienaip ar kitaip pagerina šių protokolų veikimą esant tam tikroms aplinkybėms.

1 lentelė. Protokolų palyginimas

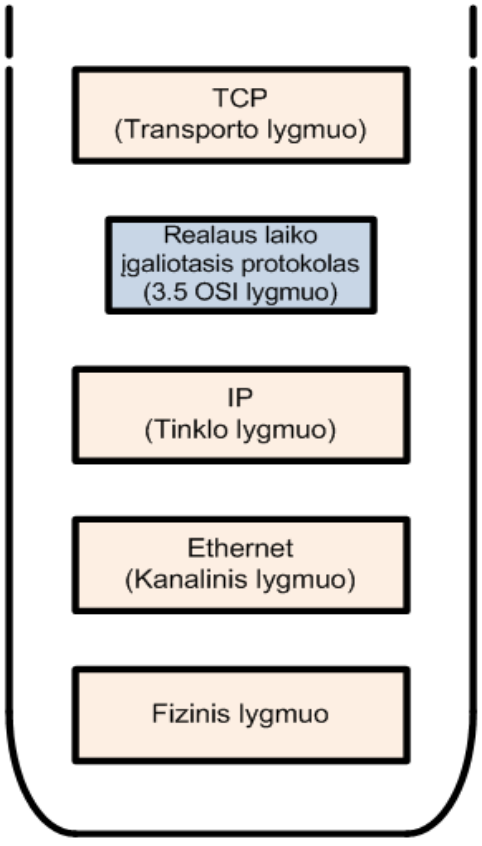
Protokolas	Tęstinis ryšys	Patikimas ryšys	Duomenų persiuntimas	Realaus laiko duomenys	OSI lygmuo
TCP	√	√	Globalus	-	Transporto
I-TCP	-	√	Vietinis	-	Transporto
Kanalinio lygmens retransliavimas	√	√	Vietinis	-	Kanalinis
Šniukštinėjimo protokolas	√	√	Vietinis ir globalus	-	Transporto
Realaus laiko įgaliotasis protokolas	√	Dalinis	Vietinis	√	Transporto

Aukščiau esančioje lentelėje pateiktas trumpas aptartųjų protokolų palyginimas. Kaip matyti iš lentelės dauguma protokolų veikia OSI modelio transporto sluoksnio lygmenyje, palaiko tęstinį ryšį bei patikimą ryšį. Tačiau tik vienas Realaus laiko įgaliotasis protokolas turi galimybę palaikyti realaus laiko duomenis nevienalyčiame tinkle, kuriame fizinis sujungimas (*paskutiniame topologijos šuolyje*) yra bevielis ryšys. Įvertinant jog Realaus laiko įgaliotasis protokolas yra universaliausias tiek siunčiant realaus laiko duomenis, tiek siunčiant ne realaus laiko duomenis, magistro darbo nagrinėjimui pasirinktas būtent šio protokolo analizė. Ruošiant magistro darbo projektinę dalį bus

nagrinėjamas šis protokolas įvertinant šio protokolo galimybes vengti DoS atakų realaus laiko duomenų srauto paslaugoms bei ne realaus laiko duomenų srauto paslaugoms.

3. Realaus laiko įgaliotojo protokolo savybės

Standartinėje heterogeninio tinklo architektūroje duomenų siuntimas vykdomas iš darbinės stoties (*serverio*) ar vartotojo, esančio laidinėje tinklo dalyje. Esant tokiai tinklo architektūrai Realaus laiko įgaliotasis protokolas yra konfigūruojamas ant laidinio ir bevielio tinklų ribos – bevieliame prieigos taške. Šioje tinklo topologijos vietoje esantis bevielis prieigos taškas kaupia įeinančius paketus iš laidinės tinklo dalies ir saugo juos buferinėje atmintyje. Paketų persiuntimas bevieliam mazgui kontroliuojamas pagal esamą ryšio kanalo kokybę: esant prastesnei ryšio kanalo kokybei iš bevielio mazgo dingsta daugiau paketų, todėl negaunami patvirtinimai apie pristatytus paketus prieigos taške. Pagal gautus bei negautus patvirtinimus minėtasis protokolas nustato kurie paketai turi būti pakartotinai siunčiami bevielio ryšio kanalu, o kurie paketai turi būti ištrinami iš buferinės atminties. Susieti patvirtinimą su išsiųstu paketu, kuris ir įtakojo patvirtinimo išsiuntimą naudojamas papildomas informacija – laiko žymė. Ji yra unikali, vienoda tiek patvirtinimui tiek paketui[7,8].



3 pav. Realaus laiko įgaliotojo protokolo padėtis OSI protokolų dėkle

Trečiajame paveiksle parodyta Realaus laiko įgaliotojo protokolo vieta OSI modelio protokolų dėkle. Šiame dėkle minėtasis protokolas reziduoja tarp trečiojo (*tinklo*) ir ketvirtojo (*transporto*) lygmens protokolų. Šis protokolas nėra tiesioginė tęstinio ryšio dalis, tačiau jis naudoja transporto lygmens protokolo duomenis sudaryti tęstiniam ryšiui. Šis protokolas dar kartais laikomas 3.5 lygio protokolu OSI modelio protokolų dėkle, tačiau toks priskyrimas yra subjektyvus todėl tiriamojoje bei mokslinėje literatūroje plačiai nenaudojamas.

Protokolo veikimo modelis bei principas aprašomas dvejomis pagrindinėmis procedūromis: duomenų bei patvirtinimų. Duomenų procedūra iššaukiama kiekvieną kartą priimant paketą iš laidinio tinklo pusės. Patvirtinimų procedūra iššaukiama priimant patvirtinimus, kuriuos apie priimtus paketus išsiunčia mobilus (*bevielis*) tinklo mazgas[8].

3.1 Duomenų procedūra

Visi siunčiami paketai iš laidinės tinklo dalies į mobilų tinklo mazgą, per bevielį prieigos tašką, yra valdomi duomenų procedūros. Paketai turi šiuos duomenis: sekos numerį, žymintį baitus TCP sraute, kuriuos paketas talpina ir laiko žymę, pažyminčią laiko momentą, kuriuo siuntėjas išsiuntė paketą. Realaus laiko srautui naudojamas dar vienas objektas paketams aprašyti – kritinės ribos matas. Šis matas nustato realaus laiko duomenų srauto paketo galiojimo trukmę. Ne realaus laiko srautui ši vertė yra prilyginama begalybei, nes šio tipo srautui nėra būtinas greitas persiuntimas, ribojamas tam tikrų laiko verčių. Tačiau net ir nesant kritinės ribos matui ne realaus laiko duomenų srautui paketai negali tinkle egzistuoti amžinai. Abiejų tipų duomenų srautų mechanizmuose, TCP protokolo logikoje yra įgyvendintas TTL mechanizmas, naikinantis paketą, praėjusį tam tikrą skaičių tinklo šuolių (*maršrutizatorių*). Paketo antraštėje nustatyta TTL vertė, kuri su kiekvienu paketo šuoli tinkle mažėja viena skaitine reikšme. TTL laukui pasiekus nulinę vertę paketo antraštėje, paketas nustoja galioti.[8,9]

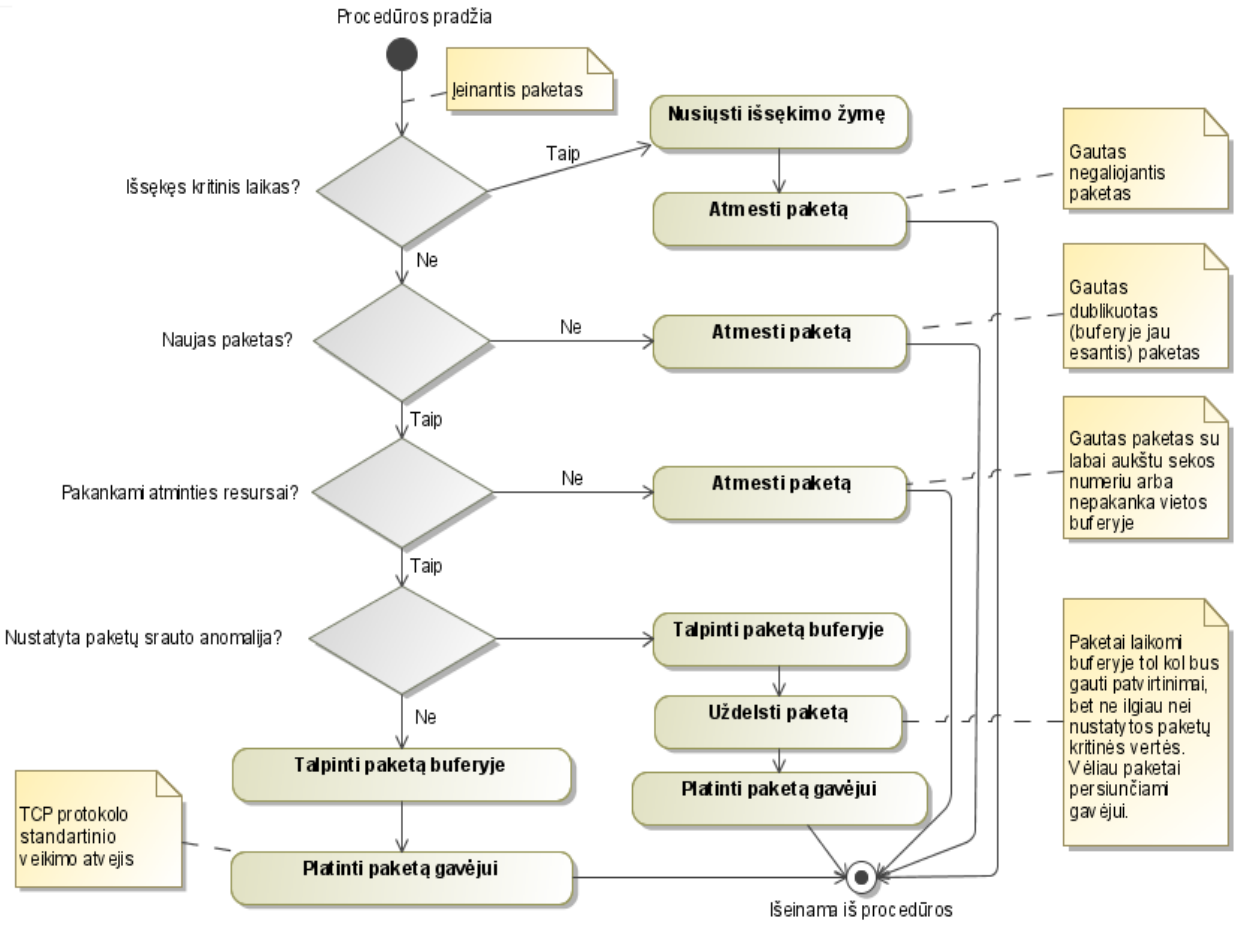
Paketai priimami protokolo teoriškai gali būti skirstomi į tris grupes:

1) Naujas paketas (*paketas dar neegzistuojantis kaupiklyje*). Jei priimamo paketo sekos numeris yra daug didesnis nei paskutinio paketo, kurio patvirtinimas buvo gautas iš mobilaus vartotojo, sekos numerį, priimamas paketas į buferį įrašomas tik tada, kai

yra pakankamai laisvos vietos. Nesant pakankamai vietos buferyje paketai yra atmetami, taip apsaugomas buferis nuo paketų užsipildymo kurie turi didelį sekos numerį. Paketai iš buferio keliauja į kaupiklį, iš kurio yra persiunčiami gavėjui. Jei paketai iš laidinės tinklo dalies atkeliauja didesniu greičiu nei sugebama juos apdoroti kaupiklyje, yra naudojamas srauto kontrolės/spūsties mechanizmas – mažinamas paketų perdavimo greitis.

Kai paketas įrašomas į buferį ir padedamas į kaupiklį, nesant tinkle jokioms anomalijoms paketas persiunčiamas tiesiai gavėjui. Jei pastebimos tinkle anomalijos (*nustatytas paketų praradimas*), kaupiklyje paketai uždelsiami trumpą laiko tarpą, tol kol bus gauti patvirtinimai iš siuntėjo apie priimtus paketus arba tol kol išseks paketų kritinė vertė. Tai nutikus paketai yra naikinami iš tinklo. Šiuo principu grindžiamas realaus laiko paslaugų duomenų srauto apdorojimas[9].

2) Dublikuoti paketai (*paketai jau esantys kaupiklyje*). Dublikuoti paketai arba paketai kurių patvirtinimai yra gauti yra atmetami. Jei gaunamas dublikuotas paketas jau esantis buferyje siuntėjui nebėra siunčiamas patvirtinimas apie gautą paketą. Taip nesumažinamas pralaidumas.



4 pav. Realaus laiko įgaliotojo protokolo duomenų procedūros veikimo schema, panaudojant UML notaciją

3) Paketai su pasibaigusiu galiojimo laiku (*paketas pasiekęs kritinę gyvavimo vertę tinkle*). Kai gaunamas toks paketas, jis yra atmetamas, tačiau patvirtinimas apie gautą paketą išsiunčiamas siuntėjui. Tai yra daroma todėl, jog laikomasi prielaidos jog ne tik šis bet ir kiti siuntėjo buferyje esantys paketai gali turėti išsekusias kritines vertes. Naudojant šią logiką leidžiama kitiems, siuntėjo buferyje esantiems paketams su nepasibaigusia gyvavimo kritine verte pasiekti gavėjo pusę.

Ketvirtame paveiksle parodytas Realaus laiko įgaliotojo protokolo duomenų procedūros veikimo schema panaudojant UML[14] notaciją. Pirmame šios procedūros žingsnyje yra tiriama, ar į tinklo mazgą atėjusio paketo kritinė vertė yra neišsekusi. Paketai su išsekusią kritine verte yra atmetami, patvirtinimai apie tokius atmestus paketus yra išsiunčiami siuntėjui. Jei kritinė vertė nėra

[14] UML (Unified Modeling Language angl.) – Universali modeliavimo kalba

išsekus, antrame žingsnyje yra tikrinama ar atėjęs duomenų paketas yra unikalus. Kai atėjęs paketas yra sistemoje esančio paketo dublikatas, paskiausiai atėjęs paketas yra atmetamas. Jei situacija yra su unikaliu paketu, toliau sistemoje yra tikrinami fizinės atminties resursai (*tinklo mazgo atminties buferis*). Esant pakankamiems tinklo mazgo atminties resursams, ir kai paketas yra siunčiamas ne per įgaliotąjį paslaugų tiekėją, gautasis paketas yra tiesiai persiunčiamas gavėjui. Tai nutinka dažnu atveju, kai ryšio kokybė nevienalyčiame tinkle yra pakankamai gera (*standartinis TCP/IP protokolo veikimo atvejis esant idealioms sąlygoms tinkle*). Kitu atveju, kai ryšio kokybė nėra pakankama paketai siunčiami per įgaliotąjį paslaugų tiekėją: paketai yra talpinami tarpiniame tinklo mazgo buferyje, vėliau kaupiklyje, uždelsiami ir persiunčiami tik paskutiniame tinklo šuolyje, kur yra bevielis ryšio kanalas[7,8,9].

