

Univerza v Ljubljani
Fakulteta za računalništvo in informatiko

Nives Bricman

**Zasnova fizičnega omrežja za potrebe novega
ponudnika internetnih storitev**

DIPLOMSKO DELO
UNIVERZITETNI ŠTUDIJSKI PROGRAM PRVE STOPNJE
RAČUNALNIŠTVO IN INFORMATIKA

prof. dr. Miha Mraz
MENTOR

Ljubljana, 2018

© 2018, Univerza v Ljubljani, Fakulteta za računalništvo in informatiko

Rezultati diplomskega dela so intelektualna lastnina Fakultete za računalništvo in informatiko Univerze v Ljubljani. Za objavljanje ali izkoriščanje rezultatov diplomskega dela je potrebno pisno soglasje Fakultete za računalništvo in informatiko ter mentorja.

Tematika naloge:

Kandidatka naj v svojem delu izvede osnovno demografsko analizo vzorčnega izbranega geografskega področja. Na osnovi slednje naj vzpostavi model fizičnega računalniškega omrežja, ki bo pokrivalo to področje za potrebe novega ponudnika internetnih storitev. Model omrežja naj vzpostavi v razvojnem okolju OMNeT++ in opravi simulacije za potrebe zmogljivostnih analiz načrtovanega omrežja.

IZJAVA O AVTORSTVU DIPLOMSKEGA DELA

Spodaj podpisana izjavljam, da sem avtor dela, da slednje ne vsebuje materiala, ki bi ga kdorkoli predhodno že objavil ali oddal v obravnavo za pridobitev naziva na univerzi ali drugem visokošolskem zavodu, razen v primerih kjer so navedeni viri.

S svojim podpisom zagotavljam, da:

- sem delo izdelala samostojno pod mentorstvom prof. dr. Mihe Mraza,
- so elektronska oblika dela, naslov (slov., angl.), povzetek (slov., angl.) ter ključne besede (slov., angl.) identični s tiskano obliko in
- soglašam z javno objavo elektronske oblike dela v zbirki "Dela FRI".

— Nives Bricman, Ljubljana, februar 2018.

Univerza v Ljubljani
Fakulteta za računalništvo in informatiko

Nives Bricman

Zasnova fizičnega omrežja za potrebe novega ponudnika internetnih storitev

POVZETEK

V diplomskem delu obravnavamo problem postavitve poenostavljenega modela fizičnega omrežja na območju Upravne enote Slovenj Gradec, ki jo sestavljata občini Slovenj Gradec in Mislinja. Na začetku diplomskega dela najprej opišemo lastnosti analitičnega in simulacijskega pristopa. Slednjega uporabimo kot metodo za reševanje obravnavanega problema.

Nadaljujemo z analizo opazovanega območja občin Slovenj Gradec in Mislinja, s pomočjo katere določimo grobe ocene števila bivanjskih in poslovnih uporabnikov, ki bodo priključeni v naše omrežje. V nadaljevanju določimo oceno količine podatkov na posamezen tip uporabnika. Osredotočimo se na količino prometa poslanega do uporabnika in količino prometa generiranega s strani uporabnika. Analizi uporabnikov sledi združitev naselij v območja, ki jih uporabimo kot začetne generatorje prometa našega modela omrežja.

Odprtokodno simulacijsko orodje OMNeT++ uporabimo za implementacijo gradnikov modela omrežja, izvedbo dejanske simulacije in njene analize. Po izvedbi simulacijskih eksperimentov podrobneje analiziramo in predstavimo rezultate metrik čakalnih in potovalnih časov ter uporabnostnih faktorjev povezav.

Ključne besede: model računalniškega omrežja, OMNeT++, računalniške simulacije

University of Ljubljana
Faculty of Computer and Information Science

Nives Bricman

Design of the wired network for an internet service provider

ABSTRACT

In this thesis we discuss the problem of a simplified physical organization of network on area of Administrative Office of Slovenj Gradec, which consists of municipalities of Slovenj Gradec and Mislinja. First, we describe the characteristics of analytical and simulation approach. The last one is used as the method for solving the discussed problem.

Next is the analysis of the observed area of municipalities Slovenj Gradec and Mislinja, which helps us to evaluate roughly number of household and business users, connected to our network. We continue with the quantity assessment of data, according to each type of users. The center of our attention is the data traffic download and upload. User analysis is followed by gathering the settlements into areas, which are used as primal generators of data traffic of our network model.

Open source simulation tool called OMNeT++ is used for implementation of network model components, running of actual simulation and its analysis. After the run of simulation experiments there is also detailed analysis and presentation of metric results such as channel utilization factor, packet waiting and travelling time.

Key words: computer network model, OMNeT++, computer simulations

ZAHVALA

Najprej bi se zahvalila mentorju prof. dr. Mihi Mrazu za vsa strokovno pomoč, podporo in vodenje pri izdelavi diplomske naloge. Prav tako pa se zahvaljujem tudi staršem, ki so me skozi vsa leta študija podpirali in mi stali ob strani.

— Nives Bricman, Ljubljana, februar 2018.

KAZALO

Povzetek	i
Abstract	iii
Zahvala	v
1 Uvod	1
1.1 Opredelitev problema	2
1.2 Metodološki pristop	3
1.3 Postavitev modela in simulacije	4
1.4 Struktura naloge	4
2 Določitev bremena omrežja	5
2.1 Analiza prebivalstva Upravne enote Slovenj Gradec	5
2.2 Analiza delovno aktivnega prebivalstva	8
2.3 Prostorska opredelitev opazovanega območja	11
2.4 Metoda za oceno količine prometa	13
2.5 Združitev naselij v območja	16
3 OMNeT++	19
3.1 Uvod v OMNeT++	20
3.2 Gradniki modelov v OMNeT++	20
3.2.1 Enostavni moduli	21
3.2.2 Sestavljeni moduli	21
3.2.3 Sporočila	21
3.3 Opisni jezik NED	22
3.3.1 Vrata	22

3.3.2	Kanali	23
3.3.3	Parametri	23
3.4	Analiza in zbiranje rezultatov	23
4	Implementacija gradnikov omrežja	25
4.1	Sestavljeni moduli	25
4.1.1	Naselje	26
4.1.2	Območje	26
4.1.3	Usmerjevalnik	27
4.1.4	Oddajno sprejemna enota	27
4.1.5	Internet	28
4.2	Enostavni moduli	28
4.2.1	Podatki	29
4.2.2	Generator prometa območje	29
4.2.3	Generator prometa internet	30
4.2.4	Oddaja prometa	30
4.2.5	Sprejem prometa	30
4.2.6	Strežnik	31
4.2.7	Usmerjanje	31
4.2.8	Usmerjevalna tabela	32
4.2.9	Konfiguracija omrežja	32
5	Model omrežja	35
5.1	Vrste in topologije omrežij	35
5.2	Smernice za postavitve modela omrežja	37
5.3	Postavitev modela omrežja	38
5.4	Generiranje prometa	41
5.5	Zagon simulacije	41
5.6	Metrika čakalnih časov	44
5.7	Metrika časov potovanja paketov	48
5.8	Metrika uporabnostnega faktorja povezav	49
5.9	Optimizacija	50
6	Zaključek	57

1 Uvod

Realizacija kompleksnejših računalniških sistemov, kot je postavitve omrežne infrastrukture, je lahko cenovno zelo draga, prav tako pa tudi kasnejše preureditve in posodobitve že postavljenega sistema, če pride do neučinkovitega delovanja. V nadaljevanju se osredotočimo na računalniški sistem paketnega omrežja, ki naročnikom storitve nudi žični prenos IP paketov od uporabnika (angl. *upload*) in do njega (angl. *download*). Cilj internetnega ponudnika (angl. *ISP - internet service provider*) je vzpostavljeno omrežje, ki bo kupce - bodoče uporabnike njegovih storitev privabljal in v veliki meri tudi obdržal. Omrežje mora biti tako sposobno učinkovito in v čim krajšem času ugoditi potrebam uporabnikov, zato sta pri implementaciji ključna hitrost prenosa po žičnih povezavah in čas obdelave IP paketov na aktivnih napravah.

Za internetnega ponudnika je takojšnja realizacija omrežja brez vpogleda v neznan teren zelo draga in lahko nevarna, zato je pred dejansko izvedbo omrežja priporočljiva postavitve modela, ki bo čim bolj povzemalo lastnosti še nerealiziranega sistema. Prednost izvedbe modelov je predvsem v cenejšem poseganju v strukturo modela omrežja, kjer lahko z izvajanjem simulacij nad vzpostavljenim modelom spremljamo dinamiko sis-

tema, zaznavamo ozka grla, dobimo vpogled v delovanje v primeru odpovedi posameznih povezav, opazujemo obnašanje omrežja ob zamašitvi, ob podpovprečni obremenitvi itd. Morebitnih nevšečnosti se tako zavedamo in smo boljše pripravljene na odpravo težav. Posamezne simulacijske eksperimente lahko analiziramo in model popravljamo glede na naš cilj.

1.1 Opredelitev problema

Predpostavimo, da se na nas obrne nov ponudnik internetnih storitev, ki želi na območju upravne enote Slovenj Gradec postaviti žično infrastrukturo paketnega omrežja. Upravno enoto sestavljata Mestna občina Slovenj Gradec in občina Mislinja, ki ju skupno zaseda 33 manjših naselij na površini 286 km². Internetni ponudnik želi načrtovati omrežje, ki bo predstavljalo kar se da optimalno razmerje med ceno in zmogljivostjo omrežja tako zanj, kot tudi za uporabnike. K priključitvi v njegovo omrežje želi privabiti čim več uporabnikov, ki bi po pristopu bili zadovoljni v čim večji meri in bi še naprej uporabljali njegovo storitev.

Večine internetnih uporabnikov tehnične podrobnosti izvedbe ne zanimajo [1]. Zanje sta pomembna predvsem dejavnika zmogljivosti in zanesljivosti delovanja omrežja. Zmogljivost se izraža v hitrosti pretoka podatkov do uporabnika (angl. *download*) in od njega (angl. *upload*), kjer velika večina uporabnikov teži k čim večji hitrosti prenosa. Omenimo lahko, da uporabniki velikokrat enačijo dosegljivost omrežja in dosegljivost storitev, kar pomeni, da v primeru odpovedi določene storitve (npr. spletnega ali poštnega strežnika) pomislijo na težave v omrežju pri njihovem internetnem ponudniku. Pogostokrat pa se izkaže, da ne gre za probleme v dosegljivosti omrežja, ampak prihaja do drugih motenj na samih storitvah, do katerih dostopa uporabnik.

Internetni ponudnik naj bi uporabnikom nudil čim večjo dosegljivost omrežja, kjer je pogostost napak minimalna. S svojimi aktivnimi komponentami in žičnimi povezavami naj bi vzpostavil in vzdrževal zmogljivo omrežje, ki bo sposobno obdelovati količino prometa, ki ga obremenjuje. Tako morajo biti aktivne naprave in vodniki dovolj zmogljivi, da ne prihaja do ponovnega pošiljanja paketov. V primeru uporabe premalo zmogljivih resursov lahko pride do predolgega čakanja zahtev v vrsti ali celo do izgube paketov, kar vodi v nezadovoljstvo strank, ki se lahko odločijo za prehod k drugemu internetnemu ponudniku.

V diplomski nalogi obravnavamo problem postavitve poenostavljenega modela fizičnega omrežja, ki bo temeljil na statistikah opazovanega območja Upravne enote Slovenj Gradec in hkrati ustrezal zahtevam internetnega ponudnika ter potrebam uporabnikov. Omenjene statistike zajemajo podatke o številu prebivalcev po posameznih naseljih in podatke o številu zaposlenih po obravnavanih naseljih. Model bo torej v osnovi zasnovan na podlagi števila potencialnih uporabnikov ter njihovih navadah uporabe predpostavljenih internetnih storitev. Slednje se nanašajo na pogostost uporabe internetnih storitev posameznega bivanjskega in poslovnega uporabnika. Za vsak tip uporabnika posebej bomo ocenili krivuljo povprečnega pretoka omrežnih podatkov v odvisnosti od dnevnega časovnega obdobja. Vzpostavitev posameznih ocen podrobneje opišemo v poglavju 2. S pomočjo izvajanja in analiziranja simulacij je naš cilj postaviti končni model omrežja, ki bi ga internetni ponudnik lahko upošteval kot osnovo za izvedbo fizičnega omrežja.

1.2 Metodološki pristop

Opisani problem lahko rešujemo z analitičnim ali simulacijskim pristopom [2]. Analitičen pristop poteka na podlagi matematičnih zakonitosti teorije strežbe, medtem ko si pri simulacijskem pomagamo z računalnikom kot simulatorjem, s katerim zgrajen model simuliramo, testiramo in analiziramo. Analitični pristop zajema aktivnost strežnega sistema v točno določenem trenutku, kar pomeni neupoštevanje dejanske porazdelitve procesa, torej z njim izgubimo na splošnosti [3]. Prav tako ne upošteva strukture obremenitve, zato ne moremo pridobiti vseh podatkov o možnih stanjih sistema. Vendar do dejanskih rezultatov v primerjavi s simulacijskim modelom pridemo zelo hitro. Zaradi kompleksnosti postavitve modela žičnega omrežja in z željo po opazovanju večih zmogljivostnih metrik uporabimo simulacijski pristop, katerega prednost je vpogled v dinamiko postavljenega računalniškega sistema. Detektiramo in preprečujemo lahko morebitna ozka grla in model postavljamo v različna stanja. Tako lahko predvidimo delovanje postavljenega omrežja, ko je sistem v stanju podpovprečne, povprečne ali nadpovprečne obremenitve. Slabost simulacijskega pristopa je v časovnosti postavitve modela, saj je potrebno dobro razumevanje delovanja opazovanega sistema, prav tako pa je zaradi izvedbe velikega števila dogodkov ključen tudi čas izvajanja simulacije na računalniku. Ne glede na način postavitve modela je potrebna validacija modela in zaupanje vanj, saj zaključki na osnovi modela, ki ne predstavlja realnega stanja, ne bodo pravilni.

1.3 Postavitev modela in simulacije

Rezultat modeliranja predstavlja bolj ali manj poenostavljen model nekega realnega ali še neobstoječega sistema [2]. V našem primeru gre za namišljeno internetno omrežje, ki mora ustrezati pogojem definirane območja in našim željam. Izgradnja modela, ki čim bolj aproksimira dejansko stanje realnega ali namišljenega modela, je ključna, saj se želimo z modelom čim bolj približati dejanskemu stanju opazovanega sistema za čim bolj uspešno nadaljnjo analizo. V realnih sistemih sta nadgradnja in večji poseg v delujoč sistem mnogokrat nepraktična in draga, zato je pri spremembah potrebna previdnost. Zavedanje o morebitnih težavah in njihovih posledicah predstavlja veliko prednost, saj nas seznanjenost s problemom vodi v pravočasno in hitrejše iskanje rešitve. Zgrajen model tako omogoča enostavno spreminjanje vrednosti vhodnih spremenljivk in dodajanje ali odstranjevanje raznih resursov, s katerimi težimo k optimizaciji problema novega internetnega ponudnika. Simulacija nam pomaga pri vrednotenju, ocenjevanju in analiziranju delovanja opazovanega sistema, ki je velikokrat pogoj pred spreminjanjem sistema. S simulacijo modela želimo identificirati ozka grla našega sistema in pojavitev teh minimizirati. Vprašamo se lahko, koliko uporabnikov lahko vključimo v omrežje, preden bo prišlo do preobremenitve našega omrežja in bo potrebno v omrežje vključiti nove resurse.

1.4 Struktura naloge

Drugo poglavje diplomske naloge namenimo določitvi bremena omrežja, v katerem najprej predstavimo opazovano območje Upravne enote Slovenj Gradec. Na podlagi analiz prebivalstva in delovno aktivnih oseb vzpostavimo oceni števila bivanjskih in poslovnih uporabnikov. Sledi določitev bremena omrežja po posameznih definiranih območjih. V tretjem poglavju predstavimo uporabljeno simulacijsko orodje OMNeT++ in opišemo nekatere izmed njegovih najbolj uporabljenih komponent. Četrto poglavje vsebuje opis delovanja posameznih modulov oz. gradnikov našega modela omrežja, implementiranih z uporabo orodja OMNeT++. V petem poglavju opišemo postopek postavitve modela našega omrežja, nad katerim izvedemo simulacijska eksperimenta in predstavimo njune rezultate.

2 Določitev bremena omrežja

V pričujočem poglavju predstavimo analizo naselij opazovanega območja Upravne enote Slovenj Gradec, ki je sestavljena iz občin Slovenj Gradec in Mislinja. Pri tem se osredotočimo na prebivalce naselij in njihovo zaposleno populacijo. Na podlagi pridobljenih podatkov ocenimo število bivanjskih in poslovnih uporabnikov, ki bodo priključeni v naše omrežje in s tem prispevali h generiranju prometa. Za nas je pomembna predvsem ocena količine prenosa podatkov posamezne skupine uporabnikov. Pri tem upoštevamo količino prenosa do uporabnika (angl. *download*) in količino prenešenih podatkov od uporabnika (angl. *upload*). Na koncu poglavja obravnavana naselja združimo v območja, katerim na podlagi števila bivanjskih in poslovnih uporabnikov določimo količino celodnevnega prometa celotnega območja.

2.1 Analiza prebivalstva Upravne enote Slovenj Gradec

Upravno enoto Slovenj Gradec sestavljata občini Slovenj Gradec in Mislinja. S Statističnega urada republike Slovenije smo pridobili podatke obeh občin. Tabela 2.1 prikazuje število prebivalcev, ki so bili popisani v letu 2015. Novejših podatkov nismo zajeli, saj

pri nadaljnjem iskanju podatkov o prebivanju ljudi po posameznih naseljih nismo našli. V omenjeni tabeli se poleg števila prebivalcev obeh občin nahaja površina ozemlja in število naselij, ki občini pripadajo. Opazovano območje je tako v celoti veliko 286 km^2 , sestavlja pa ga 33 naselij, ki bodo podrobneje predstavljena v naslednjem razdelku. Omenimo še, da celotno populacijo Upravne enote Slovenj Gradec za leto 2015 predstavlja 21.378 prebivalcev.

