



KAUNO TECHNOLOGIJOS UNIVERSITETAS  
INFORMATIKOS FAKULTETAS  
SISTEMINĖS ANALIZĖS KATEDRA

Arvydas Burdulis

**Internetinė platforma interaktyviam komandiniam  
darbui realiu laiku**

Magistro darbas

Darbo vadovas  
M. Kavaliauskas

Kaunas  
2012

## Turinys

|      |  |    |
|------|--|----|
| 1.   | ĮVADAS.....  | 3  |
| 2.   | PLATFORMOS KŪRIMO POREIKIS .....                     | 4  |
| 3.   | REALIZAVIMO GALIMYBIŲ TYRIMAS .....                  | 9  |
| 3.1. | Galimi architektūros tipai.....                      | 9  |
| 3.2. | Rinkoje esančių projektų apžvalga.....               | 10 |
| 3.3. | Reikalavimai kuriamai platformai.....                | 13 |
| 3.4. | Platformos architektūros pasirinkimas .....          | 14 |
| 4.   | PLATFORMOS PROJEKTAVIMAS .....                       | 17 |
| 4.1. | Komunikavimo protokolo pasirinkimas.....             | 17 |
| 4.2. | Platformą sudarantys komponentai .....               | 18 |
| 4.3. | Klientų konkurencijos problema .....                 | 19 |
| 4.4. | Centrinis serveris.....                              | 22 |
| 4.5. | Žaidimų serveris.....                                | 25 |
| 4.6. | Žaidimo karkasas .....                               | 27 |
| 4.7. | Žaidimų portalas.....                                | 28 |
| 4.8. | Komunikavimo tarp komponentų detalus aprašymas ..... | 30 |
| 5.   | KONCEPCINĖS VERSIJOS REALIZACIJA .....               | 33 |
| 5.1. | Programavimo aplinkos pasirinkimas .....             | 33 |
| 5.2. | Realizuotas komandų rinkinys .....                   | 34 |
| 6.   | PLATFORMOS VEIKIMO TYRIMAS .....                     | 36 |
| 7.   | IŠVADOS .....  | 40 |
| 8.   | LITERATŪRA .....                                     | 41 |
| 9.   | TERMINŲ IR SANTRUMPŲ ŽODYNAS .....                   | 44 |

## 1. ĮVADAS

Didėjant informacinių technologijų įtakai ir pritaikymui įvairiose sferose, didėja ir jų naudojimas mokymo procese. Jau dabar mokyklose pamokų metu mokytojai sėkmingai naudojami specializuotais internetiniais resursais, tokiais kaip [www.mokinukai.lt](http://www.mokinukai.lt), [www.moko.lt](http://www.moko.lt). Kompiuterinių mokomųjų priemonių privalumas prieš knygas – galimybė perteikti animaciją, garsus, interaktyvumą. Tai sustiprina mokinių motyvaciją mokytis, taip pat pagerina žinių įsisavinimą.

Deja, kompiuterinės mokymo(si) priemonės paprastai yra labai ribotos ir nesuteikia galimybės mokiniams bendrauti, dalyvauti komandiniame darbe sprendžiant bendras užduotis. Tai yra vieni iš pagrindinių trūkumų, lyginant su tradiciniu mokymo procesu. Norint juos pašalinti, reikalinga speciali internetinė platforma, įgalinanti mokinius bendrauti realiu laiku, dalyvauti komandiniame darbe.

Pati platforma architektūros atžvilgiu gali būti labai įvairi. Norint gauti geriausią rezultatą, prieš pradėdant projektavimą reikia atidžiai išnagrinėti reikalavimus, turimas technines galimybes, atsižvelgti į galimą pritaikymą kitose srityse.

Šio darbo tikslai:

1. Ištirti tokios platformos poreikį;
2. Išanalizuoti šia tema nuveiktus darbus ir rinkoje esančius panašios paskirties produktus;
3. Absibrėžti funkcinis reikalavimus ir suprojektuoti juos atitinkančią platformą;
4. Realizuoti platformos koncepcinę versiją, ištirti jos veikimo sklandumą, atitikimą užsibrėžtiems reikalavimams.

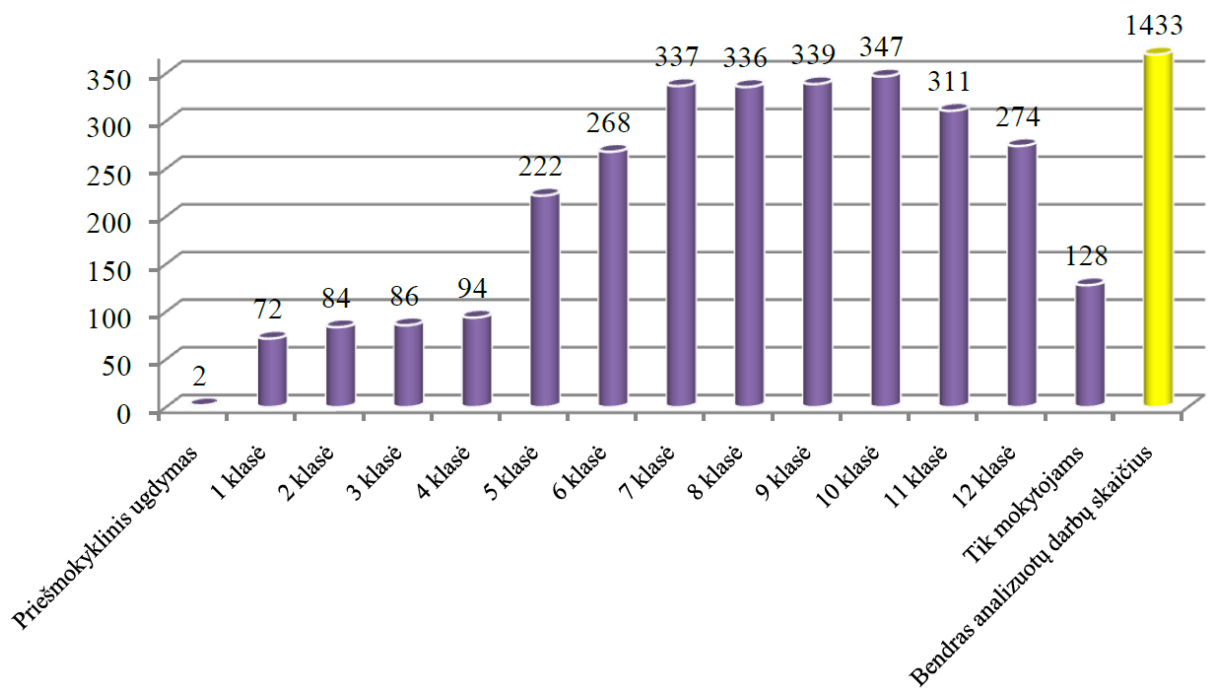
Sukurtoji platforma skirta realiam pritaikymui pradinių klasių mokinių mokymo procese, mokomajame portale [www.mokinukai.lt](http://www.mokinukai.lt), šis pritaikymas yra atskiras etapas ir į magistrinį darbą neįeina.

## 2. PLATFORMOS KŪRIMO POREIKIS

Informacinių technologijų panaudojimas žmonių gyvenime vis didėja. Atsiranda vis naujų sričių, kuriose kompiuteriai palengvina darbą ar įgalina atlikti tai, kas anksčiau buvo neįmanoma. Tiesa, norint išnaudoti atsiradusias naujas galimybes, būtina žmonių kompetencija informacinių ir komunikacinių technologijų (IKT) panaudojime. Tam tikslui rengiami apmokymai, siekiama didinti kompiuterinį raštingumą. Jaunoji karta su kompiuteriais ir jų teikiama nauda supažindinama jau mokykloje. Mokyklose kompiuteris sėkmingai naudojamas mokslams, ten mokinių kompiuterinis raštingumas pasiekiamas per praktiką – vienu šūviu nušaunami du zuikiai.

Kompiuterinių mokomųjų priemonių (KMP) panaudojimas mokymo procese yra viena svarbiausių švietimo sistemos tobulinimo krypčių, tai patvirtina LR švietimo ir mokslo ministro įsakymas „Dėl informacinių ir komunikacinių technologijų diegimo į bendrąjį lavinimą ir profesinį mokymą 2008-2012 metų strategijos patvirtinimo“ [1]. Jame teigiama, kad IKT turi būti taikomos ugdymo procese siekiant, kad ir mokymas, ir mokymasis kokybiškai pakistų, kad jis vis labiau atitiktų atskiro individo poreikius, kad iš esmės pagerėtų mokyklų, mokytojų ir mokinių darbo veiksmingumas, o pats mokymas ir mokymasis taptų patrauklesnis mokiniui, kad kiekvienam šalies piliečiui būtų suteikiama daugiau galimybių kūrybinėms galioms skleisti ir savarankiškos veiklos poreikiams tenkinti. Interneto ir tinklų technologijos turi apimti visas švietimo sritis, tapti kasdienėmis priemonėmis, kuriomis mokiniai naudotųsi ir mokykloje, ir namie. Modernios komunikacijos priemonės turi užtikrinti naują bendradarbiavimo kultūrą mokykloje, skatinti mokymąsi bendradarbiaujant. Norint pasiekti šiuos tikslus, mokyklos aprūpinamos kompiuteriais ir kita susijusia įranga, skatinamas ir finansuojamas KMP lietuvių kalba kūrimas, užsienio kūrėjų sukurtų KMP lokalizavimas.

Šiuo metu pastebimas žymus KMP panaudojimo mokyklose augimas. Jo svarba ir teigiama nauda tampa vis aiškesnė: panaudojant animaciją ir garsus vaizdžiau perteikiama dėstoma medžiaga, suteikiamas interaktyvumas. Tai sustiprina mokinių motyvaciją mokytis, pagerina žinių įsisavinimą. Švietimo ir mokslo ministerijos informacinių technologijų centro tyrimo [2] duomenimis, iš šiuo metu mokyklose naudojamų KMP daugiausiai jų skirta 7-10 klasėms, mažiausiai – pradinėms (1 pav.). Natūralu, kad mokomosios medžiagos pateikimas IKT priemonėmis pradinukams yra sudėtingesnis, nei aukštesnių klasių mokiniams. Mažesnis KMP, skirtų 11-12 klasėms, skaičius aiškinamas tuo, kad mokytojai, ruošdami mokinius egzaminams, yra mažiau linkę eksperimentuoti su naujomis technologijomis ir prioritetą skiria tradiciniams mokymo metodams.



1 pav. KMP panaudojimo pasiskirstymas pagal klases

Remiantis minėtojo įsakymo [1] duomenimis, 97 proc. mokyklų naudoja kompiuterius, 95 proc. mokyklų turi interneto prieigą. Mokyklose daugėja iniciatyvių ir motyvuotų mokytojų, kurie yra įgiję kompiuterinį raštingumą ir naudoja IKT ugdymo procese. Didėja jaunesniosios kartos kompiuterinis raštingumas ir noras naudotis įvairiomis technologijomis, tik 1 proc. mokinių visiškai neturi IKT taikymo įgūdžių (ES vidurkis 4 proc.). Tai, kad kompiuteriu ir internetu daugiausiai naudojasi jaunimas, galima išvelgti ir statistiniuose duomenyse [3]. Kompiuterių naudojimas namų ūkiuose taip pat yra didelis. Statistikos departamento duomenimis [4] 62,1 proc. šalies gyventojų naudojami kompiuteriu, 60,5 proc. internetu. Namų ūkių apsirūpinimas kompiuteriais: 53,8 proc., internetu: 54,9 proc. Jaunimo tarpe, žinoma, kompiuterio pagrindinė paskirtis yra pramogos, tačiau jis sėkmingai naudojamas ir mokslams. Atlikta mokinių, jų tėvų, pedagogų ir ekspertų apklausa [5] atskleidė, kad namų darbų ruošimui, mokinių ir mokytojų nuomone, mokiniai dažnai naudoja asmeninį kompiuterį. Apie 87 proc. mokinių ir apie 92 proc. mokytojų teigia, kad mokiniai asmeninį kompiuterį namų darbų ruošimui naudoja bent jau kartais (taip mano 98 proc. mokyklų vadovų ir 87 proc. tėvų), tačiau retai naudojami mokomosiomis kompiuterinėmis programomis. Tyrimo rezultatuose akcentuojama tokia IKT nauda:

- galimybė sužinoti daugiau informacijos, atlikti daugiau bandymų tikslių mokslų pamokose;
- galimybė greitai susirasti papildomos informacijos;
- lengviau įsiminti pateikiamą informaciją;

- lengviau suprasti naujai pateikiamą medžiagą;
- pamokos tampa įdomesnės, vaizdingesnės;
- taikant IKT gerėja įgūdžiai dirbti kompiuteriu bei įvairiomis programomis.

Mokytojai pažymi, kad tokiose pamokose didėja mokinių susidomėjimas, motyvacija mokytis. IKT pagalba vaizdinę medžiagą galima pateikti koncentruotai, kompaktiškai, ugdomas mokinių kritinis mąstymas ir kūrybiškumas. Mokytojams naudojant IKT taip pat yra lengviau ir greičiau pasiruošti pamokoms. IKT leidžia nesunkiai įgyvendinti mokinių ugdymo individualizavimą, atsižvelgiant į skirtingus poreikius, kai tuo tarpu tradiciniame ugdyme tai reikalauja žymiai daugiau mokytojo pastangų. IKT taip pat skatina mokinių savarankiškumą mąstant, atliekant užduotis. Būtent šie veiksniai ir yra minimi kaip vieni svarbiausių pradinio ir pagrindinio ugdymo bendrosios programos atnaujinimo kryptių [6].

Nepaisant IKT pripažinimo ir naudojimo mokyklose ugdymo procese, egzistuoja įvairūs trūkumai, trukdantys pilnai išnaudoti IKT galimybes. Atlikus IKT diegimo į bendrąjį lavinimą ir profesinį mokymą SSGG analizę [1], įvardintos tokios silpnybės:

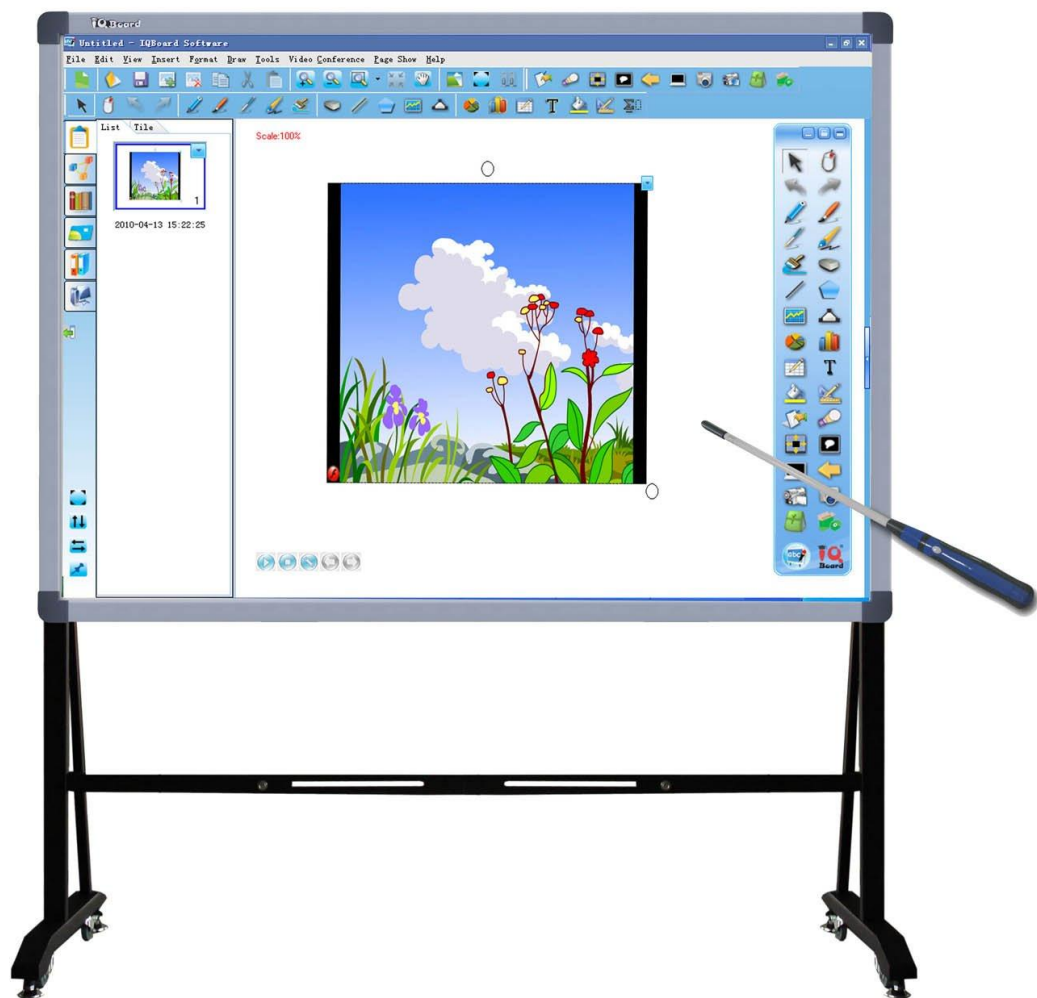
- Trūksta mokiniams prieinamos lietuviškos skaitmeninės mokomosios medžiagos. Pastebima, kad 38 proc. mokyklose kylančių problemų yra susijusių su IKT taikymui tinkamu mokymo turiniu, 34 proc. mokytojų teigia, kad jų turima skaitmeninė medžiaga yra labai blogos kokybės.
- IKT pernelyg menkai naudojamos pamokų metu mokinių mokymuisi individualizuoti, kuriant kokybiškesnį diskursą tarp mokinių ir pedagogų. 84 proc. mokinių norėtų, kad mokytojai dažniau naudotų IKT pamokų metu.

