

Univerza v Ljubljani
Fakulteta za elektrotehniko

Andrej Vilhar

**Optimizacija postopkov za
hierarhično upravljanje z
mobilnostjo v internetu**

DOKTORSKA DISERTACIJA

Mentor: prof. dr. Janez Bešter
Somentor: prof. dr. Gorazd Kandus

Ljubljana, 2009

Moji družini. Juliji, Janu Aleksandru in Andreji.

Izjava

Doktorska disertacija z naslovom “Optimizacija postopkov za hierarhično upravljanje z mobilnostjo v internetu,” je v celoti, z izjemo označenih navedb literature, plod mojega lastnega raziskovalnega dela. Opravljena je bila pod mentorstvom prof. dr. Janeza Beštra s Fakultete za elektrotehniko Univerze v Ljubljani ter somentorstvom prof. dr. Gorazda Kandusa na Odseku za komunikacijske sisteme Instituta “Jožef Stefan” v Ljubljani.

Zahvala

Za uspešno vodenje skozi vsa leta mojega podiplomskega izobraževanja se zahvaljujem mentorju prof. dr. Janezu Beštru in somentorju prof. dr. Gorazdu Kandusu.

Posebna zahvala gre dr. Romanu Novaku, za uspešno uvajanje v raziskovalno delo ter številne strokovne nasvete. Za uspešno sodelovanje in dragocene izkušnje se zahvaljujem tudi ostalim sodelavcem Odseka za komunikacijske sisteme na Institutu "Jožef Stefan."

Raziskovalno delo sta v okviru projekta mladih raziskovalcev finančno omogočila Ministrstvo za visoko šolstvo, znanost in tehnologijo ter Agencija za raziskovalno dejavnost Republike Slovenije.

Povzetek

V doktorski disertaciji obravnavamo hierarhični pristop k upravljanju mobilnosti v internetu. Po tem pristopu se mobilnost upravlja ločeno na globalnem in lokalnem nivoju, kar znižuje režijske stroške in pospeši izročanje. Osredotočimo se na delovanje na lokalnem nivoju, kjer mobilni terminali entitete za upravljanje mobilnosti izbirajo dinamično. Po protokolu HMIPv6 se entitete imenujejo sidrne točke mobilnosti, njihova izbira pa vpliva na učinkovitost protokola. V doktorski disertaciji predlagamo nove metode za izbiranje sidrnih točk, nove modele testnih topologij in nove metode za postopek evaluacije. Pridobljeni rezultati kažejo na prednosti naših predlogov in na omejitve do sedaj uporabljenih testnih okolij.

V uvodu naše raziskovalno delo umestimo v širše področje. Predstavimo pojma globalne oziroma makro-mobilnosti in lokalne oziroma mikro-mobilnosti ter podamo pregled vidnejših predstavnikov protokolov. Izpostavimo, da so principi upravljanja mobilnosti ne glede na izbrani protokol zelo podobni. Naštejemo tudi druge možnosti za izboljšanje učinkovitosti upravljanja mobilnosti, ki so lahko bodisi alternativna, bodisi komplementarna rešitev mikro-mobilnemu pristopu. Nazadnje opišemo pomanjkljivosti dosedanjih raziskav na področju algoritmov za izbiranje sidrnih točk. Le-te predstavljajo izhodišče raziskovalnega dela v okviru doktorske disertacije.

V drugem poglavju opišemo delovanje makro-mobilnega protokola Mobile IPv6 in mikro-mobilnega protokola HMIPv6, po katerih delujejo rešitve, ki jih v disertaciji analiziramo. Čeprav je poznavanje njunega delovanja nujno za razumevanje jedra disertacije, to nista edina protokola, za katera veljajo naše rešitve in dognanja. Zaradi podobnih principov delovanja se vsebina disertacije nanaša na vse protokole, ki za upravljanje mobilnosti uporabljajo eno entiteto na globalnem in eno na lokalnem nivoju. V poglavju je predstavljen tudi pojem mobilnih omrežij, to je omrežij, ki se premikajo kot celota. Gre za mobilne entitete, na katerih bo zaradi predvi-

dljivosti gibanja uporaba naših algoritmov za izbiro sidrnih točk še posebej privlačna.

Jedro disertacije z izvirnimi prispevki se prične v tretjem poglavju. Najprej podamo pregled algoritmov za izbiranje sidrnih točk, ki jih delimo v 5 skupin. Nato predstavimo idejo izbiranja sidrnih točk s predvidevanjem gibanja. Predlagamo tri različice prediktivnega algoritma in jih primerjamo s tremi reprezentativnimi obstoječimi algoritmi. Vseh šest zapišemo s psevdokodami. Učinkovitost algoritmov primerjamo s simulacijami na sintetičnih in realističnih topologijah. Za izvedbo simulacij na realističnih, predhodno razvijemo metodo za označevanje vlog posameznih vozlišč in območij pokritij sidrnih točk. Rezultati potrjujejo boljšo učinkovitost predlaganih algoritmov in upravičenost ponujanja storitev sidrnih točk v več avtonomnih sistemih. Nakazujejo tudi na to, da je razlika v učinkovitosti primerjanih algoritmov odvisna od topologije omrežja.

Za podrobnejšo raziskavo vpliva topologije omrežja na učinkovitost algoritmov za izbiranje sidrnih točk je bilo potrebno bazo simulacijskih omrežij razširiti. V ta namen v četrtem poglavju razvijemo lastni model za določanje tipov poslovnih relacij v omrežjih avtonomnih sistemov. Model omogoča analizo upravljanja mobilnosti na širokem naboru topologij. V uvodu poglavja predstavimo pojem poslovnih relacij in njihov pomen. Nato opišemo zadnje dosežke na sorodnih področjih, področju prepoznavanja tipov relacij v resničnem internetu in področju generiranja omrežij avtonomnih sistemov. Sledi formalna definicija problema TRR, ki vsebuje pet prepoznanih karakterističnih lastnosti interneta. Verodostojen model mora upoštevati vseh pet. Razvijemo tudi analitično funkcijo, ki opisuje porazdelitev relacij enakovreden-z-enakovrednim. Algoritem za določanje tipov poslovnih relacij, ki ga predlagamo, sestoji iz štirih korakov. Njegovo kakovost preverimo glede na hierarhične lastnosti, porazdelitev razdalj med vozlišči in ujemanje posameznih tipov relacij. Rezultati kažejo na visoko verodostojnost algoritma. Nazadnje preverimo njegovo obnašanje pri uporabi na sintetičnih topologijah.

V petem poglavju raziskujemo vpliv topologije omrežja na učinkovitost algoritmov. V ta namen analiziramo najprej medsebojni vpliv evaluacijskih metrik in njihovih parametrov, ki se uporabljajo na tem področju. Iz analize sledi, da sta parametra povprečne oddaljenosti od sidrne točke d in povprečnega števila zamenjanih dostopovnih vozlišč na eno zamenjano sidrno točko h ključna. Uvedemo pojem dh -izboljšane izbire, ki pomeni, da sta bila d in h izboljšana hkrati. Z dh -izboljšano izbiro izboljšamo učinkovitost algoritma ne glede na uporabniške parametre. Nato analitično dokažemo, da

dh -izboljšana izbira ni dosegljiva v drevesnih topologijah, ki jih uporablja velika večina raziskovalcev s tega področja. S simulacijami ugotovitev potrdimo in pokažemo, da realistične topologije nimajo te omejitve. Nazadnje identificiramo, katere topološke lastnosti vodijo k boljšim pogojem za dh -izboljšano izbiro. Lastnosti izrazimo analitično v enačbi, katere vrednost imenujemo prekrivna metrika. Višja vrednost prekrivne metrike pomeni boljše možnosti za dh -izboljšano izbiro. Ustreznost analitičnega izraza preverimo s postopnim modificiranjem topologij, pri čemer opazujemo razmerje med prekrivno metriko in parametroma d in h .

Doktorsko disertacijo sklenemo v šestem poglavju s pregledom vsebine, v katerem izpostavimo glavne zaključke in z oceno, katerim odprtim problemom se je vredno na tem področju posvetiti v prihodnosti.

Abstract

The topic of the dissertation is a hierarchical approach to mobility management in the internet. By such an approach, mobility is managed at the local level separately from the global level, which leads to lower overhead costs and faster handovers. The focus is given on the local level, where mobility management entities are selected dynamically by Mobile Nodes (MNs). In the HMIPv6 protocol, these entities are referred to as Mobility Anchor Points (MAPs). The manner of their selection influences the efficiency of the protocol. We propose new methods for MAP selection, new models of simulation topologies and new methods for the evaluation procedure. The results confirm the anticipated advantages of our proposals and show the limitations of the testing environments that have been used in the field.

An introduction to the research field is given in the first chapter. The concepts of global or macro-mobility and local or micro-mobility are presented and a survey of notable protocols from both levels is given. Regardless of the selected protocol, the basic mobility management principles remain. Other possibilities for improving the efficiency of mobility management are also described. They can be used either as an alternative or a complementary solution to the micro-mobility approach. In the last part, the deficiencies of the state-of-the-art MAP selection algorithms are described, and these constitute the starting-point of the research work reported in the dissertation.

The operation of macro-mobility protocol Mobile IPv6 and micro-mobility protocol HMIPv6 is described in Chapter 2. The analysed solutions are based on these two protocols. Although understanding their operation is essential for the concepts in the core of the dissertation, our solutions and findings are not limited to Mobile IPv6 and HMIPv6. Due to the similarity of operation principles, the dissertation content is valid for any protocols that use one entity for managing mobility at the global and one at the local

level. The concept of network mobility is also presented in this chapter. It describes networks that move as a whole and are attractive for the application of our MAP selection algorithms, due to the predictability of their movement.

Chapter 3 and the following chapters constitute the core of the dissertation. A survey of existing MAP selection algorithms, which we divide into 5 groups, is given. The novel idea of selecting MAPs by predicting the future movement of the MNs is then presented. We propose three versions of predictive algorithm and compare them to three representative existing algorithms. All six are given formally by pseudocodes. The efficiency of the algorithms is compared by simulations using synthetic and realistic topologies. The ability to run simulations using realistic topologies was achieved by developing a specialized annotation method. The method differentiates the network nodes by the roles they play in the mobility management and determines the MAP coverage areas. The results confirm the greater efficiency of the proposed algorithms over the existing ones, and justify the spread of MAP service availability over several Autonomous Systems (ASs).

In order to investigate the influence of the network topology on the efficiency of MAP selection algorithms, the set of simulation networks has to be expanded. For this purpose, a model for assigning business relationship types in AS networks is developed in Chapter 4. The model enables analysis of the mobility management in a broad scope of topologies. In the introduction to the chapter, the notion of business relationships and their importance are explained. The description of the state-of-the-art follows. Inferring the type of relationships in the real internet and generating the AS networks are the most related research problems. Next, we formally define the Type-of-Relationship-Problem (TRR), which comprises five recognized characteristics of the real internet. We also develop an analytical function that describes the distribution of peer-to-peer relationships. The final algorithm for relationship assignment is given in four steps. Its quality is verified using hierarchical decompositions, distance distribution and by matching individual relationship types. The results imply that the proposed algorithm satisfies all five requirements of the TRR problem. Finally, its behaviour when applied to synthetic topologies is examined.

In Chapter 5 the influence of network topology on the efficiency of MAP selection algorithms is investigated. For this purpose, the mutual dependence of frequently used evaluation metrics and their parameters is analysed. The analysis shows that the average distance from MAPs, d , and the average number of changed access points per MAP change, h , are of key

importance. We introduce the notion of dh -improved selection, by which d and h are improved simultaneously. By dh -improved selection, the efficiency of the algorithm is improved, regardless of the given user parameters. A formal proof is given, stating that dh -improved selection is not achievable in the tree topologies used by the majority of researchers in the field. By simulations, we confirm this finding and demonstrate that realistic topologies are not subject to such a limitation. Finally, the topological characteristics that lead to better conditions for dh -improved selection are identified. The grade of their occurrence is expressed analytically as the overlap metric. The possibilities of achieving dh -improved selection are higher if the overlap metric is higher. The adequacy of the equation is verified by gradual modification of topologies, observing the relation between the overlap metric and parameters d and h .

In Chapter 6, we conclude by describing the main findings. Some promising directions for future work are given.

Izvirni prispevki

Doktorska disertacija vsebuje naslednje prispevke k znanosti:

- **Razvoj treh tipov prediktivnega algoritma za izbiranje sidr-
nih točk.** V primerjavi z obstoječimi, naš algoritem izbira sidrne
točke, ki so v povprečju bližje, zamenja pa jih manj pogosto. Posle-
dično se poveča učinkovitost upravljanja mobilnosti. (Poglavje 3.2).
- **Razvoj metode za označevanje poljubnih topologij, ki omo-
goča analizo upravljanja mobilnosti.** Na tem področju so do se-
daj za analizo uporabljali drevesne topologije, ki jih zaradi enostavne
hierarhije v ta namen ni bilo potrebno posebej označevati. Naša me-
toda omogoča študijo realnih in drugih omrežij s strukturo, ki ni dre-
vesna. (Poglavji 3.4 in 4.6).
- **Ocena prednosti, ki jih zagotavlja ponujanje storitve sidrne
točke izven matičnega avtonomnega sistema.** Večina razisko-
valcev s področja te možnosti ne predvideva, čeprav v specifikaciji
HMIPv6 protokola ni prepovedana. Po gibalnih vzorcih, ki smo jih
uporabili, lahko mobilno vozlišče na ta način izvede menjave sidrnih
točk do 5 krat manj pogosto. (Poglavji 3.4 in 5.4).
- **Identifikacija petih karakterističnih lastnosti realnega inter-
neta s poznanimi tipi poslovnih relacij.** Karakteristične lastno-
sti smo formalno zapisali v obliki definicije problema TRR. Metoda,
ki ustrezno reši ta problem, verodostojno modelira realne strukturne
vzorke. (Poglavje 4.4).
- **Razvoj modela za določanje tipov poslovnih relacij.** Model de-
luje po relativno enostavni metodi, ki že vsebuje informacijo o struk-
turi poslovnih relacij in zato na vhodu ne potrebuje drugih podatkov

kot vhodno topologijo brez označenih tipov povezav. Uporaben je na vseh področjih, ki vključujejo verodostojno modeliranje realnih internetnih omrežij. (Poglavji 4.5 in 4.6).

- **Določitev medsebojne odvisnosti evaluacijskih metrik in njihovih parametrov, ki jih uporabljajo za vrednotenje algoritmov za izbiro sidrnih točk.** Iz ugotovljenih razmerij izhaja, da se pri hkratnem izboljšanju povprečne oddaljenosti mobilnega vozlišča od sidrne točke d in povprečnem številu zamenjanih dostopovnih točk na eno zamenjano sidrno točko h , učinkovitost protokola izboljša neodvisno od uporabniških parametrov. (Poglavji 5.1 in 5.2).
- **Formalni dokaz, da v drevesnih topologijah dh -izboljšana izbira ni mogoča.** Formalno smo pokazali, da je dh -izboljšana izbira mogoča v vseh ostalih omrežjih, ki nimajo drevesne strukture, s simulacijami pa smo potrdili še, da je v realističnih omrežjih dh -izboljšana izbira mogoča. (Poglavji 5.3 in 5.4).
- **Identifikacija topoloških vzorcev, ki ugodno vplivajo na stopnjo dh -izboljšanja.** Prisotnost vzorcev v dani topologiji smo izrazili analitično v obliki prekrivne metrike. Višja vrednost metrike pomeni boljše pogoje za doseganje dh -izboljšanja. (Poglavje 5.5).

Kazalo

Kazalo	i
Slike	v
Tabele	vii
1 Uvod	1
1.1 Makro-mobilnost	1
1.2 Mikro-mobilnost	4
1.3 Algoritmi za izbiranje sidrnih točk	5
2 Opis izbranih protokolov	9
2.1 Mobile IPv6 protokol	9
2.1.1 Zahteve	9
2.1.2 Osnove delovanja in izpolnitev zahtev	11
2.1.3 Optimizacija usmerjanja	13
2.1.4 Varnost	15
2.1.5 Primer poteka komunikacije	16
2.2 HMIPv6 protokol	17
2.2.1 Mobilnost omrežij	20
3 Prediktivni algoritmi za izbiro sider	25
3.1 Pregled obstoječih algoritmov	25
3.1.1 Algoritmi, zasnovani na oddaljenosti	26
3.1.2 Algoritmi, zasnovani na hitrosti	26
3.1.3 Algoritmi, zasnovani na zgodovini	27
3.1.4 Adaptivni algoritmi	27
3.1.5 Ostali algoritmi	28
3.2 Opis predlaganih algoritmov	28

3.2.1	Osnovna ideja	28
3.2.2	Delovanje analiziranih algoritmov	29
3.3	Analiza na sintetičnih topologijah	37
3.3.1	Simulacijski model	37
3.3.2	Evaluacijske metrike	38
3.3.3	Simulacijski rezultati za sintetične topologije	39
3.4	Analiza na realističnih topologijah	43
3.4.1	Označevanje in uporaba realističnih topologij	43
3.4.2	Simulacijski model in metrike	45
3.4.3	Simulacijski rezultati za realistične topologije	46
4	Modeliranje realnih omrežij	49
4.1	Pomen poznavanja poslovnih relacij	50
4.1.1	Tipi poslovnih relacij	50
4.1.2	Usmerjevalne poti brez dolin	50
4.1.3	Posledice neupoštevanja relacij	51
4.2	Prepoznavanje tipov poslovnih relacij	52
4.3	Generiranje omrežij avtonomnih sistemov	54
4.3.1	Osnovni topološki generatorji	54
4.3.2	Relacijski generatorji	55
4.4	Problem določanja relacij v naključnih grafih	56
4.4.1	Definicija problema	56
4.4.2	Utemeljitev zahtev	57
4.5	Porazdelitev relacij enakovreden–z–enakovrednim	59
4.5.1	Verjetnostna porazdelitev	60
4.5.2	Skalirni model	63
4.5.3	Končni zapis funkcije	66
4.6	Generator poslovnih relacij	67
4.7	Kakovost algoritma	68
4.7.1	Natančnost hierarhije	69
4.7.2	Porazdelitev razdalj	71
4.7.3	Primernost koraka 2 in ujemanje posameznih relacij	72
4.8	Generator relacij na naključnih topologijah	73
5	Vpliv strukture omrežij na izbiro sider	77
5.1	Metrike	77
5.2	<i>dh</i> -izboljšana izbira	81
5.3	Topološke zahteve za <i>dh</i> -izboljšano izbiro	82
5.3.1	Vzorci vstopanja in izstopanja iz območij pokritij	84

5.3.2	Poti mobilnih vozlišč in zaporedja sidrnih točk	85
5.3.3	dh -izboljšanje v drevesih in obrnjenih drevesih	86
5.3.4	dh -izboljšanje v topologijah z delnim prekrivanjem	91
5.4	Testiranje drevesnih in realističnih topologij	92
5.4.1	Opis simulacijskih topologij	93
5.4.2	Vzorci gibanja mobilnih vozlišč	93
5.4.3	Simulacijski rezultati	95
5.5	Topološke lastnosti za visoko dh -izboljšano izbiro	97
5.5.1	Prekrivna metrika	98
5.5.2	Prekrivna metrika v drevesih in realističnih topologijah	99
5.5.3	Prekrivna metrika v modificiranih topologijah	100
6	Sklep	105
6.1	Pregled vsebine z glavnimi zaključki	105
6.2	Smernice za nadaljnje delo	108
	Literatura	111

Slike

2.1	Dvosmerno tuneliranje (po [1])	12
2.2	Problem trikotniškega usmerjanja (po [1])	14
2.3	Prikaz mobilnosti končnih vozlišč s primerom	16
2.4	Primerjava protokola HMIPv6 s protokolom Mobile IPv6	18
2.5	Prikaz razlike v signalizaciji pri lokalnem in regionalnem premiku	19
2.6	Shematičen prikaz mobilnega omrežja	20
2.7	Usmerjevalna pot pri mobilnih omrežjih s hierarhičnim pristopom	22
2.8	Usmerjevalna pot po optimizaciji usmerjanja	22
3.1	Primer izreza 36 radijskih celic	37
3.2	Povprečno število menjav (1)	40
3.3	Povprečna razdalja (1)	41
3.4	Povprečno število menjav (2)	41
3.5	Povprečna razdalja (2)	42
3.6	Primer topologij z 10 vozlišči	44
3.7	Časovni delež izbranih sidrnih točk glede na hierarhični nivo . .	47
4.1	Primer omrežja avtonomnih sistemov	51
4.2	Parcialna verjetnostna funkcija za višje range (nižje stopnje) . .	61
4.3	Parcialna verjetnostna funkcija za nižje range (višje stopnje) . .	62
4.4	Iskanje funkcije za parameter a	63
4.5	Iskanje funkcij za parametra b in c	64
4.6	Izris parametra a na log-log skali	65
4.7	Izris funkcije $pp(x, y)$ za $R_{max} = 100$ in $\gamma = 1$	66
4.8	Porazdelitev razdalj za merjene in generirane označbe grafov . .	72
5.1	Relacije med evaluacijskimi metrikami	80
5.2	Drevo	83

5.3	Obrnjeno drevo	83
5.4	Premaknjeno drevo	83
5.5	Neskončno drevo z $b = 4$	94
5.6	Povprečne razdalje	96
5.7	Razlike v povprečnih razdaljah	96
5.8	Povprečni kvocienti izročanja	97
5.9	Razlike v povprečnih kvocientih izročanja	97
5.10	Parametri, ki vplivajo na prekrivno metriko O	98
5.11	Tipični primeri parov sidrnih točk	99
5.12	Δh , Δd in O za topologije primerljivih velikosti	100
5.13	Δh in Δd glede na število generacij - izvor drevo	102
5.14	Δh in Δd glede na število generacij - izvor generirani model	102
5.15	Povprečne vrednosti O , Δh in Δd - izvor drevo	103
5.16	Povprečne vrednosti O , Δh in Δd - izvor generirani model	103

Tabele

3.1	Definicije uporabljenih razredov	30
3.2	Povprečna porazdelitev sidrnih točk	38
3.3	Primerjava algoritmov (sidrne točke so na robu interneta)	46
3.4	Primerjava algoritmov (sidrne točke so na vseh nivojih)	47
4.1	Modelirani grafi	60
4.2	Parametri modela za grafe z vseh treh let	63
4.3	Porazdelitev vozlišč v 5-nivojski hierarhiji	70
4.4	Porazdelitev vozlišč v hierarhiji ponudnik–stranka	71
4.5	Ujemanje s korakom 2 določenih relacij z merjenimi relacijami	73
4.6	Ujemanje posameznih relacij	73
4.7	5-nivojska hierarhija označenih Inet topologij	74
4.8	Hierarhija ponudnik–stranka označenih Inet topologij	75
5.1	Cene kot funkcije različnih parametrov	78
5.2	Opis parametrov	79
5.3	Trojke premikov za tri obravnavane topologije	84
5.4	Menjave sidrnih točk na poti A	91
5.5	Menjave sidrnih točk na poti B	92
5.6	Povprečne razdalje d in povprečni kvocienta izročanja h	92

Poglavje 1

Uvod

V moderni informacijski družbi potreba po dostopu do interneta stalno narašča. Zahteve uporabnikov se ne kažejo samo v možnosti dostopa do interneta na več lokacijah in z večjo pasovno širino, pač pa tudi v nemoteni uporabi globalnega omrežja med gibanjem. Razvoj informacijskih in komunikacijskih tehnologij prinaša veliko možnosti za uresničitev teh potreb. Na fizičnem in povezavnem sloju protokolnega sklada je v zadnjih letih prišlo do vidnega napredka brezžičnih tehnologij. Pokritost terena s signalom se skokovito izboljšuje. Prej ali slej bo nastopil trenutek razvoja, ko bo stalen dostop do interneta v fizikalnem smislu krajevno in časovno nemejen. Za doseganje stalne dosegljivosti in neprekinjenosti vzpostavljene povezave med gibanjem, ki sta glavna cilja upravljanja mobilnosti [2, 3], pa je potrebna tudi izvedba postopkov na višjih slojih.

1.1 Makro-mobilnost

Protokoli za upravljanje mobilnosti morajo zagotavljati globalno oziroma makro-mobilnost. Na področju zagotavljanja makro-mobilnosti v internetu obstaja več standardnih rešitev, izvedenih na različnih slojih protokolnega sklada [4–6]. V raziskovalnih krogih sta kot pomembnejši med njimi prepoznani rešitvi z aplikacijskim protokolom SIP (Session Initiation Protocol) [5, 7] in z omrežnim protokolom Mobile IP [1, 8, 9]. Med ostalimi predlogi izstopata transportni protokol SCTP [10, 11] in protokol HIP [12], ki uvaja poseben sloj za upravljanje mobilnosti.

Protokol SIP je bil v osnovi namenjen signalizaciji pri upravljanju z mul-

timedijskimi sejami. Definira ga dokument RFC 3261 [7], sprejet v okviru združenja IETF. SIP upravlja z več parametri, pomembnimi za vzdrževanje multimedijskih sej. Med njimi je tudi uporabnikova trenutna lokacija, ki omogoča gostovanje uporabnika v tujih omrežjih. Za stalno dosegljivost skrbi lokacijski strežnik, ki vzdržuje povezavo med uporabnikovo identiteto in njegovo lokacijo. Komunikacija poteka neposredno med mobilnim in dopisnim terminalom. Z dopolnitvijo ažurnega osveževanja informacije o novi lokaciji dopisnim terminalom tudi med potekom seje SIP postane protokol, ki omogoča neprekinjenost vzpostavljene povezave [5, 13]. V okviru združenja 3GPP je bil SIP sprejet kot signalizacijski protokol v IMS arhitekturi [14].

Protokol Mobile IP je bil v osnovi načrtovan za zagotavljanje mobilnosti v internetu. Starejša različica (MIPv4), zasnovana za protokol IP verzije 4, je definirana z dokumentom RFC 3344 [8], novejša (MIPv6), ki temelji na verziji 6 pa z RFC 3775 [9]. Obe verziji temeljita na ideji uvedbe vsaj dveh IP naslovov za mobilni terminal. Domači naslov določa identiteto in lokacijo mobilnega terminala v domačem omrežju, začasni naslov pa njegovo trenutno lokacijo v globalnem omrežju. Za stalno dosegljivost skrbi domači agent, ki vzdržuje povezavo med obema naslovoma. V najosnovnejši različici protokola MIPv4 dopisni terminal pakete vedno pošilja na domači naslov. Domači agent jih prestreže in tunelira tujemu agentu v omrežje, kjer mobilni terminal trenutno gostuje. Tuji agent jih nato posreduje mobilnemu terminalu. V nasprotni smeri mobilni terminal pošilja pakete neposredno dopisnemu terminalu, brez posredovanja domačega agenta. Različica MIPv6 tujega agenta ne predvideva, tunel se vzpostavi neposredno med domačim agentom in mobilnim terminalom. V MIPv6 je vgrajena tudi funkcionalnost optimizacije usmerjanja, ki omogoča neposredno dvosmerno komunikacijo med mobilnim in dopisnim terminalom. Po protokolu Mobile IP mobilni terminal vsako spremembo začasnega naslova takoj sporoči domačemu agentu, v različici MIPv6 pa še dopisnim terminalom, s katerimi komunicira po optimiranih usmerjevalnih poteh. S tem je zagotovljena neprekinjenost vzpostavljene povezave.

Kljub podobnim principom zagotavljanja stalne dosegljivosti in neprekinjenosti vzpostavljene povezave, pokaže primerjava protokolov Mobile IP in SIP tudi pomembne razlike. Protokol SIP zagotavlja neprekinjenost vzpostavljene povezave samo aplikacijam, ki ne temeljijo na transportnem protokolu TCP [13, 15]. Razlog je v spreminjanju IP naslova, ki služi kot eden od identifikatorjev posamezne TCP seje. Mobile IP protokol ta pojav prepreči z metodo uporabe stalnega domačega naslova, ki mobilnost viš-

jim slojem prikrije [1]. Pri SIP protokolu identiteta uporabnika ni vezana na IP naslov, kar omogoča t.i. osebno mobilnost [5], to je možnost, da se uporabnik lahko registrira preko različnih terminalov. Po drugi strani je SIP protokol zaradi procesiranja na aplikacijskem sloju [16] in pridobivanju novega IP naslova preko protokola DHCP [6, 17] podvržen večjim zakasnitvam pri izročanju. Pri MIPv4 protokolu brez optimizacije usmerjanja zakasnitve pri izročanju naraščajo z oddaljenostjo od domačega omrežja, pri SIP in MIPv6 protokolu pa z oddaljenostjo od dopisnega terminala [15, 17]. Zaradi naštetih prednosti oziroma slabosti opisanih rešitev, so mnogi avtorji predlagali različne načine hibridne uporabe protokolov SIP, MIPv4 in MIPv6 [13, 15, 17, 18]. Ne glede na uporabljeno rešitev pa se izkaže, da je vpliv oddaljenosti mobilnega terminala od dopisnih terminalov in domačega agenta možno omejiti le v primeru uporabe dodatnih postopkov, kot je na primer mikro-mobilnost [6].

Primer protokola za zagotavljanje mobilnosti na transportnem sloju predstavlja mobilna verzija protokola SCTP (Stream Control Transmission Protocol) [19]. Osnovni SCTP protokol [10] je bil načrtovan kot alternativna možnost v internetu najpogosteje uporabljenemu protokolu TCP, zaradi možnosti prilagajanja zanesljivosti, pa predstavlja alternativo tudi protokolu UDP. Ena od posebnosti transportnega SCTP protokola je podpora večdomnosti, to je lastnosti, ki omogoča hkratno uporabo več omrežnih vmesnikov za isto sejo. V osnovni različici SCTP protokola igra večdomnost predvsem vlogo povečanja zanesljivosti. V mobilni različici, ki dopušča tudi možnost spreminjanja IP naslovov med vzpostavljeno sejo, pa je večdomnost uporabljena kot osnova za zagotavljanje neprekinjenosti povezave [11]. Eden glavnih problemov mobilnega SCTP protokola je zagotavljanje stalne dosegljivosti. Protokol namreč ne predvideva upravljanja s trenutno lokacijo mobilnega terminala, zaradi česar je potrebno to funkcionalnost zagotoviti na drugem nivoju, na primer s protokolom Mobile IP [4]. Tudi ostali predlogi protokolov za upravljanje mobilnosti na transportnem sloju bodisi ne predvidevajo upravljanja s trenutno lokacijo, bodisi slonijo na sistemu DNS [20], ki pa zaradi problema ažurnosti ne predstavlja primerne rešitve [21].

V literaturi zasledimo tudi protokol HIP (Host Identity Protocol) [12], ki vsebuje posebni sloj za upravljanje mobilnosti. Predvsem zaradi velike kompleksnosti potrebnih sprememb na komunikacijskih entitetah v obstoječem internetu, se ta rešitev v raziskovalnih krogih ni posebej uveljavila [4].

1.2 Mikro-mobilnost

Za doseganje večje učinkovitosti, je možno upravljanje mobilnosti na makro-nivoju dopolniti z upravljanjem na lokalnem oziroma mikro-nivoju. Glavna cilja protokolov za upravljanje mobilnosti na mikro-nivoju sta zmanjšanje režijske signalizacije in hitrejšo izročanje [3, 22]. To se dosega na način, da mobilnemu terminalu ni potrebno ob vsakem lokalnem premiku domačemu omrežju oziroma dopisnim terminalom sporočati nove lokacije, pač pa se informacija o natančni lokaciji hrani le lokalno.

Čeprav obstajajo predlogi za zagotavljanje mikro-mobilnosti s SIP protokolom [23], večina mikro-mobilnih protokolov temelji na razširitvi protokola Mobile IP. Vidnejši predstavniki so protokoli HAWAII [24], Cellular IP [25], Mobile IPv4 Regional Registration [26] in Hierarchical Mobile IPv6 (HMIPv6) [1, 27]. Kot poseben primer lahko obravnavamo protokol IDMP [28], ki je sicer omrežni mikro-mobilni protokol, vendar se za zagotavljanje makro-mobilnosti ne omejuje na uporabo Mobile IP protokola. Od vseh naštetih je protokol HMIPv6 edini, ki je bil v okviru združenja IETF sprejet kot dokument RFC [27].

Razlike v delovanju mikro-mobilnih protokolov se kažejo predvsem v načinu osveževanja natančne lokacije mobilnega terminala in načinu posredovanja paketov na lokalnem nivoju, ki lahko zajema različne sloje protokolnega sklada. Kljub tem razlikam avtorji v [22] ugotavljajo, da so osnovni principi mikro-mobilnih protokolov enaki in temeljijo na sistemu posebnih lokalnih entitet. Posamezna stopnja tovrstnih entitet vzdržuje povezavo med identiteto mobilnih terminalov in naslovom naslednje stopnje. Skupek vseh povezav na usmerjevalni poti do mobilnega terminala vsebuje informacijo o njegovi natančni lokaciji.

V primeru protokola HMIPv6 [27] se opisana lokalna entiteta imenuje sidrna točka mobilnosti (MAP - Mobility Anchor Point), specifikacija pa predvidena le eno stopnjo. Kot identifikator mobilnega terminala v sidrni točki služi regionalni IP naslov, ki je topološko pravilen na njeni povezavi in je dodeljen mobilnemu terminalu. Isti naslov uporablja domači agent mobilnega terminala kot podatek o njegovi lokaciji, medtem ko ga sidrna točka povezuje z informacijo o dejanskem lokalnem naslovu. Če se sidrna točka zamenja, se zamenja tudi regionalni naslov, kar je s signalizacijo potrebno sporočiti domačemu agentu. Na učinkovitost HMIPv6 protokola pomembno vpliva lokacija sidrne točke v lokalnem omrežju [29]. Še več, lokacija sidrnih točk oziroma ustreznih ekvivalentov je parameter, ki ima najpomembnejši vpliv na učinkovitost izročanja kateregakoli drugega mikro-mobilnega pro-

tokola [22].

Poleg opisanih principov zagotavljanja mikro-mobilnosti obstajajo še druge metode za izboljšanje izročanja. Hitro izročanje s protokolom FMIPv6 [30] temelji na predhodni zaznavi potrebe po izročitvi na podlagi ustreznih informacij s povezavnega sloja ter vzpostavitvi tunela med staro in novo dostopovno točko, ki omogoča nemoteno komunikacijo med postopkom osveževanja lokacije na globalnem nivoju. Bi-casting metoda podatkovni tok podvoji in ga pošilja proti stari in novi dostopovni točki mobilnega terminala hkrati [25, 31]. Uporaba principa večdomnosti pa mobilnemu terminalu zagotavlja hkratno omrežno povezavo na več omrežnih vmesnikih [32, 33]. Primerjava opisanih principov pokaže, da se pri učinkovitosti izročanja dosega boljše rezultate, če so v uporabi kombinirane rešitve, ki vključujejo tako mikro-mobilnost, kot tudi druge metode, med katerimi se v literaturi najpogosteje pojavlja hitro izročanje [1, 34–36]. S stališča enostavnosti uvedbe in režijske uporabe virov, pa izstopa uporaba mikro-mobilnih postopkov samih [35, 37].

Uporaba principov mikro-mobilnega HMIPv6 protokola je posebej privlačna tudi pri upravljanju mobilnosti omrežij. Mobilna omrežja so omrežja, ki se v internetu premikajo kot celota [38]. Standardna rešitev za upravljanje mobilnosti omrežij [39] ne predvideva optimizacije usmerjanja. Večina obstoječih predlogov za izvedbo optimizacije usmerjanja temelji na principih makro-mobilnosti [40–43]. Predlog rešitve v [44] pa z uporabo sidrne točke zaobide domače omrežje mobilnega omrežja. S tem je za komunikacijske tokove na globalnem nivoju dosežena optimalnost usmerjanja, hkrati pa so zagotovljene vse prednosti principa mikro-mobilnosti, ki so v primeru velikih mobilnih omrežij s stališča režijske signalizacije še bolj izrazite.

1.3 Algoritmi za izbiranje sidrnih točk

Po protokolu HMIPv6 vsaka sidrna točka oglašuje svojo prisotnost s sporočili, ki se širijo po omrežju do dostopovnih usmerjevalnikov, ti pa sporočilo posredujejo mobilnim terminalom. Mobilni terminal v splošnem prejme več takih sporočil in posledično lahko izbira med več sidrnimi točkami. Protokol HMIPv6 določa enostaven algoritem, po katerem mobilni terminali izberejo topološko najbolj oddaljeno sidrno točko [27]. Namen takega pristopa je zmanjšati pogostost osvežitve lokacije mobilnega terminala v oddaljenem domačem agentu in dopisnih terminalih. Sidrna točka, ki je bolj oddaljena, namreč običajno pokriva večje število dostopovnih usmerjevalnikov, s tem

pa je verjetnost potrebe po menjavi sidrne točke manjša. Avtorji protokola HMIPv6 dopuščajo možnost obstoja boljšega algoritma.

Slabosti izbire najbolj oddaljene sidrne točke prepoznavajo številni avtorji [29, 36, 45–65]. Najpogosteje ugotavljajo, da so najbolj oddaljene sidrne točke preobremenjene in da je pot signalizacijskih sporočil za mobilne terminale, ki se gibljejo v mejah pokritja bližje sidrne točke, nepotrebno dolga. Avtorji nato predlagajo različne algoritme za izbiranje primernejših sidrnih točk. Na podlagi analize ugotavljamo, da imajo predlagane metode določene pomanjkljivosti. Štiri pogloblitve pomanjkljivosti, ki jih prepoznavamo, so:

- **Neupoštevanje možnosti predvidevanja gibanja mobilnega terminala.** Predlagani algoritmi temeljijo na različnih idejah, ki upoštevajo hitrost mobilnega vozlišča, zgodovino njegovih premikov, njegovo komunikacijsko aktivnost, porazdelitev bremena v omrežju, povprečno signalizacijsko zakasnitev in podobno. To pomeni, da je pri odločanju o naslednji sidrni točki pomembno bodisi trenutno, bodisi preteklo stanje terminala oziroma omrežja. Nobeden od predlogov ne raziskuje možnosti izboljšav s predvidevanjem gibanja mobilne entitete, kar je še posebej privlačno v primeru javnih prevoznih sredstev, ki imajo utečeno pot.
- **Analiza predlaganih algoritmov le na drevesnih topologijah.** Kljub velikemu številu opravljenih raziskav, se velika večina opira na rezultate, dobljene z analizo preprostih drevesnih topologij. Izjemo predstavljajo članki [49, 50] in [55], v katerih poskušajo avtorji obravnavati predlagani algoritem neodvisno od topološke strukture omrežja oziroma območje pokritja posamezne sidrne točke obravnavajo opisno. Nobeno od naštetih del ne analizira učinkovitosti algoritmov na realnih modelih internetnih omrežij.
- **Uporaba raznolikih metrik za ocenjevanje učinkovitosti algoritmov.** Različni avtorji uporabljajo različne metrike, s čimer je otežkočena medsebojna primerjava rezultatov. Še več, nekateri avtorji uporabljajo t.i. skupne cene, kjer različne metrike združijo tako, da jih ustrezno utežijo, pri tem pa ne pojasnijo vpliva izbora uteži na rezultate. Nobena od raziskav ne analizira medsebojnih odvisnosti v širokem naboru metrik, ki se pojavljajo na tem področju.
- **Omejevanje delovanja sidrne točke na eno domeno.** Avtorji ponavadi predpostavljajo, da je delovanje posamezne sidrne točke ome-

jeno na eno samo domeno. V realnosti je pričakovati, da bodo terminali med gibanjem do interneta dostopali preko različnih dostopovnih tehnologij različnih internetnih ponudnikov. Če so preklopi pogosti in vsaka sidrna točka skrbi samo za mobilne terminale znotraj svoje domene, bo tudi menjava sidrnih točk pogosta, s čimer se učinkovitost protokola zmanjša.

Raziskave, opravljene v okviru doktorskega dela, temeljijo na iskanju rešitev za našete pomanjkljivosti.

Poglavje 2

Opis izbranih protokolov

V doktorski disertaciji opisane rešitve so zasnovane na mikro-mobilnem protokolu HMIPv6, ki predstavlja razširitev makro-mobilnega protokola Mobile IPv6. V tem poglavju natančneje opišemo njuno delovanje. Čeprav smo izbrali prav ta dva predstavnika, so v disertaciji opisani principi veljavni tudi za druge protokole, ki uporabljajo eno entiteto za upravljanje mobilnosti na globalnem in eno na lokalnem nivoju. V našem primeru sta to domači agent in sidrna točka mobilnosti.

