

SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

DIPLOMSKI RAD br. 1340

**ANALIZA MREŽNOG PROMETA PRI INTERAKCIJI DVAJU
KORISNIKA U VIŠEKORISNIČKOJ 3D VIDEOIGRI**

Dorotea Šimunović

Zagreb, lipanj 2020.

SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

DIPLOMSKI RAD br. 1340

**ANALIZA MREŽNOG PROMETA PRI INTERAKCIJI DVAJU
KORISNIKA U VIŠEKORISNIČKOJ 3D VIDEOIGRI**

Dorotea Šimunović

Zagreb, lipanj 2020.

DIPLOMSKI ZADATAK br. 1340

Pristupnica: **Dorotea Šimunović (0036492897)**

Studij: Informacijska i komunikacijska tehnologija

Profil: Telekomunikacije i informatika

Mentor: prof. dr. sc. Maja Matijašević

Zadatak: **Analiza mrežnog prometa pri interakciji dvaju korisnika u višekorisničkoj 3D videoigri**

Opis zadatka:

Vaš zadatak je proširiti raniju inačicu 3D interaktivne računalne igre "Wizard Wars" razvijene primjenom programskog okvira Unity unutar diplomskog projekta poboljšanim načinima međusobne interakcije korisnika (primjerice, odabirom složenijih "čarolija", potporom za naprednije vizualne efekte). Osmislite i u lokalnoj mreži provedite eksperiment u kojem ćete snimiti mrežni promet prilikom interakcije dvaju igrača. Ako bude izvedivo, ponovite eksperiment s povezivanjem preko mreže širokog područja putem virtualne privatne mreže. Analizirajte odabrane značajke mrežnog prometa i ocijenite iskustvenu kvalitetu igranja uz različite uvjete u mreži. Svu potrebnu literaturu i uvjete za rad osigurat će Vam Zavod za telekomunikacije.

Rok za predaju rada: 30. lipnja 2020.

Za Domija, najdražeg suputnika na svakom putovanju.

Sadržaj

Uvod	1
1. Prethodno razvijena videoigra „Wizard wars“	2
2. Proširenje videoigre „Wizard wars“	6
2.1. Dizajn proširenja videoigre	6
2.1.1. Dizajn čarolije napada.....	6
2.1.2. Dizajn animacije napada avatara	7
2.2. Programska izvedba.....	8
2.2.1. Kreiranje složene čarolije napada	8
2.2.2. Kreiranje čarolije u Unityju	9
2.2.3. Kreiranje animacije napada avatara.....	17
2.2.4. Ostale animacije	19
2.3. Funkcionalnosti nakon nadogradnje	20
3. Analiza mrežnog prometa.....	22
3.1. Korišteni programi za provođenje eksperimenta	23
3.1.1. Wireshark.....	23
3.1.2. LogMeIn Hamachi.....	24
3.2. Provođenje eksperimenta.....	25
3.2.1. Međudolazno vrijeme	25
3.2.2. Jednaka dolazna vremena	26
3.3. Identifikacija distribucije slučajne varijable	26
3.3.1. Kolmogorov-Smirnovljev test	27
4. Rezultati i diskusija	31
4.1. Eksperiment 1.....	31
4.1.1. Exponpow	32
4.2. Eksperiment 2.....	33

4.2.1. Genextreme	35
4.3. Eksperiment 3.....	37
4.3.1. FatigueLife	40
4.4. Duljine paketa.....	41
Zaključak	44
Literatura	45
Sažetak.....	48
Summary	49

Uvod

Razvoj videoigara u aspektima bolje komunikacije, priča i računalne grafike, uzrokovao je i rast pripadajuće industrije. Počevši od igara za jednu osobu (engl. singleplayer), preko igara za više osoba ali na istom resursu, danas su višekorisničke umrežene videoigre (engl. multiplayer) neizostavni faktor tržišta videoigara. Od pojave višekorisničkih videoigara mijenjao se način na koji se višekorisničke igre mogu igrati. Danas su višekorisničke igre pretežito preko Interneta, ali se također mogu igrati i preko lokalne mreže.

Za analizu mrežnog prometa pri interakciji dvaju korisnika, kao baza, korištena je višekorisnička videoigra „Wizard wars“, dizajnirana i implementirana u okviru kolegija Diplomski projekt. Prethodno izrađena videoigra predviđena je za rad u lokalnoj mreži (engl. LAN), te je u sklopu ovog rada osmišljeno rješenje povezivanja preko Interneta, koristeći virtualnu privatnu mrežu (engl. VPN). U VPN-u se provode tri eksperimenta između umrežena dva korisnika, koja se međusobno razlikuju po bežičnom ili žičnom načinu spajanja na usmjeritelj. Prije snimanja mrežnog prometa, videoigra je dodatno nadograđena dodavanjem nove čarolije i animacije napada. Mrežni promet prilikom provođenja eksperimenta snima se pomoću analizatora paketa, nakon čega se provodi analiza snimljenog mrežnog prometa, specifično, duljine paketa i međudolaznih vremena. Cilj rada je identifikacija odgovarajuće slučajne varijable uzevši u obzir međudolazna vremena snimljenih paketa.

Rad je strukturiran u četiri poglavlja. U prvom poglavlju opisuje se višekorisnička umrežena videoigra korištena kao baza za diplomski rad. U drugom poglavlju opisan je dizajn i implementacija proširenja spomenute videoigre. Treće poglavlje opisuje analizu mrežnog prometa, potrebne alate za izvođenje analize i postupak provođenja eksperimenata. Dobiveni rezultati eksperimenata prodiskutirani su u četvrtom poglavlju.

1. Prethodno razvijena videoigra „Wizard wars“

Kao temelj diplomskog rada korištena je višekorisnička videoigra pod nazivom „Wizard wars“. Razvijena videoigra nastala je koristeći Unity (verzija 2019.2.8f), u sklopu kolegija Diplomski projekt [1]. Tema izrađene videoigre je magična bitka između više korisnika povezanih u umreženom lokalnom okruženju. Korisnik upravlja avатарom čarobnjaka koji ima mogućnost proizvoljnog kretanja po površini te upravljanja čarobnim štapićem koji mu služi za čaroliju napada te za podizanje obrambenog čarobnog štita u svrhu obrane od napada protivnika.

U videoigri implementirane su i funkcionalnosti poput:

- pridruživanje igrača u videoigru u lokalnoj mreži;
- napad igrača u obliku stvaranja čarolije iz štapa;
- obrana igrača podizanjem štita;
- animacije napada, trčanja i mirovanja;
- povratka na početnu točku prilikom potpunog gubitka života (engl. *respawn*);
- praćenja zdravlja igrača pomoću linije zdravlja iznad njega (engl. *health bar*);
- označavanja igrača prilikom pridruživanja u igru obojenom kapsulom u svrhu razlikovanja.

Avatar igrača i njegove tadašnje animacije preuzeti su kao gotovi programski paketi, gdje je avatar čovjekolika forma s prethodno definiranim kosturom, zglobovima, mišićima i teksturom. Tekstura avatara sastoji se od bijelog materijala za lice i tijelo te smeđeg materijala za šešir. Avatar također uključuje i štap, kao element koji se nalazi u koordinatnom sustavu samog avatara. U nastavku je prikazan korišteni avatar čarobnjaka (Slika 1.1) te čarobnjak u virtualnom okruženju (Slika 1.2).



Slika 1.1. Avatar čarobnjaka



Slika 1.2. Igrač u virtualnom okruženju

Naknadno je igraču dodano stvaranje čarolije nakon odgovarajuće animacije napada, pri čemu se ispred avatarovog štapa stvara čestični sustav (engl. *particle system*) oblika prikazanog na sljedećoj slici (Slika 1.3). Čestični sustav preuzet je kao gotovi programski paket nevezan uz avatara. Dodatno, riješena je funkcionalnost stvaranja

čestičnog sustava prilikom napada te utjecaj čestičnog sustava na ostale objekte okoline.

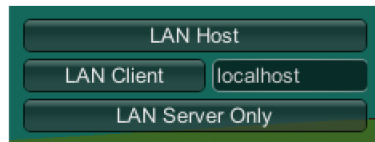


Slika 1.3. Čarolija napada

Videoigra je implementirana na temelju modela klijent – poslužitelj. Umreženost igre, u ovom slučaju u LAN mreži, postignuta je pomoću Mirrora verzije 5.0.2., sučelja koje pripada HL API alatima (engl. High Level Application Programming Interface). Mirror rješava komunikaciju između klijenta i poslužitelja, te u svojoj knjižnici definira oznake ispred metoda koje to omogućavaju:

- [Server] / [Client] - omogućava ograničavanje koda na izvođenje isključivo poslužiteljskih ili klijentskih dijelova;
- [Command] - koristi se pri komunikaciji klijent – poslužitelj;
- [ClientRpc] / [TargetRpc] - koriste se pri komunikaciji poslužitelj – klijent;
- [SyncVar] - koristi se za sinkronizaciju varijabli diljem umrežene okoline.

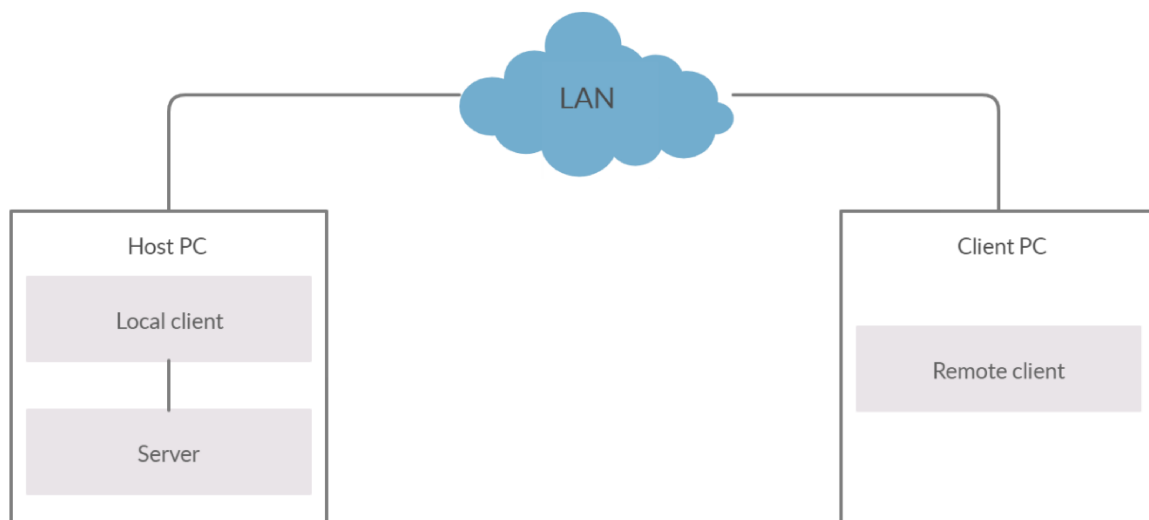
Mirror također rješava i uključivanje igrača u videoigru. Prilikom pokretanja igre, korisniku se prikazuje ekran za uključivanje igrača (Slika 1.4.), gdje korisnik ima tri opcije, „LAN Host“, „LAN Client“ i „LAN Server Only“.



Slika 1. 4. Sučelje za uključivanje igrača u igru

U slučaju odabira gumba „LAN Host“, korisnik postaje poslužitelj, no ujedno ima i pokrenutu instancu klijenta. Druga opcija, gumb „LAN Client“, koristi se ako se radi o dodavanju novog klijentskog igrača u igru, nakon što je poslužitelj već kreiran. Pored gumba „LAN Client“, korisnik unosi IP adresu poslužitelja na kojeg se spaja. Treća opcija je „LAN Server Only“, pomoću kojeg pokrenuta instanca služi isključivo kao poslužitelj ostalim računalima.

Arhitektura korištena u izradi diplomskog projekta, gdje jedno računalo ima ulogu poslužitelja i instancu klijenta, a drugo računalo ima instancu klijenta spojenog na poslužitelja prikazan je na slici 1.5.



Slika 1. 5. Arhitektura klijent - poslužitelj

2. Proširenje videoigre „Wizard wars“

Postojeći sustav proširen je poboljšanim načinima međusobne interakcije. Točnije, videoigra „Wizard wars“ je proširena u dva aspekta:

- kreiranje složenije čarolije napada
- kreiranje animacije napada avatara

2.1. Dizajn proširenja videoigre

Prije implementacije nadogradnje videoigre, dizajnirana je čarolija napada čarobnjaka te animacije napada, mirovanja i trčanja.

2.1.1. Dizajn čarolije napada

Za nadogradnju prethodne čarolije, bilo je potrebno dizajnirati čaroliju napada koja bi konkretnije identificirala avatara čarobnjaka. Ideja je bila kreirati čaroliju čije je kretanje poput projektila, dok se izgledom prepoznaju fantastični elementi. Za dizajniranje čarolije napada bilo je potrebno obratiti pozornost na sljedeće faktore:

1. Utjecaj čarolije
2. Stvaranje čarolije
3. Boja čarolije
4. Brzina čarolije
5. Oblik čarolije
6. Posljedica čarolije
7. Ponovno korištenje čarolije

Čarolija napada je zamišljena tako da avatar prvo izvede pokret tijelom za stvaranje čarolije na kraju čijeg se izvođenja u određenom trenutku čarolija stvara te ispucava u smjeru prema kojem igrač gleda. Ponavljanje čarolije moguće je tek nakon što se prethodna animacija izvede do kraja. Čarolija napada svakom novom kolizijom s protivničkim igračima smanjuje količinu života za jednu desetinu od moguće količine zdravlja igrača. Ako dođe do kolizije s bilo kojim elementom scene (igračem ili scenom prirode) čarolija trenutno nestaje. Bilo je potrebno obratiti pozornost na oblik i veličinu čarolije, kako bi se umjetnom elementu pridodao dojam fizikalnih svojstava. Pri tome je bilo potrebno razraditi ponašanje čarolije prilikom prolaska kroz zrak, kako bi se stvorio dojam aerodinamičnosti same čarolije, pri čemu uveliko značenje ima i sam

oblik čarolije. Dizajniranje je također uključivalo i procjenjivanje brzine čarolije, pri čemu je potrebno da je brzina čarolije u skladu s brzinom animacije napada.

2.1.2. Dizajn animacije napada avatara

Animacija napada prethodi stvaranju čarolije, te se pokreće prilikom korisnikove ulazne naredbe za napad. Kod dizajniranja animacije napada avatara bilo je potrebno razmotriti sljedeće faktore:

1. Fizikalna svojstva tijela
2. Prijenos energije
3. Raspoređenost vremena

Dizajn animacije napada sadrži elemente borilačkih vještina poput Kung-Fua i Tai Chija, kao i elemente inspirirane animiranim i igranim filmovima fantastične tematike. Naglasak kod animacije napada nalazi se u pokretu tijela, te prenošenju težine tijela s jedne noge na drugu. Budući da se animacija izvršava na čovjekolikom avataru, bitno je da pokreti budu u skladu sa stvarnim mogućnostima zglobova i mišića čovjeka.

Kod animacije napada bilo je bitno ostvariti prijenos energije, kako bi energija animacije mogla biti uzrok energiji čarolije koje slijedi, čime je postignuta ravnoteža toka između uzroka i posljedice. Posebice je potrebno precizno odrediti trajanje animacije napada. Prilikom izvođenja animacije napada, onemogućeno je kretanje igrača, te dulje trajanje napada znači veću izloženost riziku.

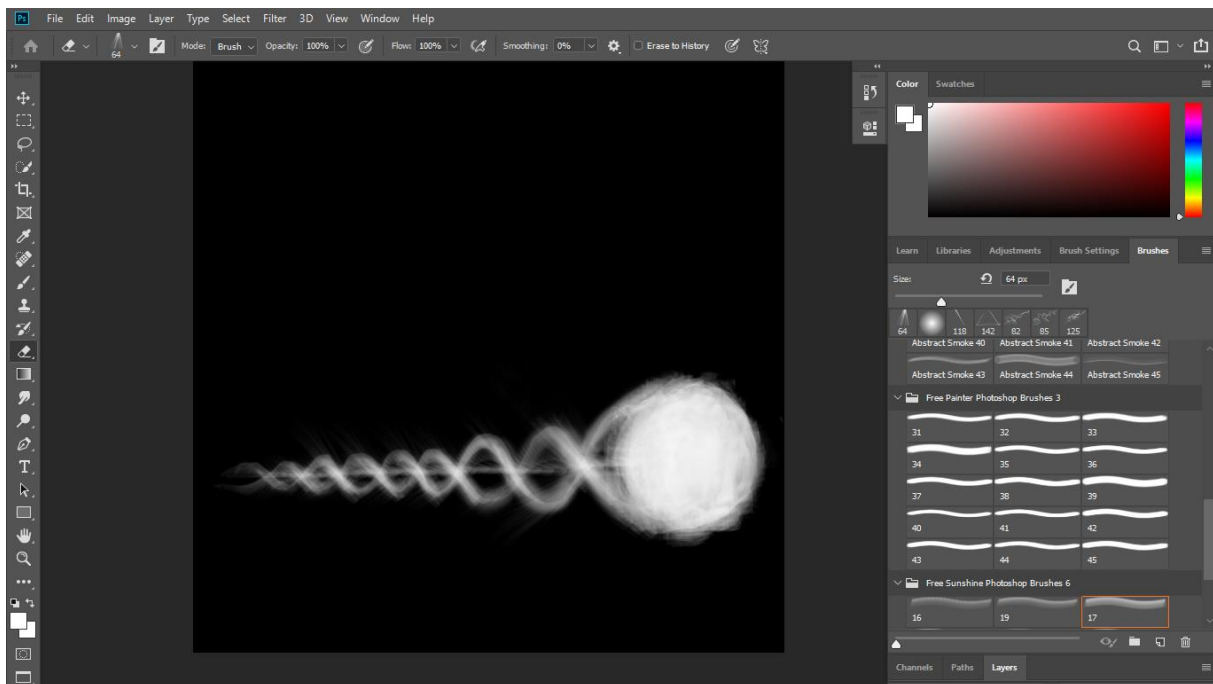
2.2. Programska izvedba

Za programsku izvedbu animacije napada korišten je čovjekoliki avatar preuzet iz Assets store-a prilikom izrade Diplomskog projekta, dok izrada čarolije napada započinje u Photoshopu (Free trial, verzija 21.1).