Občina	Število prebivalcev	Površina območja [km^2]	Število naselij
Slovenj Gradec	16.785	173,70	22
Mislinja	4.593	112,30	11
Skupaj	21.378	286,00	33

Tabela 2.1 Tabela osnovnih demografskih značilnosti občin Slovenj Gradec in Mislinja na osnovi popisa iz leta 2015.

Celotno populacijo Upravne enote Slovenj Gradec na podlagi podatkov iz Statističnega urada razdelimo po posameznih naseljih, kar je prikazano v tabeli 2.2. Zgornjemu delu navedene tabele pripadajo naselja iz občine Slovenj Gradec, spodnji del tabele pa predstavljajo naselja občine Mislinja. Tabela vsebuje oštevilčena imena naselij in pripadajoče vrednosti o številu prebivalcev. Na tem mestu smo postavili oceno števila bivanjskih uporabnikov, ki bodo priključeni v internetno omrežje novega internetnega ponudnika. Pri oceni števila uporabnikov uporabimo predpostavko, da so prebivalci od vključno 10 do 59 leta starosti najpogostejši in s tem ključni uporabniki interneta. Omenjeni interval je naša predpostavka, ki ne temelji na realnem stanju, saj za območje Upravne enote Slovenj Gradec nismo našli podatkov o starostni porazdelitvi najpogostejših uporabnikov interneta. V oceno števila bivanjskih uporabnikov določenega naselja smo torej zajeli vse prebivalce obravnavanega naselja od 10 do 59 leta starosti. Podatke smo zajeli iz starostnega popisa prebivalstva po naseljih objavljenega na strani Statističnega urada Republike Slovenije.

Skupno število bivanjskih uporabnikov na podlagi naše predpostavke znaša 14.219, kar predstavlja 66,51% celotne populacije Upravne enote Slovenj Gradec. Največ bivanjskih uporabnikov se nahaja v naseljih Slovenj Gradec in Mislinja.

Ime naselja	Število prebivalcev	Ocena števila uporabnikov	Površina območja [km ²]
1 Brda	291	195	4,10
2 Gmajna	466	334	6,10
3 Golavabuka	176	124	9,40
4 Gradišče	312	221	5,50
5 Graška Gora	110	77	4,20
6 Legen	1.061	703	18,40
7 Mislinjska Dobrava	745	501	7,00
8 Pameče	1.314	944	15,80
9 Podgorje	977	674	18,70
10 Raduše	228	153	7,00
11 Sele	238	161	11,40
12 Slovenj Gradec	7.477	4.837	5,60
13 Spodnji Razbor	192	132	10,40
14 Stari trg	642	420	4,70
15 Šmartno pri Slovenj Gradcu	1.237	810	5,90
16 Šmiklavž	142	95	4,00
17 Tomaška vas	129	74	0,50
18 Troblje	325	214	3,80
19 Turiška vas	143	90	0,70
20 Vodriž	111	70	3,60
21 Vrhe	329	223	8,90
22 Zgornji Razbor	140	96	18,00
23 Dovže	330	231	2,10
24 Gornji Dolič	508	362	4,40
25 Kozjak	408	273	12,80
26 Mala Mislinja	192	140	8,50
27 Mislinja	1.869	1.209	43,00
28 Paka - del	48	29	3,60
29 Razborca	104	71	9,20
30 Srednji Dolič	249	163	5,00
31 Šentilj pod Turjakom	305	201	1,70
32 Tolsti Vrh pri Mislinji	156	97	7,20
33 Završe	424	295	14,80
Upravna enota Slovenj Gradec	21.378	14.219	286,00

Tabela 2.2 Ocena števila uporabnikov po naseljih.

2.2 Analiza delovno aktivnega prebivalstva

Za oceno števila poslovnih uporabnikov našega omrežja smo kot vir privzeli podatke s Statističnega urada prikazane v tabeli 2.3. Podatki prikazujejo zaposlene v podjetjih na nivoju cele Slovenije, ki na delovnem mestu uporabljajo informacijsko-komunikacijske tehnologije. Podatki so razdeljeni glede na zaposlene, ki na delovnem mestu uporabljajo računalnik, računalnik z internetnim dostopom in zaposlene, ki jim je bila dodeljena prenosna naprava z vsaj 3G tehnologijo. Slednji zaposleni za nas niso relevantni, saj gre za brezžično infrastrukturo, ki ne bo del naših storitev. Odstotek zaposlenih na nivoju cele Slovenije, ki svoje delo opravljajo na računalniku z internetnim dostopom, je približno 50%, kar predstavlja oceno števila poslovnih uporabnikov znotraj posameznega naselja.

Število zaposlenih	Uporaba računalnika	Računalnik z internetnim dostopom	Prenosna naprava z vsaj 3G tehnologijo
359.326	197.062	173.888	61.592

Tabela 2.3 Uporaba informatike med zaposlenimi v podjetjih na nivoju cele Slovenije na osnovi popisa iz leta 2015.

Tabela 2.4 prikazuje delovno aktivno prebivalstvo, ki dela v posameznih naseljih. Delovno aktivne predstavljajo zaposlene in samozaposlene osebe, ki imajo delovno mesto v obravnavanem naselju, ne glede na to, kje imajo prijavljeno stalno prebivališče. V število delovno aktivnega prebivalstva so torej zajete vse osebe, ki prebivajo v določenem naselju in delo opravljajo v njem, ter tudi druge morebitne osebe, ki v naselju ne prebivajo ampak v njem delajo.

Podatke smo pridobili s Statističnega urada Republike Slovenije za leto 2015. Za nekatera naselja ti podatki niso na voljo, saj je bilo število delovno aktivnih prebivalcev manjše od praga zaupnosti. Stolpec omenjenih naselij je označen s poševnico (/). V nadaljevanju smo pri teh naseljih ocenili, da v njih ne obstajajo zaposleni v poslovnih podjetjih, zato je vrednost drugega stolpca enaka 0. V ostalih naseljih, kjer so podatki znani smo oceno števila poslovnih uporabnikov nastavili tako, da smo iz števila delovno aktivnega prebivalstva naselja izvzeli polovico oseb. Ponovno gre za približne ocene števila poslovnih uporabnikov, ki svojo dejavnost opravljajo v poslovnih objektih in bodo poleg prebivajočih uporabnikov prispevali h generiranju prometa.

Ime naselja	Število delovno aktivnega prebivalstva	Ocena števila uporabnikov
1 Brda	31	16
2 Gmajna	197	99
3 Golavabuka	14	7
4 Gradišče	29	15
5 Graška Gora	/	0
6 Legen	145	73
7 Mislinjska Dobrava	256	128
8 Pameče	822	411
9 Podgorje	198	99
10 Raduše	18	9
11 Sele	40	20
12 Slovenj Gradec	5.285	2.643
13 Spodnji Razbor	/	0
14 Stari trg	68	34
15 Šmartno pri Slovenj Gradcu	179	90
16 Šmiklavž	/	0
17 Tomaška vas	7	4
18 Troblje	69	35
19 Turiška vas	10	5
20 Vodriž	/	0
21 Vrhe	30	15
22 Zgornji Razbor	/	0
23 Dovže	23	12
24 Gornji Dolič	62	31
25 Kozjak	41	21
26 Mala Mislinja	23	12
27 Mislinja	234	117
28 Paka - del	/	0
29 Razborca	31	16
30 Srednji Dolič	/	0
31 Šentilj pod Turjakom	89	45
32 Tolsti Vrh pri Mislinji	33	17
33 Završe	47	24
Upravna enota Slovenj Gradec	7.981	3.998

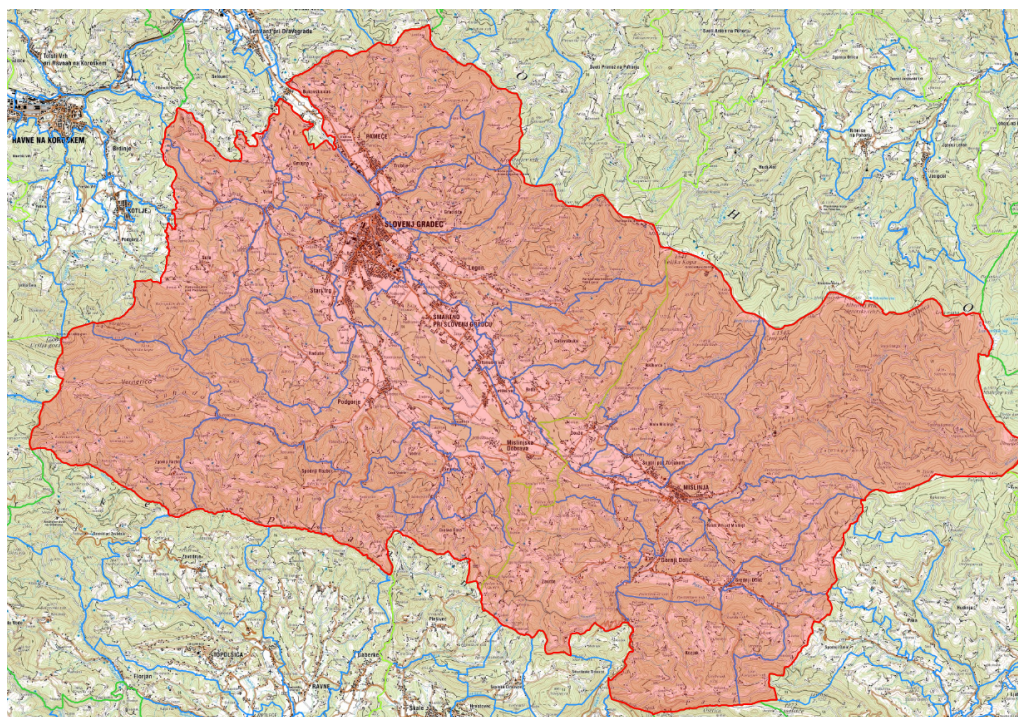
Tabela 2.4 Delovno aktivno prebivalstvo glede na delovno mesto po naseljih.

Oceno števila poslovnih uporabnikov predstavlja 3.998 zaposlenih. Omenjeni oceni iz tabele 2.2 prištejemo še bivanjske uporabnike, ter na ta način pridobimo 18.217 uporabnikov. Odstotek zaposlenih uporabnikov glede na celotno populacijo ocenjenih uporabnikov znaša 21,95%. Precej manjša količina prometa bo torej generirana s strani zaposlenih v poslovnih objektih, ki svojo dejavnost opravljajo v dopoldanskem osem urnem delovniku. Promet značilen za zaposlene tako ne bo vplival na porast količine prometa proti večeru, temveč bo prispeval k pribitku omrežnega prometa ob predpostavljenih delovnih urah. Celoten omrežni promet po območjih je tako sestavljen iz navad bivanjskih uporabnikov in zaposlenih v poslovnih objektih.

Na tem mestu omenimo še pesimističnost našega modela, ki ga bomo vzpostavili v nadaljevanju diplomskega dela. Pri oceni količine prometa po območjih bomo namreč predpostavili, da celotni množici bivanjskih in poslovnih uporabnikov neodvisno prispevata h generiranju prometa v celotnem dnevu. Bolj optimistične rezultate bi lahko dosegli tako, da bi pri generiranju prometa bivanjskih uporabnikov v delovnem času podjetij izvzeli poslovne uporabnike, ki se v času obratovanja podjetja nahajajo na svojem delovnem mestu. Vendar bi pri tem morali pridobiti dejanske podatke o številu prebivalcev, ki imajo prijavljeno stalno prebivališče v naselju in v njem tudi delajo. Nadaljnje bi za bolj natančne ocene za vsakega bivanjskega uporabnika potrebovali podatek v katerem naselju dela in kakšen je njegov delovni čas. Poleg tega pa bi morali ločeno upoštevati še vse šolo obvezne uporabnike. Naša ocena torej predpostavlja, da podmnožica bivanjskih uporabnikov, ki pripada tudi podmnožici poslovnih uporabnikov hkrati prispeva h generiranju prometa.

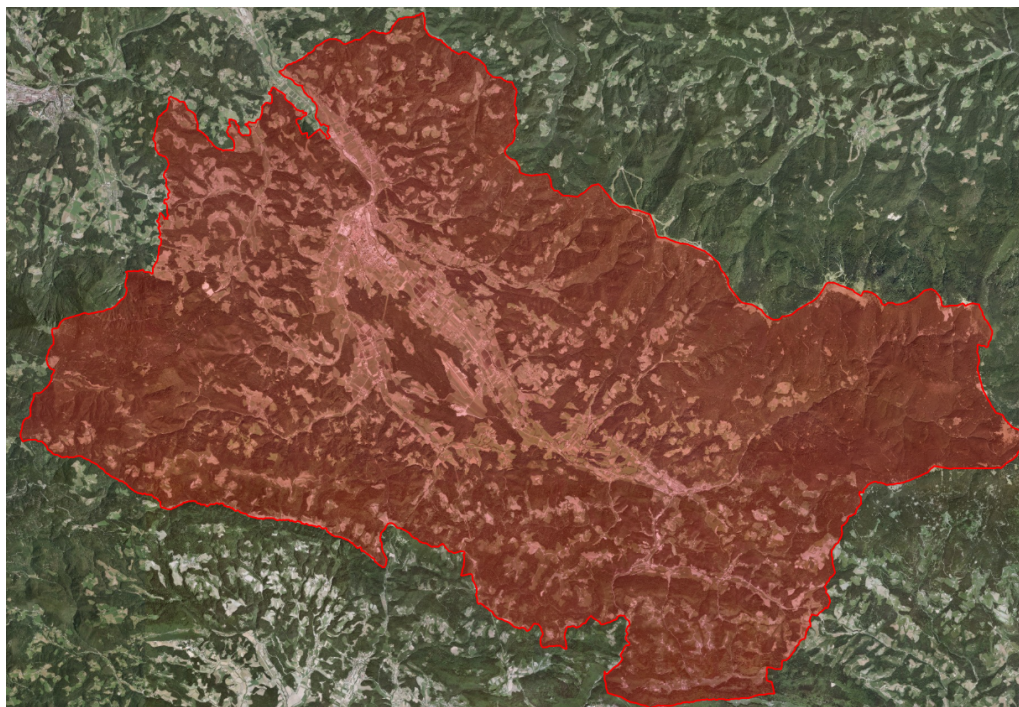
2.3 Prostorska opredelitev opazovanega območja

Za prikaz prostorske opredelitve izbranega območja smo uporabili slike s spletnega portala Geopedia [4]. Slika 2.1 prikazuje topografijo Upravne enote Slovenj Gradec, na kateri so stavbe posebej označene. Zaradi velikosti opazovanega območja, je vidljivost stavb na manj poseljenih, travnatnih in gozdnatih terenih slaba. Kljub temu pa lahko s slike identificiramo območja, ki bodo predstavljala največjo obremenitev modela žičnega omrežja. V prejšnjem delu tekočega poglavja, kjer smo tabelirali lastnosti posameznih naselij, smo omenjena območja že lahko identificirali na podlagi ocene števila uporabnikov. Pridobljene vrednosti primerjamo s sliko, kjer opazimo, da je največja gostota stavb v istoimenskih centrih opazovanih občin Slovenj Gradec in Mislinja. Slednji naselji namreč vsebujeta največje število bivanjskih in poslovnih uporabnikov. Velike količine interneta prometa pričakujemo še v naseljih Šmartno pri Slovenj Gradcu, Legen, Podgorje, Mislinjska Dobrava, Pameče, Stari trg in Gornji Dolič.



Slika 2.1 Prikaz lokacij stavb Upravne enote Slovenj Gradec.

Upravno enoto predstavimo še v ortografski obliki na sliki 2.2, ki občini opredeli kot kotlini - središči hribovitih površin. Največji del prebivalstva je skoncentriran v centrih. Na obrobjih najdemo veliko kmetijskih zemljišč in cest, ki vodijo do manjših in težje dosegljivih naselij znotraj gozdov. Kot uporabnike omrežne storitve - dostopa do interneta, bomo v diplomski nalogi upoštevali vse pripadnike množice ocenjenega števila uporabnikov po posameznih naseljih. Potrebno je poudariti, da je celotni populaciji težko zagotoviti žični dostop do interneta. Razlog je predvsem v velikosti finančne investicije, ki je potrebna za izgradnjo novega ali nadgradnjo že obstoječega omrežja. Le-ta naj bi za internetnega ponudnika prinašala dobiček. Poleg denarnih omejitev omenimo še nedostopnost posameznih območij kot so npr. odmaknjene kmetije, planinske kočje itd.

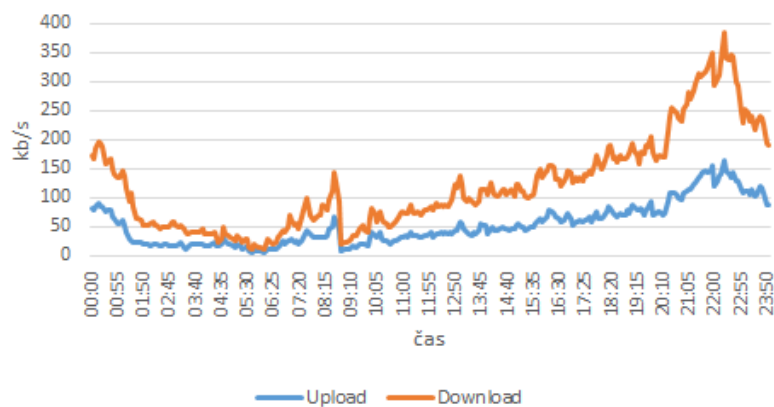


Slika 2.2 Ortophografška slika opazovanega območja.

