KAUNO TECHNOLOGIJOS UNIVERSITETAS
INFORMATIKOS FAKULTETAS
PROGRAMŲ INŽINERIJOS KATEDRA

Vytautas Bruzgulis

IPv6 tinklo topologijos monitoringo sistema

Magistro darbas

Darbo vadovas
dr. Rimantas Kavaliūnas

Kaunas, 2006

KAUNO TECHNOLOGIJOS UNIVERSITETAS
INFORMATIKOS FAKULTETAS
PROGRAMŲ INŽINERIJOS KATEDRA

Vytautas Bruzgulis

IPv6 tinklo topologijos monitoringo sistema

Magistro darbas

Kalbos konsultantė

dr. Jurgita Mikelionienė 2006-05-29

Vadovas

dr. Rimantas Kavaliūnas 2006-05-29

Recenzentas

dr. Gytis Vilutis 2006-05-29

Atliko
IFM-0/2 gr. stud.
Vytautas Bruzgulis 2006-05-29

Kaunas, 2006

TURINYS

Lentelių sąrašas

Paveikslėlių sąrašas

SUMMARY

IPv6 network topology monitoring system.

Outage of computer network detection and continuous network state monitoring are one of the most important aspects to ensure network service quality. As usage of the next generation internet protocol (IPv6) is vastly increasing, so does the need for protocol parameters monitoring and operational problems detection.

Centralised data collection is the most oftenly used method in monitoring systems. Main disadvantages of such approach are: heavily loaded central node, possible excessive network cappacity usage when transmiting monitoring data, common network view only from the single point in the network topology. Distributed monitoring partialy eliminates these problems.

When monitoring data networks, it is necessery to know not only the availability of the backbone network to the edge networks, but also the reachability among several edge networks, availability for the egde network to reach the backbone network and the global network. It is possible to use distributed network monitoring technique for this purpose. The agents, which perform network querying, should be placed at the topology edges and shold query all needed network nodes (not only the part topologicaly close to the agent). Thus it is possible to get the multipoint common network state view.

The monitoring software, described here, is created using this distributed multipoint network monitoring technique and adapting it for monitoring IPv6.

1. ĮVADAS

Sutrikimų kompiuterių tinkle aptikimas ir pastovus tinklo būklės monitoringas yra vienos iš svarbiausių veiklų tinklo veikimo kokybiškumui užtikrinti. Sparčiai plintant naujos kartos interneto protokolo (IPv6) panaudojimui, taip pat auga poreikis šio protokolo parametrams stebėti ir protokolo veikimo sutrikimams aptikti.

Dažniausiai duomenims surinkti naudojamas centralizuotas monitoringas. Didžiausi šio metodo trūkumai: didelis apkrovimas centriniam mazgui, galimas perteklinis tinklo pralaidumo išnaudojimas tarnybinei informacijai perduoti, tinklo būsenos vaizdas tik vieno taško atžvilgiu. Paskirstyto monitoringo metodas iš dalies pašalina šias problemas.

Stebint duomenų perdavimo tinklą, aktualu žinoti ne tik pateikiamumą iš magistralinio tinklo į kraštinius tinklus, bet ir pasiekiamumą tarp kelių kraštinių tinklų arba nuo kraštinio tinklo į magistralinius tinklus ar internetą. Tam galima panaudoti paskirstyto monitoringo metodą, stebėjimo taškus išdėstant tinklo kraštuose ir iš visų taškų stebint visą tinklą (o ne tik topologiškai atitinkamam agentui artimą jo dalį). Tokiu būdu gaunamas pilnas tinklo būsenos vaizdas iš keleto perspektyvų.

Panaudojus tokį monitoringo traktavimą ir specializavus programinę įrangą IPv6 protokolo monitoringui, sukurta IPv6 tinklo monitoringo sistema.

2. TECHNOLOGIJŲ IR PROGRAMŲ SISTEMŲ APŽVALGA

Monitoringas šiuolaikinėse tinklo infrastruktūrose yra vis svarbesnė bei tuo pačiu sudėtinga ir daug pastangų reikalaujanti užduotis. Galimybė stebėti tinklą leidžia jo operatoriams gauti pastoviai atnaujinamą informaciją apie tinklo ir jo veikimo būklę, nedelsiant pastebėti, kad įvyko pasikeitimas. Nuolatinis šios informacijos pateikiamumas svarbus daugeliui tinklo veikimo aspektų (sangrūdoms (*bottleneck*) aptikti, paslaugų lygmens susitarimams (SLA) tikrinti, paslaugos kokybės garantijoms (QoS) užtikrinti ir kt.) [1].

2.1. Monitoringo technologijų apžvalga

Didėjant ir sudėtingėjant šiuolaikiniams kompiuterių tinklams, tokiomis tampa ir tinklo monitoringo sistemos. Tai verčia tinklų operatorius kurti ir vystyti lanksčią monitoringo infrastruktūrą, galinčią nepaliaujamai stebėti visą tinklą. Tinklui plečiantis, būtina pritaikyti esamą monitoringo sistemą taip, kad monitoringo procesas nesustotų [2].

Dauguma tinklo valdymo programinės įrangos produktų naudoja standartinius, praktiškai visos aparatūrinės įrangos ir operacinių sistemų palaikomus apklausos ir apsikeitimo duomenimis metodus: interneto valdymo žinučių protokolą (ICMP), perdavimo valdymo protokolą (TCP), vartotojo duomenų fragmentų protokolą ir paprastą tinklo valdymo protokolą (SNMP). Naudojant pirmus tris metodus, valdančioji stotis siunčia atitinkamo protokolo žinutes apklausiamam įrenginiui ir tikisi gauti standartinį to protokolo atsakymą. Naudojant SNMP, valdymo stotis apklausia SNMP agentus (atskiras vartotojo lygmens programas), įdiegtus stebimuose tinklo įrenginiuose, kurie grąžina informaciją apie tų įrenginių būklę [3].

Identifikuojamos keturios pagrindinės monitoringo architektūros: kliento-serverio, hierarchinis statinis, silpnai mobilus ir stipriai mobilus [4]. Pirmasis modelis yra standartinis centralizuotas monitoringas, o likę trys yra paskirstyto monitoringo atmainos. Mobiliųjų modelių veikimas susijęs su kodo persiuntimu tarp centrinio mazgo ir tarpinių agentų, kas labai pagerina tinklo valdymą, tačiau monitoringui tokio funkcionalumo nereikia, taigi šiuos modelius taikyti monitoringo užduotims netikslinga.

Vertinant modelių efektyvumą, problema yra atsitiktinio kintamojo X pakitimo pastebėjimas atsitiktiniame stebimame mazge (SM). Kintamųjų pasikeitimai laikomi atsitiktiniu procesu, kurio dažnis λ (laiko tarpai tarp pasikeitimų yra nepriklausomi eksponentiniai atsitiktiniai kintamieji, kurių reikšmės yra $1/\lambda$) bet kuriame SM. Pagrindinė

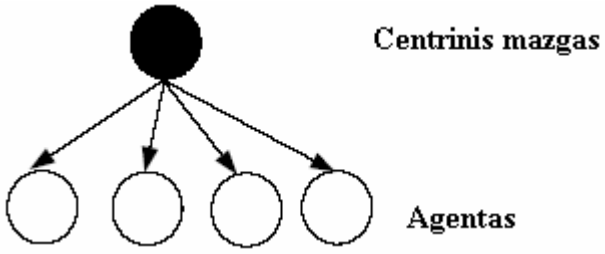
efektyvumo metrika yra laikas T, reikalingas aptikti pasikeitimą. Aptikimo kainos parametrai yra:

C tinklo srauto kiekis (bps), matuojamas centriniame mazge ir SM;

U vidutinis procesoriaus pajėgumų panaudojimas centriniame mazge ir SM.

2.1.1. Centralizuotas monitoringas

Centralizuoto monitoringo atveju naudojama kliento-serverio architektūra su vienu centriniu serveriu ir vienu ar daugiau klientų. Tokia architektūra turi labai svarbių privalumų. Vieni iš svarbiausiųjų yra jos paprastumas bei tinkamumas daugeliui tinklų, nereikalaujančių paskirstyto administravimo, dažno įrenginių apklausimo ar dažnų skaičiavimų, turinčių didelio pralaidumo linijas tarp centrinio mazgo ir agentų ir perduodančių nedidelius monitoringo informacijos kiekius [5].



1 pav. Kliento-serverio monitoringo architektūra

Pagrindiniai centralizuoto monitoringo trūkumai yra metodo nelankstumas ir nepakantumas gedimams (sugedus pagrindiniam mazgui, visa sistema tampa nefunkcionali) [5].

SNMP paremtos valdymo duomenų surinkimo sistemos dažniausiai yra centralizuotos, todėl pasižymi dideliu valdymo stoties apkrovimu (procesoriaus ir atminties resursai) ir savo persiuntimais apkrauna duomenų perdavimo tinklo resursus [6]. Šie papildomi nuostoliai dar padidėja, kai įrenginių apklausimo dažnis yra didelis. Dažnas apklausinėjimas yra būtinas, norint įvykdyti QoS garantijas ir greitai aptikti sutrikimus. Siekiant sumažinti tinklo įrenginio procesoriaus apkrovimą, nemažai dėmesio skiriama paties SNMP protokolo primityvų tobulinimui [7].

Centralizuotos tinklo monitoringo sistemos atveju, atsakymai į užklausas turi būti siunčiami į centrinį įrenginį. Tai leidžia matyti visą tinklą, tačiau sudaro galimybę atsirasti duomenų spūsčiai tinklo segmentuose, esančiuose netoli centrinio valdymo įrenginio [8]. Taip pat tokiu atveju tinklas matomas tik iš centrinio mazgo perspektyvos, tai reiškia, kad pateikiamumas taip pat bus matuojamas tik iš centrinio mazgo pozicijos. Dar viena centralizuoto monitoringo problema yra tai, kad įvykus gedimui centrinio mazgo prijungimo

prie tinklo taške, jokia tinklo stebėjimo ir valdymo informacija jo negalės pasiekti, todėl visas tinklas bus traktuojamas kaip nepasiekiamas.

Centralizuoto monitoringo atveju, kintamasis X (žr. 2.1 skyrelį) gaunamas iš centrinio mazgo pastoviais intervalais apklausiant SM. Metodas charakterizuojamas tokiais kintamaisiais [9]:

- Δ apklausimo intervalas;
- i_q užklausos pranešimo dydis (bitais);
- i_r atsakymo pranešimo dydis (bitais);
- h paketo antraštės dydis (bitais);
- t_c užklausos ar atsakymo pranešimo apdorojimo laikas SM;
- t_q užklausos pranešimo apdorojimo laikas centriniame mazge;
- t_r atsakymo pranešimo apdorojimo laikas centriniame mazge.

Kintamojo pasikeitimas vienodai tikėtinas bet kuriuo apklausos intervalo Δ metu. Vidutinis laikas iki sekanti užklausa pasieks SM po kintamojo pasikeitimo yra $\Delta/2$. Gavęs užklausą, SM išsiunčia atsakymą centriniam mazgui. Sudėjus visus laikus, tikėtinas laikas, po kurio centrinis mazgas aptiks kintamojo pasikeitimą yra [9]:

$$T_{CS} = \frac{\Delta}{2} + 2t_c + t_r + \bar{t}_p(N). \quad (1)$$

Kiekvienas apklausimas sudarytas iš užklausos-atsakymo pranešimų poros, taigi kiekvienas SM turi apdoroti $(i_q + i_r + 2h) / \Delta$ b/s. Centrinis mazgas turi apdoroti srautą iš N mazgų [9]:

$$C_{CS} = \frac{N(i_q + i_r + 2h)}{\Delta} \text{ b/s.} \quad (2)$$

Spūsties atsiradimas duomenų linijoje prie centrinio mazgo akivaizdžiai tiesiogiai priklausomas nuo N reikšmės.

Kiekviename stebimame mazge yra po nesudėtingą programėlę (agentą), taigi atminties sunaudojimas nėra problema. Agentas kiekvieno intervalo Δ metu užklausai arba atsakymui apdoroti sunaudoja laiką t_c. Laiko dalis, reikalinga apdoroti apklausimo paketams SM yra $2t_c/\Delta$. Svarbesnis yra laiko tarpas, reikalingas apdoroti paketams centriniame mazge, kuris yra [9]:

$$U_{CS} = \frac{N(t_q + t_r)}{\Delta}. \quad (3)$$

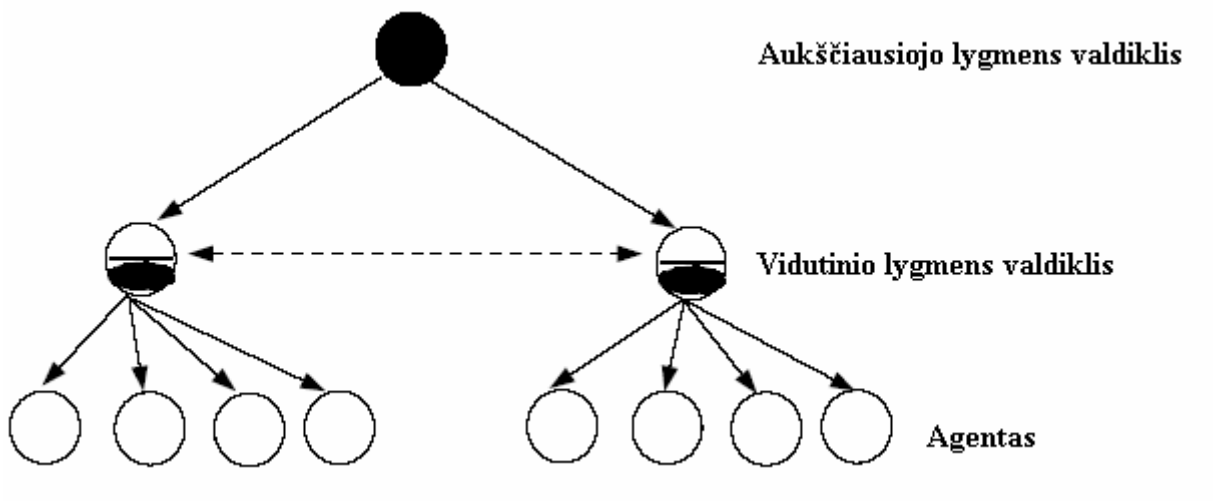
Apdorojimo perkrova centriniame mazge atsiras tada, kai tinklo dydis bus $N \geq \Delta / (t_q + t_r)$, kadangi tokiu atveju šis įrenginys bus užimtas visą laiką.

2.1.2. Paskirstytas monitoringas

Bandant pašalinti centralizuotu monitoringu paremtų sistemų efektyvumo ir patikimumo problemas, pastebėta, kad išskirstant, padalinant resursus bei darbus, sumažinamas tiek vieno centrinio mazgo apkrovimas, tiek vieno centrinio mazgo sutrikimo pasekmės [10]. Toks metodas vadinamas paskirstytu monitoringu.

Paskirstymas sumažina papildomą nereikalingą resursų panaudojimą, todėl tokios sistemos yra lankstesnės ir gali apdoroti didesnį stebimų objektų skaičių; be to, monitoringo sistema tampa patikimesnė (įvykus gedimui vienoje vietoje, kitos dalys tebefunkcionuos). Kita vertus, paskirstymas įneša ir naujų problemų: centralizuoto valdymo nebuvimas, sinchronizacija, sumažėjusi standartizacija bei padidėjęs sudėtingumas [5]. Kadangi perduodant duomenis atsiranda dar viena tarpinė grandis, aktuali ir saugumo bei duomenų patikimumo problema.

Hierarchinio statinio modelio atveju egzistuoja L vidurinio lygmens valdiklių, atliekančių dalies tinklo elementų N/L monitoringą. Dažniausiai sutinkama dviejų lygmenų hierarchija, tačiau galimos ir daugiau lygmenų turinčios architektūros. Kiekviena tinklo dalis apklausiama kliento-serverio metodu, o vidurinio lygmens valdikliai gali komunikuoti tiek su centriniu mazgu, tiek tarpusavyje [9].



2 pav. Hierarchinė monitoringo architektūra

Paskirstyto modelio atveju, monitoringo funkciją centrinis mazgas deleguoja L tarpinio lygmens valdikliams, kurie vienu metu kliento-serverio monitoringo būdu vykdo N/L potinklių monitoringą. Informacijos šifravimo ir autentifikacijos nuostoliai įvertinami kaip papildomas paketų apdorojimo vėlinimas t_s siuntėjui ir gavėjui bei paketo padidėjimas h_s baitų. Papildomas vėlinimas dėl saugumo t_s gali būti nereikšmingas, nes naudojami šifravimo

algoritmai ir aparatūrinė įranga yra daugmaž vienodi. Apklausimo dažniui esant vienodam, laikas, reikalingas tarpiniam mazgui aptikti kintamojo X reikšmės pakitimą, yra [9]:

$$T_{HS} = \frac{\Delta}{2} + 2t_c + t_r + 3t_s + \bar{t}_p(\frac{N}{L}). \quad (4)$$

Šis laikas yra geresnis, negu (1), kai $3t_s + \bar{t}_p(N/L) < \bar{t}_p(N)$.

