KAUNO TECHNOLOGIJOS UNIVERSITETAS

INFORMATIKOS FAKULTETAS

Gediminas Jančys

TRANSPORTO PROTOKOLŲ ĮTAKOS INTERAKTYVIAM DARBUI PER VPN TYRIMAS

Baigiamasis magistro projektas

Vadovas
lekt. dr.. K. Paulikas

KAUNAS, 2016

KAUNO TECHNOLOGIJOS UNIVERSITETAS

INFORMATIKOS FAKULTETAS

TRANSPORTO PROTOKOLŲ ĮTAKOS INTERAKTYVIAM DARBUI PER VPN TYRIMAS

Baigiamasis magistro projektas

Programų sistemų inžinerija (M4046M21)

Vadovas

lekt. dr. Kęstutis Paulikas

Recenzentas

lekt. dr. Dangis Rimkus

Projektą atliko

Gediminas Jančys

KAUNAS, 2016

Gediminas Jančys. *TRANSPORTO PROTOKOLŲ ĮTAKA INTERAKTYVIAM DARBUI PER VPN TYRIMAS*. Magistro baigiamasis projektas vadovas doc. lek. Kęstutis Paulikas; Kauno technologijos universitetas, informatikos fakultetas.

Mokslo kryptis ir sritis:

Reikšminiai žodžiai: *VPN, NAT, User space, VPN tunnel, user land VPN*

Kaunas, 2016. 80 p.

SANTRAUKA

Šiomis dienomis augant interneto naudotojų skaičiui ir didėjant paklausai dirbti iš namų ar keliaujant didėja poreikis pasiekti vidinius įmonės resursus per nuotolį. Šiam tikslui pasiekti naudojami įvairūs VPN sprendimai. Tačiau ne visi pilnai atitinka specifinius reikalavimus.

Šiame darbe analizuojamos VPN sprendimo galimybės išskiriant svarbiausius reikalavimus, veikimo principus, bei realizacijos technologijas. Atsižvelgiant į surinktus reikalavimus buvo suprojektuota ir realizuota programinė įranga. Tyrimo metu buvo tiriama realizuota programinė įranga, bandant nustatyti kuris transportavimo protokolas veikia sparčiausiai ir suteikia didžiausią sujungimą vienokiu ir kitokiu tinklo modeliu.

Jančys Gediminas. *Master's* thesis in *STUDY OF TRANSPORT PROTOCOLS INFLUENCE ON THE INTERACTIVE WORK VIA THE VPN CONNECTION* supervisor assoc. prof. lec. Kęstutis Paulikas. The Faculty of Informatics Kaunas University of Technology.

Research area and field:

Key words: VPN, User space, VPN tunnel, user land VPN

Kaunas, 2016. 80 p.

SUMMARY

Nowadays due to the increasing number of the internet users and demand to work from home or while traveling there is a fast growing necessity to remotely access company's internal resources. For this task VPN solutions are used. However, not all of the VPN solutions fully meet specific company's requirements needs.

In this work the VPN solutions are analyzed by distinguishing main requirements, operating principles as well as technologies needed to build those solutions. Based on the solutions analyzed the data software was built to test them. Software testing was carried using several network configurations to figure out which transport protocol is the most appropriate, has the highest bandwidth, the lowest latency and the highest accessibility over the internet.

KAUNO TECHNOLOGIJOS UNIVERSITETAS

Informatikos fakultetas

(Fakultetas)

Gediminas Jančys

(Studento vardas, pavardė)

Programų sistemų inžinerija (M4046M21)

(Studijų programos pavadinimas, kodas)

„Baigiamojo projekto pavadinimas“

AKADEMINIO SĄŽININGUMO DEKLARACIJA

20 16 m. gegužės d.

Kaunas

Patvirtinu, kad mano, Gedimino Jančio, baigiamasis projektas tema „Transporto protokolų įtaka interaktyviam darbui per VPN tyrimas“ yra parašytas visiškai savarankiškai ir visi pateikti duomenys ar tyrimų rezultatai yra teisingi ir gauti sąžiningai. Šiame darbe nei viena dalis nėra plagijuota nuo jokių spausdintinių ar internetinių šaltinių, visos kitų šaltinių tiesioginės ir netiesioginės citatos nurodytos literatūros nuorodose. Įstatymų nenumatytų piniginių sumų už šį darbą niekam nesu mokėjęs.

Aš suprantu, kad išaiškėjus nesąžiningumo faktui, man bus taikomos nuobaudos, remiantis Kauno technologijos universitete galiojančia tvarka.

(vardą ir pavardę įrašyti ranka) *(parašas)*

TURINYS

LENTELIŲ SĄRAŠAS

PAVEIKSLĖLIŲ SĄRAŠAS

1 ĮŽANGA

1.1 Dokumento paskirtis

Šio dokumento paskirtis yra pateikti informaciją susijusią su kurtos ir tyrinėtos sistemos – „gNet virtualaus privataus tinklo sprendimas naudotojo teisėmis“ prototipo realizacija, atliktais tyrimais ir jų rezultatais. Dokumentas sudarytas iš skyrių, kuriuose pateikiama atlikta sistemos analizė, jos projektavimas ir tyrumas. Analizės skyriuje apžvelgiami galimi sistemos realizacijos būdai, priemonės ir svarbiausi principai. Projektinėje dalyje pateikiama informacija apie sistemos realizaciją ir naudotus įrankius. Tyrime aprašoma vykdyti tyrimai, sistemos plėtojimo perspektyvos ir kitos įžvalgos.

1.2 Darbo tikslai

Magistro studijų metu buvo sukurta prototipinė programinės įrangos sistema, kuri yra nemokoma, laisvai platinama, paskirstyta, lengvai naudojama mažo kompiuterinio raštingumo naudotojų, lengvai diegiama informacinių technologijų administratorių, programinės įrangos parametrai lengvai keičiami naudojant centralizuotą nustatymo sprendimą, nereikalauja papildomų virtualių tinklo įrenginių diegimo į kompiuterį, užtikrinta didžiausią saugumą ir ryšio užmezgimo galimybę. Pagrindinis kuriamos programinės įrangos sistemos akcentas lengvas ir greitai įvykdomas sujungimo užmezgimas tarp naudotojų.

Šio darbo tikslas yra nustatyti kaip įtakoja duomenų persiuntimą sukurtos sistemos naudojami skirtingi duomenų transportavimo protokolai. Bus tiriamas vėlinimas, duomenų pralaidumas ir naudingos informacijos perduodamas kiekis.

2 ANALITINĖ DALIS

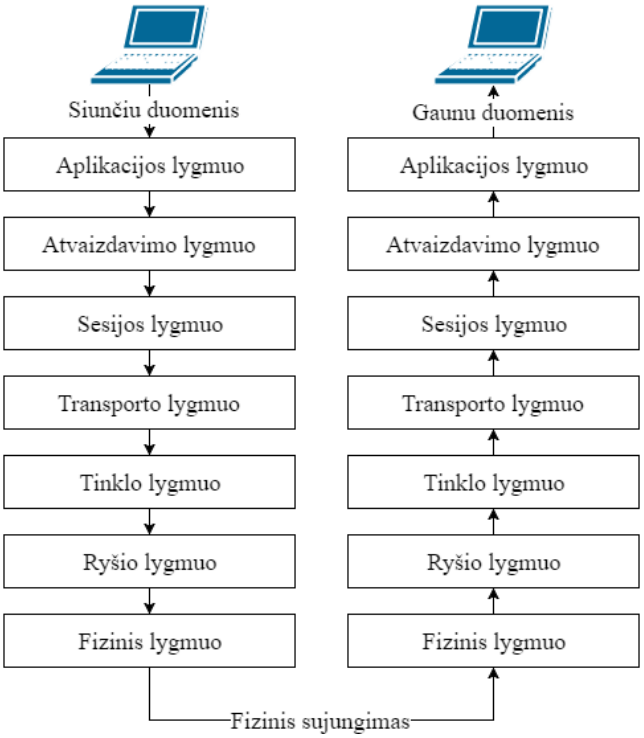
Šiame skyriuje apžvelgiama su virtualiais privačiais tinklais ir komunikacijos procesu tarp įrenginių, naudojant internetą. Taip pat analizuojama kuriame komunikacijos lygmenyje turėtų veikti kuriama sistema, koks sujungimo modelis turi būti realizuotas bei kaip šifruojami duomenys. Taip pat apžvelgiami transporto protokolai, kuriais bus perduodami duomenys, bei apžvelgiamos pagrindinės problemos, dėl kurių kyla kliūtys įrenginių komunikacijai internete.

2.1 Virtualus privatus tinklas

Virtualus privatus tinklas, toliau VPN, yra tinklo technologija, kurios pagalba yra sukuriama saugi jungtis per viešąjį tinklą. Šia technologija dažnai naudojasi įmonės ir organizacijos norėdamos nutolusiems naudotojams suteigti prieigą prie vidinių resursų. Ši technologija leidžia sujungti nutolusias vietas, padalinius, kurie yra už tūkstančių kilometrų.

2.2 Komunikacija tarp įrenginių naudojant internetą

Prieš pradedant nagrinėti VPN sujungimų tipus reikia paminėti tai, kad kiekvienas duomenų perdavimas internetu tarp kompiuterių turi būti apdorojamas loginių ir fizinių struktūrų. Šias struktūras galima nagrinėti per TCP/IP ir OSI modelius. Šiame darbe nagrinėsime naudodami OSI modelį.



1 pav. Kompiuterių komunikacijos analizė naudojant OSI modelį

OSI modelis, 1 paveikslas, suskirstytas į septynis lygmenis: fizinį, ryšio, tinklo, transporto, sesijos, atvaizdavimo ir aplikacijos. Fizinis lygmuo atsakingas už fizinį sujungimą, pavyzdžiui variniais laidais, radijo bandomis ar šviesolaidžiais.

Antrasis lygmuo yra ryšio lygmuo. Jo pagrindinė užduotis patikrinti ar fiziniame lygmenyje perduodama informacija nebuvo pakeista. Pakeitimai gali būti dėl laidų nutrūkimų, radijo bangų trukdžių ar šviesos elementų gedimų.

Trečiasis tinklo lygmuo atsakingas už duomenų maršruto sudarymą. Šio lygmens pagalba duomenys pasiekia norimą gavėją.

Transporto lygmuo atsakingas už duomenų suskaidymą taip, kad būtų galima siųsti tinklu ir atsakingas už duomenų apjungimą perduodant aukštesniam lygmeniui.

Penktasis sesijos lygmuo sinchronizuoja darbą tarp siuntėjo ir gavėjo aplikacijų. Atvaizdavimo lygmuo atsakingas už duomenų parengimą aukštesniam lygmeniui. Jo pareigas sudaro duomenų glaudinimas ir išskleidimas, bei duomenų vertimas, pavyzdžiui: skirtingos architektūros kompiuteriai skirtingai perduoda skaičius tinklu. Šis lygmuo išverčia gautą skaičių taip, kokį siuntė siuntėjas, nepriklausomai nuo architektūros.

Septintasis, lygmuo yra aplikacijos. Šiame lygmenyje veikia vartotoj programa priklausomai nuo gautų duomenų.

Apžvelgus komunikacijos koncepciją ir susipažinus su jos lygmenimis tolimesniuose skyriuose apžvelgsime kokio tipo sujungimai galimi kiekviename lygmenyje.

2.3 Sujungimų tipai

Šiame skyriuje apžvelgiama kokie sujungimų būdai taikomi kiekviename iš OSI modelio lygmenų.

2.3.1 Fizinis lygmuo

Fizinis lygmuo- žemiausias OSI modelio lygmuo. Šis lygmuo atsakingas už duomenų perdavimą fizine terpe. Populiariausios terpės yra trys- elektros impulsai variniais laidais, radijo bangos ir šviesos impulsai šviesolaidžiais. Pasinaudojant šiomis trimis terpėmis galima sujungti du taškus (tinklus ar klientus). Šis būdas yra pats greičiausias ir sunkiausiai keičiamas. Šiame lygmenyje naudojami šios sujungimo technologijos:

- Tiesioginis klientų ar tinklų sujungimas laidais, radijo bangomis ar šviesolaidžiu
- Ryšio teikėjo linijos nuoma
- Ryšio tiekėjo laiko nuoma

Ne kiekvienas asmuo ar įmonė gali sau lesti šio tipo sujungimus nutolusiems darbuotojams. Dėl aukštos kainos ir mobilumo trūkumo VPN technologijos buvo vystomos aukštesniuose OSI modelio lygmenyse. Šio lygmens sujungimai teritoriniai ir vietiniai tinklai.

2.3.2 Ryšio lygmuo

Ryšio lygmuo- antrasis OSI modelio lygmuo. Jis atsakingas už nepakitusių duomenų perdavimą per fizinį lygmenį. Šiame lygmenyje, siekiant įgyvendinti saugų duomenų perdavimo sujungimą naudojamos šios technologijos:

- Frame Relay
- OpenVPN (tinklų tiltas)
- MTLS VPN
- APN
- PPTP
- L2TP

Šios technologijos yra pigesnės nei pirmojo lygmens ir dažniau sutinkamos pramoniniuose tinkluose. Šio tipo technologijos leidžia įgyvendinti teritorinio tinklo ir vietinio tinklo sujungimą.

2.3.3 Tinklo lygmuo

Tinklo lygmuo- trečiasis OSI modelio lygmuo. Šis lygmuo atsakingas už duomenų perdavimą tarp tinklų ir kelio parinkimą iki gavėjo. Šiame lygmenyje naudojami šie saugaus sujungimo sprendimai:

- IPSec
- OpenVPN (maršrutizuojami tinklai)
- GRE
- Hamachi
- TeamViewer
- Įvairūs SSL VPN sprendimai

Visi šie sprendimai leidžia sujungti kompiuterius taip lyg jie būtų gretimuose tinkluose, taip pat leidžiant jiems komutuoti nuo trečio ir aukštesnio lygmens. Šiame lygmenyje yra realizuota daug saugaus sujungimo sprendimų, kurie artimi mažo kompiuterinio raštingumo naudotojams.

2.3.4 Transporto lygmuo

Transporto lygmuo – ketvirtasis OSI modelio lygmuo. Šis lygmuo atsakingas už duomenų garantijas perduodant duomenis gavėjui. Jis atlieka duomenų patikrą ir užtikrinta, kad gavėjas tikrai duomenis gaus. Šiame lygmenyje naudojamos šios saugaus duomenų perdavimo technologijos:

- Stunnel
- SSL/ TLS (HTTPS, IMAP4s)

Šiame lygmenyje sujungiami ne kompiuteris su kompiuteriu ar tinklas su tinklu, o tik programa su programa. Šis būdas plačiai naudojamas internato tinklapių saugiam duomenų perdavimui.

2.4 Saugumas

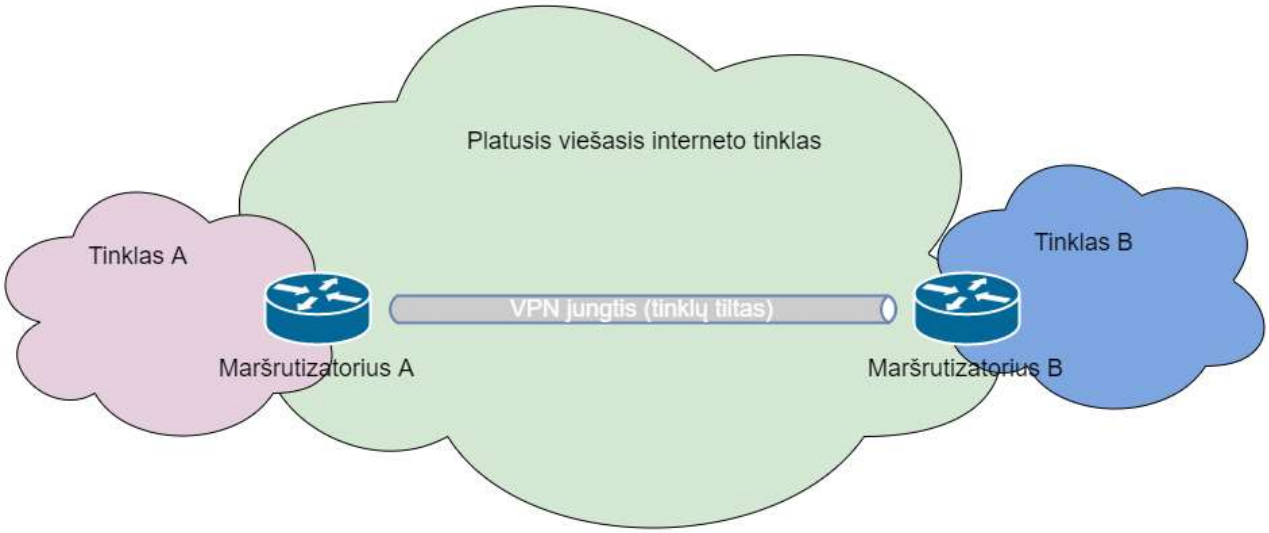
Šiame skyrių apžvelgiamos technologijos susijusio sus saugumu ir pateikiamos rekomendacijos geriausių principų realizacijai.

2.4.1 Tinklo saugumas

Siekiant suteikti didžiausią saugumą tiek sujungtiems klientams, tiek kitiems tinklų naudotojams peržvelgiama kuris sujungimo tipas yra saugiausias iš kenkėjiškų programų plitimo perspektyvos. Tinklų ir operacinių sistemų ar kitos programinės įrangos saugumo spragomis dažnai naudojasi kenkėjiškos programos.

2.4.1.1 Tinklų tilto sujungimas

Atlikus šį sujungimą tinklai tampa vienu vietiniu tinklu. Šis sujungimas pavaizduotas 2 paveiksle.



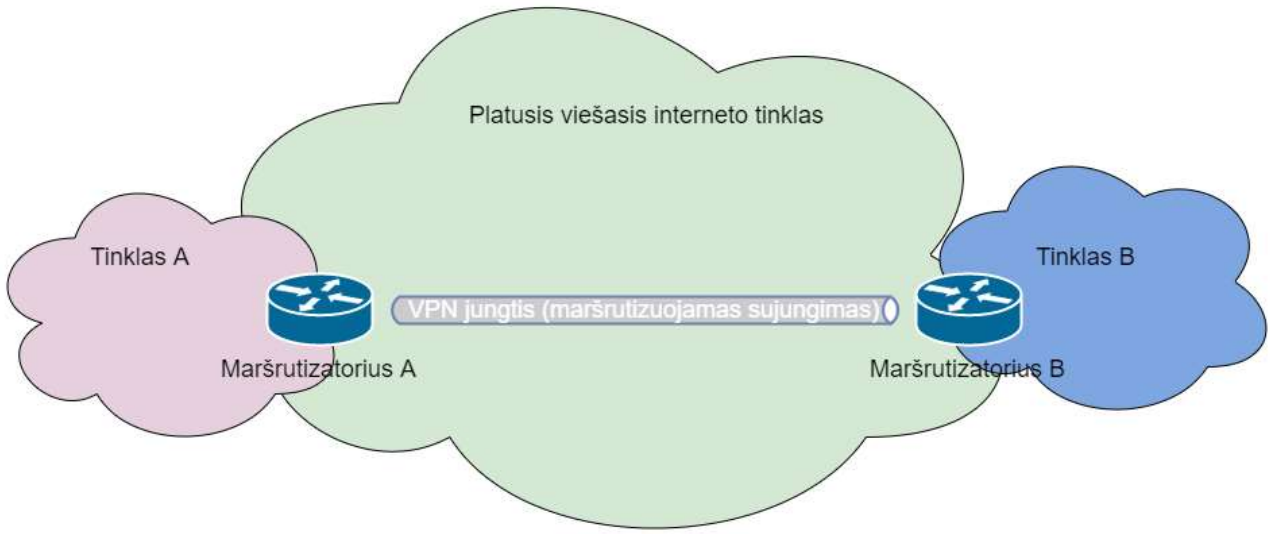
2 pav. Tinklų tilto sujungimo pavyzdys

Atlikus tinklų sujungimą pagal tinklų tilto principą 2 paveikslo atveju, tinklas A ir tinklas B tampa vienu vietiniu tinklu AB. Visi duomenys tarp tinklų persiunčiami laisvai, be jokių apribojimų, nes šis sujungimas veikia antrame OSI modelio lygmenyje (1 paveikslą), todėl apsikrėtus kenkėjiška programine įranga vienam iš kompiuterių esančių tinkle AB ir visas tinklas AB gali būti apkrėstas. Kenkėjiškos programos plitimo nesustabdys jokia ugniasienė. Tinklo AB kompiuterių saugumas priklausys nuo individualių apsaugos priemonių. Čia galima laikyti saugumo lygį labai mažu.

Taip pat šis sujungimas yra neefektyvus duomenų perdavimo prasme, nes transliavimo visiems duomenys bus transliuojama per tinklų tilto sujungimą ir bus mažinamas jos pralaidumas.

2.4.1.2 Maršrutizuojamas tinklų sujungimas

Maršrutizuojamas sujungimas atliekamas tinklo lygmenyje pagal OSI modelį (1 paveikslas). Šio sujungimo pavyzdys pateikiamas 3 paveiksle.



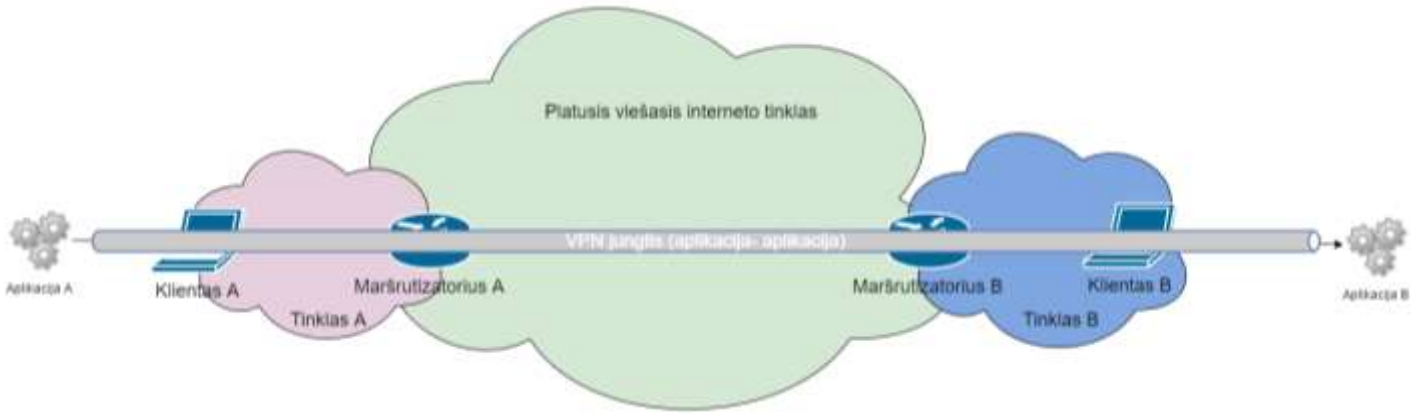
3 pav. Maršrutizuojamo sujungimo pavyzdys

Maršrutizuojamasis sujungimas nuo tinklų tilto sujungimo pagal sujungimo pavyzdį (2 paveikslas) niekuo nesiskiria. Skiriasi tuo, kad sujungus tinklą A su tinklu B, pagal 3 paveikslą, sujungti tinklai lieka dviem atskirais tinklais- tinklu A ir tinklu B. Maršrutizatoriai A ir B yra atsakingi už saugų duomenų perdavimą. Visas srautas tarp tinklų A ir B keliauja per maršrutizatorius, todėl gali būti filtruojamas. Užsikėtus kenkėjiška programine įranga vienam iš kompiuterių, sakykime tinkle A, jos išplitimas yra sudėtingas ir dažnai neįmanomas į tinklo B esančius kompiuterius. Dėl šios savybės galime teigti, kad šio tipo sujungimo saugumo lygis yra normalus. Taip pat būtina paminėti, kad šiuo sujungimo būdu gali užmegzti saugų sujungimą tiek tinklas su tinklu, tiek kompiuteris su tinklu, tiek kompiuteris su kompiuteriu.

Šis sujungimas yra efektyvesnis lyginant su tinklų tilto sujungimu, duomenų kiekio perdavimo atžvilgiu, nes yra perduodami tik reikalingi duomenys tarp tinklų.

2.4.1.3 Aplikacijos su aplikacija sujungimas

Aplikacijos su aplikacija sujungimo pavyzdys pateikiamas 4 paveiksle. Šį sujungimo modelį galime tratuoti kaip saugiausią iš analizuojamų, nes yra sujungiama tik aplikacija su aplikacija. Šiame sujungime gali būti apkrėstas kompiuteris kenkėjiška programine įranga ar visas tinklas. Jis neplis per saugų sujungimą toliau. Šis sujungimas dirba ketvirtame, transporto, OSI modelio lygmenyje.



4 pav. Aplikacija- aplikacija sujungimo pavyzdys

Pagal pateiktą pavyzdį, pavaizduotą 4 paveiksle., matome, kad aplikacija A, veikianti kliento A kompiuteryje norėdama užmegzti saugų sujungimą su aplikacija B, veikiančia kliento B kompiuteryje užmezga saugų ryšį tarp kliento A kompiuterio esančio tinkle A per maršrutizatorių A per viešąjį internato tinklą, per maršrutizatorių B iki kliento kompiuterio B kol galiausiai pasiekia norimą aplikaciją B. Kad šis sujungimas įvyktų reikalinga specifinė tinklo įrangos (maršrutizatorių A ir B) konfigūracija, kuri sudarytų sujungimą iki aplikacijos veikiančios vietiniame tinkle. Atsižvelgiant į šio sujungimo savybes galiu teigti, kad šio sujungimo saugumo lygis- aukštas, nes tarpiniai įrenginiai neturi prieigos prie perduodamų duomenų.

Taip pat būtina paminėti, kad duomenų kiekis perduodamas per šio tipo sujungimą yra pats optimaliausias, nes perduodami tik aplikacijų duomenys.

2.4.2 Duomenų saugumas

Duomenų saugumas priklauso nuo duomenų šifravimo algoritmo ir nuo duomenų šifravimo rakto apsikeitimo būdo. Toliau apžvelgsiu kokie saugumo metodai yra naudojami.

2.4.2.1 Šifravimo raktas

Siekiant gauti didžiausią šifravimo greitį visi šifravimo algoritmai naudoja simetrinį raktą. Bet skirtingi saugaus duomenų perdavimo metodai jo susitarimui naudoja skirtingus būdus. Visus metodus galime skirstyti į dvi pagrindines grupes- simetriško šifravimo ir dinaminio šifravimo.

2.4.2.1.1 Nešifruotas duomenų perdavimas

Nešifruotas duomenų perdavimas yra naudojamas šiose technologijose:

- Fizinių laidų tiesime, nuomoje
- Frame relay
- MPLS

Šie sprendimai neužtikrina duomenų saugumo. Jie suformuoja tik kanalą tarp siuntėjo ir gavėjo. Tačiau yra tikimybė, kad kanalo duomenis matys trečios šalys.

2.4.2.1.2 Statinio šifravimo grupė

Statinėje grupėje sesijos užmezgimo raktai yra įrašomi į nustatymus ir yra naudojami visą laiką komunikacijai tarp siuntėjo ir gavėjo. Įsilaužimo atveju visuose įrengimuose, kurie dalyvavo komunikacijoje reikia keisti pradinį raktą. Perėmęs raktą įsibrovėlis gali perskaityti visą duomenų srautą kuris buvo siunčiamas anksčiau, bei kuris bus siunčiamas iki rakto pakeitimo. Dėl šių savybių šis metodas moderniuose sistemose nėra naudojamas. Į šią grupę priskirčiau šiuos sprendimus:

- GRE
- IPSec (statinis raktas)
- OpenVPN (statinis raktas)

2.4.2.1.3 Dinaminė grupė

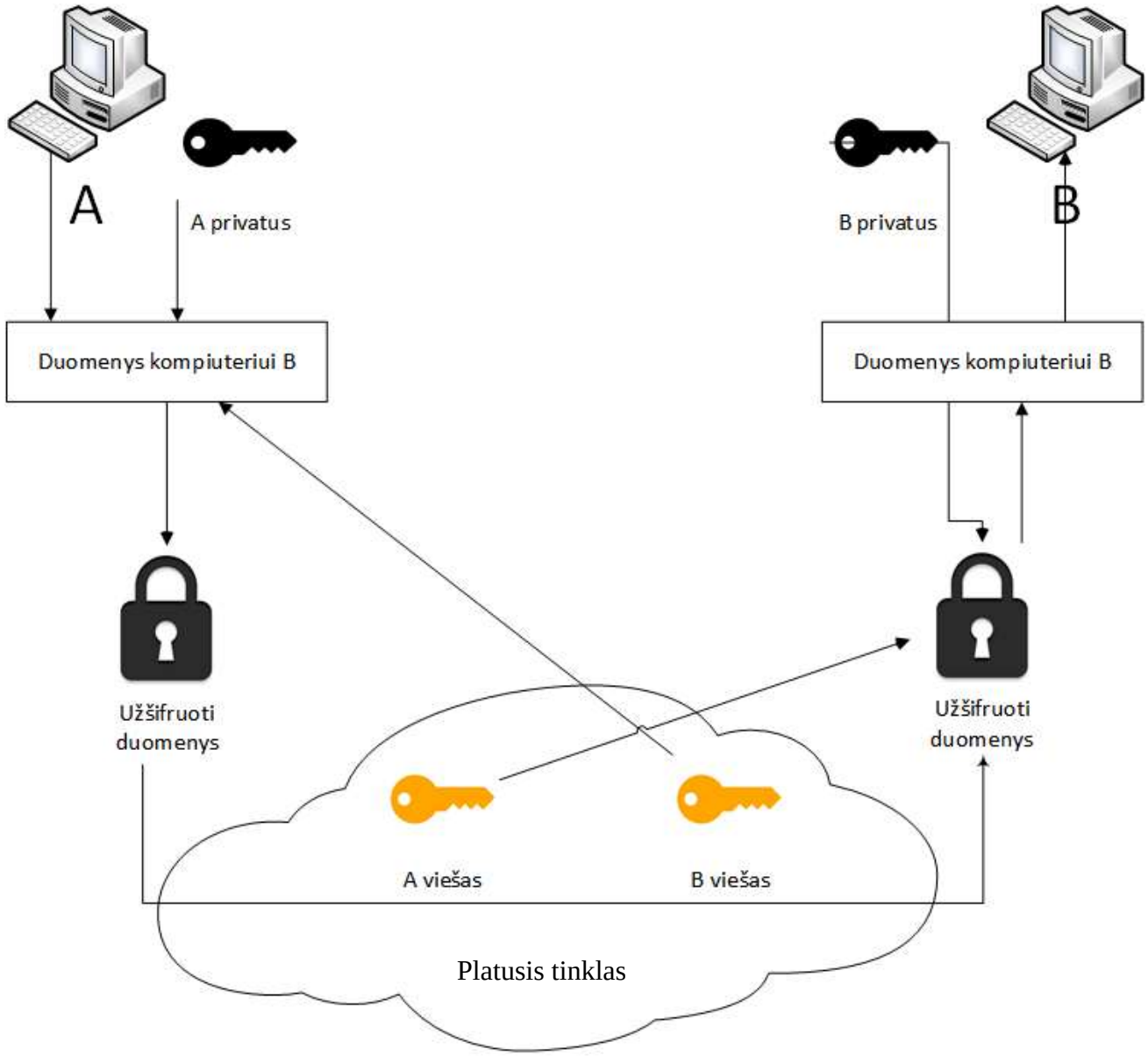
Dinaminių raktų grupė išsiskiria tuo, kad šifravimo raktai nėra įrašomi į nustatymų failą. Dėl jų yra susitariama duomenų persiuntimo pradžioje ir jie galioja susitartą laiką. Praėjus tam laikui yra susitariama dėl naujų raktų. Šis metodas yra daug saugesnis, nes negalima taip lengvai gauti šifravimo rakto kaip iš nustatymų failo. Šį metodą naudoja šie sprendimai:

- IPSec (IKE apsikeitimas)
- OpenVPN (sertifikatų naudojimas)
- Hamachi
- TeamViewer

- Stunnel

Populiariausias metodas apsikeisti simetriniu raktu viešajame internete yra Diffie-Hellman schema. Šiame metode naudojamas viešas ir privatus raktai. Šiuos raktus kiekvienas naudotojas gali susigeneruoti pats. Naudotojas privalo viešą raktą skelbti viešai, kad bet kas norintis keistis duomenimis galėtų jį paimti ir juo naudotis, bet privatų raktą kiek įmanoma labiau saugoti, kad niekas daugiau jo negalėtų paimti ir jo pagalba pasinaudoti kitu naudotoju. Šis metodas naudojamas apsikeisti sesijos rakto apsikeitimui.

Duomenų šifravimui yra naudojamas gavėjo viešasis raktas (kompiuteris B). Toks šifravimas užtikrina, kad siunčiamus duomenis galės perskaityti tik gavėjas. Siekiant užtikrinti, kad šifruotus duomenis siuntė būtent siuntėjas (kompiuteris A) siunčiami duomenys yra pasirašomi siuntėjo privačiu raktu.. Ši procedūra pavaizduota 5 paveiksle.



