

VILNIAUS UNIVERSITETAS

Liudvikas Kaklauskas

**Fraktalinių procesų kompiuterių tinkluose stebėsenos ir valdymo  
metodų tyrimas**

**Daktaro disertacija**

Fiziniai mokslai, informatika (09 P)

Vilnius, 2012

Disertacija rengta 2008–2012 metais Vilniaus universiteto Matematikos ir informatikos institute

Mokslinis vadovas:

Prof. habil. dr. Leonidas Sakalauskas (Vilniaus universitetas, fiziniai mokslai, informatika – 09 P)

## Padėka

Disertacijos autorius nuoširdžiai dėkoja moksliniam vadovui prof. habil. dr. Leonidui Sakalauskui už suteiktas labai atsakingas bei vertingas mokslines konsultacijas per doktorantūros studijas, už patarimus ir pasiūlymus bei nuolatinį skatinimą tobulėti, Matematikos ir informatikos instituto direktoriui prof. habil. dr. Gintautui Dzemydai už visapusišką paramą studijuojant doktorantūroje, MII Duomenų analizės skyriaus operacijų tyrimo sektoriaus darbuotojams bei doktorantams už bendradarbiavimą, pagalbą ir supratimą, disertacijos recenzentams prof. habil. dr. Kaziui Kazlauskui bei doc. dr. Igoriui Belovui, atidžiai perskaičiusiems disertaciją ir pateikusiems vertingų patarimų bei kritinių pastabų.

Dėkoju Šiaulių universiteto Matematikos ir informatikos fakulteto dekanui doc. dr. Dariui Šiaučiūnui bei visiems Informatikos katedros darbuotojams, draugams ir artimiesiems už jų paramą, kantrybę, supratingumą.

Dėkoju už finansinę pagalbą, skirtą lietuvių kalbos redagavimui ir vertimui į anglų kalbą iš 7 bendrosios mokslinių tyrimų, technologijų plėtros ir demonstracinės veiklos programos INTEGER projekto (Nr. 266638) „Institucijos kaita, skatinanti moterų ir vyrų lygybę moksle“ (angl. k. Institutional Transformation for Effecting Gender Equality in Research).

Už nuolatinį palaikymą, kantrybę ir supratingumą dėkoju savo šeimai ir žmonai Danutei.

Liudvikas Kaklauskas

## Turinys

1 skyrius. Įvadas .....	18
1.1. Tyrimų sritis .....	18
1.2. Problemos aktualumas.....	22
1.3. Tyrimų objektas.....	24
1.4. Tyrimų tikslas ir uždaviniai.....	24
1.5. Tyrimo metodai .....	24
1.6. Mokslinis naujumas.....	25
1.7. Praktinė darbo reikšmė.....	26
1.8. Ginamieji teiginiai.....	26
1.9. Rezultatų aprobavimas .....	27
1.10. Darbo rezultatų publikavimas .....	28
1.11. Disertacijos struktūra.....	29
2 skyrius. Kompiuterių tinklų modeliavimas .....	31
2.1. Tinklų technologijos.....	31
2.2. Tinklo informacinio srauto modeliavimas .....	39
2.2.1. Tinklo komponentų modeliavimas .....	39
2.2.2. Paketinio kompiuterių tinklo srauto modeliai.....	43
2.2.3. Paketinio tinklo srauto aptarnavimo modeliai .....	48
2.3. Skyriaus išvados.....	53
3 skyrius. Fraktalinių srautų kompiuterių tinkluose matavimo technologijos ir metodai .....	55
3.1. Technologiniai tinklo srauto matavimo sprendimai.....	56
3.2. Tinklo srauto parametrų vertinimas taikant chaoso teoriją.....	59
3.3. Savastingumas ir jo analizės metodai kompiuterių tinkluose .....	61
3.3.1. Hurst'o koeficiento vertinimas .....	63
3.3.2. Stabiliųjų modelių parametrų vertinimas.....	66
3.3.3. Prieaugio santykio statistika (IR statistika) .....	68
3.4. Skyriaus išvados.....	69
4 skyrius. Fraktališkumo analizė kompiuterių tinkluose .....	70

4.1. Laiko eilučių formavimas.....	71
4.1.1. Laiko eilučių agregavimas.....	72
4.1.2. Agreguotų laiko eilučių formavimas realiu laiku.....	75
4.2. Tinklo srauto analizės priemonės.....	76
4.3. Analizės metodų testavimas.....	78
4.4. Kompiuterių tinklo apkrovos laiko eilučių analizės rezultatai.....	83
4.5. Skyriaus išvados.....	89
5 skyrius. Fraktalinių procesų imitavimas.....	91
5.1. Fraktalinių procesų imitavimo programinė įranga.....	92
5.2. Vieno kanalo tinklo mazgo aptarnavimo modelis.....	94
5.3. Vieno kanalo tinklo mazgo modelis.....	96
5.4. Vieno kanalo tinklo mazgo tyrimo rezultatai.....	100
5.5. Paketinio kompiuterių tinklo mazgo tyrimai.....	104
5.6. Apibendrintas stochastiškai apribotas tinklo modelis.....	105
5.7. Tinklo modelio aprašymas.....	109
5.8. Tyrimo rezultatai GI/G/m//N tinklo mazge.....	113
5.9. Skyriaus išvados.....	122
6 skyrius. Savastingumo matavimo realiu laiku metodas.....	125
6.1. Kompiuterių tinklo srauto valdymo modelis.....	126
6.2. Savastingumo matavimo realiu laiku metodų apžvalga.....	128
6.3. Tinklo srauto analizės realiu laiku metodas.....	132
6.4. Matematinis siūlomo metodo aprašymas.....	134
6.5. Pateikto išradimo apibrėžtis.....	137
6.6. Tinklo srauto analizės realiu laiku metodo testavimas.....	138
6.7. Skyriaus išvados.....	141
7 Išvados.....	142
Literatūra.....	144

## Žymenys ir santrumpos

$\oplus$	<i>max-plus</i> algebros dekonvoliucijos (angl. <i>deconvolution</i> ) operatorius, aprašantis dviejų funkcijų $f$ ir $g$ dekonvoliuciją, kuri gaunama pagal formulę $f \oplus g(t) \equiv \inf_{s \geq 0} \{f(t+s) - g(s)\}$ .
$t_i$	$i$ -ojo paketo atėjimo laikas.
$\otimes$	<i>max-plus</i> algebros konvoliucijos (angl. <i>convolution</i> ) operatorius, aprašantis dviejų funkcijų $f$ ir $g$ konvoliuciją, kuri gaunama pagal formulę $f \otimes g(t) \equiv \sup_{0 \leq s \leq t} \{f(s) + g(t-s)\}$ .
$x_i$	paketo aptarnavimo trukmė.
$\tau_i$	trukmė tarp gretimų paketų pasirodymo mazge.
$S_\alpha(\beta, \sigma, \mu)$	$\alpha$ -stabilusis dėsnis.
<i>ad hoc</i>	tai tinklas, kuris apjungia kompiuterius tiesiogiai, nenaudojant specializuotų tinklo mazgų ir yra iš karto yra parengtas darbui, dar kartais vadinamas „lygus su lygiu“ tinklu, t. y. visi tinklo komponentai turi vienodas teises.
<i>AR</i>	tai autoregresinis tinklo modelis (angl. <i>Autoregressive</i> ).
<i>ARIMA</i>	tai autoregresinis integruoto slenkančių vidurių tinklo metodas (angl. <i>AutoRegressive Integrated Moving Average</i> ).
<i>ARMA</i>	tai autoregresinis slenkančio vidurkio tinklo modelis (angl. <i>Autoregressive Moving Average</i> ).
<i>CSMA/CD</i>	tai <i>Ethernet</i> standarto naudojamas tinklo resursų prieigos metodas: daugkartinės prieigos su nešančiojo signalo kontrole ir konfliktų aptikimu (angl. <i>Carrier Sense Multiple Access/Collision Detection</i> ).
<i>D</i>	tai laiko eilutės pokyčių įvertinimo dinamikos fraktalinis matas.
<i>DL</i>	dialogo lygmuo.
<i>end-to-end</i>	tai kompiuterių tinklų perdavimo principas užtikrinantis duomenų paketų perdavimą iš gavėjo siuntėjui per tarpinius

	tinklo mazgus.
<i>FIFO</i>	paketų eilės aptarnavimo metodas, veikiantis pagal principą pirmas įėjai, pirmas aptarnautas (angl. <i>First Input First Output</i> ).
<i>GI/G/1//N</i>	tai tinklo srauto modelis su rekurentiniu įėjimo srautu, bendro pobūdžio aptarnavimo srautu, su vienu aptarnavimo kanalu, kai į sistemą ateina N paketų/paraiškų.
<i>GI/G/m//N</i>	tai tinklo srauto modelis su rekurentiniu įėjimo srautu, bendro pobūdžio aptarnavimo srautu, su m aptarnavimo kanalu, kai į sistemą ateina N paketų/paraiškų.
<i>gSBB</i>	generalizuotas stochastiškai apribotas srautas (angl. <i>Generalized Stochastically Bounded Burstiness</i> ).
<i>H-sssi</i>	tai savastingasis procesas su stacionariais pokyčiais.
<i>Hurst</i>	tai laiko eilučių ilgalaikės atminties matas, dar vadinamas <i>Hurst'o koeficientu</i> .
<i>IEEE-802.3</i>	tai standartas dar vadinamas <i>Ethernet</i> standartu ir plačiai taikomas vietiniuose kompiuterių tinkluose, kai naudojamas <i>CSMA/CD</i> tinklo resursų prieigos metodas.
<i>IPP</i>	tinklo modelis, kuris duomenų kanalą aprašo kaip sistemą su dviem skirtingomis būsenomis: 1 – kai jis apkrautas ir veikia, 2 – kai tinklo kanale nėra paraiškų ir jis laisvas ir neveiksnius (angl. <i>Interrupted Poisson Process</i> ).
<i>IR</i>	tai statistika pagrįsta proceso laiko eilutės dalinių sumų skaičiavimu, taikant antros eilės intervalus arba skirtumus (angl. <i>Increment Ratio</i> ).
<i>LIFO</i>	paketų eilės aptarnavimo metodas, veikiantis pagal principą paskutinis įėjai, pirmas aptarnautas (angl. <i>Last Input First Output</i> ).
<i>LRD</i>	tolima priklausomybė (angl. <i>Long-Range Dependence</i> ) arba ilgalaikė proceso atimtis matematiškai apibūdinama kaip

	lėtas laipsniškas proceso koreliacinės funkcijos bei kitų priklausomybę atspindinčių skaitinių charakteristikų gesimas, kai atstumas tarp laiko momentų neapbrėžtai didėja, šia savybe pasižymi packetinio kompiuterių tinklo savastingasis srautas.
<i>M/M/m</i>	tai tinklo srauto modelis, kai įeina paprastas srautas per vieną aptarnaujančią liniją, turinčią neriboto ilgio eilę ir m įrenginių paketų aptarnavimui.
<i>MAIG</i>	paketų traukinio arba didžiausio leistino atotrūkio tarp paketų tinklo modelis (angl. <i>Maximum Allowed Inter-Car Gap</i> ).
<i>Max-Plus</i>	tai algebra, kuri pakeičia standartinės algebros operatorius taip, kad maksimizuočiau dviejų funkcijų konvoliuciją.
<i>Min-Plus</i>	tai algebra, kuri pakeičia standartinės algebros operatorius taip, kad minimizuočiau dviejų funkcijų konvoliuciją.
<i>MMPP</i>	<i>Markovo</i> moduluoto <i>Puasono</i> proceso tinklo modelis (angl. <i>Markov Modulated Poisson Processes</i> ).
<i>MMRP</i>	<i>Markovo</i> moduluoto dažnio proceso tinklo modelis (angl. <i>Markov Modulated Rate Processes</i> ).
<i>MulNodSimSys</i>	tinklo mazgo srauto modeliavimo daugiakanalė sistema (angl. <i>Multichannel Node Simulation System</i> ).
<i>Network Calculus</i>	kompiuterių tinklų modeliavimo/skaičiavimo teorija.
<i>ON-OFF</i>	tai tinklo modelis, kai tinklo srautas gali turėti dvi stacionarias būsenas <i>ON</i> , kai duomenų paketai perduodami ir <i>OFF</i> – kai paketų nėra.
<i>OSI</i>	informacijos perdavimo procesų funkciniam lygmenyje aprašymo modelis (angl. <i>Open System Interconnection</i> )
<i>PaL</i>	paketų lygmuo.
<i>PIL</i>	pliūpsnio lygmuo.



<i>PP</i>	į tinklo mazgą įeinančio ir aptarnaujamo srauto tipai yra <i>Puasono</i> .
<i>PpSiL</i>	paslaugų programų ar servisų išskvietimo lygmuo.
<i>PrL</i>	prieigos lygmuo.
<i>PS</i>	į tinklo mazgą įeinančio srauto tipas <i>Puasono</i> , aptarnaujamo – savastingasis.
<i>QoS</i>	kompiuterių tinklo srauto paslaugos kokybės rodiklis.
<i>SL</i>	sujungimo lygmuo.
<i>SP</i>	į tinklo mazgą įeinančio srauto tipas savastingasis, aptarnaujamo – <i>Puasono</i> .
<i>SRD</i>	paketinio kompiuterių tinklo srauto savastingumo savybė, kai srautas pasižymi trumpalaikėmis atmintimis (angl. <i>Short-range dependence</i> ).
<i>SRLAP</i>	tinklo srauto savastingumo realiu laiku analizės paketas.
<i>SS</i>	į tinklo mazgą įeinančio srauto tipas savastingasis, aptarnaujamo – savastingasis.
<i>SSE</i>	tai programinių modulių biblioteka, naudojama savastingumo analizei (angl. <i>Self-Similarity Estimator</i> ).
<i>SSE-ON</i>	tai programinių modulių biblioteka, naudojama savastingumo analizei realiu laiku (angl. <i>Self-Similarity Estimator On-Line</i> ).
<i>TCP</i>	tai <i>TCP/IP</i> naudojamas duomenų perdavimo kompiuterių tinklo protokolas, turintis klaidų atitaisymo mechanizmą (angl. <i>Transmission Control Protocol</i> ).
<i>TCP/IP</i>	tai standartinis duomenų perdavimo protokolų rinkinys, kurio pagrindu veikia internetas ir daugelis vietinių kompiuterių tinklų.
<i>TTL</i>	paketo, perduodamo kompiuterių tinklo, gyvavimo laikas (angl. <i>Time to Live</i> ).
<i>UDP</i>	tai <i>TCP/IP</i> naudojamas duomenų perdavimo kompiuterių

tinklu protokolas, neturintis klaidų atitaisymo mechanizmo  
(angl. *User Datagram Protocol*).

- $\alpha$   $\alpha$ -stabilaus dėsnio stabilumo parametras.
- $\beta$   $\alpha$ -stabilaus dėsnio asimetrijos parametras.
- $\mu$   $\alpha$ -stabilaus dėsnio poslinkio parametras.
- $\sigma$   $\alpha$ -stabilaus dėsnio mastelio parametras.

## Savokų žodynelis

Ad hoc	Tai lotyniška frazė, kuri reiškia „šiuo tikslu“ ir naudojama apibrėžti sprendimams, kurie reikalingi čia ir tuoj pat. Pagal TechTerms.com tinklalapį < <a href="http://www.techterms.com/definition/adhocnetwork">http://www.techterms.com/definition/adhocnetwork</a> >.
Atraktorius	Tai geometrinė struktūra, nusakanti sistemos elgseną fazinėje erdvėje, kai laikas artėja į begalybę (angl. <i>Attract – pritraukti</i> ).
Buferis	Laikina kompiuterio atminties sritis arba tinklo įtaisas, reikalingas duomenų mainų spartai, duomenų blokų matmenims ir kt. skirtybėms suderinti (Enciklopedinis kompiuterijos žodynas 2012). Aptarnavimo teorijoje buferis vadinamas eile.
Duomenų paketas	Tai duomenų porcija, siunčiama tinklu kaip nedalomas vienetas. Paketą sudaro duomenų paketo antraštė ir patys duomenys (Enciklopedinis kompiuterijos žodynas 2012).
Ethernet	Lokaliųjų tinklų technologija, aprašoma pagal standartą IEEE 802.3, turinti kai kurias globaliųjų tinklų aplikacijas. Pagal šią technologiją maksimalus tinklu perduodamo paketo dydis yra 1518B. Standartas sukurtas 1972 metais, o 1982 metais pasirodė jo patobulinta Ethernet II versija. Šiuose tinkluose informacija siunčiama naudojant CSMA/CD metodą (angl. <i>Carriers Sense Multiple Access / Collision Detection</i> ). Jo esmė ta, kad duomenys siunčiami porcijomis po kiekvienos nustatant ar duomenų perdavimo linija neužimta. Jei siuntimo įrenginys nustato, kad magistralėje yra duomenų paketas tai siuntimas atidedamas. Jei į magistralę vienu metu patenka du ar daugiau paketų, tai įvyksta jų susidūrimas – kolizija. Po susidūrimo daroma

	pauzė ir tinklo darbas atnaujinamas.
Fazinė erdvė	Tai abstrakti erdvė, kurios koordinatės nusako analizuojamos sistemos laisvės laipsnius.
FIFO Taildrop	Tai eilės (buferio) aptarnavimo disciplina veikianti pagal principą – eilei užsipildžius naikinami vėliausiai atėję paketai ir jų vietoje talpinami naujai atėję.
Fraktalas	Tai matematinio objekto geometrinė išraiška, kurios fraktalinis matas išreiškiamas trupmena ir yra matematinis fraktalo netikslumo įvertis (McBride 2005).
GRID	Tai sujungti skaičiavimo išteklių kai įvairių tipų procesorių, duomenų saugyklų, programinės ar šalutinės įrangos visuma, panaudojama didelės apimties skaičiavimams atlikti. Pagal Mokslo Lietuvos straipsnį < <a href="http://mokslasplius.lt/mokslo-lietuva/2002-2004/200502/pirmas_puslapis.htm">http://mokslasplius.lt/mokslo-lietuva/2002-2004/200502/pirmas_puslapis.htm</a> >
Hausfordo matas	Tai atraktoriaus fraktalinio mato matematinis įvertis. Žymimas $D$ .
Kartotuvai	Tai tinklo įrenginys gautą paketą perduodantis į gavėjo prievadą.
Kompiuterių tinklas	Tai kompiuterių ir kitų įrenginių grupė, sujungtų ryšio linijomis taip, kad būtų galimas keitimasis duomenimis tarp jų (Enciklopedinis kompiuterijos žodynas 2012).
Komutatorius	Tinklo mazgo įrenginys į vieną tašką jungiantis daugelį ryšio linijų, ateinančių iš įvairių tinklo mazgų arba tinklo klientų, ir paskirstantis duomenų paketus tinkle atsižvelgiant į siuntėjo bei gavėjo adresus (Enciklopedinis kompiuterijos žodynas 2012) (angl. <i>Switch</i> ).
LIFO Frontdrop	Tai eilės (buferio) aptarnavimo disciplina veikianti pagal principą – eilei užsipildžius, naikinami ilgiausiai užsibuvę paketai ir jų vietoje talpinami naujai atėję.

Maršrutizatorius	Tai įrenginys apdorojantis ryšį tarp dviejų arba daugiau paketiniu būdu perjungiamų tinklų, skirtas dideliems tinklams formuoti, informacijos perdavimo srautų maršrutams aprašyti, įvairių rūšių ir paskirčių tinklams apjungti į bendrą tinklą (angl. <i>Router</i> ).
Paketinio kompiuterių tinklo pliūpsnis	Tai kompiuterių tinklo srauto neprognozuojamas padidėjimas.
Paketinis duomenų persiuntimo kompiuterių tinklu būdas	Duomenų siuntimo būdas, kai išsiuntimo vietoje duomenys (duomenų failas) skaidomi į duomenų paketus ir persiunčiami tinklu. Tų pačių duomenų paketai paskirties vietą gali pasiekti įvairiais maršrutais, priklausomai nuo tuo metu esančių duomenų judėjimo sąlygų tinkle. Paskirties vietoje iš gautų paketų surenkamas pradinis duomenų rinkinys (failas). Duomenų skaidymas į paketus išsiuntimo vietoje ir surinkimas paskirties vietoje eina per visus protokolo lygmenis; skaidymas – žemėjančiai, o surinkimas – atvirkščia linkme (Enciklopedinis kompiuterijos žodynas 2012).
Prievadas	Tai sąsaja, duomenims arba komandoms persiųsti tarp kompiuterio ir jo išorinių įtaisų, į kitą kompiuterį, tinklą arba iš kito kompiuterio, tinklo ir pan. (Enciklopedinis kompiuterijos žodynas 2012).
Ryšio linija	Tai fizinis ryšio kanalas, naudojamas duomenų paketų perdavimui kompiuterių tinklu per laidą, šviesolaidį, radijo bangomis ar kita duomenų perdavimo terpę ir jungiantis du tinklo komponentus. Naudojami šie jungimo būdai: mazgas-mazgas, mazgas-klientas ir klientas-klientas.
Serveris	Tinklo mazgas, atliekantis tam tikras funkcijas pagal kitų tinkle esančių klientų kompiuterių paraiškas (užklausas).

	(Enciklopedinis kompiuterijos žodynas 2012).
Stacionarusis įvykių srautas	Įvykių srautas vadinamas stacionariuoju, jei jų skaičiaus bei kitų srauto laiko parametrų pasiskirstymas bet kuriuo laiko intervalu priklauso tik nuo to intervalo trukmės, bet nepriklauso nuo intervalo vietos laiko ašyje (Sakalauskas 2000).
Šakotuvus	Tai tinklo mazgo įrenginys gautą paketą padauginantis tiek kartų, kiek jis turi prievadų, išskyrus siuntusįjį prievadą (angl. <i>Hub</i> ).
Tiltas	Tinklo komponentas, siejantis tinklus, suderinantis skirtingus siejamų tinklų įvairių lygmenų protokolus, konvertuojantis (jei reikia) duomenis, persiūsdamas juos iš vieno tinklo į kitą (Enciklopedinis kompiuterijos žodynas 2012) (angl. <i>Bridge</i> ).
Tinklo klientas	Tai kompiuteris, spausdintuvas, mobilusis ar kitas įrenginys, naudojamas kompiuterio tinklo vartotojo, besinaudojančio tinklo paslaugomis aptarnavimui.
Tinklo mazgas	Tai įrenginys paskirstantis kompiuterių tinklu perduodamą duomenų srautą pagal tinklo administratoriaus ar automatiškai sudarytą schemą.
Užklausa	Tai pageidavimas gauti kokią nors informaciją (duomenis) (Enciklopedinis kompiuterijos žodynas 2012).

## Naudojamų paveikslų sąrašas

<b>1 pav.</b> OSI modelis, naudojamas informacijai surinkti .....	32
<b>2 pav.</b> Duomenų failo perdavimas kompiuterių tinklu.....	34
<b>3 pav.</b> Virtualaus kanalo nustatymas.....	35
<b>4 pav.</b> Komutatoriaus schema .....	41
<b>5 pav.</b> Analizuojamo tinklo mazgo struktūrinė schema .....	56
<b>6 pav.</b> Tinklo srauto registravimo schema.....	57
<b>7 pav.</b> Registruojamų paketų ilgių pasiskirstymas .....	58
<b>8 pav.</b> Charakteringas vienos paros tinklo mazge užfiksuoto srauto grafikas .....	70
<b>9 pav.</b> SSE komponentų sąveikos schema .....	78
<b>10 pav.</b> Savastingumo įverčių su Fractan 4.4, Selfis 0.1b ir SSE pasiskirstymas .....	80
<b>11 pav.</b> Tinklo srauto SRD (kairėje) ir LRD (dešinėje) įverčių programa Fractan pasiskirstymas, kai $\beta \in [-1,1], \Delta\beta = 0,1$ .....	81
<b>12 pav.</b> Selfis Hurst'o reikšmių pasiskirstymas, kintant $\beta$ (kairėje – SRD, dešinėje – LRD) .....	81
<b>13 pav.</b> Selfis Hurst'o reikšmių pasiskirstymas, kintant $\beta$ , kai rezultatas rodo triukšmą .....	81
<b>14 pav.</b> SSE programos $\alpha$ koeficiento reikšmių pasiskirstymas, kintant $\beta$ .....	82
<b>15 pav.</b> SSE $\beta$ reikšmių pasiskirstymas.....	82
<b>16 pav.</b> Charakteringi Hurst'o eksponentės grafikai .....	84
<b>17 pav.</b> Hurst'o koeficiento reikšmių įverčiai, kai naudotas agregavimo intervalas 100ms.....	85
<b>18 pav.</b> Hurst'o koeficiento reikšmių įverčiai, kai naudotas agregavimo intervalas 500ms.....	86
<b>19 pav.</b> Hurst'o koeficiento reikšmių įverčiai, kai naudotas agregavimo intervalas 1000ms.....	86
<b>20 pav.</b> McCulloch koreliacijos matrica .....	87
<b>21 pav.</b> Regresijos koreliacijos matrica .....	88
<b>22 pav.</b> Momentų koreliacijos matrica .....	88
<b>23 pav.</b> Šiaulių universiteto LitNet tinklo mazgo modelis, parengtas programa Opnet .....	92
<b>24 pav.</b> Vieno kanalo aptarnavimo sistemos schema .....	94
<b>25 pav.</b> Vieno kanalo aptarnavimo sistema su LIFO buferiu.....	95
<b>26 pav.</b> Paraiškų aptarnavimas, kai eilės disciplina FIFO.....	97
<b>27 pav.</b> Paraiškų aptarnavimas, kai eilės disciplina LIFO .....	98
<b>28 pav.</b> Paraiškų aptarnavimo algoritmas MulNodSimSys, kai mazge naudojama LIFO eilė.....	99
<b>29 pav.</b> Daugiakanalė aptarnavimo sistema GI/G/10//N .....	109
<b>30 pav.</b> $P_L, \bar{T}_{SP}$ ir $\bar{T}_{delay}$ priklausomybė nuo $\lambda$ , kai $SP_{disc} = FIFO$ PP srautui ...	116
<b>31 pav.</b> $P_L, \bar{T}_{SP}$ ir $\bar{T}_{delay}$ priklausomybė nuo $\lambda$ , kai $SP_{disc} = LIFO$ PP srautui ...	116
<b>32 pav.</b> $P_L, \bar{T}_{SP}$ ir $\bar{T}_{delay}$ priklausomybė nuo $\lambda$ , kai $SP_{disc} = FIFO$ PS srautui....	117
<b>33 pav.</b> $P_L, \bar{T}_{SP}$ ir $\bar{T}_{delay}$ priklausomybė nuo $\lambda$ , kai $SP_{disc} = LIFO$ PS srautui....	117
<b>34 pav.</b> $\bar{T}_{SP}$ įverčiai, kai $\rho > 1$ PP tipo srautams .....	118

<b>35 pav.</b> $\bar{T}_{SP}$ įverčiai, kai $\rho > 1$ PS tipo srautams.....	118
<b>36 pav.</b> $\bar{T}_{SP}$ įverčiai, kai $\rho > 1$ SP tipo srautams.....	119
<b>37 pav.</b> $\bar{T}_{SP}$ įverčiai, kai $\rho > 1$ SS tipo srautams.....	119
<b>38 pav.</b> $\bar{T}_{delay}$ įverčiai, kai $\rho > 1$ PP tipo srautams.....	119
<b>39 pav.</b> $\bar{T}_{delay}$ įverčiai, kai $\rho > 1$ PS tipo srautams .....	120
<b>40 pav.</b> $\bar{T}_{delay}$ įverčiai, kai $\rho > 1$ SP tipo srautams .....	120
<b>41 pav.</b> $\bar{T}_{delay}$ įverčiai, kai $\rho > 1$ SS tipo srautams.....	120
<b>42 pav.</b> Atviras tinklo mazgo valdymo modelis su grįžtamuoju ryšiu .....	127
<b>43 pav.</b> Kompiuterių tinklo srauto analizės realiu laiku būdas.....	133
<b>44 pav.</b> Kompiuterių tinklo srauto analizės realiu laiku būdo realizavimo algoritmas .....	135
<b>45 pav.</b> Programinių modulių bibliotekos SSE-OL komponentų schema.....	138
<b>46 pav.</b> SSE-OL naudojimo tinklo mazge schema .....	140
<b>47 pav.</b> SSE-OL naudojimo, kai matavimo kompiuteris prijungtas prie tinklo mazgo schema .....	140



## *Naudojamų lentelių sąrašas*

<b>4.1 lentelė.</b> <i>Laiko eilučių savastingumo procentiniai</i> .....	87
<i>įverčiai, naudojant skirtingas programas bei metodus</i> .....	87
<b>4.2 lentelė.</b> <i><math>\beta</math> įverčiai regresijos metodu</i> .....	89
<b>5.1 lentelė.</b> <i>Tinklo srauto savybių įtakos mazgo darbui, pagal aptarnavimo intensyvumą, įverčiai</i> .....	102
<b>5.2 lentelė.</b> <i>Tinklo srauto savybių įtakos mazgo darbui, pagal mazgo buferio dydį, įverčiai</i> .....	103
<b>5.3 lentelė.</b> <i>Tinklo srauto savybių įtakos mazgo darbui, pagal eilutės stabilumo parametrą, įverčiai</i> .....	104
<b>5.4 lentelė.</b> <i><math>N_{buf}</math> ir <math>SP_{disc}</math> įtakos priklausomiems kintamiesiems įverčiai</i> .....	115
<b>5.5 lentelė.</b> <i>Kanalų apkrovos įverčiai, kai <math>\rho &gt; 1</math></i> .....	121
<b>1 priedas, 1 lentelė.</b> <i>Hurst'o koeficiento reikšmių pasiskirstymas</i> .....	172
<b>1 priedas, 2 lentelė.</b> <i>Hurst'o koeficiento pasikliautieji intervalai</i> .....	172
<b>2 priedas, 1 lentelė.</b> <i>Hurst'o koeficiento reikšmių pasiskirstymas</i> .....	174
<b>2 priedas, 2 lentelė.</b> <i>Hurst'o koeficiento įverčiai</i> .....	175

## 1 skyrius. Įvadas

### 1.1. Tyrimų sritis

Žinoma, kad per kompiuterių tinklo mazgą perduodamų duomenų paketų srautas yra dinaminis, sunkiai prognozuojamas ir primena triukšmą. Tokio srauto tradiciniais matematiniais metodais dažniausiai neįmanoma aprašyti (Kleinrock 2002, Ilnickij 2004, He, Gao, Hou, Park 2004, Kaklauskas, Sakalauskas 2008). *Ethernet* tinklo paketinio srauto tyrimai, atlikti 1989 m. A. Erramilli, O. Narayan ir W. Willinger, patvirtino, kad šio srauto charakteristikos pasižymi fraktališkumu (Zhang, Wu, Yang 2004) bei savastingumu<sup>1</sup> (angl. *Self-Similarity*) su ilgalaike atmintimi (angl. *Long-Range Dependence* – toliau *LRD*). Tolesni empiriniai kompiuterių tinklo srauto tyrimai patvirtino jo savastingumą (Crovella, Bestavros 1996, Beran 1998, Taqqu Teverovsky 1998, Roughan, Veitch, Abry 2000, Kaj 2002, Samorodnitsky 2006a, Shena, Zhua, Lee 2007, Kaklauskas, Sakalauskas 2008, Ginno, Gastron 2009, Asars, Petersons 2009, Kaklauskas, Sakalauskas 2011b ir kt.). Pastaroji savybė leidžia prognozuoti kompiuterių tinklo srauto kaitą, ir prognozės rezultatus taikyti srauto aptarnavimo kokybei *QoS* (angl. *Quality of Service* – toliau *QoS*) gerinti, reguliuojant tinklo srauto paketų vėlinimą, srauto fluktuacijas bei paketų praradimą (Abdelnaser, Amarnath 2000, Soldatos, Vayias, Stathopoulos, Mitrou 2004, Kulikovs, Petersons 2008, Kaklauskas, Sakalauskas 2010a, Kaklauskas, Sakalauskas 2010b, Kaklauskas, Sakalauskas 2011a).

Yra sukurta daug ir įvairiomis metodikomis grįstų tinklo srauto savastingumo vertinimo metodų. Gana populiarūs paketinio tinklo srauto

---

<sup>1</sup> Kadangi terminas *self-smilarity* neturi visuotinai pripažinto atitikmens lietuvių kalboje, yra naudojamas prof. habil. dr. L. Sakalausko pasiūlytas terminas „savastingumas“. Jį savo darbuose ir disertacijoje naudoja dr. A. Kabašinskas. Terminas taip pat minimas leidinyje Athena. Filosofijos studijos, žurnale „Sociologija, mintis ir veiksmas“. Lietuviškoje mokslinėje literatūroje dažnai vietoj savastingumo termino naudojamas jo lietuviškas atitikmuo – savipanašumas. Šį terminą savo darbuose naudoja prof. habil. dr. D. Surgailis, prof. dr. J. Valantinas ir kiti.

savastingumo vertinimo metodai pagrįsti laiko sekų analize: liekanų dispersijos (angl. *Variance of Residuals*) (Peng 1994), dispersijos (angl. *Variance*) (Erramilli, Narayan, Willinger 1996), *R/S* (angl. *Rescaled Adjusted Range* – toliau *R/S*) (Beran 1998), absoliutinių reikšmių (angl. *Absolute Value*) kartais vadinamų absoliutinių momentų (angl. *Absolute Moments*). Taip pat naudojami dažninių/banginių procesų bangos vilnelių (angl. *Wavelet*) savybių įverčiais grindžiami metodai: sumažinimo (angl. *Whittle*) (Kokoszka, Taqqu 1996), *Abry-Veitch* (Abry, Veitch 1996), periodogramos (Taqqu, Teverovsky 1998). Laiko eilutę galima analizuoti nagrinėjant stabilijų atsitiktinių procesų stabilumo parametrus. (Fama, Roll 1971) pasiūlė *Fama-Roll* metodą simetrinių stabilijų parametrų įverčiams. (McCulloch 1986) savo darbuose išvystė *Fama-Roll* sukurtus stabilijų dydžių vertinimo metodus ir pateikė interpoliacines lenteles. *Regresijos* metodą stabilijų dydžių parametrų įvertinimui pasiūlė Kotrouvelis (Kotrouvelis 1981), jo taikymus finansinėms eilutėms plačiai naudojo I. Belovas, A. Kabašinskas ir L. Sakalauskas (2006). Momentų metodą stabilijų dydžių parametrų įvertinimui pasiūlė S. J. Press (1972). Laiko eilučių vertinimo metodai pastoviai tobulinami ir pasiūloma vis naujesnių jų įverčių (Maulik, Resnick 2003, Peng 2004, Lopez-Guerrero, Orozco-Barbosa, Makrakis 2005, Fukuda, Takayasu, Takayasu 2005, Elagha, AlShafee 2007, Gospodinov, Gospodinova 2007, Rutka 2008, Melakessoua, Sorgera, Suchaneckia 2009, Millán, Kaschel, Lefranc 2010, Elbiaze, Zhani, Cherkaoui, Kamoun 2010, Kaklauskas, Vaičiulis 2011). 2008 metais (Surgailis, Teyssiere, Vaičiulis 2008) laiko eilutės savastingumo vertinimui pasiūlė *IR* (angl. *Increment Ratio* – toliau *IR*) statistikos metodą. *IR* statistika pirmą kartą pritaikyta kompiuterių tinklo srauto analizei, nustatant jo savastingumą ir *LRD* (Kaklauskas, Sakalauskas 2011b).

Kompiuterių tinklo paketinio srauto savastingumo analizei naudojami matematiniai tinklo srauto modeliai. Tinklo procesams modeliuoti taikomi netiesinės analizės metodai, atsižvelgiantys į didelius tinklo srautų pliūpsnius,

aprašomus tikimybiniais modeliais su „sunkiomis uodegomis“<sup>2</sup> (angl. *Heavy-Tail*) (Sheluhin, Smolskiy, Osin, 2007). Tokiuose modeliuose analizuojamas ateinančių paketų kiekis arba laiko tarpai tarp ateinančių paketų. Kadangi tinklo srautas sunkiai prognozuojamas, todėl jam modeliuoti sėkmingai taikomi stochastiniai tinklo srauto modeliai (Kaj 1998). Seniausiai naudojamas tinklo srauto modelis yra *Puasono* modelis, kuris aprašo tinklo srauto paketų atėjimus kaip nepriklausomus sekos narius, pasiskirsčiusius pagal eksponentinį dėsnį (Paxson, Floyd 1995). Vienas iš pagrindinių šio modelio trūkumų yra tas, kad jis netinka srautams su dideliais pliūpsniais aprašyti. Kitas rečiau naudojamas yra paketų traukinio (srauto) modelis, kuris modeliuoja paketų grupes tarp dviejų galinių taškų A ir B, kai laikas tarp paketų atėjimų tarp yra mažesnis nei tam tikras skaičius, vadinamas maksimaliu leistinu atstumu tarp paketų (Jain, Routhier 1986, Willinger, Paxson, Taqqu 1998). Tyrimais nustatyta, kad šiuolaikinio kompiuterių tinklo srauto parametrus netiksliai aprašo klasikiniai Markovo modeliai (Lombardo, Morabito, Schembra 1998, Kleinrock 2002), kurie plačiai naudojami įvertinant klasikinių telefoninių tinklų rodiklius. Šiuolaikinio paketinio kompiuterių tinklo srauto modeliams rengti ne visada naudotinas nepertraukiamo laiko Markovo grandinių metodas (Robert, Le Boudec 1995, Klemm, Lindemann, Lohmann 2003). Autoregresiniai tinklo srauto modeliai geriausiai tinka realaus laiko tinklo srautų modeliavimui (Jonga, Penzer 2004, Alheraish 2004, Weisang, Awazu 2008). Šiuolaikinio tinklo srauto analizei plačiai naudojami *ON-OFF* tinklo srauto modeliai, kurie pagrįsti kompiuterių tinklo srauto savybėmis. Jų pagrindu sukurtas IPP (angl. *Interrupted Poisson Process*) modelis, kuris tinklo kanalą vaizduoja dviejose būsenose: pirmoje būsenoje jis apkrautas ir veikia, o antroje yra laisvas ir neužimtas (Dombry, Kaj 2011). Tinklo srauto simuliacijai dažnai taikomas Pareto skirstinys, leidžiantis sukurti nepriklausomas sekas pasižyminčias pliūpsniškumu (Abdelnaser 1997).

---

<sup>2</sup> Šį terminą savo darbuose naudoja prof. habil. dr. R. Leipus, habil. dr. R. Norvaiša ir kt. Termino apibrėžimą galima rasti straipsnyje R. Leipus, R. Norvaiša „Finansų rinkos teorijų pagrindai“, Pinigų studijos 2003-4, Ekonomikos teorija ir praktika, 16 psl.

Savastingojo srauto modeliavimui sėkmingai naudojami aptarnavimo teorijos instrumentai (Boxma, Cohen 1999). Tinklo procesų aprašymui naudojami netiesinės analizės metodai, įvertinantys tinklo srautų pliūpsnius, aprašomus tikimybiniais modeliais su sunkiomis uodegomis (Jeong, McNickle, Pawlikowski 1999, Pawlikowski, Yau 2000, Jeong, Lee, McNickle 2005, Sheluhin, Smolskiy, Osin 2007, Jeong, Lee, McNickle, Pawlikowski 2007, Kulikovs 2008, Kaklauskas, Sakalauskas 2008, Schmidt, Pawlikowski, McNickle 2009, Xiaolong, Geyong 2009, McNickle, Ewing, Pawlikowski 2010, Halima, Zhani, Cherkaoui, Kamoun 2010, Vieira, Bianchi, Lee 2010, Nogueira, Salvador, Valadas, Pacheco 2010, Kaklauskas, Sakalauskas 2010b). Tinklo srauto savybes vertinti galima tinklo srautą modeliuojant lanksčiais  $\alpha$ -stabiliaisiais skirtiniais, kurie paklūsta centrinei ribinei teoremai, aprašo sunkias uodegas bei asimetriją ir pasižymintys leptokurtiškumu (Gallardo, Makrakis, Orozco-Barbosa 2000, Janicki, Weron 2000, Yin, Lin, Sebastien, Li, Min 2005, Kabašinskas, Rachev, Sakalauskas, Sun, Belov 2006, Kaklauskas, Sakalauskas 2008, Surgailis, Teyssiere, Vaičiulis 2008, Kulikovs, Petersons 2008, Grigelionis 2008, Lavancier, Philippe, Surgailis 2009, Minkevičius 2010, Kaklauskas, Vaičiulis 2011, Kaklauskas, Sakalauskas 2011a). Tinklo srauto savastingumas nustatytas ir naujos kartos technologiniuose tinklų sprendimuose, pavyzdžiui, paskirstytų skaičiavimų tinkluose GRID (Loiseau, Gonçalves, Dewaele, Borgnat, Abry, Primet 2010), mobiliuosiuose tinkluose (Yanga, Yang, Kim, Kim 2010).

Naudojant savastingųjų tinklo srautų modelius galima vertinti tinklo srauto savybių įtaką tinklo mazgo (komutatorius, maršrutizatorius, serveris ir kt.) parametrų (Manjunath, Sikdar 2002, Gyires 2003, Kaklauskas, Sakalauskas 2010b). Tokiuose modeliuose naudojamas sintetinis tinklo srautas, sugeneruotas pagal tyrėjo nurodytus parametrus, arba realiame kompiuterių tinkle surinkti srauto duomenys. Kai tinklo srautas neturi prioritetinių duomenų paketų, tinklo mazguose eilių aptarnavimui naudojamos *FIFO* ir *LIFO* eilių disciplinos (Kesselman, Lotkeryx, Mansour, Patt-Shamiryx, Schieberz, Sviridenkoz 2001, Huang, Chang, Cheng, Lee 2007, Pranevičius 2008). J.

Nzouonta, T. Ott ir C. Borcea (2009) analizuodami *ad hoc* tinklų *UDP* srautą nustatė, kad *LIFO* ir *FIFO* aptarnavimo disciplinos turi vienodus vėlinimus, kai srautas mažas, tačiau kai srautas didelis, tai *LIFO* Frontdrop pasižymėjo daugiau nei dvigubai mažesniu vėlinimu nei *FIFO* Taildrop. Mazgo eilės ilgio įtaką jo darbui nagrinėjo G. Appenzeller, I. Keslassy, M. Enachescu, Y. Ganjali ir daugelis kitų mokslininkų (Appenzeller, Keslassy, McKeown 2004; Enachescu, Ganjali, Goel, McKeown, Roughgarden 2005, Ramanan, Cao 2002, Kulikovs, Petersons 2008). Tyrimais nustatyta, kad mazguose aptarnaujančiuose didelės spartos (apie 30Gb/s) ir pralaidos kompiuterių tinklus pilnai pakanka 15-20 paketų buferio (Nzouonta, Ott, Borcea 2009). Tinklo srauto savastingumo savybė sėkmingai naudojam tinklo paketų transporto srauto perkrovų reguliavimui (He, Gao, Hou, Park 2004, Chung, Seol, Choi, Yeoum, Lim 2008, Kaklauskas, Sakalauskas 2010b).

Vertinant paketinio tinklo srauto savybes realiu laiku, atsiranda galimybė dinamiškai prognozuoti srauto kaitą (Mark, Ramamurthy 1998, Roughan, Veitch, Abry 2000, Hagiwara, Doi, Tode, Ikeda 2001, Chandramathi, Shanmugavel 2003, Younis, Fahmy 2005, Fu 2006, Zaidi, Hakami, Landfeldt, Moors 2010, Baccarelli, Zhu 2011, Cordeschi, Patriarca 2012). Tinklo srauto dinaminis įverčius galima taikyti srauto aptarnavimo kokybės *QoS* gerinimui, taip užtikrinant optimalų atėjusių paketų aptarnavimą, minimalius paketų praradimus tinklo mazge (Quan, Chung 2004, Mao 2005, Smith 2006, Lianming, Jianping, Xiaoheng 2011). Nustatyta, kad tinklo srauto aptarnavimas mazge gerėja parenkant optimalų eilės ilgį, eilės aptarnavimo discipliną, mažinant paketų vėlinimą bei keičiant kitus mazgo parametrus taip, kad jie geriausiai atitiktų dinaminis tinklo srauto savybių įverčius (Kaklauskas, Sakalauskas 2011a).

## **1.2. Problemos aktualumas**

Kompiuterių tinklai, naudojami nuotolinėms studijoms, dažnai patiria neprognozuojamas perkrovas dėl studijų proceso specifiškumo. Siekiant sumažinti tinklo perkrovas ir su tuo susijusius perduodamos informacijos

praradimus reikia stebėti tinklo ir jo mazgų būseną ir, remiantis gautais rezultatais, prognozuoti tinklo bei jo mazgų apkrovas bei perkrovas. L. Kaklauskas ir L. Sakalauskas 2008-2011 metais atliko empirinius Šiaulių universiteto e. studijų kompiuterių tinklo paketinio srauto tyrimus patvirtinti, kad šio tinklo srautas pasižymi savastingumu. Tokį tinklo srautą galima analizuoti kaip fraktalinį procesą, pasižymintį antros eilės statistiniu savastingumu, charakterizuojamu fraktaliniu matu, atraktorais, autokoreliacijos funkcijos savybėmis bei fazinės erdvės matu. Gausūs savastingųjų procesų kompiuterių tinkluose tyrimai rodo, kad, remiantis šia savybe, galima prognozuoti srauto kaitą ir prognozės rezultatus taikyti tinklo srauto aptarnavimo kokybei užtikrinti.

Reikia pažymėti, kad nėra išsamių tyrimų, kuriuose būtų įvertinta tinklo mazgo buferio talpos ir eilės aptarnavimo disciplinos įtaka savastingojo tinklo srauto aptarnavimo kokybei. Vertinant tinklo srauto savastingumą realiu laiku galima dinamiškai prognozuoti srauto kaitą. Prognozės rezultatus galima taikyti tinklo srauto aptarnavimo kokybės *QoS* gerinimui, reguliuojant paketų užlaikymą ir trumpinant jų aptarnavimo laiką tinklo mazge, mažinant srauto fluktuacijas.

Lietuvos mokslininkai dažniausiai analizuoja teorinius tinklo srauto savastingumo analizės modelius, tačiau eksperimentinis modeliavimas nėra atliekamas (Rindzevičius, Tervydis 2003, Rutka 2008 Pranevičius 2008). Tinklo apkrovos savastingumo indikatoriai realiu laiku, taikant  $\alpha$ -stabiliuosius modelius, nėra išsamiai tirti nei Lietuvoje, nei užsienyje. Reikia pažymėti, kad tinklo srauto apkrovos savastingumo savybių tyrimo realiu laiku metodai dar nėra pakankamai ištirti. Ši problema yra aktuali ir dar nepakankamai išnagrinėta teoriniu bei praktiniu aspektais. Šio darbo tyrimų sritis apima kompiuterių tinklo paketinio srauto fraktalinių procesų stebėjimą ir analizę, srauto savastingumo savybių realiu laiku įvertinimo metodų tyrimą bei srauto valdymo modelių eksperimentinį tyrimą, atsižvelgiant į tinklo srauto savybes bei tinklo mazgo parametrus.

### ***1.3. Tyrimų objektas***

Disertacijos tyrimų objektas yra paketinio kompiuterių tinklo srauto savybės, tinklo mazgo savybių įtaka paketinio kompiuterių tinklo srauto aptarnavimui, realaus laiko tinklo srauto savybių analizės metodai ir jų taikymas tinklo srauto kaitos dinaminiam prognozavimui.

### ***1.4. Tyrimų tikslas ir uždaviniai***

Darbo tikslas ištirti fraktalinius procesus kompiuterių tinkluose, remiantis gautais rezultatais parinkti metodus tinkamus tinklo srauto analizei realiu laiku ir sukurti savastingumo matavimo realiu laiku metodiką bei ją pritaikyti kompiuterių tinklų aptarnavimo kokybei gerinti.

Siekiant šio tikslo, darbe sprendžiami šie uždaviniai:

- išnagrinėti kompiuterių tinklų fraktališkumo (savastingumo) tyrimo metodus;
- parinkti metodus tinklo srauto savastingumo analizei realiu laiku;
- sukurti kompiuterių tinklo srauto savastingumo matavimo realiu laiku metodiką bei atitinkamus įrankius;
- sukurti kompiuterių tinklų informacinių srautų imitavimo sistemą ir pritaikyti ją sukurtai metodikai testuoti;
- pritaikyti sukurtą metodiką kompiuterių tinklų aptarnavimo kokybei *QoS* gerinti.

### ***1.5. Tyrimo metodai***

Analizuojant kompiuterių tinklų matematinius modelius, tinklo srauto fraktališkumo vertinimo metodus, tinklų modeliavimą naudoti mokslinės informacijos paieškos, lyginamosios analizės ir rezultatų apibendrinimo metodai. Paketinio kompiuterių tinklo srauto eilučių formavimui pritaikytas klasterinės analizės *k*-vidurkių metodas. Savastingumo metodai vertinti panaudojant koreliacinę analizę. Matematinis modeliavimas panaudotas tinklo srauto savastingumo analizei ir imitaciniam vienakanalių bei daugiakanalių



tinklo mazgų modeliavimui. Taikant eksperimentą imituoti fraktaliniai procesai kompiuterių tinkluose, o eksperimento rezultatai analizuoti taikant tiesinę regresinę analizę bei sukinius duomenų grupavimui.

### ***1.6. Mokslinis naujumas***

Darbe gauti tokie rezultatai:

1. Parengtas tinklo srauto savybių analizės paketas, panaudotas analizuoti, vertinti ir palyginti kompiuterių tinklų fraktališkumo ir savastingumo tyrimo metodus;
2. Ištirti paketinio kompiuterių tinklo srauto savastingumo analizės metodai:
  - a. pagrįsti laiko sekų analize,
  - b. paremti dažninių/banginių procesų savybių vertinimu,
  - c. laiko eilutės stabilumo parametrų įverčiais.
3. Sudarytas tinklo srauto savastingumo realiu laiku analizės paketas (toliau SRLAP).
4. Naudojant *SRLAP* savastingumo matavimui realiu laiku atrinktas robastinis *regresijos* metodas ir *IR* statistika.
5. Sukurta kompiuterių tinklo srauto savastingumo matavimo realiu laiku metodika, kuri rekurentiškai formuoja eilutę, skaičiuoja statistinius jos parametrus, ir rezultatus saugo laikinojoje atmintyje. Ši metodika nuo kitų skiriasi tuo, kad realiu laiku agreguoja tinklo srautą taikant judančių vidurkių glotninimo arba perduotų duomenų srauto sumos per laiko intervalą metodą, suformuotą eilutę saugo laikinoje atmintyje, rekurentiškai skaičiuoja dinamiškai formuojamos eilutės savastingumo parametrus, apskaičiuojant  $\alpha$  stabilumo indeksą, *IR* įvertį ir *Hurst* koeficientą, skaičiavimo rezultatus tolesnei analizei saugo laikinojoje arba/ir pastoviojoje atmintyje. Metodas užpatentuotas Lietuvos respublikos valstybiniame patentų biure, patento numeris LT20011099.

6. Sukurta unikali kompiuterių tinklo savastingojo ir *Puasono* srautų imitavimo sistema ir, ja naudojantis, iširta tinklo mazgo savybių įtaka paketinio tinklo srauto aptarnavimo kokybei.
7. Sukurta metodika pritaikyta kompiuterių tinklo aptarnavimo kokybei *QoS* gerinti. Remiantis tinklo srauto savastingumo matavimo realiu laiku metodika, suprojektuotas tinklo srauto valdymo realiu laiku modelis.

### ***1.7. Praktinė darbo reikšmė***

Darbe gauti tokie praktiniai rezultatai:

1. Sukurta kompiuterių tinklo duomenų srauto savastingumo įvertinimo realiu laiku metodika bei įrankiai;
2. Sukurta kompiuterių tinklo savastingųjų srautų su sunkiomis uodegomis imitavimo metodika bei įrankiai;
3. Pasiūlyti savastingojo kompiuterių tinklo srauto su sunkiomis uodegomis aptarnavimo kokybės *QoS* sprendimai, gaunami įvertinus informacinių srautų savastingumą ir jų pagrindu suprojektuotas tinklo srauto valdymo realiu laiku modelis.

### ***1.8. Ginamieji teiginiai***

Pagrindiniai darbo teiginiai, pateikiami gynimui, yra šie:

1. Kompiuterių tinklo srauto savastingumo vertinimui naudojamos laiko sekų analize, dažninių/banginių procesų savybių vertinimu bei eilutės stabilumo parametrų įverčiais grįstos metodų grupės. Naujai pritaikyti laiko eilutės stabilumo parametrų įverčiai tinklo srauto savastingumo vertinimui realiu laiku, kurie gaunami apskaičiuojant: uodegų indeksą pagal robastinę *regresijos* metodą ir *IR* statistikos įverčius.
2. Kompiuterių tinklo paketinio srauto fraktaliniai procesai įtakoja tinklo srauto aptarnavimo kokybę *QoS* tinklo mazge ir priklauso nuo tinklo mazgo parametrų. Pagal tinklo srauto savybių įverčius,

keičiant tinklo mazgo charakteristikas, eilės ilgį ir eilės aptarnavimo discipliną, mažėja paketų praradimo tikimybė, vidutinis paketo buvimo laikas tinklo mazgo eilėje ir mazge, pilno eilės užpildymo laikas bei vėlinimas. Tokie mazgo savybių pokyčiai užtikrina geresnę tinklo srauto aptarnavimo kokybę tinklo mazge.

3. Naujas kompiuterių tinklo srauto savastingumo vertinimo realiu laiku metodas leidžia dinamiškai keisti tinklo mazgo savybes, taip užtikrinant geresnį srauto aptarnavimą tinklo mazge.

### ***1.9. Rezultatų aprobavimas***

Tyrimų rezultatai buvo pristatyti ir aptarti:

1. On network traffic statistical analysis. LMD XLIX konferencija, 2008 m. birželio 25-26.
2. Seminaras „Kompiuterių tinklų informacinių srautų savastingumo tyrimas“. Matematikos-informatikos institutas, duomenų analizės skyrius, 2008 rugsėjo 24.
3. Seminaras G. Samorodnitsky. “Long Range Dependence”. H. Orsted Institute of the University of Copenhagen, 2008 October 20-23.
4. Seminaras „Fraktalinių procesų kompiuterių tinkluose tyrimas“. Šiaulių universitetas, 2008 lapkričio 21.
5. Application of Chaos Theory to Analysis of Computer Network Traffic. XIII International Conference “Applied Stochastic Models and Data Analysis”, Vilnius Lithuania, 2009 m. birželio 30 – liepos 3d.
6. Tinklo apkrovos fraktališkumo tyrimas realiu laiku. XIV tarptautinė kompiuterininkų konferencija, 2009 m. rugsėjo 26.
7. Interaktyvus *Hurst* statistikų analizatorius. LMD 51 konferencija, 2010 m. birželio 17-18 d.

8. Seminaras „Fraktalinių procesų kompiuterių tinkluose stebėsenos ir valdymo metodų tyrimas“. Klaipėdos universitetas, 2012 balandžio 13.

### ***1.10. Darbo rezultatų publikavimas***

Tyrimo rezultatai publikuoti šiuose moksliniuose leidiniuose:

*Užsienio mokslo leidiniuose, įtrauktuose į Mokslinės informacijos instituto pagrindinių žurnalų sąrašą:*

1. L. Kaklauskas, L. Sakalauskas (2011). Study of the Impact of Self-Similarity on the Network Node Traffic. Electronics and electrical engineering, No. 5(111), p. 27-32.
2. L. Kaklauskas, L. Sakalauskas (2011). Study of on-line measurement of traffic self-similarity. Central European Journal of Operations Research, DOI: 10.1007/s10100-011-0216-5. Accepted on 16-06-2011.  
<http://www.springerlink.com/content/0036x207656pq211/>

*Tarptautinių konferencijų darbuose, įtrauktuose į Mokslinės informacijos instituto sąrašą:*

1. L. Kaklauskas, L. Sakalauskas (2009). Application of Chaos Theory to Analysis of Computer Network Traffic. Proceedings of XIII International Conference “Applied Stochastic Models and Data Analysis”, Vilnius Lithuania, 407-412. (ISI Proceedings).

*Recenzuojamuose Lietuvos ir užsienio leidiniuose:*

1. L. Kaklauskas, L. Sakalauskas (2008). On network traffic statistical analysis. Lietuvos matematikos rinkinys. LMD darbai, 48/49, 2008, pp. 314–319.