2.2.1. Kreiranje složene čarolije napada

Čarolija napada osmišljena je kao objekt oblika projektila, koji se sastoji od „glave“, okruglastog elementa na početku čarolije te spiralnog repa s povezanog s glavom, koji se postupno sužava prema kraju. Ideja prenesena na virtualno platno prikazana je na sljedećoj slici (Slika 2.1.).

Konstrukcija čarolije napada započinje u programu Adobe Photoshop (Free trial). Osim postojećih osnovnih alata za skiciranje, korišteni su dodatno preuzeti alati za sjenčanje, crtanje i dodavanje efekata. Ponajprije, bilo je potrebno postaviti crnu pozadinu u jedan sloj (engl. *layer*), te na to dodati drugi prozirni sloj, kako bi se crtanje alatima u bijeloj boji moglo očitovati. Za izradu čarolije projektila korištena je bijela boja, čime su otvorene opcije da se poslije u Unityju čarolija oboja u proizvoljnu boju, sadržavajući i dalje bijelu jezgru koja cijelom sustavu daje efekt svjetlosnog izvora.



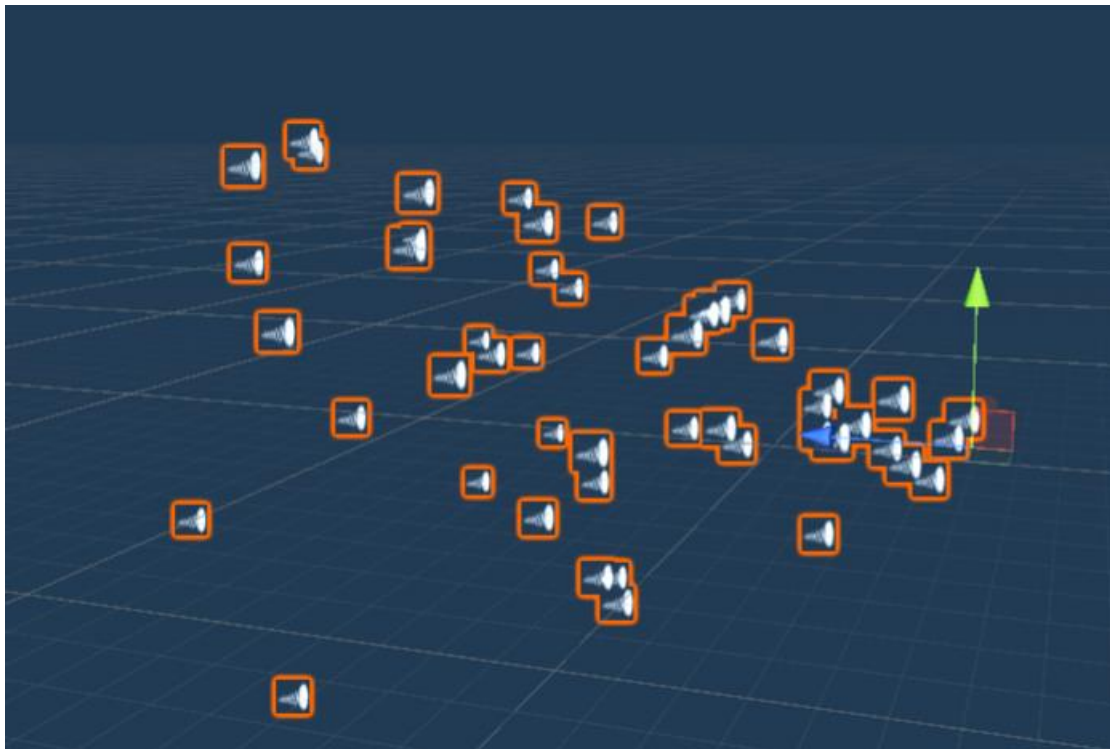
Slika 2. 1. Photoshop okruženje - početna skica projektila

Za izradu početne skice korišteni su alati za crtanje koji simuliraju slikarske kistove i četke, dim i maglu, te zrake sunca. Kreirani efekt je zatim izrezan tako da prati oblik projektila te mu je uklonjena pozadina. Na kraju, skica je morala biti izvedena u formatu koji podržava transparentnost, a to je .png format.

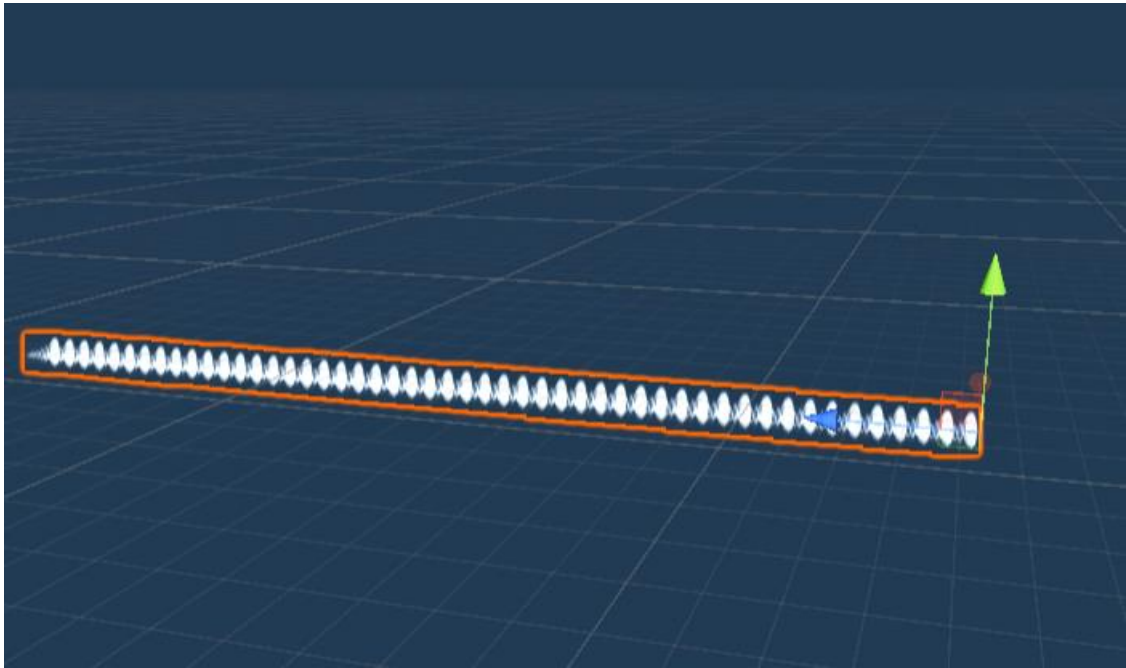
2.2.2. Kreiranje čarolije u Unityju

Čarolija je objekt koji se sastoji od tri čestična sustava: projektila, tragova i repa. Svakom čestičnom sustavu dodijeljena je tekstura čestice i ponašanje kroz niz postavki.

Kreiranje čarolije je započeto tako da je kreiran Game objekt u hijerarhiji scene. Objekt je nazvan PS_MagicProjectile te je njemu kao dijete dodijeljen čestični sustav PS. Zatim je kreiran novi materijal koji ima svojstvo da mu se može dodati vanjska tekstura, te mu je kao tekstura predana skica projektila kreirana u Photoshopu. Takav novo konstruirani materijal dodijeljen je čestičnom sustavu PS. Time je postignuto da sve čestice koje emitira izvor čestičnog sustava imaju teksturu projektila. Budući da je neželjeno ponašanje da postoji više raspršenih projektila u sceni kao što je to prikazano na slici 2.2, čestičnom sustavu se miče svojstvo oblika, čime se dobiva efekt da se čestice kreću po liniji, prikazano na slici 2.3, gdje su narančastom bojom prikazane obujmice pojedinih čestica.

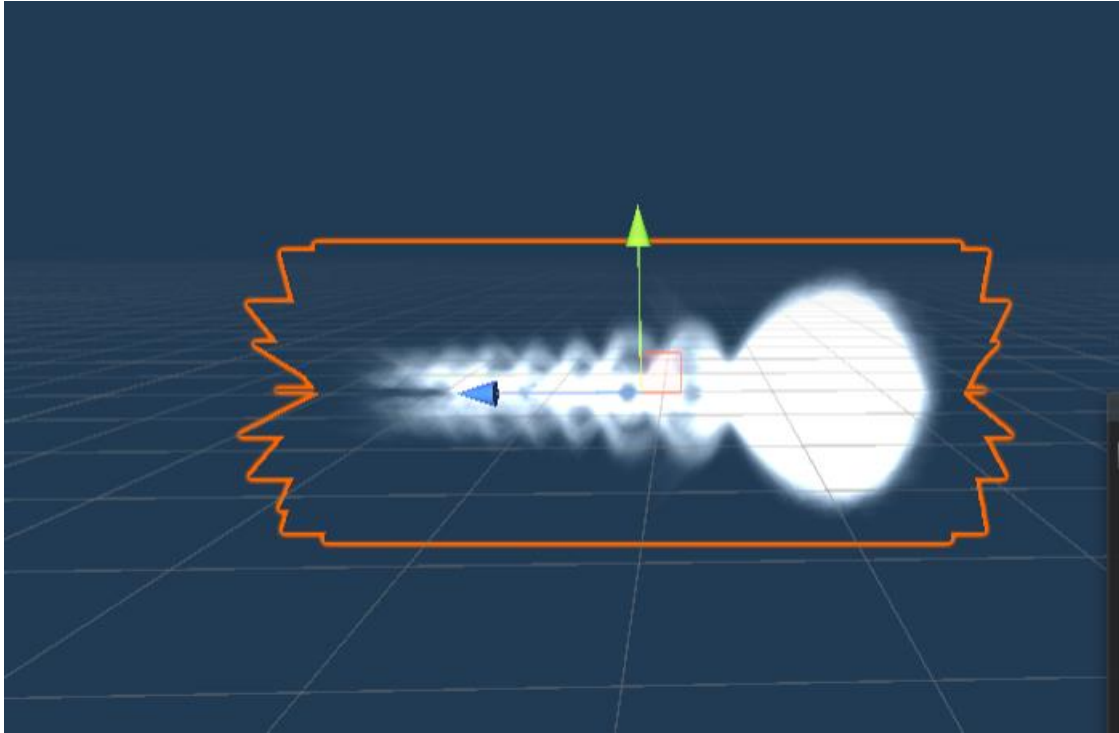


Slika 2. 2. Raspršeni čestični sustav



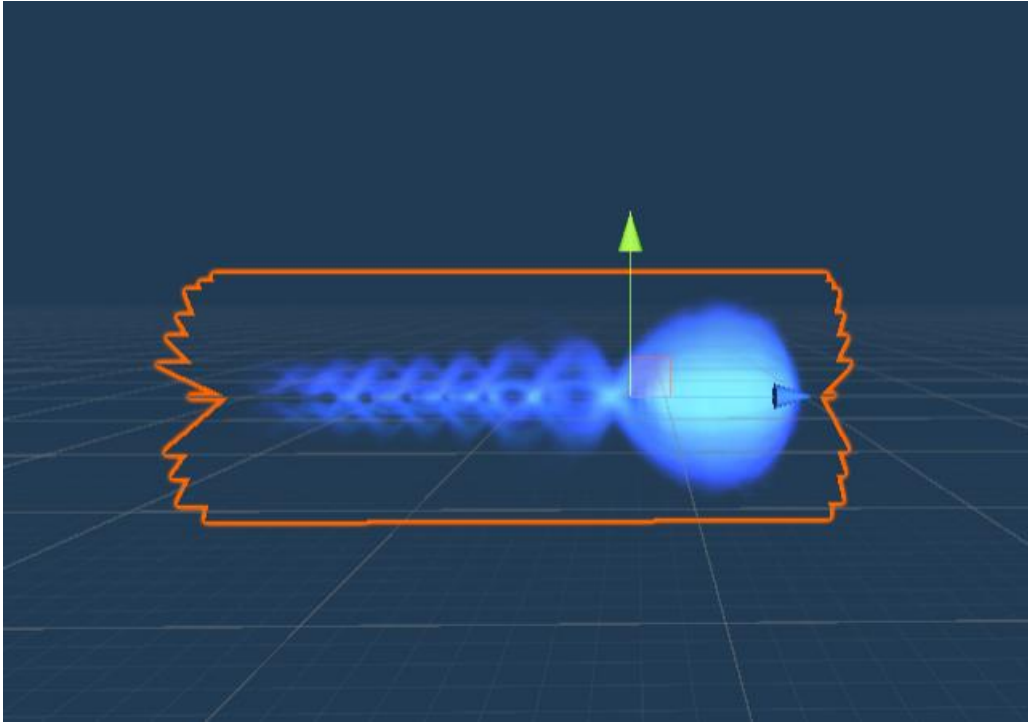
Slika 2. 3. Čestični sustav s linijskom putanjom

Zatim je uklonjena Z komponenta, pri čemu se gubi efekt da čestica putuje po liniji, te se čestica sada stvara uvijek na istim koordinatama. Kako bi se dobio trodimenzionalni efekt projektila iz čestice čiji je materijal dvodimenzionalan, sustavu se dodaje svojstvo rotacije kroz životni vijek, kojim je omogućeno da se čestični izvor rotira u smjeru x-osi te dobije efekt prikazan na slici 2.4.



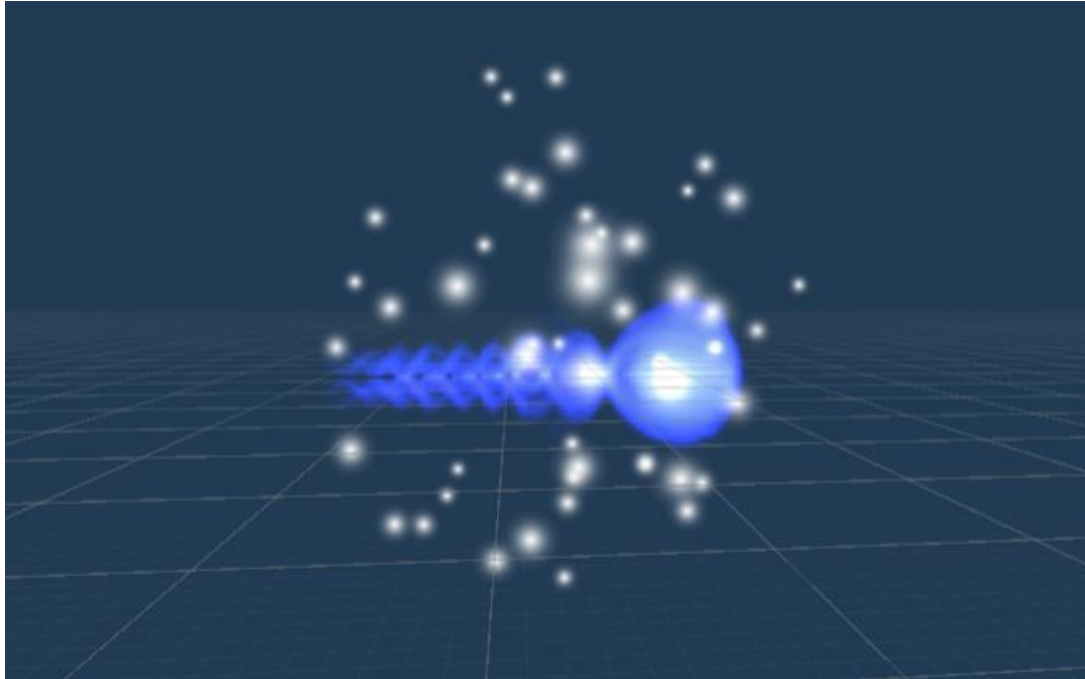
Slika 2. 4. Rotacija dvodimenzionalne slike

S obzirom na to da je početna skica bila bijele boje, čestičnom sustavu su sada dodane dvije boje te će se svakoj novonastaloj čestici pridružiti nasumična vrijednost boje u intervalu između te dvije konstante. Projektilu su još dodatno promijenjene postavke poput: količina novonastalih čestica po jedinici vremena, transparentnost čestica, duljina života, brzina rotacije te poravnavanje čestice u odnosu na koordinatne osi. Sustavu je također dodano svjetlo točkastog izvora, što uzrokuje da projektil putujući po sceni svijetli plavičastim svjetlom. Promjene na spomenutim postavkama očituju se na konačnom projektilu prikazanom na slici 2.5.