Panašiai mano ir ekspertai-konsultantai, atliktoje apklausoje [5] sutikdami, kad mokant vis dar nepanaudojamas darbas virtualioje erdvėje, grupinio darbo galimybės, individuali mokinio veikla prie kompiuterio. Mokinių ugdymo individualizavimas suprantamas kaip būtinybė ir yra stipriai akcentuojamas naujose KMP, tuo tarpu komandinio darbo galimybės kolkas lieka neišnaudotos. Tą iš dalies įtakoja techninės kliūtys: KMP individualizavimas įgyvendinamas gana paprastai – esami mokomieji objektai (MO) gali būti tiesiog praplečiami nauju funkcionalumu, tuo tarpu komandiniam darbui reikalinga atskira platforma, papildomi techniniai pajėgumai.

Nors grupinis darbas laikomas viena daugiausiai žadančių inovacijų ugdymo procese [7], tačiau šiuo metu Lietuvos mokyklose jis praktiškai neišnaudojamas. Viena iš IKT priemonių, leidžiančių įgyvendinti grupinį darbą, yra interaktyvi lenta (2 pav.). Nemaža dalis mokyklų aprūpintos šia nauja technologija, tačiau, remiantis apklausos [5] duomenimis, ja naudojasi labai retai. Pagrindinės tai lemiančios priežastys:

- reikalinga speciali programinė įranga;

- reikalingas pedagogų apmokymas;
- pritaikymo sritis labai siaura, ją riboja techninės savybės;
- mokiniai nėra identifikuojami – visi prisilietimai prie lentos suprantami vienodai, negalima išskirti atskirų mokinių indėlio į bendrą darbą;
- didelė kaina.
- pigesniuose modeliuose vaizdas kuriamas projektoriaus pagalba, tai sukelia papildomų nepatogumų vartotojui.



2 pav. Interaktyvi lenta

Akivaizdu, kad šis techninis sprendimas nėra sėkmingas. Šioje vietoje būtų galima panaudoti jau turimas IKT – asmeninius kompiuterius. Lyginant su interaktyvia lenta, toks variantas būtų visapusiškai pranašesnis:

- šiuo metu naudojami mokomieji portalai galėtų suteikti grupinio darbo galimybę, nereikalaujanti jokių papildomų veiksmų ar programinės įrangos iš vartotojų;

- nereikalingas pedagogų apmokymas, nes nereikėtų viską valdančio ir prižiūrinčio asmens;
- pritaikymo sritis žymiai platesnė – galimas didesnis vienu metu dirbančių mokinių skaičius, didesnė veiksmų laisvė (interaktyvi lenta paprastai reaguoja tik į prisilietimą pirštais, tuo tarpu kompiuteris gali turėti informacijos įvedimą klaviatūra, pele, internetine kamera ir t. t.);
- kiekvienas mokinys būtų identifikuojamas – aiškus indėlis į bendrą darbą, mokytojas galėtų juos įvertinti, stebėti pažangą ir t. t.;
- nereikalinga papildoma techninė įranga, tam būtų panaudojamos turimos IKT – asmeniniai kompiuteriai ir internetas, mokiniai galėtų prisijungti prie sistemos tiek mokyklose pamokų metu, tiek iš namų.

Čia, kalbant apie grupinį darbą, turima galvoje vienos užduoties atlikimą dalyvaujant iškart keliems mokiniams, jiems siekiant bendro tikslo, arba atvirkščiai – konkuruojant tarpusavyje. Toks interaktyvumo lygis, kai dalyviai mato vieni kitų veiksmus realiu laiku, vadinamas sinchroniniu. Šiam tipui taip pat priskiriamos momentinės žinutės, video pokalbiai. Tuo tarpu asinchroninis grupinis darbas (forumai, naujienų grupės, el. paštas) nėra susietas su laiku, nesuteikia tokio interaktyvumo. Sinchroninio grupinio darbo, kai dalyviai realiu laiku mato vieni kitų veiksmus, veikimo modelis iš esmės toks pat, kaip ir dabar populiarių daugelio vartotojų žaidimų (angl. *Massively Multiplayer Online, MMO*), su kuriais gerai susipažinusi jaunoji karta, tad neturėtų kilti adaptacinių problemų.

Šiuo metu Lietuvos mokyklos neturi KMP, veiksmingai įgyvendinančios interaktyvų mokinių komandinį darbą realiu laiku. Pasaulyje egzisuoja analogų, tiesa, dauguma jų yra tiesiog atskiri, nepriklausomi žaidimai su edukaciniu turiniu, o ne mokomosios programos, orientuotos į mokyklas.



### 3. REALIZAVIMO GALIMYBIŲ TYRIMAS

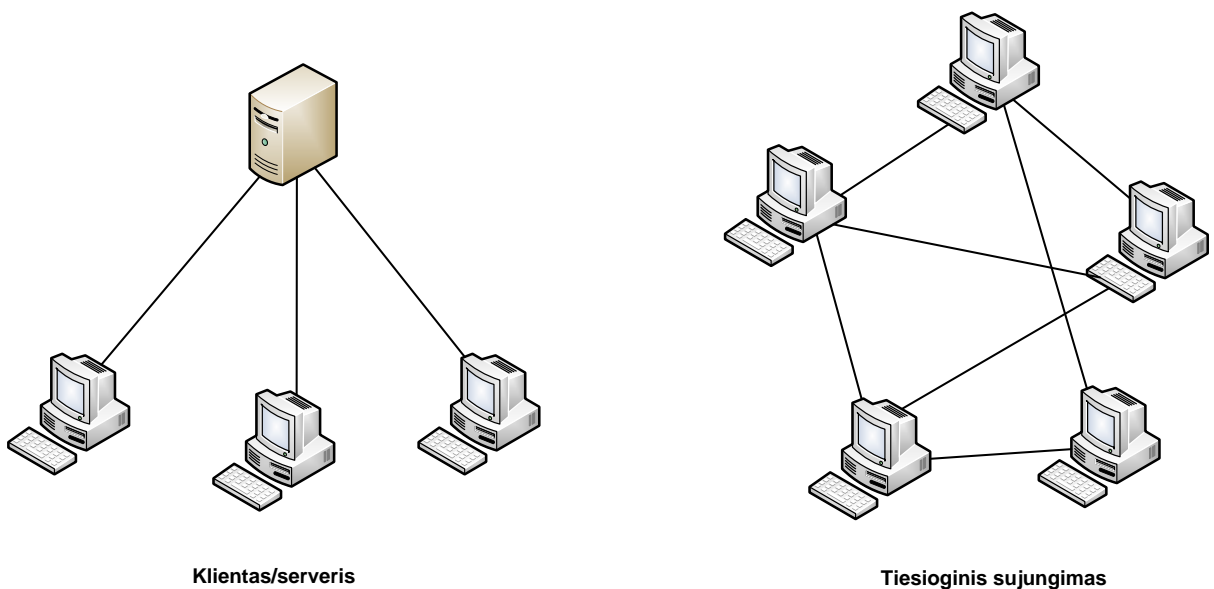
Techniniu požiūriu, norint įgyvendinti interaktyvų komandinį darbą realiu laiku, reikia realizuoti duomenų perdavimą tarp klientų. Vienam žaidėjui atlikus veiksmą, apie tai turi sužinoti visi kiti tą patį žaidimą žaidžiantys žaidėjai. Duomenys turi būti perduodami automatizuotai, t. y. nereikalaujant papildomų veiksmų iš kitų žaidėjų. Internetinės sveitainės šios funkcijos atlikti negali – jos tik statiškai perteikia informaciją, tad reikalinga atskira platforma, sujungianti vartotojus, pasirūpinanti duomenų perdavimu iš vieno kitam. Prieš projektuojant tokią platformą reikia išsiaiškinti galimus realizacijos būdus, išanalizuoti rinkoje esančių projektų privalumus ir trūkumus.

#### 3.1. Galimi architektūros tipai

Duomenų perdavimas tarp vartotojų, pagal sujungimo topologiją, skirstomas į dvi pagrindines architektūrinės grupes: kliento/serverio ir tiesioginio sujungimo (angl. *peer to peer*, *P2P*) [8] (3 pav.).

Kliento/serverio architektūroje vartotojai (klientai) jungiasi prie serverio, per kurį vyksta visas komunikavimas. Skaičiavimai gali būti atliekami tiek kliento, tiek serverio pusėje, pagal skaičiavimų santykį išskiriamos dvi kategorijos: ploni ir stori klientai (angl. *thin clients*, *fat clients*). Plonas klientas – tai toks, kuris vykdo labai mažai skaičiavimų. Jis paprastai tik surenka informaciją apie vartotojo atliktus veiksmus (mygtukų paspaudimai, valdymo komandos ir pan.) ir persiunčia ją serveriui, kur ir atliekamas visas darbas. Storame kliente atvirkščiai, didesnė dalis darbo tenka būtent klientui, o serveriui paliekami tik patys svarbiausi, reikalaujantys didesnio saugumo veiksmai.

Tiesioginio sujungimo architektūroje atskiro serverio nėra – sukuriama decentralizuota sistema, kurioje klientai jungiasi tarpusavyje, o dalis jų (arba visi) atlieka serverio vaidmenį. Toks sujungimas tinkamiausias tais atvejais, kai nėra būtinybės sujungti visų klientų iškart ir saugumo klausimas nėra kritinis. Tokia architektūra labai plačiai naudojama failų apsikeitimo sistemose: susijungimas sukuriamas tik su tais mazgais, su kuriais reikia apsikeisti duomenimis.



*3 pav. Kliento/serverio ir tiesioginio sujungimo modeliai*

### 3.2. Rinkoje esančių projektų apžvalga

Yra sukurta nemažai platformų, įgalinančių žaidimus išplėsti į daugelio žaidėjų modelį. Absoliuti dauguma tokių platformų paremtos kliento/serverio architektūra: žaidimų kūrėjams suteikiami įrankiai serverio dalies programavimui ir karkasas klientinei daliai, skirtas integravimui į žaidimą. Kuriant žaidimą, pritaikytą tokiai platformai, reikia sukurti serverinę dalį, kuri valdys žaidimo eigą ir pasirūpins duomenų perdavimu tarp žaidėjų, taip pat į žaidimą integruoti pateiktą karkasą, kuris palengvina komunikavimą su serveriu ir atlieka kitas specifines platformos funkcijas. Toliau apžvelgiamos kelios plačiausiai naudojamos tokios platformos.

#### **SmartFoxServer**

Viena populiariausių platformų, naudojama tokių kompanijų kaip



Zynga, Disney, Sony Online Entertainment ir kt. [9]. Serverinė dalis programuojama Java kalba, klientinės dalies karkasas realizuotas šioms platformoms: Flash, Unity3D, iOS, Java, Android. Platforma pasižymi dideliu duomenų srauto optimizavimu, taip pat turi galimybę TCP/UDP paketus tuneliuoti HTTP protokolu.

#### **PlayerIO**

Platforma orientuota į Flash žaidimus, taip pat



neseniai pridėtas ir Unity3D palaikymas [10]. Serverinė dalis programuojama C# (.Net) kalba. Plačiausiai naudojama nedideliuose, mažų studijų žaidimuose.

### **ElectroServer**

Plačiai naudojama platforma (tarp vartotojų – kompanijos Ubisoft, Disney, Pixar), palaikanti daugumą populiariausių klientinių sąsajų – Flash, Unity3D, Java, iOS, Android, JavaScript/HTML5, XNA [11]. Kaip ir SmartFoxServer, turi įvairių papildomų galimybių: HTTP tuneliavimą, aukšto laipsnio duomenų suspaudimą, automatizuotą žaidimo sesijų valdymą ir t. t. Viena iš įdomesnių savybių yra galimybė serverio pusėje naudoti kelias programavimo kalbas, galima rinktis iš Java, JavaScript ir ActionScript. Platforma orientuota į didesnius žaidimus, turinčius didelį žaidėjų kiekį vienu metu. Maksimalus palaikomas žaidėjų kiekis yra virš 300 000.



### **Union Platform**

Gana paprasta platforma, serverinei daliai naudojanti Java (ir kitas JVM aplinkoje veikiančias kalbas), klientinei – Flash ir JavaScript [12]. Klientinė dalis gali būti nesunkiai pritaikoma kitoms programavimo kalboms. Platforma tarp žaidimų kūrėjų nėra populiari, tačiau plačiai naudojama sporto varžybų statistikai realiu laiku atvaizduoti, pokalbių programoms ir pan.



### **Photon Server**

Platforma orientuota į aukštos klasės, daug žaidėjų turinčius žaidimus (MMO, FPS), yra naudojama tokių žaidimų kūrimo studijų kaip Codemasters, Acronym Games ir kt. [13]. Serverinė dalis programuojama C# kalba, o klientinė, be standartinių Flash, Unity3D, iOS, Android ir HTML, palaiko dar ir .NET, Mono ir Windows Phone.



1 lentelė. Egzistuojančių panašių platformų palyginimas

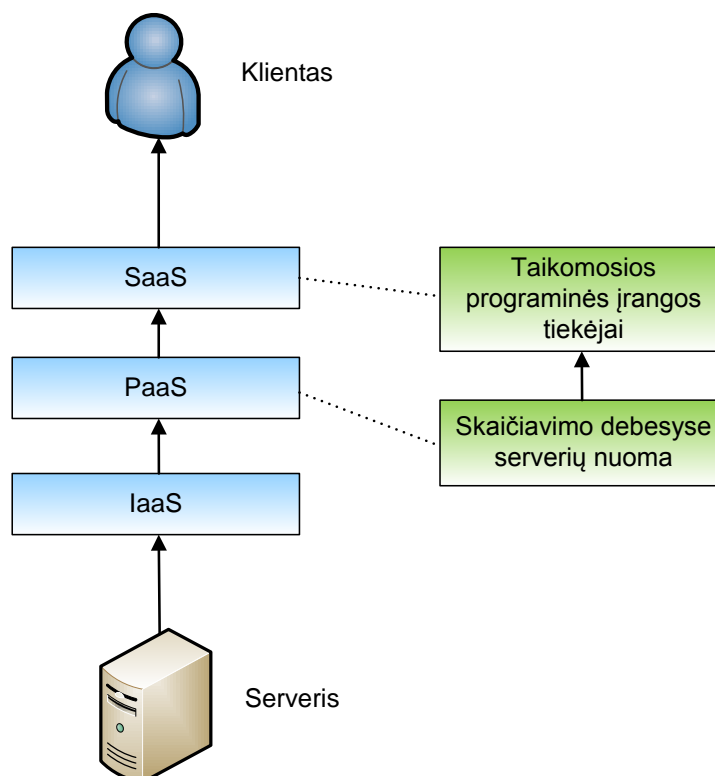
| Platformos pavadinimas | Paleidimo data | Serverinės dalies programavimo kalba | Klientinės dalies programavimo kalba                       | UDP palaikymas | Maksimalus vartotojų skaičius nemokamoje versijoje |
|------------------------|----------------|--------------------------------------|--|----------------|--|
| SmartFoxServer         | 2004           | Java                                 | Flash, Unity3D, iOS, Java, Android                         | +              | -  |
| PlayerIO               | 2004           | C#                                   | Flash, Unity3D, C#   | -              | Nenurodyta   |
| ElectroServer          | 2002           | Java, JavaScript, ActionScript       | Flash, Unity3D, Java, iOS, Android, JavaScript/HTML5, XNA  | +              | 50   |
| Union Platform         | 2001           | JVM (Java, JavaScript, Ruby, Python) | Flash, JavaScript/HTML5                                    | -              | 1000   |
| Photon Server          | 2003           | C#                                   | Flash, Unity3D, iOS, Android, JavaScript/HTML5, .NET, Mono | +              | 100  |

Verta pabrėžti, kad visos iš apžvelgtų platformų yra pritaikytos veikti naudojant skaičiavimą debesyse (angl. *Cloud Computing*). Pats skaičiavimas debesyse yra nauja IT šaka, kurios tikslas – teikti skaičiavimo išteklius kaip paslaugą nutolusiems vartotojams [14]. Tradicinis programinės įrangos priėjimo būdas – įdiegti ją į vartotojų kompiuterius, tuo tarpu skaičiavimo debesyse modelyje programinė įranga veikia nutolusiame serveryje, ją vartotojai pasiekia internetu, naudodami, pvz., interneto naršyklę. Galutinis vartotojas abiem atvejais gauna tą patį rezultatą ir nejaučia skirtumo, jam nebūtina žinoti, kaip veikia skaičiavimo debesyse modelis ir kaip keliauja duomenys. Skaičiavimo debesyse fizinis modelis skirstomas į tris sluoksnius [15] (4 pav.):

- Infrastruktūra (angl. *Infrastructure as a Service, IaaS*) – tai techninė įranga, suteikianti skaičiavimo, duomenų talpinimo ir perdavimo išteklius. Vartotojas turi pats pasirūpinti operacine sistema ir kita programine įranga.
- Platforma (angl. *Platform as a Service, PaaS*) – infrastruktūra kartu su specializuota operacine sistema ir programine įranga. Vartotojui paruošiama terpė, belieka tik įterpti savo taikomąją programinę įrangą, kuri turi būti pritaikyta platformai – naudoti jos

palaikomą API.