5 pav. Duomenų šifravimas ir autorizavimas viešo ir privataus raktų metodu

Taip pat, 5 paveiksle pavaizduota ir dešifravimo procedūra Diffie-Hellman algoritmu. Dešifravime naudojimas siuntėjo viešasis raktas ir tada gavėjo privatus raktas. Šių raktų pagalba galėsime iššifruoti duomenis ir žinoti, kad duomenis gali perskaityti tik gavėjas ir, kad duomenis siuntė kompiuteris A.

2.4.2.2 Šifravimo algoritmai

Norint užtikrinti, kad duomenys nebūtų perskaityti trečių šalių kol jie keliauja viešuoju tinklu yra šifruojami. 1 lentelėje pavaizduota koks sprendimas kokius šifravimo algoritmus palaiko.

1 lentelė. Naudojami šifravimo algoritmai

	Frame relay	MPLS	OpenVPN	IPSec	GRE	Hamachi	TeamViewer	Stunnel
3DES			+	+	+			+
AES			+	+		+	+	+
RC4			+					+

Iš 1 lentelės duomenų galima teigti, kad visuose sprendimuose naudojami tokie patys šifravimo algoritmai.

2.4.2.2.1 AES

Advanced Encryption Standart (AES) – blokinis šifravimo algoritmas, dar žinomas kaip Rinjdael. Šis šifravimo metodas gali šifruoti 32, 128 ir 256 bitų duomenų blokais. Juos atitinkamai šifruoja 128, 12 ir 256 bitų šifrais. Šis algoritmas plačiai paplitęs šiuolaikinėse sistemose ir iki šiandien nėra žinoma nulaužimo atvejų.

2.4.2.2.2 DES ir 3DES

Data Encryption Standard (DES) – vienas seniausių blokinio šifravimo algoritmų. Jis sukurtas 1977 metais. Dėl itin mažo šifro rakto dydžio, kuris sudaro tik 56 bitus, yra laikomas nesaugiu, nes su šiandieniniais kompiuterių pajėgumais jį įveikti trunka iki dienos. Kadangi šis algoritmas yra plačiai paplitęs aparatinėje įrangoje ir senose sistemose buvo sugalvotas algoritmo atnaujinimas (3DES), kuris leidžia algoritmui gyvuoti šiandienėse sistemose.

Triple Data Encryption Algorithm (3DES) – tas pats algoritmas kaip DES, tik duomenys yra šifruojami tris kartus. Dėl šio ciklo šifravimo rakto ilgis išauga iki 168 bitų. Šis algoritmas buvo sukurtas kaip senosios technologijos palaikymas. Be to ši algoritmo versija yra atsparesnė visų reikšmių perrinkimo atakai.

2.4.2.2.3 RC4

Jis nuo pat pradžių buvo kuriamas kaip srauto šifratorius. Dėl algoritmo įgyvendinimo paprastumo labai plačiai paplito mobiliuose prietaisuose. Deja, jis yra pakankamai lengvai nulaužiamas. Žinomos atakos:

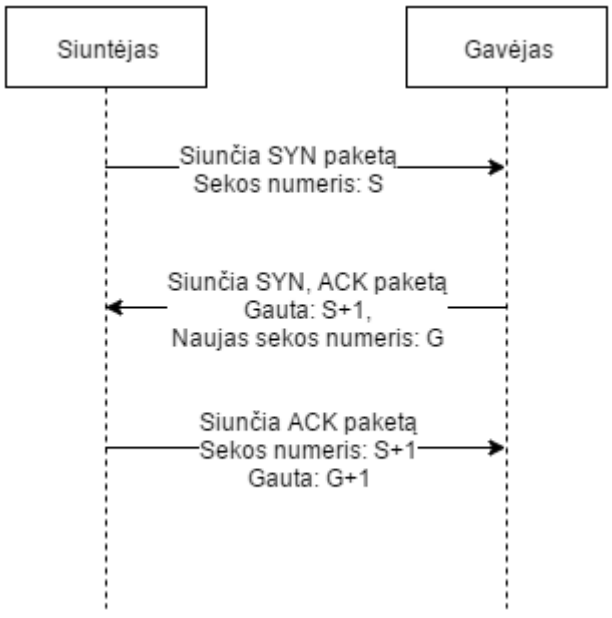
- Fluhrer, Mantin ir Shamir ataka
- Klein's ataka
- Royal Holloway ataka

2.4.2.3 Duomenų perdavimas

Kuriant saugų tunelį nuo siuntėjo iki gavėjo reikia apsibrėžti kokios garantijos duomenims bus teikiamos. Tyrimo metu bus bandoma įvertinti kuris transporto protokolas optimalus naudojamai programinei įrangai.

2.4.2.3.1 Transmission Control Protocol

Transmission Control Protocol (TCP) (1) (2) – vienas populiariausių ir dažniausiai naudojamų transporto protokolų interneto tinkle . Šis protokolas pasižymi patikimu persiunčiamų duomenų kiekio valdymo mechanizmu. Šis protokolas pristatytas visuomenei 1981 metais ir iki šiol yra mažai pasikeitęs. TCP protokolas prieš pradėdamas bet kokius duomenų siuntimus užmezga sesiją. Sesijos užmezgimas pavaizduotas 6 paveiksle (3). Ji vadinama trijų etapų pasisveikinimu, angliškai „Three Way Handshake“.



6 pav. TCP protokolo sesijos užmezgimas

Sesijos užmezgimas (6 paveikslas) suteikia šiam protokolui patikimumo, nes yra žinoma, kad gavėjas yra pasiekiamas ir yra pasiruošęs gauti duomenis. Didinant patikimumą protokolas siunčia daug tarnybinės informacijos. Po tam tikro nusiųstų duomenų kiekio laukiama patvirtinimo iš gavėjo,

kad jie gauti. Negavus atsakymo visi siųsti duomenys yra pakartotinai siunčiami dar kartą, kol bus gautas patvirtinimas iš gavėjo.

Protokolo pralaidumas priklauso nuo tinklo patikimumo ir gavėjo siuntėjo spartumo, bendro kanalo pralaidumo, bei nuo protokolo realizacijos kokybės. Jei gavėjas atsakinėja apie gautus duomenis greitai tikrinimas yra atliekamas vis rečiau. Ši savybė vadinama lango taisykle.

TCP protokolas naudojamas aplikacijose, kuriose reikalingas patikimumas ir užtikrintumas, kad visi duomenys pasiekė gavėją. Šio protokolo paketo struktūra pavaizduota 7 paveiksle.

<table>
<tr><td colspan="3">Siuntėjo prievadas 16</td><td>Gavėjo prievadas 16</td></tr>
<tr><td colspan="4">Sekos numeris 32</td></tr>
<tr><td colspan="4">Patvirtinimo numeris 32</td></tr>
<tr><td>Antraštė 4</td><td>Rezervuota 6</td><td>Požymiai 6</td><td>Lango dydis 16</td></tr>
<tr><td colspan="3">Patikros suma 16</td><td>Pirmumo rodyklė 16</td></tr>
<tr><td colspan="4">Opcijos</td></tr>
<tr><td colspan="4">Duomenys</td></tr>
</table>

7 pav. TCP paketo struktūra

Iš TCP paketo struktūros, 7 paveikslas,matyti, kad jame perduodama daug tarnybinės informacijos, kuri sudaro 144 baitus. Vienu TCP protokolo paketu daugiausiai galime perduoti 65391 baitų naudingos informacijos. Šis paketas gali būti išskaidytas į mažesnius dėl žemesnių lygmenų protokolų apribojimų.

2.4.2.3.2 User Datagram Protocol

User Datagram Protocol (UDP) – priešingybė TCP protokolui (4) (2). Šis protokolas neužmezga sesijos su gavėju, nereikalauja patvirtinimų apie gautus duomenis. UDP tiesiog išsiunčia duomenis ir pamiršta, kad duomenys buvo išsiųsti. Gavėjas gali net negauti išsiųstų duomenų dėl nepatikimo tinklo ir protokolas to neparodys ir nesiims jokių veiksmų duomenų pakartojimui. Visas šias funkcijas turi atlikti aukštesnio lygmens protokolai arba aplikacijos.

Šis protokolas pasižymi didele sparta, nes naudoja tik būtiniausią informaciją, kad paketas būtų išsiųstas gavėjui. UDP naudojamas aplikacijose, kuriose svarbu greitaveika, o pamesti ar neatkeliavę duomenys laiku nebebus aktualūs.

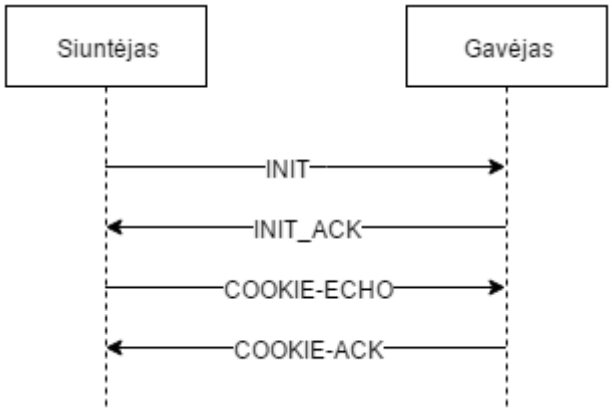
Siuntėjo prievadas 2	Gavėjo prievadas 2
Duomenų ilgis 2	Duomenų patikros reikšmė 2
Duomenys	

8 pav. UDP paketo struktūra

8 paveiksle pateikiama UDP paketo struktūra. Iš jos galime matyti, kad tarnybinė informacija sudaro tik 4 baitai. Šis protokolas vienu paketu gali perduoti 65507 baitus duomenų. Šis paketas gali būti išskaidytas į mažesnius dėl žemesnių lygmenų protokolų apribojimų.

2.4.2.3.3 Stream Control Transmission Protocol

Steram Control Transmission Protocol (SCTP) (5) – protokolas turintis visas gerąsias TCP ir UDP protokolų savybes. Šis protokolas prieš siųsdamas duomenis su gavėju užmezga sesiją. Sesijos užmezgimas pavaizduotas 9 paveiksle. Ją sudaro keturi etapai, todėl ji vadinama keturių etapų pasisveikinimu, angliškai „4-way handshake“.



9 pav. SCTP protokolo sesijos užmezgimas

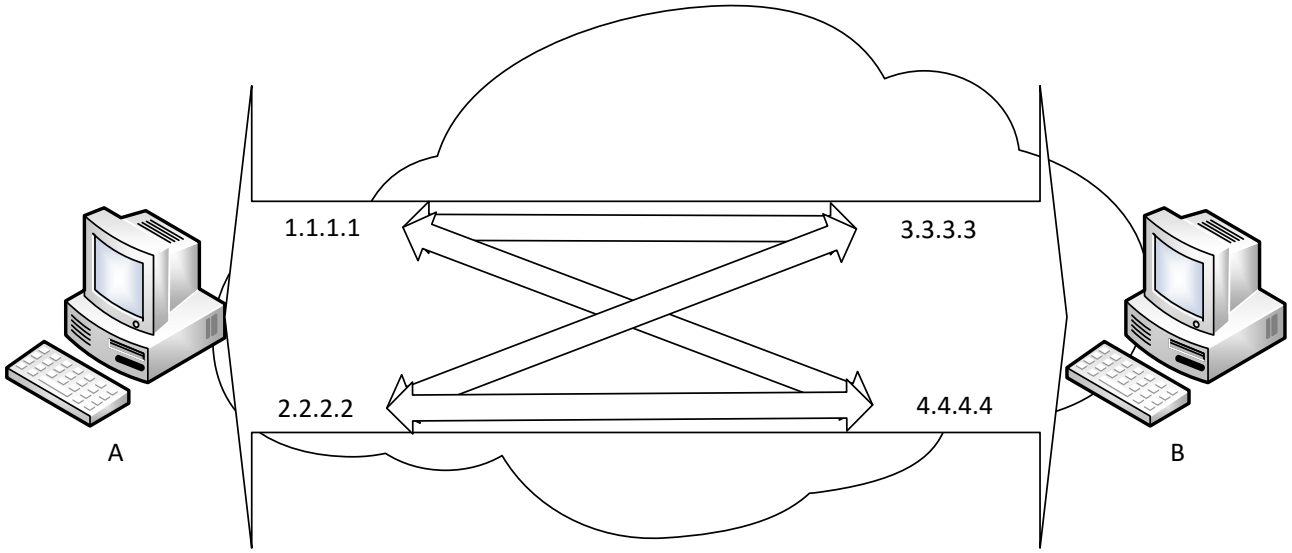
Dėl sesijos užmezgimo, protokolas, užtikrina, kad gavėjas gaus duomenis. Duomenų perdavimui naudojama panaši paketo struktūra kaip UDP protokolo pakete. Ši struktūra pavaizduota **10** paveiksle.

Siuntėjo prievadas 2	Gavėjo prievadas 2
Patikros žymė 4	
Duomenų patikros suma 4	
Duomenys	

10 pav. SCTP protokolo paketo struktūra

Šio protokolo pakete (**10** paveikslas) tarnybinė informacija užima 12 baitų, likę 65495 baitai yra skirti duomenims. SCTP protokolas leidžia skirstyti duomenis į srautus. Kiekvienam srautui pridedama

papildoma 4 baitų antraštė. Perduodant duomenis SCTP protokolo paketu, srautų antraštės užima duomenims skirtą vietą. Taip pat šis protokolas pasižymi savybe, kuri leidžia sesiją užmėgsti visais siuntėjo ir gavėjo adresais. Ši savybė pavaizduota 11 paveiksle.



11 pav. SCTP protokolo sesijos palaikymas su visais kompiuterio adresais

Ši savybė leidžia judant tarp tinklų, keičiantis adresams, nenutraukti duomenų perdavimo.

Atlikęs transporto protokolų analizę nustačiau, kad kiekvienas protokolas yra savitai vertingas ir gali veikti mano sprendime, netgi UDP protokolas, kuris neužtikrina duomenų gavimo. Realizuojant šį protokolą reikia pasirūpinti duomenų perdavimo kontrole, pačioje kuriamoje programinėje įrangoje. Glaustas transporto protokolų savybių apibendrinimas 2 lentelėje.

2 lentelė. Transporto protokolų apibendrinimas

	TCP	UDP	SCTP
Užmezga sesiją	Taip	Ne	Taip
Patikimas perdavimas	Taip	Ne	Taip
Užmezga vieną sesiją su visais turimus IP adresus	Ne	Ne	Taip
Palaiko srautus sesijoje	Ne	Ne	Taip

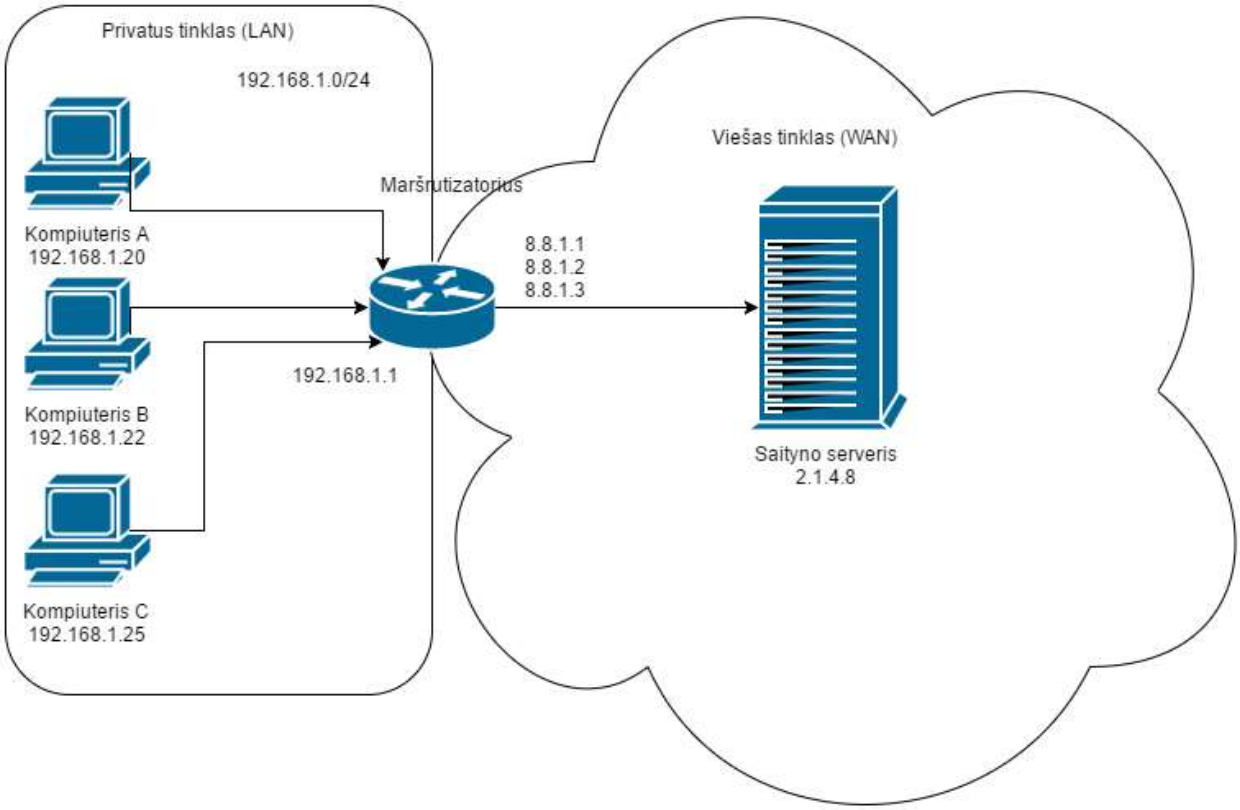
2.5 Tinklų adresų perrašymo technologija

Tinklų adresų perrašymo technologijos (angliškai network address translation) (6) (7), toliau NAT, pagrindinė paskirtis perrašyti siunčiamų duomenų interneto adresus. Ši technologija buvo kuriama lengvesniam duomenų nukreipimui interneto tinklu nepernumeruojant kiekvieno kompiuterio (NAT technologija vienas su vienu). Tačiau išaugus interneto naudotojų skaičiui ir pradėjus trūkti globalių adresų buvo patobulinta (NAT technologija daug su vienu) ir pradėta plačiai diegti interneto

tiekėjų. Taip pat yra galimybė naudoti daugiau nei vieną globalų adresą. Ši NAT technologija vadinama daug su daug.

2.5.1 NAT technologija vienas su vienu

NAT technologija vienas su vienu, arba dar vadinama statiniu NAT, išleidžia į internetą kiekvieną klientą su savo globaliu adresu. Šio tipo NAT technologijai yra reikalinga tiek globalių adresų kiek yra vidinių prietaisų, norinčių pasiekti platųjį interneto tinklą. Šios technologijos pavyzdys pateikiamas 12 paveiksle.

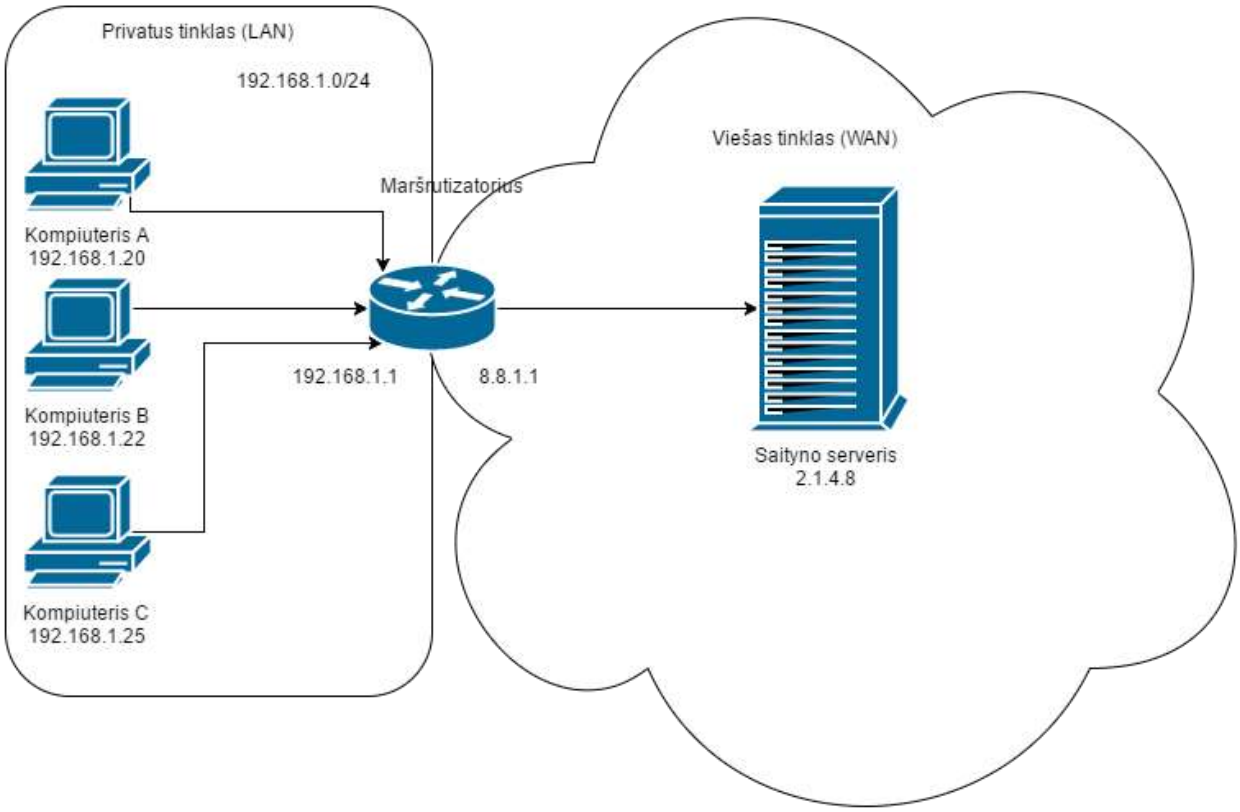


12 pav. NAT technologijos vienas su vienu pavyzdys

Pagal pateiktą pavyzdį 12 paveiksle kiekvienas klientas turi po sau priskirtą globalų adresą. Sakykime klientas A turi priskirtą adresą 8.8.1.1, o klientai B ir C atitinkamai 8.8.1.2 ir 8.8.1.3. Siųsdami duomenis saityno serveriui jų vietinio tinklo adresai bus perrašomi į jiems priskirtus globalius adresus. Sakykime klientas A siunčia duomenis saityno serveriui per maršrutizatorių. Maršrutizatorius perrašys siuntėjo adresą iš vietinio adreso į viešąjį adresą 8.8.1.1 ir persiųs duomenis į saityno serverį. Serveris matys, kad duomenys atėjo iš globalaus adreso 8.8.1.1. Siunčiant duomenis klientui B maršrutizatorius perrašys vidinį adresą 192.168.1.22 į viešąjį adresą 8.8.1.2 ir persiųs saityno serveriui. Tuomet saityno serveris matys, kad duomenys atėjo iš įrenginio turinčio globalų adresą 8.8.1.2. Šio tipo NAT leidžia vietinius įrenginius pasiekti iš viešojo tinklo netrukdomai ir visuomet.

2.5.2 NAT technologija daug su vienu

Šios technologijos leidžia išleisti į platųjį internetą N prietaisų, perrašydama siuntėjo adresą į viešąjį ir modifikuodama siuntėjo prievadą. Perrašydama siuntėjo prievadą pasižymima kuriam klientui ir į kokį jo prievadą turi būti perduodami grįžtantys duomenys. Tokiu būdu nėra pametami duomenys ir užtikrinamas bendravimas naudojant vieną globalų adresą. Šios technologijos pavyzdys pavaizduotas 13 paveiksle.

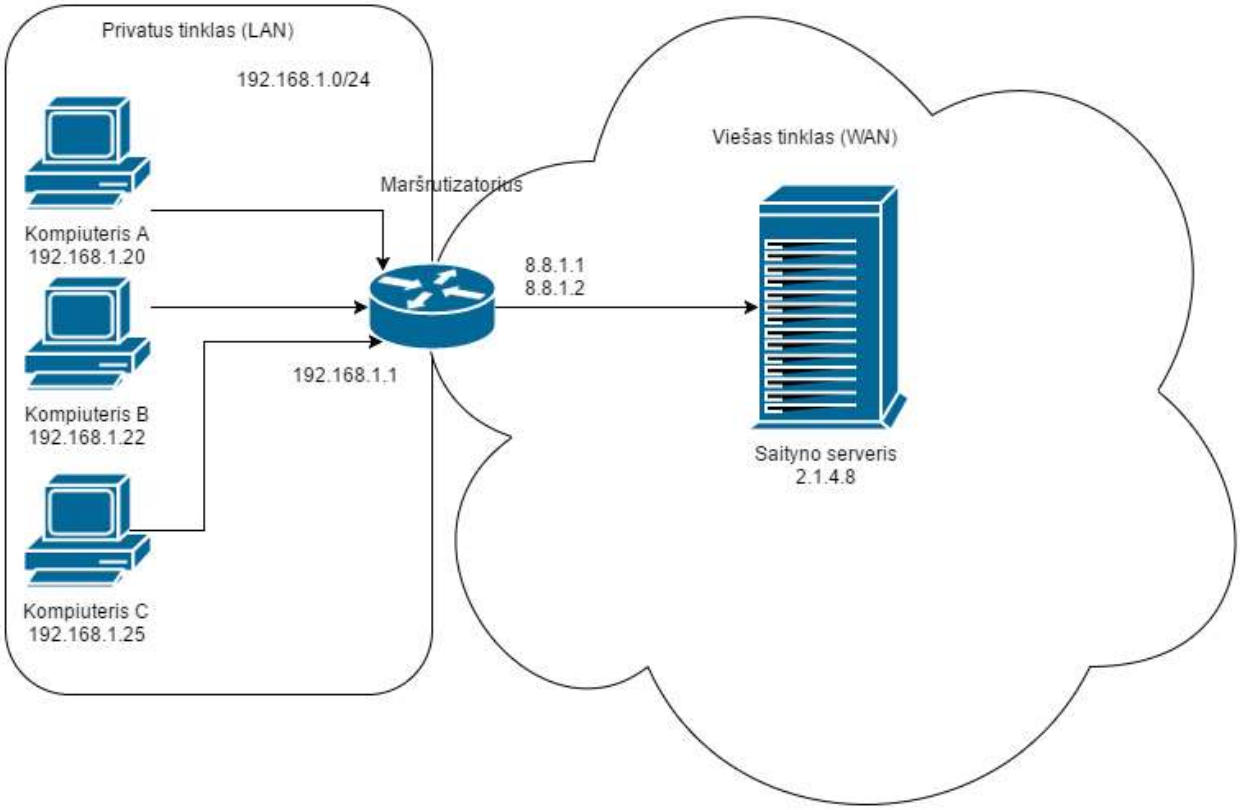


13 pav. NAT technologijos daug su vienu pavyzdys

Pagal pateiktą pavyzdį, 13 paveikslas, klientai A,B ir C siunčia duomenis saityno serveriui, adresu 2.1.4.8. Šių klientų duomenys perduodami per maršrutizatorių. Maršrutizatoriuje pažymima, kad klientas A iš prievado PA siunčia duomenis į saityno serverio 80 prievadą. Tuomet perrašo siuntėjo adresą į maršrutizatorius globalų adresą- 8.8.1.1 ir išsiunčia naudodamas prievadą MP. Saityno serveris gaudamas duomenis mato, kad juos siunčia maršrutizatorius iš adreso 8.8.1.1 ir prievado MP, o ne klientas A iš prievado PA. Siųsdamas atsakymą saityno serveriui siunčia adresu 8.8.1.1 į prievadą MP. Maršrutizatorius gavęs duomenis į prievadą MP, juos persiunčia klientui A į prievadą PA. Tokiu būdu klientai A,B ir C gali bendrauti su saityno serveriui naudojant vieną globalų adresą.

2.5.3 NAT technologija daug su daug

NAT technologija daug su daug leidžia bendrauti N įrenginių viešajame internete naudojant M globalių adresų. Šios technologijos pavyzdys pateikiamas 14 paveiksle.



14 pav. NAT technologijos daug su daug pavyzdys

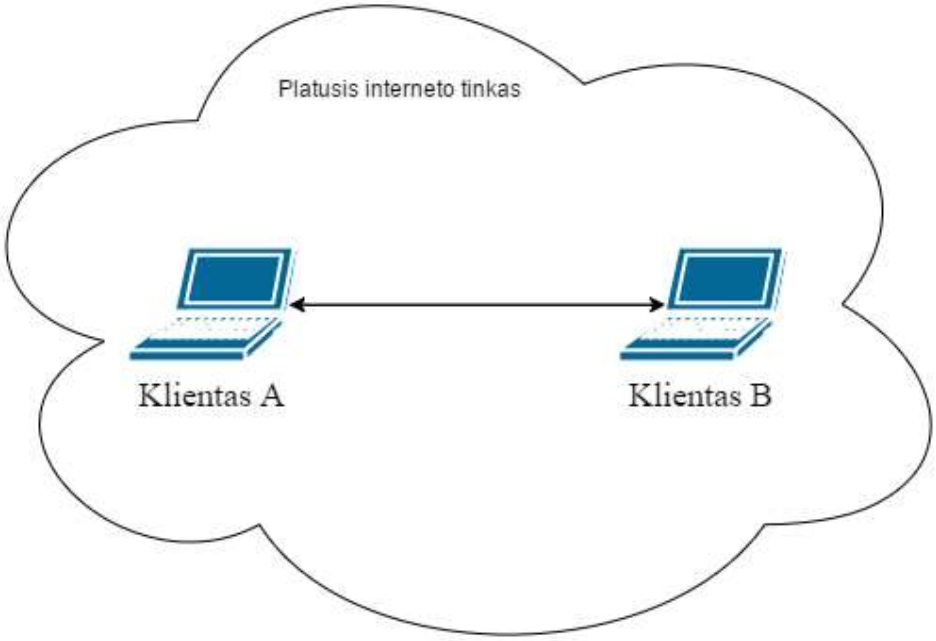
Pagal pateiktą pavyzdį 14 paveiksle matome, kad klientas A,B ir C su viešuoju internetu gali bendrauti naudojant du viešuosius adresus- 8.8.1.1 ir 8.8.1.2. Ši technologija veikia taip pat kaip NAT technologija daug su vienu, tačiau klientų A,B ir C duomenys siunčiami saityno serveriui per maršrutizatorių vieną kartą, gali būti siunčiami naudojant 8.8.1.1 globalų adresą, kitą kartą naudojant kitą globalų adresą- 8.8.1.2. Ši technologija palaiko M globalių adresų skaičių.

2.6 Programinės įrangos architektūra

Kuriama programinė įranga turi užtikrinti didžiausią įmanomą pasiekiamumą, todėl jos architektūra turi didelę įtaką įgyvendinant šį reikalavimą. Dažniausiai naudojamos klientas- klientas sujungimo architektūra arba klientas- serveris. Tolimesniuose skyriuose šiuos architektūros modelius analizuosiu atskirai.

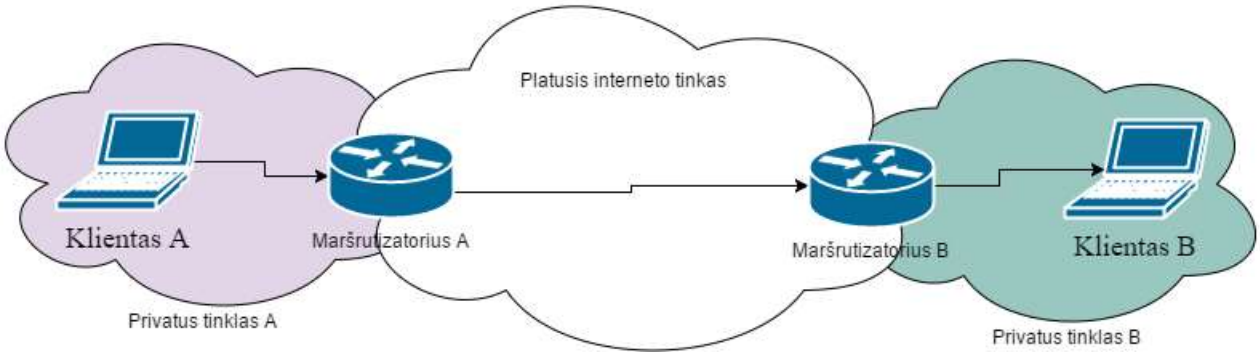
2.6.1 Klientas- klientas architektūra

Klientas- klientas architektūros principas pagrįstas tuo, kad klientai bendrauja tiesiogiai vienas su kitu be tarpininko. Šią architektūrą dažnai naudoja realaus laiko bendravimo sistemos. Pirminis šios architektūros modelis pavaizduotas 15 paveiksle.



15 pav. Klientas- klientas architektūros pavyzdys

15 paveiksle matome, kad klientas A su klientu B bendrauja tarpusavyje tiesiogiai, be papildomų tarpininkų. Šis bendravimo tipas galimas abiem klientams turint globalius adresus. Šios architektūros pavaizdavimas (15 paveikslas) buvo plačiai paplitęs ankstyvuosiuose interneto gyvavimo metuose. Šiandieninė šios architektūros schema turi būti perbraižyta pagal 16 paveikslą Šis pasikeitimas įvyko dėl interneto vartotojų skaičius augimo ir NAT technologijos naudojimo.