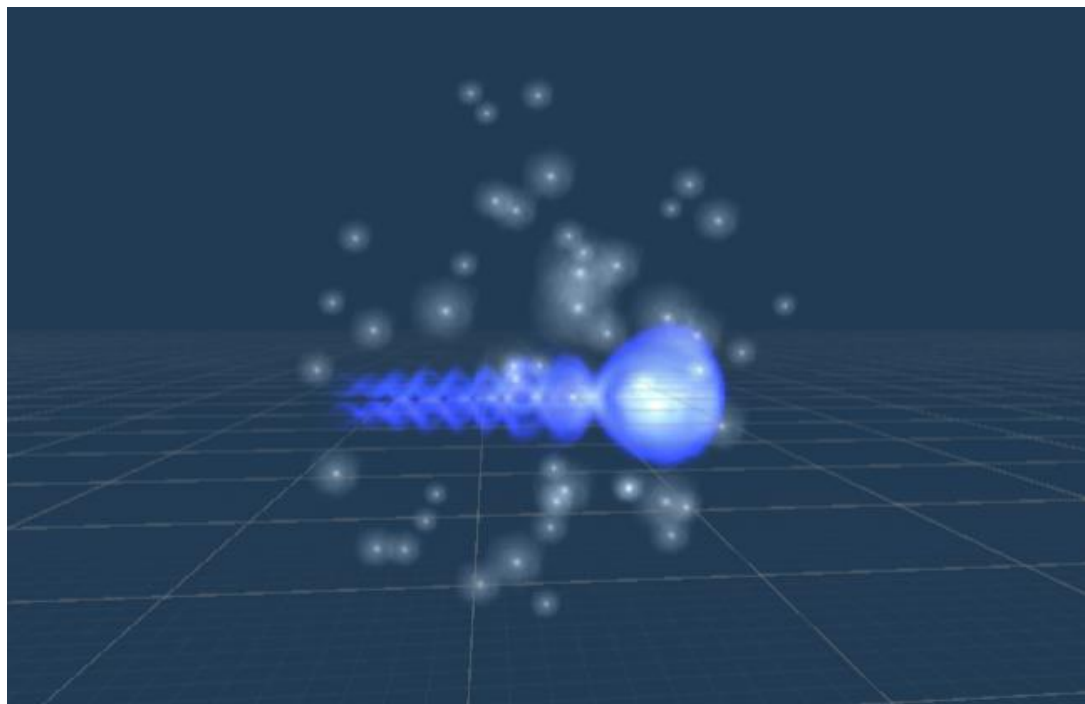


Slika 2. 5. Projektil

Nakon toga dodan je novi čestični sustav trenutnom projektilu, koji predstavlja tragove koje projektil ostavlja svojom rotacijom i translacijom kroz scenu. Kako bi tragovi projektila i projektil činili cjelinu, između njih je uspostavljena hijerarhija roditelj-dijete, pri čemu se dijete (tragovi projektila) nalazi u koordinatnom sustavu roditelja (projektil). Time je dobiveno da se svaka operacija translacije, rotacije ili skaliranja projektila odražava i na njegove tragove. Čestica korištena kao tekstura za novi čestični sustav također je skicirana u Photoshopu, te uvezena u Unity na isti način kao i skica projektila. Na slici 2.6. prikazan je novi čestični sustav s početnim česticama, te je na slici 2.7. prikazan sustav nakon što mu je dodana skicirana tekstura.



Slika 2. 6. Početni čestični sustav



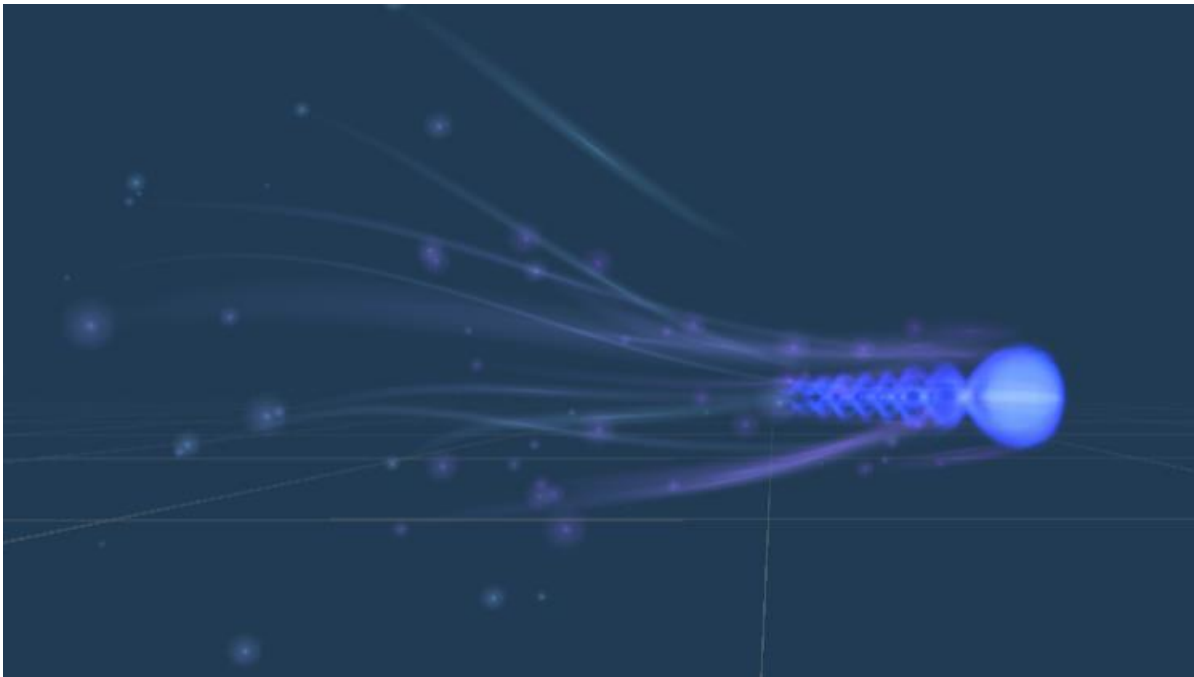
Slika 2. 7. Čestični sustav s dodanom teksturom čestice

Nad novim čestičnim sustavom su zatim napravljene sljedeće promjene:

1. Oblik smjera širenja čestica je postavljen na oblik stošca, kojem mu slijedno tome smanjen radijus kako bi smjer širenja čestica pratio oblik projektila;

2. Nasumično vrijeme života postavljeno je između dvije zadane konstante, čime se dobiva efekt prirodnijeg nestajanja čestica;
3. Nasumična veličina postavljena je između dvije konstante, čime se uvodi raznovrsnost čestica;
4. Postavljen je vektor brzine za svaku česticu, gdje se svakoj čestici dodaje vektor brzine vrijednosti između $[-2, -2, -2]$ i $[2, 2, 2]$, čime se dobiva efekt da čestice idu blago raspršenim putanjama iza projektila;
5. Sustavu je dodana ista boja kao i projektilu;
6. Veličina čestice se mijenja tijekom vremena, od veće prema manjoj;
7. Dodani su tragovi čestica, gdje se 40% čestica pridružuje trag koji prati putanju kretanja čestice;
8. Dodan je šum traga čestice, postavka kojom se dodaje šum linearnom tragu čestice te mijenja putanja pomoću mijenjanja frekvencije i intenziteta šuma.

Navedenim promjenama stvoren je efekt prikazan na slici 2.8.



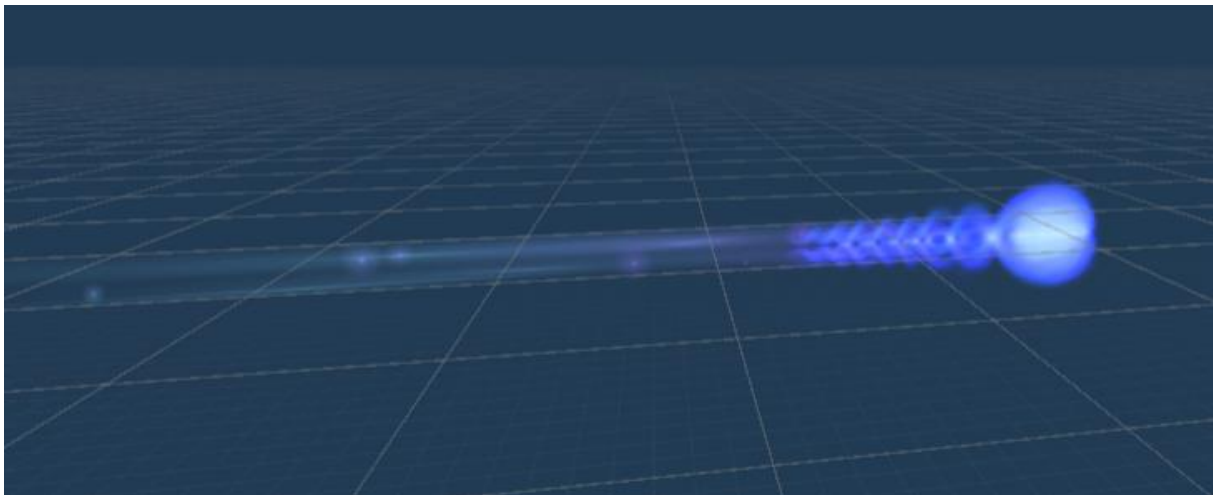
Slika 2. 8. Efekt nakon dodanog čestičnog sustava

Na kraju je dodan još jedan čestični sustav koji služi kao rep projektila, koji za razliku od prošlog čestičnog sustava nema šumova, zbog čega točno prati putanju projektila.

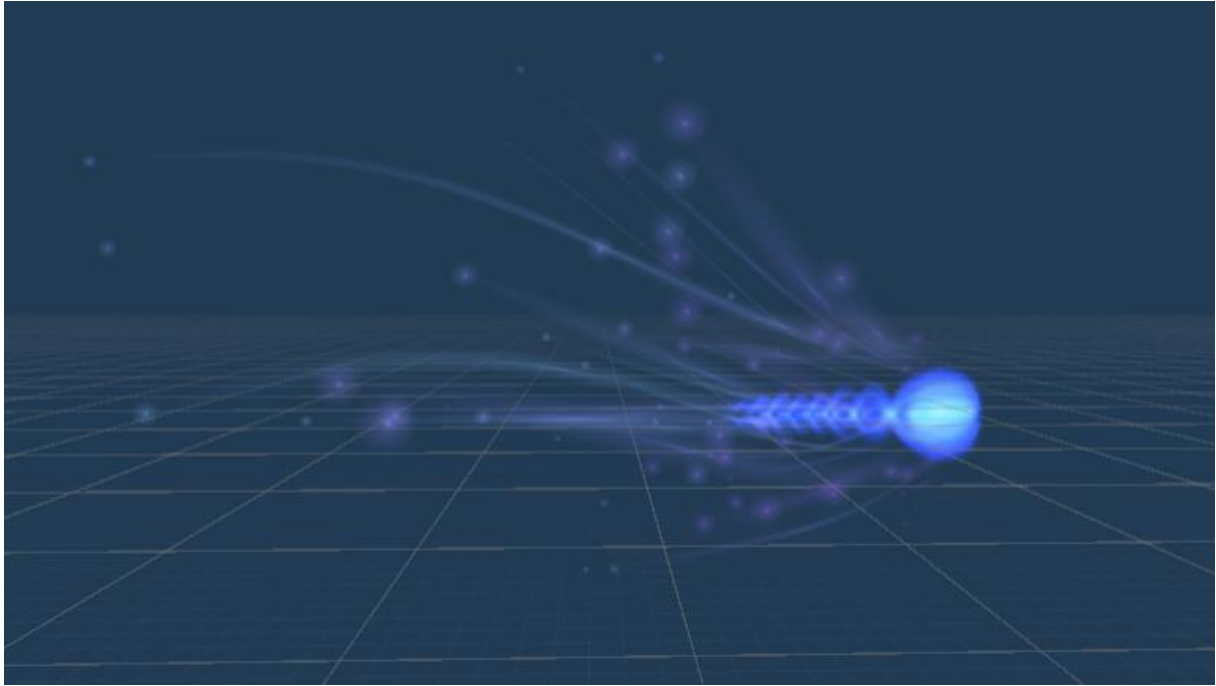
Čestični sustav repa je kao i trag projektila dodan kao dijete roditeljskom objektu (projektilu) te je konačna hijerarhija roditelj-djeca prikazana na slici 2.9. Rep je translaticiran tako da izvor sustava počinje na sredini projektila. Prostor kretanja je bio valjak, kojemu je promjer smanjen na iznos pri kojem je bio uži od samog projektila. Sve ostale postavke su postavljene isto kao i na prošlom čestičnom sustavu, uz iznimku da je trag čestica postavljen na 100% čestica podsustava. Projektil sa samo čestičnim sustavom repa prikazan je na slici 2.10., a sveukupni rezultat prikazan je na slici 2.11.



Slika 2. 9. Hijerarhija objekata projektila



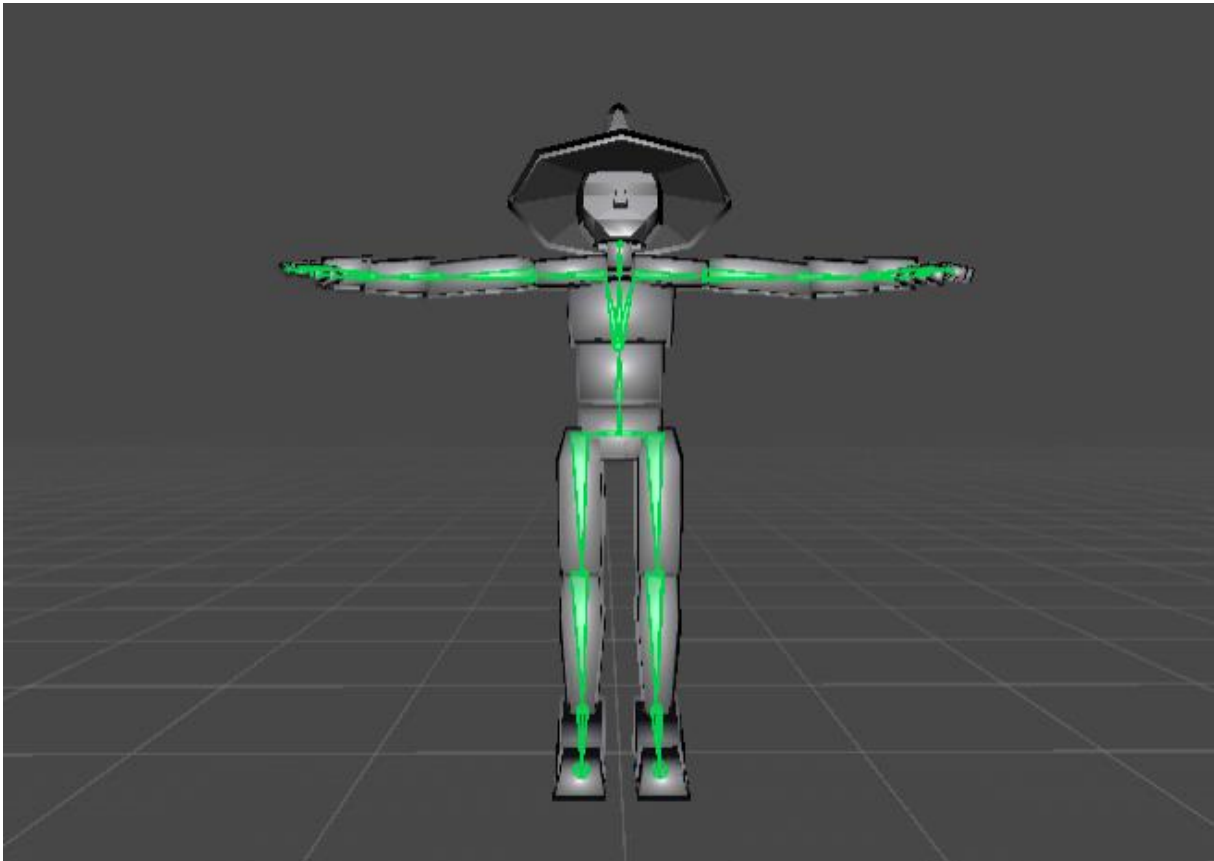
Slika 2. 10. Rep čestičnog sustava



Slika 2. 11. Konačni čarobni projektil

2.2.3. Kreiranje animacije napada avatara

Za kreiranje animacije napada korišten je model čarobnjaka preuzet iz Asset store-a. Model sadrži kosti i mišiće omogućene za animiranje. Prikaz kostura čarobnjaka prikazan je zelenom bojom na slici 2.12.