2. L. Kaklauskas, L. Sakalauskas (2010). Tinklo apkrovos savastingumo tyrimas realiu laiku. „Informacijos mokslai“, 53 tomas, pp. 100-105.
3. L. Kaklauskas, L. Sakalauskas (2010). Savastingumo įtakos tinklo mazgo darbui tyrimas. Jaunųjų mokslininkų darbai, Nr. 3, 100-107.
4. Kaklauskas Liudvikas, Marijus Vaičiulis (2011). Skirstinio uodegos indekso MAX-DPR įvertis. Jaunųjų mokslininkų darbai, Nr. 1, p 126-130.

### ***1.11. Disertacijos struktūra***

1 skyriuje pateikiamas disertacijos tikslas, uždaviniai, tyrimo objektas. Aptariamas mokslinis naujumas, praktinė darbo reikšmė. Pateikiamas darbo rezultatų aprobavimo ir publikavimo aprašas.

2 skyriuje aprašomas kompiuterių tinklų matematinis modelis, šiuolaikinių tinklų technologiniai sprendimai, tinklo įrangos ir informacinių srautų perdavimo technologiniai sprendimai, paketų struktūra. Aprašomi kompiuterių tinklo paketinio srauto modeliai ir modeliai naudojantys aptarnavimo teorijos instrumentus.

3 skyriuje pateikiami fraktalinių srautų kompiuterių tinkluose matavimo technologijos. Aptariami technologiniai tinklo srauto matavimo sprendimai. Analizuojamas tinklo srauto parametrų įvertinimas pagal chaoso teoriją, skaičiuojant *Hurst*'o statistikas, gaunant stabiliųjų parametrų įverčius ir taikant IR statistiką.

4 skyriuje aptariamos tinklo srauto analizės priemonės, srauto analizės metodų testavimas, realaus srauto laiko eilučių formavimas ir agregavimo problematika bei apibendrinami kompiuterių tinklo srauto laiko eilučių analizės rezultatai.

5 skyriuje analizuojama fraktalinių procesų imitavimo programinė įranga. Aptariamas vieno kanalo tinklo mazgo aptarnavimo imitavimo modelis, apibendrinami jo tyrimo rezultatai. Panaudojant stochastiškai apribotą tinklo

komunikavimo modelį analizuojamas GI/G/m//N tinklo mazgas, įvertinami jo tyrimo rezultatai.

6 skyriuje pateikiamas savastingumo matavimo realiu laiku metodas bei nagrinėjamas paketinio kompiuterių tinklo valdymo modelis. Aptariami analogiški, pasaulyje naudojami, matavimo metodai realiu laiku, jų privalumai ir trūkumai, aprašomas pateiktas išradimas – tinklo srauto analizės realiu laiku metodas. Analizuojamas atviras tinklo mazgo valdymo modelis su grįžtamuju ryšiu, kur išradimas pritaikytas dinaminiam tinklo mazgo valdymui.

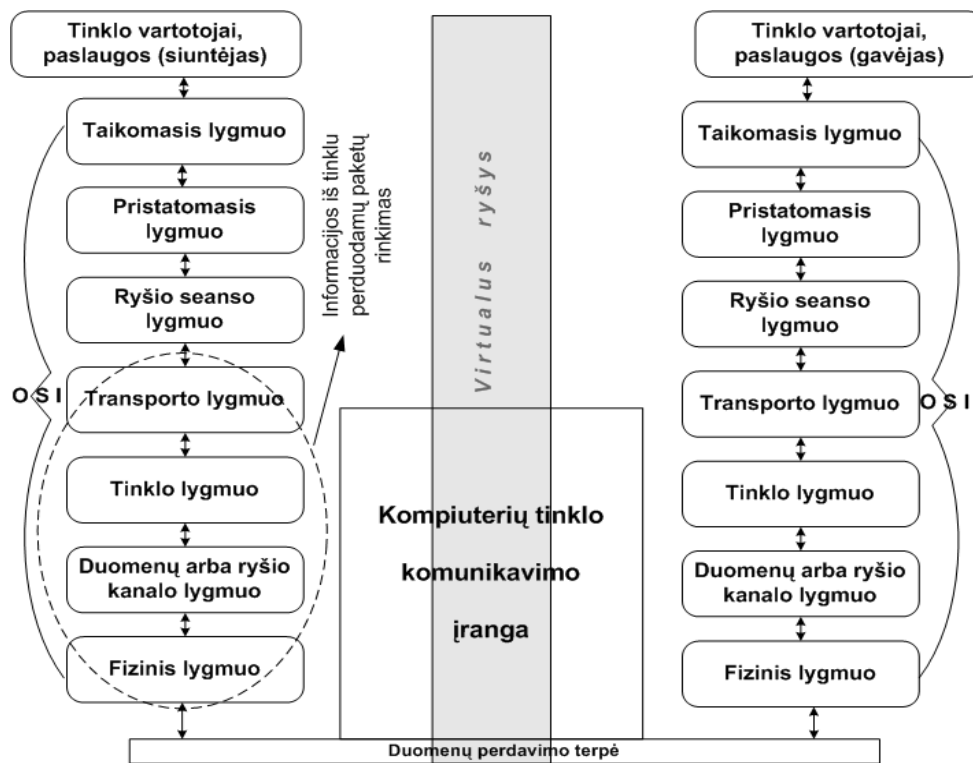
## **2 skyrius. Kompiuterių tinklų modeliavimas**

Kompiuterių tinklas – tai grupė kompiuterių ir kitų įrenginių, sujungtu ryšio linijomis taip, kad užtikrintų duomenų keitimąsi tarp jų. Kompiuterių tinklas gali būti skaidomas į potinklius, kurie yra tinklo poaibiai. Kompiuterių tinklą sudaro: tinklo mazgai, tinklo klientai ir ryšio linijos. Tinklo mazgai paskirsto kompiuterių tinklu perduodamą duomenų srautą. Tinklo mazguose duomenų paketų paskirstymui dažniausiai naudojami maršrutizatoriai, komutatoriai, šakotuvai bei tiltai. Maršrutizatorius naudojamas didelių tinklų formavimui bei jų skaidymui į potinklius, užtikrina informacijos perdavimą pagal iš anksto aprašytus maršrutus, apjungia įvairių rūšių ir paskirčių tinklus į bendrą tinklą (Kaklauskas 2003). Jis kartais dar vadinamas maršruto parinktuvu, nes parenka optimalų paketo perdavimo gavėjui kelią. Komutatorius paskirsto duomenų paketus tinkle, atsižvelgiant į siuntėjo ir gavėjo adresus, dažnai naudojamas tinklo potinklių arba segmentų aptarnavimui (Kaklauskas 2003). Paprastesnis įrenginys yra vadinamas šakotuvu arba kartotuvu. Jis gautą paketą padauginą tiek kartų, kiek jis turi prievadų, išskyrus siuntusiją. Toks paketų dauginimas didina tinklo apkrovą, todėl šie įrenginiai ypač retai naudojami kompiuterių tinkluose. Tiltas naudojamas apjungti skirtingų standartų ir technologinių sprendimų tinklus. Kompiuterių tinklo paslaugomis naudojami tinklo klientų kompiuteriai, spausdintuvai, mobilieji ar kiti įrenginiai. Tinklo klientai dažniausiai jungiami prie tinklo mazgo, o paslaugas gauna iš paslaugų serverio. Kompiuteriai ir kiti tinklo įrenginiai apjungiami ryšio linijomis, t. y. įvairių konstrukcijų variniais laidais, šviesolaidiniais laidais bei numatytų dažnių radijo bangomis.

### ***2.1. Tinklų technologijos***

Kompiuterių tinkluose informacijos perdavimo procesai funkciname lygmenyje aprašomi remiantis OSI (angl. *Open System Interconnection*) modeliu, kurį sudaro septyni sluoksniai (žr. 1 pav.). Pagal jį informacijos

mainų tarp tinklo objektų procesas išskaidomas į septynis uždavinius, kurių kiekvienas atitinka OSI modelio tam tikrą sluoksnį. Informacijos apie paketinio tinklo srautą surinkimui pakanka keturių žemiausių sluoksnių: fizinis – teikia informaciją apie tinklo įrenginių fizinius adresus, duomenų arba ryšio kanalo – teikia informaciją apie duomenų iškraipymą, paketo ilgį ir paketo atėjimo laiką, tinklo – teikia informaciją apie siuntėjo ir gavėjo adresus, transporto – apie naudojamos paslaugos tipą.



1 pav. OSI modelis, naudojamas informacijai surinkti

Kompiuterių tinklai yra standartizuojami, t. y. tinklo įrangos parametrai aprašomi remiantis tam tikru standartu. Analizuoti *Ethernet Type II* standarto paketai, kurie pagal Project 802<sup>3</sup> standartą, išleista 1980 metų vasario mėnesį, atitinka 802.3 kategoriją. Taip pat nagrinėti 802.1, 802.2, 802.8 ir 802.11 kategorijos paketai.

Pagal *Ethernet Type II* standartą yra nustatyti reikalavimai tinklo paketo ilgiui. Paketo ilgis, matuojant baitais (dažnai vadinamas oktetais, nes baitą

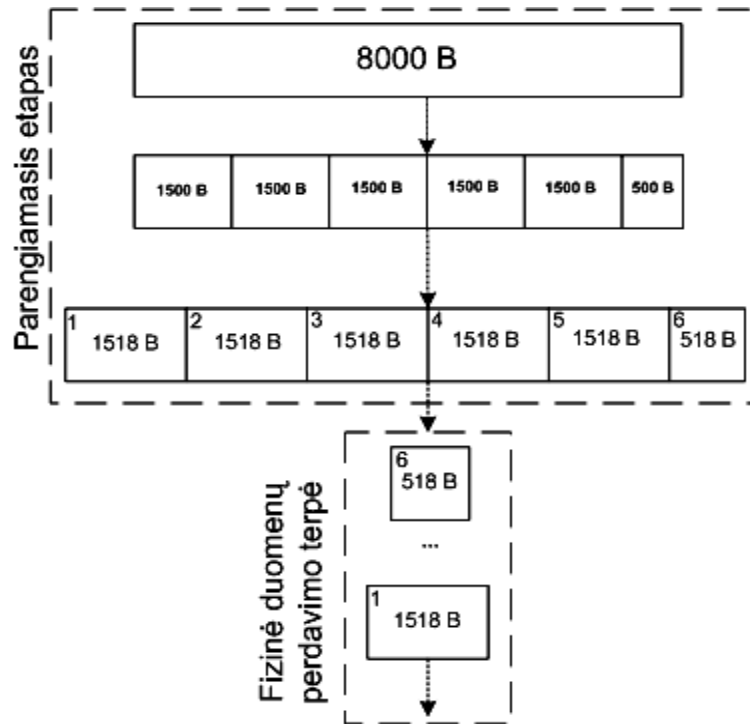
<sup>3</sup> IEEE 802 standartų komitetas. <<http://www.ieee802.org/>>.



sudaro 8 bitai) turi priklausyti intervalui [64,1518]. Maksimalus paketo ilgis žymimas *MTU* (angl. *Maximum Transmission Unit*) ir daugumoje šiuolaikinių paketinių kompiuterių tinklų yra 1500B be antraštės (Grimm, Schlüchtermann 2008). Antraštę, kurioje saugoma informacija apie siuntėjo/gavėjo adresus, paketo būseną ir kiti įrašai, užima 18 baitų. Atsižvelgiant į šiuos reikalavimus, informacija, kurią siunčia vartotojas tinklu yra skaidoma į porcijas – paketus (žr. 2 pav.), kurie tenkina šio standarto reikalavimus. Aišku, tam yra gaištamasi papildomas duomenų išskaidymo/surinkimo laikas, kuris yra įtrauktas į vartotojo laiką  $t_v$ . Kaip matyti 2 paveiksle duomenų failas, kurio apimtis 8000 baitų yra skaidomas į paketus, juos numeruojant. Gauname penkis pilnus paketus po 1500B ir vieną 500B paketą. Suskaidyti paketai papildomi tarnybine informacija, kuri talpinama jų antraštėse, todėl kiekvieno paketo ilgis padidėja 18B. Po to paketai po vieną išsiunčiami gavėjui. Pas gavėją išskaidyti paketai surenkami, pagal antraštėje nurodytus požymius, patikrinant ar gautas paketas yra nesugadintas, tam panaudojant antraštėje įrašytas kontrolines paketų sumas. Tinklo paketai gali būti sugadinti arba prarasti dėl jų susidūrimų dar vadinamų kolizijomis arba dėl per ilgų eilių tinklo mazguose.

Tinklo lygmenyje (žr. 1 pav.) duomenų paketai perduodami, panaudojant interneto protokolą IP (angl. *Internet Protocol*) aprašantį siuntėjo ir gavėjo adresus. Sėkmingam kiekvieno IP duomenų paketo perdavimui yra parenkamas geriausias einamu momentu maršrutas. Tai atliekama pagal pakete įrašytą informaciją, t. y. sprendžiamas geriausio paketų maršruto parinkimo uždavinys. Tinklo mazguose tam gali būti naudojamos administratoriaus sudarytos arba automatiškai formuojamos maršrutų lentelės. Automatinį lentelių formavimą atlieka maršrutizavimo protokolai, *CIDR* (angl. *Classless Inter-Domain Routing*), *OSPF* (angl. *Open Shortest Patch First*) ir kiti. Jie maršrutus tinkluose nustato automatiškai, tam naudodami specialius algoritmus. Pavyzdžiui, *OSPF* ryšius tarp kitų tinklo mazgų kuria taikydamas Dijkstros algoritmą (Dijkstra 1959). Pagal šį algoritmą susietiems grafams  $G=(N,E)$ , kur  $N$  – tinklo mazgų aibė, o  $E$  – tinklo mazgus jungiančių ryšio

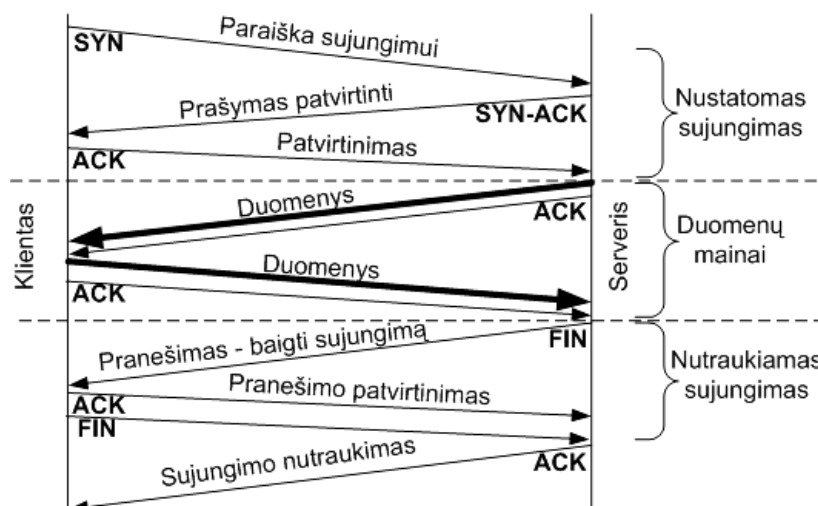
linijų aibė, yra kuriami trumpiausieji maršrutai – keliai  $p$  nuo siuntėjo iki gavėjo (Kaklauskas 2003).



**2 pav.** Duomenų failo perdavimas kompiuterių tinklu

Trumpai algoritmą (Dijkstra 1959), pagal kurį sukuriamas tinklo maršrutų sąrašas, galima apibūdinti taip:

- mazgas  $M$ , kuriame sudaromos maršrutų lentelės išsiuntinėja visiems savo tinklo mazgams pranešimą atsiusti savo maršruto svorio koeficientus pagal visas prie jo prijungtas ryšio linijas, įvertinantį linijos spartą;
- nustatomi kaimynai ir jų maršrutų svorio koeficientai įrašomi į maršrutų lentelės pirmą eilutę, o kitų svorio koeficientai toje eilutėje pažymimas kaip begalybė;
- dabar nuo mazgo  $M$  artimiausio kaimyno skaičiuojami svorio koeficientai iki jo kaimynų ir surašomi į maršrutų lentelės antrąją eilutę, atmetant pirmoje eilutėje dubliuojančius maršrutus, o iki kitų mazgų svorio koeficientai toje eilutėje pažymimi kaip begalybė;
- skaičiavimai tęsiami tol kol nebus įvertinti toliausiai nutolusio nuo  $M$  tinklo mazgo kiekvieno kaimyno svorio koeficientas.



**3 pav.** Virtualaus kanalo nustatymas

Duomenų perdavimui iš siuntėjo gavėjui transporto lygmenyje (žr. 1 pav.) yra naudojamas *UDP* protokolas, tada paketai vadinami datagramomis, arba *TCP* protokolas, tada paketai vadinami segmentais.

Prieš perduodant paketus klientas su serveriu apsikeičia tarnybiniais paketais ir taip sukuria virtualaus ryšio kanalą, naudojamą duomenų persiuntimui. 3 paveiksle pavaizduotas virtualaus sujungimo realizavimo procesas (Grimm, Schlüchtermann 2008). Pirmiausiai klientas siunčia serveriui užklausa dėl sujungimo *SYN* ir gauna patvirtinimą *SYN-ACK* apie sujungimo pradžią, tokį pat patvirtinimą *ACK* siunčia ir klientas serveriui. Po to pradedamas duomenų paketų siuntimas iš kliento kompiuterio serveriui. Kadangi tinkluose dažniausiai naudojamas asinchroninis duomenų perdavimo būdas, tai po kiekvieno paketo persiuntimo siunčiamas patvirtinimo paketas *ACK*. 3 paveiksle pavaizduotas duomenų mainų procesas yra dvikryptis (angl. *Duplex*). Baigus duomenų persiuntimą siunčiamas pranešimas apie sujungimo pabaigą *FIN*, kurį patvirtina klientas nusiudamas *ACK* ir *FIN* paketus. Po to patvirtinimą persiunčia serveris ir virtualus ryšio kanalas panaikinamas.

Kadangi *UDP* protokolas neturi paketų perdavimo greičio reguliavimo mechanizmo, tai jis dar vadinamas sujungimo protokolu. *TCP* protokolas turi integruotus du srauto kontrolės mechanizmus:

1. Srauto kontrolė, kuri padeda išvengti paketų praradimų, kai gavėjas yra nepajėgus priimti daugiau paketų arba laikinai užimtas. Apie savo užimtumą gavėją informuoja siuntėjas kiekvienoje paketo siunčiamo siuntėjui antraštėje, įrašydamas informaciją apie laisvą atmintį (paprastai buferį), kurios siuntėjas negali viršyti siusdamas paketus.
2. Perkrovos valdymo kontrolės mechanizmas padeda išvengti tinklo komponentų (komutatorių, maršrutizatorių ir pan.) perkrovų. Čia informacija siuntėjas sužino apie tinklo komponentų perkrovą, jei negauna duomenų gavimo patvirtinimo paketų *ACK*. Siuntėjas pakartoja tų paketų persiuntimą gavėjui, kuriems nebuvo atsiusta pavirtinimo (Kaklauskas 2003).

Didėjant kompiuterių tinklų spartai ir apkrovoms kuriami nauji srauto kontrolės mechanizmai. Siekiant užtikrinti, kad paketas, perduodamas kompiuterių tinklu, nesėkmės atveju būtų panaikintas, į jį įrašomas paketo gyvavimo laikas *TTL* (angl. *Time to Live* – toliau *TTL*). *TTL* reikšmė yra kintama ir priklauso nuo galimos perdavimo tarp siuntėjo ir gavėjo trukmės. Kompiuterių tinklu siunčiamus duomenų paketus galima skaidyti į tris pagrindines grupes (Grimm, Schlüchtermann 2008):

1. Tarnybiniai paketai – tai nedidelio ilgio paketai, dažniausiai jų ilgis svyruoja nuo 64B iki 200B, jie perduoda tinklo mazgo ar kito įrenginio darbui reikalingą informaciją apie maršrutus, kompiuterių darbą ir pan.;
2. Duomenų paketai, tai informaciją nešantys paketai:
  - a. iš kliento perduodami paketai apie pageidaujamą informaciją iš nutolusio serverio arba vartotojo medžiaga talpinama virtualiose aplinkose ir t.t.,
  - b. iš tinklo mazgo arba serverio perduodami duomenų paketai su vartotojo pareikalauta informacija, virtualios aplinkos vizualizavimo duomenimis vartotojo naršyklėje ir t.t.;
3. Kiti paketai – tai neatpažinti, „pasiklydę“ tinkle, su pasibaigusiu gyvavimo laiku *TTL* ar kitais paketo sugadinimo požymiais. Juos, jei

nenumatyta kitaip, tinklo mazgas arba serveris naikina. Reikia pažymėti, kad tvarkinguose tinkluose tokių paketų kiekis sudaro mažiau nei vieną viso duomenų srauto procentą.

Ryšio seanso, pristatomajame ir taikomajame lygmenyse veikiantys protokolai užtikrina kokybišką paslaugų teikimą tinklo vartotojui. Transporto protokolus duomenų perdavimui taip pat naudoja taikomojo lygmens protokolai:

- raidinių ir skaitinių adresų susiejimo protokolai (duomenų perdavimui naudojamas *UDP*) – *DNS* (angl. *Domain Name Protocol*) naudoja *UDP*, *ARP* (angl. *Address Resolution Protocol*) naudoja *UDP* ir pan.;
- tinklo valdymo ir kontrolės protokolai – *SNMP* (angl. *Simple Network Management Protocol*) naudoja *UDP*;
- pranešimų perdavimo tinklu protokolai, tokie kaip *ICMP* (angl. *Internet Control Message Protocol*) formuoja savo transporto tarnybinius paketus, *RSVP* (angl. *Resource Reservation Protocol*) formuoja savo transporto tarnybinius paketus ir pan.;
- vartotojų identifikavimo protokolai – *RADIUS* naudoja *TCP* ir pan.;
- nutolusios prieigos prie tinklo objekto resursų protokolai – *TELNET*, *SSH* naudoja *TCP* ir pan.;
- paslaugų teikimo vartotojui protokolai – *HTTP*, *SMTP*, *POP*, *IMAP* ir *FTP* naudoja *TCP* ir pan.;
- tinklo objektų sąveikos formavimo protokolai – *NFS* ir *SMB* naudoja *UDP* ir pan.;
- realaus laiko video ir garso perdavimo užtikrinimo protokolai – *RTP* ir *RTCP* naudoja *UDP*, *RTSP* naudoja *TCP* ir pan.;
- Kiti taikomojo lygmens protokolai.

Šiame skyrelyje aptarti protokolai informaciją perduoda kompiuterių tinklu įvairaus ilgio paketų pavidalu, kurie formuoja mūsų analizuojamą kompiuterių tinklo *UDP*, *TCP* transporto arba tarnybinių paketų, tokių kaip *ICMP*, srautą.

Pagal Grimm, Schlüchtermann (2008), atsižvelgiant į tinklo *OSI* architektūrą, *TCP/IP* tinkluose formuojamą duomenų paketų srauto modeliavimo procesą galima skaidyti į šešis lygmenis:

1. prieigos lygmenį (toliau *PrL*) – čia vartotojas prisijungia prie kompiuterių tinklo resursų, o šio lygmens sugaišto laiko matas gali siekti nuo kelių minučių iki kelių valandų;
2. paslaugų programų ar servisų iškvietimo lygmuo (toliau *PpSiL*) ( $PrL \subseteq PpSiL$ ) – čia vartotojas, gavęs prieigą, gali kviešti keletą paslaugų, šio lygmens sugaišto laiko matas gali siekti nuo kelių minučių iki kelių valandų;
3. dialogo lygmuo (toliau *DL*) ( $PpSiL \subseteq DL$ ) – čia programos reaguoja į vartotojui reikalingų duomenų pasirinkimą ir inicijuoja reikalingų duomenų perdavimą, šiame lygmenyje sugaišto laikas gali siekti nuo kelių sekundžių iki kelių minučių;
4. sujungimo lygmuo toliau *SL* ( $DL \subseteq SL$ ) – pagal inicijuotus perdavimus formuojami sujungimai duomenų perdavimui, šiame lygmenyje sugaištas laikas gali siekti nuo kelių milisekundžių iki kelių minučių;
5. pliūpsnio lygmuo *PIL* ( $SL \subseteq PIL$ ) – *TCP* segmentų ir *UDP* datagramų perdavimo formavimas, taikant aktyvias ir pasyvias duomenų perdavimo būsenas (*ON-OFF* srauto modelis, kurį aptarsime vėliau),
6. paketų lygmuo *PaL* ( $PIL \subseteq PaL$ ) – čia formuojami duomenų paketai ir, atsižvelgiant į fizinio tinklo parametrus, perduodami, šiame lygmenyje sugaištas laikas gali siekti nuo kelių mikrosekundžių iki kelių sekundžių, o jo modeliavimui dažnai taikomi matematiniai multifraktaliniai srauto modeliai.

Reikia pažymėti, jog prieigos lygmens paketinio kompiuterių tinklo srauto modelis pagal Grimm, Schlüchtermann (2008) yra sudarytas iš kitų penkių lygmenų tinklo srautų modelių.

## 2.2. Tinklo informacinio srauto modeliavimas

Modeliuojant informacinį kompiuterių tinklo srautą yra du pagrindiniai parametrai, kurie generuojami taikant kompiuterių tinklo modelius: paketų ilgių skirstinys ir laikų tarp paketų atėjimų skirstinys. Senesnieji tinklo srauto modeliai, tokie kaip *Puasono* su eksponentiniu paketų atėjimo laiko skirstiniu, intensyviai buvo naudojami iki srauto savastingumo savybės nustatymo (Leland, Taqqu, Willinger, Wilson 1994). Savastingojo kompiuterių tinklo srauto su sunkiomis uodegomis modeliavimui reikia taikyti kitus – sudėtingesnius srauto modelius. 2005 metais L. Carvalho, J. Angeja ir A. Navarro aprašė sudėtingesnę paketų praradimo modelį beveliams tinklams su tinklo srautu turinčiu vaizdo ir garso dedamąsias (Carvalho, Angeja, Navarro 2005). Siekiant subtilesnių modeliavimo rezultatų galima atsižvelgti ir į tinklo komponentų savybes, įtakančias informacinio srauto aptarnavimo rezultatus.

### 2.2.1. Tinklo komponentų modeliavimas

Duomenų paketai perduoti per kompiuterių tinklą į paslaugų serverį pirmiausiai patenka į tinklo plokštę (angl. *NIC Network Interface*). Čia paketo aptarnavimui sugaištamas  $t_{nic}$  laikas. Po to operacinės sistemos (angl. *OS Operating System*) branduolio (angl. *Kernel*) tvarkyklės (angl. *Drivers*) iššifruoja gautus paketus iki tinklo lygmens (angl. *Network Layer*), tam užtrukdamas  $t_s$  laiką. Vėliau paketas perduodamas į vartotojo aptarnavimo zoną, kurioje tęsiamas duomenų šifravimas, čia duomenų paketas iššifruojamas iki taikomojo lygmens ir perduodamas į vartotojo iškvieštą programą/aplikaciją, tam sugaištama  $t_v$  laikas. Laikas sugaištamas apdorojant paketą sistemos branduolyje  $t_s$  nepriklauso nuo paketo ilgio, nes čia analizuojama informacija, naudojama paketo perdavimui tinklu. Vartotojo zonoje nustatomas naudojamas servisas ir, jei tai duomenų paketas, tai jis surenkamas, todėl čia laikas priklauso ir nuo apdorojamo paketo ilgio. Būtina pažymėti, kad  $t_{nic} \leq t_s \leq t_v$  (Lemoine, Pham. Lefèvre 2004, Pu, Bhatia, Consel, Coll 2006). Įvertinant visus užlaikymus, kurie naudojami serveryje apdorojant

vieną paketą, pilnas paketo sugaištas laikas serveryje yra  $t_{server} = t_{nic} + t_s + t_v$  (angl. *Nodal Processing Delay*). Reikėtų pažymėti, kad siekiant tikslaus įvertinimo būtina įvesti serverio našumo įvertį, atsižvelgiantį į serverio procesoriaus spartą, įvairių atminčių tipų spartą ir pan.

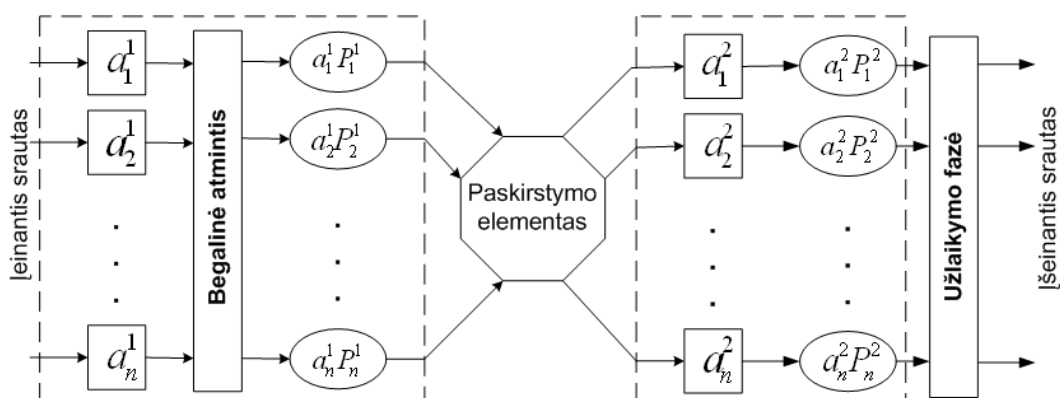
Jei kompiuterių tinklo mazge srauto aptarnavimui naudojamas maršrutizatorius, tai būtina įvertinti jo įtaką perduodamiems duomenų paketams kompiuterių tinklu. Pagal klasikinį tinklo maršrutizavimo uždavinį, paketinio tinklo maršrutizavimą galima aprašyti formule  $\min \sum_{i,j} D_{ij}(F_{ij})$ , kai parinktiems maršrutų srautams tenkinamas apribojimas  $\{x_p \mid w \in W, p \in P_w\}$ , o maršrutai  $p$  tenkina sąlygas:  $\sum_{p \in P_w} x_p = r_w$  visiems  $w \in W$  ir  $x_p \geq 0$  visiems maršrutams  $p \in P_w$ . Čia naudojami tokie žymėjimai:  $W$ – visų įmanomų porų  $w$  „siuntėjas–gavėjas“ aibė;  $P_w$ – siuntėjo ir gavėjo galinius taškus (angl. *Host*) jungiančių porų  $w$  aibė;  $x_p$ – kelio  $p$  srautas;  $r_w$ – sujungimo poros  $w$  reikalavimai tinklo srautui;  $F_{ij}$ – suminis srautas praeinantis  $e_{ij}$  linija, t. y.  $F_{ij} = \sum x_p$  ir sumavimas atliekamas sudedant visus maršrutus  $p$  perduodamus per  $e_{ij}$  linija;  $D_{ij}(F_{ij})$ – svorinė funkcija, susieta su linijos  $e_{ij}$  techninėmis charakteristikomis, tokiomis kaip paketų praradimų koeficientas, trikdžiai, greitimeika (Grimm, Schlüchtermann 2008, Sakalauskas 2000).

Jei kompiuterių tinklo mazge aptarnavimui naudojamas komutatorius su  $n$  kanalų (žr. 4 pav.), naudojančių dvikryptį paketų perdavimą, tai bendruoju atveju galima apskaičiuoti aptarnautų jame paketų kiekį. Sakykime įeinantį srautą formuoja  $n$  kanalų  $a_i$ , kur  $i = 1, 2, \dots, n$ , srauto intensyvumas, patenkantis į komutatorių  $\lambda = \sum_{i=1}^n a_i$ . Jei į tinklo mazgą įeinančių paketų ilgiai yra pasiskirstę pagal eksponentinį dėsnį su parametru  $f$ , tai paketų ilgių pasiskirstymo funkcija užrašoma  $L(x) = 1 - e^{-fx}$  (Sakalauskas 2000). Paketo priėmimo laikas yra proporcingas jo ilgiui, t. y.  $\xi_i^1 = g_i \xi$ ,  $g_i > 0$ , kur  $g_i$ –  $i$ -ojo paketo ilgis, o  $\xi$  paketų priėmimo laikas. Jei komutatoriuje taikomas *FIFO*



(angl. *First Input First Output*) metodas, tai kiekvienas gautas nuo jo aptarnavimo pradžios įrašomas į komutatoriaus atmintį ir aptarnaujamas pagal jo atėjimo tvarką. Ateinančių paketų eilė yra begalinė (Sakalauskas 2000). Tai pirmoji paketų aptarnavimo fazė, kuri apima  $n$  nepriklausomų tiesinių kanalų. Komutatoriuje paskirstymo elementą sudaro centrinis komutatoriaus procesorius vykdamas kanalų komutavimą pagal komutatoriuje įdiegtą algoritmą naudojant komutavimo matricą arba bendrąją magistralę, arba paskirstomą atmintį, arba kitą metodą. Priimti paketai į vieną iš  $m$  perdavimo kanalų  $j$  patenka su tikimybe  $R_j$ , kai  $\sum_{j=1}^m R_j = 1$ . Galima teigti, kad į  $j$ -ąjį kanalą patenka srautas  $\lambda_j = R_j \lambda$ , kur  $j = 1, 2, \dots, m$ . Kaip ir įeinančio srauto atveju paketo  $c_j$  perdavimo  $j$ -uoju kanalu laikas  $\xi_j^1 = c_j \xi$ ,  $c_j > 0$ . Tai antroji paketų aptarnavimo fazė, kuri apima  $m$  nepriklausomų tiesinių kanalų.

Pranešimas, kuris yra perduotas vienu iš išėjimo kanalų paprastai dar yra saugomas operatyviojoje atmintyje (žr. 4 pav. Užlaikymo fazė) tam tikrą laiką  $\tau = a e^{-b\xi}$ , kur  $\xi$  – paketo ilgis, o  $a \geq 0$  ir  $b \geq 0$  ir priklauso nuo konkretaus komutatoriaus parametrų. Ši atmintis reikalinga tam atvejui, jei išsiustas paketas būtų prarastas, tada jį galima greitai pakartoti. Tai trečioji paketų aptarnavimo fazė. Čia patenkančio srauto intensyvumas yra tapatingai lygus pirmųjų dviejų fazių suminiams srautams, t. y.  $\lambda = \sum_{j=1}^m \lambda_j = \sum_{i=1}^n a_i$ .



**4 pav.** Komutatoriaus schema

Jei pirmąją, antrąją ir trečiąją paketų srauto aptarnavimo fazes laikysime stacionariomis ir nepriklausomas, tai suminė paketų esančių komutatoriuje apimtis bet kuriuo laiko momentu gal būti apskaičiuojama įvertinant kiekvienoje fazėje aptarnaujamų paketų apimtį. Pirmoje fazėje aptarnaujamų paketų apimtis, pritaikius Laplaso-Stiltjeso transformaciją, apskaičiuojama pagal formulę:

$$\delta^I(s) = \prod_{i=1}^n \left( 1 - a_i g_i \left[ \frac{1}{f} + \frac{f}{(s+f)^2} \right] \right), \quad \text{jo pirmasis momentas}$$

$$r_1 = \frac{2}{f^2} \sum_{i=1}^n a_i g_i, \quad \text{o dispersija } r_2 - r_1^2 = \frac{2}{f^3} \sum_{i=1}^n a_i g_i \left( 3 - \frac{2a_i g_i}{f} \right).$$

Antroje fazėje aptarnaujamų paketų apimtis, pritaikius Laplaso-Stiltjeso apimties transformaciją, apskaičiuojama pagal formulę:

$$\delta^{II}(s) = (s+f)^{3m} \prod_{j=1}^m \frac{1 - \rho_j}{[(s+f)^2 + \lambda_j c_j s](s+f - \lambda_j c_j)}, \quad \text{jo pirmas}$$

$$\text{momentas } \delta_1^{II} = \frac{1}{f} \sum_{j=1}^m \frac{\rho_j (2 - \rho_j)}{1 - \rho_j}, \quad \text{o dispersija}$$

$$\rho_2^{II} - (\rho_1^{II})^2 = \frac{1}{f} \sum_{j=1}^m \frac{\rho_j (\rho_j^3 + 2\rho_j^2 + 8\rho_j + 6)}{(1 - \rho_j)^2}, \quad \text{kur } \rho_j = \frac{\lambda_j c_j}{f}.$$

Trečioje fazėje aptarnaujamų paketų apimtis, pritaikius Laplaso-Stiltjeso apimties transformaciją, apskaičiuojama pagal formulę:

$$\rho^{III}(s) = \exp \left[ - \frac{\lambda a f s}{(b-f)(s+b+f)} \right], \quad \text{jo pirmas momentas}$$

$$\rho_1^{III} = \frac{\lambda a f}{(b+f)^2}, \quad \text{o dispersija } \rho_2^{III} - (\rho_1^{III})^2 = \frac{2\lambda a f}{(b+f)^2}.$$

Bendra aptarnaujamų paketų apimtis apskaičiuojama  $\rho(s) = \rho^I(s) \cdot \rho^{II}(s) \cdot \rho^{III}(s)$ .

### 2.2.2. Paketinio kompiuterių tinklo srauto modeliai

Išsamesnei kompiuterių tinklo paketinio srauto analizei pasitelkiami matematiniai tinklo srauto modeliai. Pagal klasikinę kompiuterių tinklo srauto modeliavimo teoriją, kompiuterių tinklo paketinį srautą galima aprašyti taikant tris modelius (Grimm, Schlüchtermann 2008):

1. Įeinančio srauto modelis. Čia įeinantis srautas formuojamas iš duomenų paketų ar duomenų vienetų, siunčiamų kompiuterių tinklu. Proceso modeliavimui naudojamas paketo atėjimo laikas  $t_i$  ir laiko tarpai tarp dviejų sėkmingai atėjusių paketų  $\tau_i$ .
2. Aptarnavimo modelis. Čia modeliuojamas paketų aptarnavimas tinklo įrenginyje, o kiekvienas atėjęs paketas fiksuojamas kaip paraiška reikalaujanti aptarnavimo. Proceso modeliavimui naudojamas diskretus reikalavimų aptarnavimo procesui skaičius ir tolydus kiekvieno paketo aptarnavimo laikas.
3. Procesų būsenos modelis. Realizuojant modelį atsižvelgiama į dvi procesų būsenas: 1 – diskretus atėjusių paketų skaičius  $k$  sistemoje, 2 – tolydus paketo buvimo sistemoje laikas  $T_i$ , kuris lygus paraiškos laukimo laiko  $w_i$  ir jos aptarnavimo laiko  $x_i$  sumai.

Pritaikius Kendal žymėjimą (Kendall 1951) kompiuterių tinklo srauto modelis trumpai užrašomas:  $A/B/C/D$ . Čia  $A$  – įeinančio srauto aprašymo standartas,  $B$  – aptarnavimo laiko skirstinį aprašantis standartas,  $C$  – aptarnavimo kanalų skaičius, kartais vadinamas serverių skaičiumi, modeliuojamoje sistemoje ir yra natūralusis skaičius lygus 1 arba didesni už 1,  $D$  – vietų skaičius sistemoje apimantis sistemos buferio talpą ir einamu momentu aptarnaujamas paraiškas, kuris gali būti ir begalinis. Šį žymėjimą galima ir praplėsti iki šešių elementų, tada modelį galima užrašyti  $A/B/C/D/E/F$ . Čia  $E$  – paraiškų populiacijos dydis išreiškiamas natūraliuoju skaičiumi, jei šis argumentas praleistas, tai turima omenyje, kad paraiškų skaičius begalinis.  $F$  – aptarnavimo įrenginio eilės aptarnavimo disciplina, jei

šis argumentas yra praleistas paprastai turima omenyje, kad naudojama *FIFO* disciplina.

*IP* kompiuterių tinklo procesų modeliavimui ir aprašymui taikomi netiesinės analizės metodai, įvertinant neprognozuojamus tinklo srauto pliūpsnius, aprašomus tikimybiniais modeliais su sunkiomis uodegomis (Sheluhin, Smolskiy, Osin, 2007).

Pagal (Wilson 2011), visus paketinio kompiuterių tinklo srauto modelius galima suskirstyti į modelius, kurių taikymas savastingojo tinklo srauto modeliavimui yra komplikuoatas ir modelius, tinkančius savastingojo srauto modeliavimui. Apžvelgsime standartinių modelių naudojimo informacinio srauto modeliavimui ypatumus. Analizuojant mokslinę literatūrą, galima rasti įvairių modelių modifikacijų bei kombinuotų modelių, kurie taikomi specifinėms tinklo srauto modeliavimo reikmėms.

*Puasono* matematinis modelis yra seniausias ir dažniausiai naudojamas telefoninio tinklo paketinio srauto modeliavimo įrankis. Modelis pagrįstas teiginiu, kad duomenų paketų atėjimo laikai į aptarnavimo mazgą yra atsitiktiniai ir nepriklauso nuo kitų paketų atėjimo laikų. *Puasono* procesas yra charakterizuojamas kaip savaimė atsinaujinantis procesas. Čia duomenų paketų atėjimo laikai aprašomi pagal eksponentinį dėsnį, kai srauto intensyvumas  $\lambda$ , formule:  $P\{A_n \leq t\} = 1 - \exp(-\lambda t)$ . *Puasono* modelį galima panaudoti, kai reikia palyginti didelį kiekį nepriklausomų srautų. Be to būtinos dvi pradinės sąlygos: 1 – tinklo paketų siuntėjų skaičius yra begalinis, 2 – srauto aptarnavimas yra atsitiktinis. Modelio įvykių skirstinio funkcija:  $F(t) = 1 - e^{-\lambda t}$ . Tankio funkcija:  $f(t) = \lambda e^{-\lambda t}$ .

Kombinuotas *Puasono* modelis standartinį modelį praplečia tuo, kad atlieka paketų grupių analizę. Grupių atėjimo laikai yra pasiskirstę pagal eksponentinį dėsnį, o grupės dydis ir laikai tarp grupių atėjimo yra atsitiktiniai ir pasiskirstę pagal eksponentinį dėsnį. Taikant šį modelį reikia aprašyti įeinančio srauto intensyvumą  $\lambda$  ir partijos parametą  $\rho \in (0,1)$ . Remiantis

aprašytais parametrais paketų skaičius grupėje apskaičiuojamas  $n = \frac{1}{\rho}$ , o laiko tarpų tarp grupių atėjimo vidurkis  $\bar{\tau} = \frac{1}{\lambda}$ , tada per laiko tarpą  $t$  vidutinis atėjusių paketų skaičius  $\bar{n} = \frac{t\lambda}{\rho}$  (Jain, Routhier 1986).

Kaip ir kombinuotas *Puasono* modelis paketų traukinio arba didžiausio leistino atotrūkio tarp paketų *MAIG* (angl. *Maximum Allowed Inter-Car Gap*) modelis taip pat analizuoja paketų grupių perdavimą. Skirtingai nuo kombinuotojo modelio čia aprašomas grupių maršrutas iš vieno galinio taško A į kitą galinį tašką B, todėl modelis tinka kompiuterių tinklams, kurių mazguose yra tiltas ar kitas analogiškas tinklo mazgo įrenginys. Modelį plačiau aprašė R. Jain ir S. Routhier (1986).

Klasikiniai Markovo ir Markovo grandinių metodai modeliuoja paketinių kompiuterių tinklo srautą, kai jo būsenų skaičius yra ribotas. Modelio tikslumas ir sudėtingumas priklauso nuo aprašomų sistemos būsenų kiekio. Vienas ir pagrindinius Markovo modelio teiginių – tai, kad kiekviena sekanti sistemos būseną priklauso nuo einamosios, t. y. jei  $X_n$  yra einamoji sistemos būseną, tai  $X_{n+1}$  būseną gaunama tik iš  $X_n$ , o ne iš kitos būsenos  $X_i$ , kur  $i < n$ . Atsitiktinių kintamųjų, aprašančių skirtingas būsenas seka formuoja diskrečiąją Markovo grandinę  $\{X_n\}$ . Duomenų paketų atėjimo laikai Markovo modeliuose aprašomi eksponentiškai su skirtingais srauto intensyvumais  $\lambda_i$ . Markovo savaime atsinaujinančiame modelyje kiekviena sekanti sistemos būseną priklauso nuo einamosios ir nuo laikų tarp duomenų paketų atėjimo į sistemą  $\tau_i$ . Plačiau tokių modelių taikymą nagrinėjo A. Lombardo, G. Morabito, ir G. Schembra (1998).

*Markovo* moduluoto *Puasono* proceso *MMPP* (angl. *Markov Modulated Poisson Processes*) modelis skirtas komunikacinių tinklų, kuriais perduodamas garsas analizei (Heffes, Lucantoni 1986). Markovo moduluoto dažnio proceso *MMRP* (angl. *Markov Modulated Rate Processes*) pagrįstas matricos tikrinės

reikšmės nustatymu, kuris sudėtingėja didėjant aprašomų kintamųjų kiekiui (Daigle, Langford 1986). Šie modeliai yra naudojami modeliuoti video paketų perdavimo procesą komunikaciniuose tinkluose (Maglaris, Anastasiou, Sen, Karlsson, Robbins 1988).

Markovo grandinės bandoma taikyti ir modeliuojant savastingąjį tinklo srautą. Vieną iš tokių pritaikymų aprašė (Robert, Le Boudec 1995, Elsayed, Perros 1994).

Autoregresiniai *AR* (angl. *Autoregressive*) tinklo srauto modeliai geriausiai tinka tinklo, kuriame svarbūs perduodamų duomenų paketų prioritetai, t. y. garso perdavimo, vaizdo perdavimo ir analogiškų srautų modeliavimui. Šiame modelyje išeinančio srauto  $y_n$  prognozavimui yra naudojama ankstesnių išeinančių srautų aibė  $\{y_k\}$ , įeinantis srautas  $x_n$  ir ankstesnių įeinančių srautų aibė  $\{x_k\}$ , kur  $k < n$ . *AR*( $p$ ) yra žymimas  $p$ -tosios eilės autoregresinis modelis ir matematiškai aprašomas formule:  $X_t = C + \sum_{i=1}^p R_i X_{t-i} + W_t$ . Čia  $W_t$  – baltasis triukšmas,  $R_i$  – realieji skaičiai, modelio parametrai,  $x_t$  – aprašyti koreliuojantys atsitiktiniai skaičiai,  $C$  – konstanta (Abdelnaser 1997).

Autoregresiniai slenkančio vidurkio modeliai *ARMA* (angl. *Autoregressive Moving Average*), dar vadinami Box-Jenkins modeliais, išeinančio srauto prognozavimui naudoja ne tik įeinantį srautą, bet ir anksčiau išėjusio srauto parametrus. Taikant modelį analizuojamos dvi jo dalys – tai autoregresinė dalis *AR* ir slenkančio vidurkio dalis *MA*. Šį modelį apibūdina du kintamieji *ARMA*( $p, q$ ), čia  $p$  aprašo  $p$ -tosios eilės autoregresinį modelį *AR*( $p$ ), o  $q$  –  $q$ -osios eilės slenkančio vidurkio modelį *MA*( $q$ ). Jis išreiškiamas formule  $X_t = \varepsilon + W_t + \sum_{i=1}^p S_i W_{t-i}$ . Čia  $W_t$  – baltasis klaidų triukšmas, nepriklausomų tolygiai pasiskirsčiusių kintamųjų aibė,  $\varepsilon - x_t$  išimtis, kurios reikšmė dažnai lygi 0,  $s_i$  – realieji skaičiai, modelio parametrai.

Autoregresinio integruoto slenkančių vidurkių metodo *ARIMA* (angl. *Autoregressive Integrated Moving Average*) esmė ta, kad jis apjungia

autoregresijos, diferencijavimo ir slenkamųjų vidurkių metodo galimybes ir naudojamas tiesinių laiko eilučių modeliavimui. Ši modelį apibūdina trys kintamieji  $ARIMA(p,d,q)$ , čia  $p$  aprašo  $p$ -tosios eilės autoregresinį modelį  $AR(p)$ ,  $q$  –  $q$ -osios eilės slenkančio vidurkio modelį  $MA(q)$ , o  $d$  – diferencijavimo eilė, nusakanti kiek kartų reikia diferencijuoti eilutę, kad ją suvesti į stacionarų pavidalą (Li 2002).

*ON-OFF* tinklo modelis dažniau naudojamas, kai analizuojama kompiuterių tinklo paketinio srauto struktūra, esant dideliems tinklo srauto svyravimams. Naudojant modelį galima aprašyti tinklo srautą nuo fizinio lygmens iki taikomojo *OSI* lygmens. Šiame modelyje tinklo srautas gali turėti dvi stacionarias būsenas *ON*, kai duomenų paketai perduodami ir *OFF* – kai paketų nėra. Formuojant tinklo paketų modelį būtina atsižvelgti, kad paketų perdavimo šaltiniai yra statistiškai identiški ir nepriklausomi. Siunčiamos paraiškos aptarnaujamos pastoviu greičiu  $C$ , kai eilės ilgis yra baigtinis ir lygus  $M$ . *ON-OFF* paketų šaltinis charakterizuojamas  $L$  – tai vidutinis paketų kiekis persiustas esant *ON* sistemos būsenai, pliūpsniu metu perduodamas  $S$  paketų kiekis, o šaltinio paketų generavimo dažnis yra  $r$ . Šie faktoriai lemia *ON-OFF* periodų vidutinės trukmės. *ON* būsenos pusiausvyros tikimybė apskaičiuojama:  $\gamma = r/S$ . *ON-OFF* periodai yra eksponentiškai pasiskirstę, o paketų šaltinis gali būti modeliuojamas kaip dviejų būsenų Markovo grandinė. Vidutinis paketų generavimo dažnis yra griežtai didesnis už 1. *ON* ir *OFF* būsenų persijungimas apskaičiuojamas taip: iš *OFF* į *ON*  $\frac{\gamma S}{L(1-\gamma)}$ ; iš *ON* į

*OFF*  $\frac{S}{L}$  (Abdelnaser 1997, Žvironienė, Navickas, Rindzevičius 2006).

Remiantis *ON-OFF* modeliu buvo sukurtas nutraukto *Puasono* proceso *IPP* (angl. *Interrupted Poisson Process*) modelis, kuris tinklo kanalą aprašo kaip sistemą su dviem skirtingomis būsenomis: 1 – kai jis apkrautas ir veikia, 2 – kai tinklo kanale nėra paraiškų ir jis laisvas ir neveiksnius (Dombry, Kaj 2011).

Pliūpsniškiems srautams modeliuoti dažnai taikomi  $\alpha$ -stabilieji modeliai. Jie paklūsta apibendrintai centrinei ribinei teoremai ir gerai aprašo sunkias uodegas, asimetriją (Gallardo, Makrakis, Orozco-Barbosa 2000, Janicki, Weron 2000, Yin, Lin, Sebastien, Li, Min 2005, Kaklauskas, Sakalauskas 2008, Surgailis, Teyssiere, Vaičiulis 2008, Kulikovs, Petersons 2008, Grigelionis 2008, Kabašinskas, Rachev, Sakalauskas, Sun, Belov 2009, Lavancier, Philippe, Surgailis 2010). Pagal Samorodnitsky (2006b) savastingasis simetriškas procesas pasižymintis begaline dispersija, yra  $\alpha$ -stabilusis procesas, kai  $0 < \alpha < 2$  (Samorodnitsky, Taqqu 1994), jei kiekvienam atsitiktiniam procesui  $Y(t)$  galima aprašyti sunkias uodegas pagal formulę  $P(|Y(t)| > x) \sim cx^{-\alpha}$ , kai  $x \rightarrow \infty$ , o  $c > 0$ . Vertinant bet kokį stabilųjį atsitiktinį skirstinį  $S_\alpha(\beta, \sigma, \mu)$  rekomenduojama įvertinti keturis stabilumo parametrus:

- $\alpha$  – stabilumo parametras  $\alpha \in (0, 2]$ , dar vadinamas uodegų parametru, nusakantis proceso pliūpsniškumą,
- $\beta$  – asimetrijos parametras  $\beta \in [-1, 1]$ , nusakantis proceso poslinkį atžvilgiu nulio,
- $\sigma$  – mastelio parametras,  $\sigma > 0$  ir nusako proceso elementų dydį,
- $\mu$  – padėties parametras  $\mu \in R$ .

Prenkant stabilumo parametro reikšmę  $\alpha \in (0, 1]$  savastingasis procesas pasižymi greitai mažėjančia priklausomybe *SRD*, o kai  $\alpha \in (1, 2]$ , tai procesas pasižymi lėtai mažėjančia priklausomybe *LRD*.

### 2.2.3. Paketinio tinklo srauto aptarnavimo modeliai

(Manjunath, Sikdar 2002, Gyires 2003) teigė, kad taikant savastinguosius tinklo srauto modelius galima parinkti geriausias tinklo mazgo savybes, kad būtų optimaliai aptarnaujamas informacinis duomenų srautas. Reikia pažymėti, kad tinklo mazge labai svarbų vaidmenį vaidina buferis, saugantis paketus eilėje iki jų aptarnavimo. Tinkle, neaptarnaujančiame prioritetinių duomenų paketų (nenaudojamos vaizdo ar garso transliacijos bei kitos realaus laiko



paslaugos), eilių aptarnavimui naudojamos *FIFO* ir *LIFO* eilių disciplinos. Jų įtaką tinklo srauto aptarnavimui analizavo (Kesselman, Lotkeryx, Mansour, Patt-Shamiryx, Schieberz, Sviridenkoz 2001, Huang, Chang, Cheng, Lee 2007). 2009 metais Nzouonta, Ott ir Borcea (Nzouonta, Ott, Borcea 2009) analizuodami *Ad Hoc* tinklų *UDP* srautą nustatė, kad *LIFO* ir *FIFO* aptarnavimo disciplinos turi vienodus vėlinimus, esant mažam srautui, o kai srautas didelis, tai *LIFO* Frontdrop pasižymėjo daugiau nei dvigubai mažesniu vėlinimu nei *FIFO* Taildrop. Informacinio srauto aptarnavimo našumui nemažą įtaką daro mazgo eilės ilgis. Šią problemą nagrinėjo Appenzeller, Keslassy, Enachescu, Ganjali ir daugelis kitų mokslininkų (Appenzeller, Keslassy, McKeown 2004; Enachescu, Ganjali, Goel, McKeown, Roughgarden 2005, Ramanan, Cao 2002, Kulikovs, Petersons 2008). Tyrimais buvo nustatyta, kad mazguose aptarnaujančiuose 30 Gb/s ir didesnės pralaidos kompiuterių tinklus pilnai pakanka 15-20 paketų talpos buferio (Nzouonta, Ott, Borcea 2009). (He, Gao, Hou, Park 2004, Chung, Seol, Choi, Yeoum, Lim 2008) paketinio kompiuterių tinklo savastingumo savybę sėkmingai panaudojo tinklo paketų transporto srauto perkrovų reguliavimui.

Aptarnavimo teoriją sukūrė A. K. Erlong (Erlong 1909). Analizuodamas telefonų tinklą Erlong siekė parengti ryšių kanalų skaičiaus nustatymo metodą, leidžiantį aptarnauti visus prie tinklo besijungiančius telefono abonentus, minimizuojant iškvietimų praradimus. Aptarnavimo teorijos pagrindus toliau plėtojo A. Chinčin (Хинчин 1955), T. Engset (Engset 1998), D. Kendall, A. Kolmogorov (Хинчин 1963) ir daugelis kitų mokslininkų. (Boxma, Cohen 1999) pateikė aptarnavimo teorijos priemonių panaudojimo aspektus aprašant tinklo objektų sąveiką. Danų mokslininkas Iversen aptarnavimo teoriją pavadino telesrauto (angl. *Teletraffic*) teorija ir jos taikymus aprašė knygoje (Iversen 2011). Lietuvoje aptarnavimo teoriją vysto Rindzevičius, Sakalauskas ir kiti mokslininkai. Aptarnavimo teorija integravo nemažai ryšių technologijoms būdingų sąvokų: įėjimo srauto intensyvumas –  $\lambda$ , aptarnavimo srauto intensyvumas –  $\mu$ , aptarnavimo trukmė –  $x$ , apkrovos koeficientas –  $\rho$ , kanalo apkrovos koeficientas –  $q$ , aptarnavimo kanalų kiekis –  $m$ , vidutinė

vienos paraiškos aptarnavimo trukmė –  $\bar{X}$ , vidutinis paraiškų skaičius sistemoje –  $\bar{N}$  ir pan. (Sakalauskas 2000).