Kad surasti laiką, po kurio kintamojo reikšmės pokytis bus užfiksuotas centriniame mazge, reikia pridėti papildomą laiką $t_r + 3t_s + \bar{t}_p(N)$; taigi, pilnutinis laiko tarpas, po kurio bus pastebėtas būsenos pasikeitimas centriniame mazge yra [9]:

$$T_{HS} = \frac{\Delta}{2} + 2t_c + 2t_r + 5t_s + \bar{t}_p(\frac{N}{L}) + \bar{t}_p(N). \quad (5)$$

Aptikimo laikas ilgesnis, negu (1) dėl papildomo laiko, sunaudojamo saugumui užtikrinti, ir ilgesnio srauto kelio, atsirandančio dėl tarpinių valdiklių.

Esant vienodam apklausimo intervalui Δ, kiekvieno SM veikimas nedaug tesiskiria nuo jų veikimo centralizuoto monitoringo atveju. Kiekvienas SM turės apdoroti $(i_q + i_r + 2h + 2h_s)/\Delta$ b/s srautą, kuris yra šiek tiek didesnis dėl atsiradusio papildomo srauto, norint užtikrinti perdavimo saugumą. Kiekvienas tarpinis valdiklis turės apdoroti $N(i_q + i_r + 2h + 2h_s)/L\Delta$ b/s, stebėdamas N/L įrenginių bei išsiųsti vidutiniškai $N\lambda(i_r + h + h_s)/L$ b/s centriniam mazgui, jei pasikeitė visų kintamųjų reikšmės. Tarpiniai valdikliai yra naudingi, jei jie atlieka duomenų filtravimą ir praneša centriniam mazgui tik pasikeitusias kintamųjų reikšmes. Centriniam mazgui nebereikia siųsti užklausos pranešimų ir jis gauna tik bendrą atsakymų srautą [9]:

$$C_{HS} = N\lambda(i_r + h + h_s) \text{ b/s} \quad (6)$$

Nors srautas (6) vis tiek tiesiogiai priklausomas nuo N, jis turėtų būti mažesnis, negu centralizuoto modelio atveju (2), jei apklausimo dažnis $1/\Delta$ yra didesnis, negu kintamojo X pasikeitimo dažnis λ.

Apdorojimo resursai yra žymiai svarbesnė problema, negu atminties resursų panaudojimas (kuris kaip ir centralizuoto modelio atveju yra pastovus). Kiekvienas tarpinio lygmens valdiklis apdorodamas paketus praleidžia $N(t_q + (1 + \lambda\Delta)(t_r + 2t_s))/L$ laiko tarpą kiekvieno apklausimo intervalo Δ metu. Tai reiškia, kad laiko dalis, praleista apdorojant paketų srautą yra: $N(t_q + (1 + \lambda\Delta)(t_r + 2t_s))/L\Delta$. Kad išvengti tarpinio lygmens valdiklių pilno apkrovimo apdorojant srautus, reikia mažiausiai $L > N(t_q + (1 + \lambda\Delta)(t_r + 2t_s))/\Delta$ tarpinio lygmens valdiklių. Laiko dalis, praleista apdorojant srautą centriniame mazge yra [9]:

$$U_{HS} = N\lambda(t_q + t_s). \quad (7)$$

Ši vertė bet kuriuo atveju yra priklausoma nuo bendro kintamųjų pasikeitimo skaičiaus tinkle. Apdorojimo perkrovimo galimybė centriniame mazge yra iš dalies sumažinta ir jis gali apdoroti iki $N = 1/\lambda(t_q + t_s)$ mazgų.

2.1.3. Išvados

Tradicinio centralizuoto modelio efektyvumo bei lankstumo charakteristikos yra blogesnės, negu statinio hierarchinio. Statinis hierarchinis modelis, panaudojant pakankamai daug tarpinių agentų, gali sėkmingai pašalinti srauto bei apdorojimo perkrovimus, kurie tikėtini naudojant centralizuotą metodą. Visgi, naudojant paskirstyto monitoringo modelį, reikia atsižvelgti ir į papildomas pinigines išlaidas, kurios bus didesnės negu naudojant centralizuotą modelį, kadangi papildomai kainuos tarpinius agentus aptarnaujanti aparatūrinė įranga.

2.2. IPv6 monitoringo sistemos

Šiuo metu jau egzistuoja tiek komercinės, tiek nemokamos tinklo topologijos monitoringo sistemos, kuriose be IPv4 iš dalies realizuotas ir IPv6 monitoringo palaikymas.

2.2.1. CAIDA „Macroscopic IPv6 Topology measurements“ projektas

CAIDA renka ir analizuoja mikroskopinės IPv6 topologijos duomenis, bendradarbiauja su WAND tyrimo grupe iš Waikato universiteto Naujojoje Zelandijoje [11].

Projekto tikslai

Pagrindiniai projekto tikslai yra šie:

- charakterizuoti mikroskopinę IPv6 aktyvių adresų erdvės topologiją,
- ištirti IPv6 pasiekiamumo charakteristikas,
- vizualiai pavaizduoti IPv6 pasiekiamumo paplitimą,
- pateikti topologijos duomenis visuomenei modeliavimo simuliacijos ir analizės tikslais.

Duomenų surinkimo metodai

Duomenims surinkti naudojami du šaltiniai: aktyvūs IPv6 kelio informacijos matavimai ir globalios BGP maršrutizavimo lentelės informacija [11].

Kelio informacijos matavimai

IPv6 tinklo lygmens sujungimų duomenims gauti naudojama specialiai tam tikslui sukurta programa *scamper*. Ši programa periodiškai siunčia paketus iš monitoringo šaltinių iki nurodytų IPv6 adresų ir įrašo šių paketų nueitą kelią bei paketo keliavimo laiką tiek iki

tarpinių, tiek iki galutinio įrenginio. *Scamper* veikimo metodologija labai panaši į IPv4 monitoringo programos *skitter* veikimą.

IPv6 gavėjų sąrašas buvo sudarytas iš tokių šaltinių (jame yra 4235 adresai) [11]:

Lentelė Nr. 1. CAIDA projekto duomenys

Šaltinis	Metodas	Adresų kiekis
6bone.db failas	Programos ir tunelių adresai	867
Google API paieška raktiniam žodžiui ‚IPv6'	IPv6 žiniatinklio serveriai, rasti pirmame 1000 rezultatų	123
Testiniai *scamper* paleidimai	Tarpiniai mazgai	480
1.0.0.2.ip6.arpa	DNS užklausos	144 DNS serveriai, 2445 pavadinti IPv6 adresai
henk@ripe.net	RIPE TTM monitoringas	11
Sugeneruoti adresai	Naudojami 0::1 galūnės, kadangi tai dažniausiai naudojami adresai	165

Topologijai aptikti naudojamas metodas panašus, į *traceroute*: kiekvienas tarpinis adresas yra nustatomas siunčiant paketus su vis didesne *Hop Limit* reikšme IPv6 antraštėje. Eilinis IPv6 kelio elementas randamas pagal tarpinio įrenginio grąžinamą ICMP klaidos pranešimą apie per mažą *Hop Limit* reikšmę.

Kad paketo keliavimo laiko intervalas būtų tikslesnis, kūrėjai sukūrė FreeBSD Unix operacinės sistemos branduolio modulį, kurį naudoja apklausiančioji programa. Šis modulis nėra skirtas tiksliems vienpusiams matavimams ar detaliam koreliacijos tyrimui, tačiau jis pakankamas, kad būtų galima pastebėti našumo netolygumus infrastruktūroje. Palygindami įvairių šaltinių duomenis, analitikai gali nustatyti kanalo užsikimšimo arba našumo sumažėjimo taškus bei rasti sritis, kuriose reikia tobulinti technologijas bei įrangą.

Kelio pasikeitimai kiekvienos užklausos metu visada lyginami su prieš tai gautais rezultatais, kad būtų pastebėti topologijos pasikeitimai. Laiko tarpas, per kurį surandamas pilnas kelias iki konkretaus galinio įrenginio turi būti pakankamai trumpas, kadangi reikalingas momentinis kelias. Be to, vienu metu reikia rasti kelius iki daugelio galinių įrenginių, dėl to reikalingas lygiagretus apklausimas. Dėl reikalavimų konfliktavimo, būtina rasti kompromisą [12].

BGP informacijos rinkimas

Vykdant *Route Views* projektą, buvo pradėti rinkti IPv6 maršrutų lentelės duomenys. Projekto pagrindinis tikslas – rinkti BGP IPv4 maršrutų lenteles iš įvairių pasaulio interneto

paslaugų tiekėjų. Du tiekėjai be IPv4 informacijos sutiko pateikti ir IPv6 maršrutų duomenis [11].

Route Views projekto pradinis tikslas buvo sukurti įrankį leidžiantį interneto operatoriams gauti realaus laiko informaciją apie globalią maršrutizavimo lentelę iš skirtingų perspektyvų įvairiose interneto vietose. Projektas skiriasi nuo įvairių „padidinamojo stiklo“ tipo programų tuo, kad topologijos būklė matoma iš keleto perspektyvų realiu laiku.

Darbo principas toks: *Route Views* maršrutizatorius užmezga *multi-hop* BGP sesijas su magistraliniais dominančių tinklų maršrutizatoriais ir paima jų globalios maršrutų lentelės duomenis. Maršrutai gauti iš šių maršrutizatorių niekur neskelbiami, o ir pats duomenis surenkantis maršrutizatorius neskelbia jokios savo maršrutų informacijos kitiems BGP sesijų dalyviams [13].

Darbo rezultatai

Pirmasis globalus *scamper* testas buvo atliktas 2003 metų birželį. Įrankis buvo paleistas vienu metu iš keturių geografiškai nutolusių vietų: CAIDA (San Diego, JAV), WIDE (Japonija), WAND (Naujoji Zelandija) ir Oregono universiteto (Eugene, JAV). Matavimams buvo naudojami 4235 galiniai taškai [11].

Šio bei įvairių vėlesnių testavimų grafai pateikti *skitter* puslapyje [14].

Išvados

CAIDA IPv6 topologijos matavimų patirtis, surinkti duomenys bei metodiniai sprendimai gali būti panaudojami daugeliui monitoringo tikslų pasiekti. Informacijos surinkimo metodologija gali būti pritaikyta tiek kuriant automatinio topologijos aptikimo bei pasikeitimų topologijoje aptikimo sistemą, tiek ir pasiekiamumo monitoringo sistemoje. Surinkti duomenys leidžia palyginti kitomis priemonėmis gautą topologinę globalios lentelės informaciją su *scamper* teikiamais rezultatais ir taip turėti tikslesnį globalios BGP lentelės būklės vaizdą. Tokia informacija itin aktuali dideliems interneto paslaugų tiekėjams, turintiems keletą autonominių sistemų bei tinklo pasiekiamumo taškų (NAP).

2.2.2. HP „Network Node Manager“ praplėtimas IPv6 palaikymui

HP monitoringo sistemos praplėtimas maršrutizavimui skirtas surasti tinklo įrenginius ir pateikti kuo daugiau informacijos apie esamą tinklo būklę naudojant įvairias tinklo diagnostikos galimybes. Produktas skirtas HP-UX arba Solaris operacinėms sistemoms. Be kitų technologijų šiame praplėtime realizuotas ir IPv6 palaikymas, kurio pagrindinės galimybės tokios [15]:

- automatiškai aptinka IPv6 įrenginius tinkle,
- atlieka tiek IPv4, tiek IPv6 protokolo monitoringą iš vieno stebėjimo įrenginio,
- panaudoja ICMPv6 protokolą būsenai stebėti,
- integruoja būsenos pasikeitimo įvykius į NNM įvykių apdorojimo posistemę,
- pateikia grafinę IPv6 tinklo lygmens topologijos schemą,
- dinamiškai atvaizduoja IPv6 įrenginių būsenos pasikeitimus,
- palaikomi Hitachi, NEC, Juniper ir Cisco kompanijų įrenginiai.

Topologijos aptikimas

Topologijos aptikimas nėra absoliučiai automatinis, todėl prieš jį paleidžiant, reikalinga nustatyti kelis parametrus. Kad būtų aptikta kuo daugiau IPv6 įrenginių, galima nurodyti norimos įrangos IPv6 adresus, DNS vardus bei suteikti pavadinimus norimoms prefiksų grupėms. Taip pat sistema gali neatpažinti, kad keletas IPv6 adresų iš tikrųjų priklauso vienam įrenginiui ir atvaizduoti kelis įrenginius vietoje vieno. Siekiant to išvengti, taip pat reikia atlikti papildomą konfigūraciją.

Aptikimo procesas paima momentinius duomenis apie konkrečiu apklausimo momentu egzistuojančią topologiją. Duomenys traktuojami neatsižvelgiant į tai, kokie buvo prieš tai buvusių aptikimo ciklų rezultatai. Tai reiškia, kad išsaugoti bus tik tie ryšiai, kurie egzistuoja aptikimo ciklo metu (tarp ciklų atsiradę ryšiai ar įrenginiai bus pastebėti tik sekančio ciklo metu). Be to, kita informacija (įskaitant ir IPv6 adresus) surenkama tik iš tų įrenginių, kurie yra pasiekiami ciklo metu. „Pasiekiamas“ įrenginys turi atsakyti į SNMP užklausas iš NNM, todėl reikia NNM sukonfigūruoti taip, kad būtų naudojamos teisingos SNMP c*ommunity* eilutės.

Įvykdžius IPv6 automatinį topologijos aptikimo procesą, aptinkami tiek globalūs, tiek *site-local*, tiek *link-local* IPv6 adresai, tačiau kad automatinis aptikimas vyktų sklandžiai, visi įrenginiai turi veikti abiem protokolų rinkiniais (turi funkcionuoti tiek IPv4, tiek IPv6 adresai) [16].

Topologijos peržiūra

NNM IPv6 topologijos vaizdavimas paremtas aptiktų mazgų klasifikacija į grupes pagal prefiksus, kuriems priklauso IPv6 adresai. Topologijos peržiūra yra trijų tipų: globali peržiūra, prefiksų grupių peržiūra ir *site-local* tinklo peržiūra.

Globalios peržiūros žemėlapyje atsispindi visi mazgai, kurie turi sukonfigūruotus globaliai maršrutizuojamus IPv6 adresus. Vaizduojamos prefiksų grupės, prie kurių prijungiami joms priklausantys IPv6 mazgai. Galima pasirinkti, kad būtų matomi tik tinklo

įrenginiai arba visa tinklo topologija, įskaitant ir galinius vartotojų įrenginius. Parinkus konkretų įrenginį, matoma detali jo informacija bei pasiekiamumo parametrai.

Prefiksų grupių vaizdas rodo pačias prefiksų grupes kaip esybes, kurios sujungtos tarpusavyje. Jų būsenos yra gaunamos tikrinant visų tai grupei priklausančių interfeisų būsenas, t.y. esant bent vienam nepasiekiamam adresui, visa prefiksų grupė žemėlapyje žymima kaip nepasiekiama.

Site-local topologijos žemėlapis yra analogiškas globalių prefiksų žemėlapiui, tik šiuo atveju yra vaizduojami *site-local* prefiksai ir mazgai, turintys šio tipo IPv6 adresus [16].

Tinklo problemų diagnostika

Įrenginių pasiekiamumas tikrinamas periodiniais konfigūruojamo dažnumo ICMPv6 ping paketais. SNMP užklausos pasiekiamumo problemoms aptikti nenaudojamos.

Interfeisų būsena nustatoma pagal jiems priklausančių IPv6 adresų pasiekiamumą, o mazgo būsena analogiškai gaunama tikrinant jam priklausančių interfeisų būsenas. Prefiksų grupių būsenos nustatomos tiesiogiai iš jiems priklausančių adresų pasiekiamumo būsenų. Būsenų priklausomybė ir tarpusavio įtaka pavaizduota 2 lentelėje.

Lentelė Nr. 2. HP NNM būsenų tarpusavio įtaka

Prefiksų grupės būsena ←	**Adreso pasiekiamumo būsena**	**Interfeiso būsena** →	**Mazgo būsena** →
Būsena globaliame vaizde	Globalaus adreso būsena	Būsena globaliame vaizde	Būsena globaliame vaizde
Būsena site-local vaizde	Site-local adreso būsena	Būsena site-local vaizde	Būsena site-local vaizde
Bendra būsena	Globalaus + site-local adreso būsena	Bendra būsena	Bendra būsena

Pastebėjus būsenos pasikeitimą generuojamas naujas IPv6 įvykis, kuris išsaugomas ir vėliau gali būti peržiūrimas kartu parodant ir susijusią informaciją [16].

Išvados

HP „Network Node Manager“ – tai galinga, daug įvairių tinklo technologijų bei standartų palaikanti programinė sistema. Ji reikalauja daug aparatūrinės įrangos resursų savo užduotims įvykdyti, todėl geriausiai tinkama didelėms organizacijoms, turinčioms arba prižiūrinčioms stambius tinklus, kuriuose realizuota daug įvairių protokolų ir tinklo technologijų.

Sistema yra komercinė, todėl mažus ar vidutinės apimties tinklus naudojančioms organizacijoms ji nėra optimaliausias sprendimas ir dėl savo kainos.

Nors sistemos praplėtime maršrutizavimo stebėjimui realizuotas IPv6 topologijos palaikymas, tai nėra specializuota IPv6 tinklų stebėjimo sistema, todėl ji naudoja tik standartinius mechanizmus tiek topologijai aptikti, tiek stebėjimams atlikti. Kadangi būsenai stebėti nenaudojamas SNMP protokolas, negalimas asinchroninis klaidos pranešimas, kas stabdo klaidų tinkle aptikimą.

Sistema veikia tik HP-UX ir Solaris operacinėse sistemose, kurios abi yra komercinės, todėl gali būti didelė problema mažesnius biudžetus turinčioms organizacijoms.