2.1 Mobile IPv6 protokol

2.1.1 Zahteve

Cilj načrtovalcev protokola Mobile IPv6 je bil izpolniti naslednje zahteve [1]:

- **Neprekinjenost vzpostavljene komunikacije.** Menjava IP naslova ne sme vplivati na že vzpostavljene seje. TCP protokol na primer loči posamezne seje po t.i. vtičnicah (*angl. socket*), ki so definirane s petimi podatki, od katerih je eden tudi IP naslov mobilne naprave. Če se spremeni IP naslov, se spremeni tudi vtičnica in seja se poruši. Samo iskanje načina, kako obvestiti napravo na drugi strani zveze o novem IP naslovu, ne zadostuje.
- **Stalna dosegljivost.** Internet je prvotno deloval predvsem po principu odjemalec-strežnik. Danes pa je že močno prisotna uporaba principa vsak z vsakim (*angl. peer-to-peer*). Uporabniki želijo tudi sami

uporabljati funkcije strežnika, zato morajo biti dosegljivi vedno na istem naslovu. Rešitev, ki sloni na strežniku domenskih imen (DNS - Domain Name Server), ni ustrezna. Ažurnost namreč ni dovolj dobra, saj je v primeru menjave naslova potrebno osvežiti podatke v več DNS strežnikih hkrati. Ker to ni neskončno hiter proces, so v prehodnem času v različnih strežnikih različni vnosi. Odprto vprašanje je tudi obseg zbirke imen in naslovov [3].

- **Neodvisnost od zgornjih slojev.** Ker gre za premik v topologiji omrežja, mora biti le-ta tudi v celoti izveden na omrežnem sloju. Vpeljevanje upravljanja mobilnosti v transportni ali celo aplikacijski sloj pomeni, da mora biti za vsak protokol oziroma aplikacijo, ki jo terminal poganja, funkcionalnost mobilnosti podvojena. Tudi sam princip medsebojne neodvisnosti posameznih slojev je na ta način kršen. Za transportni in aplikacijski sloj namreč želimo, da sta usmerjanje in dostava paketov za njiju transparentna, četudi je prisotna mobilnost.
- **Neodvisnost od spodnjih slojev.** Za spodnje sloje velja podobno kot za zgornje. Nižjim slojem sicer že po funkcionalnosti mobilnost ne more biti skrita, saj je t.i. izročanje (*angl. handover*) na povezavnem sloju potrebno. Vseeno pa je potrebno zagotoviti, da izročanje na omrežnem sloju ni odvisno od spodaj ležeče tehnologije, zato da ne pride do podvajanja upravljanja mobilnosti za vsak primer posebej. V zadnjem času je močno prisoten trend, da se bo v globalnem omrežju IP protokol uveljavil kot stična točka nad različnimi tehnologijami in pod številnimi aplikacijami. Tudi takšnimi, ki šele prihajajo. Neodvisnost upravljanja mobilnosti od vseh ostalih slojev je zato zelo pomembna.
- **Ohranitev kontrole na strani terminalov.** Usmerjevalnike, ki skrbijo za dostavo paketov v internetu, je potrebno čim bolj razbremeniti, zato morajo v službi mobilnosti končnih naprav opravljati čim manj funkcionalnosti. V nasprotnem primeru nastopita vsaj dva problema. Prvič, ob morebitnih potrebnih spremembah je potrebno nadgraditi vse usmerjevalnike, kar je v globalnem omrežju prevelik zalogaj. In drugič, usmerjevalniki so obremenjeni z dodatnim procesiranjem paketov, kar vnaša v komunikacijo dodatne zakasnitve.

2.1.2 Osnove delovanja in izpolnitev zahtev

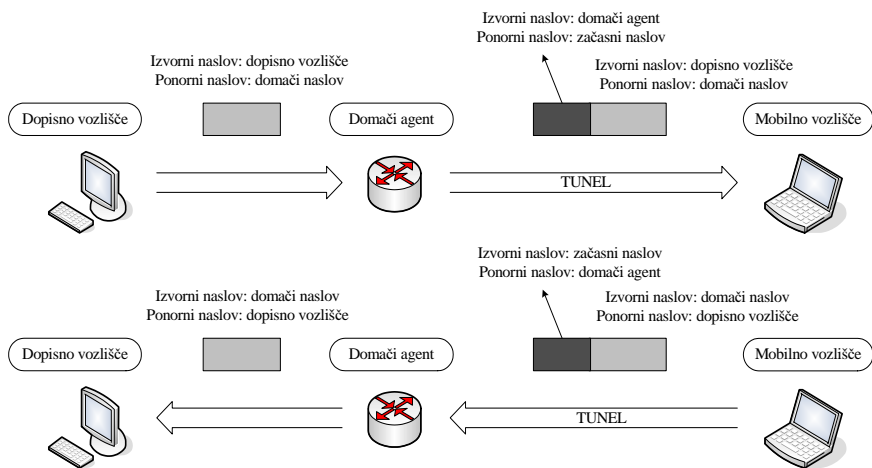
Protokol Mobile IPv6 je bil sprejet kot standardna rešitev z oznako RFC 3775 leta 2004 [9]. V tem dokumentu je natančno opisano njegovo delovanje z vsemi protokolarnimi postopki. V nadaljevanju se osredotočamo na osnovne principe, ki so potrebni za razumevanje, ter pri posameznih funkcionalnostih protokola opozorimo, kateri izmed naštetih zahtev iz poglavja 2.1.1 je bilo zadoščeno.

Osnovna ideja protokola Mobile IPv6 je, da za mobilna vozlišča (MN - Mobile Node) uvede poleg začasnega topološko odvisnega IP naslova (*angl. care-of address*) še stalen domači naslov (*angl. home address*). Domači naslov oblikuje mobilno vozlišče v izbranem domačem omrežju in ga ne spreminja. S tem sta njegova identiteta in topološka lokacija v domačem omrežju enolično določeni. Ko dopisna vozlišča (CN - Correspondent Node), t.j. vozlišča, s katerimi mobilno vozlišče komunicira, želijo vzpostaviti komunikacijo z mobilnim vozliščem, naslovijo pakete na njegov domači naslov. Če mobilnega vozlišča takrat ni v domačem omrežju, prestreže njemu namenjene pakete t.i. domači agent (HA - Home Agent) in jih s postopkom tuneliranja preusmeri na začasni naslov mobilnega vozlišča. S tem je dosežena **stalna dosegljivost** mobilnega vozlišča.

Informacijo o tem, kateri začasni naslov pripada kateremu domačemu naslovu, hrani domači agent v t.i. pomnilniku vezi (*angl. binding cache*). Mobilno vozlišče to informacijo pošilja domačemu agentu s sporočili osvežitev vezi (BU - Binding Update). Pravilna vez v vsakem trenutku komunikacije je ključnega pomena, saj bi izguba vezi pomenila prekinitev komunikacije ali celo nedosegljivost. Zato je pomembno, da je protokol osveževanja vezi zanesljiv. V ta namen domači agent na osvežitev vezi odgovori s potrditvijo vezi (BA - Binding Acknowledgement). Če mobilno vozlišče potrditve ne prejme, ponovno pošilja sporočilo osvežitev vezi, dokler pri tem ne uspe oziroma dokler se ne izteče časovnik. Osveževanje vezi in potrditev vezi poteka preko t.i. podaljšanih glav (*angl. extension headers*). Vsaka vez je veljavna le določen čas, nato pa jo je potrebno znova osvežiti. V nasprotnem primeru se vez izbriše, s čimer se izogne neveljavnim vezem za vozlišča, ki niso več aktivna.

Tunel med domačim agentom in mobilnim vozliščem je dvosmeren. To pomeni, da tudi pakete, ki so namenjeni dopisnemu vozlišču, mobilno vozlišče najprej tunelira do domačega agenta, ta pa jih pošlje naprej dopisnemu vozlišču. Postopek dvosmernega tuneliranja s pripadajočimi izvornimi in ponornimi IP naslovi prikazuje slika 2.1. Če bi mobilno vozlišče pošiljalo

pakete neposredno dopisnemu vozlišču in pri tem uporabilo začasni naslov, bi to pri transportnem protokolu TCP zaradi spremenjene vtičnice povzročilo prekinitev seje. Problem je sicer možno rešiti z implementacijo takšnega transportnega protokola (na primer SCTP), ki bi to spremembo ustrezno upošteval in seje ne bi prekinil, vendar potem ni izpolnjena zahteva po neodvisnosti od zgornjih slojev. Če pa bi mobilno vozlišče pošljalo iz gostujočega omrežja pakete z izvornim domačim naslovom, bi prišlo do vstopnega filtriranja in paketi sploh ne bi prišli do cilja. Z dvosmernim tuneliranjem je torej zagotovljeno, da dopisna vozlišča ne glede na lokacijo mobilnega vozlišča vidijo samo njegov domači naslov, medtem ko so začasni naslovi skriti. S tega stališča dopisna vozlišča mobilnosti sploh ne zaznajo in je za njih transparentna. Po odstranitvi ovojnice (*angl. decapsulation*) velja podobno tudi na strani mobilnega vozlišča. Glede na izvorni naslov notranje glave je učinek isti, kot če bi z dopisnim vozliščem komunicirala neposredno. S tem sta izpolnjeni zahtevi **neprekinjenosti vzpostavljenе komunikacije** in **neodvisnosti od zgornjih slojev**.



Slika 2.1: Dvosmerno tuneliranje (po [1])

Zahtevo o **neodvisnosti od spodnjih slojev** smo predpostavili že v osnovi. Če je zagotovljena povezava z dostopovnim usmerjevalnikom, potem vsa komunikacija pri vzpostavljanju potrebnega tunela poteka na omrežnem sloju.

Skoraj popolna **ohranitev kontrole na strani terminalov** pa je zagotovljena z vgrajeno funkcionalnostjo za upravljanje mobilnosti v mobilnih vozliščih in v nekaj izbranih usmerjevalnikih, ki opravljajo funkcijo domačega agenta. V principu zadošča že en domači agent na omrežje, dopuščeno

pa je tudi, da jih je več (na primer, ko gre za večje omrežje, za zagotavljanje večje zanesljivosti, itd.). Ostali usmerjevalniki v internetu ostanejo nespremenjeni. K ohranitvi kontrole na strani terminalov prispeva tudi optimizacija usmerjanja (*angl. route optimization*), ki je razložena v nadaljevanju.

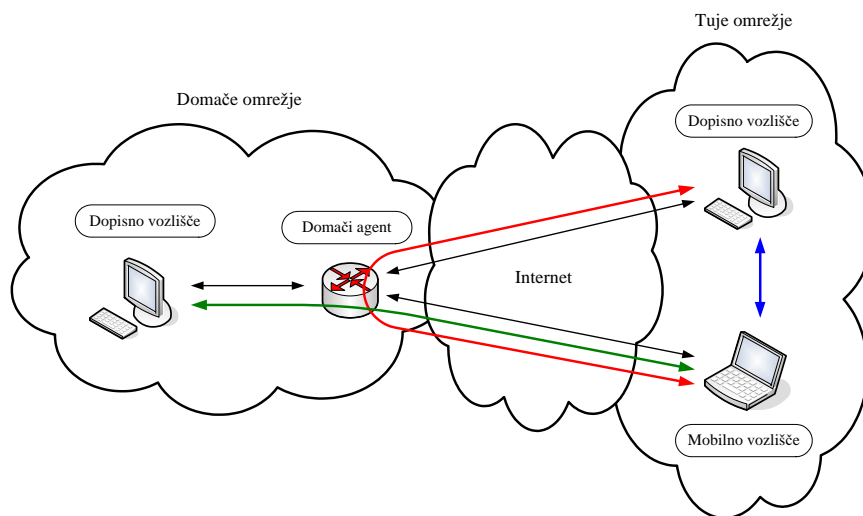
2.1.3 Optimizacija usmerjanja

Postopek upravljanja mobilnosti za končne terminale, kot smo ga opisali do sedaj, s stališča vzpostavljanja in vzdrževanja komunikacije zadostuje. Če izkoriščenost omrežnih virov ne bi predstavljala nikakršnega problema in če bi bilo procesiranje paketov ter njihov transport neskončno hiter, bi ta rešitev povsem zadostovala. Ker pa pri načrtovanju omrežij in sistemov vedno stremimo k čim boljši izkoriščenosti in čim manjšim zakasnitvam, je potrebna nadaljnja dopolnitev.

Ob posredovanju domačega agenta namreč prihaja do neoptimalnega usmerjanja. Usmerjanje med dopisnim vozliščem in domačim agentom ter domačim agentom in mobilnim vozliščem je sicer optimalno, oba dela skupaj pa v splošnem tvorita neoptimalno pot. Celotna pot je optimalna le v posebnem primeru, ko domači agent po naključju leži na pravem mestu v topologiji. Problem po analogiji spominja na trikotniško neenakost iz matematike in je v literaturi znan kot problem trikotniškega usmerjanja (*angl. triangular routing problem*). Zasluditi je mogoče tudi termin usmerjanje po pasji nogi (*angl. dog-leg routing*).

Ilustracija problema za najboljši in najslabši primer je prikazana na sliki 2.2. Najslabši primer nastopi, ko se dopisno vozlišče in mobilno vozlišče nahajata v istem omrežju, domači agent pa je lociran drugje. Optimalno usmerjanje bi bilo po poti označeni z modro puščico, vendar komunikacija dejansko poteka po poti, označeni z rdečo, ki je očitno neoptimalna. Kot najboljši primer pa je prikazana situacija, ko sta dopisno vozlišče in domači agent v istem omrežju. Optimalna usmerjevalna pot tu po naključju sovpada z dejansko (zelena puščica).

Posledic, ki izhajajo iz problema trikotniškega usmerjanja, je več. Prvič, paketi morajo prepotovati daljšo usmerjevalno pot, kar vnaša večje zakasnitve. Drugič, zaradi daljše usmerjevalne poti je v rabi več omrežnih virov (usmerjevalnikov in pasovne širine) in je torej izkoristek slab. To je še posebej problematično, če je zelo veliko mobilnih vozlišč. Prevelika obremenitev omrežij namreč zopet vodi k naraščanju zakasnitev. In tretjič, vpeljava domačega agenta pomeni zgoščevanje prometa v eni točki. Če ta odpove,



Slika 2.2: Problem trikotniškega usmerjanja (po [1])

postanejo vsa mobilna vozlišča, ki jim domači agent služi, nedosegljiva.

Da bi se naštetim posledicam izognili, je potrebno vpeljati dopolnitev. Prva možnost, ki ne deluje v splošnem, je ta, da mobilno vozlišče namesto domačega uporabi kar začasni naslov. Ta rešitev je primerna le za časovno kratke seje, ki jih vzpostavlja mobilno vozlišče in za katere se pričakuje, da bodo prekinjene, še preden bo mobilno vozlišče znova zamenjalo omrežje ter s tem začasni naslov. Primera aplikacij te vrste sta brskanje po spletu in komunikacija s strežnikom DNS.

Druga možnost dopolnitve, imenovana optimizacija usmerjanja, pa deluje v splošnem, vendar je za njo potrebno nekaj več režijske komunikacije. Zaradi tega je primerna predvsem za daljše seje, deluje pa ne glede na to, ali sejo vzpostavlja mobilno ali dopisno vozlišče.

Cilj optimizacije usmerjanja je, da dopisno in mobilno vozlišče komunicirata neposredno, brez posredovanja vmesnih vozlišč. V ta namen mora mobilno vozlišče obvestiti dopisno vozlišče o novi lokaciji. To stori na podoben način, kot pri obveščanju domačega agenta. Dopisnemu vozlišču pošlje osvežitev vezi z domačim in trenutnim začasnim naslovom. Dopisno vozlišče spravi vez v pomnilnik vezi in odgovori s potrditvijo vezi. Od tu naprej vozlišči lahko komunicirata neposredno, vendar morata, če želita obdržati neprekinjenost vzpostavljenih povezav, to storiti na določen način. Še vedno je namreč potrebno poskrbeti za prikritost mobilnosti višjim slojem in pri tem paziti na vstopno filtriranje. Pri optimizaciji usmerjanja je to rešeno z uporabo podaljšane glave. Vsak paket, ki potuje neposredno med vozli-

ščema ima v glavi pravilni izvorni in ponorni naslov, domači naslov pa je pripet kot opcija v podaljšani glavi. Ko posamezno vozlišče paket prejme, v glavi najprej začasni naslov nadomesti z domačim in ga šele nato posreduje višjim slojem. Postopek je razviden s slike 2.3.

Tehnično bi za neposredno komunikacijo med dopisnim in mobilnim vozliščem lahko uporabljali tudi tuneliranje, na enak način kot je bilo to uporabljeno med domačim agentom in mobilnim vozliščem. V standardu pa je vendarle sprejeta rešitev z uporabo podaljšanih glav. Rešitev je bolj posledica razpleta dogodkov pri razvoju standarda kot nekih tehtnih razlogov. Načeloma bi delovalo oboje enako dobro.

2.1.4 Varnost

Zagotavljanje varnosti pri mobilnosti v internetu je obsežna tema, ki v večini presega namene tega poglavja. Tu opisujemo le principe, ki so pomembni za celovito razumevanje poteka optimizacije usmerjanja.

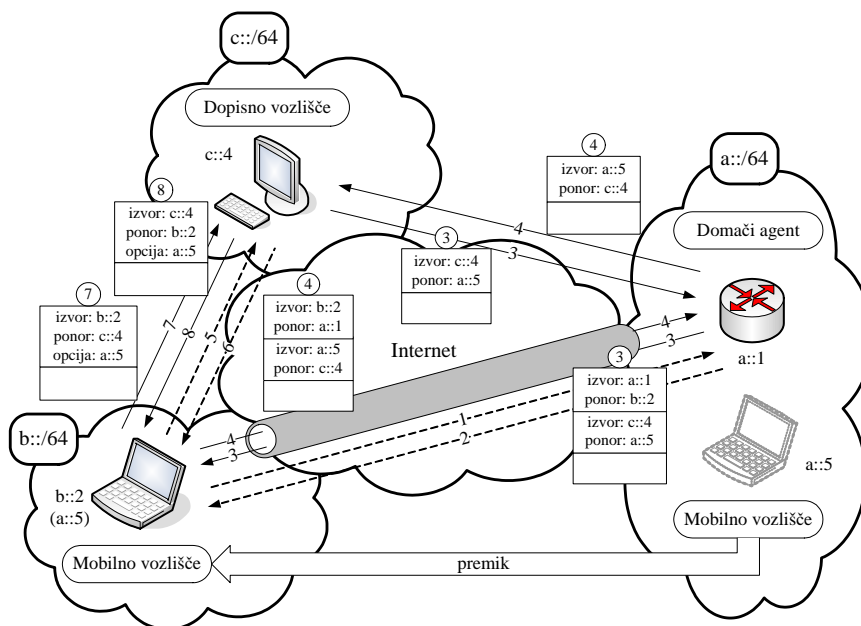
Varnost Mobile IPv6 protokola se zagotavlja na dva načina, v odvisnosti od tega, s kom komunicira mobilno vozlišče. Pri relaciji mobilno vozlišče - domači agent se uporablja protokol IPsec. Natančneje tega načina ne bomo opisovali, ker IPsec ni specifičen za mobilno okolje, pač pa se ga uporablja v splošnih IPv6 omrežjih. Njegovo delovanje ni ključno za metode in principe, ki jih opisujemo v disertaciji.

Pri relaciji mobilno vozlišče - dopisno vozlišče pa se izvaja postopek RR (Return Routability Procedure), ki je specifičen za mobilno okolje. Namen tega postopka je, da dopisno vozlišče preveri, ali je mobilno vozlišče res tisto, za katerega se izdaja. Če dopisna vozlišča tega ne bi preverjala, bi osvežitev vezi s svojim naslovom in domačim naslovom mobilnega vozlišča lahko pošiljala tretja vozlišča, ki bi na ta način prestrezala promet, namenjen mobilnemu vozlišču. Postopek je sledeč. Ko se mobilno vozlišče premakne in že pridobi novi začasni naslov, pošlje dopisnemu vozlišču dve sporočili, v kateri vključi svoja piškotka (*angl. cookie*). Sporočilo HOTI (HOMe address Test Init) tunelira preko domačega agenta, sporočilo COTI (Care-Of address Test Init) pa pošlje neposredno proti dopisnemu vozlišču. Ta mobilnemu vozlišču ravno tako preko obeh usmerjevalnih poti odgovori s sporočili HOT (HOMe Test) in COT (Care-Of Test). V njiju vključi od mobilnega vozlišča prejeta piškotka in doda svoje, enolično generirane varnostne podatke. Na podlagi tako prejetih podatkov mobilno vozlišče sestavi ključ, ki zagotavlja, da sta oba naslova zares dodeljena prav temu vozlišču. Ključ se uporablja pri preverjanju pristnosti (*angl. authentication*) sporočil

osvežitev vezi in potrditev vezi.

2.1.5 Primer poteka komunikacije

Razlago delovanja protokola Mobile IPv6 v tem poglavju ilustrirajmo še s tipičnim primerom, prikazanim na sliki 2.3.



Slika 2.3: Prikaz mobilnosti končnih vozlišč s primerom

Mobilno vozlišče se premakne iz domačega omrežja s predpono a::/64, kjer ima naslov a::5, v tuje omrežje s predpono b::/64, kjer pridobi začasni naslov b::2. O tem s sporočilom osvežitev vezi obvesti domačega agenta (puščica 1), ta pa vez potrdi (puščica 2). Dalje so na sliki prikazani paketi s koristno vsebino. Prazno polje pri tem pomeni vsebino paketa, polje nad njim pa glavo paketa z ustreznimi naslovi. Ko želi dopisno vozlišče, ki ima naslov c::4, z mobilnim vzpostaviti zvezo, pošlje pakete na njegov domači naslov. Ker mobilnega vozlišča ni v domačem omrežju, pakete prestreže domači agent z naslovom a::1 in jih tunelira na začasni naslov mobilnega vozlišča (puščica 3). Pri tuneliranju sta glavi dve, notranja (srednje polje) in zunanja (zgornje polje). Mobilno vozlišče dopisnemu odgovarja preko istega tunela (puščica 4). Na tem mestu algoritem v mobilnem vozlišču določi, naj se izvede optimizacija usmerjanja. V ta namen se najprej izvede postopek preverjanja RR, ki na sliki zaradi preglednosti ni prikazan. Mobilno vozlišče pošlje sporočili HOTI in COTI, dopisno pa odgovori s sporočili HOT

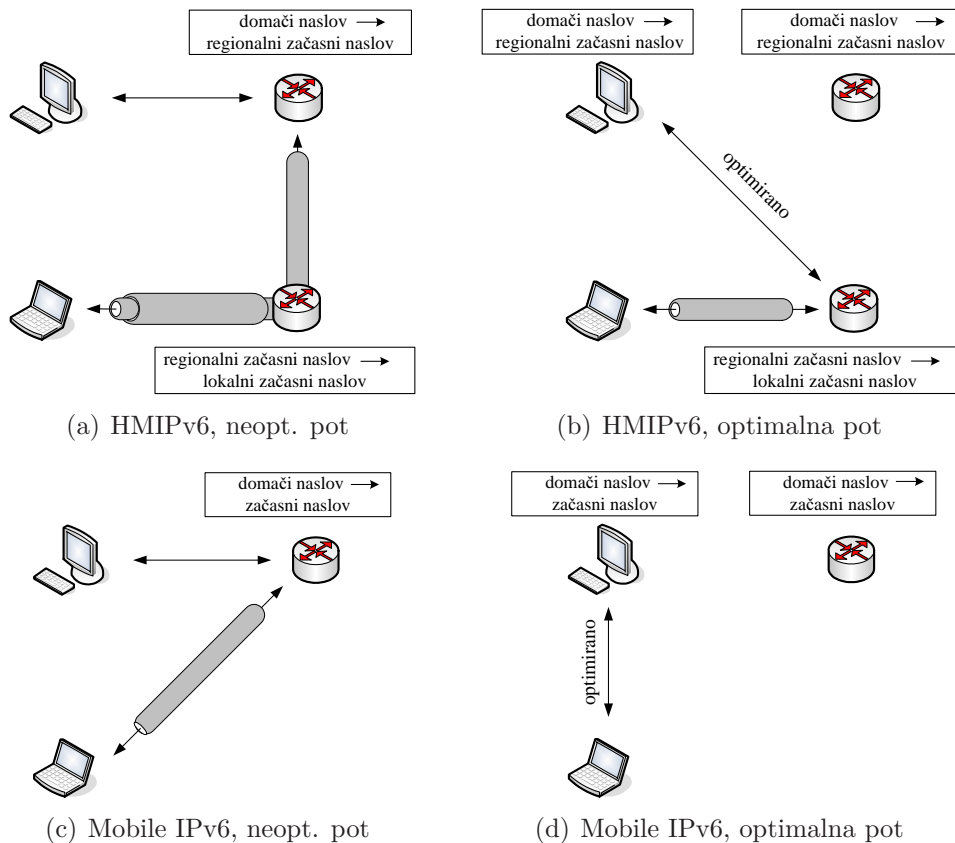
in COT. Mobilno vozlišče nato sestavi ključ in pošlje dopisnemu vozlišču sporočilo osvežitev vezi, ki je opremljeno s podatki za preverjanje pristnosti (puščica 5). Z le-temi je opremljeno tudi sporočilo potrditev vezi (puščica 6). Nadaljnja komunikacija poteka brez posredovanja domačega agenta, neposredno med mobilnim in dopisnim vozliščem. Pri tem je domači naslov pripet kot opcija v podaljšani glavi (puščici 7 in 8), kar je na sliki prikazano v glavi paketa v vrstici "opcija:". Če bi se po opisanem postopku mobilno vozlišče znova premaknilo v novo tuje omrežje, bi moralo o tem z osvežitvijo vezi ponovno obvestiti domačega agenta ter sprožiti postopek RR in poslati osvežitev vezi še dopisnemu vozlišču.

2.2 HMIPv6 protokol

Hierarhični protokol HMIPv6 (Hierarchical Mobile IPv6) [27] je bil zasnovan kot razširitev Mobile IPv6 protokola z namenom, da se izboljša izročanje. Za delovanje HMIPv6 protokola je potrebna uvedba nove entitete, imenovane sidrna točka mobilnosti (MAP - Mobility Anchor Point). Sidrna točka igra zelo podobno vlogo kot domači agent mobilnega vozlišča, le da je njeno delovanje omejeno na neko območje. V topologiji je postavljena nekoliko globlje, tako da zagotavlja povezljivost v hrbtenično IP omrežje več dostopnim usmerjevalnikom.

Ko se mobilno vozlišče premakne v območje pokritja sidrne točke, izve preko signalizacijskega sporočila RA (Router Advertisement) za njeno prisotnost. Nato oblikuje naslov, imenovan regionalni začasni naslov (RCoA - Regional CoA), ki je topološko pravilen na njeni povezavi (*angl. link*). Ta naslov igra podobno vlogo kot domači naslov pri domačem agentu. Mobilno vozlišče pošlje sidrni točki sporočilo osvežitev vezi, ki povezuje regionalni začasni naslov z lokalnim začasnim naslovom (LCoA - on-Link CoA). Lokalni začasni naslov je topološko pravilen na njegovi trenutni dostopni povezavi. Sidrna točka si to vez zapiše v pomnilnik vezi in odgovori s potrditvijo. Po prejeti potrditvi pošlje mobilno vozlišče osvežitev vezi še domačemu agentu. Pri tem ne pošlje vezi med domačim in lokalnim začasnim naslovom, temveč vez med domačim in regionalnim začasnim naslovom. Domači agent tako prestrežene pakete tunelira do sidrne točke, ta pa jih vnovič prestreže in tunelira na lokalni začasni naslov mobilnega vozlišča. Tuneliranje pri HMIPv6 protokolu torej poteka hierarhično, v dveh nivojih, kot je prikazano na sliki 2.4(a). Mobilno vozlišče lahko izvede tudi optimizacijo usmerjanja, na podoben način kot pri osnovnem Mobile IPv6 protokolu.

Dopisnemu vozlišču pošlje sporočilo osvežitev vezi, ki povezuje njegov domači naslov z regionalnim začasnim naslovom. Vse pakete dopisnih vozlišč torej zopet prestreza sidrna točka in jih tunelira na lokalni začasni naslov mobilnega vozlišča, kot prikazuje slika 2.4(b).

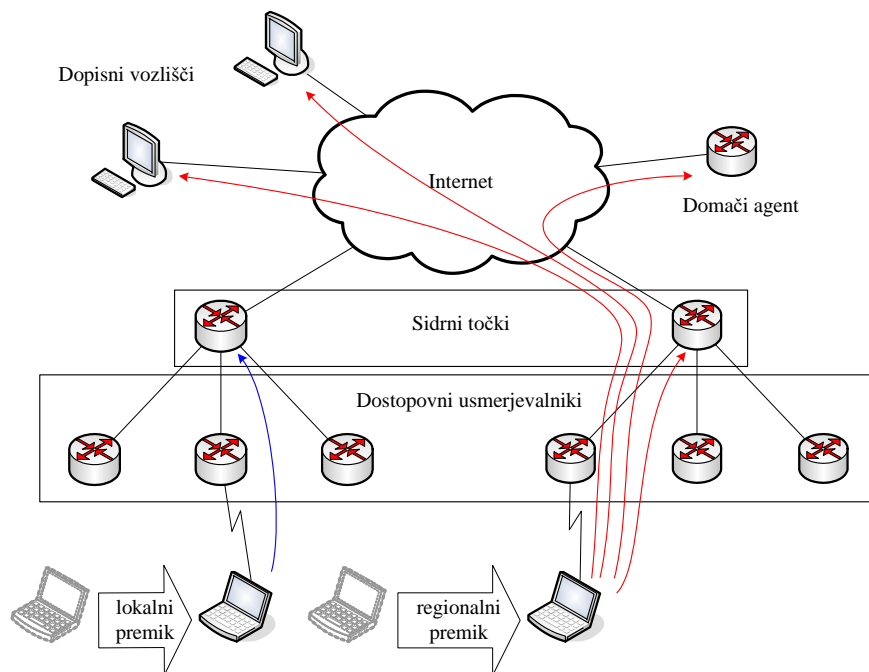


Slika 2.4: Primerjava protokola HMIPv6 s protokolom Mobile IPv6

Opisano delovanje na prvi pogled vnaša dodatno neoptimalnost pri usmerjanju, saj morajo paketi pri Mobile IPv6 protokolu pred optimizacijo usmerjanja (slika 2.4(c)) prečkati le eno entiteto, medtem ko morajo pri HMIPv6 protokolu prečkati dve (slika 2.4(a)). Še več, po optimizaciji usmerjanja se pri protokolu Mobile IPv6 (slika 2.4(d)) paketi izognejo kakršnemukoli posredovanju, medtem ko pri HMIPv6 protokolu (slika 2.4(b)) sidrna točka še vedno posreduje.

Prednost tega pristopa je skrita v dinamičnem izbiranju najprimernejše lokacije sidrne točke za mobilno vozlišče. Domači naslov je namreč stalen, z njim pa je stalna tudi lokacija domačega agenta. Če je mobilno vozlišče zelo oddaljeno, je osveževanje vezi lahko dolgotrajno. Po drugi strani mobilno

vozlišče sproti izbira najprimernejšo sidrno točko. Če izbere sidrno točko na primerni lokaciji, to pospeši osveževanje vezi in hkrati zmanjša režijo, pri čemer optimalnost usmerjevalne poti ostane. Pri dobri izbiri sidrne točke namreč mobilno vozlišče ob menjavi dostopovnega omrežja tipično zamenja le lokalni začasni naslov, medtem ko regionalni začasni naslov ostane enak. Mobilno vozlišče pošlje v tem primeru osvežitev vezi z novim lokalnim naslovom samo sidrni točki, ki je topološko blizu, kot je prikazano z modro puščico na sliki 2.5. Dopisnih vozlišč in domačega agenta ni potrebno obveščati, saj imajo v pomnilniku vezi zapisan regionalni začasni naslov. Kadar se zgodi regionalni premik, pa HMIPv6 protokol proti običajnemu Mobile IPv6 protokolu celo nekaj izgubi, saj je potrebno o novem regionalnem začasnem naslovu po standardnem postopku obveščati vsa dopisna vozlišča in domačega agenta, poleg tega pa je potrebna menjava sidrne točke, kar vnaša dodatno režijo. Signalizacija ob regionalnem premiku je na sliki 2.5 ponazorjena z rdečimi puščicami. Lokacija sidrne točke je torej zelo pomembna. Izbrana mora biti dovolj blizu mobilnega vozlišča, da je osveževanje vezi hitro in da usmerjevalna pot ohrani optimalnost ter hkrati dovolj daleč, da je ni potrebno pogosto menjati.



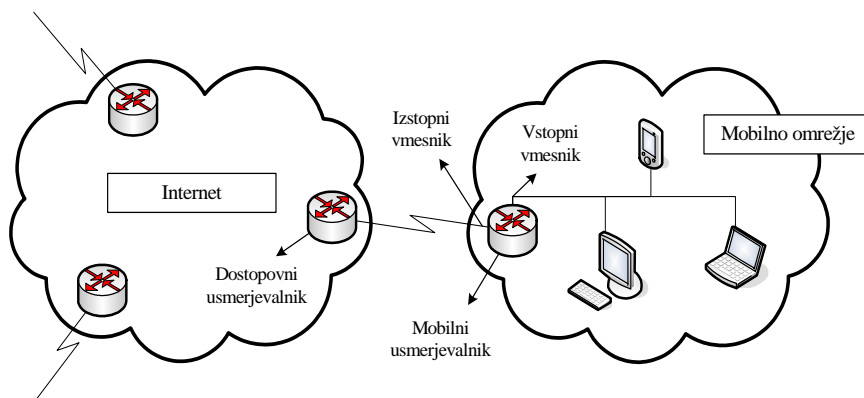
Slika 2.5: Prikaz razlike v signalizaciji pri lokalnem in regionalnem premiku

Kot prednost opisanega delovanja lahko štejemo tudi deloma zagotovljeno lokacijsko zasebnost. Dopisna vozlišča in domači agent namreč ne

poznajo lokalnega začasnega naslova, pač pa le regionalnega. Sledenje mobilnemu vozlišču je do neke mere tako še vedno možno, vendar je njegova natančnost bistveno manjša.

2.2.1 Mobilnost omrežij

Uporaba protokola HMIPv6 je še posebej privlačna pri upravljanju mobilnosti t.i. mobilnih omrežij. Gre za omrežja, ki se premikajo kot celota in dinamično spreminjajo točko povezave v internet. Z vpeljavo mobilnih omrežij so tehnološke zahteve za terminale nižje, t.i. izbruhi osveževanja vezi so izločeni, kontrola pa je centralizirana. Mobilna omrežja sestojijo iz enega ali več IP podomrežij in so povezana v internet preko enega ali več mobilnih usmerjevalnikov (MR - Mobile Router) [39, 66]. Ti delujejo kot stične točke, ki se na izstopnem vmesniku (*angl. egress interface*) povežejo v internet, na vstopnem vmesniku (*angl. ingress interface*) pa omogočajo povezljivost terminalnim napravam. Spreminjanje interne konfiguracije mobilnega omrežja in spreminjanje točke povezave izstopnega vmesnika v internet, sta pri tem ortogonalna procesa, ki ne vplivata drug na drugega. Shematičen prikaz mobilnega omrežja je prikazan na sliki 2.6.



Slika 2.6: Shematičen prikaz mobilnega omrežja

Tipična primera mobilnih omrežij sta osebno omrežje (PAN - Personal Area Network) in omrežje vozila (VAN - Vehicular Area Network) [38]. Slednjega lahko obravnavamo kot lokalno omrežje (LAN - Local Area Network), ki se nahaja v vozilu in nudi povezljivost v internet.

Optimizacija usmerjanja pri mobilnih omrežjih

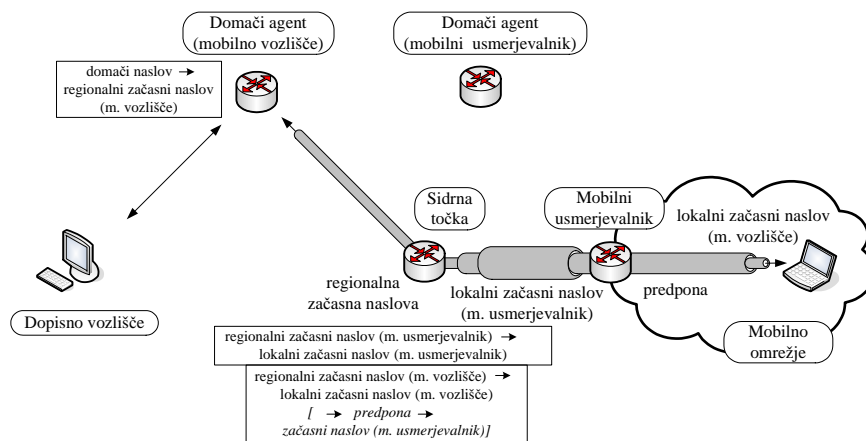
Delovanje osnovne podpore mobilnosti omrežij opisuje RFC 3963 [39]. Protokol je zasnovan kot razširitev protokola Mobile IPv6 in omogoča vsem terminalom znotraj omrežja, tudi tistim, ki nimajo podpore mobilnosti, neprekinjenost vzpostavljene komunikacije in stalno dosegljivost, pri čemer je zagotovljena neodvisnost od zgornjih in spodnjih slojev. Podobno kot pri protokolu Mobile IPv6, se vzpostavi dvosmerni tunel med domačim agentom in mobilno entiteto, ki je v tem primeru mobilno omrežje oziroma natančneje mobilni usmerjevalnik.

V nasprotju z osnovnim Mobile IPv6 protokolom za končna vozlišča, pa osnovna podpora mobilnosti omrežij, postopka optimizacije usmerjenja ne predvideva. To za seboj potegne vrsto težav, opisanih v poglavju 2.1.3, ki pa so v primeru mobilnih omrežij zaradi večjega števila končnih vozlišč še večje. Zato je bilo v literaturi predlaganih več metod za optimizacijo usmerjanja pri mobilnih omrežjih. Delimo jih na rešitve, ki delujejo po principu makro-mobilnosti in rešitve, ki vključujejo mikro-mobilnost. V prvo skupino štejemo predstavnike kot so Sporočila PSBU [67], protokol ORC [40], Oprtani naslovi [41] in OptiNets [42]. Predstavniki druge skupine pa vpeljuje optimizacijo usmerjanja s hierarhičnim pristopom [44] in tako obenem ponuja tudi prednosti lokaliziranega upravljanja mobilnosti.

Hierarhični pristop k optimizaciji usmerjanja

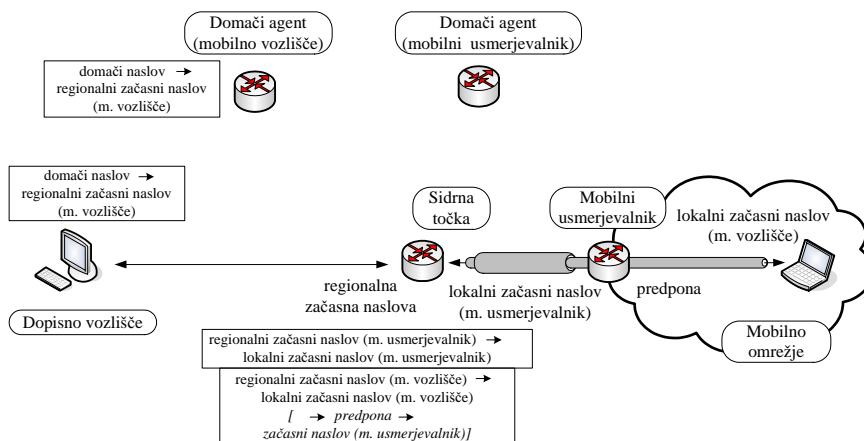
Hierarhični pristop k optimizaciji usmerjanja za mobilna omrežja temelji na HMIPv6 protokolu za končna vozlišča. Mobilna vozlišča, ki so znotraj mobilnega omrežja, na povezavi sidrne točke na običajen način oblikujejo svoj regionalni začasni naslov in ga z osvežitvijo vezi povežejo s svojim lokalnim začasnim naslovom. Mobilni usmerjevalnik na povezavi sidrne točke prav tako oblikuje svoj regionalni začasni naslov, vendar poleg svojega lokalnega začasnega naslova pošlje sidrni točki tudi vez s predponami, ki jih oglašuje znotraj mobilnega omrežja. Sidrna točka nato na podlagi vseh prejetih vezi oblikuje sestavljeno vez. Ta povezuje regionalni naslov mobilnega vozlišča z njegovim lokalnim naslovom, le-tega pa dalje z njegovo predpono, ki je oglaševana znotraj mobilnega omrežja in je povezana z lokalnim naslovom mobilnega usmerjevalnika (nagnjena pisava v oglatem oklepaju na sliki 2.7). Domači agent mobilnega usmerjevalnika je s trenutkom, ko sidrna točka in domači agent mobilnega vozlišča prejmeta vse vezi, izvzet iz usmerjevalne poti (slika 2.7).

