



Univerza v Mariboru

Fakulteta za elektrotehniko, računalništvo in informatiko

mag. Karl Benkič, univ. dipl. inž. str.

**PROMETNO URAVNOTEŽENI USMERJEVALNI
ALGORITMI ZA BREZZIČNA SENZORSKA
OMREŽJA**

doktorska disertacija

Maribor, Julij 2010

Teza: Prometno uravnoreženi usmerjevalni algoritmi za brezžična
senzorska omrežja

Kandidat: **mag. Karl Benkič, univ. dipl. inž. str.**
status: mladi raziskovalec

štev. raziskovalca: 27565
Štev. indeksa: 95029183

Študijski program: telekomunikacije
Štev. izvodov: 5

Mentor: dr. Žarko Čučej
red.prof. za avtomatiko in robotiko
red.prof. za telekomunikacije

So-mentor: izr. prof. dr. Boštjan Brešar

Zahvala

Zahvaljujem se svoji družini, staršem ter bratu za podporo, ki so mi jo izkazali študiju.

Za vodenje bi se rad posebej zahvalil najboljšim; mentorju prof. dr. Žarku Čučēju ter somentorju prof.dr. Boštjanu Brešarju! Brez vaše pomoči mi ne bi uspelo! Hvala!

Hvala celotnemu kolektivu laboratorija za obdelavo signalov in daljinska vodenja, posebej ljubiteljev jutranje kave: Dušan, Marko, še en Marko in Matej, v čast mi je, da nismo "zgolj" sodelavci!

Nenazadnje bi se rad zahvalil vsem, ki ste verjeli vame. Vsem tistim, ki ste se zavedali in verjeli, da moja pot ni lahka, pa čeprav je bila prijetna.

Per aspera ad Astra.

Maribor, 1. 6. 2010
Številka: DR 42/2010/425-MGM

Na osnovi 287., 140., 142. in 144. člena Statuta Univerze v Mariboru (Statut UM-UPB8, Ur. l. RS, št. 1/2010) ter sklepa 32. redne seje Senata Univerze v Mariboru z dne 1. 6. 2010 v zvezi z vlogo doktorskega kandidata mag. Karla Benkiča za sprejem odločitve o predlagani temi doktorske disertacije in mentorja,

izdajam naslednji

SKLEP

Odobri se tema doktorske disertacije mag. Karla Benkiča s Fakultete za elektrotehniko, računalništvo in informatiko z naslovom »Prometno uravnoteženi usmerjevalni algoritmi za brezžična senzorska omrežja«. Kandidatu se za mentorja imenuje red. prof. dr. Žarko Čučej, za somentorja pa izr. prof. dr. Boštjan Brešar. Kandidat mora članici predložiti izdelano doktorsko disertacijo v zadostnih izvodih najpozneje do 31. 5. 2014.

Obrazložitev:

Kandidat mag. Karl Benkič je dne 13. 1. 2010 na Fakulteti za elektrotehniko, računalništvo in informatiko vložil vlogo za potrditev teme doktorske disertacije z naslovom »Prometno uravnoteženi usmerjevalni algoritmi za brezžična senzorska omrežja«. Za mentorja je bil predlagan red. prof. dr. Žarko Čučej, za somentorja pa izr. prof. dr. Boštjan Brešar.

Senat Fakultete za elektrotehniko, računalništvo in informatiko je na osnovi pozitivnega mnenja komisije za oceno teme doktorske disertacije, ki je ugotovila, da kandidat izpolnjuje pogoje za pridobitev doktorata znanosti, in ocenila, da je predlagana tema ustrezna, sprejel pozitivno mnenje in poslal predlog teme doktorske disertacije s predlogom mentorja in somentorja v odobritev Senatu univerze.

Senat Univerze v Mariboru je po proučitvi vloge in na osnovi določil Statuta Univerze v Mariboru sprejel svojo odločitev o predlagani temi doktorske disertacije in imenoval mentorja in somentorja, kot izhaja iz izreka.

V skladu s 144. členom Statuta Univerze v Mariboru mora kandidat za pridobitev doktorata znanosti najpozneje v štirih letih od dneva izdaje tega sklepa, članici predložiti izdelano doktorsko disertacijo v zadostnih izvodih. Kandidatu je bil določen rok glede na datum sprejetja teme na pristojnem organu.

Pouk o pravnem sredstvu:


Zoper ta sklep je možna pritožba na Senat Univerze v Mariboru v roku 8 dni od prejema tega sklepa.

Obvestiti:

1. Kandidata.
2. Fakulteto.
3. Arhiv.



Rektor:
Prof. dr. Ivan Rozman





Fakulteta za elektrotehniko,
računalništvo in informatiko

Smetanova ulica 17
2000 Maribor

Številka: 95029183
Datum: 09. 07. 2010

Na osnovi 330. člena Statuta Univerze v Mariboru (UPB 4) izdajam

POTRDILO O DOKTORIRANJU

Fakulteta za elektrotehniko, računalništvo in informatiko Univerze v Mariboru potrjuje, da je **mag. Karl Benkič, univ. dipl. inž. str.**, rojen 16. 07. 1981 v kraju Maribor, opravil vse obveznosti po doktorskem študiju Elektrotehnika in dne **09. 07. 2010** javno in uspešno zagovarjal doktorsko disertacijo z naslovom

PROMETNO URAVNOTEŽENI USMERJEVALNI ALGORITMI ZA BREZŽIČNA SENZORSKA OMREŽJA

in s tem ob izpolnjevanju pogoja iz 157. člena Statuta Univerze v Mariboru pridobil pravico do podelitve znanstvenega naslova **doktor znanosti s področja elektrotehnike**.

Potrdilo je izdano, ker promocija doktorjev še ni bila opravljena. Kandidat lahko začne uporabljati znanstveni naslov po opravljeni promociji v doktorja znanosti. Skladno z 21. točko 28. člena Zakona o upravnih taksah (Ur. l. RS št. 42/07 - ZUT UPB3, 126/07) je to potrdilo takse prosto.



DEKAN
prof. dr. Igor Tičar

Vsebina

Vsebina	· xi
Uporabljene kratice in simboli	· xvii
Kazalo	· xix
Slike	· xxiii
Tabele	· xxv
1 Uvod	· 1
2 Teorija grafov	· 15
3 Usmerjanje prometa	· 27
4 Model omrežja in prometna uravnoveženost	· 37
5 Predlog algoritma optimalnega polprirejanja	· 43
6 Predlog usmerjevalnega protokola: SPaRCRoute	· 59
7 Analiza SPaRCRoute protokola	· 67
8 Zaključek	· 85
Dodatek	· 87
A O kandidatu	· 89
B SPaRCMosquito in SPaRCSoft	· 91
C Rezultati simulacij	· 95
Literatura	· 103

PROMETNO URAVNOTEŽENI USMERJEVALNI ALGORITMI ZA BREZZIČNA SENZORSKA OMREŽJA

Ključne besede: brezžična senzorska omrežja, protokoli usmerjanja, teorija grafov, uravnoteževanja obremenitev

UDK: 681.586.5:621.396(043.3)

Povzetek

Brezžične komunikacije, kot na primer GSM tehnologija, WiFi vstopne točke, digitalna televizija in drugo postajajo v naših življenjih vedno bolj prisotna. Cenovna dostopnost komponent in nagel industrijski razvoj je vzpodbudil uporabo brezžičnih komunikacij tudi v osebne namene (kot primer podajmo samo GSM telefon in Bluetooth slušalko). Ljudje smo vedno bolj vpeti v svet komunikacij pa se često tega niti ne zavedamo.

Vedno manjše, cenejše in zmogljivejše komponente so pripomogle k uporabi brezžičnih komunikacij v prej nepredstavljenih aplikacijah. Eno izmed takšnih aplikacij predstavljajo tudi **brezžična senzorska omrežja** (BSO). Brezžična senzorska omrežja so omrežja sestavljena iz majhnih, baterijsko napajanih, pametnih senzorjev sposobnih brezžične komunikacije. Njihova radijska vidljivost je ponavadi majhna, cena pa tako nizka, da senzorske enote po uporabi preprosto zavržemo. Namenjena so spremljanju različnih fenomenov (sezmiologija, spremljanje habitata, spremljanje požarov, vojaške aplikacije...).

Med intenzivnejše raziskave brezžičnih senzorskih omrežij že od vsega začetka spadajo raziskave usmerjevalnih algoritmov. Standardni usmerjevalni algoritmi uporabljeni v standardih IEEE 802.11x zaradi posebnosti BSO niso uporabni ali pa je njihova uporaba v BSO nesmiselna (zaradi velike potrošnje procesorskih ali spominskih virov ter energije). Posledično so raziskave usmerjene v za brezžična senzorska omrežja posebej prilagojene protokole usmerjanja prometa.

V tezi smo se omejili na raziskave prometno uravnoteženih algoritmov, ki sporočila pošiljajo po najkrajši možni poti (minimalno število etap). Raziskovali smo usmerjanje v statičnih BSO, kjer senzori s časom ne spreminjajo svoje lege ali pa jo spreminjajo v intenziteti, ki ni bistvena za delovanje algoritmov.

Predlagan usmerjevalni protokol je sestavljen iz dveh algoritmov: BFS algoritma in optimalnega polprirejanja. Algoritem za izračun minimalnega števila potrebnih etap, da je sporočilo poslano od vozlišča do bazne postaje je v bistvu dodelan BFS algoritem. Z BFS algoritmom izračunamo nivo vsakega vozlišča (nivo predstavlja oddaljenost od bazne postaje v etapah) v omrežju. Vozlišča iz dveh sosednjih nivojev za potrebe algoritma iskanja optimalnega polprirejanja predstavimo kot virtualni dvodelni graf. Teh virtualnih grafov je za ena manj kot število nivojev vozlišč ($n - 1$). Na vsakem virtualnem dvodelnem grafu posebej izračunamo optimalno polprirejanje. Cilj optimalnega polprirejanja je uravnotežitev prometa med vozlišči. Skupen rezultat obeh algoritmov je topologija imenovana *topologija prirejanja*. Topologija prirejanja je v bistvu vpeto drevo, ki ga uporablja protokol usmerjanja. Kvaliteto

uravnovežitve na vseh nivojih vpetega drevesa ocenimo po metriki faktorja uravnovežitve (Chebyshevo sumo).

V delu predlagamo tudi nov, hitrejši algoritem za izračun optimalnega polpripravljanja. Eksperimenti so pokazali, da je izvajanje algoritma vsaj 15 % hitrejše kot pri ostalih, do sedaj znanih algoritmih.

Za testiranje in simuliranje usmerjevalnega protokola smo uporabili standardni MAC protokol (IEEE 802.15.4), temelječ na CSMA-CA izmikanju kolizij, kateremu smo dodali še RTS/CTS kontrolne okvirje.

Za potrditev teze smo uporabili simulacijsko okolje OPNET kjer smo razvili model prototipa brezžičnega senzorskega vozlišča SPaRCMosquito razvitega v laboratoriju. Rezultati simulacij so potrdili, da protokol zaradi svojega načina delovanja pripomore k manjši porabi energije celotnega senzorskega omrežja in krajšim latentnim časom sporočil poslanih od senzorskih vozlišč do bazne postaje.

Predlagan usmerjevalni algoritem smo implementirali na testno brezžično senzorsko omrežje sestavljeno iz 9 vozlišč SPaRCMosquito.

LOAD BALANCED ROUTING PROTOCOLS FOR WIRELESS SENSOR NETWORKS

Key words: Wireless sensor networks, routing protocol, graph theory, load balancing

UDK: 681.586.5:621.396(043.3)

Summary

Abstract

Wireless communications, such as GSM telephony, WiFi access points, digital television over the air, and others have great impact on our lives. Low prices of electronic components and rapid hardware development are encouraging personal use of a wireless communications (i.e., GSM phones, Bluetooth headsets). We are becoming part of a digital communication society without really realizing it.

The foundations of wireless technology were set with smaller, cheaper and more powerful wireless components were foundations. Nowadays they are a common part in building previously unimaginable applications. One area of expertise in wireless communications are Wireless Sensor networks. Wireless sensor networks are assembled from small, battery powered, smart sensors capable of a wireless communication called Sensor Nodes (or joust nodes). Node radio range of sight is normally limited to a few ten meters or less. These nodes are low-cost hardware equipment, thus in majority of cases after battery depletion, they are discarded.

One of the most intensely researched topics in WSN are routing algorithms and protocols. Standard routing algorithms, like those used in IEEE 802.11x are, due to WSN characteristics (e.g., limited energy consumption, low CPU and memory resources...) not appropriate for use in the wireless sensor networks.

In the PhD thesis we have focused on traffic balanced algorithms which sends messages over shortest path possible or available (expressed as minimum number of hops). Our research target was static wireless sensor network, where the sensors do not change position in time, or the speed of a moving node does not impact the algorithm calculations.

Routing algorithm implements two sub algorithms: BFS and semi matching. Algorithm for determining minimum number of hops needed from message to be send from a node to the base station is basically a BFS algorithm. BFS algorithm calculates the level of each node (hop distance from the BS) in the network. Nodes from the two neighboring levels are then assigned as members of a virtual bipartite graph. On every bipartite graph semi matching is calculated. Semi matching algorithm balances the load evenly across nodes. Result of the BFS and semi matching algorithm is a topology (called matching topology) – a spanning tree which is later used in a routing protocol.

We accomplish to write new, fastest algorithm for calculating optimal semi matching. The experimental results showed that our proposed algorithm gives optimal results and preforms at least 15% faster than other state of art algorithms.

For testing and simulation purposes we used standard IEEE 802.15.4 PHY and MAC protocol, based on CSMA-CA algorithm. Later on we added RTS and CTS control packets for avoiding hidden node problem.

Simulations for confirming thesis were done in OPNET simulation tool, where we have developed a model of a wireless sensor network node called SPaRCMosquito. Simulation results have confirmed that using protocol SPaRCRoute results in lower power consumption, lower latency, and a higher throughput of the network.

Proposed routing algorithm SPaRCRoute was implemented on a test network consisted of 9 SPaRCMosquito nodes.