2.2.3. Nagios monitoringo sistema

Nagios – tai nemokama (GPL licencija) programinė sistema, skirta sistemos ir tinklo monitoringui. Pagrindinis jos tikslas – stebėti nurodytus įrenginius, jų teikiamus servisus ir pranešti, kai pastebimas kurio nors parametro pasikeitimas. *Nagios* sukurta Linux operacinei sistemai, tačiau gali veikti ir kituose UNIX variantuose. Pagrindinės programos galimybės tiesiogiai susijusios su monitoringu yra šios [17]:

- Tinklo servisų monitoringas
- Galinio įrenginio resursų monitoringas
- Galimybė praplėsti programos funkcionalumą praplečiamomis paprogramėmis
- Lygiagrečios užklausos
- Galimybė nurodyti topologinę hierarchiją naudojant „tėvo“ įrenginius bei dėl suskirstymo atsirandanti galimybė atskirti neveikiančius įrenginius nuo nepasiekiamų
- Pranešimo siuntimas, kai įrenginio(-ių) būsena pasikeičia (e-paštu, pranešimų gavikliu ar kitu nurodytu metodu)
- Galimybė esant konkrečiam įvykiui paleisti atitinkamą paprogramę, padėsiančią išspręsti atsiradusią problemą

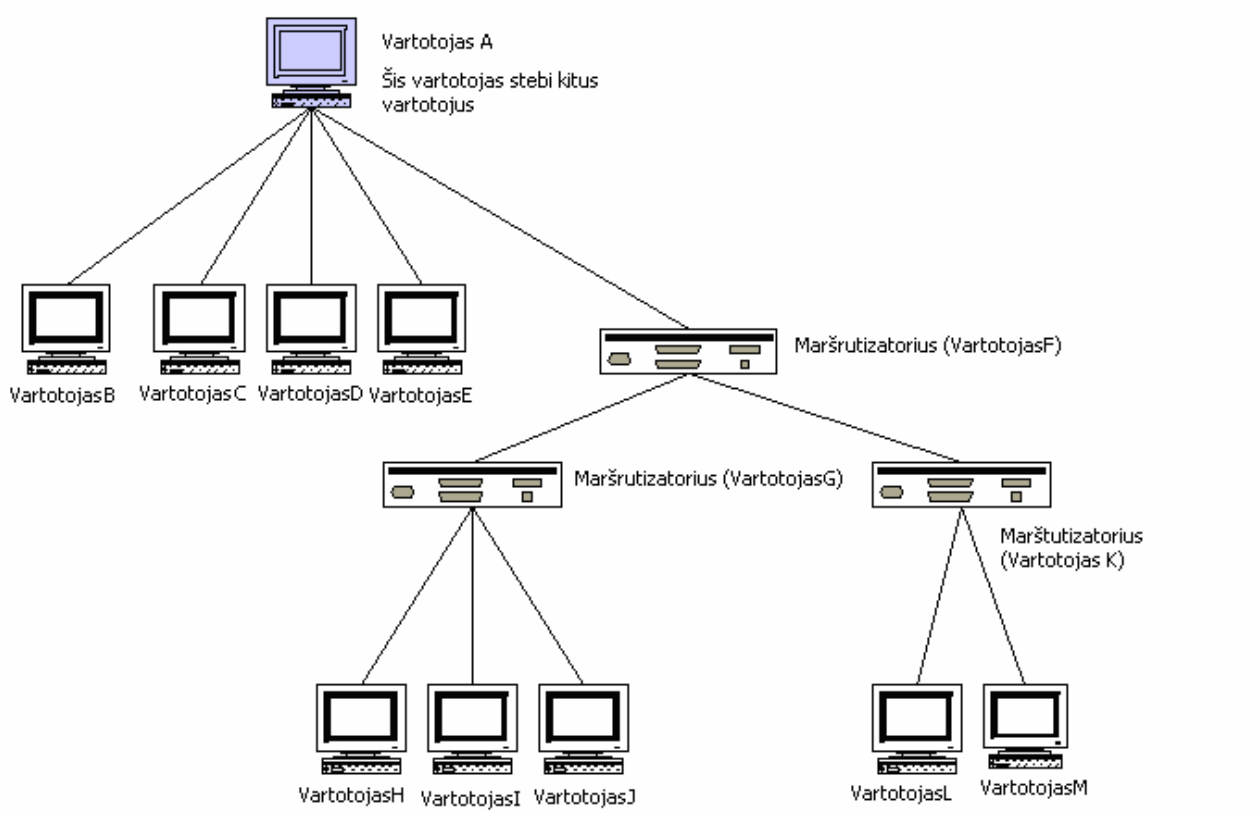
Topologijos sudarymas

Visi stebimi įrenginiai sistemoje laikomi objektais. Automatinio topologijos aptikimo galimybė nėra realizuota, todėl topologijos informaciją turi nurodyti administratorius.

Topologinis vaizdas sudaromas prie atitinkamų objektų nurodant jų „tėvo“ objektus. „Tėvo“ objektai dažniausiai būna maršrutizatoriai, komutatoriai, ugniasienės ir kt. įrenginiai, kurie yra tarp stebinčios sistemos ir stebimų objektų. Įrenginys, kuris yra topologiškai arčiausiai stebimojo objekto, laikomas to objekto „tėvu“. Jei stebimas įrenginys yra tame

pačiame tinklo segmente, kaip ir stebinčioji sistema, jis laikomas kaip vietinio tinklo objektas ir toks objektas neturės „tėvo“ objektų [17].

Tokiu būdu sudaroma medžio tipo hierarchinė topologijos schema:



3 pav. Medžio tipo tinklo hierarchija

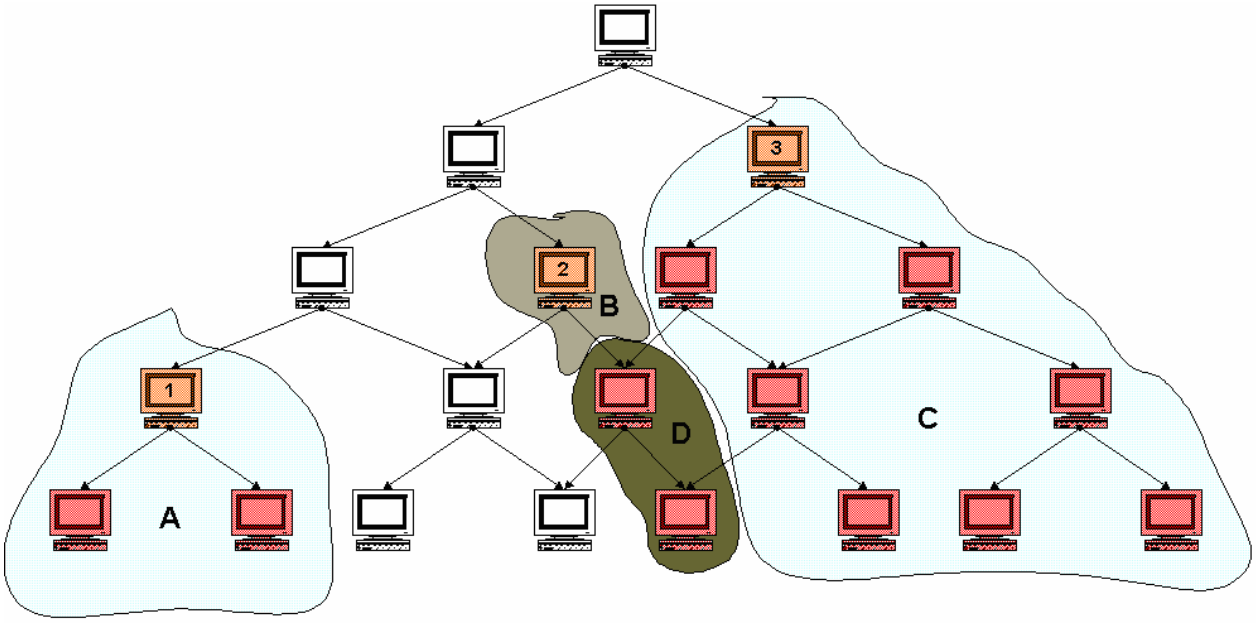
Tinklo nepasiekiamumo aptikimas

„Tėvo“ objektų nurodymas leidžia sistemai atskirti, kada stebimas objektas yra neveikiantis, o kada jis yra nepasiekiamas dėl tarpinių objektų neveikimo. Tokiu būdu atsiradus nepasiekiamumui, atitinkami įrenginiai pažymimi arba kaip nepasiekiami, arba kaip neveikiantys.

Pusprogramė *outages* gali atpažinti, dėl kurio įrenginio kiti įrenginiai tapo nepasiekiami. Toks įrenginys turi būti pažymėtas kaip neveikiantis arba nepasiekiamas ir bent vienas iš jo „tėvų“ turi būti pasiekiamas ir veikiantis. Objektai atitinkantys šiuos kriterijus pažymimi kaip galimi problemos šaltiniai. Jei visi pažymėtų įrenginių hierarchiškai žemesni objektai yra neveikiantys arba nepasiekiami ir neturi veikiančių „tėvo“ objektų, pažymėtasis objektas ir yra tinklo nepasiekiamumo priežastis.

Outages be galimo problemos šaltinio suradimo taip pat gali apytiksliai nurodyti atsiradusios problemos mastą. Pagal 4 pav. matyti, kad 1 objektas blokuoja du „vaikus“ (srityje A). Objektas 2 blokuoja tik pats save (sritis B), o objektas 3 atsakingas už septynių

hierarchiškai žemesnių objektų nepasiekiamumą (sritis C). Nepasiekiami yra ir du objektai srityje D, kuriuos blokuoja tiek objektas 2, tiek objektas 3, kadangi vienareikšmiškai pasakyti, kuris objektas atsakingas už šių įrenginių blokavimą, negalima [17].



4 pav. Tinklo neveikimo priežastys ir pasekmės

Outages padeda tinklo administratoriams greičiau atrasti tinklo neveiksnumo priežastis bei jas pašalinti. Ši pusprogramė gali tik nurodyti objektą, dėl kurio greičiausiai atsirado problema, tačiau negali nustatyti, kas konkrečiai ją sukėlė. Taigi, problemos išaiškinimas paliekamas administratoriui [17].

Praplėtimai

Nagios praplėtimai – tai sukompiliuotos programos arba pusprogramės (PHP, shell, ir kt.), kurios gali būti paleistos iš komandinės eilutės, kad patikrintų objekto ar serviso būseną. Sistema naudoja šių praplėtimų rezultatus ir be praplėtimų ji nefunkcionali [17].

Toks sistemos lankstumas leidžia pritaikyti ją darbui su IPv6 tinklais. *Nagiosplug* projektas buvo sukurtas tam, kad praplėsti standartinio *Nagios* monitoringo galimybes ir naujoje šio paketo versijoje jau yra keletas pusprogramių, palaikančių IPv6 protokolą [18]:

- IPv6 ping – palaikomas
- TCPv6 services – palaikomas
- UDPv6 services – palaikomas
- SNMPv6 – nepilnai palaikomas

Kadangi visas kodas yra atviras, administratoriai gali sukurti savo praplėtimus, padėsiančius jiems geriau pritaikyti sistemą savo reikmėms. Tam galima naudoti bet kokius programavimo įrankius arba tiesiog panaudoti bet kokią jau veikiančią programą, kuri išveda informaciją apie pasiekiamumą paprastu tekstu ar kita lengvai interpretuojama forma (skaičiumi arba programos užbaigimo kodu) [19].

Priedai

Nagios sistema turi papildomas programas, kurios, paleistos stebimuose įrenginiuose, gali pačios informuoti monitoringo sistemą apie konkretaus objekto būsenos pasikeitimus. Yra dvi tokio tipo programos: NRPE ir NSCA.

NRPE – tai priedas, įgalinantis centrinę sistemą paleidinėti praplėtimo paprogrames nutolusiame įrenginyje. *Check_nrpe* praplėtimas paleidžiamas centriniame *Nagios* mazge ir yra naudojamas praplėtimo paleidimo užklausoms siųsti NRPE agentui. Šis, gavęs tokią užklausą, paleidžia atitinkamą praplėtimą ir grąžina jo rezultatus centrinei *Nagios* sistemai. Čia juos gavęs *check_nrpe* praplėtimas persiunčia juos taip, lyg tai būtų jo paties rezultatai.

NSCA – tai praplėtimas, leidžiantis siųsti pranešimus iš nutolusio įrenginio į centrinę monitoringo sistemą apie to įrenginio būklę. Komunikacija tarp kliento ir serverio gali būti užšifruota [17].

Išvados

Nagios – tai pakankamai lanksti sistema, leidžianti pritaikyti jos teikiamas galimybes savo reikmėms tiek konfigūracijos pagalba, tiek sukuriant savo pačių praplėtimus bei panaudojant priedus. Ji nėra komercinė bei yra kuriama nekomercinėms operacinėms sistemoms, todėl populiariausia *Nagios* nekomercinėse arba nedidelio biudžeto organizacijose.

Nors *Nagios* yra numatyta galimybė stebėti tinklo topologiją, ji nėra orientuota į tokio tipo monitoringą: čia nėra automatinio topologijos aptikimo mechanizmo, problemų aptikimas apribotas tik galimo probleminio tarpinio mazgo aptikimu pagal hierarchinį topologijos medį, o ir pats topologijos sudarymas pagal „tėvus“ nesuteikia pakankamo lankstumo nurodant topologiją.

Minimalus IPv6 monitoringo palaikymas galimas panaudojant *nagiosplug* ir sukuriant savo praplėtimus, tačiau sistema iš esmės orientuota į IPv4, todėl pilnavertis jos pritaikymas IPv6 prilygtų naujos sistemos kūrimui.

2.2.4. Kitos programų sistemos

Be jau išnagrinėtų sistemų, egzistuoja ir daugiau tiek komercinių, tiek nemokamų monitoringo sprendimų, tačiau kol kas visi jie orientuoti į IPv4 tinklus, o IPv6 monitoringas, jei egzistuoja, tai realizuojamas kaip priedas. Populiariausios iš komercinių sistemų yra šios [20]:

- IBM NetView – IPv6 palaikymo nėra
- CiscoWorks – testuojama Campus Manager IPv6 versija. Orientuota į Cisco aparatūrinę įrangą.
- InfoVista – IPv6 palaikymas neplanuojamas

Egzistuoja ir kelios nemokamos sistemos, bent iš dalies palaikančios IPv6:

- Sysmon [21] – palaiko IPv6 ping užklausas,
- Argus [22] – palaikomos IPv6 *ping*, TCP, UDP ir SNMP užklausos bei administratorius gali pats pritaikyti savo praplėtimus,
- ASpath-tree [23] – realizuotas IPv6 BGP lentelės grafinis vaizdavimas, parodant stabilumo statistiką ir neįprastus maršrutų skelbimus.

2.3. Programų sistemų savybių kiekybinis ir kokybinis palyginimas

Lyginti konkrečių programinių sistemų nėra prasmės, kadangi jų skaičius pakankamai didelis, o savybės pakankamai panašios, todėl tikslinga išskirti dvi sprendimų grupes: produktai, orientuoti į stambius tinklus turinčias organizacijas ir į valdančias vidutinius arba mažus tinklus. Sistemų palyginimas pateikiamas 3 lentelėje.

Lentelė Nr. 3. Sistemų palyginimas

Kriterijus	Stambioms organizacijoms skirti produktai	Smulkioms organizacijoms skirti produktai
Kaina	Didelė. Tai paprastai komerciniai sprendimai.	Maža. Tai dažniausiai nemokami produktai, kuriuos reikia pritaikyti savo reikmėms.
Operacinės sistemos	Komercinės operacinės sistemos (Solaris, BSD 4.2, HP-UX, Windows).	Nemokamos atviro kodo operacinės sistemos (Linux, Free/Open/NetBSD).
Stebimi įrenginiai	Dažniausiai aparatūriniai maršrutizatoriai (Cisco, Juniper, HP, Hitachi).	Paprastai orientuota į PC tipo įrenginius, veikiančius kaip maršrutizatoriai.
Topologijos automatinis aptikimas	Realizuotas. Išnaudojamos specifinių gamintojų sukurtos priemonės įrangos atpažinimui.	Dažniausiai nerealizuotas. Stebimi įrenginiai turi būti konfigūruojami statiškai.
Peržiūra	Pilnai grafinė. Yra galimybė grafiškai konfigūruoti topologiją.	Statiškai generuojami topologijos žemėlapiai arba tik tekstinė informacija.
Problemų aptikimas	Naudojami periodiniai apklausimai. Galimi asinchroniniai SNMP pranešimai.	Naudojami periodiniai apklausimai. Galimi asinchroniniai papildomos programinės įrangos arba SNMP pranešimai.
Galimybė gauti informaciją iš nutolusių objektų	Tik per SNMP pranešimus. Nėra galimybės įterpti administratoriaus praplėtimų.	Galimi SNMP pranešimai bei administratoriaus praplėtimai.
IPv6 palaikymas	Nėra arba kaip priedas prie esamos IPv4 infrastruktūros.	Nėra arba kaip priedas prie esamos IPv4 infrastruktūros.
IPv6 informacijos gavimo būdai	ICMP, SNMP, Neighbour Discovery.	Dažniausiai ICMP bei TCP ir UDP užklausos.

2.4. Išvados

Šiuo metu tinklo monitoringas yra vienas iš pagrindinių gerą tinklo veikimo kokybę užtikrinančių procesų. IPv4 paremtų tinklų monitoringo produktų pasirinkimas labai platus ir įvarus tiek kalbant apie jų kainą, jų kokybę ar jų stebimo tinklo dydį.