Ko mobilna vozlišča izvedejo z dopisnimi vozlišči še optimizacijo usmer-



Slika 2.7: Usmerjevalna pot pri mobilnih omrežjih s hierarhičnim pristopom

janja po protokolu Mobile IPv6, pa je izvzet še domači agent mobilnega vozlišča in pot je v celoti optimalna (slika 2.8).



Slika 2.8: Usmerjevalna pot po optimizaciji usmerjanja

Hierarhični pristop optimizacije usmerjanja je v osnovi zasnovan za vozlišča, ki podpirajo upravljanje mobilnosti. Za vozlišča, ki nimajo te podpore, pa hierarhični pristop uvaža na mobilnem usmerjevalniku funkcionalnost t.i. namestnika (*angl. proxy*).

V primeru, ko je lokacija sidrne točke izbrana dobro, lastnosti hierarhičnega pristopa močno presegajo lastnosti ostalih predlogov. Izročanje je bistveno hitrejše, povečljivost pa zagotovljena ne glede na velikost omrežja ali število dopisnih vozlišč, saj je ob tipičnem lokalnem premiku potrebna le ena osvežitev vezi, to je lokalni začasni naslov mobilnega usmerjevalnika z njegovim regionalnim začasnim naslovom. Za primere, ko se mora zaradi

visoke geografske dinamičnosti mobilnega omrežja sidrna točka pogosto menjavati, pa se ti dve lastnosti močno poslabšata. Ob menjavi sidrne točke je namreč potrebna dodatna režija, ki raste z velikostjo mobilnega omrežja. Zato je učinkovitost protokola pri mobilnih omrežjih še bolj občutljiva na način izbiranja sidrnih točk kot pri običajnem protokolu HMIPv6 za končna vozlišča.

Poglavje 3

Prediktivni algoritmi za izbiro sider

V preteklosti je bilo predlaganih mnogo metod za izbiranje sidrnih točk. V tem poglavju na kratko opišemo njihovo delovanje. Nato predstavimo idejo, na kateri temelji naša metoda. Reprezentativne algoritme formalno zapišemo in jih primerjamo s simulacijami. Kot prvi na področju algoritme ovrednotimo tudi na realnih internetnih topologijah. Rezultati potrjujejo prednost predlaganega prediktivnega načina izbiranja sidrnih točk in smiselnost izbiranja sidrnih točk, ki so izven domene mobilnega vozlišča.

3.1 Pregled obstoječih algoritmov

Dosedanje predloge algoritmov za izbiro sidrnih točk po načinu delovanja delimo v 5 skupin:

- algoritme, zasnovane na oddaljenosti;
- algoritme, zasnovane na hitrosti;
- algoritme, zasnovane na zgodovini;
- adaptivne algoritme;
- ostale algoritme.

Medsebojno primerjavo nekaterih algoritmov najdemo v [29].

3.1.1 Algoritmi, zasnovani na oddaljenosti

V skupino algoritmov, zasnovanih na oddaljenosti, spada algoritem, privzet v specifikaciji HMIPv6 protokola [27], ki določa izbor najbolj oddaljene sidrne točke. Cilj tega pristopa je zmanjšati pogostost menjave sidrne točke. Protokol določa tudi, da lahko sidrne točke v signalizacijskih sporočilih oglašujejo t.i. preferenčno vrednost, ki določa pripravljenost sidrne točke, da sprejme v upravljanje novo mobilno vozlišče. S to razširitvijo se lahko na primer preobremenjene sidrne točke izognejo dodatnim obremenitvam. V isto skupino sodijo tudi algoritmi, ki temeljijo na izboru najbližje sidrne točke.

V vsakem primeru gre za pristop, ki ne upošteva nikakršnih specifičnih lastnosti mobilnih terminalov, zato ga avtorji obravnavajo v glavnem kot referenco za primerjavo.

3.1.2 Algoritmi, zasnovani na hitrosti

V tej skupini je oddaljenost izbrane sidrne točke neposredno povezana s hitrostjo mobilnega terminala. Avtorji predpostavljajo, da se hitrejši terminali gibljejo po večjem geografskem območju in je torej bolje, da izbirajo bolj oddaljene sidrne točke, ki predvidoma pokrivajo več dostopovnih usmerjevalnikov.

Za oceno hitrosti v [46] uporabljajo število izročanj v opazovanem intervalu. V [54] merijo realno hitrost v kilometrih na uro, pri čemer prepotovane razdalje znotraj celic ocenijo pavšalno glede na njihovo površino. Te ocene so zaradi nepredvidljivega gibanja mobilnih vozlišč lahko napačne. Isti avtorji poskušajo to pomanjkljivost popraviti z uteženim povprečjem, ki upošteva preteklo hitrost terminala in kasneje z zmanjšanjem vpliva celic, znotraj katerih je ocenjena hitrost bistveno previsoka [68]. Učinkovitost tako izboljšanega algoritma preizkusijo tudi na sintetičnih topologijah, ki so sicer podobne drevesnim, vendar se območja pokritij sidrnih točk naključno prekrivajo [55]. Pri tem upoštevajo tudi obremenjenost sidrnih točk.

Natančnost meritev hitrosti poskušajo izboljšati tudi drugi avtorji. V [56] namesto uteženega povprečja hitrosti računajo uteženo povprečje preživetega časa v posameznih celicah, pri čemer utež določajo dinamično glede na standardno deviacijo časa. Enak princip določanja uteži, vendar s standardno deviacijo hitrosti, je uporabljen v [64].

V [36] primerjajo učinkovitost merjenja hitrosti v številu izročanj na časovno enoto in v razdalji na časovno enoto. Predstavijo tudi prednosti

integracije HMIPv6 protokola s hitrim izročanjem.

Zadnja dva predstavnika algoritmov iz te skupine poleg hitrosti terminala, ki izbira sidrno točko, upoštevata tudi hitrosti ostalih terminalov v omrežju [58, 60]. Terminal je uvrščen v primeren nivo hierarhije sidrnih točk glede na njegovo relativno hitrost, pri čemer se upošteva tudi, kako obremenjene so sidrne točke.

3.1.3 Algoritmi, zasnovani na zgodovini

Tudi avtorji algoritmov iz te skupine zagovarjajo, da je primernost oddaljenosti sidrne točke odvisna od velikosti območja, po katerem se gibljejo mobilni terminali, vendar ugotavljajo, da velikost območja ni odvisna samo od hitrosti. Zato predlagajo metode, ki beležijo, kje so se mobilni terminali gibali v preteklem obdobju in izberejo sidrno točko, ki najbolje ustreza temu vzorcu.

Po algoritmih, predlaganih v [45] in [48], mobilni terminali znotraj vnaprej določenega intervala beležijo, katere sidrne točke so bile razpoložljive. Ko se interval izteče, se med točkami, ki so bile razpoložljive cel interval, algoritem odloči za najbližjo. V [53] dopuščajo, da izbrana sidrna točka ni bila razpoložljiva skozi celoten interval, pač pa je presegla neko vnaprej določeno mejo razpoložljivosti. Upoštevajo tudi obremenjenost sidrnih točk.

3.1.4 Adaptivni algoritmi

Avtorji adaptivnih algoritmov vpeljejo dve izboljšavi. Prva izboljšava je, da poleg mobilnosti terminalov upoštevajo tudi njihovo aktivnost v smislu komunikacijskega prometa. Manjša kot je aktivnost v razmerju do mobilnosti, bolj oddaljena mora biti izbrana sidrna točka. Druga izboljšava pa je, da izbor najprimernejše sidrne točke ni odvisen samo od razmerja aktivnosti proti mobilnosti, pač pa tudi od trenutnega stanja v omrežju in od vrednostnega sistema posameznega mobilnega terminala.

Da bi upoštevali čim več dejavnikov, vpeljejo avtorji pojma cene dostave paketov in cene signalizacije. Mobilni terminal sproti izračunava, kakšna bo skupna cena izbire ene izmed razpoložljivih sidrnih točk ter izbere tisto, za katero bo skupna cena minimalna [47, 51]. Avtorji metode v [61] predlagajo izboljšavo, ki poleg aktivnosti terminala in njegove hitrosti upošteva tudi obseg premikanja. S tem je dosežen podoben učinek, kot so ga želeli doseči snovalci algoritmov, zasnovanih na zgodovini. Še naprej uporabljajo klasičen pojem cen.

Nekoliko posebna je metoda, po kateri avtorji ne merijo aktivnosti mobilnega vozlišča v prenesenem prometu na časovno enoto, pač pa v številu dopisnih vozlišč, s katerimi ima mobilno vozlišče vzpostavljeno sejo [62]. S tem uspejo natančneje upoštevati ceno signalizacije ob menjavi sidrnih točk, če komunikacija poteka po optimiziranih usmerjevalnih poteh.

V skupino adaptivnih algoritmov štejemo tudi predstavnika, ki sicer uporabljata razmerje aktivnosti proti mobilnosti in adaptivno prilagajanje, vendar ne z namenom izbora najprimernejše sidrne točke. Uporabljata ju za izbor najprimernejšega števila nivojev sidrnih točk v več nivojski strukturi [52] oziroma za določitev najprimernejše velikosti območja, ki naj ga izbrana sidrna točka pokriva [49]. Kot izboljšavo slednjega dela so v [63] predlagali še upoštevanje zgodovinskih vzorcev za izbor najprimernejše sidrne točke v prej določenem območju.

3.1.5 Ostali algoritmi

V to skupino štejemo preostale algoritme, ki jih po načinu delovanja ne moremo uvrstiti v nobeno od zgornjih skupin, razlikujejo pa se tudi med seboj.

V [57] uporabljajo metodo, ki temelji samo na porazdelitvi bremena med sidrnimi točkami. Pri tem uporabljajo drseče povprečje, s katerim ocenjujejo, kako obremenjene bodo sidrne točke v prihodnje.

Metoda v [59] sloni na ideji, da je potrebno minimizirati zakasnitve pri menjavah sidrnih točk, saj je to ključnega pomena za zagotavljanje kakovosti. V ta namen po vzoru cen, ki jih uporabljajo adaptivni algoritmi, izpeljejo izraz za povprečno zakasnitev izročanja. Algoritem izbira tiste sidrne točke, za katere je povprečna zakasnitev minimalna.

Avtorji študije [65] poskušajo delovanje HMIPv6 protokola izboljšati z več metodami hkrati. Sidrne točke izbirajo na podlagi hitrosti mobilnega vozlišča in zgodovine premikov, pri izročanju pa uporabljajo tudi hitro izročanje in oddajanje več prejemnikom (*angl. multicast*).

3.2 Opis predlaganih algoritmov

3.2.1 Osnovna ideja

Večina obstoječih predlogov temelji na popolni naključnosti gibanja mobilnega vozlišča. Realno pa se mobilna vozlišča premikajo po določenih ponavljajočih se vzorcih. Frekvenca ponovitev lahko variira od reda ur ali

dni do tednov, mesecev ali celo let. Ilustrativni primeri ponavljajočih vzorcev gibanja povprečnega človeka so pot v službo, na športne aktivnosti, na izlete, počitnice itd. V nekaterih primerih so ti vzorci zelo natančno definirani. Javna prevozna sredstva, kot so avtobusi, vlaki in letala, se premikajo po vnaprej določenih poteh. Pričakovano je, da se bo na teh prevoznih sredstvih uporabljalo koncept mobilnih omrežij [39], opisanih v poglavju 2.2.1. V teh omrežjih bodo vlogo posrednika med zunanjim internetom in terminali v vozilu igrali mobilni usmerjevalniki. Ker bodo mobilni usmerjevalniki upravljali mobilnost namesto terminalov, jim bodo na voljo informacije o dostopnih sidrskih točkah na ponavljajočih se poteh, s tem pa bo mogoča napoved oz. predikcija prihodnjih premikov.

Procedure za izbiranje sidrskih točk so lahko izvedene na katerikoli napravi, ki upravlja mobilnost, to pomeni bodisi na mobilnem usmerjevalniku, bodisi na končnem mobilnem vozlišču. V nadaljevanju zaradi preglednosti vse takšne naprave imenujemo mobilna vozlišča.

Naš predlog temelji na predpostavki, da se poznavanje dostopnosti sidrskih točk v prihodnosti lahko uporabi za izboljšanje izbire sidrskih točk. Predlogi algoritmov, ki so najbolj sorodni našemu predlogu, so algoritmi, ki temeljijo na zgodovini premikov [45, 48, 53]. Pri teh predlogih temelji izbira sidrskih točk na poznavanju gibanja mobilnega vozlišča v bližnji preteklosti. Pristop predstavlja poizkus upoštevanja vzorcev gibanja mobilnih vozlišč, vendar vodi k izbiranju sidrskih točk, ki so optimalne na delih poti, ki so bile že prepotovane.

3.2.2 Delovanje analiziranih algoritmov

Za oceno prednosti, ki jih ponuja poznavanje prihodnjih premikov, smo primerjali tri obstoječe referenčne algoritme s tremi tipi našega prediktivnega algoritma. V nadaljevanju je opisano delovanje vseh šestih. Za vsak algoritem je podana pripadajoča psevdokoda. Definicijo razredov, ki so uporabljeni v psevdokodah, podaja tabela 3.1. Funkcija posameznih spremenljivk je razvidna iz spodnjega opisa in analize v poglavju 3.3. Algoritmi so zapisani v obliki, ki omogoča enostavno razumevanje in niso optimizirani z vidika hitrega delovanja. Mobilno vozlišče je vedno registrirano pri izbrani sidrski točki kolikor časa je to mogoče. Nova sidrna točka je izbrana šele potem, ko mobilno vozlišče zapusti območje pokritja prejšnje sidrne točke.

Obstoječi referenčni algoritmi, ki smo jih uporabili, so osnovni algoritem, ki temelji na najdaljši razdalji, algoritem, ki temelji na hitrosti in algoritem, ki temelji na zgodovini premikov. Imenujemo jih *'najdlje'*, *'hitrost'* in *'zgo-*

```

class sidrna_tocka
    RazdaljaDoMV
    CasVidnosti
    ZadnjiRob
    PrednjiRob
endclass

class dostopovna_tocka
    SeznamVidnihST
    PorazdelitevHitrosti
endclass

class mobilno_vozlisce
    Hitrost
    VerjetnostPremikaNaprej
    OpazovaneDT
    ZahtevanaPrisotnost
endclass

```

Tabela 3.1: Definicije uporabljenih razredov

dovina *š* *najbližja*'. '*Najdlje*' (algoritem 1) je algoritem, ki je bil predlagan v [27]. Med sidrnimi točkami, ki so na največji razdalji od mobilnega vozlišča, algoritem naključno izbere eno. Pri algoritmu '*hitrost*' (algoritem 2) najhitrejša mobilna vozlišča izberejo najbolj oddaljene sidrne točke in obratno. Porazdelitev hitrostnih intervalov v odvisnosti od oddaljenosti sidrnih točk je definirana vnaprej. Med simulacijami se porazdelitev ne spreminja. Pri '*zgodovina* *š* *najbližja*' (algoritem 3) mobilna vozlišča opazujejo sidrne točke, ki so bile na razpolago na pretekli poti. Dolžina opazovanega intervala, merjena v številu zamenjanih dostopovnih točk, je določena vnaprej. Mobilno vozlišče naključno izbere najbližjo sidrno točko, ki je trenutno vidna in je bila vidna skozi celoten opazovani interval. Če taka sidrna točka ne obstaja, izbere mobilno vozlišče najbližjo sidrno točko izmed tistih, ki so bile v opazovanem intervalu prisotne najdlje.

Prva dva od treh tipov našega predlaganega algoritma temeljita na algoritmu '*hitrost*' z razširitvijo zmožnosti predikcije. Z uporabo '*hitrost*' algoritma najprej določita skupino sidrnih točk s primerno razdaljo od mobilnega vozlišča. V naslednjem koraku algoritma izmed sidrnih točk v tej skupini izbereta tisto, za katero je predvideno, da bo prisotna najdlje. Algoritma se razlikujeta v natančnosti predvidevanja. Prvi, označen kot '*hi-*

Algoritem 1 'najdlje'**Vhod:** *sidrna_tocka* *PrejsnjaST*, *dostopovna_tocka* *TrenutnaDT***Izhod:** *sidrna_tocka* *NaslednjaST*

```

1: if PrejsnjaST  $\notin$  TrenutnaDT.SeznamVidnihST then
2:   NaslednjaST  $\leftarrow$  TrenutnaDT.SeznamVidnihST[0]
3:   for all sidrna_tocka ST  $\in$  TrenutnaDT.SeznamVidnihST do
4:     if ST.RazdaljaDoMV > NaslednjaST.RazdaljaDoMV then
5:       NaslednjaST  $\leftarrow$  ST
6:     end if
7:   end for
8:   return NaslednjaST
9: else
10:  return PrejsnjaST
11: end if

```

trost \mathcal{E} *prihodnost*' (algoritem 5) ima na razpolago popolno informacijo o dostopnosti sidrnih točk v prihodnosti. Zato v skupini sidrnih točk s primerno oddaljenostjo vedno izbere tisto, ki bo res prisotna najdlje. Drugi algoritem, ki ga označujemo kot '*hitrost* \mathcal{E} *smer*' (algoritem 4), pa ima na voljo le informacijo o tem, kam se bo vozlišče premikalo dolgoročno. Ker je trenutno gibanje mobilnega vozlišča naključno, z algoritmom '*hitrost* \mathcal{E} *smer*' izbrana sidrna točka ne bo vedno optimalna.

Tretji tip našega prediktivnega algoritma, ki ga označujemo kot '*prihodnost* \mathcal{E} *najbližja*', deluje na podoben način kot '*zgodovina* \mathcal{E} *najbližja*'. Glavna razlika je v tem, da '*prihodnost* \mathcal{E} *najbližja*' (algoritem 6) namesto preteklega opazovanega intervala uporablja selekcijski interval, ki se nanaša na prihodnost. Dolžina intervala je določena vnaprej. Algoritem izbere najbližjo sidrno točko, ki bo na voljo skozi celoten selekcijski interval. Informacija o dostopnosti sidrnih točk v prihodnosti je popolna. Če ni sidrnih točk, ki bi bile na voljo skozi celoten selekcijski interval, izbere algoritem izmed tistih, ki bodo na voljo najdlje, najbližjo. Daljši selekcijski interval vodi k manj pogostim menjavam sidrnih točk, vendar so te v povprečju bolj oddaljene in obratno. V praksi bo dolžina intervala izbrana v odvisnosti od trenutnih razmer in preferenc mobilnega vozlišča.

Algoritem 2 'hitrost'

Vhod: sidrna_točka $PrejsnjaST$, dostopovna_točka $TrenutnaDT$, mobilno_vozlisce $MobilnoVozlisce$ **Izhod:** sidrna_točka $NaslednjaST$

```

1: if  $PrejsnjaST \notin TrenutnaDT.SeznamVidnihST$  then
2:    $IzbranaRazdalja \leftarrow 0$ 
3:   while  $MobilnoVozlisce.Hitrost >$ 
      $TrenutnaDT.PorazdelitevHitrosti[IzbranaRazdalja]$  do
4:      $IzbranaRazdalja \leftarrow IzbranaRazdalja + 1$ 
5:   end while
6:   for all sidrna_točka  $ST \in TrenutnaDT.SeznamVidnihST$  do
7:     if  $ST.RazdaljaDoMV = IzbranaRazdalja$  then
8:        $NaslednjaST \leftarrow ST$ 
9:       break
10:    end if
11:  end for
12:  return  $NaslednjaST$ 
13: else
14:  return  $PrejsnjaST$ 
15: end if

```

Algoritem 3 'zgodovina & najbližja'

Vhod: sidrna_točka *PrejsnjaST*, dostopovna_točka *TrenutnaDT*, mobilno_vozlisce *MobilnoVozlisce*

Izhod: sidrna_točka *NaslednjaST*

```

1: if PrejsnjaST  $\notin$  TrenutnaDT.SeznamVidnihST then
2:   SeznamSTZgodovina  $\leftarrow$  {}
3:   for all dostopovna_točka DT  $\in$  MobilnoVozlisce.OpazovaneDT
4:     do
5:       for all sidrna_točka ST  $\in$  DT.SeznamVidnihST do
6:         if ST  $\in$  SeznamSTZgodovina then
7:           ST.CasVidnosti  $\leftarrow$  ST.CasVidnosti + 1
8:         else
9:           ST.CasVidnosti  $\leftarrow$  1
10:          SeznamSTZgodovina  $\leftarrow$  {SeznamSTZgodovina, ST}
11:        end if
12:      end for
13:    end for
14:    MaksPrisotnost  $\leftarrow$  0
15:    SeznamPrimernihST  $\leftarrow$  {}
16:    for all sidrna_točka ST  $\in$  TrenutnaDT.SeznamVidnihST do
17:      if ST.CasVidnosti  $\geq$  MobilnoVozlisce.ZahtevanaPrisotnost
18:        then
19:          SeznamPrimernihST  $\leftarrow$  {SeznamPrimernihST, ST}
20:        else if ST.CasVidnosti  $>$  MaksPrisotnost then
21:          MaksPrisotnost  $\leftarrow$  ST.CasVidnosti
22:        end if
23:      end for
24:    if SeznamPrimernihST = {}  $\wedge$  MaksPrisotnost  $>$  0 then
25:      for all sidrna_točka ST  $\in$  TrenutnaDT.SeznamVidnihST do
26:        if ST.CasVidnosti = MaksPrisotnost then
27:          SeznamPrimernihST  $\leftarrow$  {SeznamPrimernihST, ST}
28:        end if
29:      end for
30:    if SeznamPrimernihST = {} then
31:      NaslednjaST  $\leftarrow$  TrenutnaDT.SeznamVidnihST[0]
32:    else
33:      NaslednjaST  $\leftarrow$  SeznamPrimernihST[0]
34:      for all sidrna_točka ST  $\in$  SeznamPrimernihST do
35:        if ST.RazdaljaDoMV  $<$  NaslednjaST.RazdaljaDoMV then
36:          NaslednjaST  $\leftarrow$  ST
37:        end if
38:      end for
39:    end if
40:    return NaslednjaST
41:  else
42:    return PrejsnjaST
43: end if

```

Algoritem 4 *'hitrost & smer'*

Vhod: sidrna_točka *PrejsnjaST*, dostopovna_točka *TrenutnaDT*, mobilno_vozlisce *MobilnoVozlisce***Izhod:** sidrna_točka *NaslednjaST*

```

1: if PrejsnjaST  $\notin$  TrenutnaDT.SeznamVidnihST then
2:   IzbranaRazdalja  $\leftarrow$  0
3:   while MobilnoVozlisce.Hitrost >
     TrenutnaDT.PorazdelitevHitrosti[IzbranaRazdalja] do
4:     IzbranaRazdalja  $\leftarrow$  IzbranaRazdalja + 1
5:   end while
6:   for all sidrna_točka ST  $\in$  TrenutnaDT.SeznamVidnihST do
7:     if ST.RazdaljaDoMV = IzbranaRazdalja then
8:       NaslednjaST  $\leftarrow$  ST
9:       break
10:    end if
11:  end for
12:  for all sidrna_točka ST  $\in$  TrenutnaDT.SeznamVidnihST do
13:    if MobilnoVozlisce.VerjetnostPremikaNaprej > 0.5 then
14:      if ST.RazdaljaDoMV = IzbranaRazdalja  $\wedge$  ST.PrednjiRob >
        NaslednjaST.PrednjiRob then
15:        NaslednjaST  $\leftarrow$  ST
16:      end if
17:    else
18:      if ST.RazdaljaDoMV = IzbranaRazdalja  $\wedge$  ST.ZadnjiRob <
        NaslednjaST.ZadnjiRob then
19:        NaslednjaST  $\leftarrow$  ST
20:      end if
21:    end if
22:  end for
23:  return NaslednjaST
24: else
25:  return PrejsnjaST
26: end if

```

Algoritem 5 *'hitrost & prihodnost'*

Vhod: *sidrna_tocka* *PrejsnjaST*, *dostopovna_tocka* *TrenutnaDT*, *obilno_vozlisce* *MobilnoVozlisce***Izhod:** *sidrna_tocka* *NaslednjaST*

```

1: if PrejsnjaST  $\notin$  TrenutnaDT.SeznamVidnihST then
2:   IzbranaRazdalja  $\leftarrow$  0
3:   while MobilnoVozlisce.Hitrost >
     TrenutnaDT.PorazdelitevHitrosti[IzbranaRazdalja] do
4:     IzbranaRazdalja  $\leftarrow$  IzbranaRazdalja + 1
5:   end while
6:   SeznamKandidatkST  $\leftarrow$  {}
7:   for all sidrna_tocka ST  $\in$  TrenutnaDT.SeznamVidnihST do
8:     if ST.RazdaljaDoMV = IzbranaRazdalja then
9:       ST.CasVidnosti  $\leftarrow$  1
10:      SeznamKandidatkST  $\leftarrow$  {SeznamKandidatkST, ST}
11:     end if
12:   end for
13:   for i = 1 to size of(MobilnoVozlisce.OpazovaneDT) do
14:     dostopovna_tocka DT  $\leftarrow$  MobilnoVozlisce.OpazovaneDT[i - 1]
15:     for all sidrna_tocka ST  $\in$  DT.SeznamVidnihST do
16:       if ST  $\in$  SeznamKandidatkST  $\wedge$  ST.CasVidnosti = i then
17:         ST.CasVidnosti  $\leftarrow$  ST.CasVidnosti + 1
18:       end if
19:     end for
20:   end for
21:   MaksPrisotnost  $\leftarrow$  0
22:   for all sidrna_tocka ST  $\in$  SeznamKandidatkST do
23:     if ST.CasVidnosti > MaksPrisotnost then
24:       MaksPrisotnost  $\leftarrow$  ST.CasVidnosti
25:     end if
26:   end for
27:   for all sidrna_tocka ST  $\in$  SeznamKandidatkST do
28:     if ST.CasVidnosti = MaksPrisotnost then
29:       NaslednjaST  $\leftarrow$  ST
30:       break
31:     end if
32:   end for
33:   return NaslednjaST
34: else
35:   return PrejsnjaST
36: end if

```

Algoritem 6 'prihodnost & najbližja'

Vhod: sidrna_točka *PrejsnjaST*, dostopovna_točka *TrenutnaDT*, mobilno_vozlisce *MobilnoVozlisce*

Izhod: sidrna_točka *NaslednjaST*

```

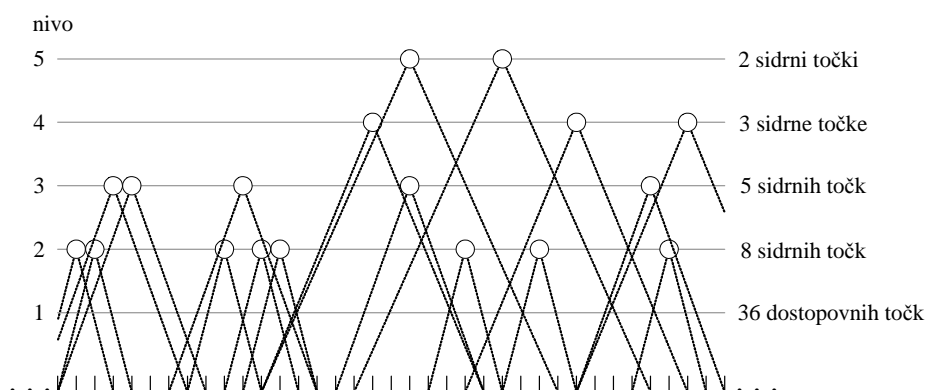
1: if PrejsnjaST  $\notin$  TrenutnaDT.SeznamVidnihST then
2:   SeznamKandidatkST  $\leftarrow$  {}
3:   for all sidrna_točka ST  $\in$  TrenutnaDT.SeznamVidnihST do
4:     ST.CasVidnosti  $\leftarrow$  1
5:     SeznamKandidatkST  $\leftarrow$  {SeznamKandidatkST, ST}
6:   end for
7:   for i = 1 to size of(MobilnoVozlisce.OpazovaneDT) do
8:     dostopovna_točka DT  $\leftarrow$  MobilnoVozlisce.OpazovaneDT[i - 1]
9:     for all sidrna_točka ST  $\in$  DT.SeznamVidnihST do
10:      if ST  $\in$  SeznamKandidatkST  $\wedge$  ST.CasVidnosti = i then
11:        ST.CasVidnosti  $\leftarrow$  ST.CasVidnosti + 1
12:      end if
13:    end for
14:  end for
15:  MaksPrisotnost  $\leftarrow$  0
16:  SeznamPrimernihST  $\leftarrow$  {}
17:  for all sidrna_točka ST  $\in$  SeznamKandidatkST do
18:    if ST.CasVidnosti  $\geq$  MobilnoVozlisce.ZahtevanaPrisotnost then
19:      SeznamPrimernihST  $\leftarrow$  {SeznamPrimernihST, ST}
20:    else if ST.CasVidnosti > MaksPrisotnost then
21:      MaksPrisotnost  $\leftarrow$  ST.CasVidnosti
22:    end if
23:  end for
24:  if SeznamPrimernihST = {} then
25:    for all sidrna_točka ST  $\in$  SeznamKandidatkST do
26:      if ST.CasVidnosti = MaksPrisotnost then
27:        SeznamPrimernihST  $\leftarrow$  {SeznamPrimernihST, ST}
28:      end if
29:    end for
30:  end if
31:  NaslednjaST  $\leftarrow$  SeznamPrimernihST[0]
32:  for all sidrna_točka ST  $\in$  SeznamPrimernihST do
33:    if ST.RazdaljaDoMV < NaslednjaST.RazdaljaDoMV then
34:      NaslednjaST  $\leftarrow$  ST
35:    else if ST.RazdaljaDoMV = NaslednjaST.RazdaljaDoMV  $\wedge$ 
      ST.CasVidnosti > NaslednjaST.CasVidnosti then
36:      NaslednjaST  $\leftarrow$  ST
37:    end if
38:  end for
39:  return NaslednjaST
40: else
41:   return PrejsnjaST
42: end if

```

3.3 Analiza na sintetičnih topologijah

3.3.1 Simulacijski model

Simulacijski model je bil implementiran v programskem jeziku C#. Uporabili smo sintetične topologije, podobne tistim, ki so jih predlagali v [55]. Čeprav se razvejanost širi od višjih nivojev k nižjim, topologije niso pravilna drevesa, območja sidrnih točk pa se naključno prekrivajo. V modelu je 100 sidrnih točk, ki so razporejene nad 233 dostopnimi točkami. Vsaka dostopna točka pokriva eno radijsko celico. Primer izreza 36 celic je prikazan na sliki 3.1. Topologijo sestavlja 5 hierarhičnih nivojev. Dostopne točke na najnižjem nivoju v sliki niso prikazane zaradi preglednosti. Sidrne točke so razporejene v 4 višje nivoje. Vsak višji nivo je od radijske celice bolj oddaljen v smislu usmerjevalnih hopov in posledično v smislu zakasnitve prenosa. Sidrne točke so naključno porazdeljene po vnaprej določeni verjetnosti porazdelitve. Na višjih nivojih je sidrnih točk manj, vendar pokrivajo večje število radijskih celic. Gledano po vrsti od najnižjega k najvišjemu nivoju, sidrne točke na nivojih 2, 3, 4 oziroma 5 pokrivajo 4, 8, 12 oziroma 16 radijskih celic.



Slika 3.1: Primer izreza 36 radijskih celic

Vsako mobilno vozlišče prične svojo pot v centru topologije. V simulacijskem času prepotujejo mobilna vozlišča med 0 in 100 celic, glede na izbrano konstantno hitrost. Porazdelitev hitrosti med mobilnimi vozlišči je enakomerna. Smer gibanja mobilnega vozlišča je izbrana za vsak premik med dvema radijskima celicama posebej. Verjetnost, da bo izbrana posamezna smer, je določena vnaprej. Topologija je dovolj velika, da je s stališča mobilnih vozlišč virtualno neomejena, kar pomeni, da se niti za najhitrejša

mobilna vozlišča, ki se premikajo venomer v eni smeri, končni pogoji ne razlikujejo od začetnih.

Za vsak testni algoritem in vsako verjetnost premikanja naprej (v desno smer), smo simulacijski tek ponovili 10000-krat. Za vsak tek je bila naključno generirana nova topologija in izbrana nova hitrost mobilnega vozlišča. Povprečna porazdelitev sidrnih točk s pripadajočimi standardnimi deviacijami v odvisnosti od razdalje od radijskih celic je podana v tabeli 3.2.

nivo	št. sidrnih točk
5	13.91 ± 0.03
4	19.69 ± 0.03
3	27.68 ± 0.03
2	38.71 ± 0.04

Tabela 3.2: Povprečna porazdelitev sidrnih točk

3.3.2 Evaluacijske metrike

Za oceno učinkovitosti algoritmov je pomembnih veliko komunikacijskih parametrov. Najbolj opazni so signalizacijska zakasnitev, procesorska moč in poraba pasovne širine. Vrednosti teh parametrov so močno odvisne od izbranih karakteristik simulacijske topologije, pomembnost enega parametra v primerjavi z drugim pa je nejasna. Zato je evaluacijski proces otežkočen.

V tej analizi uporabimo pristop, podoben tistemu v [53]. Opazovali smo dve lahko merljivi metriki, povprečno razdaljo od izbranih sidrnih točk in povprečno število menjav sidrnih točk v simulacijskem času. Prva je tesno povezana s signalizacijsko zakasnitvijo, optimalnostjo usmerjevalnih poti znotraj domen [69] in porazdelitvijo bremena, medtem ko povprečno število menjav sidrnih točk neposredno vpliva na količino režijske komunikacije in dodatnih signalizacijskih zakasnitev. Za doseganje čim boljše učinkovitosti algoritma, morata biti obe metriki tem nižji.

Tudi za ti dve metriki velja, da je relativna pomembnost ene v primerjavi z drugo nejasna in odvisna od preferenc posameznega mobilnega vozlišča. Vendar, če je vrednost ene od dveh metrik enaka za primerjana algoritma, potem primerjava vrednosti druge metrike nedvoumno pokaže, kateri algoritem je boljši. Enak zaključek velja, če sta vrednosti obeh metrik nižji pri enem od obeh primerjanih algoritmov. V obeh primerih trdimo, da sta algoritma *primerljiva*. Če pa je pri enem od obeh algoritmov vrednost ene

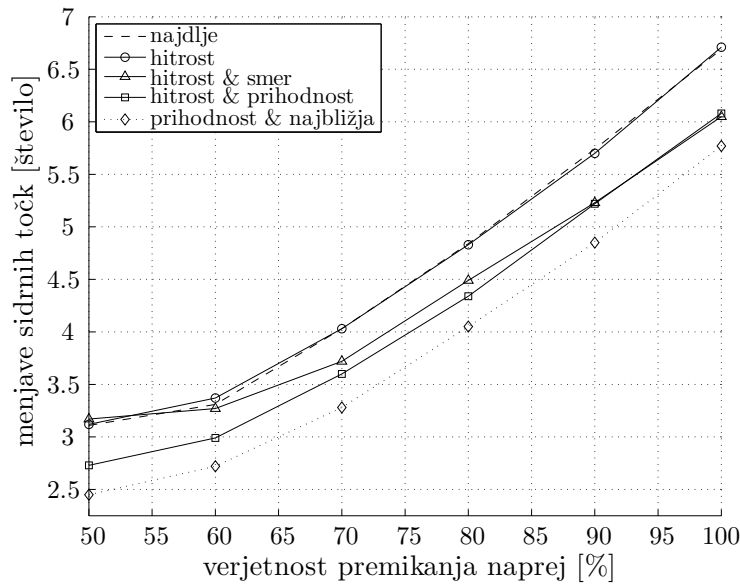
metrike nižja in hkrati vrednost druge višja, ni mogoče enostavno zaključiti, kateri algoritem je boljši. V tem primeru sta algoritma *neprimerljiva*.

3.3.3 Simulacijski rezultati za sintetične topologije

Pri vsakem od analiziranih algoritmov, razen pri *'najdlje'*, je relacijo med merjenima metrikama moč spremeniti s spreminjanjem parametrov algoritma. Tako na primer ena izbrana kombinacija parametrov vodi h krajšim razdaljam do izbranih sidrnih točk za ceno bolj pogostih menjav sidrnih točk, medtem ko druga kombinacija vodi k daljšim razdaljam pri manj pogostih menjavah. Analiza rezultatov je podana v dveh delih. V prvem delu smo parametre algoritmov izbirali tako, da so bili le-ti primerljivi z algoritmom *'najdlje'*. V drugem delu dajemo prednost algoritmom in parametrom, ki vodijo k bližje izbranim sidrnim točkam.

Da bi dosegli primerljivost med algoritmoma *'najdlje'* in *'hitrost'*, smo prilagodili porazdelitev hitrosti pri *'hitrost'* tako, da so mobilna vozlišča izbirala v povprečju bolj oddaljene sidrne točke. Verjetnost, da bo mobilno vozlišče izbralo bolj oddaljeno sidrno točko, smo dvigali toliko časa, dokler nismo dosegli, da je bilo število menjav sidrnih točk pri obeh algoritmih primerljivo. Kot je prikazano na sliki 3.2, se vrednosti zelo dobro ujemajo, ne glede na izbrano verjetnost premikanja naprej. Za iste parametre *'hitrost'* izbira sidrne točke, ki so v povprečju bližje, kot je prikazano na sliki 3.3. Iz tega sledi, da je *'hitrost'* boljši algoritem, kar potrjuje izsledke sorodnih raziskav.

Dalje smo študirali učinkovitost naših algoritmov *'hitrost & prihodnost'* in *'hitrost & smer'* glede na obstoječ algoritem *'hitrost'*, pri čemer smo obdržali isto porazdelitev hitrosti. Ker je porazdelitev hitrosti nespremenjena, so povprečne razdalje od izbranih sidrnih točk zelo podobne za vse tri algoritme, kot prikazuje slika 3.3. V nasprotju s povprečnimi oddaljenostmi, pa so povprečne vrednosti števila menjav sidrnih točk pri naših dveh algoritmih do 12,5% manjše. To pomeni, da je samo z upoštevanjem informacije o prihodnjih premikih mobilnih vozlišč izbira sidrnih točk izboljšana. Natančnejša analiza grafa na sliki 3.2 pokaže tudi povezavo med natančnostjo predvidevanja in zmožnostjo izboljšanja izbire sidrnih točk. Na skrajno levi strani x-osi je predvidevanje, da se bo mobilno vozlišče gibalo v smeri naprej, v polovici primerov pravilno. Implicitno to pomeni, da za algoritem *'hitrost & smer'* na tej točki ni nikakršne uporabne informacije o prihodnosti gibanja. Iz tega razloga je *'hitrost & smer'* pri 50% verjetnosti premikanja naprej enako učinkovit kot *'hitrost'*, ki predvidevanja sploh ne vsebuje.