3.2 Patvirtinimų procedūra

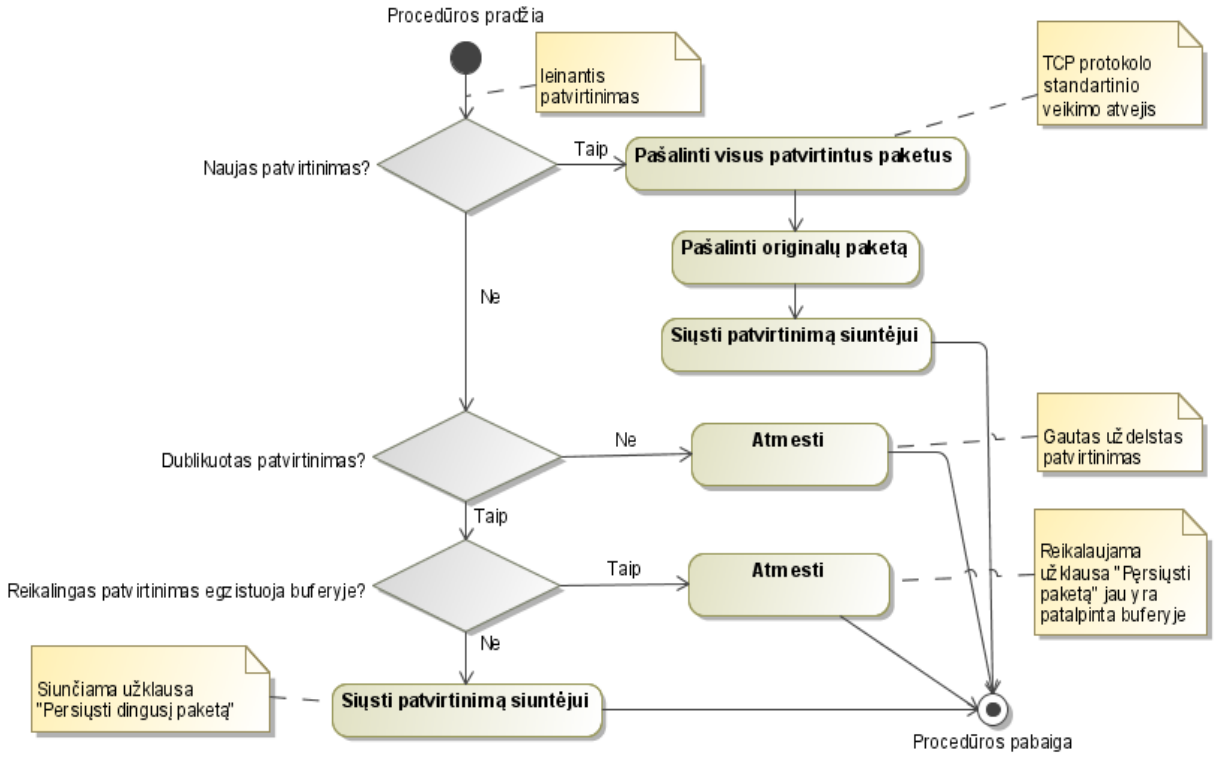
Patvirtinimų procedūra naudojama duomenims gautiems iš kitos nevienalyčio tinklo pusės – mobilaus vartotojo. Iš gaunamų patvirtinimų galima gauti šią informaciją: realaus laiko laikmačio vertę (*RTT*[15]) bevielio tinklo pusėje, informaciją kuriuos paketus galima pašalinti iš kaupiklio nes jie yra priimti gavėjo bei kurie paketai galimai buvo prarasti bevieliame tinklo kanale. Kadangi paketai talpinami kaupiklyje pagal jų sekos eilę, galima manyti, jog prarastas paketas yra tas, kuris dar talpinamas kaupiklyje bei kurio sekos numeris yra mažesnis už paskutinio, patvirtinimą gavusio paketo sekos numerį.

Realaus laiko įgaliotojo protokolo patvirtinimų procedūra gali traktuoti patvirtinimus trejopai:

1) Naujas patvirtinimas. Tai standartinis TCP/IP protokolo veikimo atvejis nesant anomalijoms tinkle. Priimamas patvirtinimas turi didesnį sekos numerį nei prieš tai buvusysis. Šiuo atveju visi patvirtinti paketai ištrinami iš kaupiklio.
2) Dublikuoti patvirtinimai. Dublikuotas patvirtinimas įvyksta kai paketas dingsta laidinėje nevienalyčio tinklo dalyje arba kai pasikeičia paketų tvarka priimančiojoje pusėje. Kadangi tai vyksta tinklo dalyje, šiuos nesklandumus įtakoja tinklo srautų spūstys.

[15] RTT (Round Trip Time angl.) – laiko tarpsnis reikalingas nukeliauti iš siuntėjo iki gavėjo ir sugrįžti atgal

3) Netikras patvirtinimas. Tai toks patvirtinimas, kurio sekos numeris yra mažesnis nei prieš tai buvusio patvirtinimo. Šiuo pašalinamas paketas sukėlęs šį patvirtinimą, o patvirtinimas atmetamas.



5 pav. Realaus laiko įgaliotojo protokolo patvirtinimų procedūros veikimo schema, panaudojant UML notaciją

Penktame paveiksle parodytas Realaus laiko įgaliotojo protokolo patvirtinimų procedūros veikimo schema panaudojant UML notaciją. Pirmame šios procedūros žingsnyje tikrinamas patvirtinimo naujumas. Jei gautas patvirtinimas yra naujas vykdoma standartinė TCP/IP protokolų veikimo schema esant normalioms sąlygoms tinkle: iš kaupiklio šalinami visi patvirtinti paketai, šalinamas originalus paketas, kurio patvirtinimas gautas paskutinis, siunčiamas patvirtinimas siuntėjui apie sėkmingai atliktą procedūrą. Jei įeinantysis patvirtinimas nėra naujas, toliau tikrinama ar tai nėra dublikuotas patvirtinimas. Ne dublikuoto patvirtinimo atveju patvirtinimas yra laikomas kaip uždelstas. Toks patvirtinimas yra atmetamas. Dublikuoto patvirtinimo atveju toliau tikrinama ar egzistuoja reikalaujamas patvirtinimas buferyje. Jei patvirtinimas egzistuoja, vadinasi užklausa „Persiųsti paketą“ jau buvo inicijuota seniau. Jei

neegzistuoja, talpina patvirtinimą „Persiųsti paketą“ buferyje, kuri vėliau persiunčia siuntėjui. [8,9]

3.3 Realaus laiko įgaliotojo protokolo naudojimu grindžiami elementai

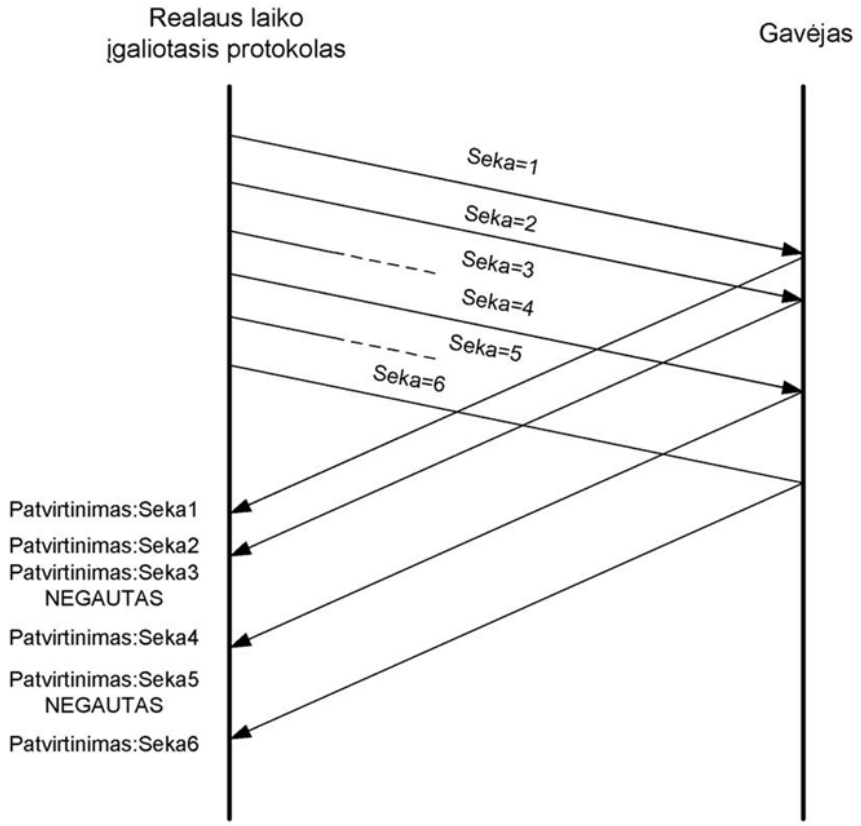
a) Duomenų kaupiklis

Visi priimami paketai yra laikomi duomenų kaupiklyje. Paketas į kaupiklį gali būti priimamas tik tokiu atveju, jei turi sekos numerį, didesnį už prieš tai buvusį patvirtintą paketą. Tai yra daroma norint apsaugot vietą kaupiklyje nutikus paketų persirikiavimui. Jei paketas negali būti talpinamas į buferį automatiškai jis nepatenka ir į kaupiklį. Toks paketas yra atmetamas.

Visi paketai esantys duomenų kaupiklyje lieka jame tol kol būna persiunčiami gavėjui (*mobiliam tinklo mazgui*) arba tol kol paketų kritinės vertės išsenka. Kai tai nutinka paketas pašalinamas iš kaupiklio.

Buferį galima skaidyti į dvi logines dalis: pirmo siuntimo bei pakartotinio siuntimo. Pirmo siuntimo buferyje yra talpinami visi priimami paketai. Kol pakartotinio siuntimo buferis yra tuščias visi paketai iš pirmo siuntimo buferio siunčiami gavėjui. Jei egzistuoja paketai persiuntimo buferyje, bus uždelstas paketų siuntimas iš pirmo siuntimo buferio tol kol neatsilaisvins pakartotinio siuntimo buferis.[10]

Kadangi pirmo siuntimo buferyje paketai išdėstomi pagal jų sekos numerį, nesunku nustatyti, kurie paketai turi būti persiunčiami: paketas, kurio sekos numeris yra žemesnis už paketą, kuris sukėlė patvirtinimą pirmo siuntimo buferyje, turi būti talpinamas į pakartotinio siuntimo buferį ir persiunčiamas.[10,11].



6 pav. Paketų praradimai naudojant sekų numerius

Šeštame paveiksle vaizduojamas paketų praradimas skaičiuojant bei vertinant sekų numerius. Paketai iš pakartotinio siuntimo buferio naudoja panašią techniką nustatyti kada yra poreikis persiųsti paketą: persiuntimas gali prasidėti kai patvirtinimas yra inicijuotas paskutinio paketo, kuris priimtas persiuntimo buferyje arba kai išsenka paskutinio paketo, esančio persiuntimo buferyje RTO vertė[10,11].

b) Tvarkaraštis

Tvarkaraštis yra naudojamas rikiuoti paketus duomenų kaupiklyje. Galimas vienų paketų prioritetizavimas prieš kitus paketus. Tai naudojama kai tinkle esama keletas srautų ir vienas ar keli srautai yra svarbesni už kitus duomenų srautus (*pvz. realaus laiko duomenų srautas turės didesnį prioritetą nei paprastas duomenų srautas*). Tai pat tai yra naudojama kai gavėjo

priimamu duomenų langas yra mažas: negalima perduoti daug duomenų vienu pliūpsniu, todėl reikia tiksliai prioritetizuoti paketus tam kad butų efektyviai išnaudotas kanalas.

Normalioms sąlygoms tinkle esant naudojamas FIFO[16] tvarkaraščio principas. Pirmiausias patalpintas paketas buferyje yra pirmiausiai išsiunčiamas bei patvirtinimas apie šį paketa yra gaunamas pats pirmasis. Tačiau minėta situacija aprašo idealųjį tinklo veikimo modelį be jokių pašalinių trukdžių. Taip pat gali būti naudojamas dar vienas tvarkaraštis, skaitantis paketų ribines išsekimo vertes, bei pagal jas rikiuojantis paketus i buferį bei kaupiklį. Paketai su greit išseksiančiomis ribinėmis vertėmis traktuojami kaip turintys didesnį prioritetą prieš kitus paketus. Tokio tvarkaraščio naudojimas yra naudingas apsisaugant nuo paketų tvarkos pakeitimo laidinėje heterogeninio tinklo dalyje.[9,10]

Tvarkaraštis gali būti naudingas tik tokiu atveju jei atitinkamai yra parinktas persiuntimo algoritmas bevieliu kanalu. Jei persiuntimo algoritmas bevieliu kanalu yra „Pastovus siuntimas“, tada tvarkaraščio naudojimas netenka prasmės, nes į jo duomenis nėra atsižvelgiama. Jei naudojamas algoritmas „Sustoti ir palaukti“, tokiu atveju galima tikėtis neblogų rezultatų.[9,10]

c) Bendriniai patvirtinimai

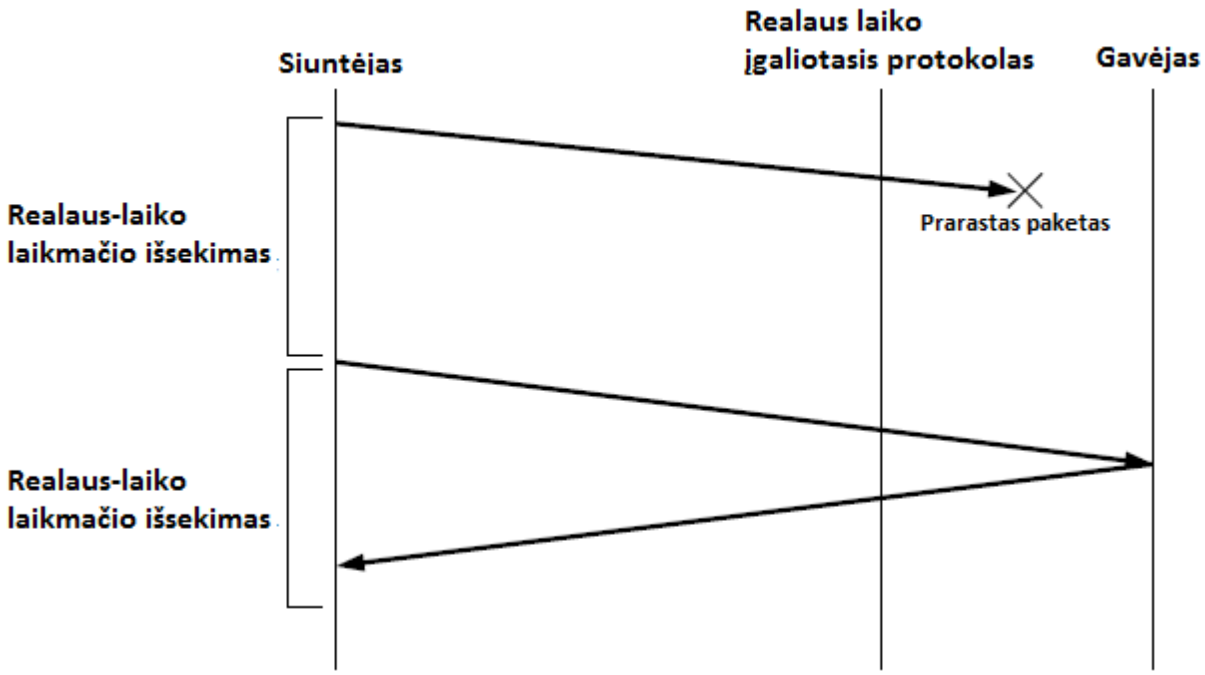
TCP protokolas netenka efektyvumo kai siuntimo kanale yra klaidų, nes visos šios klaidos yra traktuojamos kaip srauto spūstys. Taip nutikus yra mažinamas persiunčiamų duomenų langas, tuo pačiu mažėja ir perduodamų duomenų sparta. Realaus laiko įgaliotasis protokolas turi savybę paslėpti daugelį klaidų bevieliame kanale nuo siuntėjo taip apgaunant siuntėjo pusę, kuri nenutuokia apie esamą situaciją, nemažina persiunčiamų duomenų lango dydį, o kartu nemažėja ir persiunčiamų duomenų perdavimo sparta[1,2,10].