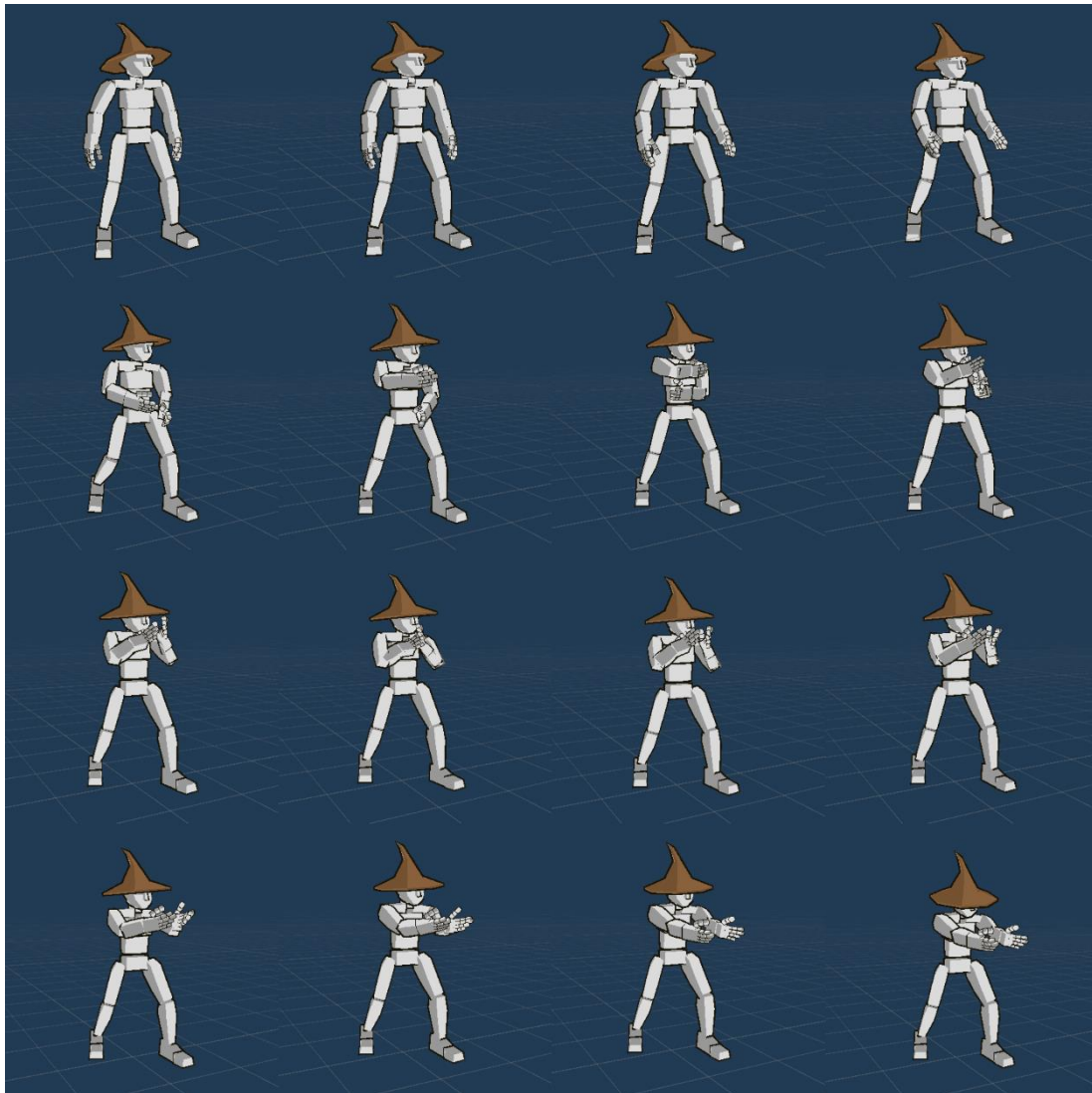
Slika 2. 12. Kostur avatara

Model čarobnjaka podijeljen je u hijerarhijski posložene dijelove tijela, gdje pojedini dio sadrži svoj lokalni koordinatni sustav te se nad njime mogu raditi operacije translacije, rotacije i skaliranja. Hijerarhija modela s 51 dijelom tijela prikazana je na slici 2.13.



Slika 2. 13. Hijerarhija kostura

Za kreiranje animacije korišten je „Unity Animations“ prozor, u kojem je kreiran prazan isječak animacije. „Animations“ prozor omogućuje kreiranje isječka animacije pomoću translacije, rotacije i skaliranja objekta kroz vrijeme. Animacija napada počinje iz ručno izrađenog početnog stanja (engl. *idle*), kako bi prijelaz iz stanja mirovanja u stanje napada bio neprimjetan. Inicijalno, animacija napada čarobnjaka trajala je 1.10 sekunde, uz mogućnost da se naknadno trajanje pomnoži s multiplikatorom te se time animacija uspori ili ubrza. Cijela animacija napada sastoji se od 16 kadrova (engl. *key frame*), koji su svi prikazani na slici 2.14.



Slika 2. 14. Animacija napada

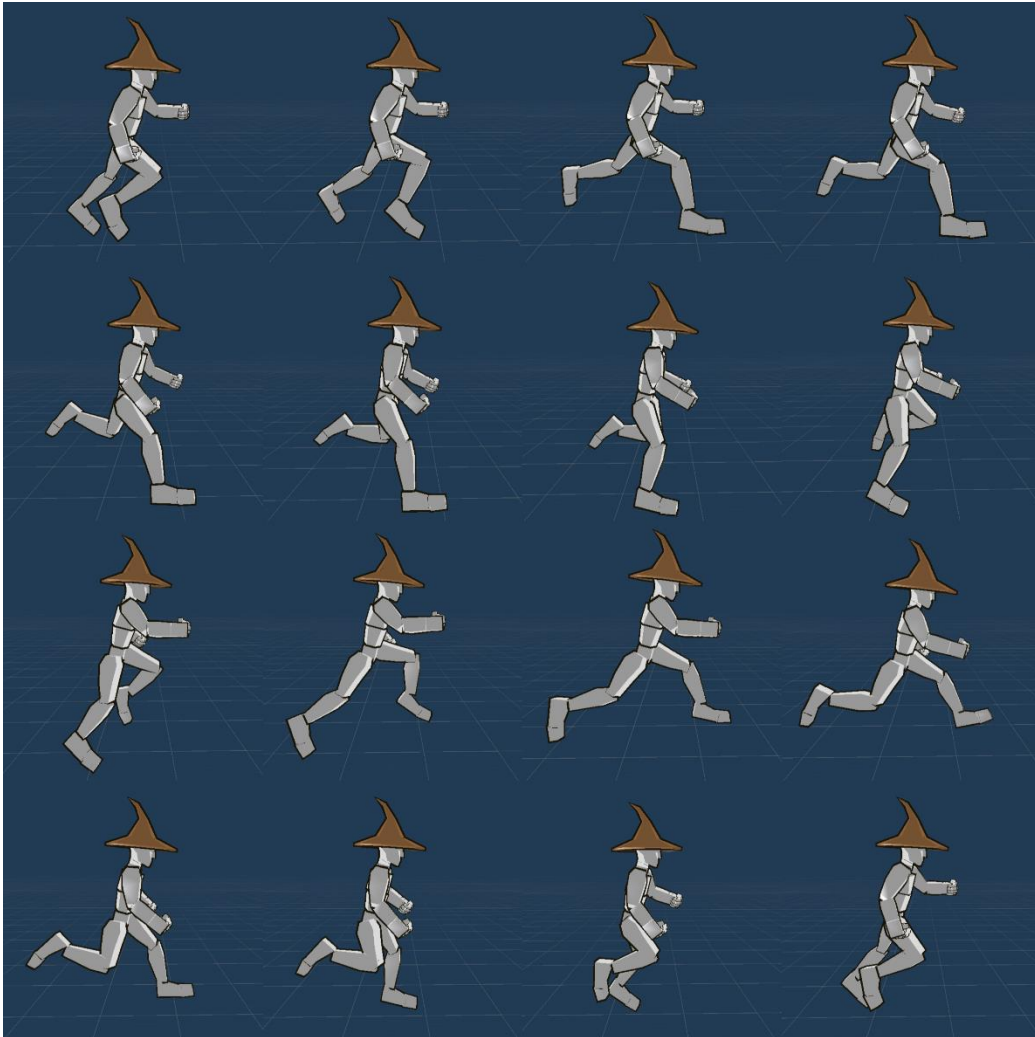
2.2.4. Ostale animacije

Osim animacije napada, pomoću Unityja i modela čarobnjaka kreirane su još i animacija početnog stanja (engl. *idle*) i animacija kretanja.

Animacija početnog stanja je animacija koja se prikazuje kad avatar nije niti u stanju trčanja niti napada. Tom animacijom pridonosi se oživljavanju avatara budući da animacija prikazuje disanje blagim kretnjama prsnog koša, ramena, ali i savijanje zglobova nogu, čime se na preuveličani način prikazuje animacija mirovanja.

Animacija trčanja je animacija koja se prikazuje prilikom kretanja korisnika u bilo kojem smjeru. To je animacija koja je onemogućena u slučaju napada korisnika gdje se pomoću lijevog klika miša trenutno prekida. Trajanje animacije je 0.16 sekundi te je

animacija usporena naknadno nakon iskustvenog ispitivanja, a sastoji se od 16 kadrova, navedenih na slici 2.15.



Slika 2.15. Animacija trčanja

2.3. Funkcionalnosti nakon nadogradnje

Nakon što se kreirane nova čarolija napada i animacije napada, mirovanja i trčanja, u suradnji s kolegom, dodane su dvije vrste AI ne-igračkih elemenata, neprijatelja s mogućnošću praćenja igrača te stacionarnog neprijatelja. Konačno, nadograđena verzija igre „Wizard wars“ sadrži sljedeće glavne funkcionalnosti:

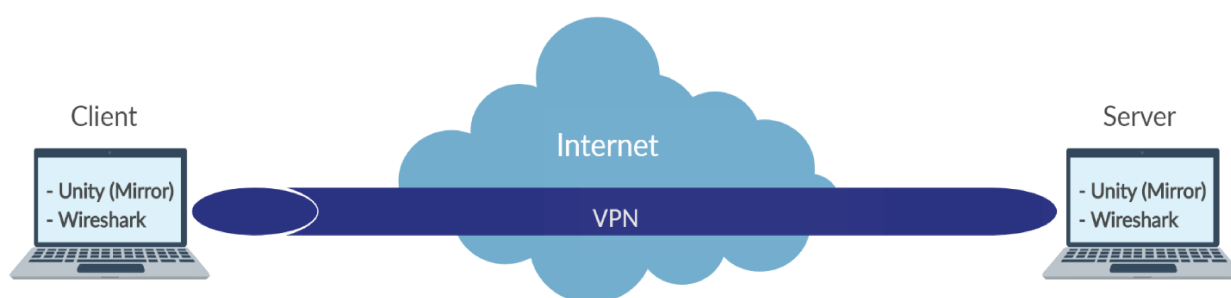
1. Pridruživanje igri
2. Kretanje u sceni
3. Napad avatara
4. Napad AI neprijatelja

5. Podizanje štita
6. Animacije avatara
7. Animacije AI neprijatelja

3. Analiza mrežnog prometa

Analizom mrežnog prometa moguće je saznati informacije o promatranoj računalnoj mreži, načinima njenog korištenja te podacima koji se njome prenose. Postupak praćenja mrežnog prometa često se koristi u aktivnostima kao što su otkrivanje problema u radu računalne mreže i/ili mrežnih servisa ili primjerice otkrivanju sigurnosnih propusta, no jednako tako ga mogu koristiti korisnici s namjerom provođenja malicioznih aktivnosti. Osim toga, analiza mrežnog prometa može se koristiti i za prikupljanje statistike mrežnog prometa, što je upravo i promatrano u ovom radu.

U snimanju mrežnog prometa sudjelovala su dva računala s instaliranim Unityjem s nadograđenom verzijom videoigre „Wizard wars“, aplikacijom za kreiranje virtualne privatne mreže i analizatorom paketa za snimanje prometa. Predmeti analize su međudolazna vremena paketa i duljine paketa prilikom interakcije dvaju korisnika. Snimanje pomoću analizatora podataka provedeno je tako da se snima isključivo preko sučelja virtualne privatne mreže, kako bi se izolirao ostali mrežni promet. Jedno računalo osim za pokretanje klijentske instance videoigre služilo je i kao poslužitelj, dok je drugo računalo imalo pokrenuto klijentsku instancu. Općenita arhitektura za provedene eksperimente prikazana je na slici 3.1.



Slika 3. 16. Arhitektura za provedene eksperimente

S obzirom na način spajanja pojedinih računala na svoje usmjeritelje, razlikuju se tri eksperimenta nad kojima je provedena mrežna analiza:

1. Eksperiment 1 - jedno računalo je spojeno preko WiFi-a, dok je drugo spojeno u Ethernetu;
2. Eksperiment 2 - oba računala su spojena preko WiFi-a;
3. Eksperiment 3 - oba računala su spojena preko Etherneta.

Za svaki eksperiment snimljeno je 10 snimki prometa, unutar kojih su provedeni koraci testiranja funkcionalnosti videoigre. Na poslijetku pojedinog eksperimenta, iz snimaka prometa izračunato je međudolazno vrijeme paketa te je objedinjeno u jednu .csv datoteku prilagođenu za daljnju identifikaciju slučajne varijable. Za svaki eksperiment također su izdvojene i duljine paketa.

3.1. Korišteni programi za provođenje eksperimenta

Za potrebe analize, korišten je besplatni analizator paketa Wireshark, a za spajanje korisnika u virtualno okruženje korištena je privatna virtualna mreža pomoću aplikacije LogMeIn Hamachi, detaljnije opisana u nastavku.

3.1.1. Wireshark

Programska rješenja za analizu mrežnog prometa nazivaju se analizatori paketa (engl. *sniffers*). To su alati koji imaju mogućnost prikazati pakete bez zaglavlja (engl. header), odnosno sposobni su dekodirati informacije specifične za protokole danog paketa te ih prikazati u čitljivom obliku. Definiraju se kao računalni program ili dio računalnog hardvera koji može presresti i zabilježiti promet koji prolazi mrežom ili dijelom mreže. Wireshark je besplatni alat otvorenog koda (engl. *open source*) korišten u radu za analizu mrežnog prometa. Posljednja stabilna verzija Wireshark-a je verzija 3.2.4. te je ista korištena za analizu. Budući da u pravilu programi ne mogu pristupiti memoriji rezerviranoj u jezgri operacijskog sustava, kako bi se informacije mogle očitati s mrežnog stoga (engl. *network stack*) to se treba učiniti preko programskog sučelja (engl. *application programming interface – API*). Tako, ovo softversko rješenje u pozadini koristi tzv. pcap (engl. *packet capture*), biblioteku koja programima pruža programsko sučelje za dohvaćanje paketa s mrežnih sučelja na operacijskim sustavima Windows i Linux/Unix. Primjer Wireshark sučelja s prikazanim TCP prometom prikazan je na slici 3.2.

No.	Time	Source	Destination	Protocol	Length	Info
2432	13.082271	25.94.70.91	25.57.56.123	TCP	83	7777 → 49286 [PSH, ACK] Seq=35241 Ack=2742 Win=62731 Len=29
2433	13.082329	25.57.56.123	25.94.70.91	TCP	54	49286 → 7777 [ACK] Seq=2742 Ack=35270 Win=65113 Len=0
2434	13.082335	25.57.56.123	25.94.70.91	TCP	54	[TCP Dup ACK 2433#1] 49286 → 7777 [ACK] Seq=2742 Ack=35270 Win=65113 Len=0
2435	13.098345	25.94.70.91	25.57.56.123	TCP	83	7777 → 49286 [PSH, ACK] Seq=35270 Ack=2742 Win=62731 Len=29
2436	13.100823	25.94.70.91	25.57.56.123	TCP	83	7777 → 49286 [PSH, ACK] Seq=35299 Ack=2742 Win=62731 Len=29
2437	13.100931	25.57.56.123	25.94.70.91	TCP	54	49286 → 7777 [ACK] Seq=2742 Ack=35328 Win=65055 Len=0
2438	13.100938	25.57.56.123	25.94.70.91	TCP	54	[TCP Dup ACK 2437#1] 49286 → 7777 [ACK] Seq=2742 Ack=35328 Win=65055 Len=0
2439	13.114258	25.94.70.91	25.57.56.123	TCP	83	7777 → 49286 [PSH, ACK] Seq=35328 Ack=2742 Win=62731 Len=29
2440	13.116256	25.94.70.91	25.57.56.123	TCP	83	7777 → 49286 [PSH, ACK] Seq=35357 Ack=2742 Win=62731 Len=29
2441	13.116303	25.57.56.123	25.94.70.91	TCP	54	49286 → 7777 [ACK] Seq=2742 Ack=35386 Win=64997 Len=0
2442	13.116309	25.57.56.123	25.94.70.91	TCP	54	[TCP Dup ACK 2441#1] 49286 → 7777 [ACK] Seq=2742 Ack=35386 Win=64997 Len=0
2443	13.118627	25.94.70.91	25.57.56.123	TCP	94	7777 → 49286 [PSH, ACK] Seq=35386 Ack=2742 Win=62731 Len=40
2444	13.132314	25.94.70.91	25.57.56.123	TCP	83	7777 → 49286 [PSH, ACK] Seq=35426 Ack=2742 Win=62731 Len=29
2445	13.132357	25.57.56.123	25.94.70.91	TCP	54	49286 → 7777 [ACK] Seq=2742 Ack=35455 Win=64928 Len=0
2446	13.132361	25.57.56.123	25.94.70.91	TCP	54	[TCP Dup ACK 2445#1] 49286 → 7777 [ACK] Seq=2742 Ack=35455 Win=64928 Len=0
2447	13.134261	25.94.70.91	25.57.56.123	TCP	83	7777 → 49286 [PSH, ACK] Seq=35455 Ack=2742 Win=62731 Len=29
2448	13.136592	25.94.70.91	25.57.56.123	TCP	68	7777 → 49286 [PSH, ACK] Seq=35484 Ack=2742 Win=62731 Len=14
2449	13.136630	25.57.56.123	25.94.70.91	TCP	54	49286 → 7777 [ACK] Seq=2742 Ack=35498 Win=64885 Len=0
2450	13.136635	25.57.56.123	25.94.70.91	TCP	54	[TCP Dup ACK 2449#1] 49286 → 7777 [ACK] Seq=2742 Ack=35498 Win=64885 Len=0
2451	13.150147	25.94.70.91	25.57.56.123	TCP	83	7777 → 49286 [PSH, ACK] Seq=35498 Ack=2742 Win=62731 Len=29
2452	13.150232	25.94.70.91	25.57.56.123	TCP	83	7777 → 49286 [PSH, ACK] Seq=35527 Ack=2742 Win=62731 Len=29
2453	13.150263	25.57.56.123	25.94.70.91	TCP	54	49286 → 7777 [ACK] Seq=2742 Ack=35556 Win=64827 Len=0
2454	13.150267	25.57.56.123	25.94.70.91	TCP	54	[TCP Dup ACK 2453#1] 49286 → 7777 [ACK] Seq=2742 Ack=35556 Win=64827 Len=0
2455	13.164349	25.94.70.91	25.57.56.123	TCP	83	7777 → 49286 [PSH, ACK] Seq=35556 Ack=2742 Win=62731 Len=29
2456	13.167127	25.94.70.91	25.57.56.123	TCP	83	7777 → 49286 [PSH, ACK] Seq=35585 Ack=2742 Win=62731 Len=29
2457	13.167174	25.57.56.123	25.94.70.91	TCP	54	49286 → 7777 [ACK] Seq=2742 Ack=35614 Win=64769 Len=0
2458	13.167178	25.57.56.123	25.94.70.91	TCP	54	[TCP Dup ACK 2457#1] 49286 → 7777 [ACK] Seq=2742 Ack=35614 Win=64769 Len=0

> Frame 2444: 83 bytes on wire (664 bits), 83 bytes captured (664 bits) on interface \Device\NPF_{7C5AF22C-8592-4D01-807D-72991DF8085D}, id 0
> Ethernet II, Src: 7a:79:19:5e:46:5b (7a:79:19:5e:46:5b), Dst: 7a:79:19:39:38:7b (7a:79:19:39:38:7b)
> Internet Protocol Version 4, Src: 25.94.70.91, Dst: 25.57.56.123
> Transmission Control Protocol, Src Port: 7777, Dst Port: 49286, Seq: 35426, Ack: 2742, Len: 29
> Data (29 bytes)

```

0000  7a 79 19 39 38 7b 7a 79 19 5e 46 5b 08 00 45 00  zy-98{zy -^F[...E-
0010  00 45 28 77 40 00 80 06 20 cf 19 5e 46 5b 19 39  -E(w@... -^F[-9
0020  38 7b 1e 61 c0 86 a3 db 71 db ee 34 f9 bd 50 18  8{a-...q-4-...P-
0030  f5 0b 9f 0e 00 00 00 00 00 19 ba 84 14 00 dc cd  -.....
0040  70 f5 11 93 ba ed 3e 4e 30 d4 3e 2e ab e8 3e c9  p-...>N 0->...>-
0050  c8 48 3f                                     -H?