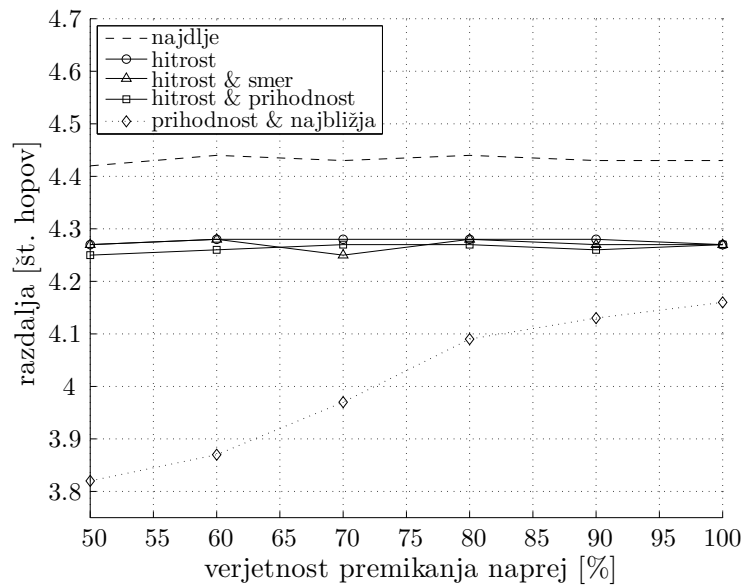


Slika 3.2: Povprečno število menjav (1)

Na nasprotni, desni strani x-osi, je predvidevanje, da se bo mobilno vozlišče premikalo naprej, vedno pravilno. V tem primeru je *'hitrost & smer'* enako učinkovit kot *'hitrost & prihodnost'*, ki razpolaga s popolno informacijo o gibanju mobilnega vozlišča v prihodnosti. Med obema ekstremoma je *'hitrost & smer'* slabši algoritem kot *'hitrost & prihodnost'* in boljši kot *'hitrost'*. Dejansko izboljšanje izbire sidrnih točk je torej neposredno odvisno od pravilnosti predvidevanja.

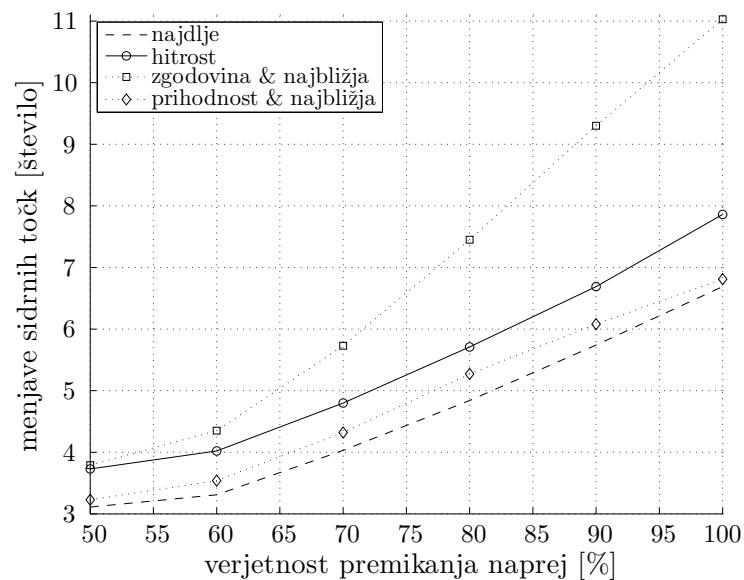
V prvem delu analize je bila izbrana neskončna dolžina selekcijskega intervala algoritma *'prihodnost & najbližja'*. Kot je razvidno iz grafov na slikah 3.2 in 3.3, je *'prihodnost & najbližja'* še učinkovitejši algoritem kot *'hitrost & prihodnost'*. Ker oba algoritma razpolagata s polno informacijo o prihodnjem gibanju mobilnega vozlišča, je učinkovitost prediktivnih algoritmov očitno odvisna tudi od načina uporabe te informacije. V tem primeru se izbiranje najbližjih sidrnih točk med tistimi, ki bodo prisotne v prihodnosti, izkaže za boljši način od tistega, ki temelji na hitrosti mobilnega vozlišča.

V drugem delu analize obravnavamo algoritme s pripadajočimi parametri, ki v povprečju izbirajo bližje sidrne točke in jih pogosteje menjavajo. Kot prikazujeta grafa na slikah 3.4 in 3.5, ti algoritmi niso več primerljivi z algoritmom *'najdlje'*, so pa primerljivi med seboj. Opazovani interval pri *'zgodovina & najbližja'* in selekcijski interval pri *'prihodnost & najbližja'*



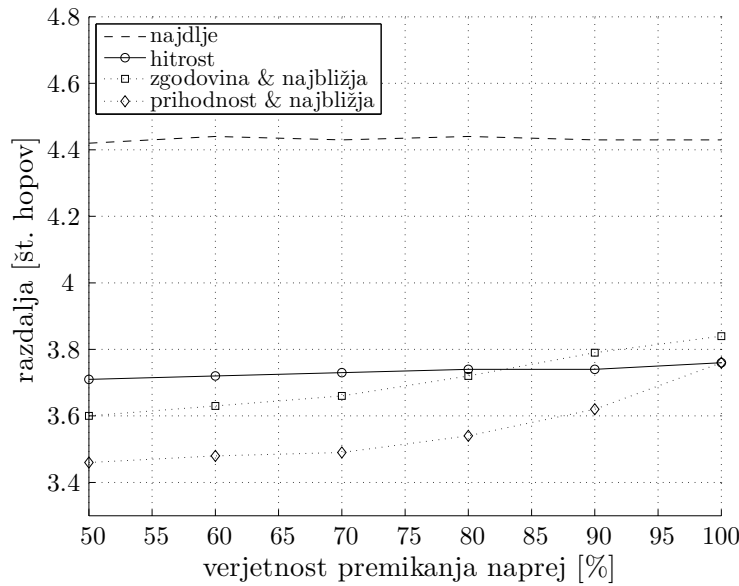
Slika 3.3: Povprečna razdalja (1)

sta bila oba nastavljeni na dolžino 4. Hitrostni intervali pri *'hitrost'* so bili glede na oddaljenost od sidrnih točk enakomerno porazdeljeni.



Slika 3.4: Povprečno število menjav (2)

Algoritem *'zgodovina & najbližja'* je učinkovitejši od algoritma *'hitrost'*,



Slika 3.5: Povprečna razdalja (2)

če se mobilno vozlišče z enako verjetnostjo premika naprej ali nazaj. Z naraščajočo verjetnostjo premikanja v eno smer, se število menjav sidrnih točk pri *'zgodovina & najbližja'* zelo hitro povečuje. Hitreje kot pri *'hitrost'*, pa se povečuje tudi povprečna oddaljenost od sidrnih točk. Pri 60% in 70% verjetnosti premikanja naprej, algoritma nista primerljiva. Pri 80%, kjer algoritma znova postaneta primerljiva, je algoritem *'hitrost'* že učinkovitejši od *'zgodovina & najbližja'*. Ko verjetnost premikanja naprej doseže 100%, postane algoritem *'hitrost'* izrazito učinkovitejši.

Razlog za takšne rezultate je v tem, da *'zgodovina & najbližja'* izbira le med tistimi sidrnimi točkami, ki so bile prisotne dovolj dolgo časa v preteklosti. To pravilo izloči nekatere sidrne točke, ki bodo prisotne dolgo časa v prihodnosti. Metoda se izkaže za učinkovito le v primeru, ko se vzorci gibanja ponavljajo pogostokrat znotraj iste poti.

Glede na sliki 3.4 in 3.5 je naš algoritem *'prihodnost & najbližja'* vedno učinkovitejši od obstoječih rešitev *'zgodovina & najbližja'* in *'hitrost'*. Natančna primerjava z algoritmom *'najdlje'* ni mogoča, vendar intuitivna ocena izrazito kaže na večjo učinkovitost algoritma *'prihodnost & najbližja'*, še posebej za 50% in 100% verjetnosti premikanja naprej.

Iz analize rezultatov, dobljenih s primerjavo algoritmov na sintetičnih topologijah, zaključujemo, da je algoritem *'prihodnost & najbližja'* najučinkovitejši.

3.4 Analiza na realističnih topologijah

Raziskovalci algoritmov za izbiranje sidrnih točk so do sedaj izvajali simulacije na regularnih sintetičnih topologijah, temelječih na preprosti hierarhiji, najpogosteje drevesni. Kljub temeljitemu pregledu v znanstveni literaturi nismo zasledili nikakršnega poskusa preučevanja tega problema z uporabo realističnih topologij. Verjamemo, da je z uporabo realističnih topologij moč doseči večjo zanesljivost in relevantno rezultatov. V nadaljevanju je predstavljen prvi takšen poskus [70].

3.4.1 Označevanje in uporaba realističnih topologij

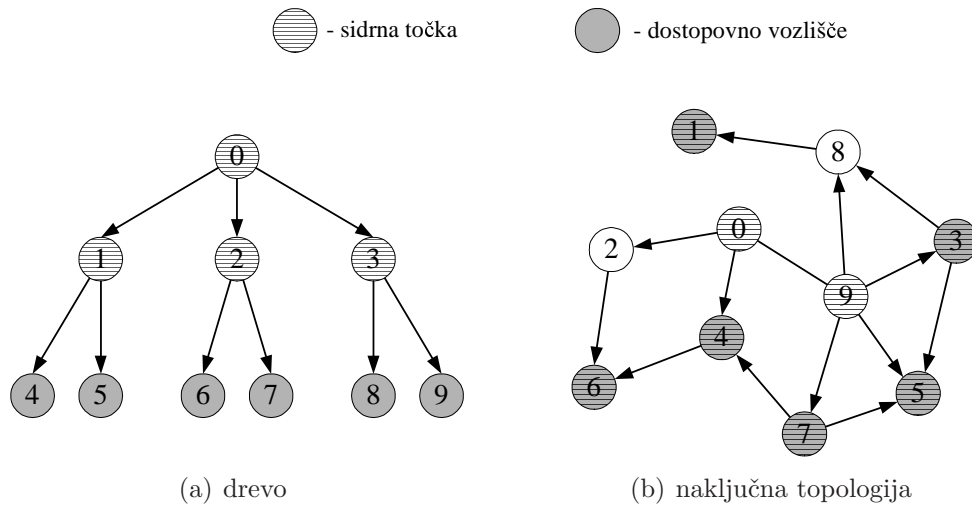
Topologije, primerne za analizo upravljanja mobilnosti, morajo izpolnjevati tri zahteve:

- (a) razlikovanje med dostopovnimi vozlišči in ne-dostopovnimi vozlišči,
- (b) razlikovanje med vozlišči s sidrnimi točkami in vozlišči brez sidrnih točk,
- (c) določenost območij pokritij sidrnih točk.

Topologija na sliki 3.6(a) predstavlja tipično drevesno strukturo, ki jo uporabljajo raziskovalci s tega področja. Dostopovna vozlišča, vozlišča s sidrnimi točkami in območja pokritij sidrnih točk, so določena z upoštevanjem enostavne nivojske hierarhije. Sidrne točke so postavljene na višjih nivojih, dostopovne točke pa na najnižjem nivoju. Puščice kažejo v smeri razširjanja signalizacijskih sporočil sidrnih točk. V realističnih topologijah hierarhija ni natančno določena. Posledično je potreben bolj sistematičen pristop k označevanju topologij.

Struktura resničnega interneta se kaže na dveh osnovnih nivojih, nivoju usmerjevalnikov in nivoju avtonomnih sistemov, kjer predstavlja avtonomni sistem skupek omrežij in usmerjevalnikov s skupnimi pravili usmerjanja. Hierarhični vzorci so bolje raziskani na nivoju avtonomnih sistemov, kar je eden od razlogov, zakaj smo za nadaljnjo analizo izbrali prav ta nivo.

Da bi zadostili zahtevama (a) in (b), smo uporabili hierarhično dekompozicijo avtonomnih sistemov v 5 hierarhičnih nivojev, kot je to predlagano v [71]. Najnižje v hierarhiji so *stranke*, ki ne prenašajo tranzitnega prometa. Stranke skupaj z *majhnimi ponudniki* internetnega dostopa predstavljajo rob interneta, kjer se pričakuje, da bodo končni uporabniki dostopali do omrežja. Stranke in majhne ponudnike zato določimo za dostopovna



Slika 3.6: Primera topologij z 10 vozlišči

vozlišča. Preostali avtonomni sistemi predstavljajo jedro interneta in so razporejeni v tri nivoje, *gosto jedro*, *tranzitno jedro* in *zunanje jedro*. Največ tranzitnega prometa se prenaša preko najvišjega nivoja, to je gostega jedra. V nasprotju z zagotavljanjem brezžičnega dostopa, bi storitve sidrnih točk lahko bile prisotne tudi v jedru, saj so infrastrukturne zahteve relativno majhne, območje pokritja pa zelo široko. V nadaljnji analizi preizkusimo dva scenarija. V prvem so sidrne točke prisotne samo na robu interneta, to je v strankah in majhnih ponudnikih, v drugem pa so prisotne tudi v jedru.

Na sliki 3.6(b) je primer majhnega omrežja avtonomnih sistemov. Vsako vozlišče predstavlja en avtonomni sistem. Vozlišča lahko zavzamejo tudi obe vlogi hkrati, kot dostopno vozlišče in kot vozlišče s sidrno točko (vozlišča 1, 3, 4, 5, 6 in 7) ali pa ostanejo brez vlog (vozlišči 2 in 8).

Za zadovoljitev zahteve (c) je potrebno označiti povezave v topologiji. Med pari avtonomnih sistemov v resničnem internetu obstajajo različne poslovne relacije, med katerimi so najpogostejše ponudnik-stranka in enakovreden-z-enakovrednim [72, 73]. Prve označujemo kot usmerjene, druge pa kot neusmerjene povezave. Ponudniki so ponavadi večji in za plačilo ponujajo dostop do preostalega interneta strankam. Tudi storitve sidrnih točk bodo verjetno ponujane v isti smeri, od ponudnika k strankam. Povezave so zatoj usmerjene k strankam. Signalizacijska sporočila sidrnih točk se razširjajo v smeri povezav in lahko prečkajo poljubno število vozlišč. Dovolimo tudi razširjanje preko povezav enakovreden-z-enakovrednim, vendar le, če povezava predstavlja prvi hop na propagacijski poti.

Propagacijo signalizacijskih sporočil sidrnih točk ilustriramo z naslednjimi primeri na sliki 3.6(b). Sidrno točko v vozlišču 7 oglašujejo dostopovna vozlišča 6, 4, 5. Oglaševana je tudi v lastnem vozlišču 7, saj je to tudi dostopovno vozlišče. Sidrna točka v vozlišču 3 je vidna v dostopovnih vozliščih 1, 3 in 5. Signalizacijska sporočila so preko vozlišča 8 posredovana vozlišču 1, v vozlišču 8 pa ne pride do oglaševanja, ker le-to ni dostopovno vozlišče. Sidrna točka v vozlišču 5 je oglaševana le v tem lastnem vozlišču, sidrni točki v vozliščih 0 in 9 pa sta oglaševani v vseh dostopovnih vozliščih v topologiji.

3.4.2 Simulacijski model in metrike

V tem delu analize primerjamo dva algoritma, *'prihodnost & najbližji'*, ki je najučinkovitejši v testnih sintetičnih omrežjih in *'najdlje'*, kot bazično referenco.

Za simulacije uporabljamo merjene topologije interneta z znanimi poslovnimi relacijami, ki jih zagotavlja organizacija CAIDA (Cooperative Association for Internet Data Analysis) [74]. Raziskovalci te organizacije povzemajo podatke o obstoječih povezavah med avtonomnimi sistemi s posnetkov na strežnikih RouteViews [75], ki so narejeni vsakih 8 ur preko 5-dnevnih period. Tipe relacij nato prepoznavajo z algoritmom, opisanim v [73]. Vsak teden je na voljo nova, na tak način dobljena, internetna topologija. Na izbranih topologijah nato izvedemo še zgoraj opisani postopek označevanja. Kot rezultat dobimo merjeno internetno topologijo, ki zadošča vsem trem zahtevam, (a), (b) in (c).

Za vsak set parametrov smo simulacijski tek ponovili 10000-krat. Pri vsakem teku je mobilno vozlišče zamenjalo 500 dostopovnih vozlišč. Način, kako mobilno vozlišče izbere naslednje dostopovno vozlišče, smo zasnovali na predpostavki, da je izbor topološko bližjih dostopovnih vozlišč bolj verjeten od tistih, ki so topološko dlje. Privzeli smo, da je verjetnostni faktor premikanja enak 3, kar pomeni, da so dostopovna vozlišča, ki so en hop bližje, izbrana s trikrat večjo verjetnostjo.

Opazovali smo povprečno razdaljo med dostopovnim vozliščem in vozliščem s sidrno točko ("razdalja") ter povprečno število zamenjanih dostopovnih točk na eno menjavo sidrne točke ("menjave"). Slednja metrika je tesno povezana s povprečnim številom menjav sidrnih točk, omogoča pa tudi dodatno analizo. Če je ta metrika mnogo višja od 1, je ponujanje storitev sidrnih točk preko večih avtonomnih sistemov dobro upravičeno. Višja kot je metrika, bolj učinkovit je algoritem.

3.4.3 Simulacijski rezultati za realistične topologije

Za simulacije smo uporabili topologije, ki jih je CAIDA pridobila v različnih časovnih obdobjih v letih 2004-2007. Razlike med posameznimi časovnimi obdobji so minimalne. V nadaljevanju so predstavljeni rezultati za topologijo iz septembra 2007.

V tabeli 3.3 so prikazani rezultati za scenarij, v katerem so sidrne točke prisotne samo v strankah in majhnih ponudnikih. Ne glede na verjetnostni faktor premikanja, je vrednost metrike “menjave” praktično enaka za oba algoritma, medtem ko je “razdalja” malo manjša pri uporabi algoritma *'prihodnost & najbližji'*. V tem scenariju bazični algoritem *'najdlje'* očitno zadostuje, saj niti *'prihodnost & najbližji'* ne kaže pomembnega izboljšanja. Ne glede na to je ponujanje storitev sidrnih točk iz oddaljenih avtonomnih sistemov dobro upravičeno, še posebej za izrazito lokalizirano gibanje mobilnega vozlišča s faktorjem premikanja 10 ali več.

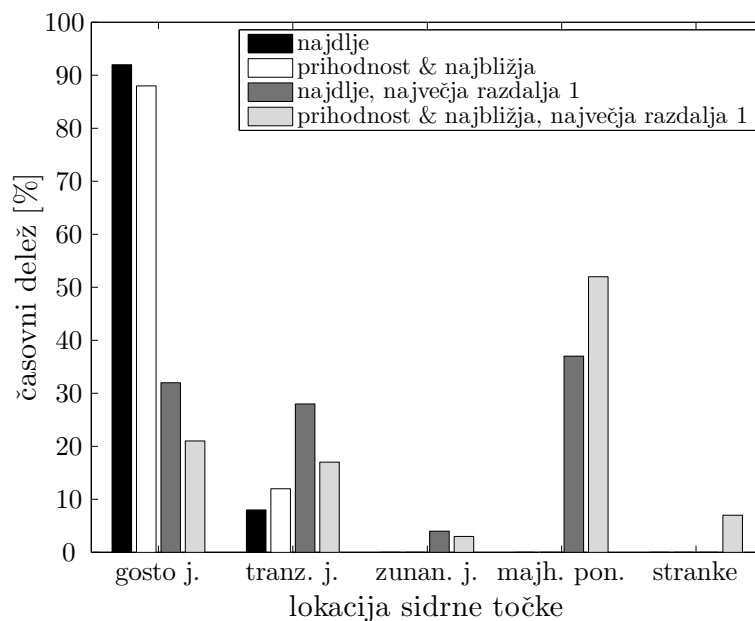
faktor premikanja	metrika	<i>'najdlje'</i>	<i>'prihodnost & najbližja'</i>
3	razdalja	0,65	0,47
	menjave	1,83	1,85
10	razdalja	0,56	0,50
	menjave	3,43	3,44

Tabela 3.3: Primerjava algoritmov (sidrne točke so na robu interneta)

Rezultati v tabeli 3.4 so veljavni za scenarij, kjer so storitve sidrnih točk ponujane tudi iz jedra. V tem primeru postane razlika v učinkovitosti med obema algoritmoma očitna. Algoritem *'prihodnost & najbližji'* je veliko učinkovitejši, še posebej pri metriki “menjave”. Vendar pa so večino časa izbrane sidrne točke, ki so v gostem jedru. S slike 3.7 lahko razberemo, da je to približno 90% časa gibanja mobilnega vozlišča. To se ne zdi zelo realistično, ker ni verjetno, da bodo avtonomni sistemi v gostem jedru dejansko ponujali storitve sidrnih točk v taki meri. Če omejimo razdaljo med dostopovnim vozliščem in vozliščem s sidrno točko na največ 2 hopa, upade časovni delež na 50%, medtem ko omejitev razdalje na največ 1 hop vodi do 30% časovnega deleža ali manj (slika 3.7). Tudi s to strogo omejitvijo na največ 1 hop, ki vodi do bolj realističnih rezultatov, ostane razlika med algoritmoma relevantna, uporaba sidrnih točk izven meja lastnih avtonomnih sistemov pa upravičena.

največja razdalja	metrika	'najdlje'	'prihodnost & najbližja'
∞	razdalja	2,56	2,41
	menjave	40,52	58,00
2	razdalja	1,73	1,49
	menjave	4,56	7,54
1	razdalja	0,87	0,68
	menjave	1,87	2,66

Tabela 3.4: Primerjava algoritmov (sidrne točke so na vseh nivojih)



Slika 3.7: Časovni delež izbranih sidrnih točk glede na hierarhični nivo

Poglavje 4

Modeliranje realnih omrežij

Analiza algoritmov za izbiranje sidrnih točk na realističnih omrežjih iz prejšnjega poglavja je omejena na merjene internetne topologije. Če želimo analizo poglobiti, moramo nabor simulacijskih topologij razširiti, na primer z uporabo topoloških generatorjev, ki imitirajo strukturne lastnosti realnega interneta. Poudarek današnjih generatorjev je predvsem na topološki povezanosti vozlišč omrežja. Za verodostojnost modela pa se morajo ohranjati tudi hierarhični vzorci, izraženi s poslovnimi relacijami med avtonomnimi sistemi.

V tem poglavju predstavimo lastno metodo, po kateri topologijam, ki so brez informacij o poslovnih relacijah, le-te pripišemo tako, da so vzorci njihovega pojavljanja podobni tistim, ki jih najdemo v resničnem internetu. Gre za prispevek, ki ni omejen samo na upravljanje mobilnosti, zato ga predstavimo celovito, s pripadajočo uvrstitvijo v raziskovalno področje.

Poglavje začnemo z natančnejšim opisom poslovnih relacij in njihovega pomena. Nadaljujemo s pregledom metod za prepoznavanje relacij v resničnem internetu in metod za generiranje realističnih topologij. Nato definiramo problem in opišemo njegovo rešitev. V zadnjem delu preverimo še kakovost metode in jo preizkusimo na primerih sintetičnih omrežij.

4.1 Pomen poznavanja poslovnih relacij

4.1.1 Tipi poslovnih relacij

Pionirsko delo na področju analize poslovnih relacij je bilo objavljeno leta 2001 [72]. Delo prvič definira tri različne tipe relacij: ponudnik–stranka (provider–to–customer), enakovreden–z–enakovrednim (peer–to–peer) in soroden–s–sorodnim (sibling–to–sibling).

Relacija ponudnik–stranka najpogosteje obstaja v smeri od večjega avtonomnega sistema proti manjšemu, pri čemer stranka plačuje ponudniku dostop do globalnega interneta. Ponudnik stranki prek protokola BGP izvaža celotno usmerjevalno tabelo, torej svoje lokalne zapise, zapise lastnih ponudnikov, strank, enakovrednih in sorodnih avtonomnih sistemov. Nasprotno, stranka ponudniku posreduje le lokalne zapise, zapise lastnih strank in sorodnih sosedov. Zapisov drugih ponudnikov in enakovrednih avtonomnih sistemov stranka ponudniku ne izvaža. Enake omejitve veljajo med enakovrednimi avtonomnimi sistemi. V tem primeru imata avtonomna sistema sklenjen vzajemen sporazum o medsebojni izmenjavi njunega prometa ter prometa njunih strank in sorodnih sistemov. Relacija soroden–s–sorodnim je ponavadi vzpostavljena med avtonomnima sistemoma, ki pripadata sorodni ali kar isti organizaciji. Sorodna avtonomna sistema brez omejitev izmenjujeta tako promet kakor tudi zapise usmerjevalnih tabel. Avtonomni sistem lahko pri sklepanju relacij z drugimi avtonomnimi sistemi nastopa v vseh naštetih vlogah hkrati.

Relacije soroden–s–sorodnim so redke, ponavadi njihov delež znaša le približno 1% vseh relacij v internetu [71, 72]. Poleg opisanih obstajajo tudi druge, kompleksnejše poslovne relacije, vendar v internetu nastopajo še redkeje [71, 73].

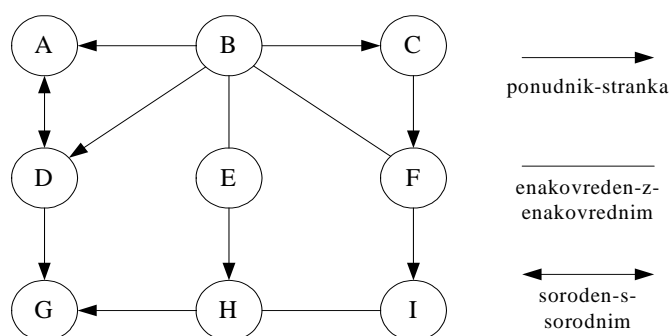
4.1.2 Usmerjevalne poti brez dolin

Posledica doslednega upoštevanja zgoraj opisanih izvoznih politik je, da je potek usmerjevalnih poti podrejen določeni strukturi. Podatkovni paket ne more prečkati povezave stranka–ponudnik ali enakovreden–z–enakovrednim po prečkanju povezave ponudnik–stranka ali enakovreden–z–enakovrednim. Obstoj strukture je dokazala Gao [72] in poimenovala tovrstne poti kot usmerjevalne poti *brez dolin*. V literaturi najdemo tudi termin *veljavne usmerjevalne poti* [71, 73, 76].

Usmerjevalna pot brez dolin je torej sestavljena iz treh segmentov. Se-

gment navkreber sestoji iz nič ali več povezav tipa stranka-ponudnik ali soroden-s-sorodnim. Sledi segment z največ eno povezavo enakovreden-z-enakovrednim. Zadnji segment je segment navzdol in vsebuje nič ali več povezav tipa ponudnik-stranka ali soroden-s-sorodnim.

Na sliki 4.1 je prikazan primer omrežja avtonomnih sistemov s pripadajočimi poslovnimi relacijami. Usmerjevalne poti brez dolin v tem omrežju so na primer (D, B, E, H), (B, C, F, I) in (C, B, A, D), medtem ko usmerjevalne poti (D, G, H), (F, B, E, H) in (F, I, H) niso brez dolin.



Slika 4.1: Primer omrežja avtonomnih sistemov

4.1.3 Posledice neupoštevanja relacij

Komunikacijski promet poteka samo po usmerjevalnih poteh brez dolin, kar omejuje možnosti usmerjanja. Poleg tega imajo avtonomni sistemi vedno možnost izbire, po kateri izmed razpoložljivih usmerjevalnih poti brez dolin bodo pošiljali promet. Tipično ima prednost usmerjanje prek strank [77]. Posledično so uporabljene usmerjevalne poti daljše od topološko najkrajših. Pojav imenujemo napihovanje poti (path inflation) [78]. V [77] so pokazali, da je zaradi napihovanja poti najmanj 45% usmerjevalnih poti daljših vsaj za eno povezavo in da podaljševanje lahko prinese do 9 dodatnih povezav.

V primeru na sliki 1 je med vozliščema D in H topološko najkrajša usmerjevalna pot (D, G, H), vendar ni brez dolin, ker segmentu navzdol (D, G) sledi segment navkreber (G, H). Najkrajša usmerjevalna pot brez dolin med D in H je (D, B, E, H). Med vozliščema B in I je topološko najkrajša usmerjevalna pot (B, F, I) sicer brez dolin, vendar avtonomni sistem B izbere usmerjanje prek stranke C namesto prek enakovrednega avtonomnega sistema F, ker tako narekuje njegova politika usmerjanja. Posledično je uporabljena pot (B, C, F, I).

Simulacije na modelu omrežja, ki ne vsebuje informacij o poslovnih relacijah, zanemarijo efekt napihovanja poti. Simulirane usmerjevalne poti so na splošno prekratke. Neupoštevanje tipov relacij vodi tudi k prevelikemu številu alternativnih poti in k manjšim prometnim obremenitvam na določenih povezavah [79]. Zato so rezultati simulacij izpostavljeni napakam.

V nekaterih primerih komunikacijskih scenarijev pa upoštevanje tipov relacij predstavlja celo pogoj za izvedbo simulacije. Pri analizi upravljanja mobilnosti brez poznavanja tipov relacij ne moremo sklepati o vidnosti sidrnih točk v sosednjih avtonomnih sistemih. Območja pokritij sidrnih točk glede na poslovne relacije določamo po metodologiji, opisani v poglavju 3.4.1.

V primeru na sliki 4.1 je sidrna točka v avtonomnem sistemu B vidna mobilnim napravam v avtonomnih sistemih B, A, D, G, C, F in I, medtem ko je sidrna točka v avtonomnem sistemu F vidna samo mobilnim napravam v avtonomnih sistemih F in I.

4.2 Prepoznavanje tipov poslovnih relacij

Upravitelji avtonomnih sistemov imajo na splošno podatke o poslovnih relacijah za zaupne. Zato je prepoznavanje tipov relacij v resničnem internetu velik izziv. Predlaganih je bilo več metod.

Prvo metodo je predlagala avtorica tipizacije relacij v [72]. Informacije o usmerjevalnih poteh črpa iz javno dostopnih usmerjevalnih tabel na strežnikih RouteViews [75]. Metoda temelji na predpostavki, da je na meji med segmentom navkreber in segmentom navzdol najverjetneje tak avtonomni sistem, ki ima v grafu omrežja najvišjo stopnjo med vozlišči na dani poti. Avtonomni sistemi z višjo stopnjo so ponavadi večji, zato pogosteje nastopajo v vlogi ponudnika storitev manjšim avtonomnim sistemom. Avtorica tudi ugotavlja, da relacija soroden–s–sorodnim obstaja med sistemoma, ki drug drugemu zagotavljata prehod tranzitnega prometa, ter da relacijo enakovreden–z–enakovrednim ponavadi vzpostavijo avtonomni sistemi podobnih dimenzij. Ugotovitve upošteva predlagana hevrstika, ki iz dejanskih usmerjevalnih tabel določi relacije med realnimi avtonomnimi sistemi.

Naslednje pomembno delo [71] formalizira problem prepoznavanja poslovnih relacij kot optimizacijski problem v teoriji grafov. Rešitev problema ToR (Type-of-Relationship) pri danem grafu $G = (V, E)$ in naboru usmerjevalnih poti P je tak nabor relacij za povezave iz E , da je število usmerjevalnih poti brez dolin v P maksimalno. Avtorji so verjeli, da gre za NP-poln

problem, vendar jim tega ni uspelo dokazati. Novost njihovega pristopa k reševanju problema je v uporabi več javno dostopnih BGP usmerjevalnih tabel. V nasprotju z [72] metoda ne uporablja informacije o stopnji avtonomnega sistema, temveč iz povprečnega položaja danega avtonomnega sistema v celotnem naboru usmerjevalnih poti sklepa o tem, kako blizu jedra interneta se sistem nahaja. Primerjava položajev dveh sosednjih avtonomnih sistemov je osnova predlagane hevrstike za prepoznavanje tipa relacije med njima. Hevrstika loči le med relacijami ponudnik–stranka in enakovreden–z–enakovrednim.

NP-polnost problema ToR je pokazana v [76] in [80]. Avtorji neodvisno pridejo do sklepa, da rešitev problema ne določa natančnega položaja relacij enakovreden–z–enakovrednim. Predlagane hevrstike so matematično najbolj eksaktne, dosegajo največje deleže usmerjevalnih poti brez dolin, a so omejene le na relacije tipa ponudnik–stranka. Vprašanje ujemanja relacij z dejanskimi ostane odprto.

V [81] zgradijo delno rešitev problema z zbiranjem podatkov iz podatkovnih zbirk “*BGP community*” in “*Internet Routing Registers*” [82], v katerih nekateri avtonomni sistemi javno objavijo podatke o poslovnih relacijah. Z upoštevanjem pravil, izpeljanih iz lastnosti usmerjevalnih poti brez dolin, je mogoče določiti tudi relacije nekaterim sosednjim povezavam. Večkratna ponovitev postopka določi velik delež vseh relacij v omrežju. Avtorji za povezave, katerih algoritem ne zajame, uporabijo hevrstiko iz [72]. Analiza pokaže, da tak pristop izboljša predvsem natančnost prepoznavanja relacij tipa enakovreden–z–enakovrednim.

Članek [83] kritično analizira pravilnost pristopa k prepoznavanju relacij z reševanjem problema ToR. Avtorji navajajo, da lahko v nekaterih primerih pripišemo povezavam poljuben tip relacije, ne da bi pri tem spremenili število usmerjevalnih poti brez dolin. Ugotovijo tudi, da ignoriranje relacij tipa soroden–s–sorodnim vodi ne samo do zamegljene slike o realni strukturi interneta, temveč tudi do napačno prepoznanih relacij tipa ponudnik–stranka. S tem pokažejo, da zgolj hevrstično izboljševanje rešitve problema ToR ne vodi k večji verodostojnosti prepoznanih relacij. Predlagajo razširitev problema z dodatnim optimizacijskim kriterijem, ki temelji na gradientu stopenj. Gradient stopenj za izbrano smer opazovane usmerjevalne poti pove, ali stopnja sosednjih avtonomnih sistemov narašča ali pada. Z upoštevanjem na novo vpeljanega kriterija postane avtonomni sistem z višjo stopnjo z večjo verjetnostjo ponudnik avtonomnemu sistemu z nižjo stopnjo. Ideja o gradientu stopenj je podobna filozofiji prve objavljene hevrstike [72], vendar je to pot matematično izpopolnjena in nastopa v kombinaciji z reševanjem

problema ToR.

V zadnjem pomembnejšem delu [73] avtorji poudarijo pomanjkljivost algoritmov za reševanje problema ToR, da nekatere poznane usmerjevalne poti nepravilno razglasijo kot poti brez dolin in posledično vsaj eno relacijo na poti določijo napačno. Predlagana heuristika upošteva nove ugotovitve in ugotovitve predhodnikov ter določa vse tri tipe relacij. Relacije ponudnik–stranka in enakovreden–z–enakovrednim določa upoštevajoč gradient stopenj in maksimalno število usmerjevalnih poti brez dolin. Metoda presenetljivo dosega najboljše rezultate tedaj, ko ima gradient stopenj bistveno večjo težo kot maksimizacija števila usmerjevalnih poti brez dolin. Najugodnejše razmerje dosega vrednost 100:1. Za prepoznavanje relacij soroden–s–sorodnim, se predlagani algoritem opira na podatkovne zbirke “*Internet Routing Registers*”. Pri evalvaciji svoje heuristike so se avtorji obrnili na upravitelje avtonomnih sistemov. Rezultati potrjujejo veliko natančnost prepoznanih relacij, ki dosega 94,2%.

Pomemben prispevek avtorjev zadnje heuristike je tudi tedensko osvežena javno dostopna zbirka avtonomnih sistemov s pripadajočimi relacijami, kot jih določa njihov algoritem [74].

4.3 Generiranje omrežij avtonomnih sistemov

4.3.1 Osnovni topološki generatorji

Področje generiranja naključnih internetnih topologij je dokaj dobro raziskano, kar velja tako za topologije na ravni usmerjevalnikov, kakor tudi za topologije povezovanja med avtonomnimi sistemi.

Začetki segajo v leto 1988, ko je Waxman predlagal prvi generator, s katerim je poskušal imitirati povezovalne vzorce v takratnem internetu [84]. Osnova generatorja je naključni proces, odvisen le od evklidske razdalje med naključno porazdeljenimi vozlišči na ravnini. Za reprodukcijo čedalje bolj očitnih hierarhičnih vzorcev v internetu to ni zadostovalo, zato so v naslednji fazi nastali tako imenovani strukturni generatorji [85, 86]. Striktno nivojska hierarhija teh generatorjev je odvisna od številnih vhodnih parametrov. Leta 1999 se je razumevanje strukture internetnih topologij korenito spremenilo z ugotovitvami bratov Faloutsos [87]. Na podlagi obsežnejših meritev so ugotovili, da se porazdelitev stopenj tako avtonomnih sistemov, kakor tudi posameznih usmerjevalnikov, podreja potenčnemu zakonu. Te lastnosti strukturni generatorji ne upoštevajo, zato je nastala povsem nova skupina tako imenovanih stopenjskih generatorjev [88–92]. Primarni cilj teh

generatorjev je avtentična reprodukcija porazdelitve stopenj, medtem ko hierarhično strukturo omrežne topologije povsem ignorirajo. Kljub temu se pozneje izkaže, da stopenjski generatorji prekašajo strukturne generatorje tudi v tem pogledu [93].

Po zadnjih izsledkih je natančnost generiranih modelov mogoče še nadalje izboljšati z upoštevanjem tako imenovanih dK -serij verjetnostnih porazdelitev [94]. Gre za formalizem, po katerem se s korelacijami stopenj opiše, kako so v modeliranem grafu med seboj povezani podgrafi poljubnih velikosti d . Z višanjem velikosti d se doseže večjo natančnost, vendar za ceno višje kompleksnosti. V limiti d postane enak velikosti modeliranega grafa, s čimer model postane identičen originalu. Avtorji so pokazali praktično uporabnost tega pristopa za $d = 2$ in $d = 3$. Pri $d = 1$ generator postane klasičen stopenjski generator, ki upošteva le porazdelitev stopenj posameznih vozlišč. Isti avtorji so predlagali tudi razširitev, ki omogoča, da se modelirani in izvorni graf razlikujeta po velikosti [95].

4.3.2 Relacijski generatorji

Kljub širokemu naboru metod za prepoznavanje poslovnih relacij v resničnem internetu in velikemu številu predlaganih osnovnih topoloških generatorjev, ni bilo veliko poskusov prepoznano strukturo relacij vključiti v generične modele topologij. V nadaljevanju opišemo tri primere.

V [96] so ubrali evlucijski pristop. V članku raziščejo različne kriterije, ki vplivajo na odločitve avtonomnih sistemov, katere poslovne relacije je potrebno vzpostaviti na novo in katere obnoviti, tako da bo vrednost avtonomnega sistema čim višja. Njihov model omogoča študijo preteklih vplivov, ki vodijo do trenutnega stanja in napovedovanje, kako se bo topologija razvijala v prihodnje.

Naslednji predlog temelji na filozofiji stopenjskih generatorjev z razširitvijo stopenjske porazdelitvene funkcije na tipe relacij [79]. Razširjena porazdelitvena funkcija pove, koliko avtonomnih sistemov v omrežju ima določeno kombinacijo števila ponudnikov, strank in enakovrednih avtonomnih sistemov. Pri realizaciji algoritma so avtorji izhajali iz metode, uporabljene pri osnovnem stopenjskem generatorju PLRG [91]. Postopek generacije naključnega omrežja poteka v dveh korakih. V prvem se generira N trojic števil za N zelenih avtonomnih sistemov. Posamezna trojka pomeni število relacij ponudnik–stranka, stranka–ponudnik in enakovreden–enakovrednim. V drugem koraku generator naključno poveže vozlišča tako, da zadovolji zahtevanemu številu relacij. Presežek relacij določenega tipa, ki nimajo

ustreznega para, generator zanemari. Nazadnje odstrani še topološke nepravilnosti, kot so zanke in podvojene povezave med vozlišči.

Tudi zadnji primer predstavlja razširitev osnovnega na relacijski generator [97]. Osnovan je na dK -serijah verjetnostnih porazdelitev [94]. Tako kot izvorni, tudi razširjeni formalizem dopušča poljubno velikost podgrafov d . Kljub temu avtorji izvedejo generator za največ $d = 2$. Pri $d = 1$ je generator identičen predhodniku [79], že pri $d = 2$ pa so zaradi širokega nabora neodvisnih spremenljivk porazdelitvene funkcije zelo redke. Zato pri praktični izvedbi avtorji uvedejo tako imenovane povzetke porazdelitvenih funkcij, ki združujejo podatke več neodvisnih spremenljivk. S tem je natančnost manjša kot jo predvideva formalizem, izvedba pa lažja.

Pri vseh naštetih relacijskih generatorjih je modeliranje tipov relacij sestavni del procesa generiranja topologije. Če pri raziskavah osnovnih stopenjskih generatorjev pride do novih izsledkov, tudi iz njih izvedeni relacijski generatorji postanejo zastareli. Poleg tega je celotna metoda kompleksnejša, še posebej pri zadnjem predlogu [97], kjer je rešitev zastavljena tako široko, da jo je moč uporabiti tudi za druge tipe omrežij, na primer sociološke ali biološke.