Uporabljene kratice in simboli

$\chi'(G)$	Kromaticni indeks grafa
$\chi(G)$	Kromaticno število grafa
$\epsilon_G = E $	velikost grafa
λ	Valovna dolžina
$\nu_G = V $	Red grafa
θ	Faktor uravnovežitve (Chebysheva suma)
A_{SM}	Algoritem polprirejanja
BFS	Iskaje v širino (ang.: Breadth-first search)
BP	Bazna Postaja
BS	Base Station
BSO	Brezžicno Senzorsko Omrežje
$C(M)$	Skupna cena polprirejanja
$CDMA$	Code Division Multiple Access
$cost_M(v)$	Cena polprirejanja vozlišča v
$d_G(A)$	Stopnja vozlišča A grafa G
$deg_M(v)$	Stopnja prirejanja vozlišča v
DFS	Iskanje v globino (ang.: Depth-first search)
$E(G)$	Množica povezav grafa E
FFD	Full Function Device
$G(V(G), E(G))$	Graf
G_{d_x}	dvodelni graf nivoja x
G_o	Ojacitev oddajne antene
G_s	Ojacitev sprejemne antene
$GPRS$	General Packet Radio Service
GPS	Global positioning system
H_x	Pograf grafa x
M	Prirejanje; množica povezav E
M_x	Matrika sosednosti
MAC	Media Access Control
$MANET$	Mobile Ad-Hoc NETWORKS
NP	Problemi nepolinomske racunske zahtevnosti
O	Racunska zahtevnost

<i>P</i>	Pot
P_o	Oddajna moc senzorske enote
P_s	Sprejemna moc senzorske enote
<i>QoS</i>	Quality of Service
<i>QUDG</i>	Quasi Unit Disk Graph
<i>RFD</i>	Reduced Function Device
<i>RFID</i>	Radio-frequency identification
<i>Rt – WSN</i>	Real-time Wireless Sensor Networks
<i>SPaRC</i>	Signal Processing and Remote Control
<i>T</i>	Drevo
<i>U</i>	Leva vozlišča
<i>UDG</i>	Unit-Disk Graph
<i>UWB</i>	Ultra Wide Band
<i>V</i>	Desna vozlišča
$V(G)$	Množica vozlišč grafa G
<i>WiFi</i>	Wireless Fidelity (brežično omrežje)
<i>WPAN</i>	Wireless Personal Area Networks
<i>WSN</i>	Wireless Sensor Network

Vsebina	xi
Povzetek	xiii
Summary	xv
Uporabljene kratice in simboli	xvii
Kazalo	xix
Slike	xxiii
Tabele	xxv
1 Uvod	1
1.1 Termini in kratice	1
1.2 Osnovni pojmi	2
1.3 Senzor	2
1.3.1 Vozlišča	3
1.3.2 Bazna vozlišča	3
1.3.3 Topologija	4
1.3.4 Usmerjanje	4
1.3.5 Samoorganizacija brezžičnih senzorskih omrežij	4
1.3.6 Metrika	5
1.4 Uporabnost in perspektiva WSN	5
1.4.1 Vrste brezžičnih senzorskih omrežij	6
1.4.2 Raziskovalni izzivi v brezžičnih senzorskih omrežjih	7
1.5 Formulacija problema	9
1.6 Pregled stanja raziskovalnega področja teze	10
1.7 Področja in prispevki v tezi	11
1.7.1 Pričakovani prispevki v tezi	12
1.7.2 Uporabljena orodja v tezi	12
1.8 Sprehod po tezi	13
2 Teorija grafov	15
2.1 Osnovni pojmi	15

2.1.1	Podgrafi in vpeti grafi	17
2.1.2	Usmerjeni in neusmerjeni grafi	19
2.2	Drevesa	20
2.2.1	Kruskalov in Primov algoritem	21
2.3	Barvanje grafov	23
2.4	Algoritmi	23
2.4.1	Razredi problemov	24
2.5	Pregledovanje grafov	24
2.6	Najkrajše poti	25
3	Usmerjanje prometa	27
3.1	Topologija	30
3.1.1	Primer uporabe kontrole topologije	31
3.1.2	Vpliv fizične plasti na topologijo	32
	Modeli prenosnih kanalov	33
4	Model omrežja in prometna uravnoteženost	37
4.1	Matematični model omrežja	37
4.1.1	Znani matematični modeli omrežij	37
4.1.2	Drevesna struktura v modelu	38
4.2	Uravnoteženost prometa	39
4.3	Kako oceniti uravnoteženost?	40
5	Predlog algoritma optimalnega polprirejanja	43
5.1	Algoritmi polprirejanja	43
5.1.1	Predpostavke v algoritmih polprirejanja	44
5.1.2	Znani algoritmi polprirejanja	47
	A_{SM1} algoritem za optimalno polprirejanje	47
	A_{SM2} algoritem za optimalno polprirejanje	47
5.2	Predlog alternativnega algoritma optimalnega polprirejanja	49
5.2.1	Dodajanje in odstranjevanje vozlišč(a)	51
5.3	Primerjava algoritmov polprirejanja	51
5.3.1	Primerjava delovanja algoritmov	51
5.3.2	Primerjava hitrosti izvajanja algoritmov	54
	Naključni grafi	55
	Mrežni grafi	56
	Grafi vrvi	56
5.4	Algoritmi polprirejanja za utežene grafe	58
6	Predlog usmerjevalnega protokola: SPaRCRoute	59
6.1	Predpostavke in uvod	59

6.2	Začetna in dejanska topologija ter topologija polprirejanja	59
6.2.1	Dejanska topologija in topologija polprirejanja	60
6.3	Identifikacija vozlišča – Naslavljanje	62
6.4	Delovanje protokola	63
6.4.1	SPaRCRoute	63
6.4.2	ESR (EYES Source Routing Protocol)	64
7	Analiza SPaRCRoute protokola	67
7.1	Model vozlišča in omrežja	67
7.1.1	Model senzorskega vozlišča SPaRCMosquito	67
	Delovanje modela senzorskega vozlišča SPaRCMosquito	69
	Lastnosti modela senzorskega vozlišča SPaRCMosquito	70
7.1.2	Vpliv MAC plasti na rezultate simulacij	73
	Skrito vozlišče	73
	Izpostavljeno vozlišče	77
7.2	Simulacije protokola usmerjanja	78
7.2.1	Kolizijske domene	78
7.2.2	Enakomerna porazdelitev prometa	80
	Posamezna vozlišča pri enakomerni porazdelitvi prometa	81
7.2.3	Simulacija dveh manjših senzorskih omrežij	82
8	Zaključek	85
	Dodatek	87
A	O kandidatu	89
B	SPaRCMosquito in SPaRCSoft	91
B.1	SPaRCMosquito	91
B.2	SPaRCSoft in druga programska oprema	93
C	Rezultati simulacij	95
C.0.1	Simulacije: Vpliv MAC plasti na rezultate simulacij	95
	Vpliv skritega vozlišča	95
	Vpliv izpostavljenega vozlišča	95
C.0.2	Simulacije protokola usmerjanja	98
	Simulacije: Vpliv kolizijskih domen na rezultate simulacij	98
	Uravnoteženo in neuravnoteženo drevo	98
	Simulacija dveh manjših senzorskih omrežij	98
	Literatura	103

1.3-1	Strojna oprema sensorja v brezžičnem senzorskem omrežju	2
1.3-2	Referenčni ISO/OSI model in modela vozlišč RFD in FFD. Področje raziskav so funkcije tretje plasti v FFD (osenčeno polje na sliki 1.3-2c in njihove interakcije z funkcijami narejene uporabniške plasti in podrejenih podplasti LLC in MAC	3
1.3-3	Prenos podatkov preko FFD	4
2.0-1	Problem Königsberg mostov	15
2.0-2	Uporaba teorije grafov	16
2.1-1	Primer preprostega grafa G	16
2.1-2	Graf G in njegov komplementarni graf \overline{G}	17
2.1-3	Omrežje in model modeliran z grafom	18
2.1-4	Graf G , podgraf H_P in vpeti podgraf H_V	18
2.1-5	Posebne vrste grafov	18
2.1-6	Primer usmerjenega G_U in neusmerjenega G_N grafa	19
2.1-7	Utežen graf: Oddaljenost večjih Slovenskih mest	20
2.2-1	Primer drevesa T	21
2.2-2	Petersenov graf G in eno od njegovih možnih vpetih dreves T	21
2.2-3	Utežen graf G in njegovo vpeto drevo T	22
2.3-1	Graf G in njegovo barvanje	23
3.0-1	Schema preprostega brezžičnega senzorskega omrežja	27
3.0-2	Delitev protokolov glede na predhodno poznavanje poti	28
3.0-3	Delitev protokolov glede na omrežno strukturo in operacije	30
3.1-1	Umestitev topologije v ISO/OSI model	31
3.1-2	Primer kontrole topologije	31
3.1-3	Različne topologije	32
3.1-4	Primer odbojnega modela	34
4.1-1	Primer virtualne bazne postaje r	38
4.1-2	Rezultat BFS algoritma izvedenega na grafu G	39
4.2-1	Različni tipi uravnovešitev	40
4.2-2	Drevo najkrajših poti in uravnovešeno drevo	40
4.3-1	Označeno (BFS) uteženo drevo T	42
5.0-2	Graf G in izpeljava dvojnih grafov	43
5.1-1	Problem švicarske banke modeliran z dvojnimi grafom G	44
5.1-2	Graf G in primer polprirejanja	45
5.1-3	Primer visoke in nizke cene $C(M)$	45
5.1-4	Primer znižanja stroškov z uporabo alternativnih poti	46
5.2-1	Optimalno polprirejanje na dvodelnem grafu G	50
5.2-2	Primer dodajanja vozlišča	51
5.3-1	Primerjava delovanja algoritmov za optimalno polprirejanje	52

5.3-2	Primerjava delovanja algoritmov za optimalno polprirejanje	53
5.3-3	Primerjava delovanja algoritmov za optimalno polprirejanje	53
5.3-4	Čas izvajanja algoritma na naključnem grafu	56
5.3-5	Čas izvajanja algoritma na mrežnem grafu	57
5.3-6	Čas izvajanja algoritma na mrežnem grafu	57
6.2-1	Od začetne do topologije polprirejanja	60
6.2-2	Od realne topologije do topologije polprirejanja	60
6.2-3	Računanje topologije polprirejanja	61
6.3-1	Naslovna struktura predlaganega algoritma	62
6.3-2	Drevo polprirejanja z naslovi	63
6.4-1	Shema delovanja SPaRCRoute	64
7.1-1	OPNET model in scenarij	68
7.1-2	Osnovni IEEE802.15.4 paket	69
7.1-3	Vpenjanje podatkov v okvir	69
7.1-4	<i>Mosq</i> /MAC proces modela SPaRCMosquitoModel (Slika 7.1-1a)	70
7.1-5	Graf ustvarjenih paketov	72
7.1-6	Graf energijske porabe za ustvarjene pakete	72
7.1-7	Graf energijske porabe za ustvarjene pakete: skupna poraba, poraba enote "node_0" in "node_1"	74
7.1-8	Analiza hitrosti prenosa podatkov za dva vozlišča (Slika 7.1-5a)	74
7.1-9	Analiza hitrosti prenosa podatkov za dva vozlišča in zavrženih paketov (Preplavitev vrste v MAC)	75
7.1-10	Scenarij ocenjevanja razmer zaradi vpliva MAC (RTS/CTS)	75
7.1-11	Analiza količine uporabnega prometa na kanalu	76
7.1-12	Scenarij testiranja več etap v BSO	77
7.1-13	Stanje energije, kolizij in uspešnosti paketov v več etapni komunikaciji	78
7.2-1	Kolizijske domene	78
7.2-2	Scenarija testiranja kolizijskih domen	79
7.2-3	Scenarij simulacije uravnoveženega in neuravnoveženega drevesa	80
7.2-4	Simulacijski scenarij uravnoveženega in ne-uravnoveženega omrežja)	82
B.2-1	Moduli programa SPaRCSoft	93
B.2-2	SPaRCSoft programsko orodje	94
C.0-3	Vpliv MAC plasti, paketne in energijske razmere: 5 vozlišč	95
C.0-4	Vpliv MAC plasti, paketne in energijske razmere: 7 vozlišč	96
C.0-5	Vpliv MAC plasti, paketne in energijske razmere: 9 vozlišč	96
C.0-6	Vpliv izpostavljenega vozlišča: 1 etapa	97
C.0-7	Vpliv izpostavljenega vozlišča: 3 etape	97
C.0-8	Vpliv izpostavljenega vozlišča: 6 etap	98
C.0-9	Kolizijske domene v uravnoveženem drevesu: brez skupne kolizijske domene	99
C.0-10	Kolizijske domene v uravnoveženem drevesu: skupna kolizijska domena	99
C.0-11	Simulacijski rezultati uravnoveženega drevesa	100
C.0-12	Simulacijski rezultati neuravnoveženega drevesa	100
C.0-13	Simulacijski rezultati uravnoveženega scenarija	101
C.0-14	Simulacijski rezultati neuravnoveženega scenarija	102

1.3-1	Primerjava brezžičnih senzorskih in MANET omrežij	4
2.4-1	Časovna zahtevnost in notacija velikega O povzeto po [115].	24
5.1-1	Izračun cenovne vrednosti $C(M)$ za (Slika 5.1-3a) in (Slika 5.1-3b)	46
5.3-1	Faktorji uravnovežitve θ za grafe (Slika 5.3-1)	51
5.3-2	Faktorji uravnovežitve θ za grafe (Slika 5.3-2)	52
5.3-3	Faktorji uravnovežitve θ za grafe (Slika 5.3-3)	52
6.2-1	Zagon A_{SM3} algoritma na grafu (Slika 6.2-3a)	62
7.1-1	Energijske in časovne razmere pošiljanja enega paketa (teoretično)	71
7.1-2	Vpliv MAC plasti: Analiza paketnih razmer	76
7.1-3	Energija, uspešnost paketov in število kolizij v več etapnem sistemu	77
7.2-1	Energijska analiza scenarija (Slika C.0-9) in (Slika C.0-10)	79
7.2-2	Analiza uspešnosti prenosa podatkov scenarijev (Slika C.0-9) in (Slika C.0-10)	80
7.2-3	Uravnovežitveni faktor θ za scenarija (Slika 7.2-3b) in (Slika 7.2-3c)	80
7.2-4	Paketne in časovne razmere za scenarija (Slika 7.2-3b) in (Slika 7.2-3c)	81
7.2-5	Energijske razmere za scenarija (Slika 7.2-3b) in (Slika 7.2-3c)	81
7.2-6	Energijske razmere za scenarija (Slika 7.2-3b) in (Slika 7.2-3c), vozlišča 1., 2. in 3.	81
7.2-7	Uravnovežitveni faktor θ za scenarija (Slika 7.2-3b) in (Slika 7.2-3c)	82
7.2-8	Paketne in časovne razmere za scenarija (Slika 7.2-3b) in (Slika 7.2-3c)	82
7.2-9	Energijske razmere za scenarija (Slika 7.2-3b) in (Slika 7.2-3c)	83
B.1-1	SPaRCMosquito verzija 1. in verzija 2.	91

Brezžična senzorska omrežja (BSO) se od ostalih brezžičnih omrežij razlikujejo v osebkih, ki medsebojno komunicirajo. To so senzori, katerih osnovna funkcija je zaznavanje okolja, na primer meritve temperature, tlaka, pospeška, hitrosti in drugo, ali video nadzora prostora in podobno kompleksnih sistemov zaznavanja okolja, ter sporočanje teh podatkov nadzornim sistemom. S komunikacijskega stališča so senzori viri podatkov, katerih lastnosti so v večini primerov konstantne in vnaprej predpisane. Tokovi podatkov, ki jih ustvarjajo, so usmerjeni k nadzornim sistemom. V nasprotni smeri, od nadzornih sistemov k senzorem, je tok krmilnih podatkov. Tudi ta ima vnaprej znane lastnosti.