- Taikomoji programinė įranga (angl. *Software as a Service, SaaS*) – vartotojas turi priėjimą tik prie taikomosios programinės įrangos, gali ja naudotis nuomos principu.



4 pav. Skaičiavimo debesyse paslaugos sluoksniai

Visose iš apžvelgtųjų platformų, išskyrus Photon Server, skaičiavimo debesyse galimybė įgyvendinta kaip PaaS paslauga: išlaikomas tas pats tradicinis modelis, kuriame vartotojai gali programuoti specifinius serverio modulius, skaičiuojančius konkrečių žaidimų logiką. Tokio panaudojimo privalumas – vartotojui nebereikia rūpintis serverio technine dalimi. Photon Server šioje vietoje pasielgė kiek kitaip, sukūrė atskirą produktą Photon Cloud, kuris veikia vien tik skaičiavimo debesyse platformoje. Vartotojui neleidžiama keisti serverinės dalies, ją galima tik konfigūruoti, bet ne rašyti nuosavus skaičiavimo modulius. Visa logika ir skaičiavimai vyksta klientinėje programoje, serveris atlieka tik duomenų tarpininko vaidmenį. Toks modelis skaičiavimą debesyse paverčia jau nebe PaaS, o SaaS paslauga, o vartotoją dar labiau atitolina nuo tiesioginių darbų su serveriu.

### 3.3. Reikalavimai kuriamai platformai

Kuriamoji platforma turi atitikti tiek dabar esančius, tiek numanomas, galinčius atsirasti ateityje, poreikius. Ji turi būti universali ir atitikti tokius reikalavimus:

- Galimas pritaikymas įvairioms sritims, ne vien žaidimams.
- Portabilumas, veikimas įvairiose operacinėse sistemose (Windows, Linux).
- Lengvas plečiamumas – neapsiriboti vienu fiziniu serveriu.
- Klientiniai karkasai Flash, Unity3D, JavaScript ir kitoms platformoms.
- Užtikrinamas prieinamumas apribotiems vartotojams, pvz. esantiems už ugniasienės.
- Įvairios statistinės informacijos kaupimas ir patogi peržiūra.

Platforma orientuota į konkretų pritaikymą portale [www.mokinukai.lt](http://www.mokinukai.lt), iš to atsiranda daugiau papildomų sąlygų ir savybių, į kurias reikia atsižvelgti:

- Portale yra didelis kiekis mokomųjų žaidimų, tad serverio loginės dalies programavimas turi būti kaip įmanoma labiau automatizuotas.
- Žaidimai realizuoti Flash programavimo kalba.
- Žaidimai yra maži, žaidėjų grupės numatomos nuo 2 iki 10.
- Tą patį žaidimą vienu metu gali žaisti neribotas žaidėjų grupių skaičius.
- Žaidimų loginė dalis yra paprasta, nereikalaujanti labai greito duomenų perdavimo ar tikslumo laiko atžvilgiu.
- Žaidimai žaidėjui neneša jokios piniginės naudos ar žalos, todėl apsauga nuo apgaulingų veiksmų nėra prioritetinis dalykas.
- Portale turi būti matomas vykstančių žaidimų sesijų sąrašas.
- Dauguma žaidėjų yra iš Lietuvos teritorijos.
- Portale žaidėjai sudaro socialinį tinklą, tad turi būti galimybė matyti, ką žaidžia draugai, žaisti kartu su jais.
- Žaidimus galės žaisti tik registruoti portalo vartotojai, žaidimuose reikalingas žaidėjų autentifikavimas.

### 3.4. Platformos architektūros pasirinkimas

Kompiuteriniuose žaidimuose, įgyvendinant daugelio žaidėjų vienu metu (multiplayer) galimybę, beveik visais atvejais yra renkama kliento/serverio architektūra. Tokį pasirinkimą sąlygoja visų pirma pačių žaidimų struktūra: visi žaidėjai veikia viename bendrame pasaulyje. Kad pasaulio būseną būtų atvaizduojama vienodai visiems žaidėjams, ją turi apskaičiuoti ne kiekvienas žaidėjas individualiai, o nepriklausoma grandis – serveris. Pasikeitus pasaulio būsenai, serveris apie tai informuoja visus žaidėjus. Didžiausias tokios architektūros trūkumas tas, kad reikia programuoti serverinę žaidimo dalį, kur būtų apskaičiuojama dalis žaidimo logikos. Kai žaidimai maži ir jų daug, toks darbas pasidaro labai problematiškas, serveris apkraunamas daugybe modulių, skirtų kiekvienam konkrečiam žaidimui (tradicinis „vienas

serveris vienam žaidimui“ požiūris). Įgyvendinus tiesioginį klientų sujungimą (P2P), visa serverio logika persikeltų į klientus, o serverio, kaip atskiro fizinio vieneto, neliktų. Deja, nemaža dalis programavimo platformų, tarp jų ir Flash, neturi tiesioginio sujungimo galimybės, tad tokiu sprendimu būtų smarkiai apribojamas galimų žaidėjų skaičius.

Kuriamai platformai pasirinkta mišri architektūra, turinti ir kliento/serverio, ir tiesioginio sujungimo savybių, iš abiejų paimant reikalingas dalis. Žiūrint iš techninės pusės, tai – kliento/serverio modelis: žaidimai jungiasi prie serverio ir tiesiogiai vieni kitų nemato, visi duomenys tarp klientų keliauja per serverį. Tačiau visa žaidimo logika skaičiuojama kliento pusėje, serveris yra tik tarpininkas ir visiškai neįtakoja žaidimo eigos, tad, žiūrint iš kliento pusės, tai tarsi tiesioginis sujungimas. Tokiame modelyje išlieka vienas svarbiausių tiesioginio sujungimo trūkumų – palaikomas nedidelis į vieną grupę sujungtų žaidėjų skaičius, tačiau šiame taikyme, orientuotame į mažus žaidimus, šis trūkumas nėra aktualus.

Komercinėse platformose tokia mišri architektūra nėra plačiai taikoma, pagrinde dėl savo apribojimų. Tai yra jau minėtas nedidelis į vieną grupę įeinančių vartotojų skaičius, taip pat padidėjęs reikalingos perduoti informacijos kiekis, didesnė vartotojų sukčiavimo galimybė. Pavyzdžiui, minėtasis Photon Cloud peleistas tik prieš pusmetį, o per pirmuosius 4 mėnesius pritraukė 4500 žaidimų kūrėjų [16]. PlayerIO taip pat turi paruoštą specialų modulį, kuriuo sukuriamas skaidrus ryšys tarp klientų, t. y. serveris iš vieno kliento gautą žinutę persiunčia visiems kitiems jos nmodifikuodamas. Tačiau toks veikimas yra pernelyg primityvus, kad būtų plačiai naudojamas žaidimuose.

Apskritai tokia mišri architektūra yra gan nauja serverių vystymo kryptis ir šiuo metu yra daugiau idėjiniame lygmenyje. Švedijos Gioteburgo universitete buvo sukurtas panašios platformos prototipas [17]. Projekto tikslas – sukurti universalų žaidimų serverį, palaikantį kelis skirtingus žaidimus vienu metu. Tiesa, universalumas ten įgyvendintas kiek kita prasme: žaidimų serverinės dalies moduliai kuriami tuo pačiu tradiciniu principu, serveris tik valdo (sukuria/naikina) žaidimo sesijas. Kiekvienai žaidimo sesijai virtualioje mašinoje paleidžiamas žaidimo logikos modulis, tokiu būdu skirtingose sesijose gali veikti skirtingi žaidimai. Universalumo atžvilgiu tai yra tas pats, ką siūlo apžvelgtos komercinės platformos. Kitas projektas, kuriame siekiama sukurti universalų žaidimų serverį, yra žaidimų portalas skatgame.net. Projektas orientuotas į ėjimais grįstus (angl. *turn-based*) žaidimus (šachmatai, go ir pan.), kuriuose vienu metu žaidžia du žaidėjai ir veiksmus atlieka paeiliui, bet ne abu iškart [18]. Ši platforma, net neskaitant apribojimo žaidimų žanrui, taip pat nėra visiškai universali – norint įtraukti naują žaidimą, reikia nežymiai modifikuoti serverinę dalį, t. y. serveris, nors ir nedaug, bet yra susijęs su žaidimų logika.

Iš apžvelgtų pavyzdžių aišku, kad architektūrinio sprendimo pasirinkimas labai

priklauso nuo konkrečių poreikių. Galima platformos serverinę dalį visiškai atskirti nuo žaidimo logikos ir daryti universalią, pvz. kaip minėtas PlayerIO skaidraus kanalo modulis, arba apkrauti įvairiomis žaidimams reikalingomis funkcijomis. Problema ta, kad abu variantai turi trūkumų: pirmu atveju žaidimo sesijų valdymas turėtų būti įgyvendintas kliento pusėje, o tai yra labai problematiška ir nepraktiška; antru atveju serveris praranda universalumą.

Sprendžiant, kurį variantą naudoti šiame darbe, vėlgi pasirinktas aukso vidurys – iš abiejų paimtos geriausios savybės. Nuspręsta serverį visiškai atskirti nuo žaidimų logikos, kad, norint pridėti naują žaidimą, nereikėtų nei kiek modifikuoti serverio kodo. Tačiau, skirtingai nei PlayerIO sprendime, čia serveris užsiima ne vien tik žinučių persiuntimu tarp žaidėjų, bet ir žaidimų sesijų valdymu. Žaidimo sesijas valdant serverio pusėje, žinutes galima persiųsti tik žaidėjams, priklausantiems tai sesijai, kaip ir siuntėjas, o ne visiems prisijungusiems prie serverio, tokiu būdu sumažinant bendrą duomenų srautą. Žinoma, sesijų valdymas serverį šiek tiek susieja su žaidimais, pvz. žaidžiant šachmatus serveris turi neleisti prie vienos sesijos prisijungti daugiau kaip dviems žaidėjams, tačiau įvairūs tokie žaidimų parametrai gali būti saugomi duomenų bazėje ir serveris vis dar išlaikytų svarbiausią savybę – būtų universalus, t. y. jo nereikėtų kaskart perprogramuoti pridėdant naują žaidimą.

Iš kitos pusės, serverio logikos dalis niekur nedingsta, ji tiesiog perkeliama į kliento pusę. Kliento architektūra tampa sudėtingesnė, atsiranda naujas papildomas loginis sluoksnis.



## 4. PLATFORMOS PROJEKTAVIMAS

Platforma projektuojama „iš viršaus žemyn“ principu (angl. *top-down approach*), pradėdant stambiausiomis dalimis, baigiant smulkiausiomis. Toks metodas pasirinktas pagrįste dėl to, kad turimi pradiniai duomenys yra platformos reikalavimai, kurie būtent ir aprašo abstraktų visos platformos vaizdą, o smulkesnis suskaidymas yra ieškomoji dalis. Pirmame žingsnyje atliekamas komunikavimo protokolo pasirinkimas, taip sukuriama tarsi pagrindas tolimesniems darbams. Toliau, pagal „iš viršaus žemyn“ metodą, apibrėžiamas bendras karkasas, gilinamasi į jį sudarančius komponentus ir jų vidinę struktūrą.

### 4.1. Komunikavimo protokolo pasirinkimas

Platformos, sudarytos iš kelių tarpusavyje komunikuojančių komponentų, projektavime labai svarbus žingsnis yra komunikavimo protokolo pasirinkimas. Skirtingi protokolai turi skirtingus apribojimus ir galimybes, į kuriuos turi būti atsižvelgta projektuojant platformą, ir yra visos platformos pagrindas, todėl jų pasirinkimas turėtų būti pats pirmasis projektavimo žingsnis. Pvz. tai, ar duomenys perduodami sinchroniškai, ar asinchroniškai, gana smarkiai įtakoja vidinį platformos komponentų veikimą.

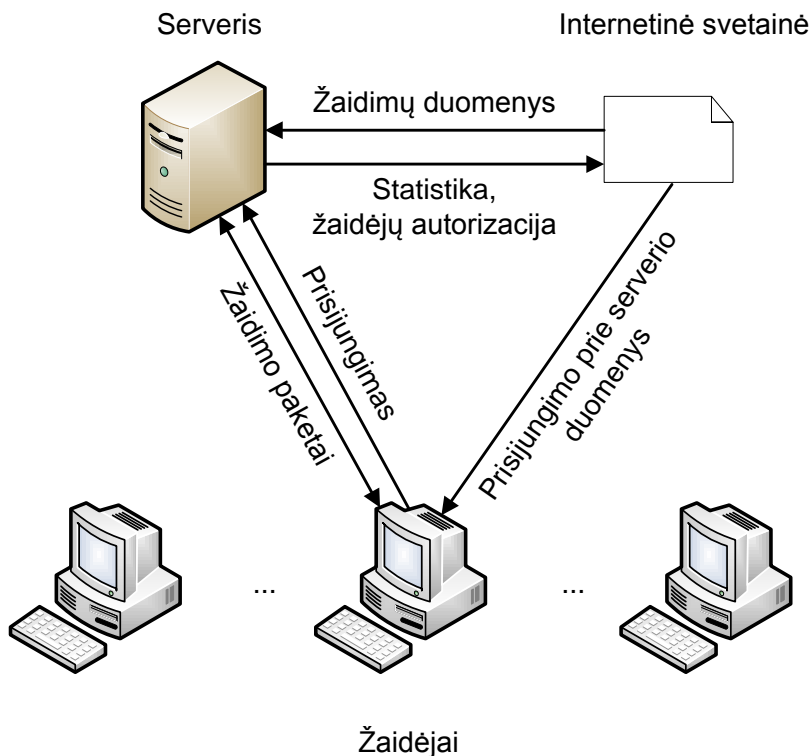
Internetinėse sistemose plačiausiai naudojami du OSI transportinio lygmens protokolai: TCP ir UDP. Pagrindinis jų skirtumas: TCP protokolas užtikrina, kad išsiųsti duomenys nukeliaus teisingu eiliškumu ir nesugadinti, tuo tarpu UDP to negarantuoja. Taip pat TCP paketai yra šiek tiek didesni, o komunikavimas kai kuriais atvejais lėtesnis. Verta paminėti, kad kompiuterių vartotojai neretai būna apriboti ugniasienės, praleidžiančios tik HTTP srautą. HTTP yra taikomojo lygmens protokolas, veikiantis TCP pagrindu. Ribojimas gali būti ir dar griežtesnis, leidžiantis prisijungimus tik prie 80-o prievado, kuris yra standartinis HTTP protokolui. Norint apeiti šį apribojimą, siunčiamus duomenis galima inkapsuliuoti į HTTP, toks veiksmas vadinamas HTTP tuneliavimu.

Iš atlikto panašių rinkoje esančių projektų palyginimo galima pastebėti, kad, pagal naudojamus duomenų perdavimo protokolus, jie sudaro dvi grupes: vieni apsiriboja vienu protokolu ir palaiko tik TCP, kiti stengiasi būti kuo universalesni ir palaiko TCP, UDP ir HTTP. Šis pasirinkimas lemia platformos universalumą, pvz. HTTP palaikymas būtinas, norint pasiekti žaidėjus, esančius už ugniasienės, taip pat HTTP yra pageidaujamas, norint turėti žaidimų karkasą JavaScript/HTML5 platformai. Kalbant apie UDP palaikymą, jis nėra būtinybė – platformos, palaikančios UDP, paprastai palaiko ir TCP, jis labiau orientuotas į sudėtingesnius, greitesnės sinchronizacijos reikalaujančius žaidimus, kuriuose stengiamasi išnaudoti kiekvieną įmanomą duomenų srauto ar keliavimo laiko sumažinimo būdą.

Projektuojamai platformai pasirinktas TCP protokolas su HTTP tuneliavimo galimybe. TCP protokolas tokio pobūdžio sistemose yra *de facto* pasirinkimas, pagrinde dėl to, kad yra plačiausiai palaikomas tarp žaidimo kūrimo platformų. Turint TCP, įgyvendinti HTTP palaikymą yra gana nesudėtinga, o toks žingsnis dar labiau padidina žaidėjų pasiekiamumą. Tuo tarpu UDP palaikymui reikalingų darbų ir gaunamos naudos santykis yra žymiai prastesnis. Taip pat projektuojama sistema orientuota į mažesnius žaidimus, nereikalaujančius ypatingai greito duomenų perdavimo ar tikslumo, tad UDP palaikymas, jei būtų įgyvendintas, greičiausiai taip ir liktų nepanaudotas.