16 pav. Klientas- klientas architektūros pavyzdys su privačiais tinklais

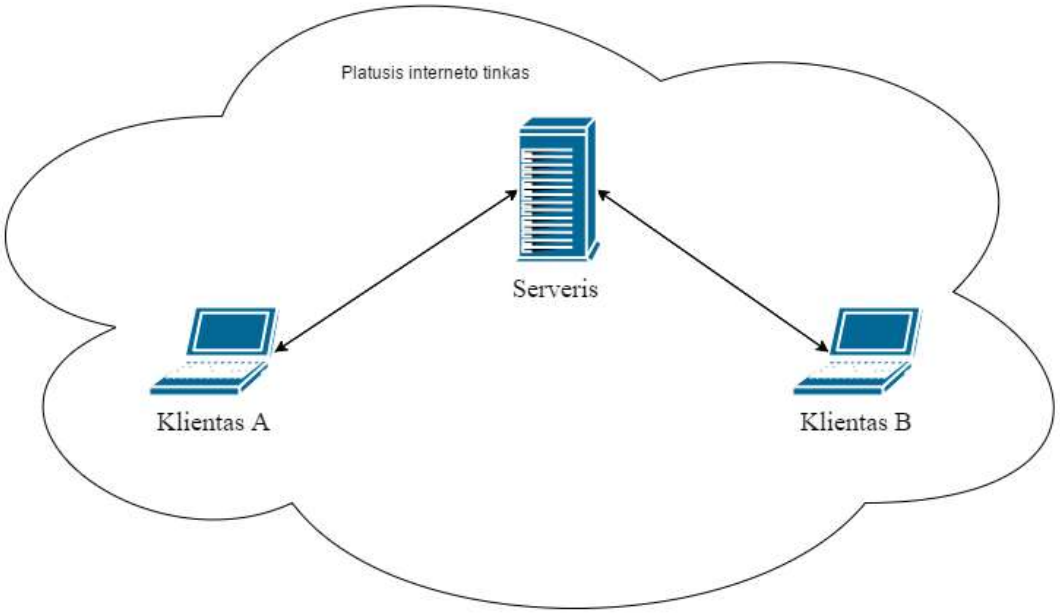
Pagal pavyzdį su privačiais tinklais, žiūrėti 16 paveikslą, matome, kad klientas A iš vietinio tinklo A per maršrutizatorių A, naudodamasis platųjį, viešojo interneto tinklą nori pasiekti klientą B, esantį privačiame tinkle B, eidamas per maršrutizatorių B. Šiam scenarijui kliudo maršrutizatoriai dirbantys

su NAT technologija. Šis scenarijus negalimas be papildomų abiejų maršrutizatorių nustatymų arba be papildomų protokolų naudojimo kaip UPnP (8), tačiau ne visi maršrutizatoriai palaiko šį protokolą.

Atsižvelgiant į kuriamos sistemos reikalavimus, norint naudoti šią architektūrą šiandien reikėtų įdiegti protokolų palaikymą, kurie leistų užmegzti ryšius su klientais esančiais už NAT įrenginių.

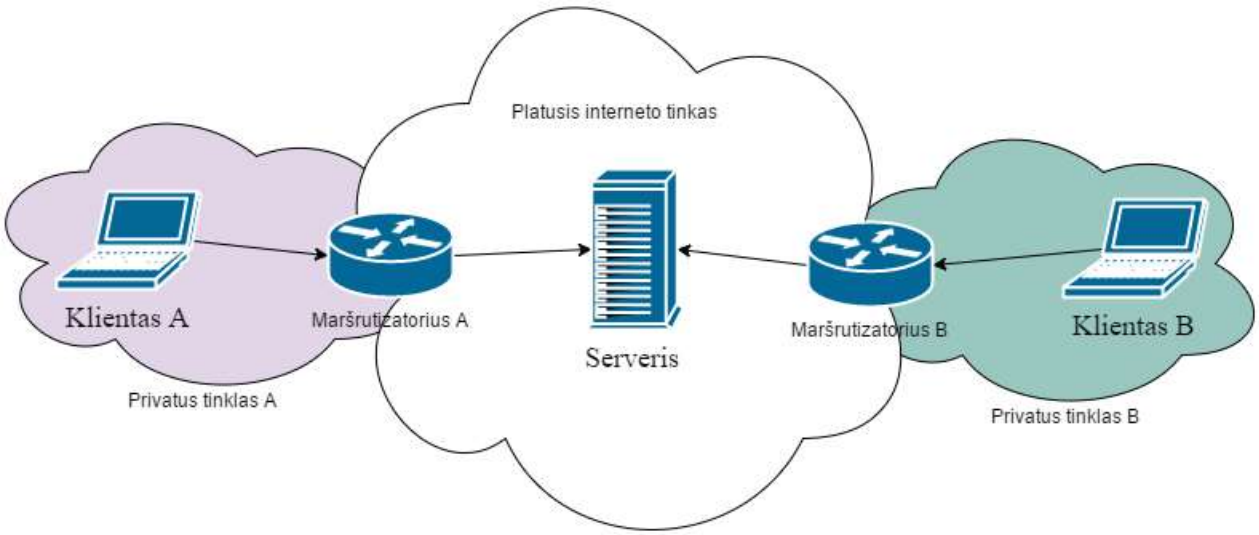
2.6.2 Serveris- klientas architektūra

Serveris- klientas architektūros principas pagrįstas tuo, kad visa komunikacija tarp klientų vyksta per tarpininką- centrinį serverį. Ši architektūra dažnai naudojama nuolatiniam resursų pasiekiamumui kaip interneto svetainės, pokalbių kambariai ar žaidimų serveriai. Šis modelis realizuojamas naudojant 3 globalius adresus. Ankstyvajame internete ši architektūra pavaizduota 17 pav.



17 pav. Serveris- klientas architektūros pavyzdys

Pagal pavyzdį pateiktą 17 paveiksle matome, kad klientai A ir B jungiasi į centrinį serverį. Klientas A norėdamas nusiųsti informaciją klientui B pirmiausia ją turi siųsti serveriui, o tuomet serveris ją persiųs klientui B. Dėl tokio duomenų perdavimo scenarijaus pailgėjo duomenų perdavimo laikas. Scenarijus pavaizduotas 17 paveiksle buvo naudojamas ankstyvaisiais interneto metais. Išaugus interneto klientų skaičiui ir įdiegus NAT technologiją ši architektūra pasikeitė į pavaizduotą 18 paveiksle.



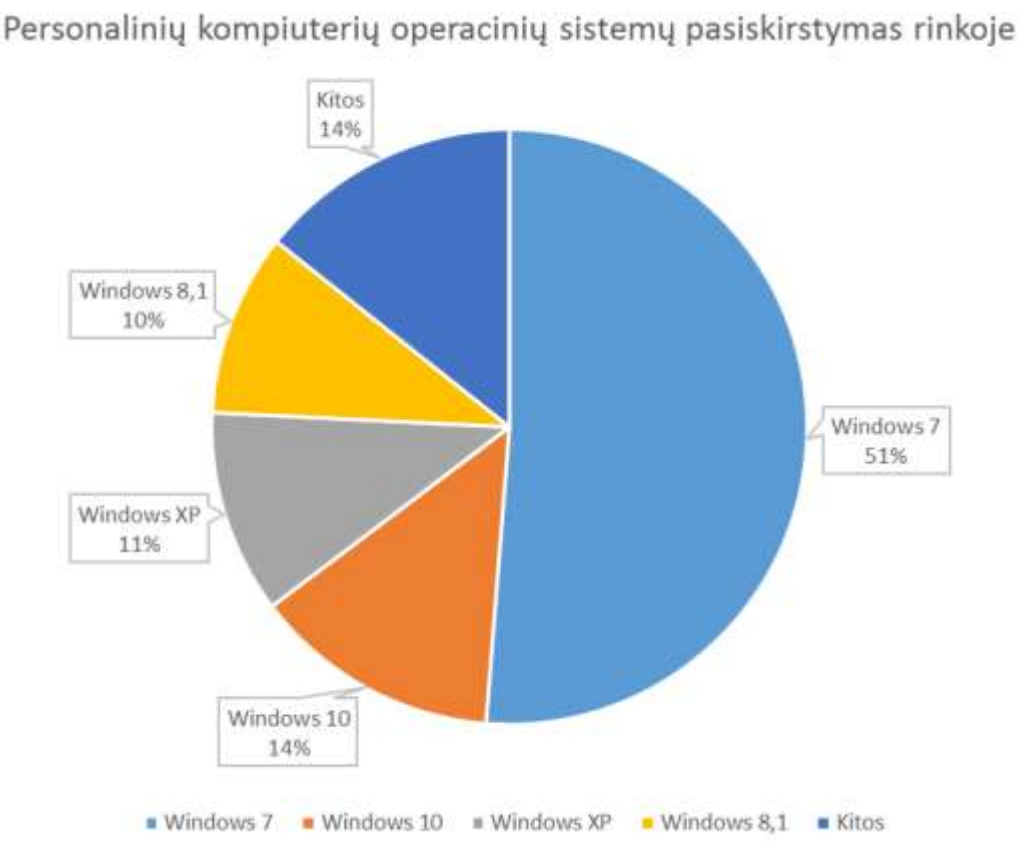
18 pav. Serveris- klientas architektūros pavyzdys su privačiais tinklais

Pagal schemą pateiktą 18 paveiksle matome, kad klientas A norėdamas užmegzti sujungimą su klientu B turės jungtis į centrinį serverį iš privataus tinklo A per maršrutizatorių A, taip pat klientas B turės jungtis į centrinį serverį iš privataus tinklo B per maršrutizatorių B. Abiem klientams prisijungus prie centrinio serverio, centrinis serveris turės suorganizuoti duomenų persiuntimą iš kliento A pas klientą B. Ši architektūra nereikalavo papildomų protokolų įgyvendinimo pridėjus maršrutizatorius su NAT įrenginiais, tačiau visą ryšių komutavimą tarp klientų turi stabiliai realizuoti centrinis serveris. Šis modelis realizuojamas mažiausiai naudojant trys globalius adresus, o komutuojamų sesijų skaičius priklauso nuo klientų poreikio.

Šis modelis suteikia didžiausią klientų pasiekiamumą ir nereikalauja papildomų veiksmų pridėjus papildomus įrenginius- maršrutizatorius. Dėl šių veiksmų kuriamoje sistemoje turėtų būti realizuotas šis architektūros modelis.

2.7 Operatyviosios sistemos pasirinkimas

Siekiant pasiekti kuo daugiau naudotojų reikia pasirinkti populiariausią operacinės sistemos aplinką kuriamos sistemos kliento programinei įrangai. Remiantis netmarketshare (9) tinklapio 2016 metų kovo mėnesio duomenimis, nustačiau, kad pati populiariausia operacinė sistema šiai dieniai- Windows 7, žiūrėti 19 paveikslą.



19 pav. Personalinių kompiuterių operacinių sistemų pasiskirstymas rinkoje. Atnaujinta 2016-05-06

Statistikoje, pavaizduota 19 paveiksle, buvo analizuojama tik personalinių kompiuterių rinka. Operacinės sistemos, kurios turėjo mažiau nei 5% rinkos, nebuvo išskiriamos į atskirą šaką, o pridedamos prie šakos- kitos.

Centrinio serverio programinė įranga bus kuriama naudojant Debian Linux (10) operacinės sistemos aplinką. Šis pasirinkimas atliktas nes:

- Linux šeimos operacinės sistemos turi žemo lygmens SCTP protokolo palaikymą
- Linux šeimos operacinės sistemos reikalauja mažiau resursų nei Windows šeimos serverių operacinės sistemos
- Įvertinus darbuotojų kvalifikaciją buvo pasirinkta Debian operacinė sistema
- Debian operacinei sistemai teikiami dažni ir dažniausiai stabilūs atnaujinimai

Atlikus operacinės sistemos rinkos analizę nustačiau, kad kliento programinė įranga turėtų veikti Windows operacinių sistemų aplinkoje. Minimali palaikoma versija turi būti Windows 7, nes Windows XP nuo 2014 m. balandžio 8d. nėra gamintojo palaikoma. Centrinio serverio programinė įranga kuriama naudojant Debian Linux operacinę sistemą ir turi veikti su kitomis Linux šeimos operacinėmis sistemos aplinkomis.

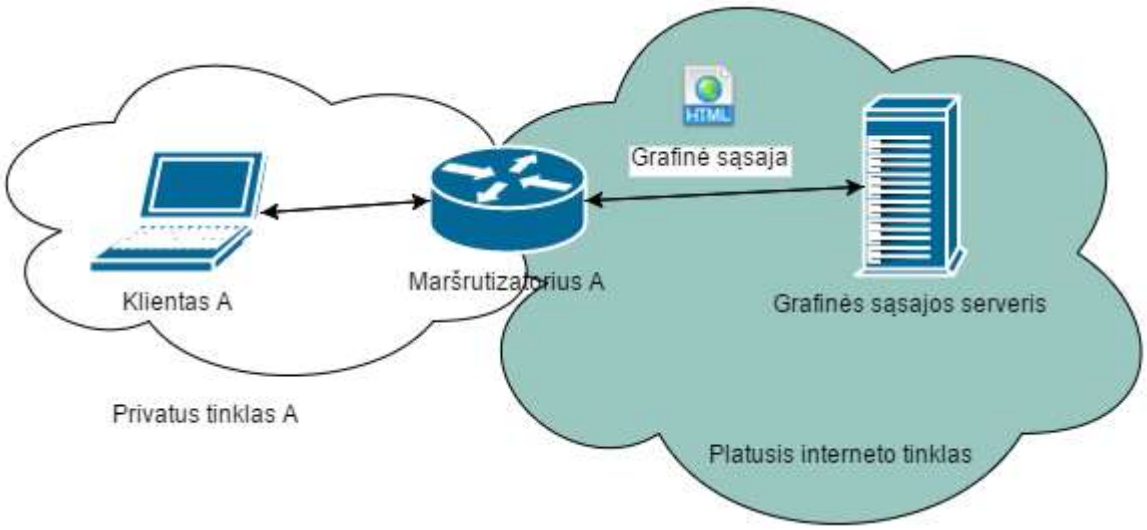
2.8 Centralizuoto nustatymo sistema

Mažinant laiko sąnaudas kuriamos sitemos paleidimui, reikia numatyti centralizuotą būdą nustatymams nustatyti. Įgyvendinus šį būdą sutrumpės kuriamos sitemos paleidimo laikas. Taip pat šis būdas būtinas norint orientuoti kuriamą programinę įrangą į verslo sektorių. Atsižvelgiant į tai, kad kalbama tik apie kliento programinę įrangą ir kad ji veiks Windows šeimos operacinės sitemos aplinkoje šiam tikslui pasiekti galima naudoti Windows Active Directory Service (11) su Group Policy (12) infrastruktūra. Kliento programinė įranga nustatymus turės saugoti operacinės sitemos registruose (13). Suteikiant galimybę kiekvienam vartotojui turėti unikalius nustatymus, kliento programinės įrangos nustatymai, bus saugomi dabartinio vartotojo registrų šakoje.

2.9 Grafinė sąsaja

Grafinė sąsaja turi būti realizuota tik kliento programinei įrangai siekiant taupyti serverio resursus. Kliento ir centrinio serverio programinės įrangos derinimo tikslams turi būti realizuota tekstinė sąsaja.

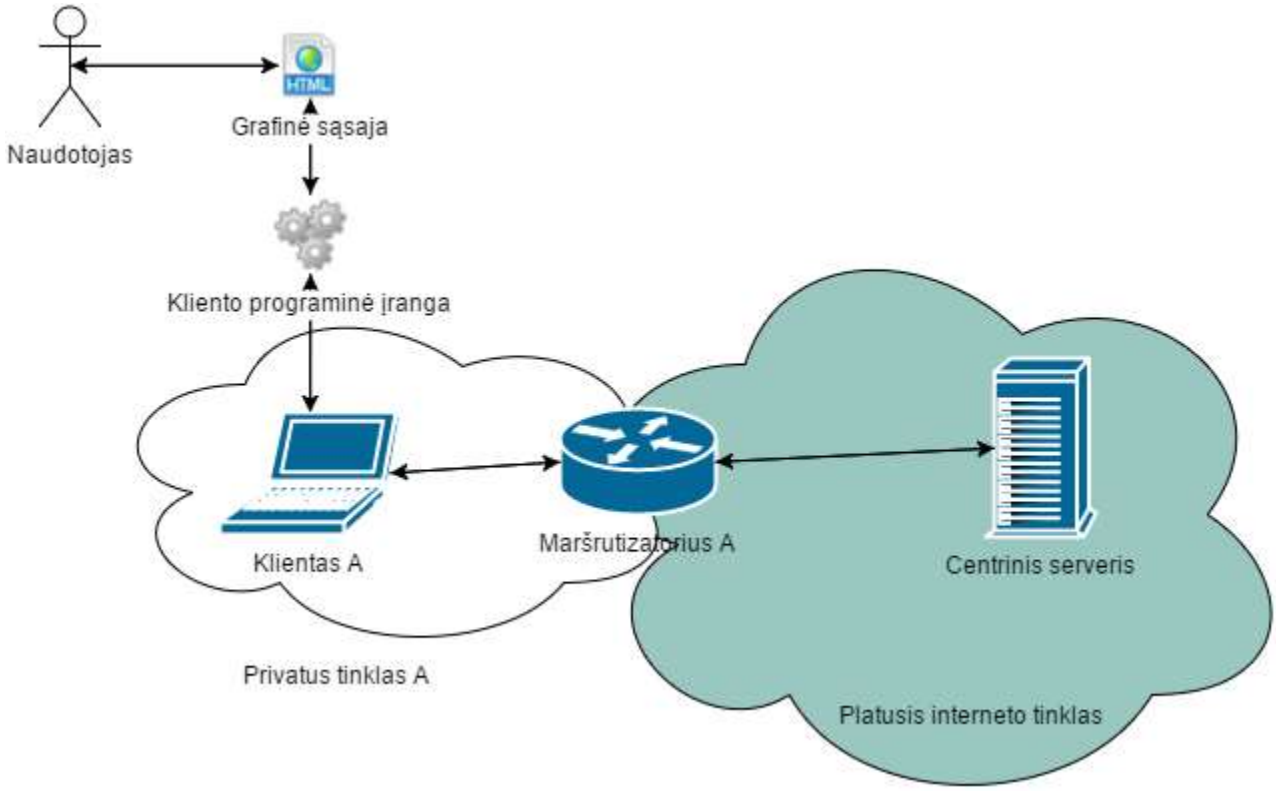
Kliento programinės įrangos grafinė sąsaja turi būti patraukli suprantama mažo raštingumo naudotojui, lengvai prižiūrima, bei lengvai keičiama. Šiuos reikalavimus puikiai atitinka grafinė sąsaja pagrįsta internetiniu tinklapiu. Taip pat tokia sąsaja leistų išskirstyti kuriamą sistemą po atskirus serverius. Galimas vartotojo grafinės sąsajos išdėstymas pateikiamas 20 pav.



20 pav. Grafinės sąsajos veikimo principas

Pagal analizuojama principą, pateiktą 20 paveiksle, grafinė sąsaja talpinama grafinės sąsajos serveryje. Ji saugoma interneto tinklapio pavidalu. Ją kompiuteris A iš privataus tinklo A, per maršrutizatorių A pasiekia ir parsisiunčia sau. Joje būtų tik statiniai komponentai reikalingi nupiešti grafinę sąsają, tai yra, stiliai, paveiksliukai, komponentų vietos ir scenarijai reikalingi piešti ir formuoti

grafinei sąsajai. Pagal pateiktą grafinės sąsajos principą, 20 paveikslas, naudotojas galėtų dirbti su kuriama sistema principu pavaizduotu 21 paveiksle.



21 pav. Naudotojo darbo su kliento programinės įrangos grafine sąsaja principas

Naudotojui atsisiuntus statinę grafinės sąsajos dalį , žiūrėti 20 paveikslą, į savo kompiuterį, toliau darbą atlieka kliento programinė įrangą, kuri užpildo grafinės sąsajos dinaminę dalį, t.y., klientų ir sujungimų sąrašą. Taip pat šios grafinės sąsajos pagalba galima inicijuoti sujungimus su kitais klientais ir juos nutraukti. Bendravimui, tarp kliento programinės įrangos ir grafinės sąsajos, reikalingas duomenų apsikeitimo standartas. Dažniausia naudojami standartai:

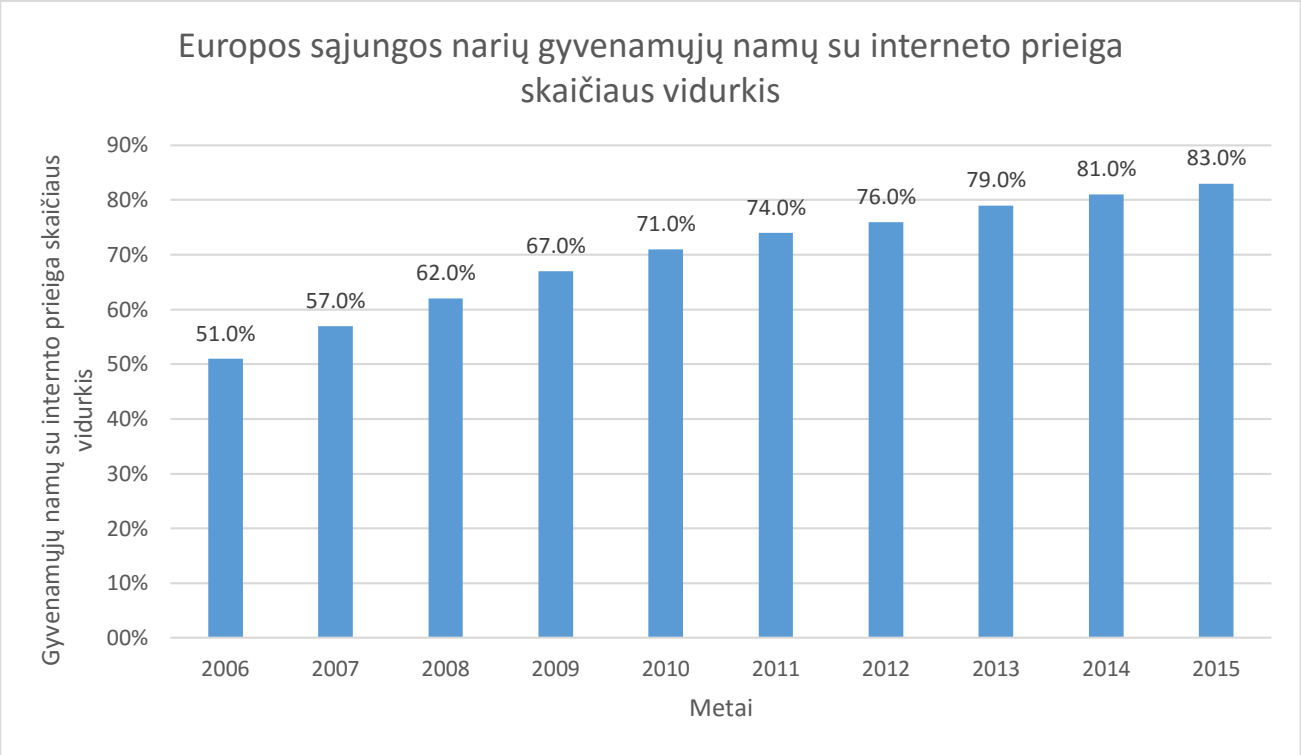
- XML (14)
- JSON (15)

Pagrindinis skirtumas tarp šių duomenų perdavimo standartų toks kad naudojant JSON standartą yra perduodama mažiau duomenų. Taip pat JSON formatas yra geriau suderinamas su JavaScript programavimo kalba dėl savo panašumo į ją.

2.10 Sistemos potencialumas

Norėdamas išsiaiškinti, ar kuriama sistema būtų potenciali ir naudojama, analizavau Europos sąjungos statistikos duomenis. Pasinaudodamas Eurostat statistikos duomenų bazę analizavau namų su interneto prieigą skaičių ir nutolusių darbuotojų skaičių.

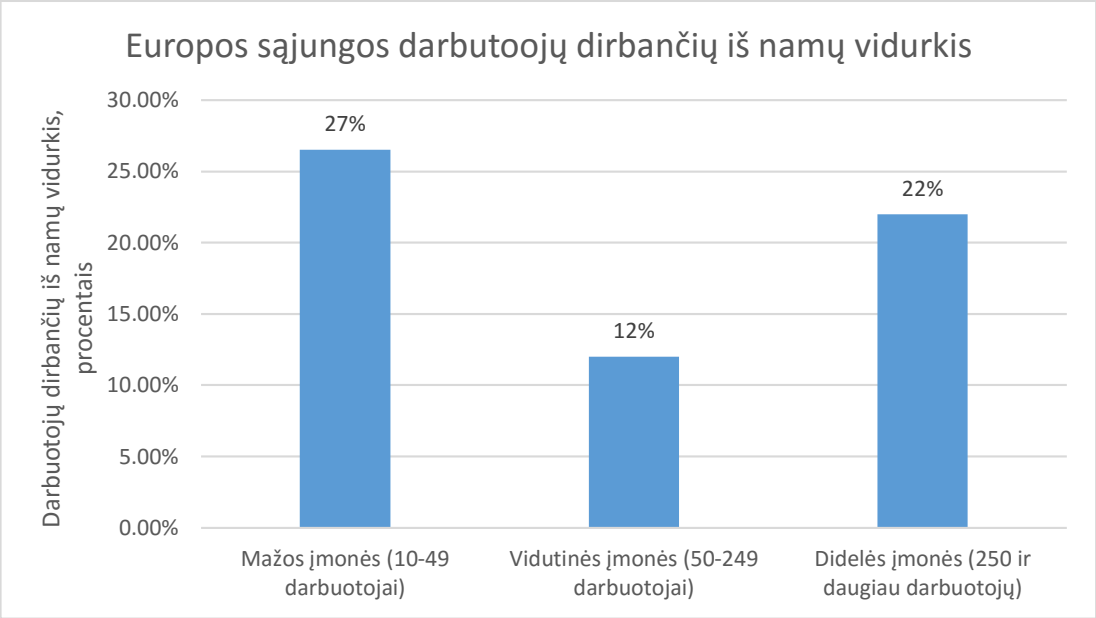
Analizuojant gyvenamųjų namų prieigą prie internato Europos Sąjungos narėse pastebima augimo tendencija, žiūrėti 22 paveikslas. Todėl kiekvienais metais vis daugiau gyventojų gauna galimybę dirbti interneto pagalba iš namų.



22 pav. Europos sąjungos narių gyvenamųjų namų su internato prieiga vidurkis

Europos sąjungoje gyvenamųjų namų su interneto prieiga vidurkis išaugo nuo 51 procento iki 83 procentų augimas stebimas nuo 2006 metų iki 2015 metų. Išsamūs duomenys pateikti 13 lentelėje prieduose.

Eurostat statistikos duomenų bazėje pateikiama darbuotojų, dirbančių iš namų tik 2006 metų statistika, žiūrėti 23 paveikslą.



23 pav. Europos sąjungos nutolusių darbuotojų skaičius procentais, 2006 metai

Pagal pateiktą statistiką, žiūrėti 23 paveikslą, matome, kad 2006 metais mažose įmonėse 27% darbuotojų dirbo iš namų naudodami internetą. Vidutinėse ir didelėse įmonėse šie skaičiai pasiskirstę atitinkamai 12% ir 22%.Atsižvelgiant į tai, kad šie duomenys yra dešimties metų senumo ir, kad nuo 2006 metų namų su interneto prieiga išaugo nuo 20%, galima manyti, kad dirbančių iš namų skaičius taip pat augo. Išsami statistika pateikiama 14 lentelėje prieduose.

Apibendrinant statistikos analizę galima teigti, kad kuriama sistema turės didelį potencialą naudojimui Europos sąjungos narėse šiomis dienomis.

2.11 Siūlomas sprendimas

Atlikus kuriamos sitemos analizę ir surinkus pagrindinius reikalavimus buvo nustatyta kokie sprendimai ar technikos turi būti realizuotos kuriamoje sistemoje. Siekiant užtikrinti didžiausią sistemos prieigą turi būti realizuota:

- TCP protokolas
- UDP protokolas
- SCTP protokolas
- Serveris- klientas architektūra

Paskirstytos sistemos sukūrimui sistema turi būti padalinta į tris dalis:

- Kliento programinę įrangą
- Serverio programinę įrangą
- Grafinės sąsajos serverį

Maksimaliam saugumui užtikrinti kuriamoje sistemoje turi būti:

- Aplikacija- aplikacija tipo sujungimas

- Šiandienos šifravimo standartus atitinkantis šifravimas (TLS, dinaminis raktų apsikeitimas, AES šifravimo algoritmas)

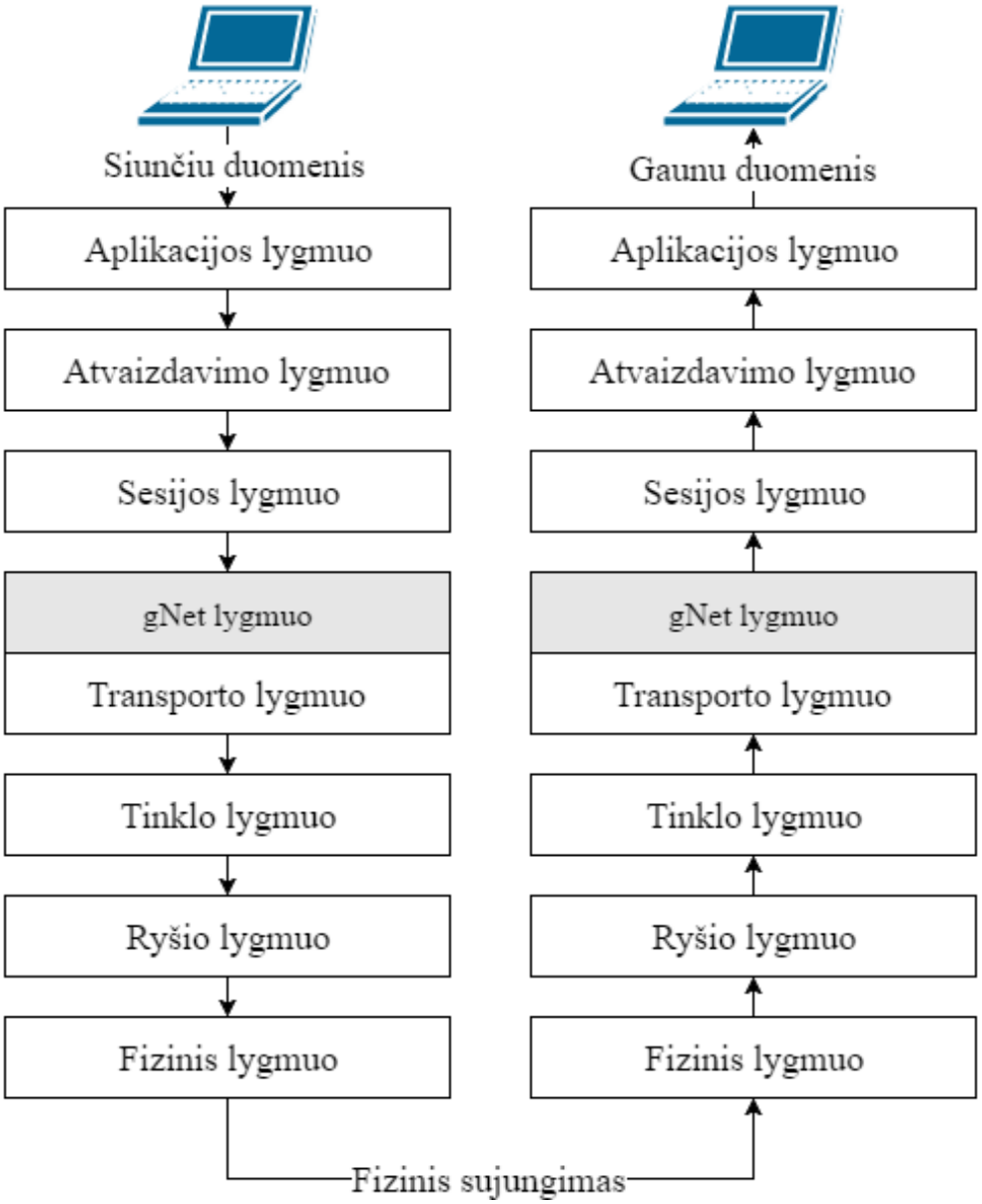
Siekiant pateikti programinę įrangą kuo didesnei naudotoju grupei reikia imtis šių veiksmų:

- Realizuoti dvi sąsajas. Tekstinę- derinimo darbams, grafinę- kasdieniam naudojimui.
- Platinti kuriamą programinę įranga nemokamai
- Platinti programinę įrangą naudojant atvirojo kodo licenziją

Siekiant pritaikyti kuriamą programinę įrangą didelėms įmonėms reikia realizuoti:

- Centralizuotai nustatyti programinės įrangos parametrus

Kliento programinė įranga turėtų veikti tarp atvaizdavimo ir sesijos lygmenų, 24 paveikslas, kad būtų galima įgyvendinti komunikaciją tarp klientų per centrinį serverį be virtualių tinklo prietaisų kūrimo.