2.4 Metoda za oceno količine prometa

Pred načrtovanjem modela, ki bo glavna tema naslednjih poglavij, predstavimo še postopek metode za oceno količine internetnega prometa. V tem delu se osredotočimo na obravnavana tipa uporabnikov in sicer na bivanjske in poslovne uporabnike. Navedeno populacijo bivanjskih uporabnikov predstavimo z vzorcem 30 študentov Študentskega doma v Ljubljani. Zaradi oteženega zbiranja podatkov dovolj velikega vzorca uporabnikov, ki bi dobro opisoval razmere na celotni populaciji, si delo olajšamo in privzamemo, da internetni dostop študentov dovolj dobro opisuje količino prometa, ki jo potrebujemo za nadaljnje delo. Ponovno poudarimo, da gre za nepreverjeno predpostavko, za katero ne moremo potrditi, da bo dovolj dobro odražala dejansko stanje.

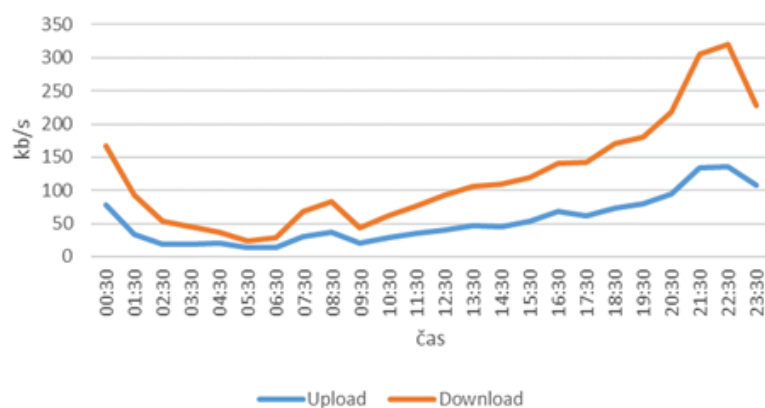
Za merjenje parametrov kot sta količina prenosa od usmerjevalnika proti internetu (angl. *upload*) in količina prenosa od usmerjevalnika proti uporabniku (angl. *download*) smo uporabili orodje Cacti. Spremljali smo enodnevno statistiko usmerjevalnika v vzorcu, kjer smo posamezne meritve zabeležili v 5 minutne odseke. Zbrane podatke smo povprečili na enega študenta, saj smo s tem pridobili oceno količine prometa na posameznega bivanjskega uporabnika. Povprečne vrednosti prometa na študenta iz obravnavanega vzorca se nahajajo na sliki 2.3.



Slika 2.3 Količina prometa na enega uporabnika v petminutnih intervalih.

Opazimo, da za bivanjskega uporabnika - v našem primeru študenta, količina prenešenega prometa do njega (angl. *download*) z linearno rastjo začne v dopoldanskih urah. Proti večeru hitrost strmo naraste do maksimuma, nato pa krivulja do naslednjega dne pada.

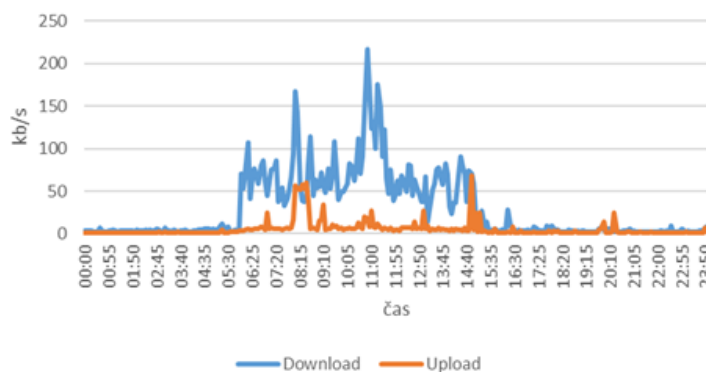
Omrežje torej postaja v jutranjih urah vedno manj obremenjeno. Funkcija za količino prenosa od uporabnika (angl. *upload*) ima enake lastnosti kot prva, doseže pa le za približno polovico manjše vrednosti. Krivulji s slike 2.3 zgladimo tako, da meritve znotraj posameznih ur povprečimo. S tem se znebimo hitrih in ostrih prehodov, ki jih prvotni funkciji dosegata. Prednost pri nadaljnjem simuliranju tako predstavlja tudi manjše definicijsko območje časovnih intervalov, katerih vrednosti bomo uporabili kot konstantne generatorje zahtev znotraj pripadajočega urnega intervala. Breme omrežja do uporabnika in od njega je prikazano na sliki 2.4.



Slika 2.4 Zglajena količina prometa na enega bivanjskega uporabnika po urah.

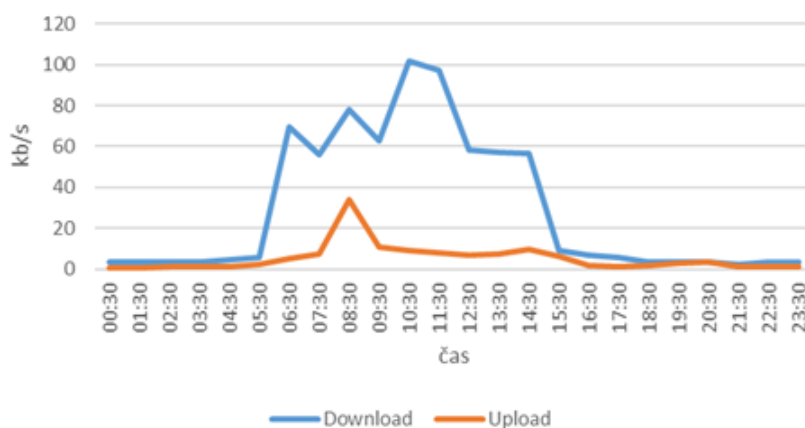
Omenimo, da je maksimalna hitrost v smeri proti uporabniku (angl. *download*) in v smeri od uporabnika (angl. *upload*) dosežena v času med 22:00 in 23:00. Maksimalna *download* hitrost znaša 320,73 kb/s, medtem, ko je maksimalna *upload* hitrost 135,89 kb/s. Bivanjski uporabnik bo omrežje najmanj obremenjeval v času med 5:00 in 6:00, saj sta bili takrat ocenjeni minimalni hitrosti *download* - 23,26 kb/s in *upload* - 12,13 kb/s.

Za oceno količine prenosa podatkov zaposlenih v poslovnih objektih smo pridobili meritve organizacije, ki dejavnost opravlja v osem urnem delavniku. IP naslovni prostor opazovane organizacije omogoča dodelitev IP naslovov 62 različnim napravam, kar smo uporabili kot imenovalec pri deljenju s celotno količino internetnega prometa organizacije. Podatkov o številu zaposlenih katerih primarno delo je uporaba računalnikov in internetnega dostopa nismo pridobili, zato smo kot oceno uporabili kar razpoložljivo število IP naslovov. Slika 2.5 prikazuje povprečno število prenešenih podatkov od in do uporabnika v 5 minutnih časovnih intervalih.



Slika 2.5 Količina prometa na enega poslovnega uporabnika (5 minutne meritve).

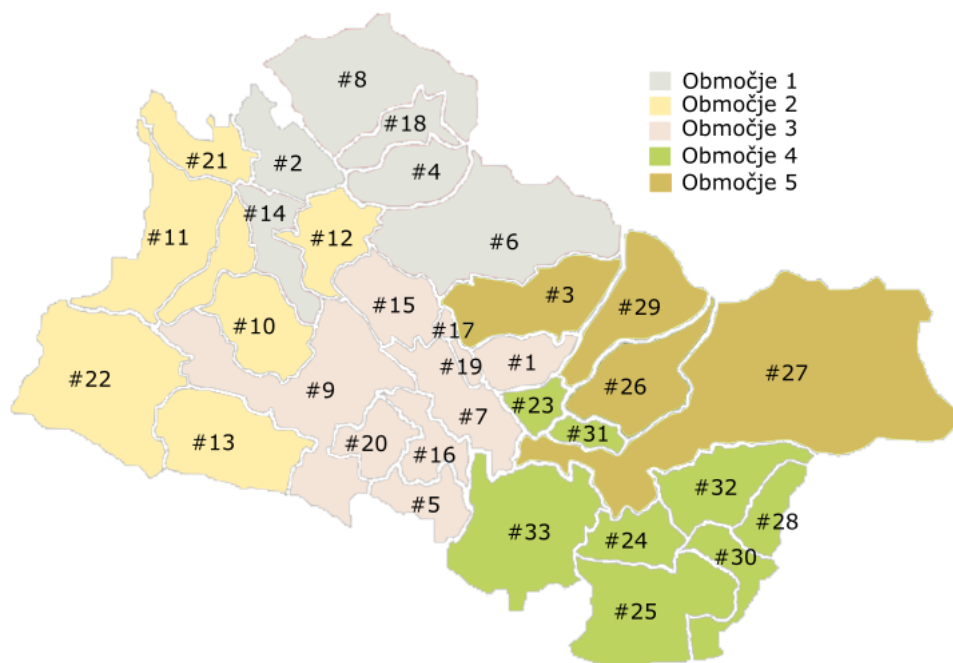
Zaradi predhodno navedenih razlogov smo funkciji količine prometa za zaposlene v poslovnih objektih povprečili znotraj meritev urnega intervala, kar je predstavljeno na sliki 2.6. Z grafa na sliki 2.6 lahko razberemo, da poslovni uporabnik generira največ *download* prometa med 10:00 in 11:00 uro, največ *upload* prometa pa je s strani zaposlenega uporabnika generiranega v času med 8:00 in 9:00. Navedeni vrednosti sta v primerjavi s količino prometa bivanjskega uporabnika bistveno manjši. Razlika je lahko posledica uporabe zahtevnejših spletnih storitev s strani študentov, kot so prenos medijskih vsebin, ogled videoposnetkov na zahtevo, igranje spletnih iger itd. Medtem je primarna uporaba spleta v organizacijah osredotočena na lažje storitve kot so npr. elektronska pošta, iskanje informacij, telefoniranje itd. Prav tako opazimo, da sta funkciji pretežno omejeni znotraj dveh pasov, ki naj bi predstavljala začetek in konec delavnega časa organizacije.



Slika 2.6 Zglajena količina prometa na enega poslovnega uporabnika (urne meritve).

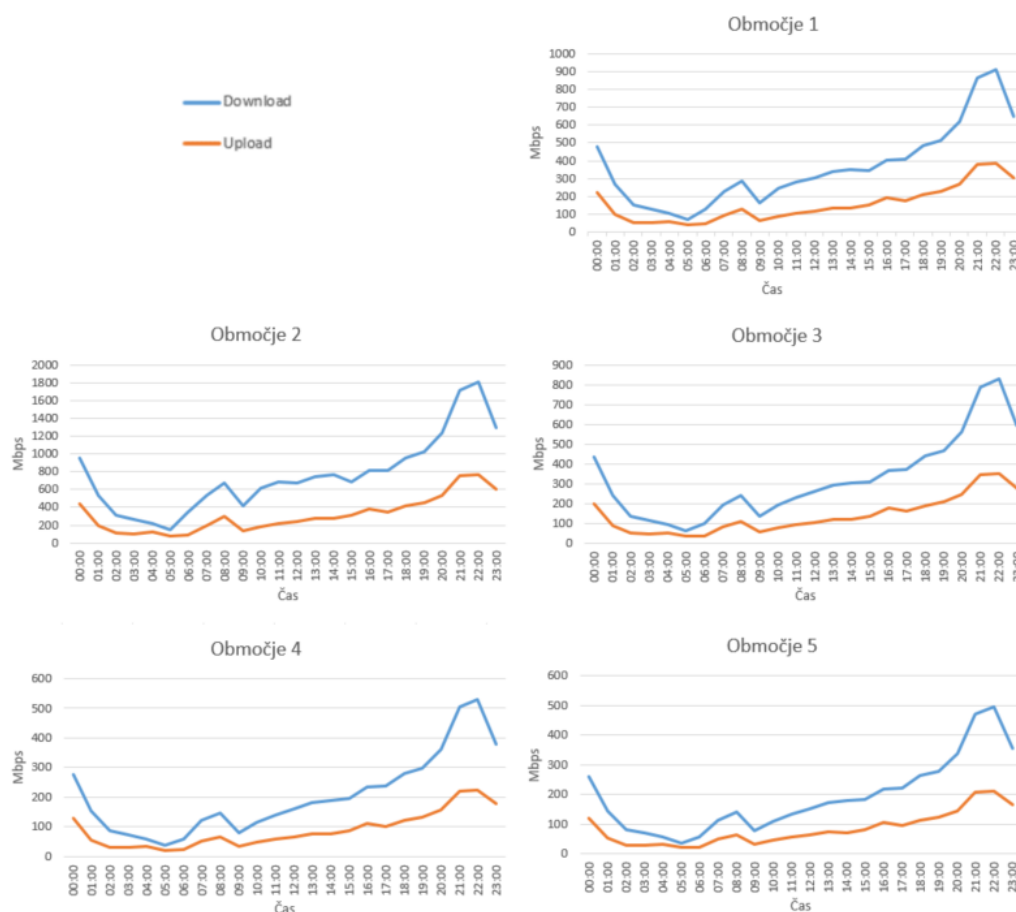
2.5 Združitev naselij v območja

Slika 2.7 prikazuje poenostavljeno shemo združenih naselij, ki bodo v nadaljevanju predstavljala območja izvorov generiranja količine prometa s strani uporabnikov. Naselja smo glede na njihovo lokacijo združili v 5 območij. Posamezno območje smo na sliki 2.7 ponazorili z enolično barvo in označili pripadajoča naselja z njihovimi identifikacijskimi številkami. Največ generiranih zahtev, ki jih bo moral model žičnega omrežja skozi simulacijo postreči, bo izviralo s področij z največ uporabniki. Zaradi reliefne oblike opazovanih občin bo sistem najbolj obremenjen v samih centrih, ki bodo predstavljali hrbtenico omrežja. Telekomunikacijske povezave na obrobjih bodo zaradi manjšega števila uporabnikov manj zmogljive, posamezna podomrežja pa se bodo priključevala hrbtenici. Posledica priključevanja povezav v stičišče omrežja je še dodatna obremenitev fizičnih povezav, ki ne bodo prenašale le paketov uporabnikov stanujočih v bližini hrbtenice, ampak bodo morale postreči še promet s preostalimi naselji.



Slika 2.7 Okvirna shema postavitve modela omrežja.

Na tem mestu na sliki 2.8 predstavimo grafe količin omrežnega prometa po definiranih območjih. Količina prenosa v smeri proti območju je izračunana kot utežena vsota izmerjenih *download* funkcij posameznega bivanjskega uporabnika s slike 2.4 in poslovnega uporabnika s slike 2.6. Količino generiranega prometa s strani območja pa določimo kot uteženo vsoto izmerjenih *upload* funkcij posameznega tipa uporabnikov. Uteži predstavljajo oceno števila bivanjskih in poslovnih uporabnikov ustreznih območij, ki jih povzamemo po tabeli 2.5.



Slika 2.8 Okvirna shema postavitve modela omrežja.

Oceno števila bivanjskih in poslovnih uporabnikov definiranih območij prikažemo v tabeli 2.5. Slednja tabela vsebuje tudi ocene hitrosti izhodnih in vhodnih povezav posameznih območij. Prvi tip povezave prenaša promet generiran s strani območja (angl.

upload), vhodna povezava pa skrbi za prenos prometa proti območju (angl. *download*). Hitrosti vhodnih povezav smo nastavili tako, da smo za posamezno območje pridobili maksimalno količino *download* prometa ter pridobljen maksimum pomnožili z vrednostjo 1,50. Hitrost povezave je torej nastavljena tako, da maksimalna količina generiranega prometa predstavlja $\frac{2}{3}$ zasedenost povezave. Izhajamo namreč iz predpostavke, da pri $\frac{2}{3}$ zasedenosti omrežja prihaja do uporabniku zaznavne upočasnitve delovanja sistema. Podobno kot smo določili oceno hitrosti vhodne povezave, izračunamo še oceno hitrosti izhodne povezave, pri kateri smo poiskali maksimum generiranega *upload* prometa posameznega območja.

ID območja	Bivanjski uporabniki	Poslovni uporabniki	Izhodna povezava [Mbps]	Vhodna povezava [Mbps]
Območje 1	2.836	667	1.368	579
Območje 2	5.602	2.687	2.709	1.146
Območje 3	2.586	342	1.246	528
Območje 4	1.651	150	795	337
Območje 5	1.544	152	744	315

Tabela 2.5 Umestitev pridobljenih ocen po območjih.

Zaključujemo poglavje v katerem smo predstavili opazovano območje Upravne enote Slovenj Gradec, ki ga sestavljata občini Mislinja in Slovenj Gradec. Občini smo natančneje analizirali po 33 naseljih, kjer smo iz števila prebivalcev in delovno aktivnih oseb pridobili oceno uporabnikov. Uporabnike smo razdelili na bivanjske, ki bodo internetni dostop uporabljali v osebne namene in na uporabnike, ki internet uporabljajo na delovnem mestu. Izmerili smo povprečno količino internetnega prometa na študenta v študentskem domu in zaposlenega v dnevnem časovnem obdobju. Privzeli smo, da množica študentov dovolj dobro ponazarja obnašanje bivanjskih uporabnikov. Opazovana naselja smo zaradi poenostavitve združili v pet večjih območij in jih opredelili s funkcijami, ki aproksimirajo IP prenos v smeri do in od uporabnikov.

