

ŠIAULIŲ UNIVERSITETAS  
MATEMATIKOS IR INFORMATIKOS FAKULTETAS  
INFORMATIKOS KATEDRA

Evaldas Papaurėlis  
Informatikos specialybės magistratūros II kurso neakivaizdinio skyriaus studentas

## **INTERNETO PASLAUGOS KOKYBĖS GERINIMO TYRIMAS**

MAGISTRO DARBAS

Darbo vadovas:  
Prof. L. Sakalauskas

Recenzentas:  
Doc. V. Sirius

Šiauliai, 2006/2007 m.m.

# Turinys

Ivadas.....	3
Teorinė dalis.....	4
Temos analizė.....	4
Darbinės srities analizė.....	5
Reikalavimai kompiuteriniams tinklams.....	5
Našumas.....	5
Patikimumas ir saugumas.....	7
Išplėtimas ir augimas.....	8
Skaidrumas.....	8
Įvairių srauto tipų palaikymas.....	9
Valdymas.....	10
Suderinamumas.....	11
Aptarnavimo kokybė.....	11
Komutatoriai ir jų veikimas.....	13
Darbinės srities modelis.....	14
Projektinė dalis.....	17
Įrankių ir priemonių pasirinkimo analizė.....	17
Projekto (darbo) vykdymo planas.....	18
Pradinis projekto aprašymas.....	19
Pradiniai duomenys.....	19
Duomenų bazės struktūra.....	20
Lentelės.....	20
Darbo eigos aprašymas.....	22
Darbo eigos grafas.....	22
Problemų ir jų sprendimų aprašymai ir pagrindimai.....	22
Galutinio projekto stovio aprašymas.....	24
Galutinė duomenų bazė.....	25
Papildomai atsiradusios lentelės.....	25
Funkcijos.....	26
Sistemos dinaminis vaizdas.....	27
Darbo veikimo modelis.....	28
Vykdomas failas.....	28
Galutinis darbo rezultatas.....	29
Darbo rezultatų analizė.....	30
Išvados.....	31
Literatūros sąrašas.....	32
Anotacija.....	33

## Ivadas

Per paskutinius metus yra ženkliai išaugęs interneto vartotojų skaičius, kuris ženkliai auga ir dabar. Didžiausios šalyje tyrimų bendrovės „TNS Gallup“ atliktas tyrimas parodė, kad per paskutinius metus kompiuterius turinčių šalies gyventojų skaičius ir prie interneto prijungtų kompiuterių skaičius išaugo po 18 procentų, o interneto sklaida šių metų pradžioje siekė beveik 40%.

Jau kuris laikas personalinis kompiuteris Lietuvos gyventojų namuose nėra prabanga. Jis jau senai yra naudojamas kaip darbo priemonė darbovietėse, ir vis labiau plinta namuose kaip laisvalaikio praleidimo forma. Be tradicinio interneto puslapių vartymo ieškant informacijos, bendravimo pokalbių ar pažinčių svetainių, kompiuterinių žaidimų žaidimo, muzikos klausimo ar filmų žiūrėjimo į namus vis smarkiau skverbiasi IP telefonija ir IP televizija. Vartotojų, kuriems kompiuteris yra kasdieninė darbo ar pramogų dalis, skaičius kasdien auga. Kompiuterių skverbimasis į žmonių kasdienybę skatina kompiuterinės technikos ir interneto paslaugų kainų mažėjimas.

Iš viso stalo ar nešiojamus kompiuterius 2007 metų pradžioje savo namuose turėjo 43% šalies gyventojų ir per metus šis skaičius padidėjo 8 procentiniais punktais. Tuo pačiu laikotarpiu praėjusiais metais kompiuterius turėjo kiek daugiau nei trečdalis gyventojų.

Šių metų pradžioje didelė dalis Lietuvos gyventojų turimų kompiuterių buvo prijungti prie interneto - net 7 iš 10 visų kompiuterių turėjo interneto ryšį. Lyginant pateiktuosius duomenis su tuo pačiu laikotarpiu praėjusiais metais, prie interneto prijungtų kompiuterių skaičius išaugo daugiau kaip 13 procentinių punktų.

Interneto aktualumas žmonėms kasdieniame gyvenime didėja. Šiais metais daugiau nei trims ketvirtadaliams (76%) vartotojų naršymas internete buvo pagrindinė jų veikla naudojantis kompiuteriu. Praėjusiais metais šis skaičius siekė 68 %.

Daugėja ir tų žmonių, kurie internetu naudojami nuolat – 2006 metų žiemą bent kartą per savaitę internetu naudojami beveik 28% šalies gyventojų, o šių metų žiemą nuolatinių interneto vartotojų skaičius sudarė 32%. Apskritai internetu naudojami beveik 40% šalies gyventojų.

Augant vartotojų skaičiui neišvengiamai auga ir tinklo apkrautumas. Kuo daugiau apkrautas tinklas, tuo daugiau kyla sunkumų jį prižiūrėti ir aptarnauti. Vartotojų poreikiai yra dideli ir nuolat keičiasi, todėl tinklas turi dirbti stabiliai ir nepriekaištingai. Klientų poreikiams patenkinti reikia labai daug pastangų ir darbo, kad būtų tinkamai atliekama tinklo priežiūra ir analizė.

# Teorinė dalis

## Temos analizė

Tiek tinklo priežiūra, tiek ir našumo analizė gaunama iš tinklo topologijos žinių. Nors tinklo administratoriai atsakingai prižiūri tinklą, bet klaidos neišvengiamos. Kitų tinklą aptarnaujančių žmonių klaidingi veiksmai gali administratorius palikti be realios tinklo topologijos schemas. Automatiškai gaudami tinklo topologijos schemą administratoriai gali užkirsti kelią problemoms. Jie gali netikėtai atrasti problemą ir jos šaknis, bet ir analizuoti tinklo darbą. Šitos žinios leidžia jiems numatyti problemas ir jų išvengti.

Nors tinklo topologijos informacija, ypač LAN lygiu, yra svarbi tinklo priežiūrai ir naudojimui, tačiau tokią informaciją yra sunku gauti. Dauguma tinklo priežiūros įrankių pasikliauja tiksliai IP lygio maršrutizatorių radimu, ir reikalauja tinklo administratorių įtraukti antro lygio įrenginius, tokius kaip Ethernet komutatoriai rankiniu būdu.

Sudėtingas tinklo topologijos radimas atsirado iš Ethernet komutatoriams būdingo permatomumo. Galiniai taškai nieko nežino apie komutatorių būvimą tinkle. Komutatoriai bendrauja tiksliai su savo kaimynais, specialiais protokolais ir jis nėra naudojamas įvairiose aplinkose. Vienintelis komutatorių pranašumas - tai jų peradresavimo duomenų bazė (FDB, forwarding database), kuri yra naudojama ateinančio srauto nukreipimui į tam tikrą jungtį. Laimei ši informacija yra lengvai prieinama ir pakankama topologijos analizei ir radimui.

Darbo tikslas yra sukurti sistemą kuri pasinaudodama komutatorių FDB duomenų lentelėmis atvaizduotų tikslų šių komutatorių išsidėstymą tinkle. Gata tokia tinklo komutatorių topologijos schema galima:

- Palengvinti tinklo plėtimo ar rekonstrukcijos planavimą;
- Matyti esančias tinklo topologijos klaidas atsiradusias statant tinklą ir jas pašalinti;
- Užkirsti kelias galimoms problemoms;
- Užtikrinti gera naujos paslaugo funkcionavimą;
- Padėti gerinti tinklo aptarnavimo kokybę šalinant atsiradusius gedimus;

## Darbinės srities analizė

### Reikalavimai kompiuteriniams tinklams

Pagrindinis reikalavimas tinklams yra kokybiškų paslaugų rinkinio, kuriam yra skirtas tinklas, tiekimas. Kaip pavyzdžiui: interneto paslaugos, leidimas naudotis failų archyvais (FTP), internetinių puslapių talpinimas, elektroninio pašto paslaugos, IP telefonija, IP televizija ir t.t.

Visi likusieji reikalavimai – našumas, patikimumas, suderinamumas, valdymas, apsauga, plėtra ir apimtys susijusios su kokybišku pagrindinės užduoties atlikimu. Nors visi šie reikalavimai yra svarbūs, dažnai supratimas „kokybiškas aptarnavimas“ (Quality Of Service, QOS) kompiuteriniuose tinkluose traktuojama daug siauriau – į jį įskaitomos tik dvi pačios svarbiausios tinklų charakteristikos – galingumas ir patikimumas.

### Našumas

Potencialiai aukštas našumas – tai vienas iš pagrindinių skirstomųjų tinklų galimybių, kuriems priklauso kompiuteriniai tinklai. Ta ypatybė principinga, bet gaila ne visada praktiškai realizuojama galimybė padalinti darbus tarp kelėtos kompiuterių tinklų.

Egzistuoja keletas pagrindinių tinklo našumo charakteristikų:

- reakcijos laikas;
- duomenų perdavimo greitis;
- perdavimo vėlinimas ir jo variacija.

*Tinklo reakcijos laikas* yra integralinės charakteristikos našumas iš vartotojo pusės. Šia charakteristiką turi galvoje sakydami: šiandien tinklas dirba lėtai.

Bendrai, reakcijos laikas yra nustatomas kaip laiko intervalas, per kurį tinklų tarnybos atsako į vartotojo užklausas. Šio rodiklio reikšmė priklauso nuo tarnybos į kurią vartotojas kreipiasi, į kokį serverį kreipiasi, kokie tinklo įrenginiai stovi tarp vartotojo ir serverio į kurią kreipiasi (maršrutizatoriai, komutatoriai, tiltai, keitikliai ir t.t.). Tinklo apkrautumas labai priklauso nuo dienos meto.

Tinklo reakcijos laikas paprastai susideda iš kelių sudedamųjų dalių. Bendrai į jį įeina užklausų paruošimo laikas vartotojo kompiuteryje, užklausos perdavimo laikas tarp vartotojo ir serverio, per visus tarpinius tinklo įrenginius, užklausos apdorojimas serveryje, užklausos atsakymo

laikas nuo serverio iki vartotojo, ir vartotojo kompiuterio atsakymo apdorojimas. Vartotojui yra svarbus galutinis rezultatas, o tinklo administratoriams yra svarbus reakcijos laikas tarp įrenginių kuriuos jie administruoja.

Žinant reakcijos laiką tarp tinklo įrengtinių, galima įvertinti tinklo našumą tarp atskirų tinklo segmentų, atrasti „siauras“ vietas ir modernizuoti tinklą, taip didinant jo našumą.

Duomenų perdavimo greitis atspindi duomenų kiekį, perduota tinklu arba jo dalimi, per laiko vienetą. Duomenų perdavimas matuojamas bitais per sekundę arba paketais per sekundę. Duomenų pralaidumas gali būti momentinis, maksimalus ir vidutinis.

Vidutinis greitis skaičiuojamas dalijant bendrą perduotą duomenų kiekį iš laiko, per kurį buvo perduoti duomenys, be to parenkamas ganėtinai ilgas laiko tarpas – valanda, para, savaitė.

Momentinis greitis skiriasi nuo vidutinio tuo, kad vidurkiui išrenkamas labai mažas laiko tarpas 10 ms arba 1 s.

Maksimalus greitis – tai daugiau momentinis duomenų pralaidumas užfiksuotas stebėjimo periodu. Maksimaliai pasiektas duomenų perdavimas vadinamas duomenų pralaidumu tinkle.

Projektuojant, derinant ir optimizuojant tinklus dažniausiai naudojami tokie rodikliai kaip: vidutinis ir optimalus duomenų pralaidumas. Vidutinis pralaidumas leidžia vertinti tinklo pralaidumą dideliu laiko tarpu, kuriam esant patenka ir piko metas kada tinklas labiausiai apkrautas. Taip kompensuoja darbo intensyvumai vienas kitą. Duomenų pralaidumas leidžia vertinti tinklo sugebėjimą dirbti piko apkrovomis vietiniame tinkle (ryte, kai visi darbuotojai registruojasi darbui tinkle, tikrinasi pašta, jungiasi prie duomenų bazių), arba globaliame (vakare kai dauguma grįžta namo ir sėda dirbti prie kompiuterių).