24 pav. Kuriamos sistemos lygmuo pagal OSI modelį

3 PROJEKTINĖ DALIS

Šiame skyriuje pateikiama informacija kaip atsižvelgiant į analizės pateiktą informaciją buvo sukurta programinė. Apžvelgiamas pagrindinis funkcionalumas, nefunkciniai reikalavimai, pateikiami panaudos atvejai, supažindinama su sistemos išdėstymo vaizdu, sistemos statinis vaizdas, bei aprašomos sistemoje naudojamos komandos. Taip pat aprašoma saugumo išimtys taikomos prototipinei programinei įrangai.

3.1 Sistemos pagrindinis funkcionalumas

Kuriamos sistemos pagrindinis tikslas suorganizuoti saugų sujungimą tarp dviejų, tiesioginio sujungimo neturinčių, klientų ir juo perduoti duomenis. Šios sistemos veikimo principas pagrįstas serveris- klientas architektūra ir veikia taip: centrinis serveris veikia plačiajame internete ir yra pasiekiamas viešuoju adresu X.X.X.X. Klientai, norėdami užmėgsti saugų sujungimą vienas su kitu, jungiasi į centrinį serverį ir jame užregistruoja savo sujungimą su serveriu. Tuomet, vienas iš klientų inicijuoją sujungimą tarp klientų per centrinį serverį. Centrinis serveris suorganizuoja sujungimo kanalą tarp klientų ir veikia kaip tarpininkas, priimdamas duomenis iš vieno kliento ir persiųsdamas kitam sujungimo klientui.

3.2 Nefunkciniai reikalavimai

Kuriamai sistemai taikomi šie nefunkciniai reikalavimai:

- Kliento programinė įranga turi veikti Windows operacinėse sistemose. Nuo Windows 7 ir naujesnėse
- Kliento programinė įrangos nustatymai turi būti saugomi Windows operacinės sitemos registruose, po dabartinio naudotojo registrų šaka.
- Kliento programine įranga turi veikti paprasto naudotojo teisėmis
- Kliento programinė įranga turi veikti su tekstine ir grafine sąsaja
- Kliento programinė įranga turi veikti kitame kompiuteryje nei centrinis serveris
- Kliento programinė įranga turi veikti be virtualių tinklo įrenginių
- Centrinio serverio programinė įranga turi veikti Linux operacinėje sistemoje
- Centrinio serverio programinė įranga turi palaikyti 2 ir daugiau vienu metu prisijungusių klientų
- Centrinio serverio programinė įranga turi veikti be administratorius teisių
- Grafinės sąsajos programinė įranga turi būti realizuota internato technologijomis
- Grafinė sąsaja turi būti pateikta su šių dienų vyraujančiomis dizaino tendencijomis
- Grafinės sąsajos programinė įranga turi turėti galimybę būti patalpinata atskirame serveryje nei kliento programinė įranga ar centrinio serverio programinė įranga

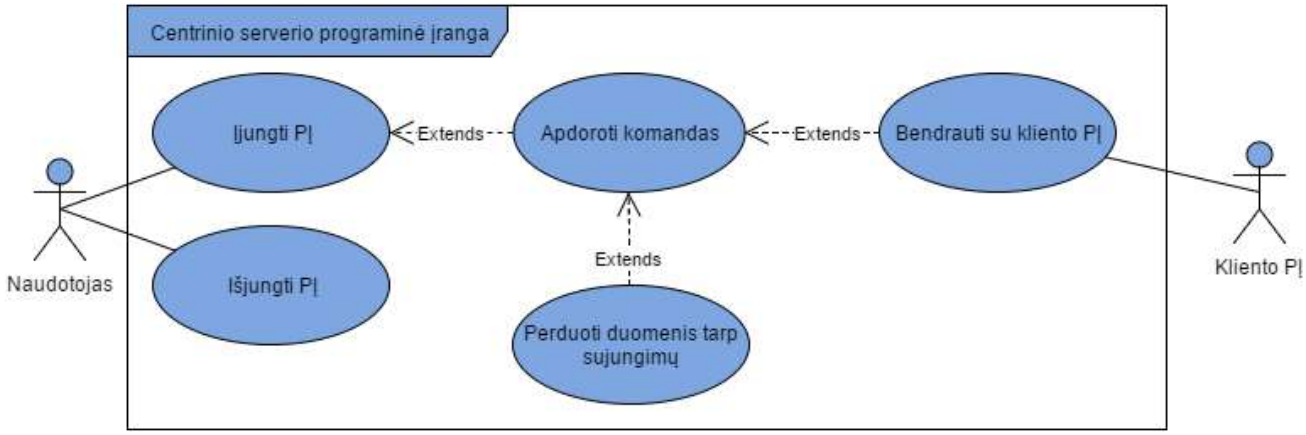
- Grafinės sąsajos programinė įranga turi keistis informacija JSON duomenų formatu
- Formuojamais sujungimais tarp klientų turi būti persiunčiami tik TCP protokolo duomenys

3.3 Panaudos atvejų diagrama

Šiame skyriuje pateikiama kuriamos sitemos panaudos atveju diagramos. Panaudos atvejų diagramos pateikiamos atskirai kiekvienai sistemos daliai ir trumpai aprašomos.

3.3.1 Centrinio serverio programinės įrangos panaudos atvejai

Šiame skyriuje pateikiama centrinio serverio panaudos atveju diagrama, žiūrėti 25 pav.



25 pav. Centrinio serverio programinės įrangos panaudos atvejai

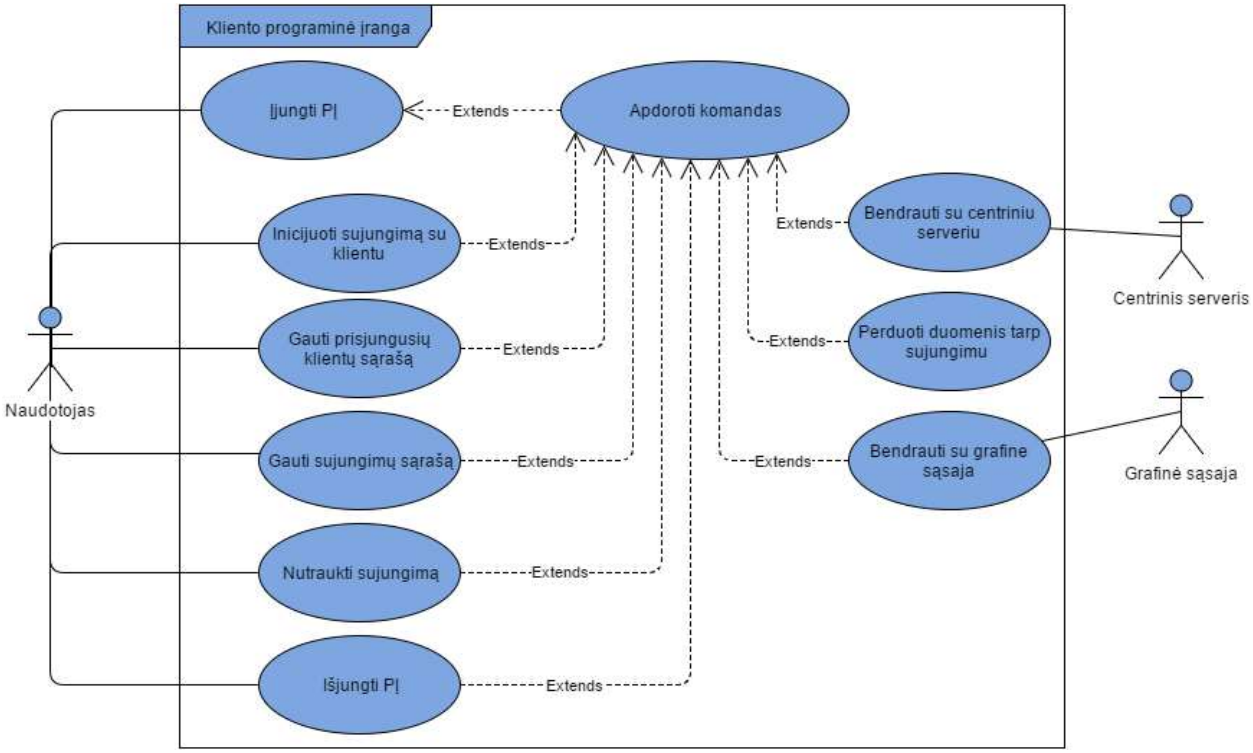
Iš centrinio serverio panaudos atvejų diagramos, žiūrėti 25 paveikslą., matome, kad naudotojas gali tik įjungti ir išjungti programinę įrangą. Tai yra dėl to, kad tolimesnis darbas ir manipuliacija programine įranga yra vykdoma kliento programinės įrangos pagalba. Taip pat centrinio serverio programinė įranga turi šiuos panaudos atvejus, kurių nepasiekia naudotojas:

- Apdoroti komandas
- Perduoti duomenis tarp sujungimų
- Bendrauti su kliento PĮ

Panaudos atvejis „apdoroti komandas“ skirtas apdoroti gaunamas komandas iš kliento PĮ. Tai gali būti prisijungusių klientų sąrašo gražinimas, sujungimo tarp klietų iniciavimas, inicijuoto sujungimo nutraukimas ir kitos. Sekantis panaudos atvejis „bendrauti su kliento PĮ“ skirtas darbui su kliento PĮ, tai yra, apdoroti prijungimą, duomenų gavimą, duomenų siuntimą ir atsijungimo veiksmus. Panaudos atvejis „perduoti duomenis tarp sujungimų“ skirtas pažymėti, kad duomenys iš vieno kliento nurodyto sujungimo yra perduodami į atitinkamo kliento atitinkamą sujungimą taip, kad jie nepapultų į neteisingo kliento neteisingą sujungimą.

3.3.2 Kliento programinės įrangos panaudos atvejai

Šiame skyriuje pateikiama kliento programinės įrangos panaudos atvejai. Jie pavaizduoti 26 paveiksle.



26 pav. Kliento programinės įrangos panaudos atvejai

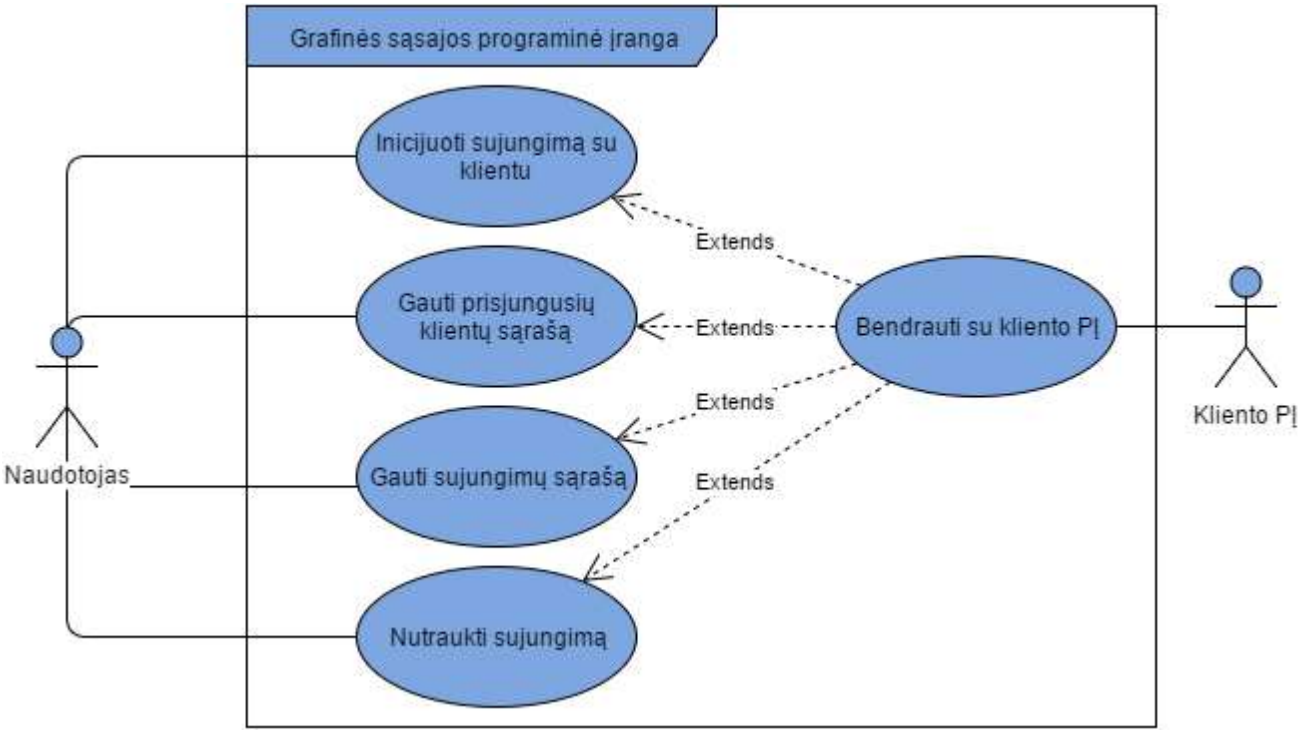
Kliento programinė įranga turi šiuos panaudos atvejus (žiūrėti 26 paveikslą):

- Įjungti PĮ
- Apdoroti komandas
- Inicijuoti sujungimą su klientu
- Gauti prijungusių klientų sąrašą
- Gauti sujungimų sąrašą
- Nutraukti sujungimą
- Išjungti PĮ
- Bendrauti su centriniu serveriu
- Perduoti duomenis tarp sujungimų
- Bendrauti su grafine sąsaja

Programinės įrangos vartotojui yra pasiekiami tik įjungti PĮ, inicijuoti sujungimą su klientu, gauti prisijungusių klientų sąrašą, gauti sujungimų sąrašą, nutraukti sujungimą ir išjungti PĮ panaudos atvejai. Liekę panaudos atvejai- apdoroti komandas, bendrauti su centriniu serveriu, perduoti duomenis tarp sujungimų ir bendrauti su grafine sąsają yra vykdomi automatiškai, kai startuoja programinė įranga.

3.3.3 Grafinės sąsajos programinės įrangos panaudos atvejai

Šiame skyriuje pateikiami kuriamos sistemos grafinės sąsajos panaudos atvejai, 27 paveikslas.



27 pav. Grafinės sąsajos panaudos atvejai

Kuriamos sitemos grafinę sąsają sudaro šie panaudos atvejai (27 paveikslas):

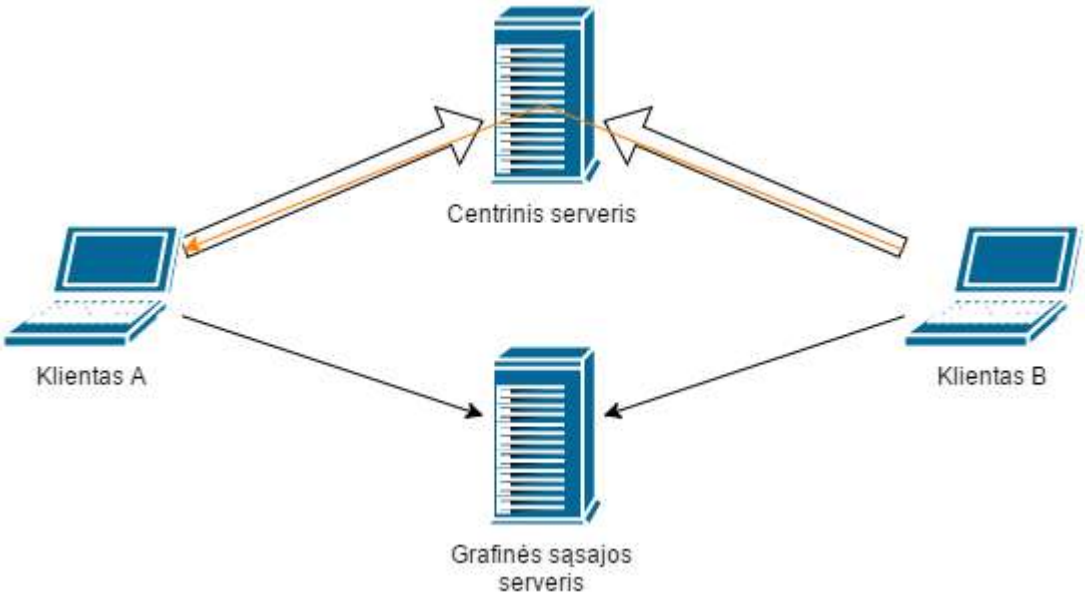
- Inicijuoti sujungimą su klientu
- Gauti prisijungusių klientų sąrašą
- Gauti sujungimų sąrašą
- Nutraukti sujungimą
- Bendrauti su kliento PĮ

Panaudos atvejis inicijuoti sujungimą su klientu skirtas užmegzti sujungimą su norimo kliento, norima programa per centrinį serverį. Panaudos atvejis gauti prisijungusių klientų sąrašą skirtas gauti visų, išskyrus save, prisijungusių klientų sąrašą iš centrinio serverio. Panaudos atvejis, gauti sujungimų sąrašą, skirtas gauti visų sujungimų, kuriuose dalyvauju sąrašą iš kliento programinės įrangos. Panaudos atvejis, nutraukti sujungimą," skirtas inicijuoti norimo sujungimo su norimu klientu nutraukimą. Visi šie keturi panaudos atvejai yra pasiekiami naudotojui ir jais galima manipuliuoti programine įranga. Likęs panaudos atvejis, bendrauti su kliento PĮ, yra sisteminis ir jo naudotojas nepasiekia. Panaudos atvejis „bendrauti su Kleino PĮ" skirtas prisijungimui ir keistis duomenimis tarp grafinės sąsajos ir kliento programinės įrangos.

3.4 Sistemos prototipo projektavimas

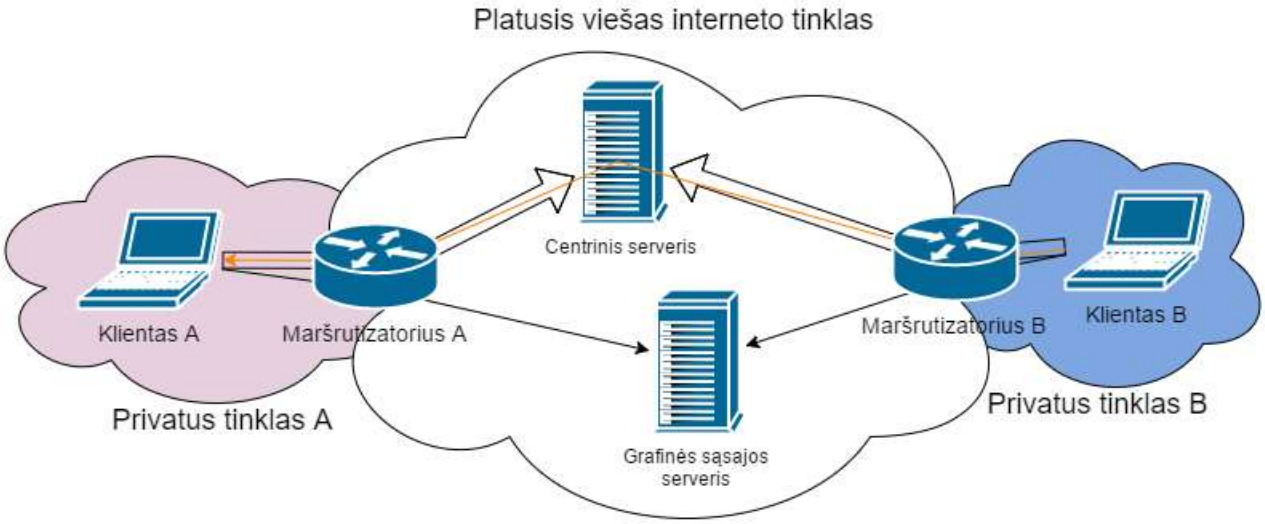
3.4.1 Išdėstymo vaizdas

Šiame skyriuje pateikiamas kuriamos sistemos išdėstymas. Loginis išdėstymas pateikiamas 28 paveiksle.



28 pav. Kuriamos sistemos loginis išdėstymas

Šis loginis išdėstymas gali būti realizuotas daugybę būdų, vienas iš jų pateikiamas 29 paveiksle.



29 pav. Kuriamos sistemos galimas išdėstymas

Pagal pateiktą vieną iš išdėstymo būdų, žiūrėti 29 paveikslą, matome, kad centrinis serveris ir grafinės sąsajos serveris yra patalpinti viešajame internete. Juos klientai A ir B, esantys privačiuose tinkluose A ir B, pasiekia per maršrutizatorius A ir B. Klientas B norėdamas perduoti duomenis klientui A praneša centriniam serveriui apie norimą jungtį iki kliento A. Centrinis serveris informuoja klientą A apie norimą sujungimą, o klientą B apie sujungimo įgyvendinimą. Tuomet kliento B

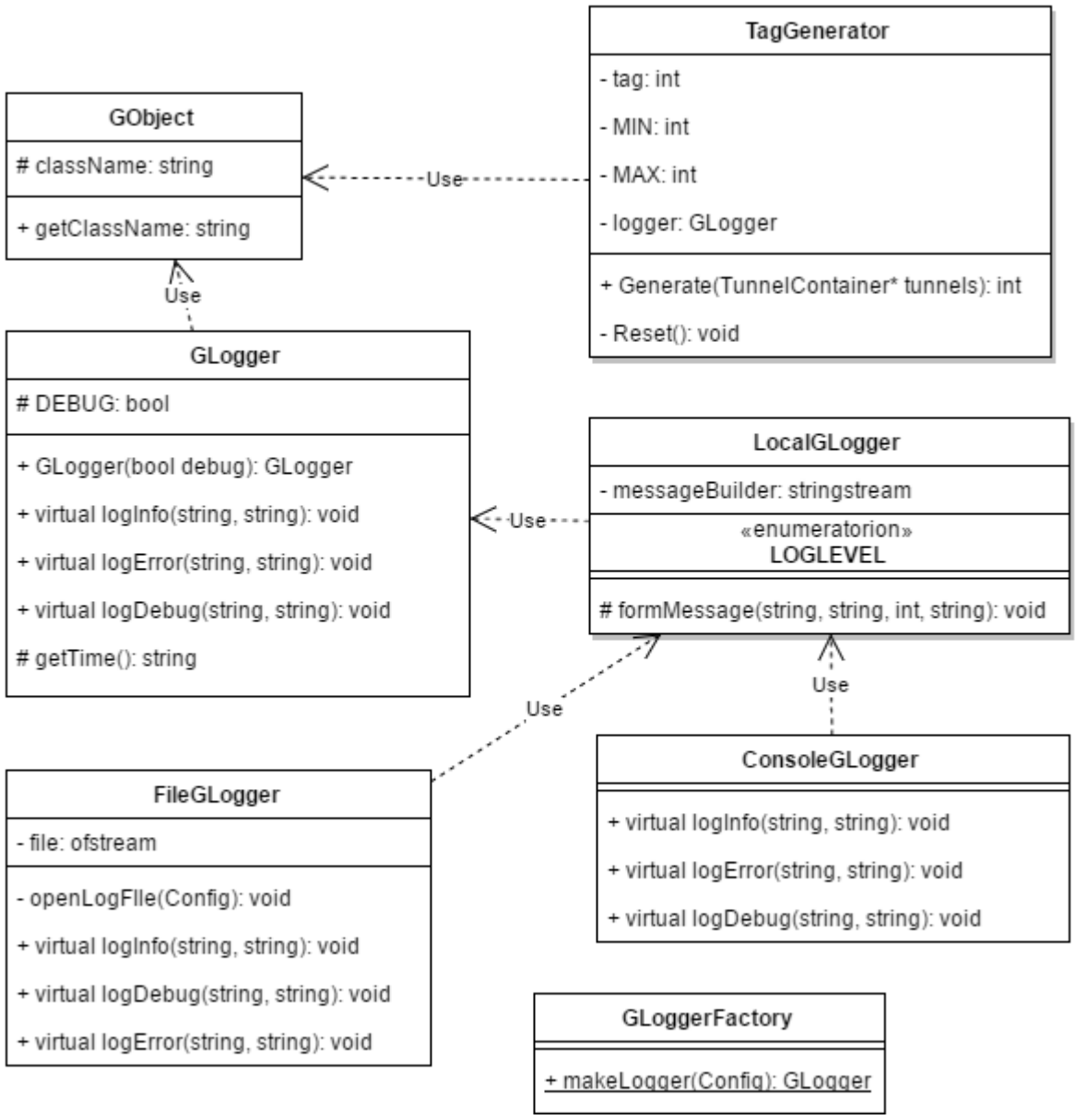
duomenys skirti klientui A bus persiųsti per centrinį serverį. Šį sujungimo procesą galima inicijuoti iš grafinės sąsajos, kurią klientai pasiekia per viešąją tinklą.

3.4.2 Sistemos statinis vaizdas

Šiame skyriuje pateikiami sistemos statiniai vaizdai. Kiekvienos sistemos dalies statiniai vaizdai bus pateikiami atskirai tolimesniuose skyriuose.

3.4.2.1 Centrinio serverio programinės įrangos statinis vaizdas

Dalis centrinio serverio programinės įrangos statinio vaizdo pavaizduota 30 paveiksle. Likusios dalys pateikiamos prieduose- 58 ir 59 paveiksluose.



30 pav. Centrinio serverio programinės įrangos statinis vaizdas, 1 dalis

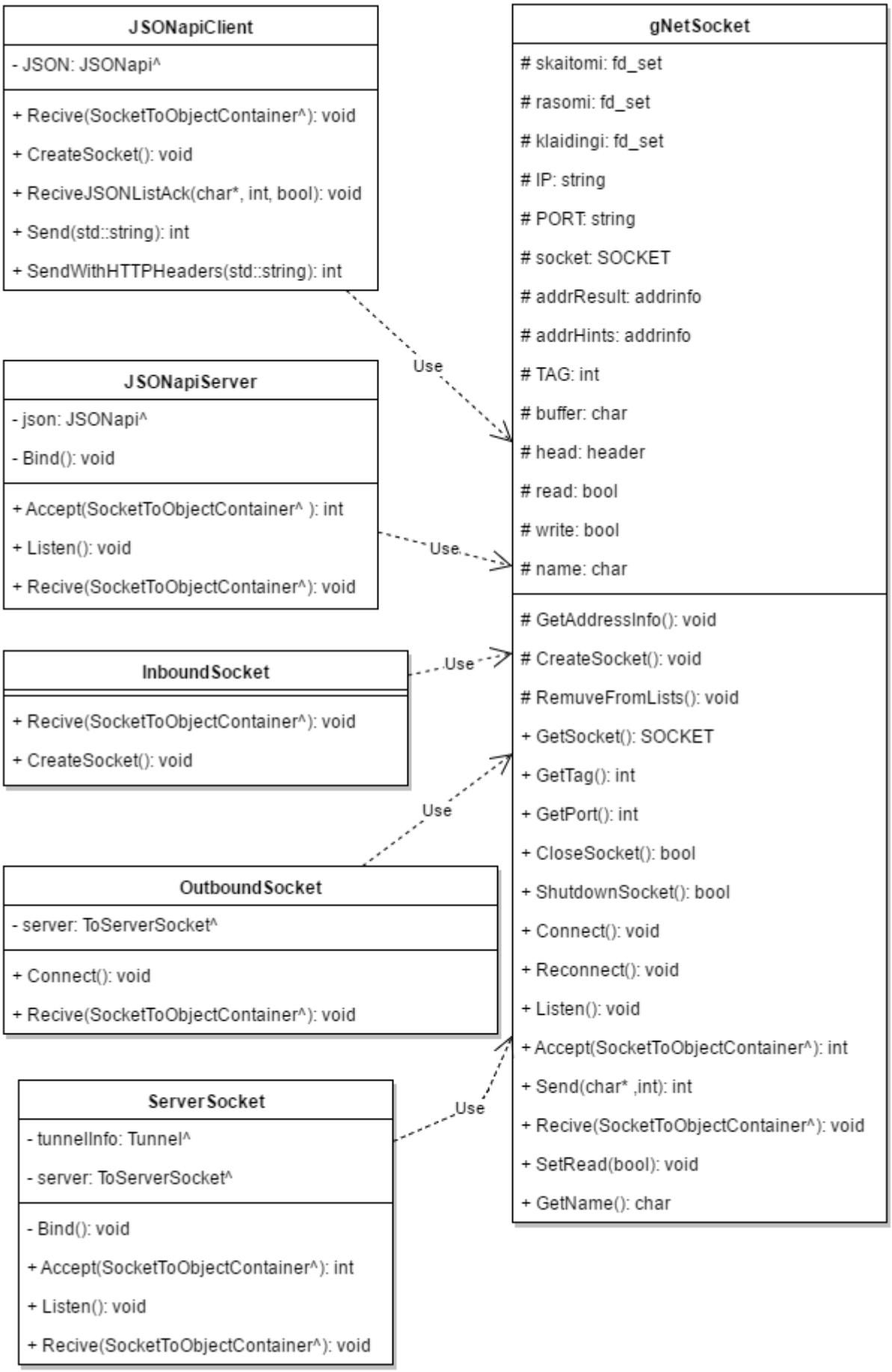
Centrinio serverio programinę įrangą sudaro šie objektai:

- **GObject**. Pagrindinis objektas, saugantis pagrindinius objektų parametrus. Šį objektą paveldi visi darbiniai objektai
- **GLogger**. Objektas aprašantis pranešimų apdorojimo objektų sąsają, bei įgyvendinantis pagrindines funkcijas
- **LocalGLogger**. Objektas naudojamas lokalių pranešimų apdorojimui
- **FileGLogger**. Objektas išvedantis pranešimų informaciją į nurodytą failą
- **ConsoleGLogger**. Objektas išvedantis pranešimus į konsolę
- **GLoggerFactory**. Objektas gražinantis pagal pateiktus parametrus pranešimų objektą
- **TagGenerator**. Objektas generuojantis unikalius sujungimų identifikavimo numerius
- **GConfig.** Objektas dirbantis su konfigūraciniu failu
- **GCommandExecution.** Objektas nuskaitantis gautas komandas, jas analizuoja ir paruošia atsakymą klientui
- **GClientContainer**. Objektas saugantis visų prisijungusių klientų sąrašą
- **GTunnelContainer**. Objektas saugantis visų užmegztų sujungimų tarp klientų duomenis
- **GSocket**. Objektas skirtas aprašyti objektų, dirbančių su klientų sujungimais sąsają, realizuoti pagrindinius metodus, bei saugoti pirminius kintamuosius
- **TCPGSocket**. Objektas skirtas realizuoti GSocket sąsają pritaikant darbui su TCP protokolu
- **TCPClientGSocket**. Objektas skirtas tiesioginiam darbui su klientu TCP protokolu
- **TCPServerGSocket**. Objektas skirtas priimti jungtis TCP protokolu
- **UDPGSocket**. Objektas skirtas realizuoti GSocket sąsają pritaikant darbui su UDP protokolu
- **UDPClientGSocket**. Objektas skirtas tiesiogiai dirbti su klientu UDP protokolu
- **UDPServerGSocket**. Objektas skirtas priiminėti jungtis UDP protokolu
- **SCTPGSocket**. Objektas skirtas realizuoti GSocket sąsajai pritaikant darbui SCTP protokolui
- **SCTPClientGSocket**. Objektas skirtas tiesiogiai dirbti su klientu SCTP protokolu
- **SCTPServerGSocket**. Objektas skirtas priiminėti jungtis SCTP protokolu
- **Structures**. Objektas skirtas aprašyti komandų struktūras
- **exitCodes**. Objektas skirtas aprašyti išėjimo kodus

3.4.2.2 Kliento programinės įrangos statinis vaizdas

Dalis kliento programinės įrangos statinio vaizdo pavaizduota 31 paveiksle. Likusios dalys pavaizduotos prieduose 60 ir 61 paveiksluose. Šią programinę įrangą sudaro šie objektai:

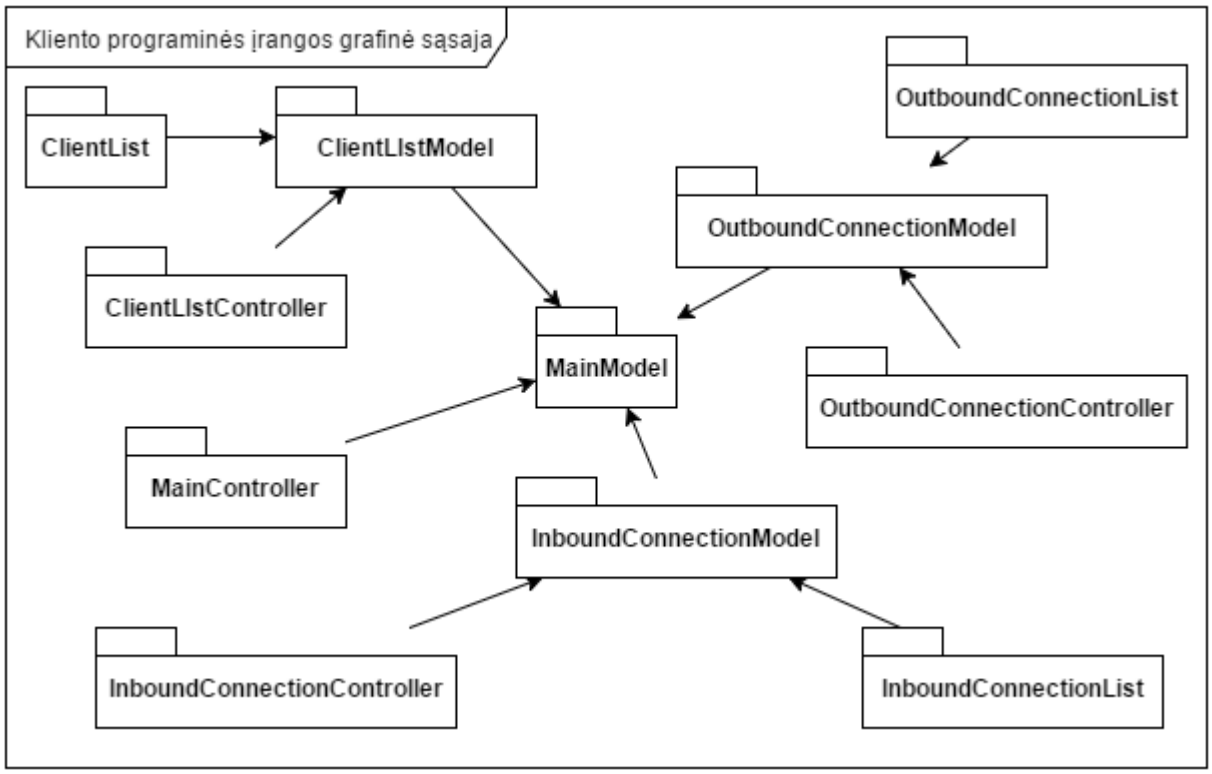
- **gNetSocket**. Objektas aprašantis sujungimų sąsają ir realizuojantis pagrindines sujungimų funkcijas
- **JSONapiClient**. Objektas skirtas bendrauti su JSON užklausų jungtimi. Naudojamas bendravimui tarp grafinės sąsajos
- **JSONapiServer**. Objektas skirtas priimti grafinės sąsajos jungtis
- **InboundSocket**. Objektas skirtas dirbti su užmezgamų sujungimų įeinamomis jungtimis
- **OutboundSocket**. Objektas skirtas dirbti su užmezgamų sujungimų išeinančiomis jungtimis
- **ServerSocket**. Obejktas skirtas priimti užmezgamų sujungimų jungtis
- **ToServerSocket**. Objektas skirtas aprašyti sąsają bendravimui su centriniu serveriu ir įgyvendina pagrindines funkcijas
- **SCTPToServerSocket**. Objektas skirtas dirbti su centriniu serveriu SCTP protokolu
- **TCPToServerSocket**. Objektas skirtas dirbti su centriniu serveriu TCP protokolu
- **UDPToServerSocket**. Objektas skirtas dirbti su centriniu serveriu UDP protokolu
- **JSONapi**. Objektas skirtas bendravimui tarp grafinės sąsajos JSON formatu
- **TunnelContainer**. Objektas skirtas saugoti užmegztų sujungimų informacijai
- **SocketToObjectContainer**. Objektas skirtas saugoti jungties objektą pagal indentifikatorių
- **Structures**. Objektas skirtas aprašyti komandų struktūras
- **SettingsReader**. Obejktas skirtas dirbti su operacinės sistemos registrais
- **TagGenerator**. Objektas skirtas generuoti unikalų sujungimo identifikatorių



31 pav. Kliento programinės įrangos statinis vaizdas. 1 dalis

3.4.2.3 Grafinės sąsajos programinės įrangos statinis vaizdas

Kliento programinės įrangos grafinės sąsajos statinis vaizdas pateikiamas 32 paveiksle.