IPv6 tinko monitoringo programos – tai daugiausia IPv4 monitoringui sukurtos sistemos, kurios vėliau buvo pritaikytos IPv6 tinklams stebėti. Nors yra ir kokybiškų monitoringo sistemų, jos neišnaudoja kai kurių standartinėje IPv6 specifikacijoje numatytų protokolo savybių, galinčių palengvinti monitoringo užduotis ir pagreitinti problemos aptikimą. Neišnaudojamas kaimynų aptikimo mechanizmas (IPv4 ARP protokolo pakaitalas) ir jam priklausantis kaimynų nepasiekiamumo aptikimas, neturintis būsenų automatinis adresų ir kitos informacijos automatinis konfigūravimas (*stateless address autoconfoguration*) ir kt.

IPv6 yra palyginti naujas protokolas, todėl daugelis interneto paslaugų tiekėjų nenori ir/arba neturi galimybės naudoti IPv6 visuose savo tinklo segmentuose, tačiau jaučia poreikį turėti bent iš dalies veikiantį IPv6 paremtą tinklą. IPv6 palaikančios aparatūrinės įrangos trūkumas sprendžiamas sujungiant IPv6 tinklo saleles tuneliais, transportuojant srautą IPv4 tinklu. Dabartinės IPv6 monitoringo sistemos neatsižvelgia į tokio tipo topologijos ypatumus, todėl dėl gedimo tuneliuotuose sujungimuose atsirandančios paslaugos pateikiamumo problemos dažnai neidentifikuojamos.

Kad bent iš dalies pašalinti tokius trūkumus, reikia sukurti vien į IPv6 paremtų tinklų monitoringą orientuotą programinę įrangą, kuri atsižvelgtų tiek į IPv6 protokolo, tiek į tinklo topologijos ypatumus.

3. MONITORINGO SISTEMOS PROJEKTAS

3.1. Sistemos paskirtis

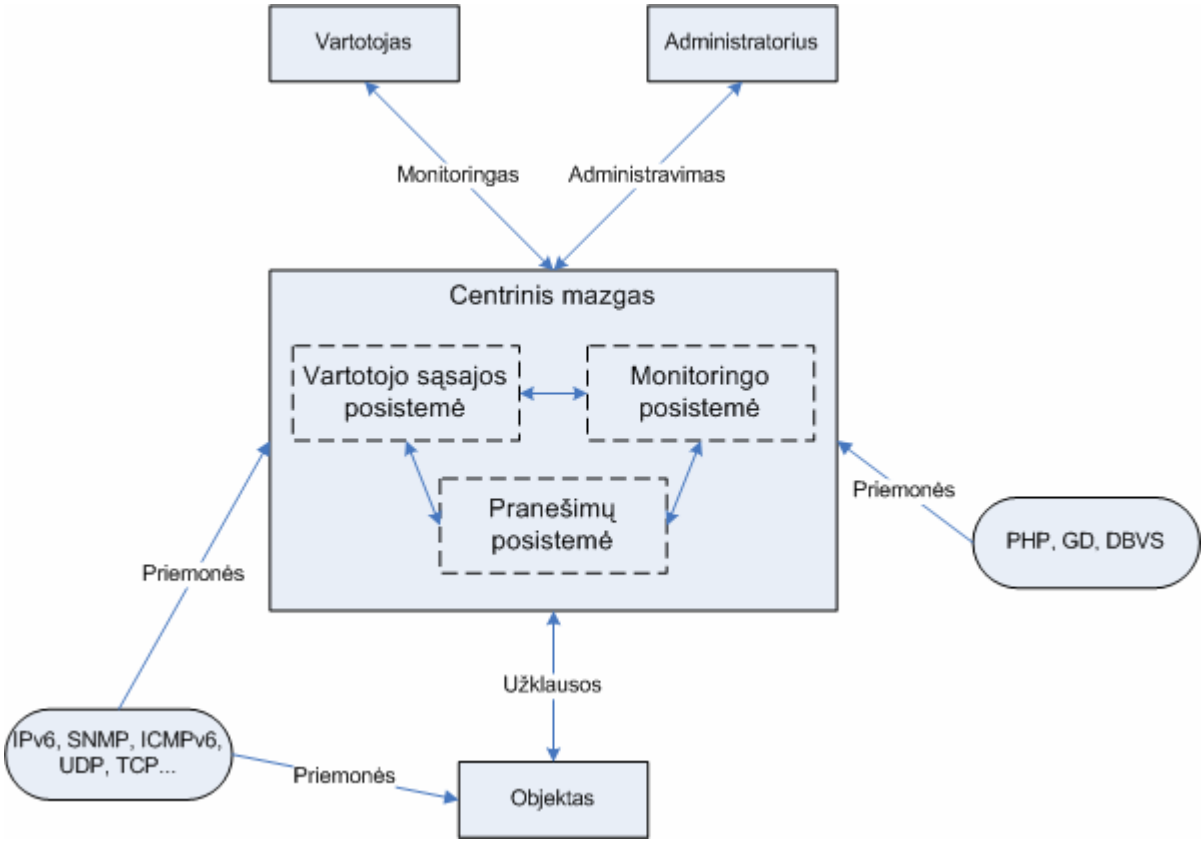
Sistema orientuota konkrečiai į IPv6 tinklo monitoringą ir diagnostiką, todėl ji pagrinde remiasi įvairiais vien tik šio protokolo parametrais bei jo veikimo principu. Tai padeda geriau ir greičiau išspręsti tokiame tinkle iškilusias problemas bei pastoviai stebėti jo funkcionalumą.

Šiuo metu IPv6 tinklų operatoriai jau naudoja vienokius ar kitokius monitoringo įrankius. Šie yra pakankamai tobuli ir lanksčiai konfigūruojami, kad laiku praneštų apie tinkle atsiradusius nesklandumus. Daugelis šių įrankių yra daugiafunkciniai, taigi, pagal jų suteikiamą informaciją galima matyti ne tik laikinus sutrikimus, bet ir sudarinėti tinklo veikimo kokybės tendencijas, pagal kurias po to lengviau tiek kurti tinklo tobulinimo arba plėtimo strategijas, tiek gilintis į atitinkamas paties IPv6 protokolo vystymo kryptis.

Šiuo metu esančios IPv6 monitoringo programų sistemos nėra pakankamai gerai ištobulintos, kad tenkintų šiuolaikinių tinklų reikalavimus. Taip yra visų pirma todėl, kad tokio tipo sistemos jų kūrimo pradiniame etape buvo orientuotos į „senųjų“ IPv4 tinklų monitoringą ir tik vėliau, paplitus IPv6, buvo pritaikytos ir šių tinklų stebėjimui. Kadangi IPv6 turi daugumą parametrų, kuriuos turi IPv4, tokių sistemų darbas iš esmės apsiriboja tik šių bendrųjų parametrų stebėjimu ir analizavimu.

Sistemos kūrimo pagrindinis tikslas – efektyviai ir kuo mažiau apkraunant kompiuterių tinklą surinkti informaciją apie jo topologiją bei įrenginių pasiekiamumą, atlikti atitinkamus skaičiavimus, parodančius tinklo atskirų įrenginių pasiekiamumo ar kitokias statistikas ir pateikti šią informaciją vartotojui patogia forma. Visuose savo darbo etapuose sistema turėtų išnaudoti IPv6 protokolo galimybes, kad informacija būtų kiek įmanoma pilna ir tiksli. Dar vienas reikalavimas sistemai –kuo efektyvesnis nesklandumų tinkle aptikimas ir pranešimas apie juos vartotojui, tiksliai identifikuojant šių nesklandumų atsiradimo laiką, vietą ir tikėtinas priežastis.

3.2. Sistemos kontekstas



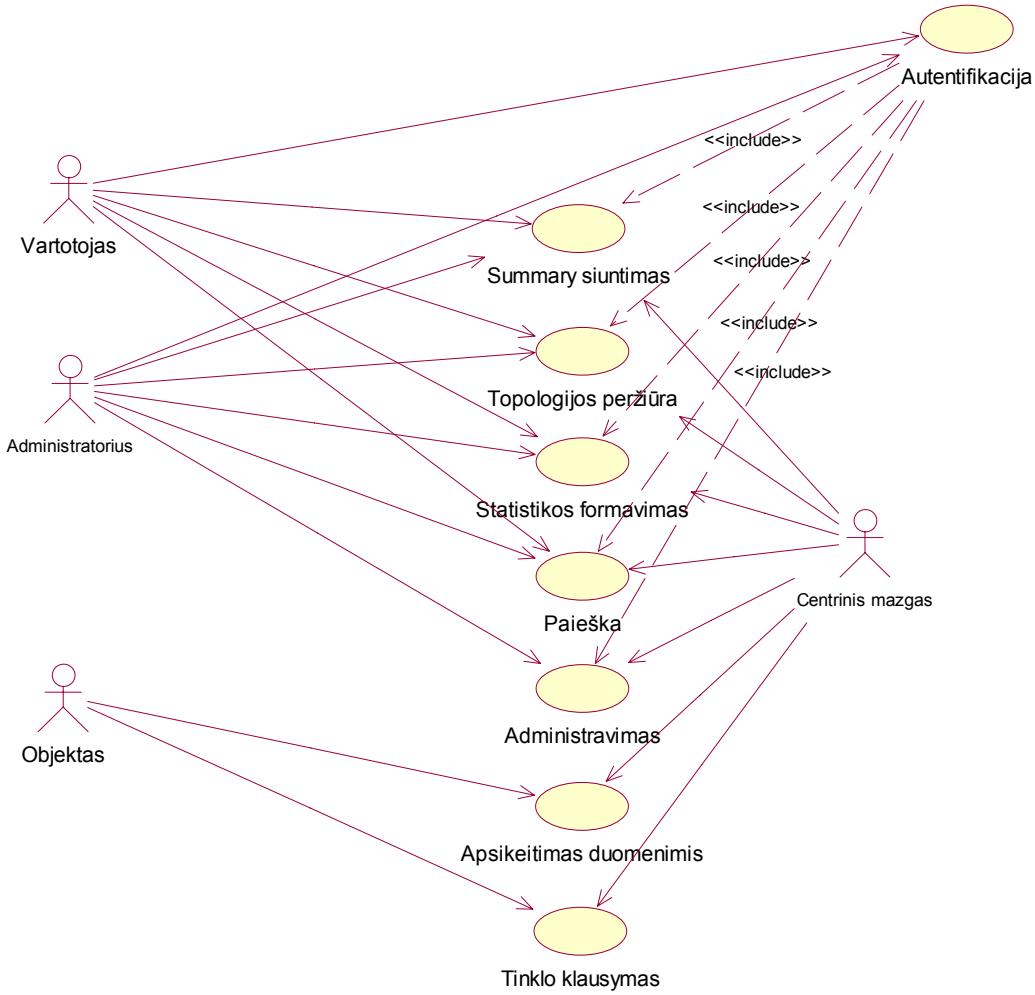
5 pav. Sistemos kontekstinė schema

Lentelė Nr. 4. Veiklos padalinimas

Eil. Nr.	***Įvykio pavadinimas***	***Įeinantys/Išeinantys informacijos srautai***
1	Centrinis mazgas apdoroja vartotojo užklausą ir grąžina rezultatą.	Užklausos duomenys (įeinantis) Grąžinamas rezultatas (išeinantis)
2	Centrinis mazgas periodiškai siunčia užklausas (TCP, UDP, ICMP, SNMP) objektams ir laukia atsakymo.	Atsakymai į užklausas (įeinantis) Įvairių Protokolų užklausos (išeinantis)
3	Administratorius konfigūruodamas sistemą siunčia konfigūracines užklausas.	Konfigūracinių duomenų užklausos (įeinantis) Duomenų gavimo patvirtinimas (išeinantis)
4	Centrinis mazgas klausosi tinklo, tikėdamasis gauti IPv6 paketus, iš kurių gali spręsti apie tinklo būklę.	IPv6 paketai (ND, RA, ...) (įeinantis)
5	Vartotojas, naudodamasis sistema, siunčia užklausas.	Užklausų duomenys (įeinantis) Suformatuoti puslapiai (išeinantis)

6	Aptikęs būsenos pasikeitimą, centrinis mazgas siunčia pranešimą vartotojams e-paštu.	Būsenos pasikeitimo pranešimas (išeinantis)
7	Centrinis mazgas periodiškai siunčia apibendrintus tinklo būklės pranešimus vartotojams (elektroniniu paštu).	Apibendrinti pranešimai (išeinantis)

3.3. Sistemos panaudojimo atvejai



6 pav. Sistemos panaudojimo atvejų vaizdas

3.3.1. Sistemos aktorių aprašymas

Vartotojas. Vartotojas yra bet kuris asmuo, turintis galimybę prisijungti prie sistemos tam, kad peržiūrėtų jos teikiamą informaciją. Į vartotoją orientuota bet kurio tipo sistemos teikiama informacija. Vartotoju realioje situacijoje turėtų būti tinklo, kurį stebi sistema, administratorius, tačiau šio aktoriaus galimybės yra ribotos, kadangi jis gali tik stebėti

sistemos teikiamą informaciją, bet negali keisti sistemos konfigūracijos.

Administratorius. Administratorius yra sistemą valdantis aktorius. Jo galimybės, lyginant su vartotojo, yra praplėstos, todėl jis gali visapusiškai keisti sistemos parametrus, pridėti naujus objektus, redaguoti jau stebimus bei trinti nereikalingus. Taip pat administratorius gali redaguoti vartotojus, t.y. pridėti naujus, keisti jų parametrus bei juos ištrinti.

Centrinis mazgas. Centrinis mazgas – tai sistemos pagrindas. Be jo neįmanomas sistemos funkcionavimas, kadangi bet kuris panaudojimo atvejis ir bet kuri realizuota funkcija tiesiogiai susijusi su šiuo aktoriumi. Informacijos rinkimas, apdorojimas ir pateikimas vartotojui yra pagrindinės centrinio mazgo funkcijos. Sistemos konfigūravimas taip pat atliekamas centriniame mazge.

Objektas. Objektu gali būti bet kuris bent minimaliai tinklo darbą užtikrinantis aparatūrinis įrenginys. Vartotojas ir administratorius negali gauti informacijos tiesiogiai iš objekto, bet objektas gali apsikeisti monitoringo informacija su centriniu mazgu. Pagrindinės objektų funkcijos yra leisti centriniam mazgui paimti monitoringui reikalingą informaciją ir, jei tai įmanoma, pranešti centriniam mazgui apie atsiradusius parametrų pakitimus asinchroniškai.

3.3.2. Panaudojimo atvejų aprašymas

Autentifikacija. Skirtas autentifikuoti prisijungusį vartotoją, tikrinant jo įvestus duomenis (vartotojo vardą ir slaptažodį), ir priskirti prisijungusiam vartotojui atitinkamas teises.

Summary siuntimas. Skirtas periodiškai nustatytu intervalu siųsti vartotojams apibendrintus pranešimus apie tinklo būseną ir pasikeitimus (savaitės, mėnesio, metų apibendrinta statistika).

Topologijos peržiūra. Sugeneruojamas ir vartotojui parodomas stebimo tinklo topologinis vaizdas, sudarant galimybę detalizuoti rodomą vaizdą iki konkretaus objekto vaizdo. Rodomame žemėlapyje turi atsispindėti geografinių segmentų arba objektų (priklausomai nuo parinktos žemėlapio detalizacijos) būsenos.

Statistikos formavimas. Skirtas grafiškai ir skaitmenine forma parodyti sugeneruotus pasirinkto objekto pasiekiamumo statistinius rodiklius.

Paieška. Suteikia vartotojui galimybę ieškoti regiono, objekto, adreso ar užklausos pagal nurodytus parametrus.

Administravimas. Tikslas – pakeisti pasirinkto vartotojo/objekto duomenis, ištrinti egzistuojantį vartotoją/objektą arba pridėti naują, taip pat redaguoti stebimus objektus ir jų parametrus.

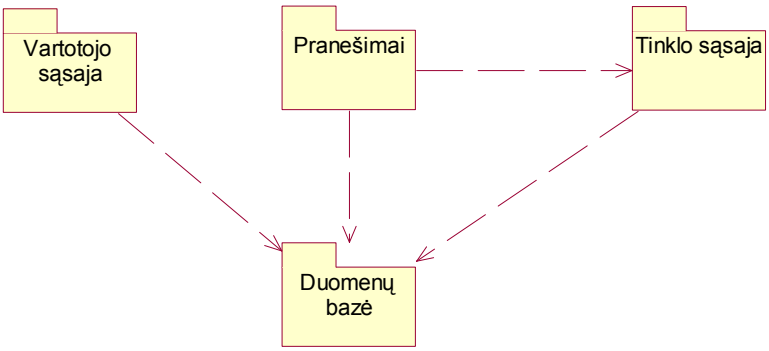
Apsikeitimas duomenimis. Skirtas apsikeisti duomenimis tarp centrinio mazgo ir atitinkamo objekto, periodiškai siunčiant užklausas ir laukiant atsakymo (aktyvus monitoringas).

Tinklo klausymas. Pasyviai klausantis tinklo, aptinka IPv6 signalinius paketus ir juos užfiksuoja (pasyvus monitoringas).