```

Slika 3.2. Wireshark okruženje

3.1.2. LogMeln Hamachi

Kako bi povezivanje preko mreže širokog područja bilo izvedivo, korištena je besplatna aplikacija LogMeln Hamachi, koja omogućava ostvarivanje virtualne privatne mreže. Pomoću Hamachi-ja moguće je ostvariti direktnu vezu između računala, iza kojih se nalazi NAT (engl. network address translation) i vatrozid, i to bez potrebe za rekonfiguracijom. Drugim riječima, između računala se uspostavlja veza preko Interneta koja oponaša vezu koja bi postojala kad bi računala bila spojena u lokalnu mrežu (engl. *local area network* – LAN).

Cjelokupno gledano, Hamachi predstavlja VPN sustav sa središnjim upravljanjem, koji se sastoji od skupine poslužiteljskih računala (engl. *server cluster*) i od klijentskog softvera, instaliranog na računalima krajnjih korisnika. Klijentski softver dodaje sučelje virtualne mreže računalu, a koristi se za presretanje dolaznog i ubacivanje odlaznog VPN prometa. Dolazni promet šalje operativni sustav na sučelje, te se promet zatim dostavlja do klijentskog softvera, gdje se promet šifrira i ovjerava, zatim šalje do određiškog VPN čvora.

3.2. Provođenje eksperimenta

Nakon konfiguracije, započeto je snimanje u Wireshark-u na oba računala, koristeći sučelje pod nazivom „Hamachi“ kako bi se omogućilo snimanje isključivo VPN prometa. Poslužiteljski igrač je zatim pokrenuo u Unityju poslužiteljsku instancu igre, nakon čega je klijentski igrač pokrenuo klijentsku instancu. U snimanom dvoboju su zatim provedene i testirane sljedeće funkcionalnosti:

1. Pridruživanje igri
2. Kretanje u sceni
3. Napad avatara
4. Napad AI neprijatelja
5. Podizanje štita
6. Animacije avatara
7. Animacija AI neprijatelja

Nakon što su sve funkcionalnosti isprobane, nakon otprilike od 40 do 90 sekundi, snimanje je završeno i spremljeno. Postupak je ponovljen ukupno 10 puta.

Snimljeni promet je na kraju filtriran s uvjetom da je protokol TCP te da je port 7777, čime je izolirani promet spreman za daljnju analizu. Time je promet protokola SSDP (engl. *Simple Service Discovery Protocol*) odbačen, budući da je to protokol sličan HTTP-u koji služi za oglašavanje i otkrivanje mrežnih servisa.

Snimljeni promet početnog formata .pcapng zatim je izvezen kao datoteka formata .csv (engl. comma-separated values), kako bi mogao biti obrađen u Excel-u.

3.2.1. Međudolazno vrijeme

Za analizu mrežnog prometa promatrano je međudolazno vrijeme paketa snimljenih u Wiresharku. Sučelje Wiresharka bilježi vrijeme, odnosno vremenske oznake (engl. *timestamp*) koje dodjeljuje svakom uhvaćenom paketu. Sam program ne generira vrijednosti vremenskih oznaka, već ih dohvaća iz WinPcap biblioteka i prikazuje u sučelju. Kako se vremenske oznake stvarno bilježe, ovisi o različitim operacijskim sustavima. Dok se u Unix-u time bave mrežni upravljački programi, kod Windows-a su vremenske oznake bilježene pomoću WinPcap upravljačkog programa. Konačno,

međudolazno vrijeme paketa definira se kao vrijeme koje potrebno do dolaska sljedećeg paketa, gledano od trenutka dolaska trenutno promatranog paketa.

3.2.2. Jednaka dolazna vremena

Operacijski sustavi funkcioniraju u dva osnovna načina rada rukovanja događajima: „Polling“ i „Interrupt“. Razlika između ta dva pristupa je što Polling neprekidno provjerava ulazno/izlazne uređaje bila usluga procesora potrebna ili ne, dok Interrupt pomoću ulaznih/izlaznih uređaja prekida procesor kako bi ga obavijestio da su njegove usluge potrebne. Iz toga razloga je moguće, da operacijski sustav u Interrupt pristupu isporuči više paketa odjednom kako bi se rukovanje paketima uravnotežilo. To posljedično može dovesti do slučaja da vremenske oznake mogu biti iste za više slijednih paketa, barem prema definiranoj preciznosti postupka vremenskog označavanja. Takav slučaj je zabilježen i prikazan slici 3.3, gdje se može vidjeti da paketi 7824-7826 dijele istu vremensku oznaku.

No.	Time	Source	Destination	Protocol	Length	Info
7823	39.088370	25.94.70.91	25.57.56.123	TCP	83	7777 → 54691 [PSH, ACK] Seq=150325 Ack=2906 Win=62401 Len=29
7824	39.088371	25.94.70.91	25.57.56.123	TCP	83	7777 → 54691 [PSH, ACK] Seq=150354 Ack=2906 Win=62401 Len=29
7825	39.088371	25.94.70.91	25.57.56.123	TCP	83	7777 → 54691 [PSH, ACK] Seq=150383 Ack=2906 Win=62401 Len=29
7826	39.088371	25.94.70.91	25.57.56.123	TCP	94	7777 → 54691 [PSH, ACK] Seq=150412 Ack=2906 Win=62401 Len=40
7827	39.088422	25.57.56.123	25.94.70.91	TCP	54	54691 → 7777 [ACK] Seq=2906 Ack=150452 Win=65229 Len=0

Slika 3.3. Različiti paketi s istom vremenskom oznakom

Nakon što su podaci izvezeni u Excel, međudolazna vremena su izračunata kao razlika vremenske oznake trenutnog paketa i prethodnog. Budući da je uočeno da u rjeđim slučajevima neki paketi imaju jednake vremenske oznake, za te promatrane parove su međudolazna vremena jednaka 0. Vrijednost 0 kao vrijednost međudolaznog vremena nije realna, stoga su iz međudolaznih vremena uklonjene sve takve vrijednosti, koje bi inače pogrešno utjecale na daljnju statističku analizu. Iz istog razloga, uklonjene su ekstremne vrijednosti koje bi u daljnjoj obradi podataka predstavljale šum.

3.3. Identifikacija distribucije slučajne varijable

Međudolazna vremena iz svih 10 ponavljanja spojena su u jednu .csv datoteku te predana skripti za identifikaciju distribucije slučajne varijable dorađene u sklopu

domaće zadaće kolegija Teorija prometa. Skripta je izrađena u programskom jeziku Python, verzije 2.7.14.

Varijabli klase „Distribution“ dodane su sve kontinuirane razdiobe iz knjižnice Scipy.stats. Na početku izvođenja, program čita podatke o međudolaznim vremenima pohranjene u .csv datoteku, nakon čega se kreira instanca klase Distribution. Nakon toga se poziva funkcija Fit(), u kojoj se skroz petlju iterira kroz sve kontinuirane razdiobe definirane u klasi. Zatim se poziva funkcija iz knjižnice Scipy.stats pod istim nazivom fit(). Prema definiciji, funkcija fit računa pomoću maksimiziranja funkcije log-izglednosti (engl. *log-likelihood*), s mehanizmom kažnjavanja za sve uzorke izvan dometa promatrane distribucije. Funkcija log-izglednosti daje predodžbu koliko dobro dani podaci (međudolazna vremena) zadovoljavaju parametre promatrane distribucije. Ne garantira se da su vrijednosti koje funkcija vraća globalna procjena najveće izglednosti (engl. *Maximum Likelihood Estimator – MLE*), već su potencijalno lokalno optimalna, iako postoji šansa da sveukupno optimizacija završi neuspješno. MLE se definira kao metoda procjenjivanja parametara distribucije pomoću maksimiziranja funkcije izglednosti, tako da su promatrani podaci unutar pretpostavljenog statističkog modela najviše vjerojatni. Funkcija fit() dakle vraća izračunate parametre koji bi najbolje odgovarali trenutno promatranog distribuciji za učitane podatke.

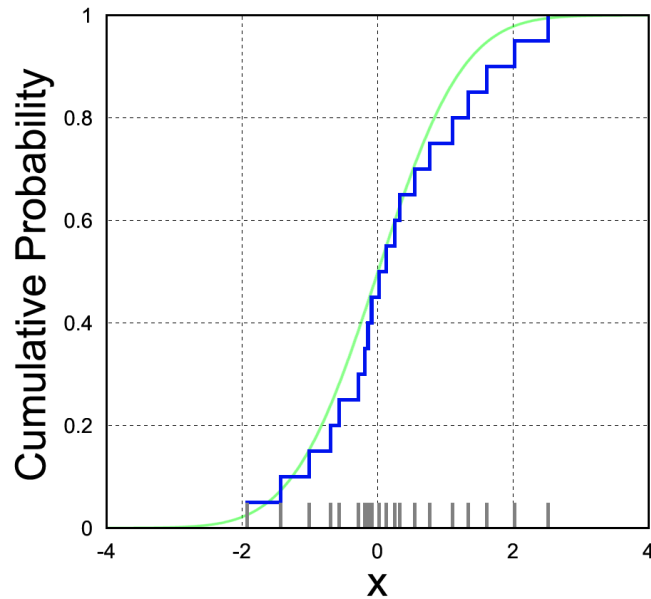
3.3.1. Kolmogorov-Smirnovljev test

Nakon što je funkcija fit() izračunala parametre za trenutnu distribuciju, radi se Kolmogorov-Smirnovljev test (skraćeno KS test). To je neparametrizirani test (ne ovisi o predodređenoj distribuciji kao primjerice T-test) za kontinuirane razdiobe, koji se koristi da bi se procijenilo dolazi li uzorak iz promatrane kontinuirane razdiobe. Temelji se na dvije početne hipoteze:

1. Nulta hipoteza (H_0): podaci dolaze iz specificirane distribucije
2. Alternativna hipoteza (H_1): barem jedna vrijednost se ne poklapa sa specificiranom razdiobom

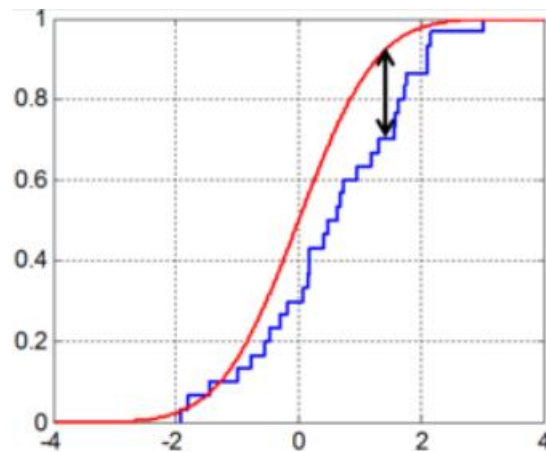
Koraci za provođenje KS test su sljedeći:

- Potrebno je kreirati empirijsku distribucijsku funkciju (skraćeno EDF), koja predstavlja funkciju kumulativne razdiobe, s empirijskom mjerom uzorka. To je step funkcija u kojoj je svaki skok odnosno korak $1/n$ od ukupno n podataka, čija je općenita vizualizacija prikazana plavom bojom na slici 3.4.



Slika 3.4. Step funkcija - EDF

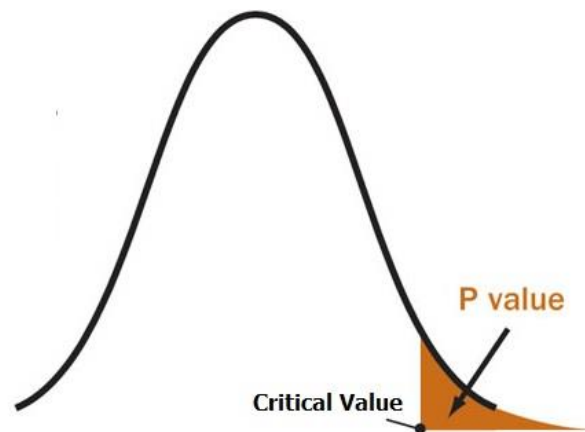
- Potrebno je definirati distribuciju s kojom se radi usporedba s EDF-om, što je u ovom slučaju trenutno promatrana distribucija iz Scipy.stats izražena preko parametara dobivenih fit() funkcijom.
- Zatim se obje distribucije prikazuju na istom grafu.
- Mjeri se vertikalna udaljenost D između dvije distribucije, prikazana na slici 3.5.



Slika 3.5. Vertikalna udaljenost D

- Računa se statistički test.
- Pronalazi se kritična vrijednost u KS tablici.
- Radi se usporedba s kritičnom vrijednosti, te ako je D manji od kritične vrijednosti, nulta hipoteza, odnosno hipoteza da se radi o istim distribucijama se prihvaća.

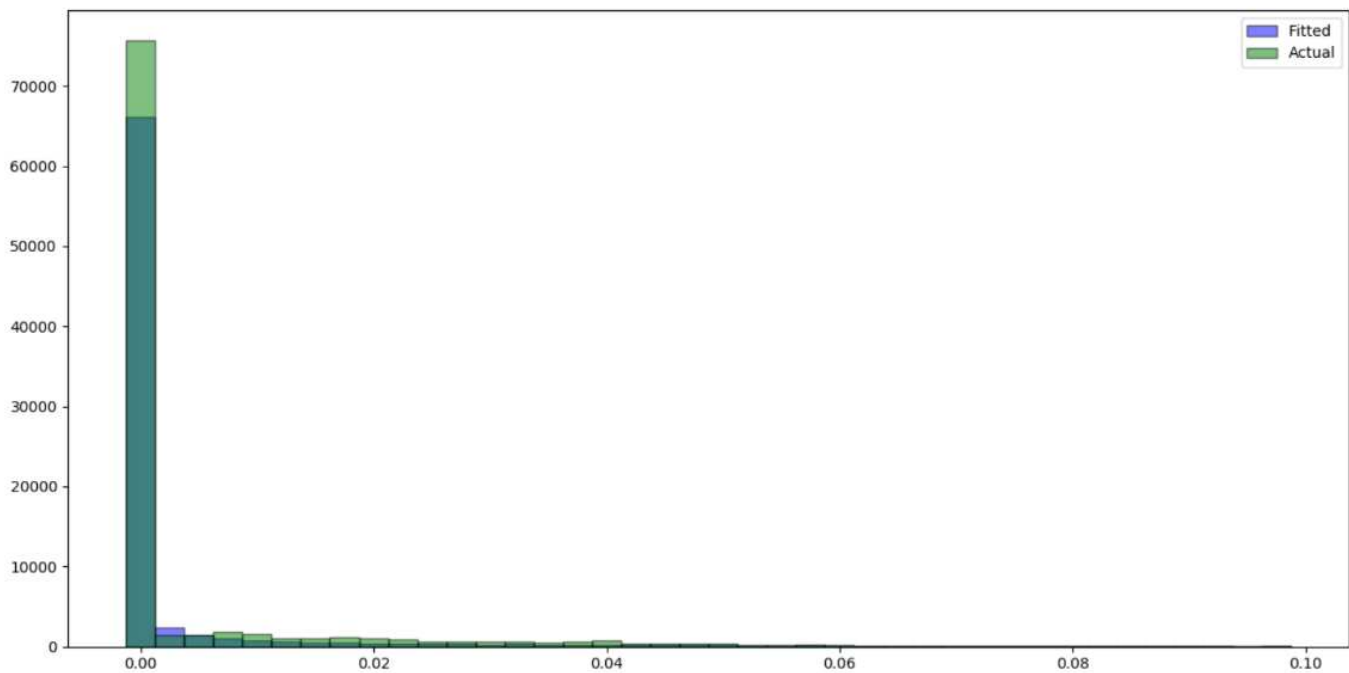
Test osim razlike D vraća i P-vrijednost (engl. *P value*), vrijednost korištenu u testiranju hipoteza u svrhu lakšeg odbijanja ili prihvatanja hipoteze. P-vrijednost se smatra dokazom protiv nulte hipoteze. Povezanost je sljedeća: što je P-vrijednost manja, dokaz je čvršći da bi se nulta hipoteza trebala odbaciti, odnosno ako je P-vrijednost velika, ne odbacujemo nultu hipotezu i uzorak pripada distribuciji. Grafički prikazano, P-vrijednost predstavlja područje „repa“ distribucije, prikazano na slici 3.6.