32pav. Kliento programinės įrangos grafinės sąsajos statinis vaizdas

32 paveiksle pavaizduoti grafinės sąsajos statinis vaizdas. Jį sudaro šie komponentai:

- **MainModel**. Objektas atsakingas už grafinės sąsajos atvaizdavimą
- **MainController**. Objektas atsakingas už sąsajos funkcionalumą
- **ClientList**. Objektas talpinantis klientų sąrašo duomenis
- **ClientListModel**. Objektas atsakingas už klientų sąrašo atvaizdavimą
- **ClientListController**. Objektas atsakingas už klientų sąrašo funkcionalumą
- **OutboundConnectionList**. Objektas atsakingas už išeinančių jungčių duomenų talpinimą
- **OutboundConnectionController**. Objektas atsakingas už išeinančių jungčių sąrašo funkcionalumą
- **OutboundConnectionModel**. Objektas atsakingas už išeinančių jungčių sąrašo atvaizdavimą
- **InboundConnectionList**. Objektas atsakingas už įeinančių jungčių duomenų talpinimą
- **InbuondConnectionController**. Objektas skirtas už įeinančių jungčių sąrašo funkcionalumą

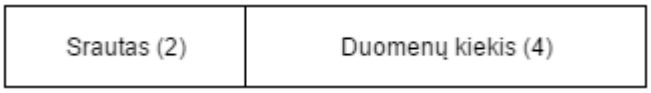
- **InboundConnectionModel**. Objektas atsakingas už įeinančių jungčių sąrašo atvaizdavimą

3.4.3 Duomenų vaizdas

Kuriamoje sistemoje nėra saugomi duomenys duomenų bazėje ar įrašomi į failus. Ši sistema dirba komandų apsikeitimo principu. Toliau šiame skyriuje pateikiamos komandų duomenų struktūros.

3.4.3.1 Antraštė

Kiekvienam duomenų rinkiniui, perduodamam per sistemą yra uždedama antraštė. Jos struktūra pavaizduota 33 paveiksle.

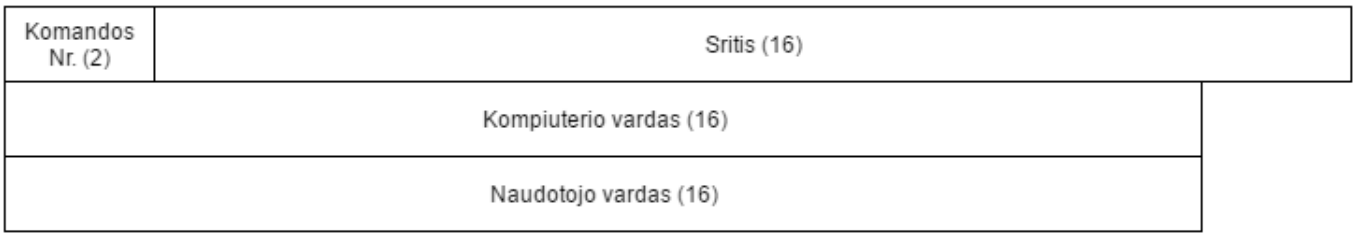


33 pav. Duomenų siunčiamų per sistemą antraštės struktūra

Duomenų antraštę , 33 paveikslas, sudaro du laukai- srautas ir duomenų kiekis. Laukas srutas skirtas nurodyti kuriam srautui perduoti duomenis. Šio lauko dydis – 2 baitai. Antrasis laukas- duomenų kiekis. Jis nurodo kiek duomenų yra perduodama. Šio lauko dydis- 4 baitai. Antraštė sudaro 6 baitus.

3.4.3.2 Komanda HELLO

Komanda HELLO skirta prisistatyti serveriui pateikiant kliento srities pavadinimą, kompiuterio pavadinimą ir vartotojo vardą. Šios komandos struktūra pavaizduota 34 paveiksle.



34 pav. Komandos HELLO struktūra

34 paveiksle pavaizduotą komandos HELLO paketas sudarytas iš 4 laukų, nurodant jų dydį skliaustuose, baitais:

- Komandos nr. Laukas nusakantis, kokia komanda atėjo
- Sritis. Kompiuterio srities pavadinimas (darbo grupės arba įmonės srities (actine directory doamin).
- Kompiuterio vardas. Siunčiančio šį paketą kompiuterio vardas.
- Naudotojo vardas. Naudotojo vardas, kuriuo paleista kliento programinė įranga.

Komandos paketo dydis- 50 baitų. Komandos paketas su antrašte sudaro 56 baitus.

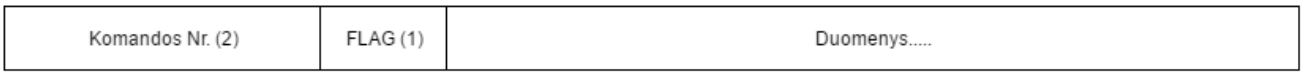
3.4.3.3 Komanda LIST

Komanda LIST skirta paprašyti prie serverio prijungusių klientų sąrašo. Šią komandą sudaro du laukai- komandos numeris ir puslapio numeris. Laukas, komandos numeris, skirtas nurodyti, kad atėjo LIST komanda, jo dydis 2 baitai. Laukas, puslapis, skirtas nurodyti kurio puslapio pageidaujame, dydis- 4 baitai. Sistemoje klientų sąrašas yra puslapiuojamas po 20 klientų, kad nesiųsti viso klientų sąrašo iš karto. Šios komandos struktūra pavaizduota 35 paveiksle.

35 pav. Komandos LIST struktūra

3.4.3.4 Komanda LIST_ACK

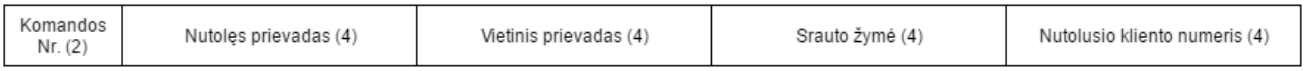
Komanda LIST_ACK skirta gražinti klientui prisijungusių klientų sąrašą. Šią komandą sudaro du butini laukai- Komandos numeris, 2 baitų dydžio, nurodantis, komandos numerį ir laukas FLAG, 1 baito dydžio, nurodantis ar sąrašas tuščias ar ne. Trečiasis laukas- duomenys, gražinamas tuomet kai yra prajungusių klientų pagal prašomą puslapio numerį. Šios komandos struktūra pateikiama 36 paveiksle.



36 pav. Komandos LIST_ACK struktūra

3.4.3.5 Komanda INIT_CONNECT

Komanda INIT_CONNECT skirta inicijuoti sujungimą su nurodytu klientu per pagrindinį serverį. Šios komandos struktūra pavaizduota 37 paveiksle.



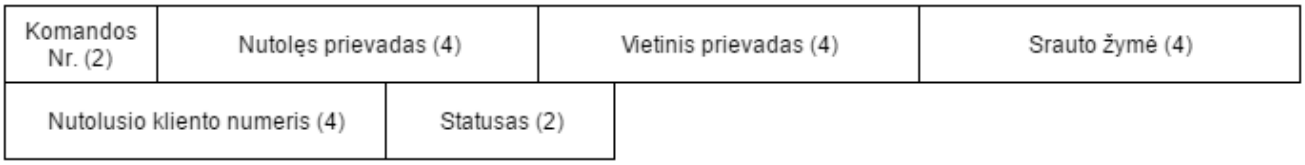
37 pav. Komandos INIT_CONNECT struktūra

Komandą INIT_CONNECT, 37 paveikslas, sudaro penki laukai:

- Komandos numeris. Naudojamas identifikuoti komandą. Dydis 2 baitai
- Nutolęs prievadas. Naudojamas nurodyti prievadą, su kuriuo norima užmegzti ryšį nutolusio kliento kompiuteryje. Dydis 4 baitai.
- Vietinis prievadas. Naudojama nurodyti vietiniame kompiuteryje atverto prievado numerį, per kurį bus pasiekiamas nutolusio kompiuterio prievadas. Dydis 4 baitai.
- Srauto žymė. Skirta nurodyti kokią srauto žymę naudoja kliento PĮ. Dydis 4 baitai.
- Nutolusio kliento numeris. Naudojama nurodyti prie kurio kliento norimą jungtis pagal identifikatorių. Dydis 4 baitai.

3.4.3.6 Komanda INIT_CONNECT_ACK

Komanda INIT_CONNECT_ACK skirtas pranešti sujungimo iniciatoriui apie sujungimo statusą. Šios komandos struktūra pavaizduota 38 paveiksle.



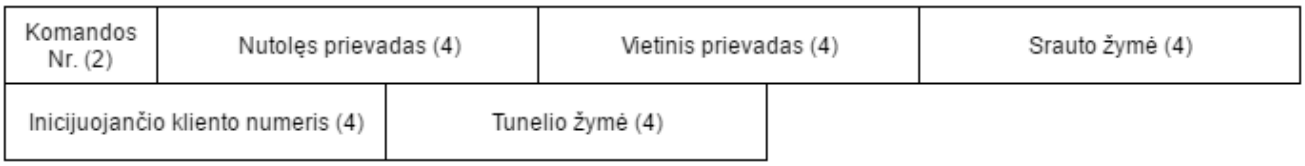
38 pav. Komandos INIT_CONNECT_ACK struktūra

Šią komandą sudaro šeši laukai:

- Komandos numeris. Naudojamas identifikuoti komandą. Dydis 2 baitai
- Nutolęs prievadas. Naudojamas nurodyti prievadą, su kuriuo norima užmegzti ryšį nutolusio kliento kompiuteryje. Dydis 4 baitai.
- Vietinis prievadas. Naudojama nurodyti vietiniame kompiuteryje atverto prievado numerį, per kurį bus pasiekiamas nutolusio kompiuterio prievadas. Dydis 4 baitai.
- Srauto žymė. Skirta nurodyti kokią srauto žymę naudoja kliento PĮ. Dydis 4 baitai.
- Nutolusio kliento numeris. Naudojama nurodyti prie kurio kliento noriu jungtis pagal identifikatorių. Dydis 4 baitai.
- Statusas. Laukas nurodantis pavyko ar nepavyko prisijungti prie nurodyto prievado nutolusio kliento kompiuteryje.. Dydis 2 baitai.

3.4.3.7 Komanda CONNECT

Komanda CONNECT naudojama inicijuoti sujungimą iš centrinio serverio, pas nutolusį klientą. Šią komandą siunčia serveris gavęs INIT_CONNECT komandą. Jos struktūra pavaizduota 39 paveiksle.



39 pav. Komandos CONNECT struktūra

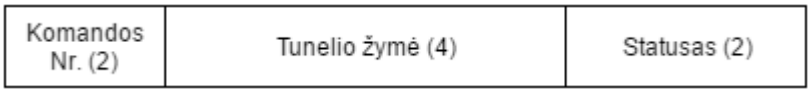
Šią komandą sudaro 6 laukai:

- Komandos numeris. Naudojamas identifikuoti komandą. Dydis 2 baitai
- Nutolęs prievadas. Naudojamas nurodyti prievadą, su kuriuo norima užmegzti ryšį nutolusio kliento kompiuteryje. Dydis 4 baitai.
- Vietinis prievadas. Naudojama nurodyti vietiniame kompiuteryje atverto prievado numerį, per kurį bus pasiekiamas nutolusio kompiuterio prievadas. Dydis 4 baitai.
- Srauto žymė. Skirta nurodyti kokią srauto žymę naudoja kliento PĮ. Dydis 4 baitai.

- Inicijuojančio kliento numeris. Naudojama nurodyti kuris klientas iniciavo sujungimą. Dydis 4 baitai.
- Tunelio žymė. Nurodantis laukas kuriam tuneliui gražinti atsakymą. Dydis 4 baitai.

3.4.3.8 Komanda CONNECT_ACK

Komanda CONNECT_ACK naudojama nusiųsti atsaką į centrinį serverį, į nurodytą tunelį, gražinant statusą ar pavyko prisijungti prie nurodyto prievado ar ne. Šios komandos struktūra pavaizduota 40 paveiksle.



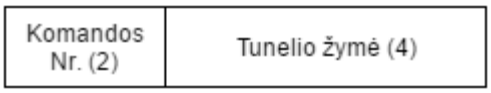
40 pav. Komandos CONNECT_ACK struktūra

Šią komandą sudaro trys laukai:

- Komandos numeris. Naudojamas identifikuoti komandą. Dydis 2 baitai
- Tunelio žymė. Laukas nurodantis kuriam tuneliui gražinamas atsakymas. Dydis 4 baitai.
- Statusas. Nurodantis apie sujungimo statusą. Dydis 2 baitai.

3.4.3.9 Komanda BEGIN_READ

Komanda BEGIN_READ naudojama įjungti skaitymą kitame tunelio gale esančioje programoje. Šis pranešimas siunčiamas iš iniciatoriaus kompiuterio į centrinį serverį. Šios komandos struktūra pavaizduota 41 paveiksle.



41 pav. Komandos BEGIN_READ struktūra

Komandą BEGIN_READ, 41 paveikslas , sudaro du laukai:

- Komandos numeris. Naudojamas identifikuoti komandą. Dydis 2 baitai
- Tunelio žymė. Laukas nurodantis kuriam tunelyje pradėti skaitymą. Dydis 2 baitai.

3.4.3.10 Komanda BEGIN_READ_ACK

Komanda BEGIN_READ_ACK naudojama perduoti BEGIN_READ komandą kitam tunelio klientui. Ši komanda siunčiama iš centrinio serverio nurodytam klientui. Šios komandos struktūra (žiūrėti **41** paveikslą) ir laukai tokie patys kaip BEGIN_READ komandos.

3.4.3.11 Komanda CLOSE_TUNNEL

Komanda CLOSE_TUNNEL siunčiama tuomet, kai kuris nors iš tunelio klientų uždaro tunelio sujungimą. Ši komanda skirta pranešti kitam tunelio klientui apie uždarymą, per centrinį serverį. Šios komandos struktūra tokia pati kaip BEGIN_READ, 41 paveikslas.

3.4.3.12 Grafinės sąsajos komandos

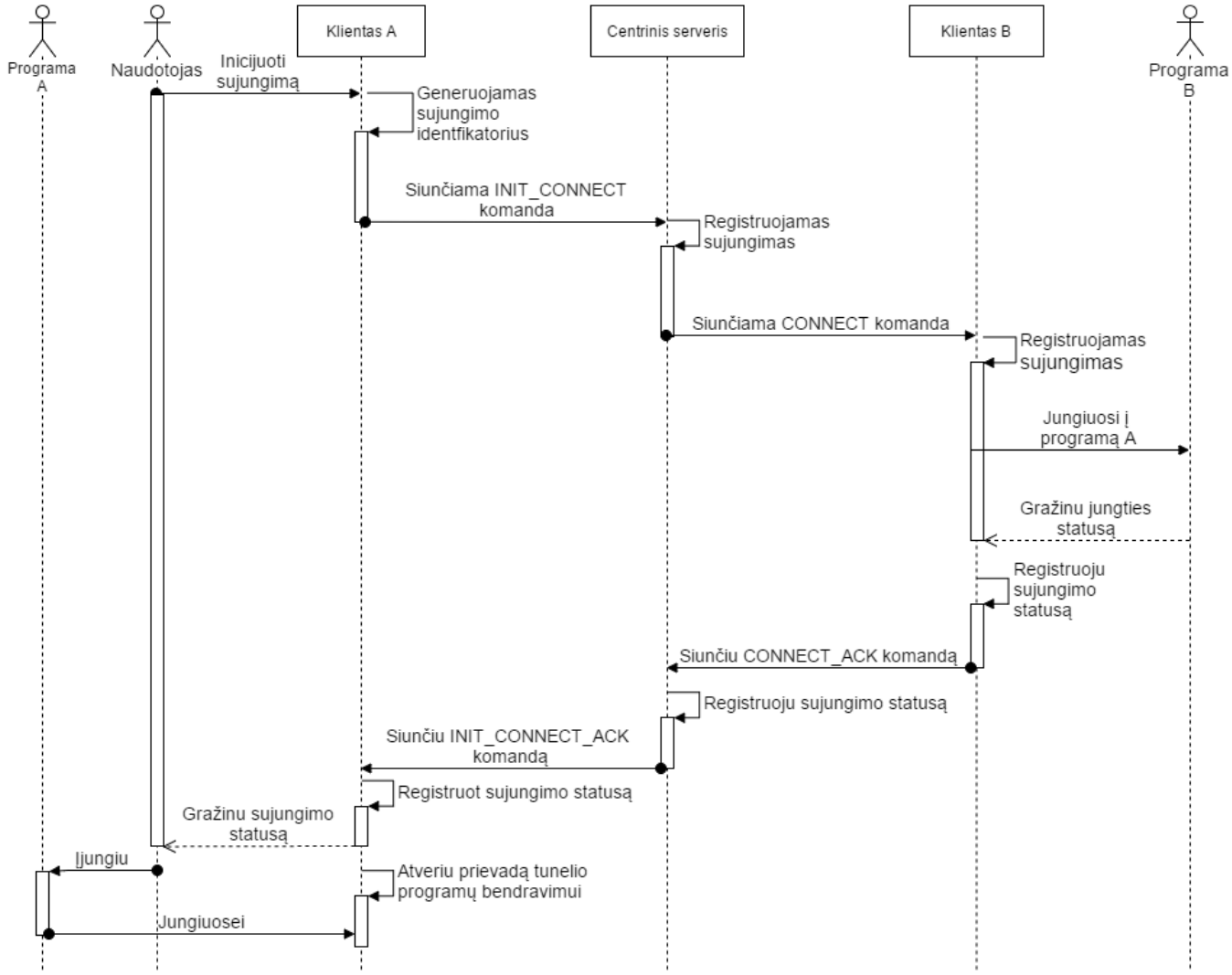
Siekiant užtikrinti grafinės sąsajos funkcionalumą yra naudojamos šios komandos:

- JSON_LIST. Tokia pati kaip LIST
- JSON_LIST_ACK. Tokia pati kaip LIST_ACK
- JSON_INIT_CONNECT. Tokia pati kaip INIT_CONNECT
- JSON_INIT_CONNECT_ACK. Tokia pati kaip INIT_CONENCT_ACK
- JSON_CONNECT. Tokia pati kaip CONNECT.
- JSON_CONNECT_ACK. Tokia pati kaip CONNECT_ACK

JSON tipo komandos skiriasi nuo paprastų, tuo, kad yra pridedamas vienas papildomas laukas socketID. Jo paskirtis nurodyti kuriam sujungimui gražinti gražinamus duomenis. Lauko dydis 4 baitai.

3.5 Veikimo principas

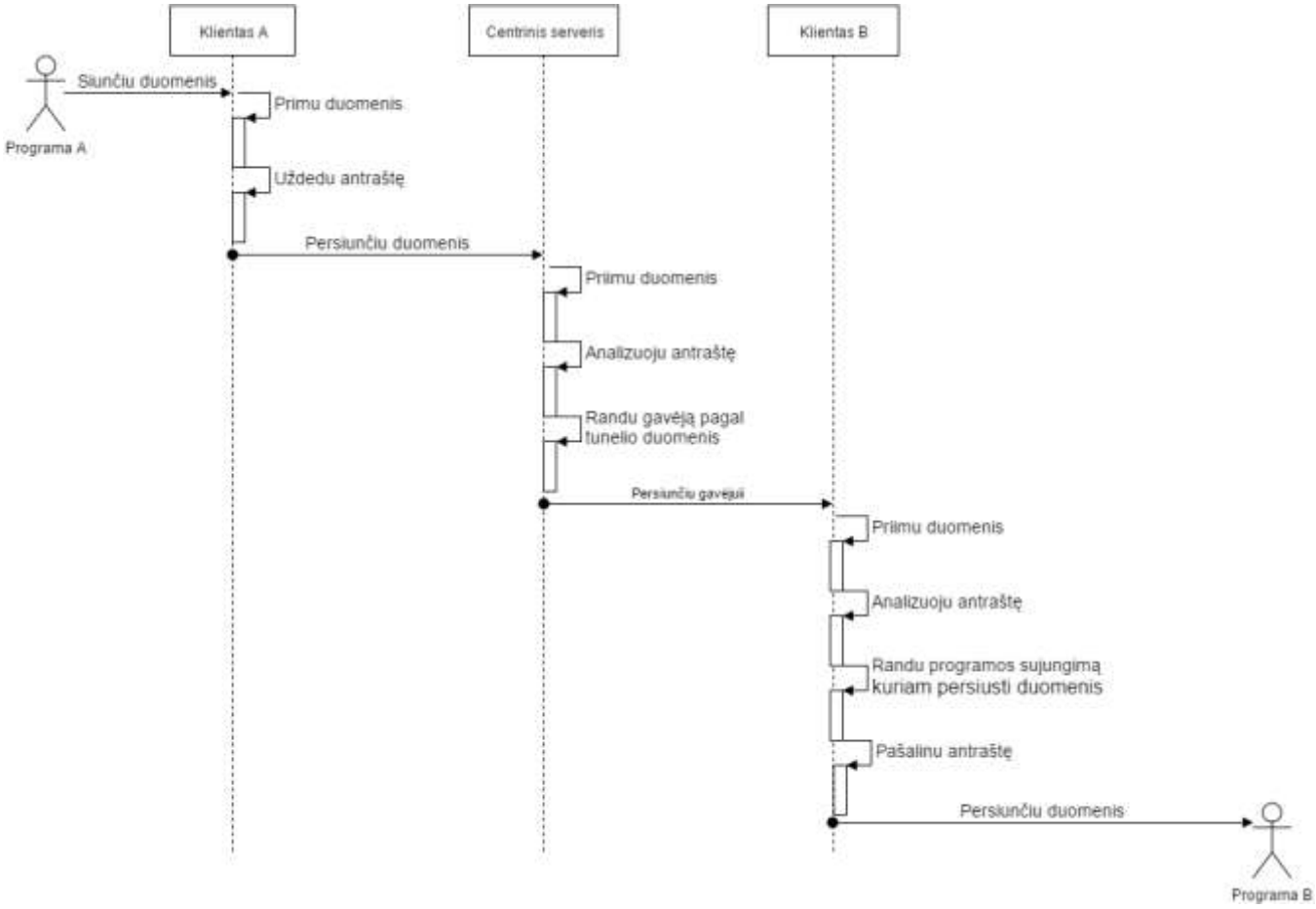
Šiame skyriuje pateikiama pagrindinių funkcijos veikimo vizualizacija. Sujungimo tarp nutolusių klientų programų ryšio užmezgimo schema pavaizduota 42 paveiksle.



42 pav. Sujungimo užmezgimo su nutolusio kliento programa veiklos diagrama

Schemoje, 42 paveikslas, pateikiama, kad naudotojas norėdamas tarp nutolusių programų A ir B užmegzti ryšį per kliento programinę įrangą A pirmiausia nurodo kliento programine įrangai inicijuoti sujungimą su kliento programine įranga B. Kliento programinė įranga A registruoja norimą sujungimą ir siunčia INIT_CONNECT sujungimo komandą centriniam serveriui. Centrinis serveris registruoja sujungimą ir informuoja kliento B programinę įrangą apie norimą sujungimą CONNECT komanda. Kliento B programinė įranga registruoja sujungimą ir bando jungtis į programą B. Sujungimo su programa B statusą registruoja ir perduoda centriniam serveriui CONNECT_ACK komanda. Centrinis serveris registruoja sujungimo statusą ir perduoda iniciatoriui, kliento A programinei įrangai, INIT_CONNECT_ACK komandos pagalba. Ji registruoja gautą statusą ir pateikia naudotojui. Naudotojas savo ruoštu jungią norimą programą A į nurodytą prievadą ir tuomet komunikaciją per sujungimą prasideda tarp programų A ir B.

Iniciavus sujungimą reikia perduoti duomenis tarp programų. Duomenų perdavimo schema pateikta 43 paveiksle.



43 pav. Duomenų perdavimo tarp klientų veiklos diagrama

Duomenų perdavimo schemoje, 43 paveikslas, duomenų transportavimo tuneliu schema. Programa A siunčia duomenis į užmegztą tunelį. Kliento A programinė įranga priima duomenis ir jiems uždeda antraštę, reikalingą persiųsti duomenis per kuriamą sistemą. Uždėjus antraštę toliau

persiunčia centriniam serveriui. Centrinis priėmęs duomenis iš kliento A programinės įrangos analizuoja antraštę ir ieško kuriam klientui ir kuriam jo tuneliui skirti duomenys. Radęs gavėją ir tunelį centrinis serveris persiunčia duomenis gavėjui- kliento B programinei įrangai. Kliento B programinė įranga priima duomenis, išanalizuoja antraštę ir ieško kuriam sujungimui reikia persiųsti duomenis. Suradus tinkamą sujungimą, programa B, juos išsiunčia.

3.6 Sistemos saugumas

Realizuojama prototipinė programinė įranga gali neturėti naudotojo autorizavimo bei duomenų šifravimo mechanizmų.

4 EKSPERIMENTINĖ DALIS

Eksperimentu siekiama ištirti sukurtos programinės įrangos duomenų perdavimo pralaidumą, vėlinimą ir naudingų duomenų kiekį perduodant duomenis skirtingais protokolais.

4.1 Tiriamosios priemonės

Siekiant ištirti sukurtos programinės įrangos duomenų perdavimo pralaidumą ir vėlinimą buvo sukurta programinė įranga,kurią sudaro dvi dalys- klientas, toliau siuntimo klientas, ir serveris, toliau sintimo serveris. Siuntimo klientas skirtas prisijungti prie siuntimo serverio ir siųsti nurodytą duomenų kiekį, nurodytą laiką. Į siunčiamų duomenų kiekį yra įrašoma laiko žymė, kad siuntimo serveris žinotų kada buvo išsiųsti duomenys. Toks duomenų siuntimas leidžia išmatuoti kanalo tarp siuntimo kliento ir siuntimo serverio didžiausią pralaidumą ir vėlinimą. Siuntimo serveris skirtas priimti jungtis iš siuntimo kliento ir skaičiuoti kiek užtruko duomenų persiuntimas , taip pat nustatyti duomenų perdavimo momentinį, vidutinį, didžiausią ir mažiausią greičius, bei kiek duomenų buvo persiųsta.

Tiriant naudingą duomenų kiekio perdavimą bus naudojama tinklo srautų analizavimo programinė įranga wireshark. Jos pagalba bus peržiūrima kiek duomenų buvo persiųsta ir kiek iš siųstų duomenų yra naudingos informacijos. Šis tyrimas bus atliekamas atskirai nuo anksčiau minėto, kad nebūtų įtakojama duomenų perdavimo sparta.

Tyrime bus naudojama įvairi kompiuterinė įranga. Jos žymėjimai ir parametrai pateikiami 3 lentelėje.

3 lentelė Tyrimo metu naudojamos kompiuterinė technika

Simbolis	Pavadinimas	Aprašas
	Kompiuteris A	Procesorius: Intel Xeon-0 Operatyvios atmintis kiekis: 4GB Tinklo jungtis: RealTek Semiconductor RTL8168/8111 PCI-E Gigabit Ethernet NIC Maksimalus tinklo jungties greitis: 1000Mbps Operacinė sistema: Windows Server 2012 R2
	Kompiuteris B	Procesorius: Intel Xeon-0 Operatyvios atmintis kiekis: 4GB Tinklo jungtis: Atheros AR8131 PCI-E Gigabit Ethernet Controller Maksimalus tinklo jungties greitis: 1000Mbps Operacinė sistema: Windows Server 2012 R2
	Centrinis serveris A	Procesorius: Intel(R) Xeon(R) CPU E5-2620 Operatyvios atmintis kiekis: 1GB Tinklo jungtis: Intel 82540EM Gigabit Ethernet Controller Maksimalus tinklo jungties greitis: 1000Mbps Operacinė sistema: Debian 8.4
	Komutatorius A	Modelis: HP ProCurve V1910-48G Jungčių skaičius: 48 Maksimalus tinklo jungties greitis 1000Mbps
	Kompiuteris C	Procesorius: Intel Xeon-0 Operatyvios atmintis kiekis: 4GB Tinklo jungtis: RealTek Semiconductor RTL8168/8111 PCI-E Gigabit Ethernet NIC Maksimalus tinklo jungties greitis: 1000Mbps Operacinė sistema: Windows Server 2012 R2
	Maršrutizatorius A	Procesorius: Pentium(R) Dual-Core CPU E5700 Operatyvios atmintis kiekis: 2GB Tinklo jungtis: Intel 82574L Gigabit Network Connection Maksimalus tinklo jungties greitis 1000Mbps Operacinė sistema: Debian 6
	Maršrutizatorius B	FortiGate D60 Maksimalus tinklo jungties greitis 1000Mbps
	Centrinis serveris B	Procesorius: Intel(R) Xeon(R) CPU E5-2630 Operatyvios atmintis kiekis: 2GB Tinklo jungtis: Intel 82540EM Gigabit Ethernet Controller Maksimalus tinklo jungties greitis: 1000Mbps Operacinė sistema: Debian 8.4

4.2 Duomenų pralaidumo ir vėlinimo tyrimas

Šio tyrimo metu siekiama išsiaiškinti sukurtos programinės įrangos užmezgamo sujungimo tarp klientų per centrinį serverį duomenų pralaidumą ir vėlinimą. Tyrimai bus atliekami naudojant skirtingus duomenų perdavimo protokolus tarp kliento ir centrinio serverio, bei tai atliekant skirtingose tinklo konfigūracijose. Tyrimų metu bus naudojama siuntimo kliento ir serverio programinė įranga, kuri bus paleista vieną minutę ir matuos perduodamų duomenų srautus ir vėlinimus.