Perdavimo greitį galima išmatuoti tarp bet kurių dviejų tinklo mazgų, tarp kliento kompiuterio ir serverio, tarp dviejų komutatorių ar kitų tinklo įrenginių. Tinklo analizei ir derinimui labai svarbu žinoti atskirų tinklo dalių pralaidumą. Didinant perdavimo greitį, reikia pirmiausia modernizuoti prasčiausiais tinklo vietas, tik po to didinti bendrą pralaidumą. Kartais pravartu operuoti bendru tinklo pralaidumu, kuris nustatomas kaip maksimalus perduotos informacijos kiekis, perduotas tarp visų tinklo mazgų per laiko vienetą. Šis rodiklis charakterizuoja tinklų kokybę apskritai, ne diferencijuodamas jį atskirais segmentais ar įranga.

Perdavimo vėlinimas nustatomas kaip sulaikymas tarp momento duomenų atėjimo į koki nors tinklo įrenginį ar tinklo dalį ir duomenų išėjimo momento iš įrenginio ar tinklo dalies. Šis parametras artimas tinklo reakcijos laikui, bet skiriasi tuo, kad visada charakterizuoja duomenų perdavimo tinkle etapus. Paprastai tinklų kokybė charakterizuojama maksimaliu perdavimo vėlinimu. Ne visi srauto tipai yra jautrūs perdavimo vėlinimui, kuris charakteringas kompiuteriniams tinklams – paprastai vėlinimas neviršija 100 ms, o rečiau keletos milisekundžių.

Tokios paslaugos kaip IP telefonija ar video vaizdų perdavimas labai reaguoją į vėlinimą. Taip atsiranda garso iškraipymai, aido efektas, ar vaizdo drebėjimas.

Pralaidumo galimybės ir vėlinimas - nepriklausomi parametrai, todėl tinklas gali pasižymėti dideliu pralaidumu, bet turėti žymius vėlinimus.

### **Patikimumas ir saugumas**

Svarbu skirti keletą patikimumo aspektų. Dėl techninių įrenginių patikimumo naudojami tokie rodikliai, kaip vidutinis išdirbimo laikas iki gedimo, gedimo tikimybė, gedimų intensyvumas. Šie patikimumo vertinimo rodikliai tinkami tikrai paprastiems įrenginiams, kurie gali būti tik dvejuose stoviuose – veikiantys arba neveikiantys. Sudėtingos sistemos, sudarytos iš daugelio elementų, išskyrus paminėtus stovius, gali turėti ir kitų tarpinių stovių, kurių šios charakteristikos neįskaityti. Tokių sistemų patikimumo įvertinimui naudojami charakteristikų rinkiniai.

Pasirengimas arba pasirengimo koeficientas, reiškia laiko dalį, per kurią sistema gali būti panaudojama. Pasirengimas gali būti padidintas dubliuojant tinklo įrenginius taip, kad vienam sugedus sistemos funkcionavimas būtų perduotas kitam įrenginiui.

Kad kompiuterinė sistema turėtų aukštą patikimumą, ji turi turėti mažiausiai aukštą pasirengimą, bet to neužtenka. Būtina užtikrinti duomenų apsaugą nuo praradimo. Pasirengimo atveju turi būti vienodi duomenys saugomi keliuose serveriuose, ir jie laikas nuo laiko turi būti suvienodinami.

Taip kaip tinklas dirba paketų perdavimo principu, iš vieno mazgo į kitą mazgą, tai viena iš charakteringų patikimumo apibūdinimų yra paketo pristatymas mazgui be iškraipymų. Šalia šios charakteristikos gali būti naudojami ir kiti rodikliai: paketų pametimo galimybė (dėl įvairių priežasčių – dėl maršrutizatoriaus buferio perpildymo, dėl kontrolinės sumos nesutapimo, dėl tinklo dalies nedarbingumo ir t.t.), tikimybė perduodamų duomenų atskiro bito iškraipymo, santykis pamestų ir pristatytų paketų.

Kitu aspektu bendro patikimumo yra apsauga - tai gebėjimas apsaugoti sistemą nuo nesankcionuoto prisijungimo. Paskirstytoje sistemoje tai padaryti daug sunkiau negu centralizuotoje. Tinkluose duomenys dažnai eina ryšių linijomis, kurios dažniausiai eina per bendras patalpas, kuriose gali būti sumontuotos duomenų sekimo įrangos. Kita rizikinga vieta gali būti palikti be priežiūros asmeniniai kompiuteriai. Jei tinklas turi išėjimą į globalų tinklą, visada yra išsilaužymo pavojus.

Dar viena naudingumo charakteristika yra gedimo tolerancija. Pavyzdžiui, kad vartotojas nepastebėtų vieno serverio (jei jis turi jį pavaduojantį serverį) sustojimo. Taip gali tikrai padidėti atsako laikas, bet sistemos funkcionavimas nenutrunka.

## Išplėtimas ir augimas

Terminai, išplėtimas ir augimas, kai kada naudojami kaip sinonimai, bet tai netikslu, kiekvienas iš jų turi tiksliai nustatytą reikšmę.

Išplėtimas (extensibility) - tai reiškia galimybę lengvo papildymo atskirų tinklo elementų (vartotojų, kompiuterių, tarnybų), tinklo segmentų augimo, esamos aparatūros pakeitimą galingesne. Kartais gali būti, kad lengvas sistemos plėtimas galimas tikrai labai ribotoje erdvėje. Kaip pavyzdžiui ethernet tinklas sujungtas koaksialiniu kabeliu lengvai leidžia plėsti tinklą todėl, kad nesudėtinga prijungti naujus mazgus. Bet tokie tinklai gali turėti ribotą skaičių mazgų. Esant tokiam ribojimui atsiranda blogas tinklo augimas esant gerai plėtimo galimybei.

Augimas (scalability) reiškia, kad tinklą galima plėsti ir didinti labai plačiose ribose, kai nenukenčia tinklo darbo kokybė. Geru tinklo augimu pasižymi daugiasegmentiniai tinklai, turintys hierarchinę struktūrą naudojant komutatorius, maršrutizatorius. Tokie tinklai gali įjungti keletą tūkstančių kompiuterių ir tuo pačių metu aprūpinti kiekvieną tinklo vartotoją kokybišku aptarnavimu.

## Skaidrumas

Tinklų skaidrumas pasiekiamas tuo atveju, kai tinklas pateikiamas vartotojui ne kaip atskirų kompiuterių visuma, tarpusavyje sujungtais sudėtinga kabėlių sistema, o kaip viena tradicinė skaičiavimo mašina su laiko skirstymo sistema. Žinomas kompanijos Sun Microsystems šūkis – „tinklas tai kompiuteris“, sako kaip tik apie tokį skaidrų tinklą.

Skaidrumas gali būti pasiektas dviejuose skirtinguose lygiuose: vartotojo ir programuotojo. Vartotojo lygyje skaidrumas reiškia, kad darbui su nutolusiais resursais jis naudoja tas pačias komandas ir jam įprastas procedūras, kurias naudoja ir darbui su lokaliniais resursais. Programiniame lygyje skaidrumas pasireiškia tuo, kad priedu priėjimui prie nutolusių resursų naudojami tie patys kreipiniai, kaip ir priėjimui prie lokalių resursų. Skaidrumas vartotojo lygyje pasiekiamas paprasčiau todėl, kad visos procedūrų savybės nuo vartotojo yra slepiamos programuotojų, kurie sukuria priedus.

Tinklas turi slėpti visas operacinių sistemų ypatybes ir vartotojų kompiuterių tipų skirtumus. Macintosh kompiuterio vartotojas turi turėti galimybę kreiptis į resursus palaikomus UNIX sistemos, o UNIX vartotojas turi galėti dalintis informacija su Windows sistemos vartotojais. Didžioji dalis vartotojų nenori žinoti apie vidinius failų formatus arba apie UNIX komandų sintaksę. IBM 3270 terminalo vartotojas turi turėti galimybę keistis pranešimais su personalinių kompiuterių vartotojais nesigilindami į sunkiai įsimenamus adresus.



Skaidrumo koncepcija gali būti panaudota skirtingais tinklų aspektais. Pavyzdžiui skaidrumo išdėstymas reiškia, kad iš vartotojo nereikalaujama žinių apie programų išdėstymą ir aparatinis resursus, tokių kaip procesorius, spausdintuvas, rinkmena ar duomenų bazė. Resurso vardas neturi įjungti informacijos apie jo išdėstymo vietą, tokie vardai kaip `machine1:prog.c` arba `\\ftp_serv\pub` nėra skaidrūs. Analogiška, skaidrumo perkėlimas reiškia, kad resursai turi laisvai judėti iš vieno kompiuterio į kitą be savo vardų pakeitimo. Dar vienas galimas skaidrumo aspektas yra paralelizmo skaidrumas, aiškinamas tuo, kad skaičiavimo proceso išlygiagretavimas vyksta automatiškai be programuotojo dalyvavimo, tuo pačiu sistema pati paskirsto priedų paralelines šakas procesoriams ir kompiuterių tinklams. Negalima sakyti, kad skaidrumo savybė pilnai patenkinama, tai greičiau tikslas, kurio siekia dabartinių tinklų administratoriai.

### **Įvairių srauto tipų palaikymas**

Kompiuterių tinklai iš pradžių buvo skirti vartotojų priėjimui prie bendrų kompiuterinių išteklių: spausdintuvų, rinkmenų ar kitos informacijos. Šiomis tradicinėmis kompiuterinių tinklų tarnybomis sukurtas srautas turi savo ypatumus ir tikrai skiriasi nuo pranešimų srauto telefoniniuose tinkluose arba nuo kabelinės televizijos tinklų. 90-ieji metai tapo metais kai į kompiuterinius tinklus pradėjo skverbtis multimėdinių duomenų srautas, kai vaizdas ir garsas perduodamas skaitmenine forma. Kompiuteriniai tinklai tapo naudojami video konferencijoms, mokslui ir net pramogoms. Aišku, kad dėl dinaminio multimėdijos srauto perdavimo yra reikalingi kiti algoritmai, protokolai ir atitinkamai kiti įrenginiai. Nors multimedinio srauto dalis kol kas nedidelė, ji jau pradėjo veržtis į lokalius ir globalius kompiuterinius tinklus, ir šis procesas didės su interneto greičio didėjimu.

Balso ar vaizdo dinaminio perdavimo metu atsirandančiam srautui keliami griežti reikalavimai perduodamiems sinchroniniams pranešimams. Dėl kokybiško nepertraukiamo srauto atkūrimo, kurie yra garso virpesiai arba šviesos intensyvumo pakitimas vaizde, būtina gauti išmatuotų ir užkoduotų amplitudžių signalus tuo pačiu dažniu, kuriuo jie buvo išmatuoti perdavimo pusėje. Esant pranešimų vėlavimui pastebimi iškraipymai.

Kompiuterinių duomenų srautas charakterizuojasi nevienodu pranešimų patekimo į tinklą intensyvumu esant griežtam sinchroninio tų paketų pristatymo reikalavimui. Pavyzdžiui, vartotojui dirbant su tekstu nutolusiame diske, generuojamas atsitiktinis pranešimų srautas tarp nutolusio ir lokalaus kompiuterio, priklausomai nuo vartotojo veiksmų redaguojant tekstą, tad pristatymo užlaikymai atitinkamose ribose turi mažai įtakos tinklo vartotojo aptarnavimo kokybei. Visi kompiuterinių tinklų algoritmai, atitinkami protokolai ir komutaciniai įrenginiai buvo apskaičiuoti būtent tokiam pulsuojančiam srautui, todėl būtinumas perdavinėti multimėdinį srautą reikalauja

padaryti tiek protokolų pakeitimus, tiek ir įrangos pakeitimus. Šiandien praktiškai visi nauji protokolai viename ar kitame laipsnyje turi multimedinio srauto palaikymą.