Slika 3.6. P-vrijednost

Na kraju iteracije svih distribucija, provedeni algoritam na temelji vrijednosti D odlučuje koja je od tih distribucija najbolja, tako da odabire distribuciju s najmanjom D vrijednošću.

Zatim se poziva funkcija Plot() pomoću kojeg se iscrtava histogram stvarnih vrijednosti i Fitted histogram. Fitted histogram se crta pomoću izračunatih parametara distribucije i nasumičnih podataka.



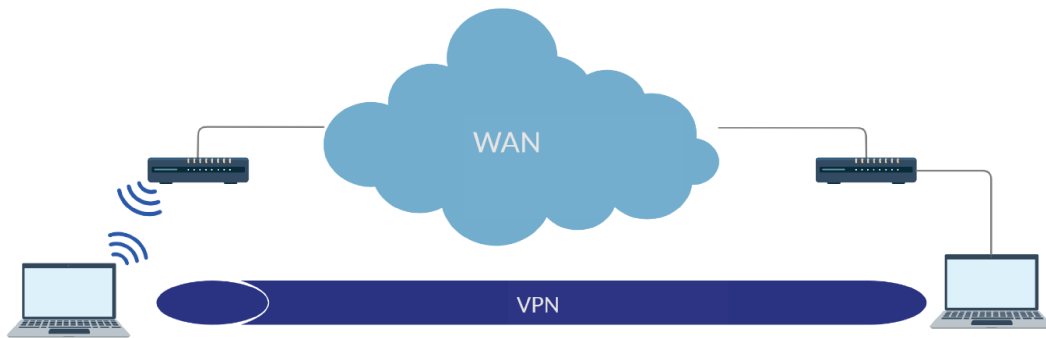
Slika 3.7. Histogram rezultata

Također se na istom gradu iscrta i histogram stvarnih vrijednosti (Actual). U histogramu, os apscise prikazuje vrijeme u sekundama, dok je na osi ordinata prikazan broj paketa.

4. Rezultati i diskusija

4.1. Eksperiment 1

U prvom eksperimentu prvo računalo bežično je spojeno na usmjeritelj, dok je drugo računalo spojeno UTP kabelom. Mrežna topologija za prvi eksperiment prikazana je na slici 4.1.



Slika 4. 1. Skica mrežne topologije u prvom eksperimentu

Informacije o snimljenom mrežnom prometu za eksperiment 1 prikazane su u tablici 4.1.

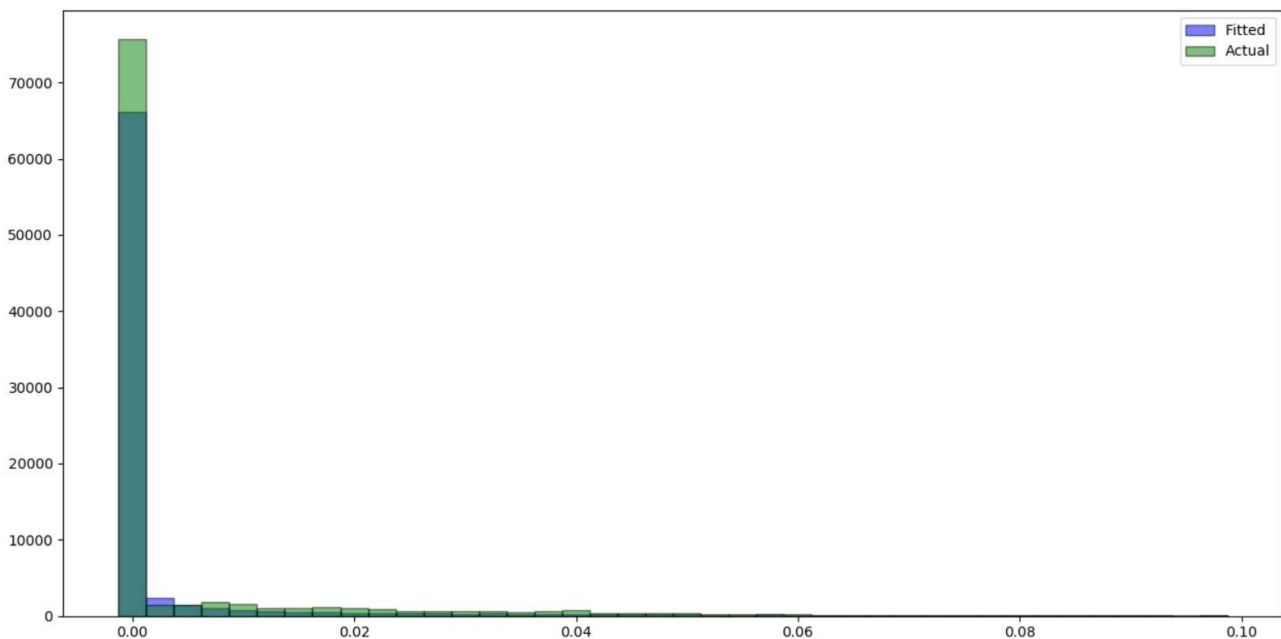
Tablica 4.1. Eksperiment 1 – informacije o mrežnom prometu

Ukupan broj snimljenih paketa	94 665
Ukupno trajanje snimke prometa [s]	558.951013
Trajanje najkraće snimke prometa [s]	37.585760
Trajanje najdulje snimke prometa [s]	90.927806
Najkraće međudolazno vrijeme [s]	0.0000010
Najdulje međudolazno vrijeme [s]	0.5008130
Srednje međudolazno vrijeme [s]	0.0051630
Najveća duljina paketa [byte]	1418
Najmanja duljina paketa [byte]	54
Srednja duljina paketa [byte]	68.5

U slučaju prvog eksperimenta, izračunato je pomoću skripte da je razdioba koja najbolje opisuje podatke o međudolaznim vremenima kontinuirana slučajna varijabla pod nazivom „exponpow“, što se može vidjeti iz ispisa konzole na slici 4.2. Dobiveni rezultati za međudolazna vremena s prikazom histograma stvarne vrijednosti i histograma generiranih vrijednosti.

```
Distribution name for time between packets: exponpow
Parameters: (0.07098549474834973, 9.999999999999997e-07, 0.07516516993918007)
```

Slika 4. 2. Ispis konzole za prvi eksperiment



Slika 4. 3. Histogrami prvog eksperimenta

Za izračunate histograme, na slici 4.3. je vidljivo da je poklapanje sa stvarnim podacima veliko, budući da se na većini slike niti ne vidi histogram generiranih vrijednosti (plava boja), nego većinu slike čini tamnozeleno, nastala preklapanjem plave boje generiranih vrijednosti i zelene boje stvarnih podataka.

4.1.1. Exponpow

Exponpow je vrsta eksponencijalne kontinuirane slučajne varijable, definirana u knjižnici Scipy.stats programskog jezika Python. Definirana je u standardnom eksponencijalnom formatu uz dodatne parametre oblika koji upotpunjuju njezinu

specifikaciju. Zapis funkcije gustoće od exponpow, gdje je x veći ili jednak 0, a parametar oblika b veći od 0, prikazan je na slici 4.4.

$$f(x, b) = bx^{b-1} \exp(1 + x^b - \exp(x^b))$$

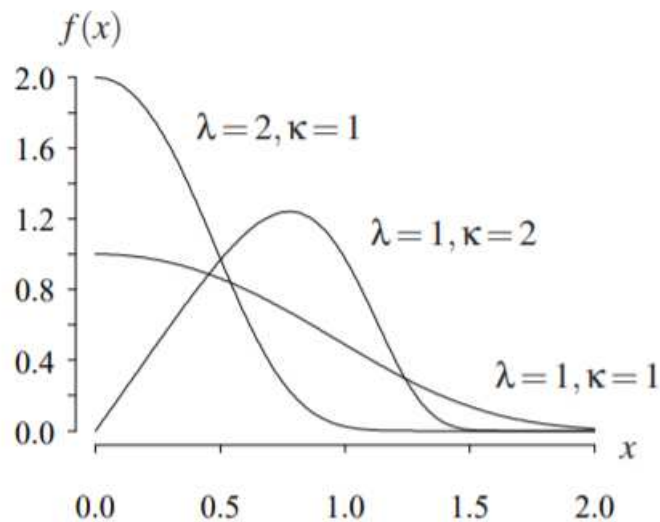
Slika 4. 4. Funkcija gustoće - standardizirani oblik

Ako funkciju gustoće zapišemo preko λ , κ , gdje je λ parametar razmjera, a κ pozitivan parametar oblika, dobiva se zapis prikazan na sljedećoj slici.

$$f(x) = \left(e^{1-e^{\lambda x^\kappa}} \right) e^{\lambda x^\kappa} \lambda \kappa x^{\kappa-1} \quad x > 0.$$

Slika 4.5. Funkcija gustoće exponpow

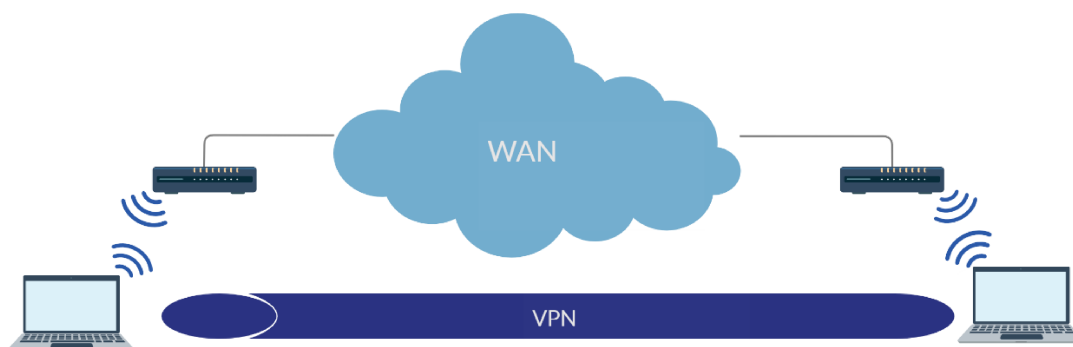
Sada primjerice, ako mijenjamo parametre λ , κ , može se vidjeti kako oni utječu na oblik krivulje funkcije gustoće.



Slika 4.6. Funkcija gustoće u ovisnosti o parametrima

4.2. Eksperiment 2

U drugom eksperimentu oba računala su bežično spojena na različite usmjeritelje, te je mrežna topologija za drugi eksperiment prikazana na slici 4.7.



Slika 4. 7. Mrežna topologija za drugi eksperiment - oba bežično

Informacije o snimljenom mrežnom prometu za eksperiment 2 prikazane su u tablici 4.2.

Tablica 4.2. Eksperiment 2 – informacije o mrežnom prometu

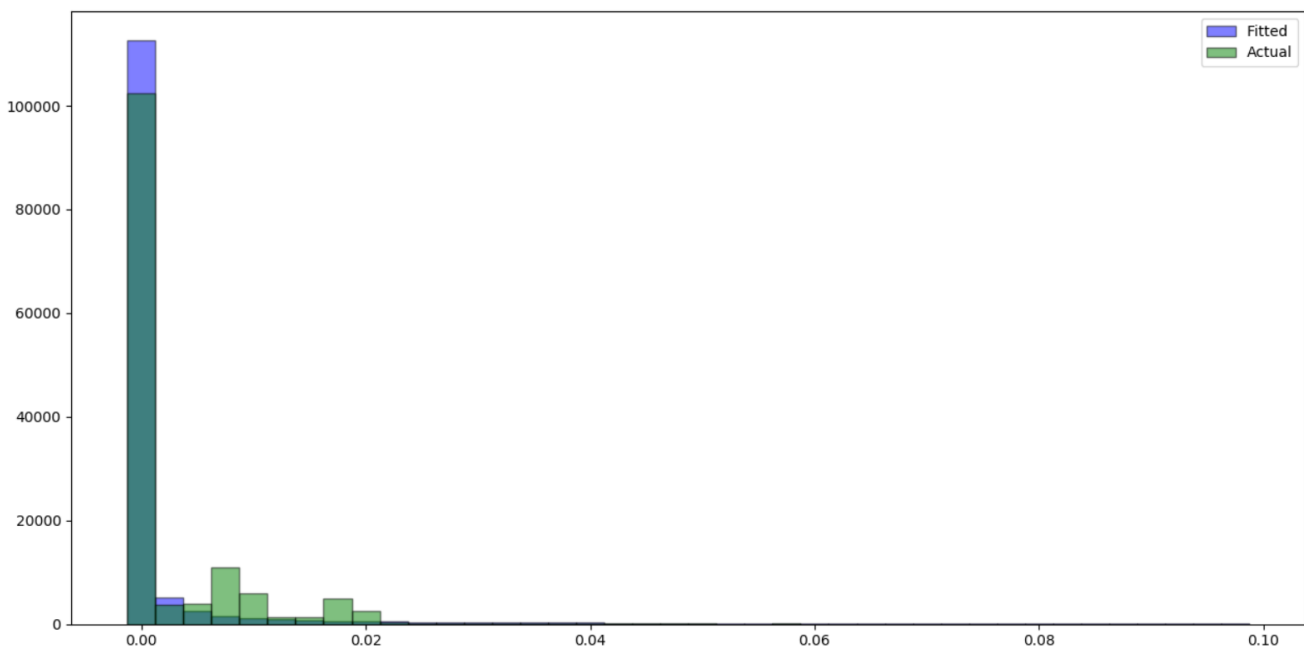
Ukupan broj snimljenih paketa	139040
Ukupno trajanje snimke prometa [s]	581.300424
Trajanje najkraće snimke prometa [s]	41.036980
Trajanje najdulje snimke prometa [s]	89.098056
Najkraće međudolazno vrijeme [s]	0.0000010
Najdulje međudolazno vrijeme [s]	0.7709780
Srednje međudolazno vrijeme [s]	0.0000035
Najveća duljina paketa [byte]	1418
Najmanja duljina paketa [byte]	54
Srednja duljina paketa [byte]	58

Kao rezultat drugog eksperimenta, pomoću skripte je dobiveno da je razdioba po kojoj se međudolazna vremena najviše ravnaju, slučajna varijabla pod nazivom „genextreme“ definirana u knjižnici Scipy.stats. Ispis konzole nakon izvršenja skripte

za međudolazna vremena dobivena u drugom eksperimentu prikazan je na slici 4.8, dok su histogrami stvarnih i generiranih vrijednosti prikazan na slici 4.9.

```
Distribution name for time between packets: genextreme  
Parameters: (-3.0114406833634915, 2.338981055732463e-05, 6.797601161977299e-05)
```

Slika 4.8. Ispis konzole za drugi eksperiment



Slika 4.9. Histogrami drugog eksperimenta

Na slici 4.9. može se vidjeti kako je poklapanje manjih međudolaznih vremena s generiranim vrijednostima poprilično dobro, dok se za vrijednosti većih međudolaznih vremena, očituju veće razlike između stvarnih vrijednosti i generiranih vrijednosti.

4.2.1. Genextreme

Genextreme predstavlja generaliziranu distribuciju ekstremnih vrijednosti (engl. generalized extreme value – GEV). To je kontinuirana distribucija vjerojatnosti razvijena unutar teorije ekstremnih vrijednosti. Teorija ekstremnih vrijednosti ili analiza ekstremnih vrijednosti je grana statistike koja se bavi ekstremnim odstupanjima od

medijana vjerojatnosti raspodjele. Nastoji iz uređenog uzorka dane slučajne varijable procijeniti vjerojatnost događaja ekstremnijih od bilo kojih prethodno promatranih. Analiza ekstremnih vrijednosti koristi se u raznim disciplinama, kao što građevinsko inženjerstvo, financije, predviđanje prometa i geološko inženjerstvo. Takva vrsta analize bi se primjerice mogla koristiti u hidrologiji, za izračun procjene vjerojatnosti neobično velikih poplava (primjerice stogodišnja poplava).

Funkcija gustoće za genextreme u standardiziranom obliku prikazana je na slici 4.10.