BSO: Brezžično senzorsko omrežje

Dejstvo, da se podatki med senzori in nadzornim sistemom izmenjujejo z radijskim signalom, pomeni, da imajo senzori avtonomni vir energije. V nasprotnem primeru bi seveda uporabljali manj zapleteni komunikacijski sistem po vodnikih. Avtonomne vire energije ima tudi večina drugih naprav, ki so povezane v brezžična omrežja, vendar je velika razlika v trajanju avtonomije. Med tem ko ta, na primer pri mobilnih telefonih, prenosnih računalnikih in podobnih napravah, traja od nekaj ur do nekaj dni, se pri senzorjih pogosto zahteva, da traja več let oziroma vso življenjsko dobo sensorja. To veliko avtonomijo se dosega s strogim varčevanjem potrošnje energije. To pa zahteva razvoj novih telekomunikacijskih konceptov, tehnik in tehnologij.

Brezžična senzorska omrežja imajo tudi mnogo skupnega z ostalimi brezžičnimi omrežji. Na primer, zagotovitev robustnosti na motne v okolju, sinhronizacija, zanesljivost, avtentikacija in ne nazadnje samo-organiziranje omrežja. Zato je mnogo rešitev iz ostalih brezžičnih omrežij zelo uporabnih tudi v brezžičnih senzorskih omrežjih.

1.1 Termini in kratice

V strokovni literaturi s področja telekomunikacij običajno mrgoli raznih kratic. Te so praviloma angleškega izvora in dobro udomačene v vseh jezikih, tudi slovenskem. Z njimi se skuša v literaturi in pogovoru zagotoviti jedrnat in pregleden opis konceptov, tehnik in tehnologij. Zato jih tudi mi uporabljamo povsod v originalni in uveljavljeni obliki povsod, kjer so v pomoč, predvsem v tabelah in slikah. Njihov pomen opišemo ob prvem zapisu kratice, na primer, brezžična senzorska omrežja bomo mnogokrat na kratko označili z WSN. V seznamu uporabljenih kratic je razlagi v izvornem jeziku dodan prevod v slovenski jezik.

WSN: Wireless Sensors Network

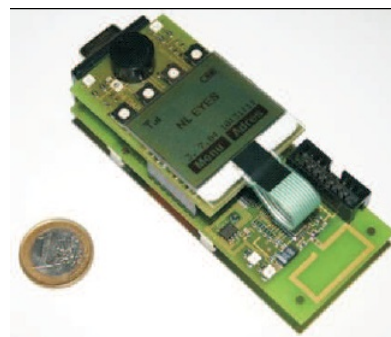
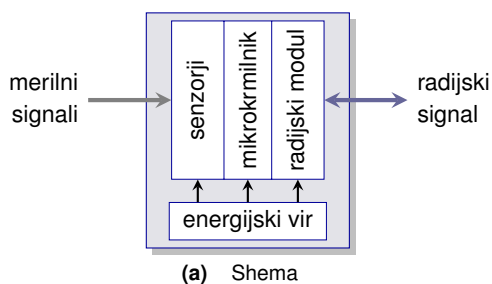
1.2 Osnovni pojmi

Glavni in ponavadi edini gradnik senzorskih omrežij je senzor. V brezžičnih senzorskih omrežjih je to naprava, ki je sposobna z različnimi tipali meriti signale fizikalnih količin, pretvoriti signale v podatke, podatke obdelati in preko brezžične povezave poslati uporabniku podatkov. Za te naloge mora imeti določeno "inteligenco", ki jo zagotavlja strojna in programska oprema.

Ko se sklicujemo na *brezžično komunikacijo*, bomo vedno imeli v mislih radijsko komunikacijo, četudi so na razpolago druge vrste brezžične komunikacije [111].

1.3 Senzor

Strojna oprema senzorja ima štiri sklope (Slika 1.3-1a),



Slika 1.3-1

Strojna oprema senzorja v brezžičnem senzorskem omrežju.

katerih polprevodniški elementi so v mnogih senzorjih združeni v enem samem integriranem vezju. Njihove funkcije so:

- Radijski modul:** radijska komunikacija z drugimi vozlišči. V njem se vrši ojačenje, modulacija in demodulacija radijskega signala, merijo jakost signala in druge s komunikacijami povezane funkcije.
- Senzorski modul:** merjenje fizikalnih količin. Njegov izvedba je odvisna od namena senzorja. lahko vsebuje več pretvornikov fizikalnih količin v električne signale.
- Mikrokrmilnik:** nadzor nad delovanjem senzorja, obdelava merilnih signalov, izvajanje komunikacijskih protokolov.
- Energijski vir:** napajanje senzorja z električno energijo. Energijo ponavadi zagotavlja baterija, intenzivno pa se raziskujejo možnosti, da se energija zagotovi iz okolja senzorja (na primer iz v prostoru pristne energije elektro-magnetnega valovanja).

Nosilne frekvence radijskega signala so ponavadi iz frekvenčnih pasov, za katere ni potrebno imeti licence za uporabo, ali pa se uporabljajo široki pasovi in modulatorske tehnike ter moči signalov, ki ne motijo druge signale [92]. Pogoje uporabe frekvenc določajo mednarodni standardi, na primer ITU-R [52]. Večina teh frekvenčnih območij je med 433 MHz in 10 GHz [9, 39].

Programsko opremo senzorjev lahko razdelimo na tri dele

1. operacijski sistem
2. uporabniški program
3. komunikacijski protokol

1.3.1 Vozlišča

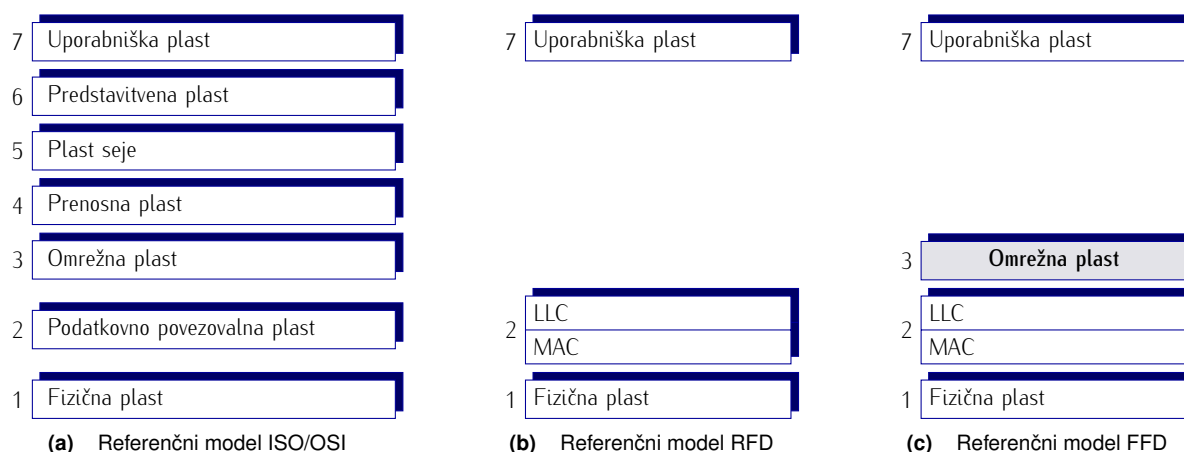
Brezžična senzorska omrežja raziskujemo s komunikacijskega stališča, zato senzorje obravnavamo kot *vozišča* v komunikacijskem omrežju. Medsebojno jih razlikujejo po njihovi zmožnosti komuniciranja. Glede na njo, jih delimo v dve skupini:

1. *Vozlišča z omejenimi funkcijami* (RFD), ki so lahko samo *vir* podatkov in *ponori* sporočil.
2. *Vozlišča s vsemi funkcijami* (FFD), ki so lahko *vir* in *posredniki* podatkov ter *ponori* in *posredniki* krmilnih sporočil.

RFD: Reduced
Function Device

FFD: Full Function Device

Razliko med RFD in FFD vidimo iz opisa njihovih modelov z referenčnim modelom ISO/OSI (Slika 1.3-2).



legenda:

LLC: Logic Link Control *krmiljenje logične povezave*

MAC: Media Access Control *krmiljenje dostopa do prenosnega medija*

Slika 1.3-2

Referenčni ISO/OSI model in modela vozlišč RFD in FFD. Področje raziskav so funkcije tretje plasti v FFD (osenčeno polje na sliki 1.3-2c) in njihove interakcije z funkcijami narejene uporabniške plasti in podrejenih podplasti LLC in MAC.

Iz te primerjave sledi, da brezžično senzorsko omrežje mora vsebovati vsaj eno vozlišče FFD in dve vozlišči RFD. V tem primeru je FFD hkrati ponor podatkov za nadzorni sistem. Vozlišča s to funkcijo imenujemo tudi *bazna vozlišča*.

1.3.2 Bazna vozlišča

Vozlišča, na katere je priključeno nadzorni sistem, bomo imenovali *bazna vozlišča* in zanje uporabljali kratico BS. Glavna naloga teh vozlišč je zbrati vse podatke iz senzorjev in jih prenesti v nadzorni sistem, zato ponavadi nimajo vgrajenih senzorjev. Ta

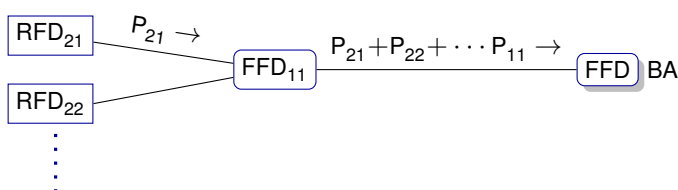
BS: Base Station

naloga jim daje privilegirani položaj, saj lahko z analizo sprejetih podatkov ugotovijo število vozlišč in pot podatkov med senzorji in baznim vozliščem (Slika 1.3-3).

1.3.3 Topologija

Topologija (iz grščine "topos" = kraj, mesto in "logos" = študij) je matematična disciplina, ki proučuje lastnosti prostora. V računalniškem svetu pa je topologija definirana kot povezovanje elementov (naprav, vozlišč...) z povezavami (žičnimi ali brezžičnimi) v omrežja.

V senzorskih brezžičnih omrežjih s topologijo opišemo strukturo komunikacijskih povezav med vozlišči. Prikažemo jo z grafi (Slika 1.3-3).



Slika 1.3-3
Prenos podatkov preko FFD.

Ker pri brezžičnih omrežjih ni fizičnih povezav z vodniki, topologijo določajo logične povezave med njimi. Postopki določanja teh povezav je glavna tema raziskav, na katerih temelji ta disertacija.

1.3.4 Usmerjanje

Usmerjanje je v podatkovnih omrežjih proces izbire poti v omrežju, po kateri si dva uporabnika izmenjujeta podatke oziroma eden pošlje podatke drugemu. Usmerjanje vršijo usmerjevalni protokoli, ki pa za svoje delo morajo poznati topologijo omrežja.

1.3.5 Samoorganizacija brezžičnih senzorskih omrežij

Od senzorskih brezžičnih omrežjih pričakujemo, da so senzorji se sposobni sami organizirati v omrežje. V tem postopku se določi topologija omrežja. Podobno lastnost imajo mobilna ad-hoc omrežja (MANET) [43]. Zato lahko na brezžična senzorska omrežja gledamo kot na posebni tip ad-hoc omrežij. Posebnosti izhajajo iz namena obeh omrežij

MANET: Mobile Ad-hoc
...NETworks

Tabela 1.3-1
Primerjava brezžičnih senzorskih in MANET omrežij.

	BSO	MANET
Normalna uporaba	sistem zbiranja podatkov	porazdeljeni računalniški sistemi
Uporabniki	običajno samo en uporabnik	množica enakopravnih uporabnikov
Uporabniki	običajno samo en uporabnik	množica enakopravnih uporabnikov
Število naprav	tudi do več tisoč	nekaj deset
Fizični naslov	več vozlišč ima lahko isti naslov	vsaka naprava ima unikatni naslov
Odvisnost od uporabe	da	ne

1.3.6 Metrika

Pri ocenjevanju lastnosti brezžičnih senzorskih omrežij s komunikacijskega vidika bomo uporabljali cenilke, s katerimi bomo te lastnosti objektivno izmerili. Glede na posebnosti teh omrežij smo si izbrali naslednje mere – našteje so po svoji pomembnosti:

- skupna moč napajanja vseh senzorjev
- skupna energija napajanja vseh senzorjev
- variacija razmerja *moč napajanja sensorja/skupna moč napajanja omrežja*
- latentni čas
- prenosna zmogljivost omrežja

1.4 Uporabnost in perspektiva brezžičnih senzorskih omrežij

Brezžična senzorska omrežja so v zadnjem desetletju doživela velik razcvet. Od pionirskih projektov, med njimi so veliko pozornost vzbudili projekti "SmartDust" [91], "WiSeNts" [119], "Eyes" [41], je razvoj polprevodniške tehnologije, komunikacijskih tehnik dramatično izboljšal uporabnost brezžičnih senzorskih omrežij. Z njimi danes lahko merimo in nadzorujemo tam, kjer prej, zaradi fizičnih ali drugih omejitev, to ni bilo mogoče.

Pred razmahom uporabe brezžičnega prenosa podatkov je veljala paradigma o gradnji velikih, robustnih naprav za zajemanje podatkov, ki so poleg senzorskega dela vsebovale še ogromno napajalno enoto in enote za lokalno shranjevanje podatkov (tračne ali diskovne). Team tehničnega osebja in znanstvenikov je bil prisiljen postaviti senzorsko enoto in se čez čas vrniti po podatke. Če se je medtem zgodila kakšna okvara ali nesreča je tehnično osebje zadnjo zvedelo šele, ko se je vrnilo po omenjeno napravo. Poleg izgube naprave pa je še bolj pomembna izguba podatkov in časa, ko naprava ni delovala.