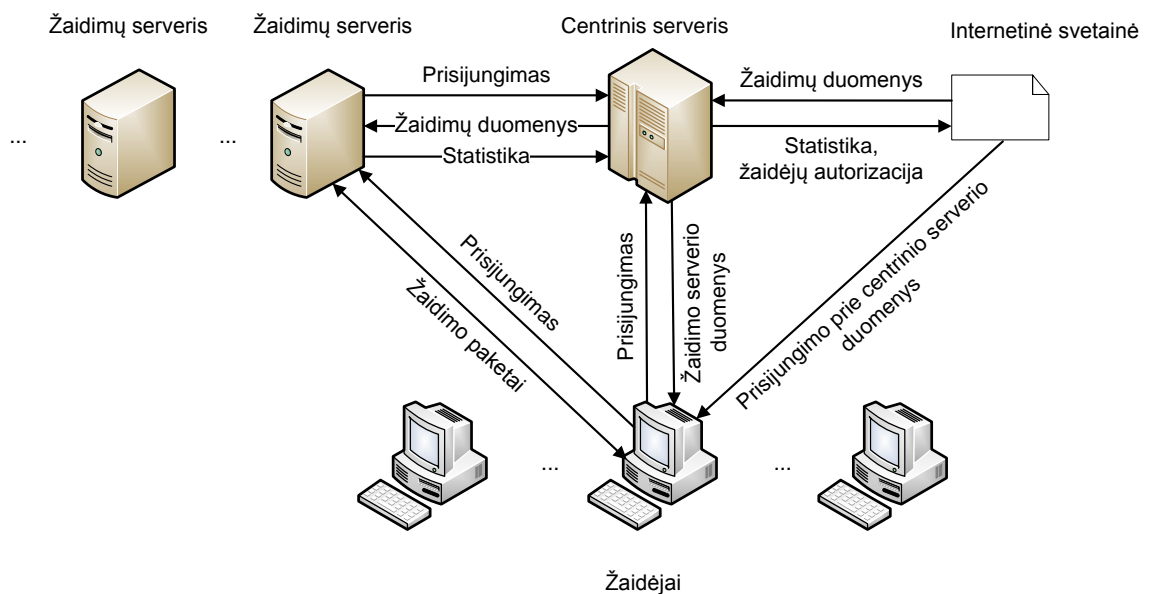
#### 4.2. Platformą sudarantys komponentai

Kokie komponentai sudarys platformą, lemia apsibrėžti reikalavimai. Iš karto galima pastebėti, kad tarp komponentų būtinai turi būti serveris, vartotojai (žaidimo karkasas) ir internetinė svetainė (5 pav.). Vartotojai prie serverio jungiasi TCP protokolu, prisijungimui jiems reikia žinoti serverio IP adresą ir prievado numerį (angl. *port*). Kadangi žaidimai pasiekiami per svetainę, prisijungimo prie serverio informacija taip pat pateikiama per ją. Prie serverio prisijungę žaidėjai turi būti autentifikuojami, tuo tikslu žaidimų serveris irgi susiejamas su svetaine, kurioje saugomi žaidėjų duomenys. Šis ryšys taip pat skirtas žaidimo rezultatams ir kitai statistikai išsaugoti, žaidimų duomenims gauti.



5 pav. Platformą sudarantys komponentai

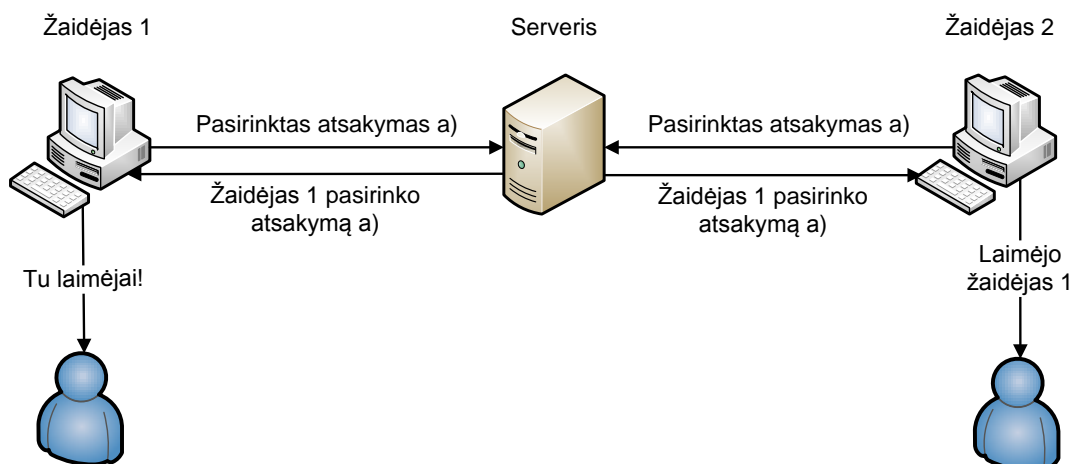
Ši apibrėžta struktūra yra minimali, tačiau visų numatytų reikalavimų netenkina. Norint turėti lengvą ir neribotą sistemos plečiamumą reikia tokios struktūros, kurią būtų galima plėsti pridėdant naujus mazgus ir nemodifikuojant pačios sistemos. Tai reiškia, kad turi būti palaikomas ne vienas žaidimų serveris, o neribotas jų skaičius. Taigi, serveris iš vienos esybės skyla į centrinį serverį ir žaidimų serverius. Centrinis serveris yra tik vienas, jo darbas yra žaidimų serverių valdymas ir žaidėjų į juos nukreipimas. Žaidimų serveriuose, kurių gali būti vienas ar daugiau, vyksta visas pagrindinis darbas, t. y. žaidimų eiga. Šioje patobulintoje struktūroje žaidėjai iš svetainės gauna prisijungimo prie centrinio serverio duomenis, o iš jo – informaciją apie vieną iš žaidimų serverių, tokiu būdu žaidėjų prisijungimo prie serverio procesas taip pat skyla į du etapus (6 pav.).



6 pav. Papildyta platformos struktūra

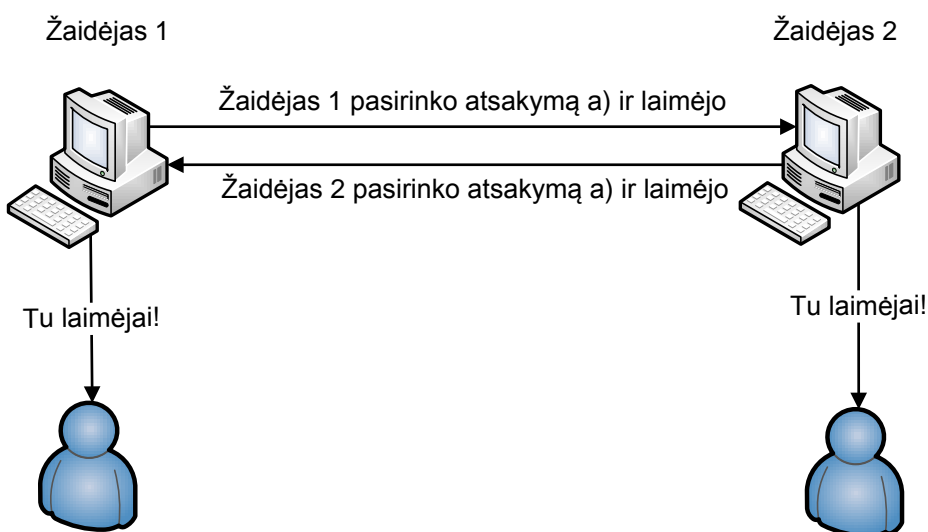
### 4.3. Klientų konkurencijos problema

Sistemoje, turinčioje kelis vienodo prioriteto veikėjus, neretai susidaro situacija, kai keli veikėjai vienu metu bando atlikti tą patį veiksmą. Pvz. testų programoje, kurioje žaidėjams užduodamas klausimas ir reikia pirmam pasirinkti teisingą atsakymo variantą, gali susidaryti tokia situacija, kai du žaidėjai pasirenka teisingą atsakymą tuo pat metu. Kliento/serverio modelyje tokia problema išsispredžia savaime: kadangi yra vienas serveris, valdantis žaidimo logiką, jis kažkurio vieno kliento žinutę priima pirmiau negu kito ir konkurencija paprasčiausiai nesusidaro (7 pav.).



7 pav. Kliento/serverio architektūroje konkurencija nesusidaro

Tačiau, esant tiesioginiam sujungimui, kai klientai patys atsako už save, tokia situacija nepageidaujama: kol duomenys keliauja tarp klientų, pačių klientų būseną gali pasikeisti ir nebesutapti (8 pav.).



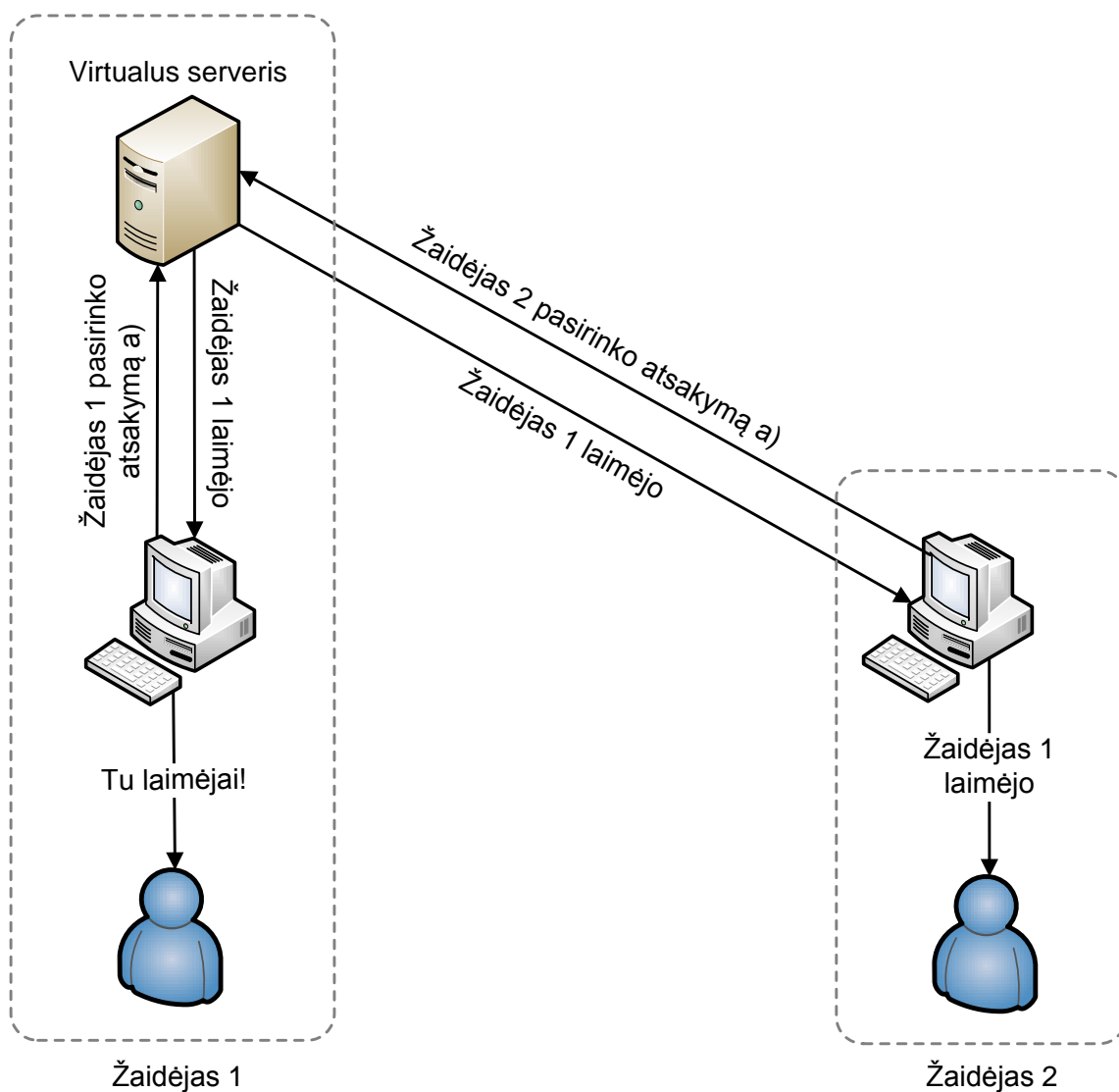
8 pav. Konkurencijos problema tiesioginio sujungimo modelyje

Klientų konkurencija įvardijama kaip viena didžiausių tiesioginio sujungimo modelio pritaikymo žaidimams kliūčių [19]. Tačiau tokios situacijos galima nesunkiai išvengti imituojant kliento/serverio modelį, tam tereikia įgyvendinti dvi sąlygas:

1. Vienas iš klientų turi papildomai atlikti ir serverio funkcijas.
2. Klientai savo atliktiems veiksams neturi suteikti pirmenybės prieš kitų klientų veiksmus.

Pirmoji sąlyga lemia kliento skilimą į dvi dalis, veikiančias nepriklausomai viena nuo kitos. Ši sąlyga labiau reikalinga norint patogiai valdyti žaidimo eigą, pvz. testų programos atveju serverio darbas būtų naujo klausimo parinkimas, atsakymo variantų atsitiktinės tvarkos generavimas, žaidimo užbaigimas. Tokie skaičiavimai gali būti atliekami tik griežtai vienoje vietoje, kitaip kiltų chaosas.

Antroji sąlyga įgyvendinama dar paprasčiau: klientas, atlikdamas veiksmą, pats nevykdo skaičiavimų (nereaguoja į būsenos pasikeitimą), o tik išsiunčia žinutę žaidimo serveriui. Skaičiavimai vykdomi tik gavus atsakymą iš serverio, t. y. atgal tą pačią žinutę, kuri buvo išsiųsta (9 pav.). Pvz. testo atveju, žaidėjui pasirinkus teisingą atsakymą, žaidimas apie tai žaidėjui praneša ne iškart, o tik gavęs žinutę iš serverio. Kliento reagavimas tik į iš serverio gautas komandas sukuria griežtą žinučių eiliškumą ir yra esminis žingsnis konkurencijai panaikinti.



9 pav. Konkurencijos panaikinimas tiesioginio sujungimo modelyje

#### 4.4. Centrinis serveris

Centrinis serveris yra tarpininkas tarp žaidėjų ir žaidimų serverių, taip pat tarp žaidimų serverių ir svetainės. Jis atlieka tokias funkcijas:

- Iš svetainės gauna visų galimų žaidimų informaciją (identifikaciniai numeriai, maksimalus žaidėjų skaičius).
- Prisijungus žaidimų serveriui, jam persiunčia informaciją apie svetainėje esančius žaidimus.
- Kaupia informaciją apie prisijungusius žaidimų serverius.
- Iš žaidimų serverių gautą statistiką persiunčia į svetainę (išsaugo duomenų bazėje). Į statistiką įeina žaidimų serverio užbaigtų žaidimo sesijų kiekis, vykdomų sesijų sąrašas, bendras apkrovimas procentais.
- Prisijungus naujam žaidėjui jam persiunčia vieno atrinkto žaidimų serverio prisijungimo duomenis.

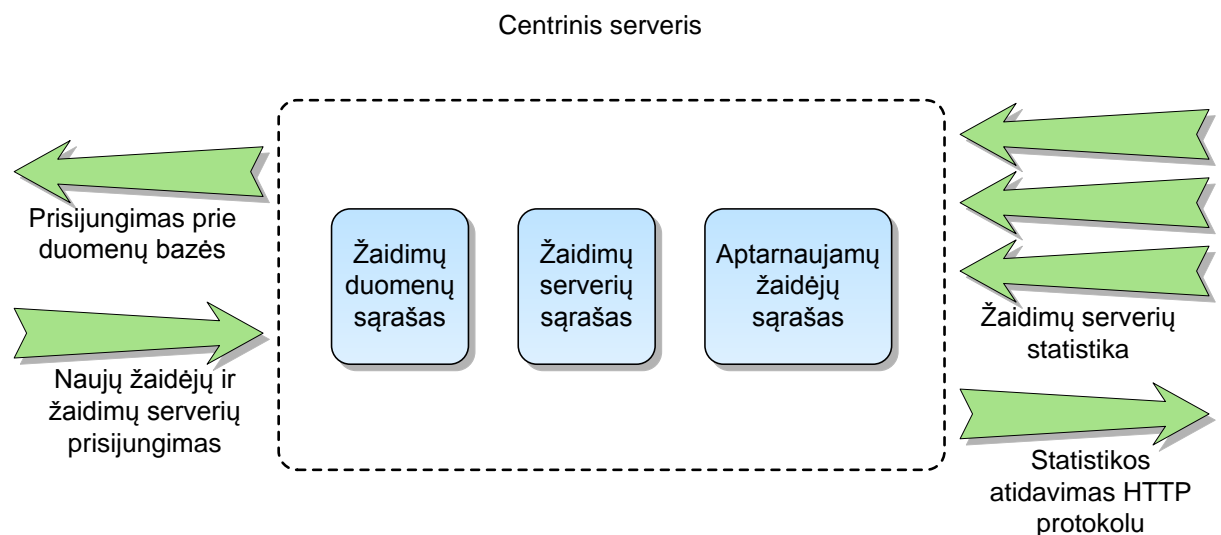
Kadangi žaidimų sesijų informacija turi būti ne vien tik įrašoma į duomenų bazę, bet ir realiu laiku atvaizduojama svetainėje, centrinis serveris privalo turėti galimybę statistiką pagal pareikalavimą perduoti HTTP protokolu. Tam reikalingas specialus servisas, priimančias HTTP užklausas ir generuojantis atsakymą į jas.

2 lentelė. Centrinio serverio reakcija į įvykius

| Išoriniai ir vidiniai įvykiai           | Reakcija į įvykius (atliekami veiksmai)  |
|---|--|
| Centrinio serverio paleidimas           | Pradedama fiksuoti žaidėjų ir žaidimų serverių prisijungimus;<br>Sukuriamas prisijungimas prie duomenų bazės |
| Prisijungta prie duomenų bazės          | Gaunamas žaidimų sąrašas   |
| Žaidimų serverio prisijungimas          | Žaidimų serveris įtraukiamas į sąrašą, jam išsiunčiamas žaidimų sąrašas                                      |
| Gauta nauja žaidimų serverio statistika | Statistika įrašoma į duomenų bazę  |
| Naujo žaidėjo prisijungimas             | Surandamas tinkamiausias žaidimų serveris, žaidėjui išsiunčiami prisijungimo prie jo duomenys                |
| Gauta HTTP užklausa statistikai gauti   | HTTP protokolu išsiunčiami statistikos duomenys  |
| Žaidimų serverio atsijungimas           | Žaidimų serveris pašalinamas iš sąrašo   |

Prisijungę žaidėjai centriniame serveryje nėra kaupiami, sudaroma tik laikina lentelė, kurioje įrašomi tuo metu aptarnaujami žaidėjai. Aptarnavus žaidėją, t. y. jam išsiuntus rekomenduojamo žaidimų serverio duomenis, jis iš laikinosios lentelės pašalinamas. Šis laikinas sąrašas reikalingas, nes, esant naujam prisijungimui iš išorės, nėra žinoma, ar prisijungė žaidėjas, ar žaidimų serveris. Tai sužinoma vėliau, atėjus pirmai prisijungusio kliento užklausiai.