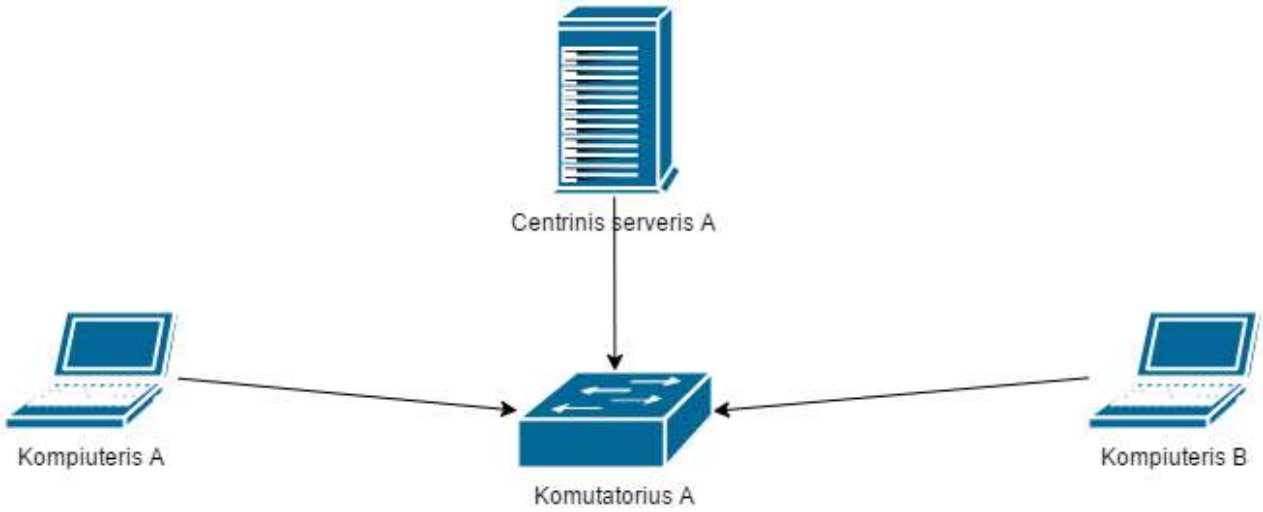
4.2.1 Tyrimo eiga

Siekiant nustatyti tikrąjį pralaidumą ir vėlinimą bus atliekami matavimai nesinaudojant sukurta sistema. Matavimai bus atliekami skirtingose tinklo konfigūracijose- vietinis ir platus viešojo interneto tinklų.

Vietinio tinklo tyrime pirmiausia bus nustatoma maksimalus kanalo pralaidumas ir vėlinimas tarp kompiuterių A ir B sujungtų per komutatorių A, naudojant siuntimo kliento ir serverio pagalba. Šio tyrimo schema pavaizduota 44 paveiksle.

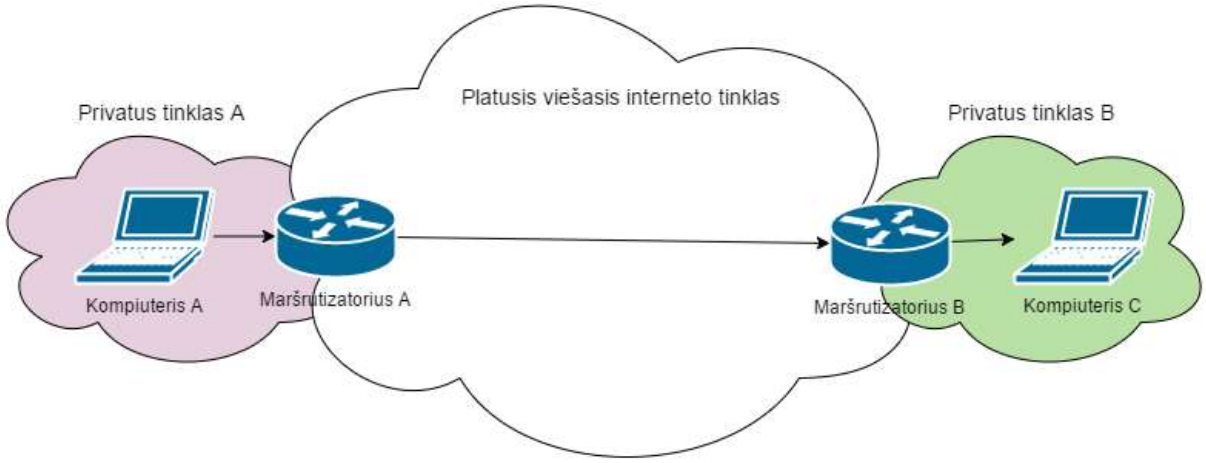
44 pav. Vietinio tinklo (LAN) tyrimas be centrinio serverio

Ištyrus maksimalų sujungimo pralaidumą ir vėlinimą tarp kompiuterių A ir B toliau tirsime duomenų pralaidumą ir vėlinimą sukurtos programos. Šio tyrimo schema pavaizduota 45 paveiksle. Šio tyrimo metu siuntimo klientas ir serveris bus sujungti per sukurtą sistemą. Tyrimo metu matuosime koks gaunamas vėlinimas vietinio tinklo konfigūracijoje.



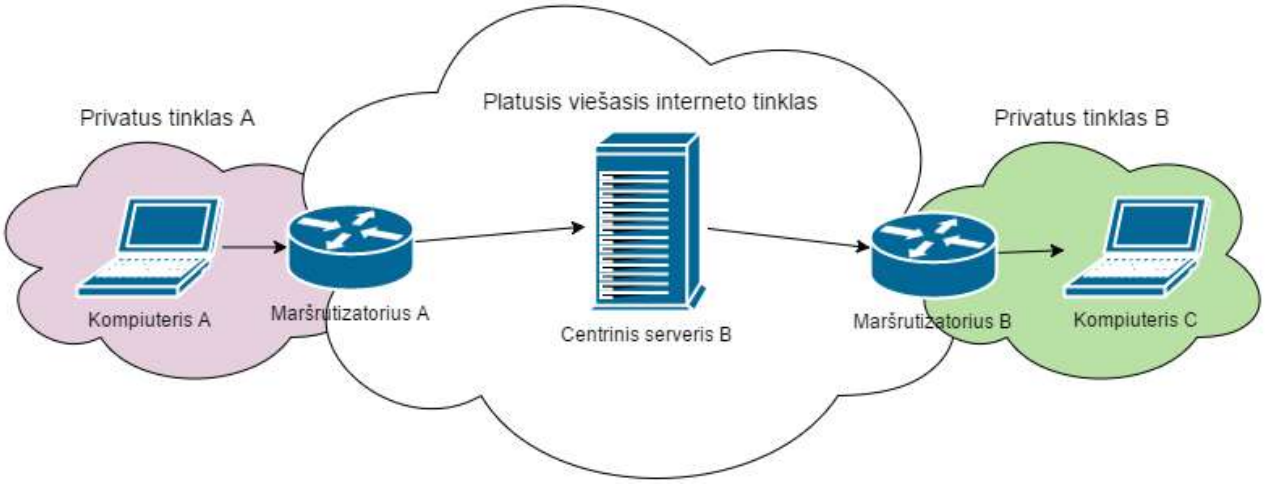
45 pav. Vietinio tinklo (LAN) su gNET programine įranga tyrimo sujungimo schema

Tiriant pralaidumą ir vėlinimą plataus tinklo konfigūracijoje bus naudojami kompiuteriai A ir C ir centrinis serveris B. Šio tyrimo schema apvaizduota 46 paveiksle.



46 pav. Plataus tinklo (WAN) tyrumo sujungimo schema

Plataus tinklo tyrime bus matuojamas pralaidumas ir vėlinimas jungiantis iš kompiuterio A į kompiuterį C per maršrutizatorius A ir B. Maršrutizatoriuje B bus atvertas reikalingas prievadas siekiant suteikti prieigą kompiuteriui A prie kompiuterio C. Atlikus ši tyrimą bus atliekamas plačiojo tinklo tyrimas naudojant kurtą programinę įrangą. Šio tyrimo schema pavaizduota 47 paveiksle.



47 pav. Plataus tinklo (WAN) su gNet programine įranga tyrumo sujungimo schema

Šiam tyrimui (schema pavaizduota 47 paveiksle) bus naudojamas centrinis serveris C, kuris yra plačiajame internete. Kompiuteris A jungsis prie centrinio serverio B per maršrutizatorių A. Kompiuteris C jungsis prie centrinio serverio B per maršrutizatorių B. Prisijungus kompiuteriams bus inicijuojamas sujungimas tarp kompiuterių A ir C. Bus matuojamas šio sujungimo duomenų pralaidumas ir vėlinimas.

4.2.2 SCTP protokolas

Kliento programinėje įrangoje SCTP protokolas nebuvo įgyvendintas. Prototipo kurimo metu buvo tik viena funkcionuojanti SCTP tvarkyklė (16), tačiau jos realizacijoje buvo klaidų. Kviečiant select funkciją operacinėje sistemoje kildavo kritinė klaida, kuri priversdavo kompiuterį persikrauti. Taip pat buvo bandyta pritaikyti naudotojo srities bibliotekas (17), kurios dirba su SCTP protokolu. Šių bibliotekų nepavyko pritaikyti kuriamai sistemai, nes jos neleido tiesiogiai dirbti su jungtimis. Vienas iš naudotojo srities bibliotekos kūrėjų Michael Tüxen pateikė atsakymą, kodėl SCTP protokolas nėra palaikomas Windows operacinės sistemos kūrėjų.

„... However, the Windows kernel extensions is not maintained anymore. It just needs someone with Windows expertise to do the work.

SCTP is commercially used in telephony signalling networks, especially mobile networks, and therefore the above selection of operating systems makes sense. As far as I know, Microsoft is not active in that area. Recently, SCTP is also used as a transport protocol in WebRTC. For integrating it into Web browsers and running on top of DTLS, you need a userland implementation. Therefore, incentives for Microsoft to integrate SCTP into their operating system might be little.“ Michael Tüxen, Münster University of Applied Sciences, 2016 05 05.

Iš atliktų bandymų ir gautos informacijos nustatyta, kad SCTP protokolo palaikymas operacinės sistemos branduolio lygmenyje yra nepatenkinamas. Dėl to nebuvo pritaikytas kuriamoje sistemoje ir tyrimai su juo nebuvo atliekami.

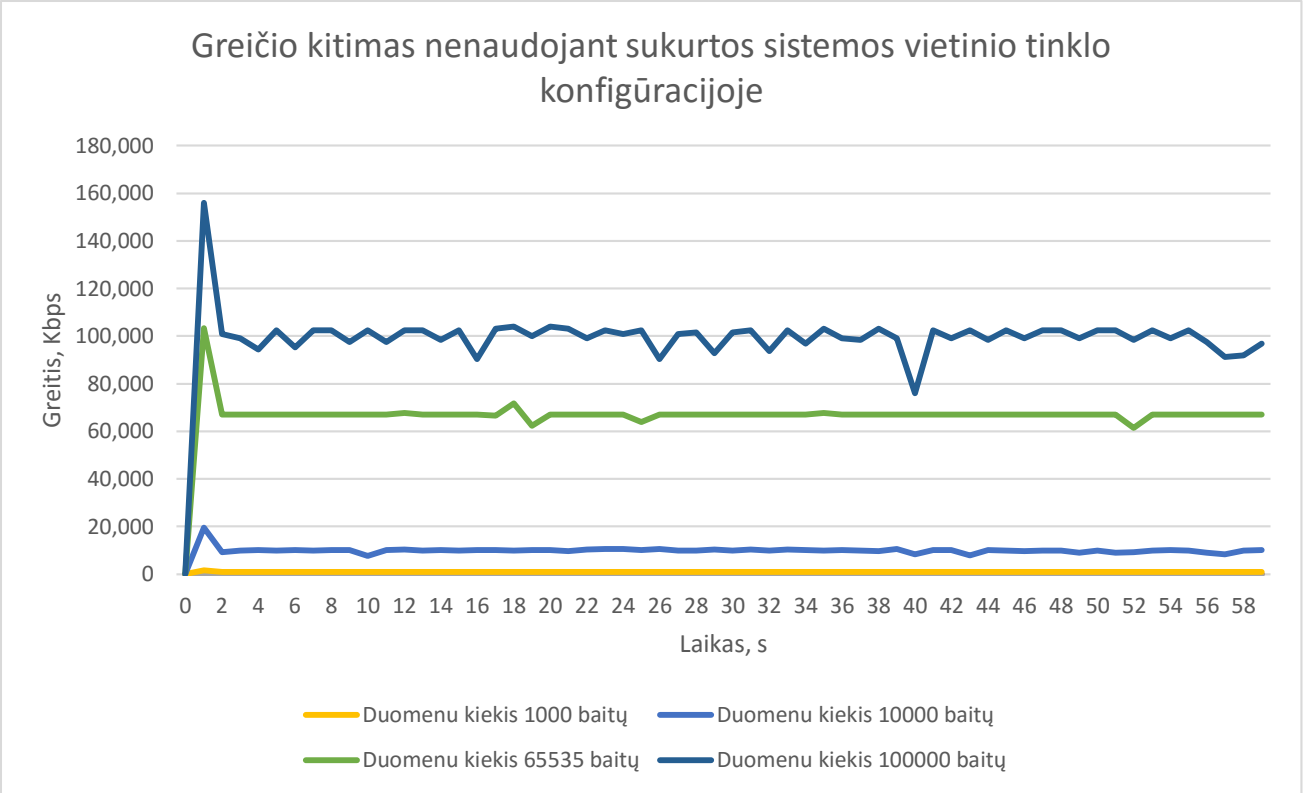
4.2.3 Vietinio tinklo tyrimas

Pralaidumui ir vėlinimui nustatyti šioje tinklo konfigūracijoje (44 paveikslas) naudojama siuntimo kliento ir serverio programinė įranga, perduodanti duomenis TCP protokolu. Vietinio tinklo matavimų rezultatai pateikiami 4 lentelėje. Siuntimo greičio kitimo grafikas pateikiamas 48 paveiksle, o tikslūs greičiai pateikiamai 15 lentelėje.

4 lentelė Vietinio tinklo tyrimo rezultatai

Siunčiamų duomenų dydis, baitais	1	10	100	1000	10000	65535	100000
Persiųstas duomenų kiekis, Kb	60,808	594,32	5888,8	59648	592720	3987673,68	5920800
Paketų skaičius, vnt	7600	7428	7360	7455	7408	7605	7400`
Vidutinis vėlinimas, ms	0,05	0,165	0,05	0,04	0,21	0,23	0,25
Vidutinis greitis, Kb/s	1,01	10,18	97,12	1 002,54	9 811,67	67 085,31	95 232,52

Tyrimo metu pastebėtos logaritminės priklausomybė tarp duomenų segmento dydžio ir vidutinio greičio, bei tarp duomenų segmento dydžio ir persiųstų duomenų kiekio. Keičiant duomenų segmento dydį tyrimo metu išsiųstų paketų skaičius nežymiai svyravo- nuo 7360 iki 7605 vienetų. Taip pat užfiksuota, kad duomenims išaugus 100000 kartų vėlinimas pakito tik 0,5 milisekundėmis.



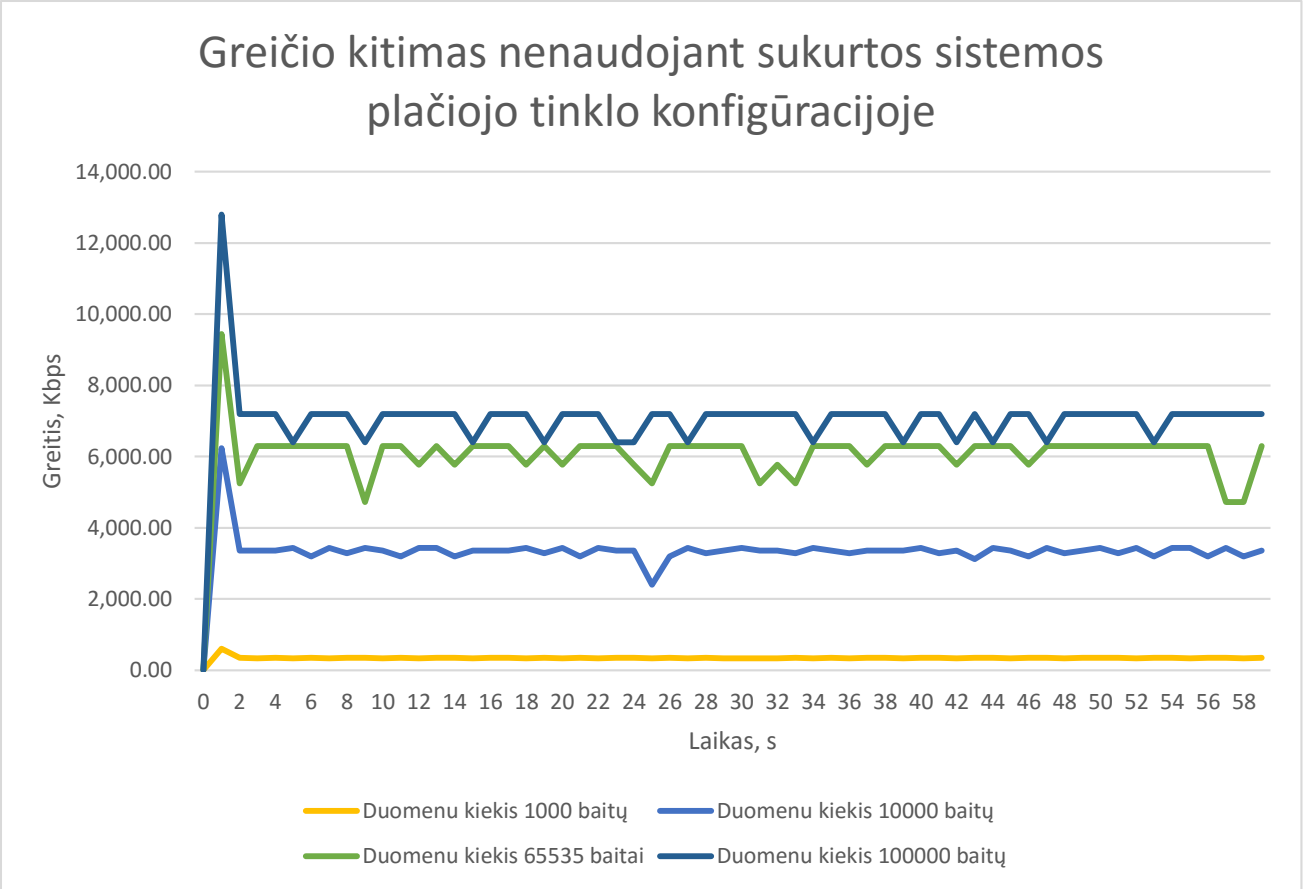
48 pav. Greičio kitimas nenaudojant sukurtos sistemos vietinio tinklo konfigūracijoje

Tyrimo metu (48 paveikslas) tiriant momentinį greitį buvo nustatyta, kad duomenų perdavimo greitis vietiniu tinklu svyruoja kas atitinkamą duomenų kiekį. Šį reiškinį turėtų įtakoti TCP protokolo atsakymų siuntimas apie gautą duomenų kiekį. Duomenų segmentą padidinus daugiau nei TCP protokolas vienu paketu gali perduoti greičio svyravimai tampa ryškesni.

Šio tyrimo pagalba buvo nustatyti pradiniai vietinio tinklo parametrai. Jų pagalba bus galima apskaičiuoti nuostolius naudojantis sukurta programine įranga.

4.2.4 Plačiojo internato tinklo tyrimas

Pralaidumui ir vėlinimui nustatyti plačiojo tinklo konfigūracijoje (46 paveikslas) taip pat naudojama siuntimo kliento ir serverio programinė įranga. Plačiojo tinklo matavimų rezultatai pateikiamai 49 paveiksle ir 5 lentelėje.



49 pav. Greičio kitimas nenaudojant sukurtos sistemos plačiojo tinklo konfigūracijoje

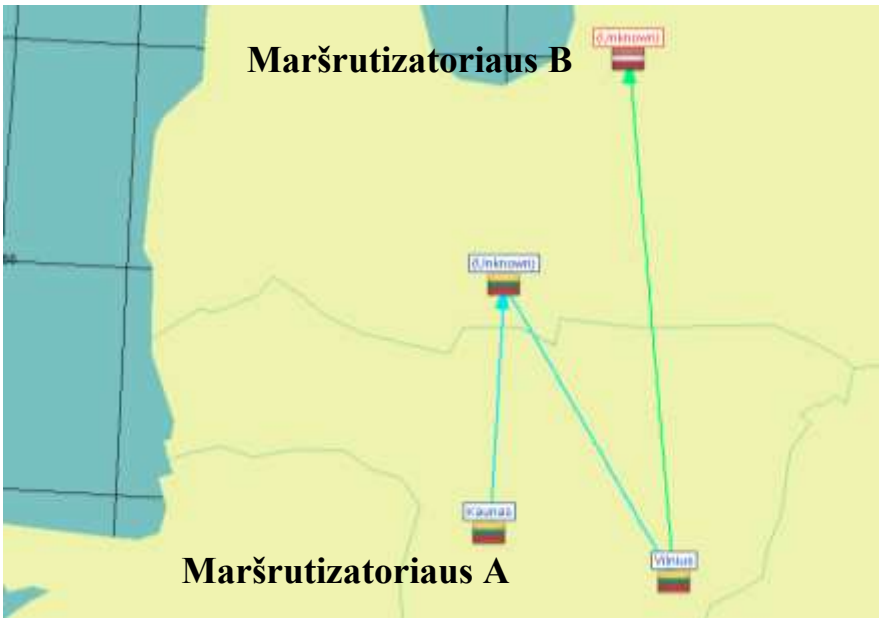
Šio tyrimo metu (49 paveikslas) pastebimi žymesni siuntimo greičio svyravimai. Jie galėjo atsirasti dėl išorinių veiksnių kaip tarpinių maršrutizatorių skaičius, jų apkrova, bei pralaidumai tarp jų.

Iš tyrimo rezultatų (5 lentelė) nustatyta, kad šioje konfigūracijoje taip pat egzistuoja logaritminė priklausomybė tarp duomenų segmento dydžio ir vidutinio greičio, bei duomenų segmento dydžio ir persiųstų duomenų kiekio. Taip pat pastebima, kad vėlinimas, kol nepasiekia vieno TCP paketo transportuojamų duomenų dydžio išlieka pastovus apie 31milisekunde, tačiau padidinus duomenų segmento dydį išauga neproporcingai.

5 lentelė Plačiojo tinko tyrimo rezultatai

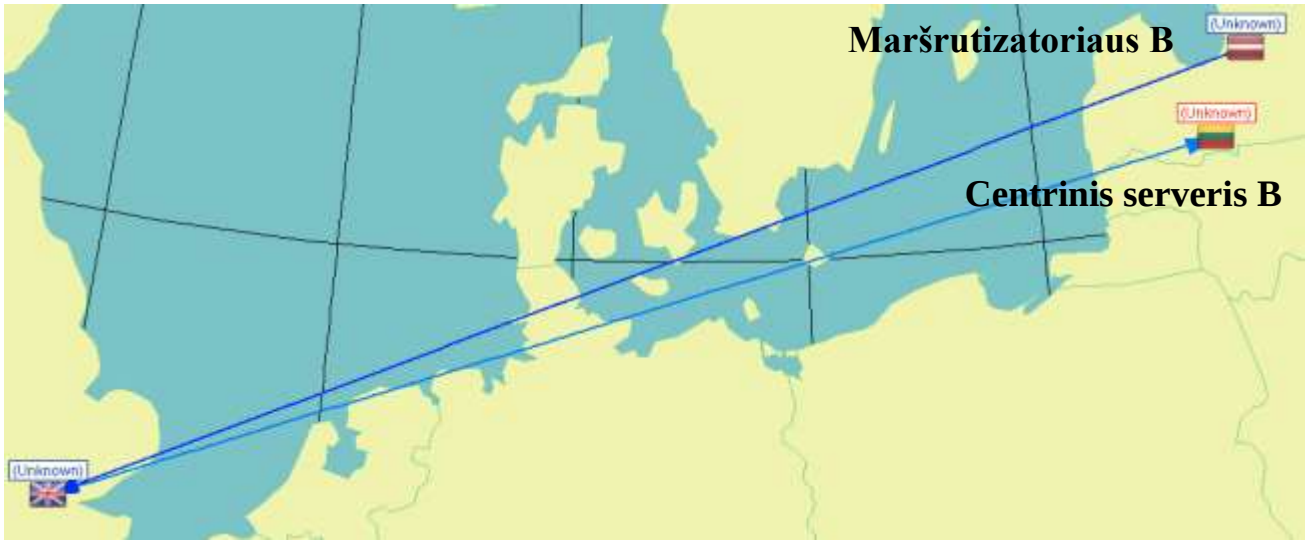
Siunčiamų duomenų dydis baitais	1	10	100	1000	10000	65535	100000
Persiųstas duomenų kiekis, Kb	20,408	204	2040,8	20352	199440	361228,92	420800
Paketų skaičius, vnt	2550	2549	2550	2543	2492	688	525
Vidutinis vėlinimas, ms	31,22	31,33	31,20	31,30	31,99	156,51	211,01
Vidutinis greitis, Kb/s	3,46	34,56	345,76	3 448,14	33 789,83	61 136,38	71 186,44

Toliau pateikiama maršruto informacija, kuriuo buvo atliekamas tyrimas, bei apžvelgiamas numatomas maršrutas plačiojo interneto tyrimui naudojant centrinį serverį B.



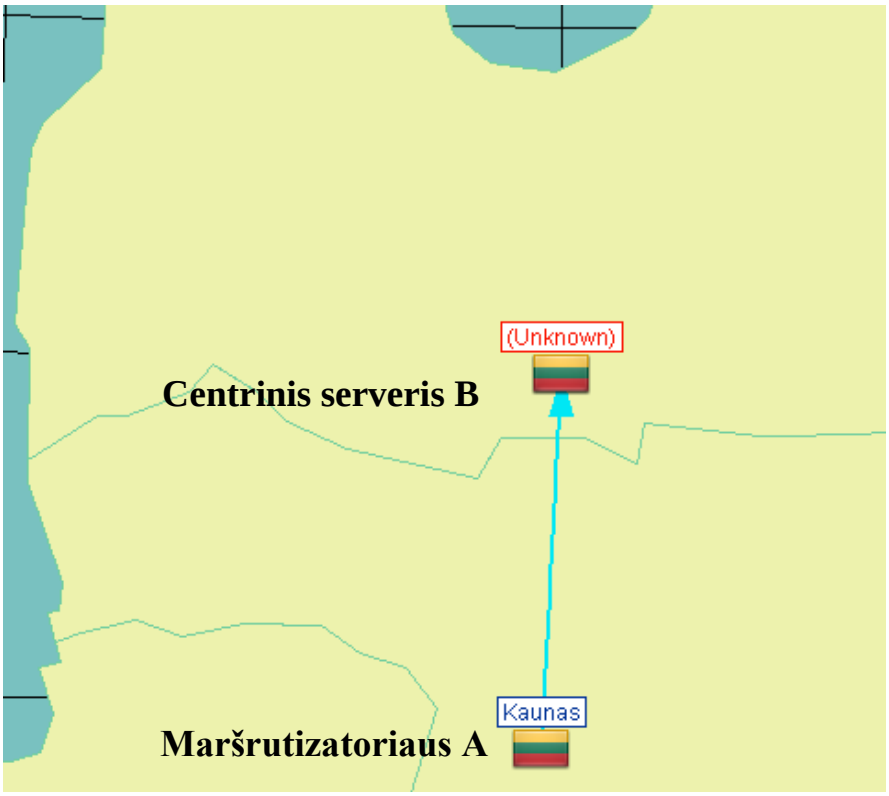
50 pav. Maršrutizatoriaus A maršrutas iki maršrutizatoriaus B

Maršrutas tarp maršrutizatoriaus A ir maršrutizatoriaus B pavaizduotas 50 paveiksle. Siunčiami duomenys tarp įrenginių keliauja per aštuonis tarpinius maršrutizatorius kol pasiekia gavėją.



51 pav. Maršrutizatoriaus B maršrutas iki centrinio serverio B

Maršrutizatoriaus B maršrutas iki centrinio serverio B pavaizduotas 51 paveiksle. Siunčiami duomenys keliauja per devynis tarpinius maršrutizatorius kol pasiekai gavėją.



52 pav. Maršrutizatoriaus A maršrutas iki centrinio serverio B

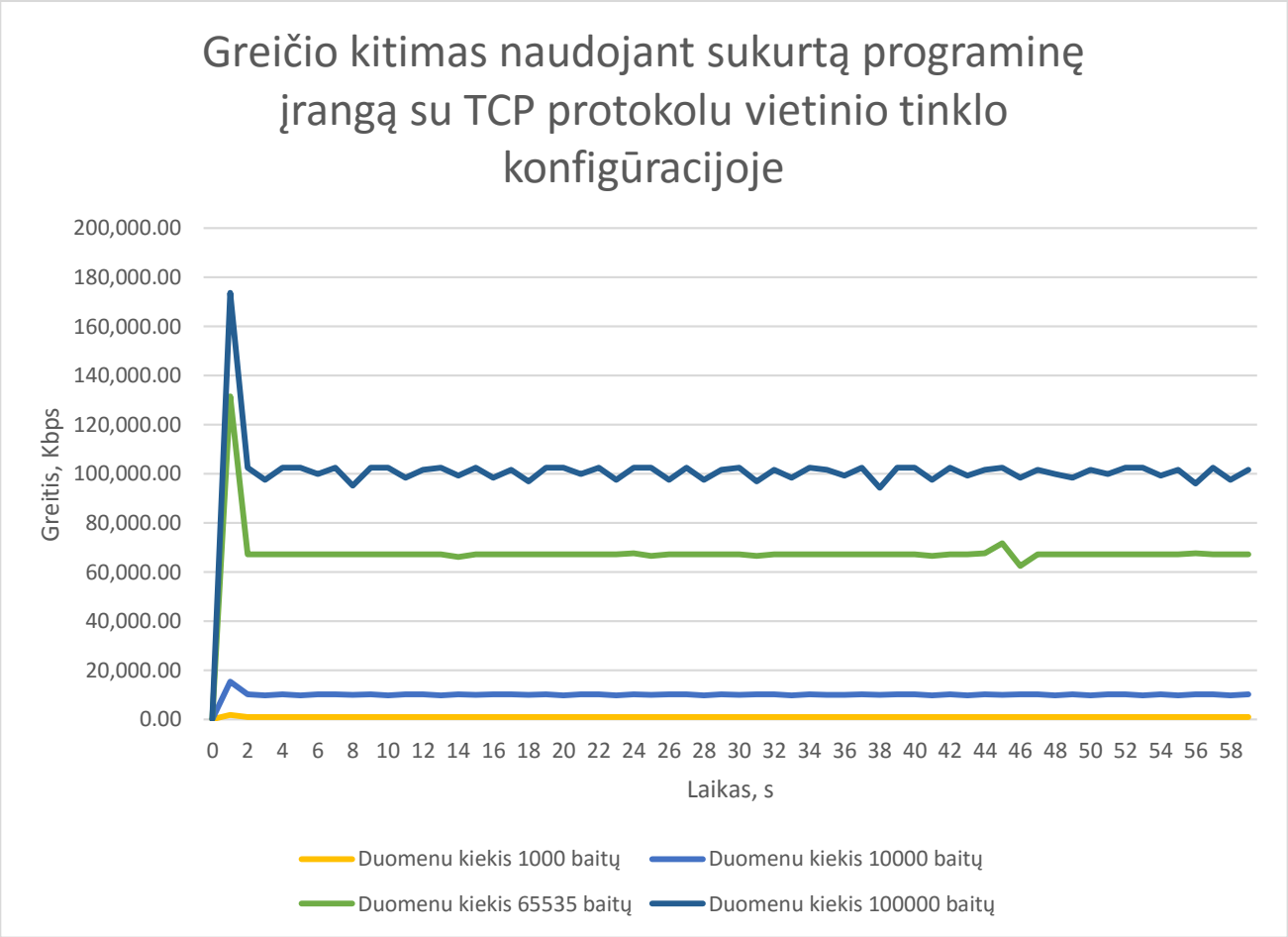
Maršrutas tarp maršrutizatoriaus A ir centrinio serverio B pavaizduotas 52 paveiksle. Siunčiami duomenys keliauja per šešis tarpinius maršrutizatorius. Maršrutų skirtumas gali įtakoti gaunamus rezultatus.

4.2.5 TCP protokolas

Siekiant ištirti TCP protokolo įtaką duomenų perdavimui sukurta programine įranga, TCP protokolas bus naudojamas duomenų perdavimui tarp kliento programinės įrangos, veikiančios klientų kompiuteriuose, ir centrinio serverio. Bus tiriamos vietinio ir plačiojo tinklų konfigūracijos. Šie tyrimai pateikiami tolimesniuose skyriuose.

4.2.5.1 Vietinio tinklo tyrimas

Šio tyrimo metu buvo tiriama vietinio tinklo konfigūracija pagal schemą pavaizduota 44 paveiksle. Tyrimo rezultatai pateikiamai 53 paveiksle ir 6 lentelėje.



53 pav. Greičio kitimas naudojant sukurtą programinę įrangą su TCP protokolu vietinio tinklo konfigūracijoje

Tyrimo metu (53 paveikslas) pastebėta, kad kol duomenų segmentas telpa į vieną TCP paketą tol duomenų siuntimo greitis laikosi pastoviame lygmenyje. Tai parodo duomenų segmento dydžiai nuo 1 baito iki 10000 baitų. Duomenų segmentui pasiekus kritinę ribą- 65535 baitus pastebimi duomenų perdavimo greičio svyravimai. Akivaizdus greičio svyravimas pastebimas kuomet duomenų segmento dydis padidinamas iki 100000 baitų. Tai atsitinka dėl to, kad šį duomenų segmentą TCP transporto protokolas perduoda naudodamas daugiau nei vieną paketą.