3 OMNeT++

Za potrebe postavitve modela omrežja je na voljo ogromno odprtokodnih, prostih in plačljivih simulacijskih orodij [5]. Prvotna licenčna oprema se v večini pojavlja v obliki programskih knjižnic, paketov in ogrodij, kjer je zahtevana uporaba programiranja, saj v nasprotnem primeru ne moremo realizirati kompleksnejših modelov, oziroma smo pri tem omejeni. Z željo po uporabi orodja s prijaznim uporabniškim vmesnikom in vgrajenimi objekti brez pisanja lastne kode, smo preverili tudi skupino plačljivih programov, kjer je večina orodij primarno osredotočenih na točno določena področja kot so zdravstvo, industrija, promet, računalništvo itd. Vsem je skupen vmesnik povleci in spusti (angl. *drag and drop*). V večini se razlikujejo po vsebovanih funkcionalnostih (npr. vizualizacija v 3D prostoru, orodja za analizo simulacij, grafični prikazi rezultatov, animacija dinamike sistema, skriptni programski jezik itd.). Navadnim uporabnikom je takšna programska oprema zaradi cene precej nedostopna, vendar nekatera izmed podjetij nudijo uporabo demo programov ali licenčnih paketov za akademsko delo. Podrobneje smo pregledali funkcionalnosti plačljivih orodij kot so Enterprise Dynamics, Anylogic in Goldsim. Težava takšnih orodij je v njihovi omejenosti npr. velikosti modela, prekoračitvi dovolje-

nega števila zahtev v sistemu ali oteženega spreminjanja delovanja obstoječih komponent. Zaradi takšnih pomanjkljivosti in narave našega problema smo se odločili, da celotno delovanje komponent implementiramo sami z uporabo orodja OMNeT++ [6]. Opis izbranega orodja v nadaljevanju povzemamo iz virov [7], [8] in [9].

3.1 Uvod v OMNeT++

OMNeT++ je objektno orientirano diskretno dogodkovno ogrodje (angl. *framework*) namenjeno modeliranju, simuliranju, vizualizaciji in kontroliranju dinamičnih procesov. Zaradi njegove generične arhitekture je uporabno v različnih domenah problemov. Le te so sledeče:

- modeliranje žičnih in brezžičnih komunikacijskih omrežij,
- modeliranje protokolov,
- modeliranje strežnih sistemov,
- modeliranje multiprocesorjev in ostalih porazdeljenih strojnih sistemov,
- validacija strojne arhitekture,
- splošna uporaba v sistemih z značilnostjo diskretnega izvajanja dogodkov.

Samo orodje kot tako ne vsebuje komponent za modeliranje računalniških omrežij, strežnih sistemov ali drugih področij, ampak uporabnikom nudi infrastrukturo in orodja za pisanje ter poganjanje simulacij. Poleg simulacijskega jedra in programskih knjižnic vsebuje integrirano razvojno okolje (angl. *IDE - integrated development environment*) bazirano na okolju Eclipse, ter grafični vmesnik za prikaz simulacij, s katerim pridobimo dober vpogled v dinamiko modeliranega sistema. Za neprofitno in akademsko uporabo je ogrodje brezplačno, na voljo pa je za najpogostejše operacijske sisteme vključno z Linux, Windows in Mac OS. Obstaja še posebna verzija OMNeT++, imenovana OMNEST, ki je namenjena komercialni uporabi.

3.2 Gradniki modelov v OMNeT++

Postavitev modela v ogrodju OMNeT++ je modularno zasnovana, kar pomeni, da je takšen sistem razdeljen na manjše medsebojno neodvisne komponente imenovane moduli. Prednost takšnega pristopa je prilagodljiv in nadgradljiv sistem, kjer so komponente

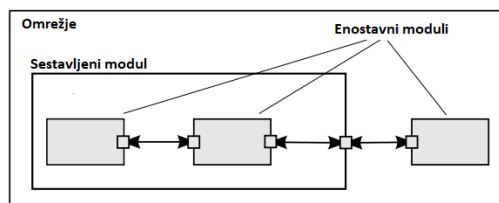
prenosljive v druge modele [10]. Simulacijski modeli v OMNeT++ sestojijo iz enostavnih in sestavljenih modulov, povezav med njimi in sporočil.

3.2.1 Enostavni moduli

Enostavni moduli predstavljajo najosnovnejše aktivne komponente, v katerih se prožijo dogodki. Delovanje enostavnih modulov je z uporabo simulacijske knjižnice implementirano v programskem jeziku C++. Uporabniki enostavne module definirajo z dedovanjem razreda `cSimpleModule`. Dedovanje omogoča razširitev razredov, kjer lahko uvedemo dodatne metode ali lastnosti ter implementiramo delovanje že obstoječih virtualnih metod, pridobljenih iz nadrazredov. V OMNeT++ torej komponente predstavljajo objekti izpeljani iz obstoječih osnovnih razredov simulacijske knjižnice.

3.2.2 Sestavljeni moduli

Enostavne module lahko združujemo v sestavljene ter na ta način dosežemo hierarhično gnezdenje komponent. Najvišji nivo hierarhije predstavlja poseben sestavljen modul imenovan Omrežje, katerega notranjost sestavljajo enostavni ali sestavljeni moduli. Sestavljeni moduli ne vsebujejo dejanske implementacije, ampak služijo le kot vsebovalniki podmodulov 3.1.



Slika 3.1 Sestavljeni modul, enostavni moduli in omrežje.

3.2.3 Sporočila

Moduli lahko med seboj komunicirajo s predajo sporočil, ki lahko potujejo preko povezav med moduli ali so poslana direktno do ciljne komponente. Sporočila v OMNeT++ lahko predstavljajo pakete, dogodke, opravila, stranke ali kakršnekoli druge entitete. Predstavljena so kot objekti razreda `cMessage` ali njegovega podrazreda `cPacket`. Takšna sporočila, bolje poimenovana paketi, vsebujejo poleg vseh lastnosti navadnih sporočil še ostale lastnosti namenjene opisu paketov. Nekatere od njih so sledeče:

- *velikost paketa v bitih ali zlogih*: simulacijsko jedro vrednost uporablja za izračun časa prenosa paketa po prenosnem kanalu z določeno hitrostjo prenosa podatkov ali pri modeliranju napak pri prenosu bitov,
- *enkapsulacija paketa*: uporabna je pri modeliranju omrežnih protokolov,
- *zastavica bita napake*: vsebuje rezultat modeliranja napak pri prenosu, potem ko je sporočilo potovalo skozi prenosni kanal z določeno verjetnostjo napake pri prenosu paketa ali bita.

Navadna sporočila so v modulih pogosto uporabljena na način, kjer modul pošlje sporočilo samemu sebi. Na ta način realiziramo dogodke značilne za samo delovanje modula.

3.3 Opisni jezik NED

Opisovanje strukture modulov in modelov poteka s posebnim opisnim jezikom imenovanim NED (angl. *network description*). V NED datotekah tako deklariramo enostavne ali sestavljene module in s tem določimo njihovo arhitekturo. Enostavnim modulom lahko z opisnim jezikom NED določimo parametre in vrata preko katerih bo modul povezan z zunanjim svetom. Struktura sestavljenih modulov lahko poleg vrat in parametrov vsebuje še deklaracije podmodulov in morebitnih povezav med njimi. Na tem mestu omenimo, da je poleg opisa modela v jeziku NED možno model opisati še v posebni konfiguracijski datoteki s končnico *.ini*. Dvostopenjski opis modela omogoča razlikovanje modela od eksperimentov. V konfiguracijskih datotekah tako lahko definiramo eksperimente in parametre, ki jih želimo opazovati in spreminjati znotraj posameznega eksperimenta.

3.3.1 Vrata

Vrata predstavljajo povezavo modula z zunanjim svetom. Modulu lahko določimo vhodna vrata, preko katerih sporočila vstopajo v modul, izhodna iz katerih sporočila izstopajo in dvosmerna vrata, preko katerih potekata pošiljanje in sprejemanje sporočil. Število vrat lahko v modulu precej hitro naraste, zato je vsa vrata možno predstaviti kot vektorsko spremenljivko.

3.3.2 Kanali

Na voljo so trije tipi kanalov imenovani idealni kanal, kanal z zakasnitvijo in prenosni kanal. Kanali prvega tipa nimajo določenih parametrov. Sporočila poslana preko takšnih povezav do cilja prispejo brez zakasnitve ali drugih stranskih učinkov. Zakasnitvenemu kanalu lahko nastavimo dva parametra. Prvi parameter predstavlja propagacijsko zakasnitev sporočila, drugi parameter pa določa dosegljivost povezave. V primeru dosegljivega kanala bodo sporočila uspešno poslana na povezavo, v nasprotnem primeru bodo vsa sporočila zavržena. Zadnji prenosni kanal poleg vseh prej omenjenih parametrov vsebuje še hitrost prenosa podatkov oziroma pasovno širino kanala, ki se uporablja pri izračunu hitrosti prenosa paketa. Lastnosti takšnega kanala sta še verjetnost napake pri prenosu paketa ali bita.

3.3.3 Parametri

Gre za spremenljivke vezane na delovanje modulov. Parametri lahko zasedajo vrednosti različnih tipov določenih v NED ali konfiguracijskih datotekah. Prav tako je parametre možno določiti preko grafičnega vmesnika že pred samim začetkom simulacije. Enostavni moduli lahko s programsko kodo dostopajo do vrednosti parametrov ter na ta način implementirajo določeno delovanje. Tako lahko npr. čakalni vrsti, predstavljeni z enostavnim modulom, določimo parameter njene kapacitete.

3.4 Analiza in zbiranje rezultatov

OMNeT++ nudi vgrajeno podporo zapisovanju, shranjevanju in analizi simulacijskih rezultatov. Omenjeno je implementirano z uporabo izhodnih vektorjev in izhodnih skalarjev. Vektorji predstavljajo časovna zaporedja podatkov pridobljenih iz enostavnih modulov ali kanalov. Izhodne vektorje lahko uporabimo za beleženje poljubnih zakasnitev paketov, dolžin čakalnih vrst, strežnih časov, statusov modulov, izkoriščenosti povezav, števila zavrženih paketov itd. Vrednost izhodnega skalarja se izračunava tekom simulacije, končna vrednost pa je shranjena na koncu. Predstavlja lahko števce, povprečja, standardne odklone, vsote, minimalne ali maksimalne vrednosti itd.

Rezultate je možno zbirati in zapisovati na dva načina in sicer na podlagi signalov z uporabo deklariranih statistik ali zapisovanjem direktno v C++ kodi s simulacijsko knjižnico. Na kratko opišemo le uporabo druge metode, saj gre za priporočljiv pristop k meritvi

vrednosti, ki ga uporabimo tudi v našem modelu. Prvi korak uporabe signalov je deklaracija v NED datotekah, kar omogoča fleksibilnost pri izbiri kaj bomo beležili in v kakšni obliki. Poleg signalov v opisni datoteki deklariramo statistiko, katera določa opazovani signal in način zbiranja vrednosti. V zadnjem koraku v izvorni datoteki enostavnega modula najprej z metodo `registerSignal()` registriramo signal, sledi še oddaja signala v določenih trenutkih. Za beleženje statistike števila poslanih paketov bi torej enostavni modul oddal signal z vrednostjo spremenljivke števila oddanih paketov vsakič, kadar je paket poslan iz modula.

OMNeT++ je zaradi njegove generične arhitekture možno uporabiti v različnih domenah problemov, saj samo orodje ni področno orientirano, temveč lahko z uporabo obstoječe simulacijske knjižnice implementiramo poljubne gradnike, ki bodo predstavljali določeno komponento obravnavanega sistema. Modularna zasnova gradnje modelov omogoča implementacijo najosnovnejših do najzahtevnejših modelov, ki jih lahko z uporabo grafičnega vmesnika enostavno dopolnjujemo, simuliramo in analiziramo. Več o razvojnem okolju OMNeT++ si lahko bralec prebere v virih [6, 7, 9].

4 Implementacija gradnikov omrežja

V tem poglavju bomo opisali zasnovo in delovanje gradnikov implementiranih z orodjem OMNeT++. Cilj pri načrtovanju je bil predvsem čim bolj opredeliti delovanje posameznih komponent glede na potrebe obravnavanega problema in jih glede na njihovo funkcionalnost razdeliti v manjše enote, imenovane enostavni moduli. Na začetku pričujočega poglavja najprej opišemo sestavljene module ter predstavimo njihovo vsebino. Nadaljujemo s podrobnejšo predstavitvijo implementacije njihove vsebine oziroma enostavnih modulov.

4.1 Sestavljeni moduli

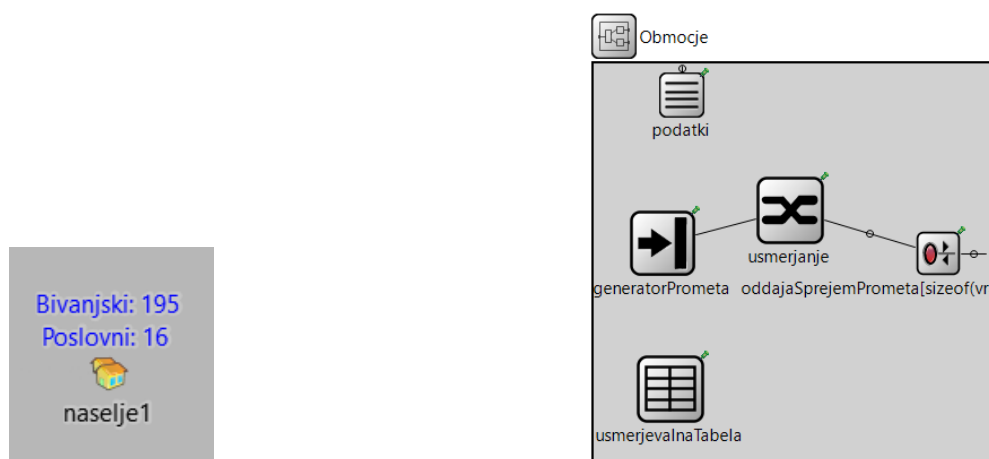
Sestavljeni moduli ne vsebujejo implementacije s programskim jezikom C++. Njihova arhitektura je določena le z opisnim jezikom NED v ustreznih datotekah. Sestavljenim modulom smo za komunikacijo z zunanjim svetom in notranjo vsebino modula določili ustrezna vrata preko katerih potekajo povezave do zunanjih ali notranjih modulov.

4.1.1 Naselje

Vsako obravnavano naselje Upravne enote Slovenj Gradec predstavimo s sestavljenim modulom tipa *Naselje*, kateremu pripnemo identifikacijsko številko naselja, določeno v tabeli 2.4. Nad modulom *Naselje* prikazanem na levi strani slike 4.1 prikažemo pripadajoče število bivanjskih in poslovnih uporabnikov. Takšen sestavljen modul ne vsebuje podmodulov, saj smo ga namenili predvsem sami grafični predstavitvi naselja in njegovemu priklopu na vozlišče imenovano območje.

4.1.2 Območje

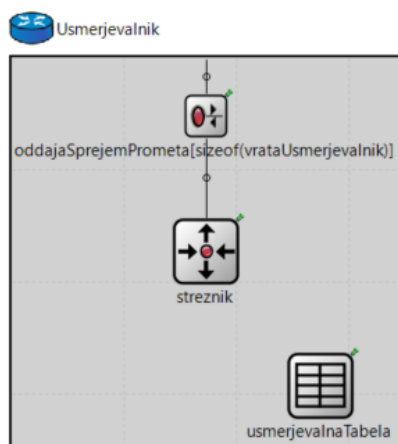
Komponenta *Obmocje* predstavlja vozlišče na katero lahko priključimo posamezna naselja in usmerjevalnike. Na desni strani slike 4.1 predstavimo podmodule, ki so vsebovani v obravnavanem modulu. Vsak modul tipa *Obmocje* je torej sestavljen iz enostavnega modula *GeneratorPrometa*, v katerem generiramo dejanske pakete. Generirani paketi so preko izhodnih vrat posredovani naprej na enostavni modul *Usmerjanje*, ki glede na naslov prejemnika v paketu določi na katera vrata naslovimo paket naprej. Tako imamo za vsako povezavo iz območja do usmerjevalnika postavljen sestavljeni modul *Oddaja-SprejemPrometa*, preko katerega paketi zapustijo modul *Obmocje*. Modul sestavljata še enostavna modula *Podatki* in *UsmerjevalnaTabela*. Funkcionalnosti posameznih podmodulov opišemo v nadaljevanju.



Slika 4.1 Naselje Brda z identifikacijsko številko 1 in prikaz vsebine sestavljenega modula *Obmocje*.

4.1.3 Usmerjevalnik

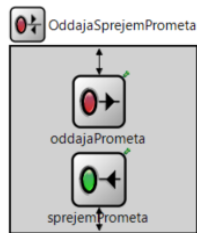
Centralna komponenta modula Usmerjevalnik je strežnik na katerega so priklopljeni sestavljeni moduli, ki igrajo vlogo oddajno sprejemne enote, deklarirane kot `oddajaSprejemPrometa`. Vsebina obravnavanega modula je prikazana na sliki 4.2. Posamezno oddajno sprejemno enoto v notranjosti predstavljata enostavna modula za oddajo in sprejem prometa. Prvi skrbi za pošiljanje paketov poslanih s strani strežnika. Pošiljanje na kanal izvaja le v primeru prostega kanala, drugače paket zadrži v čakalni vrsti. Sprejemni modul sprejme pakete poslane iz zunanjega sveta ter skupaj s strežnikom korigira prehod paketov v njegovo obdelavo. Strežnik za določeno časovno enoto prihajajoč paket zadrži ter ga na podlagi usmerjevalne tabele usmeri na ustrezna vrata. Časovna enota imenovana strežni čas paketa je implementirana kot parameter enostavnega modula tipa `Streznik`. Sestavljeni modul Usmerjevalnik ima v opisni datoteki NED definirana vektorska dvosmerna vrata, ki so namenjena priklopu drugih usmerjevalnikov ali območij.



Slika 4.2 Prikaz komponent sestavljenega modula Usmerjevalnik.

4.1.4 Oddajno sprejemna enota

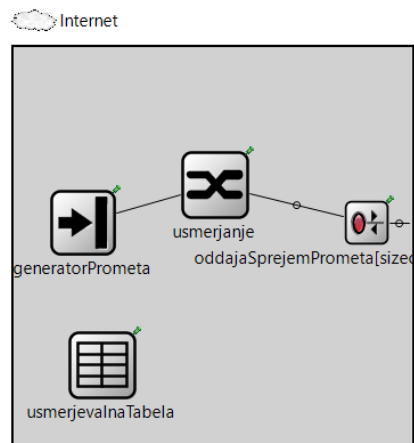
Oddajno sprejemno enoto s slike 4.3 predstavljata enostavna modula `oddajaPrometa` in `sprejemPrometa`. Ne glede na to, ali se nahajata v modulu tipa `Usmerjevalnik` ali `Obmocje`, lahko pakete sprejemata in pošiljata, zato sta oba z nadmodulom povezana preko izhodnih in vhodnih vrat.