Ypatingų sunkumų sudaro tradicinio kompiuterinio ir multimedinių srautų sujungimas viename tinkle. Išskirtinai multimedinio srauto perdavimas kompiuteriniu tinklu, kad ir surištas su tam tikrais sudėtingumais, bet iššaukia mažesnius sunkumus. Atvejis, kai egzistuoja dviejų tipų srautai su priešingais reikalavimais, yra daug sudėtingesnė užduotis iš kokybiško aptarnavimo pusės. Paprastai protokolai ir kompiuterinių tinklų įranga multimedinių srautą laiko neprivalomu, todėl jo aptarnavimo kokybė turėtų būti geresnė. Dabar dedamos didelės pastangos kuriant tinklus, neužgniaužiant kiekvieno iš srautų tipų interesų.

### **Valdymas**

Tinklo valdymas suprantamas kaip galimybė centralizuotai kontroliuoti atskirų tinklo elementų būsenas, atrasti ir išspręsti problemas išskylančias tinklo darbo metu, atlikti našumo analizę ir planuoti tinklo plėtrą. Tinklų valdymo priemonės idealiai sudaro sistemą, vykdančią stebėjimą, kontrolę ir valdymą kiekvieno tinklo elemento – nuo paprasčiausių iki pačių sudėtingiausių įrenginių. Tokia sistema mato tinklą kaip vienetą, o ne kaip atskirą rinkinį skirtingų įrenginių.

Gera valdymo sistema stebi tinklą ir pastebėjusi iškilusią problemą, atlieka tam tikrus veiksmus, taiso situaciją ir praneša administratoriui apie tai, kas įvyko ir kokių veiksmų buvo imtasi. Tuo pačiu metu sistema turi rinkti duomenis, kurių pagalba galima planuoti tinklo plėtrą. Galu gale valdymo sistema turi būti nepriklausoma nuo gamintojo ir turėti patogią sąsają, leidžiančią visus veiksmus atlikti viena konsole.

Sprendžiant taktines užduotis administratoriai ir techninis personalas susiduria su kasdieninėmis tinklo darbingumo užtikrinimo problemomis. Šios užduotys reikalauja greito sprendimo, tinklą aptarnaujantis personalas turi operatyviai reaguoti į pranešimą apie gedimą, gaunamą iš vartotojų arba automatinių tinklų valdymo sistemų. Pamažu darosi pastebimos daug bendresnės našumo problemos, tinklo konfigūravimas, klaidų apdirbimas ir duomenų saugumas, reikalaujantis strateginio priėjimo, tai yra tinklo planavimo. Planavimas įtraukia vartotojų poreikių tinklui pasikeitimo prognozes, klausimus dėl naujos įrangos ar naujų technologijų panaudojimą tinkle.

Valdymo sistemos panaudojimas ypač aiškiai pasireiškia dideliuose tinkluose: korporaciniuose ar globaliuose. Be valdymo sistemos tokiuose tinkluose reikalingi kvalifikuoti specialistai, dalyvaujantys pastatų ar padalinių esančių skirtinguose miestuose eksploatacijoje, kur yra įdiegta tinklo įranga, kas visumoje priveda prie būtinybės išlaikyti milžiniškus aptarnaujančio personalo etatus.

Dabartiniu metu tinklų valdymo sistemoje yra daug neišspręstų problemų. Aiškiai nepakanka tikrai patogių, kompaktiškų, daugiaprotolinių tinklo valdymo priemonių. Daugelis priemonių visai nevaldo tinklų, o tik vykdo jų darbo stebėjimus. Jie stebi tinklą, bet neatlieka veiksmų, jei kas nors atsitinka ar gali atsitikti tinkle. Maža augančių sistemų, galinčių aptarnauti tiek skyriaus mastelio tinklą, tiek ir korporacijos masto tinklą. Didelė dauguma sistemų valdo tik atskirus tinklo elementus ir ne analizuoja tinklo sugebėjimo atlikti kokybišką informacijos perdavimą tarp galutinių vartotojų.

### **Suderinamumas**

Suderinamumas arba integravimas reiškia, kad tinklas sugeba prijungti pačius įvairiausias programinius ir aparatinius priedus, tai yra gali egzistuoti įvairios operacinės sistemos, palaikančios skirtingus komunikavimo protokolų srautus, o taip pat įvairių gamintojų įrangą ir priedus. Tinklas susidedantis iš įvairių tipų elementų yra vadinamas nevienalyčiu arba heterogeniniu, o jei heterogeninis tinklas dirba be problemų tai ji esti integruotas. Pagrindinis integruotų tinklų sudarymo kelias – modulių naudojimas, sudarytų suderinant su atvirais standartais ir specifikacijomis.

### **Aptarnavimo kokybė**

Bendrai aptarnavimo kokybė (Quality of Service, QoS) nustato galimus įvertinimus vienokių ar kitokių reikalavimų atlikimui, kuriuos pateikia tinklo priedai arba vartotojai. Pavyzdžiui perduodant balso srautą tinklu aptarnavimo kokybė dažniausiai suprantama kaip garantija to, kad balso paketai bus pristatyti su užlaikymu nedidesniu negu  $N$  milisekundžių, tuomet užlaikymo variacija neviršyti  $M$  milisekundžių, iš šios charakteristikos išsilaikys tinkle su 0,95 tikimybe nustatytame laiko intervale. Tai yra svarbu priedui, kuris perduoda balso srautą, kad tinklas garantuotų priežiūrą būtent to, aukščiau pateikto charakteristikų rinkinio aptarnavimo kokybę. Failų servisui reikalingos garantijos vidutinės juostos pralaidumo ir jos praplėtimo nedideliuose laiko intervaluose iki kai kurio maksimalaus lygio dėl greito pulsuojančio perdavimo. Tinklas turi garantuoti ypatingus tinklo aptarnavimo kokybės parametrus, suformuluotus kiekvienam atskiram tinklo priedui. Vienok dėl suprantamų priežasčių kuriami ir jau esami QoS mechanizmai ribojasi paprastesnių užduočių sprendimais, kai kurių vidutinių reikalavimų užtikrinimu, pagrindiniams priedų tipams.

Dažniausiai figūruojantys aptarnavimo kokybės apibrėžimo parametrai, reglamentuoja sekantys tinklo darbo rodikliai:

- duomenų perdavimo greitis;
- paketų perdavimo užlaikymas;
- paketų pametimo ir iškraipymo lygis.

Aptarnavimo kokybė garantuojama tik kai kuriems duomenų srautams. Priminsime, kad duomenų srautas – tai paketų eiliškumas, turinčių kai kuriuos bendrus požymius, pavyzdžiui šaltinio adresas, informacija, identifikuojanti priedo tipą (jungties numerį TSP/UDP) ir kt. Srautams naudojami tokie supratimai, kaip agregavimas ir diferencijavimas. Taip duomenų srautas iš vieno kompiuterio gali būti pristatytas kaip srautų visuma iš skirtingų tinklo priedų, kai kurių paslaugų tiekėjų srautai iš vienos korporacijos kompiuterių agreguoti į viena duomenų srautą.

Kokybiško aptarnavimo, palaikymo mechanizmai patys nesukuria pralaidumo galimybių. Tinklas negali duoti daugiau, negu turi. Faktiška ryšio ir tranzitinių komutacinių įrenginių kanalų pralaidumo galimybė – tai tinklo resursai, esantys QoS mechanizmų darbo dėmesio centre. QoS mechanizmai valdo tiksliai skirstinius turinčius pralaidumo ypatumus atitinkančius priedų reikalavimams ir tinklo derinimus. Pats žinomiausias tinklo pralaidumo paskirstymo būdas susiveda į paketų eilių valdymą.

Kadangi duomenys, kuriais keičiasi du galiniai mazgai, pereina per keletą tarpinių tinklo įrenginių, tokių kaip koncentratoriai, komutatoriai ar maršrutizatoriai, tai QoS palaikymas reikalingas visiems tinklo elementams srauto kelyje, t.y. „nuo galo iki galo“ („end-to-end“, e2e). Bet kokios QoS garantijos ant tiek geros, ant kiek jie rūpinasi „silpniausiu“ elementu grandinėje tarp siuntėjo ir gavėjo. Todėl reikia gerai suprasti, kad QoS palaikymas tiksliai viename tinklo įrenginyje, tegul ir magistraliniame, gali visai nežymiai pagerinti aptarnavimo kokybę, arba net visai nepaveikti QoS parametrų.

QoS mechanizmų palaikymo realizacija kompiuteriniuose tinkluose yra palyginti nauja tendencija. Ilgą laiką kompiuteriniai tinklai egzistavo be tokių mechanizmų, ir tai pagrinde paaiškina dvi priežastys. Pirma daugelis tinklo įrangos naudojamos tuose tinkluose buvo nereiklios. Tokiems įrenginiams paketų užlaikymas arba vidutinio pralaidumo galimybių atjungimas pakankamai plačiame diapazone, nesukeldavo pastebimo funkcionalumo praradimo.

Antra, pačios Ethernet 10 megabitų tinklo pralaidumo galimybių daugelių atveju nesudaro deficito. Taip skirstomasis Ethernet segmentas, prie kurio buvo prijungta 10-20 kompiuterių, retkarčiais kopijuojančių nedideles tekstines rinkmenas, neviršijančias kelių šimtų kilobaitų, leidžia kiekvienos poros kartu veikiančių kompiuterių srautui keliauti tinklu taip greitai, kaip to reikalauja ši srautą sudariusieji priedai.

Rezultate dauguma tinklų dirbo su tokia transportavimo aptarnavimo kokybe, kuri užtikrino priedų reikalavimus. Tiesa jokių garantijų dėl paketų sulaikymo atsiradimo arba pralaidumo galimybių, su kuria paketai perduodami tarp mazgų, nustatytų ribų, tie tinklai neduodavo. Bet to,

prie laikinų tinklo perkrovų, kada didžioji dalis kompiuterių vienu metu pradėdavo perdavinėti duomenis maksimaliu greičiu, užlaikymai ir pralaidumo galimybės darėsi tokios, kad priedų darbas davė trikdžius – veikė labai lėtai su seanso pertraukimais, ir t.t.

Transporto servisas, kurį pateikdavo tokie tinklai, gavo pavadinimą best effort, tai yra servisas „su maksimaliomis pastangomis“. Tinklas stengiasi apdirbti patenkančią srautą kaip galima greičiau, bet tuo pačiu neduoda jokių garantijų savo pastangų rezultatui. Kaip pavyzdžiai yra dauguma populiarių technologijų sukurtų 80-taisiais metais: Ethernet, Tokian Ring, IP, X.25. „Maksimalių pastangų“ servisas pagrįstas kai kuriais teisingais eilių apdirbimo algoritmais, atsirandančių tinklo apkrovimo metu, kai per tam tikrą laiko tarpą paketų patekimo į tinklą greitis viršiją jų judėjimo tinkle greitį. Paprasčiausiu atveju eilių apdorojimo algoritmas peržiūri visu srautų paketus kaip lygiaverčius ir prastumia juos gavimo tvarka (First Input First Output, FIFO). Tuo atveju kai eilė tampa labai ilga (nebetelpa buferyje), problema sprendžiama paprastu naujai ateinančių paketų atmetimu. Matyt, kad servisas „su maksimaliomis pastangomis“ aprūpina priimtina aptarnavimo kokybe tik tais atvejais, kada tinklo našumas smarkiai lenkia vidutinius reikalavimus, tai yra perteklinis. Tokia tinklo pralaidumo galimybė pakankama netgi apkrautumo piko metu srauto palaikymui. Taip pat akivaizdu, kad toks sprendimas kraštutiniu atveju yra neekonomiškas, atsižvelgiant į šiuolaikinių technologijų ir infrastruktūrų pralaidumo galimybes, ypač globaliems tinklams. Taip kaip piko apkrautumą ir sritis, kuriose jis atsiranda, sunku nuspėti tai toks kelias neduoda ilgalaikio sprendimo.