Z uporabo brezžične tehnologije pa se je paradigma spremenila: namesto velikih, dragih naprav se je pojavila želja po cenenih, majhnih in energijsko učinkovitih napravah senzoriranja in zajemanja podatkov. Na enkrat bi lahko sodelovalo več sto ali tisoč naprav, ki bi zajemale podatke in jih pošiljale do uporabnika. V tem primeru seveda ne potrebujemo nobenega lokalnega shranjevanja podatkov, napake ali nesreče senzorske enote pa so takoj javljene do uporabnika. Druga dobra lastnost je ta, da v primeri izpada ene ali par senzorskih enot škoda oziroma izpad podatkov zanemarljiv. Individualna kalibracija senzorjev ni več potrebna saj se senzori glede na lokacijo (npr. lokalizacijski algoritmi [113]). Tudi polaganje (deploy) brezžičnih senzorskih enot na področje senzoriranja je lažje (saj jih lahko na primer preprosto zmečemo iz letala).

Za primer predpostavimo preprosto aplikacijo: nadzorovanje habitata na Pohorju. Denimo, da spremljamo vlago, temperaturo, osvetljenost in podobne parametre. Z mislijo na ta primer bomo definirali *senzorsko vozlišče (ali skrajšano vozlišče) oziroma senzorsko enoto (node)*. V tezi bomo termin senzorsko vozlišče uporabljali za preproste, baterijsko napajane, brezžične komunikacije sposobne naprave. Njihova

glavna lastno bo zaznavanje fenomenov in pošiljanje podatkov o le teh do uporabnika preko več etapnega omrežja. Termin *veliko število senzorskih enot* bo opisoval nedoločno število senzorskih vozlišč ranga med par sto do par tisoč vozlišč.

Brezžična senzorska omrežja bodo v naslednjih letih zagotovo predstavljala eno najpomembnejših orodij na področju sensoriranja. Njihova uporaba v aplikacijah skoraj ne pozna meja. Trenutno so vozlišča verjetno najmanjše in najcenejše "inteligentne" naprave, ki se uporabljajo v tehniki sensoriranja. Povzeli bi lahko, da so brezžična senzorska omrežja eden najbolj obetajočih inteligentnih sistemov za zajemanje podatkov [12].

1.4.1 Vrste brezžičnih senzorskih omrežij

Brezžična senzorska omrežja se vse bolj razvijajo v smeri "pametnega okolja" [9]. Primeri teh prizadevanj so projekti "pametna hiša" [9, 17], "pametne ceste" in drugi. V vseh so senzori ključni sestavni del, ki zagotavlja informacije o stanju okolja. Pri tem je kompleksnost njihovega komunikacijskega sistema odvisna od dveh činiteljev:

1. velikosti omrežja
2. specifične uporabe omrežja

Velikost omrežja je povezana z velikostjo področja, ki ga brezžično senzorsko omrežje pokriva [38]. Tu ločimo tri glavne kategorije:

- *Mala omrežja.* V tej skupini so omrežja nameščena v inteligentnih zgradbah, skladiščih in podobno, kje bi vse predmete v skladišču ali hiši opremili s senzori.

V trgovinah in skladiščih se na primer za avtomatizirani sistem vodenja evidence in za varnost pred krajo množično uporablja tehnika RFID. Zamenjav pasivnih elementov RFID s brezžičnimi senzorskimi omrežji omogoči povsem novo paradigmo spremljanja in iskanja blaga v skladišču [3, 102], kjer poleg ugotavljanja njihove prisotnosti lahko izmerimo še temperaturo, vlago, premikanje ali katero drugo stanje ali dogajanje z opazovanim predmetom.

- *Srednja omrežja.* V tej skupini so na primer omrežja, ki jih uporabljamo za nadzor habitata. Z njimi na primer merimo temperaturo, vlago, ali druge parametre okolja. To je trenutno najbolj razširjena uporaba brezžičnih senzorskih omrežij.
- *Velika omrežja* Primer velikih omrežij so pametne avtoceste, pametni promet v mestih in drugo. Na primer v pametnih cestah lahko brezžična senzorska omrežja merijo vremenske pogoje in gostoto prometa na cesti in te podatke sporočajo nadzornim sistemoma in direktno uporabnikom ceste.

Druga delitev BSO pa temelji na aplikacijah, ki jih BSO ponujajo [9]:

- *Vojaške aplikacije*
Brezžična senzorska omrežja so zaradi svojega načina delovanja (samorganizacije in odpornosti na izpade senzorjev) idealna za vojaške aplikacije. V vojaških aplikacijah jih lahko uporabimo za nadzorovanje bojnega polja, terena ali preprosto kot varnostne mehanizme v domači bazi. The Defense

Advanced Research Projects Agency (DARPA) klasificira BSO kot najbolj potencialno področje [12].

- *Okoljske aplikacije*

Veliko živalskih vrst in njihovega okolja lahko proučujemo z uporabo brezžičnih senzorskih omrežij. Sledimo lahko selitvam ptic ali drugih ogroženih vrstam živali. Opazovati je mogoče temperaturo in vlažnost prsti na določenem področju in podobno. Trenutno obstaja mnogo projektov, ki se ukvarjajo z proučevanjem okolja: Great Duck Island [48] ali Zebre v Keniji [122].

- *Medicinske aplikacije in aplikacije varovanja zdravja*

Zanimive aplikacije lahko BSO ponudijo starejšim in invalidom. Nadzorovanje vitalnih funkcij lahko poteka tako na domu kot v bolnišnici. Senzorji lahko tako sprožijo alarm takoj, ko zaznajo kakršno koli spremembo pri stanju pacienta. Določeni senzorji omogočajo nadzor gibanja in lahko sprožijo alarm v primeru padca ali podobne nevarnosti.

- *Aplikacije domačega okolja*

Domače okolje je še posebej zanimivo za brezžična senzorska omrežja. Predstavljajmo si, da so se električne komponente sposobne povezovati v omrežje in delovati kot celota v korist uporabnika. Istočasno pa lahko senzorska vozlišča zaznavajo nepričakovane dogodke.

- *Druge komercialne aplikacije*

Sem spadajo vse, zgoraj ne opredeljene aplikacije

Brezžična senzorska omrežja bi lahko mogoče razdelili še v dve skupini: *"grobno zrnata" brezžična senzorska omrežja*, kjer vsako senzorsko vozlišče pokriva široko področje (več 100m). Kompleksnost in cena omenjenega vozlišča je visoka. *"fino zrnata" brezžična senzorska omrežja* pa pokrivajo manjša področja (do več 10m), kompleksnost naprav je manjša, integrirani senzorji pa so navadno nizko cenovni. Topologija takega omrežja je večina več-etapna. Večje število senzorskih vozlišč kompenzira nizko kvaliteto senzorjev in rezultira v visoko kvalitetne rezultate.

1.4.2 Raziskovalni izzivi v brezžičnih senzorskih omrežjih

V kratkem poskušajmo povzeti izzive, s katerimi se spopadamo pri BSO:

- **Kvaliteta komunikacije:** BSO morajo včasih delovati pod ekstremnimi pogoji (recimo, pri namestitvi v naravi) kot so sneg, dež, ogenj... Spremembe teh pogojev lahko vplivajo na delovanje sistema, ki postane zelo nestabilen. Želja je ohraniti stabilen sistem v vseh pogojih delovanja.
- **Sovražno okolje:** Senzorske enote so lahko poškodovane ali uničene (pri vojaških aplikacijah lahko tudi namerno), zato morajo biti protokoli v BSO robustni na izpad ene ali več enot. Senzorske enote pa morajo biti posebej prilagojene sovražnemu okolju.
- **Energijska poraba:** Načrtovalci poskušamo načrtovati BSO tako, da je poraba energije nizka kot je le mogoče. Nekatere senzorske enote morajo delovati tedne ali celo mesece brez menjave baterije (pri nekaterih menjava fizično sploh ni mogoča), zato je veliko truda vloženega v raziskovanje energijske učinkovitosti. Četudi energijska učinkovitost ni vedno prva prioriteta, jo poskušamo upoštevati kolikor je le mogoče.

QoS: Quality of Service

- **Skalabilnost:** Zaradi velikega števila senzorskih enot (tudi do par 1000), ki sestavljajo BSO je skalabilnost zelo pomembna.
- **Nizka razpoložljivost virov:** Zaradi cene in energijske potrošnje so mikrokontrolerji na senzorjih (navadno) zelo omejeni pomnilnikom in računsko močjo. Pasovna širina, ki jo senzorji uporabljajo za komunikacijo je tipično nizka. Zaradi tega morajo biti BSO protokoli učinkoviti in včasih z vgrajenim QoS (Quality of Service) sistemom.
- **Omejitve strojne opreme:** Na tržišču je trenutno na razpolago samo par končanih senzorskih enot, zato je velikokrat na raziskovalcih, da izdelajo svoje senzorske enote.
- **Podatkovni tok:** Energijska omejitev in majhna pasovna širina zahtevata, da so podatki posebej pripravljeni (stisnjeni ali združeni), da zmanjšamo podatkovni tok.

Iz izzivov opisanih zgoraj lahko sklepamo, da raziskave na področju BSO potekajo na več nivojih [60, 107] različno intenzivno. tako lahko iz literature zasledimo, da je porazdelitev raziskovalnih interesov v BSO naslednja: 9,7% objavljenih člankov je obravnavalo tematiko postavitve senzorskih omrežij, 7,27% se je posvetila sledenju opazovanih objektov in okoli 6,06% objav je bilo namenjenih lokalizaciji. Avtorji so v 6,05% prispevali k zbiranju podatkov, 5,8% pa se je posvetilo usmerjanju in združevanju prometa [25].

Raziskave bomo poskušali uvrstiti v naslednje kategorije:

■ Raziskave strojne platforme

Vizija s katero se ukvarjajo BSO temelji na ideji dolgega delovanja brezžičnih vozlišč, ko so le ta razporejena. Intenzivne raziskave se izvajajo na razvoju nizko cenovnih, energijsko nizko potrošnih raijskih modulov manjšega dometa. Raziskave so med drugim usmerjene v nizko energijsko nizko potrošne mikroročunalnike čim višjih preformanc. Razvoji senzorjev in nenazadnje razvoj kompletnih brezžičnih senzorskih vozlišč lahko štejemo med raziskave strojne platforme.

■ Omrežni protokoli

Da lahko senzorska vozlišča komunicirajo med sabo potrebujejo uporabne omrežne protokole (protokole dostopa do medija – MAC, protokole topologije, protokole usmerjanja prometa, protokole gruč in podobno), ki so robustni. Več vrst redundanc (tudi dinamične) je potrebno, da zagotovimo nemoteno delovanje omrežja ob izpadu enot, napadu na enote in podobno. Pametni omrežni protokoli lahko prihranijo ogromno energije in tako podaljšajo življenjsko dobo omrežja.

■ Drugi specifični protokoli

Velika količina podatkov je neuporabna, če ne vemo kdaj so bili podatki poslani in od kod. Protokoli časovne sinhronizacije in lokalizacije so torej ključnega pomena. Protokole lokalnega delovanja v primeri izpada omrežja in protokole združitve z omrežjem lahko prav tako štejemo med specifične protokole.

■ Podatkovni protokoli in shranjevanje podatkov

Glavna naloga BSO je zaznavanje različnih fenomenov, kar pa lahko v večjem

BSO proizvede veliko količino podatkov. Kritična količina podatkov nastopi, ko le ti presežejo pasovno širino omrežja ali zapolnijo sisteme shranjevanja podatkov. Edina možnost, da se omenjeni težavi ognemo je pametno zaznavanje, komprimiranje in združevanje podatkov. Senzorska vozlišča se morajo navadno dinamično prilagoditi pogojem na omrežju.

- **Visoko nivojska desiminacija**

Ko imamo enkrat podatke na voljo se postavi vprašanje kako uporabiti le te. Nad omrežno plast navadno dodamo servis, ki uporabniku sporoča kaj je omrežje sposobno izvajati in kako. Med visoko nivojsko desiminacijo bi lahko šteli tudi prehod med BSO in široko pasovnim omrežjem v katerega pretakamo podatke.

- **Aplikacijska domena** Vsak uporabnik BSO definira aplikacijo, za namen katere bo omrežje uporabil. Večina uporabnikov BSO se ukvarja le z aplikacijami na BSO, ostale nivoje pa prepustijo komunikacijskim strokovnjakom.

Raziskave in teza je fokusirana na dve izmed prejšnjih kategorij: Omrežne protokoli in druge specifične protokole. Drugih kategorij se v tezi neposredno ne dotikamo čeprav smo jih raziskovali.

1.5 Formulacija problema

Zbiranje podatkov in posredovanje le teh do uporabnika je glavna naloga brezžičnih senzorskih omrežij. Senzorske enote zbirajo podatke in le te pošiljajo do ene ali več vozlišč imenovanih bazna postaja z uporabo več etap. Pošiljanje sporočila od vsake senzorske enote, da bazne postaje omogočajo **usmerjevalni algoritmi**.

Aplikacijska odvisnost brezžičnih senzorskih omrežij predpostavlja uporabo strogo namenskih, namesto splošnih protokolov ISO/OSI referenčnega modela.

Zaradi prej omenjene paradigme lahko v BSO najdemo mnogo strogo namenskih protokolov usmerjanja prometa. Večina le teh temelji na principu zmanjševanja porabe energije in povečanju učinkovitosti.

Aplikacije, ki potrebujejo dostavo sporočil z nizkimi zakasnitvami v osnovi ne zahtevajo energijske učinkovitosti protokola, saj sta pojma med seboj kontradiktorna. Senzorska omrežja, uporabljena v aplikacijah, ki potrebujejo nizke zakasnitvene čase imenujemo "realno časovna brezžična senzorska omrežja" (ang.:Rt-WSN). Rt-WSN navadno predvidevajo scenarije, ko je prometa v omrežju relativno malo, izbruhi pa se dogajajo lokalno in izolirano. Rt-WSN usmerjevalni algoritmi pa v večini ne upoštevajo primera, ko je prometa na omrežju izjemno veliko, pa kljub temu potrebujemo nizke zakasnitve.

Zato torej predpostavljamo, da v primerih težkega prometa znani pristopi niso zadovoljivi: poleg paradigme realnega časa je potrebno upoštevati še uravnovežitev prometa. Posledično pa pričakujemo, da bodo zaradi uravnovežitve prometa tudi energijske razmere boljše, kot pri algoritmih, ki uravnovežitve ne podpirajo.

Posledično v nalogi predlagamo prometno uravnovežen protokol usmerjanja prometa z nizkimi zakasnitvami (ang.: Latency), ki rešuje problem težkega (ang.:heavy) omrežnega prometa in sprotnega realnega časa.