Tačiau esant tokioms klaidoms, kurių nesugeba paslėpti Realaus laiko įgaliotasis protokolas nuo siuntėjo, siuntėjas ims mažint persiunčiamų duomenų lango dydį, o kartu su šiuo veiksmu mažės ir persiunčiamų duomenų sparta. Tai labai aktualu kai susiduriama su duomenų pliūpsnio klaidomis, kur bet koks bandymas persiųsti paketą bevielio ryšio kanalu tampa nesėkmingu per labai trumpą laiko tarpą.[12]

[16] FIFO (First in First out angl.) – Pirmas atėjęs, pirmas išėjęs

Sprendimas iš susiklosčiusios padėties būtų galimas toks: užtikrinti išsiunčiamą patvirtinimą tol kol neišseko RTO vertė. Tačiau toks sprendimas iš esmės pažeidžia ryšio tęstinumo principą. Naudojant Realaus laiko įgaliotąjį protokolą ryšio tęstinumo principas yra pažeidžiamas tik labai trumpais laiko momentais, kai šis protokolas nesugeba paslėpti duomenų pliūpsnio klaidų bevieliame kanale.[12]

Dažniausiai toks atvejis nutinka kai priimtas paketas yra patalpintas duomenų kaupiklyje, o laikmatis pradeda skaičiuoti laiką, kuriam pasibaigus bus siunčiamas bendrinis patvirtinimas siuntėjui. Šis bendrinis patvirtinimas bus išsiųstas jei laikas išseks ir nebus gautas patvirtinimas iš gavėjo.

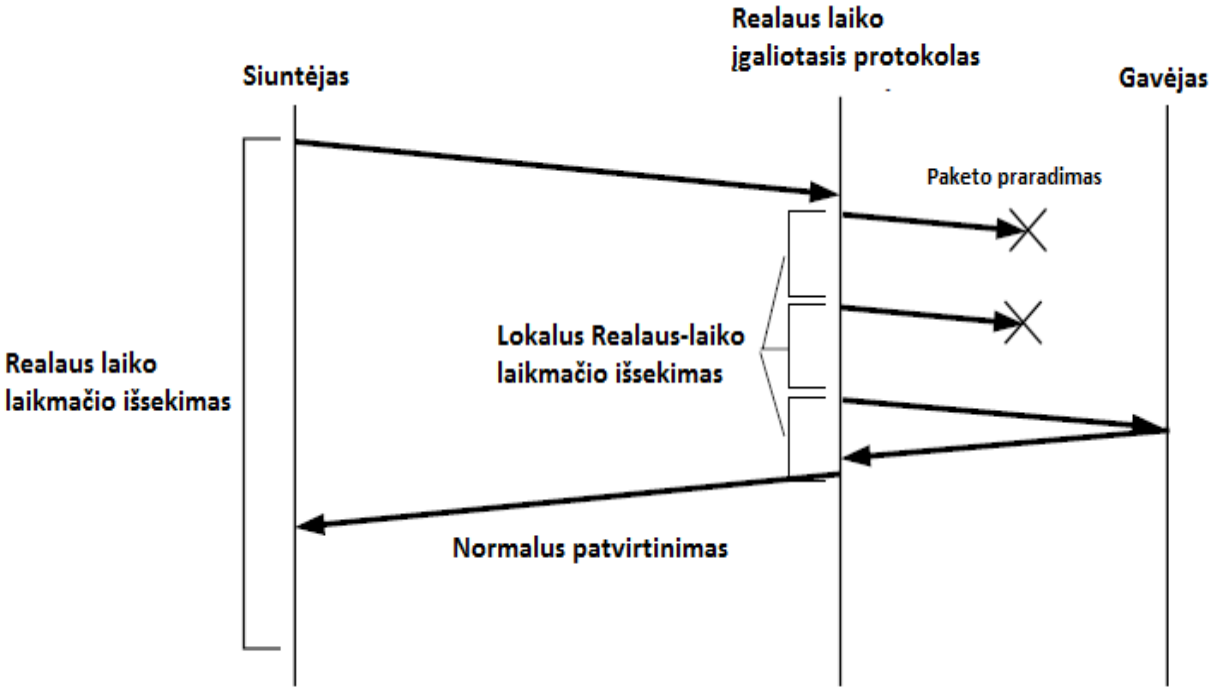


7 pav. TCP protokolo veikimas neesant anomalijoms tinkle

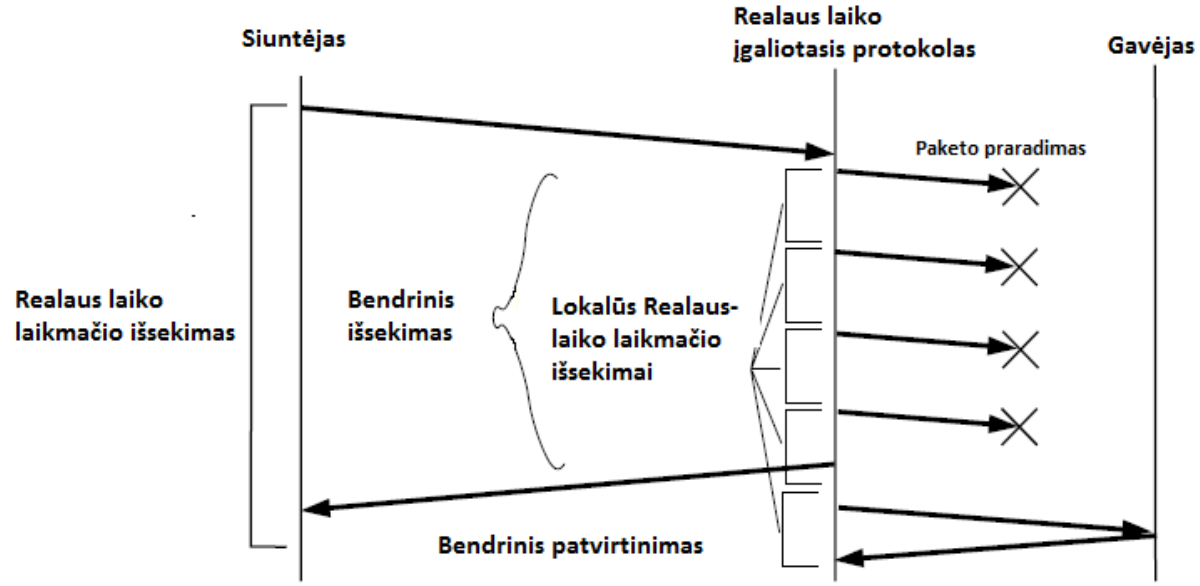
Septintame paveiksle yra parodomas TCP protokolo veikimas neesant anomalijoms tinkle. Tinklo darbas stabilus, mažai klaidų, arba jų praktiškai nėra. Kaip jau minėta anksčiau toks atvejis realiose sąlygose nėra įmanomas – tai tik idealizuota teorinė koncepcija.

Aštuntame paveiksle pateiktas Realaus laiko įgaliotojo protokolo veikimas esant anomalijoms tinkle. Patvirtinimai, kuriuos gauna siuntėjas gali būti dviejų tipų: normalus patvirtinimas, kuris praneša jog paketas yra priimtas gavėjo arba bendrinis patvirtinimas, kuris

informuoja siuntėją, jog išsiųstas paketas nepasiekė tikslo, tačiau paketo persiuntimas nėra būtinas. Šiame paveiksle yra vaizduojamas normalaus patvirtinimo bendrasis atvejis.



8 pav. Realaus laiko įgaliotojo protokolo veikimas esant anomalijoms tinkle



9 pav. Realaus laiko įgaliotojo protokolo veikimas esant anomalijoms tinkle bei naudojant bendrinius patvirtinimus

Devintame paveiksle vaizduojamas Realaus laiko įgaliotojo protokolo veikimas esant anomalijoms tinkle. Šiuo atveju protokolas vietoj normalių patvirtinimų naudoja bendrinius patvirtinimus. Šio tipo patvirtinimų įgyvendinimui reikalingas didesnės talpos buferis. Nepadidinus buferio dydžio, didėjant perdavimo spartai didėja tikimybė, jog kuo daugiau paketų bus atmesta, dėl nepakankamos talpos.

3.4 Išvados

Išanalizavus realaus laiko įgaliotojo protokolo veikimą, nustatyta, jog veikimo principas yra grindžiamas trimis pagrindiniais elementais: duomenų kaupikliu, tvarkaraščiu bei bendriniais patvirtinimais.

Duomenų kaupiklio pagrindinė funkcija yra talpinti paketus ir laikyti juos tol, kol bus gautas patvirtinimas apie sėkmingai įvykdytą duomenų pristatymą. Būtina pažymėti, jog paketai kaupiklyje yra laikomi tol kol sėkmingai yra persiunčiami gavėjui arba iki tol kol išsenka paketų kritinės gyvavimo vertės. Išsekus šioms vertėms paketai yra išmetami.

Tvarkaraščio funkcija yra rikiuoti paketus duomenų kaupiklyje pagal jų svarbą: atsižvelgiant ar paketas priklauso realaus laiko duomenų srautui ar ne realaus laiko duomenų srautui paketui priskiriama prioritetinė reikšmė. Šis tvarkaraščio veikimo principas leidžia efektyviai valdyti minėtuosius duomenų srautus (*realaus laiko ir ne realaus laiko*).

Bendrinių patvirtinimų pagrindinė funkcija yra informuoti siuntėją apie neva teisingai pristatytus duomenis gavėjui. Taip yra daroma dėl to, jog praėjus tam tikram laikui ir vis dar nepavykstant pristatyti duomenų gavėjui šie duomenys praranda aktualumą (*pvz. realaus laiko duomenų srautas*), be to informavus siuntėja apie fiktyvų duomenų pristatymą gavėjui, sumažinama kanalo apkrova, nes nebeaktualūs duomenys nėra persiunčiami. Tačiau naudojant bendrinius patvirtinimus trumpais laiko momentais yra pažeidžiamas tęstinio ryšio principas.

4. Realaus įgaliotojo protokolo galimybių tyrimo prieš DoS atakas modelis

Baigiamajame magistriniame darbe tiriamas DoS atakų poveikis heterogeniniam tinklui. Šiam tyrimui atlikti pasirinktas protokolo modelis, kuris, kaip manoma efektyviai leis pažaboti DoS atakas heterogeninio tinklo bevielėje dalyje. Pirmame šio skyriaus poskyryje pateikiama minėtojo protokolo konfigūravimo seka. Antrajame šio skyriaus poskyryje pateikiama būsimo tyrimo eiga bei modeliuojama aplinka.

4.1 Pagrindiniai DoS atakų tipai

Kompiuteriniais terminais kalbant paslaugų nutraukimo/neigimo ataka (*DoS ataka*) yra ne kas kitas kaip bandymas padaryti tinklo resursus (*kompiuteriai, darbo stotys, kuriose sukasi vartotojams reikalingos aplikacijos, spausdintuvai ir t.t.*) neprieinamus vartotojams. Nors motyvai, atakavimo būdai bei atakuojami resursai gali būti skirtingi kiekvienu DoS atakos atveju bet visas DoS atakas sieja tai, jog kažkas (*vidinio tinklo vartotojas, išorinio, hakeriai ir t.t.*) piktavališkai stengiasi pakenkti.

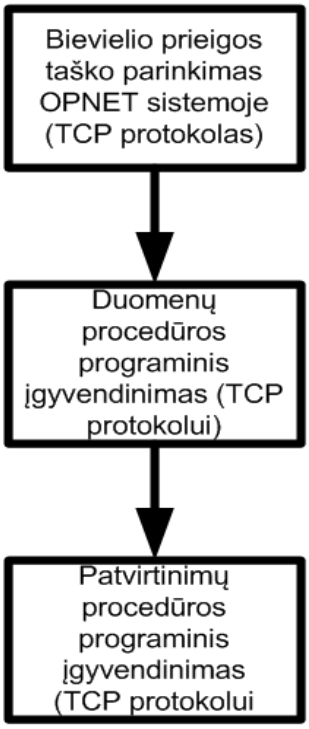
Pagrindinius DoS atakų tipus galima būtų išskirti šiuos:

a) Paskirstyta DoS ataka. Užkrečiama daug kompiuterių ar net ištisų tinklų ir iš jų yra atakuojamas pasirinktas vienas objektas.
b) ICMP užtvindymas. Vienas iš šios atakos pavyzdžių galėtų būti dar kitaip vadinamas Ping užtvindymas, kai yra atakuojamas vartotojas Ping užklausomis.
c) SYN užtvindymas. Taikant šį metodą išnaudojami visi galimi prisijungimai prie atakuojamos sistemos taip ją atkertant nuo galimų kitų vartotojų.
d) Ašaros ataka. Siunčiamas negalimo dydžio IP paketo fragmentas, taip tikintis, jog atakuojamos sistemos operacinė sistema negalės apdoroti duomenų ir užlūš.
e) Žemos spartos paslaugų nutraukimo ataka. Mažinamas persiunčiamų duomenų langas, taip imituojant prastą ryšio kanalą. Ši ataka remiasi TCP protokolo RTO mechanizmo veikimu.

Kaip matyti iš aukščiau aptartų paslaugų nutraukimo atakų baigiamajame magistriniame darbe nagrinėjamai problemai spręsti/tirti pati optimaliausia paslaugų nutraukimo ataka yra žemos spartos paslaugų nutraukimo ataką. Ji ir yra pasirinkta šiame darbe tolesniems tyrimams.

4.2 Realaus laiko įgaliotojo protokolo konfigūravimo/įgyvendinimo seka

Baigiamojo magistrinio darbo maketo modeliavimui pasirinktas tinklų analizės bei kūrimo programinis paketas OPNET. Šis programinis paketas yra universalus – juo galima tirti tiek laidinius, tiek bevielius, tiek heterogeninius tinklus. Be to šis paketas turi plačias išeities informacijos apdorojimo, pateikimo bei atvaizdavimo galimybes.



10 pav. Realaus laiko įgaliotojo protokolo konfigūravimo seka

Dešimtame paveiksle pateiktas magistrinio darbo tyrimo modelio įgyvendinimo procesas. Pirmas žingsnis šiame procese yra bevielio prieigos taško (*bazinės stoties*) parinkimas OPNET

programinio paketo aplinkoje. Parenkamas bevielis prieigos taškas turės TCP protokolo apdorojimo logiką.

Antrajame žingsnyje programiškai įgyvendinama duomenų procedūra TCP protokolui. Šiame žingsnyje pasinaudojant OPNET programinio paketo suteikiamomis galimybėmis, sumodeliuojama įgaliotojo protokolo duomenų procedūra, apdorosianti paketus tinkle esant anomalijoms. Nesant anomalijoms tinkle turi būti atsižvelgta, jog tada bevielio tinklo prieigos taško darbas neturi sutrikti ar neturi būti kaip nors įtakojamas įterptos procedūros. Normalioms sąlygoms esant bevielis prieigos taškas turi veikti standartinio TCP protokolo pagrindu. Tam kad taip nutiktų, reikalingas įprogramuoti loginis jungiklis tarp TCP protokolo bei Realaus laiko įgaliotojo protokolo. Minėtoji duomenų procedūra naudojama kontroliuoti tinklo paketų srautą nuo laidinės tinklo dalies į bevielę tinklo dalį (*paketų atmetimai, pakartotiniai persiuntimai iš siuntėjo*).