Žaidėjams siūlomas žaidimų serveris parenkamas taip, kad būtų kuo greičiau užpildomos pradėtos žaidimų sesijos, tuo pačiu stengiantis išlyginti visų serverių apkrovimą. Esant situacijai, kai visi žaidimų serveriai pilni arba jų apskritai nėra, atitinkamai siunčiami pranešimai „serveriai pilni“ ar „nėra serverių“. Detali pranešimų struktūra aptariama tolimesnėse dalyse.



10 pav. Centrinio serverio struktūra

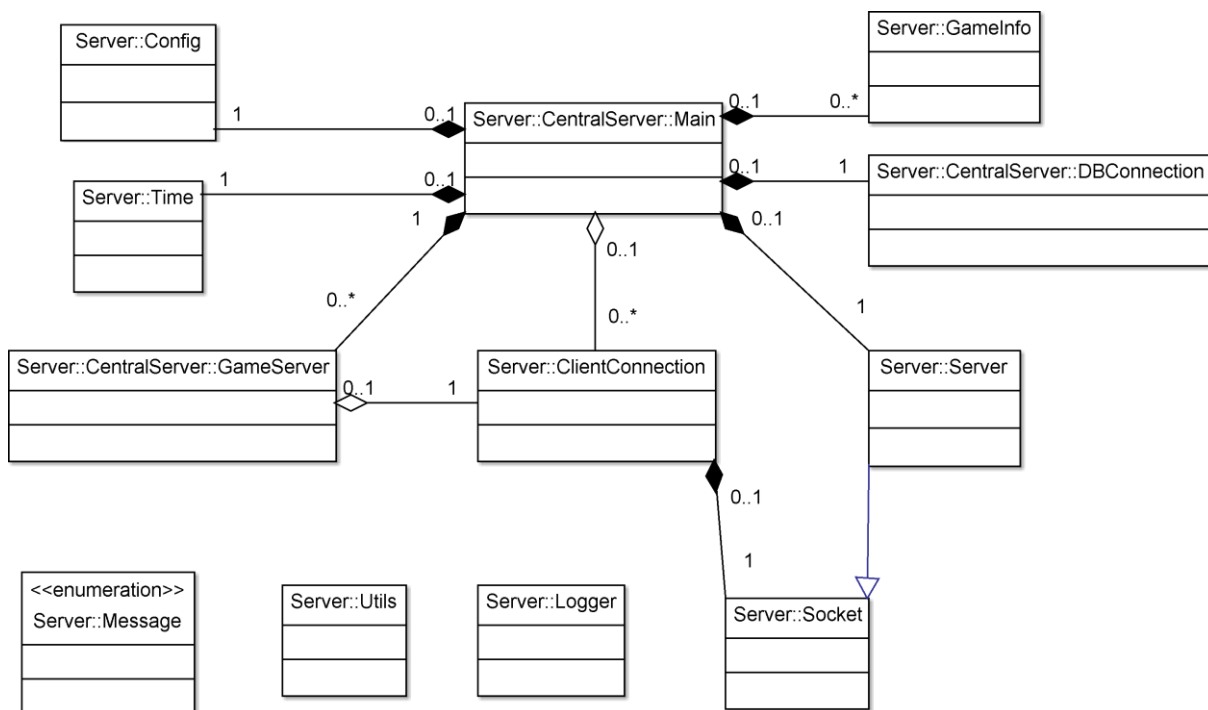
Žaidimų serveriai nėra griežtai surišti su centriniu serveriu ir gali bet kada prisijungti/atsijungti, neįtakoja vienas kito darbo. Centrinis serveris neriboja veikiančių žaidimų serverių kiekio, taip padėdamas išpildyti lengvo plečiamumo sąlygą. Plečiamumą riboja tiktai sistemos, kurioje veikia centrinis ir žaidimų serveriai, techninės galimybės (atminties kiekis, skaičiavimo pajėgumai, maksimalus leidžiamų įeinančių ir išėinančių prisijungimų skaičius, tinklo pralaidumas).

Kadangi centrinis ir žaidimų serveriai savo veikimu yra panašūs (techniniame lygyje), sudarant UML klasių diagramas dalis kodo apibrėžta kaip bendra abiem projektams, ji patalpinta Server vardų srityje. Atitinkamai specifinės centrinio serverio klasės priklauso `Server::CentralServer` vardų sričiai, žaidimų serverio – `Server::GameServer` (11 pav.). Bendrąją dalį sudaro tokios klasės (Server vardų sritis):

- Time: laiko skaičiavimas.
- Config: nustatymų failo skaitymas ir interpretavimas.
- Utils: įvairios sisteminės funkcijos.
- Logger: tekstinės informacijos išvedimas į ekraną (įspėjimai, pranešimai apie klaidas ir pan.).
- Socket: TCP prisijungimo sukūrimas, duomenų siuntimas ir gavimas.
- Server: įeinančių TCP prisijungimų fiksavimas.
- ClientConnection: siunčiamų paketų formavimas, gaunamų paketų patikrinimas ir išskaidymas.
- GameInfo: informacija apie žaidimą.

Specifinės centrinio serverio klasės (Server::CentralServer vardų sritis):

- Main: pagrindinė programos klasė.
- GameServer: žaidimo serverio prisijungimas; gaunami ir saugomi serverio statistiniai duomenys.
- DBConnection: prisijungimas prie duomenų bazės, užklausų formavimas ir gautų atsakymų interpretavimas.



11 pav. Centrinio serverio klasių diagrama

Šiame projektavimo etape nėra numatyta žaidejų autorizacija. Kadangi platforma skirta integravimui į esamas sistemas, autentifikavimas turi būti toks, koks naudojamas toje



platformoje, tad ir jo projektavimas turėtų vykti integracijos proceso metu. Dėl šios priežasties centrinis serveris neapibrėžia konkrečių autentifikavimo įgyvendinimo metodų, o tik užtikrina kanalą (prisijungimas prie duomenų bazės), kuriuo ji galėtų vykti.

#### 4.5. Žaidimų serveris

Žaidimų serveris yra pagrindinė platformos dalis, turinti didžiausią darbo krūvį. Svarbu, kad žaidimų serveris optimaliai išnaudotų jam skirtus techninius pajėgumus. Kaip buvo minėta, techninius sistemos pajėgumus aprašantys parametrai yra tokie: atminties kiekis, skaičiavimo pajėgumai, maksimalus leidžiamų įeinančių ir išėinančių prisijungimų skaičius, tinklo pralaidumas. Atminties kiekis ir maksimalus prisijungimų skaičius yra parametrai, kurių žaidimų serveris neįtakoja, tačiau, norint sėkmingai panaudoti sistemoje esančius kelis skaičiavimo vienetus (procesorius, procesoriaus branduolius) arba kelias tinklo sąsajas, reikalingas papildomas serverio darbas. Tam tikslui sistemoje paleidžiami keli žaidimų serverio egzemplioriai, jie gali veikti skirtinguose skaičiavimo vienetuose, naudoti skirtingas tinklo sąsajas. Kad būtų patogiau ir nereikėtų rankiniu būdu paleidinėti keleto žaidimų serverių, viskas vyksta automatiškai: sukuriama tiek žaidimų serverių procesų, kiek sistemoje yra skaičiavimo vienetų. Patys procesai veikia atskirai ir yra nepriklausomi vienas nuo kito.

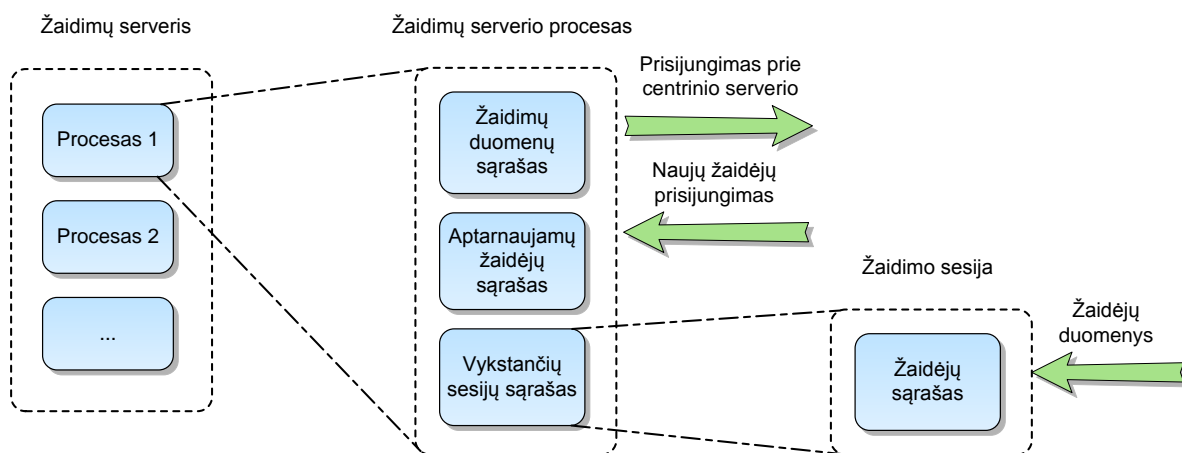
Pagrindinis žaidimų serverio darbas – duomenų perdavimas tarp klientų (žaidimų). Visi žaidėjai grupuojami į žaidimų sesijas, duomenų perdavimas atliekamas tik kiekvienos sesijos viduje. Žaidimo sesija – tai aplinka, kurioje vienas žaidimas vykdomas nuo pradžios iki pabaigos, pvz. šachmatų žaidime sesija atitinka vieną partiją. Kai kuriais atvejais sesija gali pasibaigti anksčiau, negu žaidimas, pvz. atsijungus visiems žaidėjams. Maksimalų leidžiamą žaidėjų skaičių sesijoje nusako maksimalus toje sesijoje žaidžiamo žaidimo žaidėjų skaičius. Prisijungę nauji žaidėjai iškart priskiriami serverio parinktai žaidimo sesijai. Kuriai sesijai priskirti naują žaidėją, pasirenkama taip:

- Ieškoma sesijos, kurios žaidimas sutaptų su žaidėjo pageidaujama ir ta sesija nebūtų užpildyta (turėtų mažiau žaidėjų, negu maksimaliai leidžiama norimam žaidimui). Jei tokia sesija randama, jai priskiriamas žaidėjas.
- Priešingu atveju sukuriama nauja sesija ir jai priskiriamas žaidėjas.

Esant tokiai tvarkai, kiekvienas žaidimas turi ne daugiau kaip vieną nepilną sesiją. Vykstančių sesijų skaičius nėra ribojamas, ribojamas tik maksimalus prie serverio prisijungusių žaidėjų skaičius.

| Išoriniai ir vidiniai įvykiai | Reakcija į įvykius (atliekami veiksmai)  |
|-------------------------------|--|
| Žaidimų serverio paleidimas   | Prisijungiama prie centrinio serverio.<br>Pradedama fiksuoti žaidėjų prisijungimus.          |
| Gautas žaidimų sąrašas        | Žaidimų sąrašas išsaugomas.  |
| Žaidėjo prisijungimas         | Žaidėjas priskiriamas tinkamiausiai žaidimo sesijai. Jei tokios nėra, sukuriama nauja.       |
| Iš žaidėjo gauti duomenys     | Duomenys persiunčiami kitiems (arba vienam, priklausomai nuo poreikio) tos sesijos žaidėjams |
| Baigtas sesijos darbas        | Sesija sunaikinama, pašalinama iš sąrašo.  |

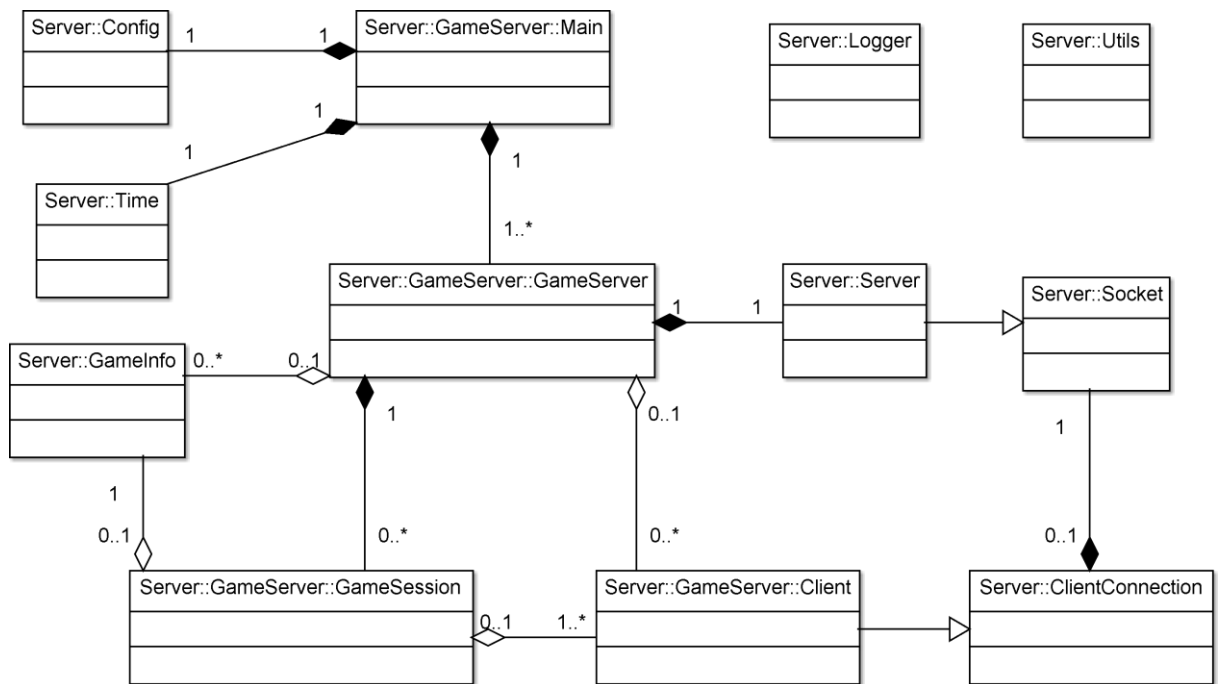
Žaidimų serverio vidinė struktūra suplanuota tokia, kad būtų visiškai atskirti tiek žaidimų serveriai tarpusavyje, tiek juos sudarantys procesai. Tokiu būdu užtikrinama, kad naujus žaidimų serverius į sistemą būtų galima įtraukti paprastai, nemodifikuojant pačios sistemos.



12 pav. Žaidimų serverio struktūra

Specifinės žaidimų serverio klasės (Server::GameServer vardų sritis):

- Client: prisijungęs žaidėjas; funkcijos paketų formavimui, siuntimui, gavimui ir apdorojimui.
- GameSession: žaidimo sesijos valdymas, klientų aptarnavimas (žinučių persiuntimas).
- GameServer: naujų žaidėjų prisijungimų fiksavimas, sesijų sukūrimas ir valdymas, žaidimų duomenų saugojimas.
- Main: reikalingo kiekio žaidimų serverių procesų sukūrimas ir valdymas.



13 pav. Žaidimų serverio klasių diagrama

#### 4.6. Žaidimo karkasas

Žaidimo karkaso tikslas – palengvinti žaidimų kūrimo darbą, realizuojant dažnai pasikartojančias, bendras keliems projektams funkcijas. Kai žaidimų daug ir jie visi atitinka tą patį šabloną, pasikartojančių dalių iškėlimas į karkasą pasitarnauja ne tik pagreitinant kūrimo procesą, bet taip pat ir klaidų taisymą: užtenka klaidą ištaisyti vienoje vietoje ir pataisa automatiškai įsigalios visuose karkasą naudojančiuose žaidimuose. Kita priežastis, kodėl žaidimų, skirtų kuriamai platformai, kūrėjams turėtų būti pateiktas karkasas – prisijungimas ir bendravimas su žaidimų serveriu yra specifinis, kelių pakopų procesas, į kurį įsigilinti pašaliniam programuotojui prireiktų nemažai pastangų, tad turi būti paruoštas jį gerai išmanančių asmenų. Karkaso atliekamos funkcijos:

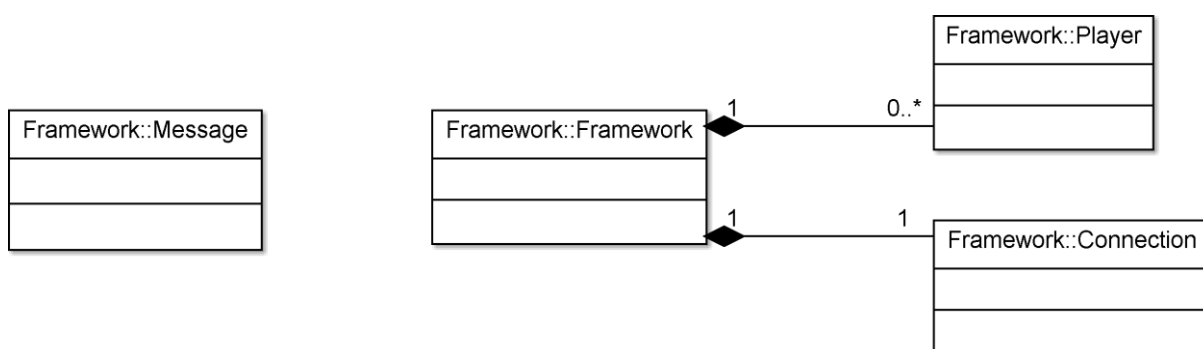
- Prisijungimas prie centrinio serverio, žaidimų serverio informacijos gavimas.
- Prisijungimas prie žaidimų serverio.
- Siunčiamų paketų formavimas.
- Gautų paketų interpretavimas, vartotojui (žaidimo programuotojui) pateikiant tik reikalingus duomenis.