Rt-WSN: Real time Wireless
Sensor Networks:

1.6 Pregled stanja raziskovalnega področja teze

Kot smo že omenili so brezžična omrežja, zaradi hitrega razvoja procesorjev in tehnologij radija deležna vse večje prozornosti pri raziskavah. Posebna veja: brezžična senzorska omrežja pa so zaradi svojega širokega spektra uporabe postala zelo razširjena in uporabna.

V koraku z brezžičnimi senzorskimi omrežji pa se morajo razvijati tudi protokoli in algoritmi plasti ISO/OSI referenčnega modela. Med drugim je veliko naporov in raziskav usmerjenih v raziskave topologij omrežja in usmerjevalnih protokolov brezžičnih senzorskih omrežij. Kot bomo natančneje razložili v (Poglavje 3) lahko algoritme usmerjanja delimo na:

- *proaktivne protokole usmerjanja* ali tabelarične protokole, kjer so usmerjevalne poti znane vnaprej. Slaba lastnost omenjenih protokolov je potreba po nenehnem periodičnem vzdrževanju poti, kar prinese precej servisnega prometa (ang.:overhead traffic). Najbolj znani predstavniki proaktivnih protokolov so DSDV [95], OLSR [105] in WRP [89].
- *reaktivne protokole usmerjanja*, ki ohranjajo poti samo in zgolj, ko je to potrebno. Problem se pojavlja pri velikih zakasnitvah sporočil in reagiranju na izpade povezav. Znani predstavniki reaktivnih protokolov so AODV [33], DSR [57] in podobni.
- *Hibridni protokoli usmerjanja*, združujejo pozitivne lastnosti proaktivnih in reaktivnih protokolov usmerjanja prometa. Predstavniki so ZRP [46] ali VBR [80] protokoli ...

Brezžična senzorska omrežja navadno prenašajo relativno majhno količino prometa kot primerjajoča se MANET omrežja vendar pa se v BSO večkrat pojavijo izbruhi prometa, kar ni tipično za MANET omrežja. V BSO imamo vedno vozlišče, ki služi kot zbirališče podatkov – bazna postaja (ena ali več) (ang.: base-station, gateway) kamor vse ostale enote pošiljajo podatke. Glede na omenjeno se pojavi logično sklepanje o strukturi drevesa. Večina drevesnih algoritmov usmerjanja se ponavadi klasificira kot proaktivnih. Nekateri algoritmi, kot recimo Plus-Tree algoritem [93] pa predlagajo določene rešitve v stilu hibridnih protokolov usmerjanja prometa.

Drugi možni pristop predstavlja model srednjih osi (ang.:Medial-Axis), omogoča učinkovito področno usmerjanje prometa v primerih, ko je znana topologija in geometrijsko področja (omejitev in razporeditev) senzorjev. Senzorske enote na Medial-Axis predstavljajo komunikacijsko hrbtenico, vse ostale senzorske enote pa so na hrbtenico povezane z uravnoveženim drevesom najkrajših poti [54].

Pristop s pametnimi antenami predlaga algoritem usmerjanja, ki izboljšajo robustnost BSO na interferenčne motnje in maksimirajo pretok podatkov skozi omrežje, če so znane krajevne smeri sosednjih vozlišč in če so ta opremljena s pametnimi antenami. V tem primeru lahko pri usmerjanju prometa upoštevamo tudi krajevno prepletanje senzorjev in se izogibamo interferenčnim motnjam [5]. Pri usmerjanju radio-frekvenčnega valovanja oziroma usmerjanja sprejema je krajevna in logična topologija BSO opisana z grafi. S to predstavitvijo je razložena formulacija optimizacijskega problema kot problem mešanega celoštevilčnega programiranja (MIP: Mixes Integer Programming). Njegova rešitev je model usmerjevalnega protokola. Za njegovo porazdeljeno izvajanje je ocenjena količina dodatnega prometa za prenos informacij potrebnih za izvajanje usmerjevalnega algoritma. Posebej je poudarjen

problem kompleksnosti algoritma, ki nastane pri sočasnem posredovanju/sprejemu prometa iz več smeri [5].

Obratne možne rešitve opisujejo in predlagajo statično ter dinamično postavitev (pozicioniranje) vozlišč. Skrbno načrtovanje predhodne izbire pozicije vozlišč je lahko ključnega pomena za optimizacijo ali doseganje zelenih parametrov. Izbran je obratni pristop kot je v navadi: namesto prilagajanja algoritmov topologiji in vozliščem, le ta prireja pozicijo vozlišč algoritmom [121].

Greedy Perimeter Stateless Routing (GPSR) algoritem usmerjanja uporablja poznavanje pozicije usmerjevalnika in destinacijo paketa za odločitev smeri posredovanja. Algoritem se obnese bolje kot algoritem najkrajše poti, sploh, ko gostota omrežja naraste [64].

V zadnjih letih se vse pogosteje omenja tudi termin realno časovna brezžična senzorska omrežja. Tu večina raziskovalcev predpostavlja, da je življenjska doba paketa paketa omejena in da mora paket, v kolikor še velja za uporabnega, prispeti na cilj preden se ta življenjska doba izteče [7]. Implementacija QoS sistemov v BSO v zadnjih letih postaja bolj intenzivno raziskano področje [7, 14, 85], vendar pa kljub temu strokovna javnost "real time BSO" ne enačijo z terminom "QoS BSO" [7]. Kljub temu pa tudi v načrtovanju Rt-BSO velja paradigma o varčevanju z energijo [7, 85].

1.7 Področja in prispevki v tezi

Problem s katerim se bomo ukvarjali v tezi predstavlja usmerjanje prometa po statičnih brezžičnih senzorskih omrežjih. Raziskali bomo fizične in logične topologije, jih opisali z grafi in s pomočjo teorije grafov poiskali optimalne rešitve glede na izbrane metrike in izbran razred brezžičnih senzorskih omrežij.

Trenutno je znanih več kot 60 algoritmov usmerjanja prometa. Med njimi le redki upoštevajo kakovost storitev in uravnoteževanje prometa [32]. Z uravnoteženjem prometa v BSO se intenzivno ukvarja več raziskovalcev [55, 81, 84, 101]. Večina predlaga uporabo dinamičnih algoritmov iskanja poti po grafu (iskanje poti, ko je le ta potrebna), podobno kot to počneta DSR ali AODV algoritma, drugi poskušajo uporabiti geografske lastnosti ali lastnosti poznavanja okolice.

Pri uravnoteženju prometa poskušamo promet enakomerno porazdeliti čez vse veje drevesa, ki ga uporablja usmerjevalni protokol. Pri statičnem modelu usmerjanja prometa lahko za doseganje prometnega uravnoteženja lahko uporabimo enega od algoritmov za uravnoteženje drevesa vendar pa nobeden od algoritmov ne zagotavlja najkrajše poti. Enako tudi nobeden od algoritmov za zagotavljanje najkrajše poti (Floyd's ali Dijkstra's algoritem) ne zagotavlja enakomerne obremenitve povezav [32].

Za usmerjevalne protokole v BSO je zaželeno, da so čim manj odvisni od uporabe BSO. Njihovo učinkovitost merimo z metriko učinkovitosti [11]. Ta je kompozit propustnosti omrežja, zakasnitev sporočil, življenjsko dobo omrežja, zakasnitev celotnega omrežja, energijsko porabo omrežja in podobno [11]. Metriko učinkovitosti ponavadi ugotavljamo s simulacijami. Za simulacije brezžičnih senzorskih mrež je na voljo precej simulatorjev (NS2, OPNET, OMNET++, GloMoSim...) vendar se lahko njihovi rezultati medsebojno razlikujejo [31]. Trenutno, v literaturi ne obstaja novejša reference, ki bi se ukvarjala z natančnostjo simulatorjev za omrežja.

Večjo učinkovitost usmerjevalnih protokolov dosežemo z *upravljanjem topologije* [16, 100]. Z njim lahko močno vplivamo na učinkovitost omrežja in energijsko porabo. S spreminjanjem oddajne moči oddajnikov lahko na primer spremenimo

stopnjo vozlišča [16].

Ker ima večina centralnih optimizacijskih algoritmov, to je algoritmov, ki gradijo vpeto drevo, za uravnoteženje prometa in iskanje najkrajših poti NP računsko zahtevnost, se v iskanju rešitev oziramo po heurističnih metodah ali pa na najbolj podobne probleme, za katere so znane rešitve v polinomskem času.

Cilj doktorske disertacije je raziskati možnosti razvoja usmerjevalnih algoritmov na osnovi metod polprirejanja grafov, ki optimirajo zastavljene metrike brezžičnih senzorskih omrežij.

1.7.1 Pričakovani prispevki v tezi

Življenjska doba BSO je odvisna predvsem od kapaciteta baterij v senzorjih in potrošni energije med delovanjem BSO. Na potrošnjo energije pa vpliva dolžina budnega stanja senzorjev, ki je odvisna tudi od hitrost dostave sporočil, enakomerne porazdelitve prometa po vejah BSO in števila etap med senzorjem in BP. Zato kot tezo disertacije trdimo:

Z uravnoteženo usmerjanjem prometom po brezžičnem senzorskem omrežju optimiramo njegove lastnosti, ki jih merimo z metrikami:

- skupna moč vseh senzorjev
- variacija razmerja *moč senzorja/skupna moč omrežja*
- latentni čas
- prenosna zmogljivost omrežja

V disertaciji pričakujemo naslednje prispevke k znanosti:

Prvi prispevek *Izboljšava algoritma za določanje polprirejanja. Predpostavljamo, da bo izvajanje algoritma hitrejše kot je izvajanje referenčnih algoritmov.*

Drugi prispevek *Nov učinkoviti algoritem usmerjanja prometa. Algoritem usmerjanja bo temeljil na vpetem drevesu, katerega bomo izračunali z algoritmom polprirejanja in algoritmi iskanja v širino.*

Pričakujemo bolj uravnotežen promet, daljšo življenjsko dobo omrežja in hitrejšo dostavo sporočil.

Tretji prispevek *OPNET simulacijski model prototipa brezžičnega senzorskega vozlišča SPaRCMosquito.*

1.7.2 Uporabljena orodja v tezi

Za pripravo teze smo v večini uporabili naslednja orodja:

- **Visual studio 2008 in Visual studio 2010**, kjer je nastajalo in še nastaja orodje za brezžična senzorska omrežja imenovana SPaRCSoft.
- **SPaRCSoft**, razvito lastno orodje za modeliranje, evaluacijo in interakcijo z brezžičnimi senzorskimi omrežji.
- **OPNET**, orodje za simuliranje brezžičnih senzorskih omrežij, v katerem smo razvili svoj lastni model senzorskega vozlišča.

- Keil uVision, orodje za programiranje in razhroščevanje ARM7 in Cortex procesorjev (procesor na SPaRCMosquito brezžičnem senzorskem vozlišču).
- WinEdit in Latex (MikTex), s katerim je nastal tekst teze.

1.8 Sprehod po tezi

Teza je razdeljena v 8 glavnih poglavij in dva dodatka.

V prvem poglavju (Poglavje 1) – Uvodu povzamemo brezžična senzorska omrežja in smernice raziskav v BSO. Nadaljujemo s formulacijo problema, pregledom stanja področja in napovedjo prispevkov v tezi.

V drugem poglavju (Poglavje 2) se posvetimo za tezo potrebni teoretičnim osnovam teorije grafov. V poglavju definiramo drevesa in predstavimo algoritme pregledovanja grafov (v globino in širino) ter iskanje najkrajših poti po grafih.

Teoretične osnove usmerjanja prometa v brezžičnih senzorskih omrežjih predstavimo v tretjem poglavju (Poglavje 3). Tu izpostavimo še topologijo in vpliv fizične plasti na višje plasti (MAC, usmerjanje prometa...).

Matematični model in prometno uravnoteženost predstavimo v poglavju štiri (Poglavje 4). V kratkem povzamemo matematične modele omrežja, ki se v raziskovalni skupnosti uporabljajo, ter posebej še, drevesne strukture v matematičnih modeli, saj teza temelji strukturah drevesa.

V poglavju 5 se posvetimo predstavitvi algoritmom optimalnega polprirejanja in razložimo, kako bomo algoritme polprirejanja uporabili v usmerjevalnem protokolu. Postavimo teoretična izhodišča algoritmov optimalnega polprirejanja in predlagamo uporabo alternativnega, hitrejšega algoritma polprirejanja. V poglavju analiziramo hitrost in delovanje predlaganega algoritma polprirejanja proti drugima referenčnima algoritmoma. Na kratko povzamemo še algoritme polprirejanja za utežene grafe.

Poglavje 6 predstavlja jedro naše teze: v poglavju predstavimo idejo usmerjevalnega protokola ter predpostavke in omejitve. Predstavimo osnovno ter vpete topologijo, ki jih bo usmerjevalni protokol uporabljal za svoje delovanje. Predstavimo še identifikacijo in naslavljanje vozlišč. Na kratko predstavimo delovanje protokola, ki smo ga poimenovali **SPaRCRoute** in njegove dopolnitve. Razložimo, da protokol SPaRCRoute za svoje delovanje uporablja še EYES protokol, ki deluje kot alternativna rešitev za tip komunikacije, ko SPaRCRoute ni primeren.

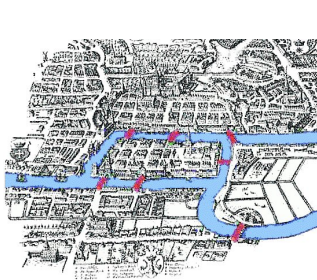
Naslednje poglavje (poglavje št. 7) predstavlja analizo protokola, ki je sestavljena predvsem iz simulacij izdelanih v OPNET programskem okolju. V poglavju na kratko predstavimo model vozlišča in omrežja, ter vpliv MAC plasti na rezultate simulacij. Nadaljujejo s simulacijami protokola v skupnih in različnih kolizijskih domenah z uravnoteženim in neuravnoteženim drevesom vozlišč. Poglavje zaključimo z rezultati simulacij dveh manjših senzorskih omrežij.

V poglavju 8 podamo zaključne misli.

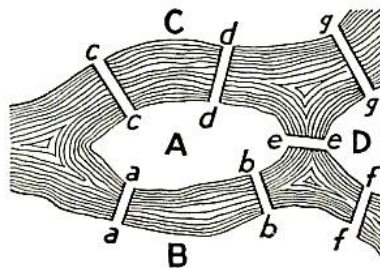
Dodatna poglavja (Poglavje B) sta namenjena predstavitvi SPaRCMosquito vozlišča in SPaRCSoft programskega orodja ter grafičnim rezultatov simulacij (Poglavje C), katere podatke smo uporabili v poglavju 7.