4.4 Problem določanja relacij v naključnih grafih

V nadaljevanju predlagamo ločitev modeliranja relacij med avtonomnimi sistemi od procesa naključnega generiranja omrežnih topologij. Menimo, da sta problema v veliki meri neodvisna in bolje rešljiva vsak zase. Razvoj osnovnih topoloških generatorjev, ki verodostojno reproducirajo bistvene lastnosti interneta, se tako lahko še naprej odvija ločeno.

4.4.1 Definicija problema

Topologija omrežja avtonomnih sistemov je osnovni vhod algoritma, ki rešuje v nadaljevanju definirani problem. Cilj je določiti tip relacije vsem povezavam v grafu na način, ki zajame osnovne lastnosti komunikacij med avtonomnimi sistemi. Rezultat je označen graf.

Identificirali smo pet potrebnih lastnosti označenega grafa. Če graf vsebuje te lastnosti, trdimo, da reproducira bistvene strukturne karakteristike realnega interneta. Zahtevane lastnosti formalno zapišemo v obliki definicije

problema, ki ga imenujemo problem TRR (Type-of-Relationship problem in Random graphs).

Definicija 1 (problem TRR). *Pri danem neusmerjenem grafu $G = (V, E)$ z množico vozlišč V in množico povezav E poišči označen graf, tako da:*

1. *med vsakim parom vozlišč iz množice V obstaja vsaj ena pot brez dolin (lastnost povezanosti vozlišč);*
2. *označeni graf odraža hierarhično strukturo, primerljivo s hierarhično strukturo v merjenih grafih (lastnost hierarhične strukture);*
3. *v jedru označenega grafa obstoji klika*, ki vsebuje le povezave tipa enakovreden–z–enakovrednim (lastnost klike jedra);*
4. *usmerjenost povezav, ki pomenijo relacijo ponudnik–stranka, sledi padajočemu gradientu stopnje, pri čemer delež izjem ne presega vrednosti 0,5% (lastnost gradienta stopnje);*
5. *označeni graf ne vsebuje ciklov[†], ki bi vsebovali urejeno zaporedje relacij ponudnik–stranka (lastnost odsotnosti ciklov ponudnik–stranka).*

4.4.2 Utemeljitev zahtev

Izpolnjevanje lastnosti povezanosti vozlišč zagotovi zmožnost komunikacije med poljubnima avtonomnima sistemoma v omrežju. Če med dvema avtonomnima sistemoma v omrežju ni usmerjevalnih poti brez dolin in se pravila izvažanja usmerjevalnih tabel dosledno upoštevajo, avtonomna sistema med seboj ne moreta komunicirati. Gre za nesprejemljiv pojav brez realne podlage, zato je povezanost vozlišč najpomembnejša lastnost označenega grafa.

Obstoj hierarhične strukture v internetu je lastnost, ki je v raziskovalnih krogih široko sprejeta. Kaže se kot neenakomerna porazdelitev tranzitnega prometa med povezavami. Ker je eksaktno težko določljiva, jo je težko izpolniti, kot tudi preveriti. Primera hierarhične dekompozicije omrežij avtonomnih sistemov, ki upoštevata tipe relacij, najdemo v [98] in [71]. Dekompoziciji temeljita na različnih postopkih razvrščanja vozlišč v hierarhične nivoje. V [98] je razvrstitev vozlišč odvisna le od relacij ponudnik–stranka,

*Klika je množica vozlišč V_1 v grafu $G = (V, E)$, za katero velja, da med vsakim parom vozlišč v V_1 obstoji povezava.

[†]Cikel je pot, za katero velja, da je začetno vozlišče hkrati tudi končno vozlišče.

pri čemer se upošteva oddaljenost vozlišča od jedra grafa v smislu omenjenih relacij. Metoda razlikuje še med čistimi strankami in ponudniki. Število hierarhičnih nivojev ni fiksno. V [71] razvrstijo vozlišča v pet nivojev. Končna razvrstitev, še posebej v višje nivoje, je močno odvisna od relacij enakovreden–z–enakovrednim. Oceno o primerljivosti hierarhične strukture označenega grafa z merjenim dobimo, če primerjamo deleže vozlišč po hierarhičnih nivojih. Od uporabljene metode dekompozicije je odvisno, ali bo ocena veljala za hierarhično strukturo relacij ponudnik–stranka ali predvsem za strukturo relacij enakovreden–z–enakovrednim. Če primerljivost hierarhične strukture modeliranega grafa z merjenim ni dosežena, obstoji večja verjetnost, da usmerjevalne poti v modeliranem omrežju ne odražajo realnih lastnosti.

V merjenih omrežjih obstojijo avtonomni sistemi, ki so med seboj gosto povezani z relacijami enakovreden–z–enakovrednim in nimajo ponudnikov. Ti avtonomni sistemi so jedro omrežja, ki ima pomembno vlogo pri zagotavljanju povezljivosti in nosi znaten delež tranzitnega prometa. Klika tovrstnih sistemov je jedro v najožjem pomenu. Za verodostojno reprodukcijo karakteristik realnega omrežja bi moral tudi označeni graf vsebovati kliko, ki je po velikosti primerljiva z velikostjo klik v merjenih grafih. Če ta lastnost ni pravilno modelirana, obstoji večja verjetnost nepravilne porazdelitve globalnih prometnih tokov. Zaradi pomembne vloge jedra v hierarhiji celotnega omrežja, se pričakuje tudi velik vpliv pravilne določitve klike na uspešnost reprodukcije lastnosti hierarhične strukture.

Pri študiju tipov poslovnih relacij v merjenih omrežjih [74] smo ugotovili, da delež avtonomnih sistemov z nižjo stopnjo v vlogi ponudnika avtonomnemu sistemu z višjo stopnjo znaša le 0,2%. Upoštevajoč morebitne deviacije, v definiciji problema TRR zahtevamo delež 0,5%. Če doseženi delež presega zahtevanega, nastopajo manjši avtonomni sistemi v večjem številu usmerjevalnih poti brez dolin, zato nosijo več tranzitnega prometa kot v resničnem internetu.

Cikel zaporedja relacij ponudnik–stranka pomeni, da avtonomni sistem posredno nastopa kot ponudnik samemu sebi. Gre za anomalijo, ki je v merjenih grafih ni, zato jo v zadnji lastnosti označenega grafa prepovemo.

4.5 Porazdelitev relacij enakovreden–z–enakovrednim

Med vsemi zahtevami problema TRR je najtežje zadostiti zahtevi po ustrezni hierarhični strukturi, ker merila niso enoumna. Na hierarhično strukturo vplivajo tako relacije ponudnik–stranka [98] kot tudi relacije enakovreden–z–enakovrednim [71]. Ker smo usmerjenost relacij ponudnik stranka zelo natančno definirali že z lastnostjo gradienta stopnje, se je potrebno osredotočiti predvsem na ustrezno modeliranje porazdelitve relacij enakovreden–z–enakovrednim. Zanima nas verjetnost obstoja relacije enakovreden–z–enakovrednim med dvema poljubnima vozliščema v odvisnosti od njunih stopenj. Cilj je poiskati analitično funkcijo, ki bo čim bolj avtentično odražala empirične vzorce, ki jih najdemo v resničnem Internetu.

Definicija 2 (Verjetnost relacij enakovreden–z–enakovrednim). *Naj bo $pp(x, y)$ funkcija, ki za dani spremenljivki x in y zavzame vrednost enako verjetnosti, da povezava (i, j) predstavlja relacijo enakovreden–z–enakovrednim, pri čemer velja $x = rang(i)$ in $y = rang(j)$. Rang vozlišča je njegova zaporedna številka v padajočem, po stopnjah urejenem zaporedju vozlišč, pri čemer se vsaka stopnja pojavi le enkrat.*

Uporaba rangov namesto stopenj vozlišč omogoča kompaktnejšo predstavitev distribucije vozlišč in s tem natančnejše ujemanje analitičnega izraza z empiričnimi podatki. Porazdelitev stopenj vozlišč v odvisnosti od njihovih rangov se namreč pokorava potenčnemu zakonu [87]. Porazdelitev vozlišč višjih stopenj je zato redka, kar pomeni, da se stopnje prisotnih vozlišč med seboj močno razlikujejo. V CAIDINEM merjenem grafu za september 2006 je najvišja stopnja vozlišča 2373, sledijo pa ji vozlišča s stopnjami 2019, 1702, 1485 in 1274. Če uporabimo predstavitev z rangi, se našteje stopnje pretvorijo v vrednosti od 1 do 5.

V nadaljevanju izpeljemo analitični izraz za verjetnostno funkcijo relacij enakovreden–z–enakovrednim. Izhajamo iz treh verjetnostnih porazdelitev relacij enakovreden–z–enakovrednim za septembrske topologije iz let 2004, 2005 in 2006. Tabela 4.1 podaja njihove osnovne podatke. Nato opišemo skalirni model, ki omogoča računanje splošne verjetnostne funkcije. Ta je z analitičnim izrazom podana na koncu podpoglavja.

topologija	# vozlišč	# povezav	% relacij enak.–z–enak.
september 2004	17848	37172	11,22
september 2005	20344	40941	8,11
september 2006	23149	47368	8,77

Tabela 4.1: Modelirani grafi

4.5.1 Verjetnostna porazdelitev

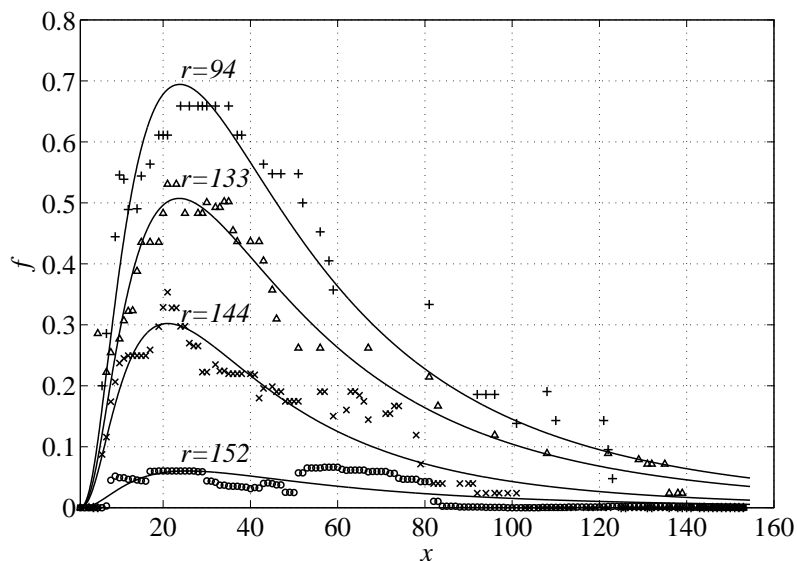
Z namenom izpeljave primernega matematičnega izraza, ki opisuje funkcijo dveh neodvisnih spremenljivk, opazujemo najprej skupino funkcij z eno spremenljivko. Tako imenovane parcialne funkcije dobimo tako, da eno od neodvisnih spremenljivk obravnavamo kot konstanto. Ko najdemo primeren matematičen izraz za parcialno funkcijo, analiziramo še odvisnost od druge spremenljivke.

Naj bo $f_r(x)$ parcialna verjetnostna funkcija, ki podaja verjetnost, da je na povezavi (i, j) vzpostavljena relacija enakovreden–z–enakovrednim. Pri tem je bodisi i bodisi j ranga r , medtem ko je rang drugega vozlišča neodvisna spremenljivka x . Z drugimi besedami, enega od argumentov funkcije $pp(x, y)$ nastavimo na fiksno vrednost r . Število povezav, ki dejansko povezujejo dani par rangov, je majhno. Zato smo za prikaz trendov pri iskanju funkcije $f_r(x)$ na slikah 4.2 in 4.3 uporabili filter tekočega povprečja dolžine 21. Pri iskanju parametrov funkcije filter ni uporabljen.

Na sliki 4.2 so prikazani podatki za topologijo iz leta 2006 in sicer za višje range: 152, 144, 133 in 94. Ti rangi pripadajo nižjim stopnjam vozlišč. Maksimalni rang R_0 v tem grafu je 153. Obliko funkcije za višje range med $R_0/2$ in R_0 aproksimiramo z lognormalno funkcijo:

$$f_r(x) = \frac{a}{x} e^{-\left(\frac{\ln(x)-b}{c}\right)^2} \quad (4.1)$$

Na sliki 4.2 so z neprekinjeno črto prikazane lognormalne krivulje, ki se prilegajo podatkom. S spreminjajočim rangom r se spreminjajo tudi parametri a , b in c , vendar vselej tako, da so relacije enakovreden–z–enakovrednim verjetnejše z vozlišči z višjo stopnjo. Verjetnost pri vozliščih z najvišjo stopnjo zopet upade, kar je v skladu z vzorcem, da so največji avtonomni sistemi predvsem ponudniki manjšim strankam, relacije enakovreden–z–enakovrednim pa imajo vzpostavljene med seboj. Izbor, da so višje stopnje predstavljene z nižjim rangom, je bil narejen zaradi lažjega iskanja primerne funkcije. Podoben način so ubrali tudi avtorji v [87].



Slika 4.2: Parcialna verjetnostna funkcija za višje range (nižje stopnje)

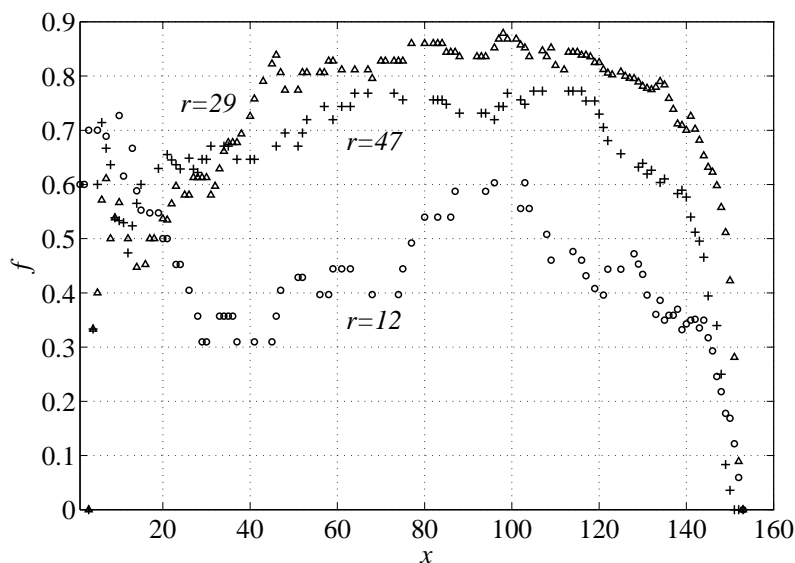
Oblika parcialne verjetnostne funkcije se pri nižjih rangih zelo spremeni, kot je prikazano na sliki 4.3. Natančna primerjava grafov na slikah 4.2 in 4.3 pokaže, da se nekatere vrednosti podatkov za isti par rangov ne ujemajo. Razlog za to je prikaz z uporabo filtra, kar pa ne vpliva na poiskano funkcijo.

Glede na obliko funkcije, ki jo izkazujejo podatki za nižje range, bi bilo pričakovati, da je potrebna nova oblika funkcije $f_r(x)$. Izkaže pa se, da to ni potrebno zaradi načina uporabe $f_r(x)$ v $pp(x, y)$. Funkcija $pp(x, y)$ je simetrična glede na x in y . Simetrijo dosežemo, če definiramo vmesno verjetnostno funkcijo $p(x, y)$ kot

$$p(x, y) = \begin{cases} f_y(x) & x \leq y \\ f_x(y) & \text{sicer} \end{cases}, \quad (4.2)$$

pri čemer $p(x, y)$ služi kot osnova za končno funkcijo $pp(x, y)$, ki jo dobimo s skaliranjem. Pri takšni definiciji je uporabljen le majhen začetni del aproksimacije z lognormalno funkcijo. Ker podatki do določene mere izkazujejo lognormalno obliko tudi za nižje vrednosti r v intervalu $0 < x < r$, uporabimo aproksimacijo s funkcijo 4.1 za vse vrednosti r .

Naslednji korak pri izpeljavi funkcije $pp(x, y)$ je modeliranje parametrov a , b in c v odvisnosti od r . Funkcijo smo poiskali z metodo najmanjših kvadratov za višje vrednosti r . Izris na sliki 4.4 pokaže potenčno odvisnost



Slika 4.3: Parcialna verjetnostna funkcija za nižje range (višje stopnje)

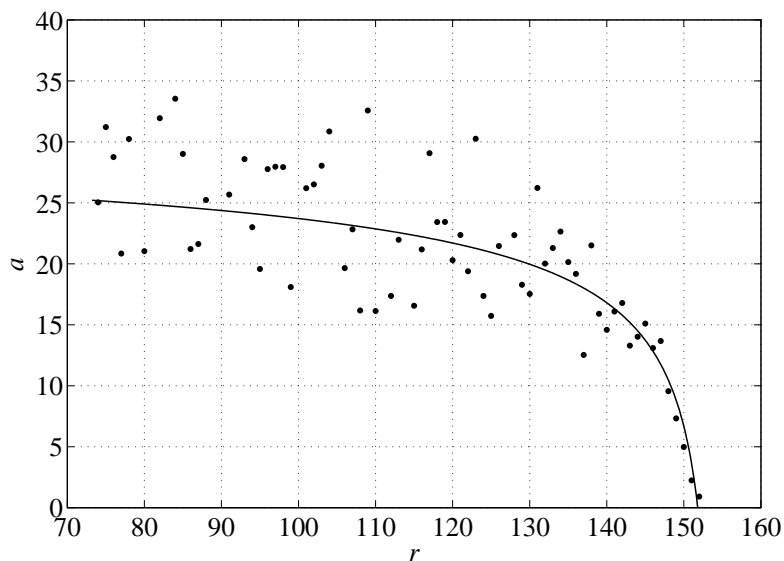
med a in r , medtem ko posamezne vrednosti b in c na sliki 4.5 variirajo okoli konstant.

Za aproksimacijo parametra a smo uporabili funkcijo, kot jo podaja spodnja enačba:

$$a = p_1 (R_0 + 1 - r)^{p_2} + p_3 \quad (4.3)$$

Da bi dodatno potrdili potenčno odvisnost, v grafu na sliki 4.6 izrisujemo $a - p_3$ v odvisnosti od $R_0 + 1 - r$ na logaritemsko-logaritemski skali. Izris izraža jasno linearnost za višje vrednosti rangov.

Grafi in enačbe, ki smo jih navajali ob razlagi, se nanašajo na podatke za september 2006. Podatki za leti 2004 in 2005 izkazujejo podobne lastnosti in vodijo do enakih zaključkov. Do variacij pride le pri vrednostih izračunanih parametrov, ki so za vsa tri leta podani v tabeli 4.2. Variacije lahko pripišemo več dejavnikom. Delež relacij enakovreden–z–enakovrednim je bil septembra 2004 11,22%, septembra 2005 je znašal 8,11%, septembra 2006 pa 8,77%. Maksimalni rang je naraščal z vrednosti 129 v letu 2004 na vrednost 135 leta 2005 in nazadnje na vrednost 153 leta 2006. Tudi preciznost anotacij je bila v zgodnejših letih morda manjša. Kljub manjšim variacijam so za opazovana leta vrednosti podobne, kakor tudi oblika dobljene funkcije $p(x, y)$. Konstantnost porazdelitve relacij enakovreden–z–enakovrednim je

Slika 4.4: Iskanje funkcije za parameter a

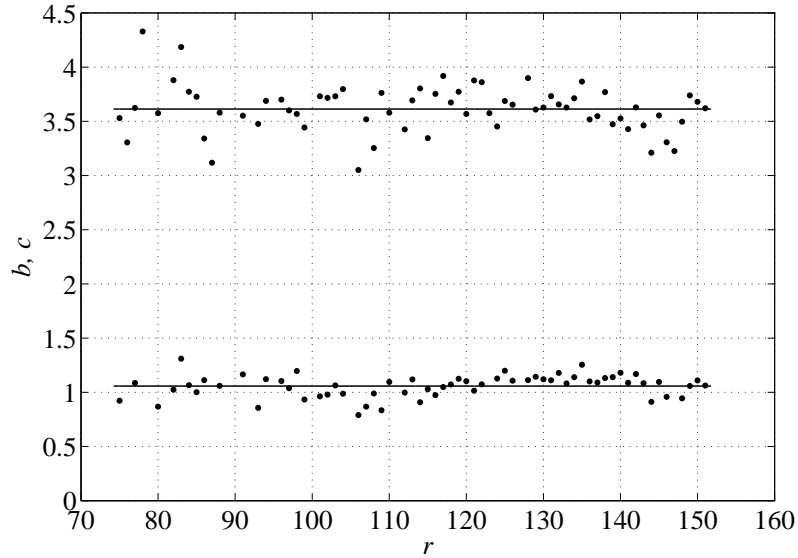
parameter	2004	2005	2006
R_0	129	135	153
p_1	-47,05	-33,58	-46,25
p_2	-0,812	-0,311	-0,353
p_3	24,29	30,72	35,02
b	3,332	3,338	3,613
c	0,941	0,953	1,057

Tabela 4.2: Parametri modela za grafe z vseh treh let

tudi v skladu z ugotovitvami v [95], da metrike, kot so na primer porazdelitev razdalj med vozlišči (*angl. distance distribution*), razvrstitveni koeficient (*angl. assortivity coefficient*), srednje grozdenje (*angl. mean clustering*) in povprečna stopnja (*angl. average degree*) ostajajo skoraj konstantne preko opazovanega petletnega obdobja.

4.5.2 Skalirni model

Glede na ugotovitev, da je oblika porazdelitve relacij enakovreden–z–enakovrednim skoraj konstantna, je model z zadnjimi izmerjenimi parametri mogoče na enostaven način skalirati in se tako prilagoditi različ-

Slika 4.5: Iskanje funkcij za parametra b in c

nim velikostim vhodnih grafov ter različnim deležem relacij enakovreden– z –enakovrednim.

Definicijsko območje funkcije $p(x, y)$ je omejeno z $x, y \in \{1, \dots, R_0\}$. V splošnem pa se maksimalni rang ciljnega grafa R_{max} razlikuje od R_0 . Da bi to upoštevali na način, ki bi ohranil obliko verjetnostne funkcije, uvedemo skalirni faktor:

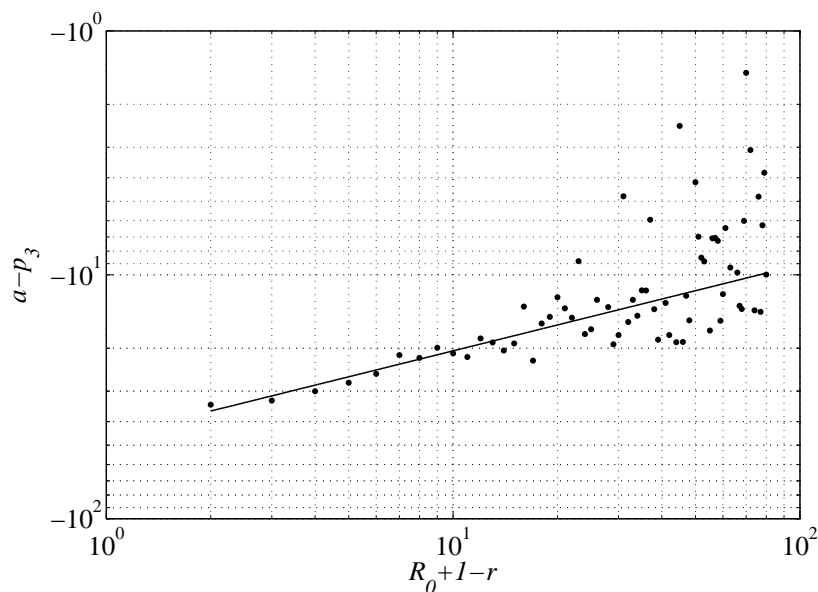
$$\lambda = \frac{R_0}{R_{max}} \quad (4.4)$$

Tudi število relacij enakovreden– z –enakovrednim, ki jih zahteva uporabnik, se lahko razlikuje od izhodiščnega grafa. Število relacij enakovreden– z –enakovrednim tako skaliramo s faktorjem

$$\gamma = \frac{N_e}{\sum_{(i,j) \in E} p(\lambda r_i, \lambda r_j)}, \quad (4.5)$$

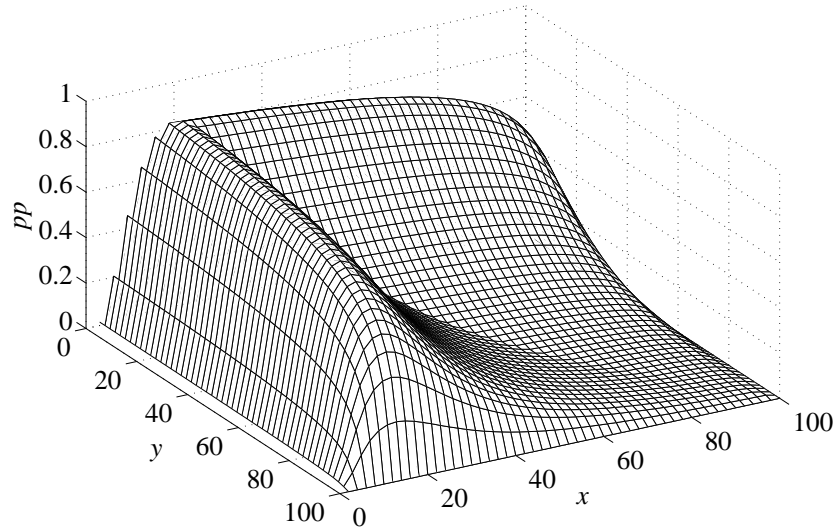
kjer je N_e število zahtevanih relacij enakovreden– z –enakovrednim. Vsota v imenovalcu predstavlja pričakovano število relacij enakovreden– z –enakovrednim v λ -skalirani topologiji. Končen zapis verjetnostne funkcije relacij enakovreden– z –enakovrednim je:

$$pp(x, y) = \gamma p(\lambda x, \lambda y). \quad (4.6)$$

Slika 4.6: Izris parametra a na log-log skali

Pri prevelikih vrednostih faktorja γ lahko funkcija $pp(x, y)$ prekorači vrednost 1. Način, ki bi vrednosti, ki so višje od 1, enostavno znižal na 1, ni ustrezen, ker nima prave osnove v realnih podatkih. Prišlo bi do popačenja oblike funkcije. Zato je potrebna bodisi primerna izbira faktorja γ , tako da do popačitve ne pride, bodisi uporaba metode, ki skalira obratno verjetnost $1-p(x, y)$, to je verjetnost, da relacija ni tipa enakovreden–z–enakovrednim. Verodostojnost slednje metode bi bilo potrebno natančneje preučiti.

Paziti je potrebno tudi na vrednost R_{max} , na velikost ciljnega grafa, na korelacije stopenj in na skupni delež relacij enakovreden–z–enakovrednim. Če te vrednosti pomembno odstopajo od izhodiščnih, potem verodostojnost modela ni zajamčena. Na primer, ker izhajamo iz podatkov za grafe reda velikosti med 15.000 in 25.000 vozlišč z deležem relacij enakovreden–z–enakovrednim med približno 8% in 11%, uporaba modela s podatki izven teh mej ne bo zanesljivo vodila do dobrih rezultatov. Na sliki 4.7 je izrisana verjetnostna funkcija relacij enakovreden–z–enakovrednim za topologijo z $R_{max} = 100$ in $\gamma = 1$.



Slika 4.7: Izris funkcije $pp(x, y)$ za $R_{max} = 100$ in $\gamma = 1$

4.5.3 Končni zapis funkcije

Zgornje izpeljave lahko združimo v en skupni zapis verjetnostne funkcije relacij enakovreden–z–enakovrednim

$$pp(x, y) = \begin{cases} \gamma \frac{a}{\lambda x} e^{-\left(\frac{\ln(\lambda x) - b}{c}\right)^2} & x \leq y \\ \gamma \frac{a}{\lambda y} e^{-\left(\frac{\ln(\lambda y) - b}{c}\right)^2} & \text{sicer} \end{cases}, \quad (4.7)$$

kjer je izračun parametrov γ , δ , a , b in c naslednji. Parameter γ je razmerje med zahtevanim številom relacij enakovreden–z–enakovrednim v ciljni topologiji in številom relacij enakovreden–z–enakovrednim, ki bi se je pojavile v λ -skalirani izhodiščni topologiji (enačba 4.5). Parameter λ predstavlja razmerje med najvišjim rangom izhodiščne topologije in najvišjim rangom ciljne topologije (enačba 4.4). Parametri a , b in c so odvisni samo od izhodiščne topologije. Njihove vrednosti so določene s tabelo 4.2 in enačbo 4.3, kjer r ustreza y , če je $x \leq y$, sicer pa x .

4.6 Generator poslovnih relacij

V tem poglavju je opisan generator, ki smo ga razvili za reševanje TRR problema. Formalno ga podaja algoritem 7. Algoritem označuje relacije na danem naključnem grafu glede na lastnosti, ugotovljene v poglavju 4.4 in glede na verjetnostno funkcijo relacij enakovreden–z–enakovrednim, kot smo jo določili v poglavju 4.5.

Algoritem 7 Generator poslovnih relacij v naključnem grafu

Vhod: neusmerjen povezan graf $G = (V, E)$ z naboroma vozlišč V in povezav E ter delež relacij enakovreden–z–enakovrednim p_e

Izhod: označen graf

- 1: Izberi nabor vozlišč V_C in povezav E_C iz gostega jedra z uporabo požrešne hevrstike v algoritmu 8. Izbrane povezave označi z relacijami enakovreden–z–enakovrednim.
- 2: Naj bo $G' = (V', E')$ graf, ki ga dobimo iz G tako, da nadomestimo nabor V_C z enim vozliščem k . Poišči drevo najkrajših razdalj T iz k do vseh ostalih vozlišč v G' , pri čemer uporabi število hopov kot metriko. Povezave usmeri navzven, tako da dobimo usmerjene poti od k do vsakega vozlišča v T . Ponovno razširi G' v G , tako da se zamenja vozlišče k s podgrafom gostega jedra, pri čemer usmeritve povezav ostanejo nespremenjene.
- 3: Naj bo E_U nabor neoznačenih povezav, za katere velja $E_U = E \setminus E_C \setminus E_T$, kjer je E_T nabor povezav v T . Izračunaj zahtevano število relacij enakovreden–z–enakovrednim kot:

$$N_e = \frac{p_e}{100} |E| - |E_C|. \quad (4.8)$$

Izračunaj γ po enačbi 4.5 tako, da je v imenovalcu uporabljen nabor E_U namesto nabora E , zato da se skaliranje ustrezno prilagodi zmanjšanemu naboru povezav. Povezave $(i, j) \in E_U$ označi z relacijami enakovreden–z–enakovrednim z verjetnostjo $pp(r_i, r_j)$, kjer sta r_i in r_j ranga vozlišč i in j .

- 4: Preostale neoznačene povezave usmeri od vozlišča z višjo stopnjo k vozlišču z nižjo ali enako stopnjo. Če pri tem nastanejo usmerjene zanke, obrni povezavo v nasprotno stran.
-

V koraku 1 identificiramo gosto jedro in povezave znotraj jedra označimo z relacijami enakovreden–z–enakovrednim, s čimer zadostimo tretji zahtevi problema TRR. S korakom 2 dosežemo lastnost povezanosti vozlišč. Pred razširitvijo grafa G' je med vsakim parom vozlišč natanko ena veljavna pot, v vozlišču k pa je vrh, kjer se usmerjenost povezav obrne. Po razširitvi

vozlišča k na gosto jedro je v vsako veljavno pot dodana največ ena povezava z relacijo enakovreden–z–enakovrednim. Tako vse povezave ostanejo veljavne. Drevo najkrajših razdalj smo izbrali zato, ker avtonomni sistemi stremijo k čim bližji razdalji od jedra.

Korak 3 preostalim neoznačenim povezavam pripiše relacije enakovreden–z–enakovrednim glede na izpeljano verjetnostno funkcijo. Število zahtevanih relacij enakovreden–z–enakovrednim N_e izračunamo tako, da od skupnega števila odštejemo $|E_C|$, ker je bilo toliko povezav že označenih z relacijo enakovreden–z–enakovrednim v koraku 1. Tudi v imenovalcu enačbe 4.5 upoštevamo, da smo nekatere povezave že označili. Zato namesto po naboru $|E|$ seštevamo po naboru $|E_U|$, ki predstavlja še neoznačene povezave. Funkcija $pp(x, y)$ bi se za še neoznačene povezave v splošnem lahko razlikovala od funkcije $pp(x, y)$ za vse povezave. S preizkusom modeliranja alternativne $pp(x, y)$, pa smo ugotovili, da pride do razlike le za range, ki pripadajo gostemu jedru, kar nima vpliva na algoritem 7.

S korakom 4 označimo vse preostale neoznačene povezave, tako da jih usmerimo v skladu z gradientom stopenj, to je od vozlišča z višjo stopnjo k vozlišču z nižjo. Peti zahtevi problema TRR zadostimo tako, da relacijo ponudnik–stranka, ki v koraku 4 ustvari cikel, obrnemo. Da pri tem ne ustvarimo novega cikla, je moč formalno dokazati s protislovjem. Do novega cikla namreč lahko pride le, če se obravnavana povezava v dveh vozliščih stika s ciklom, ki je obstajal že predhodno. Le-tega pa po prvih treh korakih v grafu ni.

Za iskanje klike gostega jedra smo uporabili požrešno hevristiko, predlagano v [71], ki smo jo priredili za neusmerjene grafe. Podana je z algoritmom 8. Kot prvo vozlišče v gostem jedru izberemo vozlišče z najvišjo stopnjo. Da je le-to vedno vsebovano v gostem jedru, so predpostavili tudi drugi avtorji [98]. Ob vsakem koraku se kot naslednje kandidate za gosto jedro obravnava vozlišča, ki so povezana z vsemi že izbranimi elementi gostega jedra. Izmed kandidatov se kot člana izbere tistega z najvišjo stopnjo. Če kandidatov ni, se postopek ustavi.

4.7 Kakovost algoritma

V tem podpoglavju analiziramo natančnost predlaganega algoritma glede na merjene grafe. Najprej primerjamo hierarhično strukturo grafa, označenega z algoritmom in merjenega grafa. Nato ocenimo zmožnost algoritma,

Algoritem 8 Hevristika za izbiro gostega jedra

Vhod: neusmerjen povezan graf $G = (V, E)$ z naboroma vozlišč V in povezav E

Izhod: gosto jedro $G_C = (V_C, E_C)$

- 1: Izberi $x \in V$ z maksimalno stopnjo. Nastavi nabora $V_C = \{x\}$ in $E_C = \{\}$.
 - 2: Naj bo $v_C \in V_C$ in naj $d(k, V_C)$ pomeni število povezav oblike (k, v_C) . Izberi $k \in V \setminus V_C$ z maksimalnim $d(k, V_C)$. Če je več vozlišč z enakim maksimalnim $d(k, V_C)$, izberi vozlišče z najvišjo stopnjo.
 - 3: Če je $d(k, V_C) = |V_C|$, potem:
 - (a) $V_C = V_C \cup \{k\}$,
 - (b) $E_C = E_C \cup \{(k, v_C) : v_C \in V_C\}$,
 - (c) pojdi na korak 2.
 - 4: Vrni (V_C, E_C) .
-

da pravilno označuje relacije na globalni ravni s primerjavo porazdelitev razdalj. Na koncu argumentiramo tudi primernost uporabe drevesa najkrajših razdalj v 2. koraku algoritma 7 in opazujemo ujemanje tipov relacij za posamezne povezave.

4.7.1 Natančnost hierarhije

Tabela 4.3 podaja porazdelitev vozlišč glede na hierarhično dekompozicijo v [71] za vse tri analizirane grafe iz let 2004-2006. Za vsako leto sta podana dva stolpca. V prvem je porazdelitev za merjen graf, kot so ga označili v CAIDI, v drugem pa porazdelitev za isti graf, ki je bil označen z našim algoritmom. V drugem stolpcu podajamo povprečje 100 ponovljenih označb grafa s pripadajočo standardno deviacijo v oklepaju. Vse vrednosti so podane v odstotkih.

Primerjava merjenih in označenih grafov pokaže dobro učinkovitost predlaganega algoritma. Vse razlike deležev vozlišč so manjše od ene odstotne točke, celo za močno prevladujoča vozlišča strank, ki jih je skupaj več kot 80%. Standardne deviacije porazdelitev vozlišč pri generiranih označbah so majhne. Število vozlišč znotraj gostega jedra je pri merjenih topologijah malo manjše in sicer znaša 8, 11 in 12 vozlišč v primerjavi z 12, 12 in 13 vozlišči pri topologijah z generiranimi označbami. Rezultati so enako prepričljivi za vsa tri analizirana leta, kljub temu da je verjetnostna funkcija

nivo [%]	meritev	generator
september 2004		
gosto jedro	0,04	0,07 (0,00)
tranzitno jedro	5,17	4,87 (0,08)
zunanje jedro	3,37	3,01 (0,08)
majhni ponudniki	9,62	9,39 (0,10)
stranke	81,80	82,66 (0,07)
september 2005		
gosto jedro	0,05	0,06 (0,00)
tranzitno jedro	4,57	4,25 (0,07)
zunanje jedro	3,12	2,70 (0,08)
majhni ponudniki	9,73	9,66 (0,09)
stranke	82,53	83,33 (0,07)
september 2006		
gosto jedro	0,05	0,06 (0,00)
tranzitno jedro	4,33	4,28 (0,06)
zunanje jedro	3,04	2,69 (0,06)
majhni ponudniki	9,97	9,64 (0,08)
stranke	82,61	83,33 (0,06)

Tabela 4.3: Porazdelitev vozlišč v 5-nivojski hierarhiji

$pp(x, y)$ osnovana na podatkih iz septembra 2006. Pri izračunu $pp(x, y)$ smo seveda upoštevali specifične podatke obravnavanih topologij R_{max} in p_e . Gledano po vrsti za leta 2004, 2005 in 2006 R_{max} znaša 129, 135 in 153, p_e pa 11,22%, 8,11% in 8,77%.

Porazdelitev vozlišč po hierarhični dekompoziciji ponudnik–stranka, kot jo predlagajo v [98], je podana v tabeli 4.4. Vsi podatki so podani v odstotkih. S pomišljajem so označene kategorije, kjer ni vozlišč. V stolpcih z rezultati za generirane označbe so podane povprečne vrednosti 100 ponovitev.

Za vsa tri analizirana leta je podobnost porazdelitve za merjene in generirane označbe visoka. Število vozlišč na 1. nivoju je vedno enako, ker je odvisno samo od topoloških lastnosti in ne od tipov relacij. Razlike med merjeno in generirano označbo so manjše od ene odstotne točke za vse kategorije, razen za stranke na 3. nivoju. Teh je številčno največ, zato so bolj podvržene višjim absolutnim razlikam. Relativno gledano se podatki dobro ujemajo tudi v tej kategoriji, saj razlika niha med 6% in 11%. Do najvišje relativne razlike pride pri ponudnikih na 5. nivoju, kjer je pri generiranih

nivo	tip	september 2004		september 2005		september 2006	
		mer.	gen.	mer.	gen.	mer.	gen.
1	ponudnik	0,07	0,07	0,06	0,06	0,06	0,06
2	ponudnik	6,91	7,49	6,55	6,97	6,31	6,78
	stranka	32,61	32,57	30,39	30,31	27,93	27,89
3	ponudnik	6,35	6,62	6,26	6,76	6,58	7,14
	stranka	38,00	40,29	38,79	42,35	40,22	44,61
4	ponudnik	1,08	0,94	1,19	1,00	1,08	0,87
	stranka	11,42	10,93	12,32	11,32	12,61	11,63
5	ponudnik	0,11	0,03	0,09	0,05	0,10	0,04
	stranka	1,59	1,03	1,42	1,10	1,27	0,92
6	ponudnik	0,01	-	0,01	0,00	0,00	-
	stranka	0,07	0,04	0,09	0,06	0,12	0,06
7	ponudnik	-	-	-	-	-	-
	stranka	0,01	-	0,00	-	-	-
Σ	ponudnik	14,54	15,14	14,16	14,85	14,14	14,89
	stranka	83,71	84,86	83,01	85,15	82,15	85,11
	nerazp.	1,75	-	2,83	-	3,71	-

Tabela 4.4: Porazdelitev vozlišč v hierarhiji ponudnik–stranka

označbah do 72% manj vozlišč kot pri merjenih. K sreči pa je v tej kategoriji le 0,09% vseh vozlišč. Pri nekaterih kategorijah se podatki zelo dobro ujemajo po obeh plateh, absolutno in relativno, medtem ko ni kategorij, kjer bi bile razlike po obeh plateh velike.