Slika 4.3 Prikaz vsebine enostavnega modula `OddajaSprejemPrometa`.

4.1.5 Internet

Gre za ciljno vozlišče, na katerega so naslovljeni vsi paketi generirani s strani modulov tipa `Obmocje`. Struktura modula `Internet` prikazanega na sliki 4.4 je ravno takšna, kot pri modulu `Obmocje`, sprememba delovanja pa se odraža le pri generiranju prometa. Takšen modul namreč skrbi za generiranje paketov proti območjem (angl. *download*), medtem ko moduli `Obmocje` generirajo lastno količino prometa, ki jo razpošljejo v omrežje (angl. *upload*).



Slika 4.4 Prikaz vsebine sestavljenega modula `Internet`.

4.2 Enostavni moduli

Delovanje enostavnih modulov je opisano s programskim jezikom C++. Vsi moduli dedujejo iz razreda `cSimpleModule`, preko katerega ustvarjeni razred pridobi virtualne metode za delo z moduli. Največkrat uporabljeni metodi, ki ju poleg konstruktorjev in destruktorjev implementiramo sami, sta `initialize()` in `handleMessage()`. Prva metoda predstavlja

inicializacijsko fazo izvedeno pred začetkom simulacije. Razlog za obstoj omenjene metode je ta, da vse kode vezane na začetno delovanje modula ne moremo implementirati v konstruktorju, saj se konstruktorji izvedejo pred samo postavitvijo simulacijskega modela. Metoda za rokovanje s sporočili se kliče ob vsakemu prejemu sporočila. Enostavne module torej ponazorimo z izvorno kodo in z opisnim jezikom NED, s katerim modulu določimo vrata in morebitne parametre.

4.2.1 Podatki

Enostavni modul **Podatki** je namenjen branju in pripravi podatkov pridobljenih iz drugega poglavja. Podatke smo zaradi lažjega branja prepisali v XML datoteke, saj je v OMNeT++ programski knjižnici na voljo razred za delo z njimi. Na podlagi identifikatorjev priklopljenih naselij na vozlišče, izračunamo vsoto števila bivanjskih uporabnikov in vsoto poslovnih uporabnikov. Vsoti števil utežimo s pripadajočima funkcijama ocenjene količine prometa. Na ta način pridobimo količino prometa vezanega na območje v katerem se nahajamo. Izračunane podatke shranimo v lokalno tabelo modula, kjer hranimo 24 vrednosti za količino prometa generiranega s strani območja in 24 vrednosti za promet generiran proti pripadajočemu območju. Število 24 ponazarja urne vzorce bremena skozi dan.

4.2.2 Generator prometa območje

Vsa sporočila poslana v smeri od uporabnika (angl. *upload*) se ustvarijo v enostavnem modulu imenovanem generator prometa. Takšen modul lahko dostopa do metod modula tipa **Podatki**, ki je opisan v prejšnjem poglavju. Glede na čas, v katerem se simulacija trenutno nahaja, izmerjeno količino prometa ponazorimo s paketom velikosti iz vzorca bremena tekoče ure. Vsem paketom poleg velikosti določimo še čas kreiranja, naslova pošiljatelja in prejemnika. Naslov prejemnika je za vse generirane pakete v modulu **Območje** enak, saj vse pakete naslovimo modulu **Internet**. Dogodke vezane na delovanje modula realiziramo s pomočjo sporočil, ki jih modul pošlje samemu sebi. Prvi takšen dogodek se zgodi ob preteku vsake ure izvajanja simulacije. Takrat se namreč lahko količina prometa spremeni, znotraj urnih intervalov pa ostaja enaka. Medprijhodni čas med generiranjem paketov prav tako realiziramo s takšnimi sporočili, kjer vrednost parametra lahko poljubno nastavljam. Modul poleg generiranja paketov deluje še kot končna postaja paketov poslanih s strani modula **Internet**.

4.2.3 Generator prometa internet

Modul se nahaja znotraj sestavljenega modula **Internet**, ki skrbi za sprejem vseh paketov poslanih iz modulov tipa **Obmocje**. Poleg sprejema takšnih paketov modul generira promet v smeri do uporabnikov (angl. *download*). Podobno kot v prejšnjem modulu dogodke ob poteku izvajanja ene ure simulacije in medprihodnem času realiziramo s sporočili poslanimi samemu modulu. Modul glede na uro, v kateri simulacija trenutno teče, generira pakete, ki jim poleg velikosti prebrane iz modula **Podatki** določenega območja dodelimo še naslov pošiljatelja in prejemnika, ki ga trenutno obravnavamo. Takšen modul bo torej v določenem trenutku generiral toliko paketov, kot je število vseh območij v omrežju ter dostopal do njihovih metod za pridobitev količine prometa. Po drugi strani pa bodo vsa sporočila poslana s strani območij pristala v tem modulu.

4.2.4 Oddaja prometa

Primarna funkcionalnost modula **oddajaPrometa** je poslušanje kanala, oziroma izhodne povezave na katero pošiljamo pakete. Paket je možno poslati naprej le v primeru, da se v tistem trenutku ne prenaša nobenih podatkov. V našem primeru želimo zagotoviti princip, da najprej postrežemo zahtevo, ki je prva prispela v modul (angl. *FIFO - first in, first out*). Modul ima v ta namen vgrajeno čakalno vrsto, v katero ob zasedenem kanalu vstavimo pakete. Delovanje ob sprejemu sporočila je torej takšno, da najprej preverimo, če smo prejeli sporočilo, ki ga je modul poslal samemu sebi. Omenjeno sporočilo ponazarja dogodek ob koncu prenosa nekega paketa. Prejem takšnega sporočila pomeni, da se je kanal sprostil, zato lahko naprej pošljemo prvi paket iz čakalne vrste, če seveda le-ta ni prazna. Ko modul prejme dejanski paket, najprej preverimo stanje kanala. Če je povezava trenutno zasedena, paket postavimo na konec čakalne vrste. V primeru prostega kanala na povezavo pošljemo prvi paket iz čakalne vrste oziroma trenutno prispeli paket, če čakalna vrste ne vsebuje drugih paketov.

4.2.5 Sprejem prometa

Enostavni modul **sprejemPrometa** predstavlja prvo postajališče paketa poslanega modulu tipov **Internet**, **Obmocje** ali **Usmerjevalnik**. Na tem mestu omenimo, da delovanje modula za sprejem prometa pride do izraza le v sestavljenem modulu **Usmerjevalnik**. Predpostavimo namreč, da je čas obdelave paketov prejetih v modulih **Internet** in **Obmo-**

cje ničelen, zato modul `sprejemPrometa` ob sprejemu paket takoj posreduje na izhodna vrata. Takšen modul v usmerjevalnikih komunicira s strežno enoto in lahko paket pošlje naprej le, če je strežnik prost. V ta namen uporabimo čakalno vrsto, v katero ob zasedenem strežniku shranjujemo na novo prispеле pakete. V razredu modula smo implementirali metodo `zahtevajPaket()`, katera je klicana s strani strežnika in katere klic oznanja sprostitvev strežne enote, zato lahko izbrani modul paket pošlje v obdelavo. Modul poleg ostalih metod vsebuje še metodo `vrniStreznikProst()`, preko katere dostopamo do objekta tipa `Streznik` oziroma njegove metode za določanje stanja strežnika.

4.2.6 Strežnik

V tem modulu vnesemo čas obdelave vsakega prispellega paketa. Vrednost strežnega časa določimo kot parameter, katerega vrednost lahko preberemo iz NED ali konfiguracyjske datoteke. Vsak modul vsebuje podatek o številu obdelanih paketov in kazalec na usmerjevalno tabelo. V modulu implementiramo sporočilo, ki ga modul pošlje samemu sebi po preteku časa obdelave paketa. V kolikor modul prejme takšno sporočilo, iz paketa končanega z obdelavo, pridobi podatek o naslovu prejemnika. V nadaljevanju s klicem metode kontaktiramo usmerjevalno tabelo, ki vrne podatek o tem na katera vrata oziroma oddajno enoto moramo posredovati paket, da bo le-ta prispel do cilja. S preusmeritvijo paketa na ustrezna vrata je obdelava paketa končana, zato po principu krožne discipline (angl. *round robin*) izberemo neprazno čakalno vrsto iz modula tipa `sprejemPrometa`, ki lahko strežniku posreduje paket. Omenjeno implementiramo s klicem funkcije izbranega modula `zahtevajPaket()`. V tem trenutku postane strežnik rezerviran, zato sprejem paketa z drugih modulov ni mogoč.

4.2.7 Usmerjanje

Modul `Usmerjanje` se nahaja v sestavljenih moduli tipa `Obmocje` in `Internet`, kjer smo predpostavili ničelen strežni čas. V omenjenih gradnikih torej nimamo strežne enote za katero bi morali preverjati kdaj je le-ta prosta. Modul ob vsakem na novo prispeltem paketu preveri njegovo vsebino. Natančneje prebere naslov prejemnika ter ga na podlagi usmerjevalne tabele usmeri na ustrezna vrata.

4.2.8 Usmerjevalna tabela

Usmerjevalna tabela služi kot shramba vrednosti kot so naslov cilja, število skokov do cilja in identifikacijsko številko vrat na katera moramo posredovati paket, da bo prispel do cilja. Razred vsebuje metodo `usmeriPaketNaVrata()`, katera kot vhodni parameter prejme naslov prejemnika in vrne identifikacijsko številko vrat na katera mora strežnik poslati paket. Število naslovnikov v našem primeru ni pretirano veliko, zato takšna usmerjevalna tabela vsebuje celotno množico naslovov, kamor je možno dostaviti paket. Postopek nastavitve usmerjevalne tabele opišemo v nadaljevanju.

4.2.9 Konfiguracija omrežja

Namen modula `konfiguracijaOmrezja` je vzpostavitev vseh usmerjevalnih tabel nahajajočih se v modulih tipov `Internet`, `Obmocje` in `Usmerjevalnik`. Uporabili smo razred `cTopology` namenjen implementaciji usmerjanja v komunikacijskih omrežjih. Objekt takšnega tipa predstavlja abstrakcijo omrežja predstavljenega s podatkovno strukturo graf, ki jo sestavljajo vozlišča in povezave med njimi. Razred implementira vrsto metod s katerimi omrežje definirano v jeziku NED preslikamo v podatkovno strukturo graf. Tako lahko v strukturo zajamemo vse module našega omrežja ali uporabimo le module, ki ustrezajo določenim kriterijem. Vsako vozlišče grafa ustreza enostavnemu ali sestavljenemu modulu in vsaka povezava grafa ustreza povezavi ali zaporedju povezav med njimi. Za uporabo omenjenega razreda smo se odločili predvsem zaradi iskanja najkrajših poti v grafu, ki je implementirana v samem razredu. Metoda `calculateUnweightedSingleShortestPathsTo(targetnode)` poišče najkrajše poti od vseh vozlišč do ciljnega vozlišča, podanega v parametru metode. V ozadju je implementiran algoritem Dijkstra, katerega rezultat je drevo najkrajših poti. Omenjena metoda predpostavlja, da imajo vse povezave enako ceno, kar uporabimo tudi v našem primeru. Ob inicializaciji modula `konfiguracijaOmrezja` najprej module shranimo v objekt tipa `cTopology`. V podatkovno strukturo zajamemo le module tipov `Obmocje`, `Internet` in `Usmerjevalnik`, saj moduli tipa `Naselje` ne sodelujejo pri generiranju ali usmerjanju prometa. V nadaljevanju se v zanki sprehodimo čez vsa vozlišča grafa, kjer za vsako vozlišče izračunamo najkrajše poti do tega vozlišča. Znotraj posamezne iteracije sledi obisk vseh preostalih vozlišč oziroma modulov katerim v usmerjevalno tabelo dodamo zapis najkrajše poti do obravnavanega vozlišča. Najkrajšo pot predstavlja naslov vrat, ki vodi do prvega vozlišča na najkrajši poti. Poleg vzpostavi-

tve usmerjevalnih tabel v obravnavanem modulu implementiramo še določitev kapacitete posameznih povezav, ki vodijo iz modula *Obmocje* do prvega usmerjevalnika na poti. Hitrost prenosa računamo posebej za izhodni in vhodni kanal tako kot smo določili v poglavju 2.5.

V tem poglavju smo predstavili enostavne in sestavljene module, ki predstavljajo gradnike našega modela omrežja. Sestavljeni moduli služijo kot vsebovalniki, v katere smo zapakirali enostavne module. Delovanje enostavnih modulov smo implementirali s programskim jezikom C++ in predstavili njihovo delovanje.

5 Model omrežja

V pričujočem poglavju opišemo model našega omrežja. Zgradimo ga iz enostavnih in sestavljenih modulov opisanih v prejšnjem poglavju. Pri tem se osredotočimo na samo postavitev in ureditev jedra omrežja ter povezav med zalednimi vozlišči v omrežju. Topologijo omrežja je smiselno izbrati glede na velikost omrežja, zato pričetek tekočega poglavja namenimo opisu različnih vrst in topologij omrežij ter nadaljujemo z umestitvijo komponent na opazovano območje občin Slovenj Gradec in Mislinja.

5.1 Vrste in topologije omrežij

Računalniško omrežje definiramo kot sistem medsebojno neodvisnih omrežnih naprav, kjer si vozlišča preko žičnih ali brezžičnih povezav izmenjujejo podatke in delijo periferne vire. Omrežja lahko različno okarakteriziramo glede na njihovo velikost, namen ali uporabljeno topologijo.

Velikost omrežja določa oddaljenost naprav v omrežju ter število pripadajočih računalnikov in omrežnih naprav. Najmanjše omrežje torej predstavljata dva računalnika, ki ju lahko povežemo v večje omrežje na nivoju prostorov, stavb in mest ter ju tako vključimo v izme-

njavo podatkov javnega medmrežja imenovanega internet. Omrežja namenjena uporabi posameznikov ali manjšemu številu uporabnikov imenujemo osebna (angl. *PAN* - *personal area network*) ali domača omrežja (angl. *HAN* - *home area network*). Glede na velikost sledijo lokalna omrežja (angl. *LAN* - *local area network*), katera pogosto povežemo v domeno krajevnih omrežij (angl. *MAN* - *metropolitan area network*). Omrežja, ki pokrivajo zelo velike geografske površine kot so države ali celine, imenujemo prostrana omrežja (angl. *WAN* - *wide area network*). Primer takšnega omrežja je internet, ki je sestavljen iz velikega števila krajevnih ali lokalnih omrežij.

Na tem mestu predstavimo še vrste topologij omrežij, njihove prednosti in slabosti. Topologija omrežja se nanaša na ureditev ali fizično postavitve računalnikov, fizičnih povezav in drugih omrežnih komponent v omrežju [11]. Najpreprostejša struktura je topologija vodila, kjer vse naprave za prenos podatkov uporabljajo skupni medij. Prednost takšne zasnove sta relativno preprosta inštalacija in cena, saj za vzpostavitev ne potrebujemo velikega števila fizičnih povezav. Naprave pripravljene za prenos podatkov najprej razpošljejo (angl. *broadcast*) pakete vsem ostalim računalnikom v omrežju, zaradi česar ob priklopu večjega števila naprav prihaja do kolizij podatkov ter posledično upočasnjenega delovanja sistema. Poleg omenjene omejitve sta slabosti takšne topologije še pomanjkljiva varnost in izpad celotnega omrežja v primeru odpovedi fizične povezave. Alternativna možnost postavitve omrežja je v obliki krožne topologije (angl. *ring network*), v kateri je vsaka enota povezana še z dvema napravama. V primerjavi s topologijo vodila tukaj ne prihaja do kolizij podatkov, zato priklop večjega števila naprav ni problematičen. Kot tretjo navajamo zvezdno topologijo, v kateri so naprave povezane s centralnim vozliščem. Slednje lahko predstavljajo stikala, usmerjevalnike ali razdelilnike. V primeru odpovedi centralnega vozlišča pride do izpada celotnega omrežja, vendar le-to ostane stabilno, če odpove posamezna povezava ali katerokoli drugo necentralno vozlišče. Najdražja, vendar proti odpovedim posameznih komponent najbolj odporna zasnova omrežja je v obliki polne topologije, kjer je vsako vozlišče povezano z vsemi preostalimi. Velika večina omrežij je zasnovana s hibridnimi topologijami, kjer so združene karakteristike osnovnih topologij.

5.2 Smernice za postavitve modela omrežja

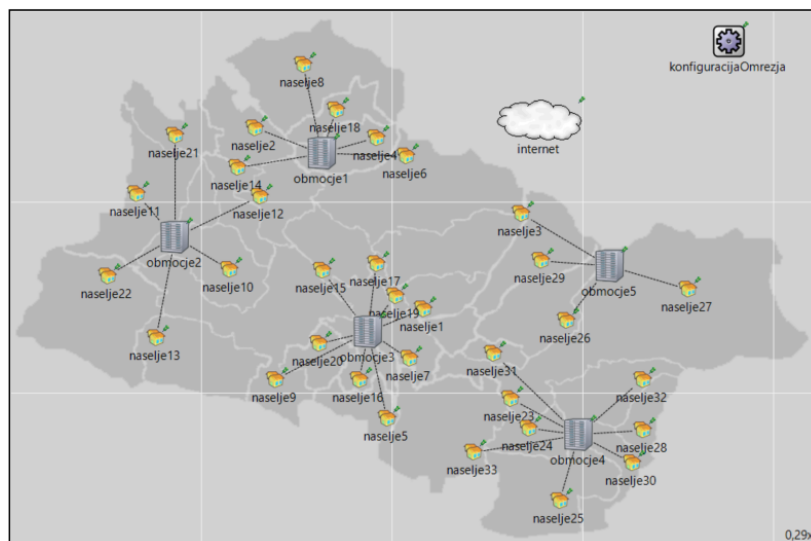
Naš model omrežja pokriva večje področje občin Slovenj Gradec in Mislinja, zato glede na velikost govorimo o bolj ali manj krajevnem tipu omrežja (angl. *MAN - metropolitan area network*). Pri načrtovanju našega modela smo želeli slediti smernicam fizične postavitve krajevnih, prostranih ali ISP omrežij, vendar oprijemljivih napotkov zanje nismo našli. Zaradi pomanjkanja podatkov smo za osnovo uporabili načrtovanje kampus ali šolskega omrežja (angl. *CAN - campus area network*) iz vira [12]. V omenjenem viru je kampus omrežje hierarhično in modularno zasnovano, kjer so posamezne komponente v hierarhiji porazdeljene po nivojih imenovanih plasti. V takšni arhitekturi govorimo o jedrni ali hrbtenični plasti (angl. *core layer*), plasti dostopa (angl. *access layer*) in porazdeljeni plasti (angl. *distribution layer*). Prednost takšne zasnove modela so njegova hierarhična struktura, modularnost, stabilnost in skalabilnost sistema. Število potujočih entitet v hierarhiji in z njimi povezane zahteve po pasovnih širinah se tako s potovanjem od najnižjega sloja hierarhije proti višjemu sloju večajo.