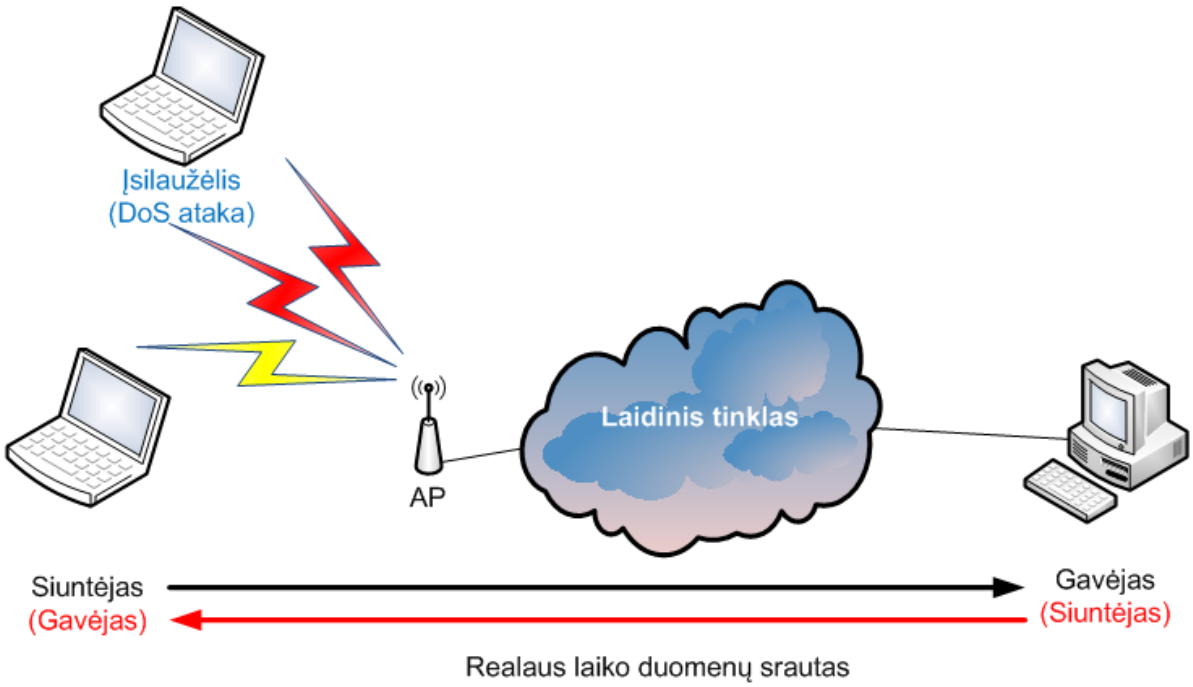
Trečiajame žingsnyje įgyvendinama patvirtinimų procedūra TCP protokolui. Ši procedūra naudojama patvirtinimų, gautų iš gavėjo (*bevielio tinklo kliento*) apdorojimui. Šiai procedūrai įgyvendinti taip pat reikalingas loginis jungiklis, kuris turi būti suprogramuotas taip kad skirtu TCP protokolo logiką nuo Realaus laiko įgaliotojo protokolo logikos. Išskiriant patvirtinimų procedūra nuo TCP protokolo logikos jungikliu pasirinkta traktuoti įeinančių patvirtinimų naujumą: jei įėjęs patvirtinimas yra naujas – naudojamas standartinė TCP protokolo veikimo logika, tačiau jei įėjęs patvirtinimas nėra naujas, logika įjungiama Realaus laiko įgaliotojo protokolo patvirtinimų procedūra, leidžianti efektyviai apdoroti neunikalius patvirtinimus.

Laidinio ir bevielio tinklo sandūroje esantis bevielis prieigos taškas tarsi turi valdymo kanalą abejomis kryptimis – tarp siuntėjo ir bevielio prieigos taško procesas valdomas naudojant duomenų procedūrą, o tarp bevielio prieigos taško ir gavėjo patvirtinimų procedūros. Kaip minėta prieš tai buvusiuose paragrafuose naudojami loginiai jungikliai tarp standartinės TCP protokolo veikimo logikos bei Realaus laiko įgaliotojo veikimo logikos (*procedūrų*). Šie loginiai jungikliai veikia tarsi ir anomalijų filtrai, galintys kaupti bei perduoti informaciją apdorojimui. Sukaupus bei susisteminus šia informaciją galima daryti išvadą apie heterogeniniame tinkle dingstančius paketus, bevielio tinklo kanalo kokybę, laidinėje tinklo dalyje vykstančias kolizijas.

4.3 Realaus laiko įgaliotojo protokolo tyrimo scenarijai

Magistro darbo metu planuojama sukonfigūruoti Realaus laiko įgaliotąjį protokolą OPNET programiniame pakete bei atlikti DoS atakų scenarijų esant dvejiems bandymų scenarijams:

1) Tinklu siunčiamas realaus laiko srautas



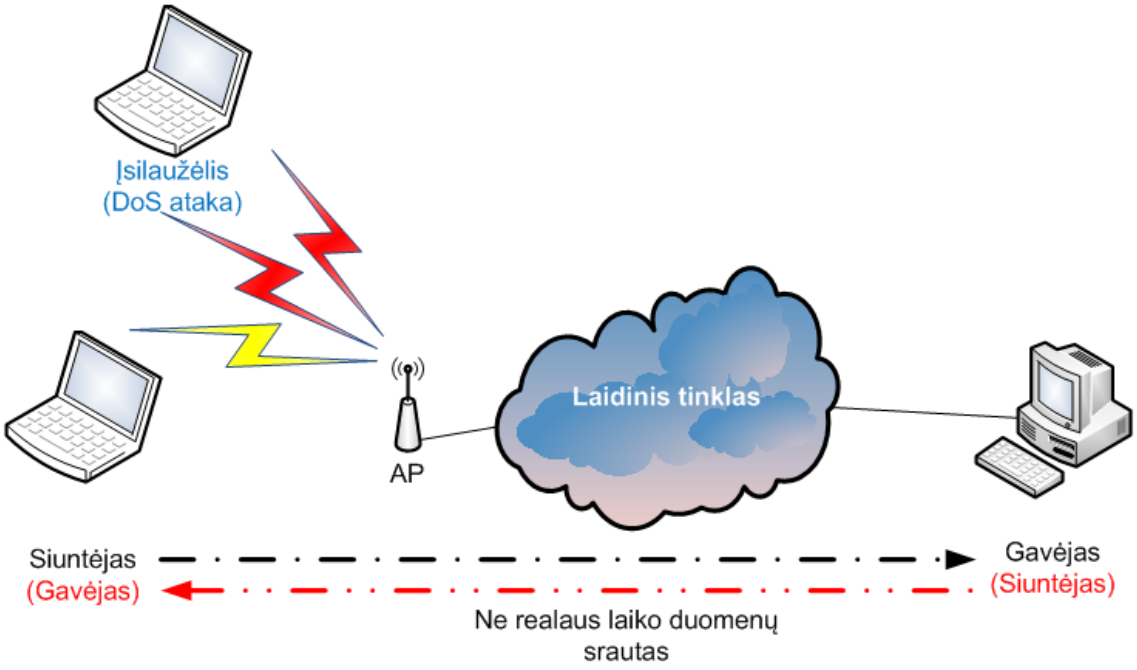
11. pav. Realaus laiko įgaliotojo protokolo tyrimas OPNET modeliavimo paketu, simuliuojant DoS atakas, naudojant realaus laiko duomenų srautą

Modeliavimui bus pasirinktas nevienalytis tinklas, kurio paskutinis šuolis sudarytas iš bevielio ryšio mazgų komunikuojančių tarpusavyje. Naudojami keli/keliolika prieigos taškai, turintys bendrą vienintelį ryšio kanalą į laidinį tinklą. Per šiuos prieigos taškus klientai (*galiniai tinklo mazgai*) komunikuoja su kitais klientais esančiais paskutiniame nevienalyčio tinklo šuolyje bei su kitais ryšio dalyviais esančiais už tinklo paskutinio šuolio ribų (*dalyviai laidinėje tinklo dalyje*).

Realaus laiko duomenų srautas yra daug jautresnis vėlinimui. Šio srauto paketai turi mažas kritines gyvavimo vertes, todėl per ilgai uždelstas paketas nustoja galioti – yra išmetamas iš tinklo. Šiuo veiksmu yra prarandamas paketas iš paketų srauto, skirto realaus laiko paslaugoms

teikti. DoS atakos atveju paketų vėlinimas tampa dažnas reiškinys tinkle. Tokiu atveju naudojant Realaus laiko įgaliotąjį protokolą atakos padariniai realaus laiko duomenų srautui tampa minimalūs: sumažėjus pralaidumui vis tiek galima būtų sėkmingai naudotis paslaugomis, nes laidinio ryšio kanalas nebūtų užkimštas. Kita vertus užblokavus kliento prieigą prie prieigos taško vis dėlto būtų galima įvykdyti pilną DoS ataką, prie šio prieigos taško prijungtiems klientams.

2) Tinklu siunčiamas ne realaus laiko srautas



12. pav. Realaus laiko įgaliotojo protokolo tyrimas OPNET modeliavimo paketu, simuliuojant DoS atakas, naudojant ne realaus laiko duomenų srautą

Tyrimų metu iš esmės bus atliekamas tas pats bandymas tik su skirtinga tinklo srauto modifikacija. Pirmuoju bandymo atveju bus imituojama DoS ataka esant realaus laiko duomenų srautui. Bus simuliuojama, jog įsilaužėlis blogina bevielio ryšio kokybę paskutiniame nevienalyčio tinklo šuolyje, taip sukurdamas Dos ataką: sublogėjus bevielio ryšio kokybei, bevielėje terpėje padidės paketų dingimo tikimybė. Dingstant vis daugiau paketų, TCP protokolo atveju, būtų laikoma, jog paketai vėlina dėl srauto spūsčių. Esant tokiai TCP protokolo logikai, siuntėjas siųs paketus tol, kol šie paketai pasieks gavėją (gavėjas priėmęs paketa siųs

patvirtinimą siuntėjui), taip užkimšdamas bendrąjį ryšio, su laidiniu tinklu, kanalą. Kai ryšio kanalas yra užkimštas, pralaidumas sumažėja arba visiškai dingsta – taip įvykdoma DoS ataka.

Ne realaus laiko duomenų srautas yra ne toks jautrus duomenų vėlinimui. Todėl DoS ataka šiam srautui neturėtų žymios įtakos: paketų siuntimas sulėtėtų iki minimumo, tačiau kadangi duomenys nėra atvaizduojami realiu laiku, vartotojui tai tik pasireikštų išeikvotomis laiko sąnaudomis. Pavyzdžiui siunčiant bylą iš serverio esančio internete bei įvykus DoS atakai heterogeninio tinklo bevielėje dalyje, bylos siuntimas vietoj įprastų 10 minučių, tęstųsi ilgiau kaip 5 valandas. Kaip minėta tai nėra labai geras rezultatas, tačiau šiuo atveju yra prarandami tik laiko resursai – duomenys anksčiau ar vėliau vis tiek pasiekia adresatą.

4.4 Išvados

Šiame skyriuje aptartas realaus laiko įgaliotojo protokolo galimybių prieš DoS atakas tyrimas. Išvardinti keli dažniausiai pasitaikantys DoS atakų scenarijai: ICMP užtvindymas, SYN užtvindymas, paskirstyta DoS ataka, ašaros ataka, žemos spartos paslaugų nutraukimo ataka.

Konstatuota, jog dėl tam tikrų savybių (*RTO mechanizmas, TCP protokolo detalusis nagrinėjimas*) baigiamojo darbo tematikai nagrinėti bus pasirinkta labiausiai planuojama maketą atitinkanti žemos spartos paslaugų nutraukimo ataka.

Paskutiniame skyrelyje aptarti keli bandymu scenarijai. Nustatyta jog bandymams bus naudojamas realaus laiko paslaugų duomenų srautas, taip pat ne realaus laiko duomenų paslaugų srautas.

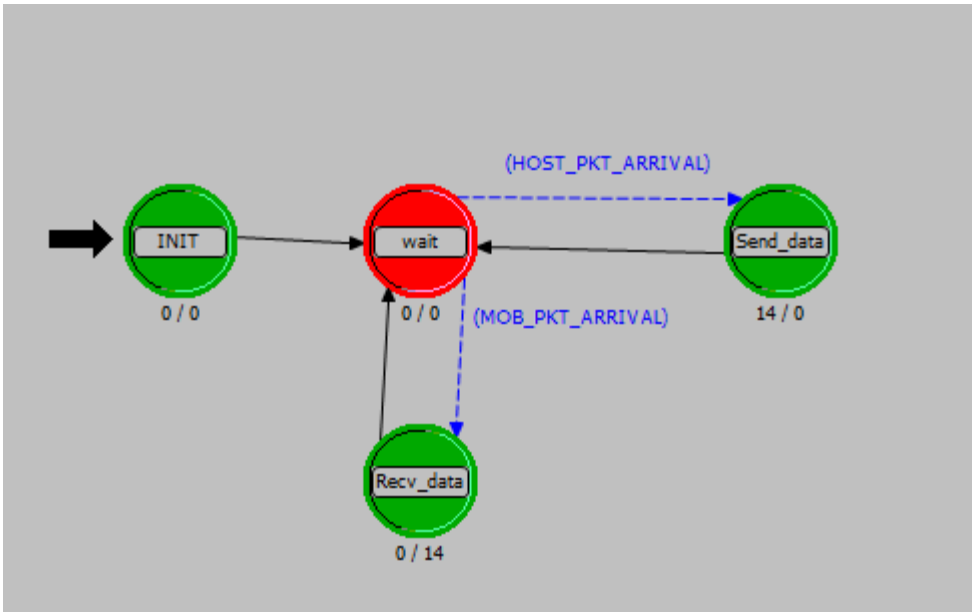
5. Realaus įgaliotojo protokolo galimybių tyrimo prieš DoS atakas modelio realizacija bei tyrimas

Magistrinio darbo modelio praktinei daliai įgyvendinti yra panaudojamas OPNET programinis paketas. Kaip minėta trečiajame skyriuje numatomas dviejų dalių realizacijos modelio tyrimas: naudojant realaus laiko duomenų srautą bei naudojant ne-realaus laiko duomenų srautą. Tam, kad atlikti numanomus tyrimus sukursime tinklo topologiją be įgyvendinto TCP sesijos įgaliojimo protokolo funkcionavimo. Sekančiuose etapuose įgyvendinsime TCP sesijos įgaliojimo protokolo funkcionavimą.

5.1 Tinklo topologijos sudarymas OPNET programiniu paketu

Tinklo topologija turi atspindėti TCP protokolo savybes esant dideliems apkrovimams bei kai yra vykdoma ataka bevieliame tinkle (*slopinamas bevielis radijo ryšys*). Vykdant DoS ataką naudojant standartinę TCP protokolo konfigūraciją numanomi paslaugų, kurios yra teikiamos tinkle trikdžiai.

Žinant, jog tinklo topologijoje paskutinis ryšio šuolis nuo paslaugų tiekėjo iki vartotojo yra bevielis ryšys, sudaryti šį šuolį tinklo konfigūracijoje naudosime OPNET programinio paketo bevielio tinklo darbinė stotį (*wlan_wkstn_adv*). Šioje darbinėje stotyje yra sukonfigūruotas paketų klaidų generatorius (*nes OPNET programinis paketas neturi galimybės simuliuoti paketų klaidas bevieliame kanale – visi išsiųsti paketai yra traktuojami kaip geri ir be bitų klaidų*). Paketų klaidų generatoriaus principas yra gana primityvus: jis tiesiog simuliuoja paketų praradimą gautus paketus išmesdamas iš tinklo. Paketų išmetimui iš tinklo naudojamas procentinis matas, nurodant kiek (*procentaliai*) turi būti išmesta paketų. Toliau yra pateikiamas paketų klaidų generatoriaus procesų modelis.