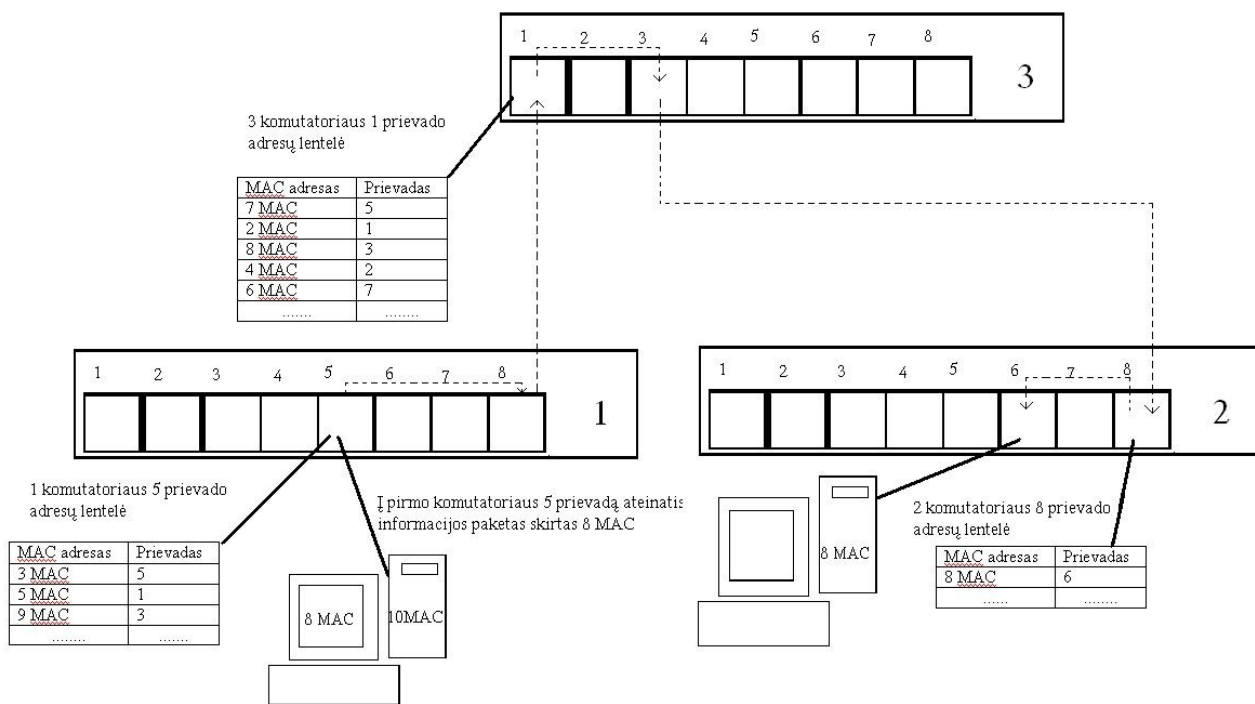
Tinklų statyba su perteklinėmis pralaidumo savybėmis, yra pats paprasčiausias būdas reikalingam aptarnavimo kokybės lygiui aprūpinimui, kartais taikomas praktikoje. Pavyzdžiui, kai kurie TCP/IP tinklo paslaugų tiekėjai duoda kokybiško aptarnavimo garantiją, pastoviai palaikydami savo magistralių nustatytą perteklinį pralaidumo savybių lygį, pagal klientų poreikius.

Sąlygomis kada dauguma aptarnavimo kokybės palaikymo mechanizmų tikrai ruošiami, perteklinio pralaidumo naudojimas dažnai tokiam tikslui pasirodo vienintelė galimybė, nors ir laikina.

### **Komutatoriai ir jų veikimas**

**Komutatorius** - tai įrenginys kuris analizuoja paketų antraščių informaciją ir pagal tai, atliekantis jos perdavimą. Kadangi komutatorius nedirba signalo kartojimo lygmenyje, net jei ta pati informacija ir persiunčiama į visus prievadus, tai reiškia, kad du įrenginiai sujungti komutatoriaus pagalba nebus toje pačioje kolizijų srityje. Kiekvienas komutatorius turi vidinę adresų perdavimo lentelę, kurioje yra įrašyti: MAC adresas, jungtis, ir laiko skaitliukas. Lentelės pagalba yra nustatoma į kurį prievadą reikia perduoti informacijos paketus, atėjusius konkrečiam fiziniam

adresui (MAC). Jei paketų gavėjo nėra lentelėje - jis perduodamas į magistralinį prievadą, kuris yra sujungtas su aukštesnio hierarchinio lygmens komutatoriumi. Kaip pavyzdys yra pateikta schema Nr.1. Kompiuteris esantis prijungtas prie pirmo komutatoriaus penkto prievado, siunčia informaciją kompiuteriui, kurio fizinis adresas yra 8 MAC. Prievadas gavęs informaciją, patikrina savo informacijos lentelę ir neradęs informacijos apie tokį gavėją, informaciją persiunčia magistraliniam aštuntam prievadui. Trečiasis komutatorius gauna informaciją iš pirmo komutatoriaus magistralinio prievado į savo pirmąjį prievadą. Patikrinęs prievado duomenų lentelę, randa informaciją, kad 8 MAC adresus turi būti perduodamas per trečiąjį prievadą, per kurį ir perduodama informacija. Toliau trečiasis komutatorius gavęs informaciją į magistralinį prievadą ir patikrinęs jo lentelę, perduoda informaciją į šeštą prievadą, prie kurio yra prijungtas informacijos gavėjas. Taip vyksta paketų perdavimai tarp komutatorių.



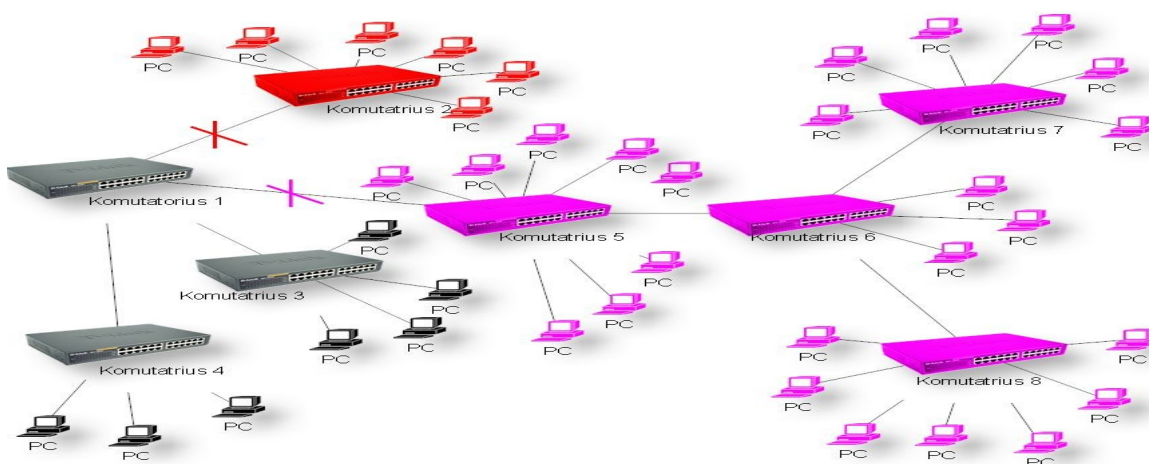
Schema Nr. 1

### Darbinės srities modelis

Dabartinėje paslaugų rinkoje egzistuoja didelė konkurencija. Tam, kad sėkmingai konkuruoti reikia užtikrinti geras ir kokybiškas paslaugas reikliems vartotojams. Interneto paslaugų tiekėjams tai didelė ir sudėtinga užduotis. Tokio tinklo aptarnavimo kokybei pagerinti reikalingos tikslios žinios apie esamą tinklo schemą. Reikia tiksliai žinoti kaip yra sujungti tarpiniai tinklo

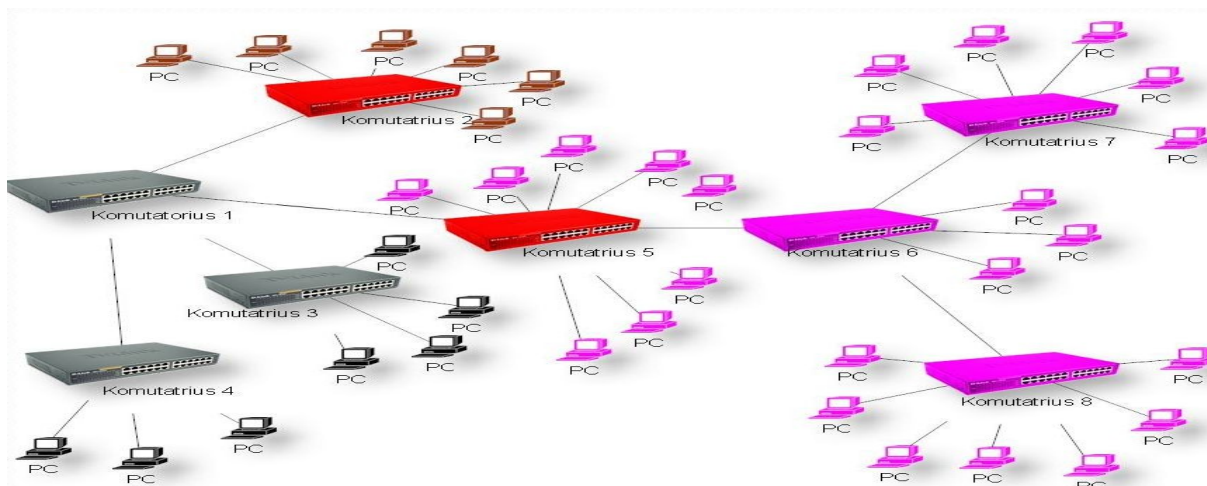
įrenginiai, šiuo atveju būtų tinklo komutatoriai. Žinant tikslią tinklo topologiją yra daug lengviau jį aptarnauti ir prižiūrėti. Tiksli topologija praverčia ieškant tinklo pažeidimų ar sugedusios tinklo įrangos vietų. Schema taip pat labai praverčia tinklo planavimui ir rekonstrukcijai diegiant naujas technologijas arba plečiant tinklo teritoriją. Turint tikslią topologiją galima iškart numatyti tinklo pažeidžiamas dalis. Kaip pavyzdžiui esant didelei tinklo komutatorių grandinei atsiranda daug tarpinių įrenginių, kur padidėja gedimų tikimybė ir tuo pačiu padidėja neveikiančio tinklo plotas.

- esant pažeistam kabeliui. Pavyzdžiui, pažeidus kabelį tarp pirmojo ir antrojo komutatoriaus (žiūrėti schemą Nr. 2), pažeidimą jaus įrenginiai ir vartotojai pažymėti raudona spalva. Pažeidus kabelį tarp pirmojo ir penktojo komutatoriaus, problemą pajaus daug didesnė dalis įrenginių ir vartotojų (pažymėti rožine spalva).



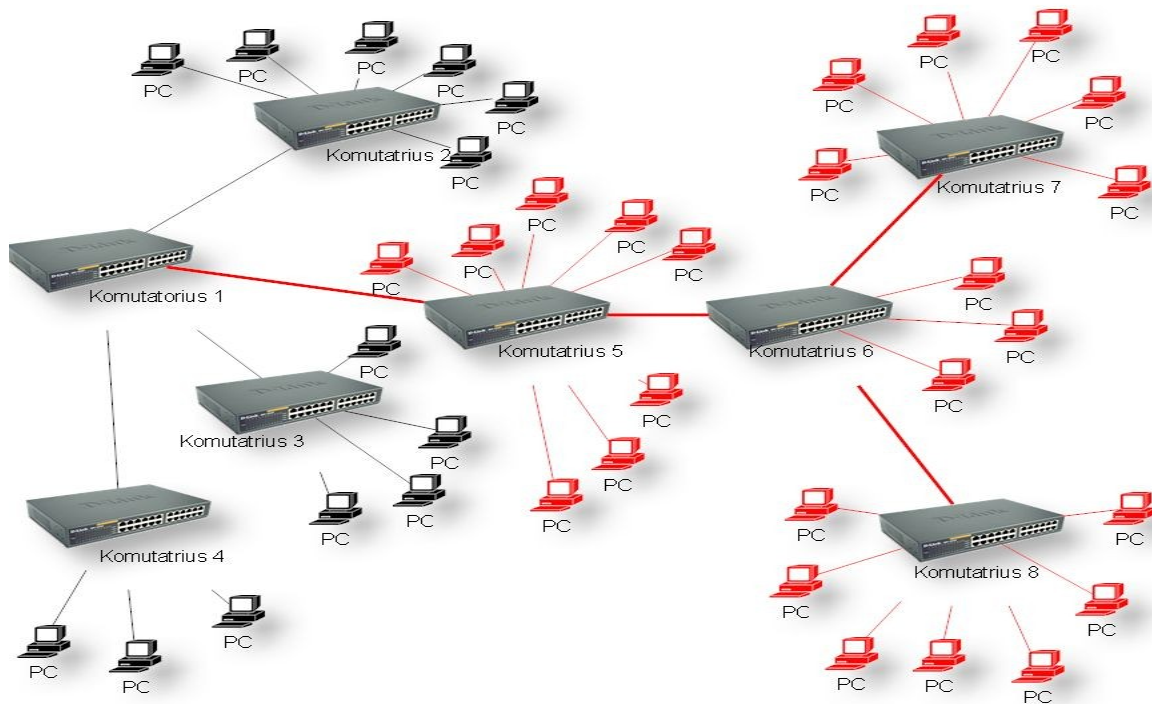
Schema Nr. 2

- sugedus vienam iš tinklo komutatorių. Gedimai gali būti: visiškas sustojimas, paketų pametimas, konfigūracijos sutrikimai ir kiti. Pavyzdžiui, sugedus antram komutatoriui, gedimas pasijaus vartotojams pažymėtiems ruda spalva, o sugedus penktam komutatoriui gedimas bus juntamas daug plačiau, pažymėta (rožine spalva). Pavyzdys pavaizduotas schemoje Nr. 3.