Remiantis aptarnavimo teorija, aprašant kompiuterių tinkle modeliuojamus informacinius srautus, naudojami specializuoti žymėjimai, o tinklo duomenų paketai vadinami paraiškomis (Saaty 1957). Aptarnaujamo srauto intensyvumas yra matuojamas erlangais (teoriją sukūrusio mokslininko garbei). 1 erlangas, lygus vienam įvykiui per valandą, paprastai žymimas  $E$  arba  $E_h$ . Vidutinis įvykių skaičius per sekundę arba milisekundę žymimi atitinkamai  $E_s$  arba  $E_{\mu s}$ . Trumpai aptarsime kitus žymėjimus. Įvykių srautas vadinamas determinuotu, jei įvykiai eina vienas po kito griežtai nustatytais, iš anksto žinomais laiko momentais. Determinuotas srautas yra reguliarusis, jei įvykiai seka vienas kitą vienodais laiko tarpais. Reguliarusis determinuotas srautas žymimas simboliu  $D$ . Paketinio kompiuterių tinklo srautai dažniausiai yra atsitiktiniai su neapibrėžtu paketų pasirodymo laiku. Tokie srautai aprašomi tikimybiniais skirstiniais. Tinklo srautą, kurio paketų atėjimo momentai  $t_n$  bet kaip pasiskirstę žymima simboliu  $G$  (angl. *General*). Rekurentiniame sraute  $GI$  (angl. *General Independent*) trukmės tarp paketų atėjimų yra nepriklausomai vienodai pasiskirstę. Šiuo atveju geriau nagrinėti ne paketų atėjimo pradžios laikus, bet trukmes tarp dviejų gretimų paketų pasirodymų:  $\tau_n = t_n - t_{n-1}$ . Tada  $n$ -ojo paketo pasirodymo momentas nusakomas nepriklausomų vienodai pasiskirsčiusių dydžių suma  $t_n = \sum_{i=1}^n \tau_i$ . Rekurentinių įvykių srautą apibūdina atsitiktinio dydžio  $\tau_n$  skirstinys:  $F(t) = P(\tau_n \leq t)$ . Jei  $f(t)$  yra skirstinio tankio funkcija vidutinę trukmę  $\bar{\tau}$  tarp dviejų viena po kitos einančių paraiškų galima apskaičiuoti  $\bar{\tau} = E\tau = \int_0^{\infty} u \cdot f(u) du$ . Įvykių srautas vadinamas stacionariuoju, jei jų skaičiaus bei kitų srauto laiko parametrų pasiskirstymas bet kuriuo laiko intervalu priklauso tik nuo to intervalo trukmės, bet nepriklauso nuo intervalo vietos laiko ašyje. Šiuo atveju stacionariojo srauto intensyvumas  $\lambda = \frac{1}{\bar{\tau}}$ . Atsitiktinis paketų srautas vadinamas ordinariuoju, jei tikimybė, jog labai

trumpam laiko tarpui teks du ar daugiau paketų, mažėja sparčiau, trumpėjant intervalui, negu šio tarpo trukmė. Tai reiškia, kad paketai į tinklo aptarnavimo sistemą patenka po vieną, o ne poromis ar trejetais ir pan. Paketų srautas vadinamas be sąveikos, jei bet kokiems dviem nesusikertantiems laiko intervalams įvykių skaičius, patenkantis į vieną iš jų, nepriklauso nuo to, kiek įvykių patenka į kitą. Srautas, kuris yra stacionarusis, ordinarusis ir be sąveikos, vadinamas paprastuoju srautu. Tinklo srauto modeliavimui dažniausiai naudojamas paprastas srautas, kuris žymimas  $\mathbf{M}$ . Paprastojo srauto įvykių skirstinys užrašomas:  $F(t) = 1 - e^{-\lambda t}$ . Atsitiktinio dydžio  $\tau$  tikėtiną reikšmę:  $\bar{\tau} = \frac{1}{\lambda}$ , dispersija:  $D^2\tau = \frac{1}{\lambda^2}$ . Paprastojo srauto įvykių, patenkančių į tam tikro ilgio intervalą, skaičius yra pasiskirstęs pagal *Puasono* arba eksponentinį dėsnį. Reguliariai išretintas paprastasis srautas yra *Erlango* srautas ir žymimas simboliu  $\mathbf{E}_r$ . HiperekspONENTINIS srautas žymimas  $\mathbf{H}_r$  ir aprašomas kelių eksponentinių funkcijų suma:  $F(t) = 1 - \sum_{j=1}^r q_j \cdot e^{-\lambda_j t}$ ,  $t \geq 0$ ,  $r \geq 1$ , o koeficientai tenkina sąlygą:  $0 \leq \sum_{j=1}^r q_j \leq 1$ . HiperekspONENTINIŲ skirstiniu modeliuojamos operacijos, kurių trukmė gali įgyti labai mažas arba labai dideles reikšmes. HipoekspONENTINIO skirstinio variacijos koeficientas visada neviršija vieneto.

Jei modeliuojamos sistemos paraiškų skaičius nėra ribojamas, tai remiantis britų matematiko D. Kendall (1951) pasiūlytais žymėjimais tokią sistemą galime pažymėti  $M/M/m$ . Ją sudaro paprastasis srautas, viena aptarnaujanti linija su neriboto ilgio eile, o paketus aptarnauja  $m$  įrenginių. Kadangi į sistemą patenka įvairaus ilgio paketai įvairiais laiko intervalais, tai įėjimo srautas gali būti aprašomas pritaikant eksponentinius pasiskirstymo dėsnius. Analizuojamo tinklo srauto paketų skaičius bei kitų parametų pasiskirstymas bet kuriuo laiko intervalu priklauso tik nuo to intervalo trukmės, bet nepriklauso nuo intervalo vietos laiko ašyje, vadinasi galima sakyti, kad tam tikrais laiko intervalais analizuojamam tinklo srautui galioja stacionarumo savybė. Į aptarnavimo

sistemos tinklą paraiškos patenka po vieną iš maršrutizatoriaus, t. y. tenkinama ordinarumo savybė. Be to srautas tenkina nepriklausomumo savybę, nes į tinklą patenkančių paketų skaičius vienu laiko intervalu visiškai nepriklauso nuo jų kiekio patenkančio kitu laiko intervalu. Remiantis aukščiau įvardintais teiginiais galima sakyti, kad nagrinėjamas tinklo srautas yra paprastas ir jo trukmę tarp paketų pasirodymų aprašantis skirstinys yra eksponentinis  $F(t) = 1 - e^{-\lambda t}$  su tankio funkcija  $f(t) = \lambda e^{-\lambda t}$ . Trukmės tarp įvykių  $\tau$  tikėtina reikšmė (vidurkis) apskaičiuojama –  $\bar{\tau} = \frac{1}{\lambda}$ , o srauto dispersija –  $D^2\tau = \frac{1}{\lambda^2}$ . Toks kompiuterių tinklo paketų srautas bendroju atveju yra pasiskirstęs pagal *Puasono* dėsnį:

$$P(N_{t,\tau} = k) = \frac{(Q_{t,\tau})^k}{k!} \cdot e^{-Q_{t,\tau}},$$

kur  $N_{t,\tau}$  yra paketų skaičius gaunamas į paketų aptarnavimo sistemą per laiko intervalą  $(t, t + \tau)$ ,  $Q_{t,\tau}$  yra vidutinis srauto intensyvumas laiko intervale  $(t, t + \tau)$  ir apskaičiuojamas  $Q_{t,\tau} = \int_t^{t+\tau} \lambda(u) du$ , čia  $\lambda(t)$  yra srauto intensyvumas laiko momentu  $t$  ir  $\lambda(t) = \lim_{\tau \rightarrow 0} \frac{Q_{t,\tau}}{\tau}$ . Tinklo srautas aprašytas pagal *Puasono* dėsnį pasižymi ribinės superpozicijos bei atsitiktinio išretinimo savybėmis (Sakalauskas 2000).

Jei neatsižvelgsime į tai, kad tinklo duomenų srautas pasižymi fraktališkumu, tai M/M/m sistemos darbą nusakančias charakteristikas galima apskaičiuoti remiantis klasikinėmis formulėmis. Vidutinis paketo buvimo sistemoje laikas  $\bar{T}_s$  randamas pagal formulę:

$$\bar{T}_s = \frac{\bar{N}_s}{\lambda} = \frac{1}{\mu} + P_s \cdot \frac{1}{m\mu - \lambda},$$

čia  $\mu$  – aptarnaujamo srauto intensyvumas,  $\lambda$  – įėjimo srauto intensyvumas  $P_s$  – tikimybė, kad paketo persiuntimo į sistemą metu visi

aptarnavimo įrenginiai bus užimti. Skaičiuojant vidutinį paketo buvimo laiką sistemoje būtina įvertinti paketų ilgių pasiskirstymą aptarnaujamoje sistemoje. Dažniausiai kompiuterių tinkle paketų ilgiai yra pasiskirstę pagal eksponentinį dėsnį, todėl skaičiuojamas jų vidutinis ilgis (Plėštys, Kavaliūnas, Vilutis 2008, Rindzevičius, Tervydis 2003).

Vidutinis paketų esančių sistemoje kiekis  $\bar{N}$ , kai sistemos apkrova  $\rho = \frac{\lambda}{\mu} = \lambda T_{pak}$  apskaičiuojamas pagal formulę:

$$\bar{N} = \sum_{k=0}^{\infty} k p_k = m\rho + \rho \frac{(m\rho)^m}{m!} \frac{P_s}{(1-\rho)^2}$$

Tikimybė, kad visi aptarnavimo įrenginiai bus užimti paketo atsiuntimo metu randama pagal formulę:

$$P_s = \frac{\frac{\rho^m}{m!}}{\left[ \frac{\rho^m}{m!} + (1-\rho) \sum_{k=0}^{m-1} \frac{\rho^k}{k!} \right]}$$

Vidutinis paketų skaičius eilėje randamas:

$$\bar{N}_Q = P_s \frac{\rho}{1-\rho}$$

Vidutinis paketų laukimo laikas eilėje apskaičiuojamas:

$$\bar{W} = P_s \cdot \frac{1}{m\mu - \lambda}$$

### 2.3. Skyriaus išvados

Empiriniai tyrimai rodo, kad šiuolaikiniai kompiuterių tinklai pasižymi pliūpsniškumu bei savastingumu, todėl klasikiniai *Puasono* ir *Markovo* paketinio tinklo srauto modeliai jų analizei netinka, o autoregresiniai tinklo srauto modeliai labiau taikytini garso, vaizdo ir kitų realaus laiko tinklo srautų modeliavimui.

Savastingojo paketinio kompiuterių tinklo srauto su sunkiomis uodegomis modeliavimui dažniausiai naudojami *ON-OFF* bei kombinuoti modeliai, pritaikant klasikinių *Markovo*, *Markovo* grandinių, *Puasono*, *MMPP*, *MMRP* ir kitų modelių savybes.  $\alpha$ -stabilieji modeliai ir aptarnavimo teorijos įrankiai mokslinėje literatūroje plačiau analizuojami teoriškai, žinomi šių modelių taikymai finansinių procesų modeliavimui, tuo tarpu praktinių jų taikymų savastingajam tinklo srautui su sunkiomis uodegomis yra mažai.

### **3 skyrius. Fraktalinių srautų kompiuterių tinkluose matavimo technologijos ir metodai**

Kompiuterių tinkle vykstančių procesų aprašymui dažnai naudojama netiesinių dinaminų sistemų teorija (Shang, Wan, Kama 2007, Xia, Liu, Iu 2009, Kaklauskas, Sakalauskas 2009), kuri dar vadinama chaoso teorija, turinti du instrumentus – atraktorius ir fraktalus. Atraktorius – tai geometrinė struktūra, nusakanti nagrinėjamos sistemos elgseną fazinėje erdvėje, kai laikas artėja į begalybę. Fazinė erdvė nusako analizuojamos sistemos laisvės laipsnius. 1963 metais E. Lorenz aprašė pirmąjį keistąjį atraktorių (Lorenz 1963). Vėliau tokius atraktorius savo darbuose aprašė (Ruelle, Takens 1971). Keistojo atraktoriaus ribos apibūdinamas atraktoriaus dimensija, o jo geometrinė išraiška – fraktalas. Keistojo atraktoriaus įvertinimui panaudojamos jo savybės: simetriškumas – nusakantis proceso savastingumą ir fraktalinis matas – matematinis fraktalo netikslumo įvertis (McBride 2005, Kaklauskas, Sakalauskas 2008).

Kompiuterių tinklo paketinio srauto fraktalinių savybių analizei pirmiausiai reikia surinkti informaciją apie srauto savybes. Tam gali būti taikomos įvairios strategijos, kurios priklauso nuo tyrime siekiamų tikslų bei numatomų rezultatų. Galima išskirti kelias skirtingas tyrimų strategijas:

1. Įvykių skaičiavimu grįsta strategija. Čia tyrėjas skaičiuoja tyrimui reikalingus įvykius, kurie įvyko per tam tikrą laiko tarpą ir juos išsaugo. Pavyzdžiui, skaičiuoja atėjusius paketus, prarastus paketus ir pan.
2. Įvykių sekimo strategija. Pagal šią strategiją yra įrašoma daugiau informacijos, nei reikia tyrimui atlikti, todėl ją taikant reikia išskirti daugiau atminties informacijos saugojimui.
3. Netiesioginis matavimas. Ši strategija naudojama tada, kai reikiama savybė negali būti pamatuojama tiesiogiai, tada tyrėjas pasirenka

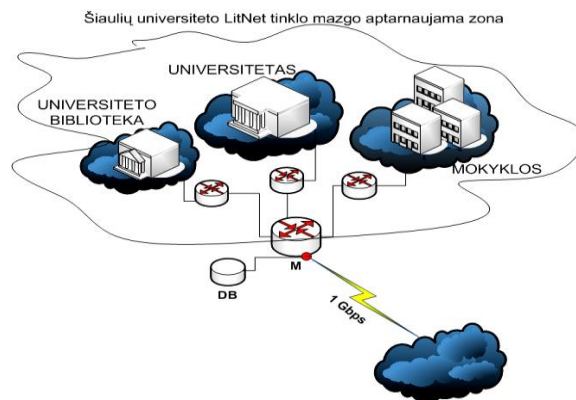
parametrus, kurie gali būti pamatuojami tiesiogiai ir iš jų gaunamas tyrimui reikalingas įvertis.

4. Epizodinis matavimas. Pagal šią strategiją yra įrašoma sistemos būseną, kad nustatyti kokie ir kokio dažnumo matavimai reikalingi.

Vykdamat matavimus dažniausiai fiksuojami tik numatytieji kompiuterių tinklo įvykiai, tačiau gali būti vertinamas bendras kompiuterių sistemose profilis, siekiant nustatyti sistemos elgseną arba fiksuojami papildomi sistemos parametrai ar įvykiai ir vertinama jų įtaka tiriamam objektui (Jain 1991, Obaidat, Boudriga 2010). Surinkta informacija analizuojama skaičiuojant jos matematinius įverčius bei apibendrinant analizės rezultatus.

### **3.1. Technologiniai tinklo srauto matavimo sprendimai**

Fraktališkumo įvertinimui reikia surinkti kompiuterių tinklo srauto paketo antraščių duomenis, kuriuos galima tiesiogiai fiksuoti kompiuterių tinklo mazge arba kliento kompiuteryje. Vadinasi, tyrimui reikalingos informacijos rinkimui reikės taikyti įvykių skaičiavimu grįstą strategiją (Boudriga 2010).



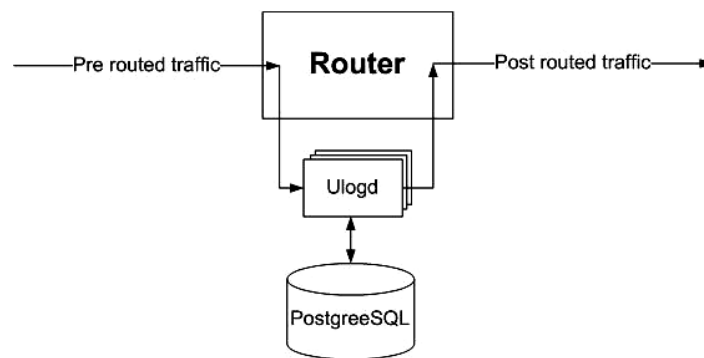
**5 pav.** Analizuojamo tinklo mazgo struktūrinė schema

Paketinio kompiuterių tinklo srautų matavimui pasirinktas Šiaulių universiteto LitNet tinklo mazgas, pasižymintis didžiausia apkrova ir srauto intensyvumu tinkle. Čia iš tarpmiestinio kanalo gaunamas 1 Gbps srautas, kuris paskirstomas universitetui bei Šiaulių regiono švietimo įstaigoms (žr. 5 pav.). Analizuoti tik į mazgą M ateinantys duomenų paketai, atmetant išsiunčiamus paketus. Gauta informacija kaupiama išorinėje duomenų bazėje Porstgree SQL



(brėžinyje pažymėta DB). Pirminiai matavimai atliekami vienos mikrosekundės tikslumu. Įrašas į duomenų bazę formuojamas iš karto kai gaunamas *TCP* arba kito protokolo duomenų paketas.

Tinklo srautų analizei darbe naudojami programiniai ir aparatiniai tinklo stebėjimo įrankiai, registruojantys duomenų paketus nurodytu laiko intervalu. Matavimams panaudota pagal *GPL* licencija platinama programa *ulogd Linux* operacinei sistemai (Ulogd 2012). Duomenys registruojami taikant skirtingus laiko diskretizavimo lygius, esant skirtingoms tinklo apkrovos lygiams bei suformuojant agreguotas laiko eilutes. Matavimo rezultatai apdorojami, skaičiuojant eilutės koreliacinį matą, analizuojant atraktorius, įvertinat fraktalinį matą, apskaičiuojant *Hurst*'o koeficientą bei įvertinant analizės rezultatų patikimumą statistiniu būdu (Leland, Wilson 1991).



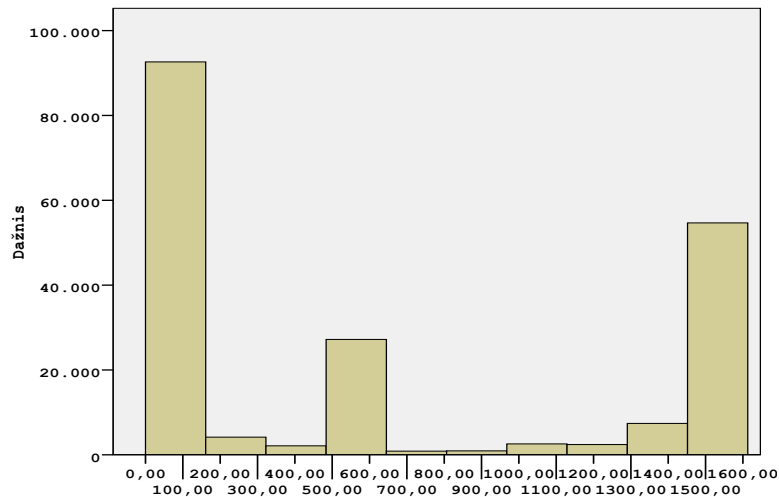
**6 pav.** Tinklo srauto registravimo schema

Matavimams panaudota pagal *GPL* licencija platinama programa *ulogd Linux* operacinei sistemai. *ulogd* – tai programa Linux operacinei sistemai, panaudodama *netfilter/iptables* lenteles įrašo informaciją apie paketus, kurie ateina į tinklo mazgą (Ulogd 2012, Kaklauskas 2007). Ji fiksuoja paketų antraščių informaciją pagal nurodytus paketų atrinkimo kriterijus. Maršrutizatoriuje *M* fiksuojami paketinio įeinančio duomenų srauto informacija. Ji talpinama į atviro kodo *PostgreSQL* duomenų bazę. (žr. 6 pav.).

Duomenų bazėje duomenys saugomi tol kol jie bus išanalizuoti. Reikia pažymėti, kad matuojamas duomenų srautas yra labai didelis, todėl vidutiniškai per valandą buvo sukaupiama iki 1500000 įrašų apie į mazgą *M* ateinančius

paketus. Iš duomenų bazės analizei pasirinkti duomenys nuo 2008-01-04 13:30:35 iki 2008-04-16 12:00:00. Per šį laikotarpį duomenų bazėje sukaupta daugiau kaip trys milijardai įrašų, kurie atitiko 8936965 sek. arba 103 paros 10 val. 29 min. ir 25 s.

Fiksuojamų paketų ilgis kinta nuo 28 baitų iki 1554 baitų. 7 paveiksle pavaizduota charakteringa paketų ilgių pasiskirstymo histograma. Apie 20,8% registruojamos informacijos sudaro paketai, kurių ilgis 1500 baitų – tai duomenų nešėjai, apie 23,4% sudaro 40 baitų paketai, tai tarnybinė informacija. Ilgesni nei 1400 baitų paketai šiame sraute sudaro apie 30,7% visų analizuotų paketų, o trumpesni nei 100 baitų – apie 45,8%. Žinoma, kad srauto sudėtis priklauso nuo kompiuterių tinklu perduodamos informacijos tipo.



**7 pav.** Registruojamų paketų ilgių pasiskirstymas

Tinklo srauto intensyvumo analizė rodo, kad didžiausias duomenų srautas mazge yra darbo dienomis, mažiausias – sekmadienį, o šeštadienis užima tarpinę poziciją. Tiriant apkrovos kitimą paros metu, kiekvieną savaitės parą nustatomos valandos, kai duomenų srautas yra didžiausias, vidutinis ir mažiausias. Pasirinktų laiko intervalų duomenų srauto intensyvumo analizei pritaikytas klasterinės analizės k-vidurkių metodas, naudojant programų statistinės analizės ir duomenų apdorojimo paketą SPSS. Buvo taikomas nehierarchinės klasterinės analizės metodas, kai žinomas klasterių skaičius ( $k=24$ , t. y. valandų kiekis paroje), o atstumai tarp klasterių ir objektų

apskaičiuojami pasinaudojant Euklido atstumo kvadrato metrika. Suformuotos laiko eilutės, atitinkančios vienos valandos matavimą, kai perduodamų paketų srautas tinkle didžiausias, vidutinis ir mažiausias. Analizei atrinktos 309 vienos valandos ilgio laiko sekos.

### ***3.2. Tinklo srauto parametrų vertinimas taikant chaoso teoriją***

Chaoso teorija pagrįsta dviem teiginiais: ateities tiksliai nusakyti neįmanoma dėl matavimo paklaidų bei visų pradinių sąlygų nežinojimo; parengtų prognozių patikimumas bėgant laikui sparčiai mažėja. Pagrindiniai šios teorijos instrumentai – atraktoriai ir fraktalai, kuriais siekiama aprašyti dinamines sistemas, nepaklūstančias klasikinėms teorijoms.

Atraktorius (angl. Attract – pritraukti) – tai geometrinė struktūra, nusakanti sistemos elgseną fazinėje erdvėje, kai laikas artėja į begalybę. Fazinė erdvė – tai abstrakti erdvė, kurios koordinatės nusako analizuojamos sistemos laisvės laipsnius. Analizuodamas kompiuteriu trim laisvės laipsniais nusakomą sistemą, sudarytą iš trijų paprastų diferencialinių lygčių su trimis konstantomis ir trimis pradinėmis sąlygomis sistemą E. Lorenz aprašė pirmąjį chaotišką arba keistąjį atraktorių (Lorenz 1963, Schnabl, Stadler, Forst, Schuster 1991). Vėliau savo darbuose aprašė D. P. Ruelle ir F. Takens (1971). Keistasis atraktorius turi tam tikras maksimalias ribas apibūdinamas atraktoriaus dimensija, o geometrinė jo išraiška yra fraktalas. Tada keistojo atraktoriaus įvertinimui galima taikyti šias fraktalo savybes: simetriškumas, kuris nusako proceso savastingumą bei fraktalinis matas išreiškiamas trupmena ir yra matematinis fraktalo netikslumo įvertis (McBride 2005, Kaklauskas, Sakalauskas 2008). Panaudojant atraktorius bei fraktalus galima aprašyti dinamines sistemas, nepaklūstančias klasikinėms teorijoms (Shang, Wan, Kama 2007, Xia, Liu, Iu 2009, Kaklauskas, Sakalauskas 2009). Dažnai šiuolaikinių kompiuterių tinklo srautą bandoma aprašyti ir analizuoti kaip multifraktalinį procesą (Chen, Cai, Li 1997, Gao, Rubin 2001a, Gao, Rubin 2001b, Lu, Ke, Huijing 2002, Liangxiu, Zhiwei, Chunbo, Chuanshan 2002, Veitch, Hohn, Abry 2005, Shang, Lu, Kamae 2008). Kompiuterių tinkle

vykstančių procesų aprašymui naudojama netiesinių dinaminų sistemų teorija dar vadinama chaoso teorija. Ji padeda paaiškinti tinklo srauto pokyčius reguliuojančius parametrus bei valdymo galimybes. Tinklo matavimų įvertinimui yra skaičiuojamai du kiekybiniai laiko eilutės parametrai: atraktoriaus dimensija ir jos jautrumas pradinėms sąlygoms (Zhang, Wu, Yang 2004).

Atraktoriaus dimensijos įvertinimui skaičiuojamas Hausfordo matas (Theiler 1990, Zhu, Fu 2008), gautas analizuojant keistą Lorencio atraktorių. Buvo nustatyta, kad sritis, kurią dengia šis atraktorius pasižymi nelygumais, kurie vertinami fraktaliniu matu (angl. *Fractal Dimension*). Fraktalinio mato panaudojimą, matuojant Anglijos pakrantes 1967 metais aprašė B. Mandelbrot (Mandelbrot 1967). Fraktalinio mato įvertinimui skaičiuojamas Hausdorfo matas  $D$ :

$$D = \lim_{\varepsilon \rightarrow 0} \frac{\ln N(\varepsilon)}{\ln(1/\varepsilon)},$$

čia  $N$  minimalus  $n$ -mačių  $\varepsilon$  briaunos ilgio kubelių kiekis, dengiančių aibės taškus, kai briaunos ilgis artėja prie nulio. Šį matą dar galima vadinti fraktaliniu matu. Mūsų nagrinėjamu atveju  $1 < D < 2$ , tada formulę galima supaprastinti:

$$D = \frac{\ln N}{\ln(1/(2 * r))},$$

čia  $N$  – elementų kiekis, panaudotas matuojant fraktalo banguotumą,  $N \rightarrow 2$  kai fraktalas yra plokštumoje,  $r$  – apskritimo spindulys naudojamas dvimatėje erdvėje. Kompiuterių tinklų srauto fraktalinis matas apibūdina gautos duomenų srauto laiko eilutės pokyčių dinamiką ir  $D \in [1,2]$ , kai naudojamas vienas kintamasis. Sekantis sveikas skaičius po fraktalinio mato  $D$  nusako apatinę modeliuojamos pokyčių dinamikos sistemos laisvės laipsnių ribą – minimalų reikalingų kintamųjų kiekį (Chakraborty, Ashir, Sukanuma, Mansfield Keeni, Roy, Shiratori 2004, Zhu, Fu 2008).

Grassberger ir Procaccia (1983) pasiūlė fraktalinį matą įvertinti vertinant koreliacinį matą, panaudojant koreliacinį integralą, kuris nusako tikimybę, kad du atraktoriaus taškai yra nutolę vienas nuo kito per atstumą  $R$ .

Sakykime  $x_t$  yra proceso  $X$  agreguota eilutė su tolygiai fiksuotais laiko intervalais  $\Delta t_n$ , kur  $i \in [1, n]$ . Sakoma, kad procesas  $X$  pasižymi  $LRD$ , jei tenkinama lygybė:  $r(k) \approx k^\beta \cdot L_i(k), k \rightarrow \infty$ . Čia  $r(k)$  yra autokoreliacinė funkcija apskaičiuojama pagal klasikinę formulę:

$$r(k) = \frac{1}{N - \tau} \cdot \frac{\sum_i (X_i - \bar{X})(X_{i+k} - \bar{X})}{\sigma^2(X)},$$

$\bar{X}$  – eilutės  $X$  vidurkis,  $\sigma^2(X)$  – dispersija,  $k \in Z_+ = \{1, 2, \dots\}$ ,  $0 < \beta < 1$ ,

$L_i$  – lėtai begalybėje kintanti funkcija, t. y.  $\lim_{t \rightarrow \infty} \frac{L_i(tx)}{L_i(t)} = 1$ , visiems  $x > 0$ .

Agreguota eilutė  $x_t$  ir neagreguotas procesas  $X$  apibūdinami autokoreliacine funkcija, kuri hiperboliškai mažėja didinant laiko intervalą. Sakoma, kad procesas  $X$  pasižymi greitai mažėjančia priklausomybe  $SRD$ , jei  $x_t$  ir  $X$  aprašomi greitai mažėjančia eksponentine formule:  $r(k) \approx \rho^k, k \rightarrow \infty, 0 < \rho < 1$  (Figueiredo, Liu, Misra, Towsley 2002, Cai, Li, Chen, Wang 2009).

### 3.3. Savastingumas ir jo analizės metodai kompiuterių tinkluose

Savastingumo fenomenas yra paaiškinamas tinklo paslaugų vartojimo pobūdžiu, kuris pasižymi pliūpsniškumu Šiuolaikinėse universitetinėse studijose plačiai naudojami kompiuterių tinklai, kurie dažnai patiria neprognozuojamas perkrovas. Efektyviam tinklo valdymui būtina atlikti tinklo mazgų stebėjimą siekiant prognozuoti tinklo mazgo apkrovas bei perkrovas. Remiantis A. Erramilli, O. Narayan ir W. Willinger 1989 m. 10 Mbps vietinio *Ethernet* tinklo srauto empiriniais tyrimais atliktais Bellcore laboratorijoje nustatyta, kad *Ethernet* tinklo srauto charakteristikos turi fraktalų savybių bei pasižymi savastingumu su  $LRD$  (Erramilli, Narayan, Willinger 1996). J. Beran tinklo srautą analizuoja kaip fraktalinį procesą, pasižymintį antros eilės

statistiniu savastingumu, charakterizuojamą fraktaliniu matu (Beran 1998). Tinklo procesų modeliavimui ir aprašymui taikomi netiesinės analizės (chaoso teorija) metodai, įvertinant sunkias uodegas, kurios charakterizuoja didelius tinklo srautų pliūpsnius (Kaklauskas, Sakalauskas 2008, Kaklauskas, Sakalauskas 2009, Kaklauskas, Sakalauskas 2010a, Kaklauskas, Sakalauskas 2011).

Pagal Samorodnitsky (Samorodnitsky 2006b) stochastinis procesas  $Y(t), t \geq 0$  yra savastingasis, jei galima rasti tokį  $H$ , kad visiems  $c > 0$ , būtų tenkinama lygybė:

$$(Y(ct), t \geq 0) \stackrel{d}{=} (c^H Y(t), t \geq 0). \quad (\text{Samorodnitsky 2006b, 213p.})$$

$\stackrel{d}{=}$  reiškia, kad ši lygybė galioja visuose funkcijos taškuose, išskyrus trūkio taškus. Jei agreguotos laiko eilutės turi stacionarius pokyčius, vadinasi dalinės sumos  $S_n = X_1 + X_2 + \dots + X_n$ , kur  $n=1,2,\dots$ , kai  $n \geq 1$ , o  $X_i = Y(i) - Y(i-1), i=1,2,\dots$  aprašo stacionarų procesą  $X = (X_1, X_2, \dots)$ , tenkinantį lygybę  $S_n = n^H S_1$ . Čia eksponentė  $H$  nusako stacionaraus proceso  $X$  pasiskirstymo reikšmingumą ir vadinama *Hurst'o* koeficientu. *Hurst'o* koeficiento reikšmė apibūdina laiko eilutės atminties tipą. Jei *Hurst'o* koeficientas  $H=0,5$ , vadinasi sekos nariai yra atsitiktiniai ir kiekvienas sekantis jos narys nepriklauso nuo ankstesnių eilutės narių, priešingu atveju galima teigti, kad ankstesni įvykiai užfiksuoti laiko eilutėse turi pastovią įtaką tolesniems procesams ir ta įtaka tuo stipresnė kuo įvykis yra artimesnis praeičiai. Tokios eilutės yra invariantinės laiko atžvilgiu.

Jei  $0 \leq H < 0,5$ , vadinasi laiko eilutės aprašomas procesas yra antipersistentinis, t. y. galima teigti, kad jei vienu laiko periodu yra stebimas kylimas, tai sekančiu bus būtinai kritimas su tuo didesne tikimybe, kuo  $H$  arčiau 0. Šiuo atveju koreliacija yra neigiama ir artėja prie 0,5. Tokios eilutės paprastai turi savybę stipriai kisti ir yra sudarytos iš dažnų kilimų bei kritimų.

Jei  $0,5 < H < 1,0$ , tai persistentinis procesas su *LRD*, dar vadinamas Markovo priklausomybe t.y. jei praeityje procesas turėjo savybę didėti, tai ir

ateityje jis ją išsaugos su tuo didesne tikimybe, kuo  $H$  artimesnis 1. Paprastai tokios eilutės vadinamos atspariomis trendams,  $H$  artėjant prie 0,5 eilutėje daugėja trendų (triukšmų). Vadinasi vertinant laiko eilutės savastingumą labai svarbi *Hurst*'o koeficiento reikšmė, t.y. intervalas, į kurį jis papuola.

### 3.3.1. *Hurst*'o koeficiento vertinimas

*Hurst*'o koeficiento vertinimui dažniausiai naudojamos dvi metodikos: laiko sekų analizės ir dažninių/banginių savybių įverčiai (Karagiannis, Faloutsos, Molle 2003). Pagal šias metodikas skaičiuojamas *Hurst*'o koeficientas laiko eilutėms  $x_t$  ir vertinama jo reikšmė.

Metodai pagrįsti laiko sekų analize yra šie:  $R/S$ , absoliutinių reikšmių kartais vadinamų absoliutinių momentų, dispersijos, liekanų dispersijos.

Seniausias savastingumo skaičiavimo metodas yra  $R/S$  statistikos naudojimas skaičiuojant *Hurst*'o koeficientą (Beran 1998). *Hurst*'o koeficientas skaičiuojamas pagal formulę  $H = \log(R/S) / \log(N/2)$ , kur  $H$  – *Hurst*'o koeficientas,  $R/S$  yra statistika skaičiuojama pagal formulę:

$$R/S = \frac{R(N)}{S(N)} = \frac{\text{Max}(\sum_{i=1}^{\tau} (X_i - \bar{X}_t)) - \text{Min}(\sum_{i=1}^{\tau} (X_i - \bar{X}_t))}{\sqrt{\frac{1}{N} \sum_{i=1}^N (X_i - \bar{X}_t)^2}}.$$

Čia  $1 \leq \tau \leq N$ , kur  $N$  – sekos narių skaičius,  $\bar{X}_t$  – vidutinė laiko eilutės  $X_t$  reikšmė, o  $\sum_{i=1}^{\tau} (X_i - \bar{X}_t)$  – suformuota kumuliatyvi eilutė aprašanti pokyčių sumą per laiką  $\tau$ . Pagal Hurst (1951) galima teigti, kad daugeliui gamtoje vykstančių reiškinių teisinga išraiška:  $M\left(\frac{R(N)}{S(N)}\right) \sim cN^H, N \rightarrow \infty$ , kur  $c$  – pastovus nepriklausomas nuo  $N$  dydis – konstanta (Park, Willinger 2000).

*Hurst*'o koeficientas glaudžiai susijęs su *Hausdorfo* matu  $D$ , kuris charakterizuoja kompiuterių tinklo duomenų srauto eilutės lokalias savybes, o *Hurst*'o koeficientas aprašo viso proceso charakteristika – proceso atmintį.

*Hurst*'o koeficientą galima įvertinti ir grafiškai tam panaudojant formulę:

$$\log\left(M\left[\frac{R(N)}{S(N)}\right]\right) = H(\log(N) + \log(c)), N \rightarrow \infty. \text{ Čia } N \text{ eilutės narių skaičius.}$$

Abscisių ašyje atidedamos  $\log(N)$  reikšmės, o ordinačių –  $\log\left(M\left[\frac{R(N)}{S(N)}\right]\right)$ .

Analizuojant gautus grafikus buvo pastebėta, kad *Hurst*'o koeficientui artėjant prie vieneto grafike mažėja triukšmų, t.y. kreivė tolydėja (Sheluhin, Smolskiy, Osin 2007).

*Absoliutinių momentų* metodas pagrįstas  $N$  ilgio laiko sekos dalinimu į  $m$  ilgio blokus, formuojant dalines sekas  $X^{(m)}(k)$ , kur  $k=1,2,\dots,[N/m]$ . Tada skaičiuojamas sekos  $n$ -tasis momentas (Belov, Kabašinskas, Sakalauskas 2006):

$$AM_n^{(m)} = \frac{1}{N/m} \sum_{k=1}^{N/m} |X^{(m)}(k) - \bar{X}|^n, \text{ kur } X^{(m)}(k) = \frac{1}{m} \sum_{i=(k-1)m+1}^{km} X_i.$$

Seka  $X^{(m)}$  asimptotiškai elgiasi kaip ir  $Cm^{n(H-1)}$  dideliems  $m$ , vadinasi gautasis  $AM_n^{(m)}$  momentas yra proporcingas  $m^{n(H-1)}$ .

Dispersijos metodas suformuotai sekai  $X^{(m)}(k)$  skaičiuoja imties dispersiją (Erramilli, Narayan, Willinger 1996):

$$\text{Var}(X^m) = \frac{1}{N/m} \sum_{k=1}^{[N/m]} (X^{(m)}(k) - \bar{X})^2.$$

Seka  $X^{(m)}(k)$  asimptotiškai elgiasi kaip  $m^{H-1}$ , jei ji turi baigtinę dispersiją, vadinasi dideliems  $N/m$  dispersijų seka asimptotiškai yra proporcinga  $m^{2H-2}$ .

Pagal Peng pasiūlytą liekanų dispersijos metodą  $m$  ilgio sekos poabiams mažiausių kvadratų metodu skaičiuojamos dalinių sumų tiesinės priklausomybės liekanų dispersijos (Peng 1994):

$$\frac{1}{m} \sum_{j=1}^m (Y(j) - a - bj)^2, \text{ kur } Y(j) = \sum_{i=1}^j X_i.$$



Visoms gautosioms dispersijoms, proporcingoms  $m^{2H}$ , skaičiuojama bendra mediana ir vertinama log-log tipo priklausomybė tarp  $m$  ir posvyrio kampo  $2H$ , jei ji tiesinė, tada regresijos pagalba įvertinamas  $H$ .

Dažniųjų/banginių savybių įverčiai grindžiami dažninėmis bangos vilnelių savybėmis. Analizei dažniausiai taikomi šie metodai: periodogramos, sumažinimo, *Abby-Veich*.

Periodogramos metodu aprašoma dažninė funkcija (Taqqu, Teverovsky 1998):

$$I(\nu) = \frac{1}{2\pi N} \left| \sum_{j=1}^N X(j) e^{ij\nu} \right|^2,$$

kur  $\nu$  – dažnis,  $N$  – sekos ilgis,  $X(j)$  – laiko eilutė. Kai  $I(\nu)$  turi baigtinę dispersiją, tai dažninė funkcija nusako sekos  $X$  tankį, kuris ilgos atminties atveju bus proporcingas  $|\nu|^{1-2H}$ , dažniams artimiems koordinatų pradžiai. Log-log regresija naudojama *Hurst'o* koeficiento įvertinimui.

Sumažinimo metodas remiasi periodogramos maksimalaus tikėtimumo funkcijos minimizavimu, kai žinoma spektrinio tankio funkcija (Kokoszka, Taqqu 1996):

$$Q^*(\eta) = \sum_{j=1}^{[(N-1)/2]} \frac{I(\nu_j)}{f^*(\nu_j, \eta)}.$$

Čia  $\eta$  yra vektoriaus minimizuojančio tikslo funkciją  $Q^*$  reikšmė, kurią apskaičiavus gaunamas *Hurst'o* koeficiento įvertis bei pasikliautasis intervalas, kai žinoma spektrinio tankio funkcija.

*Abry-Veitch* metodu *Hurst'o* koeficientas įvertinamas naudojant vilnelių sekos transformaciją (Abry, Veitch 1996, 6p.):

$$\hat{H}(j_1, j_2) = \frac{1}{2} \left[ \frac{\sum_{j=j_1}^{j_2} S_j j \eta_j - \sum_{j=j_1}^{j_2} S_j j \sum_{j=j_1}^{j_2} S_j \eta_j}{\sum_{j=j_1}^{j_2} S_j \sum_{j=j_1}^{j_2} S_j j^2 - \left( \sum_{j=j_1}^{j_2} S_j j \right)^2} \right].$$

Čia  $\eta_j = \log_2 \left( \frac{1}{n_j} \sum_k |d_x(j, k)|^2 \right)$ , svoriai –  $S_j = (n \ln^2(2)) / 2^{j+1}$ ,

$|d_x(j, k)|^2$  – proceso energijos per  $2^j k$  laiką  $2^j v_0$  dažniui matas, kai  $v_0$  parenkamas iš taip vadinamos motininės vilnelės, o  $n$  – dalinės sekos ilgis. Metodas plačiau aprašytas P. Abry ir D. Veitch straipsnyje.

### 3.3.2. Stabiliųjų modelių parametrų vertinimas

Kaip rašyta 2.2.2 skyriuje savastingasis simetriškas procesas aprašytas formulėmis ir pasižymintis begaline dispersija, yra  $\alpha$ -stabilusis procesas (Samorodnitsky 2006a). Vertinant  $\alpha$ -stabilųjį procesą  $S_\alpha(\beta, \sigma, \mu)$  reikia apskaičiuoti ir įvertinti keturis stabilumo parametrus:  $\alpha$  – stabilumo parametras,  $\beta$  – asimetrijos parametras,  $\sigma$  – mastelio parametras,  $\mu$  – padėties parametras.

*Fama-Roll* simetrinių stabilųjų parametrų įverčių metodas plačiau aptartas 1971 m. straipsnyje (Fama, Roll 1971). (McCulloch 1986) išvystė *Fama-Roll* sukurtus stabilųjų simetrinių dydžių vertinimo metodus, parengė interpoliacines lenteles. Čia stabilieji parametrai apskaičiuojami interpoliuojant reikšmes pagal duotąsias eilutes.

Regresijos metodą stabilųjų dydžių parametrų įvertinimui pasiūlė Koutrouvelis (Koutrouvelis 1981), jo taikymus finansinėms eilutėms plačiau aprašė I. Belovas, A. Kabašinskas ir L. Sakalauskas (Belov, Kabašinskas, Sakalauskas 2006).  $\alpha$  ir  $\sigma$  skaičiavimams naudojamos

sumos:  $s_1 = \sum_{k=1}^K y_k w_k$ ,  $s_2 = \sum_{k=1}^K y_k$ ,  $s_3 = \sum_{k=1}^K w_k$ ,  $s_4 = \sum_{k=1}^K w_k^2$ , kur sumose

naudojami parametrai apskaičiuojami:  $w_k = \log |t_k|$ ,

$y_k = \log(-\log(|\phi_n(t_k)|^2))$ ,  $t_k = \frac{\pi k}{25}$ .  $\alpha$  ir  $\sigma$  apskaičiuojami pagal formules

(Koutrouvelis 1981):

$$\alpha = \frac{K s_1 - s_2 s_3}{K s_4 - s_3^2}, \quad \sigma = \tilde{\sigma} \left( 0.5 \exp \left\{ \frac{s_4 s_2 - s_1 s_3}{K s_4 - s_3^2} \right\} \right)^{1/\alpha},$$

čia  $\tilde{\sigma}$  absoliutinis nuokrypis,  $K$  rekomenduojama reikšmė 10.

$\beta$  ir  $\mu$  skaičiavimui naudojamos sumos:  $s_5 = \sum_{l=1}^L u_l^2$ ,  $s_4 = \sum_{k=1}^K w_k^2$ ,  
 $s_6 = \sum_{k=1}^K q_l g_n(u_l)$ ,  $s_7 = \sum_{l=1}^L q_l u_l$ ,  $s_8 = \sum_{l=1}^L q_l^2$ ,  $s_9 = \sum_{l=1}^L u_l g_n(u_l)$ , kur sumose  
naudojami parametrai apskaičiuojami:  $u_l = \frac{\pi l}{50}$ ,  $q_l = |\sigma u_l|^\alpha \tan\left(\frac{\pi\alpha}{2}\right) \text{sign}(u_l)$ ,

o  $L=15$ .  $\beta$  ir  $\mu$  skaičiavimui taikomos formulės:

$$\beta = \frac{s_5 s_6 - s_7 s_9}{s_5 s_8 - s_7^2}, \quad \mu = \tilde{\mu} + \sigma \left( \frac{s_8 s_9 - s_6 s_7}{s_5 s_8 - s_7^2} \right) - h\pi.$$

Išsamesnį skaičiavimo metodikos aprašą galima rasti Kotrouvelis straipsnyje (Kotrouvelis 1981).

Momentų metodą stabilijų dydžių parametru įvertinimui pasiūlė S. J. Press (Press 1972). Šis metodas pagrįstas laiko eilutės empirinės charakteringosios funkcijos skaičiavimu:  $\hat{\phi}(t) = \frac{1}{n} \sum_{i=1}^n e^{itx_i}$ , kur  $n$  – eilutės elementų kiekis,  $x_i$  –  $i$ -asis eilutės elementas,  $t$  – pasirinktoji eilutės reikšmė. Stabiliuosius parametrus siūloma apskaičiuoti naudojant tokias formules (Press 1972):

$$\hat{\alpha} = \frac{\log \frac{\log |\hat{\phi}(t_1)|}{\log |\hat{\phi}(t_2)|}}{\log \left| \frac{t_1}{t_2} \right|},$$

$$\log \hat{\sigma} = \frac{\log |t_1| \log(-\log |\hat{\phi}(t_2)|) - \log |t_2| \log(-\log |\hat{\phi}(t_1)|)}{\log \left| \frac{t_1}{t_2} \right|},$$

$$\hat{\beta} = \frac{\frac{\bar{u}(t_4)}{t_4} - \frac{\bar{u}(t_3)}{t_3}}{(|t_4|^{\alpha-1} - |t_3|^{\alpha-1}) \hat{\sigma}^\alpha \tan\left(\frac{\pi\hat{\alpha}}{2}\right)} \hat{\mu} = \frac{|t_4|^{\alpha-1} \frac{\bar{u}(t_4)}{t_4} - |t_3|^{\alpha-1} \frac{\bar{u}(t_3)}{t_3}}{|t_4|^{\alpha-1} - |t_3|^{\alpha-1}}.$$

Čia  $t_1, t_2, t_3, t_4$  – pasirinktos reikšmės, tenkinančios lygybes  $t_1 \neq t_2$  ir

$$t_3 \neq t_4, \text{ o } \bar{u}(t) = \arctan \left( \frac{\sum_{i=1}^n \sin(tx_i)}{\sum_{i=1}^n \cos(tx_i)} \right). \text{ Pagal Kotrouvelis pasiūlymą geriausiai}$$

parinkti tokias reikšmes:  $t_1 = 0.2, t_2 = 0.8, t_3 = 0.1, t_4 = 0.4$ .

### 3.3.3. Prieaugio santykio statistika (IR statistika)

Tai nauja statistika, kuri pagrįsta proceso  $S_n = \sum_{t=1}^n X_t$  eilutės dalinių sumų skaičiavimu, taikant antros eilės intervalus arba skirtumus. Ji pavadinta prieaugio santykio statistika IR (angl. *Incerment ratio*) (Surgailis, Teysiere, Vaičiulis, 2008). Statistika apskaičiuojama pagal formulę:

$$IR = \frac{1}{N-3m} \sum_{k=0}^{N-3m-1} \frac{|\sum_{t=k+1}^{k+m} (X_{t+m} - X_t) + \sum_{t=k+m+1}^{k+2m} (X_{t+m} - X_t)|}{|\sum_{t=k+1}^{k+m} (X_{t+m} - X_t)| + |\sum_{t=k+m+1}^{k+2m} (X_{t+m} - X_t)|}.$$

Čia  $X_1, X_2, \dots, X_N$  yra laiko eilutė, kurios ilgis  $N$ ,  $m = 1, 2, \dots$  – pralaidos parametras. Pagal apibrėžimą IR statistika yra apribota ir kinta  $0 \leq IR \leq 1$ . Įverčių skalė nepasikeičia, kai  $X_t$  yra pakeičiama tiesine kombinacija  $aX_t + b$ , kur  $a \neq 0$ , o  $b$  yra konstanta. Autorių atlikti empiriniai proceso simuliacijos parodė, kad IR statistika yra pakankamai nejautri triukšmams, lokaliems nestacionarumams ir sunkioms uodegoms. Statistikos stabilumas aktualus analizuojant kompiuterių tinklo srautą, nes jis pasižymi dideliu pliūpsniškumu.

IR statistika sėkmingai naudojama hipotezės apie nežinomą parametą tikrinimui. Sakykime tikrinama statistinė hipotezė:

- $H_0$  – laiko eilutė  $X_1, X_2, \dots, X_N$  pasižymi SRD;
- $H_1$  – laiko eilutė  $X_1, X_2, \dots, X_N$  pasižymi LRD.

Jei tenkinama nelygybė:  $IR - \Lambda(0) > z_\alpha \sigma(0) \sqrt{\frac{m}{N-3m}}$ , tai su pasiklovimo

lygmeniu 0,95 galima teigti, kad agreguota eilutė pasižymi LRD t. y.  $H_0$  hipotezę atmetame. Nelygybės skaičiavimui autoriai rekomenduoja naudoti

tokias reikšmes:  $\Lambda(0) = 0,5881$ ,  $\sigma(0) = 0,2080$ ,  $z_{0,95} = 1,64$ , o  $m = [N^{1/3}]$ , čia  $[.]$  reiškia sveikąją skaičiaus dalį (Surgailis, Teyssiere, Vaičiulis, 2008). Apibendrinant galima teigti, kad hipotezė  $H_0$  reiškia, kad *Hurst*'o koeficientas  $0 < H < 0,5$ ,  $H_1 - 0,5 < H < 1$ . Šią statistiką pritaikėme hipotezės apie srauto savastingumą tikrinimui realiu laiku (Kaklauskas, Sakalauskas 2011b).

### ***3.4. Skyriaus išvados***

Parengta paketinio tinklo srauto savastingumo matavimo metodika, aprašyti technologiniai srauto matavimo sprendimai, ištirtas savastingumo parametrų vertinimo metodų tinkamumas savastingumo įverčių skaičiavimui realiu laiku.

Nustatyta, kad naudojant chaoso teorijos priemones, tinklo srauto savastingumo įverčiai gaunami apskaičiuojant atraktorių dimensijas, eilutės fraktalinį bei koreliacinį matus. Yra siūloma savastingumą vertinti analizuojant nubrėžtus grafikus, t. y. grafikas vaizduoja procesą pasižymintį LRD, kai jis aprašomas autokoreliacine hiperboliškai mažėjančia funkcija.

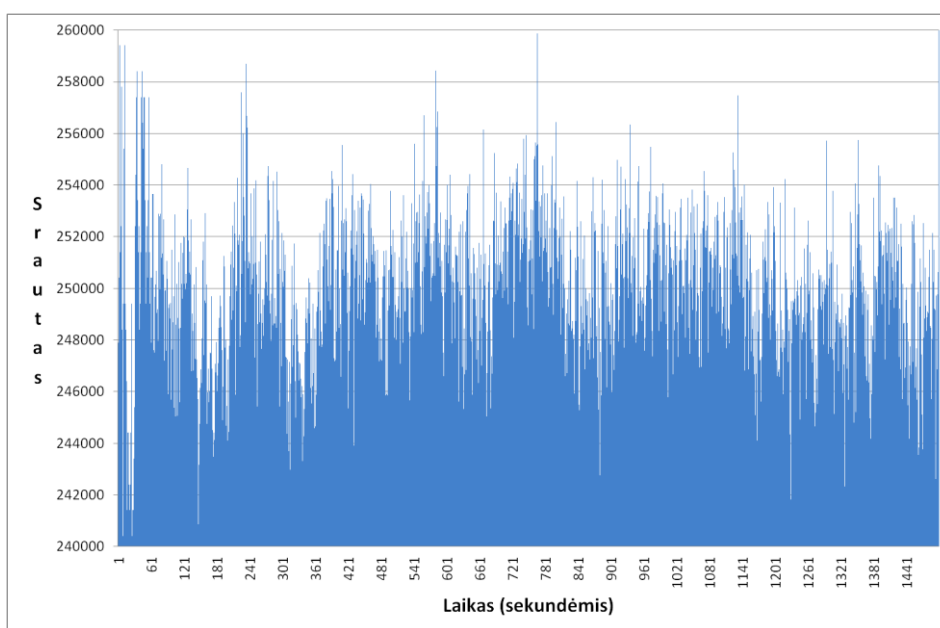
Tyrimais nustatyta, kad kompiuterių tinklo srauto savastingumo įverčiai gaunami apskaičiuojant *Hurst*'o koeficientą pagal laiko sekų analizės ir eilutės dažniųjų/banginių savybių įverčius bei skaičiuojant  $\alpha$ -stabiliojo proceso keturis stabilumo parametrus pagal sukurtą metodiką, kurioje realizuoti: *Fama-Roll*, *McCulloch*, *regresijos* ir *momentų* metodai.

Tinklo srauto laiko eilučių savastingumo analizei taikytinas naujas ir spartus, pratiškai netestuotas, Lietuvos mokslininkų D. Surgailio ir M. Vaičiulio bendradarbiaujant su G. Teyssiere pasiūlytas metodas – *IR* statistikos įverčiai.

#### 4 skyrius. Fraktališkumo analizė kompiuterių tinkluose

Fraktalo pavadinimas yra kilęs iš lotyniško žodžio “fractus” (reiškiančio “sudarytas iš dalių”). Formaliai fraktalas – geometrinis objektas, kurio atskirose dalyse galima pamatyti jį patį, ar bent jau kažką labai panašaus į visą fraktalą. Tai tarsi žaidimas su veidrodžiu, prieš kurį atsistojame pasiėmę kitą – mažesnę – veidrodį. Didžiajame veidrodyje matome mažesnįjį, kuriame savo ruožtu atsispindi didesnis su visu jo rodomu vaizdu, taigi ir mažesnis ir taip toliau – šitaip išgauname be galo smulkėjančių to paties objekto kopijų seką. Taigi fraktalas – dalinai savipanašus darinys.

Suformuotos paketinio kompiuterių tinklo srauto laiko eilutės primena determinuotą chaosą, kuris priklauso nuo: kanalo pralaidos, tinklo mazgų našumo, duomenų perdavimo protokolo, klaidų korekcijos metodų, mazgo apsaugos taisyklių ir kitų objektyvių bei subjektyvių faktorių. 8 pav. matome, kad paketų skaičius perduodamas tinklo mazge per parą pastoviai kinta. Ordinačių ašyje atidėtas suminis duomenų srautas bitais, kuris užfiksuotas per vieną sekundę mazge. Kaip minėta magistralinio mazgo, kuriame atlikti matavimai, pralaida yra 1 Gbps. Pagal 8 pav. pavaizduotą grafiką matyti, kad mazge pastoviai išnaudojama apie 25% kanalo pralaidos.



8 pav. Charakteringas vienos paros tinklo mazge užfiksuoto srauto grafikas

3.1 skyriuje aprašytas laiko eilučių formavimo ir saugojimo procesas formuoja nestacionarias laiko eilutes. Siekiant geresnių skaičiavimo rezultatų rekomenduojama tokias eilutes agreguoti, t. y. pertvarkyti taip, kad jos turėtų stacionarius pokyčius (Abry, Flandrin, Taqqu, Veitch 2000).

Agreguotų laiko eilučių analizei ieškota laisvai platinamų programų, kurios analizei taikytų dažniausiai naudojamas *Hurst'o* koeficiento skaičiavimo statistikas (plačiau 3.3.1 skyriuje), naudotų chaoso teorijos elementus (plačiau 3.2 skyriuje), taikytų  $\alpha$ -stabiliojo proceso savybes (plačiau 3.3.2 skyriuje).