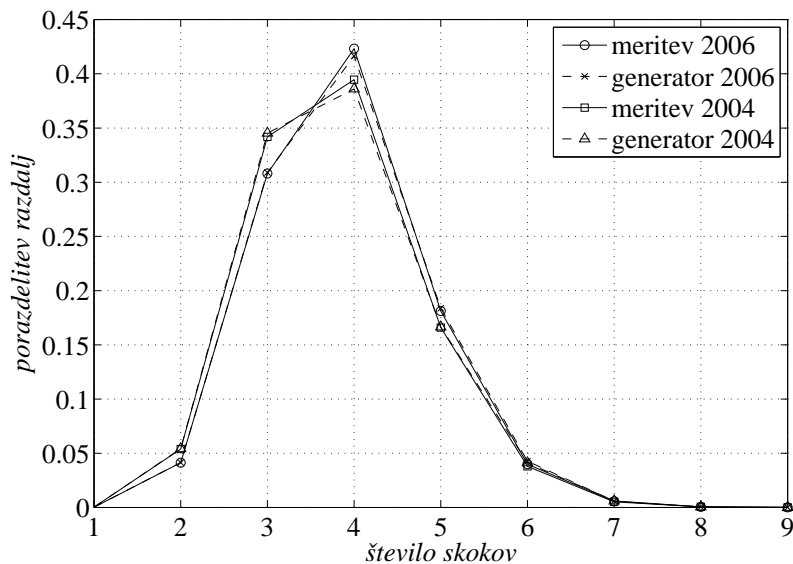
Dekompozicijski algoritem iz [98] ne razvrsti vseh vozlišč v topologijah z merjenimi označbami. Delež takšnih vozlišč je med 1,75% in 3,71%, kar lahko predstavlja enega od razlogov za razlike v tabeli 4.4. Avtorji razlagajo ta ostanek z napačno označenimi relacijami, bodisi zaradi nepravilnih usmerjevalnih podatkov, bodisi zaradi heuristike za prepoznavanje relacij. Števili vseh ponudnikov pri topologijah z generiranimi in merjenimi označbami se dobro ujemata. Vsota vseh strank je pri generiranih označbah nekoliko višja. Videti je, da razlika ustreza ravno številu nerazvrščenih vozlišč.

4.7.2 Porazdelitev razdalj

Primerjava merjenih in generiranih označb grafov po porazdelitvi razdalj, ki je na tem področju pogosto uporabljena metrika, pokaže možnost algo-

ritma, da ustrezno reproducira globalne lastnosti poslovnih relacij. Funkcija podaja število parov vozlišč na določeni razdalji v razmerju do skupnega števila parov v topologiji. V testu smo uporabili pravilo usmerjanja po poteh brez dolin [77], pri čemer se za razdaljo med dvema vozliščema upošteva najkrajšo od obstoječih poti.

Na sliki 4.8 je prikazana primerjava za topologiji iz let 2004 in 2006. Enaki zaključki veljajo za leto 2005. Za generirane označbe je prikazano povprečje 100 ponovitev. Standardna deviacija za vse meritve je manjša od 10^{-3} . Porazdelitev razdalj za grafe z generiranimi označbami zelo natančno sledi porazdelitvam za merjene označbe. Čeprav smo za generacijo označb na starejših topologijah uporabili parametre, ki temeljijo na podatkih iz leta 2006, je ujemanje porazdelitev pri topologijah iz let 2004 in 2005 enako natančno.



Slika 4.8: Porazdelitev razdalj za merjene in generirane označbe grafov

4.7.3 Primernost koraka 2 in ujemanje posameznih relacij

Algoritem za generacijo relacij 7 v koraku 2 določi relacije ponudnik–stranka na osnovi drevesa najkrajših razdalj od gostega jedra proti ostalim vozliščem. Prvotni namen tega koraka je bil zagotoviti povezljivost med vsemi pari vozlišč v topologiji. Glede na to, da v realističnem sistemu vzorec do-

mnevno ni tako enostaven, zlasti zato, ker je na tak način označenih skoraj polovico vseh relacij, ustreznost tega koraka preverjamo posebej. Rezultate prikazuje tabela 4.5. Primerjava z merjenjem in z generacijo označenih grafov pokaže, da se tako po tipu kot tudi po orientaciji ujema preko 97% povezav, določenih s korakom 2. Napačno usmerjenih je manj kot 0,5%, ostalim pa je bil določen napačen tip, od tega približno 0,7% tip soroden-s-sorodnim in nekaj več kot 1% tip enakovreden-z-enakovrednim. Iz tega sklepamo, da se s korakom 2 kakovost algoritma ne poslabša.

ujemanje [%]	september 2004	september 2005	september 2006
pravilno	97,34	97,69	97,81
napačna smer	0,48	0,44	0,38
sorod.-s-sorod.	0,64	0,76	0,67
enak.-z-enak.	1,54	1,11	1,14

Tabela 4.5: Ujemanje s korakom 2 določenih relacij z merjenimi relacijami

Opravili smo tudi primerjavo ostalih označenih povezav, čeprav natančno ujemanje merjenih in generiranih ni naš cilj, ker je generacijski algoritem po naravi naključen. Primerjava v tabeli 4.6 vseeno pokaže, da se označene povezave v veliki meri ujemajo, tudi za redkejšje relacije enakovreden-z-enakovrednim. Za primer, ko iz množice naključno izberemo 10% elementov, od katerih je 10% ugodnih, je namreč matematično upanje le 10%. V našem primeru pa je za vse tri topologije dosežen najmanj 50% uspeh.

ujemanje [%]	ponudnik-stranka			enak.-z-enak.		
	2004	2005	2006	2004	2005	2006
ponudnik-stranka	93,52	94,33	94,31	39,01	47,82	45,14
enak.-z-enak.	5,20	4,23	4,44	60,49	52,58	53,60
napačna smer	0,82	0,77	0,79	-	-	-
sorod.-s-sorod.	0,51	0,62	0,56	0,07	0,15	0,14

Tabela 4.6: Ujemanje posameznih relacij

4.8 Generator relacij na naključnih topologijah

Algoritem 7 lahko uporabimo na katerikoli naključni topologiji. Če je bila topologija generirana z uporabo zadnjih izsledkov, pričakujemo po uporabi

našega algoritma dobro ujemanje lastnosti tako dobljenega označenega grafa z realnimi merjenimi omrežji, ki smo jih analizirali v prejšnjem podpoglavju. Javno dostopnih implementacij generatorjev, ki temeljijo na dK -serijah žal ni. Zato smo za ta preizkus izbrali njihovega predhodnika, stopenjski generator Inet [90].

Algoritem smo uporabili na Inet topologijah dveh velikosti. Z uporabo prednastavljenih parametrov za generacijo Inet topologij je namreč moč doseči primerljivo povprečno stopnjo 4,1 le pri topologijah, ki so po velikosti približno polovico manjše od CAIDINIH, to je 10.000 vozlišč. Pri primerljivi velikosti 20.000 vozlišč znaša povprečna stopnja 5,22. Za dosego povprečne stopnje 4,1 pri velikosti 20.000 vozlišč bi moral biti parameter, ki določa delež vozlišč s stopnjo 1, nerealistično visok. Delež relacij enakovrednih relacij, smo nastavili na vrednost $p_e = 8,77$. Za obe študirani velikosti smo 100-krat ponovili generacijo topologije in generacijo relacij. Povprečne vrednosti porazdelitve vozlišč po obeh hierarhičnih dekompozicijah podajata tabeli 4.7 in 4.8.

V tabeli 4.7 je podana porazdelitev vozlišč po 5-nivojski hierarhiji na enak način, kot je bila za realistične topologije podana s tabelo 4.3. Porazdelitve vozlišč za Inet-ove in za CAIDINE topologije se na nivojih gostega in zunanega jedra dobro ujemajo. Na ostalih nivojih je nekoliko več odstopanj.

nivo	$ V = 10.000$ povp. st. = 4,1	$ V = 20.000$ povp. st. = 5,22
gosto jedro	0,07 (0,00)	0,06 (0,00)
tranzitno jedro	7,53 (0,15)	6,08 (0,07)
zunanje jedro	4,33 (0,09)	3,12 (0,04)
majhni ponudniki	7,43 (0,12)	4,31 (0,07)
stranke	80,64 (0,16)	86,43 (0,08)

Tabela 4.7: 5-nivojska hierarhija označenih Inet topologij

Tabela 4.8 podaja porazdelitev vozlišč po hierarhiji ponudnik–stranka in predstavlja ekvivalent tabele 4.4 za realistične topologije. Velikosti prvega nivoja Inet-ovih in CAIDINIH topologij se popolnoma ujemajo. Pri ostalih nivojih je porazdelitev nekoliko drugačna. Pri Inet-ovih topologijah je skupaj le 5 nivojev, namesto 6 oziroma 7 kot pri CAIDINIH. Tudi sicer so vozlišča v povprečju razporejena v višje nivoje, zato lahko sklepamo, da so Inet-ove topologije nekoliko bolj zgoščene. Na splošno se rezultat za

manjše generirane topologije z 10.000 vozlišči bolje ujeta z realističnimi, kot rezultat za večje topologije z 20.000 vozlišči.

nivo	tip	$ V = 10.000$ povp. st. = 4,1	$ V = 20.000$ povp. st. = 5,22
1	ponudnik	0,07	0,06
2	ponudnik	10,01	9,13
	stranka	35,08	52,41
3	ponudnik	5,22	1,38
	stranka	45,97	36,18
4	ponudnik	0,13	0,02
	stranka	3,43	0,80
5	ponudnik	-	-
	stranka	0,09	0,02
Σ	ponudnik	15,44	10,59
	stranka	84,56	89,41
	nerazp.	-	-

Tabela 4.8: Hierarhija ponudnik–stranka označenih Inet topologij

Poglavje 5

Vpliv strukture omrežij na izbiro sider

Z uporabo generatorja poslovnih relacij iz prejšnjega poglavja je algoritme za izbiro sidrnih točk moč analizirati na večjem naboru simulacijskih topologij. S tem lahko dosežemo višjo zanesljivost rezultatov, omogočena pa je tudi študija učinkovitosti algoritmov v odvisnosti od strukture omrežja.

V tem poglavju pokažemo, da imajo drevesne topologije v primerjavi z ostalimi vgrajeno omejitev, ki onemogoča od uporabniških parametrov neodvisno učinkovito izbiro sidrnih točk. Najprej analiziramo medsebojno odvisnost evaluacijskih metrik, ki so jih raziskovalci uporabljali do sedaj. Na podlagi tega uvedemo pojem *dh*-izboljšane izbire, ki zagotavlja učinkovitost, neodvisno od uporabniških parametrov. Nato formalno in s simulacijami analiziramo možnosti za *dh*-izboljšano izbiro v omrežjih različnih struktur. Poglavje končamo z uvedbo metrike, ki pove, kakšne strukturne lastnosti vodijo k boljšim pogojem za večjo učinkovitost algoritmov.

5.1 Metrike

Evaluacija algoritmov za izbiro sidrnih točk predstavlja velik izziv. Raziskovalci si v tem pogledu niso enotni. Najpogosteje uporabljene metrike so porazdelitev bremena po sidrnih točkah [45, 46, 48, 53, 54, 56–58, 60, 68], frekvenca izročanja med domenami [36, 46, 48, 53–56, 58, 60, 68], razdalja med mobilnim vozliščem in sidrno točko [53, 54, 68], zakasnitev pri izročanju [36, 45, 55, 56, 59, 61, 99, 100], cena signalizacije [29, 47, 49–

52, 58, 60, 61, 99, 100] in cena dostave paketov [29, 47, 49–52, 61, 99, 100]. Naštete metrike niso ortogonalne. V nadaljevanju razložimo odvisnost cene signalizacije C_{LU} , cene dostave paketov C_{PD} in zakasnitve pri izročanju D od frekvence izročanja med domenami f in razdalje med mobilnim vozliščem in sidrno točko d .

Cene so kompleksne metrike, ki vključujejo največ parametrov. Koncept so vpeljali Wang idr. [99, 100] in ga kasneje modificirali Xie idr. [49, 50]. Njihovemu zgledu so sledili mnogi drugi, tako na področju izbire sidrnih točk [29, 47, 51, 52, 61], kot tudi na sorodnih področjih, npr. [101]. Skupna cena C_{TOT} , dobljena kot vsota cene signalizacije C_{LU} in cene dostave paketov C_{PD} , predstavlja celovito merilo o učinkovitosti metode za izbiro sidrnih točk.

$$C_{TOT} = C_{LU} + C_{PD} \quad (5.1)$$

Druge predpostavke in natančne definicije C_{LU} in C_{PD} variirajo od avtorja do avtorja. Analiza treh reprezentativnih predlogov [49, 50], [51] in [29] pokaže odvisnosti, kot so navedene v tabelah 5.1 in 5.2.

referenca	cena signalizacije
[49, 50]	$C_{LU}(h, T, a_h, a_m, l_{hm}, d, \delta_U)$
[51]	$C_{LU}(\lambda, T, k, l_{hm}, d, \delta_U)^*$
[29]	$C_{LU}(h, l_{hm}, d, \delta_U)^*$
referenca	cena dostave paketov
[49, 50]	$C_{PD}(\eta, \lambda, \zeta, k, \alpha, \beta, \omega, l_{hm}, d, \delta_D)$
[51]	$C_{PD}(\varepsilon, \Theta_{HA}, \Theta_{MAP}, l_{hm}, d, l_{ch}, l_{cm}, \delta_D)^*$
[29]	$C_{PD}(\varepsilon, l_{hm}, d, \delta_D)^*$

* - na posamezno sejo

Tabela 5.1: Cene kot funkcije različnih parametrov

V vsaki od treh definicij cene signalizacije C_{LU} v tabeli 5.1 sta vsebovana parametra d in h . Po definiciji v [51] je h izražen posredno preko enačbe 5.2 [49, 50], kjer N predstavlja število dostopovnih točk, med katerimi mobilno vozlišče izbira. Kljub temu, da definicija C_{LU} po [29] ne upošteva parametra T , ga poleg d in h prepoznamo kot spremenljivko, ki pomembno vpliva na C_{LU} . Njegovo relacijo do f in h podaja enačba 5.3. Višja T in h ter nižji d vodijo do nižje cene C_{LU} . Preostala parametra, skupna vsem trem definicijam C_{LU} v tabeli 5.1, l_{hm} in δ_U , sta tipično konstantna.

d	Povprečna razdalja med mobilnim vozliščem in sidrno točko.
h	Kvocient izročanja. Povprečno število menjav dostopovne točke na menjavo sidrne točke.
f	Frekvenca izročanja med domenami. Povprečno število menjav sidrne točke na časovno enoto.
T	Povprečen čas povezave z isto dostopovno točko.
λ	Hitrost prihodov sej. Povprečno število sej na časovno enoto.
$l_{hn}(l_{hm}, l_{ch}, l_{cm})$	Povprečna razdalja med domačim agentom in mobilnim vozliščem (domačim agentom in sidrno točko, dopisnim vozliščem in domačim agentom, dopisnim vozliščem in sidrno točko).
$a_h(a_m)$	Cena procesiranja signalizacije pri domačem agentu (sidrni točki).
$\delta_U(\delta_D)$	Sorazmernostna konstanta signalizacije (dostave paketov).
η	Konstanta cene procesiranja pri domačem agentu.
ζ	Konstanta cene dodeljevanja pasovne širine pri sidrni točki.
k	Število dostopovnih točk znotraj domene ene sidrne točke
ω	Povprečno število mobilnih vozlišč povezanih z eno dostopovno točko
$\alpha(\beta)$	Utežnostni faktor seznama obiskovalcev (iskanja po usmerjevalni tabeli)
ε	Kvocient števila paketov, ki prečkajo domačega agenta, preden je usmerjevalna pot optimizirana.
$\Theta_{HA}(\Theta_{MAP})$	Cena procesiranja pri dostavi paketov pri domačem agentu (sidrni točki).

Tabela 5.2: Opis parametrov

$$h = 1 + \frac{N - 1}{N - k} \quad (5.2)$$

$$f = \frac{1}{hT} \quad (5.3)$$

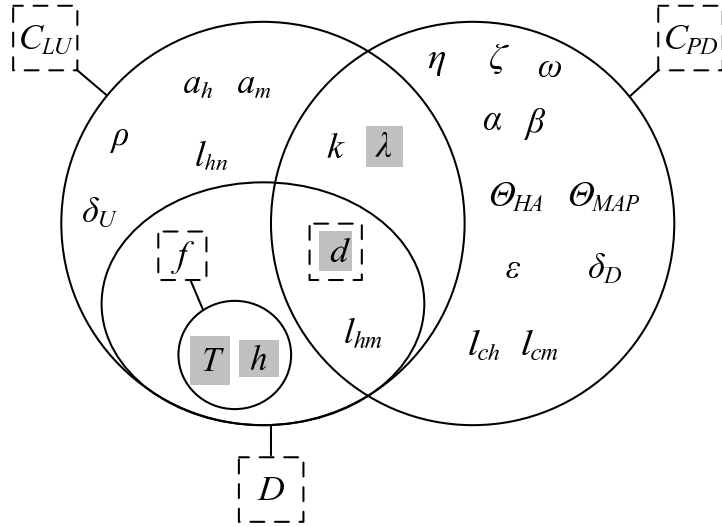
V vsaki od treh definicij metrike C_{PD} v tabeli 5.1 so vsebovani parametri l_{hm} , d in δ_D . Ker sta definiciji C_{PD} po [51] in [29] normalizirani na sejo, je potrebno enačbi pomnožiti s hitrostjo prihodov sej λ . λ tako postane še četrti parameter, prisoten v vseh treh definicijah. l_{hm} in δ_D sta tipično

konstanti. To pomeni, da sta d in h edini spremenljivki, ki se pojavljata v vseh treh definicijah C_{PD} . Nižji kot sta njuni vrednosti, nižja je cena C_{PD} .

Zakasnitve pri izročanju D ponavadi temeljijo na meritvah dejanskih zakasnitev sporočil [36, 45, 55, 56]. D je višji, če sporočila prečkajo več povezav ali če mobilno vozlišče zamenja več sidrnih točk, to je, če sta d in f višja. V [59, 99, 100] avtorji izrazijo D analitično. Pri tem definicija po [59] vključuje tako izročanje znotraj kot tudi zunaj domene izbrane sidrne točke. Skrajšano obliko njihove definicije podaja enačba 5.4. Upoštevajoč enačbi 5.3 in 5.4, je D odvisen od d , T , h in l_{hm} , kjer je vpliv l_{hm} tipično neznatn.

$$D = f_1\left(\frac{1}{T}\right)f_2(d) + f_3(f)f_4(d, l_{hm}) \tag{5.4}$$

Zgoraj opisane relacije ilustrira slika 5.1. Evaluacijske metrike v kvadratih s prekinjenimi črtami so funkcije obkroženih parametrov. Po naši študiji imajo najpomembnejši vpliv sivo obarvani parametri. Za ostale parametre avtorji bodisi predpostavijo, da so konstantni, bodisi jih ne upoštevajo vsi. Za šesto metriko, porazdelitev bremena po sidrnih točkah, analitičnega izraza ni na voljo, zato je ni moč enostavno vključiti v sliko.



Slika 5.1: Relacije med evaluacijskimi metrikami

Ugotovljeni štirje ključni parametri T , h , d in λ , se razlikujejo po pomembni lastnosti. T in λ sta spremenljivki, odvisni samo od uporabnika, zato na njiju ni moč vplivati z metodo izbiranja sidrnih točk. Spreminjanje d in h pa je ravno eden od namenov izbiranja sidrnih točk.

Glavna motivacija preteklih raziskav, ki upoštevajo uporabnikovo hitrost oziroma razmerje med uporabnikovo aktivnostjo in hitrostjo, je bila s spreminjanjem C_{PD} in C_{LU} , znižati skupno ceno C_{TOT} . Če je uporabnik na primer komunikacijsko zelo aktiven (velik λ) in se premika počasi (velik T), je izpostavljen veliki ceni C_{PD} in majhni ceni C_{LU} . Z izbiro bližjih sidrskih točk (majhen d), ki v splošnem pokrivajo manjša območja in vodijo do nizkega h , se C_{PD} lahko zniža za ceno potencialno zvišane C_{LU} . Če je zmanjšanje C_{PD} večje kot povišanje C_{LU} , potem se skupna cena C_{TOT} zmanjša.

5.2 dh -izboljšana izbira

V doktorski disertaciji se osredotočamo na izbiro sidrskih točk, ki bolje ustrezajo uporabnikom ne glede na njihova specifična parametra T in λ . V ta namen definiramo pojem dh -izboljšane izbire, ki opisuje relacijo med dvema algoritmoma za izbiro sidrskih točk. dh -izboljšana izbira je izbira, ki po enem od obeh algoritmov zagotavlja nižji d in hkrati višji h glede na drugega. Oznaka ' dh ' torej ne označuje produkta obeh metrik, ampak dejstvo, da sta obe metriki istočasno izboljšani. Spodaj je podana formalna definicija.

Definicija 3 (dh -izboljšana izbira sidrskih točk). *Naj pot mobilnega vozlišča sestoji iz določenega zaporedja dostopovnih točk. Na tej poti mobilno vozlišče izbere zaporedje sidrskih točk s povprečno razdaljo d in kvocientom izročanja h . Če je pri izbiri po algoritmu A d nižji in h višji v primerjavi z izbiro po algoritmu B , potem algoritem A izvaja dh -izboljšano izbiro glede na algoritem B .*

dh -izboljšana izbira vodi vedno k znižanju skupne cene. To pokažemo s primerom 1, kjer so uporabljene izvirne definicije cen [49, 50], na katere se raziskovalci pogosto sklicujejo.

Primer 1. *V [49, 50] sta C_{LU} in C_{PD} definirani kot:*

$$C_{LU} = \frac{a_m + 2d\delta_U}{T} + \frac{a_h + a_m + 2l_{hm}\delta_U + 2d\delta_U}{hT}, \quad (5.5)$$

$$C_{PD} = \eta\lambda + \zeta k\lambda(\alpha\omega k + \beta\log(k)) + (l_{hm} + d)\delta_D, \quad (5.6)$$

kjer sta d in h spremenljivki, ki sta odvisni od izbire sidrskih točk, T in λ sta parametra, odvisna od uporabnika, ostali parametri pa so odvisni od stanja omrežja. Za danega uporabnika v danem omrežju so parametri T in

λ ter parametri, odvisni od stanja omrežja konstantni. Enačbi 5.5 in 5.6 zato lahko skrajšano zapišemo kot:

$$C_{LU}(d, h) = K_1 + K_2(d + \frac{d}{h}) + K_3 \frac{1}{h}, \quad (5.7)$$

$$C_{PD}(d) = K_4 + K_5 d, \quad (5.8)$$

kjer so $K_1 - K_5$ konstante.

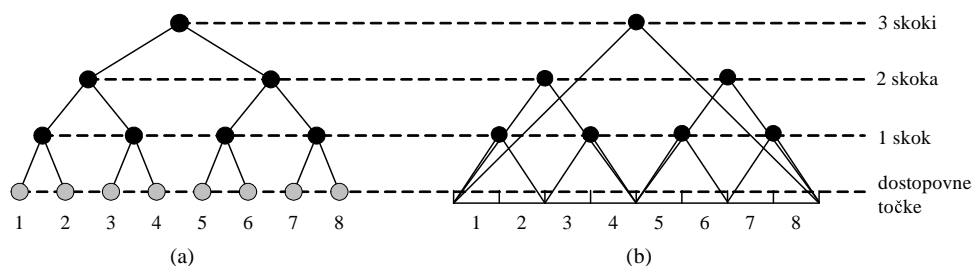
Predpostavimo, da mobilni terminal lahko izbira med sidrnima točkama A in B , za kateri velja $d_A < d_B$ in $h_A > h_B$. Iz enačb 5.7 in 5.8 sledi, da je $C_{LU-A} < C_{LU-B}$ in $C_{PD-A} < C_{PD-B}$, za poljubne $K_1 - K_5$. Če se zmanjšata obe delni ceni C_{LU} in C_{PD} , se po enačbi 5.1 posledično zmanjša tudi skupna cena C_{TOT} .

5.3 Topološke zahteve za dh -izboljšano izbiro

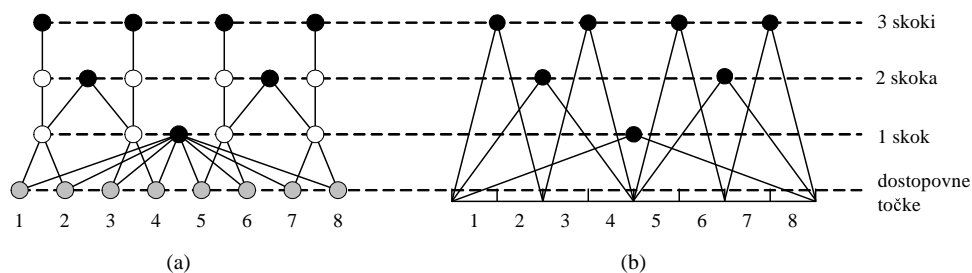
dh -izboljšane izbire sidrnih točk ni mogoče doseči v poljubni topologiji. Najpogosteje uporabljene drevesne topologije ne izpolnjujejo za to potrebnih zahtev. Zmožnost algoritmov, da sidra izbirajo na tak način, zato še ni bila raziskana.

Topološke zahteve za dh -izboljšano izbiro izpeljemo z analizo treh reprezentativnih primerov topologij, prikazanih na slikah 5.2, 5.3 in 5.4. Na levi strani je klasičen prikaz topologij, kot se uporablja v literaturi, medtem ko prikaz na desni bolj ustreza analizi d in h . Natančne povezave med posameznimi omrežnimi elementi so namreč nepomembne, zadostujeta le podatka o območju pokritja sidrnih točk in njihovih razdaljah do dostopovnih točk. Opomniti velja, da so povezave na levi samo ena od možnih konfiguracij, ki ustrezajo prikazom na desni.

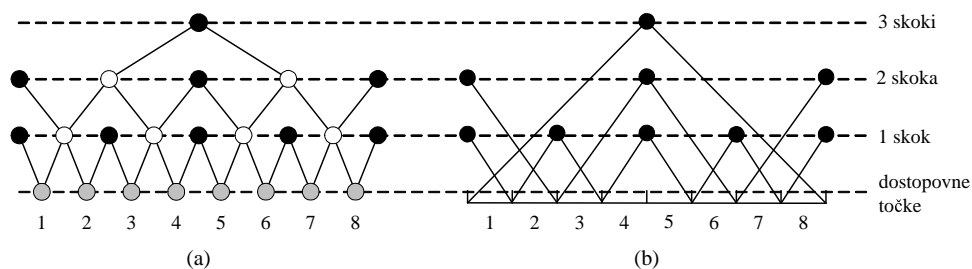
Slika 5.2 prikazuje drevesno topologijo, ki jo tipično uporabljajo raziskovalci s tega področja. Topologija na sliki 5.3 je diametralno nasprotna od drevesne. To pomeni, da najbolj oddaljene sidrne točke pokrivajo najmanjša območja in obratno. Ta tip topologije imenujemo obrnjeno drevo. Topologija na sliki 5.4 je zelo podobna drevesni, le da se območja pokritij sidrnih točk prekrivajo drugače. Imenujemo jo premaknjeno drevo. Za vse topologije predpostavimo, da so ovite okoli sebe (*angl. wrapped around*). To pomeni, da se v desno smer premikajoče mobilno vozlišče, potem ko zapusti skrajno desno dostopovno točko, poveže s skrajno levo dostopovno točko.



Slika 5.2: Drevo



Slika 5.3: Obrnjeno drevo



Slika 5.4: Premaknjeno drevo

Nobena od treh reprezentativnih topologij ne predstavlja realistične topologije. Kot pokažemo v nadaljevanju, je drevesna topologija mejni primer, pri katerem nikoli ne pride do dh -izboljšave. Obrnjeno drevo, še manj realistično kot navadno drevo, je nasprotni mejni primer, pri katerem se dh -izboljšava vedno zgodi. Premaknjeno drevo je primer topologije, ki leži med obema mejama. Poleg premaknjenega drevesa bi lahko izbrali tudi neomejeno število drugih topologij, ki ležijo v tem prostoru.

5.3.1 Vzorci vstopanja in izstopanja iz območij pokritij

Vsakokrat, ko se mobilno vozlišče premakne, lahko zapusti neko število območij, ki jih pokrivajo sidrne točke. To število lahko zavzame vrednosti med 0 in skupnim številom vseh sidrnih točk, ki pokrivajo trenutno dostopovno točko. Premik mobilnega vozlišča iz območja pokritja sidrne točke z razdaljo X označimo z M_X . Če mobilno vozlišče ostane znotraj pokritja te sidrne točke, je premik označen z $\overline{M_X}$. Premike mobilnih vozlišč v obravnavanih tri-nivojskih topologijah torej lahko označimo s trojko oblike $(\tilde{M}_1, \tilde{M}_2, \tilde{M}_3)$, kjer \tilde{M}_X predstavlja bodisi M_X , bodisi $\overline{M_X}$.

Če se mobilno vozlišče v topologijah na slikah 5.2, 5.3 in 5.4 premika samo med sosednjimi dostopovnimi točkami, je med njimi možnih 8 različnih prehodov. Ti prehodi v odvisnosti od vrste topologije prožijo različne trojke, prikazane v tabeli 5.3.

prehodi med dostop. točkami	trojke premikov		
	drevo	obr. drevo	premak. drevo
1-2, 3-4, 5-6, 7-8	$(\overline{M}_1, \overline{M}_2, \overline{M}_3)$	$(\overline{M}_1, \overline{M}_2, \overline{M}_3)$	$(\overline{M}_1, \overline{M}_2, \overline{M}_3)$
2-3, 6-7	$(\overline{M}_1, \overline{M}_2, \overline{M}_3)$	$(\overline{M}_1, \overline{M}_2, \overline{M}_3)$	$(\overline{M}_1, \overline{M}_2, \overline{M}_3)$
4-5	$(\overline{M}_1, \overline{M}_2, \overline{M}_3)$	$(\overline{M}_1, \overline{M}_2, \overline{M}_3)$	$(\overline{M}_1, \overline{M}_2, \overline{M}_3)$
1-8	(M_1, M_2, M_3)	(M_1, M_2, M_3)	$(\overline{M}_1, \overline{M}_2, \overline{M}_3)$

Tabela 5.3: Trojke premikov za tri obravnavane topologije

V tabeli 5.3, lahko opazimo več vzorcev. V drevesni topologiji zapusti mobilno vozlišče območje pokritja bolj oddaljene sidrne točke le, če hkrati zapusti tudi območja pokritij bližjih sidrnih točk. Pri obrnjenem drevesu je vzorec diametralno nasproten. Mobilno vozlišče zapusti območje pokritja bližje sidrne točke le, če hkrati zapusti tudi območja pokritij bolj oddaljenih sidrnih točk. Pri premaknjenem drevesu zapusti mobilno vozlišče največ enega od območij. Opisane tri vzorce formalno izrazimo z enačbami 5.9, 5.10 in 5.11.

- Drevo:

$$x > y \Rightarrow (M_x \Rightarrow M_y \wedge \overline{M}_y \Rightarrow \overline{M}_x) \quad (5.9)$$

- Obrnjeno drevo:

$$x > y \Rightarrow (M_y \Rightarrow M_x \wedge \overline{M}_x \Rightarrow \overline{M}_y) \quad (5.10)$$

- Premaknjeno drevo:

$$x \neq y \neq z \Rightarrow (M_x \Rightarrow \overline{M}_y \wedge \overline{M}_z) \quad (5.11)$$

Trojke premikov za drevo in obrnjeno drevo, ki so navedene v tabeli 5.3, so edine možne trojke, četudi bi se mobilno vozlišče premikalo med poljubnimi nesosednimi dostopnimi točkami. Enačbi 5.9 in 5.10 zato veljata v splošnem. Pri premaknjenem drevesu pa lahko pride tudi do drugih kombinacij, kar omejuje enačbo 5.11 na specifičen primer.

5.3.2 Poti mobilnih vozlišč in zaporedja sidrnih točk

Pot mobilnega vozlišča je definirana z zaporedjem obiskanih dostopnih točk. Označimo jo z vektorjem $j = (AP_1, AP_2, AP_3, \dots)$, kjer AP_1 , AP_2 in AP_3 označujejo obiskane dostopne točke. Zaporedje sidrnih točk na poti j pa podamo z vektorjem oblike $J = (\tilde{M}_x, \tilde{M}_x, \dots, \tilde{M}_y, \tilde{M}_y, \dots)$, kjer sta \tilde{M}_x in \tilde{M}_y definirana tako kot zgoraj. Medtem ko trojke premikov opisujejo vse vidne sidrne točke v danem trenutku, zaporedja sidrnih točk podajajo zaporedno izbrane sidrne točke na poti. Tako kot pri vektorju j je tudi pri J vsak premik označen s svojim elementom.

Za pojasnitev formalizma navajamo primer vektorjev j in J v drevesni topologiji s slike 5.2. Mobilno vozlišče obiše dostopne točke 4, 3, 4, 5, 6, 5, 6, 7 in 8. Pripadajoča pot v vektorski obliki je torej $j_{drevo} = (4, 3, 4, 5, 6, 5, 6, 7, 8)$. Na začetku izbere mobilno vozlišče sidrno točko na razdalji 1. Potem jo zamenja z drugo sidrno točko, prav tako na razdalji 1. Nazadnje izbere sidrno točko na razdalji 2. Iz tega sledi zapis zaporedja sidrnih točk $J_{drevo} = (\overline{M}_1, \overline{M}_1, M_1, \overline{M}_1, \overline{M}_1, \overline{M}_1, M_1, \overline{M}_2)$.

Iz vektorja J lahko izračunamo metriki d in h kot:

$$d = \frac{\sum_{i=1}^N X|_{J(i)}}{N} \quad (5.12)$$

$$h = \frac{N}{\sum_{i=1}^N |J(i)|} \quad (5.13)$$

V zgornjih dveh enačbah $J(i)$ predstavlja element \tilde{M}_X na i -tem položaju vektorja J . $X|_{J(i)}$ predstavlja razdaljo X pri istem elementu. $|J(i)|$ je 1, če je bila sidrna točka zamenjana, torej če je $J(i) = M_X$. V nasprotnem primeru je $|J(i)|$ 0. Skupno število premikov znaša N .

5.3.3 *dh*-izboljšanje v drevesih in obrnjenih drevesih

Z uporabo zgoraj uvedenih formalizmov v tem poglavju ugotovitev o možnostih za *dh*-izboljšano izbiro pri drevesu in obrnjenem drevesu zapišemo v obliki teorema.

Teorem 1 (*dh*-izboljšava v drevesih).

- (a) *Pri enolični izbiri v drevesu, po kateri eden od algoritmov vedno izbira enako oddaljene ali bližje sidrne točke kot drugi, dh-izboljšana izbira ni mogoča.*
- (b) *V drevesu je dh-izboljšano izbiro mogoče doseči le z dolgoročnim zaporedjem delov poti, ki imajo vsak zase enolično izbiro, v celoti pa izbira ni enolična.*

Dokaz.

- (a) Naj bosta J_{A_1} in J_{A_2} zaporedji sidrnih točk, ki ju na isti poti izbereta algoritma A_1 in A_2 . Za A_1 in A_2 naj velja, da so sidrne točke izbrane z A_2 vedno vsaj tako blizu, kot sidrne točke, izbrane z A_1 , formalno izraženo kot:

$$X|_{J_{A_1}(i)} \geq X|_{J_{A_2}(i)}, \text{ za } i \in I = \{1 \dots N\}, \quad (5.14)$$

kjer I predstavlja množico vseh zaporednih števil premikov od 1 do N . Iz tega po enačbah 5.12 in 5.14 logično sledi:

$$d_{A_1} \geq d_{A_2}. \quad (5.15)$$

Iz enačb 5.9 in 5.13 dobimo:

$$h_{A_1} \geq h_{A_2}, \quad (5.16)$$

kar dokazuje prvi del (a) teorema 1.

- (b) Če so sidrne točke, izbrane z algoritmom A_1 bodisi bližje, bodisi dlje v primerjavi s sidrnimi točkami, izbranimi z algoritmom A_2 , potem jih lahko razvrstimo v dve skupini, tako da velja:

$$X|_{J_{A_1}(i)} \geq X|_{J_{A_2}(i)}, \text{ za } i \in I' \subset I \quad (5.17)$$

$$X|_{J_{A_1}(i)} \leq X|_{J_{A_2}(i)}, \text{ za } i \in I'' \subset I, \quad (5.18)$$

pri čemer množici I' in I'' nimata skupnih elementov, v uniji pa tvorita množico I :

$$I' \cap I'' = \{\}$$
 (5.19)

$$I' \cup I'' = I$$
 (5.20)

Celotna pot j je s tem razdeljena na dva dela, ki ju označimo z j' in j'' , pripadajoče zaporedje izbranih sidrnih točk pa z J' in J'' . Kot sledi iz prvega dela (a) teorema 1, v nobenem od obeh delov poti ne pride do *dh*-izboljšanja:

$$d'_{A1} \geq d'_{A2}, h'_{A1} \geq h'_{A2}$$
 (5.21)

$$d''_{A1} \leq d''_{A2}, h''_{A1} \leq h''_{A2}$$
 (5.22)

Naj bosta N' in N'' velikosti množic I' in I'' . Po enačbi 5.12 je povprečna razdalja od mobilnega vozlišča do sidrnih točk, izbranih z algoritmom A_1 , v obeh delih poti skupaj enaka:

$$d_{A1} = \frac{\sum_{i=1}^{N'} X|_{J'_{A1}(i)} + \sum_{i=1}^{N''} X|_{J''_{A1}(i)}}{N' + N''}$$

$$d_{A1} = \frac{N' \frac{\sum_{i=1}^{N'} X|_{J'_{A1}(i)}}{N'} + N'' \frac{\sum_{i=1}^{N''} X|_{J''_{A1}(i)}}{N''}}{N}$$

$$d_{A1} = \frac{N' d'_{A1} + N'' d''_{A1}}{N}$$
 (5.23)

Za algoritmom A_2 po analogiji dobimo:

$$d_{A2} = \frac{N' d'_{A2} + N'' d''_{A2}}{N}$$
 (5.24)

Nadalje po enačbi 5.13 za algoritmoma A_1 in A_2 izračunamo povprečna kvocienta izročanja v obeh delih poti skupaj:

$$h_{A1} = \frac{N' + N''}{\sum_{i=1}^{N'} |J'_{A1}(i)| + \sum_{i=1}^{N''} |J''_{A1}(i)|}$$

$$h_{A1} = \frac{N}{\frac{\sum_{i=1}^{N'} |J'_{A1}(i)|}{N'} + \frac{\sum_{i=1}^{N''} |J''_{A1}(i)|}{N''}}$$

$$h_{A1} = \frac{N}{\frac{N'}{h'_{A1}} + \frac{N''}{h''_{A1}}} \quad (5.25)$$

in

$$h_{A2} = \frac{N}{\frac{N'}{h'_{A2}} + \frac{N''}{h''_{A2}}} \quad (5.26)$$

Za dh -izboljšano izbiro sidrnih točk na celotni poti j zahtevamo:

$$d_{A1} > d_{A2} \quad (5.27)$$

$$h_{A1} < h_{A2} \quad (5.28)$$

Po vstavitvi enačb 5.23 in 5.24 v neenačbo 5.27 in po preureditvi dobimo:

$$N'(d'_{A1} - d'_{A2}) > N''(d''_{A2} - d''_{A1}). \quad (5.29)$$

Vrednosti $\Delta d' = (d'_{A1} - d'_{A2})$ in $\Delta d'' = (d''_{A2} - d''_{A1})$ predstavljata razliko v povprečnih razdaljah do sidrnih točk, ki jih dosemeta oba algoritma. Pri tem se d' nanaša na prvi, d'' pa na drugi del poti. Po enostavni preureditvi dobimo:

$$\frac{N'}{N''} > \frac{\Delta d''}{\Delta d'} \quad (5.30)$$

Po vstavitvi enačb 5.25 in 5.26 v neenačbo 5.28 in po preureditvi dobimo:

$$N' \left(\frac{1}{h'_{A1}} - \frac{1}{h'_{A2}} \right) > N'' \left(\frac{1}{h''_{A2}} - \frac{1}{h''_{A1}} \right) \quad (5.31)$$

Obratna vrednost metrike h , ki jo definiramo kot $f_A = \frac{1}{h}$, predstavlja število zamenjav sidrnih točk na dostopovno točko. Razliki $\Delta f'_A = \left(\frac{1}{h'_{A1}} - \frac{1}{h'_{A2}} \right)$ in $\Delta f''_A = \left(\frac{1}{h''_{A2}} - \frac{1}{h''_{A1}} \right)$ predstavljata razliko tega števila znotraj posameznega dela poti glede na algoritem. Glede na definiciji 5.21 in 5.22 sta vrednosti $\Delta f'_A$ in $\Delta f''_A$ negativni, kar vpliva na usmerjenost neenačaja pri preurejanju neenačbe 5.31. Tako dobimo:

$$\frac{N'}{N''} < \frac{\Delta f''_A}{\Delta f'_A} \quad (5.32)$$

Neenačbi 5.30 in 5.32 lahko združimo v spodnjo neenačbo:

$$\frac{\Delta f''_A}{\Delta f'_A} > \frac{N'}{N''} > \frac{\Delta d''}{\Delta d'} \quad (5.33)$$

Algoritem A_2 izvaja glede na algoritem A_1 dh -izboljšano izbiro če in samo če je zadoščeno neenačbi 5.33. V nasprotnem primeru, če algoritem A_1 izvaja dh -izboljšano izbiro glede na algoritem A_2 , velja neenačba:

$$\frac{\Delta f''_A}{\Delta f'_A} < \frac{N'}{N''} < \frac{\Delta d''}{\Delta d'} \quad (5.34)$$

Večja kot je razlika med razmerjema $\frac{\Delta f''_A}{\Delta f'_A}$ in $\frac{\Delta d''}{\Delta d'}$, večja je dh -izboljšava. Za mejni pogoj $\frac{\Delta f''_A}{\Delta f'_A} = \frac{N'}{N''} = \frac{\Delta d''}{\Delta d'}$, je dh -izboljšava nična. Če ne velja nobena od neenačb 5.33 oziroma 5.34, do dh -izboljšave ni prišlo.