13. pav. Paketų klaidų generatoriaus procesų modelis

Kaip matyti iš tryliktojo paveikslo paketų klaidų generatoriaus procesų modelis sudarytas iš keturių būvių: INIT – proceso iniciacijos būvis; wait – laukimo būvis; Send_data – duomenų siuntimo būvis; Recv_data – duomenų gavimo būvis. Procesas yra inicijuojamas būvio INIT. Procesas lieka wait būvyje tol kol paketas atvyksta iš žemesnio arba aukštesnio sluoksnio. Priklausomai nuo paketo krypties (*iš aukštesnio sluoksnio į žemesnį arba atvirkščiai*) naudojami Send_data ir Recv_data būviai. Kiekviename iš šių būvių yra naudojamos instrukcijos nurodančios ar išmesti paketą iš tinklo ar jį palikti tinkle. Instrukcijos naudoja prieš tai jau aptartą kintamąjį – paketų klaidų procentinį matą. Žemiau yra pateiktas loginis paketų klaidų generatoriaus modelis:

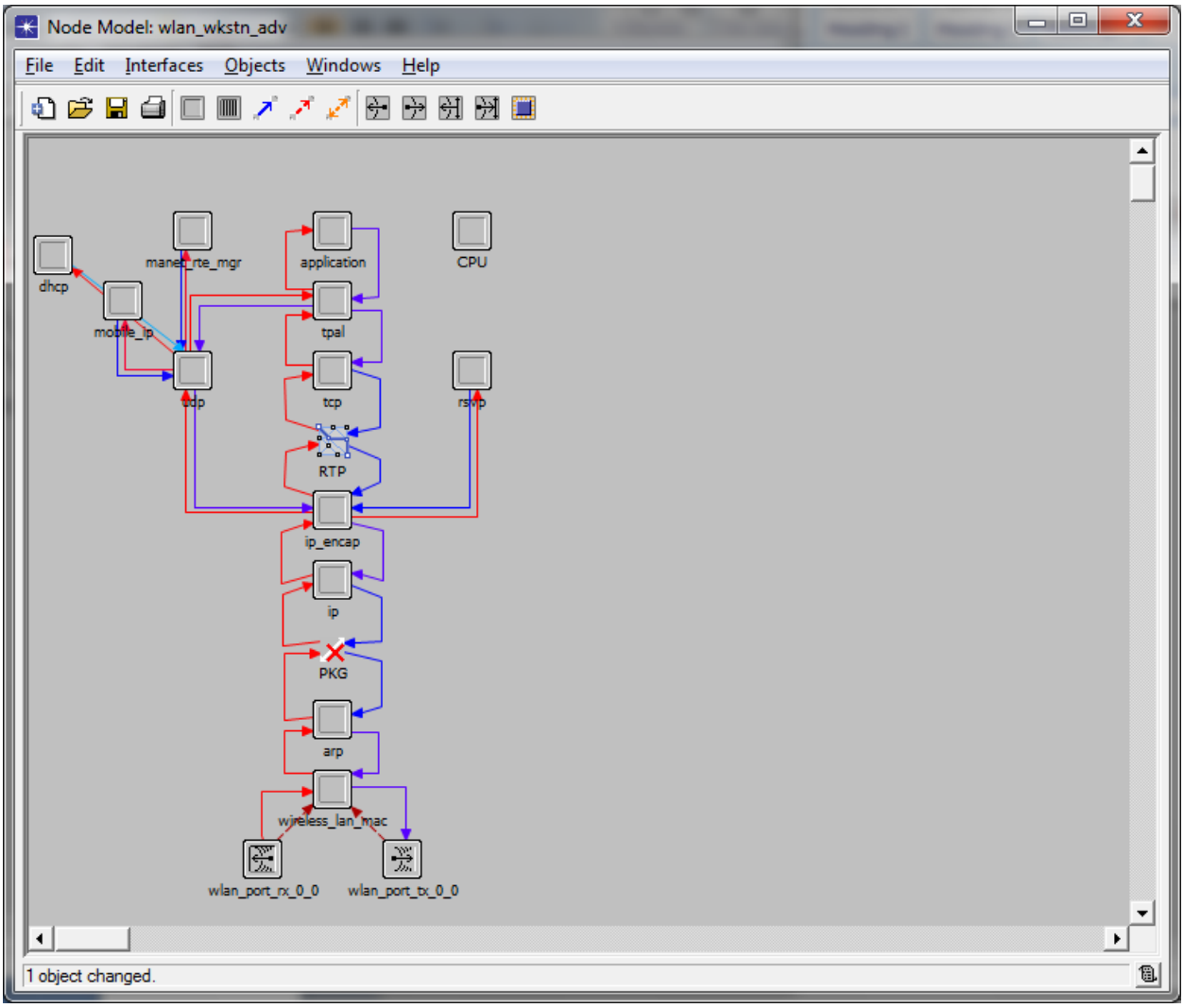
1) Gauti TCP duomenis.
2) Patikrinti ar gautas paketas yra SYN[17] arba FIN[18] tipo.
   a) Jei paketai SYN arba FIN tipo šiuos paketus perduoti į sekantį sluoksnį.
   b) Kitu atveju tikrinti ar paketų išmetimo vertė nėra viršyta. Jei vertė neviršyta perduoti paketus į aukštesnį sluoksnį.

[17] SYN tipo paketas yra siunčiamas sesijos iniciavimui

[18] FIN paketas yra siunčiamas patvirtinant siunčiantysis nebesinaudos sesija (nustos siųsti/gauti duomenis)

c) Nustačius kad yra viršyta paketų išmetimo vertė (*funkcija skaičiuojanti paketus nustato, jog gražinama vertė yra didesnė už nustatytąją išmetimo vertę*) šie paketai yra išmetami iš tinklo.

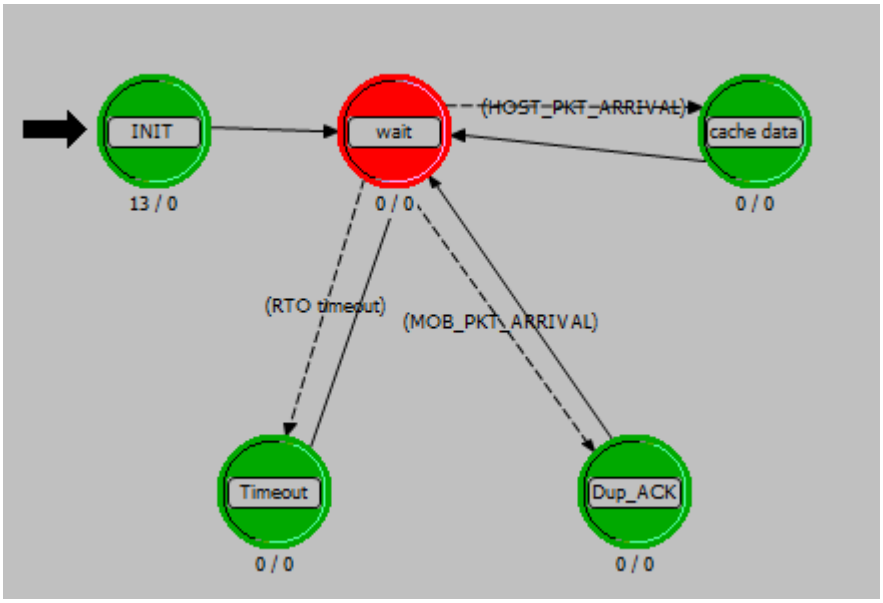
Sekančiame paveiksle yra pateikiamas OPNET programinio paketo bevielio tinklo bevielės darbinės stoties mazgo modelis. Kaip matyti iš paveikslo tinklo mazgo modelis sudarytas iš daugybes procesų modelių: application, tpal, tcp, ip_encap, ip ir t.t. Iš šio modelio yra matyti asociacija į OSI modelio lygmenis.



14. pav. Bevielės tinklo darbinės stoties mazgo modelis OPNET programiniame pakete

Taip pat iš šio paveikslo matyti jog įgyvendintas paketų klaidų generatoriaus procesų modelis. Šis modelis paveiksle yra žymimas PKG ikona. Paketų klaidų generatoriaus procesų modelis yra virš ARP procesų modelio ir vienu lygiu žemiau už IP procesų modelį.

Įgyvendinus paketų klaidų generatoriaus procesų modelį bevielio tinklo darbinėje stotyje sekantis žingsnis yra sukonfigūruoti įgaliotąjį protokolą TCP procesų modelyje. Kaip jau buvo minėta 3.3 skyriuje modifikuojami-konfigūruojami šie elementai: duomenų kaupiklis, tvarkaraštis, bendriniai patvirtinimai.



15. pav. TCP įgaliojimo procesų modelis OPNET programiniame pakete

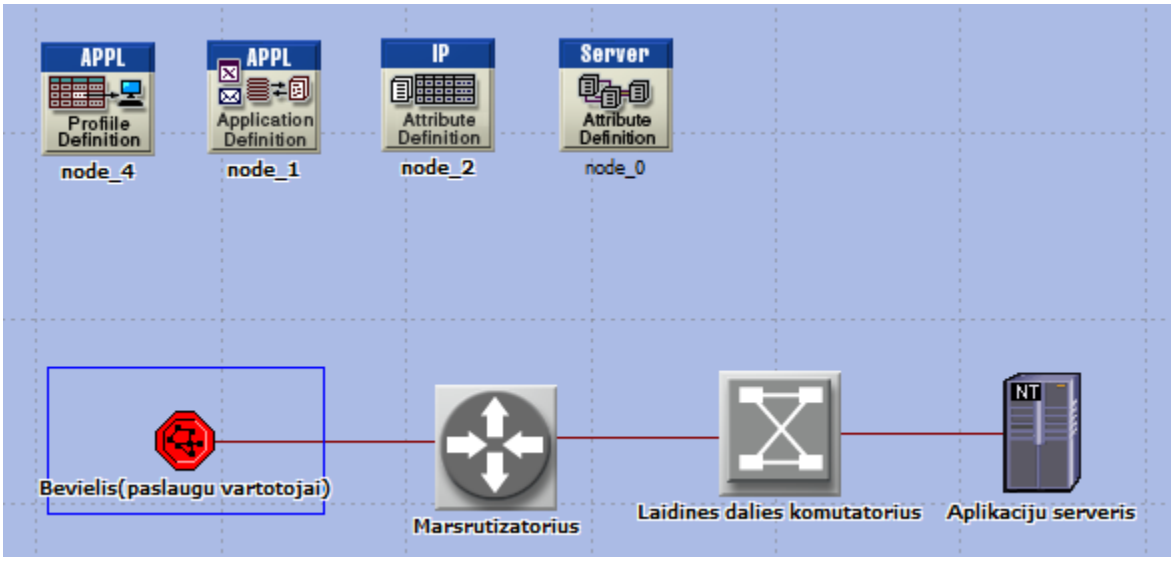
TCP įgaliojimų modelis turi penkis būvius: INIT ir wait – proceso iniciacijos būvis bei laukimo būvis (*analogiški paketų klaidų generatoriaus procesų modelio būviams*), cache_data – duomenų kaupiklio būvis, Timeout – tvarkaraščio (*išsękimo*) būvis bei Dup_ACK – dublikuotų patvirtinimų būvis. Žemiau yra pateiktas loginis TCP įgaliojimų modelis:

1) Procesų modelis yra inicijuojamas INIT būvio bei yra wait būvyje tol kol yra gaunamas (*iš aukštesnio lygio*) arba siunčiamas (*į žemesnį lygį*).
2) Jei nėra paketų praradimo paketas yra išsiunčiamas į kitą lygmenį (*standartinis TCP veikimas*). Visi priimami paketai talpinami į podėlį. Paketas į podėlį priimamas tik tuo atveju jei jo sekos numeris yra didesnis už prieš tai priimto paketo numerį (*tai yra daroma tam, kad išvengti paketų persigrupavimo*). Podėlį sudaro dvi dalys. Pirmo

siuntimo bei pakartotinio siuntimo. Antro siuntimo dalis sesijos pradžioje visada būna tuščia ir pasipildo ji tik kai yra aptinkamas paketų praradimas (*paketas, kurio sekos numeris yra žemesnis už paketą, kuris sukėlė patvirtinimą pirmo siuntimo buferyje, turi būti talpinamas į pakartotinio siuntimo buferį ir persiunčiamas*).

3) Timeout procesų modelis yra naudojamas rikiuoti paketams podėlyje. Jis įvertina srautus (*realaus laiko paslaugų duomenų srautas; ne realaus laiko paslaugų duomenų srautas*) ir surikiuoja juos pagal prioritetus (*realaus laiko duomenų srautas visada turės didesnį prioritetą*). Šis tvarkaraštis taip pat skaito ribines paketų gyvavimo vertes ir šioms vertėms išsekus išmeta paketus iš tinklo.
4) Dublikuoto patvirtinimo būvis atlieka patvirtinimų siuntimo funkciją. Jei paketas jau buvo patalpintas pakartotinio siuntimo dalyje ir jei jis jau buvo išsiųstas yra laukiama patvirtinimo apie gavimą. Šis patvirtinimas yra aktualus tol, kol neišseko paketo kritinė gyvavimo vertė. Išsekus šiai vertei paketas yra naikinamas iš tinklo (*aptartasis atvejis yra vienintelis kai pažeidžiamas tęstinumo principas*) ir TCP įgaliotasis protokolas siunčia bendrini patvirtinimą gavėjui taip nustatydamas jog duomenų nebūtina persiųsti. Normaliu atveju, neišsekus vertei yra naudojami normalūs patvirtinimai.

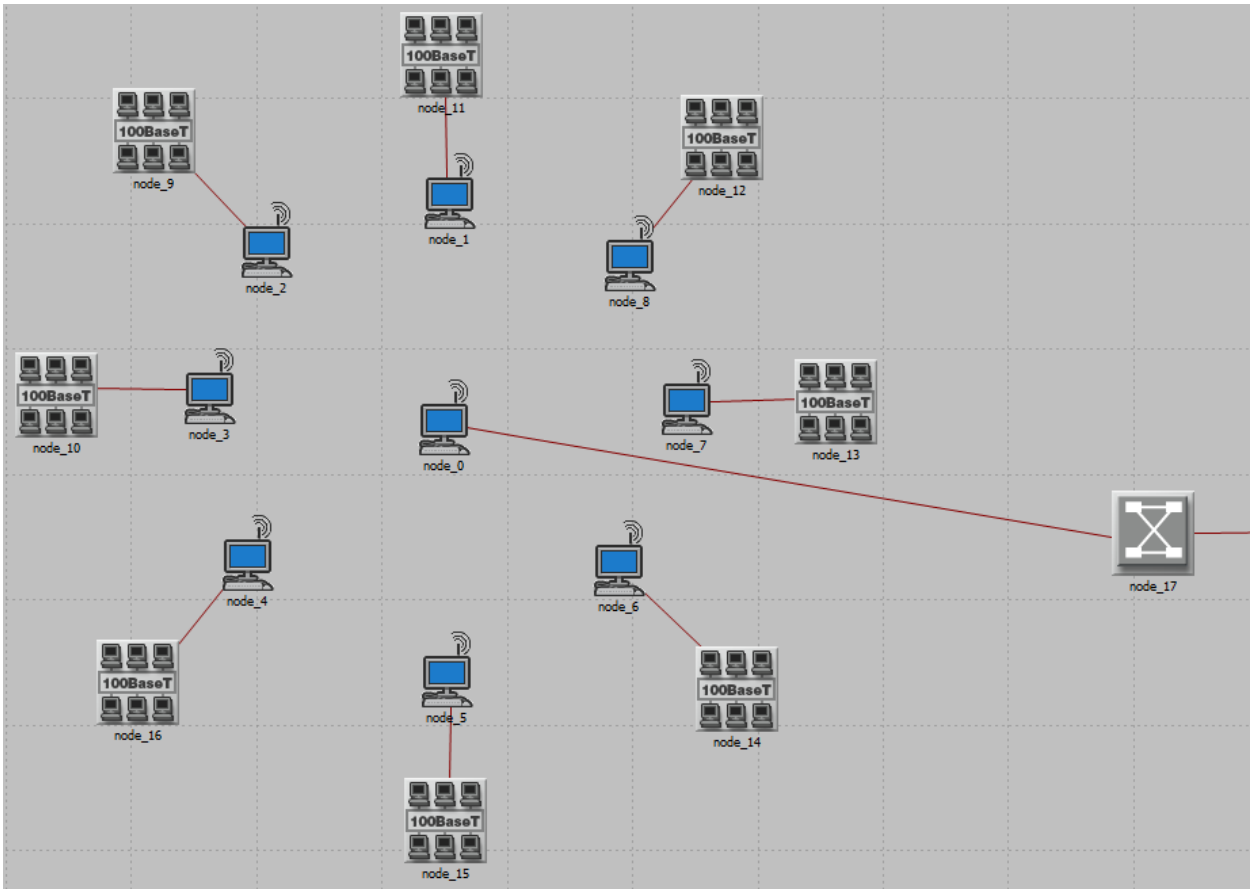
Sekančiuose paveiksluose parodoma galutinė tinklo topologija naudojama šio magistrinio darbo tyrime. Tinklo topologijai sudaryti buvo pasirinkti šie OPNET programinio paketo elementai: bevielio tinklo agentas (*wlan_wkstn_adv*), komutatoriai (*eth4_switch, eth16_switch*), maršrutizatorius (*ethernet_router_adv*), kuris kartu atlieka ir ugniasienės funkcijas, bei aplikacijų serveris (*ethernet_server_adv*).