#### 4.1. Laiko eilučių formavimas

(Willinger, Paxson, Taqqu 1998) pateikė savastingumo apibrėžimą, kuris artimesnis laiko eilučių teorijai. Sakykime stacionariai sekai  $X = \{X(i), i \geq 1\}$

atitinka agreguota seka  $X^{(m)} = \frac{1}{m} \sum_{i=(k-1)m+1}^{km} X(i)$ ,  $k = 1, 2, \dots$ , su agregavimo

lygmeniu  $m$ , kuris originalią aibę  $X$  dalija į  $k$  nepersidengiančių su apskaičiuotais vidurkais bloką. Indeksas  $k$  žymi bloko numerį. Jei  $X$  yra  $Y$  savastingojo proceso, apibrėžto 3.3 skyriuje, auganti seka ir  $X(i) = Y(i+1) - Y(i)$ . Visiems sveikiems  $m$  stacionari seka  $X = \{X(i), i \geq 1\}$  yra vadinama tiksliai savastinga, jei ji tenkina lygybę visiems  $m$  agregavimo lygmenims. Šis apibrėžimas yra glaudžiai susijęs su 3.3 skyriuje pateikiamu apibrėžimu, kai  $mX^{(m)}(\cdot)$  atitinka  $Y(a)$ . Stacionari seka  $X = \{X(i), i \geq 1\}$

vadinama asimptotiškai savastinga, jei  $X \stackrel{d}{=} m^{1-H} X^{(m)}$  galioja kai  $m \rightarrow \infty$ . Pagal (Willinger, Paxson, Taqqu 1998) agreguotos laiko eilutės taip pat išsaugo savastingumą.

Iš realaus paketinio kompiuterių tinklo srauto suformuota eilutė atitinka matavimus atliktus laiko momentais  $\tau_1, \tau_2, \dots, \tau_n$ , kur  $n$  išsaugotų paketų kiekis. Reikia pažymėti, kad laiko tarpai  $\tau_i$  tarp atėjusių duomenų paketų  $x_i$  į matavimo posistemę yra nevienodi, t. y.,  $\tau_i - \tau_{i-1} \neq \tau_{i+1} - \tau_i$ , kur  $i \in [1, n]$ . Duomenų paketai  $x_1, x_2, \dots, x_n$  atitinkamai laiko tarpais nusako kanalo užimtumą įvairiais laiko momentais. Kadangi mes suformuotas laiko eilutes

analizuosime kaip stacionarias eilutes (žr. 2.2.2 skyriaus apibrėžimą), tai jas būtina agreguoti, t. y. apskaičiuoti duomenų srautus vienodais laiko tarpais  $t_i - t_{i-1} = t_{i+1} - t_i$ .

#### 4.1.1. Laiko eilučių agregavimas

(Abry, Baraniuk, Flandrin, Riedi, Veitch 2002) rašė, kad savastingąjį procesą galima apibrėžti ir kaip nestacionarų procesą, tačiau dažniau naudojamas savastingasis procesas su stacionariais pokyčiais, kurį autoriai pavadino *H-sssi* procesu. (Cappé, Moulines, Pesquet, Petropulu, Yang 2002) pateikė tokį apibendrintą savastingumo apibrėžimą: nenutrūkstamas laiko procesas  $\{X(t)\}_{t \in R}$  yra savastingasis procesas su koeficientu  $H > 0$ , jei visiems  $a > 0$ ,  $n \geq 1$  ir  $(t_1, t_2, \dots, t_n)$ , sekos  $(X(at_1), X(at_2), \dots, X(at_n))$  ir  $(a^H X(t_1), a^H X(at_2), \dots, a^H X(at_n))$  turi vienodus skirstinius. Procesas  $\{X(t)\}_{t \in R}$  turi stacionarius pokyčius, jei visiems  $t_0 \in R$  perkeltas procesas  $\{X(t + t_0) - X(t)\}_{t \in R}$  turi identišką skirtinį, nepriklausomai nuo  $t_0$  vertės. Procesas  $\{X(t)\}_{t \in R}$  vadinamas *H-sssi* procesu, jei jis savastingas su stacionariais pokyčiais (Cappé, Moulines, Pesquet, Petropulu, Yang 2002, Sheluhin, Smolskiy, Osin 2007).

Diskretiems laiko procesams savastingumas tiksliau įvertinamas, kai pritaikomas agregavimas ir masteliavimas, kurie yra invariantiški skirstinio savybių pokyčiams, t. y. išsaugo savastingumą *LRD* ir *SRD* atvejais. Agregotu procesu duotajam procesui  $\{X_k\}$  vadinamas procesas  $\{X_k^{(m)}\}_{k \in Z}$  su agregavimo eile  $m$ . Jei  $X_k^{(m)} = \sum_{j=(k-1)m+1}^{km} X_j$ , kai  $k=1,2,\dots$ , tai agregavimas vykdomas naudojant  $m$  dydžio bloką sumas. Jei  $X_k^{(m)} = \frac{1}{m^H} \sum_{j=(k-1)m+1}^{km} X_j$  –  $m$  dydžio bloką vidurkius, o  $0 < H < 1$ . Abiem atvejais kovariacinė funkcija yra  $R^{(m)}(k)$ .



Pagal (Sheluhin, Smolskiy, Osin 2007) bet koks diskretus atsitiktinis procesas  $\{X_k, k \in Z\}$  yra griežtai savastingas plačiaja prasme (tikslus antros eilės savastingumas) su savastingumo parametru  $H$ , kai  $0,5 < H < 1$ , jei

$$R(k) = \frac{\sigma^2}{2} \left[ (k+1)^{2H} - 2k^{2H} + (k-1)^{2H} \right], \text{ bet kokiam } k \geq 1. \text{ Procesas } Y(t) \text{ yra}$$

asimptotiškai savastingas plačiaja prasme (antros eilės asimptotinis savastingumas  $H$ -ssa), jei  $\lim_{m \rightarrow \infty} R^m(k) = \frac{\sigma^2}{2} \left[ (k+1)^{2H} - 2k^{2H} + (k-1)^{2H} \right]$ .

Remiantis aukščiau aprašytais formulėmis savastingumas plačiaja prasme reiškia kovariacinę struktūrą, kuri sako, kad agreguotoms laiko eilutėms savastingumas yra išsaugomas. Griežto savastingumo  $R(k)$  atveju jis tiksliai atitinka neagreguotos eilutės savastingumą, o asimptotinio  $\lim_{m \rightarrow \infty} R^m(k)$  – gali būti nežymi paklaida.

Analogiškai procesu su pakeistu masteliu duotajam procesui  $\{X_k\}$  vadinamas procesas  $\{m^{-a} X_k^{(m)}\}_{k \in Z}$ , kai  $m \geq 1$  ir naujai gautas procesas yra vadinamas savastinguoju su parametru  $a$  (Cappé, Moulines, Pesquet, Petropulu, Yang 2002). Siekiant tikslesnių skaičiavimo rezultatų yra siekiama nustatyti optimalų  $m$  bloko dydį, kai skaičiavimams naudojamos didelės apimties eilutės su sunkių uodegų indeksu, kintančiu plačiame intervale (Stoev, Michailidis 2010, Stoev, Michailidis, Taqqu 2011).

Siekiant tikslesnių paketinio tinklo srauto savastingumo įverčių, duomenų agregavimui naudotos dvi agregavimo strategijos: judančių vidurkių glotninimas ir perduotų duomenų srauto per laiko intervalą  $\Delta t$  sumos skaičiavimas.

Taikant judančių vidurkių glotninimo metodą pasirinktu laiko intervalu  $\Delta t$  duomenų eilutei apskaičiuojamas vidutinis srautas:

$$x_k^\Delta = \frac{\sum_{\tau_i \in [t_k, t_{k-1}]} x_i}{\Delta t},$$

čia  $t_k = k \cdot \Delta t + \tau_1$ . Gautos laiko eilutės aprašo vidutinius duomenų srauto pokyčius laiko momentais  $\Delta t$ . Vidurkių glotninimo atveju vertinant savastingumą naudojama agreguota eilutė  $X_k^{(m)\Delta} := \sum_{j=(k-1)m+1}^{km} x_j^\Delta$ .

Pagal antrąjį agregavimo metodą buvo skaičiuojama perduotų duomenų srauto per laiko intervalą  $\Delta t$  suma:  $x_k^\Sigma = \sum_{\tau_i \in [t_k, t_{k-1}]} x_i$ ,

čia  $t_k = k \cdot \Delta t + \tau_1$ . Dabar savastingumo vertinimui naudojama eilutė  $X_k^{(m)\Sigma} := \sum_{j=(k-1)m+1}^{km} x_j^\Sigma$  (Kaklauskas, Sakalauskas 2011).

Kadangi tinklo mazgo pralaida, kuriame vykdome matavimus siekia iki 1 Gbps, todėl pasirinktos nedidelės  $\Delta t$  reikšmės:  $\Delta t \in [100ms, 500ms, 1000ms]$ .

Duomenų bazėje buvo sukaupta per tris milijardus paketinio duomenų srauto matavimo įrašų, kurie atitiko 8936965 sek. arba 103 dienos 10 val. 29 min. ir 25 s. srauto matavimo laiką. Siekiant įvertinti ir palyginti tinklo srauto savastingumą, kai jo intensyvumas yra skirtingas, atlikta pirminė srauto analizė. Išsamesnei analizei duomenų aibė išskaidyta į tris poaibius, pagal tinklo mazgo apkrovas vertinant savaitės dienas. Gauta, kad tinklo mazgo apkrova yra minimali – sekmadienį, vidutinė – šeštadienį ir maksimali – darbo dienomis. Suformuoti poaibiai analizuoti taikant klasterinės analizės k-vidurkių metodą, panaudojant programų paketą SPSS (angl. *Statistical Package for the Social Sciences*). Pagal nehierarchinės klasterinės analizės metodą, kai klasterių skaičius  $k=24$  (24 valandos paroje), atstumai tarp klasterių ir objektų buvo skaičiuojami pasinaudojant Euklido atstumo kvadrato metriką. Nustatyti vienos valandos laiko intervalai, kai per tinklo mazgą praeinantis informacinis srautas yra mažiausias – formuojama eilutė  $\{X_k^{\min}\}$ , vidutinis – formuojama eilutė  $\{X_k^{avg}\}$  ir didžiausias – formuojama eilutė  $\{X_k^{\max}\}$ . Taip iš trijų poaibių atrinktos neagreguotos 309 matavimų sekos, kai  $n \in (500000, 1500000)$ . Kiekviena seka agreguota taikant du skirtingus agregavimo metodus.

Pirmajame etape savastingumo analizei suformuotos 1854 eilutės, suskirstytos į 6 grupes:  $\{X_k^{\min \Delta}\}$ ,  $\{X_k^{\min \Sigma}\}$ ,  $\{X_k^{avg \Delta}\}$ ,  $\{X_k^{avg \Sigma}\}$ ,  $\{X_k^{\max \Delta}\}$ ,  $\{X_k^{\max \Sigma}\}$ .

Agreguojant kiekvieną eilutę panaudota  $\Delta t \in [100ms, 500ms, 1000ms]$ , todėl kiekviena iš 6 grupių išskaidyta į tris pogrupius ir galutinei analizei suformuotos eilutės:  $\{X_k^{\Delta \min \Delta}\}$ ,  $\{X_k^{\Delta \min \Sigma}\}$ ,  $\{X_k^{\Delta avg \Delta}\}$ ,  $\{X_k^{\Delta avg \Sigma}\}$ ,  $\{X_k^{\Delta \max \Delta}\}$ ,  $\{X_k^{\Delta \max \Sigma}\}$ . Savastingumui vertinti suformuotos 5562 eilutės.

#### 4.1.2. Agreguotų laiko eilučių formavimas realiu laiku

Formuojant agreguotą eilutę realiu laiku, pirmą kartą suformuota eilutė atnaujinama rekurentiškai, t. y. naujas eilutės narys pridedamas vietoj ilgiausiai eilutėje buvusio nario. Naujasis eilutės narys dažniausiai įrašomas į jos pabaigą, o senasis išvalomas iš jos pradžios. Tradiciškai po tokio pakeitimo kiekvieną kartą reikia perrašyti naujai suformuotą eilutę, dėl ko sistema sugaišta pakankamai daug laiko. Eilutės formavimo ir agregavimo proceso spartinimui sukūriau agreguotos eilutės perindeksavimo algoritmą, kuris keičia tik eilutės narių indeksus, taip taupydamas eilutės formavimo laiką.

Pradedant agreguoti kompiuterių tinklu perduodamą duomenų srautą pirmiausiai suformuojama agreguota eilutė  $\{X_k^{(m)}\}$ , kur  $m$  – agregavimo eilė, o  $1 \leq k \leq n$ , čia  $n$  – laiko eilutės narių skaičius. Vėliau suformuota eilutė rekurentiškai atnaujinama.

**Algoritmas.** Agreguotos eilutės formavimas naudojant perindeksavimą (autorius L.Kaklauskas).

*Tikslas:* rekurentiškai formuoti agreguotą eilutę paketiniame kompiuterių tinkle realiu laiku.

*Įėjimo parametrai:*  $t_i$  – paketo atėjimo laikas  $i$ -tuoju momentu,  $x_i$  –  $i$ -oji paraiška,  $n$  – formuojamos eilutės narių skaičius,  $ei$  – eilutės nario indeksas, kuris dabar bus keičiamas naujai atėjusiu paketu, *seka1* – požymis, 0 – reiškia, kad eilutė formuojama, 1 – pradedamas rekurentinis suformuotos eilutės atnaujinimas.

Išėjimo parametrai:  $\{X_k^{(m)}\}$  – agreguota laiko eilutė,  $ei$  – pirmojo eilutės nario indeksas.

1 žingsnis. If  $seka1=1$  tai vykdomi 2 – 4 žingsniai;

2 žingsnis.  $x_{ei} = x_i$ ;

3 žingsnis. If  $ei < n$  then  $ei=ei+1$  else  $ei=1$ ;

4 žingsnis. Suformuota eilutė  $\{X_k^{(m)}\}$  ir eilutė pradžios numeris  $ei$

perduodami savastingumo vertinimo posistemėi.

Savastingumo vertinimo posistemėje skaičiuojant suformuotos eilutės  $\{X_k^{(m)}\}$  savastingumo indeksą naudojami du ciklai, kad perrinkti visus eilutės narius: pirmasis ciklas perrinks eilutės narius su indeksais  $ei \leq i \leq n$ , o antrasis – su indeksais  $1 \leq i \leq ei - 1$ . Šis perindeksavimas panaikina bereikalingą eilutės elementų perrašymą, taip, kad jie būtų išdėstyti iš eilės, pradedant pirmuoju eilutės nariu pirmoje pozicijoje ir baigiant paskutiniu nariu eilutės pabaigoje.

#### **4.2. Tinklo srauto analizės priemonės**

Galima rasti dvi laisvai platinamas programas: 2003 metų V. Sychyov *Fractan* 4.4 (Sychyov 2003) ir T. Karagiannis 2002 metų *Selfis* 0.1b (Karagiannis 2002). Abi programos analizuoja suformuotas eilutes ir analizės rezultatus pateikia analitine ir grafine formomis. Laiko eilutės šioms programoms perduodamos kaip tekstiniai failai, o rezultatai gražinami tekstine ir grafine formomis.

*Fractan* 4.4 *Hurst'o* koeficiento skaičiavimui naudoja *R/S* statistiką, taip pat skaičiuoja koreliacijos matą, fazinės erdvės matą, autokoreliacijos funkciją išskiriant optimalų laiko intervalą, koreliacijos entropiją, *Hurst'o* koeficientą bei fraktalinį matą (žr. 3.2 ir 3.3.1 skyrius). Analitinius įverčius pateikia grafine forma, nubrėžia eilučių atraktorius. Programa gali generuoti klasikines triukšmų eilutes.

*Selfis* 0.1b agreguotose eilutėse skaičiuoja *Hurst'o* koeficientą taikant dvi metodikas:

1. laiko eilutės analizės, kur realizuoti absoliutinių reikšmių, dispersijos, R/S, liekanų dispersijos metodai,
2. dažninių/banginių savybių įvertinimo, kur realizuoti periodogramos, sumažinimo, Abby-Veich metodai.

Aukščiau išvardinti savastingumo vertinimo metodai plačiau aprašyti 3.3.1 skyriuje. Naudojant *Selfis* 0.1b galima generuoti Gauso triukšmą, skaičiuoti procentinius ir kitus standartinius statistinius įverčius, apskaičiuoti autokoreliacijos funkciją, rasti vertinamos eilutės narių reikšmių pokyčių skalę.

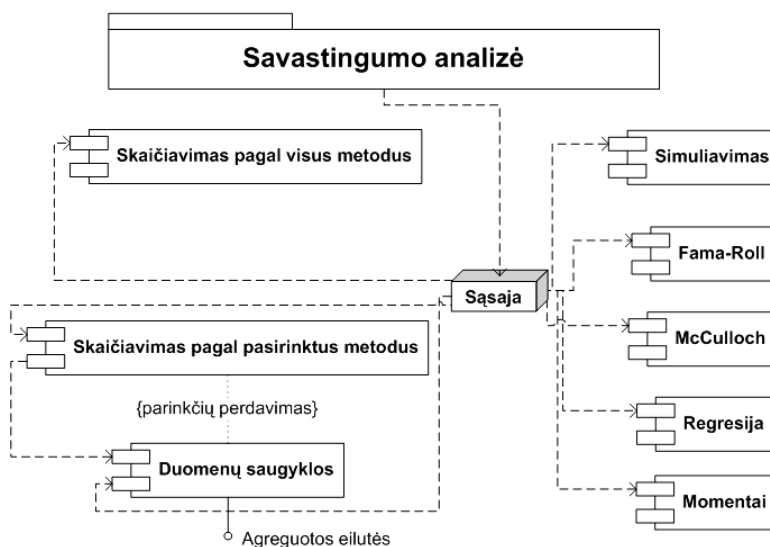
Pastebėta, kad vertinant suformuotas laiko eilutes tiek tinklo mazge matuotam srautui, tiek sintetiniam (simuliuotam) srautui, pagal skirtingas vertinimo metodikas gautieji įverčiai skiriasi. Reikia pažymėti, kad:

1. skaičiuojant pagal *dispersijos*, *R/S* statistikos, *absoliutinių momentų* ir *liekanų dispersijos* metodus papildomai apskaičiuojamas koreliacijos koeficientas;
2. skaičiuojant pagal *Abby-Veich* ir *sumažinimo* metodus papildomai apskaičiuojamas pasikliautasis intervalas su 95% patikimumu;
3. programa, įvertindama suformuotų laiko eilučių savybes, pagal kai kuriuos vertinimo metodus gali nebrėžti grafikų.

$\alpha$ -stabiliųjų skirstinių parametrų (plačiau 3.3.2 skyriuje) vertinimui, panaudojant Java programavimo kalbą, sukūriau programinių modulių biblioteka *SSE* (angl. *Self-similarity estimator* – toliau *SSE*), kurioje panaudoti keturi metodai: *Fama-Roll*, *McCulloch*, *Regresijos*, *Momentų*. 9 paveiksle matome išvardintų savastingumo tyrimo laiko eilutėse metodų modulių ir kitų bibliotekos modulių tarpusavio sąveikos schema. Papildomiems moduliams numatytos šios funkcijos:

- agreguotų laiko eilučių analizė pagal visus metodus (žr. 9 pav. Skaičiavimas pagal visus metodus),
- agreguotų laiko eilučių analizė pagal pasirinktus metodus (žr. 9 pav. Skaičiavimas pagal pasirinktus metodus),
- duomenų saugyklos (žr. 9 pav. Duomenų saugyklos);

- laiko eilutės simuliavimas pagal (Janicki, Weron 2000) siūlomą metodiką (žr. 9 pav. Simuliavimas).



9 pav. SSE komponentų sąveikos schema

SSE biblioteka parengta taip, kad galima analizuoti vieną arba keletą eilučių iš karto, arba parengti eilučių analizės vykdymo aprašą tekstiniame faile ir, remiantis juo, vykdyti faile aprašytų eilučių ir parametrų analizę pagal nurodytus savastingumo vertinimo metodus. Bibliotekos išvesties modulis gautus rezultatus gali išsaugoti viename faile arba, pagal nuorodas analizės grafiko faile, įrašyti į skirtingus failus, arba eilutės savastingumo įverčius pateikti ekrane.

#### 4.3. Analizės metodų testavimas

Naudojamų metodų tinkamumas analizei testuotas kompiuterinio imitavimo būdu. Imitavimo sistema modeliuota panaudojant *ON-OFF* tinklo srauto modelį. Remiantis šiuo modeliu bet koks paketinio kompiuterių tinklo srauto resursas turi dvi stacionarias būsenas: 1 – informaciniai paketai generuojami ir siunčiami iš resurso ir paketai nėra generuojami ir siunčiami (plačiau modelis aptartas 2.2.2 skyriuje). Pradiniai imituojamos sistemos duomenys aprašyti panaudojant  $\alpha$ -stabiliųjų procesų, pasižyminčių *LRD* ir turinčių sunkias uodegas, keturis stabilumo parametrus. Jų reikšmės parinktos, atsižvelgiant į realaus tinklo mazgo laiko eilučių analizės rezultatus, t. y.

imituojamas *LRD* srautas su sunkiomis uodegomis, kai  $\alpha=1,8$  ( $H=0,56$ ),  $\beta=0$ ,  $\sigma=1$ ,  $\mu=0$ . Eilutės narių skaičius  $n=1000000$ , simuliuojamų eilučių kiekis  $N=1000$  (plačiau apie matavimo rezultatus 4.4 skyriuje). Imitavimui panaudota *SSE* biblioteka (plačiau 4.2 skyriuje). Sintetinės laiko eilutės išsaugotos ir panaudotos skaičiuojant savastingumą su 4.2 skyriuje aprašytomis programomis. Imitavimo rezultatai palyginti su rezultatais gautais vertinant tinklo mazgo  $M$  srautą (Sheluhin, Smolskiy, Osin 2007).

Atsitiktinių srautų imitavimui naudotos standartinių stabilijų dydžių  $S_\alpha(\beta,1,0)$ , kai  $\alpha \neq 1$  generavimo formulės (Janicki, Weron 2000):

$$X = \mu + \sigma \cdot S_{\alpha\beta} \frac{\sin\{\alpha(V + B_{\alpha\beta})\}}{\{\cos(V)\}^{1/\alpha}} \left[ \frac{\cos\{V - \alpha(V + B_{\alpha\beta})\}}{W} \right]^{(1-\alpha)/\alpha},$$

$$\text{čia } B_{\alpha\beta} = \frac{\arctan\left(\beta \tan\frac{\pi\alpha}{2}\right)}{\alpha}, \quad S_{\alpha\beta} = \left\{1 + \beta^2 \tan^2\left(\frac{\pi\alpha}{2}\right)\right\}^{1/(2\alpha)}, \text{ o } V \text{ tolygiai}$$

pasiskirstęs intervale  $\left(-\frac{\pi}{2}, \frac{\pi}{2}\right)$  ir yra nepriklausomas eksponentinis kintamasis

$W$  su vidurkiu 1.

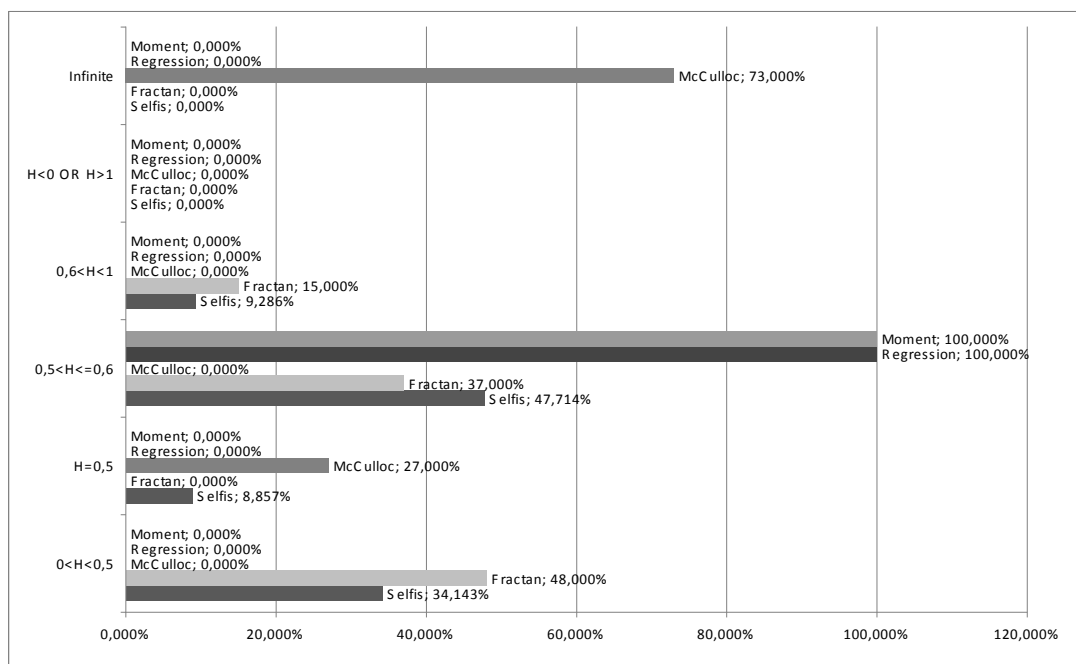
Sintetinės laiko eilutės buvo vertintos *Fractan* 4.4, *Selfis* 0.1b bei *SSE* naudojamais savastingumo vertinimo metodais. Siekiant tiksliau įvertinti simuliacijos rezultatus, savastingumo įverčių aibė išskaidyta į poaibius pagal *Hurst*'o koeficiento reikšmes:

- $H < 0$  arba  $H \geq 1$  – eilutėse nėra savastingumo;
- $0 < H < 0,5$  – eilutės pasižymi *SRD*;
- $H = 0,5$  – eilutės nustatytas Gauso triukšmas (angl. *Gaussian Noise*);
- $0,5 < H \leq 0,6$  – eilutės pasižymi nedideliu *LRD*;
- $0,6 < H < 1$  – eilutės pasižymi stipriu *LRD*;
- *Infinite* – skaičiavimo rezultatai neapibrėžti.

10 paveiksle pavaizduotas grafikas, vaizduojantis sintetinio srauto savastingumo skaičiavimo rezultatų pasiskirstymą. *SSE momentų* bei *regresijos*

metodais apskaičiuoti *Hurst*'o koeficientai 100% atitiko imituojamam srautui parinktas koeficientų reikšmes, *McCulloch* metodu net 73% reikšmių buvo neapibrėžtos, o likusios priskirtos *Gauso* triukšmo eilutėms. *Fractan* tik 37% reikšmių atitiko imituojamam srautui parinktas koeficientų reikšmes, o 15% atveju nustatyta savastingumas su stipria *LRD*. *Selfis* 47.714% atitiko generavimo rezultatus, 9.286% nustatyta savastingumas su stipria *LRD*, o likusios eilutėms priskirtos turinčioms *SRD*.

Pagal imitavimo rezultatų analizę galima teigti, kad savastingumo analizei, panaudojant sintetinį srautą, tinkamiausi *Momentų* ir *Regresijos* metodai, o *McCulloch* metodu gauti rezultatai visiškai neatitinka pirminių simuliacijos parametrų. *Selfis*, lyginant su *Fractan*, tiksliau skaičiavo sintetinio srauto savastingumo įverčius.

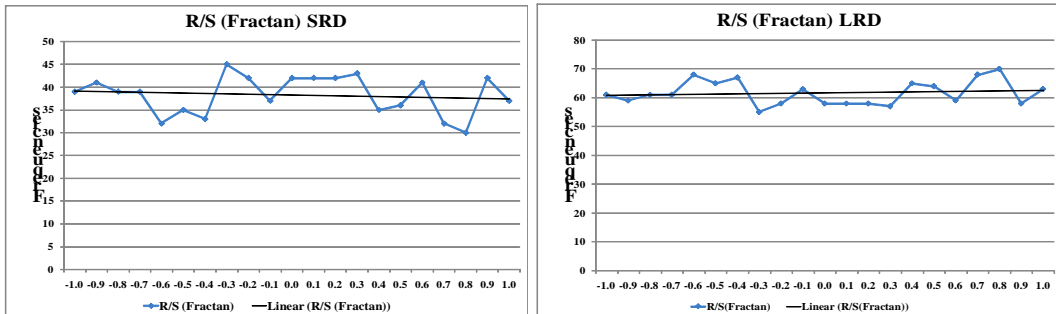


**10 pav.** Savastingumo įverčių su *Fractan* 4.4, *Selfis* 0.1b ir *SSE* pasiskirstymas

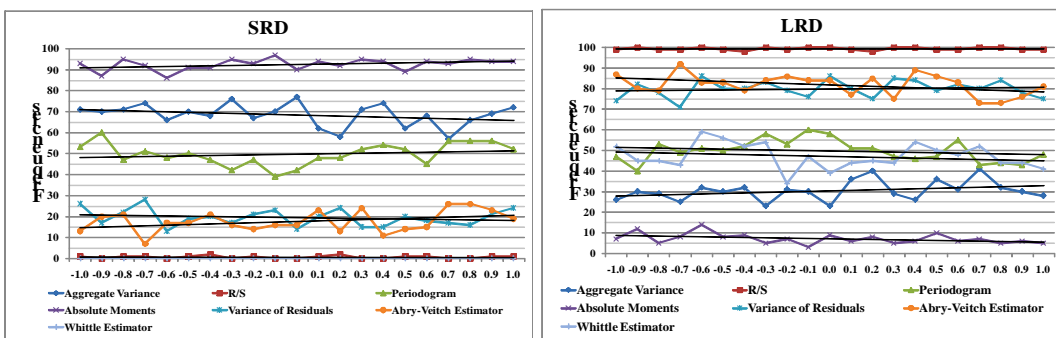
Siekiant įvertinti asimetrijos koeficiento  $\beta$  įtaką skaičiavimo rezultatams gautiems su tiriamąja programine įranga, buvo sugeneruota po 1000 eilučių  $S_{1,8}(1, \beta, 0)$ , kai  $\beta \in [-1, 1], \Delta\beta = 0,1, n = 1000000$ . Brėžiant grafikus buvo išskirti *Hurst*'o reikšmių intervalai: 1 – reikšmių intervalas nusakantis *LRD*, 2 – reikšmių intervalas nusakantis *SRD*, 3 – reikšmių intervalas nusakantis



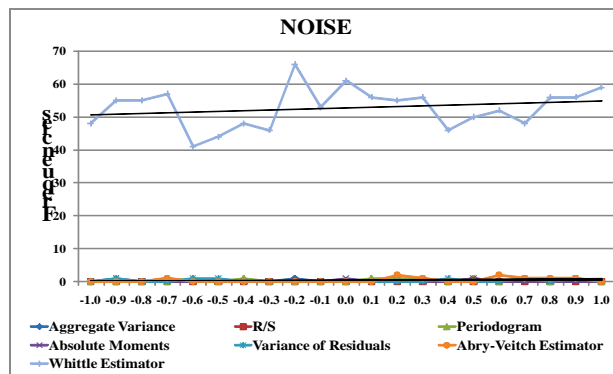
*Gauso* triukšmą. 11 paveiksle pateikiami *Fractan* gauti skaičiavimo rezultatai, 12 paveiksle ir 13 paveiksle – *Selfis* ir 14 paveiksle – *SSE* gauti rezultatai. Ordinačių ašyje atidedama *Hurst*'o koeficientų kiekis (dažnis), patenkančių į parinktą intervalą, o absčių asimetrijos koeficiento pokytis.



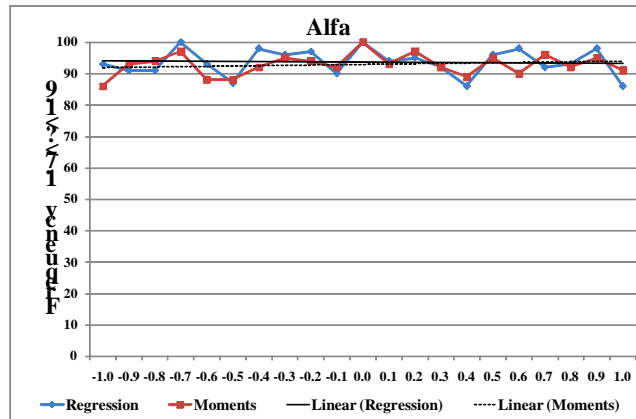
**11 pav.** Tinklo srauto SRD (kairėje) ir LRD (dešinėje) įverčių programa *Fractan* pasiskirstymas, kai  $\beta \in [-1,1], \Delta\beta = 0,1$



**12 pav.** *Selfis* *Hurst*'o reikšmių pasiskirstymas, kintant  $\beta$  (kairėje – *SRD*, dešinėje – *LRD*)



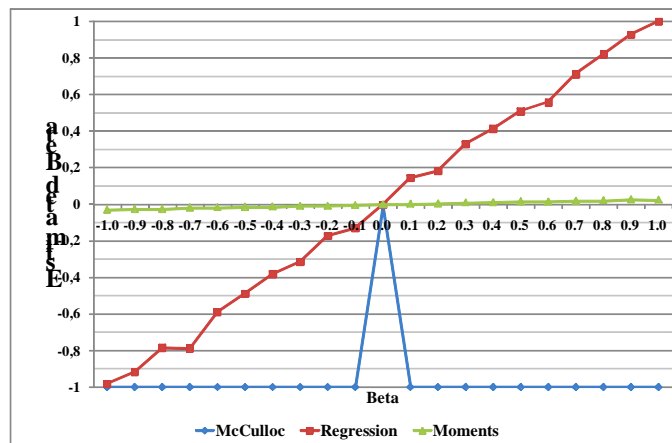
**13 pav.** *Selfis* *Hurst*'o reikšmių pasiskirstymas, kintant  $\beta$ , kai rezultatas rodo triukšmą



14 pav. SSE programos  $\alpha$  koeficiento reikšmių pasiskirstymas, kintant  $\beta$

Kaip matyti 11 pav, 12 pav, 13 pav ir 14 paveiksluose pateikiamuose grafikuose  $\beta$  koeficiento pokytis faktiškai neįtakoja apskaičiuotos savastingumo koeficiento reikšmės. Didesni nuokrypiai stebimi kai gauti koeficientai rodo, kad eilutės yra *Gauso* triukšmas (žr. 14 pav.). Gauti rezultatai visiškai neatitinka sintetinių laiko eilučių simuliuojamų pradinių parametru, todėl šie rezultatai nėra reikšmingi.

Apibendrinant asimetrijos koeficiento  $\beta$  simuliuotoms eilutėms tyrimo rezultatus, galima teigti, kad eilutėms pasižyminčioms savastingumu jo pokytis turi minimalią įtaką *Hurst*'o koeficiento bei stabilumo parametro  $\alpha$  skaičiavimui.



15 pav. SSE  $\beta$  reikšmių pasiskirstymas

Įvertintas asimetrijos koeficiento skaičiavimo tikslumas naudojant *SSE* savastingumo vertinimo metodus. Pagal gautus rezultatus nubrėžtas skirtingais metodais apskaičiuotų  $\beta$  reikšmių grafikas (žr. 15 pav.), kai simuliuojamose

eilutėse fiksuotas  $\alpha=1,8$  ( $H=0,56$ ),  $\sigma=1$ ,  $\mu=0$ , o  $\beta \in [-1,1]$ ,  $\Delta\beta = 0,1$ . Ordinačių ašyje atidėta apskaičiuota  $\beta$  reikšmė, o abscisių – simuliuota. Nustatyta, kad tiksliausi asimetrijos koeficiento  $\beta$  įverčiai gaunami *regresijos* metodu.

Pagal imitavimo tyrimo rezultatus galima teigti, kad sintetinio srauto savastingumo įverčiai tiksliausiai apskaičiuoti taikant *regresijos* metodą.

#### 4.4. Kompiuterių tinklo apkrovos laiko eilučių analizės rezultatai

Agreguotų laiko eilučių savybės įvertintos, panaudojant chaoso teorijos teiginius (plačiau žr. 3.2 skyriuje) bei programą *Fractan* (plačiau žr. 4.1 skyrių).

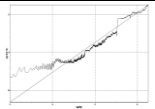
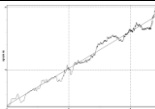
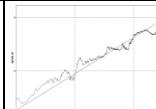
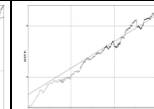
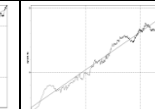
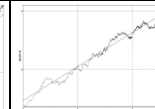
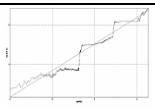
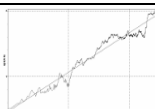
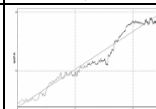
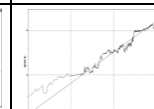
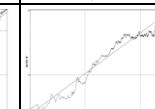
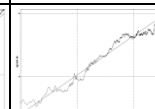
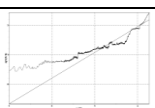
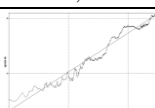
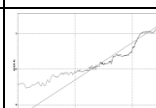
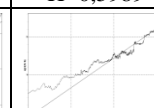
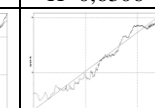
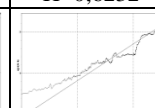
Žinoma, kad *Hurst*'o koeficientą galima įvertinti grafiškai tam panaudojant formulę:  $\log\left(M\left[\frac{R(n)}{S(n)}\right]\right) = H(\log(n) + \log(c)), n \rightarrow \infty$ . Abscisių ašyje

atidedamos  $\log(n)$  reikšmės, o ordinačių –  $\log\left(M\left[\frac{R(n)}{S(n)}\right]\right)$ . Analizuojant gautus grafikus buvo pastebėta, kad *Hurst*'o koeficientui artėjant prie vieneto grafike mažėja triukšmų, t.y. kreivė tolydėja (Feder 1988, Sheluhin, Smolskiy, Osin 2007).

*Hurst*'o koeficiento priklausomybė nuo laiko pavaizduota 16 paveiksle. Analizuojant agreguotų laiko eilučių *Hurst*'o koeficiento grafines išraiškas nustatyta, kad absoliuti grafikų dauguma tolydėja didėjant laiko intervalui, vadinasi agreguotos laiko eilutės aprašo dinaminį procesą su *LRD* (Sheluhin, Smolskiy, Osin 2007). 1 priedo, 1 lentelėje ir 1 priedo, 2 lentelėje parodyta, kad *Hurst*'o koeficientas kinta nuo 0,61 iki 0,79. Vadinasi agreguotomis eilutėmis aprašomas paketiniu kompiuterių tinklu perduodamų duomenų srautas yra persistentinis procesas su *LRD*. Galima teigti, kad apie du trečdalius gautų eilučių pasižymi savastingumu su *LRD*. Įvertinus programa *Fractan* gautus *Hurst*'o koeficiento analizės rezultatus galima teigti:

1. savastingumo įverčiai nepriklauso (sutapo 98% rezultatų) nuo eilučių agregavimo būdo;

2. savastingumo įverčiai nepriklauso nuo pasirinkto laiko agregavimo intervalo dydžio  $\Delta t \in [100\text{ms}, 500\text{ms}, 1000\text{ms}]$ , t. y. sutampa 89.4% apskaičiuotų rezultatų;
3. savastingumo įverčiai nepriklauso nuo duomenų srauto (buvo pasirinktas minimalus, vidutinis, maksimalus) ir sutapo 88% gautų rezultatų.

$x_t^\Sigma$	$\Delta t^\Sigma = 100\text{ms}$	$\Delta t^\Sigma = 500\text{ms}$	$\Delta t^\Sigma = 1\text{s}$	$\Delta t^\Delta = 100\text{ms}$	$\Delta t^\Delta = 500\text{ms}$	$\Delta t^\Delta = 1\text{s}$
$x_t^{\ominus_{\min}}$	 H=0,7870	 H=0,5891	 H=0,6794	 H=0,5190	 H=0,5888	 H=0,5820
$x_t^{\ominus_\Delta}$	 H=0,6699	 H=0,5993	 H=0,7093	 H=0,5989	 H=0,6306	 H=0,6232
$x_t^{\ominus_{\max}}$	 H=0,8736	 H=0,7079	 H=1,0524	 H=0,7819	 H=0,7059	 H=0,8885

**16 pav.** Charakteringi *Hurst*'o eksponentės grafikai

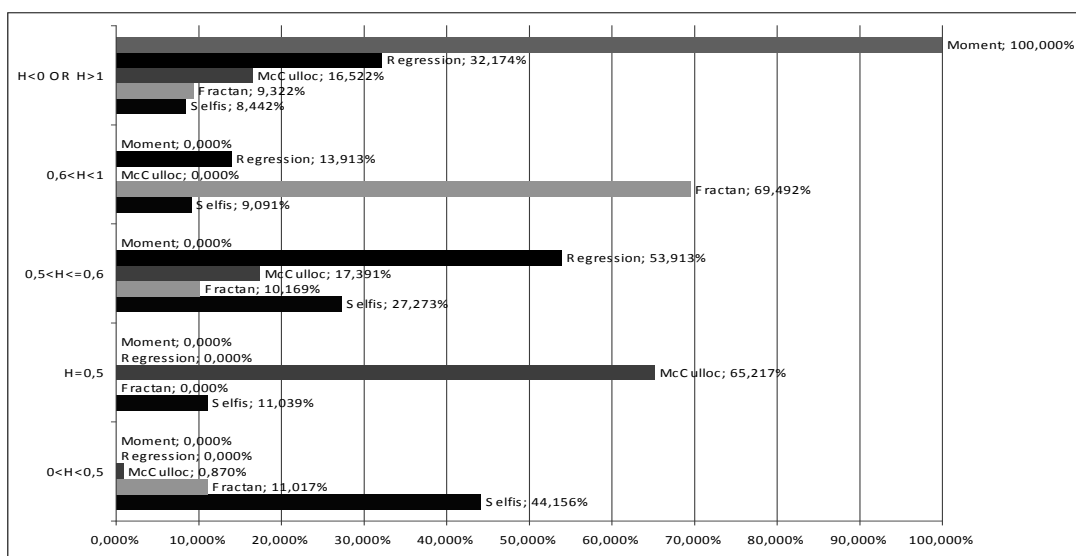
Aukščiau įvardinti teiginiai rodo, kad analizuojamos agreguotos eilutės aprašo savastingąjį procesą pasižyminti *SRD* arba *LRD*, o analizuojamos paketinio kompiuterių tinklo srauto laiko eilutės pasižymi fraktališkumu.

2 priedo, 1 lentelėje matyti, kad apie 43% matuotų eilučių pasižymi *LRD*, o *Hurst*'o koeficiento vidutinė reikšmė 0,56 rodo silpną priklausomybę, apie 47% eilučių pasižymi *SRD*, kai *Hurst*'o koeficiento vidutinė reikšmė 0,28 rodo vidutinišką priklausomybę. Atsižvelgiant į gautus rezultatus galima teigti, kad su *Selfis* nustatyta, kad didžioji dauguma eilučių pasižymi trumpalaikė arba ilgalaikė atmintimi, kas patvirtina šių eilučių savastingumą. Galima pastebėti, kad gauti rezultatai nepriklauso nuo eilučių agregavimo metodo bei nuo laiko intervalo, kas taip pat rodo jų savastingumą. Analizė su *Selfis* programa rodo, kad:

1. gauti rezultatai nepriklauso (sutapo 98% rezultatų) nuo eilučių agregavimo būdo (taikyti sumos ir vidurkių metodai);

- nustatyta labai silpna gautų rezultatų priklausomybė nuo pasirinkto laiko intervalo dydžio  $\Delta t \in [100\text{ms}, 500\text{ms}, 1\text{s}]$  (sutampa 65.63% rezultatų);
- nustatyta labai silpna gautų rezultatų priklausomybė nuo duomenų srauto, kai pasirenkamas minimalus, vidutinis, maksimalus srautas (sutapo 63.5% rezultatų).

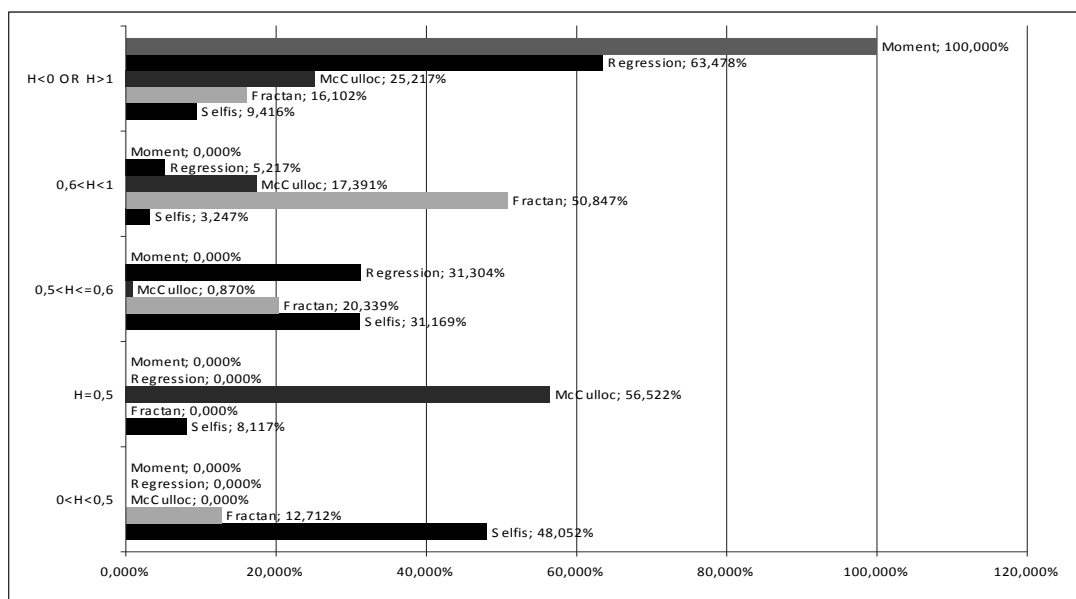
Galima teigti, kad tiriamos eilutės aprašo savastingąjį procesą pasižyminti *SRD* arba *LRD*. (Samorodnitsky 2006a) įrodė, kad tarp *Hurst*'o koeficiento ir parametro alfa yra nustatytas santykis  $H=1/\alpha$ , kai  $1 < \alpha < 2$ ,  $\beta=0$ . Siekiant tiksliau įvertinti gautus skaičiavimo rezultatus agreguotų laiko eilučių *Hurst*'o koeficiento reikšmės išskaidytos į penkis intervalus:  $0 < H$  arba  $H \geq 1.0$  – negalima įvertinti laiko eilutės savastingumą,  $0 < H < 0.5$  – eilutė aprašo savastingąjį procesą su *SRD*,  $H=0.5$  – eilutė aprašo *Gauso* triukšmą,  $0.5 < H \leq 0.6$  – eilutė aprašo savastingąjį procesą su silpnai išreikšta *LRD*,  $0.6 < H < 1.0$  – eilutė aprašo savastingąjį procesą su *LRD*.



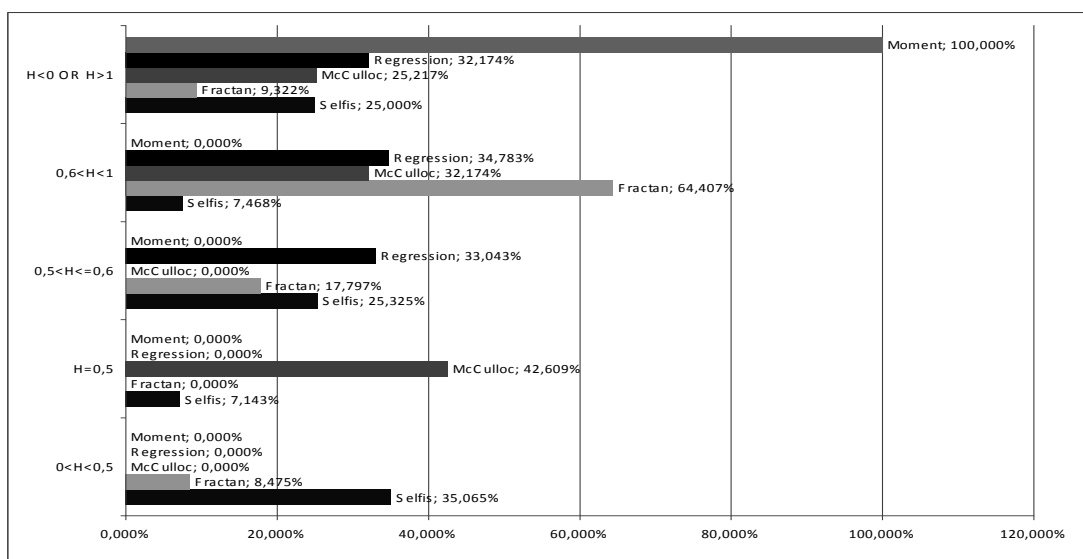
**17 pav.** *Hurst*'o koeficiento reikšmių įverčiai, kai naudotas agregavimo intervalas 100ms

Siekiant išryškinti sukurtoje programoje taikomų metodų tinkamumą agreguotų laiko eilučių vertinimui, kiekvienu metodu gauti rezultatai vaizduojami grafike, o *Fractan* ir *Selfis* programomis gauti skaičiavimo rezultatai apibendrinami, pritaikant procentinius įverčius. Gauti įverčiai

iškaidyti pagal laiko intervalus  $\Delta t \in [100\text{ms}, 500\text{ms}, 1000\text{ms}]$  pavaizduoti 17 pav, 18 pav ir 19 paveiksluose.



18 pav. Hurst'o koeficiento reikšmių įverčiai, kai naudotas agregavimo intervalas 500ms



19 pav. Hurst'o koeficiento reikšmių įverčiai, kai naudotas agregavimo intervalas 1000ms

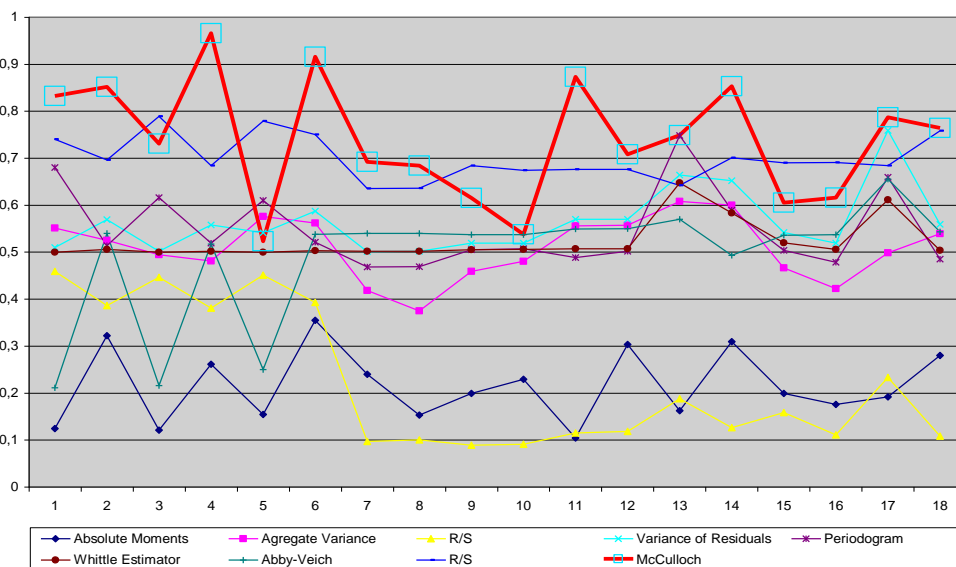
Iš grafikų galima teigti, kad momentų metodu gauti įverčiai yra netikslūs, nes jie visiškai neatitinka rezultatų gautų vertinant eilutes kitomis programomis. Regresijos ir McCulloch metodais gauti rezultatai yra artimiausi

su *Selfis* bei *Fractan* apskaičiuotiems rezultatams. Reikėtų pažymėti, kad programa *Fractan*, lyginant su kitomis skaičiavimo priemonėmis, toms pačioms laiko eilutėms daugiau nei du kartus dažniau yra apskaičiuojami *Hurst*’o koeficientai aprašantys savastingąjį procesą su *LRD*.

Analitinis skaičiavimo rezultatų įvertinimas pateiktas 4.1 lentelėje, čia pateikiami procentiniai *Hurst*’o koeficientų įverčiai, juos išskiriant į tris dalis: 1 – savastingieji, turintys *SRD*, 2– savastingieji, turintys *LRD*, 3 – *Gauso* triukšmas. Matome, kad *McCulloch* metodu daugiau nei pusėje suformuotų laiko eilučių nustatytas *Gauso* triukšmas, o momentų metodu gauti rezultatai viršijo numatytąsias ribas. Tiksliausiai agreguotų laiko eilučių skaičiavimo rezultatai atitiko koeficientus gautus standartinėmis programomis taikant *Regresijos* metodą.

4.1 lentelė. Laiko eilučių savastingumo procentiniai įverčiai, naudojant skirtingas programas bei metodus

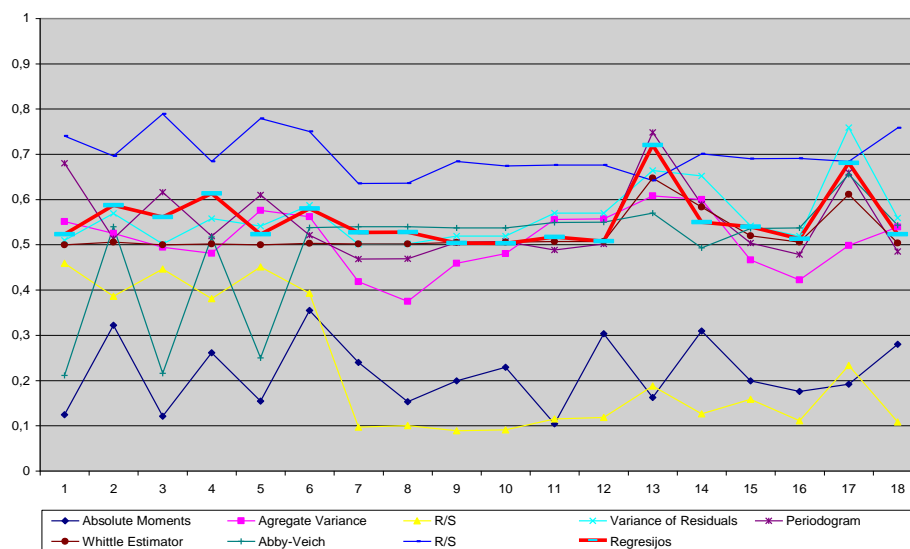
Eilutės savybė	<i>Selfis</i>	<i>Fractan</i>	<i>McCulloch</i>	<i>Regresijos</i>	<i>Momentų</i>
<i>SRD</i>	47,06%	11,87%	0,290%	0,000%	0,000%
<i>LRD</i>	<b>43,94%</b>	<b>81,08%</b>	<b>22,609%</b>	<b>68,116%</b>	0,000%
<i>Gaussian Noise</i>	8,766%	0,000%	54,783%	0,000%	0,000%



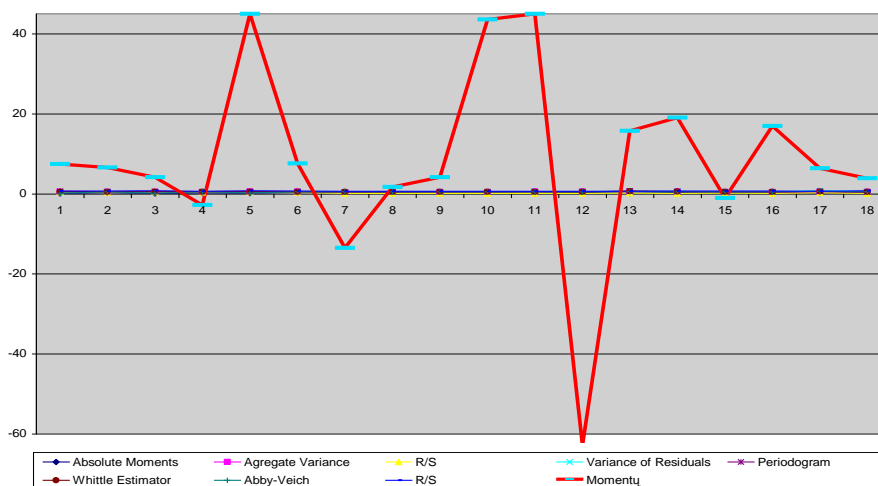
20 pav. *McCulloch* koreliacijos matrica

Siekiant įvertinti atitinkamų eilučių koeficientus, gautus taikant skirtingus skaičiavimo metodus programomis *Fractan* ir *Selfis* bei palyginti juos su *SSE*

programos rezultatais, buvo apskaičiuotos koreliacinės Pirsono sutapimų matricos. Pagal gautus rezultatus nubrėžti grafikai. Kontroliniai duomenys vaizduojami storesne linija. Kaip matyti 20 paveiksle *McCulloch* metodu gauti skaičiavimo rezultatai geriausiai atitiko R/S metodu gautus skaičiavimo rezultatus programa *Fractan*.