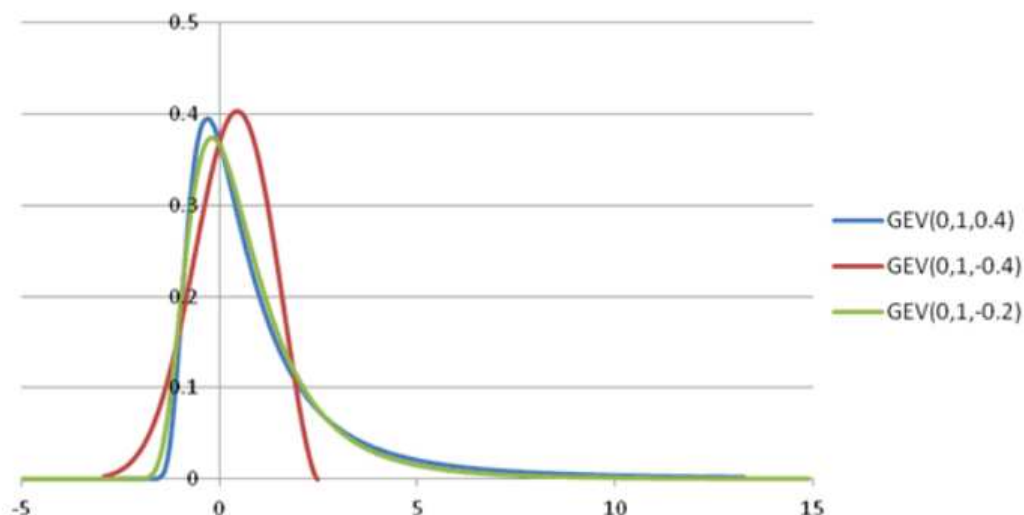
$$f(x, c) = \begin{cases} \exp(-\exp(-x)) \exp(-x) & \text{for } c = 0 \\ \exp(-(1 - cx)^{1/c})(1 - cx)^{1/c-1} & \text{for } x \leq 1/c, c > 0 \end{cases}$$

Slika 4.10. Funkcija gustoće generalizirane distribucije ekstremnih vrijednosti

U ovom obliku, funkcija gustoće je definirana da prima parametar c , parametar koji određuje oblik funkcije.

Generalizirana distribucija ekstremnih vrijednosti, poznata je još pod nazivom Fisher-Tippetova distribucija, fleksibilan je model s tri parametra koji kombinira Gumbelovu, Fréchetovu i Weibullovu distribuciju maksimalne ekstremne vrijednosti. Ovisno o tome je li parametar c manji, veći ili jednak nuli, GEV razdioba bit će ekvivalentna jednoj od tri kombinirane distribucije. Prema teoriji ekstremnih vrijednosti, raspodjela GEV-a je granična raspodjela normaliziranih maksimuma niza neovisnih i identično raspodijeljenih slučajnih varijabli.

Pa tako, ako se razdiobu GEV promatra u obliku $GEV(a, b, c)$, na slici 4.11. se očituje kako mijenjanje pojedinih parametara utječe na oblik funkcije.



Slika 4.11. Generalizirana distribucija ekstremnih vrijednosti u ovisnosti o parametrima

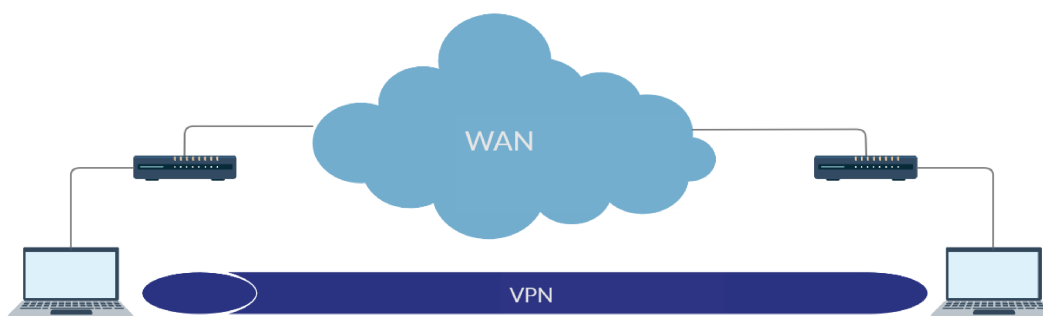
Pomoću parametra a , kontrolira se lokacija krivulje funkcije, dok parametar b predstavlja parametar razmjera. Već spomenuti parametar c određuje oblik funkcije.

Izračunato je da je parametar oblika funkcije c za ovaj eksperiment iznosa -3.01144 , što je uzrokovalo da histogram prati strmi oblik. Parametri lokacije i razmjera su malih ali pozitivnih iznosa, čime je iscrtani histogram blago pomaknut u desno, kako bi počeo od vrijednosti većima od 0, budući da su nule izbačene iz izračuna.

Razlog zašto je ova distribucija odabrana kao distribucija s najboljim poklapanjem s podacima međudolaznih vremena je taj, što se u mjerenjima međudolaznih vremena nalazi jako puno paketa s međudolaznim vremenima blizu 0. Iz tog razloga broj paketa s međudolaznim vremenima blizu 0, uvelike odstupaju od svih ostalih vrijednosti, što je na kraju očitano kao ekstremna vrijednost, čime je najbolja distribucija upravo generalizirana distribucija ekstremne vrijednosti.

4.3. Eksperiment 3

U trećem i zadnjem eksperimentu oba računala su žično spojena na različite usmjeritelje pomoću UTP kabela, te je mrežna topologija za treći eksperiment prikazana na slici 4.12.



Slika 4.12. Mrežna topologija trećeg eksperimenta

Informacije o snimljenom mrežnom prometu za eksperiment 3 prikazane su u tablici 4.3.

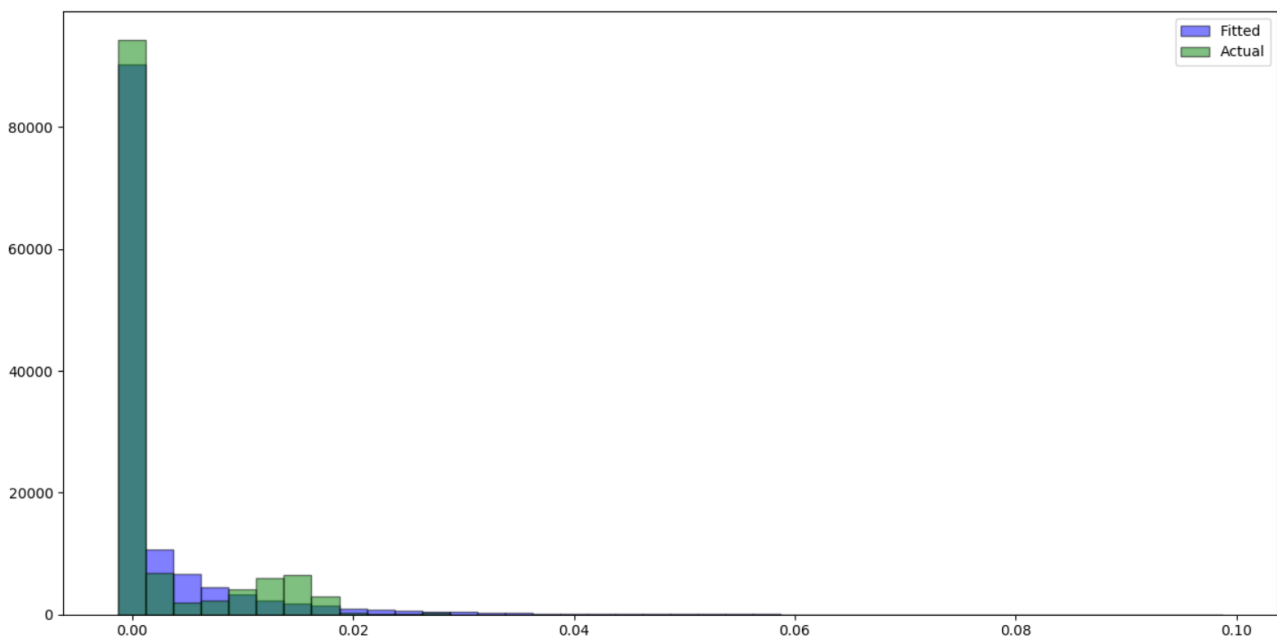
Tablica 4.3. Eksperiment 3 – informacije o mrežnom prometu

Ukupan broj snimljenih paketa	126 851
Ukupno trajanje snimke prometa [s]	440.018246
Trajanje najkraće snimke prometa [s]	33.8871380
Trajanje najdulje snimke prometa [s]	59.6202120
Najkraće međudolazno vrijeme [s]	0.0000010
Najdulje međudolazno vrijeme [s]	0.5020240
Srednje međudolazno vrijeme [s]	0.0339215
Najveća duljina paketa [byte]	1418
Najmanja duljina paketa [byte]	54
Srednja duljina paketa [byte]	58

Pomoću skripte za identifikaciju distribucije slučajne varijable je dobiveno, da se međudolazna vremena najviše ravnaju po distribuciji u Scipy.stats knjižnici definiranoj kao „fatiguelife“. Na slici 4.13. prikazan je ispis konzole s nazivom distribucije i njenim parametrima, dok je na slici 4.14. iscrtan rezultat u obliku dva histograma, stvarnih i generiranih vrijednosti.

```
Distribution name for time between packets: fatiguelife  
Parameters: (5.6242760533067475, 2.7614363721748883e-07, 0.00019302247961368904)
```

Slika 4.13. Ispis konzole drugog eksperimenta



Slika 4.13. Histogrami trećeg eksperimenta

Na slici 4.14. može se vidjeti kako je poklapanje manjih međudolaznih vremena s generiranim vrijednostima za manja međudolazna vremena poprilično dobro, dok se u se kod većih međudolaznih vremena (između 0.01 i 0.02 sekunde) može primijetiti kako stvarni podaci prate trend rasta paketa prema većem međudolaznom vremenu, dok generirane vrijednosti prate trend pada paketa prema većem međudolaznom vremenu.

4.3.1. FatigueLife

Distribucija pod nazivom fatigue life poznata još kao Birnbaum-Saundersova distribucija. To je distribucija koja se uvelike koristi za modeliranje vremena ispada u promatranom sustavu.

Funkcija gustoće za spomenutu distribuciju u standardiziranom obliku prikazana je na slici 4.15:

$$f(x, c) = \frac{x + 1}{2c\sqrt{2\pi x^3}} \exp\left(-\frac{(x - 1)^2}{2xc^2}\right)$$

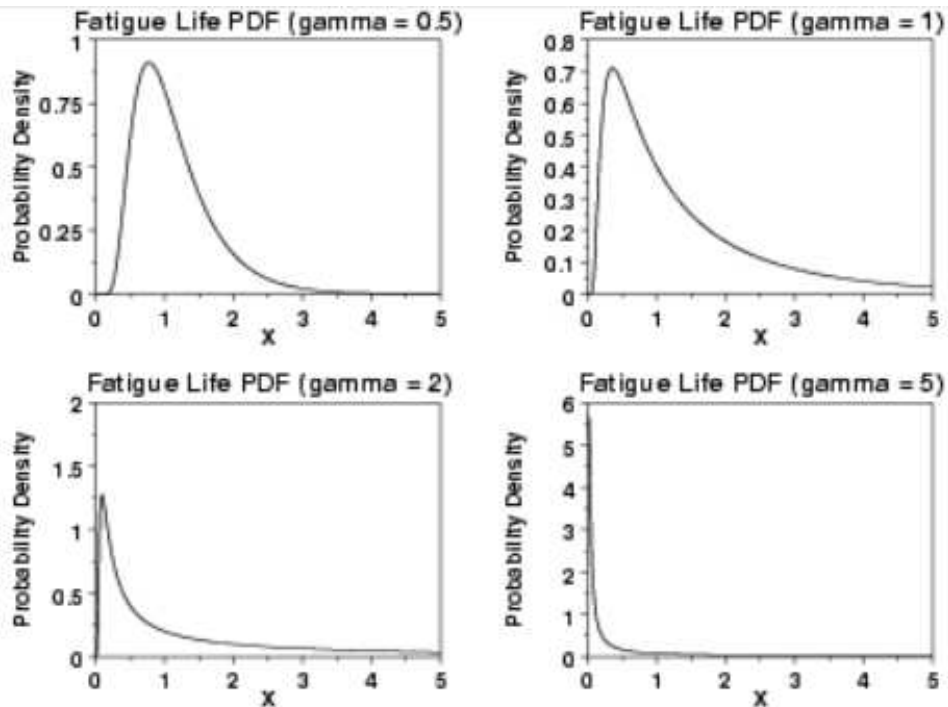
Slika 4.14. Funkcija gustoće fatigueLife, standardizirani oblik

gdje je $x \geq 0$ i $c > 0$ (c predstavlja parametar oblika funkcije), dok je generalizirani oblik funkcije gustoće, prikazan pomoću parametara γ , μ i β , prikazan na slici 4.16:

$$f(x) = \frac{\sqrt{\frac{x-\mu}{\beta}} + \sqrt{\frac{\beta}{x-\mu}}}{2\gamma(x-\mu)} \phi\left(\frac{\sqrt{\frac{x-\mu}{\beta}} - \sqrt{\frac{\beta}{x-\mu}}}{\gamma}\right) \quad x > \mu; \gamma, \beta > 0$$

Slika 4.15. Funkcija gustoće fatigueLife, parametrizirani oblik

gdje je pomoću parametra γ numerički definiran oblik funkcije, pomoću parametra μ izražen lokacijski parametar, a parametrom β izražen parametar razmjera funkcije. Funkcija također ima uključen $\Phi()$, koji predstavlja funkciju gustoće standardne normalne distribucije. Na slici 4.17. može se vidjeti kako povećavanjem parametra oblika γ , funkcija mijenja oblik.



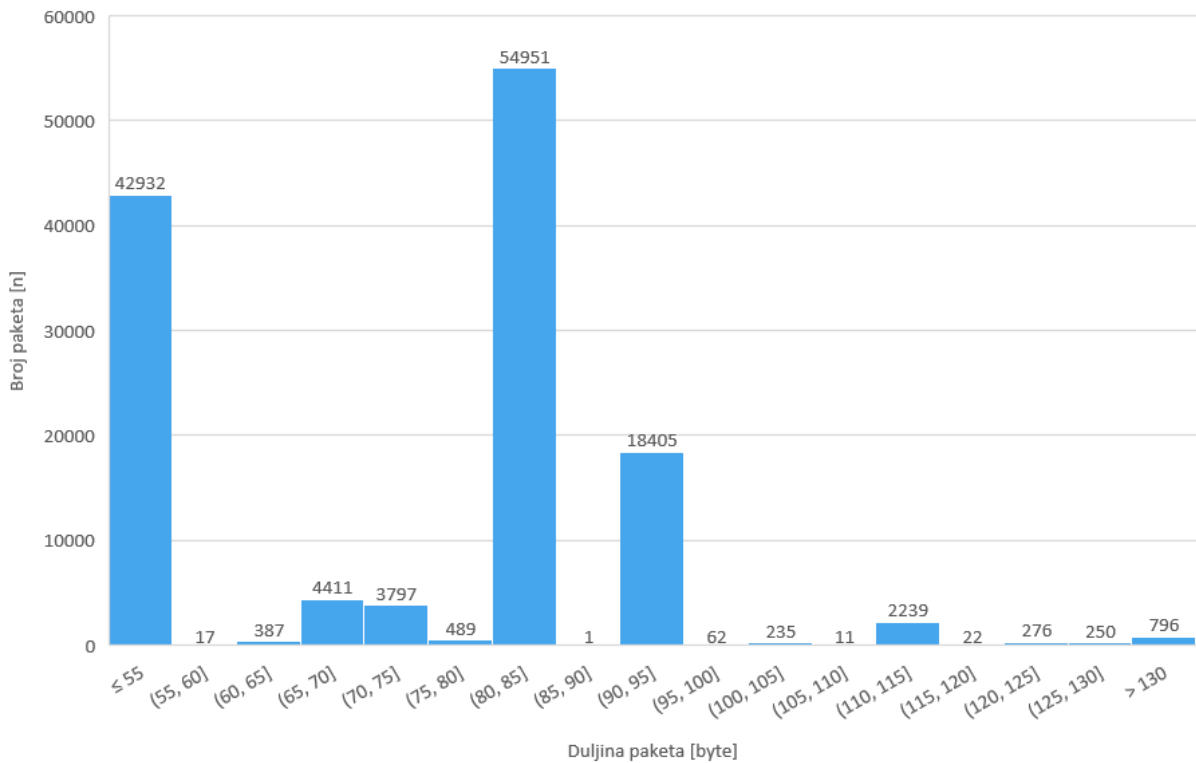
Slika 4.16. Ovisnost fatigueLife o parametru gama

Izračunato je da je za treći eksperiment $\gamma = 5.62427$, što uzrokuje da histogram ima oštar pad na osi ordinata, odnosno najveći broj paketa ima međudolazno vrijeme malo veće od 0, nakon čega broj paketa za sva veća međudolazna vremena opada. Parametar lokacije je 2.76×10^{-7} , što se očituje kao blagi pomak, budući da su vrijednosti 0 izbačene iz izračuna. Parametar razmjera β , koji utječe na razvučenost u smjeru apscisa iznosi 0.00019, što uzrokuje da vrijednosti histograma nisu razvučene, nego prate više skupljeni oblik.