Najvišje v hierarhiji se nahaja hrbtenična plast, ki zagotavlja internetno povezljivost ostalim napravam na nižjih nivojih, zato morajo biti komponente jedra hitre in ekstremno odporne. Pri načrtovanju modela omrežja moramo upoštevati morebitna vzdrževalna dela ter odpovedi posameznih povezav in vozlišč. Eden izmed pristopov k povečanju stabilnosti sistema je uporaba redundantnih povezav. Redundantnost v sistemu lahko dosežemo s polno povezano topologijo, vendar takšna zasnova ne zagotavlja konsistentne konvergence v primeru izpada povezave ali vozlišča. Konvergenca pomeni stanje v omrežju, ko vse usmerjevalne tabele v usmerjevalnikih vsebujejo skladne informacije o topologiji omrežja. Slabosti polno povezane topologije sta tudi težavna vzpostavitev ter vzdrževanje usmerjevalnih tabel in velikega števila vmesnikov naprav v jedrih, kar poveča ceno in kompleksnost omrežja. Iz članka [12] razberemo, da naj bodo redundantne povezave v jedrih zasnovane z uporabo trikotne topologije, s katero dosežemo hitrejšo in bolj deterministično konvergenco. Nivo nižje od plasti jedra se nahaja porazdeljena plast, v katero se stekajo vozlišča na najnižjem dostopovnem nivoju. Porazdeljena plast v glavnem ščiti jedro pred visoko obremenitvijo ter zagotavlja logično izolacijo posameznih dostopovnih enot. Visoko dosegljivost komponent v porazdeljeni plasti dosežemo z uporabo (redundantnih) enakih povezav do naprav na dostopovni in jedrni plasti. Uporaba dvojnih povezav predstavlja usmerjevalno strategijo pri kateri se naslednji skok

paketa (angl. *next-hop*) določa preko dveh najboljših poti (povezav z najnižjo ceno). Implementacija takšne strategije usmerjevalnikom omogoča uravnavanje prometa (angl. *load balance*), torej lahko usmerjevalnik v izogib zamašitve na določeni povezavi, paket posreduje na drugo povezavo z enako ceno. Na najnižjem nivoju hierarhije imamo plast dostopa, katera predstavlja prvo točko vstopa končnih naprav v omrežje. V naši zasnovi modela omrežja dostopovne plasti ne bomo obravnavali, saj z generiranjem prometa začnemo v sami porazdelitveni plasti in nas v nadaljevanju zanima pretočnost med jedrom in porazdeljeno plastjo.

5.3 Postavitev modela omrežja

Načrtovanje modela omrežja pričnemo z zasnovo porazdelitvene plasti, namenjene združitvi posameznih naselij. Slednjo ponazorimo z moduli tipa *Naselje*, katerim dodelimo enolični identifikator določen v tabeli 2.4 in jih nanesemo na pripadajoči prostor v orisu občin Slovenj Gradec in Mislinja. Vsako naselje predstavljata oceni števila bivanjskih in poslovnih uporabnikov. Eden izmed ciljev zasnove modela omrežja je izolacija naselij, s katero olajšamo vzdrževanje omrežja in izboljšamo stabilnost sistema. Izolacijo smo dosegli tako, da smo opazovano površino razdelili na približno pet enako velikih delov, posamezni del pa smo poimenovali območje. V večini primerov smo naselja glede na njihovo oddaljenost priključili najbližjemu območju, kar je prikazano na sliki 5.1.



Slika 5.1 Model omrežja povezanih naselij.

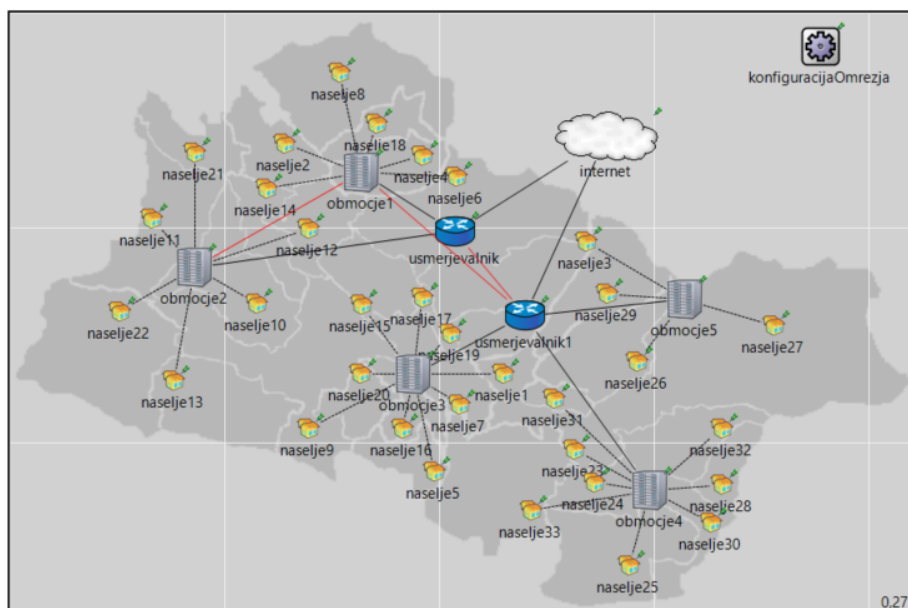
Moduli tipa **Obmocje** najprej na podlagi števila uporabnikov poskrbijo za izračun količine prometa od samega modula (angl. *upload*) in izračun količine prometa do njega (angl. *download*). Takšni moduli skrbijo za generiranje paketov naslovljenih izven našega omrežja, torej modulu **Internet**, medtem ko modul **Internet** pošilja pakete, ki vstopajo v naše omrežje. Posamezna naselja so na centralno vozlišče **Obmocje** simbolično povezana v obliki zvezdne topologije. Povezave med naselji in območjem črtkano ponazorimo, saj po takšnih povezavah prometa v modelu omrežja ne prenašamo.

Tabela 5.1 prikazuje število uporabnikov povezanih na posamezno območje, iz katere lahko razberemo, da bodo največje količine prometa generirane s strani modulov **obmocje2**, **obmocje1** in **obmocje3**.

Naziv območja	Bivanjski uporabniki	Poslovni uporabniki	Skupaj
obmocje1	2.836	667	3.503
obmocje2	5.602	2.687	8.289
obmocje3	2.586	342	2.928
obmocje4	1.651	150	1.801
obmocje5	1.544	152	1.696

Tabela 5.1 Število bivanjskih in poslovnih uporabnikov po območjih.

Nadaljujemo s postavitvijo najbolj obremenjenega dela, torej gradnikov v hrbtenični ali jedrni plasti. V hrbtenično plast našega modela omrežja vključimo dva usmerjevalnika, preko katerih generirani paketi zapuščajo model omrežja in vanj vstopajo. Na usmerjevalnika priključimo najbližja območja ter ju razbremenimo tako, da vsakemu usmerjevalniku določimo povezavo do modula **Internet**. Naša implementacija modula za vzpostavitev usmerjevalnih tabel ne predpostavlja dinamičnega usmerjanja, zato priporočljive redundantne povezave v plasti jedra in porazdeljevanja nakažemo na posameznem primeru. Redundantne povezave vključene v trikotno topologijo so na sliki 5.2 obarvane z rdečo barvo. Zaradi statične nastavitve usmerjevalnih tabel pred začetkom simulacije odpovedi posameznih komponent v modelu ne bomo simulirali, temveč se bomo osredotočili na prepustnost povezav po katerih se bodo dejansko prenašali paketi.



Slika 5.2 Model omrežja z nakazanimi redundantnimi povezavami.

V tabeli 5.2 prikažemo planirane zmogljivosti pasovnih širin posameznih povezav v smeri proti posameznemu območju (angl. *download*) in planirane pasovne širine v smeri iz območja (angl. *upload*).

Oznaka povezave	Download [Mbps]	Upload [Mbps]
obmocje1-usmerjevalnik1	1.368	579
obmocje2-usmerjevalnik1	2.709	1.146
usmerjevalnik1-internet	4.077	1.725
obmocje3-usmerjevalnik2	1.246	528
obmocje4-usmerjevalnik2	795	337
obmocje5-usmerjevalnik2	744	315
usmerjevalnik2-internet	2.785	1.179

Tabela 5.2 Tabela ocenjenih pasovnih širin posameznih povezav.

Povezave med moduli tipa Območje in Usmerjevalnik prepišemo iz tabele 2.5. Kapacitete povezav med zunanjim svetom in našim omrežjem določimo kot vsoto ocenjenih pasovnih širin povezav, ki vodijo iz usmerjevalnika do posameznih modulov tipa *Območje*. Zmogljivost fizične povezave usmerjevalnik1-internet iz tabele 5.2 je določena kot vsota

planiranih pasovnih širin povezav `obmocje1-usmerjevalnik1` in `obmocje2-usmerjevalnik1`.

5.4 Generiranje prometa

V tem razdelku opišimo še način generiranja prometa, saj uporabljena metoda ne predstavlja dejanskega delovanja omrežij. Količino omrežnega prometa posameznega območja smo določili glede na enačbi 5.1 in 5.2. Prva enačba predstavlja količino prenosa i -tega območja v smeri proti območju (angl. *download*). Izračunana je kot utežena vsota izmerjenih funkcij posameznega bivanjskega uporabnika s slike 2.4 (N_{down}) in poslovnega uporabnika s slike 2.6 (P_{down}). Enačba 5.2 predstavlja količino generiranega prometa s strani i -tega območja (angl. *upload*), ki je določena kot utežena vsota izmerjenih *upload* funkcij posameznega tipa uporabnikov (N_{upload} in P_{upload}). Uteži predstavljajo oceno števila bivanjskih in poslovnih uporabnikov i -tega območja, določenih v tabeli 5.1 ($st_{i,navadni}$ in $st_{i,poslovni}$).

$$obmocje_iDownload = st_{i,navadni} * N_{down} + st_{i,poslovni} * P_{down} \quad (5.1)$$

$$obmocje_iUpload = st_{i,navadni} * N_{upload} + st_{i,poslovni} * P_{upload} \quad (5.2)$$

Na ta način smo pridobili vrednosti količine prometa območja v urnih intervalih. V nadaljevanju količino prometa območja simuliramo tako, da vsako sekundo simulacije generiramo en paket, s katerim ponazorimo količino predvidenega prometa. Velikost posameznega paketa v sekundnem intervalu torej ustreza ocenjeni količini prometa posamezne sekunde.

Za takšno generiranje paketov smo se odločili, ker naše omrežje predstavlja veliko število uporabnikov, ki skupaj generirajo precejšnjo količino prometa. Bolje bi bilo, če bi že v prvi fazi zbiranja podatkov pridobili podatke o številu paketov poslanih s strani uporabnika in število prejetih paketov. Čeprav bi podatke lahko pridobili, bi bilo generiranje tj. ustvarjanje velikega števila objektov tipa `cPacket` prostorsko in časovno zelo požrešno.

5.5 Zagon simulacije

Pred zagonom simulacije pripravimo inicializacijsko datoteko, v kateri lahko manipuliramo z vhodnimi podatki posameznih eksperimentov. Koda 5.1 prikazuje izsek pripra-

vljene inicializacijske datoteke.

Koda 5.1 Inicializacijska datoteka modela omrežja.

```
[General]
network = omrezjeISP
sim-time-limit = 24h
**.podatki.datotekaPrometNavadni = xmldoc("podatki/prometNavadni.xml")
**.podatki.datotekaPrometPoslovni = xmldoc("podatki/prometPoslovni.xml")
**.podatki.datotekaUporabniki = xmldoc("podatki/users.xml")

**.channel.utilization.result-recording-modes = vector, max, mean
**.channel.utilization.statistic-recording = true

**.casVcakalniVrsti.result-recording-modes = vector, max, mean
**.casVcakalniVrsti.statistic-recording = true

**.casPotovanjaPaketa.result-recording-modes = vector, max, mean
**.casPotovanjaPaketa.statistic-recording = true

[Config Simulacija1]
**.medPrihodniCasPrometa = 1s
**.strezniCasPaketa = 5us
**.konfiguracijaOmrezja.hitrostPovezave = 1
```

Prvi del kode znotraj značke `[General]` vsebuje lastnosti in parametre, ki so skupni vsem definiranim eksperimentom. Na tem mestu določimo omrežje poimenovano `omrezjeISP` predstavljeno na začetku tekočega poglavja. Sledi nastavev časa trajanja simulacije, kjer vrednost spremenljivke nastavimo na 24 ur. V nadaljevanju določimo poti do datotek v katerih smo shranili podatke kot so število bivanjskih in poslovnih uporabnikov posameznega naselja ter ocenjeni količini prometa posamezne skupine uporabnikov. Odločili smo se za beleženje statistik uporabnostnega faktorja povezav (angl. *utilization factor*), čakalnih časov paketov in časov potovanja paketov od pošiljatelja do prejemnika. Za posamezno metriko v inicializacijski datoteki določimo način beleženja vrednosti, kjer poleg maksimalnih in povprečnih vrednosti zbiramo še podatke v vektorski obliki. Vektorska oblika podatkov zajema čase meritev statistik in njihovih vrednosti.

V drugem delu inicializacijske datoteke nanizamo posamezne eksperimente z različnimi vrednostmi vhodnih spremenljivk. Na tem mestu ustvarimo konfiguracijo poimenovano `[Config Simulacija1]` v kateri deklariramo spremenljivke medprihodnega časa generiranja prometa, strežnih časov in faktor zmanjšanja ali povečanja hitrosti povezav. Vrednost

medprihodnih časov na vseh modulih z lastnostjo generiranja paketov nastavimo na 1 sekundo, kar pomeni, da bodo moduli tipov Internet in Obmocje generirali paket enkrat na sekundo. Referenca `**generatorPrometa.medPrihodniCasPrometa` pomeni nanašanje na vse module z določeno lastnostjo `medPrihodniCasPrometa`. Čas procesiranja na usmerjevalnikih oz. njihovih strežnih enotah nastavimo na $5 \mu\text{s}$. Zadnji parameter `hitrostPovezave` določa faktor zmanjšanja ali povečanja ocenjenih hitrosti povezav. V prvotnem eksperimentu nas zanima pretočnost našega modela omrežja z ocenjenimi hitrostmi prenosa, zato v prvi konfiguraciji hitrosti povezav ne zmanjšamo ali povečamo.

Slika 5.3 prikazuje stanje našega modela omrežja po končanem izvajanju 24 urne simulacije.



Slika 5.3 Stanje modela omrežja po izvedbi prvega simulacijskega eksperimenta.

S posameznih modulov lahko poleg ocenjenih hitrosti povezav razberemo število poslanih in prejetih paketov. Opazimo, da zaradi določenega determinističnega medprihodnega časa paketov, posamezen modul generira tolikšno število paketov, kot traja čas izvajanja celotne simulacije v sekundah (tj. 84.000 paketov). Modul Internet generira v določeni časovni enoti tolikšno število paketov, kot je število območij v omrežju, zato je njegovo število poslanih paketov enako produktu generiranih paketov na območje in števila območij v sistemu. Opazimo, da so vsi moduli tipa Obmocje prejeli toliko paketov kot so jih poslali, kar pomeni, da je naše omrežje v 24 urnem simuliranem času bilo sposobno vse generirane *download* pakete prenesti do njihovega ciljnega naslova. Opa-

zimo lahko tudi, da modul `Internet` dveh paketov ob koncu izvajanja 24-urne simulacije še ni sprejel. Ob nadaljnjem raziskovanju smo ugotovili, da sta zadnja paketa do modula `Internet` prispela v času simulacijskega časa 84.001 s.

5.6 Metrika čakalnih časov

Obraunavana metrika se izračunava tekom potovanja posameznega paketa po njegovi prenosni poti. Paket lahko ob prenosu do njegovega cilja najprej pristane v čakalni vrsti oddajne enote modula `OddajaPrometa` v katerem je bil generiran. Ob prihodu paketa v usmerjevalnik lahko paket čaka v čakalni vrsti pred samo strežbo in po njegovi obdelavi ponovno čaka na sprostitev prenosnega kanala. Za potrebe beleženja metrike čakalnih časov smo objektom izpeljanih iz razreda `cPacket` poleg časovne značke določili še spremenljivko, ki predstavlja seštevek vseh čakalnih časov nekega paketa. Ob prihodu paketa v čakalno vrsto torej vrednost časovne značke nastavimo na čas, v katerem se simulacija trenutno nahaja, saj na ta način zabeležimo trenutek prihoda paketa v čakalno vrsto. Kadar paket čakalno vrsto zapusti s pomočjo prej določene časovne značke in trenutnega simuliranega časa, izračunamo čakalni čas in rezultat prištejemo vrednosti spremenljivke, ki predstavlja vsoto vseh čakalnih časov obravnavanega paketa. V nadaljevanju posamezne čakalne čase paketov razvrstimo glede na njihovo pripadnost modulom tipa `Obmocje`.