S tem je dokazan drugi del (b) teorema 1.

□

Teorem 2 (dh -izboljšava v obrnjenih drevesih).

- (a) *Pri enolični izbiri v obrnjenem drevesu, po kateri eden od algoritmov vedno izbira enako oddaljene ali bližje sidrne točke kot drugi, je izbira vedno dh -izboljšana.*

- (b) V obrnjenem drevesu je izbira, ki ni dh -izboljšana, mogoče doseči le z dolgoročnim zaporedjem delov poti, ki imajo vsak zase enolično izbira, v celoti pa izbira ni enolična.

Dokaz. Teorema 1 in 2 sta si diametralno nasprotna. Dokaz teorema 2, je zato analogen dokazu teorema 1 z diametralno nasprotnimi neenačaji. \square

Doseganje dh -izboljšane izbire v drevesu po metodi iz dela (b) teorema 1 je le teoretična možnost in ni primerna iz dveh razlogov. Prvič, na posameznih delih poti, kjer je izbira enolična, le-ta ni dh -izboljšana. Prednosti zato niso takojšnje in so posledično nepomembne, še posebej, če so ti deli poti dolgi. Drugič, potrebni pogoji za doseg takšne izboljšave so nerealistični, kot je to pokazano s primerom 2.

Primer 2. V drevesih je dh -izboljšana izbira mogoča dolgoročno, če izbira na skupni poti ni enolična in če veljajo specifična razmerja med $\Delta f'_A$, $\Delta f''_A$, $\Delta d'$ in $\Delta d''$.

Neenačba 5.33 v razširjeni obliki postane:

$$\frac{\left(\frac{1}{h''_{A2}} - \frac{1}{h''_{A1}}\right)}{\left(\frac{1}{h'_{A1}} - \frac{1}{h'_{A2}}\right)} > \frac{N'}{N''} > \frac{(d''_{A2} - d''_{A1})}{(d'_{A1} - d'_{A2})} \quad (5.35)$$

Če se mobilno vozlišče giblje naključno, lahko predpostavimo, da je kvocient izročanja h proporcionalen povprečni velikosti območja pokritja sidrskih točk. V drevesni topologiji območje pokritja narašča eksponentno z b^d , kjer osnova b predstavlja število otrok, eksponent d pa povprečno razdaljo do izbranih sidrskih točk. Dobimo torej:

$$h = Ab^d, \quad (5.36)$$

kjer je konstanta A odvisna od vzorca gibanja mobilnega vozlišča. Upoštevajoč to relacijo, neenačba 5.35 postane:

$$\frac{\left(\frac{1}{b^{d''_{A2}}} - \frac{1}{b^{d''_{A1}}}\right)}{\left(\frac{1}{b^{d'_{A1}}} - \frac{1}{b^{d'_{A2}}}\right)} > \frac{N'}{N''} > \frac{(d''_{A2} - d''_{A1})}{(d'_{A1} - d'_{A2})} \quad (5.37)$$

Kot sledi iz neenačbe 5.37, morata biti za čim višjo dh -izboljšavo, števec na levi strani neenačbe in imenovalec na desni strani neenačbe čim višja, imenovalec na levi strani in števec na desni strani pa čim nižja. Te zahteve

niso skladne, ker na primer visoka razlika $d'_{A_1} - d'_{A_2}$ da tudi visoko razliko $\frac{1}{b^{d'_{A_1}}} - \frac{1}{b^{d'_{A_2}}}$. Možna je sicer delna kompenzacija te relacije, tako da d'_{A_1} in d'_{A_2} zavzameta visoki vrednosti, kar zniža vrednosti $\frac{1}{b^{d'_{A_1}}}$ in $\frac{1}{b^{d'_{A_2}}}$ ter posledično razliko $\frac{1}{b^{d'_{A_1}}} - \frac{1}{b^{d'_{A_2}}}$, vendar se podoben problem v inverzni obliki pojavi tudi za pot j'' .

Z drugimi besedami, na poti j' morajo biti sidrne točke z obema algoritmom izbrane zelo daleč, medtem ko morajo biti na poti j'' zelo blizu. Zaradi zahtevanih razlik to velja še posebej za algoritem A_1 . Tako drastično spreminjanje razdalj do izbranih sidrnih točk pa ni niti realistično niti optimalno za katerikoli algoritem.

5.3.4 *dh*-izboljšanje v topologijah z delnim prekrivanjem

Čeprav so diametralno nasprotna, delijo drevesa in obrnjena drevesa pomembno skupno lastnost: ne vsebujejo območij sidrnih točk, ki bi se delno prekrivala. To pomeni, da pri vsakem prekrivanju ena od sidrnih točk v celoti pokrije območje druge sidrne točke.

Enolična izbira enako oddaljenih ali bližjih sidrnih točk je pri topologijah z delnim prekrivanjem območij sidrnih točk lahko bodisi *dh*-izboljšana, bodisi ni *dh*-izboljšana. Za ilustracijo je podan primer 3.

Primer 3. Mobilno vozlišče naj se giblje po dveh različnih neskončnih poteh v premaknjenem drevesu. Na poti A se mobilno vozlišče premika venomer v isti smeri, medtem ko se na poti B premika dolgoročno v eno smer, vendar se občasno premakne tudi v drugo. Naj bodo A_1 , A_2 in A_3 trije enostavni algoritmi za izbiranje sidrnih točk, ki le-te izbirajo na konstantnih razdaljah 1, 2 in 3. Vzorci premikanja, skupaj s spremembami trenutno izbranih sidrnih točk, so podani v tabelah 5.4 in 5.5. V obeh tabelah je poudarjena perioda vzorca gibanja. Tabela 5.6 prikazuje povprečne razdalje d in povprečne kvociente izročanja h v označeni periodi.

dost. točka	8	1	2	3	4	5	6	7	8	1
J_{A_1}	M_1	$\overline{M_1}$	M_1	$\overline{M_1}$	M_1	$\overline{M_1}$	M_1	$\overline{M_1}$	M_1	$\overline{M_1}$
J_{A_2}	$\overline{M_2}$	M_2	$\overline{M_2}$	M_2	$\overline{M_2}$	M_2	$\overline{M_2}$	M_2	$\overline{M_2}$	M_2
J_{A_3}	$\overline{M_3}$	M_3	$\overline{M_3}$	M_3	$\overline{M_3}$	M_3	$\overline{M_3}$	M_3	$\overline{M_3}$	M_3

Tabela 5.4: Menjave sidrnih točk na poti A

Za vzorec gibanja na poti B je z algoritmom A_1 dosežena *dh*-izboljšana izbira glede na algoritem A_2 . To je tudi edini primer *dh*-izboljšane izbire

d. točka	8	1	2	3	2	3	4	5	4	5	...
J_{A1}	$\overline{M_1}$	$\overline{M_1}$	$\overline{M_1}$	$\overline{M_1}$	$\overline{M_1}$	$\overline{M_1}$	$\overline{M_1}$	$\overline{M_1}$	$\overline{M_1}$	$\overline{M_1}$...
J_{A2}	$\overline{M_2}$	$\overline{M_2}$	$\overline{M_2}$	$\overline{M_2}$	$\overline{M_2}$	$\overline{M_2}$	$\overline{M_2}$	$\overline{M_2}$	$\overline{M_2}$	$\overline{M_2}$...
J_{A3}	$\overline{M_3}$	$\overline{M_3}$	$\overline{M_3}$	$\overline{M_3}$	$\overline{M_3}$	$\overline{M_3}$	$\overline{M_3}$	$\overline{M_3}$	$\overline{M_3}$	$\overline{M_3}$...

...	6	7	6	7	8	1	8	1	2	3
...	$\overline{M_1}$	$\overline{M_1}$	$\overline{M_1}$	$\overline{M_1}$	$\overline{M_1}$	$\overline{M_1}$	$\overline{M_1}$	$\overline{M_1}$	$\overline{M_1}$	$\overline{M_1}$
...	$\overline{M_2}$	$\overline{M_2}$	$\overline{M_2}$	$\overline{M_2}$	$\overline{M_2}$	$\overline{M_2}$	$\overline{M_2}$	$\overline{M_2}$	$\overline{M_2}$	$\overline{M_2}$
...	$\overline{M_3}$	$\overline{M_3}$	$\overline{M_3}$	$\overline{M_3}$	$\overline{M_3}$	$\overline{M_3}$	$\overline{M_3}$	$\overline{M_3}$	$\overline{M_3}$	$\overline{M_3}$

Tabela 5.5: Menjave sidrnih točk na poti B

	pot A		pot b	
	d	h	d	h
A_1	1	2	1	4
A_2	2	4	2	2.67
A_3	3	8	3	5.33

Tabela 5.6: Povprečne razdalje d in povprečni kvocienti izročanja h

v obravnavanem primeru. Algoritem A_1 tako na primer ne doseže dh -izboljšane izbire v primerjavi z algoritmom A_3 pri istem vzorcu gibanja oziroma ne doseže dh -izboljšane izbire v primerjavi z algoritmom A_2 pri vzorcu gibanja na poti A.

Premaknjeno drevo je bolj sorodno drevesu kot obrnjenemu drevesu. V premaknjenem obrnjenem drevesu, bi bilo parov algoritmov z doseženo dh -izboljšano izbiro več.

5.4 Testiranje drevesnih in realističnih topologij

Struktura realističnih topologij ni drevesna in vsebuje delna prekrivanja območij sidrnih točk. dh -izboljšana izbira sidrnih točk je zato v realističnih topologijah dosegljiva. V tem poglavju preverimo z uporabo računalniških simulacij pravilnost te trditve. Kot merilo zmožnosti za dh -izboljšano izbiro, uporabimo primerjavo para algoritmov z največjo poznano razliko v d in h . To sta algoritem za izbiro najbolj oddaljene sidrne točke in prediktivni algoritem [70, 102], definirana v poglavju 3.2.2.

5.4.1 Opis simulacijskih topologij

Primerjamo tri različne tipe topologij: drevesa, generirane modele realističnega interneta in merjene modele realističnega interneta. Vse primerjane topologije so predstavljene na nivoju avtonomnih sistemov. V merjenih topologijah so poslovne relacije med avtonomnimi sistemi že poznane. Ostala dva tipa te informacije ne vsebujeta. Tipe poslovnih relacij zato pri drevesih in generiranih modelih določimo z našim algoritmom, ki smo ga definirali v poglavju 4.6.

Drevesa so sestavljena iz l nivojev vozlišč z enim samim vozliščem na vrhu hierarhije. Vsako vozlišče, razen na najnižjem nivoju, se razveji na b otrok. Skupno število vozlišč v takšni topologiji je:

$$n = \sum_{i=0}^{l-1} b^i \quad (5.38)$$

Število možnih velikosti drevesnih topologij je omejeno. V nadaljevanju so predstavljeni rezultati za velikosti od 40 do 29524 vozlišč, s parametrom b 3 in 4.

Za generacijo modelov realističnega interneta smo izbrali Inet [90], stopenjski generator na nivoju avtonomnih sistemov. Velikost generiranih topologij je lahko poljubno določena. Rezultati, predstavljeni v nadaljevanju, veljajo za velikosti med 21 in 27646 vozlišč.

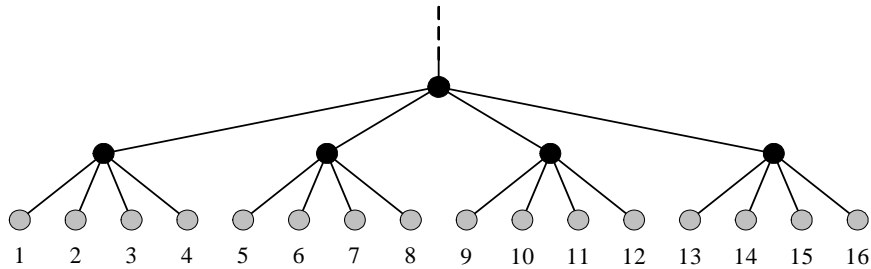
Merjene modele smo pridobili od organizacije CAIDA [74]. Analiziramo 5 topologij iz let 2004 do 2008, za vsako leto eno. Njihove velikosti so med 16301 in 27646 vozlišč.

Topologije za potrebe analize upravljanja mobilnosti označimo po metodi, opisani v poglavju 3.4.1 z nekaj manjšimi spremembami. Dostopovne točke so prisotne samo na najnižjem nivoju, nivoju strank, kar ustreza načinu razporejanja dostopovnih točk v drevesnih topologijah, ki ga uporabljajo raziskovalci s tega področja. Sidrne točke se nahajajo na vseh petih nivojih, tudi na nivoju strank. Razdalja med dostopovno in sidrno točko je omejena na 2 hopa. Pri višjih razdaljah pridemo do enakih zaključkov, vendar so le-ti manj zanesljivi zaradi majhnega števila menjav sidrnih točk, ki je lahko celo nič, če je omejitev razdalje nastavljena previsoko.

5.4.2 Vzorci gibanja mobilnih vozlišč

Modeliranje vzorcev gibanja mobilnega vozlišča je velik izziv, ker študirane topologije ne vsebujejo informacije o geografski bližini dostopovnih točk.

Za drevesa raziskovalci ponavadi predpostavijo enostavno geografsko razporeditev, ki direktno sovпада s pogosto uporabljeno grafično predstavitvijo topologije. Slika 5.5 prikazuje primer drevesa z $b = 4$, kjer je dostopna točka 3 geografsko sosednja točkama 2 in 4, točka 11 točkama 10 in 12 itd.



Slika 5.5: Neskončno drevo z $b = 4$

Analitična izpeljava, ki sledi, velja za primer, ko se mobilno vozlišče giblje levo in desno z enako verjetnostjo. Od vsake dostopovne točke v drevesu je topološko najbližja dostopovna točka vedno oddaljena 2 hopa. Najprej analiziramo b takšnih točk, ki imajo skupno sidrno točko oddaljeno en hop. Na sliki 5.5 so to na primer dostopovne točke 5, 6, 7 in 8. Za ostale dostopovne točke je postopek izpeljave enak. Mobilno vozlišče je lahko povezano s katerokoli od b dostopovnih točk in se lahko premakne v katerokoli od dveh možnih smeri. To dá $2b$ možnih premikov, kar je v študiranim primeru enako 8. Samo 2 od $2b$ možnih premikov sprožita menjavo dostopovne točke, ki je od trenutne oddaljena več kot 2 hopa. Pri ostalih premikih je nova dostopovna točka od trenutne oddaljena 2 hopa. Verjetnost, da bo nova dostopovna točka od trenutne oddaljena 2 hopa, je torej:

$$p_2 = \frac{2b - 2}{2b} = \frac{b - 1}{b} \quad (5.39)$$

Verjetnosti za liha števila hopov so enake nič, ker v drevesih ni parov dostopovnih točk s takšno oddaljenostjo. Za ostale verjetnosti sodih števil hopov je izpeljava podobna. Za verjetnosti oddaljenosti 4 in 6 hopov naprimer dobimo enačbi:

$$p_4 = \frac{b - 1}{b^2} \quad (5.40)$$

$$p_6 = \frac{b - 1}{b^3} \quad (5.41)$$

Porazdelitev verjetnosti razdalje med dvema obiskanima dostopnima točkama za poljubno drevo je torej:

$$p_x = \begin{cases} 0 & \text{za } x = 2n - 1, n \in \mathbb{N} \\ \frac{b-1}{b^{\frac{x}{2}}} & \text{za } x = 2n, n \in \mathbb{N} \end{cases} \quad (5.42)$$

V splošni topologiji je mogoča katerakoli razdalja med dvema dostopnima točkama. Enačbo 5.42 zato posplošimo z razširitvijo verjetnosti na lihe razdalje. Po tem koraku je potrebna še normalizacija, ki obdrži vsoto vseh verjetnosti 1.

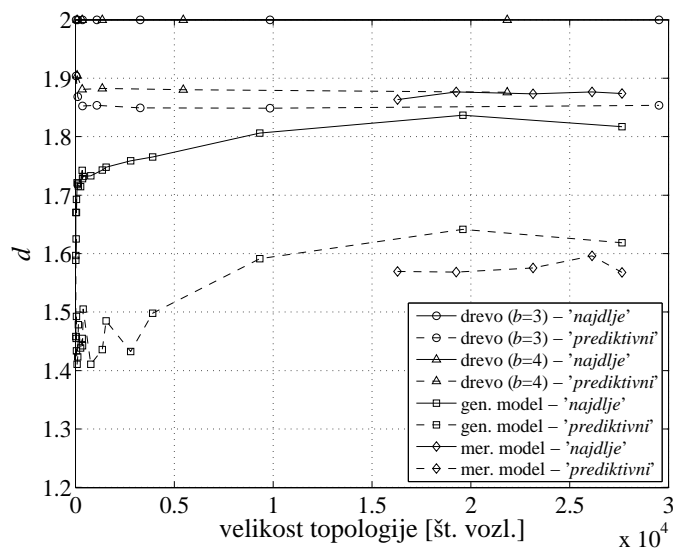
$$p_x = \frac{b-1}{b^{\frac{x}{2}}} \frac{1}{\sum_x \frac{b-1}{b^{\frac{x}{2}}}} \quad \text{za } x \in \mathbb{N} \quad (5.43)$$

Pri simulacijah z nedrevesnimi topologijami smo b nastavili na 4. Ko je razdalja glede na verjetnostno porazdelitev izbrana, mobilno vozlišče naključno izbira med dostopnimi točkami na tej razdalji.

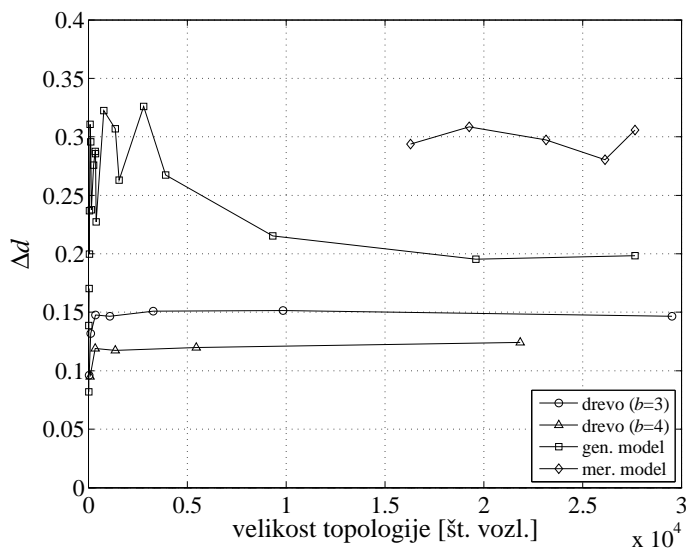
5.4.3 Simulacijski rezultati

Najprej analiziramo povprečno razdaljo d med mobilnim vozliščem in izbranimi sidrnimi točkami. Povprečne vrednosti za algoritma '*najdlje*' in '*prediktivni*' so prikazane na sliki 5.6. Če je v drevesih uporabljen algoritem '*najdlje*', so izbrane sidrne točke vedno na največji dovoljeni razdalji, to je 2 hopa. V realističnih topologijah pa razdalja variira, ker so razdalje dostopnih točk od sidrne točke, ki jih pokriva, različne. Poleg tega sidrne točke na razdalji 2 niso nujno vedno na razpolago za izbiro s trenutne dostopovne točke. Posledično je d za '*najdlje*' manjši v realističnih topologijah. Ko je uporabljen algoritem '*prediktivni*', se glede na '*najdlje*' d izboljša tako v drevesnih kot tudi v realističnih topologijah. Razlika v povprečni razdalji Δd med obema algoritmoma je prikazana na sliki 5.7. Kljub v osnovi višjemu d pri uporabi '*najdlje*' v drevesih, je izboljšanje bolj izrazito v realističnih topologijah, še posebej v merjenih.

Povprečni kvocienti izročanja h so prikazani na sliki 5.8. V drevesih, še posebej tistih z visoko razvejanostjo b , so doseženi višji kvocienti izročanja kot v realističnih topologijah. Vendar se z uporabo algoritma '*prediktivni*' v primerjavi z algoritmom '*najdlje*' kvocient h v realističnih topologijah izboljša, medtem ko se v drevesnih zniža. Razlike Δh so prikazane na sliki 5.9. Rezultati potrjujejo ugotovitve iz poglavja 5.3. Z algoritmom '*prediktivni*' se je izbira sidrnih točk v realističnih topologijah dh -izboljšala,

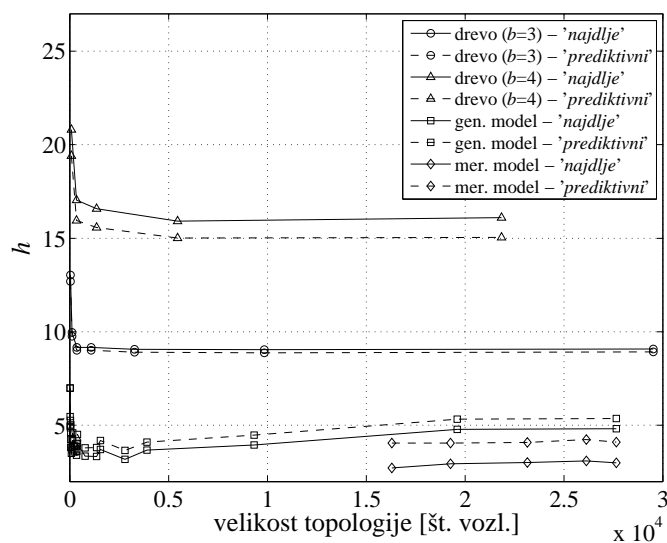


Slika 5.6: Povprečne razdalje

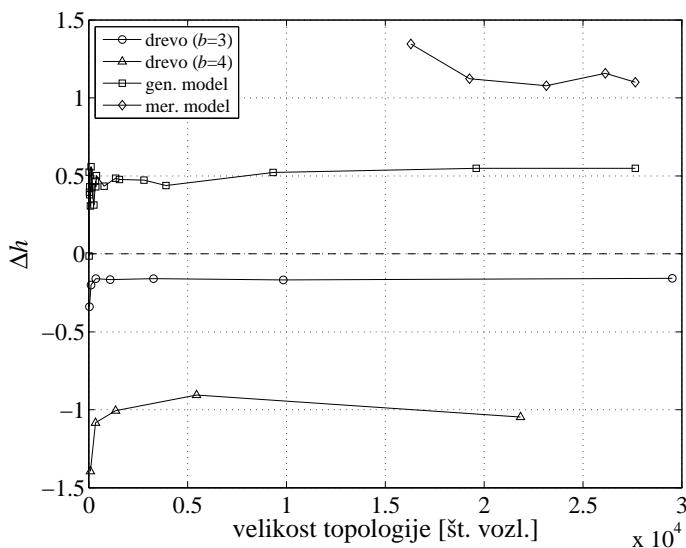


Slika 5.7: Razlike v povprečnih razdaljah

medtem ko v drevesih do dh -izboljšave v primerjavi z algoritmom 'najdlje', ni prišlo niti z najboljšim poznanim algoritmom.



Slika 5.8: Povprečni kvocient izročanja



Slika 5.9: Razlike v povprečnih kvocientih izročanja

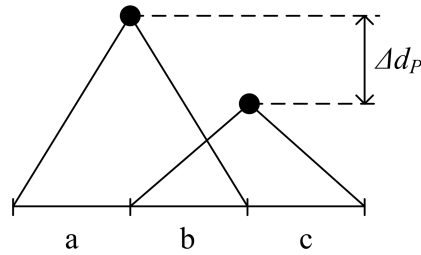
5.5 Topološke lastnosti za visoko dh -izboljšano izbiro

Sodeč po rezultatih iz prejšnjega poglavja različni tipi topologij vodijo do različnih stopenj dh -izboljšane izbire sidrnih točk. V tem poglavju to od-

visnost nadalje analiziramo. Definiramo novo metriko, prekrivno metriko, za katero verjamemo, da dobro odraža lastnosti, relevantne za stopnjo dh -izboljšanja. Predpostavke preverimo s simulacijami. Najprej primerjamo prekrivne metrike topologij iz prejšnjega poglavja. Nato modificiramo drevesne in generirane realistične topologije, tako da razlika med algoritmoma 'najdlje' in 'prediktivni' narašča, ter opazujemo spremembe v prekrivni metriki.

5.5.1 Prekrivna metrika

Intuitivno bodo višje dh -izboljšave dosežene v topologijah z večjo količino parov sidrnih točk, za katere velja, da se območje bližje sidrne točke razteza preko območja bolj oddaljene sidrne točke in še čez. Ilustracija takšnega para je podana na sliki 5.10. Poleg velikega števila takih parov, morajo biti za visoko dh -izboljšavo visoke tudi razlike v razdaljah sidrnih točk (Δd_P), število dostopovnih točk, ki so skupne obema sidrnima točkama (b) in število dostopovnih točk, ki jih pokriva samo bližja sidrna točka (c).

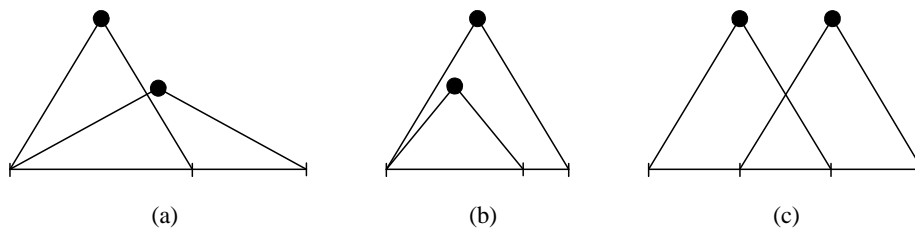


Slika 5.10: Parametri, ki vplivajo na prekrivno metriko O

Upoštevajoč zgoraj opisane povezave, uvedemo prekrivno metriko O :

$$O = \frac{\sum_{i=1}^N \Delta d_{P_i} \frac{c_i}{a_i + b_i + c_i} \frac{b_i}{a_i + b_i}}{N - M} \quad (5.44)$$

V enačbi 5.44 je parameter c normaliziran z velikostjo območja pokritja obeh sidrnih točk skupaj, parameter b pa z velikostjo območja pokritja bolj oddaljene sidrne točke. Topologija sestoji iz N parov sidrnih točk z vsaj eno skupno dostopovno točko. Za vsak tak par so parametri Δd_P , normaliziran c in normaliziran b zmnoženi, produkti pa seštet v števcu. Parameter M predstavlja število parov sidrnih točk z $a = 0$ in $c > 0$, to je parov, kjer vse dostopovne točke, ki jih pokriva bolj oddaljena sidrna točka, pokriva tudi bližja sidrna točka in ne obratno (slika 5.11(a)).



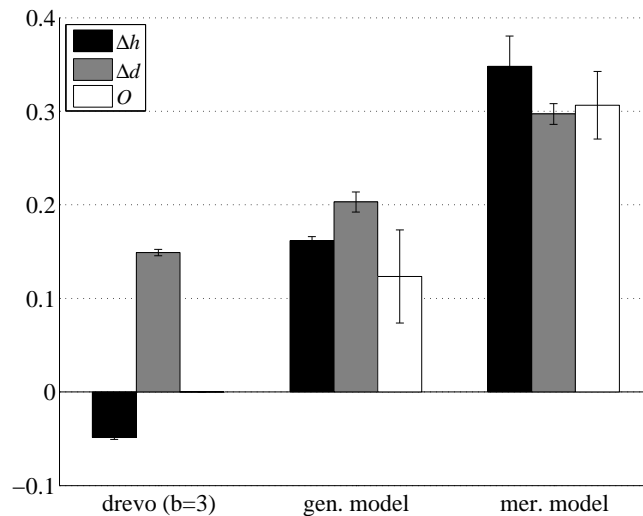
Slika 5.11: Tipični primeri parov sidrnih točk

Po definiciji 5.44 je O realno število na intervalu $[0, \infty)$. Mejna vrednost 0 pomeni, da izbira sidrnih točk v dani topologiji ne more biti dh -izboljšana, medtem ko gre v topologijah, kjer vedno pride do dh -izboljšane izbire, O proti neskončnosti. To pomeni, da je $O = 0$ v drevesih in $O \rightarrow \infty$ v obrnjenih drevesih. V drevesih za vse pare sidrnih točk velja $c = 0$, kar pomeni, da vse dostopovne točke, ki jih pokriva bližja sidrna točka, pokriva tudi bolj oddaljena sidrna točka (slika 5.11(b)). V obrnjenih drevesih pa za vse pare sidrnih točk velja $a = 0$ in $c > 0$, kot je ilustrirano na sliki 5.11(a). Opozoriti velja, da O zavzame vrednost 0 tudi za topologije, kjer sta za vseh N parov sidrnih točk obe sidrni točki enako oddaljeni (slika 5.11(c)). Za vmesne topologije, ki imajo lastnosti drugačne od zgoraj naštetih, je O pozitivno realno število in je predvidoma večje, če so pogoji za dh -izboljšavo boljši. Za primer premaknjenega drevesa s slike 5.4 iz poglavja 5.3, je O enak 0,0167.

5.5.2 Prekrivna metrika v drevesih in realističnih topologijah

Slika 5.12 prikazuje povprečne vrednosti in standardne deviacije metrik Δh , Δd in O za drevesa z $b = 3$ ter za generirane in merjene realistične topologije velikosti med 9331 in 29524 vozlišč. Spremembe Δh , Δd in O na tem intervalu velikosti niso velike, kot je prikazano z relativno majhnimi standardnimi deviacijami. Povprečne velikosti vseh treh skupin so primerljive in znašajo 19682 vozlišč za drevesa, 18862 vozlišč za generirane in 22500 vozlišč za merjene realistične topologije. Zaradi boljše primerljivosti so vrednosti v grafu na sliki 5.12 predstavljene relativno.

V drevesih je povprečna vrednost Δh negativna, Δd pa pozitivna. Do dh -izboljšanja torej ni prišlo, zato je vrednost metrike O za drevesa po pričakovanjih enaka 0. V realističnih topologijah sta Δh in Δd pozitivni. Lastnosti merjenih topologij vodijo k višjim dh -izboljšavam, ker sta vre-

Slika 5.12: Δh , Δd in O za topologije primerljivih velikosti

dnosti Δh in Δd višji. Posledično je višja tudi prekrivna metrika O , kar potrjuje naše predpostavke.

Iz tega dela študije sledi še zanimiva postranska ugotovitev. Med analiziranimi topologijami ravno merjene topologije, ki najboljše modelirajo lastnosti resničnega Interneta, izkazujejo najboljše možnosti za dh -izboljšano izbiro sidrskih točk. To je vzpodbudno, saj je s tem dobro upravičen razvoj novih, boljših algoritmov za izbiranje sidrskih točk.

5.5.3 Prekrivna metrika v modificiranih topologijah

Metoda za modificiranje

Ideja za uporabljeno metodo modificiranja topologij je povzeta po ideji evlucijskih algoritmov. V splošni shemi evlucijskega algoritma je vsaka generacija podvržena selekciji, rekombinaciji, mutaciji in evalvaciji [103]. Ker operiramo neposredno s topologijami, bi rekombinacija zahtevala neko posebno metodo deljenja topologij na polovice in nato združevanja polovic. Poleg tega je cilj modificiranja topologij poiskati topološke lastnosti, ki vodijo k boljšim pogojem za dh -izboljšano izbiro in ne iskanje optimalne topologije. Zato smo korak rekombinacije izpustili, ostale korake pa obdržali.

Začetno populacijo sestavlja poljubno izbrana izvorna topologija. Zaradi kompleksnosti metode so lahko modificirane samo relativno majhne topologije, približno do reda 1000 vozlišč. Merjene realistične topologije so

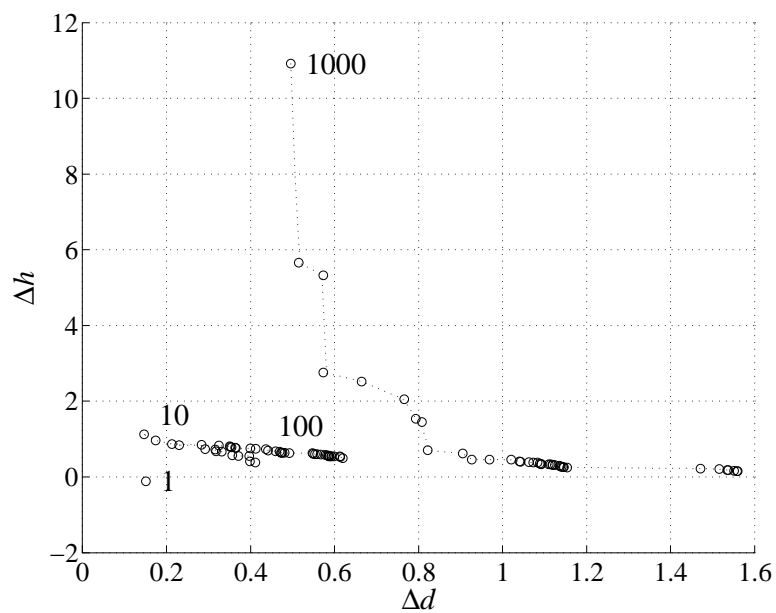
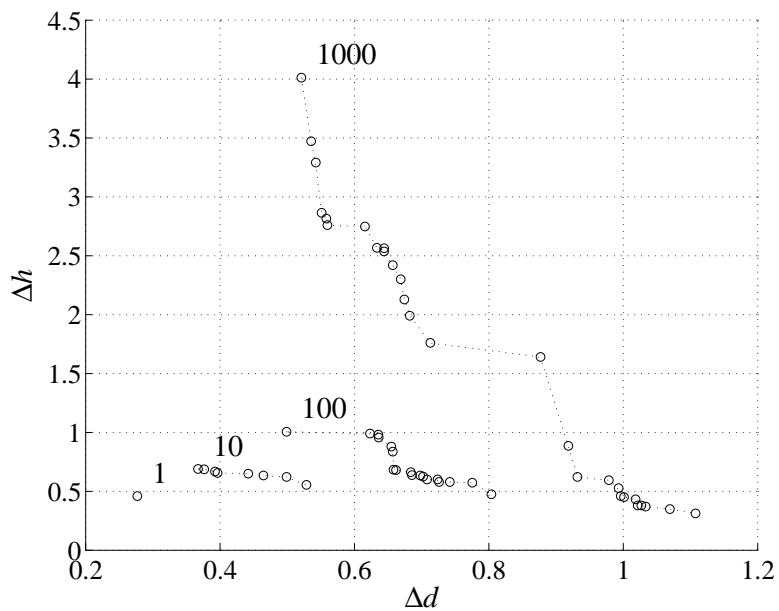
za modifikacijo prevelike. Izvorna topologija je s simulacijami evalvirana v smislu Δh in Δd , kot je opisano v poglavju 5.4. Sledi iterativni proces kreiranja novih generacij. Vsaka topologija v populaciji postane starš 5 otrokom z neposrednim kopiranjem in mutacijo. Mutacija je izvedena z naključnim prevezovanjem do 20% vseh vozlišč na način, da se število vseh povezav ohrani in da ostane topologija povezana v eno celoto. Vsak otrok je na novo označen in evalviran. Med starši in otroki so izbrane topologije za naslednjo generacijo v dveh korakih. V prvem so eliminirane dominirane rešitve, to so topologije, za katere sta Δh in Δd nižja kot za poljubno drugo topologijo. S tem korakom zagotovimo izločevanje za dh -izbiro manj ugodnih topologij, medtem ko ugodnejše ostajajo. Namen drugega koraka je omejitev kompleksnosti. Izvede se, če populacija presega 100 osebkov. V tem primeru so naključno eliminirane rešitve, ki so si med seboj najbolj podobne v smislu evklidske razdalje opazovanih metrik Δh in Δd . Eliminacija poteka tako dolgo, dokler v populaciji ni natanko 100 osebkov. Preostale topologije predstavljajo novo generacijo.

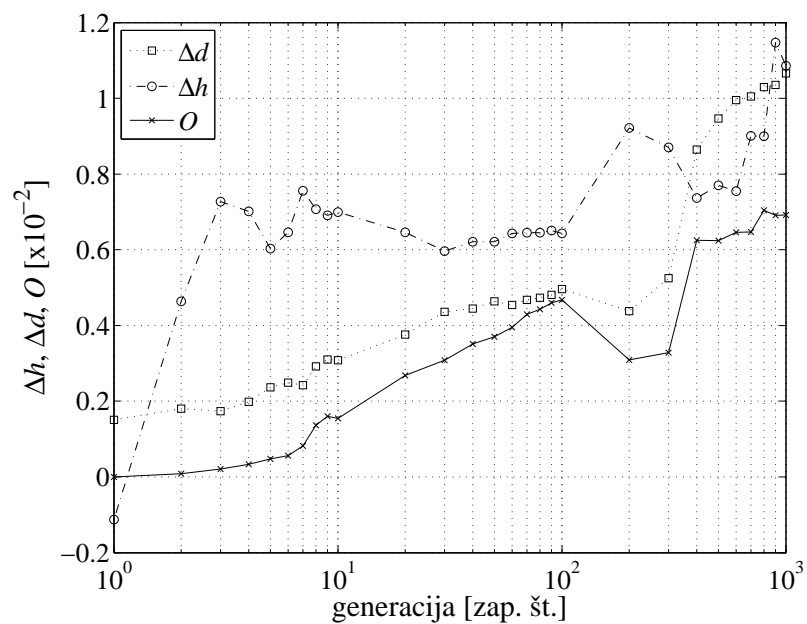
Rezultati

Modificirali smo drevesa in generirane realistične topologije različnih velikosti in različnih maksimalnih oddaljenosti sidrnih točk. Ne glede na izbrane parametre so zaključki enaki. Na spodnjih slikah so prikazani rezultati za največjo študirano velikost 1093 vozlišč in maksimalno oddaljenost sidrnih točk 2 hopa. Obe izvorni topologiji, drevo in generirani model, sta bili modificirani skozi 1000 generacij.