Teorija grafov

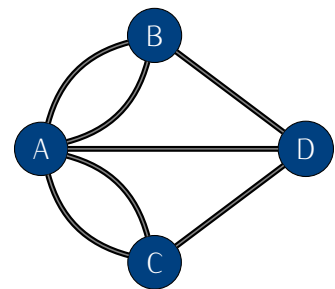
Za prvi problem iz teorije grafov štejemo problem iz leta 1736, ko je Euler rešil problem Königsberg-ških mostov. Problem, ki ga je Euler reševal je bil na prvi pogled izjemno preprost: ali obstaja takšen sprehod, po katerem bi vse Königsberg-ške mostove prečkali natanko enkrat in se po sprehodu vrnili na isto, začetno točko? Isti problem je Euler objavil tudi v prvem delu iz teorije grafov [90].



(a) Königsberg mesto



(b) Königsberg mostovi



(c) Königsberg ponazoritev z grafom

Slika 2.0-1

Problem Königsberg mostov.

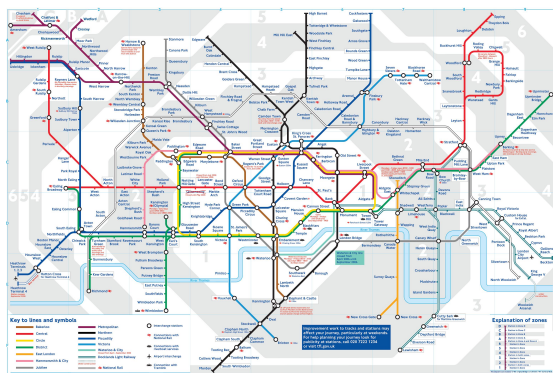
Ena prvih knjig na temo teorije grafov je bila izdana leta 1936 ("Theorie der endlichen und unendlichen Graphen") [66]. Od takrat pa do danes se je teorija grafov razvila kot posebna veja matematike. Teorija grafov se ukvarja z problemi "entitet, ki so povezane z drugimi" entitetami, kar pomeni, da je teorijo grafov možno uporabiti skoraj povsod.

Teorijo grafov lahko med drugim uporabimo v optimizacijskih problemih (koliko vlakov moramo zamenjati, da pridemo iz točke a v točko b (Slika 2.0-2a)), ena od bolj znanih aplikacij, ki uporablja teorijo grafov pa je zagotovo GPS navigacija (Slika 2.0-2b).

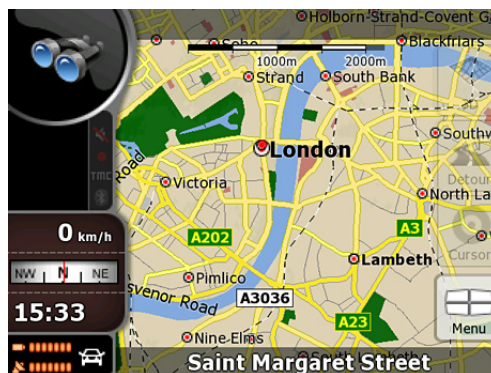
2.1 Osnovni pojmi

Grafi so matematične strukture, ki jih uporabljamo za modeliranje objektov in relacij med njimi [116]. Graf $G = (V(G), E(G))$ je določen z množico vozlišč $V(G)$ in

$G = (V(G), E(G))$:
Graf



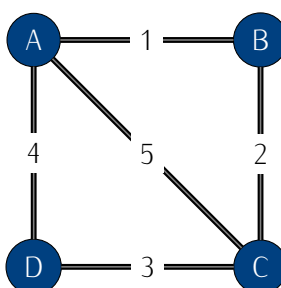
(a) Zemljevid podzemne železnice v Londonu



(b) Zemljevid GPS navigacije

Slika 2.0-2

Uporaba teorije grafov.

**Slika 2.1-1**

Primer preprostega grafa G .

povezav $E(G)$ [37].

Kadar je jasno, kateri graf obravnavamo bomo le tega predstavili s krajšim zapisom kot $G = (V, E)$, kjer bo pri nas število vozlišč $|V|$ in število povezav $|E|$ vselej končno število. (Slika 2.1-1)¹ predstavlja enostaven graf, to je graf brez zank in večkratnih povezav. Včasih namreč dovolimo, da je med dvema vozliščema več kot ena povezava. Mi se bomo omejili predvsem na enostavne grafe.

Vozlišča na grafu G (Slika 2.1-1) zapišimo kot:

$$V = \{A, B, C, D\} \quad (2.1)$$

Ter povezave kot:

$$E = \{AB, BC, CD, DA, AC\} = \{1, 2, 3, 4, 5\} \quad (2.2)$$

Graf pa lahko med drugim predstavimo tudi z **matriko sosednosti** in **incidenčno matriko** [37].

Matrika sosednosti nam pove ali sta vozlišči med seboj sosednji; tedaj je vrednost elementa v matriki enaka ena, v nasprotnem primeru pa je vrednost v matriki navadno 0. Stolpce in vrstice v matriki predstavljajo vozlišča v grafu. Iz incidenčne matrike lahko razberemo ali je določena povezava omejena z določenim vozliščem ali ne. Če je vozlišče krajišče povezave, je vrednost v matriki enaka 1, v nasprotnem primeru pa 0.

¹ Do preklica bomo z številkami na povezavah označevali povezave in ne uteži povezav!

Poglejmo primer matrike sosednosti (enačba 2.4) in incidenčne matrike (enačba 2.4) za (Slika 2.1-1):

$$M_S = \begin{matrix} & \begin{matrix} A & B & C & D \end{matrix} \\ \begin{matrix} A \\ B \\ C \\ D \end{matrix} & \begin{bmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \end{bmatrix} \end{matrix} \quad (2.4)$$

$$M_I = \begin{matrix} & \begin{matrix} 1 & 2 & 3 & 4 & 5 \end{matrix} \\ \begin{matrix} A \\ B \\ C \\ D \end{matrix} & \begin{bmatrix} 1 & 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 \end{bmatrix} \end{matrix} \quad (2.5)$$

Število povezav v vozlišču imenujemo *valenca* ali *stopnja vozlišča*. Označimo jo z $d_G(z)$, kjer G predstavlja graf, z pa določeno vozlišče. Stopnja vozlišča A na (Slika 2.1-1) je torej $d_G(A) = 3$, stopnja vozlišča B znaša $d_G(B) = 2 \dots$

Grafu G lahko določimo še dve pomembni značilnosti: *red grafa*:

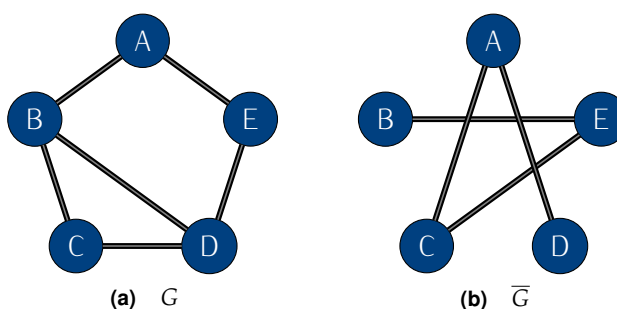
$d_G(A)$: Stopnja vozlišča
(A v grafu G)

$$\nu_G = |V| \quad (2.6)$$

ter *velikost grafa*:

$$\epsilon_G = |E| \quad (2.7)$$

Grafe lahko uporabimo tudi za modeliranje oz. modele s katerimi opišemo dane situacije. Predpostavimo množico 5 ljudi za katere bomo definirali njihove znance in neznance.



Slika 2.1-2

Graf G in njegov komplementarni graf \bar{G} .

Prej opisan problem modeliramo na (Slika 2.1-3), kjer graf G predstavlja model znancev in graf \bar{G} njegov komplement: model neznancev.

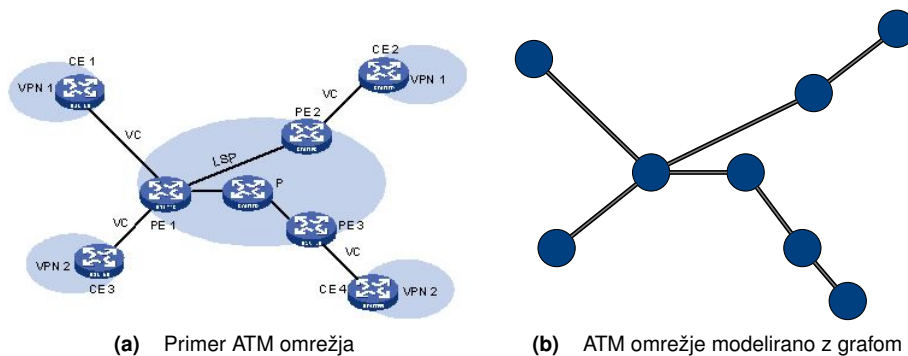
Zapišemo lahko torej, da je *komplement* \bar{G} grafa G graf z množico vozlišč $V(G)$, definiranih z $uv \in E(\bar{G})$ natanko takrat, ko velja, da $uv \notin E(G)$.

V teoriji komunikacij grafe uporabljamo za modeliranje in ponazarjanje omrežij. (Slika 2.1-3b) predstavlja modelirano ATM omrežje (Slika 2.1-3a).

2.1.1 Podgrafi in vpeti grafi

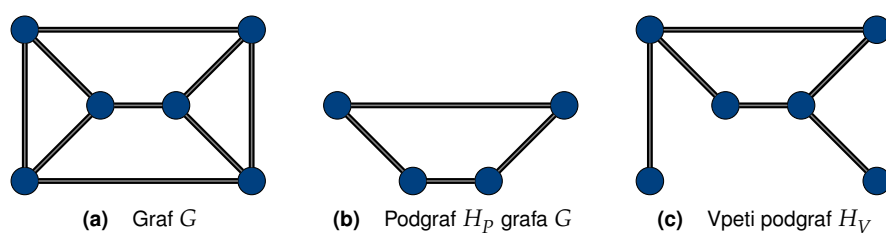
Graf H_p (Slika 2.1-4b) je **podgraf** grafa G (Slika 2.1-4a), $H_V \subseteq G$, če velja: $V_{H_p} \subseteq V_G$ in $E_{H_p} \subseteq E_G$.

Podgraf H_V (Slika 2.1-4c), $H_V \subseteq G$ je **vpeti** podgraf grafa G , če velja, da v je vsako vozlišče iz G tudi v H_V ($V_{H_V} = V_G$).



Slika 2.1-3

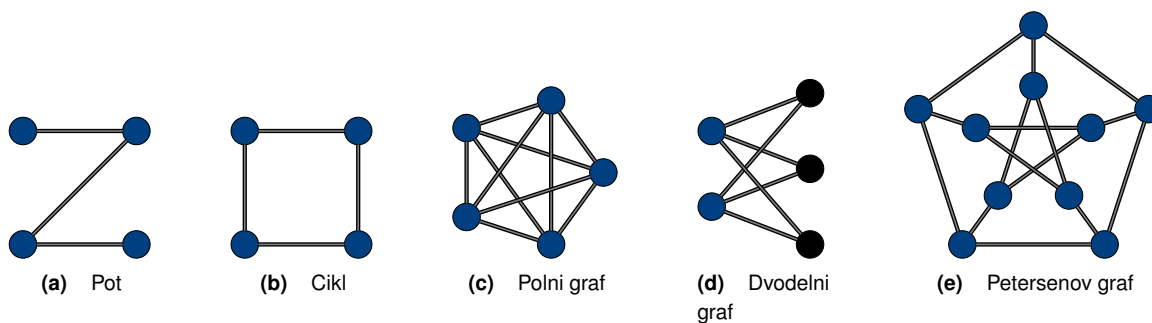
Omrežje in model modeliran z grafom.



Slika 2.1-4

Graf G , podgraf H_P in vpeti podgraf H_V .

Določene grafe štejemo med posebne vrste grafov saj zanje veljajo določene zakonitosti.



Slika 2.1-5

Posebne vrste grafov.

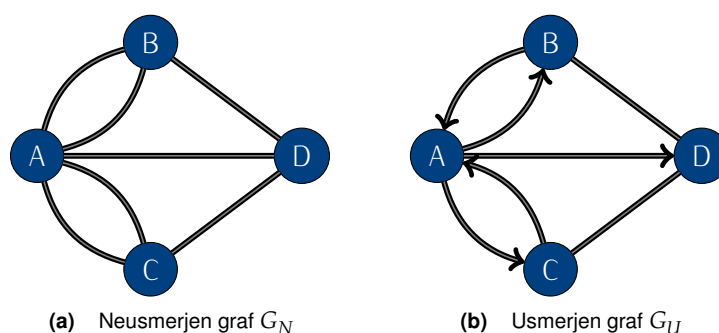
- **Cikli** (Slika 2.1-5b) so regularni povezani grafi z valenco 2 in n vozlišči, ponavadi ga označujemo s C_n .
- **Poti** (Slika 2.1-5a) so grafi, ki nastanejo z odstranitvijo ene poti iz cikla. Zanje velja, da je valenca vozlišča med 1 in 2.
- **Polni grafi**, (Slika 2.1-5c) so grafi kjer je vsak par vozlišč povezan s (točno eno) povezavo. Valenca vozlišč znaša $n - 1$, kjer n predstavlja število povezav. Označujemo jih kot K_n .
- **Dvodelni grafi**, (Slika 2.1-5d) so grafi pri katerih lahko množico vozlišč razbijemo na podmnožico A in B (kjer velja $V = A \cup B$ in $A \cap B = \emptyset$),

tako da vsaka povezava povezuje po eno vozlišče iz A in eno vozlišče iz B . Sestavljeni so iz $n = |A| + |B|$ vozlišč, označimo jih kot $K_{|A|,|B|}$.

- *Petersenov graf*, (Slika 2.1-5e) je neusmerjen graf z 10 vozlišči in 15 povezavami s kromatičnim številom 3. Petersenov graf se pogosto uporablja za potrjevanje ali zavračanje domnev.

2.1.2 Usmerjeni in neusmerjeni grafi

Povezave na grafih so lahko usmerjene ali neusmerjene. Od tod tudi delitev grafov na **usmerjene** in **neusmerjene** [37]. Usmerjen graf je urejen par množic $G = (V(G), E(G))$, kjer je V poljubna množica vozlišč, E pa množica urejenih parov vozlišč, ki jih imenujemo usmerjene povezave grafa G . Če v E ni urejenih parov oblike (v, v) dobimo **enostaven usmerjen graf**.