4.4. Duljine paketa

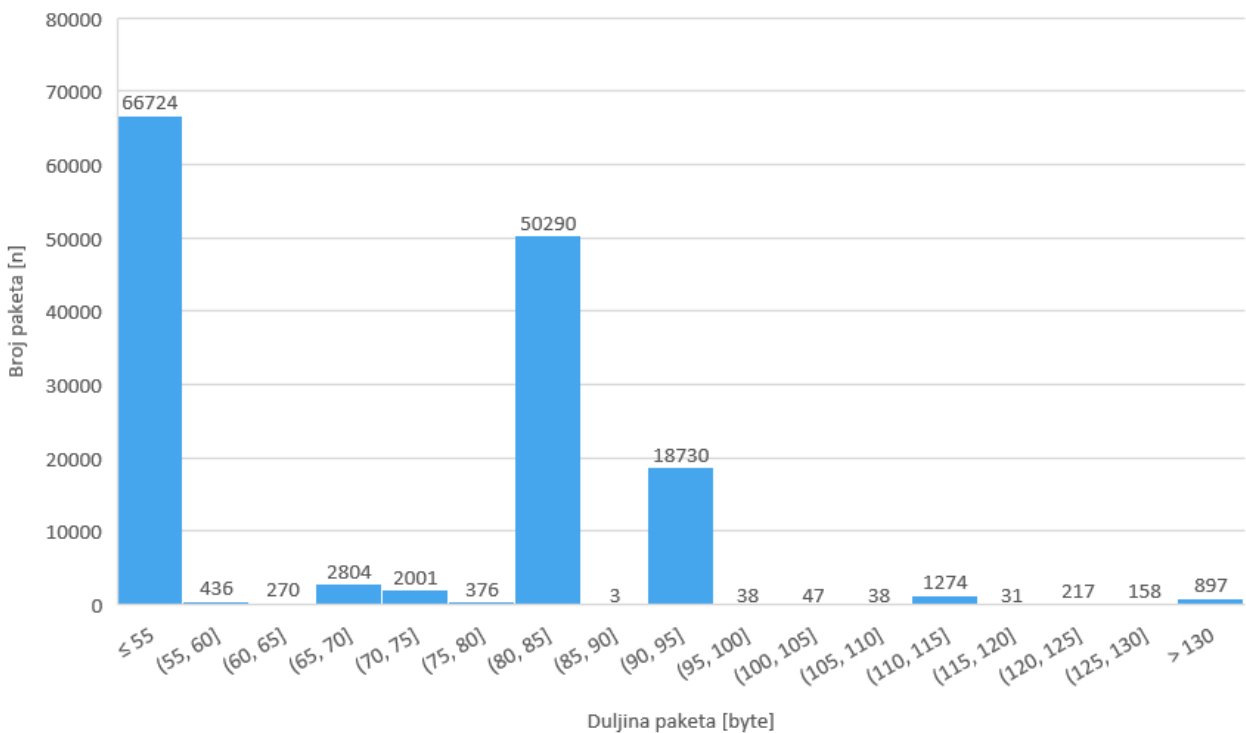
Osim međudolaznih vremena promatrana je i duljina paketa u mrežnom prometu. Nakon što je mrežni promet snimljen, za pojedini eksperiment, rezultati od 10 uzastopnih mjerenja prikupljeni su u pojedinu Excel datoteku. Zatim su pomoću alata u Excelu kreirani histogrami s prikazom duljine paketa.

Za eksperiment 1, slučaj kad je jedan korisnik na usmjeritelj spojen bežično, a drugi žično, kreiran je histogram prikazan na slici 4.18.



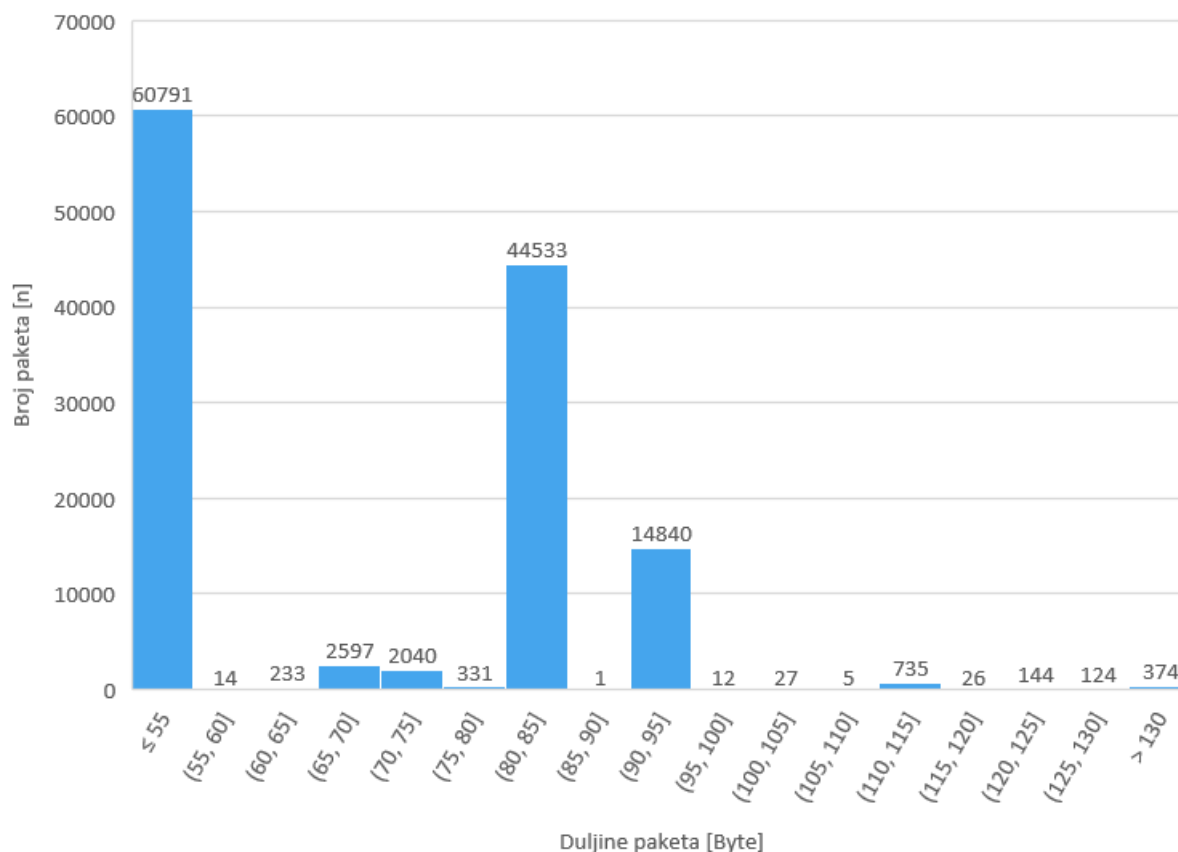
Slika 4.17. Histogram duljine paketa za prvi eksperiment

Za eksperiment 2, u kojem su oba korisnika bila spojena bežično na svoje usmjeritelje, kreiran je histogram prikazan na slici 4.19.



Slika 4.18. Histogram duljine paketa za drugi eksperiment

U eksperimentu 3, oba korisnika su bila spojena žično na svoje usmjeritelje, te je histogram duljine paketa za taj slučaj prikazan na slici 4.20.



Slika 4.20. Histogram duljine paketa za treći eksperiment

Kod sva tri eksperimenta može se primijetiti kako se ponavljaju iste veličine paketa slični broj puta. Radi se o šifriranim TCP paketima, čiji sadržaj ne možemo potvrditi. No proučavanjem snimljenog prometa u Wiresharku, uočeno je da paketa duljine 54 bajta ima najviše, te se radi o paketima s oznakom ACK, što bi značilo da se radi o slanju potvrde prethodno primljenog segmenta poslanog od poslužitelja. Ostale poruke koje se šalju su podaci u TCP segmentima čije su zastavice na mjestima PUSH postavljene na 1. Tako se traži slanje spremljenih podataka u međuspremniku, umjesto da se čeka prvo da spremnik popuni pa pošalje. Time aplikacija informira TCP da podaci trebaju odmah biti poslani, a PSH zastavica postavljena na vrijednost 1, informira primatelja da primljeni podaci trebaju odmah biti prosljeđeni aplikaciji.

Zaključak

Postojeća umrežena višekorisnička videoigra „Wizard wars“, kreirana na kolegiju Diplomski projekt, poboljšana je dodavanjem novih animacija te dodavanjem samostalno izrađene čarolije napada. Čarolija napada, skicirana u Photoshopu i animirana u Unityju, poboljšava vizualni dojam videoigre, čineći igru više fantastičnu nego realističnu. Dodavanjem nove animacije napada, mijenja se međusobna interakcija igrača te videoigra dobiva na težini.

Dovršeni produkt zatim je proveden kroz analizu mrežnog prometa prilikom interakcije dvaju korisnika. Provedena su tri eksperimenta s deset ponavljanja za korisnike spojene u privatnu virtualnu mrežu pomoću LogMeIn Hamachi-ja. Eksperimenti su se međusobno razlikovali po načinu spajanja korisnika na svoje usmjeritelje razlikujući tri slučaja: prvi slučaj kad je jedan korisnik spojen bežično, a drugi žično na usmjeritelj, drugi slučaj kad su oba korisnika spojena bežično i treći slučaj kad su oba korisnika spojena žično. Pomoću skripte, identificirane su tri kontinuirane razdiobe za međudolazna vremena: eksponencijalna kontinuirana razdioba (definirana u Scipy.stats knjižnici kao `exponpow`), generalizirana distribucija ekstremnih vrijednosti i Birnbaum-Saundersova distribucija (`fatigueLife` u Scipy.stats). Razdiobe su vraćene kao rješenje s pripadajućim parametrima oblika, lokacije i razmjera, koji iako međusobno različiti, uspoređujući dobivene histograme daju slična rješenja. Sva tri eksperimenta prate sličan uzorak histograma, gdje se za niske vrijednosti međudolaznih vremena očituje najveći broj paketa, dok povećanjem međudolaznih vremena se primjećuje drastičan pad broja paketa, i to pad sličan eksponencijalnoj funkciji. U umreženoj komunikaciji, takvo ponašanje je očekivano, budući da su niska međudolazna vremena preduvjet za kontinuiranu komunikaciju. Što se tiče duljina paketa iz snimljenih eksperimenata, uočeno je da se nekoliko veličina paketa uzastopno ponavlja. Promatranjem mrežnog prometa, može se primijetiti kako se koristi mehanizam koji ne čeka punjenje cijelog TCP paketa, nego kako bi komunikacija mogla glađe teći se koristi slanje više manjih paketa.

Literatura

- [1] Maček D., Šimunović D., *Izrada višekorisničke video igre*. Diplomski projekt. Sveučilište u Zagrebu Fakultet elektrotehnike i računarstva, 2020.
- [2] Jared Koh, 28. kolovoza, 2018., *Avatar Element Animation*, <https://www.youtube.com/watch?v=jU44SJwAKHU>, pristupljeno 23. ožujka 2020.
- [3] Lesley Yarbrough, 10. veljače, 2020., *How to Install & Use Photoshop Brushes*, <https://creativemarket.com/blog/how-to-install-use-photoshop-brushes>, pristupljeno 24. ožujka 2020.
- [4] Unity Technologies, 10. siječnja 2020., ParticleSystem, *Scripting API: ParticleSystem–Unity*, <https://docs.unity3d.com/ScriptReference/ParticleSystem.html>, pristupljeno 25. ožujka 2020.
- [5] Unity Technologies, 15. travnja 2020., Creating humanoid animation, *Creating humanoid animation - Unity – Manual*, <https://docs.unity3d.com/2018.3/Documentation/Manual/TimelineWorkflowHumanoidAnimation.html>, pristupljeno 2. travnja 2020.
- [6] 17. ožujka 2020., *Packet analyzer*, https://en.wikipedia.org/wiki/Packet_analyzer, pristupljeno 15. travnja 2020.
- [7] Awake, *Network Traffic Analysis*, <https://awakesecurity.com/glossary/network-traffic-analysis/>, pristupljeno 18. travnja 2020.
- [8] 9. rujna 2020., CARNet, *Analiza mrežnog prometa*, <https://www.cis.hr/www.edicija/LinkedDocuments/CCERT-PUBDOC-2004-09-90.pdf>, pristupljeno 23. travnja, 2020.
- [9] 29. travnja 2020., *LogMeIn Hamachi*, <https://www.cis.hr/www.edicija/LinkedDocuments/CCERT-PUBDOC-2004-09-90.pdf>, pristupljeno 30. travnja 2020.
- [10] Ivan Belcic, 14. svibnja 2020., *What is a Sniffer, and How Can I Protect Against Sniffing?*, <https://www.avast.com/c-sniffer>, pristupljeno 20. svibnja 2020.
- [11] Elvidence, 25. rujna 2015., *Understanding time stamps in Packet Capture Data (.pcap) files*, <https://www.elvidence.com.au/understanding-time-stamps-in-packet-capture-data-pcap-files/>, pristupljeno 28. svibnja 2020.

- [12] The SciPy community, 19. prosinca 2019., *scipy.stats.rv_continuous.fit*, https://docs.scipy.org/doc/scipy/reference/generated/scipy.stats.rv_continuous.fit.html, pristupljeno 2. lipnja 2020.
- [13] Myung I.J., Tutorial on maximum likelihood estimation, *Journal of mathematical psychology*, 47/03, 2003., 90-100
- [14] Bojana Dalbelo Bašić, Jan Šnajder, Algoritam maksimizacije očekivanja, 2013., *Strojno učenje*, <https://www.fer.unizg.hr/download/repository/SU-14-AlgoritamMaksimizacijeOcekivanja.pdf>, pristupljeno 4. lipnja 2020.
- [15] Vose Software, 2017., *Generalized extreme value distribution*, <https://www.vosesoftware.com/riskwiki/Generalizedextremevaluedistribution.php>, pristupljeno 11. lipnja 2020.
- [16] Nist Sematech, Kolmogorov-Smirnov Goodness-of-Fit Test, *Engineering Statistics Handbook*, <https://www.itl.nist.gov/div898/handbook/eda/section3/eda35g.htm>, pristupljeno 6. lipnja 2020.
- [17] Stephanie Glen, 5. srpnja 2016., Kolmogorov-Smirnov Goodness of Fit Test, *Statistics How To*, <https://www.statisticshowto.com/kolmogorov-smirnov-test/>, pristupljeno 6. lipnja 2020.
- [18] Teorija prometa, Slučajne varijable. Razdiobe slučajne varijable, *Teorija prometa*, Sveučilište u Zagrebu Fakultet elektrotehnike i računarstva, 2019.
- [19] Stephanie Glen, 13. siječnja 2014., P-Value in Statistical Hypothesis Tests: What is it?, *Statistics How To*, <https://www.statisticshowto.com/p-value/>, pristupljeno 7. lipnja 2020.
- [20] The SciPy community, 19. prosinca 2020., Statistical functions (scipy.stats), *Scipy.org*, <https://docs.scipy.org/doc/scipy/reference/stats.html>, pristupljeno 8. lipnja 2020.
- [21] Bali T.G., The generalized extreme value distribution, *Economics Letters*, 79/03, 2003., 423-427
- [22] Nist Sematech, Fatigue life (Birnbaum-Saunders), *Engineering Statistics Handbook*, <https://www.itl.nist.gov/div898/handbook/eda/section3/eda35g.htm>, pristupljeno 9. lipnja 2020.

[23] Nist Sematech, Fatigue Life Distribution, *Engineering Statistics Handbook*, <http://atomic.phys.uni-sofia.bg/local/nist-e-handbook/e-handbook/eda/section3/eda366a.htm>, pristupljeno 9. lipnja 2020.

[24] Arous G.B., Voiculescu D. V., Free extreme values, *The annals of probability*, 34/06, 2006., 2037–2059

[25] Relja A., *Neki statistički aspekti prepoznavanja motiva*, Diplomski rad, Sveučilište u Zagrebu Prirodoslovno-matematički fakultet, srpanj 2014.

[26] Maček D., *Analiza mrežnog prometa pri interakciji korisnika s računalno generiranim likovima u 3D videoigri*. Diplomski rad. Sveučilište u Zagrebu Fakultet elektrotehnike i računarstva, 2020.

Sažetak

U ovom diplomskom radu višekorisnička umrežena videoigra „Wizard wars“ nadograđena je kreiranjem novih animacija pomoću Unityja te nove čarolije napada korištenjem Photoshopa. Nakon toga provedena je analiza mrežnog prometa pri interakciji dvaju korisnika u virtualnoj privatnoj mreži, gdje je mrežni promet zabilježen pomoću analizatora paketa Wireshark. Provedena su tri eksperimenta međusobno različita po mrežnoj konfiguraciji, odnosno spajanju korisnika žično ili bežično na svoje usmjeritelje. Iz snimljenog TCP prometa izdvojena su međudolazna vremena i duljine paketa za daljnju analizu. Nadalje je provedena identifikacija slučajne varijable, te su identificirane razdiobe za međudolazna vremena, te slikovito prikazani histogrami izračunatih distribucija i stvarnih podataka. Nad duljinama paketa provedena je usporedba između dobivenih histograma za tri provedena eksperimenta.

Ključne riječi: višekorisnička videoigra, analiza mrežnog prometa, distribucija slučajne varijable, međudolazno vrijeme, duljine paketa

Summary

In this Master's thesis, existing multiplayer network videogame „Wizard wars“ has been upgraded with new animations using Unity and new magic attack using Photoshop (Free trial). After that, an analysis of network traffic was performed after capturing network traffic using packet analyzer during the interaction of two users in a virtual private network. Three experiments with different network configurations were performed, differentiating from each other with the way of connecting wirelessly or wired to their routers. For further analysis, the interarrival time and packet length were extracted from captured TCP traffic. From the processed interarrival time for each experiment, identification of a random variable was conducted and plotted to show two histograms, one showing fitted values, and the other actual values. For package length, the comparison was made between histograms for all three performed experiments.

Key words: multiplayer videogame, network traffic analysis, random variable distribution, interarrival time, packet length