Kitaip tariant, karkasas paslepia visą komunikaciją tarp žaidimo ir centrinio bei žaidimų serverių, programuotojui suteikia tik tą informaciją, kuri jam svarbi. Iš programuotojo pusės visas komunikavimas matomas kaip esant tiesioginiam žaidimų sujungimui: galima išsiųsti duomenis nurodytam žaidėjui (ar visiems iškart), taip pat gauti

duomenis iš kitų žaidėjų.

Karkasą sudarančios klasės (14 pav.):

- Message: valdymo komandų konstantos.
- Connection: prisijungimas prie centrinio/žaidimų serverio, paketų siuntimas, priėmimas.
- Player: žaidėjo duomenys.
- Framework: prisijungimo prie žaidimų serverio proceso vykdymas, paketų formavimas, sisteminių komandų (pvz. naujo žaidėjo prisijungimas) apdorojimas.



14 pav. Žaidimų karkaso klasių diagrama

#### 4.7. Žaidimų portalas

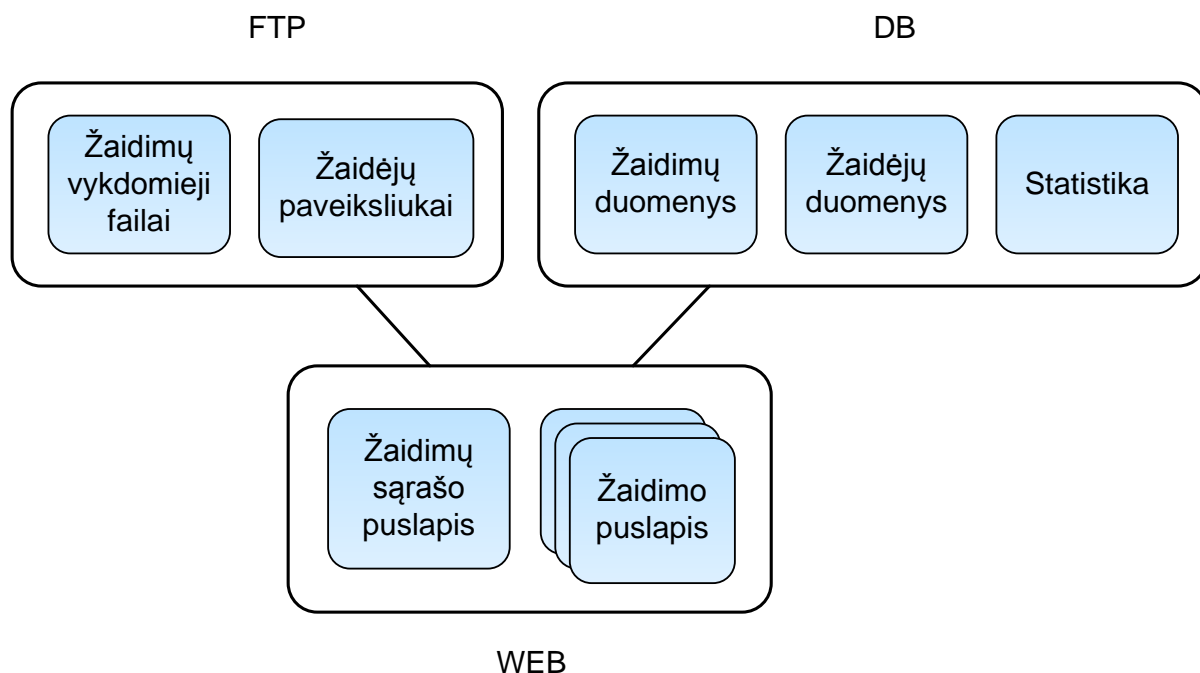
Žaidimų portalas, kaip ir žaidėjų autentifikavimas, neįeina į platformos projektavimą, tačiau, turint galvoje būsimą platformos ir portalo integraciją, būtina bent abstrakčiai absibrėžti galimą portalo struktūrą ir komunikavimo su platforma metodus.

Numatomos tokios portalo, į kurį bus integruojama platforma, savybės:

- Žaidėjai – registruoti vartotojai, kuriems, prisijungus prie žaidimo, reikalingas autentifikavimas.
- Kiekvienas žaidėjas turi unikalų vardą ir sau priskirtą paveiksluką, prieinamą HTTP protokolu.
- Informacija apie žaidėjus ir žaidimus, taip pat įvairi statistika, saugoma duomenų bazėje.
- Žaidėjai žaidimus renkasi iš sąrašo: prie kiekvieno žaidimo matomi naujausi vykstančios jo sesijos duomenys (prisijungusių žaidėjų vardai).
- Žaidimas pateikiamas WEB puslapyje, žaidime turi matytis prie jo prisijungusių žaidėjų paveikslukai.

Portalą sudarančios dalys galėtų būti tokios (15 pav.):

- FTP serveris: žaidimų vykdomųjų failų ir žaidėjų paveikslukų saugojimas.
- Duomenų bazė: saugomi žaidėjų profilių duomenys, žaidimų informacija, žaidėjų statistika.
- Internetinė svetainė: puslapiai žaidimų sąrašui pateikti, žaidimams žaisti.



15 pav. Portalo struktūra

Kaip buvo minėta, žaidimai prisijungimo prie žaidimų serverio informaciją gauna iš centrinio serverio, tad jiems tereikia žinoti, kaip prisijungti prie pastarojo. Šiuos duomenis žaidimui pateikia svetainė žaidimo puslapyje. Teoriškai žaidimai galėtų jungtis tiesiai prie žaidimų serverio, apeinant centrinį. Centrinio serverio būtų galima visiškai atsisakyti, tačiau toks veikimas būtų neracionalus:

- Žaidimų serverių informacija turėtų būti saugoma svetainėje. Žaidimų serveriai suprojektuoti taip, kad juos būtų galima bet kada įjungti/išjungti, tad svetainė ne visada galėtų tiksliai žinoti tikrąją kiekvieno serverio būseną.
- Kiekvienas žaidimų serveris turėtų apsiimti žaidėjų autentifikavimu, tad išaugtų prisijungimų prie duomenų bazės skaičius.
- Ženkliai padidėtų svetainei tenkantis darbo kiekis.
- Platformos integravimui tektų ne tik perdaryti žaidimus, panaudojant pateiktą karkasą, bet ir ženkliai modifikuoti svetainę.

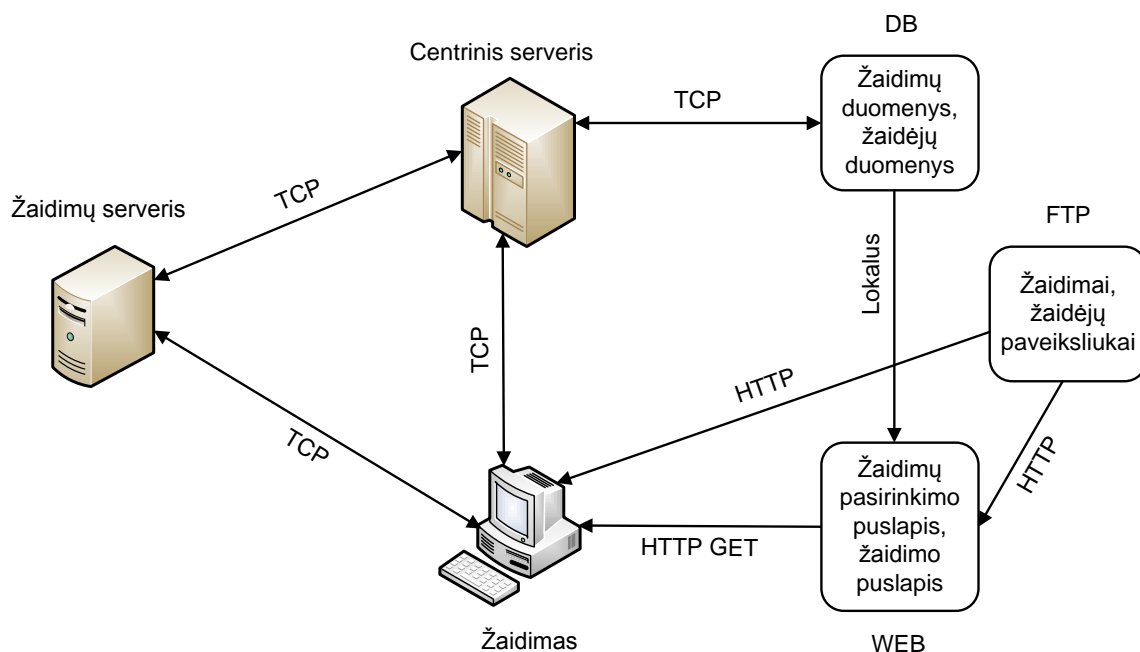
Žaidimai, be centrinio serverio prisijungimo informacijos, iš svetainės taip pat gauna

savo identifikacinį numerį, pagal kurį žaidimų serveryje susiejami su duomenų bazės informacija.

#### 4.8. Komunikavimo tarp komponentų detalus aprašymas

Žinant visus į platformą įeinančius komponentus ir jų vidinį veikimą, galima smulkiau detalizuoti komunikavimo tarp jų kanalus ir paketų struktūrą. Komponentų tarpusavio komunikavimo formatai yra tokie (16 pav.):

- Failai iš FTP pasiekiami HTTP pagalba. Svetainė iš FTP nuskaito žaidimus ir žaidėjų paveikslukus, o žaidimas – žaidėjų paveikslukus (jie naudojami į sesiją įeinančių žaidėjų atvaizdavimui).
- Svetainė prie duomenų bazės jungiasi lokaliu prisijungimu, o centrinis serveris – TCP. Centrinio serverio prisijungimą prie duomenų bazės galima įgyvendinti ir per svetainę, tuomet bendravimas vyktų HTTP užklausomis. Visos didžiausios duomenų bazių valdymo sistemos (MySQL, PostgreSQL ir pan.) turi tiesioginio prisijungimo iš C/C++ programų galimybę, tad jį naudoti tikslingiau, negu jungtis per tarpininką (svetainę). Žaidimų serveris prie duomenų bazės nesijungia, reikalingą žaidėjų informaciją, taip pat informaciją apie svetainėje esančius žaidimus gauna iš centrinio serverio.
- Žaidimas iš svetainės duomenis (prisijungimo prie centrinio serverio informaciją) gauna per žaidimų puslapį, tai atitinka HTTP GET užklausą. Taip pat papildomomis HTTP užklausomis gaunami kitų žaidėjų duomenys.
- Centrinis serveris, žaidimų serveris ir žaidimas tarpusavyje bendrauja TCP protokolu. Visam bendravimui tarp šių komponentų naudojamos vienodos struktūros žinutės. TCP žinutės papildomai gali būti įterptos į HTTP protokolo žinutes (naudojamas HTTP tuneliavimas).



16 pav. Komunikavimas tarp komponentų

Duomenų perdavimui tarp centrinio serverio, žaidimų serverio ir žaidimo pasirinkta vieninga struktūra. Paketus sudaro fiksuota antraštė ir kintamo dydžio duomenų dalis. Priklausomai nuo siunčiamos komandos, duomenų dalis gali būti ir tuščia. Į antraštę įeina tokie parametrai (17 pav.):

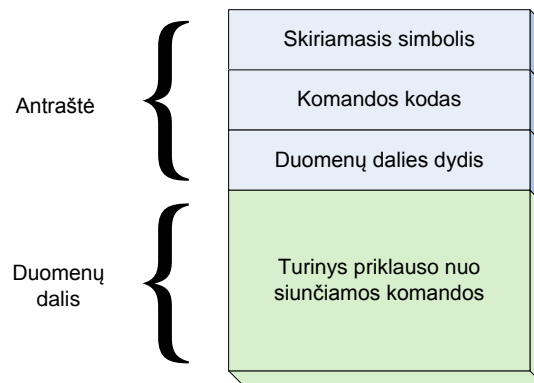
- Skiriamasis simbolis, reiškiantis paketo pradžią. Ši dalis yra vienintelė nekintanti ir nepriklausanti nuo siunčiamų duomenų.
- Vykdomos komandos kodas. Nuo šio komandos kodo priklauso, kokia informacija siunčiama duomenų dalyje.
- Duomenų dalies dydis. Skaičiuojamas tik pačios duomenų dalies dydis, neskaitant antraštės.

Tarp visų platformos komponentų išlaikant bendrą paketų perdavimo tvarką ir turint fiksuotą antraštę, tampa labai paprasta ją praplėsti. Tarkim, norint įgyvendinti UDP palaikymą, į paketus papildomai reiktų integruoti duomenų koherentiškumą leidžiančius patikrinti parametrus, pvz. kontrolines sumas, tam į antraštę būtų galima paprasčiausiai įterpti naują lauką, į kurį būtų įrašoma paketo CRC reikšmė.

Patys paketai, pagal paskirtį, gali būti skirti arba žaidimų serveriui, arba kitiems žaidėjams. Žaidimų serveriui skiriamos tokios komandos: žaidėjo prisijungimas, atsijungimas, žaidimo pradžia (nebeleisti prie sesijos prisijungti naujiems žaidėjams, nors sesija ir nepilna) ir kt. Skirtingai nuo kitiems žaidėjams skirtų komandų, į šias serveris turi reaguoti ir atlikti papildomus veiksmus: įtraukti/pašalinti žaidėją, pakeisti sesijos būseną ir pan. Paketą

perduodant kitiems žaidėjams papildomai nurodoma, ar jį perduoti visiems sesijos žaidėjams, ar tik kažkuriam vienam, taip sumažinant duomenų srautą ir bereikalingai nesiunčiant paketų tiems, kuriems iš jų nėra jokios naudos. Paketai, skirti serveriui, taip pat gali būti persiųsti kitiems žaidėjams, pvz. žaidėjui atsijungus, apie tai turi sužinoti ne tik serveris, bet ir visi kiti sesijos žaidėjai. Tik tokius paketus, skirtingai nuo skirtų persiųsti kitiems žaidėjams, serveris prieš persiuntimą dar papildomai modifikuoja.

Žaidimo pusėje paketų kodavimą/dekodavimą atlieka karkasas, žaidimo kūrėjui pateikiama tik paketo duomenų dalis. Tokiu būdu žaidimo kūrėjas gali naudoti savo specifines struktūras ir jas paprastai persiųsti kitiems, naudodamas komandas „siųsti visiems žaidėjams“ ir „siųsti vienam žaidėjui“. Jam nebūtina žinoti, kaip siunčiami paketai ir kokios yra palaikomos komandos, visu tuo pasirūpina žaidimo karkasas.



17 pav. Duomenų paketo struktūra



## 5. KONCEPCINĖS VERSIJOS REALIZACIJA

Pagal sudarytą projektą kuriama koncepcinė platformos versija. Jos tikslas – patikrinti, ar pasirinkti architektūriniai sprendimai veikia praktikoje, taip pat ištirti platformos veikimą realiomis sąlygomis. Koncepcinėje versijoje įgyvendinamos tik reikalingiausios platformos dalys, atsisakant tų, kurios nėra būtinos. Kadangi ši versija nebus integruojama į internetinį portalą, atmetamos tokios dalys kaip centrinio serverio prisijungimas prie duomenų bazės, žaidėjų autentifikavimas. Vietoj to turi būti leidžiama prisijungti visiems žaidėjams. Taip pat didelio platformos apkrovimo testavimui reikalinga papildoma programinė įranga, simuliuojanti didelį kiekį žaidėjų vienu metu. Simuliacija turi kuo tiksliau atkartoti realius žaidėjus, tam reikia suderinti prisijungimo prie žaidimo serverio/atsijungimo intervalus, iš vieno žaidimo siunčiamų duomenų srautą.

### 5.1. Programavimo aplinkos pasirinkimas

Renkantis, kokius įrankius naudoti kūrimo procese, reikia atkreipti dėmesį į platformai išskeltus reikalavimus, būtent portabilumą, plečiamumą ir greitą veikimą. Programavimui pasirinkta C++ kalba, kuri šiuo metu yra viena populiariausių. Remiantis statistiniais duomenimis [20], C++ pagal populiarumą užima trečią vietą (pirmosiose – C ir Java), o kai kuriose srityse, pvz. Google Code projektuose, yra ir dar aukščiau. Ši kalba pasižymi greitu vykdymu ir didelėmis galimybėmis, taip pat pritaikyta objektiniam programavimui ir yra plačiai naudojama ten, kur reikia žemo lygio priėjimo prie sisteminių resursų ir greito veikimo.

Pasirinkimas, kuriai operacinei sistemai taikyti kuriamą programinę įrangą, taip pat yra svarbus. Serverinėse sistemose pagrindė naudojamos dvi operacinių sistemų šeimos: Unix (64 proc.) ir Microsoft Windows (36 proc.) [21], tad nuspręsta kuriamą programinę įrangą pritaikyti veikti abejose šiose sistemose.