Slika 2.1-6

Primer usmerjenega G_U in neusmerjenega G_N grafa.

Na primerih grafov (Slika 2.1-6) pogledimo lastnosti usmerjenega in neusmerjenega grafa:

Red in velikost obeh grafov je tu še vedno enaka:

$$v(G_U) = \{A, B, C, D\} = 4 \quad (2.8)$$

$$v(G_N) = \{A, B, C, D\} = 4 \quad (2.9)$$

$$\epsilon(G_U) = 7 \quad (2.10)$$

$$\epsilon(G_N) = 7 \quad (2.11)$$

Pa tudi valenca, oziroma stopnja vozlišč grafa G_U je enaka grafu G_N . Pri usmerjenih grafih definiramo lahko še dodatna tipa stopnje vozlišča: vhodna stopnja in izhodna stopnja [37]. Stopnja vozlišča A neusmerjenega grafa G_N znaša:

$$d_A(G_N) = 5 \quad (2.12)$$

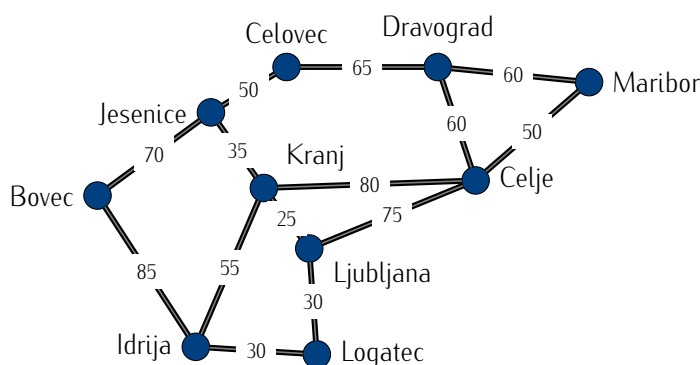
Stopnje vozlišč usmerjenega grafa G_U pa lahko zapišemo kot skupno valenco (enačba 2.13), vhodno valenco (enačba 2.14) ter izhodno valenco (enačba 2.15).

$$d_A(G_U) = 5 \quad (2.13)$$

$$d_{V_A}(G_U) = 2 \quad (2.14)$$

$$d_{I_A}(G_U) = 3 \quad (2.15)$$

Grafom, ki imajo utežene povezave pravimo **uteženi grafi**. Formalni definiciji grafa dodamo utež in dobimo: $G = (V, E, W)$, kjer W predstavlja utežno funkcijo povezav [49] $W : E \rightarrow \mathbb{R}_0^+$, $e \mapsto W(e)$.



Slika 2.1-7

Utežen graf: Oddaljenost večjih Slovenskih mest.

Dogovorimo se, za oznako $w_{(A,B)} = w(A, B)$. (Slika 2.1-7) prikazuje model zemljevida Slovenije z oddaljenostmi med kraji, ki predstavljajo utež: $w_{(Maribor, Celje)} = 50$, $w_{(Maribor, Dravograd)} = 60$, $w_{(Idrija, Logatec)} = 30$... Uteži na grafu uporabljamo za opisovanje relacije med dvema vozliščema. V primeru (Slika 2.1-7) je ta relacija razdalja. Omenili smo že, da aplikacija navigacije za svoje delovanje in izračune uporablja teorijo grafov. Predstavljajmo si, da potujemo iz Maribora v Idrijo. Na izbiro imamo veliko različnih poti ($Maribor \rightarrow Celje \rightarrow Ljubljana \rightarrow Logatec \rightarrow Idrija$ ali $Maribor \rightarrow Celje \rightarrow Kranj \rightarrow Idrija$...) izberemo pa tisto, ki ustreza določenim kriterijem (najkrajša pot, najhitrejša pot ...). Uteži na grafih nam pomagajo ovrednotiti poti glede na kriterije (Algoritme iskanja najkrajše poti bomo predstavili v poglavju 2.6).

Posebna tipa grafov predstavljajo tudi Eulerjevi in Hamiltonovi grafi. Pri teh nas zanima Eulerjev obhod ali pa Hamiltonov cikel. **Eulerjev obhod** je enostaven obhod, na katerem so vse povezave grafa in vsaka povezava natanko enkrat. **Hamiltonov cikel** pa je v grafu cikel, na katerem so vsa vozlišča grafa.

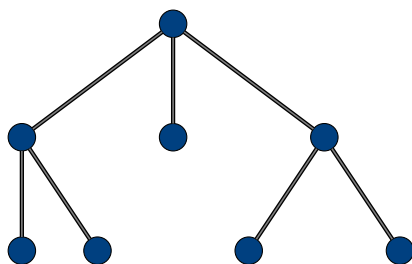
Hamiltonov cikel je mogoče uporabiti za komunikacijske topologije, kjer uporabljamo obroč. Če v komunikacijskem sistemu med postajami obstaja Hamiltonov cikel je mogoče po tem ciklu pošiljati žeton.

2.2 Drevesa

Drevo definiramo kot graf $T = (V, E)$. Graf T (Slika 2.2-1) je drevo, če je T povezan graf brez ciklov. Za drevo T veljajo trditve [118]:

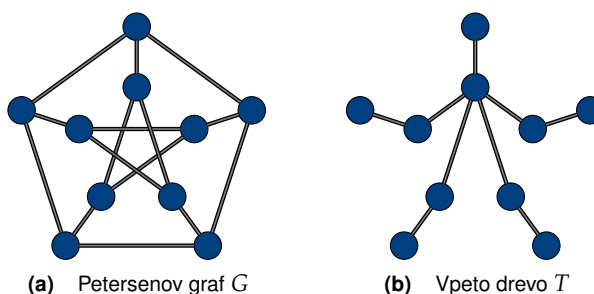
- Graf T je povezan graf in zanj velja $E = V - 1$.
- Če iz grafa T odstranimo katero koli povezavo postane graf T nepovezan.

- Vsak par vozlišč povezuje točno ena povezava.



Slika 2.2-1
Primer drevesa T .

V poglavju (Poglavje 2.1.1) smo definirali pojem vpet graf grafa G . Drevo T je vpeto drevo grafa G , če graf T zadošča pogojem drevesa [37].



Slika 2.2-2
Petersenov graf G in eno od njegovih možnih vpetih dreves T .

Drevesa (ali gozdove [112]) pogosto uporabljamo v komunikacijah. Tule je pri uteženih grafih pogosto zanimiv problem *minimalnega vpetega drevesa*. Vpeto drevo izračunamo na uteženem grafu G : Poiskati moramo podgraf T , vpeto drevo z najmanjšo skupno utežjo v povezanem grafu G .

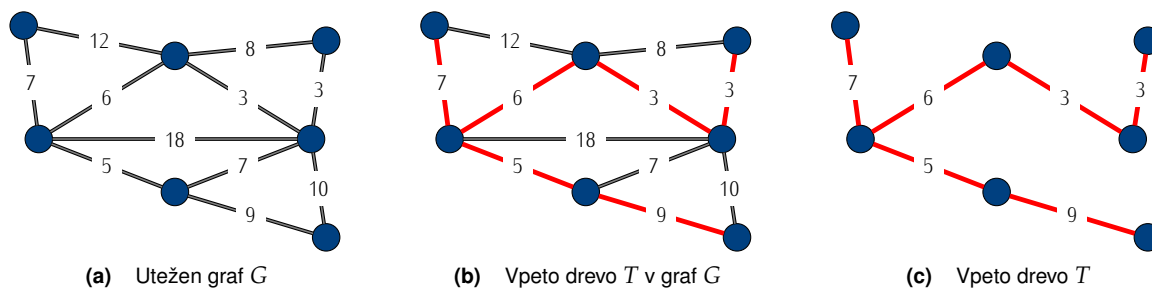
V namen računanja vpetega drevesa lahko uporabimo enega od dveh znanih algoritmov [37][112][118], ki delujeta po načelu "požrešne metode":

1. **Kruskalov algoritem**, ki zahteva računsko zahtevnost $O(|E| \log |V|)$.
2. **Primov algoritem**, ki zahteva računsko zahtevnost $O(|V|^2)$, pri izračunu na matriki in $O(|E| \log |V|)$ pri izračunu na binarni množici.

2.2.1 Kruskalov in Primov algoritem

Kruskalov algoritem deluje na principu sestavljanja minimalnega vpetega drevesa. Minimalno vpeto drevo T sestavimo tako, da po vrsti (naraščajoče uteži) izbiramo povezave in jih dodajamo v drevo, če z povezavo ne dobimo cikla. Povezave je potrebno predhodno razvrstiti po utežeh.

Primov algoritem (Algoritem 1 lahko deluje neposredno na tabeli – matriki (Poglavje 2.1) uteži in za izračun ne potrebuje strukture grafa [99].



Slika 2.2-3

Utežen graf G in njegovo vpeto drevo T .**Algoritem 1:** Primov algoritem**Data:** Graf G **Result:** T : Minimalno vpeto drevo

1 Začetni pogoji:

foreach $v \in V$ **do**2 | $D_m(v) = \infty$ 3 **end**4 $D_m(v_{zacetni}) = 0$ 5 $T = \emptyset$ 6 PQ ; množica $(D, V, \{\})$ 7 **while** $PQ \neq \emptyset$ **do**

/* Izberemo vozlišče z najmanjšo utežjo */

8 $(u, e) \leftarrow PQ.IzberiMin();$ 9 $T.Dodaj(u, e);$ 10 **foreach** $v \in Sosed(u)$ **do**11 | **if** $v \in PQ \ \&\& \ w(u, v) < D[v]$ **then**12 | | $D[v] \leftarrow w(u, v)$

/* Če je utež trenutnega vozlišča manjša od

prej poznane uteži jo zamenjamo */

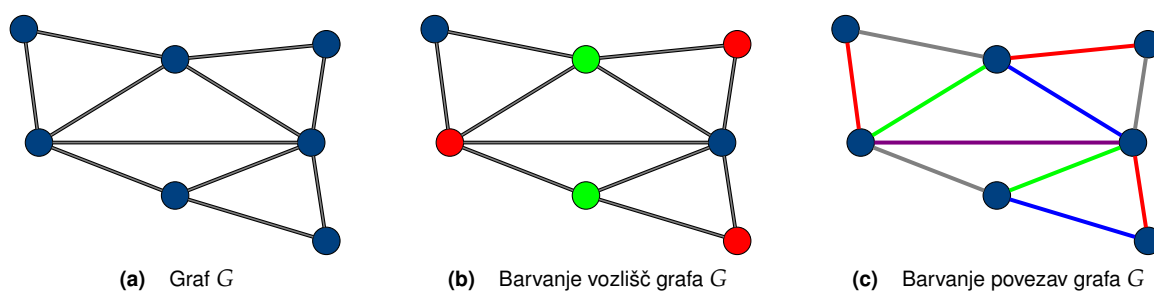
13 | | $PQ.ZnanjsajPrioritetnoVrednost(D[v], v, (u, v))$ 14 | **end**15 **end**16 **return** T 17 **end**

2.3 Barvanje grafov

Barvanje grafov je posebna vrsta označevanja grafov pri kateri dodeljujejo "barve" vozliščem ali povezavam. Od tod tudi dvoje vrst barvanja: **barvanje vozlišč** in **barvanje povezav**.

Dobro k -barvanje grafa G pomeni, dodelitev k barv vozliščem tako, da imajo vsa sosednja vozlišča različne barve. Najmanjše število za katero je graf G pobarljiv imenujemo *kromatično število*, ki ga označimo kot $\chi(G)$ [37].

Povezave v grafu so dobro pobarvane takrat, ko vsaki povezavi s skupnim krajiščem dodelimo različno barvo. Kromatični indeks ($\chi'(G)$) predstavlja najmanjše število barv k , za katero je graf G k pobarljiv [37].



$\chi(G)$:
kromatično število

$\chi'(G)$:
kromatični indeks

Slika 2.3-1

Graf G in njegovo barvanje.

(Slika 2.3-1) predstavlja dobro barvanje vozlišč (Slika 2.3-1b) in dobro barvanje povezav (Slika 2.3-1c). Kromatično število grafa G je torej $\chi(G) = 3$, saj lahko vozlišča pobarvamo s tremi različnimi barvami. Za dobro barvanje povezav pa potrebujemo vsaj 5 barv $\chi'(G) = 5$.

2.4 Algoritmi

Algoritem lahko definiramo kot zaporedje ukazov, ki so enolično razumljivi izvajalcu algoritma. Ponavadi je zahtevano še, da je algoritem končen postopek, torej, da se zaključi v končno mnogo korakov [37].

Algoritme uporabljamo za reševanje problemov. Probleme pa lahko v splošnem delimo na odločitvene ali optimizacijske.

Odločitveni problemi so določeni z množico nalog, rezultat pa je vedno enoličen odgovor: DA ali NE. Primer: *ali je število 3 večje od števila 12?* in odgovor: *NE*.

Optimizacijski problem je ponavadi določen z množico nalog in namensko funkcijo. Navadno je potrebno poiskati minimum ali maksimum namenske funkcije.

Časovna zahtevnost nam pove koliko časa se bo algoritem izvajal preden bo vrnil rezultat. Ker je natančno število operacij ponavadi precej težko ali nemogoče določiti uporabimo "**notacijo velikega O-ja**", ki označuje red rasti problema. Tabela 2.4-1 prikazuje računsko zahtevnost algoritmov in notacijo velikega O . Predpostavimo, da je velikost vhoda označena z n , k pa je neka konstanta [115].

Algoritme, ki potrebujejo v najslabšem primeru $O(n^k)$, $k > 1$ imenujemo polinomski. Pravimo, da so rešljivi v **polinomskem času**. Algoritmi, ki zahtevajo za svojo izvajanje $O(k^n)$, $k > 1$ operacij so t.i. algoritmi s **eksponentno časovno zahtev-**

Tabela 2.4-1Časovna zahtevnost in notacija velikega O povzeto po [115].

Zahtevnost	Notacija	Primer
Konstantna	$O(1)$	Ugotavljanje, ali število pripada množici sodih ali lihih števil.
Logaritemska	$O(\log n)$	Iskanje števila v urejenem polju.
Linearna	$O(n)$	Iskanje števila v ne-urejenem polju.
Loglinearna	$O(n \log n)$	Izvajanje FFT (Fast Fourier transform).
Kvadratna	$O(n^2)$	Različni algoritmi sortiranja (bubble sort, quicksort, shell sort ...)
Polinomska	$O(n^k), k > 1$	Maksimalno prirejanje v dvodelnih grafih.
EkspONENTNA	$O(k^n), k > 1$	Iskanje rešitve trgovskega potnika.

nostjo. Polinomske algoritmi so imenovani tudi učinkoviti algoritmi, nepolinomske algoritme pa lahko imenujemo tudi neučinkoviti algoritmi [37].