Oznaka območja	Čakalni čas [s]			
	Download		Upload	
	Povprečni	Maksimalni	Povprečni	Maksimalni
obmocje1	0,19	0,44	0,00	0,00
obmocje2	0,00	0,00	0,08	0,22
obmocje3	0,14	0,37	0,15	0,37
obmocje4	0,00	0,00	0,00	0,00
obmocje5	0,07	0,19	0,08	0,19

Tabela 5.3 Tabela povprečnih in maksimalnih čakalnih časov paketov po območjih.

Tabela 5.3 prikazuje povprečne in maksimalne čakalne čase paketov poslanih do posameznih območij (angl. *download*) in paketov poslanih s strani območij do skupnega ciljnega modula imenovanega `internet` (angl. *upload*). Zaradi determinističnih vhodnih podatkov našega modela omrežja kot so konstantni medprihodni časi, čas procesiranja

na strežnih napravah in konstantnih velikosti paketov znotraj urnih intervalov opazimo, da rezultati niso takšni, kot bi sprva pričakovali. Vidimo namreč, da paketi poslani s strani modula `obmocje1` ne pristanejo v nobeni čakalni vrsti, zato je njihov povprečni čakalni čas ničelen. Poleg tega lahko razberemo, da je pri modulu `obmocje2` takšen rezultat ravno obraten. Paketi poslani s strani modula `internet` na naslov modula `obmocje2` imajo povprečen čakalni čas enak 0. Na sliki 5.4 se lahko prepričamo, da je predstavljen povprečni čakalni čas paketov modula `obmocje2` posledica ničelnih čakalnih časov vseh paketov.

Vzrok takšnih rezultatov je predvsem način implementacije gradnikov in deterministično izvajanje posameznih dogodkov kot je npr. generiranje paketov. Za boljšo predstavo opišimo delovanje sistema ob njegovem izvajanju. Najprej se osredotočimo na generiranje paketov v smeri proti posameznim modulom tipa `Obmocje`. Zaradi poenostavitve razlage bomo obravnavali samo zgornjo podmnožico modela omrežja, ki vsebuje elemente `internet`, `obmocje1`, `obmocje2` in `usmerjevalnik1` ter njihove pripadajoče povezave. Modul `internet` vsako sekundo izvajanja simulacije generira dva paketa, ki ju naslovi območjema `obmocje1` in `obmocje2`. Čeprav sta paketa generirana v isti časovni enoti nista hkrati poslana na kanal, zato bomo odgovor na zgornje rezultate našli v vrstnem redu njune oddaje na povezavo. Vrstni red generiranja omenjenih paketov je odvisen od pozicije modula v podatkovni strukturi `cTopology`, s pomočjo katere obiščemo vse module našega modela omrežja, katerim moramo poslati pakete. Ob razhroščevanju (angl. *debugging*) izvajanja našega programa ugotovimo, da generator prometa v modulu tipa `internet` prvi paket naslovi na `obmocje2`. Topologija omrežja se med izvajanjem simulacije ne spreminja, zato takšni paketi vedno prvi dostopajo do resursov omrežja in zanje čakanje v čakalnih vrstah ni potrebno. Omenjeno velja samo ob predpostavki, da imamo strežne enote dovolj hitre in med njimi postavljene dovolj zmogljive povezave. Zaradi prostih omrežnih resursov paketi naslovljeni na `obmocje2` v čakalnih vrstah ne čakajo. Območje, kateremu najprej pošljemo paket, je hkrati tudi območje z največ uporabniki, čakalni čas drugega paketa pa je odvisen od hitrosti prenosa prvega po ustrezni povezavi, zato so čakalni časi paketov modula `obmocje1` odvisni od časa prenosa prejšnjega paketa.

Časi prenosov paketov po kanalih so odvisni od velikosti posameznega paketa in hitrosti povezave po kateri se prenašajo. V našem primeru smo za vsa območja kot osnovo za izračun količine prometa uporabili isti izhodiščni funkciji s slike 2.4 in slike 2.6. Funkciji smo utežili s pripadajočima številoma bivanjskih in poslovnih uporabnikov

ter na ta način določili velikosti paketov znotraj urnih intervalov. Praviloma bi naj paketi poslani s strani manj obremenjenih območij imeli manjše hitrosti prenosa, zato bi posledično prej prispeli do resursov omrežja in njihovega cilja. Prejšnja izjava seveda v celoti obvelja le v primeru modela omrežja, v katerem so vse povezave enako zmogljive. V našem primeru kapacitete povezav niso enake, temveč so določene tako, da maksimalna predvidena količina prometa predstavlja $\frac{2}{3}$ zasedenost povezave, zato imajo vsi paketi generirani s strani modulov tipa **Obmocje** do prvega usmerjevalnika zelo podobne hitrosti prenosa. Hitrosti prenosa paketov do prvega usmerjevalnika prve ure izvajanja simulacije predstavimo v tabeli 5.4.

Oznaka območja	Čas prenosa [s]	
	Download	Upload
obmocje1	0,35	0,38
obmocje2	0,35	0,38
obmocje3	0,35	0,38
obmocje4	0,35	0,38
obmocje5	0,35	0,38

Tabela 5.4 Časi prenosov paketa po povezavah do prvega usmerjevalnika.

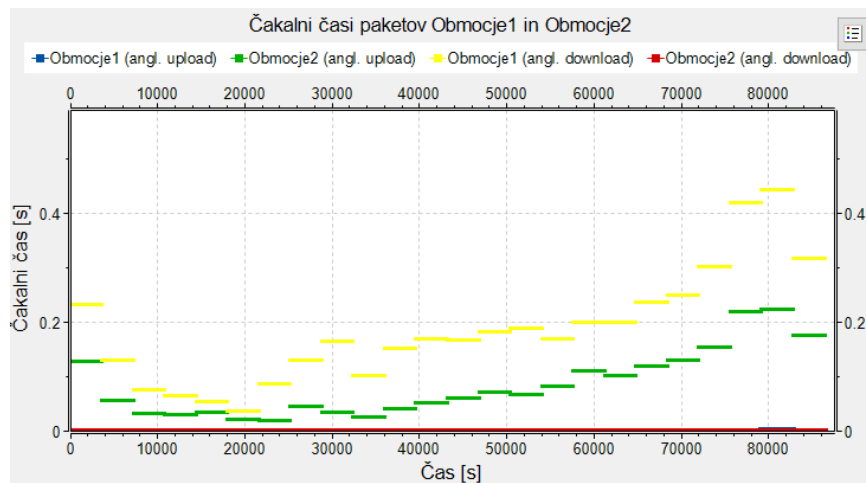
Iz tabele 5.3 razberemo, da paketi poslani s strani modula **obmocje1** ne čakajo v čakalnih vrstah. Paketi generirani s strani takšnega modula so zaradi manjšega števila uporabnikov manjši. Prav tako pa je zmogljivost povezave v primerjavi s kanalom **obmocje2-usmerjevalnik1** večja, zato omenjeni paketi do komponente na drugi strani povezave prispejo hitreje. Enak pojav lahko opazimo tudi na primeru drugega dela našega omrežja, kjer iz tabele 5.3 razberemo, da ima bolj obremenjeno **obmocje3** daljše čakalne čase v primerjavi s preostalima moduloma. Kljub temu, da ima **obmocje5** v primerjavi z modulom **obmocje4** manj uporabnikov, so čakalni časi tega modula večji. Število uporabnikov območja najdemo v tabeli 5.1. Pričakujemo lahko, da podobno velja za čase potovanja paketov od pošiljatelja do prejemnika. Metriko časov potovanj po območjih bomo predstavili v nadaljevanju.

Slika 5.4 prikazuje čakalne čase posameznih paketov poslanih in prejetih s strani modulov **obmocje1** in **obmocje2**. Zaradi determinističnega generiranja paketov in njihovih velikosti opazimo, da so čakalni časi paketov znotraj urnih intervalov konstantni. Paketi

poslani s strani modula `obmocje1` čakalno vrsto zasedajo samo v najbolj obremenjeni uri, saj so v drugih urinih intervalih velikosti generiranih paketov zaradi manjšega števila uporabnikov manjše in hitrosti pripadajočih povezav dovolj velike, da takšni paketi hitreje prispejo do resursov omrežja.

Po drugi strani paketi poslani modulu z oznako `obmocje1` v povprečju v čakalni vrsti stojijo 0,19 s, saj je bil na povezavo najprej poslan paket drugemu območju. Čakalni časi paketov naraščajo ob prehajanju izvajanja simulacijskega časa v uro z največjo količino prometa. Čakalni časi paketov poslanim modulu `obmocje1` so relativno visoki zaradi odvisnosti od velikosti predhodnih paketov, ki jih *internet* najprej naslovi na območje z največ uporabniki poimenovano `obmocje2`. Ker sta oba paketa poslana s strani modula *internet* generirana v istem trenutku in je najprej na povezavo poslan paket naslovu `obmocje2`, so čakalni časi modula `obmocje1` sestavljeni iz vsote trajanja prenosa paketov `obmocje2` in morebitnih ostalih čakalnih časov v preostalih komponentah.

Iz slike 5.4 lahko razberemo še čakalne čase prejetih in poslanih paketov modula `obmocje2`, kjer so čakalni časi prvih tipov paketov ničelni, ker je modul *internet* najprej na povezavo poslal tovrstne pakete. Prikazani čakalni časi poslanih paketov modula `obmocje2` nastanejo zaradi krajših hitrosti prenosa paketov generiranih v modulu `obmocje1`. Vsi paketi namreč do ciljnega modula potujejo po isti povezavi, zato morajo tisti, ki so do oddajne enote usmerjevalnika prispeli kasneje, najprej počakati na sprostitvev kanala.



Slika 5.4 Čakalni časi paketov modulov `obmocje1` in `obmocje2`.

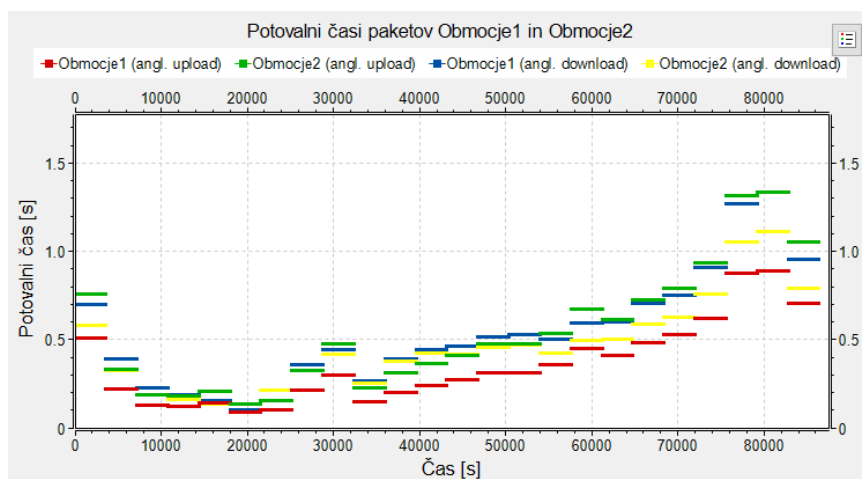
5.7 Metrika časov potovanj paketov

Metrika predstavlja čas potovanja paketa od pošiljatelja do prejemnika. Paketi generirani s strani modulov tipa `Obmocje` potujejo do modula `Internet`, medtem, ko paketi poslani s strani modula `Internet` potujejo do modulov tipa `Obmocje`. Čas potovanja posameznega paketa je določen kot vsota njegovih prenosnih časov po posameznih povezavah, časa obdelave in morebitnih čakalnih časov. Posamezne pakete tako kot v predhodni metriki združimo v skupino glede na modul kateremu pripadajo. Tabela 5.5 prikazuje potovalne čase poslanih in prejetih paketov po območjih. Metriko smo tekom simulacije izračunavali v končnem modulu, kjer smo potovalni čas izračunali kot razliko trenutnega časa simulacije in čas kreiranja paketa. Tudi tukaj se zaradi poenostavitve razlage osredotočimo na zgornji del omrežja z moduli `obmocje1`, `obmocje2`, `internet` in `usmerjevalnik1`. Kot že rečeno modul `internet` v vsakem časovnem intervalu generiranja paketov najprej naslovi paket območju z največ uporabniki tj. `obmocje2`. Zaradi omenjenega vrstnega reda paketi naslovljeni na `obmocje2` ne čakajo v čakalnih vrstah, zato je njihov potovalni čas odvisen samo od časa potovanja po povezavah in časa obdelave paketa na usmerjevalniku ter zato manjši od časa potovanja paketov naslovljenih modulu `obmocje1`. Slednji paketi morajo najprej počakati na prenos predhodnih paketov, zato njihov potovalni čas poleg časov prenosa po povezavah zajema še čakalne čase. Po drugi strani generirani paketi s strani modula `obmocje1` do prvega usmerjevalnika na poti prispejo hitreje, zato čakanje v čakalnih vrstah v povprečju zanje ni potrebno. Potovalni časi takšnih paketov so posledično odvisni samo od hitrosti prenosa paketov po pripadajočih povezavah. Generirani paketi s strani modula `obmocje2` do prvega usmerjevalnika v primerjavi s paketi modula `obmocje1` prispejo kasneje, zato morajo najprej počakati na sprostitev kanala.

Hitrosti prenosa paketov različnih modulov po povezavi `usmerjevalnik1-internet` niso več tako podobne, saj so paketi modula `obmocje2` zaradi večjega števila uporabnikov v primerjavi z območjem `obmocje1` z manjšim številom uporabnikov večji. Posledično paketi poslani ali generirani s strani modula `obmocje2` za prenos po isti povezavi potrebujejo več časa. Na sliki 5.5 predstavimo potovalne čase paketov modulov `obmocje1` in `obmocje2`. Skladno z naraščanjem čakalnih časov paketov, naraščajo tudi potovalni časi, ki vrhunec dosežejo v času generiranja največje količine prometa. Opazimo, da nekateri izmed paketov, ki so generirani v najbolj obremenjeni uri do cilja potrebujejo več kot sekundo (1000 ms).

Oznaka območja	Potovalni čas [s]			
	Download		Upload	
	Povprečni	Maksimalni	Povprečni	Maksimalni
obmocje1	0,54	1,33	0,36	0,89
obmocje2	0,47	1,11	0,54	1,33
obmocje3	0,52	1,33	0,53	1,33
obmocje4	0,86	0,40	0,34	0,86
obmocje5	0,40	1,03	0,41	1,04

Tabela 5.5 Tabela povprečnih in maksimalnih časov potovanj paketov po območjih.

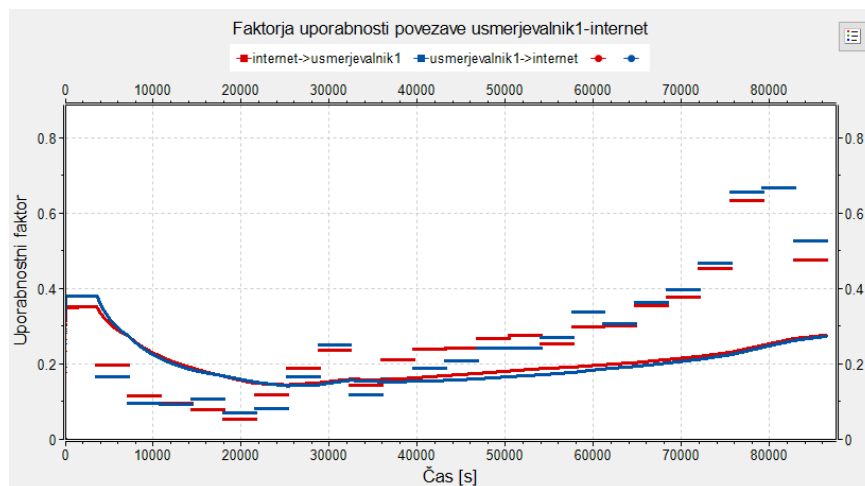


Slika 5.5 Potovalni časi paketov obmocje1 in obmocje2.

5.8 Metrika uporabnostnega faktorja povezav

Zaradi determinističnega generiranja paketov, njihovih velikosti in ocenjenih hitrosti posameznih povezav se pri predstavitvi metrike uporabnostnega faktorja osredotočimo le na povezavo usmerjevalnik1-internet. Uporabnostni faktorji posameznih povezav so si zaradi načina implementacije determinističnega modela precej podobni. Uporabnostni faktor povezave predstavlja razmerje med obremenitvijo oddajnega modula OddajaPrometa in maksimalnega prometa, s katero lahko modul pakete oddaja na pripadajočo povezavo. Omenjen maksimalni promet je določen s hitrostjo povezave. Stopničasti funkciji na grafu 5.6 predstavljata uporabnostni faktor povezave v določenem časovnem trenutku

izvajanja simulacije. Stopničasta funkcija označena z rdečo barvo predstavlja povezavo internet-usmerjevalnik1, modra pa povezavo usmerjevalnik1-internet. Iz omenjenih funkcij razberemo, da sta povezavi najbolj obremenjeni v trenutku, ko se simulacija nahaja v uri, v kateri generiramo največ prometa, medtem, ko je uporabnostni faktor najnižji v času, ko omrežje proizvaja najmanj prometa. Na sliki 5.6 prikažemo še obnašanje povprečnih vrednosti stopničastih funkcij uporabnostnega faktorja povezav skozi čas izvajanja simulacije. Zaradi determinističnih lastnosti vzpostavljenega modela omrežja, povprečne vrednosti uporabnostnih faktorjev povezav prve simulirane ure ustrezajo vrednostim stopničastih funkcij. Vse meritve znotraj urinega intervala so namreč enake, zato se tudi povprečne vrednosti v omenjenem intervalu ne spreminjajo. Desna stran grafov povprečij od diskretnih vrednosti odstopa zaradi razpršenosti izmerjenih uporabnostnih faktorjev in velikega števila meritev.



Slika 5.6 Uporabnostni faktor povezave usmerjevalnik1-internet.