6 lentelė Vietinio tinklo tyrimo naudojant sukurtą programinę įrangą ir TCP protokolą rezultatai

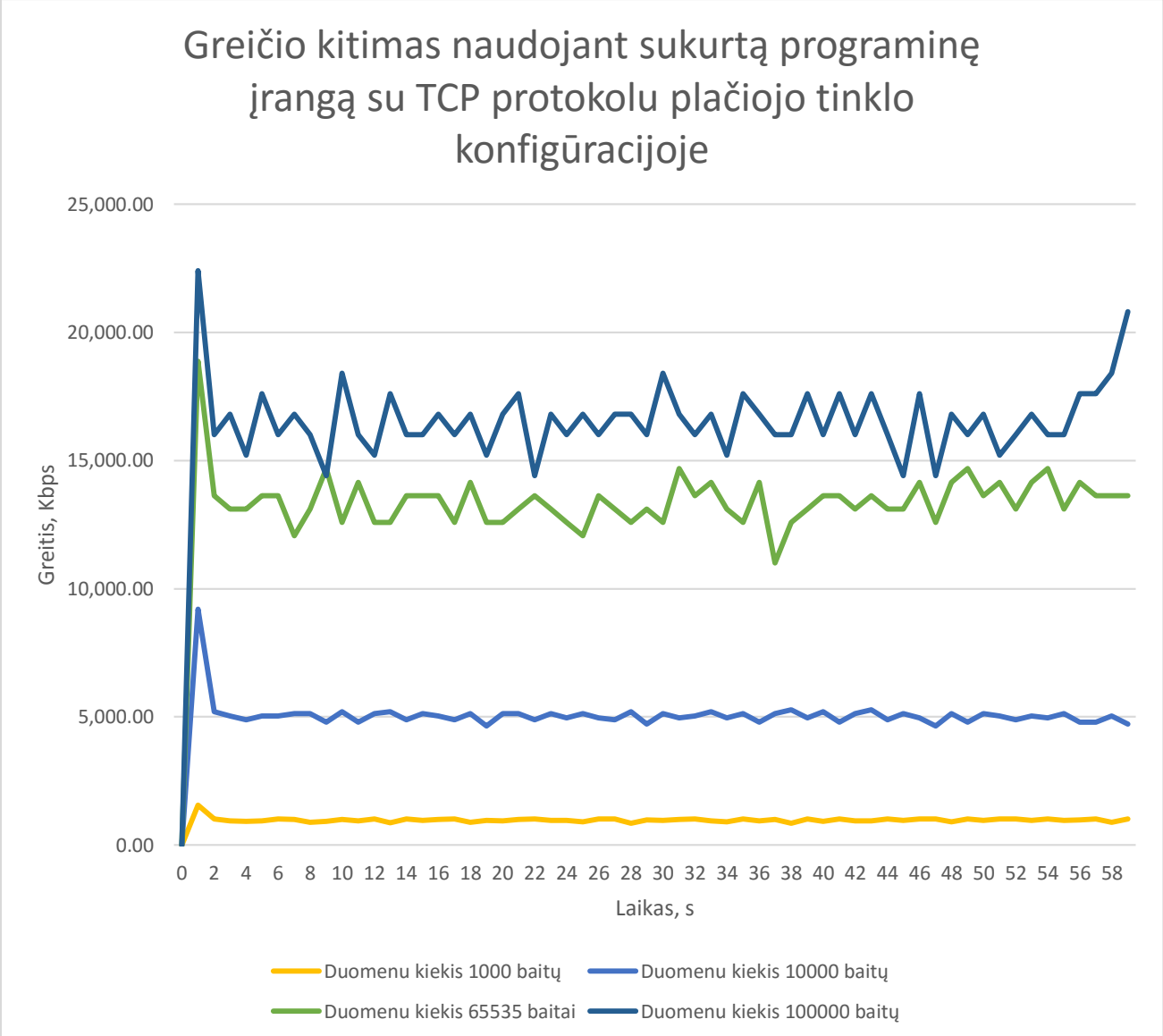
Siunčiamų duomenų dydis, baitais	1	10	100	1000	10000	65535	100000
Persiųstas duomenų kiekis, Kb	60,504	602,16	6036	60312	599760	402332,472	6001600
Paketų skaičius, vnt	7562	7526	7544	7538	7496	7673	7501
Vidutinis vėlinimas, ms	0,18	0,13	0,15	0,19	0,17	0,21	0,25
Vidutinis greitis, Kb/s	1,03	10,20	102,29	1 022,10	10 164,07	68 183,06	101 708,47

Taip pat tyrime (**6** lentelė) pastebėta, kad išliko logaritminė priklausomybė tarp duomenų segmento dydžio ir vidutinio greičio, bei tarp duomenų segmento dydžio ir persiųstų duomenų kiekio.

Būtina pabrėžti, kad tarp siunčiamų duomenų segmente dydis padidėjo 100000 kartų, o vėlinimas patiko tik 0,7 milisekundės, nuo 0.18 iki 0.25 milisekundės. Iš to galima teigti, kad sukurta programa veikianti vietinio tinklo konfigūracijoje ir naudojanti TCP protokolą duomenų persiuntimui suteikia 0,2 milisekundės vėlinimą, kuomet žmogaus toleruojama riba tarp 200 ir 600 milisekundžių. Toks sistemos naudojimas neturės jokios jaučiamos įtakos interaktyviam darbui.

4.2.5.2 Plačiojo tinklo tyrimas

Šio tyrimo metu buvo tiriama plačiojo tinklo konfigūracija pagal schemą pavaizduota 47 paveiksle. Tyrime bus naudojam sukurta sistema ir TCP protokolas duomenų perdavimui. Tyrimo rezultatai pateikiami 54 paveiksle ir 7 lentelėje.



54 pav. Greičio kitimas naudojant sukurtą programinę įrangą su TCP protokolu plačiojo tinklo konfigūracijoje

Tiriant duomenų perdavimą plačiuoju tinklu ir naudojant TCP duomenų perdavimo protokolą (54 paveikslas) pastebimas didelis duomenų perdavimo greičio svyravimas pasiekus didesnį duomenų segmentą nei vienas TCP paketas gali perduoti. Atsižvelgiant į vietinio tinklo tyrimą (53 paveikslas ir 6 lentelė) galima teigti, kad tai įtakojo išoriniai veiksniai. Staigų greičio šuolį paskutinėmis matavimo sekundėmis galėjo įtakoti išoriniai veiksmai, kaip siuntimas kitu maršrutu ar tarpinių įrenginių apkrovos sumažėjimas.

Tyriame (7 lentelė) pastebimos išlikusios logaritminės priklausomybės tarp duomenų segmento dydžio ir persiųstų duomenų, bei duomenų segmento dydžio ir vidutinio greičio.

Tyrimo metu išmatavus vidutinį vėlinimą gauta, kad jis pakito 80 kartų, nuo 0,69 iki 80 milisekundžių. Toks didelis vėlinimas dar neperkopė žmogaus toleruojamos ribos- 200-600. TCP protokolo naudojimas duomenų transportavimui šio tinklo konfigūracijoje tinkamas interaktyviam darbui ir neturės naudotojui juntamos įtakos.

7 lentelė Plačiojo tinklo tyrimo naudojant sukurtą programinę įrangą ir TCP protokolą rezultatai

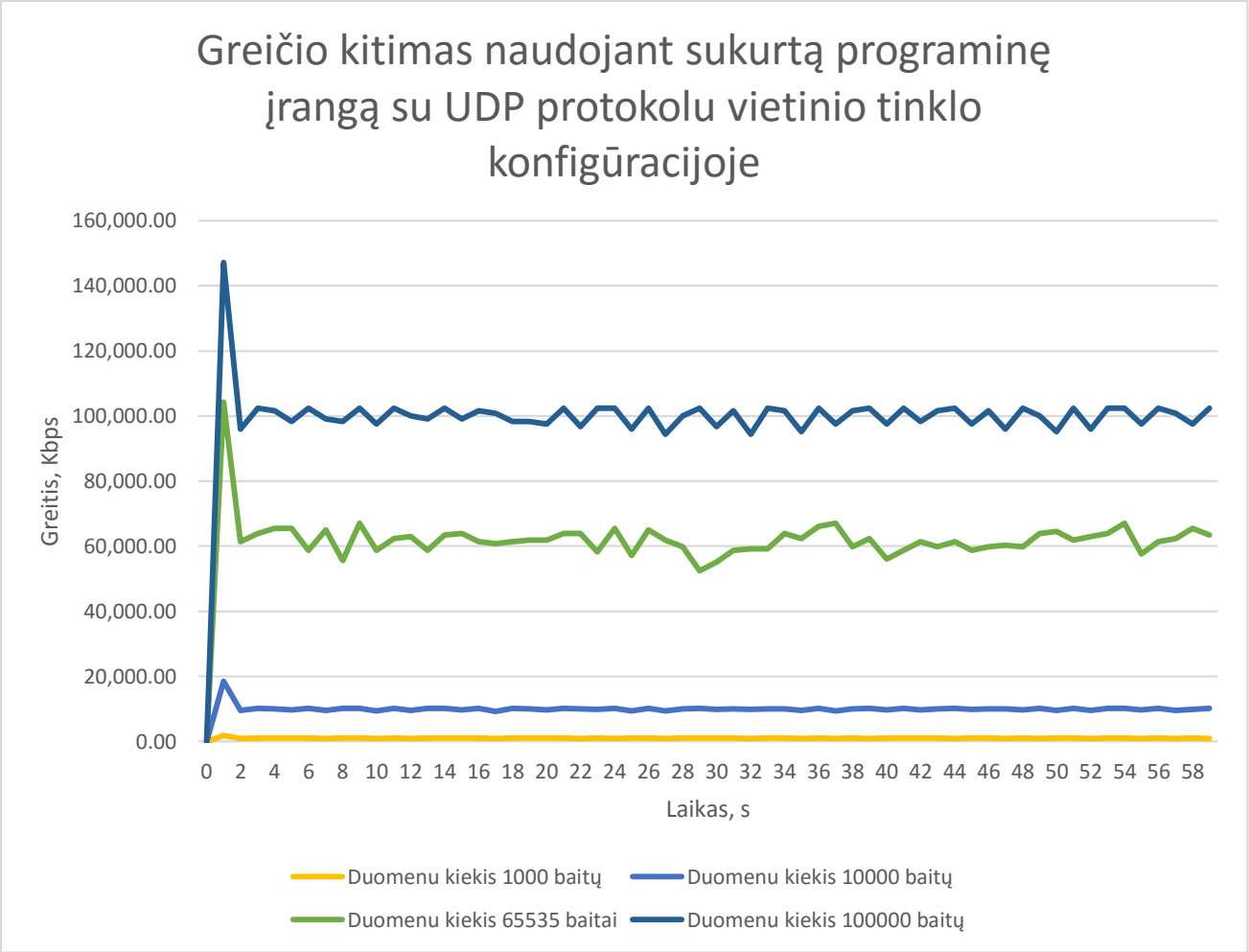
Siunčiamų duomenų dydis baitais	1	10	100	1000	10000	65535	100000
Persiųstas duomenų kiekis, Kb	57,85	585,68	5 799,20	57 752,00	299 520,00	793 235,64	980 800
Paketų skaičius, vnt	7230	7320	7248	7218	3743	1512	1225
Vidutinis vėlinimas, ms	0,69	0,58	0,65	0,71	5,52	62,20	80,07
Vidutinis greitis, Kb/s	9,80	99,25	982,78	9 787,12	50 766,10	134 357,86	166 101,69

4.2.6 UDP protokolas

Siekiant ištirti UDP protokolo įtaką duomenų perdavimui sukurta programine įranga, UDP protokolas bus naudojamas duomenų perdavimui tarp kliento programinės įrangos veikiančios klientų kompiuteriuose iki centrinio serverio. Bus tiriamos vietinio ir plačiojo tinklų konfigūracijos. Šie tyrimai pateikiami tolimesniuose skyriuose.

4.2.6.1 Vietinio tinklo tyrimas

Šio tyrimo metu siekiama nustatyti UDP protokolo naudojimo duomenų transportavimui įtaką, naudojantis sukurta programine įranga. Tyrimo rezultatai pateikiami 55 paveiksle ir 8 lentelėje.



55 pav. Greičio kitimas naudojant sukurtą programinę įrangą su TCP protokolu vietinio tinklo konfigūracijoje

Tyrimo metu padidinus duomenų segmentą iki vieno UDP paketo pernešamų duomenų dydžio ir didesnio pastebėti dideli greičio svyravimai. Šis reiškinys vyksta dėl to, kad duomenų segmentą UDP protokolas siunčia per kelis paketus.

Analizuojant tyrimo rezultatus (8 lentelė) nustatyta, kad siunčiamų duomenų kiekiui ir duomenų perdavimo vidutiniam greičiui galioja tiesinė priklausomybė. Taip pat šią priklausomybę galime įžvelgti ir tarp siunčiamų duomenų kiekio ir persiųstų duomenų kiekio.

Palyginus šio tyrimo rezultatus su atitinkamo tyrimo rezultatais (6 lentelė), naudojus TCP protokolą, galime įžvelgti, kad lyginant vidutinį greitį UDP protokolas yra 10 kartų greitesnis, nei TCP, tačiau lyginant persiųstų duomenų kiekį UDP protokolas nusileidžia TCP protokolui. Taip pat UDP protokolas patiria didesnius vėlinimus. Jo vėlinimai svyruoja nuo 0,21 iki 462557 milisekundžių. Tokie dideli svyravimai atsiranda dėl UDP protokolo duomenų pametimo perdavimo metu, ir perduoto TCP protokolo pakartotino duomenų siuntimo. Išsiųstų paketų skaičius išliko panašus lyginant abu protokolus.

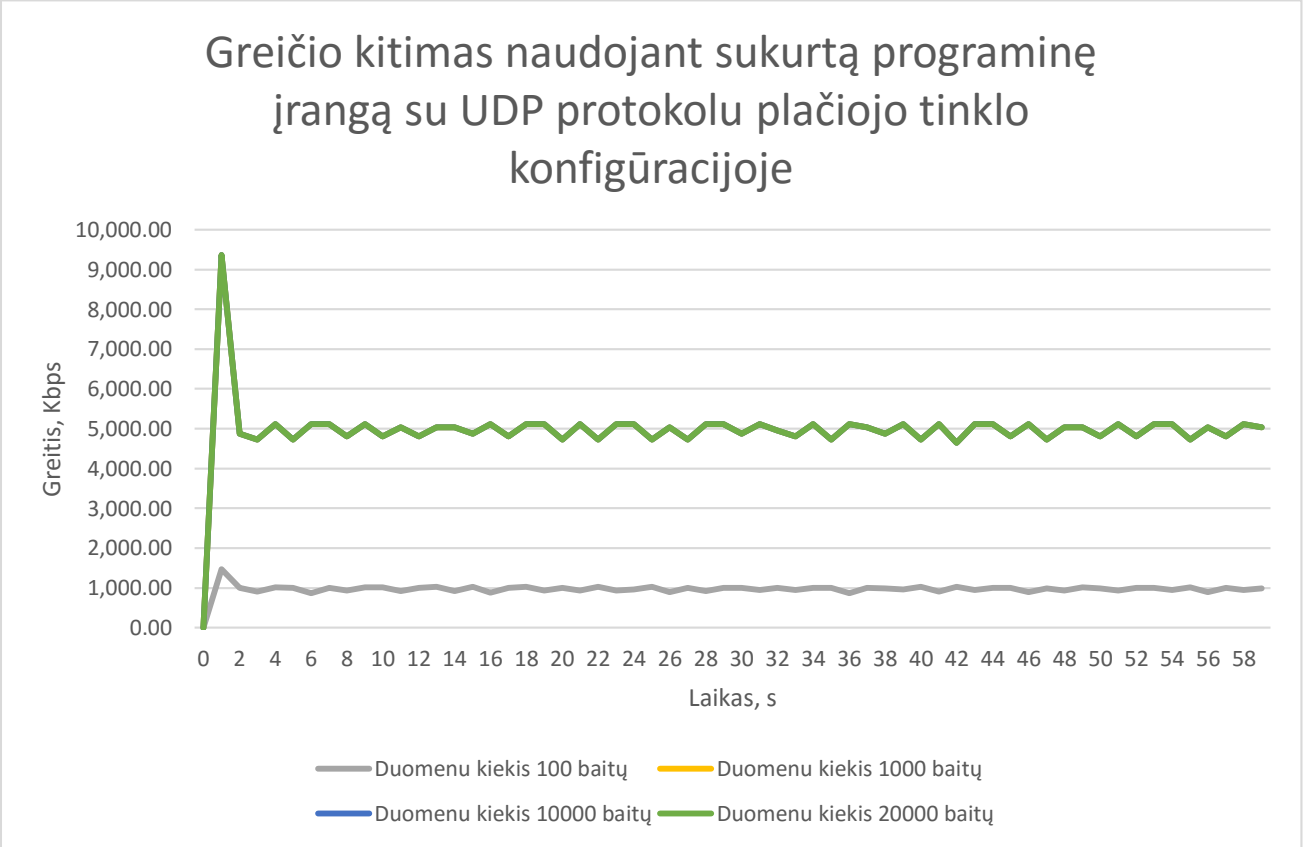
8 lentelė Vietinio tinklo tyrimo naudojant sukurtą programinę įrangą ir UDP protokolą rezultatai

Siunčiamų duomenų dydis baitais	1	10	100	1000	10000	65535	100000
Persiųstas duomenų kiekis, Kb	59,584	600,4	6000,8	59840	592320	3677299,9	5940800
Paketų skaičius, vnt	7447	7504	7500	7479	7403	7013	7425
Vidutinis vėlinimas, ms	0,30	0,26	0,22	0,21	462557,93	1,40	0,24
Vidutinis greitis, Kb/s	10,10	101,75	1 016,95	10 141,02	100 379,66	623 182,31	1 006 779,66

4.2.6.2 Plačiojo tinko tyrimas

Šio tyrimo metu buvo tiriama plačiojo tinklo konfigūracija pagal schemą pavaizduota 47 paveiksle. Tyrime bus naudojam sukurta sistema ir UDP protokolas duomenų perdavimui. Tyrimo rezultatai pateikiami 56 paveiksle ir 9 lentelėje.

Perduodant duomenis per centrinį serverį naudojant plačiojo tinklo konfigūracija (47 paveikslas) su UDP duomenų perdavimo protokolu nepavyko perduoti didesnių duomenų segmentų nei 20000 baitų. Tai galėjo įvykti dėl klaidos sukurtoje programinėje įrangoje, matavimų programinės įrangos klaidos arba dėl išorinių veiksnių. Tačiau atsižvelgiant į vietinio tinklo tyrimo rezultatus (8 lentelė) programinės įrangos ir tyrimo įrangos veiksniai turi būti atmetami.



56 pav. Greičio kitimas naudojant sukurtą programinę įrangą su UDP protokolu plačiojo tinklo konfigūracijoje

Tyrimo metu (56 paveikslas) pastebimas nuolatinis greičio svyravimas. Taip pat vienodas greitis siunčiant duomenų segmentus po 10000 baitų ir didesnius. Šis vienodo greičio reiškinys galėjo pasireikšti dėl programinėje įrangos realizacijos arba dėl išorinių veiksnių. Atsižvelgiant į vietinio tinklo tyrimus (8 lentelė) programinės įrangos realizacija turi būti atmetama. Šį reiškinį turėjo įtakoti išoriniai veiksniai.

9 lentelė Plačiojo tinklo tyrimo naudojant sukurtą programinę įrangą ir UDP protokolą rezultatai

Siunčiamų duomenų dydis baitais	1	10	100	1000	10000	20000
Persiųstas duomenų kiekis, Kb	57,656	581,68	57864	297200	297200	603040
Paketų skaičius, vnt	7206	7270	7232	3714	3714	3768
Vidutinis vėlinimas, ms	0,85	0,74	1,63	0,71	16,39	16,00
Vidutinis greitis, Kb/s	9,77	98,58	9 806,10	50 359,32	50 359,32	102 183,05

Tyrimo metu (9 lentelė) buvo pastebėta, kad perduodant duomenis segmentu po 1000 baitų ar 10000 baitų buvo pasiektas toks pats greitis ir duomenų perdavimas. Šiam reiškiniui turėto turėti įtakos išoriniai veiksniai.

Lyginant šio tyrimo rezultatus su atitinkamo tyrimo rezultatais (7 lentelė) naudojant TCP protokolą nenustatyta esminių skirtumų, išskyrus tai, kad UDP protokolu nepavyko perduoti didesnio duomenų segmento nei 20000 baitai. Tokie reiškiniai sumažina sukurtos sistemos pasiekiamumą naudojant UDP protokolą.

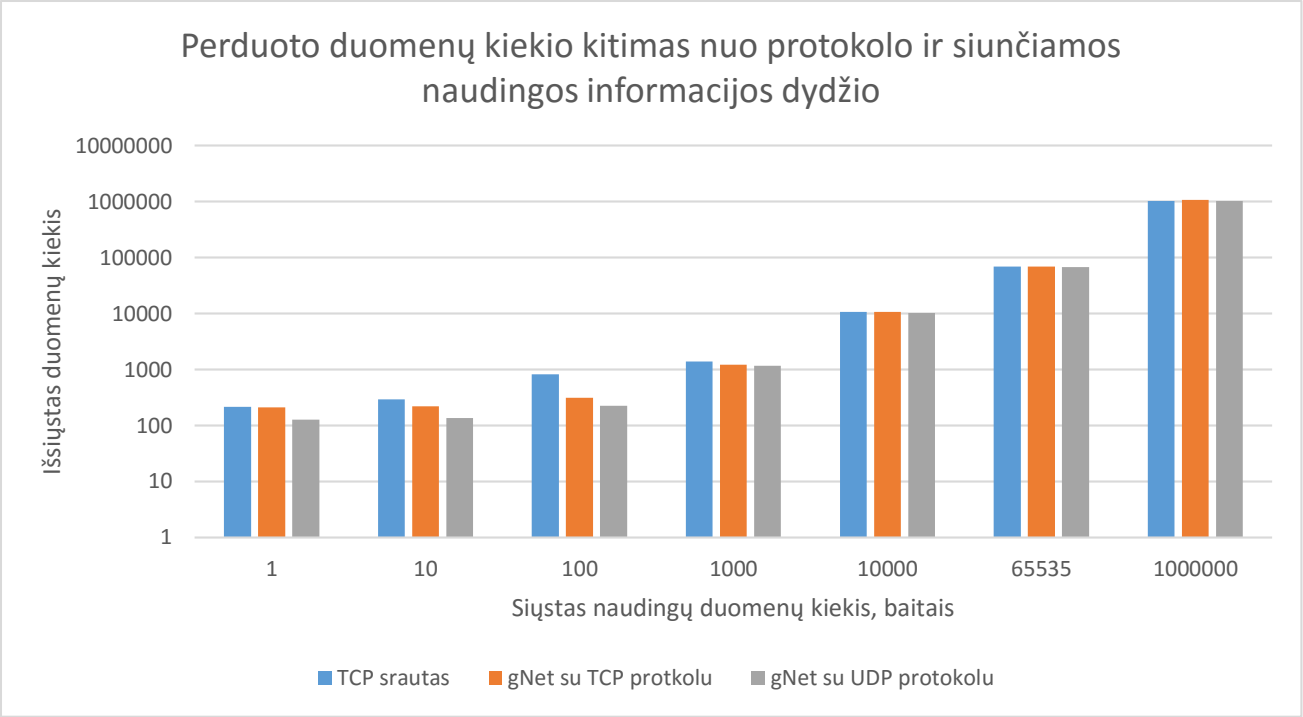
4.3 Naudingų duomenų perdavimo tyrimas

Šio tyrimo metu bus siekiama išsiaiškinti kiek naudingų duomenų yra perduodama vienu paketu. Ši informacija aktuali mobiliems naudotojams, kurių persiunčiamų duomenų kiekiai yra apmokestinami.

Matavimų rezultatai pateikiami 10 lentelėje ir 57 paveiksle.

10 lentelė Naudingų duomenų siuntimas TCP ir UDP protokolais

Siųstas nuadingų duomenų kiekis, baitais	1	10	100	1000	10000	65535	1000000
Išsisųtas duomenų kiekis naudojant TCP srautą	217	292	818	1386	10768	68409	1039000
Išsiųstas duomenų kiekis naudojant gNet su TCP protkolu	211	220	310	1210	10726	69162	1070510
Išsisųtas duomenų kiekis naudojant gNet su UDP protokolu	127	136	226	1162	10337	67212	1023690



57 pav. Perduoto duomenų kiekio kitimas nuo protokolo ir siunčiamos naudingos informacijos dydžio

Tyrimo metu (57 paveikslas ir 10 lentelė) buvo nustatyta, kad iš lygintų duomenų perdavimo būdų naudingiausia duomenis perduoti naudojant sukurtą programinę įrangą su UDP transportavimo protokolu. Klientams naudojantiems interneto ryšiu, kuris apmokestinamas pagal sunaudotą kiekį, šis būdas būtų pigiausias.

Perduodant mažus duomenų kiekius (1- 10000 baitų) naudingiau perdavinėti naudojantis sukurta programine įranga su TCP transporto protokolu nei tiesioginiu TCP protokolu.

4.4 Apibendrinimas

Pagal atliktus tyrimus (11 lentelė ir 12 lentelė) buvo nustatyta, kad vietiniame tinkle perduodant duomenis naudojantis sukurta programine įranga su UDP transporto protokolu buvo pasiekta didžiausias greitis. Jis buvo didesnis už tiesiogiai dirbantį TCP srautą. Tai turėjo lemti UDP protokolo paprastumas ir nereikalavimas patvirtinimo apie gautus duomenis.

Vietinio tinklo tyrime nustatyta, kad naudojantis sukurta programine įranga su TCP transporto protokolu galima perduoti daugiau duomenų nei naudojantis tiesioginiu TCP protokolu.

Atliekant vėlinimo tyrimą su sukurta programine įranga ir UDP transporto protokolu perduodant duomenis po 100000 baitų nepavyko nustatyti vėlinimo. Tyrime gauti neproporcingai dideli vėlinimai. Todėl šio matavimo duomenys nebuvo naudojami.

Plačiojo tinklo tyrimo metu buvo nustatyta, kad duomenis perduoti UDP protokolu ne visuomet pavyks. Tai įtakoja išoriniai veiksniai. Tačiau tiriant tinklo parametrus su sukurta programine įranga buvo gauti geresni rezultatai nei tiesioginio sujungimo. Tai galėjo įtakoti skirtingi maršrutai.

Atliekant tyrimus su sukurta programine įranga ir UDP transporto protokolu ir vidutiniais duomenų dydžiais (1000- 10000 baitų) buvo gauti tokie patys rezultatai. Šiems tyrimo rezultatams galėto turėti įtakos išoriniai veiksniai. Taip pat tyrime nustatyta, kad duomenų perdavimas su sukurta programine įranga parodė geresnius rezultatus (vėlinimas, greitis ir perduotas duomenų kiekis) nei duomenų siuntimas tiesiogiai.

11 lentelė Vietinio tinklo tyrimo rezultatai

Siunčiamų duomenų dydis, baitais	Persiųstas duomenų kiekis, Kb			Paketų skaičius, vnt			Vidutinis vėlinimas, ms			Vidutinis greitis, Kb/s		
	Pradinis	TCP	UDP	Pradinis	TCP	UDP	Pradinis	TCP	UDP	Pradinis	TCP	UDP
1	60.808	60.504	59.584	7600	7562	7447	0.05	0.18	0.30	1.01	1.03	10.10
10	594.32	602.16	600.4	7428	7526	7504	0.16	0.13	0.26	10.18	10.20	101.75
100	5888.8	6036	6000.8	7360	7544	7500	0.05	0.15	0.22	97.12	102.29	1016.95
1000	59648	60312	59840	7455	7538	7479	0.05	0.19	0.21	1002.54	1022.10	10141.02
100000	592720	599760	592320	7408	7496	7403	0.22	0.17	-	9811.67	10164.07	100379.66
65535	3987673.68	402332.472	3677299.92	7605	7673	7013	0.23	0.21	1.40	67085.31	68183.06	623182.31
1000000	5920800	6001600	5940800	7400	7501	7425	0.25	0.25	0.24	95232.52	101708.47	1006779.66

12 lentelė Plačiojo tinklo tyrimų rezultatai

Siunčiamų duomenų dydis, baitais	Persiųstas duomenų kiekis, Kb			Paketų skaičius, vnt			Vidutinis vėlinimas, ms			Vidutinis greitis, Kb/s		
	Pradinis	TCP	UDP	Pradinis	TCP	UDP	Pradinis	TCP	UDP	Pradinis	TCP	UDP
1	20.41	57.85	57.656	2550	7230	7206	31.22	0.69	0.85	3.46	9.80	9.77
10	204.62	585.68	581.68	2549	7320	7270	31.33	0.58	0.74	34.56	99.25	98.58
100	2040.80	5799.20	57864	2550	7248	7232	31.20	0.65	1.63	345.76	982.78	9806.10
1000	20352	57752	297200	2543	7218	3714	31.30	0.71	0.71	3448.14	9787.12	50359.32
100000	199440	299520	297200	2492	3743	3714	31.99	5.52	16.39	33789.83	50766.10	50359.32
65535	361228.92	793235.64	-	688	1512	-	156.51	62.20	-	61136.38	134357.86	-
1000000	420800	980800	-	5250	1225	-	211.01	80.07	-	71186.44	166101.69	-

5 IŠVADOS

Atlikus sukurtos programinės įrangos tyrumą, buvo nustatyta:

1. Dėl tarpinės infrastruktūros gali neveikti duomenų perdavimas UDP protokolu.
2. Tyrimo metu plačiuoju tinklu nutolusių klientų sujungimas naudojant sukurtą programinę įrangą parodė geresnius rezultatus, nei klientų sujungtų tiesiogiai.
3. Vietiniame tinkle perduodant duomenis naudojantis sukurta įranga ir UDP transporto protokolu buvo užfiksuoti žymiai geresni rezultatai lyginant su tiesioginiu TCP srautu ir sukurta įranga naudojančia TCP transporto protokolą.
4. Plačiajame tinkle perduodant duomenis per sukurtą programinę įranga naudojant UDP transporto protokolą buvo užfiksuoti žymiai geresni rezultatai vidutinio dydžio (100- 1000 baitų) paketams lyginant su tiesioginiu TCP srautu ir sukurta programine įranga naudojančia TCP transportavimo protokolą.
5. Tyrimo metu nustatyta, kad sukurta programinė įranga interaktyviam darbui reikalingų parametrų (vėlinimas, perduodamų duomenų kiekis, vidutinis greitis) nepablogina, o daugelyje atvejų pagerina.
6. Tyrimo metu nustatyta, kad naudojant sukurtą programinę įrangą su UDP transporto protokolu yra perduodama mažiau perteklinės informacijos nei perduodant duomenis tiesioginiu TCP srautu.

Tolimesniam programinės įrangos plėtojimui būtina realizuoti vartotojų autorizavimą ir siunčiamų duomenų šifravimą.

6 BIBLIOGRAFIJA

1. PROTOCOL SPECIFICATION. *TRANSMISSION CONTROL PROTOCOL.* [Tinkle] 1981 m. rugsėjis. [Cituota: 2014 m. lapkričio 27 d.] https://www.ietf.org/rfc/rfc793.txt.

2. CCNA Routing and Switching: Introduction to Networks. *Cisco Network Academy.* [Tinkle] [Cituota: 2013 m. rugsėjo 1 d.] https://1364609.netacad.com/courses/77360/modules.

3. Stevens W.Richard. *TCP/IP Illustrated, Vol. 1: The Protocols.* 1993.

4. User Datagram Protocol. [Tinkle] 1980 m. rugpjūčio 28 d. [Cituota: 2014 m. lapkričio 27 d.] https://www.ietf.org/rfc/rfc768.txt.

5. R. Stewart, Q. Xie, K. Morneault, C. Sharp, H. Schwarzbauer, T. Taylor, I. Rytina, M. Kalla, L. Zhang, V. Paxson. Stream Control Transmission Protocol. [Tinkle] 2000 m. rugsėjis. [Cituota: 2014 m. lapkričio 27 d.] http://tools.ietf.org/html/rfc2960.

6. K. Egevang, P. Francis. The IP Netowrk Address Translation (NAT). [Tinkle] 1994 m. liepa. [Cituota: 2014 m. 11 30 d.] https://www.ietf.org/rfc/rfc1631.txt.

7. CCNA Routing and Switching: Routing and Switching Essentials. *Cisco Network Academy.* [Tinkle] [Cituota: 2014 m. sausio 1 d.] https://1364609.netacad.com/courses/103429/modules.

8. M. Boucadair, R. Penno, D. Wing. Internet Gateway Device - Port Control Protocol Interworking Function. *Universal Plug and Play (UPnP).* [Tinkle] 2013 m. liepa. [Cituota: 2015 m. sausio 6 d.] https://tools.ietf.org/html/rfc6970.