21 pav. Regresijos koreliacijos matrica



22 pav. Momentu koreliacijos matrica

21 paveikslėlyje *regresijos* skaičiavimo rezultatai geriausiai atitinka *Aggregate Variance*, *Variance of Residuals*, *Periodogram* ir *White Estimator* metodais gautus skaičiavimo rezultatus. Iš dalies atitinka *Abby-Weight* skaičiavimo metodo rezultatus. 22 paveikslėlyje pavaizduoti *Momentu* metodu



gauti skaičiavimai rodo labai didelius skirtumus, viršijančius *Hurst'o* koeficiento numatytas ribas.

Pagal analizės rezultatus galima teigti, kad iš programoje *SSE* naudojamų skaičiavimo metodų tinkamiausias kompiuterių tinklų informacinių srautų savastingumo analizei realiu laiku yra *SSE* programoje realizuota *Regresijos* metodas.

**4.2 lentelė.**  $\beta$  įverčiai *regresijos* metodu

$\Delta t$	Average	Standard Deviation
100ms	$9.14436 \cdot 10^{-5}$	0.000207
500ms	$3.29 \cdot 10^{-5}$	0.006621
1000ms	0.001368519	0.006862

Simuliavimo rezultatų analizė parodė, kad geriausiai asimetrijos koeficientą vertina *regresijos* metodas, todėl programa *SSE* įvertintos šio koeficiento reikšmėms tinklo srautų informacinėms eilutėms skirtingais laiko intervalais:  $\Delta t \in [100\text{ms}, 500\text{ms}, 1000\text{ms}]$ . Gauti rezultatai rodo (žr. 4.2. lentelę), kad kompiuterių tinklo informacinių srautų asimetrijos koeficientą  $\beta$  galima laikyti lygiu nuliui.

#### **4.5. Skyriaus išvados**

Pasinaudojus disertacijoje sudaryta paketinio tinklo srauto savastingumo matavimo metodika, keliais metodais įvertintas realaus tinklo mazgo (1 Gbps) srauto savastingumas, naudojant kelis skirtingus programinius įrankius.

Yra žinoma, kad diskretiems laiko procesams savastingumas tiksliau įvertinamas, kai procesui pritaikomas agregavimas ir/arba masteliavimas, kurie yra invariantiški skirstinio savybių pokyčiams. Paketinio kompiuterių tinklo srauto laiko eilučių fraktališkumo savybėms įvertinti naudotini judančių vidurkių glotninimo ir perduotų duomenų srauto per laiko intervalą  $\Delta t$  sumos metodai, nes jie atitinka tikslaus antros eilės savastingumo reikalavimus keliamus šios eilutės kovariacinei struktūrai. Darbe sudarytas perindeksavimo algoritmas, kuris paspartina rekurentinį agreguotų laiko eilučių formavimą realiu laiku.

Simuliuotų laiko eilučių tyrimai su *Fractan*, *Selfis* ir *SSE* parodė, kad asimetrijos koeficiento  $\beta$  pokytis laiko eilutėms pasižyminčioms savastingumu turi minimalią įtaką *Hurst*'o koeficiento bei stabilumo parametro  $\alpha$  reikšmei. Imitavimo rezultatų, gautų su *Fractan*, *Selfis* ir *SSE*, lyginamoji analizė parodė, kad sintetinio srauto savastingumo įverčiai tiksliausiai apskaičiuojami taikant *regresijos* metodą. Realaus tinklo srauto agreguotų eilučių tyrimo rezultatai parodė, kad procesas aprašomas laiko eilutėmis yra persistentinis procesas su *LRD*, kai *Hurst*'o koeficiento reikšmė kinta nuo 0,61 iki 0,79. Be to, ištirtas kompiuterių tinklo srautas pasižymėjo fraktališkumu.

## 5 skyrius. Fraktalinių procesų imitavimas

Imitacinis fraktalinių procesų modeliavimas tai realios sistemos modelio konstravimas ir eksperimentavimas su juo, siekiant rasti fraktalinių paketinio tinklo procesų aptarnavimą užtikrinančias strategijas, esant tam tikriems iš anksto žinomiems tinklo mazgų apribojimams (Shannon 1975). Imituojant fraktalinius procesus aprašomas tikimybinis sistemos modelis, keičiantis analitinės sistemos būsenos ir rezultatų priklausomybę į analogiškas tikimybinės tinklo srauto savybių ir tinklo mazgų parametrų tarpusavio sąryšių charakteristikas (Willinger, Paxson, Taqqu 1998). Plačiau paketinio kompiuterių tinklo fraktalinių procesų modeliai apžvelgti 2.2.2 skyriuje. Fraktalinių procesų imitavimui plačiai taikomi aptarnavimo teorijos matematiniai modeliai (Хинчин 1963), kurie plačiau apžvelgti 2.2.3 skyriuje. V. B. Iversen (2011) knygoje apie kompiuterių tinklo srauto modeliavimą ir imitavimą taiko aptarnavimo teorijos idėjas aprašydamas paketinio tinklo srauto ir paketų atvykimo į sistemą matematinius skirstinius, paketų praradimų matematinius įverčius, tinklo perkrovų vertinimo metodus, taikomosios eilių teorijos panaudojimo aspektus bei daug paslaugų teikiančių sistemų modeliavimą.

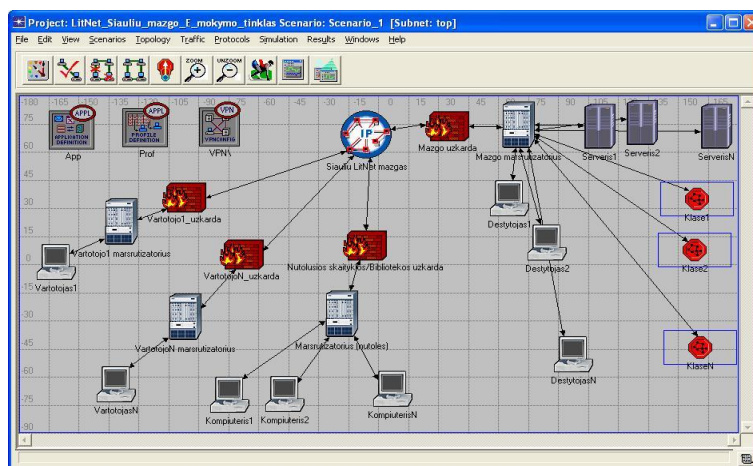
Savastingojo *LRD* tinklo srauto su sunkiomis uodegomis imitavimui dažnai pasitelkiami multifraktalų modeliai (Gao, Rubin 2001a, Gao, Rubin 2001b, Liangxiu, Zhiwei, Chunbo, Chuanshan 2002, Lu, Ke, Huijing 2002, Maulik, Resnick 2003, Shang, Lu, Kamae 2008, Vieira, Bianchi, Lee 2010). Tinklo srauto aprašymui naudojami skaičiuojami tinklo modeliai, kurie angliškai pavadinti *Network Calculus* (Xie, Jiang 2009). Kompiuterių tinklo procesų aprašymui naudojamos *Min-Plus* ir *Max-Plus* specializuotos algebros (Heidergott, Olsder, Woude 2006, Xiaolong, Geyong 2009). (Zhu, Fu 2008) tinklo srauto imitavimui siūlo naudoti dinامينius modelius. (Abdelnaser 1997, Dufour, Kurz-Kim 2010) parodė, kad savastingojo kompiuterių tinklo paketinio srauto generavimui galima sėkmingai taikyti Pareto skirstinį, kurį

naudojant sukuriamos nepriklausomas ir tolygiai pasiskirsčiusias sekas, pasižyminčios pliūpsniškumu. Moksliniai tyrimai apie šiuolaikinių kompiuterių tinklų srautų fraktalinių procesų imitavimo modelius plačiau apžvelgti 1.1 skyriuje.

### 5.1. Fraktalinių procesų imitavimo programinė įranga

Fraktalinių procesų imitaciniam modeliavimui reikalingos programų bibliotekos, kuriose galima tinklo srautą aprašyti pritaikant matematinio modeliavimo metodus bei statistiškai įvertinti gautus rezultatus. Programų bibliotekos dažniausiai turi komponentus, kurie tinklo srautą generuoja pagal vartotojo nurodytus parametrus. Galima rasti atviro kodo ir komercinių produktų, kurie skirti tinklo srauto imitavimui ir rezultatų tyrimui. Dažniausiai naudojamas komercinis produktas yra *Opnet* (Opnet 2012). Juo galima imituoti ir tirti daugelio standartų kompiuterių tinklus. *Opnet IT Guru* su ribotomis galimybėmis galima nemokamai naudoti mokymo reikmėms. Tinklo srauto imitacinio modelio kūrimas programa *Opnet*, kaip ir daugelyje tokio tipo programų, vykdomas keturiais etapais:

1. Tinklo modelio sukūrimas;
2. Reikiamos statistikos parinkimas;
3. Tinklo srauto imitacinis simuliacijavimas;
4. Rezultatų peržiūra ir analizė.



**23 pav.** Šiaulių universiteto LitNet tinklo mazgo modelis, parengtas programa *Opnet*

Tinklo srauto imitacinis projektas rengiamas, naudojantis modelių rengimo biblioteka (žr. 23 pav.). Čia aprašoma analizuojamo tinklo topologija, komponentai bei jų savybės, adresavimo būdai, naudojamos paslaugos ir srauto savybės. Modelyje nurodomi tinklo objektai, kuriuose bus surenkama informacija apie ateinantį/išeinantį tinklo srautą. Po to nurodomi tinklo simuliacijos parametrai, aprašoma ataskaita bei kiti tyrimo parametrai.

Programa dažnai naudojama tinklo srauto imitaciniams modeliavimui (Karasaridis, Hatzinakos 2001, Kajackas, Pavilanskas, Vindašius 2007, Kulikov, Petersons 2009 ir kiti). *Opnet* gauti rezultatai lyginami su rezultatais gautais kitomis tinklo modeliavimo programomis (Malowidzki 2004 ir kiti). Tiriant tinklo srautą *Opnet* programa, sumodeliuotame tinkle kiekvieną kartą srautas yra generuojamas pagal nurodytus parametrus, o tą patį srautą keletą kartų aptarnauti dinamiškai keičiant sumodeliuoto tinklo mazgų parametrus negalima. Tokio uždavinio sprendimui reikia projektuoti kelis modelius su skirtingais mazgų parametrais, kiekviename iš jų simuliuoti ir analizuoti aptarnautą srautą.

*ns-2* ir *ns-3* (*ns-2* 2012, *ns3* 2012) platinamas pagal *GNU* licenciją, naudoja *Unix* arba *Linux* operacinių sistemų platformą, ją naudojant galima analizuoti kompiuterių tinklo srautus, tinklo mazgų parametrus ir jų tarpusavio sąryšius.

*NetFPGA* (*NetFPGA* 2012) ir panašūs į šią programą simulatoriai gali būti integruojami į realiai veikiančius tinklo mazgų įrenginius ir naudojamas jų tobulinimo, savybių gerinimo tyrimams.

Galima rasti nemažai tinklo simuliacijos programų, kurios skirtos tinklo modelių rengimui, tinklo procesų valdymo įgūdžių ugdymui, tačiau moksliniams tyrimams jos turi per mažai funkcijų. Žinomiausios iš jų yra *Cisco Packet Tracer* (*Cisco* 2012), *NetSim* (*NetSim* 2012), *GNS3* (*GNS3* 2012). Šios programos yra tinkamesnės mokymo reikmėms.

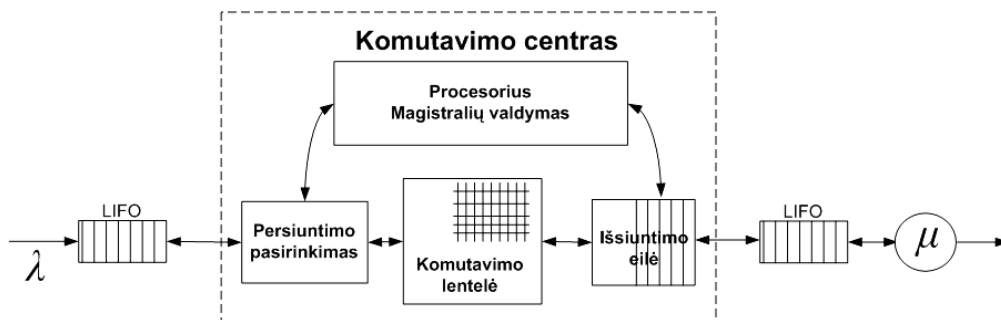
Fraktalinių procesų imitaciniam modeliavimui buvo suprojektuota ir sukurta tinklo srauto modeliavimo sistema *MulNodSimSys* (angl. *Multichannel Node Simulation System*, sistemos kūrėjas L. Kaklauskas) turi šias funkcijas:

- generuoja *Puasono* arba savastingąjį su *LRD* ir sunkiomis uodegomis tinklo srautą,
- tinklo mazgo savybės aprašomos naudojant dinaminę biblioteką, kurioje saugoma mazgo savybių pokyčių aibė,
- sugeneruotas tinklo srautas parengtame tinklo modelyje aptarnaujamas keletą kartų, dinamiškai keičiant tinklo mazgo savybes pagal dinaminės bibliotekos įrašus,
- tinklo srauto aptarnavimo rezultatai bei tinklo objektų savybių pokyčiai išsaugomi duomenų saugyklose tolesnei rezultatų analizei.

Matematinis sukurtos sistemos modelis išsamiau aprašytas 5.3, 5.6 ir 5.7 skyriuose.

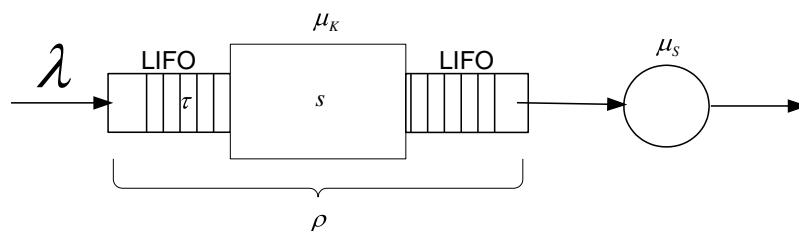
## 5.2. Vieno kanalo tinklo mazgo aptarnavimo modelis

Kompiuterių tinklą galima nagrinėti kaip tinklą sudarytą iš atskirų mazgų, atliekančių informacinių srautų apdorojimo ir paskirstymo funkcijas, ir tinklo klientų, inicijuojančių bei naudojančių šiuos srautus (plačiau 2.1 skyriuje). 24 paveiksle pateikta *IEEE-802.3* tinklo komutatoriaus, aptarnaujančio nuotolinių studijų tinklo serverius, schema. *Ethernet* standartas sukurtas 1972 m., o 1980 m. buvo suformuluoti reikalavimai *IEEE-802.3* (Kaklauskas 2003). *Ethernet* kompiuterių tinklo srautai plačiai analizuojami mokslinėje literatūroje (Leland, Taqqu, Willinger, Wilson 1994, Ge, Zhu G., Zhu Y. 2003, Samorodnitsky 2006, Sheluhin, Smolskiy, Osin 2007, Kaklauskas, Sakalauskas 2008 ir kt.)



24 pav. Vieno kanalo aptarnavimo sistemos schema

Kompiuterių tinklu perduodamas duomenų paketas yra laikomas viena paraiška. Kiekviena paraiška tinkle turi savo gyvavimo laiką, kuris patikrinamas tinklo mazge ir, jei jis yra pasibaigęs, paraiška naikinama. Siekiant išvengti paraiškų praradimo, tinklo mazguose naudojama buferinė atmintis, kurioje atėję paketai saugomi tol, kol bus aptarnauti (Lemoine, Pham, Lefèvre 2004). 24 paveiksle pavaizduotame komutatoriuje tinklo eilės aptarnavimui taikomas *LIFO* metodas, aptarnaujantis tinklą pagal principą paskutinis įėjai, pirmas aptarnaujamas. Modeliuojamo tinklo kanalo plotį galima apskaičiuoti vidutinį paketo ilgį padalinus iš paketo aptarnavimo laiko ir gautąją reikšmę padauginus iš paketų skaičiaus (Plėštys, Kavaliūnas, Vilutis 2008). Bendruoju atveju šį įrenginį galima modeliuoti kaip vienkanalę aptarnavimo sistemą su rekurentiniu įėjimo srautu, bendro pobūdžio aptarnavimo srautu, ribojimais paketų skaičiui buferyje ir aptarnavimo laukimo laikui GI/G/1//N (Олифер 2008, Plėštys, Kavaliūnas, Vilutis 2008, Rindzevičius, Tervydis 2003). Vieno kanalo aptarnavimo sistema su *LIFO* buferiu pavaizduota 25 paveiksle.



**25 pav.** Vieno kanalo aptarnavimo sistema su *LIFO* buferiu

Išnagrinėjus statistinę universiteto tinklo apkrovą, nustatyta, kad pagal tinklo apkrovos charakteristikas suformuotos laiko eilutės pasižymi savastingumu (Kaklauskas, Sakalauskas 2008). Mazgo parametrus pažymėti naudojami tokie žymėjimai:  $\lambda$  – įėjimo srauto intensyvumas,  $\mu$  – aptarnavimo srauto intensyvumas,  $\tau$  – laiko tarpas tarp dviejų paraiškų atėjimo į sistemą,  $\rho$  – apkrova,  $s$  – paketo aptarnavimo sparta (Sakalauskas 2000).

Šiuolaikinių komunikacinių tinklų srautų savybės vertinamos taikant statistinės analizės metodus bei matematinį modeliavimą. Kadangi realaus

tinklo srautai pasižymi savastingumu (Kaklauskas, Sakalauskas 2008), tinklo procesų modeliavimui ir aprašymui yra taikomi netiesinės analizės metodai, įvertinant sunkias uodegas, apibūdinančias didelius tinklo srautų pliūpsnius.

### 5.3. Vieno kanalo tinklo mazgo modelis

Imituojant tinklo mazgo darbą kompiuteriu buvo generuojamos trukmių tarp paketų pasirodymo mazge momentų seka  $(\tau_0, \tau_1, \dots, \tau_n \dots)$  ir paketų aptarnavimo trukmių seka  $(x_0, x_1, \dots, x_n \dots)$ . Pasinaudojant šiomis sekomis apskaičiuojamos paketų aptarnavimo charakteristikos, atsižvelgiant į šių sekų elementų skirstinius bei aptarnavimo tvarką. Tirtas mazgo darbo efektyvumas, kai įėjimo ir aptarnavimo srautai yra Puasono arba savastingieji. Imituojant Puasono procesą taikomos formulės:  $\tau_i = -\frac{\ln(\xi_i)}{\lambda}$ ,  $x_i = -\frac{\ln(\zeta_i)}{\mu}$ ,  $\xi_i$ ,  $\zeta_i$  yra nepriklausomi tolygiai pasiskirstę vienetiniame intervale atsitiktiniai dydžiai.

Savastingam procesui imituoti buvo naudojamos formulės:

$$\tau_i = \frac{\alpha \sqrt{\ln(\xi_i) / \ln(\omega_i)}}{\lambda}, \quad x_i = \frac{\beta \sqrt{\ln(\zeta_i) / \ln(\nu_i)}}{\mu}, \quad \xi_i, \zeta_i, \omega_i, \nu_i \text{ yra nepriklausomi}$$

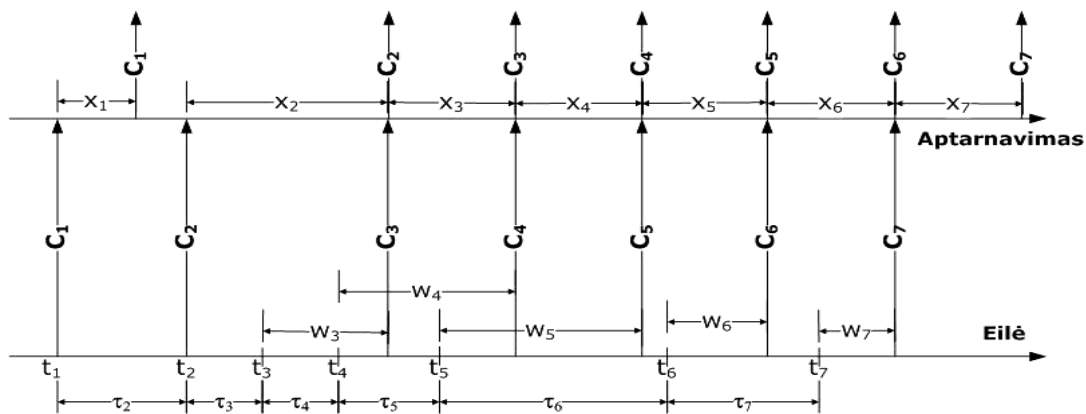
tolygiai pasiskirstę vienetiniame intervale atsitiktiniai dydžiai.

Nesunku apskaičiuoti paraiškų pasirodymo momentų seką:  $t_{n+1} = t_n + \tau_{n+1}$ ,  $t_0 = 0$ . Žinant paraiškos pasirodymo momentą bei aptarnavimo ir laukimo trukmę galima apskaičiuoti paraiškos išėjimo iš sistemos laiką:  $s_{n+1} = t_{n+1} + w_{n+1} + x_{n+1}$ . Laukimo laikas, priklausomai nuo aptarnavimo tvarkos yra apskaičiuojamas skirtingai. Jei aptarnavimo tvarka yra *FIFO*, tai laukimo laikas apskaičiuojamas taikant Lindli formulę  $w_{n+1} = \max[0, \min[w_n + x_n - \tau_{n+1}, w_{igl}]]$ , jei  $N_{n+1} \leq N_{buf}$ , kitaip  $w_{n+1} = 0$  ir paraiška patenka į prarastų paraiškų srautą, čia paprastai priimama, kad pradiniu laiko momentu sistema nebuvo apkrauta:  $w_0 = 0$ . Paraiškų skaičių, liekantį sistemoje  $n+1$  paraiškai išėjus iš sistemos, galima apskaičiuoti rekurentiniu būdu:  $N_{n+1} = \min[N_{buf}, N_n + v_{n+1} - 1]$ , čia  $v_{n+1}$  yra paraiškų, atėjusių į sistemą tarp  $n$ -osios ir  $n+1$ -osios paraiškų išėjimų, skaičius. Nesunku pastebėti, jog skaičius



$v_{n+1}$  yra lygus atėjimo momentų, patekusių į intervalą nuo  $s_n$  iki  $s_{n+1}$ , skaičiui. Prarastų paraiškų, tarp atėjusių į sistemą tarp  $n$ -osios ir  $n+1$ -osios paraiškų ir prarastų išėjimų, skaičius yra lygus  $L_{n+1} = \max[0, N_n + v_{n+1} - N_{buf} - 1]$  (Iversen 2011).

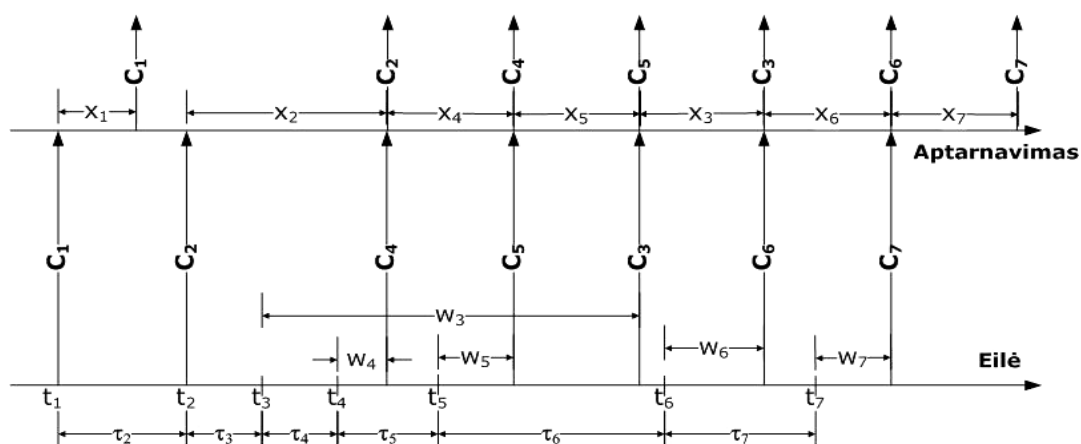
26 paveiksle pavaizduota paraiškų aptarnavimo disciplina, kai laukimo eilė *FIFO*. Čia  $t_1, t_2, \dots, t_n$  yra paraiškų atėjimo laikas į tinklo mazgą,  $x_1, x_2, \dots, x_n$  – paraiškos aptarnavimo laikas,  $w_1, w_2, \dots, w_n$  – paraiškos laukimo laikas buferyje, o  $C_1, C_2, \dots, C_n$  – aptarnautų paraiškų išėjimo iš tinklo mazgo laikas. Pirmoji paraiška į tinklo mazgą atėjo  $t_1$  laiku ir iš karto buvo aptarnauta per laiką  $x_1$ , nes tuo metu laukimo eilė buvo tuščia. Jos išėjimo iš sistemos laikas –  $C_1$ , o laukimo laikas  $w_1 = 0$ . Tuo tarpu trečioji paraiška atėjo  $t_3$  laiku, kai tinklo mazgas aptarnavo antrąją paraišką, todėl ji turėjo palaukti  $w_3$  laiką buferyje, kol bus laisvas tinklo mazgas. Šios paraiškos išėjimo iš sistemos laikas turės visus tris nenulines dedamąsias  $C_3 = t_3 + w_3 + x_3$ . Analogiškai apskaičiuojamas ir kitų paraiškų išėjimo iš tinklo mazgo laikas.



**26 pav.** Paraiškų aptarnavimas, kai eilės disciplina *FIFO*

Analizuojant mokslinius straipsnius apie paketinio kompiuterių tinklo srauto modeliavimą, išsamaus *LIFO* eilės veikimą kompiuterių tinklo mazge aprašančio algoritmo nerasta. Modelio analizei sukurtas algoritmas pagal *LIFO* tinklo mazgo eilės darbo techninius aprašus.

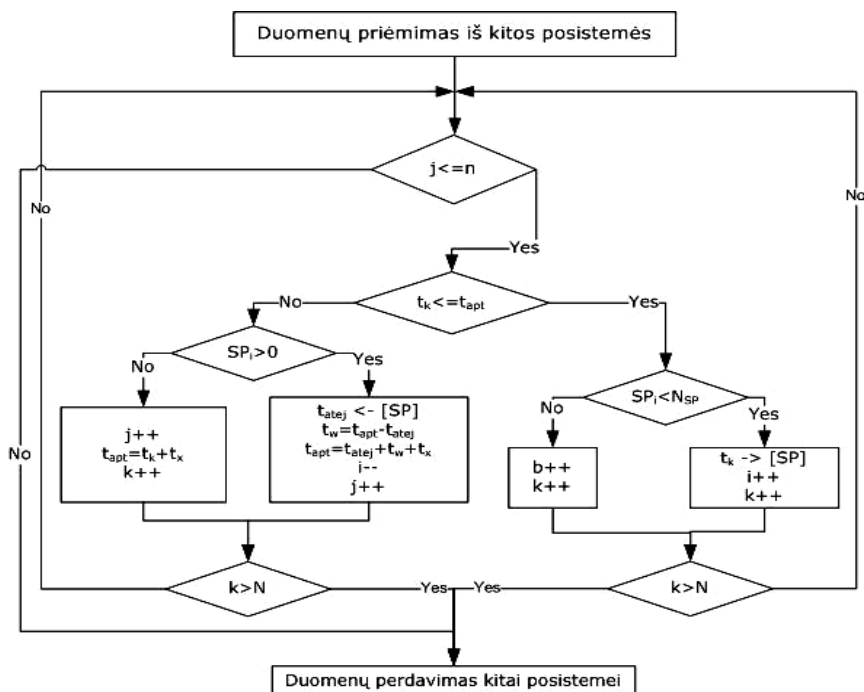
Jei aptarnavimo tvarka *LIFO*, tai laukimo laikas apskaičiuojamas sudėtingiau. 27 paveiksle pateikta paraiškų aptarnavimo laiko diagrama, kai aptarnavimo tvarka *LIFO* ir ribojama buferio talpa. Matome, kad pirmoji paraiška, atėjusi į tinklo mazgą, bus aptarnauta iš karto. Aptarnavus pirmąją paraišką, į tinklo mazgą laiku  $t_2$  ateina antroji paraiška, kuri iš karto aptarnaujama. Laiko momentais  $t_3$  ir  $t_4$  ateina paraiškos, kurios statomos į *LIFO* eilę ir saugomos tinklo mazgo buferyje. Tinklo mazgui atsilaisvinus, pirmiausia bus aptarnaujama paskutinioji atėjusi paraiška (atėjimo laikas  $t_4$ ). Jos laukimo laikas –  $w_4$ , o aptarnavimo –  $x_4$ . Aptarnaujant ketvirtąją paraišką tinklo mazgą pasiekė dar viena paraiška, kurios atėjimo laikas –  $t_5$ . Dabar ji bus pirmoji *LIFO* aptarnavimo eilėje ir iš karto bus aptarnaujama tinklo mazge, kai tik jis atsilaisvins. Tuo tarpu  $t_3$  laiku atėjusi paraiška lauks ilgiausiai ir bus aptarnaujama tik po penktosios paraiškos, kai į tinklo mazgą neateis naujų paraiškų, o buferyje bus likusi tik ji viena.



27 pav. Paraiškų aptarnavimas, kai eilės disciplina *LIFO*

Sukurtą tinklo srauto modeliavimo paketą sudaro tinklo srauto simuliacinio posistemė, tinklo mazgo modeliavimo posistemė, valdymo posistemė bei įvesties/išvesties posistemė. Visos keturios posistemės yra tarpusavyje susietos ir valdomos naudojant grafinę sąsają arba konfigūracijos failus. Galima aprašyti arba nurodyti duomenų srautų perdavimo tarp posistemių tvarką bei jų formatus. 28 paveiksle pateiktas vienas iš tinklo mazgo modeliavimo posistemės algoritmų, skirtų tinklo srauto aptarnavimui kai atėjusių paketų

eilės buferyje aptarnavimo disciplina *LIFO*. Paveiksle naudojami šie žymėjimai:  $N$  – sistemoje sugeneruotų paraiškų kiekis,  $n$  ( $n < N$ ) – numatytų aptarnauti paraiškų kiekis,  $t_k$  – paraiškos patekimo į tinklo mazgo modeliavimo posistemę laikas,  $t_{apt}$  – paraiškos išėjimo iš tinklo mazgo modeliavimo posistemės laikas,  $t_{atej}$  – iš buferio aptarnaujamos paraiškos atėjimo laikas,  $j$  ( $j \leq n$ ) – einamu momentu posistemėje aptarnautų paraiškų skaičius,  $k$  ( $k \leq N$ ) – einamu momentu į posistemę atėjusių paraiškų skaičius,  $N_{SP}$  – sistemos buferio talpa,  $SP_i$  – einamu momentu nepanaudota *LIFO* buferio talpa,  $b$  – prarastų paraiškų kiekis.  $j++$  – reiškia, kad kintamojo reikšmė didinama vienetu, o  $i--$  – mažinama vienetu.



**28 pav.** Paraiškų aptarnavimo algoritmas *MulNodSimSys*, kai mazge naudojama *LIFO* eilė

Pagal šį algoritmą į tinklo mazgą atėjusios paraiškos yra aptarnaujamos tol, kol pasiekiami tinklo srauto simuliacijos posistemės sugeneruotos eilutės pabaiga ( $N$ -tasis paketas) arba aptarnaujamas nurodytas paraiškų kiekis  $n$ . Sąlyga  $t_k \leq t_{apt}$  tikrina ar atėjusio į sistemą paketo patekimo į tinklo mazgo modeliavimo posistemę laikas neviršija šiuo metu iš sistemos išeinančio paketo aptarnavimo laiko:

- Jei sąlyga tenkinama, tai atėjęs paketas negali būti aptarnautas, posistemė užimta, todėl jis yra statomas į *LIFO* eilę, prieš tai patikrinus ar nėra pilnai užpildytas buferis. Jei buferis užpildytas, tai atėjusi paraiška prarandama;
- Jei sąlyga netenkinama, vadinasi šiuo metu posistemė yra neužimta ir paketas bus aptarnaujamas, jei buferyje nėra laukiančių paketų. Jei tokių paketų yra, tai jie bus aptarnaujami pirmiau nei naujai atėjęs paketas. Apskaičiuojamas aptarnautos paraiškos išėjimo iš posistemės laikas.

Analogiškai aprašomas ir tinklo mazgo modeliavimo posistemės algoritmas, kai atėjusių paketų buferis organizuojamas taikant *FIFO* eilę.

#### **5.4. Vieno kanalo tinklo mazgo tyrimo rezultatai**

*MulNodSimSys* srauto simuliacijos posistemėje generuotos 1000000 paraiškų turinčios įeinančio/apartarnaujamo srauto laiko eilutės. Programos *MulNodSimSys* dinaminėje bibliotekoje aprašyti šie tinklo mazgo parametrų pokyčiai:

- įeinančio/apartarnaujamo srauto tipas: *Puasono* (toliau žymėsime *P*) arba savastingasis (toliau žymėsime *S*);
- įėjimo srauto intensyvumas  
 $\lambda \in \{0,1; 0,2; 0,5; 1; 2; 5; 10; 12; 15; 20; 50\}$ ;
- aptarnavimo srauto intensyvumas  
 $\mu \in \{0,1; 0,2; 0,5; 1; 2; 5; 10; 12; 15; 20; 50\}$ ;
- savastingosios eilutės stabilumo parametras  
 $\alpha \in \{1,1; 1,2; 1,3; 1,4; 1,5; 1,6; 1,7; 1,8; 1,9\}$ .

Tiriant įeinančio ir aptarnaujamo srauto tinklo mazge intensyvumo priklausomybę nuo srauto savybių buvo generuotos laiko eilutės perrenkant visas  $\lambda$ ,  $\mu$  ir  $\alpha$  kombinacijos iš dinaminės bibliotekos. *MulNodSimSys* tinklo mazgo modeliavimo posistemėje nurodyta, kad kiekvienoje sugeneruotoje eilutėje būtina aptarnauti ne mažiau kaip 100000 įeinančio srauto paketų.

Vertinant tinklo mazge aptarnaujamo srauto savybių įtaką aptarnavimo našumui, kiekviena laiko eilutė buvo aptarnaujama keletą kartų keičiant šiuo tinklo mazgo parametrus:

- tinklo mazgo buferio eilės ilgis, kai  
 $SP_i \in \{1; 2; 5; 10; 12; 15; 20; 50; 100; 200; 500\}$ ;
- tinklo mazgo eilės aptarnavimo disciplina *FIFO*, *LIFO*.

Taip kiekvienai sugeneruotai įeinančio srauto laiko eilutei buvo gauta 11 skirtingų tos pačios eilutės aptarnavimo tinklo mazge rezultatų. Išanalizuotos 6776 sugeneruotos įeinančio srauto laiko eilutės ir gauti 74536 jų aptarnavimo tinklo mazge rezultatai. Atsižvelgiant į simuliuoto srauto tipą ir jo aptarnavimo discipliną tinklo mazgo eilėje analizei išskirtos 8 eilučių grupės:

1. *PP FIFO* – įeinantis srautas *Puasono* ir aptarnaujamas srautas *Puasono*, eilės aptarnavimo disciplina *FIFO*;
2. *PP LIFO* – įeinantis srautas *Puasono* ir aptarnaujamas srautas *Puasono*, eilės aptarnavimo disciplina *LIFO*;
3. *SP FIFO* – įeinantis srautas savastingasis ir aptarnaujamas srautas *Puasono*, eilės aptarnavimo disciplina *FIFO*;
4. *SP LIFO* – įeinantis srautas savastingasis ir aptarnaujamas srautas *Puasono*, eilės aptarnavimo disciplina *LIFO*;
5. *PS FIFO* – įeinantis srautas *Puasono* ir aptarnaujamas srautas savastingasis, eilės aptarnavimas disciplina *FIFO*;
6. *PS LIFO* – įeinantis srautas *Puasono* ir aptarnaujamas srautas savastingasis, eilės aptarnavimo disciplina *LIFO*;
7. *SS FIFO* – įeinantis srautas savastingasis ir aptarnaujamas srautas savastingasis, eilės aptarnavimo disciplina *FIFO*;
8. *SS LIFO* – įeinantis srautas savastingasis ir aptarnaujamas srautas savastingasis, eilės aptarnavimo disciplina *LIFO*.

Aptarnavimo rezultatų analizei taikyta *tiesinė regresija*, kai nepriklausomi kintamieji yra:

- santykis  $\frac{\lambda}{\mu}$ , charakterizuojantis tinklo srautą ( $\frac{\lambda}{\mu} \leq 1$  – eilė tinklo mazge baigtinė, įeinantis srautas nėra intensyvus;  $\frac{\lambda}{\mu} > 1$  – eilė begalinė, įeinantis srautas intensyvus),
- tinklo mazgo buferio apimtis  $SP_i$  ir laiko eilutės stabilumo parametras  $\alpha$ , charakterizuojantis jos savastingumą.

Priklausomi kintamieji:

- eilutės aptarnavimo tikimybė,
- buferio užpildymo tikimybė,
- paketų praradimo tikimybė.

Vertinant nepriklausomų kintamųjų  $SP_i$  ir  $\alpha$  sąryšį su kiekvienai eilutei apskaičiuotais įverčiais, atsižvelgta į srauto intensyvumą. Žinoma, kad kai  $\frac{\lambda}{\mu} \in (0,1]$ , tai įeinančio srauto intensyvumas yra mažas. Esant dideliame įeinančio srauto intensyvumui  $\frac{\lambda}{\mu} \in (1,500]$ . Atlikus skaičiavimus, sudarytos trys lentelės.

**5.1 lentelė.** Tinklo srauto savybių įtakos mazgo darbui, pagal aptarnavimo intensyvumą, įverčiai

Grupė	$0 < \frac{\lambda}{\mu} \leq 1$			$1 < \frac{\lambda}{\mu} \leq 500$		
	Aptarnavimo tikimybė	Praradimo tikimybė	Buferio užpildymo tikimybė	Aptarnavimo tikimybė	Praradimo tikimybė	Buferio užpildymo tikimybė
<i>PP FIFO</i>	1→0,907	0→0,039	0→0,871	0,688→0,002	0,311→0,998	0,629→1
<i>PP LIFO</i>	0,999→0,947	0→0,051	0→0,264	0,786→0,002	0,313→0,998	0,637→1
<i>SP FIFO</i>	0,999→0,862	0→0,138	0→0,210	0,864→0,007	0,136→0,993	0,196→1
<i>SP LIFO</i>	0,999→0,721	0→0,239	0→0,372	0,579→0,003	0,411→0,997	0,511→1
<i>PS FIFO</i>	0,941→0,456	0,069→0,545	0,024→0,621	0,571→0,382	0,561→0,621	0,501→0,581
<i>PS LIFO</i>	0,841→0,650	0,151→0,357	0,041→0,334	0,641→0,542	0,361→0,461	0,325→0,470
<i>SS FIFO</i>	0,739→0,740	0,261→0,265	0,355→0,368	0,737→0,736	0,262→0,264	0,358→0,341
<i>SS LIFO</i>	0,592→0,601	0,409→0,395	0,362→0,341	0,578→0,625	0,423→0,378	0,340→0,361

Kai tinklo mazgo aptarnavimo srautas yra *Puasono* (žr. 5.1 lentelę), tai didėjant įeinančio srauto intensyvumui aptarnavimo tikimybė mažėja, o paketų

praradimo tikimybė didėja. Kai tinklo mazgo aptarnavimo srautas yra savastingasis, tai paketų aptarnavimo tikimybė nekinta, keičiantis įeinančio srauto intensyvumui, o tinklo mazgas dirba stabiliau. Kai į tinklo mazgą įeinantis srautas yra savastingasis, tai aptarnavimo tikimybė tinklo mazge nepriklauso nuo įeinančio srauto intensyvumo. Eilės aptarnavimo disciplina *LIFO PS* tipo sraute užtikrina geresnę aptarnavimo kokybę.

5.2 lentelėje aprašyta buferio dydžio įtaka tinklo srauto aptarnavimui. Aptarnavimo tikimybė, esant mažam srauto intensyvumui, didėjant buferio dydžiui didėja, nepriklausomai nuo srauto savybių. Taip pat didėja ir buferio užpildymo tikimybė. Buferio eilės aptarnavimo disciplina *LIFO* geriau aptarnaus tinklo *PS* tipo srautą.

**5.2 lentelė.** Tinklo srauto savybių įtakos mazgo darbui, pagal mazgo buferio dydį, įverčiai

Grupė	$\frac{\lambda}{\mu} \leq 1$ , kai $1 \leq SP_i \leq 500$			$\frac{\lambda}{\mu} > 1$ , kai $1 \leq SP_i \leq 500$		
	Aptarnavimo tikimybė	Praradimo tikimybė	Buferio užpildymo tikimybė	Aptarnavimo tikimybė	Praradimo tikimybė	Buferio užpildymo tikimybė
<i>PP FIFO</i>	0,961→0,999	0,039→0,003	0,031→0,034	0,651→0,989	0,241→0,210	0,762→0,783
<i>PP LIFO</i>	0,962→0,989	0,393→0,015	0,355→0,357	0,654→0,978	0,240→0,211	0,763→0,801
<i>SP FIFO</i>	0,942→0,981	0,063→0,003	0,038→0,067	0,540→0,912	0,680→0,659	0,323→0,345
<i>SP LIFO</i>	0,873→0,982	0,137→0,005	0,098→0,126	0,179→0,245	0,824→0,761	0,645→0,973
<i>PS FIFO</i>	0,665→0,784	0,324→0,216	0,063→0,532	0,189→0,981	0,585→0,497	0,415→0,579
<i>PS LIFO</i>	0,641→0,829	0,361→0,142	0,015→0,325	0,039→0,745	0,492→0,350	0,511→0,662
<i>SS FIFO</i>	0,526→0,923	0,470→0,060	0,141→0,587	0,150→0,564	0,463→0,062	0,145→0,574
<i>SS LIFO</i>	0,385→0,811	0,621→0,184	0,140→0,540	0,400→0,783	0,601→0,201	0,142→0,520

Visiškai kita situacija, kai tinklo srautas yra intensyvus. Akivaizdžiai matyti, kad didėjant buferio talpai didėja ir įeinančio srauto aptarnavimo tikimybė, o eilės aptarnavimo disciplina jokios įtakos neturi.

5.3 lentelėje pateikti tinklo srauto aptarnavimo tikimybės įverčiai, atsižvelgiant į laiko eilutės stabilumo parametą. Būtina pažymėti, kad eilutėms su *Puasono* tipo aptarnavimo srautu, nenustatyta jokia priklausomybė nuo šio parametro. Kai tinklo mazge aptarnaujamas srautas yra savastingasis (*PS FIFO, PS LIFO, SS FIFO, SS LIFO*), tai didėjant eilutės stabilumo parametrai,

aptarnavimo tikimybė didėja. Stipresnė priklausomybė stebima, kai įeinantis ir aptarnavimo srautas yra savastingieji.

**5.3 lentelė.** Tinklo srauto savybių įtakos mazgo darbui, pagal eilutės stabilumo parametrai, įverčiai

Grupė	$\frac{\lambda}{\mu} \leq 1$ , kai $1,1 \leq \alpha \leq 1,9$			$\frac{\lambda}{\mu} > 1$ , kai $1,1 \leq \alpha \leq 1,9$		
	Aptarnavimo tikimybė	Praradimo tikimybė	Buferio užpildymo tikimybė	Aptarnavimo tikimybė	Praradimo tikimybė	Buferio užpildymo tikimybė
<i>PP FIFO</i>	–	–	–	–	–	–
<i>PP LIFO</i>	–	–	–	–	–	–
<i>SP FIFO</i>	0,772→0,689	0,252→0,284	0,190→0,483	0,670→0,241	0,324→0,765	0,263→0,973
<i>SP LIFO</i>	0,923→0,932	0,091→0,113	0,066→0,065	0,220→0,185	0,780→0,789	0,780→0,861
<i>PS FIFO</i>	0,773→0,715	0,230→0,317	0,191→0,489	0,670→0,241	0,270→0,952	0,330→0,754
<i>PS LIFO</i>	0,768→0,740	0,222→0,242	0,004→0,313	0,739→0,453	0,261→0,546	0,075→0,720
<i>SS FIFO</i>	0,563→0,915	0,441→0,151	0,359→0,378	0,555→0,894	0,443→0,089	0,558→0,917
<i>SS LIFO</i>	0,490→0,711	0,512→0,296	0,339→0,363	0,476→0,742	0,527→0,258	0,337→0,368

### 5.5. Paketinio kompiuterių tinklo mazgo tyrimai

Nors turima techninė kompiuterių tinklo valdymo įranga yra pakankamo galingumo, tačiau esant neprognozuojamoms perkrovoms jos galingumai gali būti per maži. Gausūs tinklo srauto tyrimais (Leland, Taqu, Willinger, Wilson 1994, Samorodnitsky 2006, Sheluhin, Smolskiy, Osin 2007 ir kt.) rodo, kad *Ethernet* kompiuterių tinklo srauto charakteristikos turi fraktalų savybių bei pasižymi savastingumu su *LRD*. Empiriniai Šiaulių universiteto e. studijų kompiuterių tinklo srauto tyrimai patvirtino, kad tinklo srautas pasižymi savastingumu (Kaklauskas, Sakalauskas 2008).

Kiekvienas kompiuterių tinklo mazgas turi pakankamo ilgio buferį, kad galėtų saugoti tinklo paketus eilėje, kol jie bus aptarnauti. Kompiuterių tinkluose, kai duomenų paketų perdavimo prioritetai neaktualūs, tinklo mazgo eilių aptarnavimui dažniausiai naudojamos *FIFO* ir *LIFO* disciplinos. Moksliniuose tyrimuose plačiai analizuojama šių eilių aptarnavimo disciplinų įtaka tinklo mazgo darbui. Nzouonta, Ott ir Borcea (Nzouonta, Ott, Borcea 2009) analizuodami *ad hoc* tinklą (tinklai sujungiami ir iš karto yra parengti darbui) *UDP* srautą nustatė, kad *LIFO* ir *FIFO* aptarnavimo disciplinos turi



vienodus vėlinimus, kai srautas mažas, tačiau kai srautas didelis, tai *LIFO Frontdrop* (eilei užsipildžius, naikinami ilgiausiai užsibuvę paketai ir į ją statomi naujai atėję) pasižymėjo daugiau nei dvigubai mažesniu vėlinimu nei *FIFO Taildrop* (eilei užsipildžius naikinami naujai atėję paketai). Optimalų tinklo mazgo eilės ilgį analizavo Eilės ilgio įtaka tinklo mazgo darbui nagrinėjo Appenzeller, Keslassy, Enachescu, Ganjali ir daugelis kitų mokslininkų (Appenzeller, Keslassy, McKeown 2004; Enachescu, Ganjali, Goel, McKeown, Roughgarden 2005, Ramanan, Cao 2002). Nustatyta, kad mazguose aptarnaujančiuose didelės spartos (apie 30Gb/s) ir pralaidos kompiuterių tinklus pilnai pakanka 15-20 paketų buferio (Nzouonta, Ott, Borcea 2009).

Savastingųjų procesų kompiuterių tinkluose tyrimai (Samorodnitsky 2006; Samorodnitsky, Taqqu 1994, Taqqu, Teverovsky 1998) leidžia prognozuoti srauto kaitą ir prognozės rezultatus taikyti tinklo srauto aptarnavimo kokybės užtikrinimui. Tyrimais nustatyta, kad mokomajam tinklui yra būdingos perkrovos dėl studijų sezoniškumo bei studijavimo ypatumų (Kaklauskas, Sakalauskas 2008). Reikia pažymėti, kad nėra išsamių tyrimų, kuriuose būtų įvertinta tinklo mazgo buferio talpos ir eilės aptarnavimo disciplinos įtaka tinklo srauto aptarnavimui, atsižvelgiant į to srauto savybes. Tokie vertinimai padėtų parinkti tinkamiausius tinklo mazgo parametrus, kad ir esant didelėms perkrovom bei savastingam srautui su sunkiomis uodegomis, būtų sėkmingai aptarnaujami tinklo klientai. Tai užtikrintų stabilų kompiuterių tinklo darbą, kas ypač aktualu mokymo įstaigose, kai studijų procese sparčiai populiarėja nuotolinės studijos.

### ***5.6. Apibendrintas stochastiškai apribotas tinklo modelis***

Kompiuterių tinklas nagrinėjamas kaip aptarnavimo sistema sudaryta iš atskirų mazgų (plačiau 2.2.3 skyriuje), atliekančių informacinių srautų apdorojimo ir paskirstymo funkcijas tarp tinklo klientų bei mazgų, inicijuojančių bei naudojančių šiuos srautus. Remiantis N. Vasiliou (Vasiliou 2000) tinklo paslaugų kokybė *QoS* yra sudaryta iš veiklos užtikrinimo

komponento ir paslaugų diferencijavimo komponento. Mūsų tinklo modelyje analizuojamas veiklos užtikrinimo komponentas, susijęs su kanalo pralaida, kuri tiesiogiai siejasi su paketų vėlinimu, srauto pliūpsniškumu ir paketų praradimu. Analizuojat tinklo srauto savybes bei tinklo mazgo parametrus siekiama nustatyti sąlygas, kai paketų vėlinimas ir praradimas bus minimalus, įvertinant įeinančio tinklo srauto pliūpsniškumą bei aptarnavimo sistemos pralaidą. Paslaugų diferencijavimas, taikant skirtingus paslaugos kokybės *QoS* reikalavimus skirtingoms paslaugoms šiame straipsnyje nėra nagrinėjamas.

Nagrinėjamas tinklo modelis pagrįstas stochastinio tinklo modeliu (angl. *Stochastic Network Calculus*), kuris leidžia analizuoti *end-to-end* kompiuterių tinklų *QoS* sistemas su stochastiniu įeinančiu tinklo srautu ir stochastiniais tinklo mazgais. Deterministinę tinklo paslaugų kokybės teoriją kūrė Cruz, Chang, Boudec, Thiran ir kiti (Chang 1994, Boundec, Thiran 2001) ir ją pavadino tinklų skaičiavimo teorija (angl. *Network Calculus*). Ši teorija pastoviai vystoma, 2008 metais Jiang ir Liu išleido knygą, kur analizuojamos stochastinės įeinančio ir aptarnaujamo srauto charakteristikos.

Rengiant modelį pritaikytas J. Jiang, Q. Yin, Y. Liu ir S. Jiang aprašytas apibendrintas stochastiškai apriboto srauto komunikavimo tinklams modelis (angl. *Generalized Stochastically Bounded Bursty Traffic for Communication Networks*) (Jiang, Yin, Liu, Jiang 2009) Tokiame tinkle įeinantis ir tinklo mazge aptarnaujamas srautai nepriklausomi ir abu stochastiniai. Tinklo mazgo buferio talpa determinuota ir baigtinė. Tinklo mazgas pasirengęs sekančios paraiškos aptarnavimui, kai buferis tuščias ir tinklo mazge nėra aptarnaujamos paraiškos. Tinklo darbo pradžia laikomas laikas  $t=0$ . Tinklo srauto, gaunamo per laiko intervalą  $(s,t]$ , suma  $A(s,t) = \sum_{i=s+1}^t a(i)$ . Čia  $A(t)$  pažymėtas  $A(0,t)$ , o  $a(t)$  aprašo gautą duomenų srautą per  $(t-1,t]$  laiką, t. y.  $a(t) = A(t) - A(t-1)$ .

Tinklo mazgo srautui  $\hat{A}$  yra apibrėžta funkcija  $\hat{A}(t; \rho) = \sup_{s \leq t} \{A(s,t) - (t-s)\rho\}$ , kur  $\rho$  gautas srautas. Daugiakanalėje

sistemoje  $\rho = \lambda / \mu$ , kur  $\lambda$  – į sistemą ateinančių paketų intensyvumas,  $\mu$  – paketų praeinančių per tinklo mazgo kanalus intensyvumas.

Modeliuojamas tinklo srautas yra generalizuotas stochastiškai apribotas srautas *gSBB* (angl. *Generalized Stochastically Bounded Burstiness* toliau *gSBB*). Toks srautas visiems  $t \geq 0$  bei visiems  $x \geq 0$  tenkina nelygybę  $P\{\hat{A}(t, \rho) > x\} \leq f(x)$ , kai  $\rho$  yra viršutinė srauto riba,  $f$  – aprėpties funkcija, kuri yra nedidėjanti ir  $f(x) \geq 0$  visiems  $x \geq 0$ ,  $A(0, t)$  – įeinantis srautas, tenkinantis nelygybę  $A(0, t) \ll \langle f, \rho \rangle$ . (Jiang, Yin, Liu, Jiang 2009). Įrodyta, kad jei *gSBB* tinklo srautas pasižymi stacionarumo ir ergodiškumo savybėmis,

tai  $P\left\{\hat{A}(t; \rho) > x\right\}$  bet kuriuo laiko momentu turi viršutinę ribą pastoviam eilutės ilgio skirstinyje, t. y.  $P\left\{\hat{A}(\infty; \rho) > x\right\}$  virtualioje vieno kanalo sistemoje

su pastoviu  $\rho$ . Čia  $\hat{A}(\infty; \rho)$  žymima  $\hat{A}(t; \rho)$ , kai  $t \rightarrow \infty$  (Jiang, Yin, Liu, Jiang 2009). Generuojamas tinklo srauto paketų ilgis kintamas ir tenkina *Ethernet* standarto reikalavimus.

Naudojant tinklų skaičiavimo teorija (angl. *Network Calculus*), sudėtingų tinklo sistemų analitiniam aprašui galima taikyti alternatyvias algebras *min-plus* arba *max-plus*. Panaudojant *min-plus* algebrą galima aprašyti tinklo paslaugos teikimo procesą, išeinantį srautą pagal įeinančio srauto parametrus. Mūsų modeliuojamas stochastinis kompiuterių tinklas su *gSBB* srautu pasižymi paketų praradimais, kurie iššaukia jų retransliavimo poreikį, šiame tinkle teikiamos įvairialypės tinklo paslaugos, o srautas sudarytas iš daugelio dedamųjų. Analitiniam modelio aprašymui netinka *min-plus* algebra, nes procesų analizė grindžiama suminiu virtualių procesų laiku, todėl tolesnei analizei pasirenkame *max-plus* algebros priemones. Šioje algebroje naudojami tokie žymėjimai:  $\varepsilon \stackrel{def}{=} -\infty$  – elementas nusakantis būseną niekada,  $e \stackrel{def}{=} 0$  – elementas reiškiantis nulį,  $R$  – realiųjų skaičių aibė,  $R_{\max} = R \cup \{\varepsilon\}$  – *max-plus* algebros aibė. Sakykime, kad yra aprašytos funkcijos  $f$  ir  $g$ , kur  $f, g \in R_{\max}$ ,

tada du pagrindiniai operatoriai naudojami šioje algebroje aprašomi taip (Xie; Jiang 2009):

$\otimes$  – konvoliucijos (angl. *convolution*) operatorius, t. y. dviejų funkcijų  $f$  ir  $g$  konvoliucija gaunama pagal formulę  $f \otimes g(t) \equiv \sup_{0 \leq s \leq t} \{f(s) + g(t - s)\}$ ;

$\oplus$  – dekonvoliucijos (angl. *deconvolution*) operatorius, t. y. dviejų funkcijų  $f$  ir  $g$  dekonvoliucija gaunama pagal formulę  $f \oplus g(t) \equiv \inf_{s \geq 0} \{f(t + s) - g(s)\}$ ;

Pažymėkime  $A(k)$  stacionarią ir ergodišką eilutę, kuri aprašo įeinančio paraiškų srauto į modeliuojamą tinklą laikus.  $x_j(k)$  –  $k^{th}$  paraiškos aptarnavimo  $j$  mazge laikas, kai  $t \geq 0$ .  $\sigma_j(k)$  žymimas  $k^{th}$  paraiškos aptarnavimo pradžią  $j$  eilėje. Modeliuojama sistema yra atvira ir sudaryta iš 10 mazgų. Sistemos raida aprašoma  $j+1$  ilgio vektoriumi  $x(k) = (x_0(k), \dots, x_j(k))$  ir iš to sekančia homogenine lygtimi  $x(k+1) = A(k) \otimes x(k)$  (Heidergott 2006). Tada matrica  $A(k-1)$  aprašoma taip:

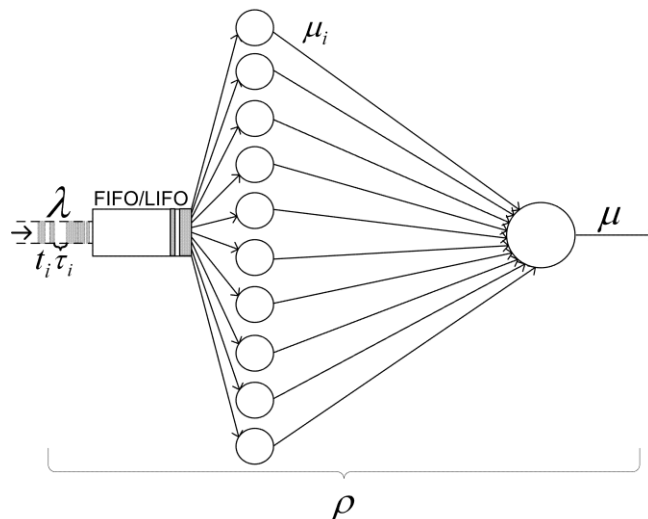
$$\begin{pmatrix} \sigma_0(k) & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \sigma_1(k) & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \sigma_2(k) & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \sigma_3(k) & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & \sigma_4(k) & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \sigma_5(k) & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \sigma_6(k) & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \sigma_7(k) & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \sigma_8(k) & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \sigma_9(k) \end{pmatrix}$$

Šį modelį pritaikysime statistinei universiteto tinklo apkrovai analizuoti. Nustatyta, kad pagal tinklo apkrovos charakteristikas suformuotos laiko eilutės pasižymi savastingumu (Kaklauskas, Sakalauskas 2008). Kaip rašyta 2.2.2 skyriuje savastingasis simetriškas procesas pasižymintis begaline dispersija yra  $\alpha$ -stabilusis procesas  $S_\alpha(\beta, \sigma, \mu)$  (Samorodnitsky 2006a, Samorodnitsky, Taggu 1994). Jo modeliavimui naudojami keturi parametrai. Reikia pažymėti, kad  $\alpha$ -stabilusis procesas pasižymi ir Pareto savybe.