3.4. Sistemos architektūra

3.4.1. Sistemos komponentai

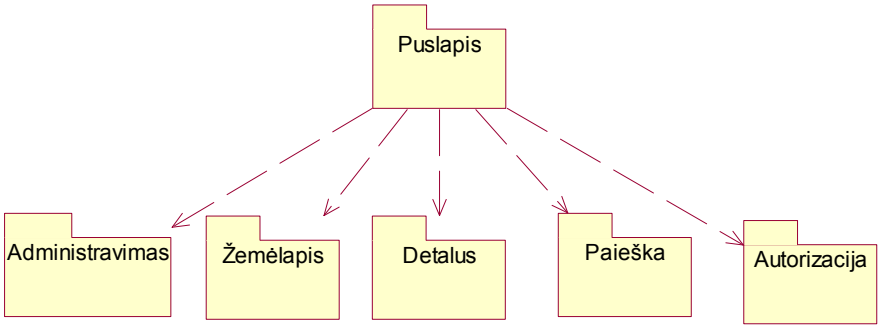
Sistema suskaidyta į paketus, pavaizduotus 7 pav.



7 pav. Sistemos paketai

3.4.1.1. Vartotojo sąsajos paketas

Paketas atlieką visą sistemos vartotojo sąsajos generavimą. Šis paketas sudarytas iš smulkesnių paketų, atliekančių tam tikras peržiūros arba administravimo funkcijas.



8 pav. Vartotojo sąsajos paketo detalizacija

Puslapis. Šis paketas yra valdantis vartotojo sąsajos generavimo elementas. Jis sudarytas iš pagrindinės klasės „GUI generate“, kurią paveldi meniu generavimo klasė „GUI_menu“. Klasė „GUI_generate“ iškviečia atitinkamas administravimo, peržiūros arba paieškos klases.

Administravimas. Paketas skirtas vartotojams bei stebimiems objektams administruoti. Naudoja išorinę paketo „Duomenų bazė“ klasę „db_interface“, kuri skirta darbui su duomenų bazės įrašais. Taip pat naudojamas paketas „Autorizacija“, tikrinantis, ar vartotojas turi reikiamas teises.

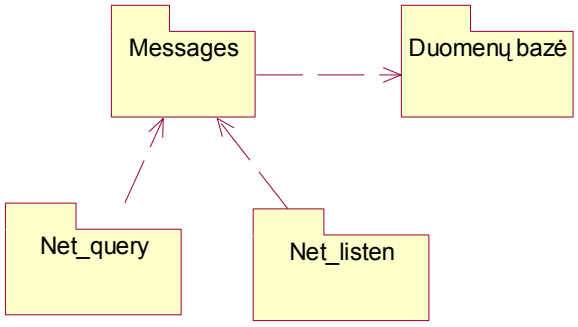
Žemėlapis. Paketas skirtas stebimos topologijos žemėlapiui generuoti. Naudojama duomenų bazės klasė bei autorizacijos paketas. Klasė „GUI_map_regions“ generuoja bendrą regionų vaizdą, o klasė „GUI_map_objects“ – pasirinktam regionui priklausančių objektų topologinį vaizdą.

Detalus. Paketas skirtas detaliai regionų, vieno regiono objektų arba konkretaus objekto informacijai matyti. Informaciją sudaro: objektų būsenos, adresų, užklausų, kaimynų bei pateikiamumo informacija. Naudojamas autorizacijos paketas bei duomenų bazės klasė.

Paieška. Paketas atlieka paiešką duomenų bazėje pagal nurodytus kriterijus ir generuoja rezultatų puslapį. Be duomenų bazės klasės, paieškos klasė naudoja autorizacijos paketą.

Autorizacija. Paketas skirtas prisijungiantiems vartotojams verifikuoti bei jų teisėms sistemoje nustatyti. Naudoja išorinę paketo „Duomenų bazė“ klasę „db_interface“, kuri skirta darbui su duomenų bazės įrašais.

3.4.1.2.Pranešimų paketas

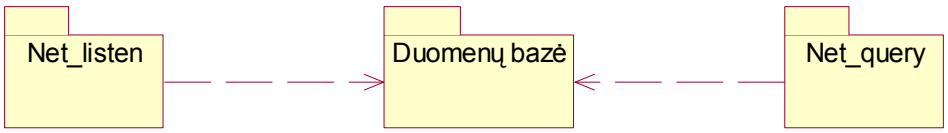


9 pav. Pranešimų paketo detalizacija

Messages. Paketas sudarytas iš valdančios klasės: „Net_message_manager“, reguliuojančios pranešimus siunčiančias klases. Klasė „Net_message_summary“ siunčia periodinius apibendrintus pranešimus, o klasė „Net_message_notify“ skirta pranešimams apie būsenos pakitimus siųsti. Abiejų tipų pranešimai siunčiami vartotojui elektroniniu paštu.

3.4.1.3.Tinklo sąsajos paketas

Paketas skirtas duomenų apsikeitimui tarp centrinio mazgo ir stebimų objektų realizuoti. Tai monitoringą atliekantis paketas. Informacijai saugoti naudojama klasė „db_interface“. Pats paketas sudarytas iš aktyvaus monitoringo paketo „Net_query“ bei pasyvaus monitoringo paketo „Net_listen“.



10 pav. Tinklo sąsajos paketo detalizacija

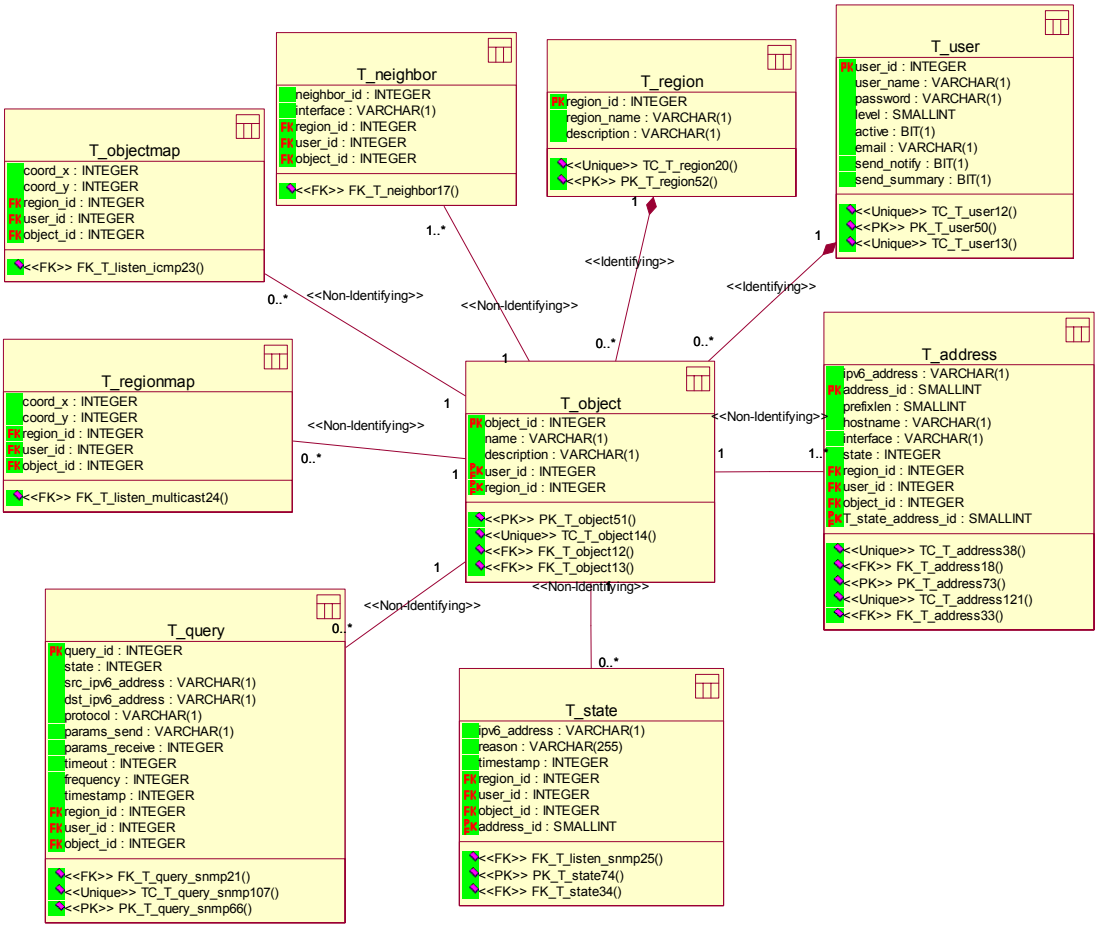
Net_listen. Paketas sudarytas iš valdančios klasės „Net_listen_manager“, kuri išskirsto gautus paketus pagal tipus bei iškviečia atitinkamą apdorojimo klasę. Duomenų saugojimui naudojama paketo „Duomenų bazė“ klasė „db_interface. Klausomasi ICMP, *multicast* ir SNMP paketų (traps).

Net_query. Paketas skirtas aktyviam monitoringui realizuoti. Jis sudarytas iš valdančiosios klasės „Net_query_manager“, reguliuojančios užklausų siuntimą. Duomenų saugojimui naudojama išorinė paketo „Duomenų bazė“ klasė „db_interface“. Užklausos siunčiamos ICMP, TCP, UDP bei SNMP protokolais, kurių kiekvieną realizuoja atskira klasė.

3.4.1.4.Duomenų bazės paketas

Šis paketas skirtas konfigūracijos bei surenkamai informacijai saugoti bei pateikti ją kitoms klasėms. Klasė „db_interface“ valdo DBVS, o klasė „konfigūracija“ skirta sistemos konfigūraciniam failui skaityti bei globaliai konfigūracijai (tokiai, kaip DBVS vartotojo vardas bei slaptažodis) gauti.

3.4.2. Duomenų vaizdas



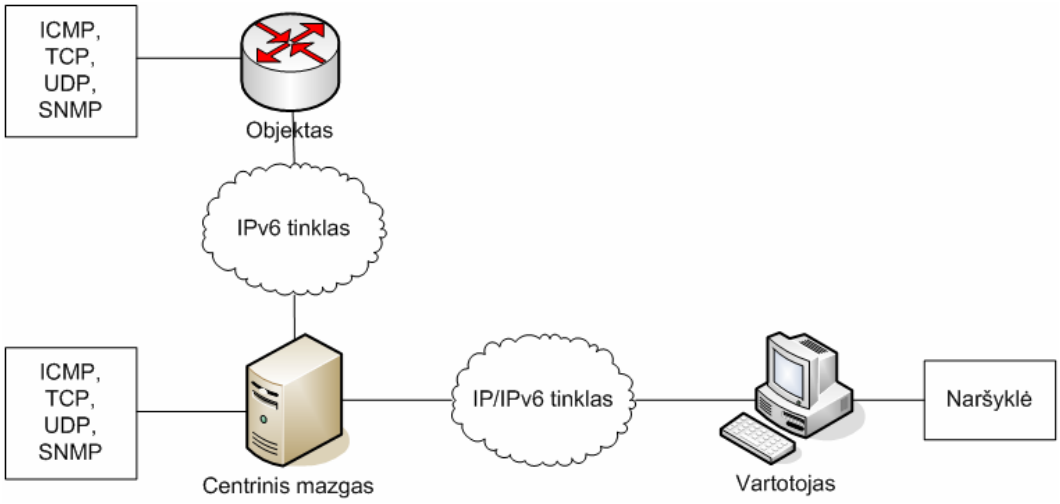
11 pav. Duomenų vaizdo schema

Lentelė Nr. 5. Esybių aprašymas

Esybė	**Aprašymas**
T_user	Sistemos vartotojų informacija. Saugomi vartotojo duomenys, jam prisikirtas lygmuo ir jo noras arba nenoras gauti pranešimus el. paštu.
T_object	Pagrindinė lentelė. Saugoma pagrindinė informacija apie stebimą objektą,siejanti jį su kitomis esybėmis.
T_region	Saugoma informacija apie regioną. Vienas objektas gali priklausyti tik vienam regionui.
T_neighbor	Saugoma informacija apie konkretaus objekto kaimynus (jei tinklo segmentas yra *multiaccess*, rodomas ryšys tik su tame tinkle esančiais maršrutizatoriais).
T_address	Objekto IPv6 adresų informacija. Saugomi visi objektą identifikuojantys IPv6 adresai bei interfeisai, kuriems jie priklauso. Taip pat nurodoma konkretaus adreso pasiekiamumo informacija duotuoju momentu.

T_state	Registruojami visi visų adresų bei užklausų būsenų pasikeitimai, nurodant: kokio adreso ar užklausos būsena pasikeitė, į kokią būseną pasikeitė, kodėl pasikeitė, kada registruotas pasikeitimas
T_query	Užklausų lentelė. Čia registruojamos visos siunčiamos aktyvaus monitoringo užklausos ir nurodomi visi tiek siunčiami, tiek laukiami parametrai. Siųsti galima ICMP, TCP, UDP ir SNMP užklausas. Čia taip pat registruojami ir laukiami SNMP trap pranešimai, kurie aprašomi kaip užklausos, tačiau nėra siunčiami, o tik laukiami.

3.4.3. Išdėstymo vaizdas



12 pav. Išdėstymo schema

Naršyklė. WWW naršyklės pagalba galima naudotis sistemos funkcijomis. WWW naršyklė gali būti bet kokia HTTP ir/arba HTTPS protokolą palaikanti programa. Reikalavimai vartotojo sistemai priklauso nuo naudojamos naršyklės. Bendras reikalavimas – centrinio mazgo pasiekiamumas duomenų perdavimo tinklu.

Vartotojas. Tai sistemos įrenginys, kuris gali būti bet kur internete, kol turi kontaktą su centriniu mazgu duomenų perdavimo tinklu. Tai paprastas kompiuteris, kuriame turi būti įdiegta WWW naršyklė.

Centrinis mazgas. Pagrindinis sistemos įrenginys, kuris valdo visą sistemos darbą. Centrinis mazgas kartu yra ir apklausą vykdantis įrenginys, ir gautus duomenis apdorojantis įrenginys, ir rezultatus vartotojui pateikiantis įrenginys.

Objektas. Stebimas tinklo įrenginys. Tai gali būti tiek kompiuteris, tiek aparatūrinis maršrutizatorius.

4. SISTEMOS TOBULINIMO TYRIMAS

4.1. Dabartinė situacija

Pagrindinis bet kurios monitoringo sistemos tikslas – greitas objekto (adreso ar užklausos) būsenos pasikeitimo aptikimas, taigi, net esant nedideliam objektų kiekiui, bet kurio iš jų neveikimą reikia pastebėti per sistemos vartotojus tenkinantį laiko tarpą.

Sukurtoji sistema skirta nedidelio objektų skaičiaus monitoringui, nors topologijos sudėtingumas (tarpinių mazgų skaičius, ryšių skaičius) beveik neįtakoja būsenos pasikeitimo aptikimo efektyvumui.

Metodas, sukurtoje sistemoje naudojamas duomenų surinkimui iš stebimų objektų, yra centralizuotas monitoringas. Kaip minėta anksčiau (žr. 2.1.1 skyrių), centralizuoto monitoringo atveju duomenų apsikeitimas vykdomas naudojant kliento-serverio architektūrą. Sistemos efektyvumas (bet kurio objekto bet kurios būsenos pasikeitimo aptikimo laikas T, tinklo srauto išnaudojimas C ir centrinio mazgo bei stebimų objektų apkrovimas U) yra tiesiogiai priklausomi nuo stebimų objektų skaičiaus.

4.2. Sistemos patobulinimai

Pagrindiniai trūkumai naudojant centralizuotą duomenų surinkimo metodą yra:

- lankstumo nebuvimas;
- visos sistemos neveiksnumas dėl vieno elemento (centrinio mazgo) gedimo;
- tik vienas apklausiantis agentas (ne visada laiku pastebimas gedimas tinkle).

Kadangi sistema orientuota į nedidelio objektų kiekio stebėjimą, lankstumas (*scalability*) nėra būtinas. Staigaus ir/arba didelio objektų skaičiaus augimo sistemai veikiant neturėtų būti.

Lengviausias būdas užtikrinti, kad sistema nenustos veikusi, jei neveikia tinklą apklausiantis įrenginys – padidinti tokių įrenginių kiekį. Taikant hierarchinį duomenų surinkimo metodą, duomenų surinkimo pareigos yra padalinamos keletui tarpinio lygmens agentų, kurie pasidalina visus stebimus objektus ir kiekvienas apklausinėja savo objektų dalį (žr. 2.1.2 skyrių). Tokiu atveju, įvykus gedimui viename iš agentų, kiti vis tiek veikia ir prarandama tik neveikiančio agento stebimų objektų būsenos informacija.

Kad dar labiau sumažinti galimą informacijos praradimą, galima atsiradusiems tarpiniams agentams patikėti viso tinklo stebėjimą. Tokią sistemą tikslingiau vadinti ne paskirstyta, o replikuota, kadangi agentai nesidalina darbo, o atlieka tas pačias užklausas pakartotinai.