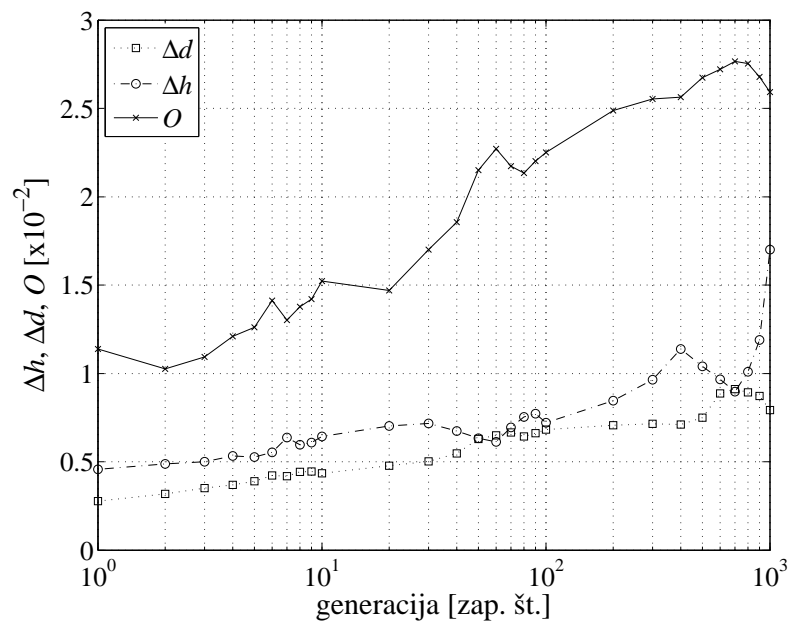
Sliki 5.13 in 5.14 prikazujeta napredek evolucije v smislu Δh in Δd . Vsaka točka v grafu predstavlja en osebek oziroma topologijo, črtkane črte pa povezujejo osebeke iste generacije. Zaporedne številke generacij so podane poleg črt. Rast vseh treh parametrov O , Δh in Δd je logaritemska in korelirana. To potrjujeta grafa na slikah 5.15 in 5.16, ki podajata povprečne vrednosti O , Δh in Δd posameznih generacij.

Rezultati potrjujejo naše predpostavke o vplivu vzorcev prekrivanja območij sidrnih točk na dh -izboljšano izbiro. Višja prekrivna metrika O v splošnem vodi do višjih vrednosti Δh in Δd . Kljub temu je mogoče v grafih opaziti tudi nekaj vzorcev, kjer relacija med O , Δh in Δd ni tako očitna. Iz tega sledi, da obstajajo verjetno tudi druge lastnosti, ki jih prekrivna metrika ne zajema, in ki vplivajo na zmožnost dh -izboljšane izbire sidrnih točk.

Slika 5.13: Δh in Δd glede na število generacij - izvor drevoSlika 5.14: Δh in Δd glede na število generacij - izvor generirani model



Slika 5.15: Povprečne vrednosti O , Δh in Δd - izvor drevo



Slika 5.16: Povprečne vrednosti O , Δh in Δd - izvor generirani model

Poglavje 6

Sklep

V internetu je mobilnost smiselno upravljati hierarhično. Z ločitvijo na globalni in lokalni nivo dosežemo zmanjšanje režijske signalizacije in hitrejše izročanje. Skupaj s prednostmi se v hierarhičnem načinu odprejo tudi novi izzivi. Eden izmed njih je posledica možnosti, da mobilno vozlišče na lokalnem nivoju lahko izbira med več entitetami za upravljanje mobilnosti. Način izbiranja vpliva na učinkovitost protokola.

Rdeča nit doktorske disertacije je bila učinkovita izbira sidrnih točk, entitet za upravljanje mobilnosti na lokalnem nivoju po protokolu HMIPv6. Iskali smo načine za boljšo izbiro in analizirali zmožnosti izboljšanja izbire v različnih tipih omrežij. V nadaljevanju povzamemo pomembne zaključke in naštejemo nekaj smernic za nadaljnje delo.

6.1 Pregled vsebine z glavnimi zaključki

V uvodnem delu, v prvem in v drugem poglavju, smo podali umestitev problematike v širše področje, našteali iztočnice za raziskovalno delo in predstavili osnove delovanja komunikacijskih protokolov, na katerih so zasnovane v disertaciji obravnavane rešitve. Jedro doktorske disertacije z izvirnimi prispevki je podano v treh delih: v tretjem, četrtem in petem poglavju. Vsako poglavje zase predstavlja zaključeno celoto, hkrati pa so vsebinsko povezana. Četrto poglavje je motivirano z izsledki tretjega, peto pa temelji na izsledkih tretjega in četrtega. V vsakem od treh delov smo opisali zadnje dosežke z ožjega področja, predstavili idejo našega prispevka ter podali metodologijo in rezultate.

V tretjem poglavju smo predlagali nov pristop k izbiranju sidrnih točk, ki temelji na predvidevanju gibanja mobilnega vozlišča. Predpostavili smo, da je ponavljajoče vzorce uporabnikovega gibanja moč izkoristiti za boljšo izbiro sider. Razvili smo tri tipe algoritmov, ki izkoriščajo informacijo o tem, koliko časa bodo prisotne trenutno vidne sidrne točke in kako oddaljene bodo v povprečju. Na podlagi te informacije izberejo algoritmi najprimernejše sidro. Prediktivni način izbiranja smo preizkusili na dveh tipih topologij, sintetičnih in realističnih.

Simulacijski rezultati na sintetičnih topologijah potrjujejo, da se z metodo predvidevanja poveča učinkovitost izbire sider. Izbrane sidrne točke so v povprečju bližje, mobilno vozlišče pa jih zamenja manj pogosto. Posledično so signalizacijske zakasnitve manjše, usmerjevalne poti znotraj domen sidrnih točk bližje optimalnim, breme na sidrnih točkah je bolje razporejeno, signalizacija izven domen pa je manj pogosta. Za izvedbo v praksi je pomembna tudi ugotovitev, da za povečanje učinkovitosti ni potrebno popolnoma natančno predvidevanje. Vsakršna informacija o prihodnjem stanju zagotavlja povečanje učinkovitosti prediktivne izbire. Popolnejša kot je informacija, boljša je učinkovitost.

Kot prvi na področju smo učinkovitost izbiranja sidrnih točk ovrednotili tudi na realističnih topologijah. V ta namen je bilo zaradi kompleksne hierarhije potrebno vpeljati posebno metodo za označevanje topologij, po kateri se določi vlogo vozlišč in območja pokritij sidrnih točk. Ocenili smo, kakšne so prednosti ponujanja storitev sidrnih točk v več avtonomnih sistemih. Rezultati kažejo, da se uporabnik lahko izogne prekomernim menjavam sidrnih točk. Prediktivni način je učinkovitejši od osnovnega načina izbiranja najbolj oddaljenih sidrnih točk tudi v realističnih topologijah. Izkaže pa se, da so razlike v določenih scenarijih realističnih omrežij lahko manjše. Iz tega sledi, da topologija omrežja vpliva na učinkovitost algoritmov za izbiranje sidrnih točk. Ugotovitev je motivirala naše nadaljnje raziskovalno delo.

Za raziskavo vpliva strukture omrežja na učinkovitost algoritmov za izbiro sidrnih točk je bilo potrebno razširiti bazo simulacijskih topologij. Ker naša metoda za določanje območja pokritij sidrnih točk temelji na poslovnih relacijah med avtonomnimi sistemi, le-te pa so bile poznane le v merjenih omrežjih, je bilo potrebno poiskati način za njihovo določanje v splošni sintetični topologiji.

Zaradi odsotnosti primernih rešitev na tem področju smo v četrtem poglavju predlagali metodo za modeliranje poslovnih relacij v internetu. Vhod predlaganega algoritma je surova topologija brez informacije o obstoju po-

slovnih relacij, izhod pa topologija, ki to informacijo vsebuje. Na podlagi študije merjenih omrežij, smo problem določanja relacij v naključnih grafih najprej formalizirali kot problem TRR. Problem zajema pet zahtev, ki temeljijo na karakterističnih lastnostih interneta. Predlagani algoritem tipe relacij v danem grafu označi tako, da zadosti vsem petim zahtevam. Pri tem na vходу ni potrebno zagotoviti nikakršnih dodatnih informacij o strukturnih lastnostih poslovnih relacij, ker jih določa že sam model. Za izvedbo algoritma je bilo potrebno izpeljati analitični izraz za porazdelitev relacij enakovreden-z-enakovrednim. Izraz temelji na empiričnih vzorcih realnega interneta.

Da bi preverili kakovost modela, smo primerjali merjene topologije z že določenimi relacijami in iste topologije z relacijami, določenimi po naši metodi. Uporabili smo več kriterijev: dve hierarhični dekompoziciji, ki temeljita predvsem na relacijah enakovreden-z-enakovrednim oziroma na relacijah ponudnik-stranka, porazdelitev razdalj med vozlišči po poteh brez dolin in ujemanje posameznih relacij. Rezultati so podobni v vseh pogledih, kar kaže na verodostojnost metode. Algoritem smo aplicirali tudi na topologijah, sintetiziranih s stopenjskim generatorjem. Rezultati hierarhičnih dekompozicij kažejo rahla odstopanja od rezultatov, doseženih na merjenih topologijah, iz česar sklepamo, da stopenjski generatorji strukturnih lastnosti ne reproducirajo najbolje.

Predlagana metoda je uporabna za splošne modele realnih omrežij in ni omejena na raziskave upravljanja mobilnosti. Podobnih rešitev ni veliko, primerjava pa pokaže, da so prednosti naše metode predvsem relativna enostavnost izvedbe in ločitev problema generiranja omrežij od določanja relacij, zaradi česar se področji lahko razvijata ločeno.

V petem poglavju smo raziskovali vpliv strukture omrežij na izbiro sidrnih točk. V ta namen smo najprej analizirali medsebojni vpliv evaluacijskih metrik in njihovih parametrov, ki jih uporabljajo raziskovalci s tega področja. Pokazali smo, da so v množici parametrov le štiri med seboj neodvisne spremenljivke, ki imajo pomemben vpliv na evaluacijske metrike. To so povprečen čas povezave z isto dostopovno točko T , hitrost prihodov sej λ , povprečno število menjav dostopovnih točk na menjavo sidrne točke h in povprečna razdalja med mobilnim vozliščem in sidrno točko d . Ostali parametri so večinoma privzeti kot konstante. Parametra T in λ sta odvisna le od uporabnika, na d in h pa lahko vplivamo z načinom izbiranja sidrnih točk. Pokazali smo, da hkratno izboljšanje obeh parametrov d in h vedno vodi k izboljšanju najpogosteje uporabljenih metrik cen, neodvisno od vrednosti uporabniških parametrov T in λ . Hkratna izboljšava parame-

trov d in h mora biti zato ključni cilj algoritmov za izbiranje sidrnih točk. Izbiro sider, ki vodi do hkratne izboljšave, smo označili z novim pojmom, imenovanim dh -izboljšana izbira.

Analitično in s simulacijami smo pokazali, da imajo drevesne topologije, ki jih uporablja velika večina raziskovalcev, vgrajeno omejitev, ki onemogoča dh -izboljšano izbiro. Poudarek dosedanjih raziskav je bil zato na iskanju optimalnega razmerja med izboljšanjem enega izmed parametrov d oziroma h za ceno poslabšanja drugega, na način, ki skupaj z danima uporabniškima parametroma T in λ vodi k izboljšanju skupne cene. Realistične topologije tovrstnih omejitev nimajo, zato je smiselno iskati algoritme, kot smo jih predlagali v tretjem poglavju. Glavni motiv prediktivnih algoritmov je ravno maksimizacija dh -izboljšanja.

Primerjali smo tudi generirane in merjene realistične topologije, pri čemer smo za generirane uporabili model poslovnih relacij s četrtega poglavja. Primerjava je pokazala, da je v primeru, ko pride do dh -izboljšanja, velikost le-tega odvisna od topologije omrežja. Vpliv strukture omrežja smo podrobneje analizirali. Poiskali smo vzorce, ki vodijo k boljšim pogojem za dh -izboljšavo in njihovo prisotnost analitično izrazili. Rezultat izraza imenujemo prekrivna metrika. Pri višjih vrednostih prekrivne metrike je moč doseči višje dh -izboljšave. Verodostojnost izpeljanega izraza smo preverili s postopno modifikacijo topologij, pri čemer smo opazovali razmerje med dh -izboljšanjem in vrednostjo prekrivne metrike. Rezultati potrjujejo, da vzorci, opisani s prekrivno metriko, pravilno opisujejo za dh -izboljšavo ugodne topološke lastnosti, hkrati pa nakazujejo, da obstajajo tudi druge lastnosti, ki jih prekrivna metrika ne zajame. Prekrivno metriko je moč uporabljati pri odločitvah o primernih algoritmih v danih omrežjih, uporabna pa je tudi kot kriterij pri načrtovanju omrežij, katerih cilj je učinkovita podpora mobilnim komunikacijam.

6.2 Smernice za nadaljnje delo

Za razvoj učinkovitih algoritmov za izbiro sidrnih točk je potrebno upoštevati realne razmere, sicer so rezultati lahko nerelevantni za uporabo v praksi. To je velik izziv, saj principi upravljanja hierarhične mobilnosti v internetu še niso v splošni rabi in realne razmere torej niso poznane. Kljub temu je relevanco raziskav na tem področju mogoče izboljšati z modeli, ki so že na voljo.

Verjamemo, da prispeva analiza na realističnih omrežjih k večji verodo-

stojnosti rezultatov. Zaradi boljše raziskanih hierarhičnih razmerij smo izhajali iz omrežij avtonomnih sistemov. V naslednjem koraku je verodostojnost mogoče nadalje izboljšati z upoštevanjem topologij na nivoju usmerjevalnikov. Pri tem bi bilo najbolje, če bi relacije med avtonomnimi sistemi ostale poznane.

Posebej zahtevno je tudi natančno upoštevanje gibanja mobilnih vozlišč. Obstoječi modeli, ki opisujejo mobilnost vozlišč v geografskem smislu in topološki modeli realnih omrežij ne zadostujejo. Primanjkujejo namreč podatki, ki bi oba tipa modelov povezali z geografsko lokacijo vozlišč v realni topologiji. Rešitev tega problema bi pomembno prispevala k širšemu področju upravljanja mobilnosti v internetu, ne samo k hierarhičnemu.

Prediktivni način izbire sidrnih točk, ki smo ga predlagali, temelji na predpostavki, da so postopki za predvidevanje že izvedeni. Iskanje dejanskih postopkov za napovedovanje prihodnjih dogodkov iz preteklih vzorcev je deloma že raziskano. Ni pa še raziskano, ali je z upoštevanjem lastnosti, specifičnih za mobilno okolje, predikcijo moč izboljšati.

Kadar je mogoče, je smiselno iskati zakonitosti, ki niso odvisne od študirane okolja in torej verodostojnost modelov na njih ne vpliva. V zadnjem delu smo tako identificirali lastnosti, ki vplivajo na zmožnost za *dh*-izboljšano izbiro. Lastnosti so neodvisne od topologije ali modela gibanja mobilnega vozlišča. Iskanje drugih lastnosti takega tipa bi poglobilo razumevanje možnosti in omejitev na področju hierarhičnega upravljanja mobilnosti.

Literatura

- [1] H. Soliman. *Mobile IPv6 mobility in a wireless Internet*. Addison Wesley, April 2004. [citirano na str. v, 1, 3, 4, 5, 9, 12, 14]
- [2] D. Saha, A. Mukherjee, I.S. Misra, and M. Chakraborty. Mobility support in IP: A survey of related protocols. *IEEE Network Magazine*, 18(6):34–40, November/December 2004. [citirano na str. 1]
- [3] I.F. Akyildiz, J. Xie, and S. Mohanty. A survey on mobility management in next generation all-IP based wireless systems. *IEEE Wireless Communications Magazine*, 11(4):16–28, August 2004. [citirano na str. 1, 4]
- [4] M. Ratola. Which layer for mobility? - comparing mobile IPv6, HIP and SCTP. HUT T-110.551 Seminar on Internetworking Sjökölla, 2004. [citirano na str. 1, 3]
- [5] H. Schulzrinne and E. Wedlund. Application-layer mobility using SIP. *ACM SIGMOBILE Mobile Computing and Communications Review*, 4(3):47–57, July 2000. [citirano na str. 1, 2, 3]
- [6] T.T. Kwon, M. Gerla, S. Das, and S. Das. Mobility management for VoIP: Mobile IP vs. SIP. *IEEE Wireless Communications Magazine*, 9(5):66–75, October 2002. [citirano na str. 1, 3]
- [7] J. Rosenberg, H. Schulzrinne, G. Camarillo, A. Johnston, J. Peterson, R. Sparks, M. Handley, and E. Schooler. SIP: Session initiation protocol. RFC 3261, June 2002. [citirano na str. 1, 2]
- [8] C. Perkins. IP mobility support for IPv4. RFC 3344, August 2002. [citirano na str. 1, 2]
- [9] D. Johnson, C. Perkins, and J. Arkko. Mobility support in IPv6. RFC 3775, June 2004. [citirano na str. 1, 2, 11]

- [10] R. Stewart, Q. Xie, K. Morneault, C. Sharp, H. Schwarzbauer, T. Taylor, I. Rytina, M. Kalla, L. Zhang, and V. Paxson. Stream control transmission protocol. RFC 2960, October 2000. [citirano na str. 1, 3]
- [11] W. Xing, H. Karl, A. Wolisz, and H. Müller. M-SCTP: Design and prototypical implementation of an end-to-end mobility concept. In *Proc. 5th Intl. Workshop The Internet Challenge: Technology and Applications*, Berlin, Germany, October 2002. [citirano na str. 1, 3]
- [12] R. Moskowitz and P. Nikander. Host identity protocol (HIP) architecture. RFC 4423, May 2006. [citirano na str. 1, 3]
- [13] E. Wedlund and H. Schulzrinne. Mobility support using SIP. In *Proc. 2nd ACM international workshop on Wireless mobile multimedia WoWMoM*, pages 76–82, Seattle, Washington, USA, August 1999. [citirano na str. 2, 3]
- [14] 3rd Generation Partnership Project. IP multimedia call control protocol based on session initiation protocol (SIP) and session description protocol (SDP). Technical Specification Group Core Network and Terminals, 3GPP TS 24.229 V7.7.0, Stage 3, Release 7, March 2007. [citirano na str. 2]
- [15] J.-W. Jung, R. Mudumbai, D. Montgomery, and H.-K. Kahng. Performance evaluation of two layered mobility management using mobile IP and session initiation protocol. In *Proc. IEEE GLOBECOM*, volume 3, pages 1190–1194, San Francisco, USA, December 2003. [citirano na str. 2, 3]
- [16] A. Dutta, S. Das, T. Chiba, H. Yokota, A. Idoue, K. D Wong, and H. Schulzrinne. Comparative analysis of network layer and application layer IP mobility protocols for IPv6 networks. In *Proc. Wireless Personal Multimedia Communications WPMC*, San Diego, CA, USA, September 2006. [citirano na str. 3]
- [17] Q. Wang, M.A. Abu-Rgheff, and A. Akram. Design and evaluation of an integrated mobile IP and SIP framework for advanced handoff management. In *Proc. IEEE International Conference on Communications ICC*, volume 7, pages 3921–3925, Paris, France, June 2004. [citirano na str. 3]
- [18] Q. Wang and M.A. Abu-Rgheff. Mobility management architectures based on joint mobile IP and SIP protocols. *IEEE Wireless Communications Magazine*, 13(6):68–76, December 2006. [citirano na str. 3]
- [19] M. Riegel and M. Tuexen. Mobile SCTP. Internet-Draft, October 2006. [citirano na str. 3]

- [20] M. Atiquzzaman and A. Reaz. Survey and classification of transport layer mobility management schemes. In *Proc. 16th Annual IEEE International Symposium on Personal Indoor and Mobile Radio Communications PIMRC*, Berlin, Germany, September 2005. [citirano na str. 3]
- [21] J. Pang, A. Akella, A. Shaikh, B. Krishnamurthy, and S. Seshan. On the responsiveness of DNS-based network control. In *Proc. 4th ACM SIGCOMM conference on Internet measurement*, pages 21–26, Taormina, Sicily, Italy, October 2004. [citirano na str. 3]
- [22] A.T. Campbell, J. Gomez, S. Kim, Z. Turanyi, C.-Y. Wan, Z.R. Turányi, and A. Valkó. Comparison of IP micro-mobility protocols. *IEEE Wireless Communications Magazine*, 9(1):72–82, February 2002. [citirano na str. 4, 5]
- [23] D. Vali, S. Paskalis, A. Kaloxylos, and L. Merakos. An efficient micro-mobility solution for SIP networks. In *Proc. IEEE GLOBECOM*, volume 6, pages 3088–3092, San Francisco, USA, December 2003. [citirano na str. 4]
- [24] R. Ramjee, K. Varadhan, L. Sargarelli, S.R. Thuel, S.-Y. Wang, and T. La Porta. HAWAII: A domain-based approach for supporting mobility in wide-area wireless networks. *IEEE/ACM Transactions on Networking*, 10(3):396–410, June 2002. [citirano na str. 4]
- [25] A.G. Valkó. Cellular IP: A new approach to internet host mobility. *ACM SIGCOMM Computer Communication Review*, 29(1):50–65, January 1999. [citirano na str. 4, 5]
- [26] E. Fogelstroem, A. Jonsson, and C. Perkins. Mobile IPv4 regional registration. Internet-Draft, October 2006. [citirano na str. 4]
- [27] H. Soliman, C. Castelluccia, K. El Malki, and L. Bellier. Hierarchical mobile IPv6 mobility management (HMIPv6). RFC 4140, August 2005. [citirano na str. 4, 5, 17, 26, 30]
- [28] A. Misra, S. Das, A. Dutta, A. McAuley, and S.K. Das. IDMP-based fast handoffs and paging in IP based 4G mobile networks. *IEEE Communications Magazine*, 40(3):138–145, March 2002. [citirano na str. 4]
- [29] S. Pack, T. Kwon, and Y. Choi. A performance comparison of mobility anchor point selection schemes in hierarchical mobile IPv6 networks. *Elsevier Computer Networks*, 51(6):1630–1642, April 2007. [citirano na str. 4, 6, 25, 77, 78, 79]
- [30] R. Koodli. Fast handovers for mobile IPv6. RFC 4068, July 2005. [citirano na str. 5]

- [31] H. Yokota and G. Dommety. Mobile IPv6 fast handovers for 3G CDMA networks. Internet-Draft, March 2007. [citirano na str. 5]
- [32] V.S. Kaulgud and S.A. Mondal. Exploiting multihoming for low latency handoff in heterogeneous networks. In *Proc. 8th International Conference on Telecommunications ConTEL*, volume 1, pages 49–55, Zagreb, Croatia, June 2005. [citirano na str. 5]
- [33] A. Vilhar, R. Novak, and G. Kandus. Multihoming benefits for network mobility in HAP networks. In *Proc. 3rd Advanced Satellite Mobile Systems Conference ASMS*, Herrsching am Ammersee, Munich, Germany, May 2006. [citirano na str. 5]
- [34] R. Hsieh and A. Seneviratne. A comparison of mechanisms for improving mobile IP handoff latency for end-to-end TCP. In *Proc. 9th annual international conference on Mobile computing and networking*, pages 29–41, San Diego, CA, USA, September 2003. [citirano na str. 5]
- [35] R. Hsieh, Z.G. Zhouand, and A. Seneviratne. S-MIP: A seamless handoff architecture for mobile IP. In *Proc. IEEE INFOCOM*, volume 3, pages 1774–1784, San Francisco, USA, March 2003. [citirano na str. 5]
- [36] E. Natalizio, A. Scicchitano, and S. Marano. Mobility anchor point selection based on user mobility in HMIPv6 integrated with fast handover mechanism. In *Proc. IEEE Wireless Communications and Networking Conference WCNC*, volume 3, pages 1434– 1439, New Orleans, LA, USA, March 2005. [citirano na str. 5, 6, 26, 77, 80]
- [37] L. Niesink. A comparison of mobile IP handoff mechanisms. In *Proc. 6th Twente Student Conference on IT*, Enschede, Netherlands, February 2007. [citirano na str. 5]
- [38] E. Perera, V. Sivaraman, and A. Senevirtane. Survey on network mobility support. *ACM Mobile Computer Communication Review*, 8(2):7–19, April 2004. [citirano na str. 5, 20]
- [39] V. Devarapalli, R. Wakikawa, A. Petrescu, and P. Thubert. Network mobility (NEMO) basic support protocol. RFC 3963, January 2005. [citirano na str. 5, 20, 21, 29]
- [40] R. Wakikawa, S. Koshiba, and K. Uehara. ORC: Optimized route cache management protocol for network mobility. In *Proc. 10th International Conference on Telecommunications ICT*, volume 2, pages 1194–1200, Tahiti Papeete, French Polynesia, February 2003. [citirano na str. 5, 21]

- [41] J. Na, J. Choi, S. Cho, and C. Kim. A unified route optimization scheme for network mobility. In *Lecture Notes in Computer Science LNCS*, volume 3260, pages 29–38. Springer-Verlag, September 2004. [citirano na str. 5, 21]
- [42] E. Perera, A. Seneviratne, and V. Sivaraman. OptiNets: An architecture to enable optimal routing for network mobility. In *Proc. International Workshop on Wireless Ad-hoc Networks*, pages 68–72, Oulu, Finland, May 2004. [citirano na str. 5, 21]
- [43] A. Vilhar and R. Novak. Home agent placement optimization for HAP based network mobility. In *Proc. International Workshop on Satellite and Space Communications IWSSC*, pages 873–877, Siena, Italy, September 2005. [citirano na str. 5]
- [44] Y. Takagi, H. Ohnishi, K. Sakitani, K. Baba, and S. Shimojo. Route optimization methods for network mobility with mobile IPv6. *IEICE Transactions on Communications*, E87-B(3):480–489, March 2004. [citirano na str. 5, 21]
- [45] Y. Xu, H.C.J. Lee, and V.L.L. Thing. A local mobility agent selection algorithm for mobile networks. In *Proc. IEEE International Conference on Communications ICC*, volume 2, pages 1074–1079, Anchorage, Alaska, USA, May 2003. [citirano na str. 6, 27, 29, 77, 80]
- [46] A. Huszak and S. Imre. Agent selection algorithm in hierarchical mobile networks. In *Proc. 5th Conference on Telecommunications ConfTele*, Tomar, Portugal, April 2005. [citirano na str. 6, 26, 77]
- [47] X. Hu, J. Song, and M. Song. An adaptive mobility anchor point selection algorithm for hierarchical mobile IPv6. In *Proc. IEEE International Symposium on Communications and Information Technology ISCIT*, volume 2, pages 1148–1151, Beijing, China, October 2005. [citirano na str. 6, 27, 77, 78]
- [48] M.-H. Liu and C.-C. Yang. A multicast extension to HMIPv6 with efficient MAP selection. In *Proc. IEEE 62nd Semiannual Vehicular Technology Conference VTC*, volume 2, pages 816–820, Dallas, Texas, USA, September 2005. [citirano na str. 6, 27, 29, 77]
- [49] J. Xie and I.F. Akyildiz. A distributed dynamic regional location management scheme for mobile IP. In *Proc. IEEE INFOCOM*, volume 2, pages 1069–1078, 2002. [citirano na str. 6, 28, 77, 78, 81]
- [50] J. Xie and I.F. Akyildiz. A novel distributed dynamic location management scheme for minimizing signaling costs in mobile IP. *IEEE Transactions on Mobile Computing*, 1(3):163–175, July-September 2002. [citirano na str. 6, 77, 78, 81]

- [51] S. Pack, M. Nam, T. Kwon, and Y. Choi. An adaptive mobility anchor point selection scheme in hierarchical mobile IPv6 networks. *Elsevier Computer Communications*, 29(16):3066–3078, October 2006. [citirano na str. 6, 27, 77, 78, 79]
- [52] S. Pack, M. Nam, and Y. Choi. Design and analysis of optimal multi-level hierarchical mobile IPv6 networks. *Springer Wireless Personal Communications*, 36(2):95–112, January 2006. [citirano na str. 6, 28, 77, 78]
- [53] T. Kumagai, T. Asaka, and T. Takahashi. Location management using mobile history for hierarchical mobile IPv6 networks. *IEICE Transactions on Communications*, 87-B(9):2567–2575, September 2004. [citirano na str. 6, 27, 29, 38, 77]
- [54] K. Kawano, K. Kinoshita, and K. Murakami. A mobility-based terminal management in IPv6 networks. *IEICE Transactions on Communications*, E85-B(10):2090–2099, October 2002. [citirano na str. 6, 26, 77]
- [55] K. Kawano, K. Kinoshita, and K. Murakami. Multilevel hierarchical mobility management in densely meshed networks. *IEICE Transactions on Communications*, E89-B(7):2002–2011, July 2006. [citirano na str. 6, 26, 37, 77, 80]
- [56] Y.-W. Chen and M.-J. Huang. A novel MAP selection scheme by using abstraction node in hierarchical MIPv6. In *Proc. IEEE International Conference on Communications ICC*, volume 12, pages 5408–5413, Istanbul, Turkey, June 2006. [citirano na str. 6, 26, 77, 80]
- [57] T. Taleb, T. Suzuki, N. Kato, and Y. Nemoto. A dynamic and efficient map selection for mobile IPv6 networks. In *Proc. IEEE GLOBECOM*, volume 5, St. Louis, Missouri, USA, November 2005. [citirano na str. 6, 28, 77]
- [58] W. Chung and S. Lee. Improving performance of HMIPv6 networks with adaptive MAP selection scheme. *IEICE Transactions on Communications*, E90-B(4):769–776, 2007. [citirano na str. 6, 27, 77]
- [59] Y.-X. Lei and Z.-M. Zeng. QoS-aware MAP selection scheme based on average handover delay for multimedia services in multi-level HMIPv6 networks. In *Lecture Notes in Computer Science LNCS*, volume 4490, chapter Computational Science – ICCS 2007, pages 777–784. Springer-Verlag, 2007. [citirano na str. 6, 28, 77, 80]
- [60] W. Chung and S. Lee. Dynamic MAP selection scheme in HMIPv6 network. In *Lecture Notes in Computer Science LNCS*, volume 4352, chapter

- Advances in Multimedia Modeling, pages 502–509. Springer-Verlag, 2007. [citirano na str. 6, 27, 77, 78]
- [61] L. Wang, B. Gaabab, D. Binet, and D. Kofman. Novel MAP selection scheme using location history in hierarchical MIPv6 networks. In *Proc. IEEE Wireless Communications and Networking Conference WCNC*, pages 2420 – 2425, Las Vegas, USA, March-April 2008. [citirano na str. 6, 27, 77, 78]
- [62] Z. Wan, Z. Wang, Z. Fang, W. Zeng, and S. Wu. An efficient mobility management scheme for hierarchical mobile IPv6 networks. In *Lecture Notes in Computer Science LNCS*, volume 3981, chapter Computational Science and Its Applications - ICCSA 2006, pages 964–973. Springer-Verlag, May 2006. [citirano na str. 6, 28]
- [63] M. Mousavi and A. Quintero. Selection mechanism in hierarchical mobile IP. In *Proc. IEEE International Conference on Wireless and Mobile Computing, Networking and Communications WIMOB*, pages 321–328, Montreal, Quebec, Canada, June 2006. [citirano na str. 6, 28]
- [64] Y.-H. Wang; K.-F. Huang; C.-S. Kuo; W.-J. Huang. Dynamic MAP selection mechanism for HMIPv6. In *Proc. 22nd International Conference on Advanced Information Networking and Applications AINA*, pages 691–696, March 2008. [citirano na str. 6, 26]
- [65] S. Gambhir, M.N. Doja, Moinuudin, and M. Gambhir. Impact of SHM attributes of mobile node on QoS for HMIPv6 mobile networks. *Advances in Wireless and Mobile Communications*, 1(1-3):25–35, 2008. [citirano na str. 6, 28]
- [66] T. Ernst. Network mobility support terminology. RFC 4885, July 2007. [citirano na str. 20]
- [67] T. Ernst, A. Olivereau, L. Bellier, C. Castelluccia, and H. Lach. Mobile networks support in mobile ipv6 (prefix scope binding updates). Internet-Draft, March 2002. [citirano na str. 21]
- [68] K. Kawano, K. Kinoshita, and K. Murakami. A study on estimation of mobility of terminals for hierarchical mobility management scheme. *IEICE Transactions on Communications*, E87-B(9):2557–2566, September 2004. [citirano na str. 26, 77]
- [69] R. Novak. Proxy MAP for intra-domain route optimization in hierarchical mobile IP. *IEICE Transactions on Communications*, E89-B(2):472–481, February 2006. [citirano na str. 38]

- [70] A. Vilhar, R. Novak, and G. Kandus. MAP selection algorithms based on future movement prediction capability in synthetic and realistic environment. *Journal of Communications Software and Systems*, 4(2), 2008. [citirano na str. 43, 92]
- [71] L. Subramanian, S. Agarwal, J. Rexford, and R. Katz. Characterizing the internet hierarchy from multiple vantage points. In *Proc. IEEE INFOCOM*, volume 2, pages 618–627, New York, USA, June 2002. [citirano na str. 43, 50, 52, 57, 58, 59, 68, 69]
- [72] L. Gao. On inferring autonomous system relationships in the Internet. *IEEE/ACM Transactions on Networking*, 9(6):733–745, 2001. [citirano na str. 44, 50, 52, 53]
- [73] X. Dimitropoulos, D. Krioukov, M. Fomenkov, B. Huffaker, Y. Hyun, kc claffy, and G. Riley. AS relationships: Inference and validation. *ACM SIGCOMM Computer Communication Review*, 37(1):29–40, January 2007. [citirano na str. 44, 45, 50, 54]
- [74] The CAIDA AS relationships dataset 2004-2008. <http://www.caida.org/data/active/as-relationships/>. [citirano na str. 45, 54, 58, 93]
- [75] University of oregon route views project (2004). [citirano na str. 45, 52]
- [76] G. Battista, M. Patrignani, and M. Pizzonia. Computing the types of the relationships between autonomous systems. In *Proc. IEEE INFOCOM*, volume 1, pages 156–165, San Francisco, California, USA, 2003. [citirano na str. 50, 53]
- [77] L. Gao and F. Wang. The extent of AS path inflation by routing policies. In *Proc. IEEE GLOBECOM*, volume 3, pages 2180–2184, Taipei, Taiwan, 2002. [citirano na str. 51, 72]
- [78] H. Tangmunarunkit, R. Govindan, and S. Shenker. Internet path inflation due to policy routing. In *Proc. SPIE ITCOM*, volume 4526, pages 188–195, 2001. [citirano na str. 51]
- [79] X. Dimitropoulos and G. Riley. Modeling autonomous-system relationships. In *Proc. Int. Workshop on Principles of Advanced and Distributed Simulation PADS*, pages 143–149, Singapore, 2006. [citirano na str. 52, 55, 56]
- [80] T. Erlebach, A. Hall, and T. Schank. Classifying customer-provider relationships in the Internet. In *Proc. Int. Conf. on Communications and Computer Networks CCN*, pages 538–545, Cambridge, USA, 2002. [citirano na str. 53]

- [81] J. Xia and L. Gao. On the evaluation of AS relationship inferences. In *Proc. IEEE GLOBECOM*, volume 3, pages 1373–1377, Dallas, Texas, USA, 2004. [citirano na str. 53]
- [82] Internet routing registry. <http://www.irr.net/docs/list.html>. [citirano na str. 53]
- [83] X. Dimitropoulos, D. Krioukov, B. Huffaker, kc claffy, and G. Riley. Inferring AS relationships: Dead end or lively beginning? In *Lecture Notes in Computer Science LNCS*, volume 3503, chapter 4th Springer Workshop on Efficient and Experimental Algorithms (WEA), pages 113–125. Springer-Verlag, 2005. [citirano na str. 53]
- [84] B.M. Waxman. Routing of multipoint connections. *IEEE Journal of Selected Areas in Communications*, 6(9):1617–1622, 1988. [citirano na str. 54]
- [85] K.L. Calvert, M.B. Doar, and E.W. Zegura. Modelling Internet topology. *IEEE Communications Magazine*, 35(6):160–163, 1997. [citirano na str. 54]
- [86] M.B. Doar. A better model for generating test networks. In *Proc. IEEE GLOBECOM*, pages 86–93, London, UK, 1996. [citirano na str. 54]
- [87] M. Faloutsos, P. Faloutsos, and C. Faloutsos. On power-law relationships of the Internet topology. *ACM SIGCOMM Computer Communication Review*, 29(4):251–262, 1999. [citirano na str. 54, 59, 60]
- [88] A. Medina, A. Lakhina, I. Matta, and J. Byers. BRITE: An approach to universal topology generation. In *Proc. MASCOTS*, pages 346–353, Cincinnati, OH, 2001. [citirano na str. 54]
- [89] D. Magoni and J.-J. Pansiot. Internet topology modeler based on map sampling. In *Proc. 7th IEEE Symposium on Computers and Communications*, pages 1021–1027, Taormina, Italy, 2002. [citirano na str. 54]
- [90] J. Winick and S. Jamin. Inet-3.0: Internet topology generator. Technical Report CSE-TR-456-02, University of Michigan, 2002. [citirano na str. 54, 74, 93]
- [91] W. Aiello, F. Chung, and L. Lu. A random graph model for massive graphs. In *Proc. Annual ACM Symposium on Theory of Computing STOC*, pages 171–180, Portland, Oregon, USA, 2000. [citirano na str. 54, 55]
- [92] C.R. Palmer and J.G. Steffan. Generating network topologies that obey power laws. In *Proc. IEEE GLOBECOM*, volume 1, pages 434–438, San Francisco, USA, 2000. [citirano na str. 54]

- [93] H. Tangmunarunkit, R. Govindan, S. Jamin, S. Shenker, and W. Willinger. Network topology generators: Degree-based vs. structural. *ACM SIGCOMM Computer Communication Review*, 32(4):147–159, 2002. [citirano na str. 55]
- [94] P. Mahadevan, D. Krioukov, K. Fall, and A. Vahdat. Systematic topology analysis and generation using degree correlations. *ACM SIGCOMM Computer Communication Review*, 36(4):135–146, 2006. [citirano na str. 55, 56]
- [95] P. Mahadevan, C. Hubble, D. Krioukov, B. Huffaker, and A. Vahdat. Orbis: rescaling degree correlations to generate annotated internet topologies. *ACM SIGCOMM Computer Communication Review*, 37(4):325–336, 2007. [citirano na str. 55, 63]
- [96] H. Chang, S. Jamin, and W. Willinger. To peer or not to peer: Modeling the evolution of the internet’s AS-level topology. In *Proc. IEEE INFOCOM*, page 1–12, Barcelona, Spain, 2006. [citirano na str. 55]
- [97] X. Dimitropoulos, D. Krioukov, A. Vahdat, and G. Riley. Graph annotations in modeling complex network topologies. *ACM Transactions on Modeling and Computer Simulation (TOMACS)*, 2009. [citirano na str. 56]
- [98] Z. Ge, D. Figueiredo, S. Jaiswal, and L. Gao. On the hierarchical structure of the logical Internet graph. In *Proc. SPIE ITCOM*, pages 208–222, Denver, Colorado, USA, 2001. [citirano na str. 57, 59, 68, 70, 71]
- [99] Y. Wang, W. Chen, and J. S. Ho. Performance analysis of mobile IP extended with routing agents. Technical Report 97-CSE-13, Southern Methodist University, 1997. [citirano na str. 77, 78, 80]
- [100] Y. Wang, W. Chen, and J.S.M. Ho. Performance analysis of mobile IP extended with routing agents. In *Proc. 2nd European IASTED International Conference on Parallel and Distributed Systems*, pages 2–3, July 1998. [citirano na str. 77, 78, 80]
- [101] A. Argyriou and V.K. Madiseti. A soft-handoff transport protocol for media flows in heterogeneous mobile networks. *Elsevier Computer Networks*, 50(11):1860–1871, 2006. [citirano na str. 78]
- [102] A. Vilhar, R. Novak, and G. Kandus. MAP selection algorithms based on future movement prediction capability. In *Proc. 15th International Conference on Software, Telecommunications and Computer Networks SoftCOM*, Split-Dubrovnik, Croatia, September 2007. [citirano na str. 92]

- [103] A.E. Eiben and J.E. Smith. *Introduction to Evolutionary Computing*. Springer, corrected 2nd printing edition, 2007. [citirano na str. 100]