16. pav. Tinklo topologija

Šešioliktame paveiksle vaizduojama tinklo topologija naudojama baigiamojo darbo tyrimams. Kaip matyti iš paveikslo taip pat tinklo topologijos konfigūracijai naudojami keturi konfigūracijos moduliai: interneto protokolo konfigūracinis modulis (*IP Attribute definition*), aplikacijų konfigūracinis modulis (*Application definition*), profilio konfigūracinis modulis (*Profile definition*), bei serverio atributų konfigūracinis modulis (*Server atribute definition*). Naudojantis šiais keturiais konfigūraciniais moduliais aprašomi tinklo srautai vidinėje tinklo dalyje, taip pat nurodomas srauto poreikis į išorinį tinklą, nustatomos serverio aplikacijos bei jų generuojami duomenų srautai. Serverio atributų modulyje nustatomi serverio darbiniai parametrai, sudedami IP protokolo nustatymai įgalinantys efektyvų serverio veikimą. Taip pat iš šio paveikslo matyti, jog tinklą sudaro vienas potinklis[19], kuris yra pavaizduotas žemiau esančiame paveiksle.

[19] Subnet angl. – potinklis, sudedamoji tinklo dalis



17. pav. Tiriamojo tinklo potinklio detalusis vaizdas

Potinklis su tinklu yra sujungtas per komutatorių, kuris turi sąsaja su maršrutizatoriumi. Potinklyje yra aštuoni ethernet tinklai (*eth_switched_lan_adv*), kurių kiekviename yra po 200 vartotojų (*kompiuterių*). Šie tinklai turi po vieną prie jų prijungtą bevielio ryšio darbinę stotį (*wlan_wkstn_adv*), kuri yra įgalinta ir maršrutizuoti. Visos šios stotys turi radijo ryšį su devintąja stotimi, kuri yra prijungta prie komutatoriaus.

Žemiau esančioje lentelėje yra pateiktas visų, tinklo topologijoje naudojamų elementų sąrašas su kiekybiniais rodikliais.

2 lentelė. Tinklo topologijoje naudojami OPNET įrenginiai/elementai bei konfigūracinai moduliai.

Elemento(įrenginio) pavadinimas	**Elemento OPNET kodas**	**Kiekis, vnt.**
Bevielio ryšio darbinė stotis	*wlan_wkstn_adv*	9
Maršrutizatorius	*ethernet_router_adv*	1
Komutatoriai	*eth4_switch*	1
	eth16_switch	1
Aplikacijų serveris	*eth_server*	1
Ethernet komutuoti tinklai	*eth_switched_lan_adv*	8
Profilio konfigūracinius elementas	*Profile_Config*	1
Aplikacijų konfigūracinius elementas	*Application_Config*	1
IP konfigūracinius elementas	*IP_Config*	1
Serverio konfigūracinius elementas	*Server_Config*	1

Kaip matyti iš aukščiau pateiktų tinklo topologijos paveikslų bei lentelės tinklo veikimo principas būtų:

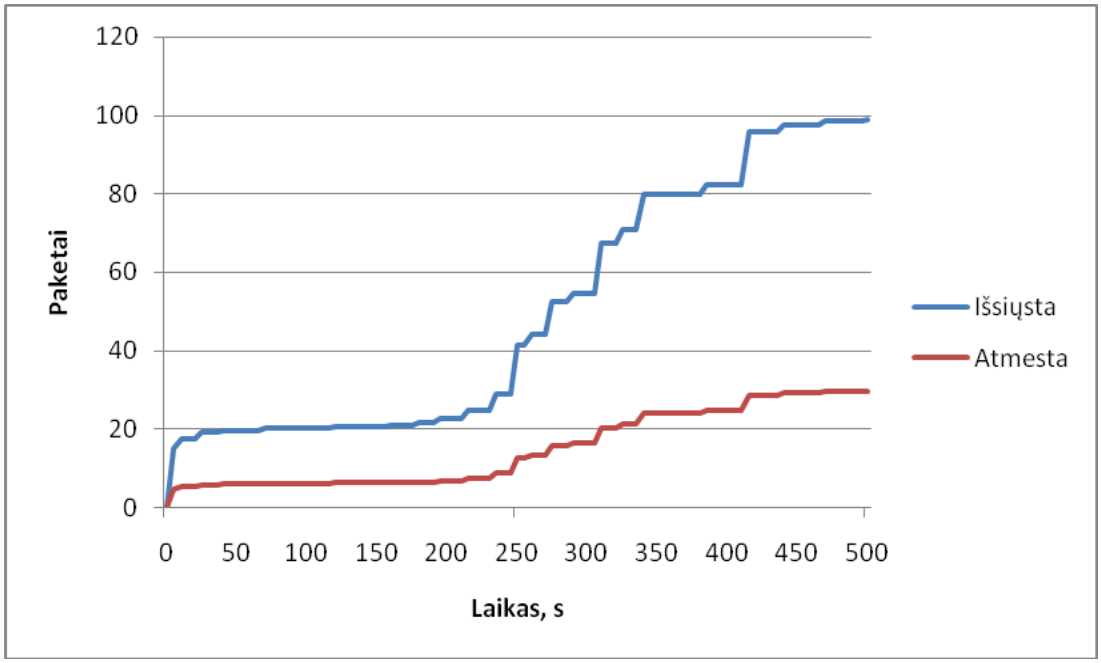
1) Serveryje sukonfigūruotos dviejų tipų paslaugos: realaus laiko duomenų srauto bei ne realaus laiko duomenų srauto. Vartotojams suteikiama prieiga prie šių paslaugų.
2) Kiekviename ethernet potinklyje (*eth_switched_lan_adv)* yra po 200 vartotojų kurie generuoja srautą – naudoja paslaugas.
3) Kiekvienas ethernet potinklis išeina į išorę per prie jo prijungtą bevielio ryšio darbinę stotį. Ši darbinė stotis turi nustatytą maršrutą į aplikacijų serverį: centrine viso potinklio bevielė darbinė stotis (*wlan_wkstn_adv)* ; maršrutizatorius (per komutatorių (*eth4_switch*)), aplikacijų serveris (per komutatorių (*eth16_switch*)).
4) Potinklio centrinėje bevielėje darbinėje stotyje yra sukonfigūruotas paketų klaidų generatorius (*kuris imituos bitų klaidas atsirandančias bevielio ryšio terpėje*) bei TCP įgaliojimas (*kuris leis mažinti paketų klaidų generatoriaus veikimo poveikį tinkle*).

Sukūrus tinklo topologiją bei sukonfigūravus visus reikiamus tinklo topologijos mazgus sekančiame skyriuje atliekamas tyrimas norint patvirtinti arba paneigti teorines doktrinas (*teiginius*) išdėstytas šio magistrinio darbo aiškinamojoje dalyje.

5.2 Tinklo topologijos normalios būsenos tyrimas

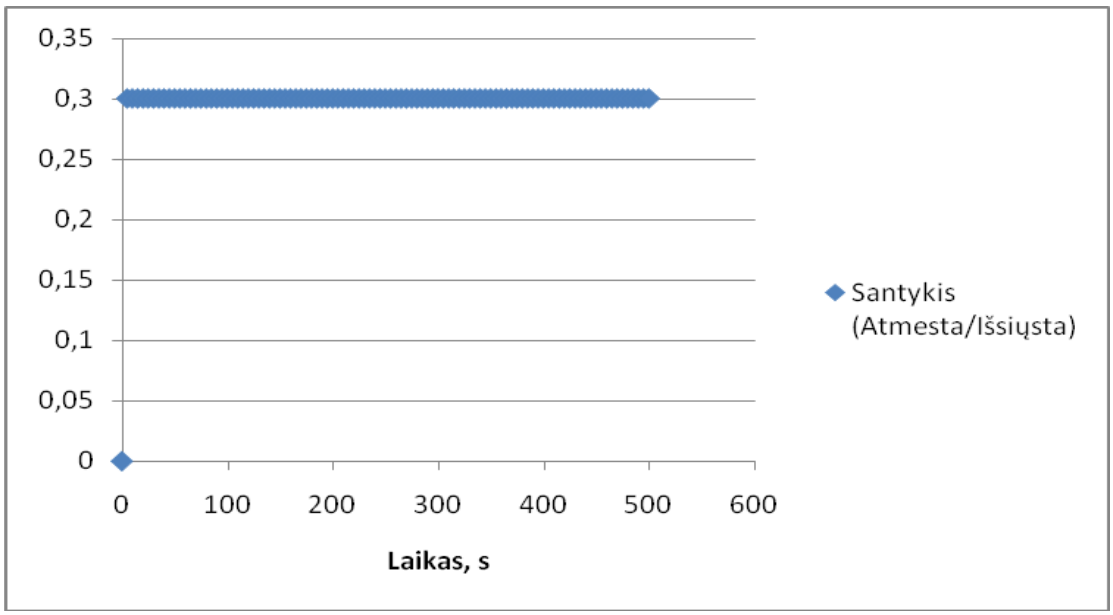
Prieš pradedant tirti sukonfigūruotą tinklo topologiją pirmiausia bus atliktas tinklo normalios būsenos tyrimas. Šį tyrimą sudaro trys susidedamosios dalys: paketų klaidų generatoriaus tyrimas (*siekiant įvertinti ar paketų generatorius korektiškai atmeta paketus pagal pasirinktus parametrus*); tinklo tyrimas kai tinkle vartojamos realaus laiko duomenų paslaugos ir nėra jokių paketų praradimų (*idealus modelis*); tinklo tyrimas kai tinkle vartojamos ne realaus laiko duomenų paslaugos (*idealus modelis, tik jau su ne realaus laiko duomenų srauto paslaugomis*). OPNET įrenginių/elementų konfigūracinė lentelė pateikta šio magistrinio darbo dešimtajame skyriuje.

Paketų klaidų generatoriuje nustatoma 30% paketų praradimas. Tokiu atveju teoriškai turėtų būti atmetami 3 iš 10 paketų atėjusių iš bevielio tinklo į centrinę tinklo potinklio bevielę darbinę stotį. Simuliacijai naudojamas ne realaus laiko paslaugų duomenų srautas (*FTP paslauga su nustatytu srautu 350 kbit/s*). Simuliacijos trukmė - 500 sekundžių.



18. pav. Paketų klaidų generatoriaus tyrimas

Kaip matyti iš aštuonioliktojo paveikslo yra pateiktas simuliacijos išsiųstų paketų bei atmestų paketų grafikas. Iš šio grafiko matyti, jog paketų praradimas turi tendenciją didėti didėjant siunčiamų paketų skaičiui. Tikslios vertės kiek sparčiai didėja atmestų paketų didėjant išsiųstų paketų skaičiui iš šio grafiko nustatyti negalime, todėl norint įsitikinti, jog paketų klaidų generatorius veikia teisingai sugeneruojamas dar vienas grafikas.

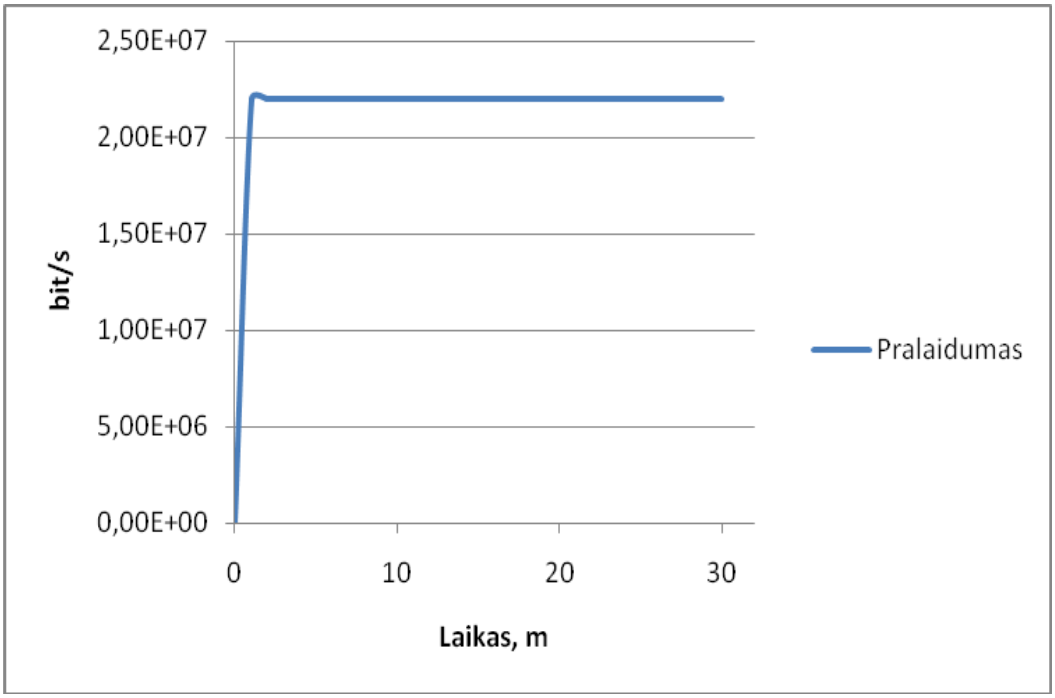


19. pav. Grafikas iliustruojantis atmestų ir išsiųstų paketų santykį

Devynioliktajame paveiksle yra parodytas grafikas vaizduojantis atmestų ir išsiųstų paketų santykį. Kaip matyti iš šio grafiko, kiekvienu laiko momentu (*išskyrus pradinį – nulinį laiko momentą, nes duotuoju laiko momentu siunčiamų paketų vertė yra lygi nuliui*) atmestų/išsiųstų paketų santykis yra lygus 0,3 vertei. Vadinasi tai atitinka iškeltą sąlygą, jog trys iš dešimties arba 30% visų siunčiamų paketų turi būti atmetama.

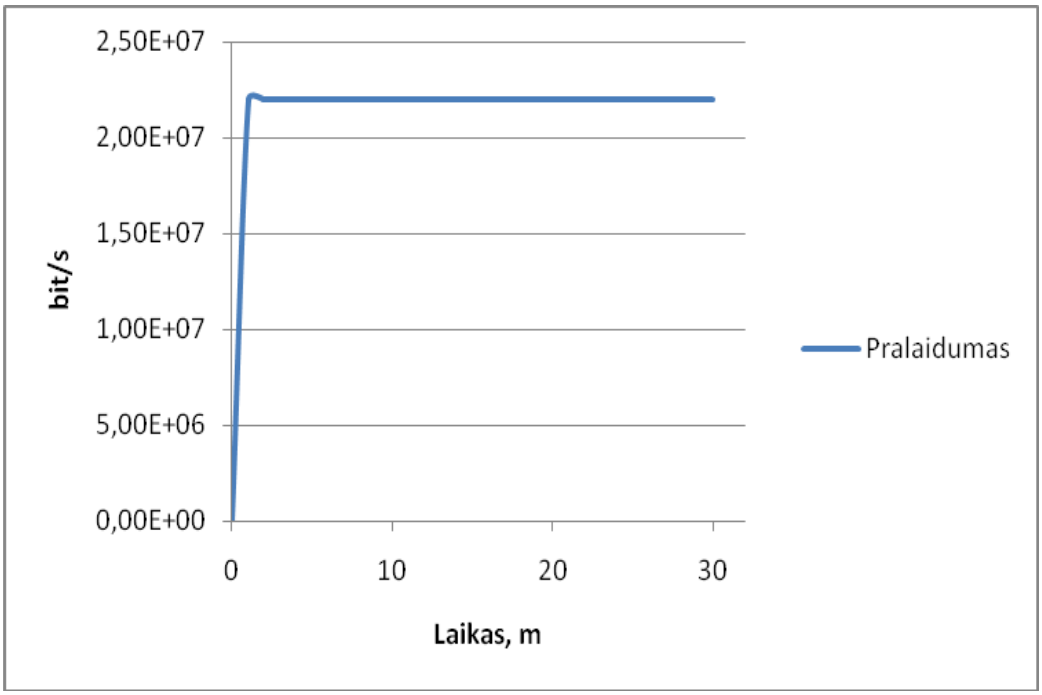
Nustačius bei įsitikinus, jog paketų klaidų generatorius veikia nepriekaištingai toliau atliekamas tinklo tyrimas esant normalioms sąlygoms (*idealusis atvejis*). Pirmiausia tyrimui bus naudojamas realaus laiko paslaugų duomenų srautas. Tinklo aplikacijų modulio konfigūraciniuose nustatymuose nurodoma, jog bus naudojamas „Realaus laiko paslaugų duomenų“ srauto profilis. Prieš tai šis profilis sukuriamas, jame nurodant, jog realaus laiko duomenų srauto paslauga yra naudojamas realaus laiko duomenų srauto multimedijos transliacija. Simuliacija atliekama 30 minučių laikiniam intervalui. Simuliacijos metu renkama

pralaidumo statistika ant ryšio linijos, kuri jungia centrinę tinklo potinklio bevielę darbinę stotį su maršrutizatoriumi.