Norint įgyvendinti portabilumo sąlygą, tiek centrinio, tiek žaidimų serverio programavime stengiamasi naudoti kuo mažiau išorinių bibliotekų. Veiksmams, kurių atlikimui C++ neturi standartinių priemonių, naudojamos sisteminės bibliotekos:

- Laiko skaičiavimas. C++ standartinės priemonės laiko skaičiavimui veikia milisekundžių tikslumu, tad, norint gauti didesnę tikslumą, reikia naudoti specifines operacinės sistemos priemones. Šios priemonės yra skirtingos Windows ir Unix sistemose, skiriasi ir jų pasiekiamas tikslumas.
- Duomenų siuntimas tinklu. Darbas su TCP protokolu skirtingose operacinėse sistemose iš dalies sutampa, tačiau turi ir skirtumų ir kai kuriose vietose veikia

nevienodai. Programuojant reikia atsižvelgti į visus neatitikimus ir juos apeiti.

- Procesų sukūrimas ir valdymas. Unix šeimos sistemose procesų valdymui standartiškai naudojama POSIX pthreads biblioteka, Windows sistemoje – savos vidinės priemonės.

Naujų operacinių sistemų palaikymo pridėjimas reikalauja šių trijų nestandartinių dalių (laiko skaičiavimo, duomenų siuntimo ir procesų valdymo) pritaikymo. Kai kurios operacinės sistemos tarpusavyje turi bendrų bruožų, tad platformos pritaikymas joms būtų dar paprastesnis, pvz. Mac OS yra artima Unix šeimai ir palaiko nemažai bendrų bibliotekų. Tuo tarpu, norint platformą pritaikyti operacinei sistemai, neturinčiai C++ programų vykdymo galimybės (pvz. Android OS), reiktų perrašyti visą platformą.

## 5.2. Realizuotas komandų rinkinys

Kaip buvo minėta, duomenų paketą sudaro antraštė (skiriamasis simbolis, komandos kodas, duomenų dalies dydis) ir duomenų dalis. Konceptinėje versijoje realizuotos tokios komandos:

- MSG\_GAME\_SERVER\_START: žaidimų serveris centriniam serveriui praneša apie savo darbo pradžią. Siunčiami duomenys (paketo duomenų dalies turinys): prisijungimo prie žaidimų serverio informacija (IP adresas, prievado numeris), maksimalus leidžiamas žaidėjų skaičius.
- MSG\_GAMES\_INFO: centrinis serveris žaidimų serveriui siunčia svetainėje esančių žaidimų sąrašą (žaidimų identifikaciniai numeriai, maksimalus žaidėjų skaičius vienoje sesijoje).
- MSG\_GAME\_SERVER\_INFO: žaidimų serveris centriniam serveriui siunčia savo statistinę informaciją (žaidėjų skaičius, vykstančių sesijų informacija, bendras apkrovimas procentais).
- MSG\_GET\_SERVER\_DATA: žaidimas iš centrinio serverio prašo žaidimų serverio informacijos. Siunčiami duomenys: žaidėjo ir norimo žaidimo identifikaciniai numeriai.
- MSG\_SERVER\_DATA: centrinis serveris žaidimui siunčia parinkto žaidimų serverio duomenis. Siunčiami duomenys: žaidimų serverio būseną, IP adresas, prievado numeris. Būsena gali būti viena iš šių: „serveris surastas“, „visi serveriai pilni“, „nėra veikiančių serverių“.
- MSG\_LOGIN: žaidimas jungiasi prie žaidimų serverio. Siunčiami duomenys: žaidėjo ir norimo žaidimo identifikaciniai numeriai.
- MSG\_LOGIN\_OK: žaidimų serveris žaidėjui patvirtina apie sėkmingą

prisijungimą.

- MSG\_LOGIN\_FAILED: žaidimų serveris žaidėjui praneša apie nesėkmingą bandymą prisijungti. Papildomai siunčiamas klaidos aprašymas, kuris gali būti vienas iš šių: „toks žaidėjas jau prisijungęs“, „neatpažintas žaidimas“, „serveris pilnas“.
- MSG\_NEW\_PLAYER: žaidimų serveris žaidėjams praneša apie naują prisijungusį žaidėją. Siunčiami duomenys: naujo žaidėjo identifikacinis numeris.
- MSG\_DROP\_PLAYER: žaidimų serveris žaidėjams praneša apie atsijungusį žaidėją. Siunčiami duomenys: atsijungusio žaidėjo identifikacinis numeris.
- MSG\_START\_GAME : žaidėjas žaidimų serveriui praneša apie žaidimo pradžia. Po šio pranešimo serveris į sesiją nebepriima naujų žaidėjų, nepriklausomai ar yra likusių laisvų vietų.
- MSG\_PUBLIC: žaidėjas siunčia žinutę, skirtą persiuntimui kitiems sesijos žaidėjams. Pati žinutė sudaro visą paketo duomenų dalį.
- MSG\_PRIVATE: žaidėjas siunčia žinutę, skirtą persiuntimui vienam konkrečiam žaidėjui. Papildomi duomenys: gavėjo identifikacinis numeris ir pati žinutė.

Žaidimo karkasas realizuotas Flash platformai. Į žaidimo karkasą įeina visos suprojektuotos funkcijos, išskyrus kitų sesijos žaidėjų atvaizdavimą ir žaidimo rezultatų išsaugojimą, t. y. tas funkcijas, kurių realizavimui reikalinga integracija su žaidimų portalu.

## 6. PLATFORMOS VEIKIMO TYRIMAS

Platformos veikimas tirtas simuliuojant didelį kiekį (3000) vienu metu prisijungusių vartotojų. Kiekvienas vartotojas žaidime praleidžia skirtingą laiką, nuo 30 iki 90 sekundžių, ir tam tikru intervalu siunčia paketus žaidimų serveriui. Eksperimento metu keičiami tokie parametrai:

- Į žaidimų sesiją įeinančių žaidėjų maksimalus kiekis, reikšmės: 4 (imituojamas mažas žaidimas) ir 100 (imituojamas didelis žaidimas).
- Žaidėjų generuojamų paketų dydis, reikšmės: 10 ir 100 baitų.
- Paketų siuntimo intervalas, reikšmės: 5 sekundės (imituojamas paprastas žaidimas, pvz. testas su atsakymų variantais) ir 0,1 sekundės (imituojamas aukšto interaktyvumo lygio, greito veiksmo žaidimas, pvz. šaudyklė, automobilių lenktynės ar pan.).

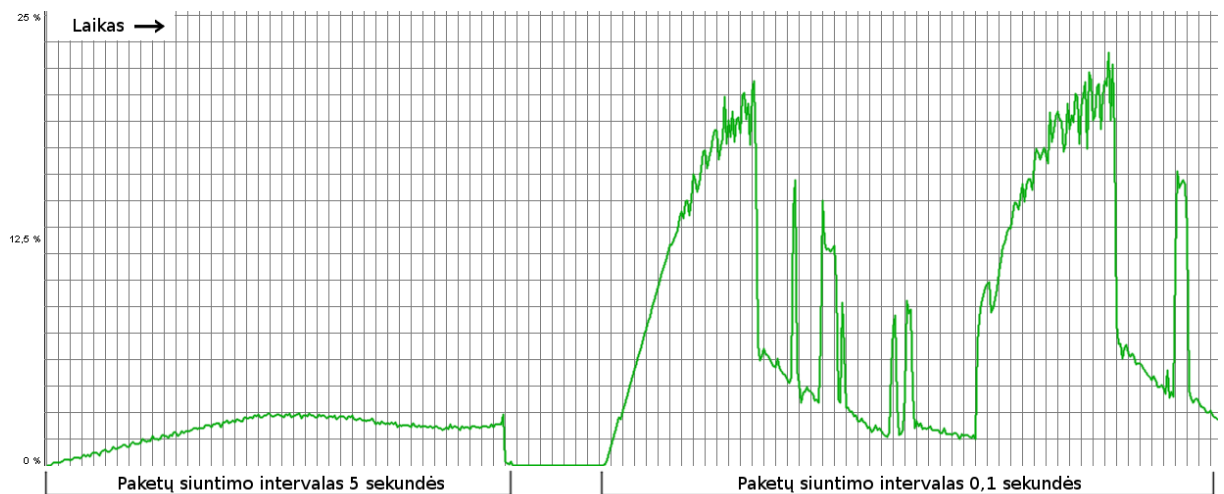
Tyrimas pakartotas 8 kartus, panaudojant visas kintamų reikšmių kombinacijas. Virtualūs žaidėjai prie žaidimų serverio jungiasi ne visi iškart, o palaipsniui, taip padedant išskirti tikrąją ribą, nuo kurios žaidimų serveris nebesusidoroja su informacijos srautu. Tyrimo metu stebėti šie sistemos, kuriame veikia žaidimų serveris, parametrai: tinklo apkrovimas (įeinantys ir išeinantys duomenys), procesoriaus apkrovimas, operatyviosios atminties sunaudojimas. Sistemos, kurioje vykdytas tyrimas, techniniai parametrai yra tokie: Intel Core 2 Duo 1,67 GHz procesorius (darbui naudotas vienas branduolys), 2 GB operatyviosios atminties, 100 Mbps tinklo sąsaja. Gauti rezultatai pateikti 4 lentelėje, tinklo apkrovimo grafikai laiko atžvilgiu – 18-21 pav.

4 lentelė. Platformos veikimo tyrimo rezultatai

| Virtualių žaidėjų skaičius | Žaidėjų skaičius sesijoje | Paketų dydis, baitais | Paketų siuntimo intervalas, s | Tinklo apkrovimas, Mbps | Procesoriaus apkrovimas, procentais | Atminties sunaudojimas, MB |
|----------------------------|---------------------------|-----------------------|-------------------------------|-------------------------|-------------------------------------|----------------------------|
| 3000                       | 4                         | 10                    | 5.0                           | 2.5                     | 22%                                 | 29                         |
| 2023                       | 4                         | 10                    | 0.1                           | 19                      | 100%                                | 22                         |
| 3000                       | 4                         | 100                   | 5.0                           | 4                       | 22%                                 | 29                         |
| 1194                       | 4                         | 100                   | 0.1                           | 56                      | 100%                                | 19                         |
| 1896                       | 100                       | 10                    | 5.0                           | 10                      | 68%                                 | 20.4                       |
| 387                        | 100                       | 10                    | 0.1                           | 18                      | 100%                                | 5.9                        |
| 2042                       | 100                       | 100                   | 5.0                           | 41                      | 100%                                | 22.3                       |
| 269                        | 100                       | 100                   | 0.1                           | 90                      | 100%                                | 6.9                        |

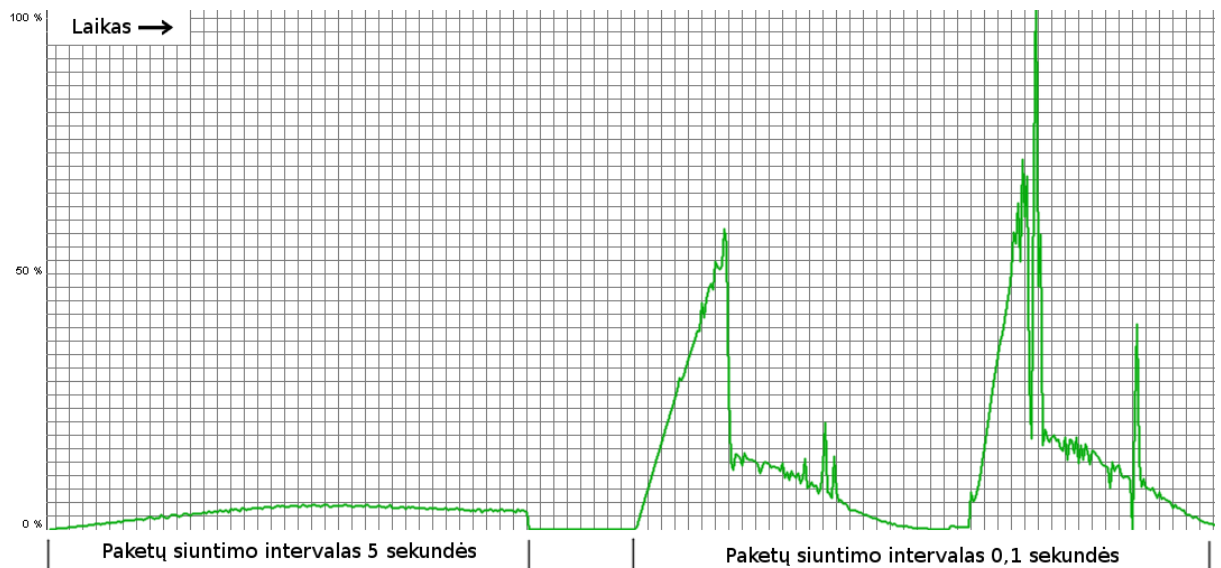
Situacija, kai sesijos dydis 4 žaidėjai, paketo dydis 10 baitų ir paketų siuntimo intervalas 5 sekundės, atitinka mažą žaidimą, kuriame žaidėjai veiksmus atlieka pakankamai retai (vidutiniškai kas 5 sekundes). Esant tokiems parametrų, žaidimų serveris susitvarko su visais 3000 žaidėjų (18 pav.): pasiekiamas 2,5 Mbps tinklo apkrovimas, 22% procesoriaus pajėgumų panaudojimas. Šioje realizacijoje maksimalus galimas siunčiamų paketų dydis yra 1024 baitai, tai lemia mažą operatyviosios atminties sunaudojimą. Atminties sunaudojimas, santykinai su kitų resursų sunaudojimu, yra labai mažas, tad šį parametru prirėikus galima drąsiai didinti.

Paketų siuntimo intervalu sumažinus iki 0,1 sekundės, o kitus parametrus išlaikant tuos pačius, platforma jau nebesusidoroja su duomenų srautu. Tinklo apkrovimas išauga iki 19 Mbps, tačiau, pasiekus 2023 žaidėjų ribą, procesorius nebespėja atlikti skaičiavimų, tad susidaro paketų eilė ir visas darbas sutrinka. Esant tokiai situacijai, nebepriimami nauji žaidėjų bandymai prisijungti, tad žaidėjų kiekis mažėja, kol galiausiai procesorius apdoroja visus susikaupusius paketus ir vėl pradėdamas sklandus darbas.



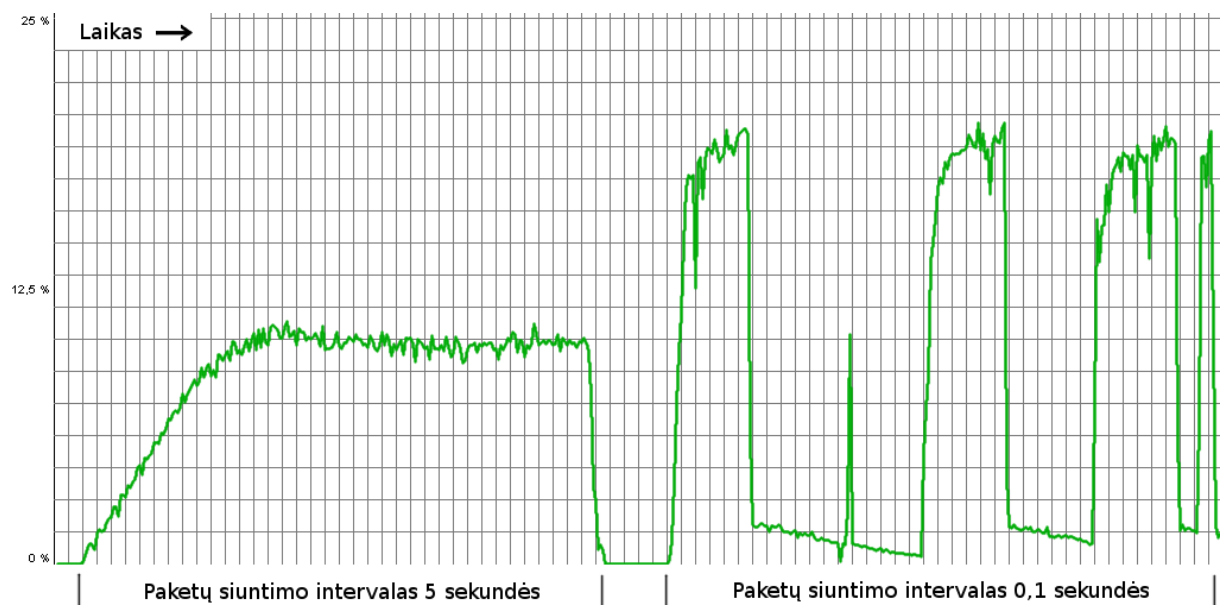
18 pav. Tinklo apkrovimas, kai sesijos dydis – 4 žaidėjai, paketo dydis – 10 baitų

Paketų dydį padidinus 10 kartų (iki 100 baitų) gaunama identiška situacija (19 pav.). Esant 5 sekundžių paketų siuntimo intervalui tinklo apkrovimas 4 Mbps, platforma susitvarko su visais 3000 žaidėjų. Paketų siuntimo intervalu sumažinus iki 0.1 sekundės, vėlgi pritrūksta procesoriaus pajėgumų, pasiekiamas 1194 prisijungusių žaidėjų skaičius.