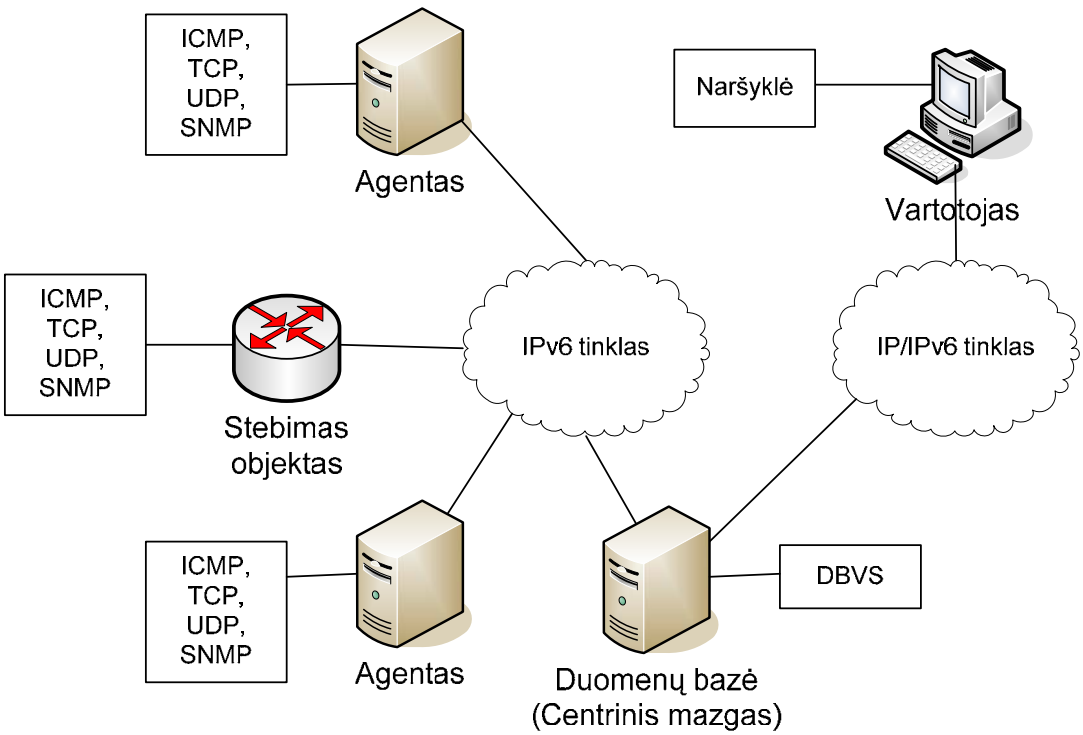
Esant replikuotai sistemai, visi stebintys agentai turi informaciją apie visus stebimus objektus (visos topologijos vaizdą iš savo pozicijos toje topologijoje). Tuo pačiu išsprendžiama ir topologijos vaizdo tik iš vieno taško problema, kadangi esant keletui visą topologiją stebinčių mazgų, ši stebima iš daugelio perspektyvų, todėl galima susidaryti pilnesnį tinklo funkcionavimo vaizdą.

4.2.1. Patobulinimai išdėstyme

Duomenys apie stebimus objektus ir reikiamų siųsti užklausų informacija laikoma centrinėje duomenų bazėje, kuri gali būti patalpinta tiek atskirame įrenginyje, tiek viename iš tinklą apklausiančių agentų. Duomenų bazę talpinančiame įrenginyje tikslinga laikyti ir vartotojo sąsajos programinę įrangą (WWW serverio rezidentinę programą ir pusprogrames, aptarnaujančias vartotojų užklausas), kadangi ši įranga dažniausiai kreipsis į duomenų bazę.

Kiekvienas agentas pradeda darbą atitinkamu dažniu iš centrinio mazgo pasiima informaciją apie stebimus objektus (adresai, užklausos) ir pradeda apklausinėti visą tinklą, saugodamas būsenų informaciją savo lokalioje saugykloje. Būsenų informacija taip pat bus perduodama į centrinę duomenų bazę.

Vartotojas, prisijungęs prie centrinio mazgo per vartotojo sąsają, gali matyti tinklo būklę bei turi galimybę ne tik peržiūrėti apibendrintą tinklo būklę, bet ir statistiką pagal bet kurio pasirinkto agento individualiai surinktus duomenis (topologijos vaizdą to agento atžvilgiu).



13 pav. Modifikuotos sistemos išdėstymo schema

Siūlomas modelis yra žymiai atsparesnis sutrikimams tiek aparatūrinėje įrangoje, tiek duomenų tinkle.

- Jei sistemai dirbant atsirastų gedimas viename iš tinklą stebinčių agentų, kaip ir hierarchinio monitoringo atveju, dingtų to agento aptarnaujamų objektų būklės informacija. Visgi, kadangi visi agentai stebi visus objektus, realiai būtų prarandama tik informacija apie topologiją iš neveikiančio agento perspektyvos; objektų būsenos informacija iš kitų agentų perspektyvos vis dar būtų prieinama.
- Atsiradus gedimui įrenginyje, kuriame patalpinta centrinė duomenų bazė, agentai liktų atkirsti nuo informacijos apie objektus bei būsenas. Ypatingai tai svarbu, jei agentas pradėtų darbą esant gedimui duomenų bazės įrenginyje. Kad išvengti problemų, agentai turi turėti lokalią visos duomenų bazės kopiją bei ją periodiškai atnaujinti. Tai nesunku pasiekti, panaudojant DBVS replikavimo funkcionalumą. Tokiu atveju visi pakitimai pagrindinėje duomenų bazėje bus perduodami visiems agentams, o jeigu centrinėje duomenų bazėje įvyktų gedimas, agentai tęstų darbą pagal lokalios duomenų bazės informaciją. Jei kuris nors agentas persikrautų, lokalios duomenų bazės turinys vis tiek išliktų, todėl, net ir neveikiant pagrindinei duomenų bazei, agento darbas nesutriktų.
- Tikėtinas ir ryšio tarp centrinio mazgo (centrinės duomenų bazės) ir agento(-ų) sutrikimas. Tokiu atveju, kol nėra ryšio, agentai vykdytų užklausas pagal lokalias informacijos apie objektus kopijas ir, jei pagrindinėje duomenų bazėje įvyko pakitimai, informacija apie juos būtų perduota agentams, sinchronizuojantis duomenų bazėms, kai ryšys atsiras. Taigi, agentų užklausų informacija blogiausiu atveju bus pasenusi, bet nebus prarasta.

4.2.2. Patobulinimai duomenų bazėje

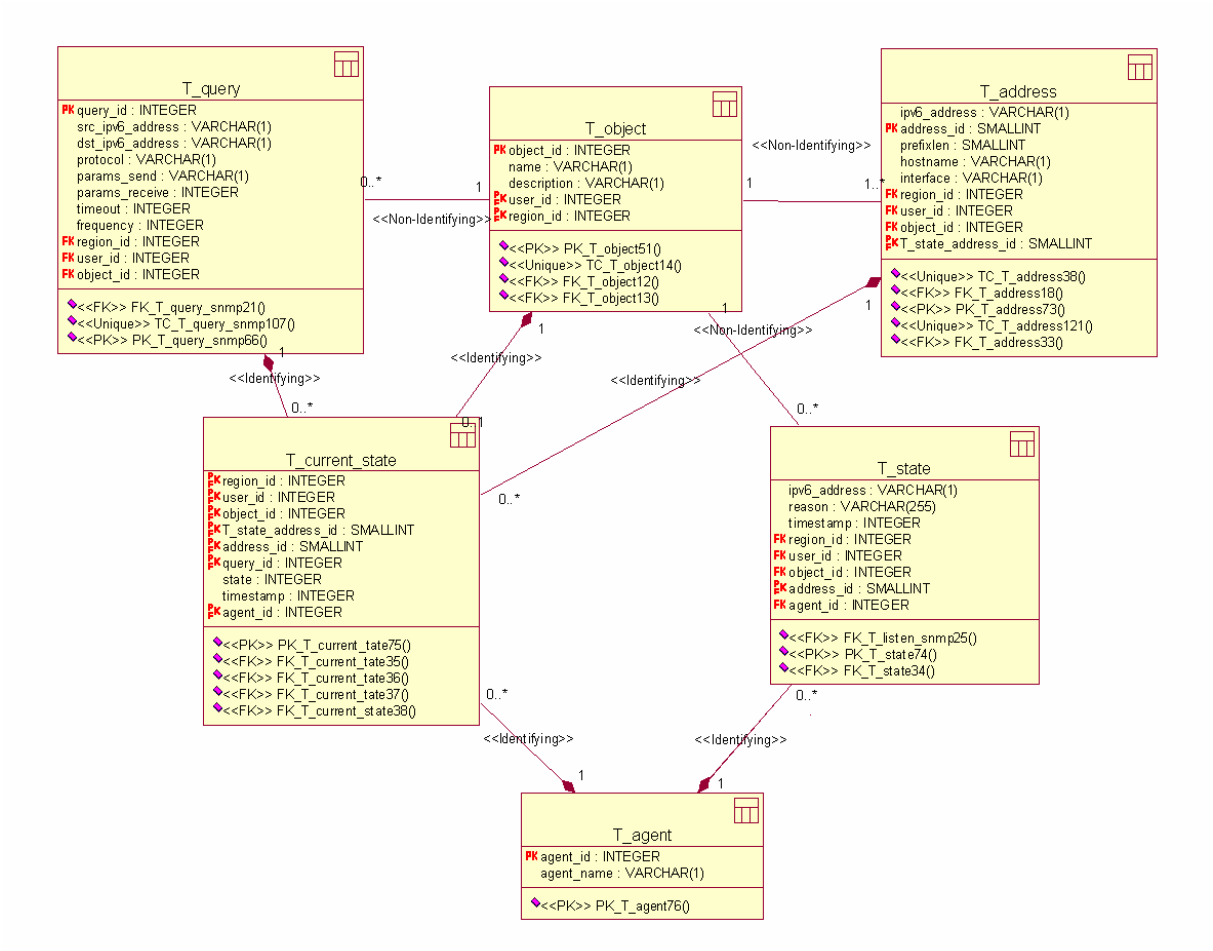
Decentralizavus monitoringą, sistema (konkrečiai centrinis mazgas) turi atskirti gaunamus duomenis pagal tai, iš kokio agento jie atėjo. Tuo tikslu reikia modifikuoti duomenų bazę taip, kad kiekvienas agentas galėtų būti unikaliai identifikuotas ir tuo pačiu būtų patogu išrinkti konkretaus agento pateiktą informaciją.

Modifikaciją atlikti patogiausia pagal jau esančią centralizuotam monitoringui pritaikytą duomenų bazę (žr. 3.4.2 skyrių). Kadangi visi agentai atliks viso tinklo apklausą, keisti reikia tik tas lenteles, kuriose fiksuojamos būsenos arba jų pakitimai.

- **t_address lentelė.** Lentelėje esantis laukelis `state` nebenaudojamas, o adresų būsenos surašomos į naujai tuo tikslu sukurtą lentelę `t_current_state`.

- **t_query lentelė.** Lentelėje esantis laukelis `state` nebenaudojamas, o užklausų būsenos surašomos į naujai tuo tikslu sukurtą lentelę `t_current_state`.
- **t_state lentelė.** Lentelės laukelių sąrašas papildomas nauju lauku `agent_id`, pagal kurį galima identifikuoti, kuris agentas užfiksavo adreso arba užklausos būsenos pasikeitimą.
- **t_current_state lentelė.** Naujai sukurta lentelė, kuri yra visų objektų adresų ir užklausų esamu momentu saugykla. Kadangi atsiranda daugiau negu vienas agentas, vienareikšmiškai spręsti apie būseną negalima, todėl reikalinga ši papildoma lentelė, sauganti būsenas pagal kiekvieno agento pateikiamus duomenis.
- **t_agent lentelė.** Naujai sukurta lentelė, kurioje nurodyti visi tinklą stebintys agentai. Pagal šios lentelės duomenis agentai randa savo identifikacijos numerį, kuriuo vėliau pažymi savo duomenis.

Pasikeitusios duomenų vaizdo schemos lentelės pavaizduotos 14 pav.:



14 pav. Pasikeitusios duomenų vaizdo schemos lentelės

4.2.3. Patobulinimai architektūroje

Architektūros patobulinimai pagrinde apima šiuos pakeitimus:

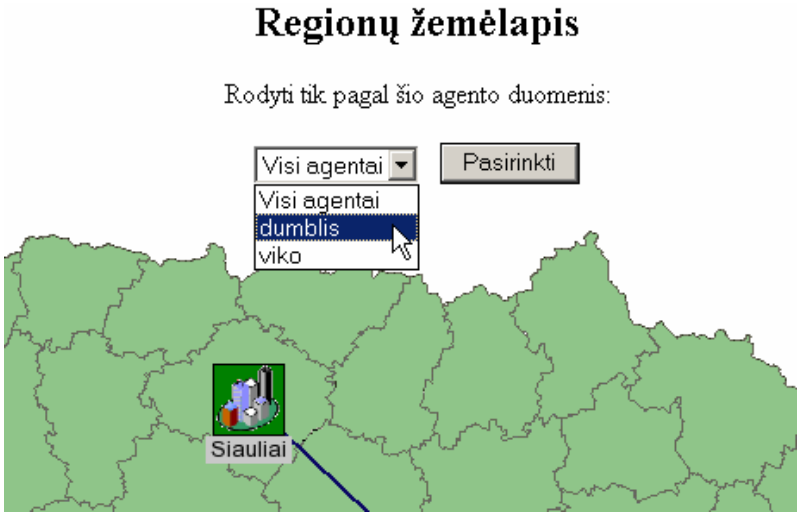
- Tinklo sąsajos pritaikymas daugelio agentų darbui.
- Vartotojo sąsajos pritaikymas daugelio agentų darbui.

Tinklo sąsajos pritaikymas sudarytas iš šių pakeitimų:

- **Agento identifikatorius.** Rezidentinė programa jau nėra vienintelė, stebinti tinklą ir yra traktuojama kaip atskiras agentas. Kiekviename agente esanti rezidentinė programa turi būti sukonfigūruota su individualiu agento vardu (kuris taip pat turi būti įrašytas pagrindinėje duomenų bazėje `t_agent` lentelėje). Pradėdamas darbą, kiekvienas agentas pagal savo vardą atranda savo agento identifikatorių, kurį naudoja įrašinėdamas savo surinktą būsenų informaciją.
- **Esamos būsenos nustatymas.** Kiekvienas agentas turi pasirūpinti savo tinklo vaizdo esamos būsenos kitimo registracija. Tam užtikrinti, agentai seka `t_current_state` lentelės įrašų kaitą ir pagal ją registruoja pasikeitimus `t_state` lentelėje. Vieno agento tinklo būsenų informacija neturi įtakoti kito agento būsenų informacijos.
- **Būsenos kitimo registravimas.** Agentas, registruodamas būsenos pasikeitimą `t_state` lentelėje, turi nurodyti savo agento identifikatorių. Tai reikalinga tam, kad būtų galima atskirti, kuris agentas užregistravo būsenos pasikeitimą (vartotojui turi būti sudaryta galimybė matyti tik vieno konkretaus agento užregistruotus būsenų pakitimus).

Vartotojo sąsajos pritaikymas sudarytas iš šių pakeitimų:

- **Agentų informacijos valdymas.** Administratoriaus teises turinčiam vartotojui turi būti sudaryta galimybė pridėti/redaguoti/trinti agentus duomenų bazėje. Tinklo monitoringo informaciją į pagrindinę duomenų bazę įrašinėti galės tik tie agentai, kurių vardai sukonfigūruoti `t_agent` lentelėje, taigi, administratorius turi galėti valdyti šios lentelės įrašus.
- **Konkretaus agento topologijos vaizdas.** Sistemos vartotojui turi būti suteikta galimybė pasirinkti ne tik apibendrintą objektų būsenų vaizdą pagal visų agentų duomenis, bet ir būsenas pagal pasirinkto vieno agento surinktą informaciją (žr. 15 pav.).



15 pav. Galimybė pasirinkti konkretaus agento duomenis

4.3. Sistemos efektyvumas

Replikuotojo modelio atveju, kaip ir paskirstame modelyje, monitoringo funkciją centrinis mazgas deleguoja L tarpinio lygmens agentams, tačiau šiuo atveju kiekvienas agentas vykdo visų N objektų monitoringą. Apklausimo dažniui esant vienodam (visi agentai naudojasi vienoda objektų informacija), laikas T_{RS}, reikalingas tarpiniam mazgui aptikti kintamojo X reikšmės pakitimą, yra toks pati, kaip (1), kadangi kiekvienas agentas bus kaip centrinis mazgas centralizuoto modelio atveju (žr. 2.1.1 skyrelį).

Tarpiniam mazgui pastebėjus būsenos pasikeitimą, atnaujinama lokali duomenų bazė bei iš karto atnaujinama ir centrinė duomenų bazė. Laikas, kol persiunčiama būsenos pasikeitimo informacija tinklu yra pakankamai mažas, todėl į jį galima nekreipti dėmesio. Tai reiškia, kad bet kurio objekto būsenos pasikeitimas centrinėje duomenų bazėje bus užfiksuotas tada, kai jis bus pastebėtas bet kuriame iš agentų. Laikas nuo būsenos pasikeitimo pačiame objekte iki pasikeitimo užfiksavimo centrinėje duomenų bazėje:

$$T_{RS} = \min (T_{HS}) + t_c + t_r. \quad (8)$$

Esant vienodam apklausimo intervalui Δ, kiekvieno SM apkrovimas tiesiogiai priklausomas nuo tarpinių agentų kiekio. Kiekvienas SM replikuojant duomenų surinkimą turės apdoroti srautą L $(i_q + i_r + 2h)/\Delta$ bps.