20. pav. Realaus laiko duomenų srauto pralaidumo grafikas (idealiuoju atveju)

Dvidešimtajame paveiksle matomas realaus laiko srauto duomenų pralaidumo grafikas. Bendras kanalo srautas yra lygus visais laiko momentais (*išskyrus pradinius laiko momentus, nes yra sukonfigūruota, jog srautas bus generuojamas/naudojamas ne nuo nulinio laiko momento*).

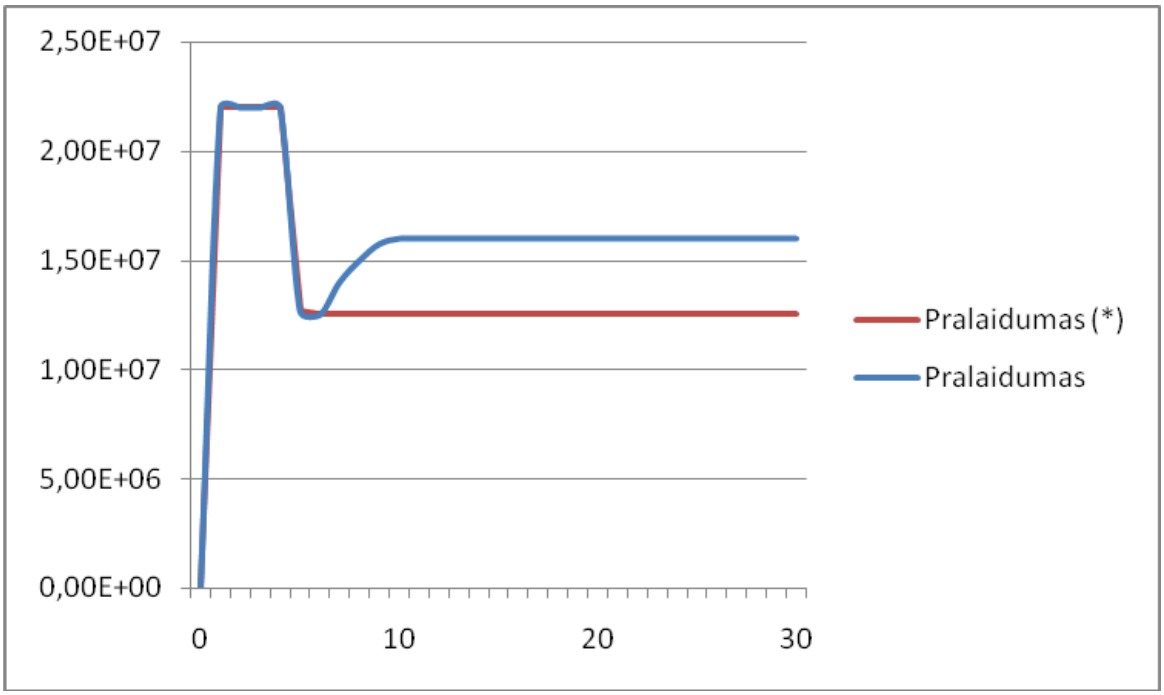


21. pav. Ne realaus laiko duomenų srauto pralaidumo grafikas (idealiuoju atveju)

Dvidešimt pirmajame paveiksle yra matomas ne realaus laiko srauto duomenų pralaidumo grafikas. Kaip ir realaus laiko duomenų srauto atveju bendras kanalo srautas yra lygus visais laiko momentais (*išskyrus pradinius laiko momentus, nes yra sukonfigūruota, jog srautas bus generuojamas/naudojamas ne nuo nulinio laiko momento.* Šiuo atveju buvo naudota FTP paslauga, su vidutinio intensyvumo/apkrovos parametrais. Iš pastarųjų dviejų paveikslų matyti, jog pralaidumas laidinėje ir bevielėje dalyje yra vienodas ir lygus *~22-23 Mbit/s*. Iš to galime daryti išvadas, jog sumodeliuotas tinklas veikia idealiuoju rėžimu: visi išsiųsti paketai pasiekia vartotoją, nėra jokių bitų klaidų bevielėje tinklo dalyje.

5.3 Tinklo topologijos būsenos tyrimas esant 30% paketų praradimo vertei

Paketų klaidų generatoriuje nustatomas 30% paketų praradimas. Generatoriaus veikimo pradžia (*uniform parametras*) nustatomas nuo penktosios scenarijaus simuliacijos minutės. Pirmiausia atliekamas tyrimas su realaus laiko duomenų srauto paslaugomis.

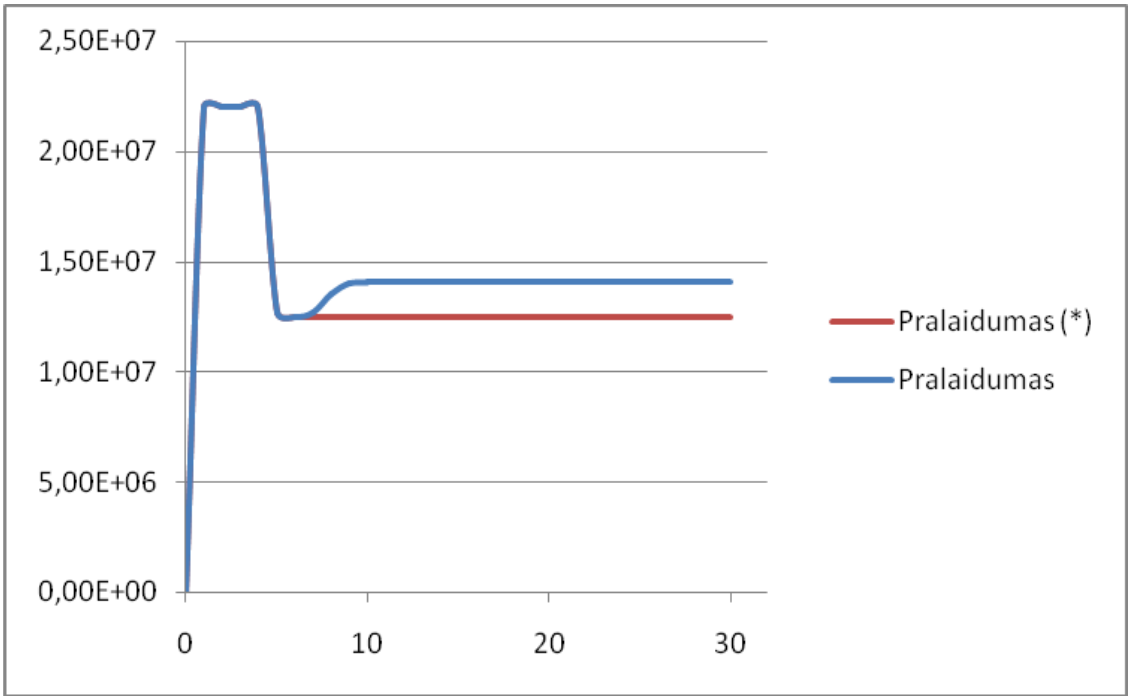


22. pav. Realaus laiko duomenų srauto pralaidumo grafikas (30% paketų praradimui esant)

Kaip matyti iš dvidešimt antrojo paveikslo tinklo pralaidumas bandymo simuliacijos pradžioje (*0 – 5 minutės*) yra analogiškas idealiųjų sąlygų bandymo simuliacijos rezultatams. Tai sąlygoją, jog pirmas penkias minutes tinkle nėra paketų bitų klaidų bevielėje tinklo dalyje. Penktąją minutę įsijungus paketų klaidų generatoriui pastebimas gana staigus pralaidumo mažėjimas. Taip yra dėl to, jog vos tik atsiradus paketų praradimui susidaro butelio kaklelio efektas[20] - bevielės tinklo dalies pralaidumas tampa daug mažesnis už laidinės dalies pralaidumą. Dėl šios priežasties mažėja bendrasis tinklo srautas: mažinamas siunčiamų segmentų langas, kuris sąlygoja mažesnį pralaidumą. Tačiau tą pačią minutę įsijungia realaus laiko TCP įgaliojimo protokolas, kuris naudoja vietinius persiuntimus, bendrinius patvirtinimus bei seka laikmačio išsekimo vertes. Iš grafiko matyti, jog nuo penktos iki dešimtos minutes šis protokolas leidžia padidinti pralaidumą dėl šių priežasčių: išsekus paketo laikmačio vertei, paketas yra išmetamas, o siuntėjui yra pranešama apie sėkmingai įvykdytą siuntimą, tuo būdų sumažinant tinko apkrovą; vietiniai persiuntimai (*su lokaliais bei bendriniais patvirtinimais*) taip pat šiek tiek sumažina bendrąjį tinklo apkrovimą, o kartu padidina ir spartą. Grafiko dalis nuo dešimtosios minutės yra

[20] Bottle neck effect angl. – Butelio kaklelio efektas

tiesė, kuri parodo, kad tinklo būvis yra nusistovėjęs ir liks toks pat kol nebus pakeistos tinklo darbo sąlygos: paketų praradimo vertė, pradiniai srautai ir t.t.



23. pav. Ne realaus laiko duomenų srauto pralaidumo grafikas (30% paketų praradimui esant)

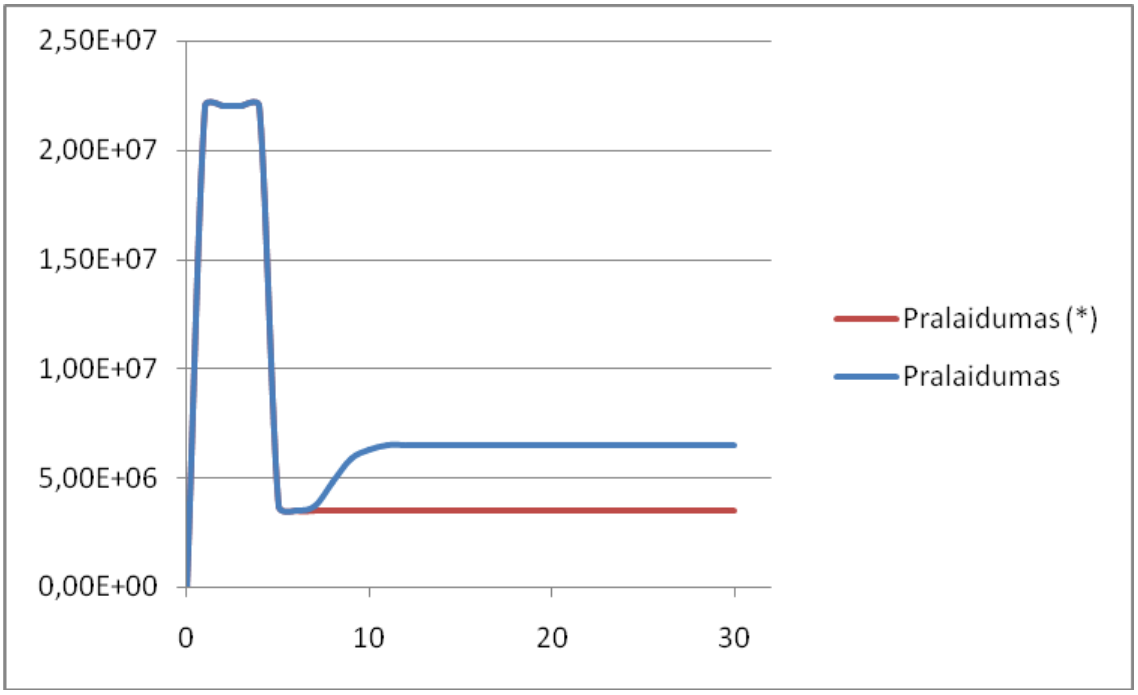
Dvidešimt trečiajame paveiksle yra parodytas ne realaus laiko duomenų srauto pralaidumo grafikas esant 30% paketų praradimui. Bandymo simuliacijos grafiko pradžios kreivė yra analogiška realaus laiko duomenų srauto pralaidumo grafiko pradžios kreivei. Situacija yra analogiška, nes paketų klaidų generatorius aktyvuojasi penktąją bandymo simuliacijos minutę. Penktąją minutę, pastebėjus anomalijas tinkle įsijungia realaus laiko TCP įgaliojimo protokolas, kuris naudodamas vietinius persiuntimus ir lokalius patvirtinimus padidina pralaidumą, tačiau ne taip smarkai kaip realaus laiko duomenų srauto bandymo atveju. Iš pastebėtų srautų skirtumų abiem atvejais galime padaryti prielaidą/išvadą, jog pralaidumas antruoju bandymu yra mažesnis, nes ne realaus laiko duomenų srauto paketai neturi kritinių paketų gyvavimo verčių. Šie paketai turi tik paprastąsias vertęs. Kadangi natūralu, jog kritinės paketų gyvavimo vertės yra daug mažesnės (*praėjus tam tikram laiko momentui realaus laiko duomenų paketai praranda aktualumą, todėl tokie paketai gali būti išmetami iš tinklo, dėl to mažėja tinklo apkrova, o pralaidumas didėja).* Ne realaus laiko duomenų srauto paketai nesėkmingo pristatymo atveju

yra persiunčiami keletą kartų dažniau (*intensyviau*), ir dėl šios priežasties matomas ne toks didelis tinklo pralaidumo atsistatymas paskutiniajame bandyme.

Dvidešimt antrajame ir trečiajame paveiksluose esančiuose grafikuose taip pat parodomas koks yra tinklo pralaidumas nenaudojant realaus laiko TCP įgaliojimo protokolo (*grafiko kreivė – Pralaidumas(*)*). Kaip matyti iš grafikų pralaidumas būtų pastovus (*konstanta*), nes dėl susidariusios butelio kaklelio padėties pralaidumas laidinėje dalyje prisitaikytų prie pralaidumo bevielėje tinklo dalyje. Tačiau nebūtų įgyvendinama joks pralaidumo atkūrimo/pagerinimo mechanizmas.

5.4 Tinklo topologijos būsenos tyrimas esant 70% paketų praradimo vertei

Paskutiniame tyrime bandysime ištirti tinklą naudojant realaus laiko duomenų srautą ir esant 70% paketų praradimui bevielėje tinklo dalyje. Tyrimo principas išlieka analogiškas prieš tai buvusiems tyrimams: paketų klaidų generatorius įjungiamas yra penktąją bandymo simuliacijos minutę, naudojamas realaus laiko duomenų srautas – multimedijos transliacija.. Bandymo rezultatų grafinė išraiška yra pateikiama dvidešimt ketvirtajame paveiksle.



24. pav. Realaus laiko duomenų srauto pralaidumo grafikas (70% paketų praradimui esant)

Paskutiniuoju bandymu norėta parodyti DoS atakos galimą scenarijų bevielėje tinklo dalyje . Atakuojamas bevielis tinklas (*tinklo paskutinis šuolis*), sukuriant 70% paketų praradimą. Esant tokia situacija suprastėja pralaidumas, o kartu nukenčia ir tinkle teikiamų paslaugų kokybė. Žinant, jog testuojamas yra realaus laiko duomenų srauto paslaugos, šis sąlyginai didelis paketų praradimo rodiklis turi didelę svarba minėtosioms paslaugoms. Kaip ir prieš tai buvusio bandymo atveju (*bandymas su 30% paketų praradimu*), pralaidumo kreivės pradžia yra analogiška. Skirtumas išryškėja nuo penktosios minutės, kai pradedamas paketų klaidų generavimas. Šiuo atveju kadangi procentinis paketų klaidų matas yra sąlyginai didelis natūralu, jog bendras kanalo pralaidumas taip pat mažėja (*22 Mbit/s sumažėja iki 3,5 Mbit/s iki ~ 15%*). Taip pat nuo penktosios minutės įsijungia realaus laiko TCP įgaliojimo protokolas, kuris naudoja laikmatį, bendrinius patvirtinimus bei lokalius persiuntimus. Dėl šių priežasčių yra pastebimas pralaidumo didėjimas nuo penktosios minutės iki dešimtosios minutes. Po dešimtosios minutės tinklo būvis nusistovi, pralaidumas tampa pastovus iki pat bandymo simuliacijos pabaigos.