5.9 Optimizacija

Pri analizi rezultatov prvega simulacijskega eksperimenta smo ugotovili, da so čakalni in potovalni časi paketov zelo veliki in v primerjavi s potovalnimi časi paketov v dejanskih internetnih omrežjih nesprejemljivi. Takšni rezultati so posledica izbranega načina generiranja paketov, zato bomo v novem simulacijskem eksperimentu način generiranja paketov spremenili. Cilj novega simulacijskega eksperimenta je torej zmanjšati čakalne in potovalne čase paketov. Na vseh generatorjih prometa povečamo frekvenco pošiljanja

paketov, kar storimo z zmanjšanjem medprihodnih časov. Vrednost parametra `medPrihodniCasPrometa` na vseh generatorjih nastavimo na 0,01 s, kar pomeni, da bo vsako vozlišče v sekundi generiralo 100 paketov. Zaradi povečanja frekvence pošiljanja paketov moramo ustrezno zmanjšati tudi velikosti paketov, zato uvedemo novo spremenljivko poimenovano `stPaketov`. Parameter uporabimo pri določanju velikosti posameznega paketa tako, da ocenjeno količino prometa zmanjšamo za faktor 100. V primerjavi s prvim simulacijskim eksperimentom torej na sekundo generiramo 100-krat več paketov, ki so 100-krat manjši. Pri tem ostajamo na deterministični porazdelitvi medprihodnih časov paketov. Inicializacijsko datoteko novega eksperimenta predstavimo s kodo 5.2.

Koda 5.2 Inicializacijska datoteka modela omrežja

```
[Config IzboljsanjeOmrezja]
**.usmerjevalnik**.streznik.strezniCasPaketa = 5us
**.generatorPrometa.medPrihodniCasPrometa = 0.01s
*.**.stPaketov = 100
```

Sprememba načina generiranja paketov je povzročila povečanje števila dogodkov v sistemu tj. paketov, ki jih prenašamo po omrežju. Posledica povečanega števila dogodkov je daljši čas izvajanja simulacije, zato smo njeno izvajanje morali prilagoditi. OMNeT++ omogoča več različnih načinov izvedbe simulacije. Eden izmed njih je zagon simulacije preko grafičnega uporabniškega vmesnika imenovanega Tkenv, s katerim lahko spremljamo dinamiko sistema. V primeru novega eksperimenta je bila takšna izvedba prepočasna, zato smo simulacijo zagnali preko alternativnega konzolnega programa imenovanega Cmdenv. Simulacijo smo izvajali na računalniku s specifikacijami navedenimi v tabeli 5.6. Izvedba 24 urnega simulacijskega časa, na računalniku z navedenimi zmogljivostmi je trajala približno 40 minut. Omenimo, da je izhodna datoteka simulacije v katero beležimo vektorske rezultate kot so potovalni in čakalni časi paketov velika 9 GB.

Procesor	Delovni pomnilnik	Tip sistema
Intel(R) Core(TM) i7-5500U CPU 2.40GHz 2.40GHz	8,00 GB	64-bit

Tabela 5.6 Specifikacije računalnika na katerem smo zagnali simulacijo.

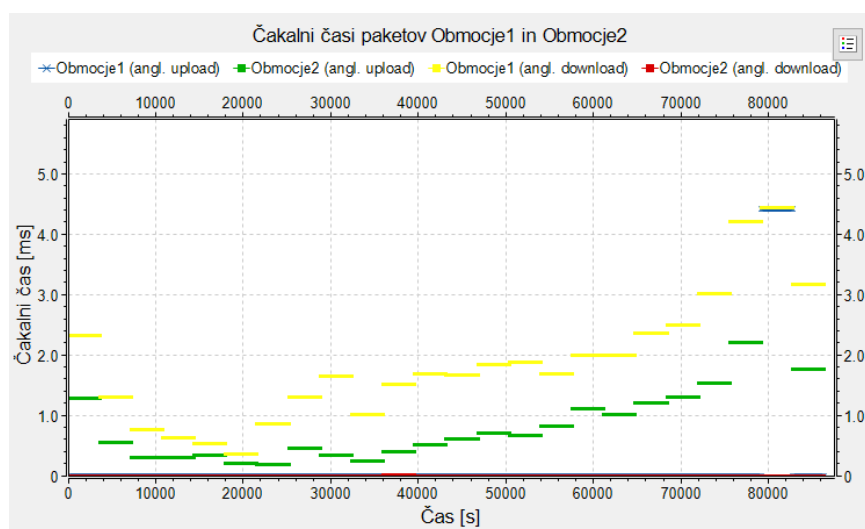
Na tem mestu najprej predstavimo čakalne čase novega simulacijskega eksperimenta. Iz tabele 5.7 razberemo, da so čakalni časi paketov v primerjavi s prvim simulacijskim eksperimentom sedaj prikazani v milisekundah. Ponovno opazimo, da paketi poslani

modulu *obmocje2* v čakalnih vrstah ne čakajo, saj je modul *Internet* na povezavo najprej poslal tovrstne pakete. Čakalni časi paketov, ki so naslovljeni modulu *obmocje1* so zaradi zmanjšanja velikosti paketov ustrezno manjši. V povprečju omenjeni paketi v čakalnih vrstah čakajo 1,86 ms, v času največje obremenitve omrežja pa čakajo 4,43 ms. Iz tabele lahko razberemo, da paketi poslani s strani modula *obmocje1* v primerjavi z modulom *obmocje2* v čakalnih vrstah čakajo manj časa. Do spremembe pride v času izvajanja simulacije z največjo količino prometa, pri čemer so paketi generirani s strani modula *obmocje1* v čakalnih vrstah zadržani dlje.

Oznaka območja	Čakalni čas [ms]			
	Download		Upload	
	Povprečni	Maksimalni	Povprečni	Maksimalni
<i>obmocje1</i>	1,86	4,43	0,18	4,43
<i>obmocje2</i>	0,00	0,00	0,75	2,20
<i>obmocje3</i>	1,43	3,69	1,30	3,68
<i>obmocje4</i>	0,00	0,00	0,00	0,00
<i>obmocje5</i>	0,74	1,90	1,02	2,89

Tabela 5.7 Tabela povprečnih in maksimalnih čakalnih časov paketov po območjih izboljšane simulacijskega eksperimenta.

Na sliki 5.7 lahko preverimo še vrednosti čakalnih časov paketov modulov *obmocje1* in *obmocje2* v odvisnosti od časa izvajanja simulacije.



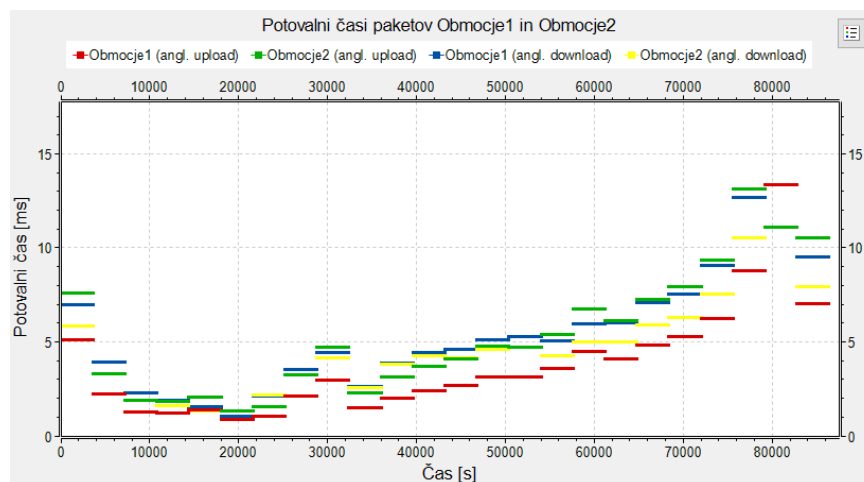
Slika 5.7 Potovalni časi paketov *obmocje1* in *obmocje2* izboljšane simulacijskega eksperimenta.

Tabela 5.8 prikazuje povprečne in maksimalne čase potovanj paketov po območjih v milisekundah. Razberemo lahko, da paketi poslani do posameznih območij v povprečju do prejemnika prispejo v manj kot 6 ms. Prav tako podobno velja za povprečne potovalne čase paketov, ki jih moduli tipa Obmocje naslovijo modulu Internet. V najbolj obremenjeni uri izvajanja simulacije potovalni časi od pošiljatelja do prejemnika potrebujejo največ 13,34 ms.

Oznaka območja	Potovalni čas [ms]			
	Download		Upload	
	Povprečni	Maksimalni	Povprečni	Maksimalni
obmocje1	5,42	13,34	3,78	13,34
obmocje2	4,67	11,10	5,33	13,13
obmocje3	5,20	13,34	5,16	13,34
obmocje4	3,32	8,58	3,42	8,58
obmocje5	4,01	10,36	4,38	10,36

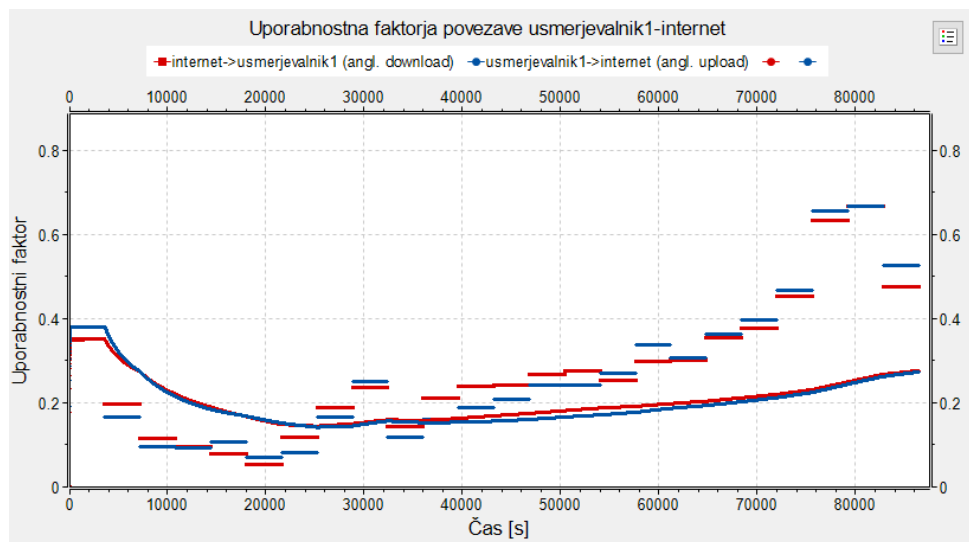
Tabela 5.8 Tabela povprečnih in maksimalnih časov potovanj paketov po območjih izboljšane simulacijskega eksperimenta.

Na sliki 5.8 prikažemo še potovalne čase paketov modulov obmocje1 in obmocje2 med izvajanjem simulacije. Potovalni časi posameznih paketov naraščajo skladno z naraščanjem čakalnih časov paketov. Maksimalni potovalni časi nastanejo v času generiranja največje količine prometa tj. med 21:00 in 23:00 uro.



Slika 5.8 Potovalni časi paketov obmocje1 in obmocje2 izboljšane simulacijskega eksperimenta.

Na koncu na sliki 5.9 predstavimo še uporabnostni faktor povezave usmerjevalnik1-internet. Rezultat je identičen rezultatu uporabnostnega faktorja obravnavane povezave prvega simulacijskega eksperimenta. V novem simulacijskem eksperimentu smo zmanjšali velikosti paketov, vendar smo skladno s tem povečali tudi frekvenco pošiljanja paketov. Količine generiranega prometa na sekundo posameznega območja s takšnim načinom generiranja nismo spremenili, zato je uporabnostni faktor povezave usmerjevalnik1-internet identičen rezultatu prvega simulacijskega eksperimenta. V obeh primerih uporabnostni faktor povezave namreč izmerimo na vsako sekundo izvajanja simulacije. Vrednosti stopničastih funkcij sta na sliki 5.9 še povprečeni glede na čas izvajanja simulacije.



Slika 5.9 Uporabnostni faktor povezave usmerjevalnik1-internet izboljšane simulacijskega eksperimenta.

S povečanjem frekvence pošiljanja paketov in zmanjšanja njihove velikosti smo vplivali na zmanjšanje povprečnih in maksimalnih čakalnih časov paketov vseh modulov. Paketom, ki so v prvem simulacijskem eksperimentu čakali na sprostitev povezave oz. na konec prenosa predhodnega paketa, smo z zmanjšanjem velikosti paketov zmanjšali čakalni čas in posledično prišli do bolj realnih potovalnih časov paketov. Čeprav smo prišli do boljših potovalnih časov, bi za boljše aproksimacijo delovanja internetnih omrežij morali frekvenco pošiljanja paketov še povečati oz. ocenjeno količino prometa predstaviti s paketi realnih velikosti. Na osnovi predstavljenih rezultatov drugega simulacijskega eksperimenta zaključimo, da smo zastavljeni cilj bolj ali manj dosegli. Na koncu omenimo še, da bodo mediji imeli standardne performance iz diskretnega prostora prenašalnih hi-

trosti, kar pomeni, da so rezultati simulacije delovanja omrežja pesimistični. Prav tako poudarimo, da deterministično generiranje paketov ne odraža dogajanja v realnih internetnih omrežjih, saj bi generiranje prometa morali aproksimirati z uporabo ene izmed verjetnostnih porazdelitev npr. Poissonove porazdelitve.

6 Zaključek

V okviru diplomske naloge smo obravnavali problem postavitve modela fizičnega omrežja za potrebe namišljenega internetnega ponudnika, ki želi na območju Upravne enote Slovenj Gradec zgraditi novo internetno omrežje. K obravnavanemu problemu smo pristopili tako, da smo najprej analizirali naselja, natančneje njihove prebivalce in zaposlene. S pomočjo analize smo določili oceno števila poslovnih in bivanjskih uporabnikov. Ločitev na dva tipa uporabnikov smo uporabili zaradi predpostavke, da so za posamezen tip uporabnika značilne drugačne potrebe. V nadaljevanju smo beležili promet študentov Študentskega doma v Ljubljani, ter na ta način določili funkcijo količine prometa bivanjskega uporabnika. Funkcijo količine prometa poslovnega uporabnika smo določili z meritvami v organizaciji, ki svojo dejavnost opravlja v osem urnem delavniku. S pomočjo orodja OMNeT++ smo implementirali zeleno delovanje komponent našega modela omrežja, ter v nadaljevanju iz zbranih podatkov postavili model. Sledila je izvedba prve simulacije in podrobnejši opis metrik čakalnih in potovalnih časov ter uporabnostnih faktorjev določenih povezav. Na koncu smo zaradi nereálnih časov potovanja paketov model omrežja izboljšali, tako, da smo spremenili način generiranja paketov. S spre-

membo smo vplivali na krajše čakalne čase in posledičnega krajšega potovanja paketov od pošiljatelja do prejemnika. Rezultat našega dela je bolj ali manj prilagojen model omrežja, ki temelji na grobih ocenah in poenostavljeni implementaciji, ki ne odraža dejanskega delovanja realnih omrežij. Glavna težava naše zasnove modela omrežja je način generiranja količine prometa in deterministično izvajanje posameznih dogodkov kot so strežni časi, medprijodni časi paketov itd. Prav tako pri naši implementaciji zaradi poenostavitve nismo upoštevali glavnih omrežnih protokolov, ki so še kako prisotni v dejanskih omrežjih. Kot primer navedimo usmerjevalni protokol OSPF (angl. *Open Shortest Path First*), ki v osnovi skrbi za dinamično vzpostavitev usmerjevalnih tabel. Internetni ponudnik bi predstavljeno rešitev sicer lahko uporabil za osnovo postavitve novega modela omrežja, vendar bi za verodostojne zaključke moral trenutno implementacijo dopolniti z delovanjem, ki bolj povzema dogajanje v realnih internetnih omrežjih. Prav tako je pri našem modelu omrežja vprašljiva verodostojnost vzpostavljenih ocen, za katere ne moremo potrditi, da odražajo dejansko stanje, saj smo bili zaradi pomanjkanja zahtevanih podatkov primorani določiti nepreverjene predpostavke.

LITERATURA

- [1] M. Mraz, M. Moškon, Modeliranje računalniških omrežij, Založba FE in FRI, Ljubljana, 2012, pp. 11–17.
- [2] A. Maria, Introduction to modeling and simulation, in: Proceedings of the 29th conference on Winter simulation, IEEE Computer Society, 1997, pp. 7–13.
- [3] G. V. Caliri, Introduction to analytical modeling, in: Int. CMG Conference, Citeseer, 2000, pp. 31–36.
- [4] Geopedia - interaktivni spletni atlas in zemljevid Slovenije, <http://www.geopedia.si>, [Online; Dostopano 1.10.2017].
- [5] List of computer simulation software, https://en.wikipedia.org/wiki/List_of_computer_simulation_software, [Online; Dostopano 1.10.2017].
- [6] OMNeT++ Discrete Event Simulator - Home, <https://www.omnetpp.org/>, [Online; Dostopano 2.1.2017].
- [7] OMNeT++ Simulation Manual, OMNeT++ version 5.2, https://www.omnetpp.org/doc/omnetpp/manual/#toc_1, [Online; Dostopano 1.12.2017].
- [8] A. Varga, R. Hornig, An overview of the OMNeT++ simulation environment, in: SimuTools, 2008.
- [9] A Q quick overview of the OMNeT++ IDE, <https://omnetpp.org/doc/omnetpp/IDE-Overview.pdf>, [Online; Dostopano 10.12.2017].
- [10] Modular programming, <https://www.techopedia.com/definition/25972/modular-programming>, [Online; Dostopano 1.11.2017].

- [11] Omrežna topologija – Računalniška omrežja, http://www.s-sers.mb.edus.si/gradiva/w3/omrezja/08_topologije/topologija.html, [Online; Dostopano 12.1.2017].
- [12] Campus Network for High Availability Design Guide - Cisco, https://www.cisco.com/c/en/us/td/docs/solutions/Enterprise/Campus/HA_campus_DG/hacampusdg.html#wp1107706, [Online; Dostopano 12.1.2017].