19 pav. Tinklo apkrovimas, kai sesijos dydis – 4 žaidėjai, paketo dydis – 100 baitų

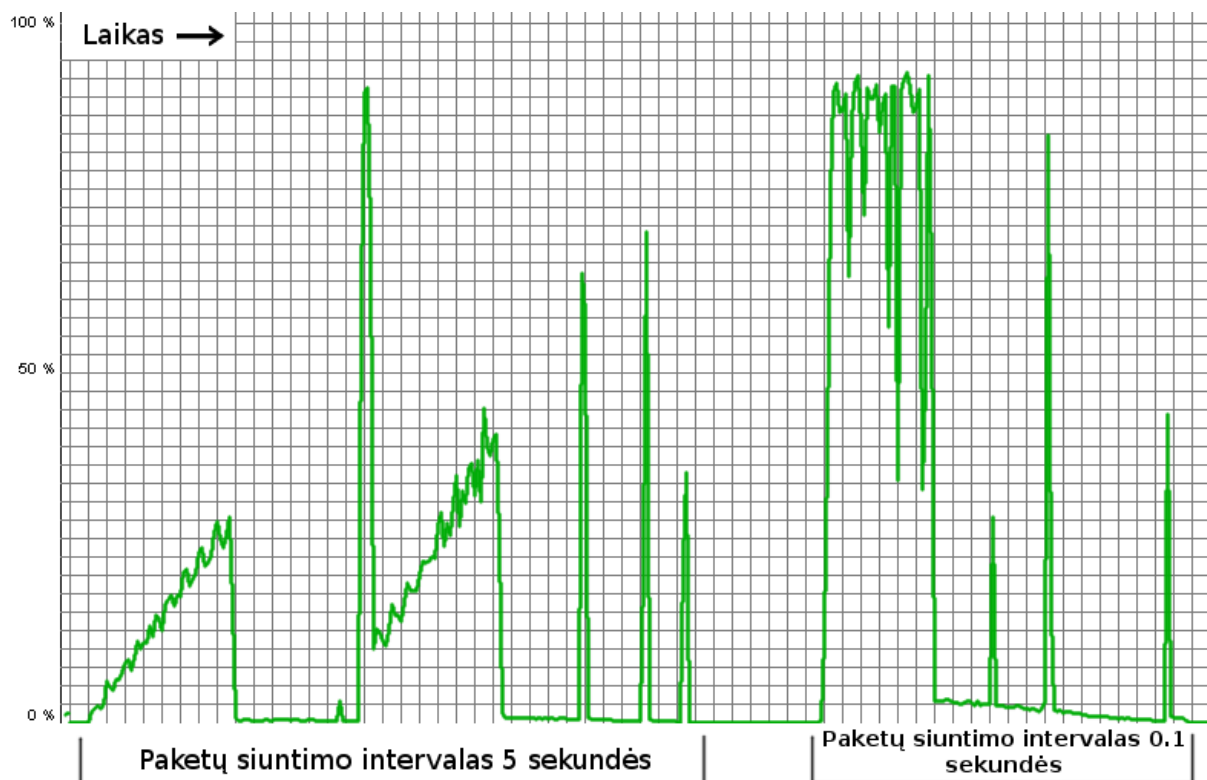
Žaidimų sesijos dydį padidinus iki 100, o paketų dydžiui esant 10 baitų, siuntimo intervalui – 5 sekundės, gaunama ribinė situacija, kai spėjama aptarnauti ne visus žaidėjus (pasiekta 1896 prisijungimų riba), tačiau paketų eilė dar nesusidaro tokia didelė, kad darbas sutriktų (20 pav.). Tuo tarpu paketų siuntimo intervalą sumažinus iki 0,1 sekundės, maksimalus žaidėjų skaičius yra tik 387.



20 pav. Tinklo apkrovimas, kai sesijos dydis – 100 žaidėjų, paketo dydis – 10 baitų

Sesijos dydžiui esant 100 o siuntimo intervalui 5 sekundės, paketų dydį padidinus iki 10 baitų jau peržengiama galimybių riba ir procesorius nebespėja apdoroti informacijos (21

pav.). Paketų siuntimo intervalą sumažinus iki 0,1 sekundės pasiekama ne tik procesoriaus, bet ir tinklo pralaidumo galimybių riba.



21 pav. Tinklo apkrovimas, kai sesijos dydis – 100 žaidėjų, paketo dydis – 100 baitų

Apibendrinant tyrimo rezultatus galima pastebėti, kad siunčiamų paketų dydis yra mažiausiai lemiantis parametras, tad, projektuojant žaidimus, tikslinga informaciją talpinti į mažesnę kiekį didelių paketų, negu daug mažų. Tinklo apkrovimas, paketų dydį padidinus 10 kartų, išaugo žymiai mažesniu santykiu. Taip yra dėl to, kad prie paketų papildomai pridedamos kitų OSI sluoksnių antraštės, taigi bendras apkrovos padidėjimas yra ne toks žymus.

Platforma puikiai susitvarko su situacija, kai sesijos dydis yra 4 žaidėjai, o paketų siuntimo intervalas 5 sekundės. Tokie parametrai būdingi mažiems žaidimams. Sesijos dydį padidinus iki 100 žaidėjų, platformai jau nebeužtenka procesoriaus pajėgumų. Paketų kiekis, didinant sesijos žaidėjų kiekį, didėja eksponentiškai, tad akivaizdu, kad universalus serveris tokiai situacijai netinka. Serveryje turi būti įgyvendintas žaidėjų grupavimas, kad, net ir esant neribojamam žaidėjų kiekiui vienoje sesijoje, duomenys būtų siunčiami ne visiems, o tik daliai jų.

## 7. IŠVADOS

- Darbe ištirtas KMP panaudojimas mokymo procese Lietuvos mokyklose. Pastebėta, kad šiuo metu mokyklų naudojamose KMP nėra įgyvendinta mokinių komandinio darbo galimybė, nors tai, remiantis švietimo ministerijos nustatyta ilgalaikė strategija ir atliktų tyrimų išvadamis, turėtų būti viena iš svarbiausių KMP vystymo krypčių.
- Suprojektuota platforma interaktyviam komandiniam darbui realiu laiku. Platformos veikimui pasirinktas mišrus kliento/serverio ir tiesioginio sujungimo modelis, kuris, nors ir yra naujas ir dar mažai taikomas, tačiau sparčiai populiarėja. Projektavimo metu išanalizuotos ir išspręstos įvairios architektūrinės problemos: greitas ir paprastas platformos plečiamumas, universalumas, klientų konkurencijos problema. Apsibrėžti platformos reikalavimai praktiniam pritaikymui mokymo procese, mokomajame portale <http://mokinukai.lt>.
- Pagal projektą sukurta koncepcinė platformos versija ir ištirtas jos veikimas. Ji labiausiai tinkama mažoms programoms, kurias sudaro nedidelis kiekis vartotojų, generuojamas nedidelis duomenų srautas. Programoms, turinčioms daug vartotojų vienu metu, arba kuriančioms ypač didelį duomenų srautą, tokia platforma netinka, joms turėtų būti naudojamas tradicinis kliento/serverio modelis.
- Platforma tinkama pritaikymui mokymo procese: mokyklose naudojamos KMP paprastai yra maži, žemo interaktyvumo lygio mokomieji žaidimai, su kuriais sukurtą platformą susitvarko geriausiai. Platformą taip pat galima plėtoti pridėdant naujas galimybes, pritaikant kitoms sritims.



## 8. LITERATŪRA

1. Informacinių ir komunikacinių technologijų diegimo į bendrąjį lavinimą ir profesinį mokymą 2008–2012 metų strategija. Lietuvos Respublikos Seimas [žiūrėta 2011-04-13]. Prieiga per internetą:  
[http://www3.lrs.lt/pls/inter3/dokpaieska.showdoc\\_l?p\\_id=312799&p\\_query=D%EB1%20reikalavim%F8%20mokytoj%F8%20kompiuterinio%20ra%F0tingumo&p\\_tr2=2#](http://www3.lrs.lt/pls/inter3/dokpaieska.showdoc_l?p_id=312799&p_query=D%EB1%20reikalavim%F8%20mokytoj%F8%20kompiuterinio%20ra%F0tingumo&p_tr2=2#)
2. Tatjana Kriliuvienė. Tyrimo „Sukauptos metodinės patirties, naudojant IKT įvairiuose dalykuose, analizė“ ataskaita. Vilnius, 2008 [žiūrėta 2011-04-13]. Prieiga per internetą: [http://www.ipc.lt/wp-content/uploads/2009/11/tyrimas\\_metodika.pdf](http://www.ipc.lt/wp-content/uploads/2009/11/tyrimas_metodika.pdf)
3. Naudojimasis kompiuteriu pagal amžiaus grupes. Lietuvos statistikos departamentas [žiūrėta 2011-04-13]. Prieiga per internetą:  
<http://db1.stat.gov.lt/statbank/selectvarval/saveelections.asp?MainTable=M9020201&PLanguage=0&TableStyle=&Buttons=&PXSID=9492&IQY=&TC=&ST=ST&rvar0=&rvar1=&rvar2=&rvar3=&rvar4=&rvar5=&rvar6=&rvar7=&rvar8=&rvar9=&rvar10=&rvar11=&rvar12=&rvar13=&rvar14=>
4. Informacinės ir žinių visuomenės plėtros statistiniai rodikliai. Lietuvos statistikos departamentas [žiūrėta 2011-04-13]. Prieiga per internetą:  
[http://www.stat.gov.lt/uploads/docs/IV\\_pletros\\_rodikliai\\_2010.xls](http://www.stat.gov.lt/uploads/docs/IV_pletros_rodikliai_2010.xls)
5. Mokyklinės informatikos didaktikos mokymo medžiaga. KTU Programų inžinerijos katedra [žiūrėta 2011-04-13]. Prieiga per internetą:  
<http://proin.ktu.lt/didaktika/medziaga.html>
6. Įsakymas dėl pradinio ir pagrindinio ugdymo bendrųjų programų patvirtinimo. Lietuvos Respublikos Seimas [žiūrėta 2011-04-14]. Prieiga per internetą:  
[http://www3.lrs.lt/pls/inter3/dokpaieska.showdoc\\_l?p\\_id=326307&p\\_query=Bendrosios%20programos&p\\_tr2=2](http://www3.lrs.lt/pls/inter3/dokpaieska.showdoc_l?p_id=326307&p_query=Bendrosios%20programos&p_tr2=2)
7. E-learning. Wikipedia, the free encyclopedia [žiūrėta 2011-04-14]. Prieiga per internetą: [http://en.wikipedia.org/wiki/E\\_learning](http://en.wikipedia.org/wiki/E_learning)
8. Client/Server Architecture Types. eHow tech [žiūrėta 2012-03-31]. Prieiga per internetą: [http://www.ehow.com/list\\_6706525\\_client-server-architecture-types.html](http://www.ehow.com/list_6706525_client-server-architecture-types.html)
9. SmartFoxServer [žiūrėta 2012-04-01]. Prieiga per internetą:  
<http://www.smartfoxserver.com/>
10. PlayerIO [žiūrėta 2012-04-01]. Prieiga per internetą:  
<http://playerio.com/>

11. Electrotank [žiūrėta 2012-04-01]. Prieiga per internetą:  
<http://www.electrotank.com/>
12. Union Platform [žiūrėta 2012-04-01]. Prieiga per internetą:  
<http://www.unionplatform.com/>
13. Photon Server [žiūrėta 2012-04-01]. Prieiga per internetą:  
<http://www.exitgames.com/>
14. What cloud computing really means. Infoworld, the leading source of information on emerging enterprise technologies [žiūrėta 2011-10-29]. Prieiga per internetą:  
<http://www.infoworld.com/d/cloud-computing/what-cloud-computing-really-means-031>
15. Mell, P., Grance, T. The NIST Definition of Cloud Computing. National Institute of Standards and Technology, 2011 [žiūrėta 2011-10-29]. Prieiga per internetą:  
<http://csrc.nist.gov/publications/nistpubs/800-145/SP800-145.pdf>
16. Photon Cloud Brings Light to Multiplayer for Everyone. iGameRadio, the Mac Gaming Podcast [žiūrėta 2012-04-01]. Prieiga per internetą:  
<http://www.igameradio.com/2012/03/02/photon-cloud-brings-light-to-multiplayer-for-everyone/>
17. Pålsson, J., Pannek, R. et al. A Generic Game Server. University of Gothenburg, 2011 [žiūrėta 2012-04-21]. Prieiga per internetą: <http://jeena.net/t/GGS.pdf>
18. Buro, M., Dardanovic, I. NECI's Generic Game Server. NEC Research Institute, 2007 [žiūrėta 2012-04-21]. Prieiga per internetą:  
<http://skatgame.net/mburo/ggsa/index.html>
19. Brandy, R. Building Flash Multiplayer Games. PlayerIO, 2010 [žiūrėta 2012-04-01]. Prieiga per internetą:  
<http://playerio.com/documentation/tutorials/building-flash-multiplayer-games-tutorial/networkarchitectures/>
20. Programming Language Popularity [žiūrėta 2012-04-28]. Prieiga per internetą:  
<http://langpop.com/>
21. Usage share of operating systems. Wikipedia, the free encyclopedia [žiūrėta 2012-05-12]. Prieiga per internetą:  
[http://en.wikipedia.org/wiki/Usage\\_share\\_of\\_operating\\_systems](http://en.wikipedia.org/wiki/Usage_share_of_operating_systems)

## **Online Platform for Interactive Teamwork in Real-Time**

### **Summary**

Usage of information technologies in people's lives is increasing, as well as increasing their application in the learning process. Unfortunately, those applications don't take advantage of team-work opportunity, although it is considered one of the most promising innovations in education.

To fill this gap, a platform, enabling educational games to integrate interactive real-time team-work, was designed. This platform is intended to be adopted to educational portal <http://mokinukai.lt>. The target is that the platform has the following characteristics: is versatile and adaptable to various areas, not only education; easily expandable; creating new games to work on the platform is as easy as possible.

For platform's architecture a mix of client/server and peer-to-peer model was chosen, it's characteristics suits the requirements best. The platform works best with educational games, having a few players and little traffic, so it is suitable for application in practice.

## 9. TERMINŲ IR SANTRUMPŲ ŽODYNAS

**API** (angl. *Application Programming Interface*) – sąsaja bendravimui tarp programinių komponentų.

**C++** – viena populiariausių programavimo kalbų, kilusi iš C kalbos.

**CRC** (angl. *Cyclic Redundancy Check*) – klaidų duomenyse aptikimo metodas.

**FPS** (angl. *First Person Shooter*) – žaidimų žanras, kuriame imituojamos kovos šaunamaisiais ginklais, o vaizdas žaidėjui perteikiamas pirmo asmens perspektyva.

**HTTP** (angl. *HyperText Transfer Protocol*) – plačiai naudojamas internete protokolas, skirtas hipertekstinei informacijai perduoti.

**IKT** – informacinės komunikacinės technologijos.

**IP** (angl. *Internet Protocol*) – protokolas duomenų perdavimui tinkle, atsakingas už paketų maršrutizavimą tarp tinklo mazgų.

**„Iš viršaus žemyn“ projektavimas** – programinėje įrangoje ir kitose srityse taikoma metodika, kai sistema projektuojama pradedant abstrakčiomis, bendresnėmis dalimis, baigiant smulkiausiomis. Priešinga „iš apačios į viršų“ metodikai.

**JVM** (angl. *Java Virtual Machine*) – virtuali mašina, skirta vykdyti Java kalbos sukompiliuotą tarpinį kodą.

**KMP** – kompiuterinė mokomoji priemonė.

**MMO(G)** (angl. *Massively Multiplayer Online Game*) – kompiuterinių žaidimų tipas, kuriame prie žaidimo pasaulio vienu metu gali prisijungti didelis žaidėjų kiekis.

**OSI** (angl. *Open System Interconnection*) – modelis, komunikacinę sistemą suskaidantis į abstrakcijos lygmenis.

**P2P** (angl. *peer-to-peer*) – dviejų ar daugiau tinklo mazgų sujungimas tiesiogiai, be tarpininkų.

**PaaS** (angl. *Platform as a Service*) – skaičiavimo debesyse sistemos sluoksnis, suteikiantis programų vykdymo terpę.

**POSIX** (angl. *Portable Operating System Interface*) – įvairių programavimo standartų grupė, skirta suderinamumui tarp skirtingų operacinių sistemų padidinti.

**SaaS** (angl. *Software as a Service*) – skaičiavimo debesyse sistemos sluoksnis, vartotojams prieinamas kaip taikomoji programinė įranga.

**Skaičiavimas debesyse** (angl. *Cloud Computing*) – skaičiavimų sritis, kurioje skaičiavimo ištekliai teikiami kaip paslauga, o vartotojai visiškai atskiriami nuo techninės įrangos.

**SSGG analizė** – analizės metodas, kuriuo aprašomos tiriamosios sistemos stiprybės, silpnybės, galimybės ir grėsmės.

**TCP** (angl. *Transmission Control Protocol*) – tinklo protokolas paketų perdavimui, užtikrinantis siunčiamų duomenų eiliškumą ir teisingumą.

**Tuneliavimas** – vienos duomenų struktūros įterpimas į kitą ir perdavimas tinklu.

**UDP** (angl. *User Datagram Protocol*) – tinklo protokolas paketų perdavimui, neužtikrinantis siunčiamų duomenų eiliškumo ir teisingumo, orientuotas į greitą veikimą.

**Ugniasienė** (angl. *Firewall*) – programinė ar techninė įranga, analizuojanti ir ribojanti duomenų perdavimą tinklu, prisijungimą prie tam tikrų prievadų.