Schema Nr. 3

- taip pat išskyla problemų didinant vietinio tinklo pralaidumą arba pradėdant teikti naują paslaugą, tokią kaip IP televizija, kuri reikalauja didelio vietinio tinklo pralaidumo. Problemos atsiranda dėl per didelio komutatoriaus jungčių apkrovimo. Esant tokiam tinklui, kaip parodyta schemeje Nr. 4 reiktų atsižvelgti į raudonai pažymėtą tinklo atšaką ir užtikrinti, kad pirmo komutatoriaus jungtis sugebės palaikyti reikiamą apkrautumą, kad neatsirastų paketų vėlinimas.



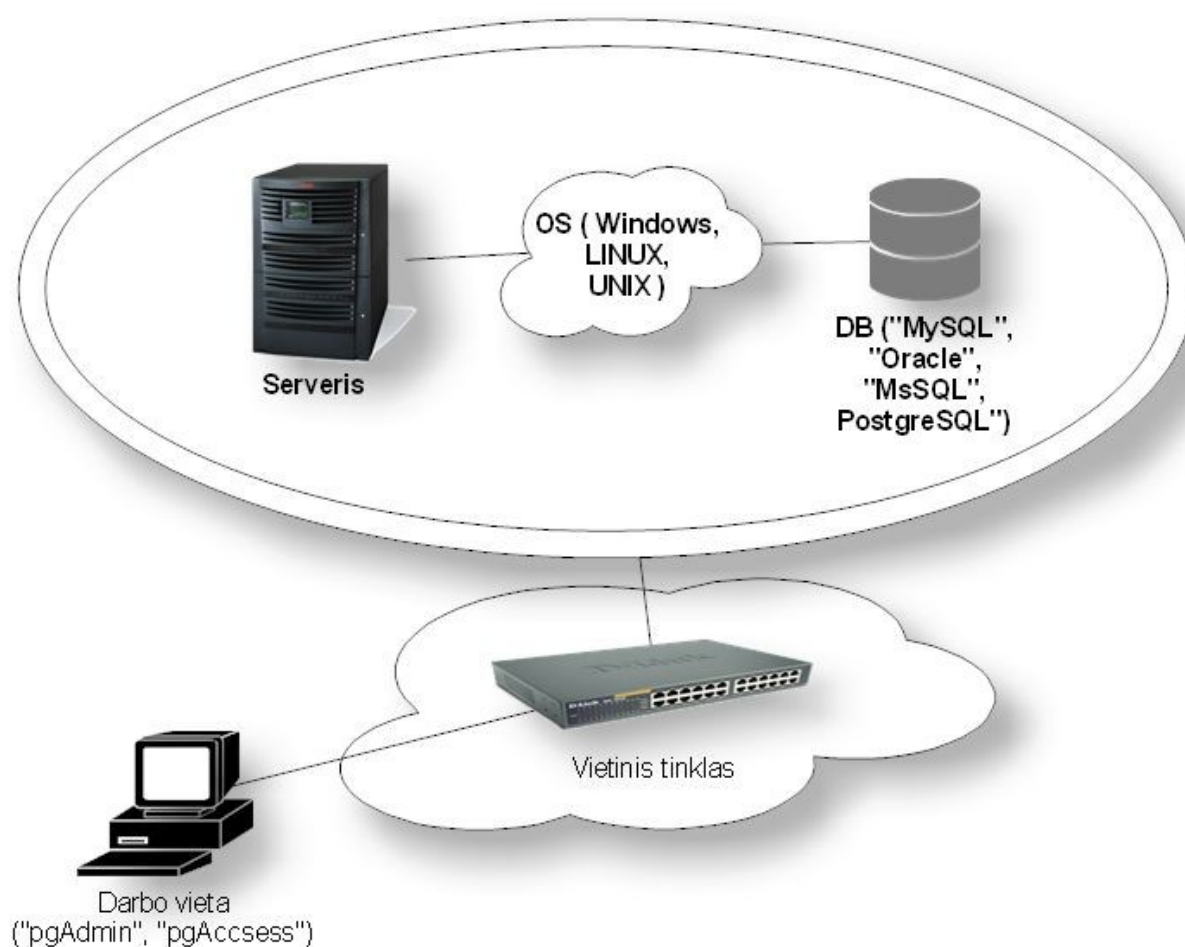
Schema Nr.4



## Projektinė dalis

### Įrankių ir priemonių pasirinkimo analizė

Šis praktinis darbas susideda iš trijų pagrindinių dalių: tai yra duomenų bazės, duomenų bazės administravimo ir grafinės priemonės. Duomenų bazėje bus saugomi: pradiniai duomenys, funkcijos duomenims apdoroti ir galutiniai duomenys. Grafinė priemonė bus naudojama galutinių duomenų grafiniam atvaizdavimui. Tokia sistema pavaizduota scheme Nr. 5.



Schema Nr. 5

Toliau bus trumpai aprašomos programos ir priemonės, kurios reikalingos užbrėžtam tikslui pasiekti. Mano pasirinkimo tikslas buvo toks, kad visos priemonės turėtų būti nemokamos.

**PostgreSQL** – atviro kodo (BSD licencija), reliacinių duomenų bazių valdymo sistema. Savo galimybėmis prilygstanti Oracle ir DB2, bet be tokių priemonių, kurios reikalingos dirbant su labai dideliais duomenų kiekiais. Ši sistema aktyviai kuriama daugiau nei 15 metų. Veikia

daugelyje operacinių sistemų, įskaitant Linux, UNIX (AIX, BSD, HP-UX, SGI IRIX, Mac OS X, Solaris, Tru64) ir Windows. Be standartinių SQL galimybių, palaiko MVCC, GiST indeksavimą, išsaugotas procedūras ir trigerius įvairiomis kalbomis (įskaitant vidinę PL/pgSQL), rafinuotą užklausų planavimą bei optimizavimą, didelių binarinių objektų laikymą bei daugybę kitų funkcijų. Turi vidinį programavimo interfeisą, pritaikytą dirbti su C, C++, Java, Perl, Python, Ruby, Tcl ir kt. Yra sukurtos ODBC tvarkyklės. Didžiąja dalimi suderinama su ANSI SQL 92/99 standartais. Priežastys nulėmusios mano pasirinkimą:

- šios duomenų bazės pilnai užtenka mano praktinio darbo poreikiams;
- nemokama reliacinė duomenų bazė;
- yra atviro kodo programa, todėl yra nuolat tobulinama;
- veikia daugelyje operacinių sistemų;
- pasižymi stabilumu.

**PL/pgSQL** –darbo kūrimui buvo naudota vidinė *PostgreSQL* užklausų kalba.

**PGAdmin** yra populiariausia ir daug savybių turinti atviro kodo programa, administruojanti PostgreSQL duomenų bazes. Ji naudoja specialias pagalbines priemones (Utilities), kurios vykdo duomenų migraciją iš kitų duomenų bazių. Priežastys nulėmusios mano pasirinkimą:

- nemokama atviro kodo sistema;
- puikiai tinka administruoti PostgreSQL duomenų bazes;
- puikiai veikia tiek Windows tiek ir LINUX (UNIX) sistemose.

**Graphviz** paketas - tai yra atviro kodo grafinio vizualizavimo programų rinkinys. Grafikus pateikia pagal duotus duomenis DOT kalba. Priežastys nulėmusios mano pasirinkimą:

- nemokama atviro kodo sistema;
- veikia įvairiose operacinėse sistemose;
- puikiai tinka atvaizduoti komutatorių tinklui.

## **Projekto (darbo) vykdymo planas**

1. išanalizuoti tinklui keliamus reikalavimus;
2. išanalizuoti tinklo komutatorių veikimo principus;
3. išgauti ir išanalizuoti duomenis, kurie yra saugomi tinklo komutatoriuose;

4. medžiagos paieška ir analizė.
5. išanalizuoti ir pasirinkti darbo priemones ir įrankius;
6. pasirinktų priemonių įdiegimas, konfigūravimas ir paleidimas;
7. suprojektuoti duomenų bazę;
8. kodo kūrimas ir jo rašymas;
9. programos bandymai ir testavimai;
10. testavimo metu iškilusių problemų ir trūkumų analizė ir šalinimas;
11. galutinio rezultato ištestavimas ir aprašymas.

## Pradinis projekto aprašymas

### Pradiniai duomenys

Pradiniais duomenimis laikysime komutatorių sąrašą, kuriame surašyti jų MAC ir IP adresai, ir komutatorių FDB lentelių duomenis, kurie yra paimami iš jų SNMP (Simple Network Management Protocol) pagalba. SNMP – vienas iš tinklo protokolų skirtas tinkle veikiančių įrenginių stebėjimui ir valdymui. Skirtingai nuo daugumos dabar naudojamų, veikia ne tik TCP/IP tinkluose. Priešingai, nei teigia pavadinimas, protokolas yra bene sudėtingiausias iš standartizuotų protokolų, turi kelis abstrakcijos lygmenis, tipų apibrėžimus ir kt. Pradiniai duomenys laikomi tekstiniuose failuose tokiu pavidalu:

- komutatorių sąrašas:

20	192.168.66.10	00:0d:88:cf:91:00
115	192.168.66.100	00:0f:3d:eb:4a:fa
116	192.168.66.101	00:0f:3d:df:5b:22
117	192.168.66.102	00:11:95:ca:07:3f
118	192.168.66.103	00:11:95:ca:07:40
119	192.168.66.104	00:11:95:ca:07:43
120	192.168.66.105	00:11:95:ca:07:5c

- FDB lentelių duomenys:

192.168.66.9	24	1	00:04:23:c7:de:0a	2007-05-03 14:01:28+03
192.168.66.9	8	4	00:11:2f:3d:1d:83	2007-05-03 14:01:28+03
192.168.66.9	24	1	00:0f:3d:86:48:57	2007-05-03 14:01:28+03
192.168.66.9	24	1	00:11:95:ca:06:bc	2007-05-03 13:31:38+03
192.168.66.9	24	1	00:11:95:ad:e1:00	2007-05-03 14:01:28+03

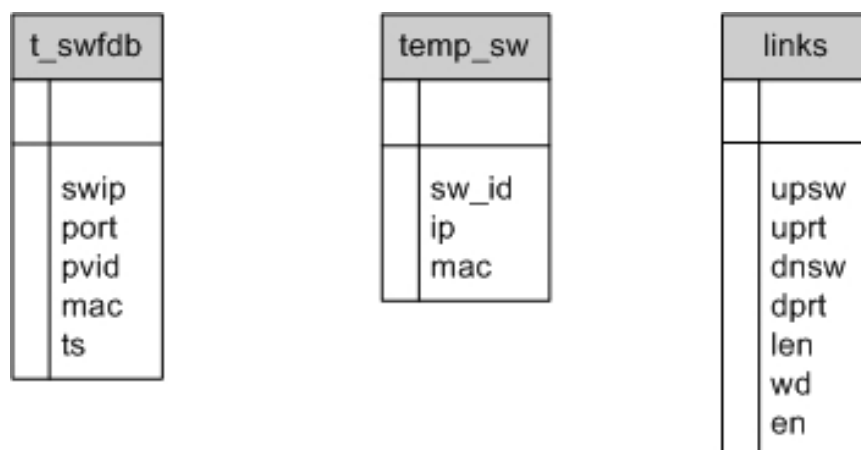
Turėdami tokius duomenis parengiame failus įterpdami PL/pgsql kalbos antraštę

COPY lenteles pavadinimas (laukų pavadinimai) FROM stdin; failo turinys \.

taip palengvindami duomenų importavimą.