9. Desktop Operating System Market Share. *Market Share Statistics for Internet Technologies.* [Tinkle] 2016 m. kovas. [Cituota: 2016 m. gegužės 6 d.] https://www.netmarketshare.com/operating-system-market-share.aspx?qprid=10&qpcustomd=0&qpcd=130&qpsp=2016&qpnp=1&qptimeframe=Y.

10. Debian -- The Universal Operating System . [Tinkle] [Cituota: 2016 m. gegužės 6 d.] https://www.debian.org/index.en.html.

11. Active Directory Domain Services. [Tinkle] [Cituota: 2016 m. gegužės 6 d.] https://technet.microsoft.com/en-us/library/dd448614.aspx.

12. Group Policy. [Tinkle] [Cituota: 2016 m. gegužės 6 d.] https://technet.microsoft.com/en-us/windowsserver/bb310732.aspx.

13. Windows registry information for advanced users. [Tinkle] [Cituota: 2016 m. gegužės 6 d.] https://support.microsoft.com/en-us/kb/256986.

14. Extensible Markup Language (XML). *The World Wide Web Consortium (W3C).* [Tinkle] 2015 m. gegužės 19 d. [Cituota: 2015 m. lapkričio 25 d.] https://www.w3.org/XML/.

15. JavaScript Object Notation (JSON). [Tinkle] 1999 m. gruodis. [Cituota: 2015 m. lapkričio 16 d.] http://www.json.org/.

16. SctpDrv: an SCTP driver for Microsoft Windows. [Tinkle] [Cituota: 2016 m. vasario 17 d.] http://www.bluestop.org/SctpDrv/.

17. Michael Tüxen, Felix Weinrank, Liam Staskawicz, Saúl Ibarra Corretgé, Gavriloaie Eugen-Andrei, He Liu, Reid Kleckner. A portable SCTP userland stack. [Tinkle] [Cituota: 2016 m. kovo 16 d.] https://github.com/sctplab/usrsctp.

18. Zhijuan Wang, Weifeng Wang. The Application of Hybrid Encryption Algorithm in Software Security. [Tinkle] 2012 m. lapričio 8 d. [Cituota: 2014 m. lapričio 19 d.] http://ieeexplore.ieee.org/xpls/icp.jsp?arnumber=6375216.

19. Ruhani Ab Rahman, Murizah Kassim. Technical comparison analysis of encryption algorithm on site-to-site IPSec VPN. [Tinkle] 2010 m. gruodžio 10 d. [Cituota: 2014 m. lapkričio 18 d.] http://ieeexplore.ieee.org/xpls/icp.jsp?arnumber=5735013.

20. stunnel: Documentation. [Tinkle] [Cituota: 2014 m. lapkričio 17 d.] https://www.stunnel.org/docs.html.

21. Ritu Agarwal, Dhiraj Dafouti, Shobha Tyagi. Peformance analysis of data encryption algorithms. [Tinkle] 2011 m. balandžio 10 d. [Cituota: 2014 m. lapkričio 19 d.] http://ieeexplore.ieee.org/xpls/icp.jsp?arnumber=5942029.

22. Hamachi. [Tinkle] [Cituota: 2014 m. lapkričio 17 d.] https://secure.logmein.com/products/hamachi/.

23. Open Systems Interconnection. [Tinkle] 1994 m. [Cituota: 2014 m. lapkričio 30 d.] https://www.iso.org/obp/ui/#iso:std:iso-iec:7498:-1:ed-1:v2:en. ISO/IEC 7498-1.

24. OpenVPN documentation. [Tinkle] [Cituota: 2014 m. lapkričio 17 d.] https://openvpn.net/index.php/open-source/documentation.html.

25. Bresson Emmanuel, Chevassut Olivier, Pointcheval, David. Dynamic Group Deffine- Hellman Key Exchange under Standart Assumptions. [Tinkle] 2002 m. vasario 12 d. [Cituota: 2014 m. lapkričio 22 d.] http://escholarship.org/uc/item/4c64w97m.

26. Houtven Laurens Van. *Crypto 101*. 2016.

27. CCNA Routing and Switching: Scaling Networks. *Cisco Network Academy*. [Tinkle] [Cituota: 2014 m. liepos 1 d.] https://1364609.netacad.com/courses/121811/modules.

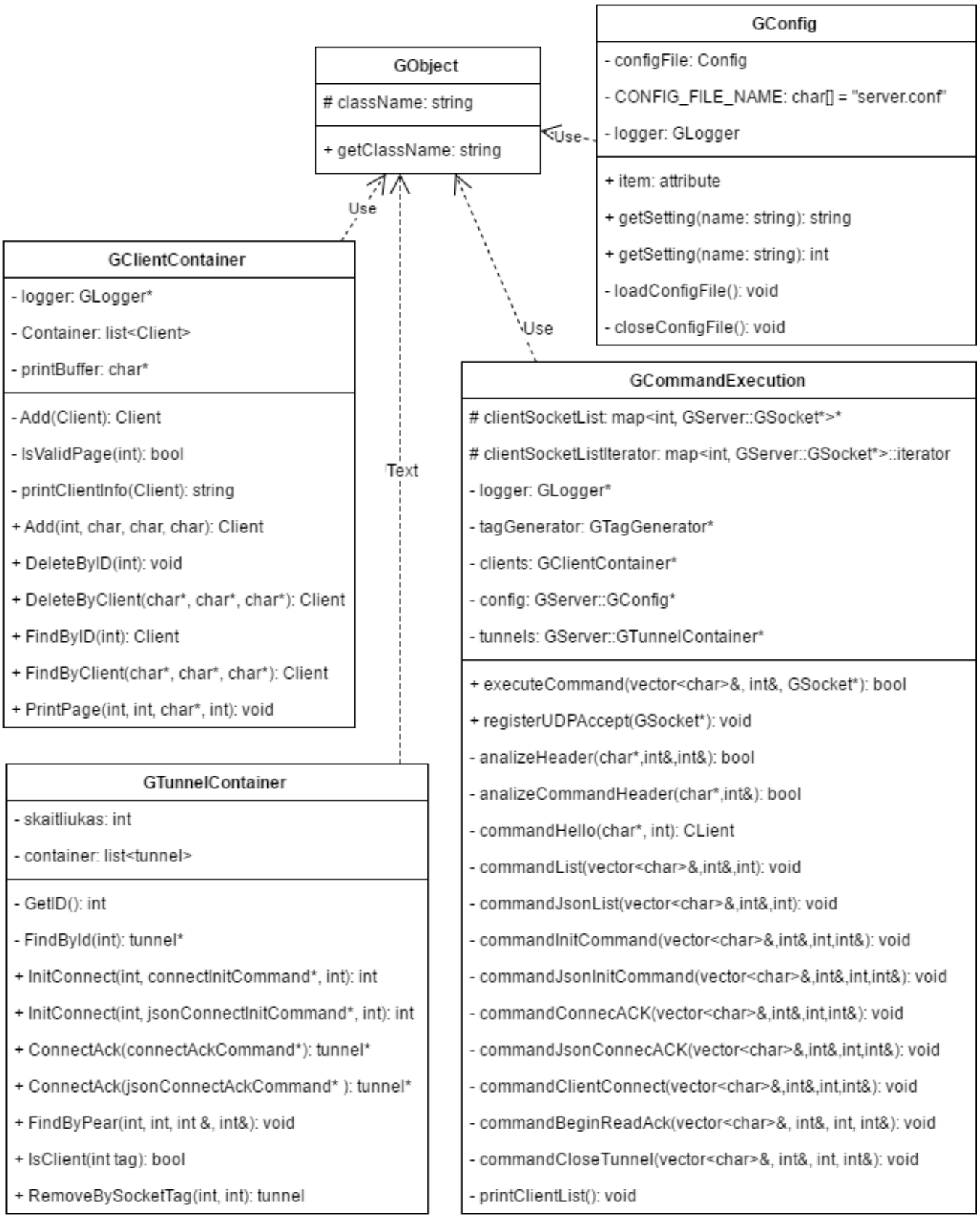
28. CCNA Routing and Switching: Connecting Networks. *Cisco Network Academy*. [Tinkle] [Cituota: 2014 m. rugsėjo 1 d.] https://1364609.netacad.com/courses/138993/modules.

7 TERMINŲ IR SANTRUMPŲ ŽODYNAS

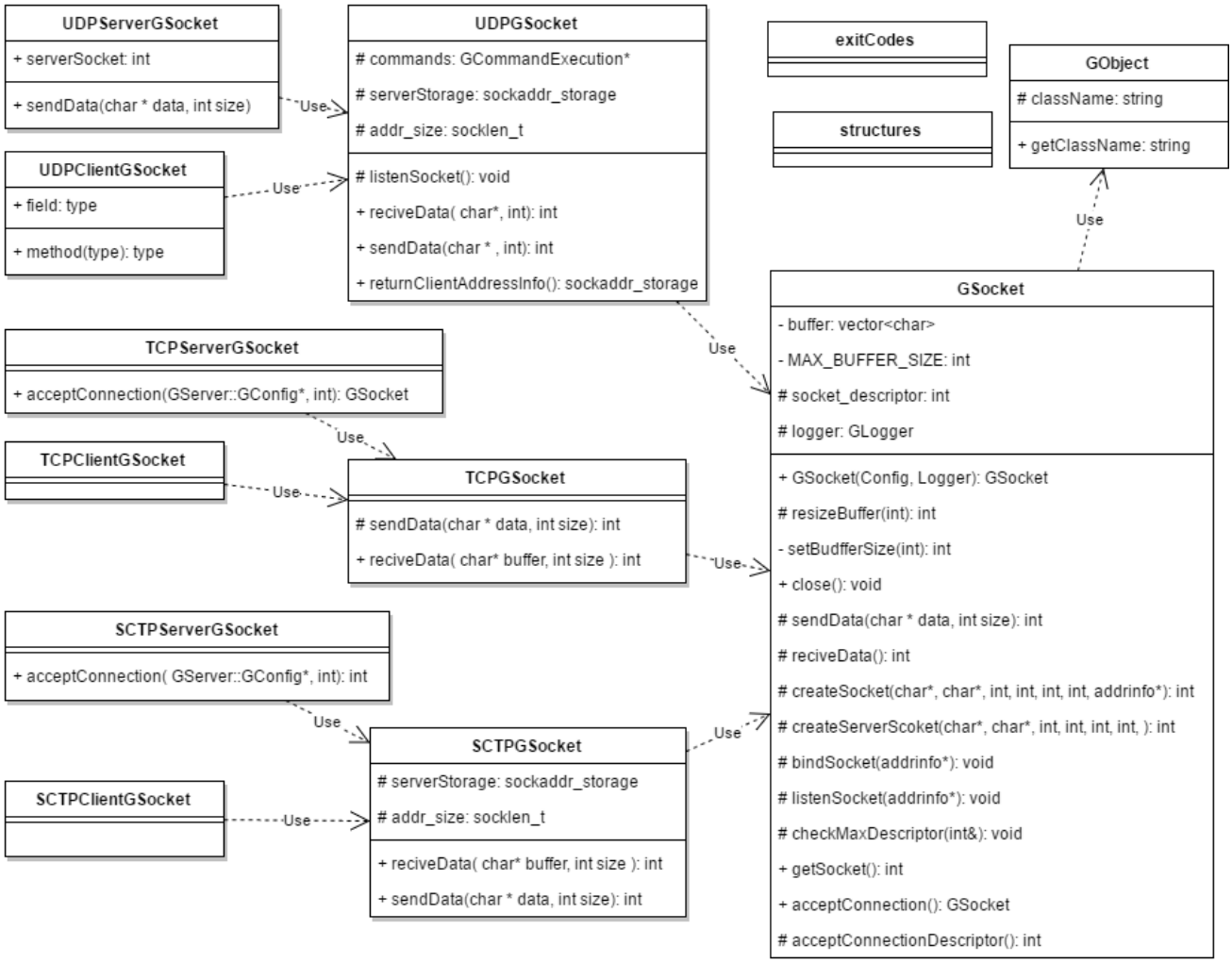
Virtualusis privatusis tinklas	(angl. Virtual Private Network) VPN. Kompiuterių tinklas, kurio tam tikros dalys susietos su internetu, tačiau visi duomenys, siunčiami internetu, yra užšifruojami naudojant įvairias priemones ir protokolus. Tokiu būdu išlaikomas tinklo privatumas. Būdingas tokio tinklo pavyzdys yra organizacijos, turinčios padalinių įvairiuose miestuose, tinklas. Pagrindinis skirtumas nuo ekstraneto – būtina internetu siunčiamų duomenų šifravimo sąlyga.
Tinklo adresų transliavimas	(angl. Network Address Translation) ANT. Metodologija, kurios dėka yra perrašomi IP paketo antraštės adresas kitu, kai keliauja per maršrutizatorių.
Maršrutizatorius	(angl. Router) Kompiuteris arba programa, apdorojanti ryšį tarp dviejų arba daugiau paketiniu būdu perjungiamų tinklų. Analizuoja gaunamus duomenų paketus, skirsto ir siunčia juos į paskirties vietas, parenka geriausią persiuntimo maršrutą.
Vietinis tinklas	(angl Local Area Network) LAN. Nedidelis kompiuterių tinklas, dažniausiai jungiantis vienos įstaigos arba atskiro jos padalinio kompiuterius su jų išoriniais įtaisais (spausdintuvais, duomenų saugyklomis ir pan.). Toks tinklas sudaro sąlygas dalytis ištekliais ir taip racionaliau juos naudoti.
Platusis tinklas	(angl Wide Area Network) WAN. Kompiuterių tinklas, apimantis didelę geografinę teritoriją, pavyzdžiui, visą valstybę. Gali aprėpti daugelį vietinių tinklų arba būti atskiras. Internetas gali būti laikomas plačiuoju tinklu, kurio teritorija – visas pasaulis.
Teritorinis tinklas	(angl. Metropolitan Area Network) MAN. Kompiuterinis tinklas, didelis geografiškai požiūriu, apimantis kelis pastatus ar net miestus.
Jungtis	Mechaninis arba elektroninis įtaisas, skirtas dviem aparatinės įrangos arba programinės įrangos komponentams arba kabeliams sujungti.

	Jungtys būna dviejų tipų: viena su išsikišusiais dantukais, o kita – su skylutėmis.
Prievadas	(angl. Port) Sąsaja, duomenims arba komandoms persiųsti tarp kompiuterio ir jo išorinių įtaisų (pvz., į spausdintuvą, iš skenerio), į kitą kompiuterį, tinklą arba iš kito kompiuterio, tinklo ir pan. Prievadą serveryje galima įsivaizduoti kaip loginę jungtį. Visi duomenys į serverio kompiuterį patenka per tą pačią jungtį ir toliau paskirstomi į prievadus pagal prievadų numerius, ateinančius kartu su duomenų srautu.
Duomenų paketas	(angl. Data packet) Duomenų porcija, siunčiama tinklu kaip nedalomas vienetas. Paketą sudaro duomenų paketo antraštė ir patys duomenys.
Tinklų tiltas	(angl. bridge connection)Tinklo komponentas, jungiantis du arba kelis vietinius tinklus į vieną tinklą. Jungiami tinklai turi būti to paties protokolo, bet jų topologijos gali būti skirtingos.

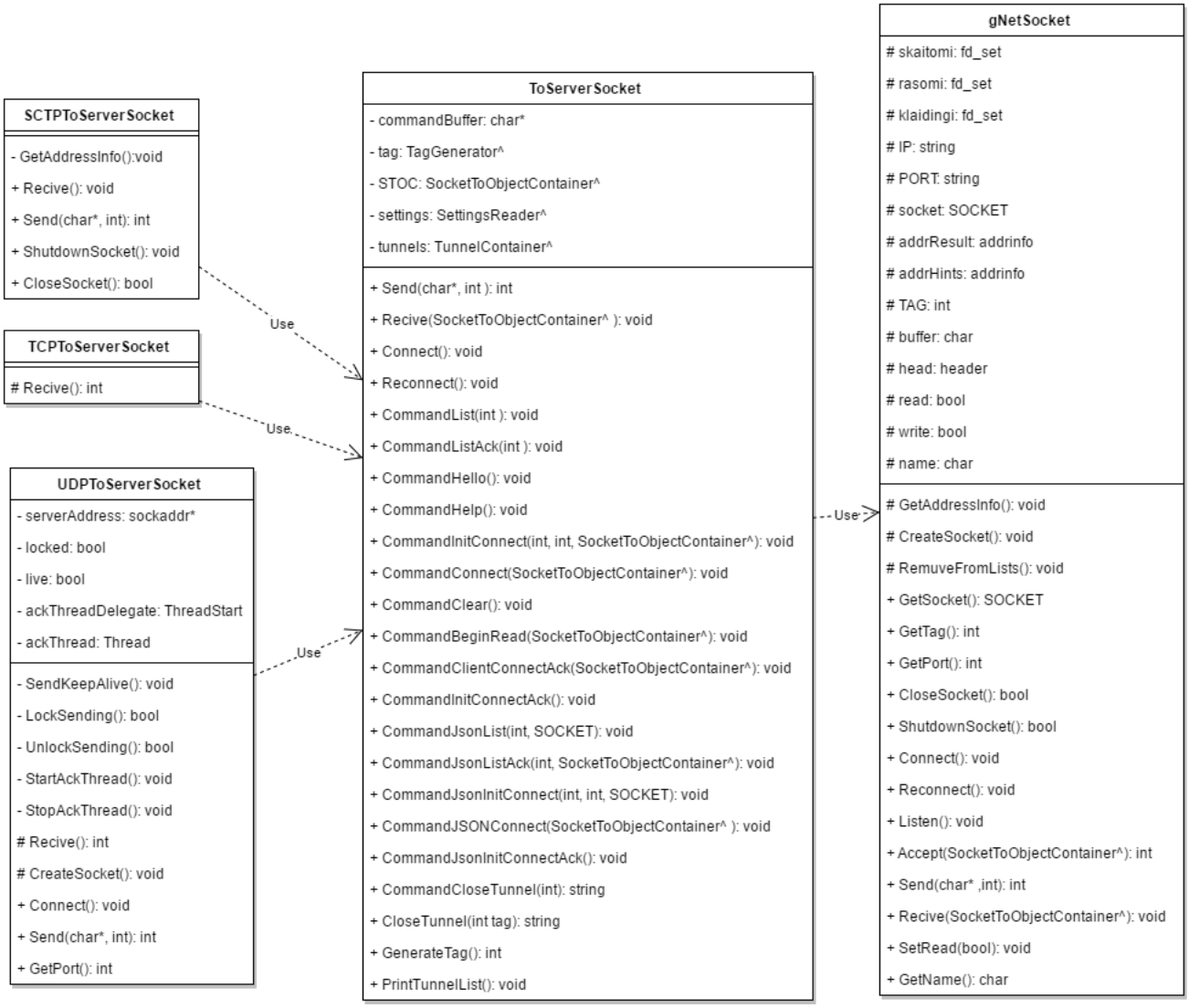
8 PRIEDAI



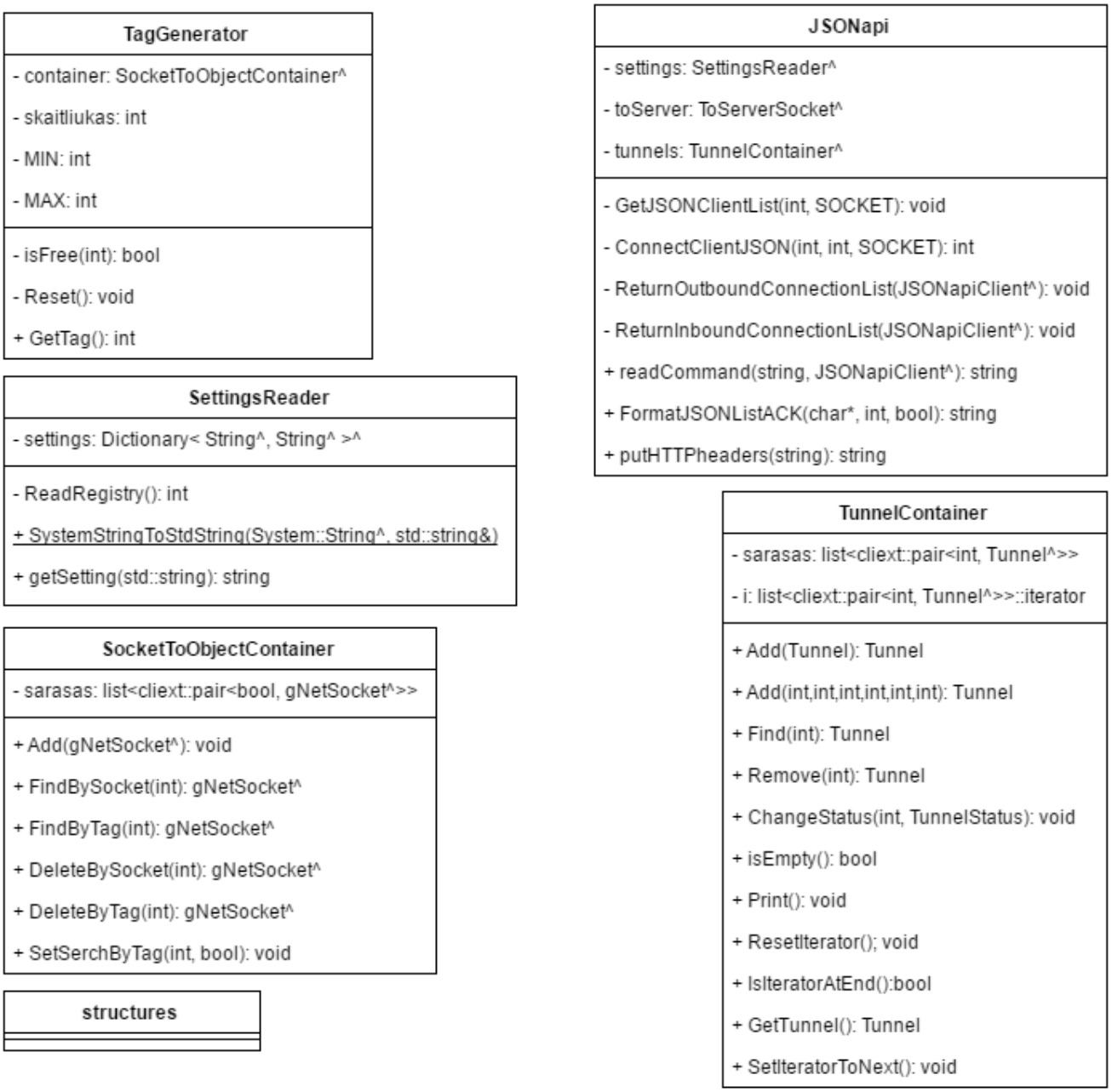
58 pav. Centrinio serverio programinės įrangos statinis vaizdas, 2 dalis



59 pav. Centrinio serverio programinės įrangos statinis vaizdas, 3 dalis



60 pav. Kliento programinės įrangos statinis vaizdas. 2 dalis



61 pav. Kliento programinės įrangos statinis vaizdas. 3 dalis

13 lentelė Europos sąjungos narių gyvenamųjų narių prijungtų prie interneto duomenys, procentais. Pagal Eurostat duomenis

Šalis/ Metai	2006	2007	2008	2009	2010	2011	2012	2013	2014	2015
Europos sąjungos vidurkis	51,0%	57,0%	62,0%	67,0%	71,0%	74,0%	76,0%	79,0%	81,0%	83,0%
Belgium	54,0%	60,0%	64,0%	67,0%	73,0%	77,0%	78,0%	80,0%	83,0%	82,0%
Bulgaria	17,0%	19,0%	25,0%	30,0%	33,0%	45,0%	51,0%	54,0%	57,0%	59,0%
Czech Republic	29,0%	35,0%	46,0%	54,0%	61,0%	67,0%	65,0%	73,0%	78,0%	79,0%
Denmark	79,0%	78,0%	82,0%	83,0%	86,0%	90,0%	92,0%	93,0%	93,0%	92,0%
Germany	67,0%	71,0%	75,0%	79,0%	82,0%	83,0%	85,0%	88,0%	89,0%	90,0%
Estonia	45,0%	52,0%	57,0%	62,0%	67,0%	69,0%	74,0%	79,0%	83,0%	88,0%
Ireland	50,0%	57,0%	63,0%	67,0%	72,0%	78,0%	81,0%	82,0%	82,0%	85,0%
Greece	23,0%	25,0%	31,0%	38,0%	46,0%	50,0%	54,0%	56,0%	66,0%	68,0%
Spain	38,0%	43,0%	50,0%	53,0%	58,0%	63,0%	67,0%	70,0%	74,0%	79,0%
France	41,0%	55,0%	62,0%	69,0%	74,0%	76,0%	80,0%	82,0%	83,0%	83,0%
Croatia	0,%	41,0%	45,0%	50,0%	56,0%	61,0%	66,0%	65,0%	68,0%	77,0%
Italy	40,0%	43,0%	47,0%	53,0%	59,0%	62,0%	63,0%	69,0%	73,0%	75,0%
Cyprus	37,0%	39,0%	43,0%	53,0%	54,0%	57,0%	62,0%	65,0%	69,0%	71,0%
Latvia	42,0%	51,0%	53,0%	58,0%	60,0%	64,0%	69,0%	72,0%	73,0%	76,0%
Lithuania	35,0%	44,0%	51,0%	60,0%	61,0%	60,0%	60,0%	65,0%	66,0%	68,0%
Luxembourg	70,0%	75,0%	80,0%	87,0%	90,0%	91,0%	93,0%	94,0%	96,0%	97,0%
Hungary	32,0%	38,0%	47,0%	53,0%	58,0%	63,0%	67,0%	70,0%	73,0%	76,0%
Malta	53,0%	54,0%	59,0%	64,0%	70,0%	75,0%	77,0%	79,0%	81,0%	82,0%
Netherlands	80,0%	83,0%	86,0%	90,0%	91,0%	94,0%	94,0%	95,0%	96,0%	96,0%
Austria	52,0%	60,0%	69,0%	70,0%	73,0%	75,0%	79,0%	81,0%	81,0%	82,0%
Poland	36,0%	41,0%	48,0%	59,0%	63,0%	67,0%	70,0%	72,0%	75,0%	76,0%
Portugal	35,0%	40,0%	46,0%	48,0%	54,0%	58,0%	61,0%	62,0%	65,0%	70,0%
Romania	14,0%	22,0%	30,0%	38,0%	42,0%	47,0%	54,0%	58,0%	61,0%	68,0%
Slovenia	54,0%	58,0%	59,0%	64,0%	68,0%	73,0%	74,0%	76,0%	77,0%	78,0%
Slovakia	27,0%	46,0%	58,0%	62,0%	67,0%	71,0%	75,0%	78,0%	78,0%	79,0%
Finland	65,0%	69,0%	72,0%	78,0%	81,0%	84,0%	87,0%	89,0%	90,0%	90,0%
Sweden	77,0%	79,0%	84,0%	86,0%	88,0%	91,0%	92,0%	93,0%	90,0%	91,0%
United Kingdom	63,0%	67,0%	71,0%	77,0%	80,0%	83,0%	87,0%	88,0%	90,0%	91,0%

14 lentelė Europos sąjungos valstybių narių įmonių nutolusių darbuotojų procentas, 2006 metai, Eurostat duomenimis

	Norway	**Iceland**	**United Kingdom**	**Sweden**	**Finland**	**Slovakia**	**Slovenia**
Mažos įmonės (10-49 darbuotojai)	89,80%	85,71%	53,06%	69,39%	48,98%	24,49%	46,94%
Vidutinės įmonės (50-249 darbuotojai)	31,20%	26,80%	19,60%	23,60%	22,40%	6,80%	12,80%
Didelės įmonės (250 ir daugiau darbuotojų)	37,60%	26,40%	31,60%	33,60%	30,80%	13,60%	26,00%
	Romania	**Portugal**	**Poland**	**Austria**	**Netherlands**	**Hungary**	**Luxembourg**
Mažos įmonės (10-49 darbuotojai)	12,24%	14,29%	6,12%	32,65%	59,18%	16,33%	32,65%
Vidutinės įmonės (50-249 darbuotojai)	3,60%	8,40%	3,20%	14,80%	22,40%	6,40%	10,00%
Didelės įmonės (250 ir daugiau darbuotojų)	8,00%	19,60%	6,00%	25,60%	34,00%	14,40%	26,40%
	Lithuania	**Latvia**	**Cyprus**	**Italy**	**Spain**	**Greece**	**Ireland**
Mažos įmonės (10-49 darbuotojai)	22,45%	10,20%	20,41%	4,08%	10,20%	28,57%	40,82%
Vidutinės įmonės (50-249 darbuotojai)	5,20%	4,80%	11,20%	2,80%	6,80%	10,00%	15,20%
Didelės įmonės (250 ir daugiau darbuotojų)	12,00%	10,80%	24,80%	9,20%	16,00%	20,80%	23,60%
	Estonia	**Germany**	**Denmark**	**Czech Republic**	**Bulgaria**	**Belgium**	**Europos sąjungos vidurkis**
Mažos įmonės (10-49 darbuotojai)	36,73%	30,61%	93,88%	30,61%	18,37%	42,86%	26,53%
Vidutinės įmonės (50-249 darbuotojai)	13,60%	15,60%	32,40%	12,40%	4,00%	20,00%	12,00%
Didelės įmonės (250 ir daugiau darbuotojų)	21,20%	26,00%	38,00%	19,20%	6,80%	28,40%	22,00%

15 lentelė Vietinio tinklo greičio priklausomybė nuo duomenų kiekio. Greitis matuojamas Kb/s

Laikas, s	Duomenų kiekis 1 baitas	Duomenų kiekis 10 baitų	Duomenų kiekis 100 baitų	Duomenų kiekis 1000 baitų	Duomenų kiekis 10000 baitų	Duomenų kiekis 65535 baitų	Duomenų kiekis 100000 baitų
0	0	0	0	0	0	0	0
1	2	17	159	1 528	19 520	103 283	156 000
2	1	10	102	1 024	9 280	67 108	100 800
3	1	10	101	1 024	10 000	67 108	99 200
4	1	10	102	920	10 240	67 108	94 400
5	1	10	102	1 024	9 840	67 108	102 400
6	1	10	98	1 024	10 240	67 108	95 200
7	1	10	102	1 040	9 840	67 108	102 400
8	1	10	99	1 040	10 240	67 108	102 400
9	1	10	102	1 040	10 240	67 108	97 600

10	1	10	102	936	7 680	67 108	102 400
11	1	10	102	1 032	10 240	67 108	97 600
12	1	10	104	1 000	10 320	67 632	102 400
13	1	10	100	1 032	9 840	67 108	102 400
14	1	10	103	1 024	10 240	67 108	98 400
15	1	10	105	1 008	9 840	67 108	102 400
16	1	10	98	1 024	10 240	67 108	90 400
17	1	10	103	1 008	10 240	66 584	103 200
18	1	10	102	1 024	10 000	71 826	104 000
19	1	10	101	984	10 240	62 389	100 000
20	1	10	102	992	10 240	67 108	104 000
21	1	10	98	1 008	9 760	67 108	103 200
22	1	9	102	992	10 320	67 108	99 200
23	1	10	98	1 024	10 640	67 108	102 400
24	1	10	84	1 016	10 560	67 108	100 800
25	1	10	102	1 024	10 240	63 962	102 400
26	1	10	99	960	10 480	67 108	90 400
27	1	10	104	1 000	10 000	67 108	100 800
28	1	10	91	1 032	9 920	67 108	101 600
29	1	9	81	1 000	10 320	67 108	92 800
30	1	9	103	1 024	9 840	67 108	101 600
31	1	10	118	1 008	10 320	67 108	102 400
32	1	10	84	984	10 000	67 108	93 600
33	1	9	102	920	10 320	67 108	102 400
34	1	10	94	1 008	10 240	67 108	96 800
35	1	10	99	952	9 920	67 632	103 200
36	1	10	103	1 016	10 240	67 108	99 200
37	1	10	102	880	10 000	67 108	98 400
38	1	8	102	1 024	9 760	67 108	103 200
39	1	10	103	936	10 640	67 108	99 200
40	1	10	102	1 008	8 240	67 108	76 000
41	1	10	100	1 032	10 080	67 108	102 400
42	1	10	58	984	10 160	67 108	99 200
43	1	9	73	1 016	7 840	67 108	102 400
44	1	9	90	1 024	10 160	67 108	98 400
45	1	10	102	1 000	9 840	67 108	102 400
46	1	10	102	1 032	9 600	67 108	99 200
47	1	10	102	1 008	10 000	67 108	102 400
48	1	10	102	1 024	9 840	67 108	102 400
49	1	9	93	1 024	9 120	67 108	99 200
50	1	10	102	976	9 840	67 108	102 400
51	1	10	103	1 024	9 120	67 108	102 400
52	1	10	104	952	9 280	61 341	98 400
53	1	10	103	1 024	9 920	67 108	102 400
54	1	10	103	1 000	10 160	67 108	99 200
55	1	10	102	936	10 000	67 108	102 400

56	1	10	99	1 024	8 960	67 108	97 600
57	1	10	98	992	8 240	67 108	91 200
58	1	10	86	1 032	9 920	67 108	92 000
59	1	10	102	992	10 240	67 108	96 800