Kadangi kiekvienas apklausiantis agentas yra tas pačias funkcijas atliekantis įrenginys, kaip centrinis mazgas centralizuoto monitoringo atveju, jo apdorojamas duomenų srautas vykdant objektų apklausimą bus toks pats, kaip ir (2). Papildomas srautas atsiranda tada, kai užfiksuojamas kurio nors objekto būsenos pasikeitimas. Tokiu atveju informaciją reikia

perduoti įrenginiui, kuriame patalpinta centrinė duomenų bazė. Papildomas srautas, kai pasikeitimą pastebi visi agentai (atsižvelgiant į būsenos pasikeitimo tikimybę λ) yra (bps):

$$C_{RS} = LN\lambda(i_r + h). \quad (9)$$

Šiuo atveju persiunčiami duomenų srautai yra didesni, negu naudojant centralizuotą duomenų surinkimo metodą.

Kiekvienas tarpinis agentas apdorodamas paketus praleidžia $N(t_q + (1 + \lambda\Delta)t_r)$ laiko tarpą kiekvieno apklausimo intervalo Δ metu. Tai reiškia, kad laiko dalis, praleista apdorojant paketų srautą, yra: $N(t_q + (1 + \lambda\Delta)t_r) / \Delta$. Laiko dalis, praleista apdorojant srautą centriniame mazge, yra:

$$U_{HS} = NL\lambda t_q. \quad (10)$$

Nesunku pastebėti, kad replikuotojo modelio atveju tiek duomenų srautai, tiek resursų išnaudojimas agentuose ir centriniame įrenginyje padidėja. Duomenų srauto padidėjimas tiesiogiai priklauso nuo tarpinių agentų skaičiaus. Apkrovimo padidėjimas stebimuose objektuose ir centrinę duomenų bazę aptarnaujančiame įrenginyje taip pat priklauso nuo tarpinių mazgų skaičiaus. Apkrovimas agentuose priklauso nuo stebimų objektų skaičiaus ir jo neįtakoja pačių agentų skaičius.

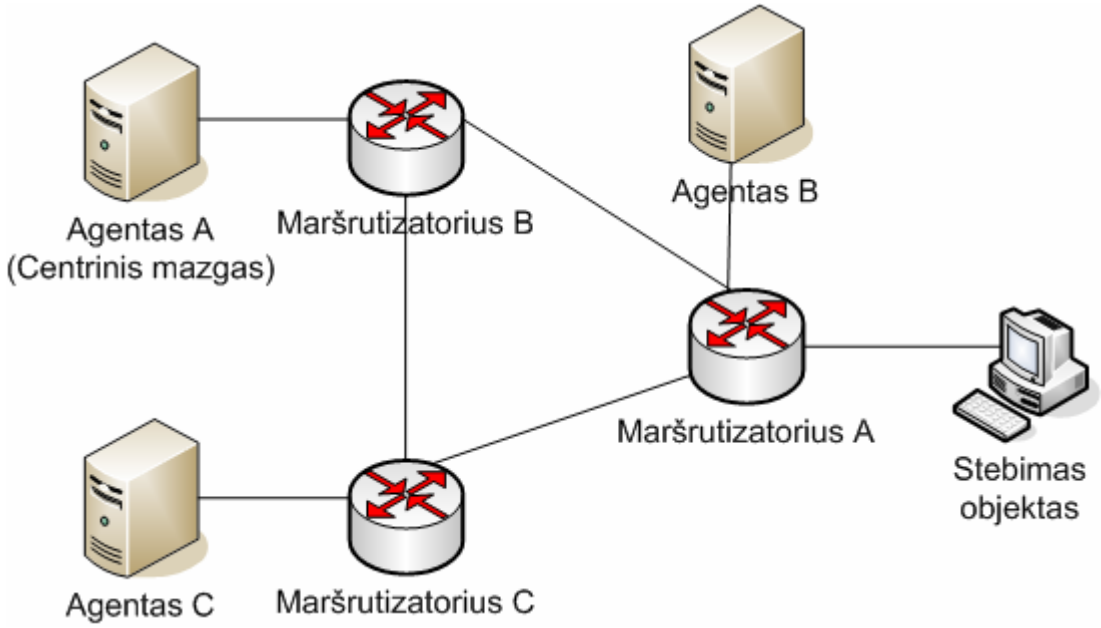
Būsenos pasikeitimo aptikimo greitis taip pat įtakojamas agentų skaičiaus bei jų išdėstymo topologijoje (tai ypač svarbu norint aptikti duomenų kanalo sutrikimus). Priklausomybė tarp laiko tarpo, reikalingo užfiksuoti būsenos pasikeitimą, ir agentų skaičiaus yra atvirkštinė, taigi, kuo didesnis agentų skaičius, tuo mažesnis tikėtinas laiko tarpas iki bus užfiksuotas būsenos pasikeitimas.

5. PASIKEITIMO VĖLINIMO EKSPERIMENTAS

Norint išmatuoti sukurtos sistemos darbo charakteristikas, atliekamas būsenos aptikimo vėlinimo eksperimentas. Tikslas – išmatuoti ir palyginti, kaip keičiasi laiko tarpas, reikalingas užfiksuoti objekto būsenos pasikeitimą centrinėje duomenų bazėje, keičiant replikuotu duomenų surinkimo metodu veikiančių agentų skaičių.

5.1. Eksperimento priemonės

Eksperimentas atliekamas specialiai tam paruoštuose įrenginiuose, kurių išdėstymo schema pateikiama (16 pav.).



16 pav. Eksperimento aparatūrinės įrangos išdėstymo schema

Eksperimente naudojama tokia aparatūrinė įranga:

- **Agentas A.** Intel Pentium III 933 MHz, 384 MB RAM, OS Linux debian.
- **Agentas B.** Intel Pentium IV 1.8 GHz, 256 MB RAM, OS Linux debian.
- **Agentas C.** Intel Pentium Pro 180 MHz, 64 MB RAM, OS Linux debian.
- **Maršrutizatorius A.** Intel Pentium IV 1.7 GHz, 256 MB RAM, OS Linux debian.
- **Maršrutizatorius B.** Intel Pentium IV 1.8 GHz, 256 MB RAM, OS Linux debian.
- **Maršrutizatorius C.** Intel Pentium II 233 MHz, 128 MB RAM, OS Linux debian.
- **Stebimas objektas.** Intel Pentium II 300 MHz, 128 MB RAM, OS FreeBSD 5.4.

5.2. Eksperimento eiga

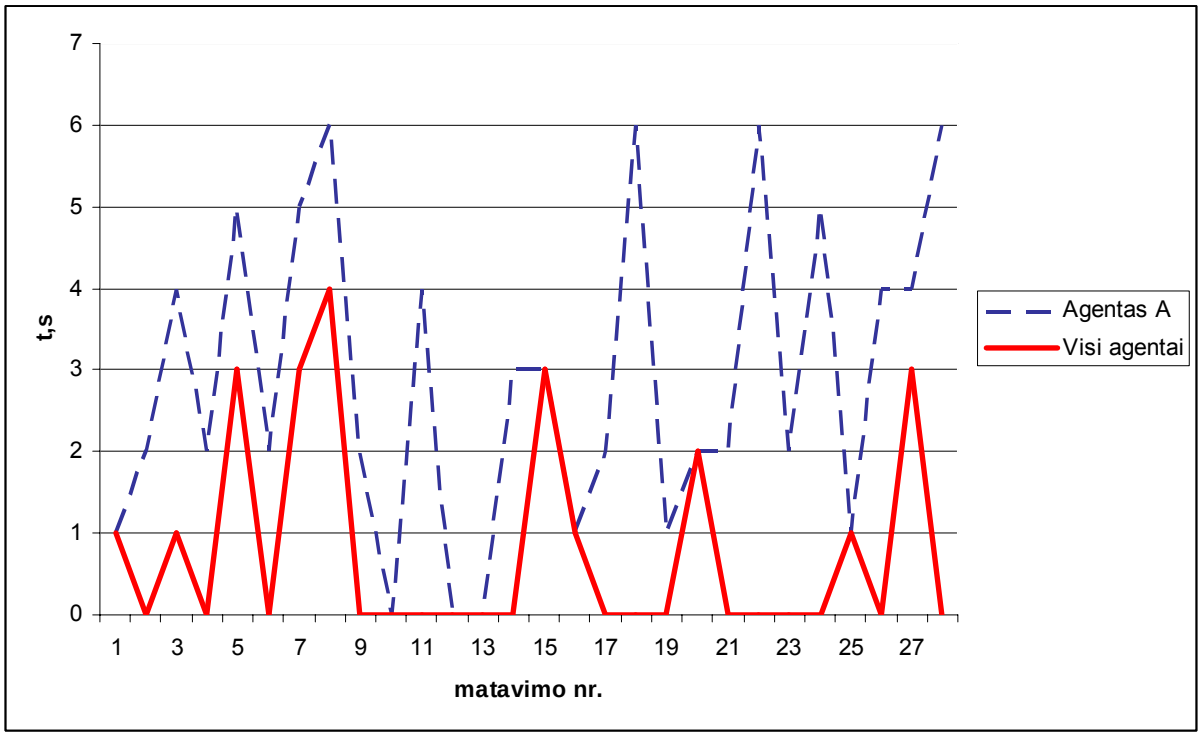
Eksperimento metu tikrinamas pasiekiamumas iki objekto ICMPv6 protokolu. Bandoma stebint būsenos pasikeitimą agento A, agento B ir agento C duomenų bazėje (po vieną agentą) ir centrinėje bazėje (visi trys agentai). Eksperimentas vykdomas 5 minutes. Neveikimo/veikimo laiko tarpai – atsitiktiniai skaičiai ne mažesni nei 5 ir nedidesni nei 15 sekundžių. Objekto apklausimas vykdomas apklausiant kas 5 sekundes (visuose agentuose vienodai).

Pasiekiamumo sutrikimas simuliuojamas, atsitiktiniams laiko intervalams maršrutizatoriuje A visiškai užblokuojant/atblokuojant srautą ICMPv6 protokolu link stebimo objekto (žr. 1 priedą).

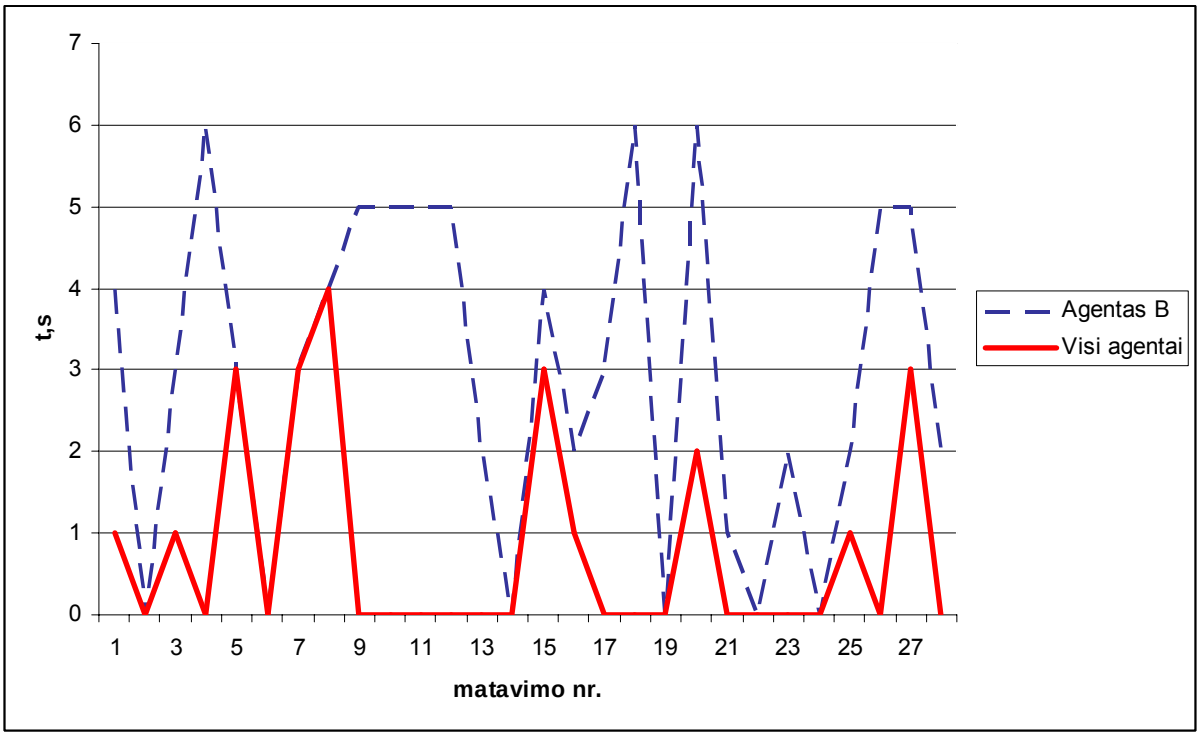
Laukiami rezultatai: visi trys agentai turi fiksuoti objekto būsenos pasikeitimą. Bet kurio vieno agento arba bet kurio dviejų agentų rezultatų derinio vėlinimas turi būti didesnis, negu visų trijų agentų rezultato derinio vėlinimas.

5.3. Eksperimento rezultatai

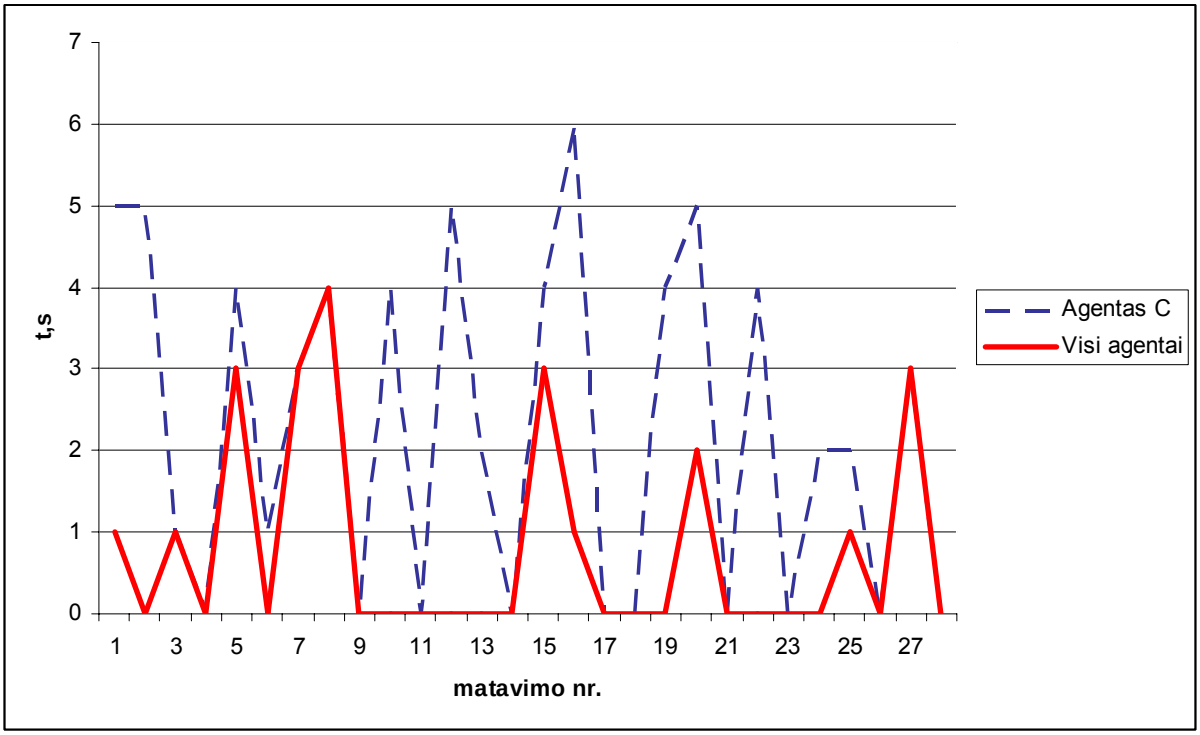
Akivaizdžiai matyti, jog turint visų trijų agentų duomenis, objekto būsenos pasikeitimas fiksuojamas greičiau, negu turint mažesnio kiekio agentų duomenis (žr. 17-22 pav.).