## Duomenų bazės struktūra

Duomenų bazė yra sudaryta iš trijų lentelių nesusietų jokiais ryšiais. Dvi lentelės ( temp\_sw ir t\_swfdb ) yra skirtos pradiniam duomenims saugoti. Pradiniai duomenys tai: komutatorių FDB lentelių informacija ir turimų komutatorių IP adresai bei jų fiziniai (MAC) adresai. Trečia lentelė (links), kuri bus skirta saugoti galutiniams duomenims. Ir keletas funkcijų kurios apdoros pateiktus duomenis iki galutinio rezultato. Galutiniai rezultatai bus išvedami į tekstinį failą DOT kalba, kuri apdoros grafikų vaizdavimo priemonė.



### Lentelės

Lentelė *temp\_sw*, yra skirta pradiniam duomenims apie komutatorius saugoti. Duomenys yra paimami iš kitos duomenų bazės, kurioje yra saugoma informacija apie visus turimus įrenginius.

Lauko pavadinimas	Tipas
sw_id	integer
ip	character varying(15)
mac	macaddr

Lentelė *t\_swfdb*, kuri yra skirta pradiniam duomenims, kurie yra gaunami iš komutatorių FDB, saugoti.

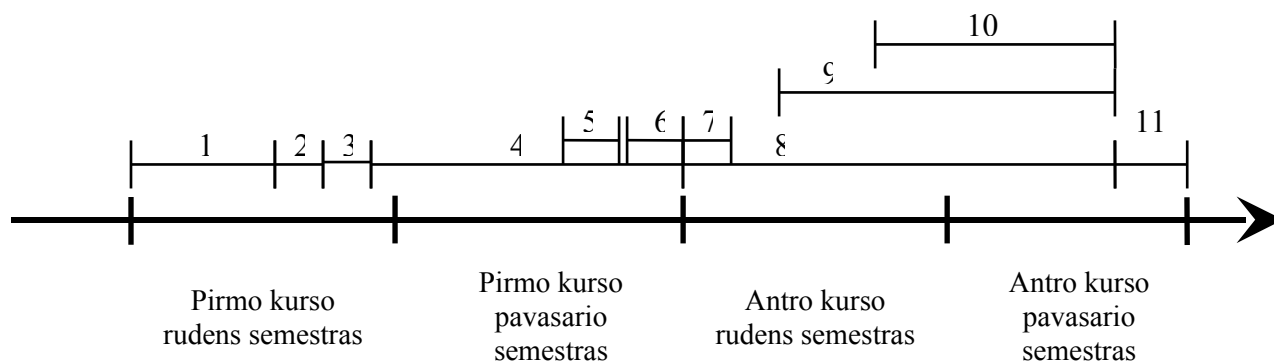
Lauko pavadinimas	Tipas
swip	character varying(15)
port	integer
pvid	integer
mac	macaddr
ts	timestamp with time zone

Lentelė *link* skirta galutiniams duomenims saugoti.

<b>Lauko pavadinimas</b>	<b>Tipas</b>
upsw	character varying(15) NOT NULL
uprt	integer NOT NULL
dnsw	character varying(15) NOT NULL
dprt	integer
len	integer
wd	integer
en	integer

# Darbo eigos aprašymas

## Darbo eigos grafas



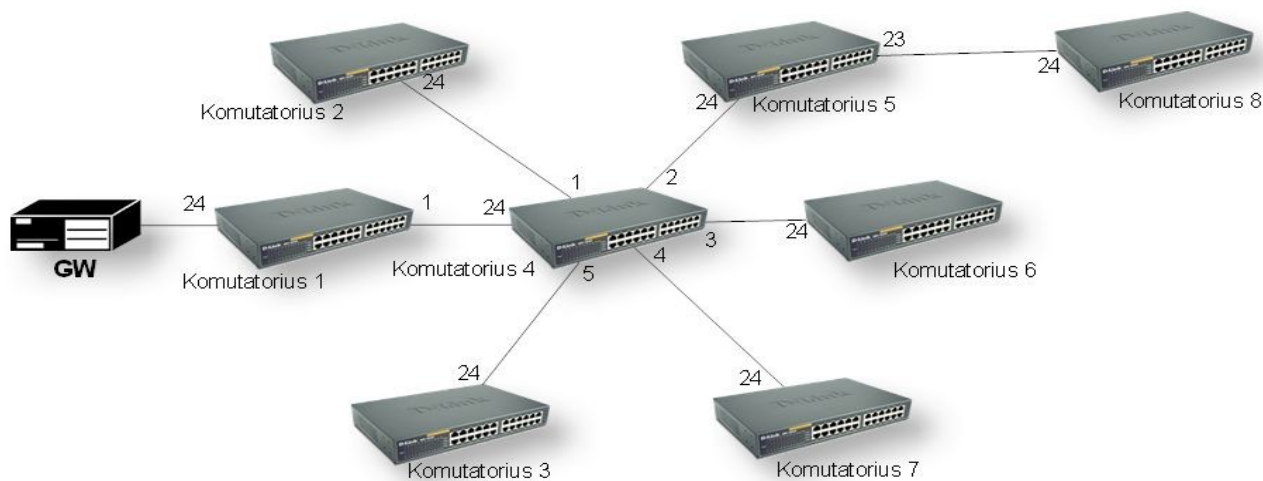
Paveikslėlyje skaičiukai atitinka projekto vykdymo planui (žiūrėti aukščiau)

## Problemų ir jų sprendimų aprašymai ir pagrindimai

Darbo rašymo ir testavimo metu iškilo kelėtas problemų kurių sprendimui buvo sugaišta nemažai laiko.

### Problema:

Pirmoji problema, su kuria teko susidurti, buvo surasti komutatorių grandinę nuo pradinio serverio iki galutinio atšakos komutatoriaus. Kaip pavyzdžiui tarp serverio (GW) ir aštunto komutatoriaus, arba tarp serverio ir šešto komutatoriaus (schema Nr. 6). Pirmu atveju yra penktas, ketvirtas ir pirmas komutatoriai. Antru atveju yra ketvirtas ir pirmas komutatoriai.



Schema Nr. 6

### Sprendimas:

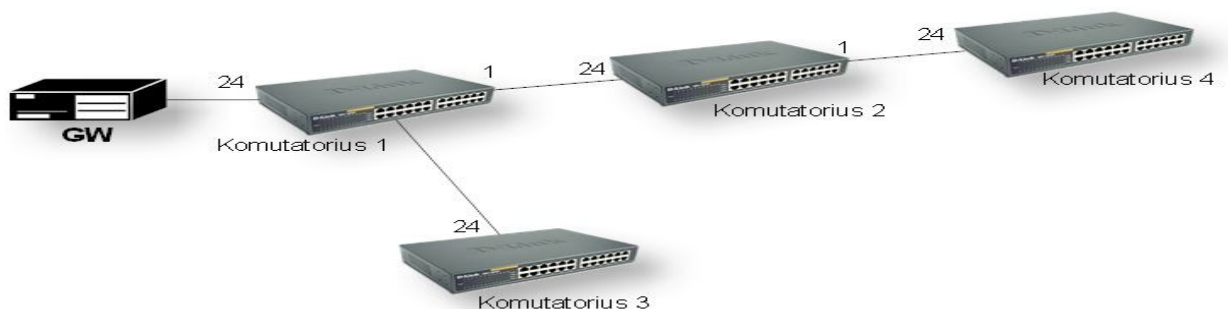
Funkcijai kaip kintamas yra paduodamas serverio fizinis (MAC) adresas ir kurio nors iš komutatorių MAC adresas. Tada imami po vieną kiti komutatoriai ir yra tikrinama ar trečias komutatorius mato pirmuosius du fizinius adresus per skirtingas jungtis ar per vienodas. Jei adresai matomi per skirtingas jungtis, tai reiškia, kad šis komutatorius yra per vidurį pirmųjų dviejų įrenginių. Tai yra jie yra sujungti į grandinę. Pavyzdžiui (žiūrėti į schemą Nr.6) funkcijai padavus serverio ir septinto komutatoriaus adresus, ketvirtasis komutatorius juos matys per skirtingas 4 ir 24 jungtis. O trečiasis komutatorius abu adresus matys per 24 jungtį.

### Problema:

Sekanti problema buvo, ta kaip atrasti komutatorių grandinės eiliškumą, kuris komutatorius eina po kurio. Nežinant komutatorių eiliškumo, gaunasi visiškai netiksli ir galima sakyti nesąmoninga tinklo topologijos schema.

### Sprendimas:

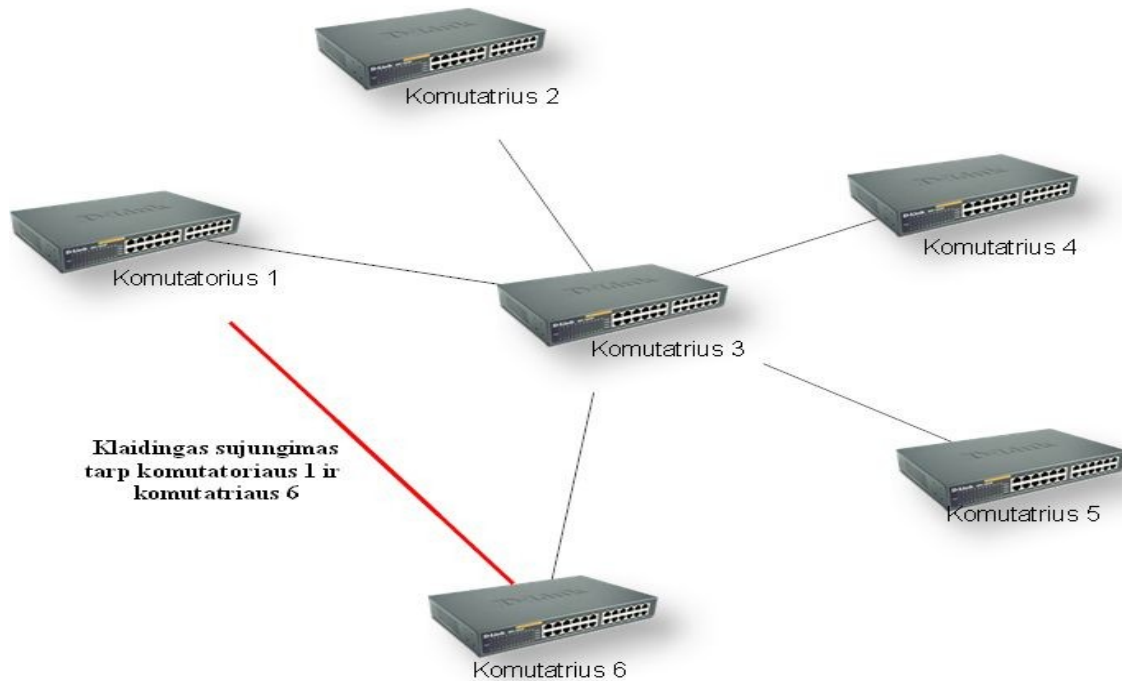
Iš rastos komutatorių grandinės, naudojant ciklą imami du komutatoriai, panaudojant visus galimus variantus ir liginama ar tarp pirmojo paimto komutatoriaus jungčių, išskyrus uplink jungtį ir tarp antrojo komutatoriaus uplink jungties FDB duomenų yra vienodų įrašų. Jei randamas nors vienas bendras įrašas, tada imama kita kombinacija. Jei bendrų įrašų nerandama, tada prie pirmojo paimto komutatoriaus yra pridedamas indeksas. Patikrinus visas galimas kombinacijas, gaunama grandinės eilės tvarka. Didžiausią indeksą turintis komutatorius yra toliausiai nuo serverio. Pavyzdžiui (žiūrėti schemą Nr. 7) paėmus pirmą ir ketvirtą komutatorius yra tikrinama ar tarp pirmojo komutatoriaus jungčių išskyrus 24 jungtį ir tarp ketvirto komutatoriaus 24 jungties FDB duomenų yra bendrų įrašų. Šiuo atveju, būtų bendri antro ir trečio komutatorių fizinių adresų duomenys. Jeigu paimtumėm ketvirtą ir antrą komutatorius ir tikrintume ar tarp ketvirto komutatoriaus jungtis išskyrus 24 jungtį ir tarp antro komutatoriaus 24 jungties FDB duomenų yra bendrų įrašų. Šiuo atveju tokių įrašų nerastume.