Kaip ir prieš tai buvusiuose bandymuose parodomas normalus tinklo veikimas (*grafiko kreivė – Pralaidumas(*)*) nenaudojant realaus laiko TCP įgaliojimo. Šis pralaidumas tampa konstanta jau nuo penktosios minutės, nes nėra naudojamas joks mechanizmas sumažėjusiam tinklo pralaidumui atkurti.

5.5 Išvados

Eksperimentinis bandymas turėjo parodyti koki efektyvumą turi realaus laiko įgaliotasis protokolas esant žemos spartos paslaugų nutraukimo atakai tinkle. Iš viso buvo atliktos šešių scenarijų simuliacijos.

Pirmojo scenarijaus simuliacija – paketų klaidų generatoriaus tyrimas, buvo atliekamas norint įsitikinti ar sumodeliuotas paketų klaidų generatorius veikia korektiškai. Kaip matyti iš skyriuje pateiktų grafikų paketų klaidų generatoriaus tyrimas pateisino rezultatus.

Antrasis ir trečiasis bandymų scenarijai – tinklo veikimas idealioms sąlygoms esant. Tinkle simuliuojamas realaus laiko ir ne realaus laiko duomenų srautas. Nustatyta, jog idealioms sąlygoms esant abiejų scenarijų matavimo rezultatas yra vienodas: viešojo kanalo pralaidumas yra lygus *~22-23 Mbit/s.*

Ketvirtasis ir penktasis bandymų scenarijai – tinklo veikimas 30% paketų dingimui esant tinklo laidinėje dalyje. Atlikus šiuos bandymus matyti, jog rezultatai yra skirtingi realaus laiko srauto duomenims bei ne realaus laiko srauto duomenims. Ketvirtajame bandyme matyti, jog realaus laiko įgaliotasis protokolas atkuria kritusį pralaidumą viešojo ryšio kanale nuo *~12,5 Mbit/s* iki *~14 Mbit/s*. Penktajame bandyme matyti, jog atkurto pralaidumo šuolis yra šiek tiek didesnis nuo *~12,5 Mbit/s* iki *~16Mbit/s*. Gavus tokius rezultatus galime daryti išvadą, jog realaus laiko įgaliotasis protokolas, kaip ir buvo numanyta, efektyviau apdoroja realaus laiko duomenų srautą nei ne realaus laiko duomenų srautą.

Šeštajame bandymų scenarijuje buvo imituojama analogiška DoS ataka kaip ir ketvirtojo bei penktojo scenarijaus atveju, tik naudojant 70% paketų praradimą. Taip pat būtina pabrėžti, jog šeštajame bandyme buvo naudojamas tik realaus laiko duomenų srautas. Nustatyta, jog maksimaliai kritęs pralaidumas (*iki ~3,5 Mbit/s nuo ~22-23 Mbit/s)* turi tendenciją atsistatyti naudojant realaus laiko įgaliotąjį protokolą iki *~6,5 Mbit/s*). Nors atkurtojo pralaidumo šuolis penktajame bandyme (*~3,5 Mbit/s*) yra didesnis, tačiau šeštajame bandyme atliktos simuliacijos rezultatai (*~2,8-3 Mbit/s*) taip pat džiugina.

6. Išvados

1. Baigiamajame magistro darbe nagrinėjamos paslaugų užkirtimo atakos (*DoS*) bei šių atakų atrėmimo būdai. Nagrinėjimui pasirinktas Realaus laiko įgaliotojo protokolo duomenų siuntimo metodas.
2. Pirmojoje magistro darbo dalyje apžvelgta Realaus laiko įgaliotojo protokolo duomenų siuntimo logika nevienalyčiuose tinkluose. Nustatyta, jog TCP protokolo veikimas yra labiausiai optimizuotas vienalyčiams laidiniams tinklams. Esant nepakankamai bevielio tinklo ryšio kokybei nevienalyčiuose (*laidinis + bevielis*) tinkluose TCP protokolas veikia neefektyviai. Taip pat nustatyta, jog heterogeninis tinklas turi trūkumų, transportuojant realaus laiko duomenis. Dėl skirtingų Ethernet protokolo (*TCP protokolas naudoja šį protokolą kanaliniame lygmenyje*) klaidų aptikimo algoritmų nėra užtikrinama, jog paketas atėjęs iš laidinio tinklo bus patalpintas į bevielį tinklą. Taip yra todėl, jog neįmanoma rezervuoti bevielio tinklo resursų (*bevielio kanalo*) tam tikru konkrečiu laiko momentu, kada paketas atvyksta iš laidinės tinklo dalies.
3. Atliekant problemos analizę nustatyta, jog problemą galima spręsti šiais būdais: išskaidant TCP sujungimą į dvi dalis; retransliuojant duomenis kanaliniame lygmenyje; naudojant greitą pakartotinį duomenų persiuntimą; panaudojant šniukštinėjimo protokolą arba realaus laiko įgaliotąjį protokolą. Pastarasis protokolas tolesniam nagrinėjimui buvo pasirinktas dėl šių priežasčių: palaiko tęstinį ryšį (*labai retais atvejais bei trumpais momentais nepalaiko tęstinio ryšio*), dalinai užtikrina patikimą ryšį, naudoja vietinį duomenų persiuntimą bei turi galimybę palaikyti realaus laiko duomenų srautus.
4. Antrojoje darbo dalyje apžvelgtas Realaus laiko įgaliotasis protokolas bei susisteminta šio protokolo veikimo logika. Įvardinta jog protokolo architektūra paremta dvejomis procedūromis: duomenų – naudojama paketams ateinantiems iš siuntėjo į bevielį prieigos tašką; patvirtinimų – naudojama paketų patvirtinimams gautiems iš mobilaus vartotojo. Šio protokolo įgyvendinimas grindžiamas TCP protokolu, prie šio protokolo logikos pridedant minėtas procedūras. Protokolų

perjungimui naudojamas įprogramuotas loginis jungiklis, kuris esant anomalijoms tinkle parenka pagal kokį protokolo modelį bus apdorojami gaunami paketai.

5. Trečiojoje darbo dalyje sudarėme planuojamo tirti tinklo topologiją. Parinkome tinklo mazgus/įrenginius, nustatėme jų konfigūracinius poreikius (*bevielio tinklo standartas, sparta, funkcionalumas ir t.t.*). Sudėliojome tinklą sudarančius potinklius bei juos apjungėme į vieną vidinį tinklą. Išskyrėme laidinę ir bevielę tiriamojo tinklo dalis.
6. Paskutinėje magistrinio darbo dalyje atliekamas sudarytos tinklo topologijos tyrimas. Šiam tyrimui atlikti naudojamas pasirinktasis realaus laiko įgaliotasis protokolas. Tyrimas modeliuojamas ir atliekamas OPNET programiniame pakete. Tyrimas atliktas naudojant dvejus analogiškus scenarijus bei dar vieną papildoma scenarijų. Pirmasis tyrimo scenarijus: tinklas veikia idealiuoju rėžimu – nėra jokių paketų praradimų bevielėje tinklo dalyje. Šiuo tyrimo scenarijumi nustatytas bendras tinklo pralaidumas. Antrasis scenarijus: tinkle imituojamas 30% paketų praradimas. Šiuo atveju matyti jog naudojant Realaus laiko įgaliotąjį protokolą sumažėjęs tinklo pralaidumas turi tendenciją atsitiesti (*didėti iki tam tikros ribos*). Trečiasis scenarijus: tinkle imituojamas 70% paketų praradimas. Šis bandymas patvirtino prieš tai buvusio bandymo rezultatus (*antrasis scenarijus*), todėl galime daryti išvadą, jog baigiamojo magistrinio darbo tyrimo rezultatai sutampa su analitinėje bei projektinėje dalyse aptartais teoriniais principais bei metodais.

7. Naudota literatūra

1. Protocol Specification. RFC: 793 [Interaktyvus]. Transmission Control Protocol. Iš *IEFT* [Interaktyvus]. 1981, Spalis [žiūrėta 2011-11-03]. Prieiga per internetą: http://www.ietf.org/rfc/rfc793.txt.
2. Standard. ISO/IEC DIS 8802-11. [Interaktyvus]. Wireless LAN Medium Access Control and Physical Layer Specifications. Iš *ISO/IEEE* [Interaktyvus]. 1999 [žiūrėta 2011-11-03]. Prieiga per internetą: http://www. standards.ieee.org/findstds/errata/802.11a-errata.pdf
3. Ajay Bakre, B.R. Badrinath. I-TCP: Indirect TCP for Mobile Hosts [Interaktyvus]. 1994, Spalis [žiūrėta 2011-11-15]. Prieiga per internetą: http://ieeexplore.ieee.org/xpl/articleDetails.jsp?reload=true&arnumber=500012&contentType=Conference+Publications
4. Hari Balakrishnan, Srinivasan Seshan, Elan Amir and Randy H. Katz. Improving TCP/IP Performance over Wireless Networks [Interaktyvus]. 1995 [žiūrėta 2011-11-15]. Prieiga per internet: http://dl.acm.org/citation.cfm?id=215544
5. Nitin Vaidya. TCP for Wireless and Mobile Hosts (MobiCOM'99 Tutorial) [Interaktyvus]. 1999 [žiūrėta 2011-11-15]. Prieiga per internetą: http://www.scribd.com/doc/18001933/54/Link-Level-Retransmissions
6. Sarma Vangala and Miguel A. Labrador. The TCP SACK-Aware Snoop Protocol for TCP over Wireless Networks [Interaktyvus]. 2003 [žiūrėta 2011-11-15]. Prieiga per internetą: http://ieeexplore.ieee.org/xpl/login.jsp?tp=&arnumber=1286032&url=http%3A%2F%2Fieeexplore.ieee.org%2Fxpls%2Fabs_all.jsp%3Farnumber%3D1286032
7. Lei Huang, Uwe Horn. Proxybased TCPfriendly streaming over mobile networks [Interaktyvus]. 200 [žiūrėta 2011-11-22]. Prieiga per internetą: http://dl.acm.org/citation.cfm?id=570794
8. Sam Liang, David Cheriton. TCP-RTM: Using TCP for real time multimedia applications [Interaktyvus]. 2002 [žiūrėta 2011-11-22]. Prieiga per internetą: http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.86.3257

9. Paolo Bellavista, Antonio Corradi, Carlo Giannelli. Mobile Proxies for Proactive Buffering in Wireless Internet Multimedia Streaming [Interaktyvus]. 2005 [žiūrėta 2011-11-22]. Prieiga per internetą: http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.75.9419
10. The Network Processing Forum (NPF). TCP proxy (RT-proxy) protocol [Interaktyvus]. 2004 [žiūrėta 2011-11-22]. Prieiga per internetą: http://www.oiforum.com/public/documents/TCPProxyBenchmarkV1.pdf
11. Andrei Gurtov, Sally Floyd. Lifetime packet discard for efficient real-time transport over cellular links [Interaktyvus]. 2002 [žiūrėta 2011-11-22]. Prieiga per internetą: http://dl.acm.org/citation.cfm?id=965738
12. Protocol Specification. RFC: 3366 [Interaktyvus]. Advice to link designers on link Automatic Repeat reQuest (ARQ) Iš *IEFT* [Interaktyvus]. 2002 Rugpjutis [žiūrėta 2011-11-22]. Prieiga per internetą: https://ebook.tools.ietf.org/html/rfc3366

8. Summary

Modeling of the lost packets retransmission method within the wireless local network

The aim of this thesis is to get clear approach regarding heterogeneous network and investigate effectiveness of the Real time TCP proxy method.

In the first part of this thesis there is a short review of available TCP improvements. After analyzing all the available TCP improvements Real time TCP proxy method has been selected for the further investigation. The main reason of this choice was that real time proxy method is suitable for real time data stream and second reason was that it could supply end to end connection.

After analyzing real time proxy model (*from theoretical scope*) there were few experiments performed to see whether results could prove theory. The results of all the experiments were positive so, to sum everything up I could say that this investigation was successful.

9. Santrumpų ir terminų žodynas

ARP (*Address Resolution Protocol angl.*) – Adreso išrišimo protokolas

BER (*Bit Error Rate angl.*) – Bitų klaidų skaičius

CSMA/CA – (*Carrier Sense Multiple Access/Collision Avoidance angl.*) – Daugkartinis prieigos nešlio metodo valdymas/kolizijų vengimas

CSMA/CD – (*Carrier Sense Multiple Access/Collision Detection angl.*) – Daugkartinis prieigos nešlio metodo valdymas/kolizijų aptikimas

DoS (*Denial of Service angl.*) – Paslaugų neigimas/užkirtimas

EDF (*Earliest Deadline First angl.*) – Trumpiausia galinė riba aptarnaujama pirmiausiai

IP (*Internet Protocol angl.*) – Interneto protokolas

OSI (*Open System Interconnection angl.*) – Atvirų sistemų aprašo modelis

PEP (*Performance Enhancing Proxy angl.*) – Veikimą pagerinantis įgaliojimas

Proxy angl. - Įgaliojimas

RTO (*Restore Time Objective angl.*) – Laiko tarpas reikalaujamas paslaugos atstatymui

RTT (*Round Trip Time angl.*) – Paketo kelionės nuo siuntėjo iki gavėjo laikas

TCP (*Transmission Control Protocol angl.*) – Siuntimo kontrolės protokolas

TTL (Time to Live angl.) – Paketo gyvavimo laikas

UML (*Unified modeling language angl.*) – Universali modeliavimo kalba

10. OPNET tinklo mazgų konfigūracinė lentelė

Darbe modifikuoti tik tie įrenginiai, kuriuose yra pasiekiamos 3-4 OSI lygmens paslaugos. Reikia pabrėžti, jog maršrutizatoriaus maršrutų parinkimo protokolas yra RIP. Įrenginio kiti maršrutizavimo parametrai (*išskyrus maršrutizuojamus tinklo adresų rėžius*) nėra pakeisti. Bevielės dalies komutatorius bei laidinės dalies komutatoriai veikia standartiniu[21] rėžimu.

3 lentelė. OPNET tinklo mazgų konfigūracinė lentelė

Mazgo pavadinimas	Default parametro reikšmė	Įdėto parametro reikšmė
wlan_wkstn_adv (node_0 – centrinė bevielė darbinė stotis)	Klaidų generatorius: neegzistuoja TCP įgaliojimas: neegzistuoja	Klaidų generatorius: Uniform(300,300) TCP įgaliojimas: Uniform(0,0)
Application_Config	FTP: UNiform(100,110) Multimedia: Uniform(100,110)	FTP: Uniform(60,60) Multimedia: Uniform(60,60)
Server_Config	OS systems definition: NT	OS systems definition: NT (nepakeista)
Profile_Config	Sample profiles (5 rows: Engineering, Reasercher, E-commerce, Sales person, Multimedia)	Real time traffic profile (Multimedia traffic:Medium load) Non-real time traffic profile(FTP traffic:Medium load)

Reikia pažymėti, jog Uniform parametras turi dvi skaitinės reikšmes: pirmoji reiškia įvykio pradžios minimalų laiką, antroji: įvykio pradžios maksimalią laiko pabaigą. Pavyzdžiui, jei multimedijos srauto parametras turi reikšmę (*100*,*110*), vadinasi OPNET programinis paketas sumodeliuotame tinkle multimedijos srautą pradės simuliuoti ne anksčiau kaip tik nuo 100-osios sekundės, bet ne vėliau kaip 110-ąją sekundę.

[21] Default angl. – standartinis, pakeitimų nereikalaujantis

11.Priedai

1. Kompaktinė plokštelė

Kompaktinės plokštelės turinys yra pateiktas žemiau:

a) Elektroninė šio darbo versija PDF bei DOC formatu.

b) OPNET modelis *wlan_wkstn_adv,* kuriame yra įgyvendintas realaus laiko įgaliotasis TCP protokolas.