### 5.7. Tinklo modelio aprašymas

Modeliuojamas tinklas yra daugiakanalė aptarnavimo sistema (žr. 29 pav.) su dešimčia kanalų, rekurentiniu įėjimo srautu, bendro pobūdžio aptarnavimo srautu, ribojimais paketų skaičiui buferyje ir aptarnavimo laukimo laikui  $GI/G/m//N$  (Pranevičius 2008, Олифер, Олифер 2008, Plėštys, Kavaliūnas, Vilutis 2008, Rindzevičius, Tervydis 2003). Modeliuojamas tinklas yra fizinio e. studijų tinklo atitikmuo, jame nėra vykdomas vaizdo ir garso perdavimas, todėl prioritetingos eilės neanalizuojamos.

Siekiant išvengti paraiškų praradimo, tinklo mazguose naudojama buferinė atmintis  $N_{SP}$ , kurioje atėję paketai saugomi tol, kol bus aptarnauti (Lemoine, Pham, Lefèvre 2004).  $N_n$  yra paketų skaičius buferyje, kai iš sistemos išėjo  $n$  paketų,  $L$  – sistemoje prarastų (neaptarnautų) paketų skaičius. Modeliuojamoje sistemoje naudojama eilių aptarnavimo disciplina *FIFO taildrop* ir *LIFO taildrop* (Nzouonta, Ott, Borcea 2009). Laikoma, kad pradinėje būsenoje visi sistemos kanalai ir eilė yra laisvi ( $t=0$ ), t. y. sistema pasirengusi tuoj pat aptarnauti atėjusią paraišką. Eilės ilgis  $N_{SP}$  yra bendras visiems kanalams. Jei, atėjus paraiškai, visi sistemos kanalai užimti, o eilės ilgis  $N_n < N_{SP}$ , tai atėjęs paketas yra talpinamas į eilę.



29 pav. Daugiakanalė aptarnavimo sistema  $GI/G/10//N$

29 paveiksle pavaizduota tokio tinklo modelio schema. Čia įeinančio srauto intensyvumas  $\lambda$ , kuris aptarnaujamas  $m=10$  vienodo prioriteto kanalų, kurių kiekviename aptarnavimo srautas žymimas  $\mu_i$ , o išeinantis srautas  $\mu = \sum_{i=1}^{10} \mu_i$ .

Bendrasis tinklo srauto pajėgumas  $\rho = \frac{\lambda}{\mu}$ . Modeliuojamo tinklo kanalo plotį galima apskaičiuoti vidutinį paketo ilgį padalinus iš paketo aptarnavimo laiko ir gautąją reikšmę padauginus iš paketų skaičiaus (Plėštys, Kavaliūnas, Vilutis 2008).

Kitiems modeliuojamos sistemos parametrai žymimi taip:  $\tau_i$  – trukmė tarp gretimų paketų pasirodymo mazge,  $x_i$  – i-ojo paketo aptarnavimo trukmė,  $\rho$  – apkrovos koeficientas,  $\bar{X}$  – vidutinis paketo ilgis,  $s$  – paketo aptarnavimo sparta,  $\bar{T}$  – vidutinis paketo buvimo sistemoje laikas,  $\bar{W}$  – vidutinis paketo laukimo laikas sistemoje,  $\bar{N}$  – vidutinis paraiškų skaičius sistemoje,  $P_L$  – paraiškos praradimo tikimybė,  $\bar{T}_{SP}$  – vidutinis serverio arba tinklo mazgo pilnos apkrovos laikas (Pranevičius 2008).

Imituojant tinklo darbą kompiuteriu generuojamos trukmių tarp paketų pasirodymo mazge momentų seka  $\tau_0, \tau_1, \dots, \tau_n \dots$  ir paketų aptarnavimo trukmių seka  $x_0, x_1, \dots, x_n \dots$ . Pasinaudojant šiomis sekomis apskaičiuojamos paketų aptarnavimo charakteristikos, atsižvelgiant į šių sekų elementų skirstinius bei aptarnavimo tvarką. Buvo tirtas mazgo darbo efektyvumas, kai įėjimo ir aptarnavimo srautai yra Markovo arba savastingieji. Imituojant Markovo procesą buvo taikomos formulės:

$$\tau_i = -\frac{\ln(\xi_i)}{\lambda}, \quad x_i = -\frac{\ln(\zeta_i)}{\mu},$$

$\xi_i, \zeta_i$  yra nepriklausomi tolygiai pasiskirstę vienetiniame intervale atsitiktiniai dydžiai. Savastingajam procesui imituoti buvo naudojamos formulės:

$$\tau_i = \frac{\sqrt[\alpha]{\ln(\xi_i) / \ln(\omega_i)}}{\lambda}, \quad x_i = \frac{\sqrt[\beta]{\ln(\zeta_i) / \ln(\nu_i)}}{\mu},$$

$\xi_i, \zeta_i, \omega_i, \nu_i$  yra nepriklausomi tolygiai pasiskirstę vienetiniame intervale atsitiktiniai dydžiai. Šios formulės išvestos pasinaudojus tuo, kad Puasono srautas su parametru, pasiskirsčiusiu nepriklausomai pagal Puasono dėsnį yra pasiskirstęs pagal Pareto dėsnį.

Nesunku apskaičiuoti paraiškų pasirodymo momentų seką:  $t_{n+1} = t_n + \tau_{n+1}, t_0 = 0$ .  $(n+1)^{th}$  paraiškos išėjimo iš sistemos laikas apskaičiuojamas pagal formulę:  $s_{n+1} = t_{n+1} + w_{n+1} + x_{n+1}$ , kur  $w_{n+1}$  yra  $(n+1)^{th}$  paketo laukimo buferyje laikas. Laukimo laikas, priklausomai nuo aptarnavimo tvarkos yra apskaičiuojamas skirtingai. Jei aptarnavimo tvarka yra *FIFO*, tai laukimo laikas  $w_{n+1}$  apskaičiuojamas taikant *gSBB*  $w_{n+1} = \max[0, \min[w_n + x_n - \tau_{n+1}, w_{lgL}]]$ , kai  $N_n < N_{SP}$ , kitaip  $w_{n+1} = 0$  ir paraiška prarandama, jei visi kanalai užimti. Paraiškų skaičių, liekantį sistemoje  $(n+1)^{th}$  paraiškai išėjus iš sistemos, galima apskaičiuoti rekurentiniu būdu:  $N_{n+1} = \min[N_{buf}, N_n + \nu_{n+1} - 1]$ , čia  $\nu_{n+1}$  yra paraiškų atėjusių į sistemą tarp  $n^{th}$  ir  $(n+1)^{th}$  paraiškų išėjimų skaičius. Nesunku pastebėti, jog skaičius  $\nu_{n+1}$  yra lygus atėjimo momentų, patekusių į intervalą nuo  $s_n$  iki  $s_{n+1}$ , skaičiui. Prarastų paraiškų  $L_{n+1}$ , tarp atėjusių į sistemą tarp  $n^{th}$  ir  $(n+1)^{th}$  paraiškų išėjimų, skaičius yra lygus  $L_{n+1} = \max[0, N_n + \nu_{n+1} - N_{SP} - 1]$  (Kleinrock 1975).

(Jiang, Yin, Liu, Jiang 2009) įrodė, kad remiantis *gSBB* savastingasis procesas yra modeliuojamas pagal formulę:

$$f^{self-similar}(x) = C_\alpha \left( \frac{\rho - m}{\delta} \right)^{-\alpha}, \quad \text{kai tenkinama nelygybė}$$

$P\left\{\hat{A}(t; \rho) > x\right\} \leq f^{self-similar}(x)$ . Čia  $C_\alpha$  ir  $m$  yra apskaičiuojami kaip  $S_\alpha(\beta, \sigma, \mu)$  ir  $x$  parametrai.

Pagal *gSBB* modelį Puasono srautas yra modeliuojamas pagal formulę:

$$f^{Poisson}(x) = 1 - (1 - \eta) \cdot \sum_{i=0}^k \left[ \frac{[\eta(i-k)]^i}{i!} e^{-\eta(i-k)} \right], \text{ kai tenkinama nelygybė}$$

$$P\left\{ \hat{A}(t; \rho) > x \right\} \leq f^{Poisson}(x). \text{ Čia } \eta = \frac{\lambda \bar{S}}{\rho}, \text{ ir } k = \left\lfloor \frac{x}{\bar{S}} \right\rfloor, \text{ kur } \bar{S} \text{ vidutinis paketų}$$

ilgis.

Daugiakanalės aptarnavimo sistemos GI/G/10//N darbas realizuotas naudojant algoritmą, kai eilės aptarnavimui naudojamos *FIFO tail drop* arba *LIFO tail drop* disciplinos.

**Algoritmas 1.** Daugiakanalės aptarnavimo sistemos mazgo aptarnavimo simuliacinio posistemės darbo algoritmas, naudojamas sistemoje *MulNodSimSys*.

*Tikslas:* aptarnauti iš tinklo srauto simuliacinio posistemės gautas paraiškas, taikant *FIFO* arba *LIFO* eilės aptarnavimo discipliną.

*Įėjimo parametrai:*  $t_i$  – paraiškų atėjimo laikų vektorius,  $x_i$  – paraiškų aptarnavimo laikų vektorius,  $N$  – sistemoje sugeneruotų paraiškų kiekis,  $n$  ( $n < N$ ) – numatytų aptarnauti paraiškų kiekis,  $j$  ( $j \leq n$ ) – einamu momentu posistemėje aptarnautų paraiškų skaičius,  $k$  ( $k \leq N$ ) – einamu momentu į posistemę atėjusios paraiškos numeris,  $N_{SP}$  – sistemos buferio talpa,  $N_{cur}$  – einamu momentu panaudota buferio talpa,  $SP_{disc}$  – eilės aptarnavimo disciplina,  $n_{ch}$  – einamo kanalo numeris,  $N(n_{ch})$  – aptarnautų paraiškų skaičiaus  $n_{ch}$  kanale,  $s(N(n_{ch}), n_{ch})$  – paraiškų aptarnavimo laikų matrica,  $apl$  – einamos paraiškos numeris,  $LIFO_i$  – *LIFO* eilės disciplinos vektorius,  $FIFO_i$  – *FIFO* eilės disciplinos vektorius,  $w(N(n_{ch}), n_{ch})$  – paraiškų laukimo laikų matrica,  $A_{served}$  – paraiškos aptarnavimo požymis,  $N_{ch}$  – kanalų skaičius tinklo mazge.

*Išėjimo parametrai:*  $s(N(n_{ch}), n_{ch})$ ,  $w(N(n_{ch}), n_{ch})$ ,  $L$  – prarastų paraiškų kiekis.

1 žingsnis. Pagal parametrus gautus iš įvesties/išvesties posistemės generuojami  $t_i$  ir  $x_i$  vektoriai, taikant (1) ir (2) formules, kai  $1 < \alpha < 2$ .



Nustatomos pradinės reikšmės  $N(n_{ch}) = 0$  visiems kanalams,  $n_{ch} = N_{ch}$ ,  
 $w(N(n_{ch}), n_{ch}) = 0$ ,  $s(N(n_{ch}), n_{ch}) = t_1 + x_1$ ,  $N_{cur} = 0$ ,  $k = 2$ ,  $j = 1$ ,  $L = 0$ .

2 žingsnis. Kiekvienam  $j \leq N$ .

3 žingsnis. If  $t_k \geq s(N(n_{ch}), n_{ch})$  paraiška atėjo, kai aptarnavimo sistemos kanalas laisvas.

4 žingsnis. If  $N_{cur} > 0$ , atsižvelgiant į  $SP_{disc}$  randamas  $apl$ ,

$N(n_{ch}) = N(n_{ch}) + 1$ ,  $w(N(n_{ch}), n_{ch}) = s(N(n_{ch}) - 1, n_{ch}) - t_{apl}$ ,

$s(N(n_{ch}), n_{ch}) = t_{apl} + w(N(n_{ch}), n_{ch}) + x_{apl}$ ,  $SP_i = SP_i - 1$ ,  $j = j + 1$ ,

$A_{served} = True$ .

5 žingsnis. Jei 4 žingsnyje užrašyta sąlyga nėra tenkinama, tai

$N(n_{ch}) = N(n_{ch}) + 1$ ,  $w(N(n_{ch}), n_{ch}) = 0$ ,  $s(N(n_{ch}), n_{ch}) = t_k + x_k$ ,  $j = j + 1$ ,

$k = k + 1$ ,  $A_{served} = True$ .

6 žingsnis.  $n_{ch} = n_{ch} - 1$ , if  $n_{ch} > 0$  ir  $A_{server} = False$ , kartoti 3-5 punktus.

7 žingsnis. If  $A_{server} = False$ .

8 žingsnis. If  $N_{cur} < N_{SP}$ , tai atsižvelgiant į  $SP_{disc}$  paraiškos numeris  $apl_k$  talpinamas į *FIFO/LIFO* eilę,  $N_{cur} = N_{cur} + 1$ ,  $k = k + 1$ .

9 žingsnis. Jei 8 žingsnyje užrašyta sąlyga nėra tenkinama, tai  $L = L + 1$ ,  
 $k = k + 1$ .

10 žingsnis. 2-9 žingsniai kartojami tol kol nebus aptarnauta  $N$  paraiškų.  
 Gauti rezultatai perduodami į rezultatų analizės posistemę, vėliau į įvesties/išvesties posistemę.

Šiame skyrelyje aprašytas matematinis tinklo srauto modelis panaudotas daugiakanalės tinklo mazgo simuliacijos sistemos *MulNodSimSys* kūrimui.

### 5.8. Tyrimo rezultatai GI/G/m//N tinklo mazge

Su *MulNodSimSys* sugeneruotos ir išanalizuotos 67760 laiko eilutės kombinuojant tinklo mazgo ir generuojamo srauto (*Puasono* arba savastingas pagal formules aprašytas 5.7 skyriuje) parametrus: kiekvienos generuojamos paraiškų eilės ilgis  $N=1000000$ ; tinklo mazge aptarnautų paraiškų kiekis

$n \geq 100000$ ; įeinančio/aparnaujamo srauto tipas:  $PP, PS, SP, SS$ ; įėjimo srauto intensyvumas, kai  $0.1 \leq \lambda \leq 50$ ; aptarnaujančio mazgo srauto intensyvumas, kai  $0.1 \leq \mu \leq 50$ ; savastingosios eilutės stabilumo parametras, kai  $1.1 \leq \alpha \leq 1.9$ . Kiekviena laiko eilutė buvo aptarnaujama keletą kartų keičiant tinklo mazgo parametrus:  $1 \leq N_{SP} \leq 200$  ir  $SP_{disc} \in [FIFO, LIFO]$ . Sugeneruotos eilutės pagal srauto tipą ir aptarnavimo discipliną buvo išskaidytos į 8 eilučių grupes:  $PP\ FIFO$  (išanalizuota 2420 eilučių);  $PP\ LIFO$  (išanalizuota 2420 eilučių);  $SP\ FIFO$  (išanalizuota 10890 eilučių);  $SP\ LIFO$  (išanalizuota 10890 eilučių);  $PS\ FIFO$  (išanalizuota 10890 eilučių);  $PS\ LIFO$  (išanalizuota 10890 eilučių);  $SS\ FIFO$  (išanalizuota 10890 eilučių);  $SS\ LIFO$  (išanalizuota 10890 eilučių). Tiriant įeinančio ir aptarnaujamo srauto tinklo mazge intensyvumo priklausomybę nuo srauto savybių buvo generuotos laiko eilutės perrenkant visas  $\lambda, \mu, \alpha$  ir  $SP_{disc}$  kombinacijos iš aprašytų aibių.

Tyrimo rezultatų analizei parinkti šie nepriklausomi kintamieji:  $N_{SP}, \lambda, \rho, \alpha$ . Analizuojant priklausomus kintamuosius, kur svarbus susidarančių paraiškų eilės ilgis, tinklo srautas buvo skaidomas į tris dalis:  $\rho < 1$  – įeinantis srautas yra mažesnis nei aptarnaujamas, eilė tinklo mazge baigtinė,  $\rho = 1$  – įeinantis ir aptarnaujamas srautai sutampa,  $\rho > 1$  – įeinantis yra didesnis nei aptarnaujamas srautas, eilė begalinė. Priklausomi kintamieji:  $P_L$  – paraiškų praradimo tikimybė,  $P_{serv}$  – eilutės aptarnavimo tikimybė,  $\bar{T}_{SP}$  – vidutinis paraiškos laukimo laikas eilėje,  $\bar{T}_{delay}$  – vėlinimas,  $\bar{N}_{aplsyst}$  – paraiškų skaičius liekantis sistemoje  $n+1$  paraiškai išėjus iš sistemos ir  $T_{SPfull}$  – visiško buferio užpildymo laikas.

Rezultatų tyrimui panaudota tiesinė regresinė analizė  $Y = a \cdot X + b$ , o gauti įverčiai sugrupuoti pagal  $SP_{disc}$  reikšmes, atsižvelgiant į įeinančio ir aptarnaujamo srautų savybes. Siekiant nustatyti ar tarp priklausomų ir nepriklausomų kintamųjų ryšys nėra atsitiktinis apskaičiuota  $F$  statistika ir tikimybė  $P_F$ , kad gauta  $F$  reikšmė atsitiktinė.

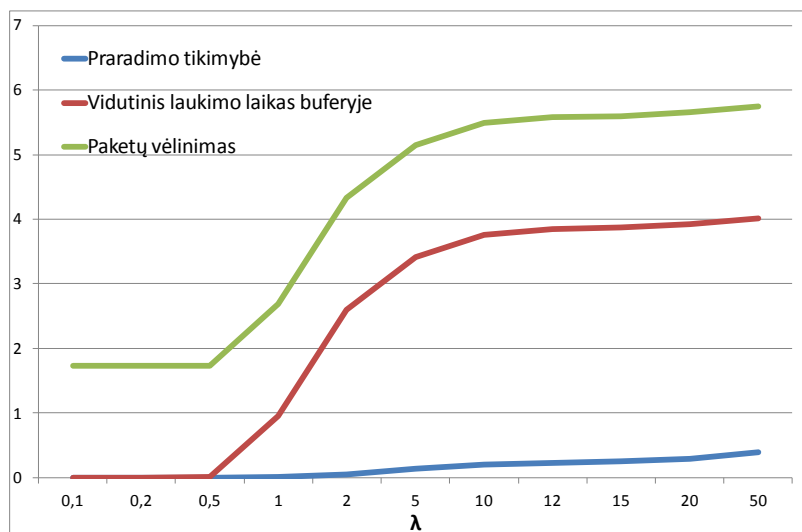
5.4 lentelė.  $N_{buf}$  ir  $SP_{disc}$  įtakos priklausomiems kintamiesiems įverčiai

	$SP_{disc} = FIFO$						$SP_{disc} = LIFO$					
	$P_L$	$P_{serv}$	$\bar{T}_{SP}$	$\bar{T}_{delay}$	$\bar{N}_{applsyst}$	$T_{SPfull}$	$P_L$	$P_{serv}$	$\bar{T}_{SP}$	$\bar{T}_{delay}$	$\bar{N}_{applsyst}$	$T_{SPfull}$
<b>PP</b>												
$a$	-0,00008	0,00008	0,10163	0,10162	0,21327	6064,52	-0,00008	0,00008	0,01072	0,1072	0,21320	-3,45204
$b$	0,14471	0,85537	-0,02904	1,70067	2,56336	1122,92	0,14469	0,85538	0,38304	2,11278	2,56440	6438,88
$r^2$	0,40692	0,40218	1	1	0,99999	1	0,40515	0,40035	0,78783	0,78751	0,99999	0,36464
$F$	4,80286	4,70925	1753822	1610126	1106408	3959315	4,76775	4,67348	25,9929	25,94205	1078099	4,01735
$P_F$	0,06453	0,06661	0	0	0	0	0,06530	0,06743	0,00140	0,00141	0	0,08507
<b>PS</b>												
$a$	-0,00008	0,00008	0,04883	0,04781	0,2099	-1,00237	-0,00008	0,00007	0,00782	0,00784	0,20995	-0,69617
$b$	0,12951	0,87058	-0,02227	1,06416	2,68813	2663,85	0,12949	0,87059	0,21482	1,29776	2,68729	2658,294
$r^2$	0,37790	0,37235	0,99999	0,9995	0,99996	0,36788	0,36361	0,3581	0,81277	0,80729	0,99996	0,28758
$F$	4,25225	4,15275	492127	143089	174069	4,07389	3,99949	3,90516	30,38682	29,32332	164586	2,82564
$P_F$	0,07812	0,08097	0	0	0	0,08333	0,08564	0,08868	0,00089	0,00099	0	0,13666
<b>SP</b>												
$a$	-0,00022	0,00022	0,09712	0,09712	0,16293	-10,02	-0,00022	0,00022	0,09342	0,09342	0,16294	-10,0211
$b$	0,11962	0,88044	-0,03029	-0,03029	2,61375	5085,48	0,11962	0,88044	0,00778	1,73703	2,61355	5084,630
$r^2$	0,59097	0,58958	0,58958	1	0,999983	0,57283	0,59119	0,58974	0,99977	0,99977	0,99998	0,57066
$F$	10,11380	10,05566	10,05566	1718320	18246	9	10,12268	10,06240	30679	30540	320208	9,30410
$P_F$	0,01549	0,01568	0,01568	0	0	0,01822	0,01546	0,01566	0	0	0	0,01858
<b>SS</b>												
$a$	-0,00018	0,00018	0,03389	0,03384	0,12163	-3,79	-0,00018	0,00018	0,03360	0,03366	0,12161	-3,65471
$b$	0,08171	0,91835	-0,01953	0,89990	2,61103	1614,44	0,08160	0,91846	-0,01878	0,90144	2,60811	1604,152
$r^2$	0,58363	0,58193	0,99997	0,99994	0,99999	0,60962	0,57105	0,56935	0,99997	0,99994	0,99999	0,56292
$F$	9,81205	9,74379	251955	117077	970050	11	9,31902	9,25456	229169	108783	756369	9,01552
$P_F$	0,01655	0,01681	0	0	0	0,01301	0,01851	0,01879	0	0	0	0,01987

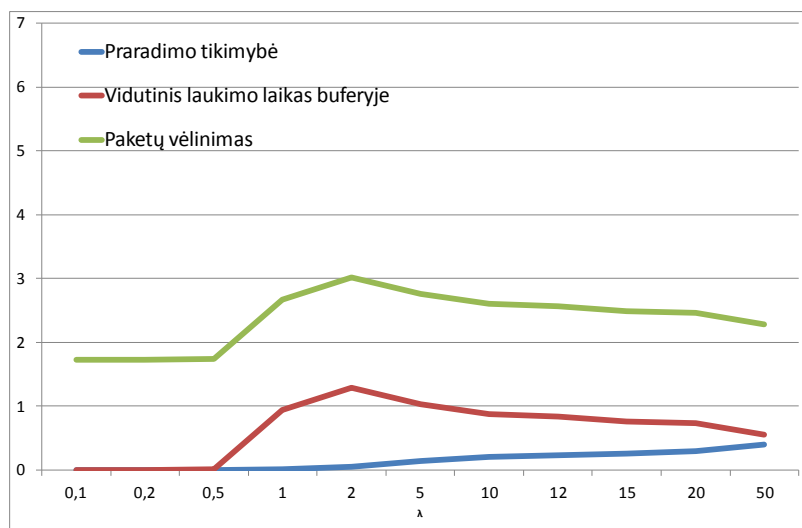
5.4 lentelėje yra analizės rezultatai, kai  $X = N_{SP}$ , o  $Y \in [P_L, P_{serv}, \bar{T}_{SP}, \bar{T}_{delay}, \bar{N}_{applsyst}, T_{SPfull}]$ . Reikia pastebėti, kad  $\bar{T}_{delay}$  PS tipo srautui yra didesnis, kai eilės aptarnavimo disciplina *FIFO*. Determinacijos koeficientas  $r^2$  rodo, kad nustatyta tiesinė priklausomybė tarp  $N_{SP}$  ir  $\bar{T}_{buf}, \bar{T}_{delay}, \bar{N}_{applsyst}, T_{SPfull}$  yra labai stipri, o tarp  $N_{SP}$  ir  $P_L, P_{serv}$  vidutiniška. Visais analizuotais atvejais Fišerio statistika  $F$  ir  $P_F$  rodo, kad stebimas ryšys tarp priklausomų ir nepriklausomų kintamųjų nėra atsitiktinis.

Išanalizavus  $\lambda$  ir  $P_L, P_{serv}, \bar{T}_{SP}, \bar{T}_{delay}, \bar{N}_{applsyst}, T_{SPfull}$  tarpusavio priklausomybę, yra stebimas ryšys tarp priklausomų ir nepriklausomų kintamųjų, kuris nėra atsitiktinis. Nustatyta, kad įeinančio srauto intensyvumas

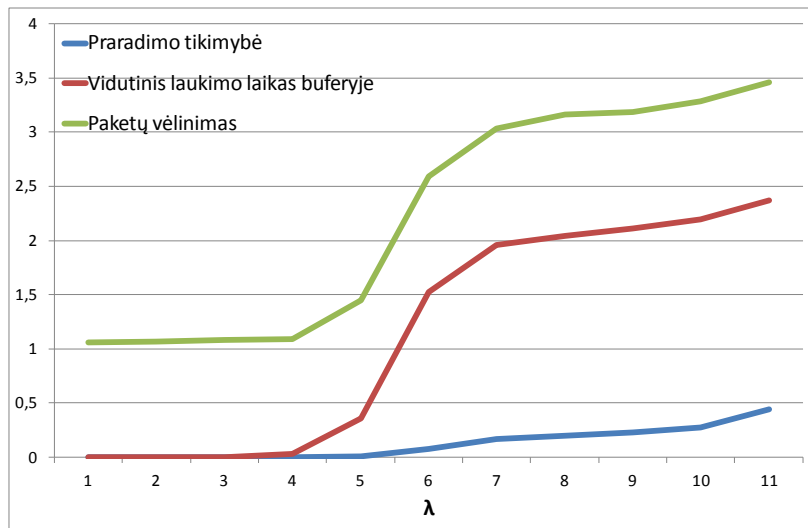
daugiausia įtakoja  $P_L, \bar{T}_{SP}, \bar{T}_{delay}$  (žr. 30 pav, 31 pav, 32 pav ir 33 paveikslus), kai srautas yra PP ir PS tipo. Reikia pažymėti, kai  $SP_{disc} = FIFO$  yra didesnis vidutinis paraiškos laukimo laikas buferyje ir paraiškos vėlinimas, t. y.  $SP_{disc} = LIFO$  geriau aptarnauja didesnę įeinantį srautą. Kai įeinantis srautas yra savastingasis (SP, SS srautai), tai buferio eilės aptarnavimo disciplina jokios įtakos paraiškų aptarnavimo parametrams neturi.



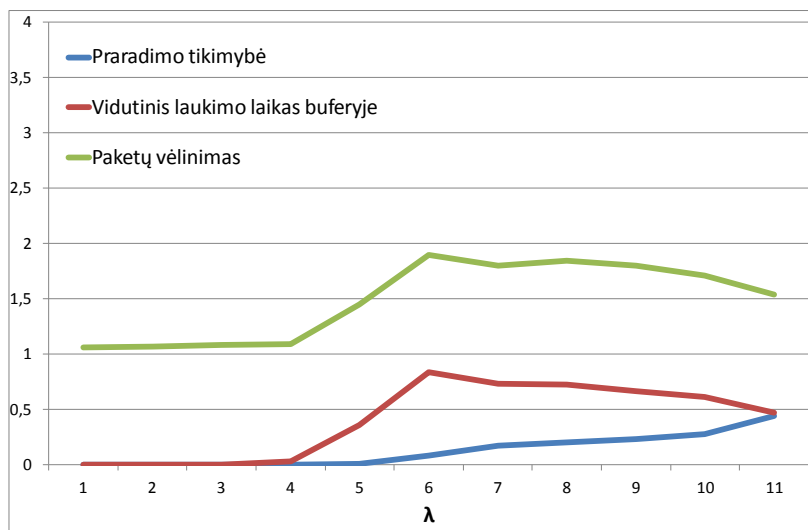
30 pav.  $P_L, \bar{T}_{SP}$  ir  $\bar{T}_{delay}$  priklausomybė nuo  $\lambda$ , kai  $SP_{disc} = FIFO$  PP srautui



31 pav.  $P_L, \bar{T}_{SP}$  ir  $\bar{T}_{delay}$  priklausomybė nuo  $\lambda$ , kai  $SP_{disc} = LIFO$  PP srautui



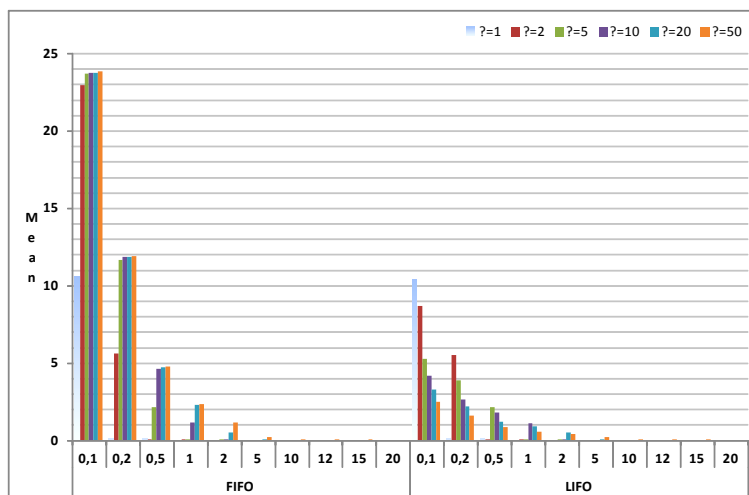
32 pav.  $P_L, \bar{T}_{SP}$  ir  $\bar{T}_{delay}$  priklausomybė nuo  $\lambda$ , kai  $SP_{disc} = FIFO$  PS srautui



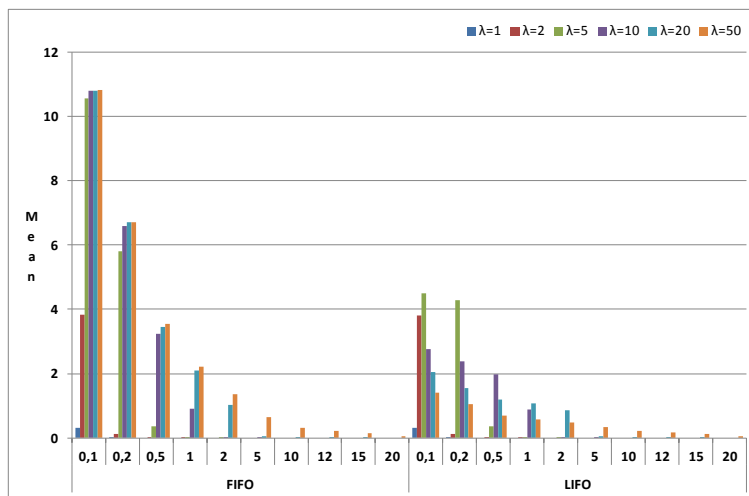
33 pav.  $P_L, \bar{T}_{SP}$  ir  $\bar{T}_{delay}$  priklausomybė nuo  $\lambda$ , kai  $SP_{disc} = LIFO$  PS srautui

Analizuotas tinklo srautas, kai  $\rho > 1$  ir sumodeliuotas tinklo mazgas patiria dideles perkrovas. Analizės tikslas – įvertinti tinklo srauto reikšmę, prie kurios paraiškų aptarnavimas tinklo mazge yra geresnis, kai eilės disciplina *LIFO*. Tam buvo fiksuotas įeinantis srautas  $\lambda \in \{1,2,5,10,12,15,20,50\}$ .  $\bar{T}_{SP}$  tyrimo rezultatų PP, PS, SP ir SS tipo srautams įvertinimai pavaizduoti 34 pav, 35 pav, 36 pav ir 37 paveiksluose.  $\bar{T}_{delay}$  tyrimo rezultatų PP, PS, SP ir SS tipo srautams įvertinimai pavaizduoti 38 pav, 39 pav, 40 pav ir 41 paveiksluose. Apskaičiuoti  $\bar{T}_{SP}$  ir  $\bar{T}_{delay}$  vidutinės reikšmės bei standartiniai nuokrypiai.

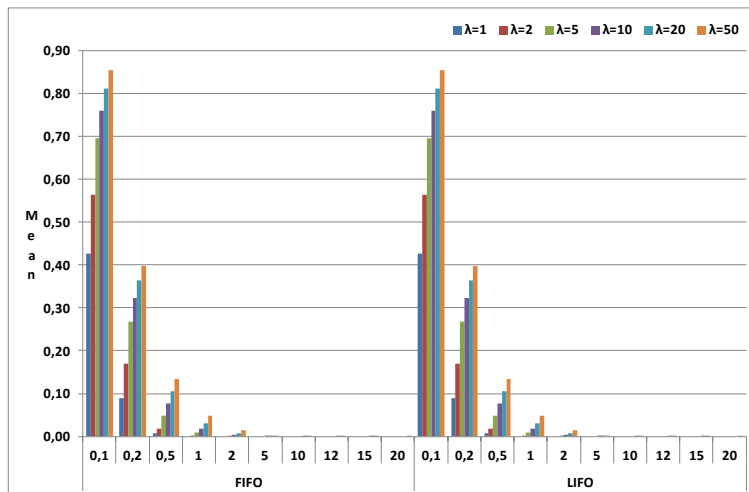
Reikia pažymėti, kad analizės rezultatai rodo, kad kai  $1 < \rho < 10$ ,  $\bar{T}_{SP(LIFO)} \approx \bar{T}_{SP(FIFO)}$  ir  $\bar{T}_{delay(FIFO)} \approx \bar{T}_{delay(LIFO)}$ . Kai  $\rho \geq 10$ , tai  $P_{serv(FIFO)} < P_{serv(LIFO)}$ , t. y. paraiškos aptarnaujamos daugiau nei du kartus sparčiau kai taikoma *LIFO* eilės aptarnavimo disciplina, o  $P_{L(FIFO)} > P_{L(LIFO)}$ . Apibendrinant tyrimo rezultatus galima teigti, kad PP tipo srautams, kai  $\lambda$  daugiau nei 10 kartų viršija  $\mu$  geriau paraiškos aptarnaujamos, kai tinklo mazgo eilės disciplina *LIFO*. Analogiškai rezultatai gauti PS tipo srautams, kai  $\lambda$  daugiau nei 10 kartų viršija  $\mu$  geriau paraiškos aptarnaujamos, kai tinklo mazgo eilės disciplina *LIFO*. SP ir SS tipo srautams *FIFO* ir *LIFO* eilių aptarnavimo disciplina  $\bar{T}_{SP}$  ir  $\bar{T}_{delay}$  jokios įtakos neturi.



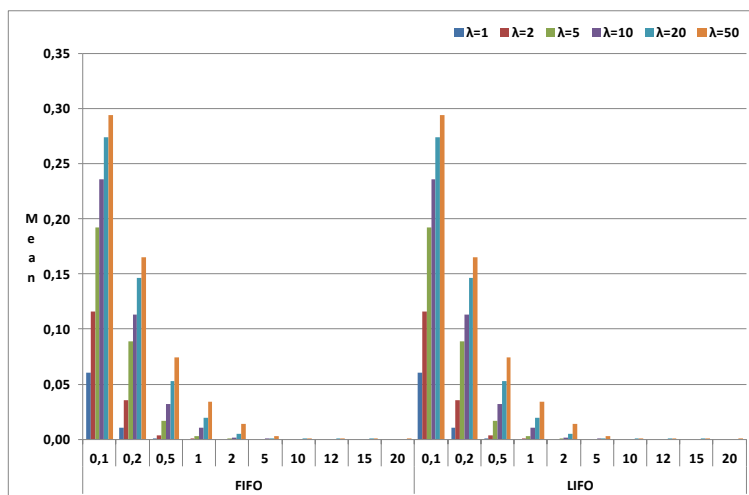
34 pav.  $\bar{T}_{SP}$  įverčiai, kai  $\rho > 1$  PP tipo srautams



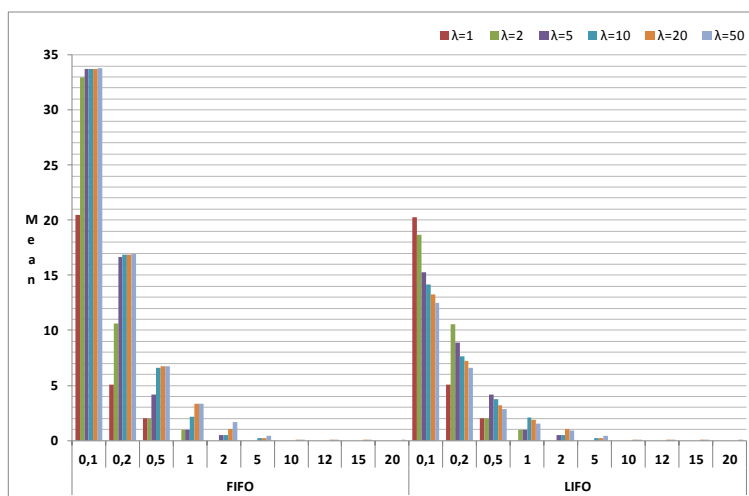
35 pav.  $\bar{T}_{SP}$  įverčiai, kai  $\rho > 1$  PS tipo srautams



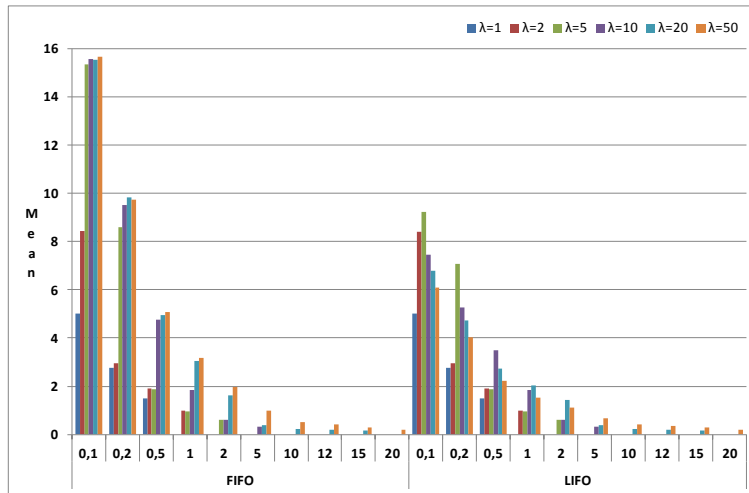
36 pav.  $\bar{T}_{SP}$  įverčiai, kai  $\rho > 1$  SP tipo srautams



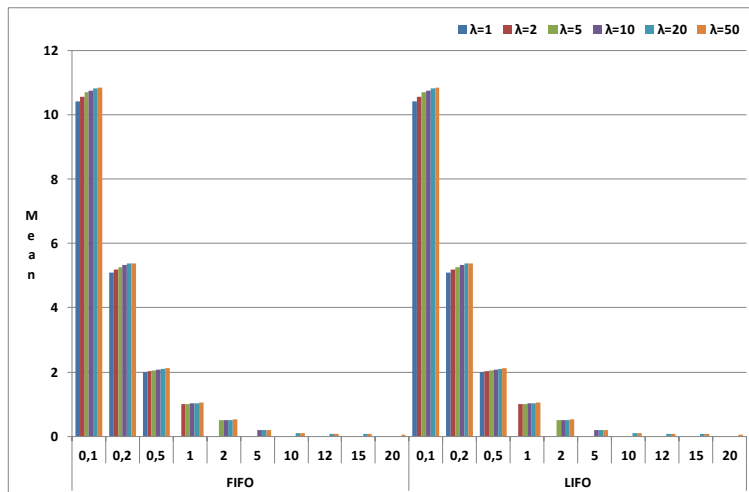
37 pav.  $\bar{T}_{SP}$  įverčiai, kai  $\rho > 1$  SS tipo srautams



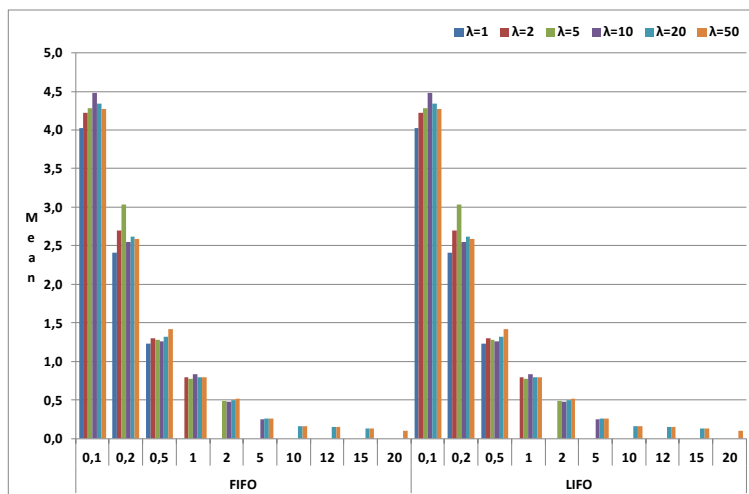
38 pav.  $\bar{T}_{delay}$  įverčiai, kai  $\rho > 1$  PP tipo srautams



39 pav.  $\bar{T}_{delay}$  įverčiai, kai  $\rho > 1$  PS tipo srautams



40 pav.  $\bar{T}_{delay}$  įverčiai, kai  $\rho > 1$  SP tipo srautams



41 pav.  $\bar{T}_{delay}$  įverčiai, kai  $\rho > 1$  SS tipo srautams



Įvertinus  $\bar{T}_{SP}$  ir  $\bar{T}_{delay}$  priklausomybę nuo srauto tipo, nustatyta, kad SP tipo srautui  $\bar{T}_{SP}$  yra daugiau 14 kartų mažesnis, o  $\bar{T}_{delay}$  daugiau nei 1,4 karto mažesnis nei PS ir PP tipams srautams. SS tipo srautui  $\bar{T}_{SP}$  yra daugiau 36 kartus mažesnis,  $\bar{T}_{delay}$  daugiau nei 3,4 karto mažesnis nei PS ir PP tipams srautams.

5.5 lentelė. Kanalų apkrovos įverčiai, kai  $\rho > 1$

	$\lambda$	0,2	0,5	1	2	5	10	12	15	20	50
<b>PP</b>											
$\overline{N(n_{ch})}$	<b>FIFO</b>	10000	10000	10000	10000	10000	9984	8362	6681	5006	2003
$\overline{N(n_{ch})}_{std}$		11609	4986,8	478,96	53,994	27,584	9,051	39,06	31,98	19,25	10,46
$\overline{N(n_{ch})}$	<b>LIFO</b>	10000	10000	10000	10000	10000	9990	8369	6667	5008	1998
Std.dev.		11609	4985,5	489,43	32,037	52,664	27,99	36,48	17,57	12,95	15,11
<b>PS</b>											
$\overline{N(n_{ch})}$	<b>FIFO</b>	10000	10000	10000	10000	10000	10000	10000	10000	9961	9179
$\overline{N(n_{ch})}_{std}$		16742	12454	10091	7495,3	4663,6	3708	4812	5079	4603	2416
$\overline{N(n_{ch})}$	<b>LIFO</b>	10000	10000	10000	10000	10000	10000	10000	10000	9960	9184
$\overline{N(n_{ch})}_{std}$		16742	12454	10091	7496	4668,1	3708	4814	5080	4596	2415
<b>SP</b>											
$\overline{N(n_{ch})}$	<b>FIFO</b>	10000	10000	10000	10000	10000	9935	9916	9878	9820	9505
$\overline{N(n_{ch})}_{std}$		4697,8	3905,8	4948,2	5444,1	5259,7	6096	7975	8956	9425	8058
$\overline{N(n_{ch})}$	<b>LIFO</b>	10000	10000	10000	10000	10000	9935	9916	9878	9820	9505
$\overline{N(n_{ch})}_{std}$		4695,8	3905,7	4949,2	5445	5260,5	6096	7975	8956	9425	8058
<b>SS</b>											
$\overline{N(n_{ch})}$	<b>FIFO</b>	10000	10000	10000	10000	10000	10000	10000	9984	9965	9808
$\overline{N(n_{ch})}_{std}$		9985	8079,8	7546,2	6832,4	5449,2	5384	6601	7059	7199	5170
$\overline{N(n_{ch})}$	<b>LIFO</b>	10000	10000	10000	10000	10000	10000	10000	9986	9963	9802
$\overline{N(n_{ch})}_{std}$		10352	8068,3	7548,8	6832,6	5462,3	5382	6598	7070	7199	5176

Išanalizavus  $\alpha$  ir  $P_L, P_{serv}, \bar{T}_{SP}, \bar{T}_{delay}, \bar{N}_{applsyst}, T_{SPfull}$  tarpusavio priklausomybę, yra stebimas ryšys tarp priklausomų ir nepriklausomų kintamųjų, kuris nėra atsitiktinis. Didėjant savastingumo koeficientui nežymiai didėja  $\bar{N}_{applsyst}$  skaičius sistemoje, tačiau priklausomybės nuo eilės aptarnavimo disciplinos nenustatyta.

Įvertintas kanalų apkrovos tolygumas, kai  $\rho < 1$ ,  $\rho = 1$  ir  $\rho > 1$ , o nepriklausomi kintamieji  $\lambda$  ir  $\mu$ . Netolygiausiai apkrauti kanalai, kai  $\rho < 1$  ir  $\rho = 1$ , nes įeinantis srautas yra mažesnis arba sutampa su mazgo aptarnavimo sistemos pajėgumais. Šiais atvejais eilės aptarnavimo disciplina jokios įtakos kanalų darbui neturi.

Kai  $\rho > 1$  atrinkti tik tie matavimai kai įeinantis srautas visada didesnis už tinklo mazgo paraiškų aptarnavimo galimybes (žr. 5.5 lentelę). Apskaičiuoti kanaluose aptarnautų paraiškų vidurkiai  $\overline{N(n_{ch})}$  ir standartiniai nuokrypiai  $\overline{N(n_{ch})}_{std}$ , didėjant įeinančio srauto intensyvumui bei keičiant įeinančio ir aptarnaujamo srautų sudėtį. Analizuojant  $\overline{N(n_{ch})}_{std}$  pokytį matyti, kad didėjant įeinančio srauto intensyvumui kanalai apkraunami tolygiau. Kai tinklo mazgo aptarnavimo srautas yra savastingasis, tai tik kai  $\frac{\lambda}{\mu} > 12$  tinklo mazgo kanalai nebepajėgūs aptarnauti visų atėjusių paraiškų ( $n = 100000$ ,  $m = 10$ ,  $\frac{n}{m} < 10000$ ), o esant Puasono srautui  $\frac{\lambda}{\mu} > 5$ .

Galima teigti, kad kai tinklo mazgo srautas pasižymi savastingumu, jo paraiškų aptarnavimo galimybės yra didesnės, esant tam pačiam įeinančio paraiškų srauto intensyvumui.

### 5.9. Skyriaus išvados

Panaudojant sukurtą tinklo srauto modeliavimo paketą *MulNodSimSys* ištirta įėjimo bei aptarnavimo srautų savastingumo įtaka tinklo mazgo darbui, atsižvelgiant į aptarnavimo eilės discipliną ir buferio talpą, apkrovos ir aptarnavimo intensyvumus bei savastingumą.

Nustatyta, kad, didėjant tinklo srauto intensyvumui, savastingumu pasižymintis srautas yra aptarnaujamas stabiliai. Aptarnaujant Puasono srautą didėja paketų praradimo tikimybė, didėjant įeinančio srauto intensyvumui.

Eilės aptarnavimo disciplina *LIFO PS* tipo sraute užtikrina geresnę aptarnavimo kokybę.

Didėjant tinklo mazgo buferio talpai, srauto, pasižyminčio savastingumu, aptarnavimo tikimybė didėja. Buferio eilės aptarnavimo disciplina *LIFO* geriau aptarnauja *PS* tipo srautą. Didėjant savastingosios eilutės stabilumo parametrai, aptarnavimo tikimybė didėja. Stipresnė priklausomybė stebima, kai srautas yra *SS* tipo.

Remiantis tiesinės regresijos rezultatais ir determinacijos koeficientu  $r^2$  tarp  $N_{SP}$  ir  $\bar{T}_{SP}, \bar{T}_{delay}, \bar{N}_{applsyst}, T_{SPfull}$  nustatyta labai stipri, o tarp  $N_{SP}$  ir  $P_{loss}, P_{serv}$  vidutiniška priklausomybė, o  $F$  ir  $P_F$  rodo, kad stebimas ryšys tarp priklausomų ir nepriklausomų kintamųjų nėra atsitiktinis. Nustatyta, kad  $\bar{T}_{delay}$  *PS* tipo srautui yra didesnis, kai eilės aptarnavimo disciplina *FIFO*.

Nustatyta, kad stebimas ryšys tarp  $\lambda$  ir  $P_{loss}, P_{serv}, \bar{T}_{SP}, \bar{T}_{delay}, \bar{N}_{applsyst}, T_{SPfull}$  nėra atsitiktinis. Gautų reikšmių analizės rezultatai parodė, kad įeinančio srauto intensyvumas daugiausia įtakoja  $P_{loss}, \bar{T}_{SP}$  ir  $\bar{T}_{delay}$ , o eilės aptarnavimas yra geresnis, kai aptarnavimo disciplina *LIFO* (kai  $\rho \geq 10$ ) *PP* ir *PS* tipo srautams. *SP* ir *SS* tipo srautams buferio eilės aptarnavimo disciplina jokios įtakos paraiškų aptarnavimo parametrų neturi.

Nustatyta, kad kai įeinantis srautas yra savastingasis, tai  $\bar{T}_{SP}$  ir  $\bar{T}_{delay}$ , nepriklausomai nuo eilės aptarnavimo disciplinos yra mažesnis, kai visi kiti generuojamo srauto parametrai yra vienodi.

Nustatyta, kad  $\alpha$  ir  $P_{loss}, P_{serv}, \bar{T}_{SP}, \bar{T}_{delay}, \bar{N}_{applsyst}, T_{SPfull}$  yra tarpusavyje priklausomi ir šis ryšys nėra atsitiktinis. Išanalizavus skaičiavimo rezultatus gauta, kad didėjant savastingumo koeficientui nežymiai didėja  $\bar{N}_{applsyst}$  skaičius sistemoje. Kitų kintamųjų ir  $\alpha$  priklausomybės visiškai neįtakoja eilės aptarnavimo disciplina.

Įvertinus kanaluose aptarnautų paraiškų  $\overline{N(n_{ch})}$  ir  $\overline{N(n_{ch})}_{std}$  gauta, kad didėjant įeinančio srauto intensyvumui kanalai apkraunami tolygiau.

Pažymėtina, kad esant tam pačiam įeinančio srauto intensyvumui savastingojo tinklo mazgo srauto aptarnavimo našumas yra daugiau nei dvigubai didesnis nei tinklo mazgo, kurio srautas yra *Puasono*.

## 6 skyrius. Savastingumo matavimo realiu laiku metodas

Kompiuterių tinklo srauto savastingumo savybę galima sėkmingai naudoti tinklo perkrovų reguliavimui (He, Gao, Hou, Park 2004, Chung, Seol, Choi, Yeoum, Lim 2008, Kaklauskas, Sakalauskas 2010b). Vertinant kompiuterių tinklo paketinio srauto savybes realiu laiku, galima dinamiškai prognozuoti srauto kaitą (Mark, Ramamurthy 1998, Roughan, Veitch, Abry 2000, Hagiwara, Doi, Tode, Ikeda 2001, Chandramathi, Shanmugavel 2003, Younis, Fahmy 2005, Fu 2006, Zaidi, Hakami, Landfeldt, Moors 2010, Baccarelli, Zhu 2011, Cordeschi, Patriarca 2012). Tinklo srauto įverčius galima taikyti jo aptarnavimo kokybės *QoS* gerinimui, užtikrinant našesnę paketų aptarnavimą tinklo mazge, mažinant paketų užlaikymą (Quan, Chung 2004, Mao 2005, Smith 2006, Lianming, Jianping, Xiaoheng 2011). Galima keisti tinklo mazgo eilės ilgį, eilės aptarnavimo discipliną bei kitus parametrus, kurie geriausiai atitinka dinamiškai įvertinto srauto savybes (Kaklauskas, Sakalauskas 2011a).

Reikia pažymėti, kad tinklo srauto apkrovos savastingumo savybių tyrimo realiu laiku metodai dar nėra pakankamai ištirti. Šio skyrelio tyrimų sritis apima kompiuterių tinklo paketinio srauto savastingumo savybių realiu laiku įvertinimo metodų tyrimą, atsižvelgiant į tinklo srauto savybes bei tinklo mazgo parametrus.

Skaičiuojant savastingumo parametrus realiu laiku atsiranda dar vienas svarbus parametras – skaičiavimo laikas. Sukurtas savastingumo skaičiavimo metodas realiu laiku vykdomas 3 etapais: 1 – eilutės formavimas realiu laiku (plačiau 4.1.1 ir 4.1.2 skyriuose), 2 – savastingumo įverčių skaičiavimas (plačiau 4.4 skyriuje), 3 – rezultatų perdavimas (plačiau 6.2 skyriuje).

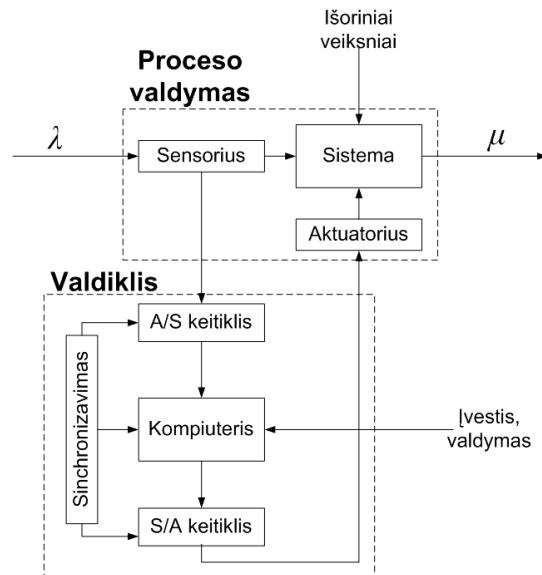
Formuojant agreguotą eilutę realiu laiku, gautoji eilutė nėra perrašoma iš naujo, o rekurentiškai atnaujinama, taikant slenkančio lango principą ir naudojant eilutės perindeksavimo algoritmą, kuris plačiau aprašytas 4.1.2 skyriuje.