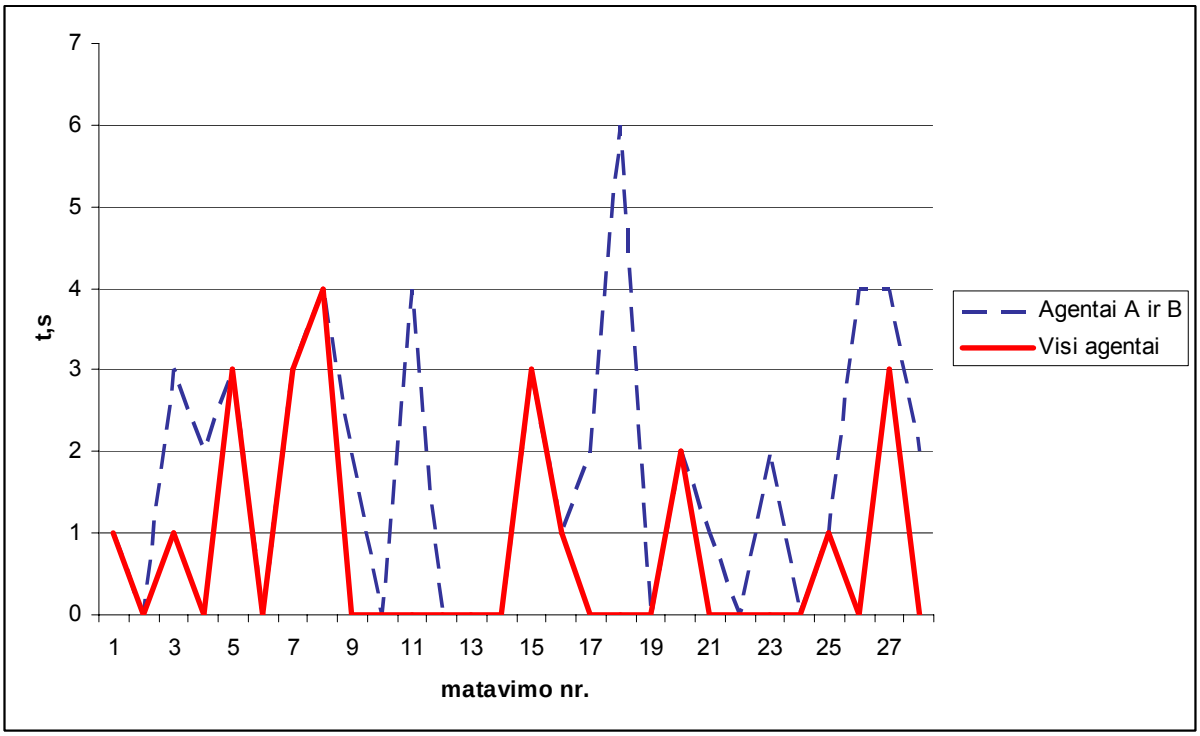
17 pav. Būsenos aptikimo vėlinimas lyginant su agento A duomenimis



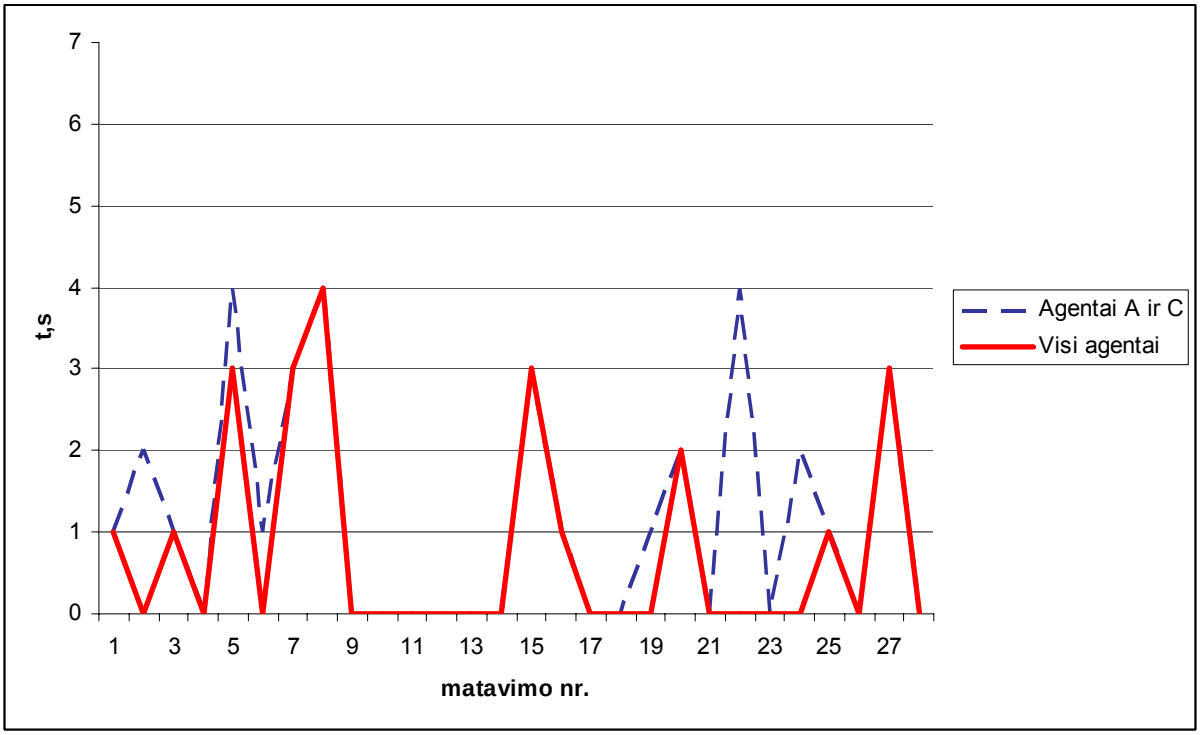
18 pav. Būsenos aptikimo vėlinimas lyginant su agento B duomenimis



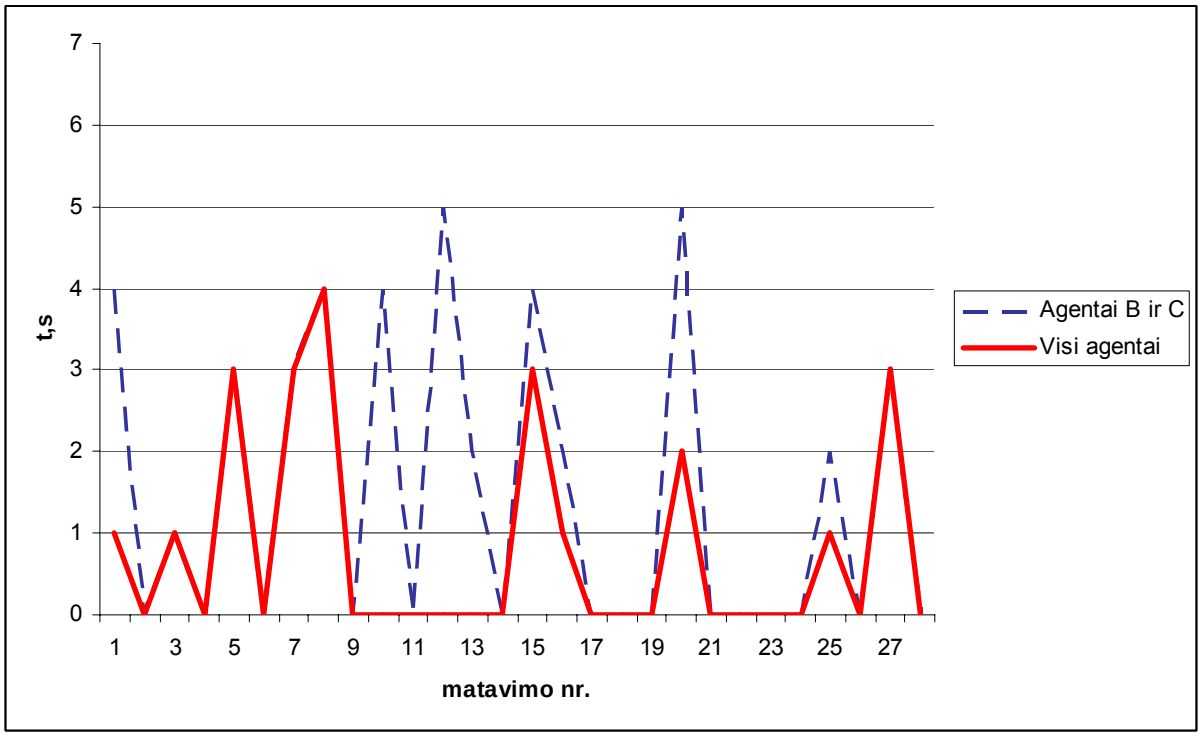
19 pav. Būsenos aptikimo vėlinimas lyginant su agento C duomenimis



20 pav. Būsenos aptikimo vėlinimas lyginant su agentų A ir B duomenimis



21 pav. Būsenos aptikimo vėlinimas lyginant su agentų A ir C duomenimis



22 pav. Būsenos aptikimo vėlinimas lyginant su agentų B ir C duomenimis

Bet kurio agento vėlinimas bent vieno matavimo metu yra didesnis lyginant su bendru skaičiumi, vadinasi bet kuriuo atveju trys agentai būsenos pasikeitimą pastebi greičiau, negu vienas agentas. Turint dviejų agentų duomenis, būsenos pasikeitimas bent vieno matavimo metu pastebimas vėliau, negu turint visų trijų agentų duomenis, taigi dviejų agentų pastebėtas būsenos pasikeitimas taip pat vėluos labiau, nei trijų agentų pastebėjimas.

6. IŠVADOS

Darbe išnagrinėtos esamos tinklo monitorinio sistemos, galinčios stebėti IPv6 tinko veikimą. Nustatyta, kad jos nepakankamai išnaudoja IPv6 protokolo savybes, palengvinančias monitoringo užduotis, ir nepakankamai atsižvelgia į IPv6 tinklų topologijos ypatumus.

Pasiūlytas specializuotas IPv6 tinklo monitoringo sprendimas, kuris skiriasi nuo žinomų tuo, kad panaudota paskirstyta apklausimo veiksmus kartojanti architektūra ir monitoringas vykdomas atsižvelgiant į IPv6 tinklų ypatumus.

Pasiūlytas naujas metodas, kuris naudoja daugiau negu vieno objektus lygiagrečiai apklausiančio agento informaciją ir toks duomenų surinkimo metodas pavadintas replikuotu. Teoriškai nustatyta, kad šis metodas turėtų greičiau pastebėti tinklo būsenos pasikeitimą, negu naudojami paskirstyto monitoringo metodai.

Darbo metu buvo sukurta IPv6 tinklo monitoringo programų sistema, naudojanti pasiūlytą replikuotą metodą.

Eksperimentiškai ištyrus sistemą, pastebėta, jog didinant apklausiančių agentų skaičių ir replikuojant monitoringo užduotis, tinklo būsenos pasikeitimas pastebimas greičiau. Teorinė prielaida patvirtinta atlikus būsenos pasikeitimo vėlinimo palyginimo eksperimentą.

7. LITERATŪRA

[1] ASGARI A., TRIMINTZIOS P., IRONS M., PAVLOU G., DEN BERGHE S. V. ir EGAN R. *A Scalable Real-Time Monitoring System for Supporting Traffic Engineering.* IEEE IP eksploatavimo ir valdymo seminaro medžiaga, 2002.

[2] THOTTAN M., LI L., YAO B., MIRROKNI VAHAB S. ir Paul S. *Distributed Network Monitoring For Evolving IP Networks.* 24 tarptautinės paskirstytų skaičiavimo sistemų konferencijos (ICDCS) pranešimų medžiaga. Tokijas, Japonija, 2004 kovas.

[3] STALLINGS W. *SNMP, SNMPv2, SNMPv3, and RMON 1 and 2*. Addison Wesley Longman, Inc, 3 leidimas, 1999.

[4] CARZANINGA A., PICCO G., ir VIGNA G. *Designing distributed applications with mobile code paradigms.* Tarptautinė programų inžinerijos konferencija, 1997.

[5] SUBRAMANYAN R, MIGUEL-ALONSO J., ir FORTES J. A. B. *A scalable SNMP-Based distributed monitoring system for heterogenous network computing*. SC2000 medžiaga. Dalasas, Teksaso valstija, 2000.

[6] THOTTAN M. ir JI C. *Using network fault prediction to enable IP traffic management.* Tinklų ir sistemų valdymo žurnalas, 9(3):327–346, 2001.

[7] BREITGAND D., RAZ D. ir SHAVITT Y. *SNMP GetPrev: An efficient way to access data in large MIB tables.* IEEE parinktų komunikacijos sričių žurnalas, 20(4):656–667, 2002.

[8] LI L., THOTTAN M., YAO B. ir PAUL S. *Distributed Network Monitoring with Bounded Link Utilization in IP Networks.* IEEE INFOCOM medžiaga, 2003.

[9] CHEN T. M. ir LIU S. *A Model and Evaluation of Distributed Network Management Approaches.* IEEE parinktų komunikacijos sričių žurnalas, 20(4), 2002 gegužė.

[10] GOLDSZMIDT G. ir YEMINI Y. *Delegated agents for network management.* IEEE komunikacijų žurnalas Nr. 36, 66–70 psl., 1998 kovas.

[11] MA A. *Macroscopic IPv6 topology measurements* [interaktyvus]. 2003 sausis [žiūrėta 2004-11-09]. Prieiga per internetą: <http://www.caida.org/analysis/topology/macroscopic/IPv6/>.

[12] MA A. *Skitter* [interaktyvus]. 2004 birželis [žiūrėta 2004-11-10]. Prieiga per internetą: <http://www.caida.org/tools/measurement/skitter/>.

[13] MEYER D. *University of Oregon Route Views Project* [interaktyvus]. 2002 gegužė [žiūrėta 2004-11-10]. Prieiga per internetą: <http://www.antc.uoregon.edu/route-views/>.

[14] LUCKIE M. *IPv6 Skitter* [interaktyvus]. 2003 balandis [žiūrėta 2004-11-10]. Prieiga per internetą: <http://www.caida.org/~mjl/>.

[15] *Network Node Manager Smart Plug-in for advanced routing* [interaktyvus]. 2004 [žiūrėta 2004-11-10]. Prieiga per internetą: <http://www.openview.hp.com/products/spi/nnm_spi_ar/>.

[16] *Using Network Node Manager Extended Topology* [interaktyvus]. 2003 sausis [žiūrėta 2004-11-10]. Prieiga per internetą: <http://ovweb.external.hp.com/ovnsmdps/pdf/j5303-90001.pdf>.

[17] GALSTAD E. *Nagios v1.0 documentation* [interaktyvus]. 2004 spalis [žiūrėta 2004-11-10]. Prieiga per internetą: <http://nagios.sourceforge.net/download/contrib/documentation/english/Nagios_1_0_Docs.pdf>.

[18] *Nagios plugin development news* [interaktyvus]. 2003 vasaris [žiūrėta 2004-11-10]. Prieiga per internetą: <http://sourceforge.net/news/?group_id=29880>.

[19] DEBISSCHOP K., GALSTAD E., GAYOSSO H., GHOSH S., HOPCROFT S., VOON

T., BOUSE J. T. *Nagios plug-in development guidelines* [interaktyvus]. 2004 [žiūrėta 2004-11-10]. Prieiga per internetą: <http://nagiosplug.sourceforge.net/developer-guidelines.html>.
[20] MUYAL S. *IPv6 network management* [interaktyvus]. 2004 gegužė [žiūrėta 2004-11-10]. Prieiga per internetą: <http://www.ipv6spring2004.be/Slides/Simon%20MUYAL%20-%20IPv6%20networks%20management.pdf>.
[21] MAUCH J. *Sysmon homepage* [interaktyvus]. 2004 spalis [žiūrėta 2004-11-10]. Prieiga per internetą: <http://www.sysmon.org/>.
[22] WEISBERG J. *Argus IPv6 support* [interaktyvus]. 2004 rugsėjis [žiūrėta 2004-11-10]. Prieiga per internetą: <http://argus.tcp4me.com/ipv6.html>.
[23] *ASpath-tree* [interaktyvus]. 2003 balandis [žiūrėta 2004-11-10]. Prieiga per internetą: <http://carmen.ipv6.tilab.com/ipv6/tools/ASpath-tree/index2.html>.

8. TERMINŲ IR SANTRUMPŲ ŽODYNAS

ARP – *address resolution protocol* (adreso suradimo protokolas).

BGP – *border gateway protocol* (kraštinių autonominės sistemos maršrutizatorių protokolas).

bps – *bits per second* (bitai per sekundę).

DBVS – duomenų bazės valdymo sistema.

DNS – *domain name system* (srities vardų sistema).

GD – *gif draw* (grafikos generavimo biblioteka).

GPL – GNU *public licence* (GNU vieša licencija).

GUI – *graphic user interface* (grafinė vartotojo sąsaja).

HTTP – *hypertext transfer protocol* (hiperteksto perdavimo protokolas).

HTTPS – *secure hypertext transfer protocol* (saugus hiperteksto perdavimo protokolas).

ICMP – *internet control messages protocol* (interneto valdymo pranešimų protokolas).

IPv4 – *internet protocol version 4* (interneto protokolo 4 versija).

IPv6 – *internet protocol version 6* (interneto protokolo 6 versija).

NAP – *network access point* (tinklo prieigos taškas).

ND – *neighbor discovery* (kaimynų aptikimas).

NNM – *network node manager* (tinklo mazgų valdiklis).

PC – *personal computer* (asmeninis kompiuteris).

PHP – *hypertext preprocessor* (išankstinė hiperteksto doroklė).

QoS – *quality of service* (paslaugos kokybės užtikrinimas).

RA – *router advertisement* (maršrutizatoriaus skelbimas).

RIPE – *réseaux internet protocol européens* (Europos interneto protokolo adresų registras).

SLA – *service layer aggreement* (paslaugos kokybės parametrų užtikrinimo sutartis).

SM – stebimas mazgas.

SNMP – *simple network management protocol* (paprastas tinklo valdymo protokolas).

TCP – *transmission control protocol* (duomenų perdavimo valdymo protokolas).

TTM – *test traffic measurements* (patikrinamojo srauto matavimai).

UDP – *user datagram protocol* (vartotojo duomenų fragmentų protokolas).

WWW – *world wide web* (pasaulinis žiniatinklis).

1 PRIEDAS. Objekto sutrikimo simuliavimo pusprogramės išeities tekstas.

```
#!/usr/bin/perl

$deny = 0;

while (1)
  {
    $rand = rand (10);
    $rand += 5;
    $time = time ();

    if (! $deny)
      {
        print ("Denied for $rand seconds\ntime: $time\n\n");
        system ("ip6tables -I FORWARD -d 2001:778:11:9::1 -p
icmpv6 -j DROP");
        $deny = 1;
      }
    else
      {
        print ("Allowed for $rand seconds\ntime: $time\n\n");
        system ("ip6tables -D FORWARD -d 2001:778:11:9::1 -p
icmpv6 -j DROP");
        $deny = 0;
      }

    sleep ($rand);
  }
```