Schema Nr. 7

### Problema:

Didžiausia ir sudėtingiausia problema buvo netikrų ryšių atsiradimas tarp dviejų komutatorių. Pavyzdys pateikiamas schemoje Nr. 8.



Schema Nr. 8

Taip nutikdavo todėl, kad kažkuris vienas komutatorius duomenų paėmimo metu neturėdavo pilnos FDB lentelės.

### Sprendimas:

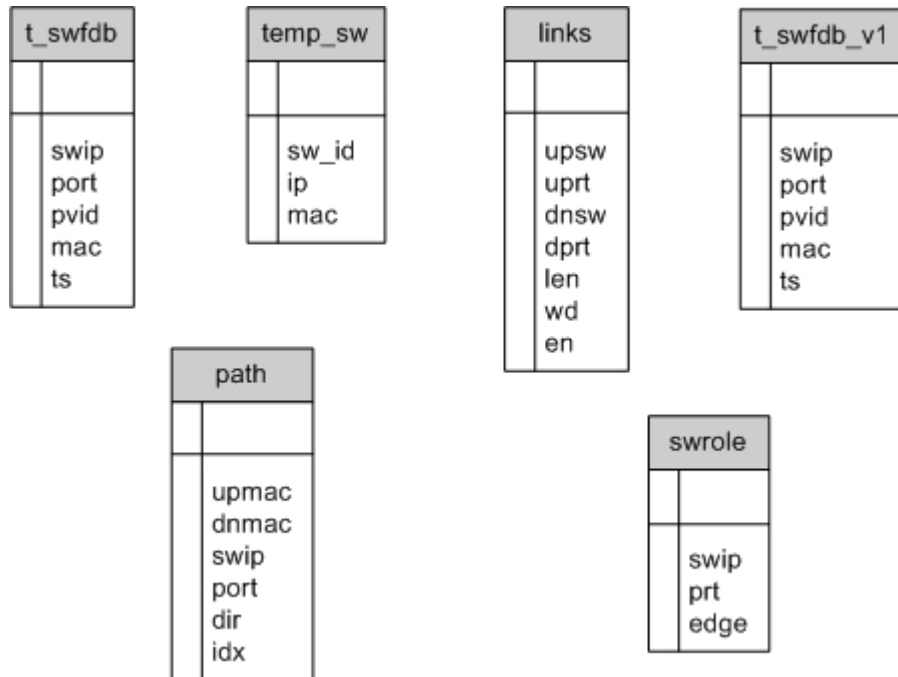
Ši problema buvo išspręsta imant du komutatorius, sujungtus ryšiu ir tikrinant jų FDB lenteles. Jei lentelėse randamas nors vienas bendras įrašas, tai reiškia, kad tarp jų yra dar vienas komutatorius ir ryšys yra klaidingas. Jei bendrų įrašų nėra – ryšys yra teisingas.

## Galutinio projekto stovio aprašymas

Po visų iškilusių problemų, teko pakoreguoti duomenų bazę ir parašyti keletą papildomų funkcijų, kurios išsprendžia visas kilusias problemas. Po viso pertvarkymo duomenų bazėje yra šešios lentelės, kurių trys buvo pridėtos tarpiniams duomenims saugoti, ir septynios funkcijos, kurias įvykužius yra gaunamas galutinis rezultatas.



## Galutinė duomenų bazė



## Papildomai atsiradusios lentelės

Lentelė *t\_swfdb\_v1* skirta tarpiniams duomenims saugoti. Šioje lentelėje yra atfiltruoti reikalingi duomenys iš lentelės *t\_swfdb*.

Lauko pavadinimas	Tipas
swip	character varying(15)
port	integer
pvid	integer
mac	macaddr
ts	timestamp with time zone

Lentelę *path* skirta tarpiniams duomenims saugoti.

Lauko pavadinimas	Tipas
upmac	macaddr
dnmac	macaddr
swip	character varying(15)
port	integer
dir	character varying(4)
idx	integer

Lentelė *swrole* skirta tarpiniams duomenims saugoti.

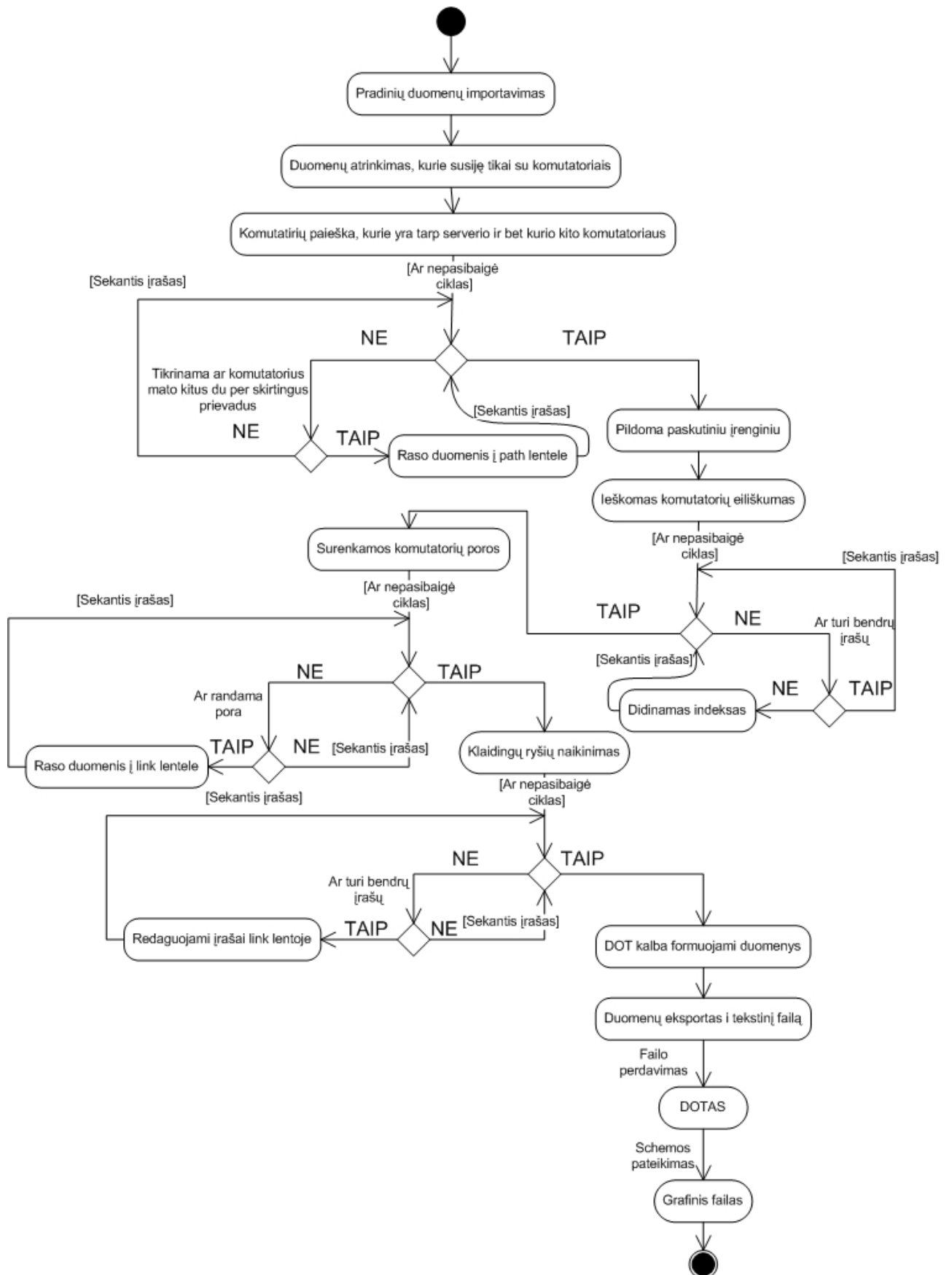
<b>Lauko pavadinimas</b>	<b>Tipas</b>
swip	character varying(15)
pri	bigint
edge	integer

### **Funkcijos**

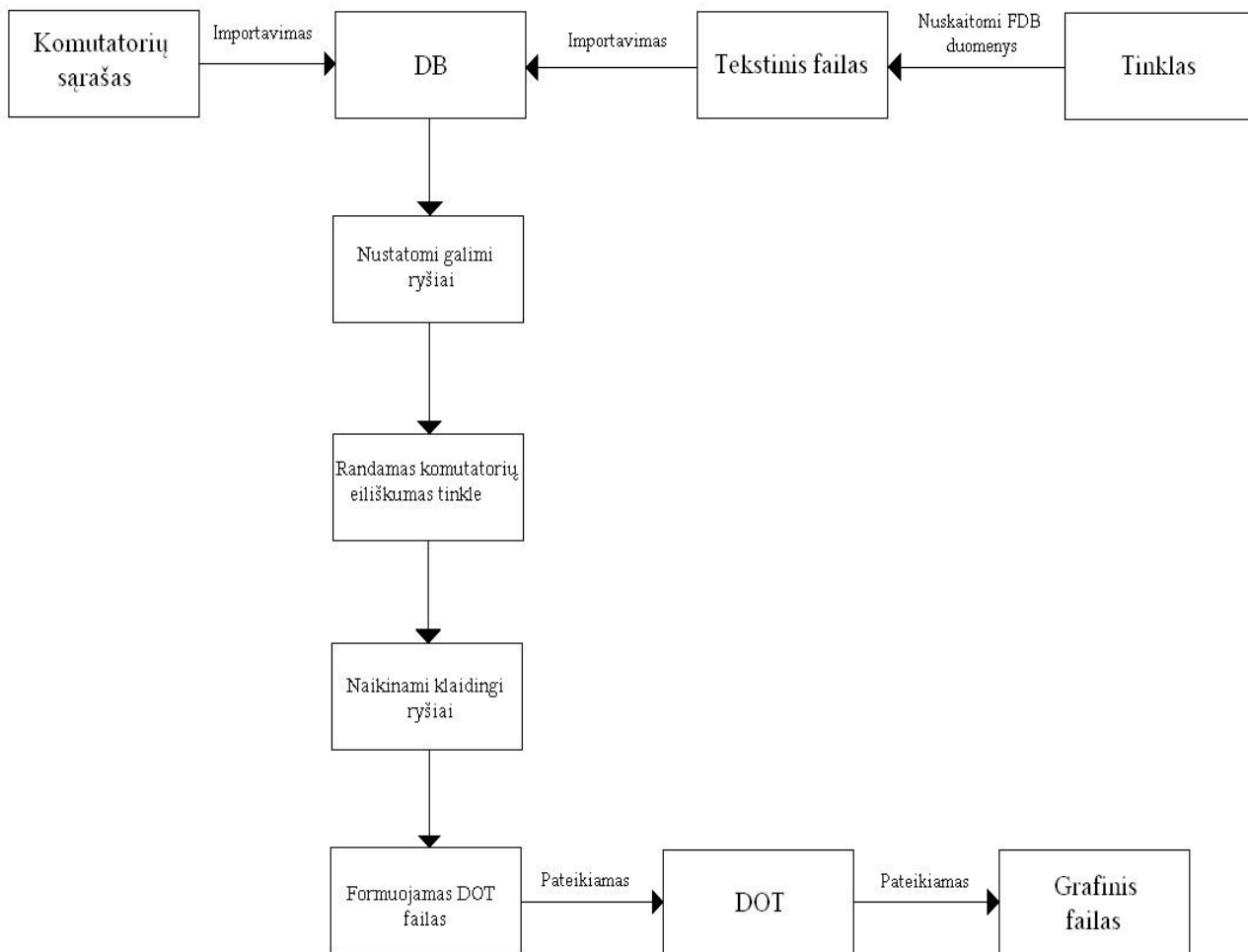
Sukurtos funkcijos, kurios atlieka visus reikiamus veiksmus su pradiniais duomenimis, ir išsprendžia darbo kūrimo ir testavimo metu iškilusias problemas:

- **draw\_path4** – funkcija, kuri išfiltruoja iš visų bendrų FDB duomenų duomenis, kurie yra susiję su komutatoriais, ir dar kreipiasi į dvi funkcijas: `order_sw_path` ir `rec_unord_path`.
- **rec\_unord\_path** – funkcija atranda komutatorių grandinės (pirmoji problema) ir surašo į lentelę *path*.
- **order\_sw\_path** – ši funkcija rastas komutatorių grandinės surikiuoja ir suindeksuoja eiliškumo tvarka (2 problema) taip, kad toliausiai nuo serverio esantis komutatorius turi didžiausią indeksą.
- **drow\_path4\_wocalc\_links** – rikiuoja ir sukelia į kitą lentelę įrašus, tai yra, kuris komutatorius yra sujungtas su kuriuo ir per kurias jungtis.
- `valinate_links` – ši funkcija tikrina ar tarp išrikiuotų komutatorių porų nėra klaidingų ryšių (trečia problema).
- **draw\_path5** – pateikia žemiau esamų komutatorių DOT kalbos stiliaus žemėlapi, nuo nurodyto norimo komutatoriaus ir jungties. Mano atveju reikalinga visa topologinė schema tai yra gaunamas visas žemėlapis.
- **f\_main** – funkcija, kuri suvykdo visas esančias funkcijas.

## Sistemos dinaminis vaizdas



## Darbo veikimo modelis

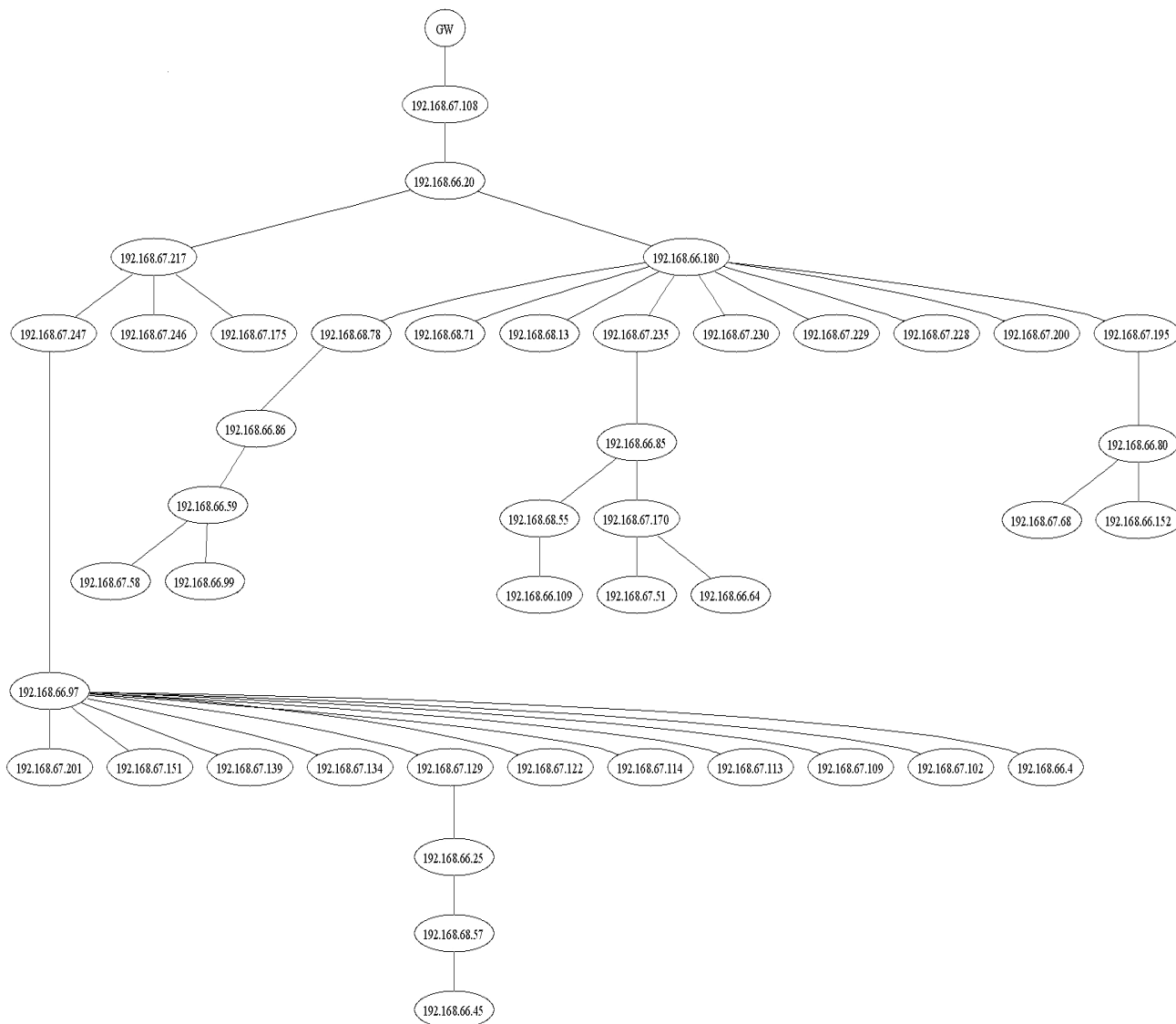


## Vykdomasis failas

Sukurtas vykdomasis failas pavadinimu „Start“. Šis failas sukuria: duomenų bazę, lenteles ir funkcijas. Sukelia pradinis duomenis į lenteles ir suvykdo visas funkcijas, kurias atlikus yra gaunamas DOT kalba parašytas tekstinis failas. Po to, tas failas yra paduodamas DOT programai, kuri apdoroja duomenis ir gražina paveiksliuką su tinklo topologijos schema. Šio vykdomojo failo sukūrimo tikslas buvo visų veiksmų atlikimo ir galutinio rezultato išvedimo palengvinimui.

## Galutinis darbo rezultatas

Galutinis darbo rezultatas - tai tinklo topologijos schema, kuri pateikiama jpeg formato paveiksliuku. Joje pavaizduota tinklo komutatorių išsidėstymas tinkle, ir jų IP adresai. Schema yra formuojama nuo pagrindinio serverio žemyn. Pavyzdys pateikiamas schemeje Nr. 9.



Schema Nr. 9

## Darbo rezultatų analizė

Gero produkto sukūrimas priklauso pirmiausia nuo geros programinės įrangos pasirinkimo. Taip pat priklauso ir nuo pradinio projekto, o nuo jo priklauso ir pats produkto veikimas bei funkcionavimas. Kuo optimaliau bus suprojektuotas, tuo greičiau ir stabiliau sistema veiks, o tai nėra iš lengviausių užduočių.

Darbas buvo atliktas pagal temą „Interneto paslaugos kokybės gerinimo tyrimas“. Remiantis šia tema buvo sukurta sistema, kuri turint pradinis duomenis pateikia tinklo topologijos schemą. Buvo sukurta duomenų bazė, kurioje saugomi pradiniai, tarpiniai ir galutiniai duomenys. Duomenų bazė buvo padaryta pagal projektą, kurį darbo metu teko koreguoti dėl iškilusių problemų pašalinimo. Tam reikėjo pridėti keltą lentelių ir funkcijų.

Mano darbo galutinis variantas - tai duomenų bazė ir funkcijų rinkinys, kurių pagalba tinklo administratoriai gauna tinklo topologijos schemą. Ja naudodamiesi gali lengviau projektuoti interneto tinklą, išvengti nereikalingų problemų ir pagerinti tinklo aptarnavimo kokybę.

## Išvados

Apibendrinat visus atliktus darbus galima parašyti tokias išvadas:

1. Išanalizuoti tinklo komutatorių veikimo principai ir jų savybės bendraujant su kitais tinklo įrenginiais.
2. Išanalizavus darbo temą, sritį ir apsibrėžus tam tikrus reikalavimus buvo sukurta sistema analizuojanti komutatorių duomenų lenteles ir grafiškai pateikianti jų išsidėstymo tinkle schemą.
3. Gauta tinklo topologijos schema palengvina tinklo plėtrą ar rekonstrukciją, padeda analizuoti atsiradusias problemas ir išvengti daugelį galimų problemų.

## Literatūros sąrašas

1. [http://citeseer.ist.psu.edu/cache/papers/cs/25556/http:zSzzSzwww.cs.wm.eduzSz~lowekamp zSzpaperszSzlowekamp-topology-sigcomm01.pdf/topology-discovery-for-large.pdf](http://citeseer.ist.psu.edu/cache/papers/cs/25556/http%3A%2F%2Fwww.cs.wm.edu%2F~lowekamp%2Fpapers%2Flowekamp-topology-sigcomm01.pdf/topology-discovery-for-large.pdf)
2. [http://citeseer.ist.psu.edu/cache/papers/cs/12975/http:zSzzSzwww.ieee-infocom.orgzSz2000zSzpaperszSz366.pdf/breitbart00topology.pdf](http://citeseer.ist.psu.edu/cache/papers/cs/12975/http%3A%2F%2Fwww.ieee-infocom.org%2F2000%2Fpapers%2F366.pdf/breitbart00topology.pdf)
3. [http://www.ieee-infocom.org/2003/papers/09\\_02.PDF](http://www.ieee-infocom.org/2003/papers/09_02.PDF)
4. Victor Olifer. [Computer Networks: Principles, Technologies and Protocols for Network Design](#)
5. <http://www.postgresql.org/docs/>
6. <http://www.pgadmin.org/index.php>
7. <http://www.graphviz.org/>
8. [http://www.tns-gallup.lt/lt/disp.php/lt\\_news](http://www.tns-gallup.lt/lt/disp.php/lt_news)



## Anotacija

E.Papaurėlis

Interneto paslaugos kokybės gerinimo tyrimas.

Augant interneto vartotojų skaičiui auga ir jų poreikiai, didėja paslaugų įvairovė. Vartotojų poreikiams patenkinti tinklas turi veikti nepriekaištingai. Šiame darbe yra aptariami tinklo reikalavimai, ir jo aptarnavimo kokybė. Darbo tikslas buvo sukurti sistemą, kuri nupieštų esamą tinklo topologiją pasinaudojant tinklo komutatorių adresų lentelėmis. Tiksliai tinklo topologijos schema padeda išvengti problemų, kurios kyla plečiant tinklą ar diegiant naujas technologijas ir paslaugas. Taip pat padeda išanalizuoti atsirandančius gedimus tinkle ir greičiau juos likviduoti. Darbo metu susipažinta su PostgreSQL, pgAdmin ir Graphviz sistemomis. Sukurta duomenų bazė ir duomenis apdorojančios funkcijos, kurios pateikia galutinį rezultatą. Galutinis rezultatas – tai JPEG formato paveikslukas, kuriame atvaizduota komutatorių tinklo topologija.

E.Papaurėlis

Analysis of internet service quality improvement

Significantly growing number of internet users, increases their needs and expands variety of services. To satisfy customers' needs and expectations the network has to work indefectibly. The dissertation examines network specifications and its maintenance quality. The main aim of the work was to create the system which would illustrate the topology of the current network, using network switch forwarding databases (FDB). The precise network topology scheme helps to avoid many problems which might turn up expanding the network or implementing new technologies or services. This also helps to analyze originating network failures and eliminate them more quickly. PostgreSQL, pgAdmin and Graphviz systems were analyzed throughout the work. Data base and data processing functions were created to produce final result. The final result – JPEG format picture which represents topology of network switches.