### **6.1. Kompiuterių tinklo srauto valdymo modelis**

Kompiuterių tinklo valdymo modelį sudaro aibė tarpusavyje susijusių komponentų, kurie veikdami kartu užtikrina efektyvų tinklo vartotojų aptarnavimą (Aström, Murray 2008). Modelio komponentai – tai tarpusavyje sąveikaujančios įvedimo, apdorojimo ir išvedimo dalys. Komponentai turi atributus, nusakančius jų savybes. Valdymo modelio elementų tarpusavio sąveiką užtikrina suprojektuoti ryšiai tarp komponentų ir atributų. Modeliuojamos sistemos valdymas aprašomas algoritmu (Adler, Sitaraman, Venkataramani 2011) arba vykdomas naudojant per grįžtamąjį ryšį gaunamą informaciją (Mahmoud, Hamid 2011), t. y. matuojant veikiančio modelio komponentų charakteristikas ir skaičiuojant jų įverčius. Suprojektuotą kompiuterių tinklo srauto valdymo modelį sudaro dinaminiai komponentai – tai dalys atliekančios veiksmus, apjungiančios veikiančius modelio elementus ir užtikrinančius kompiuterių tinklo srauto valdymą. Valdymo komandos perduodamos per sistemos kintamuosius, kurie aprašo jos būsenas. Suprojektuotą valdymo sistemą sudaro tinklo mazgas (žr. 42 pav. *Proceso valdymas*), kuriam keičiamas eilės ilgis, eilės aptarnavimo disciplina, atsižvelgiant į tinklo srauto įverčius (žr. 42 pav. *Valdiklis*). Šie pokyčiai užtikrina optimalią kompiuterių tinklo paketinio srauto aptarnavimo kokybę QoS.

Tinklo srauto savastingumo vertinimui sukurta programinių modulių biblioteka *SSE-OL* (plačiau apie biblioteką 6.6 skyriuje). Ją rekomenduojama įdiegti į kompiuterį, prijungtą prie tinklo mazgo (žr. 42 pav. *Kompiuteris*). Tinklo mazgas į kompiuterį persiunčia gaunamų paketų antraščių kopijas (plačiau 6.6 skyriuje). Suprojektuotame tinklo srauto valdymo modelyje mazgas atlieka sistemos sensorinio prietaiso vaidmenį (žr. 42 pav. *Sensorius*). Jei tinklo mazge nėra paketų antraščių kopijavimo ir perdavimo į prievadą funkcijos, tai, jei yra galimybė, galima įdiegti laisvai platinamas programinių modulių bibliotekas. Pavyzdžiui, *ulogd* programą *Linux* operacinei sistemai (Ulogd 2012). Plačiau *ulogd* naudojimas aptartas 3.1 skyriuje. Kompiuteris su

įdiegta *SSE-OL* modelyje atlieka valdymo prietaiso vaidmenį, t. y. ji matuoja paketinio tinklo srauto charakteristikas realiu laiku (žr. 42 pav. *Kompiuteris*) ir, atsižvelgiant į gautus rezultatus, per kintamuosius keičia tinklo mazgo eilės ilgį, eilės aptarnavimo discipliną (žr. 42 pav. *Aktuatorius*).



**42 pav.** Atviras tinklo mazgo valdymo modelis su grįžtamuju ryšiu

Iš proceso valdymo modulio įeinančių paketų kopijos į valdiklį perduodamos kaip analoginis signalas, kurį analoginis/skaitmeninis keitiklis keičia į skaitmeninį (žr. 42 pav. *A/S keitiklis*). Iš valdiklio į tinklo mazgą perduodamas apdorotas valdymo signalas (žr. 42 pav. *Sistema*) vėl turi būti keičiamas į analoginį, naudojant skaitmeninį/analoginį keitiklį (žr. 42 pav. *S/A keitiklis*). Procesų sinchronizavimui valdiklyje naudojamas sinchronizavimo įrenginys (žr. 42 pav. *Sinchronizavimas*). Jis užtikrina suprojektuoto modelio komponentų sąveiką ir tvarkingus informacijos mainus.

Modelyje numatyta galimybė tinklo administratoriui per išorinius įrenginius valdyti kompiuterį, t. y. koreguoti suprojektuoto modelio procesus, keičiant parametrus bei atlikti kitus programinių modulių bibliotekos *SSE-ON* valdymo veiksmus (žr. 42 pav. *Įvestis, valdymas*). Administratoriui taip pat leidžiama per išorinius valdymo įrenginius prisijungti ir koreguoti tinklo mazgo parametrus (žr. 42 pav. *Išoriniai veiksniai*). Aprašytas paketinio kompiuterių tinklo valdymo modelis gali būti realizuotas tik bendradarbiaujant

su populiariausiais tinklo įrangos gamintojais, nes tam reikia keisti tinklo mazgo programinę įrangą, kurios autorinių teisių savininkai yra įrangos gamintojai. Tokio pobūdžio tyrimai numatyti tolesniuose darbų planuose. Naudojamoje programinių modulių bibliotekoje *SSE-OL* integruoti du paketinio kompiuterių tinklo srauto savastingumo analizės realiu laiku metodai: *robastinis regresijos* metodas ir *IR* statistika.

Kitas paketinio kompiuterių tinklo srauto valdymo sprendimas yra rankinis tinklo mazgo parametrų keitimas pagal tinklo srauto savybių įverčius. Sprendimą dažniausiai priima kompiuterių tinklo administratorius. Jis, atsižvelgdamas į tinklo srauto savastingumo tyrimo rezultatus, parenka optimalius tinklo mazgo parametrus užtikrinančius geriausią tinklo srauto aptarnavimo kokybę (minimalūs paketų praradimai, mažiausias užlaikymas tinklo mazguose). Prieš priimant sprendimą reikėtų atsižvelgti į šias svarbiausias rekomendacijas:

- Jei įeinantis į tinklo mazgą srautas intensyvus ir nepasižymi savastingumu, tai rekomenduojama rinktis tinklo mazgus, kurių eilės aptarnavimo disciplina *LIFO*;
- Savastingajam srautui tinklo mazgo eilės aptarnavimo disciplina paraiškų aptarnavimui įtakos neturi;
- Savastingajam srautui didesnė tinklo mazgo buferio talpa užtikrina geresnį tinklo srauto aptarnavimą;

## **6.2. Savastingumo matavimo realiu laiku metodų apžvalga**

Perduodant kompiuterių tinklais informaciją svarbu, kad siuntėjo išsiusti duomenų paketai sėkmingai pasiektų gavėją. Žinoma, kad kiekvienas kompiuterių tinklo vartotojas gauna paslaugą tuoj pat po jos pareikalavimo, todėl atskiruose tinklo segmentuose dažnai atsiranda nenumatytos tinklo perkrovos, sąlygojančios duomenų praradimus tinkle. Siekiant užtikrinti duomenų perdavimo patikimumą, kompiuterių tinkle siekiama prognozuoti tinklo segmentų perkrovas ir imtis veiksmų, mažinančių perduodamos informacijos praradimus.



Khalid M. Khalil pasiūlė metodą ir sistemą srauto pliūpsniškumo analizei realiu laiku (Khalid 1994). Čia siūloma fiksuoti paketo atėjimo į sistemą laiką taip formuojant laiko eilutes, kurioms realiu laiku yra skaičiuojami aprašomosios statistikos parametrai: variacijos koeficientas (nulis kai yra determinuotas srautas, 1 ir didesnis nei 1 kai srautas yra *Puasono*), koreliacijos koeficientas, dispersijos koeficientas. Dispersijos koeficientas charakterizuoja srauto pliūpsniškumą. Sukurtas metodas neanalizuoja tinklo srauto pliūpsnių pasikartojimo galimybes, t. y. srauto savastingumo. Ši savybė leidžia prognozuoti būsimus tinklo srauto pliūpsnius ir, atsižvelgiant į tai, imtis prevencinių priemonių.

San-Qi Li ir Chia-Lin Hwang pasiūlė metodą ir techninį sprendimą skirtą srauto matavimui ir eilių aptarnavimo įvertinimui posistemų integravimą į kompiuterių tinklą (Li, Hwang 1994). Čia taikomos dvi technologijos, tai signalų apdorojimas ir eilių analizė. Įrankis apima tris pagrindinius komponentus: srauto matavimas, statistinė analizė ir eilių analizė. Srauto matavimo komponente naudojamos standartinės signalų skaičiavimo technologijos, kad apskaičiuoti pusiausvyros ir antros eilės statistiką srautui. Antrasis komponentas naudojamas specialios *Markovo* grandinės formavimui, kuri galim būti statistiškai palyginta su duotuoju srautu. Eilių analizės komponentas naudoja generalizuotą *Foldingo* algoritimą, kad rasti pusiausvyros ir nuo laiko priklausančius eilučių analizės sprendimus dėl naudojamo buferio dydžio, ryšio pajėgumų, praradimų dažnio, perkrovų valdymo ir aptarnavimo elgsenos.

(McKee, Phaal 1995) pasiūlytas metodas ir techninė įranga paketinio tinklo testavimui leidžia pamatuoti komunikavimo parametrus tarp dviejų specifinių tinklo stočių. Keičiant paketo dydį sekoje yra paskaičiuojamos pralaidos, vėlinimo ir tinklo vidinio paketo dydžio charakteristikos. Paketų pliūpsniškumas apsprendžia buferio dydį ir pagal tai gali būti nustatytos naujos sekos savybės. (DePertillo 2000) pasiūlė laiko eilučių *Hurst'o* eksponentės nustatymą būdą, kai pagal iš anksto suformuotas laiko eilutes ir pagal gautą statistinę informaciją apskaičiuojamas *Hurst'o* koeficientas. Lim Or Sim, Yee

Han Cheong ir kiti pasiūlė tinklo resursų stebėjimo ir matavimo sistemą bei metodą (Sim, Cheong, Jarrett, Bey, Eustace, Pettit 2003), kuriuo siūloma daugelio resursų kompiuterių tinklo informaciją surinkti į specialų ataskaitų serverį ir čia analizuoti gautą informaciją. (Dugatkin, Hannel 2004a) tinklo srauto rinkimui ir savybių analizei pasiūlė automatinio kompiuterių tinklo srauto fiksavimo ir charakterizavimo metodą, kai naudojamas grįžtamasis ryšys. Aprašomą sistemą sudaro paketų savybių surinkimo, charakterizavimo ir grįžtamojo ryšio valdymo komponentai. Čia taip pat pateikiamos tinklo srauto savybės be srauto prognostinių įverčių. Kitame išradime D. Dugatkin ir C. Hannel siūlo, panaudojant fizinius įrenginius su keliomis tinklo plokštėmis, surinkti ir automatiškai analizuoti tinklo srautą (Dugatkin, Hannel 2004b). Sistemą sudaro realus tinklas ir testuojamas tinklas. Pagal gautus tinklo srauto analizės įverčius generuojamas išeinančio srauto parametrų aprašas, kuris panaudojamas išeinančio srauto simuliacijai.

H. V. Tran pasiūlė skambučių priėmimo kontrolės nehomogeniniams resursams metodą (Tran 2005). Sukurta sistema, kuri nehomogeniniams savastingiesiems resursams parenka tinkamus tinklo parametrus. Pirmiausiai konstruojamas difuzijos modelis, nustatant, kad įeinantis srautas yra Gauso triukšmo srautų rinkinys turintis įvairius *Hurst'o* įverčius. Aproximavus srautą Weibulio funkcija ir atsižvelgiant į difuzijos modelį, prognozuojamas naujo srauto atėjimas, kuris pagrįstas apskaičiuota paketų pradžios tikimybe. W. Olesinski ir P. Rabinovitch pasiūlė trasavimu grįstą statistinį metodą srauto klasifikavimui realiu laiku (Olesinski, Rabinovitch 2007). Naudojant tinklo srauto statistinę analizę realiu laiku jis klasifikuojamas ir pagal nustatytus domenų formuojama srauto sudėties lentelė. K. Chao, M. Aziz, H. Wei ir S. Jimin pasiūlė tinklo stebėjimo sistemą, kuri skaičiuoja srauto išsibarstymą ir vidurkį (Chao, Aziz, Wei, Jimin 2007). Sistema stebi kompiuterių tinklą, kuriame vykdomas duomenų perdavimas paketais ir tikrina paketus atsižvelgiant į aprašytą požymių aibę. Čia galima nurodyti laiko intervalą, per kurį būtų apskaičiuotas nurodytų paketų variacijos koeficientas bei vidurkis. Sistema automatiškai reaguoja į srauto pliūpsnius, kai viršijama nustatyta riba.

D. Dugatkin pasiūlė įrangą ir metodą savastingojo srauto generavimui telekomunikacijų tinkluose, pagal iš anksto aprašytas *Hurst'o* koeficiento ir kitas reikšmes (Dugatkin 2007).

Artimiausią sukurtam savastingumo analizės realiu laiku metodui išradimą pasiūlė (Khalid 1994), tai duomenų srauto pliūpsniškumo analizės realiu laiku būdas, kai fiksuojamas duomenų paketo atėjimo į sistemą laikas taip formuojant laiko eilutes. Joms realiu laiku apskaičiuotos aprašomosios statistikos: variacijos koeficientas, koreliacijos koeficientas, dispersijos koeficientas. Žinoma, kad dispersijos koeficientas charakterizuoja srauto pliūpsniškumą. Reikia pažymėti, kad šis sprendimas nepateikia srauto savastingumo įverčių. Kitas patentas (Veitch, Roughan, Abry 1999) tinklo srauto analizei realiu laiku naudoja suformuotų laiko eilučių bangelių analizės būdą. Jo trūkumas yra tas, kad šiame išradime naudojamas tik vienas srauto analizės ir vienas agregavimo metodas. (Mario, Nuno, Hajduczenia, Inacio, Monteiro, Silva 2008) siūlo savastingumo laipsnį matuoti, dinamiškai skaičiuojant srauto *Hurst'o* koeficientą. Pirmas koeficientas apskaičiuojamas iš pirmos laiko eilutės, kuri suformuota skaičiavimo modulyje iš realių duomenų sekos, o kiti gaunami prie einamosios pridėjus naują narį ir apskaičiavus naują *Hurst'o* koeficientą iš naujai gautos laiko eilutės pradedant antruoju nariu. Skaičiavimams naudojama R/S statistika, laiko dispersijos, liekanos dispersijos, laiko absoliutinių momentų, siaurinio didžiausio tikėtimumo, signalo bangelių įverčių, įterptojo šakojimo, betrendės fliktuacijos analizės metodai. Čia nėra laiko eilučių agregavimo. (Mario, Nuno, Hajduczenia, Monteiro, Silva 2007) tinklo srauto analizės įrenginys automatiškai formuojama 1000 narių laiko eilutę, kuri vėliau papildoma naujais sekos nariais, nekeičiant bendro eilutės ilgio. Eilutė formuojama buferyje, čia skaičiuojama R/S statistika, laiko dispersija, liekanos dispersija, laiko absoliutiniai momentai, siaurinio didžiausias tikėtimumas, įvertinamos signalo bangelės, taikomi įterptojo šakojimo bei betrendės fliktuacijos analizės metodai. Rezultatai klasifikuojami ir saugomi išeities buferyje. Šio išradimo trūkumas tas, kad yra fiksuotas eilutės ilgis ir matavimams

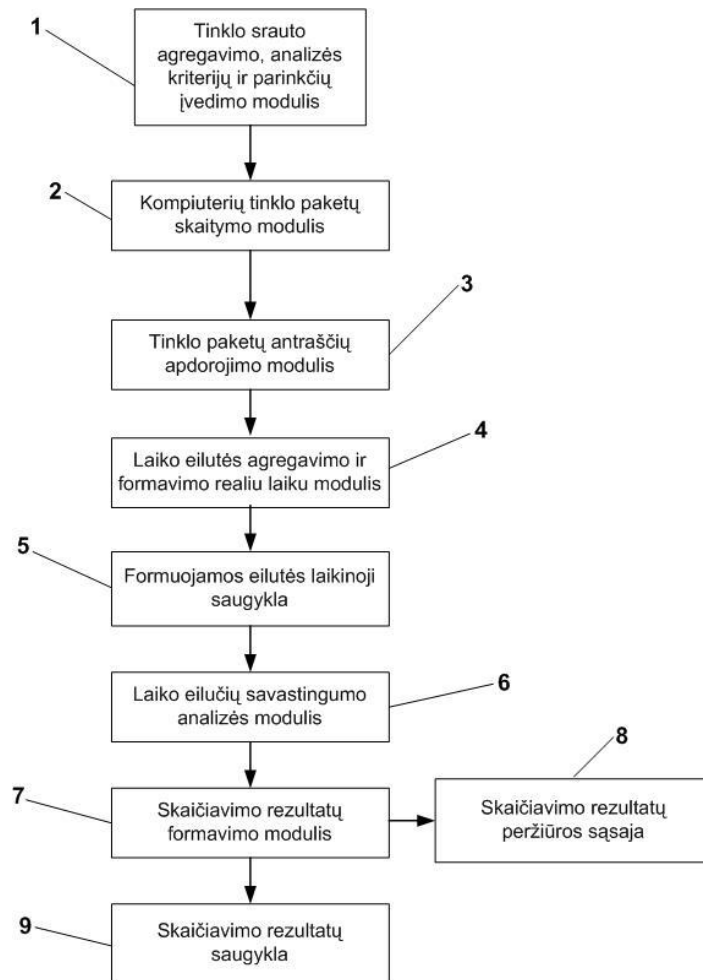
naudojamas tik vienas agregavimo metodas, o *Hurst'o* koeficientai nėra išsaugomi tolesnei analizei.

### **6.3. Tinklo srauto analizės realiu laiku metodas**

Išradimas priskiriamas skaitmeninės informacijos stebėjimo ir testavimo sistemoms, kai kompiuterių tinkluose naudojamas paketinis duomenų perdavimas ir skirtas tinklo srauto analizei realiu laiku (Kaklauskas, Sakalauskas 2011c). Išradimo tikslas padėti įvertinti tinklo perkrovų pasikartojimo galimybę ir priimti sprendimą dėl nereikšmingų srauto dedamųjų apribojimo ar techninės įrangos keitimo. Siūlomas būdas leidžia pasirinkti tinklo srauto agregavimo metodą ir parametrus, pasižymi atsparumu klaidoms bei didele skaičiavimo sparta. Vertinimas vykdomas tam tikrais griežtai apibrėžtais etapais:

- realiu laiku agreguojamas tinklo srautas taikant judančių vidurkių glotninimo metodą arba perduotų duomenų srauto sumos per laiko intervalą metodą;
- suformuotą eilutę išsaugoma laikinojoje atmintyje, kuri vėliau rekurentiškai papildoma naujais nariais, nekeičiant vartotojo nurodyto eilutės ilgio;
- rekurentiškai skaičiuojamas dinamiškai formuojamos eilutės savastingumo įverčiai taikant *robastinį regresijos* metodą, *IR* statistiką ir *R/S* statistiką;
- skaičiavimo rezultatai tolesnei analizei išsaugomi laikinojoje arba/ir pastoviojoje atmintyje;

43 paveiksle pavaizduota struktūrinė metodo schema, kuri naudojama kompiuterių tinklo srauto analizei realiu laiku.



**43 pav.** Kompiuterių tinklo srauto analizės realiu laiku būdas

Tinklo srauto analizė pradedama nuo analizės kriterijų ir parinkčių įvedimo (1 modulis). Čia nurodomas tinklo srauto agregavimo metodas, formuojamos eilutės ilgis, pasirenkamas laiko eilutės agregavimo intervalas, pasirenkami tinklo srauto savastingumo įverčiai bei nurodoma kur ir kaip saugoti gautus rezultatus. Po to valdymas perduodamas paketų nuskaitymo moduliui (2 modulis) ir tinklo paketų antraščių apdorojimo moduliui (3 modulis). Iš paketo antraštės nuskaityta informacija reikalinga agreguotos eilutės formavimui pagal įvedimo modulyje (1) pasirinktą judančių vidurkių glotninimo arba perduotų duomenų srauto sumos per laiko intervalą agregavimo metodą (plačiau 4.1.1 skyriuje). Laiko eilutės nariai formuojami ir į eilutę rašomi pagal vartotojo nurodytą agregavimo intervalą. 4 modulyje laiko eilutė formuojama rekurentiškai, taikant perindeksavimo algoritmą (plačiau 4.1.2 skyriuje). Formuojama ir vėliau rekurentiškai atnaujinama eilutė saugoma laikinojoje

saugykloje (5 modulis). Suformuota eilutė analizuojama laiko eilučių 6 savastingumo analizės modulyje, kur pagal pasirinktus analizės metodus *robastinį regresijos* metodą skaičiuojamas stabilumo parametras ir/arba skaičiuojama *IR* statistika ir/arba skaičiuojama *R/S* statistika. Skaičiavimo rezultatų formavimo modulis (7) iš savastingumo analizės modulio gauna analitinius skaičiavimo rezultatus. Skaičiavimo rezultatai saugomi laikinoje ir/arba pastovioje saugyklose (8). Skaičiavimo rezultatų formavimo modulis (7) papildomai sujungtas su skaičiavimo rezultatų peržiūros sąsaja (9), kur skaičiavimo rezultatai gali būti vaizdžiai perteikiami vartotojui.

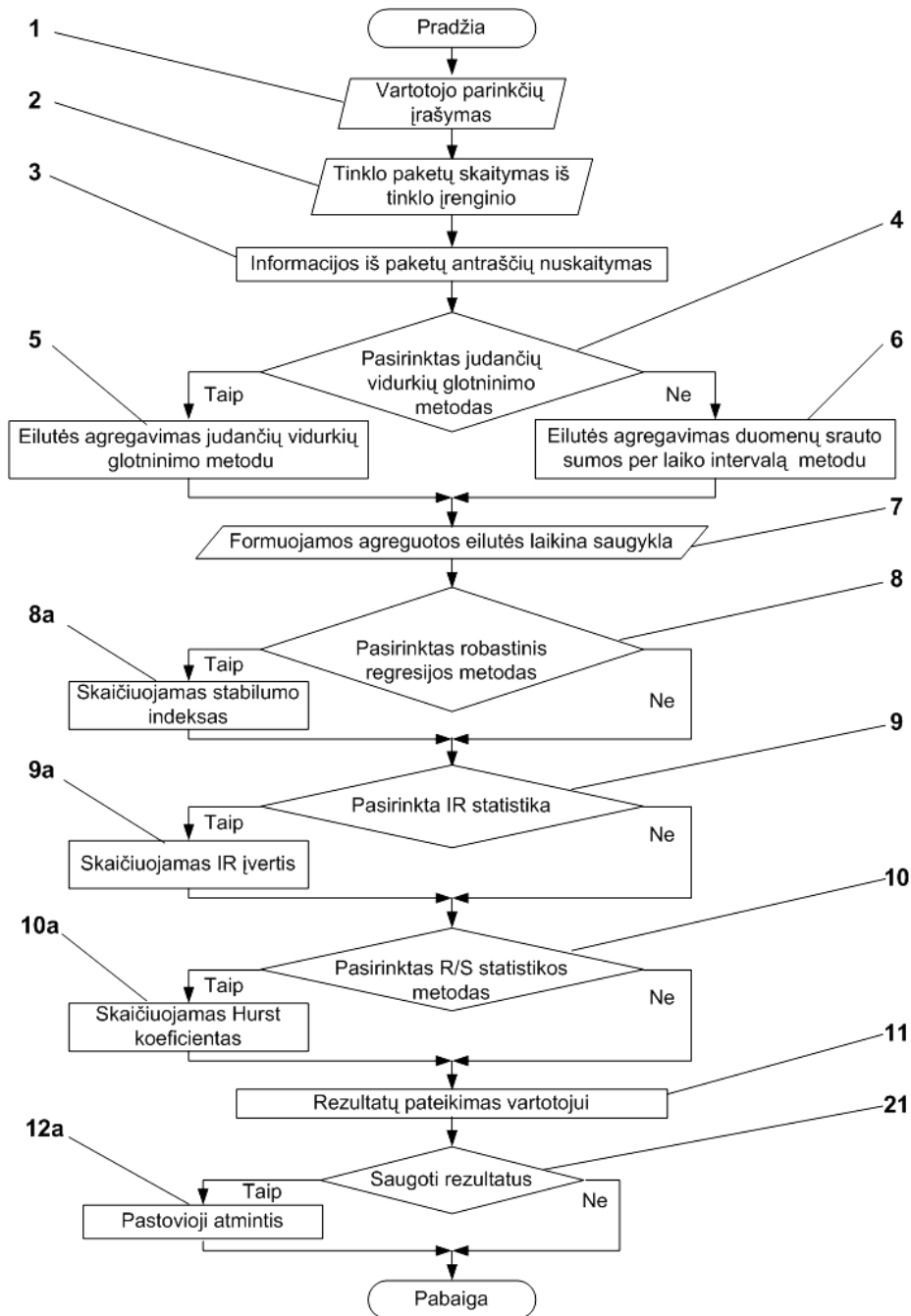
#### **6.4. Matematinis siūlomo metodo aprašymas.**

Analizuojant realiu laiku paketinio kompiuterių tinklo srautą formuojama agreguota laiko eilutė, kuri atitinka matavimus atliktus laiko tarpais  $t_1, t_2, \dots, t_N$ , kur  $i \in [1, N]$ . Agregavimo intervalas  $\Delta t = t_{i+1} - t_i$ , matuojamas milisekundėmis. Iš agregavimo intervalo  $\Delta t \in [10, 100000]$  siūloma pasirinkti laiko eilutės agregavimo intervalą. Pasirenkant  $\Delta t$  rekomenduojama atsižvelgti į matuojamo tinklo srauto intensyvumą, t. y. esant intensyvesniam srautui  $\Delta t$  pasirenkamas mažesnis, priešingu atveju – didesnis. Parenkant agregavimo intervalą galima vadovautis (Park, Rilett, Gajewski, Spiegelman, Choi 2009) ištirta ir aprašyta optimalaus agregavimo intervalo parinkimo metodika pagal keturis srauto vertinimo scenarijus.

Tinklo srauto agregavimo metodus, kai yra realaus laiko duomenų paketų dedamoji, taikant *end-to-end* modelį analizavo T. Ferrari (Ferrari 2000). Disertacijoje aptariamas tinklo modelis be prioritetinių duomenų paketų, t. y. tinklo sraute nėra realaus laiko duomenų paketų dedamosios. Tokio tinklo srauto agregavimui realiu laiku naudojami du skirtingi agregavimo metodai: judančių vidurkių glotninimo metodas ir perduotų duomenų srauto sumos per laiko intervalą metodas. Plačiau šie metodai aptarti 4.1.1 skyriuje.

*Robastinis regresijos* metodas pagal stabilumo parametą  $\alpha$  tiksliau įvertina kompiuterių tinklo srauto savastingumą nei klasikinė *R/S* statistika

apskaičiuodama *Hurst*'o koeficientą  $H$  (plačiau 4.3 skyriuje). Apskaičiuotas  $IR$  įvertis parodo ar kompiuterių tinklo srautas pasižymi savastingumu su  $LRD$ , t. y. leidžia įvertinti kompiuterių tinklo srauto kitimo tendencijas (Kaklauskas, Sakalauskas 2009). Įvardinti savastingumo vertinimo metodai plačiau aprašyti 3.3.3 skyriuje.



**44 pav.** Kompiuterių tinklo srauto analizės realiu laiku būdo realizavimo algoritmas

**Algoritmas.** Kompiuterių tinklo srauto savastingumo vertinimas realiu laiku (žr. 44 pav.).

*Tikslas:* rekurentiškai realiu laiku formuoti agreguotą eilutę, apskaičiuoti savastingumo parametrus ir rezultatą išsaugoti nurodytoje laikmenoje.

*Įėjimo parametrai:*  $t_i$  – paketo atėjimo laikas  $i$ -tuoju momentu,  $x_i$  –  $i$ -oji paraiška,  $N$  – formuojamos eilutės narių skaičius,  $e_i$  – eilutės nario indeksas, kuris dabar bus keičiamas naujai atėjusiu paketu, *seka1* – požymis: 0 – reiškia, kad eilutė formuojama, 1 – pradedamas rekurentinis suformuotos eilutės atnaujinimas.

*Išėjimo parametrai:* kompiuterių tinklo srauto įverčiai.

1 žingsnis – vartotojo parinkčių įrašymas (žr. 44 pav. – 1). Čia įvedama informaciją apie pasirinktą agregavimo intervalą, agregavimo metodą, formuojamos laiko eilutės ilgį, pasirenkami savastingumo įvertinimo metodai bei rezultatų saugojimas.

2 žingsnis – tinklo paketų skaitymas iš tinklo įrenginio (žr. 44 pav. – 2).

3 žingsnis – informacijos iš paketų antraščių nuskaitymas (žr. 44 pav. – 3).

4 žingsnis – laiko eilutės agregavimas (žr. 44 pav. – 4). Jei pasirinktas judančių vidurkių glotninimo metodas (žr. 44 pav. – 5), tai eilutė formuojama taikant šį agregavimo metodą, kitaip eilutė formuojama taikant duomenų srauto sumos per laiko intervalą metodą (žr. 44 pav. – 6).

5 žingsnis – agreguotos laiko eilutės saugojimas (žr. 44 pav. – 7) laikinoje atmintyje kol bus suformuota agreguota laiko eilutė pagal vartotojo pasirinktą laiko eilutės ilgį. Kiekviena sekanti eilutė bus formuojama rekurentiškai, t. y. naikinamas pirmasis suformuotos eilutės narys ir pridedamas naujas agreguotas narys. Siekiant sumažinti eilutės formavimo laiko sąnaudas, rekurentiškai formuojamos eilutės nariai bus įrašomi taikant agreguotos eilutės formavimo, naudojant perindeksavimą, algoritmą (plačiau 4.1.2 skyriuje).

Tikrinama ar  $i < N$ , tai  $i = i + 1$  ir pradedamas kartojimas nuo 2 žingsnio, priešingu atveju  $seka1 = 1$ ,  $e_i = i$  ir valdymas perduodamas į 6 žingsnį.



6 žingsnis – stabilumo parametru skaičiavimas (laiko eilutės narių perrinkimui ir savastingumo parametru skaičiavimui naudojami du vienas su kitu susiję ciklai: 1 – ciklas nuo  $e_i$  iki  $n$ , 2 – ciklas nuo 1 iki  $e_i-1$ ):

6.1 žingsnis – jei pasirinktas *robastinis regresijos* metodas (žr. 44 pav. – 8), tai skaičiuojamas stabilumo indeksas (žr. 44 pav. – 8a), priešingu atveju pereinama į 6.2 žingsnį,

6.2 žingsnis – jei pasirinktas *IR* statistikos metodas (žr. 44 pav. – 9), tai skaičiuojamas *IR* įvertis (žr. 44 pav. – 9a), priešingu atveju pereinama į 6.3 žingsnį,

6.3 žingsnis – jei pasirinktas *R/S* statistikos metodas (žr. 44 pav. – 10), tai skaičiuojamas *Hurst'o* koeficientas (žr. 44 pav. – 10a), priešingu atveju pereinama į 7 žingsnį.

7 žingsnis – rezultatų pateikimas vartotojui (žr. 44 pav. – 11).

8 žingsnis – jei pasirinkta – saugoti rezultatus (žr. 44 pav. – 12), tai gauti įverčiai saugomi pastoviojoje atmintyje (žr. 44 pav. – 12a), priešingu atveju algoritmas baigiamas.

### **6.5. Pateikto išradimo apibrėžtis**

Šiame skyrelyje pateikiama išradimo, registruoto Lietuvos patentų biure apibrėžimas. Čia taip pat įvertinama kuo pasiūlytas metodas skiriasi nuo anksčiau užpatentuočių analogiškų metodų.

1. Kompiuterių tinklo srauto analizės realiu laiku būdas, kai rekurentiškai formuoja eilutę, skaičiuoja statistinius jos parametrus, ir rezultatus saugo laikinojoje atmintyje, *besiskiriantis* tuo, kad:

- realiu laiku agreguoja tinklo srautą taikant judančių vidurkių glotninimo metodą,
- arba perduotų duomenų srauto sumos per laiko intervalą metodą;
- suformuotą eilutę saugo laikinoje atmintyje;
- rekurentiškai skaičiuoja dinamiškai formuojamos eilutės savastingumo parametrus;

- skaičiavimo rezultatus tolesnei analizei saugo laikinojoje arba/ir pastoviojoje atmintyje;

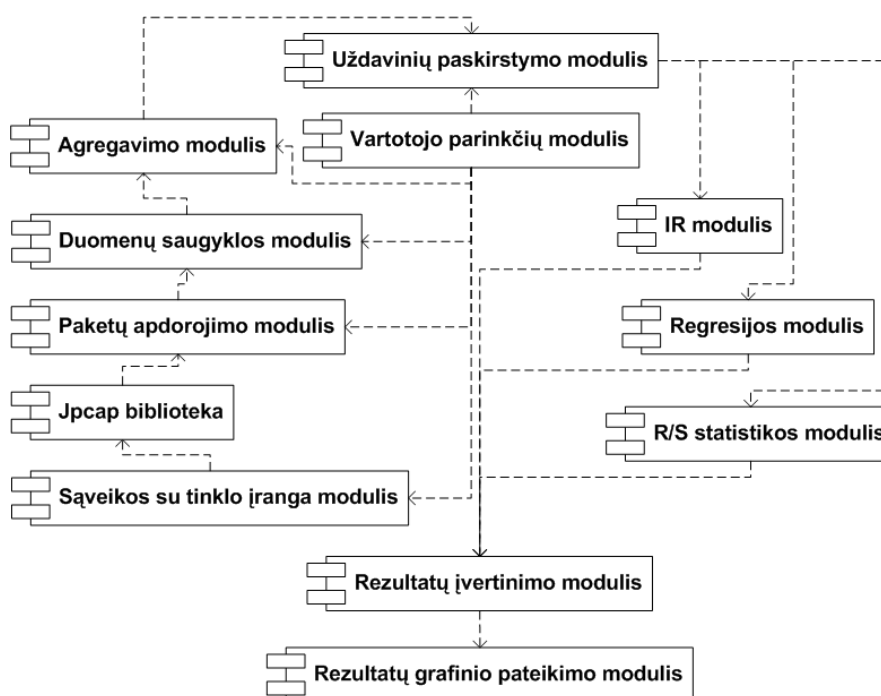
2. Būdas, pagal 1 punktą, *besiskiriantis* tuo, kad savastingumo parametrus skaičiuoja taikant *robastinį regresijos* metodą, *IR* statistiką ir *R/S* statistiką

3. Būdas, pagal 2 punktą, *besiskiriantis* tuo, kad savastingumo įvertinimui skaičiuoja  $\alpha$  stabilumo indeksą, *IR* įvertį ir *Hurst* koeficientą (Kaklauskas, Sakalauskas 2011c).

### 6.6. Tinklo srauto analizės realiu laiku metodo testavimas

Siūlomame analizės realiu laiku metode naudojamos savastingumo vertinimo statistikos yra patikrintos panaudojant kompiuterinio imitavimo metodą bei atliekant skaičiavimus su realaus tinklo duomenų paketais. Plačiau imitavimo rezultatai aprašyti 4.3 ir 4.4 skyriuose.

Tinklo srauto analizės realiu laiku metodo testavimui sukurta programinių modulių biblioteka *SSE-OL* savastingumą vertinanti realiu laiku (žr. 45 pav.).



45 pav. Programinių modulių bibliotekos *SSE-OL* komponentų schema

Fraktališkumo tyrimui realiu laiku Java kalba sudaryta *SSE-OL* (angl. *Self-Similarity Estimator On-Line*) programinių modulių biblioteka. Biblioteka realizuotos šios funkcijos:

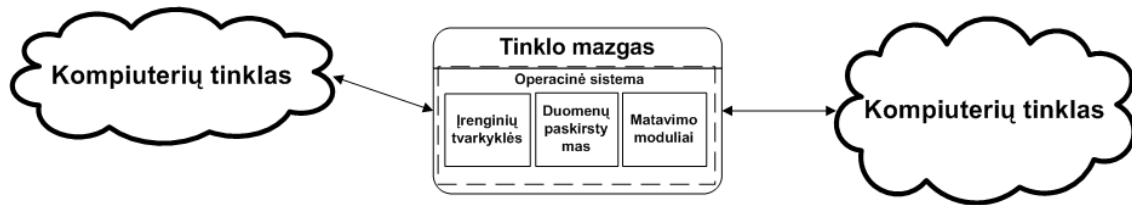
- tinklo paketų antraščių kopijų įrašymas (žr. 45 pav. – Sąveikos su tinklo įranga modulis),
- reikiamos informacijos iš antraštės kopijos išskyrimas (žr. 45 pav. – *Jpcap* biblioteka ir Paketų apdorojimo modulis),
- paketų informacijos saugojimas laikinojoje saugykloje, taikomi metodai ir algoritmai plačiau aprašyti 4.1.2 ir 6.3 skyriuose (žr. 45 pav. – Duomenų saugyklos modulis),
- suformuotos laiko eilutės agregavimas (žr. 45 pav. – Agregavimo modulis),
- laiko eilučių fraktališkumo įvertinimas, taikant metodus plačiau aprašytus 3.3.1, 3.3.2 ir 3.3.3 skyriuose (žr. 45 pav. – IR modulis, Regresijos modulis ir R/S statistikos modulis),
- rezultatų įvertinimas (žr. 45 pav. – Rezultatų įvertinimo modulis),
- grafinis gautų rezultatų pateikimas (žr. 45 pav. – Rezultatų grafinio pateikimo modulis).

Kaip žinome *Java* kalba parašytos programos prieigos prie fizinių tinklo įrenginių neturi, todėl prieigos užtikrinimui naudota atviro kodo biblioteka *Jpcap* (*Jpcap* 2012).

Tinklo apkrovos fraktališkumo tyrimas skaidomas į keturis etapus:

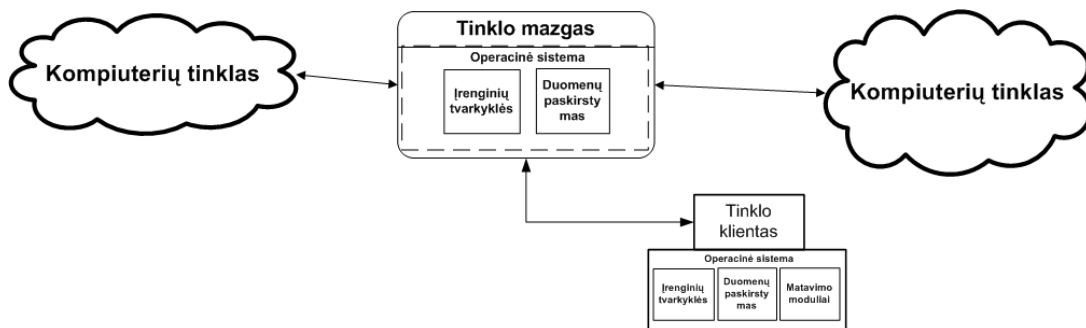
- *Pirmas etapas* – parenkamas agregavimo intervalas, atsižvelgiant į matuojamo tinklo srauto intensyvumą, nurodomas fizinis tinklo įrenginys iš kurio bus fiksuojami paketai, nurodomas duomenų saugyklos tipas, tinklo apkrovos fraktališkumo tyrimo realiu laiku metodai bei analizės rezultatų perteikimo vartotojui būdai;
- *Antras etapas* – formuojamos ir agreguojamos laiko eilutės;
- *Trečias etapas* – analizuojamos agreguotos laiko eilutės ir gauti įverčiai saugomi laikinose saugyklose;

- *Ketvirtas etapas* – rezultatai pateikiami vartotojui pagal pirmame etape nurodytas parinktis.



**46 pav.** *SSE-OL* naudojimo tinklo mazge schema

Programos darbui ir rezultatų saugojimui reikia ne mažiau nei 500MB pastovios atminties, 60MB darbinės atminties bei pakankamos spartos procesoriaus. *SSE-OL* galima naudoti paketinio kompiuterių tinklo apkrovos savastingumo matavimui kliento kompiuteryje arba tinklo mazge, kuris turi tinklinę operacinę sistemą (pageidautina su grafine sąsaja). 46 paveiksle programa *SSE-OL* integruota į tinklo mazgo operacinę sistemą. Šiuo atveju tinklo mazge atsiras papildoma apkrova, kuri mažins praeinančio srauto aptarnavimo galimybes.



**47 pav.** *SSE-OL* naudojimo, kai matavimo kompiuteris prijungtas prie tinklo mazgo schema

47 paveiksle programa *SSE-OL* integruota į tinklo kliento kompiuterį, kuris prijungtas prie tinklo mazgo. Šis jungimo variantas pagrįstas suprojektuotu tinklo mazgo valdymo modeliu (plačiau žiūrėkite 6.1 skyrių). Iš tinklo mazgo persiunčiama paketinio duomenų srauto paketų antraščių kopijos, kurios yra naudojamos srauto matavimui. Šiuo atveju tinklo mazgo papildoma apkrova mažesnė nei integruojant biblioteką į mazgą. Be to galima panaudoti tinklo

mazge integruotas paketų antraščių kopijavimo programos, protokolus, funkcijas.

Galima kliento kompiuterį prijungti ir naudojant tinklų tilto formavimo principą, tačiau šiuo atveju magistralinių tinklo segmentų srautą pamatuoti bus sudėtinga, nes tam reikės papildomų technologinių sprendimų.

Naudojant *SSE-OL* ir 46 paveiksle pavaizduotą jungimo schemą atlikti Šiaulių universiteto Nuotolinių studijų centro tinklo mazgo apkrovos matavimai realiu laiku. Matavimams naudotas vidutinio galingumo (procesoriaus indeksas 38677) tinklo mazgo serveris (Tom's Hardware 2012) su Windows 2003 Server Standard Edition operacine sistema. Tyrimo rezultatai parodė, kad procesoriaus apkrova kinta nuo 1% iki 5%, o darbinės atminties naudojama nuo 44MB iki 50MB. Savastingumo įverčių skaičiavimas pagal visus tris metodus vienai eilutei vidutiniškai trunka 14,1329ms, o eilutės formavimo laikas priklauso nuo pasirinkto eilutės agregavimo intervalo  $\Delta t$ . Apskaičiuota 300 laiko eilučių  $\alpha$  stabilumo parametras ir *Hurst'o* koeficientas. Tyrimo rezultatai parodė, kad 86,17% eilučių pasižymi *LRD*. *IR* įvertis 87,24% atvejų rodė, kad matuojamos laiko eilutės pasižymi *LRD*.

### **6.7. Skyriaus išvados**

Sukurtas ir Lietuvos patentų biure užpatentuotas paketinio kompiuterių tinklo srauto savastingumo analizės metodas realiu laiku. Įvertintos ir parinktos tinkamiausios savastingumo skaičiavimo realiu laiku metodikos: *robastinis regresijos* metodas ir *IR* statistika. Sukurtas rekurentinis eilučių formavimo realiu laiku algoritmas, naudojantis eilutės perindeksavimo metodą. Sukurtas metodas testuotas kompiuterinio imitavimo būdu. Panaudojant paketinio kompiuterių tinklo srauto savastingumo analizės realiu laiku metodą parengtas atviras tinklo mazgo valdymo modelis su grįžtamoju ryšiu.

## 7 Išvados

Klasikiniai Puasono ir Markovo tinklo srauto modeliai išsamiai ištirti ir plačiai taikomi modeliuojant tinklo srautą, kai jo būsenų skaičius ribotas. Tačiau empiriniai tyrimai rodo, kad šiuolaikiniai kompiuterių tinklų srautai pasižymi pliūpsniškumu bei savastingumu ir klasikiniai modeliai jų analizei netinka. Savastingojo paketinio srauto su sunkiomis uodegomis modeliavimui kompiuterių tinkle mokslinėje literatūroje išsamiai išanalizuoti ir dažnai taikomi *ON-OFF* modeliai. Taip pat aprašomi kompleksiniai modeliai, kuriuose panaudojami klasikiniai Markovo, Markovo grandinių bei Puasono, *MMPP*, *MMRP* ir kitų modelių tinkamiausias savybės savastingajam srautui su sunkiomis uodegomis modeliuoti.  $\alpha$ -stabilieji modeliai ir aptarnavimo teorijos įrankiai mokslinėje literatūroje yra analizuojami daugiausia teoriškai. Žinomi šių modelių pritaikymai finansinių procesų modeliavimui, tuo tarpu praktinių jų pritaikymų savastingajam tinklo srautui su sunkiomis uodegomis yra mažai. Autoregresiniai *AR* tinklo srauto modeliai geriausiai tinka garso perdavimo, vaizdo perdavimo ir kitų realaus laiko tinklo srautų modeliavimui, modeliuojas jais standartinį tinklo srautą gauti rezultatai gali būti neadekvatūs.

*Hurst*'o statistikų apskaičiavimui galima panaudoti dvi vertinimo metodikas: laiko eilučių analizės ir eilutės dažniųjų/banginių savybių įverčius. Laiko eilutėms su sunkiomis uodegomis ir pasižyminčiomis *LRD*, įverčiams apskaičiuoti naudojamos  $\alpha$ -stabiliojo proceso savybės, kai vertinami keturi parametrai:  $\alpha$  – stabilumo parametras,  $\beta$  – asimetrijos parametras,  $\sigma$  – mastelio parametras,  $\mu$  – padėties parametras. Jiems apskaičiuoti sukurta metodika, kurioje realizuoti šie metodai: *Fama-Roll*, *McCulloch*, *regresijos* ir *momentų*. Sukurta *IR* statistikos praktinio naudojimo metodika, patikimai ir greitai vertinanti tinklo srauto savastingumą.

Simuliuotų laiko eilučių tyrimai su *Fractan*, *Selfis* ir *SSE* parodė, kad asimetrijos koeficiento  $\beta$  pokytis laiko eilutėms pasižyminčioms savastingumu turi minimalią įtaką *Hurst*'o koeficiento bei stabilumo parametro  $\alpha$  reikšmei.

Imitavimo rezultatų, gautų su *Fractan*, *Selfis* ir *SSE*, lyginamoji analizė parodė, kad sintetinio srauto savastingumo įverčiai tiksliausiai apskaičiuojami taikant regresijos metodą. Realaus tinklo srauto agreguotų eilučių tyrimo rezultatai rodo, kad procesas aprašomas laiko eilutėmis yra persistentinis procesas su *LRD*, kai *Hurst*'o koeficiento reikšmė kinta nuo 0,61 iki 0,79, be to analizuotas kompiuterių tinklo srautas pasižymi fraktališkumu.

Panaudojant sukurtą tinklo srauto modeliavimo paketą *MulNodSimSys* ištirta įėjimo bei aptarnavimo srautų savastingumo įtaka tinklo mazgo darbui, atsižvelgiant į aptarnavimo eilės discipliną ir buferio talpą, apkrovos ir aptarnavimo intensyvumus bei savastingumą. Nustatyta, kad, didėjant tinklo srauto intensyvumui, savastingumu pasižymintis srautas yra aptarnaujamas stabiliai. Aptarnaujant *Puasono* srautą didėja paketų praradimo tikimybė, didėjant įeinančio srauto intensyvumui. Eilės aptarnavimo disciplina *LIFO PS* tipo sraute užtikrina geresnę aptarnavimo kokybę. Didėjant tinklo mazgo buferio talpai, srauto aptarnavimo tikimybė didėja, jei jis pasižymi savastingumu. Buferio eilės aptarnavimo disciplina *LIFO* geriau aptarnauja *PS* tipo srautą. Didėjant savastingosios eilutės stabilumo parametru, aptarnavimo tikimybė didėja. Stipresnė priklausomybė stebima, kai srautas yra *SS* tipo. Remiantis tiesinės regresijos rezultatais ir determinacijos koeficientu  $r^2$  tarp  $N_{buf}$  ir  $\bar{T}_{buf}, \bar{T}_{delay}, \bar{N}_{aplsyst}, T_{buffull}$  nustatyta labai stipri, o tarp  $N_{buf}$  ir  $P_{loss}, P_{serv}$  vidutiniška priklausomybė, o  $F$  ir  $P_F$  rodo, kad stebimas ryšys tarp priklausomų ir nepriklausomų kintamųjų nėra atsitiktinis. Nustatyta, kad  $\bar{T}_{delay}$  *PS* tipo srautui yra didesnis, kai eilės aptarnavimo disciplina *FIFO*. Nustatyta, kad stebimas ryšys tarp  $\lambda$  ir  $P_{loss}, P_{serv}, \bar{T}_{buf}, \bar{T}_{delay}, \bar{N}_{aplsyst}, T_{buffull}$  nėra atsitiktinis. Gautų reikšmių analizės rezultatai parodė, kad įeinančio srauto intensyvumas daugiausia įtakoja  $P_{loss}, \bar{T}_{buf}$  ir  $\bar{T}_{delay}$ , o eilės aptarnavimas yra geresnis, kai aptarnavimo disciplina *LIFO* (kai  $\rho \geq 10$ ) *PP* ir *PS* tipo srautams. *SP* ir *SS* tipo srautams buferio eilės aptarnavimo disciplina jokios įtakos paraiškų aptarnavimo parametrų neturi. Nustatyta, kad kai įeinantis srautas

yra savastingasis, tai  $\bar{T}_{buf}$  ir  $\bar{T}_{delay}$ , nepriklausomai nuo eilės aptarnavimo disciplinos yra mažesnis, kai visi kiti generuojamo srauto parametrai yra vienodi. Nustatyta, kad  $\alpha$  ir  $P_{loss}, P_{serv}, \bar{T}_{buf}, \bar{T}_{delay}, \bar{N}_{aplsyst}, T_{buffull}$  yra tarpusavyje priklausomi ir šis ryšys nėra atsitiktinis. Išanalizavus skaičiavimo rezultatus gauta, kad didėjant savastingumo koeficientui nežymiai didėja  $\bar{N}_{aplsyst}$  skaičius sistemoje. Kitų kintamųjų ir  $\alpha$  priklausomybės visiškai neįtakoja eilės aptarnavimo disciplina. Įvertinus kanaluose aptarnautų paraiškų  $\overline{N(n_{ch})}$  ir  $\overline{N(n_{ch})_{std}}$  gauta, kad didėjant įeinančio srauto intensyvumui kanalai apkraunami tolygiau. Pažymėtina, kad esant tam pačiam įeinančio srauto intensyvumui savastingojo tinklo mazgo srauto aptarnavimo našumas yra daugiau nei dvigubai didesnis nei tinklo mazgo, kurio srautas yra *Puasono*.

Sukurtas ir Lietuvos patentų biure užpatentuotas kompiuterių tinklo paketinio srauto savastingumo analizės metodas realiu laiku. Įvertintos ir parinktos tinkamiausios savastingumo skaičiavimo realiu laiku metodikos: *robastinis regresijos* metodas ir *IR* statistika. Sukurtas rekurentinis eilučių formavimo realiu laiku algoritmas, naudojantis eilutės perindeksavimo metodą. Panaudojant paketinio kompiuterių tinklo srauto savastingumo analizės realiu laiku metodą parengtas atviras tinklo mazgo valdymo modelis su grįžtamuju ryšiu.

## Literatūra

1. Abdelnaser A. (1997). Traffic models in broadband networks. Communications Magazine, IEEE, vol. 35, No. 7, pp. 82-89.
2. Abdelnaser A., Amarnath M. (2000) Quality of service provisioning for long-range-dependent real-time traffic. Self-similar traffic and performance evaluation, edited by Park K. and Willinger W., John Wiley and Sons, pp. 481-505.
3. Abry P., Baraniuk R., Flandrin P., Riedi R., Veitch D. (2002). Multiscale nature of network traffic. IEEE Signal processing, vol. 19, no. 3, pp. 28-46.



4. Abry P., Flandrin P., Taqqu M. S., Veitch D. (2000). Wavelets for the Analysis, Estimation, and Synthesis of Scaling Data. Self-Similar Network Traffic and Performance Evaluation. Edited by Park K., Willinger W., John Wiley & Sons, pp. 39-89.
5. Abry P., Veitch D. (1998) Wavelet analysis of long-range dependence traffic. *IEEE Transactions on Information Theory*; 4(1), pp. 2–15.
6. Adler M., Sitaraman R. K., Venkataramani H. (2011). Algorithms for optimizing the bandwidth cost of content delivery. *Computer Networks*, vol. 55, no. 18, pp. 4007–4020.
7. Alheraish A. (2004). Autoregressive video conference models. *International Journal of Network Management*, Vol. 14, No. 5, pp. 329-337.
8. Appenzeller G., Keslassy I., McKeown N. (2004) Sizing router buffers. In *SIGCOMM '04*, New York, NY, USA. ACM Press, pp. 281–292.
9. Asars A., Petersons E. (2009). Determining the optimal interval of the parameter identification of self-similar traffic. *Automatic Control and Computer Sciences*, vol. 43, No. 4, pp. 211-216.
10. Aström K. J., Murray R. M. (2008). *Feedback Systems: An Introduction for Scientists and Engineers*. Princeton University Press, pp. 424.
11. Baccarelli E., Cordeschi N., Patriarca T. (2012). *QoS Stochastic Traffic Engineering for the wireless support of real-time streaming applications*. *Computer Networks*, vol. 56, pp. 287–302.
12. Bandt C., Graf S. (1992) A Characterization of Self-similar Fractals with Positive Hausdorff Measure. *Proceedings of the American Mathematical Society*. Vol.114, No 4, pp. 995-1001.
13. Belov I., Kabašinskas A., Sakalauskas L. (2006) A Study of Stable Models of Stock Markets. *Information Technology and Control*. Vol. 35, No. 1, pp. 34-56.
14. Beran J. (1998). *Statistics for Long-Memory Processes*. USA: Capman&Hall, pp. 317.

15. Boudec J.-Y. L., Thiran P. (2001) Network Calculus: A Theory of Deterministic Queuing Systems for the Internet. Springer-Verlag, <[http://ica1www.epfl.ch/PS\\_files/NetCal.htm](http://ica1www.epfl.ch/PS_files/NetCal.htm)>.
16. Boxma O. J., Cohen J. W. (1999). Heavy-traffic analysis for the GI/G/1 queue with heavy –tailed distributions. Queueing systems, vol. 33, pp. 177-204.
17. Cai L., Li H., Chen B., Wang J. (2009). On the Chaotic Dynamics Analysis of Internet Traffic. International Workshop on Chaos-Fractals Theories and Applications, Shenyang, China, pp. 361-364.
18. Cappé O., Moulines E., Pesquet J-C., Petropulu A., Yang X. (2002). Long-range dependence and heavy-tail modelling for teletraffic data. IEEE Signal processing, vol. 19, no. 3, pp. 14-27.
19. Carvalho L., Angeja L., Navarro A. (2005). A New Packet Loss Model of the IEEE 802.11g Wireless Network for Multimedia Communications. IEEE Transactions on Consumer Electronics, vol. 51, no. 3, pp. 809-814.
20. Chakraborty D., Ashir A., Suganuma T., Mansfield Keeni G., Roy T. K., Shiratori N. (2004). Self-similar and fractal nature of Internet traffic. International Journal of Network Management, vol. 14, no. 2, pp. 119–129.
21. Chandramathi S., Shanmugavel S. (2003). Estimation of cell loss probability for self-similar traffic in ATM networks – a fuzzy approach. Applied Software Computing, vol. , No. 2, pp. 71-84.
22. Chang C. S. (1994) Stability, queue length and delay, Part I: Deterministic queuing networks. IEEE/ACM Trans. Auto. Control., vol. 39, no. 5, pp. 913-931.
23. Chao K., Aziz M., Wei H., Jimin S. (2007). Network monitoring system responsive to changes in packet arrival variance and mean. United States Patent 7242668. [interaktyvus]. [Žiūrėta 2012-02-15]. Prieiga per internetą: <http://www.patents.com/us-7242668.html>.

24. Chen H.; Cai H.; Li Y. (1997). The multifractal property of bursty traffic and its parameter estimation based on wavelets. TENCON '97. IEEE Region 10 Annual Conference. Speech and Image Technologies for Computing and Telecommunications., Proceedings of IEEE, vol. 2, pp. 791-794.
25. Chung J.-M., Seol J.-H., Choi S., Yeoum T., Lim H. (2008). Statistical delay control scheme for DiffServ networks with self-similar traffic. Electronics letters, Vol. 44, No. 9.
26. Cisco (2012). Cisco Packet Tracer. [interaktyvus]. [Žiūrėta 2012-02-15]. Prieiga per internetą: <[http://www.cisco.com/web/learning/netacad/course\\_catalog/PacketTracer.html](http://www.cisco.com/web/learning/netacad/course_catalog/PacketTracer.html)>.
27. Crovella M. E., Bestavros A. (1996) Self-similarity in World Wide Web Traffic Evidence and Possible Causes. Proceedings of the 1996 ACM SIGMETRICS international conference on Measurement and modelling of computer systems ACM New York, NY, USA, Vol. 24, No. 1, pp. 160-169.
28. Cruz R. L. (1991) A calculus for network delay, Part I: network elements in isolation. IEEE Trans. Information Theory. Vol. 37, No. 1, pp. 114-131.
29. Daigle J. N., Langford J. D. (1986). Models for analysis of packet voice communication systems. IEEE Journal on Selected Areas in Communications, Vol. 4, No. 6, pp. 847-855.
30. DePertillo P. B. (2000). Determining the hurst exponent for time series data. United States Patent 6144877. [interaktyvus]. [Žiūrėta 2012-02-15]. Prieiga per internetą: <http://patft.uspto.gov/>.
31. Dijkstra E. W. (1959). A Note on Two Problems in Connexion with Graphs. Numerische Mathematik, vol. 1, pp. 269-271.
32. Dombry C., Kaj I. (2011). The *ON-OFF* network traffic model under intermediate scaling. Queuing Systems: Theory and Applications, vol. 69, No. 1, pp. 29-44.

33. Dufour J.-M., Kurz-Kim J.-R. (2010). Exact inference and optimal invariant estimation for the stability parameter of symmetric  $\alpha$ -stable distributions. *Journal of Empirical Finance*, vol. 17, pp. 180-194.
34. Dugatkin D, Hannel C. (2004a). Automated capturing and characterization of network traffic using feedback. United States Patent 7627669. [interaktyvus]. [Žiūrėta 2012-02-15]. Prieiga per internetą: <http://www.patentstorm.us/patents/7627669.html>.
35. Dugatkin D, Hannel C. (2004b). Automated characterization of network traffic. United States Patent 7840664. [interaktyvus]. [Žiūrėta 2012-02-15]. Prieiga per internetą: <http://www.patentstorm.us/patents/7840664.html>.
36. Dugatkin D. (2007). Self-similar traffic generation. United States Patent 7257082. [interaktyvus]. [Žiūrėta 2012-02-15]. Prieiga per internetą: <http://www.wikipatents.com/US-Patent-7257082/self-similar-traffic-generation>.
37. Elagha H., AlShafee M. (2007). On the Self-Similar Nature of ATM Network Traffic. *Informing Science and Information Technology*, vol. 4, pp. 641-649.
38. Elbiaze H., Zhani M. F., Cherkaoui O., Kamoun F. (2010). A new structure-preserving method of sampling for predicting self-similar traffic. *Telecommunication Systems*, vol. 43, No. 3-4, pp. 265-277.
39. Enachescu M., Ganjali Y., Goel A., McKeown N., Roughgarden T. (2005) Part III: Routers with very small buffers. *ACM/SIGCOMM Computer Communication Review*. Vol. 35, No. 3, pp. 83-90.
40. Enciklopedinis kompiuterijos žodynas (2012). [interaktyvus]. [Žiūrėta 2012-02-15]. Prieiga per internetą: <http://www.likit.lt/term/enciklo.html>.
41. Engset T. O. (1998). On the Calculation of Switches in an Automatic Telephone System. Translated by Arne Myskja, *Teletronikk*, vol. 94, no. 2, pp. 99-142.
42. Erlong A. K. (1909). The theory of probabilities and telephone conversations. *Nyt Tidsskrift Mat*. Vol. 20, pp. 33-39.

43. Erramilli A., Narayan O., Willinger W. (1996). Experimental queuing analysis with long-range dependent packet traffic. *Networking, IEEE/ACM Transactions on*, Vol. 4, pp. 209-223.
44. Fama E. F., Roll R. (1971) Parameter Estimates for Symmetric Stable Distributions. *Journal of the American Statistical Association*. Vol. 66, No. 334, pp. 331-338.
45. Feder J. (1988) *Fractals*. New York: Plenum Press, pp. 285.
46. Ferrari T. (2000). End-to-end performance analysis with traffic aggregation. *Computer Networks*, vol. 34, no. 6, pp. 905-914.
47. Figueiredo D. R., Liu B., Misra V., Towsley D. (2002). On the autocorrelation structure of *TCP* traffic. *Computer Networks*, vol. 40, no. 3, pp. 339-361.
48. Fu H. (2006). On parameter estimation of a simple real-time flow aggregation model. *International Journal of Communication Systems*, vol. 19, No. 7, pp. 795–808.
49. Fukuda K., Takayasu M., Takayasu H. (2005). A cause of self-similarity in *TCP* traffic. *International Journal of Communication Systems*, vol. 18, No. 6, pp. 603–617.
50. Gallardo J. R., Makrakis D., Orozco-Barbosa L. (2000). Use of  $\alpha$ -stable self-similar stochastic processes for modelling traffic in broadband networks. *Performance Evaluation*, vol. 40, pp. 71–98.
51. Gallos L. K., Song C., Makse H.A. (2007) A review of fractality and self-similarity in complex networks. *Physica A: Statistical Mechanics and its Applications*. Vol. 386, pp. 686–691.
52. Gao J., Rubin I. (2001a). Multifractal modelling of counting processes of long-range dependent network traffic. *Computer Communications*, vol. 24, no. 14, pp. 1400-1410.
53. Gao J., Rubin I. (2001b). Multiplicative multifractal modelling of long-range-dependent network traffic. *International Journal of Communication Systems*, vol. 14, no. 8, pp. 783-801.

54. Ge X., Zhu G., Zhu Y. (2003). On the testing for alpha-stable distributions of *Ethernet* network traffic. *Journal of Electronics (China)*, vol. 20, No. 4, pp. 309-312.
55. Gelenbe E., Pujjole G. (1999). *Introduction to queuing networks*, second edition. John Willey & Sons, pp. 258.
56. Ginno M., Gastron L. (2009). Presentation of an Estimator for the *Hurst* Parameter for a Self-Similar Process Representing the Traffic in IEEE 802.3 Networks. *International Journal of Computers, Communications & Control*; Vol. 4, No. 2, pp. 137-147.
57. GloMoSim. (2012). Global Mobile Information Systems Simulation Library. [interaktyvus]. [Žiūrėta 2012-02-15]. Prieiga per internetą: <<http://pcl.cs.ucla.edu/projects/glomosim/>>.
58. GNS3. (2012). Graphical Network Simulator. [interaktyvus]. [Žiūrėta 2012-02-15]. Prieiga per internetą: <<http://www.gns3.net/>>.
59. Gospodinov M., Gospodinova E. (2007). Comparative analysis of *Hurst* techniques. Proceedings of the 2007 international conference on Computer systems and technologies, ACM, pp. IIIB. 16-1 IIIB. 16-5.
60. Grassberger P., Procaccia I. (1983). Estimation of the Kolmogorov Entropy from a Chaotic Signal. *Physical review A*, vol. 28, no. 4, pp. 2591-2593.
61. Grigelionis B. (2008) On stochastic processes associated with relativistic stable distributions / *Lithuanian mathematical journal*. Vol. 48, No. 1, pp. 61-69.
62. Grimm C., Schlüchtermann G. (2008). *IP-traffic theory and performance*. Springer, pp. 487.
63. Guan L., Awan I.U., Woodward M.E., Wang X. (2007) Discrete-time performance analysis of a congestion control mechanism based on RED under multi-class bursty and correlated traffic. *Journal of Systems and Software*. Vol. 80, No. 10, pp. 1716-1725.

64. Gyires T. (2003). Software Agents Architecture for Controlling Long-Range Dependent Network Traffic. *Mathematical and Computer Modelling*, vol. 38, pp. 839-848.
65. Hagiwara T., Doi H., Tode H., Ikeda H. (2001) High-speed calculation method of the *Hurst* Parameter based on real traffic. *IEICE Transactions on information and systems*, vol. E84D, No. 5, pp. 578-587.
66. Halima E., Zhani M. F., Cherkaoui O., Kamoun F. (2010). A new structure-preserving method of sampling for predicting self-similar traffic. *Telecommunications Systems*, vol. 43, pp. 265–277.
67. He G., Gao Y., Hou J.C., Park K. (2004) A case for exploiting self-similarity of network traffic in *TCP* congestion control. *Computer Networks: The International Journal of Computer and Telecommunications Networking*. Vol. 45, pp. 743-766.
68. Heffes H., Lucantoni D. (1986). A Markov Modulated Characterization of Packetized Voice and Data Traffic and Related Statistical Multiplexer Performance. *Selected Areas in Communications, IEEE Journal*, Vol. 4, No. 6, pp. 856-868.
69. Heidergott B., Olsder G. J., Woude J. W. (2006) *Max Plus at work: modelling and analysis of synchronized systems: a course on Max-Plus algebra and its applications*. Princeton University Press.
70. Huang P.-K., Chang C.-S., Cheng J., Lee D.-S. (2007). Recursive Constructions of Parallel *FIFO* and *LIFO* Queues With Switched Delay Lines. *Ieee transactions on information theory*, vol. 53, no. 5, pp. 1778-1798.
71. Hurst H. E. (1951) Long-term Storage Capacity of Reservoirs. *Transactions of the American Society of Civil Engineers*. Vol. 116, pp. 770-799.
72. Ilnickij S. (2004). Research of the Network Server in Self-Similar Traffic Environment. *Telecommunications and Electronics*, vol.4, pp. 65-72.

73. Iversen V. B. (2011). *Teletraffic Engineering And Network Planning*. Technical University of Denmark, pp. 567.
74. Jain R. (1991). *The Art Of Computer Systems Performance Analysis*. John Wiley & Sons, pp. 685.
75. Jain R., Routhier S. (1986). Packet Trains--Measurements and a New Model for Computer Network Traffic. *IEEE Journal on Selected Areas in Communications*, Vol. 4, No. 6, pp. 986-995.
76. Janicki A., Weron A. (2000). *Simulation and chaotic behaviour of a-stable stochastic processes*. New York: Marcel Dekker, pp. 360.
77. Jeong H.-D. J., Lee J.-S. R., McNickle D. (2005). Distributed steady-state simulation of telecommunication networks with self-similar teletraffic. *Simulation Modelling Practice and Theory*, vol. 13, no. 3, pp. 233-256.
78. Jeong H.-D. J., Lee J.-S. R., McNickle D., Pawlikowski K. (2007) Suggestions of Efficient Self-Similar Generators. *Modelling, Practice and Theory*, vol. 15, no. 3, pp. 328-353.
79. Jeong H.-D. J., McNickle D., Pawlikowski K. (1999) A Search for Computationally Efficient Generators of Synthetic Self-similar Teletraffic. *Australian Computer Science Communications*, vol. 21, Springer, pp. 75-86.
80. Jiang Y., Yin Q., Liu Y., Jiang S. (2009) Fundamental calculus on generalized stochastically bounded bursty traffic for communication networks. *Computer Networks*. Vol. 53, No. 12, pp. 2011-2021.
81. Jiang, Y., Liu, Y. (2008) *Stochastic Network Calculus*. Springer-Verlag London Limited.
82. Jonga P., Penzer J. (2004). The ARMA model in state space form. *Statistics & Probability Letters*, vol. 70, No. 1, pp. 119-125.
83. JPCAP. (2012). Java library for capturing and sending network packets. [interaktyvus]. [Žiūrėta 2012 m. vasario 15 d.]. Prieiga per internetą: <<http://netresearch.ics.uci.edu/kfujii/jpcap/doc/index.html>>.



84. Kabašinskas A., Rachev S., Sakalauskas L., Sun W., Belov I. (2006) Alpha-stable paradigm in financial markets, *Journal of Computational Analysis and Applications*. Vol. 11, No. 4, pp. 641–668.
85. Kaj I. (2002). *Stochastic Modeling in Broadband Communications Systems*. USA: SIAM, Philadelphia, pp. 188.
86. Kajačkas A., Pavilanskas L., Vindašius A. (2007). Synchronous Voice Applied Customer Access based on IEEE 802.11. *Electronics and Electrical Engineering*, vol. 8, no. 80, pp. 23-28.
87. Kaklauskas L. (2003). *Kompiuterių tinklai*. VšĮ ŠU leidykla, Šiauliai, pp. 236.
88. Kaklauskas L. (2007). *CCNA. Cisco Certificate Network Associate*, Cisco ID No. CSC011291846.
89. Kaklauskas L., Sakalauskas L. (2008). On network traffic statistical analysis. *Liet. mat. rink. LMD darbai*, vol. 48, no. 49, pp. 314–319.
90. Kaklauskas L., Sakalauskas L. (2009). Application of Chaos Theory to Analysis of Computer Network Traffic. *Proceedings of XIII International Conference “Applied Stochastic Models and Data Analysis”*, Vilnius Lithuania, 407-412. (ISI Proceedings).
91. Kaklauskas L., Sakalauskas L. (2010a). Tinklo apkrovos savastingumo tyrimas realiu laiku. „Informacijos mokslai“, 53 tomas, pp. 100-105.
92. Kaklauskas L., Sakalauskas L. (2010b). Savastingumo įtakos tinklo mazgo darbui tyrimas. *Jaunųjų mokslininkų darbai*, Nr. 3, pp. 100-107.
93. Kaklauskas L., Sakalauskas L. (2011a). Study of the Impact of Self-Similarity on the Network Node Traffic. *Electronics and electrical engineering*, vol. 111, no. 5, pp. 27-32.
94. Kaklauskas L., Sakalauskas L. (2011b). Study of on-line measurement of traffic self-similarity. *Central European Journal of Operations Research*, DOI: 10.1007/s10100-011-0216-5. Accepted on 16-06-2011. <<http://www.springerlink.com/content/0036x207656ppq211/>>

95. Kaklauskas L., Sakalauskas L. (2011c). Kompiuterių tinklo srauto analizės realiu laiku būdas. LR valstybinis patentų biuras, patento numeris LT20011099.
96. Kaklauskas L., Vaičiulis M. (2011). Skirstinio uodegos indekso MAX-DPR įvertis. Jaunųjų mokslininkų darbai, nr. 1, pp. 126-130.
97. Karagiannis T. (2002). *Selfis* 0.1b. <<http://www.cs.ucr.edu/~tkarag/Selfis/Selfis.html>>.
98. Karagiannis T., Faloutsos M., Molle M. (2003) A user-friendly self-similarity analysis tool. ACM SIGCOMM Computer Communication Review. Vol. 33, No. 3, pp. 81-93.
99. Karasaridis A., Hatzinakos D. (2001). Network heavy traffic modelling using  $\alpha$ -stable self-similar processes. Communications, IEEE Transactions, vol. 49, no. 7, pp. 1203-1214.
100. Kendall G. (1951). Some problems in the theory of queues. Journal of the Royal Statistical Society. Series B (Methodological), vol. 13, No. 2, pp. 151-185.
101. Kesselman A., Lotkeryx Z., Mansour Y., Patt-Shamiryx B., Schieberz B., Sviridenkoz M. (2001). Buffer Overflow Management in QoS Switches. STOC '01 Proceedings of the thirty-third annual ACM symposium on Theory of computing, pp. 520-529.
102. Khalid M. (1994). Method and system for real-time burstiness analysis of network traffic. United States Patent 5343465. [interaktyvus]. [Žiūrėta 2012-02-15]. Prieiga per internetą: <http://patft.uspto.gov>.
103. Khalid M. K. (1994). Method and system for real-time burstiness analysis of network traffic. United States Patent 5343465. [interaktyvus]. [Žiūrėta 2012-02-15]. Prieiga per internetą: [http://www.patentlens.net/patentlens/patents.html?patnums=US\\_5343465&language=&](http://www.patentlens.net/patentlens/patents.html?patnums=US_5343465&language=&).

104. Kim J.S., Goh K., Salvi I.G., Oh E., Kahng B., Kim D. (2007) Fractality in complex networks: critical and supercritical skeletons. *Physical Review E*. Vol. 75.
105. Kleinrock L. (1975). *Queueing Systems: Volume I – Theory*. Wiley Interscience. New York, pp. 448.
106. Kleinrock L. (2002). Creating a Mathematical Theory of Computer Networks. *Operations Research*. Vol. 50, No. 1, pp. 125 – 131.
107. Klemm A., Lindemann C., Lohmann M. (2003). Modeling IP traffic using the batch Markovian arrival process. *Performance Evaluation*, vol. 54, No. 2, pp. 149-173.
108. Kokoszka P.S., Taqqu M.S. (1996) Parameter estimation for infinite variance fractional ARIMA. *The Annals of Statistics*. Vol. 24, No. 5, pp. 1880-1913.
109. Koutrouvelis I. A. (1981) An iterative procedure for the estimation of the parameters of the stable law. *Communications in Statistics - Simulation and Computation*. Vol. 10, pp. 17–28.
110. Kulikovs M. (2008). Statistical parameters estimation of the self-similar input traffic for the Measurement-based Admission Control. *Telecommunications and Electronics*, pp. 37-42.
111. Kulikovs M., Petersons E. (2008). Remarks Regarding Queuing Model and Packet Loss Probability for the Traffic with Self-Similar Characteristics. *International Journal of Computer Science*, Vol. 30, pp. 84-90.
112. Kulikovs M., Petersons E. (2009). Modelling the On-line Traffic Estimator in OPNET. *Electronics and Electrical Engineering*, vol. 7, no. 95, pp. 82-86.
113. Lavancier F., Philippe A., Surgailis D. (2009) Covariance function of vector self-similar processes / *Statistics & probability letters*. Vol. 79, No. 23, pp. 2415-2421.

114. Lee K. (1995) Performance bounds in communication networks with variable rate links. In Proc. of ACM SIGCOMM, pp. 126-136.
115. Leland E., Taqqu S., Willinger W., Wilson D.W. (1994) On the Self-Similar Nature of *Ethernet* Traffic. IEEE/ACM Transactions on networking. Vol. 2, No. 1, pp. 1–15.
116. Leland W. E., Wilson D. V. (1991) High Time-Resolution Measurement and Analysis of LAN Traffic: Implications for LAN Interconnection. INFOCOM '91. Proceedings. Tenth Annual Joint Conference of the IEEE Computer and Communications Societies. Networking in the 90s., IEEE.
117. Lemoine É., Pham C. and Lefèvre L.(2004). Packet classification in the NIC for improved SMP-based Internet servers. In ICN. IEEE.
118. Li J.-S. (2002). Using autoregressive Gaussian processes with trends and aggregations to model self-similar traffic. Computer Communications, vol. 25, no. 10, pp. 964-971.
119. Li M., Lim S.C. (2008) Modeling network traffic using generalized Cauchy process. Physica A. Vol. 387, pp. 2584–2594.
120. Li S-Q., Hwang C-L. (1994). Method and apparatus for integration of traffic measurement and queuing performance evaluation in a network system. United States Patent 5583792. [interaktyvus]. [Žiūrėta 2012-02-15]. Prieiga per internetą: <http://patft.uspto.gov>.
121. Liangxiu H., Zhiwei C., Chunbo C., Chuanshan G. (2002). A new multifractal network traffic model. Chaos, Solutions & Fractals, vol. 13, no. 7, pp. 1507-1513.
122. Lianming Z., Jianping Y., Xiaoheng D. (2011). Modelling the guaranteed *QoS* for wireless sensor networks: a network calculus approach. Wireless Communications and Networking, vol. 2011, No. 1, pp. 1-14.
123. Loiseau P., Gonçalves P., Dewaele G., Borgnat P., Abry P., Primet P. V.-B. (2010). Investigating Self-Similarity and Heavy-Tailed

- Distributions on a Large-Scale Experimental Facility. Transactions on networking, vol. 18, No. 4, pp. 1261-1274.
124. Lombardo A., Morabito G., Schembra G. (1998). An accurate and treatable Markov model of MPEG-video traffic. Proceedings of IEEE INFOCOM '98, San Francisco, CA.
  125. Lopez-Guerrero M., Orozco-Barbosa L., Makrakis D. (2005). Probabilistic envelope processes for  $\alpha$ -stable self-similar traffic models and their application to resource provisioning. Performance Evaluation, vol. 61, No. 2/3, pp. 257-279.
  126. Lorenz E. N. (1963). Deterministic Nonperiodic Flow. Journal of the Atmospheric Sciences, vol. 20, no. 2, pp. 130–141.
  127. Lu X., Ke W., Huijing D. (2002). Wavelet multifractal modelling for network traffic and queuing analysis. Computer Networks and Mobile Computing, Proceedings in International Conference, pp. 260-265.
  128. M., Petersons E. (2008) Packet Loss Probability Dependence on Number of *ON-OFF* Traffic Sources in OPNET. Elektronika ir elektrotechnika. ISSN 1392-1215., vol. 85, no. 5, pp. 77-80.
  129. Maglaris B., Anastasiou D., Sen P., Karlsson G., Robbins J. (1988). Performance models of statistical multiplexing in packet video communications. IEEE Transactions on Communications, Vol. 36, No. 7, pp. 834-844.
  130. Mahmoud M. S., Hamid M. M. H. (2011). Feedback control methods for router management. IET Control Theory and Applications, vol. 5, no. 13, pp. 1509-1522.
  131. Malowidzki M. (2004). Network Simulators: A Developer's Perspective. Proceedings of International Symposium on Performance Evaluation of Computer and Telecommunication Systems (SPECTS'04), San Jose, USA.

132. Mandelbrot B. (1967). How long is coast of Britain? Statistical self-similarity and fractional dimension. *Science, New series*, vol. 156, no. 3775, pp. 636-638.
133. Manjunath D., Sikdar B. (2002). Input queued switches for variable length packets: analysis for Poisson and self-similar traffic. *Computer Communications*, vol. 25, No. 6, pp. 590-610.
134. Mao G. (2005). A real-time loss performance monitoring scheme. *Computer Communications*, vol. 28, No. 2, pp. 150-161.
135. Mario F., Nuno G., Hajduczenia M., Inacio P., Monteiro P., Silva H. (2008). Methods for estimating self-similarity degree and related estimator. European Patent EP1983687A1. [interaktyvus]. [Žiūrėta 2012-02-15]. Prieiga per internetą: <http://www.freepatentsonline.com/EP1983687.pdf>.
136. Mario F., Nuno G., Hajduczenia M., Monteiro P., Silva H. (2007). Monitoring method and apparatus of processing of a data stream with high rate/flow. European Patent EP1780955. [interaktyvus]. [Žiūrėta 2012-02-15]. Prieiga per internetą: <http://patent.ipexl.com/EP/EP1780955.html>.
137. Mark B. L., Ramamurthy G. (1998). Real-Time Estimation and Dynamic Renegotiation of UPC Parameters for Arbitrary Traffic Sources in ATM Networks. *IEEE/ACM transactions on networking*, vol. 6, No. 6, pp. 811-827.
138. Maulik K., Resnick S. (2003). The Self-Similar and Multifractal Nature of a Network Traffic Model. *Stochastic models*, Vol. 19, No. 4, pp. 549–577.
139. McBride N. (2005). Chaos theory as a model for interpreting information systems in organizations. *Information Systems Journal*, vol. 15, no. 3, pp. 233–254.
140. McCulloch J. H. (1986) Simple consistent estimators of stable distribution parameters. *Communications in Statistics - Simulation and Computation*. Vol. 15, No. 4, pp. 1109 – 1136.

141. McKee N. H., Phaal P. (1995). Method and apparatus for testing a packet-based network. United States Patent 5477531. [interaktyvus]. [Žiūrėta 2012-02-15]. Prieiga per internetą: <http://patft.uspto.gov/>.
142. McNickle D., Ewing G., Pawlikowski K. (2010). Some effects of transient deletion on sequential steady-state simulation. *Simulation Modelling Practice and Theory*, vol. 18, 2010, pp. 177-189.
143. Melakessoua F., Sorgera U., Suchaneckia Z. (2009). A multiplicative law of network traffic and its consequences. *Acta physica polonica*, vol. 40, No. 5, pp. 1507-1525.
144. Millán G., Kaschel H., Lefranc G. (2010). Discussion of the Analysis of Self-similar Teletraffic with Long-range Dependence (*LRD*) at the Network Layer Level. *Computers, Communications & Control*, vol. V, No. 5, pp. 799-812.
145. Minkevičius S. (2010) Modeling the law of the iterated logarithm in open queuing networks / *International journal of pure and applied mathematics*. Vol. 63, No. 4, pp. 487-494.
146. NetFPGA. (2012). Flexible and open platform for research, and classroom experimentation. [interaktyvus]. [Žiūrėta 2012 m. vasario 15 d.]. Prieiga per internetą: <<http://netfpga.org>>.
147. NetSim. (2012). Network Simulator. [interaktyvus]. [Žiūrėta 2012 m. vasario 15 d.]. Prieiga per internetą: <<http://www.boson.com/netsim-cisco-network-simulator>>.
148. Nogueira A., Salvador P., Valadas R., Pacheco A. (2010). Modeling self-similar traffic over multiple time scales based on hierarchical Markovian and L-System models. *Computer Communications*, vol. 33, pp. s3-s10.
149. ns-2. (2012). The Network Simulator ns-2. [interaktyvus]. [Žiūrėta 2012 m. vasario 15 d.]. Prieiga per internetą: <<http://www.isi.edu/nsnam/ns/>>.

150. ns-3 (2012). Discrete-event network simulator for Internet systems. [interaktyvus]. [Žiūrėta 2012 m. vasario 15 d.]. Prieiga per internetą: <<http://www.nsnam.org/>>.
151. Nzouonta J., Ott T., Borcea C. (2009) Impact of Queuing Discipline on Packet Delivery Latency in Ad Hoc Networks. Performance Evaluation, Vol. 66, No. 12, pp. 667-684.
152. Obaidat M. S., Boudriga N. A. (2010). Fundamentals of performance evaluation of computer and telecommunication systems. John Wiley & Sons, pp. 459.
153. Olesinski W., Rabinovitch R. (2007). Statistical trace-based methods for real-time traffic classification. United States Patent 7782793. [interaktyvus]. [Žiūrėta 2012-02-15]. Prieiga per internetą: <http://www.patentstorm.us/patents/7782793.html>.
154. Opnet – tinklo simulatorius, (2012) <<http://www.opnet.com>>.
155. Park D., Rilett L. R., Gajewski B. J., Spiegelman C. H., Choi C. (2009). Identifying optimal data aggregation interval sizes for link and corridor travel time estimation and forecasting. Transportation, vol. 36, pp. 77-95.
156. Park K., Willinger W. (2000) Self-Similar Network Traffic and Performance Evaluation. John Wiley & Sons, pp. 558.
157. Patrick Loiseau, Paulo Gonçalves, Member, IEEE, Guillaume Dewaele, Pierre Borgnat, Member, IEEE,
158. Pawlikowski K., Yau V. (2000). Analysis of Dual\_Bus Metropolitan Area Networks Using Distributed Quantitative Stochastic Simulation. Simulation, vol. 75, no. 3, pp. 157-169.
159. Paxson V., Floyd S. (1995). Wide-area Traffic: The Failure of Poisson Modeling. IEEE/ACM Transactions on Networking, pp.226-244.
160. Peng Y. (2004). Flow Controller Design and Performance Analysis for Self-Similar Network Traffic. Electronic engineering, vol.12, No. 3, pp. 167-180.



161. Plėštys R., Kavaliūnas R., Vilutis G.(2008). Kompiuterių tinklai. Technologija, Kaunas, pp. 152.
162. Pranevičius H. (2008) Sudėtingų sistemų formalizavimas ir analizė. Kauno technologijos universitetas.
163. Press S. J. (1972) Estimation in univariate and multivariate stable distribution. The Journal of the American Statistical Association. Vol. 67, pp. 842-846.
164. Pu Y. K., Bhatia C., Consel S., Coll C. (2006). Efficient Packet Processing in User-Level OSes: A Study of UML. Local Computer Networks, Proceedings 31st IEEE Conference on, 14-16 Nov., pp. 63-70.
165. Quan Z., Chung J.-M. (2004). Statistical admission control for real-time services under earliest deadline first scheduling. Computer Networks, vol. 48, pp. 137–154.
166. Rachev S. T., Mittnik S. (2000) Stable Paretian Models in Finance. Chichester: Wiley.
167. Ramanan K., Cao J. (2002) A Poisson limit for buffer overflow probabilities. In INFOCOM.
168. Rindzevičius R., Tervydis P. (2003). Balso perdavimo interneto tinklu charakteristikų tyrimas. Elektronika ir elektrotechnika, Vilnius, Nr.5(47).
169. Robert S., Le Boudec J.-Y. (1995). A Markov Modulated process for self-similar traffic. LCA-CONF-1995-002, Germany.
170. Roberts J. W. (2000). Engineering for quality of service. Self-similar traffic and performance evaluation, edited by Park K. and Willinger W., John Wiley and Sons, pp. 401-420.
171. Roughan M., Veitch D., Abry P. (2000). Real-Time Estimation of the Parameters of Long-Range Dependence. IEEE/ACM transactions on networking, vol. 8, no. 4, pp. 467-478.
172. Ruelle D. F., Takens F. (1971) On the nature of turbulence. Communications of Mathematical Physics. Vol. 20, pp. 167-192.

173. Rutka G. (2008). Network Traffic Prediction using ARIMA and Neural Networks Models. *Electronics and electrical engineering*, No. 4(84), pp. 47-52.
174. Saaty T. L. (1957). Resume of useful formulas in queuing theory. *Operations research*, vol. 5, No. 2, pp. 161-187.
175. Sakalauskas L. (2000). Masinio aptarnavimo teorija. VGTU, „Technika“, pp. 156.
176. Samorodnitsky G, Taqqu M.S. (1994) *Stable Non-Gaussian Processes: Stochastic Models with Infinite Variance*. Chapman and Hall, New York.
177. Samorodnitsky G. (2006a) Long memory and self-similar processes. *Annales de la faculté des sciences de Toulouse Sér. 6*. Vol. 15, No. 1, pp. 107 - 123.
178. Samorodnitsky G. (2006b) Long Range Dependence. *Foundations and Trends in Stochastic Systems*. Vol. 1, No. 3, pp. 163 - 257.
179. Schmidt A., Pawlikowski K., McNickle D. (2009). Sequential estimation of the steady-state variance in discrete event simulation. *Proceedings ECMS, Madrid, June 9-12*, pp. 630-635.
180. Schnabl W., Stadler P. F., Forst C., Schuster P. (1991). Full characterization of a strange attractor: Chaotic dynamics in low-dimensional replicator systems. *Physica D: Nonlinear Phenomena*, vol. 48, no. 1, pp. 65-90.
181. Shang P., Lu Y., Kamae S. (2008). Detecting long-range correlations of traffic time series with multifractal detrended fluctuation analysis. *Chaos, Solitons & Fractals*, vol. 36, no. 1, pp. 82-90.
182. Shang P., Wan M., Kama S. (2007). Fractal nature of highway traffic data. *Computers and Mathematics with Applications*, vol. 54, pp. 107–116.
183. Shannon R. E. (1975). *Systems Simulation: The Art and Science*. Prentice Hall, pp. 368.

184. Sheluhin O. I., Smolskiy S. M., Osin A. V. (2007). Self-similar processes in telecommunications. JohnWiley & Sons, pp. 314.
185. Shena H., Zhua Z., Lee T. C. M. (2007). Robust estimation of the self-similarity parameter in network traffic using wavelet transform. *Signal Processing*, vol. 87, pp. 2111–2124.
186. Sim L. O., Cheong Y. H., Jarrett A. L., Bey S., Eustace A. R., Pettit M. J. (2003). Network resource monitoring and measurement system and method. United States Patent 7376722. [interaktyvus]. [Žiūrėta 2012-02-15]. Prieiga per internetą: <http://www.wikipatents.com/US-Patent-7376722>.
187. Smith E. A. (2006). Understanding performance issues in IP networks. *BT Technology Journal*, vol. 24, No 4, pp. 167-179.
188. Soldatos J., Vayias E., Stathopoulos P., Mitrou N. (2004). Enforcing Effective Rates for Packet-Level *QoS* Control in IP Networks: Theory and Validation Based on Real Traffic Data. *Telecommunication Systems*, vol. 27, No. 1, pp. 9-31.
189. Sornette D. (2004) *Critical Phenomena in Natural Sciences: Chaos, Fractals, Self-organization and Disorder: Concepts and Tools* (second ed.), Springer, Berlin.
190. Stoev S., Michailidis G., Taqqu M.S. (2011). Estimating heavy-tail exponents through max self-similarity. *IEEE Transactions on Information Theory*, vol. 57, no. 3, pp. 1615-1635.
191. Stoev S., Michailidis, G. (2010). On the estimation of the heavy-tail exponent in time series using the max-spectrum. *Applied Stochastic Models in Business and Industry*, vol. 26, no. 3, pp. 224-253.
192. Sun W., Rachev S., Fabozzi F. J. (2007). Fractals or I.I.D.: Evidence of Long-Range Dependence and Heavy Tailedness from Modeling German Equity Market Returns. *Journal of Economics and Business*. Vol. 59, No. 6, pp. 575-595.

193. Surgailis D., Teyssiere G., Vaičiulis M. (2008). The increment ratio statistics. *Journal of Multivariate Analysis*. Vol. 99, No 3, pp. 510-541.
194. Sychyov V. (2003). *Fractan* 4.4. Available at <<http://softsearch.ru/programs/175-207-Fractan-download.shtml>>, 2003.
195. Taqqu M.S. and Teverovsky V. (1998). Estimating long-range dependence in finite and infinite variance series. In: R. Adler, R. Feldman and M.S. Taqqu (eds), *A Practical Guide to Heavy Tails: Statistical Techniques for Analyzing Heavy-Tailed Distributions*. Birkhauser, Boston, pp. 177-217.
196. Theiler J. (1990). Estimating fractal dimension. *Journal the optical society*, vol. 7, no. 6, pp. 1055-1073.
197. Tom's Hardware (2012). The authority on tech. [interaktyvus]. [Žiūrėta 2012-02-15]. Prieiga per internetą: <http://www.tomshardware.com/>.
198. Tran H. V. (2005). Call admission control for nonhomogenous sources. United States Patent 6856948. [interaktyvus]. [Žiūrėta 2012-02-15]. Prieiga per internetą: <http://www.patentstorm.us/patents/6856948.html>.
199. Ulogd. (2012). Userspace logging daemon. [interaktyvus]. [Žiūrėta 2012-01-07]. Prieiga per internetą: <http://www.netfilter.org/projects/ulogd/index.html>
200. Vasilou N. (2000) Overview of Internet *QoS* and Web Server *QoS*. UWO Reading Course Paper, The Pennsylvania State University <<http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.43.6544>>.
201. Veitch D. N. Roughan M. Abry P. (1999). Real-Time Estimation Of Long Range Dependent Parameters. WO1999040703 Patent. [interaktyvus]. [Žiūrėta 2012-02-15]. Prieiga per internetą: <http://www.wipo.int/patentscope/search/en/WO1999040703>.

202. Veitch D., Hohn N., Abry P. (2005). Multifractality in *TCP/IP* traffic: the case against. *Computer Networks*, vol. 48, no. 3, pp. 293-313.
203. Vieira F. H. T., Bianchi G. R., Lee L. L. (2010). A network traffic prediction approach based on multifractal modelling. *High Speed Networks*, vol. 17, pp. 83-96.
204. Weisang G., Awazu Y. (2008). Vagaries of the Euro: An Introduction to ARIMA Modeling”. *Case Studies in Business, Industry and Government Statistics*, Vol. 2, No. 1.
205. Weron R. (2004). Computationally intensive Value at Risk calculations, in J. E. Gentle, W. Härdle, Y. Mori (eds.) *Handbook of Computational Statistics*, Springer, Berlin, pp. 911-950.
206. Willinger W., Paxson V., Taqqu M. S. (1998). Self-Similarity and Heavy Tails: Structural Modeling of Network Traffic. *A Practical Guide to Heavy Tails: Statistical Techniques and Applications*. Editors: Adler R. J., Feldman R. E., Taqqu M. S., John Wiley & Sons, pp. 27-55.
207. Wilson M. (2011). A Historical View of Network Traffic Models. Citeseer, Technical report, pp. 1-15, <[http://www.cs.wustl.edu/~jain/cse567-06/ftp/traffic\\_models2.pdf](http://www.cs.wustl.edu/~jain/cse567-06/ftp/traffic_models2.pdf)>.
208. Xia Y., Liu N., Iu H. H. C. (2009). Oscillation and chaos in a deterministic traffic network. *Chaos, Solutions & Fractals*, vol. 42, no. 3, pp. 1700-1704.
209. Xiaolong J., Geyong M. (2009). Modelling and Analysis of Priority Queuing Systems with Multi-Class Self-Similar Network Traffic: A Novel and Efficient Queue-Decomposition Approach. *IEEE Transactions on Communications*; vol. 57, No. 5, pp. 1444-1452.
210. Xie J., Jiang Y. (2009) Stochastic Network Calculus Models under Max-Plus Algebra. *Global Telecommunications Conference, GLOBECOM 2009*. IEEE.
211. Yanga W. S., Yang E. S., Kimc H. J., Kim D. K. (2010). Estimation of spectrum requirements for mobile networks with self-

- similar traffic, handover, and frequency reuse. *Mobile Information Systems*, vol. 6, pp. 281–291.
212. Yilmaz C., Gürdal O., Sayan, H. H. (2008) Investigation of the network delay on Profibus-DP based network (2008). *Elektronika ir elektrotechnika*. ISSN 1392-1215. Vol. 87, No. 7, pp. 23-26
213. Yin H., Lin C., Sebastien B., Li B., Min G. (2005). Network traffic prediction based on a new time series model. *International Journal of Communication Systems*, vol. 18, No. 8, pp. 711–729.
214. Younis O., Fahmy S. (2005). FlowMate: Scalable On-Line Flow Clustering. *IEEE/ACM transactions on networking*, vol. 13, No. 2, pp. 288-301.
215. Zaidi A. R., Hakami S., Landfeldt B., Moors T. (2010). Real-time detection of traffic anomalies in wireless mesh networks. *Wireless Networks*, vol. 16, pp. 1675–1689.
216. Zhang W., Wu Z., Yang G. (2004). Chaotic network attractor in packet traffic series. *Computer Physics Communications*, vol. 161, no. 3, pp. 129-142.
217. Zhu R. (2011). Intelligent rate control for supporting real-time traffic in WLAN mesh networks. *Network and Computer Applications*, vol. 34, No. 5, pp. 1449-1458.
218. Zhu Z.; Fu C. (2008). A Chaotic Dynamical Model for Internet Traffic. 4th International Conference on Natural Computation, Jian, China, pp. 581-585.
219. Žvironienė A., Navickas Z., Rindzevičius R. (2006). Bursty traffic simulation by *ON-OFF* model. *Elektronika ir elektrotechnika*, vol. 6, No. 70, pp. 65-68.
220. Олифер В. Г., Олифер Н. А. (2008). Компьютерные сети. Принципы, технологии, протоколы. 3-е изд.
221. Хинчин А. Я. (1955). Математические методы теории массового обслуживания. АН СССР, pp. 3–122.

222. Хинчин А. Я. (1963). Теории массового обслуживания. Под редакцией Б. В. Гнеденко, Москва, pp. 528.

## Reziumė

Disertacijos tyrimų sritis – kompiuterių tinklo paketinio srauto savybės, tinklo mazgo savybių įtaka srauto aptarnavimui, tinklo srauto savybių realaus laiku analizės metodai ir jų taikymas kompiuterių tinklo srauto kaitos dinaminiam prognozavimui. Tyrimų objektas – kompiuterių tinklo paketinio srauto savybės, tinklo mazgo savybių įtaka paketinio kompiuterių tinklo srauto aptarnavimui, realaus laiko tinklo srauto savybių analizės metodai ir jų taikymas tinklo srauto kaitos dinaminiam prognozavimui. Darbo tikslas – iširti fraktalinius procesus kompiuterių tinkluose, remiantis gautais rezultatais parinkti metodus, tinkamus tinklo srauto analizei realiu laiku, ir sukurti savastingumo matavimo realiu laiku metodiką bei ją pritaikyti kompiuterių tinklų aptarnavimo kokybei gerinti.

Išanalizuotos tinklo komponentų matematinio modeliavimo galimybės, kompiuterių tinklo paketinio srauto modeliai ir modeliai, naudojantys aptarnavimo teorijos instrumentus. Parengtas tinklo srauto savybių analizės paketas, panaudotas kompiuterių tinklų fraktališkumo ir savastingumo tyrimo metodams analizuoti, vertinti ir palyginti. Iširti paketinio kompiuterių tinklo srauto laiko eilučių analizės, dažninių/banginių savybių įvertinimo, laiko eilutės stabilumo parametrų įverčiais grindžiami bei chaoso teorijos priemonėmis įvertinami savastingumo analizės metodai.

Sudarytas tinklo srauto savastingumo realiu laiku analizės paketas, kurį naudojant savastingumo matavimui realiu laiku atrinktas robastinis *regresijos* metodas ir *IR* statistika.

Išanalizuota fraktalinių procesų imitavimo programinė įranga. Pritaikius stochastiškai apribotą tinklo komunikavimo modelį ir algebra plus priemones aprašytas GI/G/10//N kompiuterių tinklo mazgo modelis. Sukurta kompiuterių tinklo savastingojo ir *Puasono* srautų imitavimo sistema ir, ja naudojantis, iširta tinklo mazgo savybių įtaka tinklo srauto aptarnavimo kokybei. Sukurta metodika pritaikyta kompiuterių tinklo aptarnavimo kokybei gerinti.



Sukurta kompiuterių tinklo srauto savastingumo matavimo realiu laiku metodika, kuri rekurentiškai formuoja eilutę, skaičiuoja statistinius jos įverčius ir rezultatus saugo laikinojoje atmintyje. Ši metodika nuo kitų skiriasi tuo, kad realiu laiku agreguoja tinklo srautą taikant judančių vidurkių glotninimo arba perduotų duomenų srauto sumos per laiko intervalą metodą, suformuotą eilutę saugo laikinoje atmintyje, rekurentiškai skaičiuoja dinamiškai formuojamos eilutės savastingumo parametrus, apskaičiuojant  $\alpha$  stabilumo indeksą,  $IR$  įvertį ir *Hurst'o* koeficientą, skaičiavimo rezultatus tolesnei analizei saugo laikinojoje arba/ir pastoviojoje atmintyje. Metodas užpatentuotas Lietuvos Respublikos valstybiniame patentų biure, patento numeris LT20011099. Panaudojant paketinio kompiuterių tinklo srauto savastingumo analizės realiu laiku metodą parengtas atviras tinklo mazgo valdymo modelis su grįžtamuju ryšiu.

Tyrimų rezultatai buvo pristatyti ir aptarti 2 tarptautinėse konferencijose, viename tarptautiniame seminare, 2 respublikinėse konferencijose bei 3 respublikiniuose seminaruose. Publikuoti 2 straipsniai užsienio mokslo leidiniuose, įtrauktuose į Mokslinės informacijos instituto pagrindinių žurnalų sąrašą su citavimo indeksu, vienas – tarptautinių konferencijų darbuose, įtrauktuose į Mokslinės informacijos instituto sąrašą ir 4 recenzuojamuose Lietuvos ir užsienio leidiniuose.

## Summary

The field of the dissertation research is features of computer network packet traffic, the impact of network node features on traffic service, methods of real-time analysis of network traffic features and their application for dynamic prognostication of computer network packet traffic variance. The object of the research is the features of computer network packet traffic, the impact of network node features on computer network traffic service, methods of real-time network traffic features analysis and their application for dynamic prognostication of network traffic variances.

The aim of work is to investigate fractal processes in computer networks, grounding on the results obtained to select methods suitable for real-time analysis of network traffic and to work out methods for real-time measurement of self-similarity as well as to apply it for perfection of computer networks service quality.

Possibilities for mathematical modelling of network components, computer network packet traffic models and models using service theory instruments have been analysed. The package of network traffic features analysis has been worked out; it was used for analysis, assessment and comparison of methods for computer networks fractality and self-similarity research. For assessment of self-similarity of the network traffic time lines analysis, frequency/wave feature estimates, self-similarity analysis methods based on time line stability parameters estimators and assessed by the chaos theory means have been studied.

The package of real-time network traffic self-similarity analysis has been worked out; by using it for real-time measurement of self-similarity the robust *regression* method and *IR* statistics were selected.

Software for simulation of fractal processes was analysed. After applying a stochastically limited network communication model and algebra plus means, the GI/G/10//N computer network node is described.

The simulation system of computer network self-similar and Poisson traffics was created and, by using it, the impact of network node features on quality of network traffic service was investigated. The worked methods were applied for perfection of computer network service quality.

Methods for computer network traffic self-similarity real-time measurement have been worked out; they recurrently form a queue, calculate its statistical parameters and stored results in a temporal memory. These methods differ from others by real-time aggregation of network traffic by applying the method of smoothing moving means or the sum of transmitted data traffic during a time interval, the formed queue is stored in a temporal memory, recurrently calculates self-similarity parameters of a dynamically formed queue calculating  $\alpha$ -stability index, *IR* estimator and *Hurst* coefficient, stores the calculations results in a temporal or/and permanent memory for further analysis. The method has been patented at the State Patent Bureau of the Republic of Lithuania, patent No. LT20011099. By using the method real-time analysis of computer network packet traffic self-similarity, an open network node control model including feedback has been worked out.

Research results have been presented and discussed in two international conferences, one international seminar, two national conferences and three national seminars. Two articles have been published in foreign scholarly publications included into the Institute's of Scientific Information list of main journals with the citation index, one published in international conference proceedings included into the Institute's of Scientific Information list and four appeared in reviewed Lithuanian and foreign publications.

**1 priedas.** *Hurst*’o koeficiento pasikliautieji intervalai (apskaičiuota programa *Fractan* 4.4).

**1priedas, 1 lentelė.** *Hurst*’o koeficiento reikšmių pasiskirstymas

	$0.5 < H < 1.0$						$0 \leq H < 0.5$						$1.0 \leq H$																																									
	81,92%																		11,30%																		6,78%																	
	100ms			500ms			1000ms			100ms			500ms			1000ms			100ms			500ms			1000ms																													
$x_{t=1}$	81,36%			76,27%			88,14%			10,17%			15,25%			8,47%			8,47%			8,47%			3,39%																													
	Mažiausias	Vidutinis	Didžiausias	Mažiausias	Vidutinis	Didžiausias	Mažiausias	Vidutinis	Didžiausias	Mažiausias	Vidutinis	Didžiausias	Mažiausias	Vidutinis	Didžiausias	Mažiausias	Vidutinis	Didžiausias	Mažiausias	Vidutinis	Didžiausias	Mažiausias	Vidutinis	Didžiausias																														
	88,24%	70,37%	93,33%	70,59%	74,07%	86,67%	76,47%	85,19%	93,33%	11,76%	14,81%	0,00%	11,76%	18,52%	13,33%	23,53%	11,11%	0,00%	0,00%	14,81%	6,67%	17,65%	7,41%	0,00%	0,00%	3,70%	6,67%																											
	$x_{t=100}$	80,23%									12,43%									7,34%																																		
		83,05%			76,27%			81,36%			13,56%			15,25%			8,47%			3,39%			8,47%			10,17%																												
		Mažiausias	Vidutinis	Didžiausias	Mažiausias	Vidutinis	Didžiausias	Mažiausias	Vidutinis	Didžiausias	Mažiausias	Vidutinis	Didžiausias	Mažiausias	Vidutinis	Didžiausias	Mažiausias	Vidutinis	Didžiausias	Mažiausias	Vidutinis	Didžiausias	Mažiausias	Vidutinis	Didžiausias																													
		88,24%	81,48%	80,00%	70,59%	74,07%	86,67%	88,24%	77,78%	93,33%	11,76%	14,81%	13,33%	11,76%	18,52%	13,33%	5,88%	11,11%	0,00%	0,00%	3,70%	6,67%	17,65%	7,41%	0,00%	5,88%	14,81%	6,67%																										

**1priedas, 2 lentelė.** *Hurst*’o koeficiento pasikliautieji intervalai

	$\overline{H_{\Theta}}$	$\sigma_{H_{\Theta}}$	$[\overline{H_{\Theta}} - 1,96\sigma_{H_{\Theta}}, \overline{H_{\Theta}} + 1,96\sigma_{H_{\Theta}}]$
$x_{t=100}^{\Sigma_{\min}}$	0,71	0,18	[0,74; 0,74]
$x_{t=500}^{\Sigma_{\min}}$	0,69	0,24	[0,63; 0,64]
$x_{t=1}^{\Sigma_{\min}}$	0,61	0,14	[0,64; 0,644]
$x_{t=100}^{\Delta_{\min}}$	0,68	0,16	[0,69; 0,70]
$x_{t=500}^{\Delta_{\min}}$	0,69	0,23	[0,63; 0,64]
$x_{t=1}^{\Delta_{\min}}$	0,74	0,18	[0,70; 0,704]
$x_{t=100}^{\Sigma_{\Delta}}$	0,76	0,19	[0,786; 0,79]
$x_{t=500}^{\Sigma_{\Delta}}$	0,67	0,18	[0,68; 0,69]
$x_{t=1}^{\Sigma_{\Delta}}$	0,73	0,18	[0,688; 0,692]
$x_{t=100}^{\Delta_{\Delta}}$	0,71	0,18	[0,68; 0,69]
$x_{t=500}^{\Delta_{\Delta}}$	0,67	0,17	[0,67; 0,68]
$x_{t=1}^{\Delta_{\Delta}}$	0,73	0,20	[0,69; 0,693]

$x_{t=100}^{\Sigma_{\max}}$	0,77	0,14	[0,776; 0,781]
$x_{t=500}^{\Sigma_{\max}}$	0,69	0,16	[0,67; 0,68]
$x_{t=1}^{\Sigma_{\max}}$	0,72	0,16	[0,681; 0,686]
$x_{t=100}^{\Delta_{\max}}$	0,74	0,17	[0,746; 0,752]
$x_{t=500}^{\Delta_{\max}}$	0,69	0,16	[0,67; 0,68]
$x_{t=1}^{\Delta_{\max}}$	0,79	0,15	[0,755; 0,760]
$x_t^{\Sigma_{\min}}$	0,67	0,19	[0,649; 0,653]
$x_t^{\Delta_{\min}}$	0,70	0,19	[0,660; 0,664]
$x_t^{\Sigma_{\Delta}}$	0,72	0,18	[0,694; 0,696]
$x_t^{\Delta_{\Delta}}$	0,71	0,18	[0,676; 0,679]
$x_t^{\Sigma_{\max}}$	0,72	0,16	[0,706; 0,709]
$x_t^{\Delta_{\max}}$	0,74	0,16	[0,709; 0,712]

**2 priedas.** *Hurst*'o koeficiento pasikliautieji intervalai (apskaičiuota programa *Selfis*).

**2 priedas, 1 lentelė.** *Hurst*'o koeficiento reikšmių pasiskirstymas

	<b>0&lt;H&lt;0.5</b>									<b>0.5&lt;H&lt;1.0</b>									<b>H=0.5</b>																					
	47,07%									42,82%									8,63%																					
	<b>100ms</b>			<b>500ms</b>			<b>1000ms</b>			<b>100ms</b>			<b>500ms</b>			<b>1000ms</b>			<b>100ms</b>			<b>500ms</b>			<b>1000ms</b>															
$x_{t \downarrow}$	<b>Mažiausias</b>	42,86%		<b>Mažiausias</b>	64,29%		<b>Mažiausias</b>	47,62%		<b>Mažiausias</b>	42,86%		<b>Mažiausias</b>	25,00%		<b>Mažiausias</b>	47,62%		<b>Mažiausias</b>	14,29%		<b>Mažiausias</b>	10,71%		<b>Mažiausias</b>	7,69%		<b>Mažiausias</b>	0,00%		<b>Mažiausias</b>	10,71%		<b>Mažiausias</b>	2,86%					
	<b>Vidutinis</b>	52,38%		<b>Vidutinis</b>	48,35%		<b>Vidutinis</b>	45,24%		<b>Vidutinis</b>	33,33%		<b>Vidutinis</b>	43,96%		<b>Vidutinis</b>	44,05%		<b>Vidutinis</b>	57,14%		<b>Vidutinis</b>	14,29%		<b>Vidutinis</b>	10,71%		<b>Vidutinis</b>	5,71%		<b>Vidutinis</b>	10,71%		<b>Vidutinis</b>	2,86%					
	<b>Didžiausias</b>	51,43%		<b>Didžiausias</b>	40,00%		<b>Didžiausias</b>	31,43%		<b>Didžiausias</b>	37,14%		<b>Didžiausias</b>	43,96%		<b>Didžiausias</b>	54,29%		<b>Didžiausias</b>	57,14%		<b>Didžiausias</b>	14,29%		<b>Didžiausias</b>	10,71%		<b>Didžiausias</b>	7,69%		<b>Didžiausias</b>	0,00%		<b>Didžiausias</b>	10,71%		<b>Didžiausias</b>	2,86%		
	$x_{t \Delta}$	<b>Mažiausias</b>	42,86%		<b>Mažiausias</b>	64,29%		<b>Mažiausias</b>	47,62%		<b>Mažiausias</b>	46,43%		<b>Mažiausias</b>	25,00%		<b>Mažiausias</b>	42,86%		<b>Mažiausias</b>	10,71%		<b>Mažiausias</b>	9,18%		<b>Mažiausias</b>	10,71%		<b>Mažiausias</b>	8,79%		<b>Mažiausias</b>	0,00%		<b>Mažiausias</b>	8,33%		<b>Mažiausias</b>	11,90%	
		<b>Vidutinis</b>	44,90%		<b>Vidutinis</b>	48,35%		<b>Vidutinis</b>	53,57%		<b>Vidutinis</b>	45,92%		<b>Vidutinis</b>	42,86%		<b>Vidutinis</b>	38,10%		<b>Vidutinis</b>	40,48%		<b>Vidutinis</b>	9,18%		<b>Vidutinis</b>	5,71%		<b>Vidutinis</b>	8,79%		<b>Vidutinis</b>	5,71%		<b>Vidutinis</b>	8,33%		<b>Vidutinis</b>	11,90%	
		<b>Didžiausias</b>	34,29%		<b>Didžiausias</b>	40,00%		<b>Didžiausias</b>	47,62%		<b>Didžiausias</b>	60,00%		<b>Didžiausias</b>	42,86%		<b>Didžiausias</b>	54,29%		<b>Didžiausias</b>	40,48%		<b>Didžiausias</b>	10,71%		<b>Didžiausias</b>	5,71%		<b>Didžiausias</b>	10,71%		<b>Didžiausias</b>	8,79%		<b>Didžiausias</b>	5,71%		<b>Didžiausias</b>	11,90%	

2 priedas, 2 lentelė. Hurst'o koeficiento įverčiai

	0<H<0.5				0.5<H<1.0			
	Min	Max	$\overline{H_{\Theta}}$	$\sigma_{H_{\Theta}}$	Min	Max	$\overline{H_{\Theta}}$	$\sigma_{H_{\Theta}}$
$x_{t=100}^{\Sigma \min}$	0,504	0,86	0,563	0,095374	0,101	0,451	0,2075	0,13531
$x_{t=500}^{\Sigma \min}$	0,508	0,621	0,542	0,0415	0,041	0,491	0,349	0,162226
$x_{t=1}^{\Sigma \min}$	0,504	0,93	0,6425	0,153756	0,046	0,474	0,3235	0,161881
$x_{t=100}^{\Sigma \Delta}$	0,508	0,717	0,595	0,062544	0,004	0,478	0,2615	0,134153
$x_{t=500}^{\Sigma \Delta}$	0,503	0,635	0,531	0,034011	0,027	0,487	0,264	0,161563
$x_{t=1}^{\Sigma \Delta}$	0,501	0,786	0,537	0,0627	0,018	0,498	0,203	0,150247
$x_{t=100}^{\Delta \max}$	0,501	0,751	0,606	0,08232	0,022	0,499	0,3225	0,146162
$x_{t=500}^{\Delta \max}$	0,507	0,631	0,552	0,037102	0,026	0,466	0,124	0,164835
$x_{t=1}^{\Delta \max}$	0,506	0,998	0,5595	0,11121	0,012	0,402	0,16	0,130508
$x_{t=100}^{\Delta \min}$	0,516	0,602	0,545	0,027849	0,247	0,482	0,386	0,074671
$x_{t=500}^{\Delta \min}$	0,51	0,627	0,542	0,042809	0,024	0,491	0,297	0,180603
$x_{t=1}^{\Delta \min}$	0,503	0,818	0,634	0,116554	0,044	0,487	0,343	0,17361
$x_{t=100}^{\Delta \Delta}$	0,502	0,64	0,541	0,034684	0,092	0,498	0,365	0,104591
$x_{t=500}^{\Delta \Delta}$	0,502	0,64	0,534	0,033445	0,061	0,495	0,2785	0,159197
$x_{t=1}^{\Delta \Delta}$	0,501	0,583	0,529	0,023257	0,022	0,497	0,239	0,15924
$x_{t=100}^{\Delta \max}$	0,501	0,675	0,536	0,050987	0,275	0,492	0,384	0,070016
$x_{t=500}^{\Delta \max}$	0,501	0,634	0,552	0,038453	0,109	0,484	0,2705	0,150718
$x_{t=1}^{\Delta \max}$	0,506	0,617	0,56	0,036871	0,078	0,497	0,3135	0,162648