2.4.1 Razredi problemov

V teoriji algoritmov definiramo razrede problemov glede na to, kako hitro jih rešimo. Problem je v razredu **P**, če ga lahko rešimo z polinomske algoritmom. V **NP** razredu so problemi, za katere potrebujemo algoritme tipa $O(k^n)$, $k > 1$. V razredu **NP** najdemo tudi podmnožico problemov, ki jih imenujemo **NP-polni problemi**.

Za katerikoli NP-polni problem velja: *Če bi obstajal polinomske algoritem za rešitev NP-polnega problema potem bi lahko v polinomske času rešili vse probleme iz NP razreda.* Obveljalo bi torej, da: $P = NP$.

NP težke probleme rešujemo, ali bolje rečeno: poskušamo reševati (saj iščemo dovolj dobre približne rešitve) z verjetnostnimi algoritmi (algoritem simulacijskega ohlajanja [26]), genetskimi algoritmi [88] ...

2.5 Pregledovanje grafov

Strukture grafov se pogosto uporabljajo v komunikacijah in računalniški tehniki. Pregledovanje po grafih nam omogoča iskanje določenih informacij, ki se nahajajo v strukturi.

BFS: Breadth-first search

DFS: Depth-first search

V osnovi ločimo dva tipa iskanja: **iskanje v širino** - **BFS** (ang.:breadth-first search) in **iskanje v globino** - **DFS** (ang.:Depth-first search) .

Pri pregledovanju v globino poskušamo prodreti čim dlje v globino grafa preden razvejimo pot pregledovanja.

Pri pregledovanju v širino se držimo obratnega principa: obiskati poskušamo čim več vozlišč istega nivoja preden se spustimo globlje.

Ker bomo v nadaljevanju potrebovali BFS, pogledajmo, kako deluje algoritem [30]. Za implementacijo bomo uporabili podatkovno strukturo imenovano vrsta (ang.:Queue) [117]:

Algoritem 2: BFS iskanje

Data: Povezan Graf G
Result: BFS

- 1 *Začetni pogoji:*
- 2 **foreach** $v \in V$ **do**
- 2 $v.BFS = \infty$
- 3 $v_{zacetni}.BFS = 0$
- 4 Q ; Vrsta elementov tipa v
- 5 $Q \leftarrow v_{zacetni}$
- 6 **while** $Q \geq \emptyset$ **do**
- 7 $v_T = Q.izVrste()$;
- 8 **foreach** $v \in Sosed(v_T)$ **do**
- 9 **if** $v.BFS == \infty$ **then**
- 10 $v.BFS = v_T.BFS + 1$;
- 11 $Q.vVrsto(v)$;
- 12 **return** G ;

2.6 Najkrajše poti

Eno zanimivejših vprašanj v teoriji grafov predstavlja tudi najkrajša pot med dvema vozliščema. Za iskanje najkrajših poti v grafu obstaja kar nekaj znanih algoritmov, najbolj znan pa je gotovo **Dijkstrov** algoritem [112]. Z njim izračunamo razdalje (in najkrajše poti) od izbranega vozlišča do vseh drugih vozlišč omrežja. Graf mora biti utežen s poljubnimi nenegativnimi utežmi na povezavah [37].

Algoritem 3: Dijkstrov algoritem

Data: Povezan Graf G
Result: Razdalje na grafu

- 1 *Začetni pogoji:*
- 2 **foreach** $v \in V$ **do**
- 2 $v.razdalja = \infty$;
- 3 $v.stars = null$;
- 4 Q ; Vrsta elementov tipa v
- 5 $Q \leftarrow V(G)$;
- 6 **while** $Q \geq \emptyset$ **do**
- 7 $v_T = Q.izVrste(MIN(v.dist))$;
- 8 **foreach** $v \in Sosed(v_T)$ **do**
- 9 $C_{razdalja} = v_T.razdalja + Razdalja(v_T, v)$;
- 9 */* Če je razdalja po omenjeni poti krajša jo prepisemo */*
- 10 **if** $C_{razdalja} < v.razdalja$ **then**
- 11 $v.razdalja = C_{razdalja}$;
- 12 $v.stars = v_T$;
- 13 **return** G ;

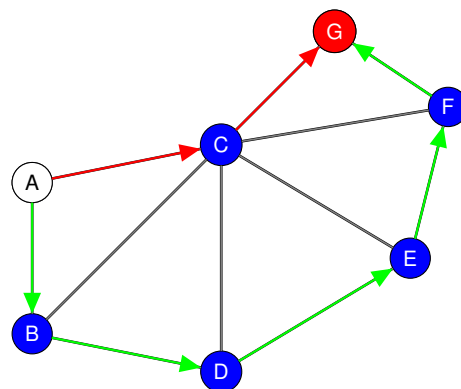
Opazimo lahko, da je struktura *node* dobila dve lastnosti in sicer *node.dist* uporabimo za zapis razdalje in *node.preNode* za določitev starša (tako lahko po verigi staršev sledimo, kje po grafu najkrajša pot poteka). Lastnost *node.preNode* je seveda opsijska in ni nujna za izračun Dijkstrovega algoritma vendar pa je, kot bomo ugotovili naprej, večkrat zelo koristna.

Usmerjanje prometa

Podatki postanejo uporabni šele, ko so na voljo uporabniku. V brezžičnem senzorskem omrežju podatki navadno opravijo več etap preden dosežejo cilj (naslovnika). V ta namen uporabljamo **protokole usmerjanja prometa**. Protokoli usmerjanja prometa za svoje delovanje uporabljajo *algoritme usmerjanja prometa*.

Algoritem, kot smo že dejali (Poglavje 2.4) sestavlja določeno število pravil, ki rešuje določen problem, kot recimo izračun najkrajše ali najmanj energijsko potratne poti.

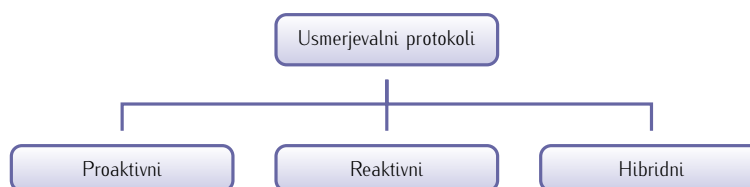
Predstavljajmo si preprosto brezžično omrežje, ki ga ponazorimo z grafom (Slika 3.0-1).



Slika 3.0-1
Schema preprostega brezžičnega senzorskega omrežja.

Na (Slika 3.0-1) smo označili dve od možnih poti, kateri lahko uporabimo za pošiljanje sporočila od točke A do točke G. V prvem primeru (izbira poti rdeče barve) sporočilo za dosego cilja potrebuje dve etapi, v drugem primeru (izbira poti zelene barve) pa pet etap. Na prvi pogled se seveda zdi, da je izbira poti trivialna, vendar pa se izkaže, da je izbira komunikacijske poti v omrežju zahteven postopek, odvisen od porabljene energije, kvalitete povezave, obremenitve povezave, topologije in še mnogih drugih podobnih faktorjev. **Naloga algoritmov usmerjanja je torej najti optimalno pot po omrežju glede na dane parametre.**

Usmerjanje prometa v brezžičnih večetapnih sistemih lahko v grobem razdelimo v tri skupine (Slika 3.0-3) [53].

**Slika 3.0-2**

Delitev protokolov glede na predhodno poznavanje poti.

Pro-aktivni usmerjevalni protokoli so protokoli, kjer imajo načeloma vsa senzorska vozlišča vedenja o vseh drugih vozliščih. Informacije o celotnem omrežju vsako vozlišče shranjuje v posebni tabeli. Tabele vozlišče obnavlja periodično, ko se spremeni topologija omrežja. Usmerjevalne poti so od vozlišča do vseh drugih znana v naprej.

Re-aktivni protokoli so protokoli, kjer ima senzorsko vozlišče vedenje le o svojih sosedih in potrebnih aktivnih poteh. Poti so shranjene tako dolgo, dokler so potrebne. Seveda pa mora vozlišče pred vsakim pošiljanjem podatkov poslati zahtevo po iskanju poti.

Hibridni usmerjevalni protokoli pa so kombinacija prej omenjenih protokolov.

V literaturi lahko zasledimo veliko predlogov algoritmov usmerjanja prometa za brezžična senzorska omrežja, raznolikost pa se pojavlja predvsem zaradi posebnosti brezžičnih senzorskih. Zaradi velikega števila senzorskih vozlišč je skoraj nemogoče uporabiti globalno shemo naslavljanja vozlišč: potrebna je namreč velika količina dodatnega prometa (overhead) za vzdrževanje ID-jev. Za razliko od drugih podobnih omrežij (MANET) je v BSO večina podatkovnega toka usmerjena proti točno določenemu vozlišču: Bazni Postaji (BP). Drugi tip prometnega toka (multicast ali točka-točka) sicer ni izključen je pa zelo redek. Brezžična senzorska vozlišča so med drugim precej omejena s procesorsko močjo, spominskimi viri in navadno baterijsko napajana. Posledično se pojavlja potreba po energijsko varčnih in preprostih a vseeno učinkovitih protokolih usmerjanja prometa.

Senzorska vozlišča so v večini primerov, ko jih položimo stacionarna, z izjemo mogoče par vozlišč. Vozlišča v drugih podobnih omrežjih se načeloma prosto premikajo po prostoru.

Brezžična senzorska omrežja so, kot smo že dejali, zelo aplikacijsko odvisna. To pomeni, da je načrtovanje omrežja, strojne opreme in protokolov v veliki meri odvisen od namena uporabe BSO. Nizko zakasnitvena natančna taktično nadzorna BSO zagotovo potrebujejo drugačen tip protokolov za usmerjanje prometa kot aplikacija v kateri periodično nadzorujemo vodostaj v velikih rezervoarjih.

Pri usmerjanju prometa v BSO v določenih aplikacijah veliko vlogo igra lokacija. V primeru intenzivnega nazorovanja samo določenega dela pokritega območja je potrebno najprej določiti lego oziroma pozicijo senzorjev. Navadno le nekaj senzorjev pozna svojo absolutno lokacijo (uporaba GPS modulov), saj implementacija GPS modulov na vsako vozlišče močno podraži senzorsko vozlišče. Večina senzorskih enot torej izračunava svojo lokacijo [6, 20, 87].

GPS: Global Positioning System

V BSO je precej verjetno, da bo večina podatkov, ki prihajajo iz istega področja redundantnih (velika verjetnost je, da bodo na primer vsi senzori na enakem področju zaznali enako temperaturo), zato je primerno, da usmerjevalni protokoli to redundantnost izvržejo in s tem prihranijo energijo in pasovno širino za druge podatke.

Iz opisanega lahko ugotovimo, da je iskanje in vzdrževanje poti po brezžičnih senzorskih omrežjih vse prej kot trivialno, še posebej zaradi konstantnega spreminjanja količine prenosa podatkov in spreminjanja topologije.

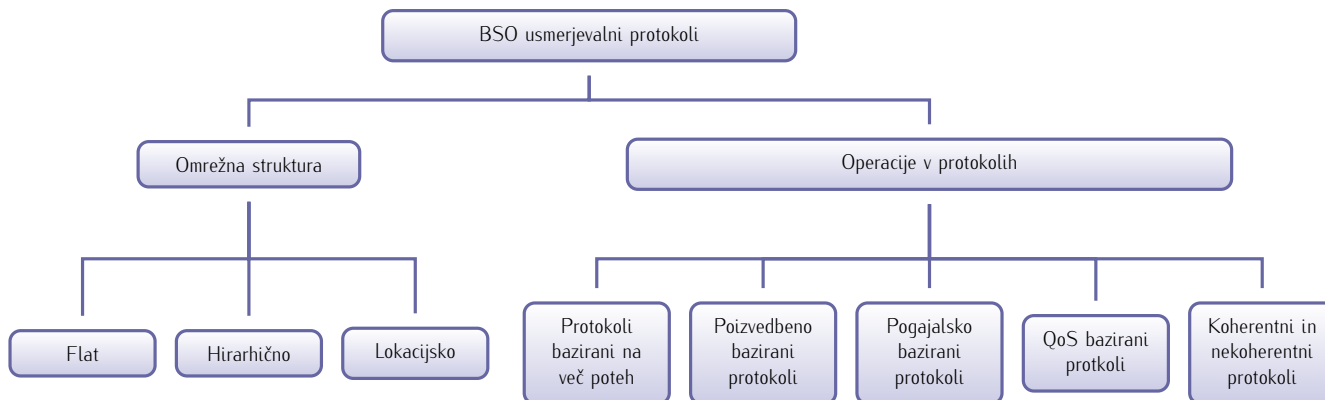
Raziskovalci, ki razvijajo nove algoritme usmerjanja prometa se morajo zato soočiti z naslednjimi izzivi [10]:

- **Nameščanje senzorske enote**, je seveda aplikacijsko odvisno in vpliva na algoritem usmerjanja. Lahko je deterministično ali naključno. Senzorske enote morajo posedovati sposobnost samo-organizacije (Ad-hoc način). Smiselno je sklepati, da bo večina prometa potrebovala za doseganje bazne postaje več etap.
- **Model podatkovnega prenosa**, je lahko časovno voden (time driven), dogodkovno voden (event driven), voden z vprašanji (query) ali hibridno. Izvedba ali izbira modela podatkovnega prenosa ima precejšen vpliv na učinkovitost usmerjevalnega algoritma.
- **Raznolikost senzorskih enot in povezav** je v večini primerov zanemarjena. Večina snovalcev usmerjevalnih protokolov predpostavlja, da imajo vse podatkovne povezave enako kakovost in vse senzorske enote na voljo približno enako računsko moč in vire. Poleg raznolike kakovosti povezave je lahko različna tudi računsko moč senzorskih enot (kar zasledimo redkeje). Usmerjevalni protokol mora torej upoštevati raznolikost kakovosti povezav in računsko moč procesorjev na senzorskih vozliščih.
- **Odpornost na napake (fault-tolerance)** senzorske enote lahko odpovedo zaradi različnih vzrokov: pomanjkanje energije, fizična poškodba, motnje na kanalu, motnje iz okolice... Seveda pa neaktivne senzorske enote ne smejo vplivati na delovanje sistema. Tako je naloga MAC protokola, topologije in usmerjevalnih protokolov, da ustvarijo nove povezave, ki omogočajo povezanost senzorskega vozlišča v omrežje (seveda, če je to fizično mogoče). Ustvarjanje novih povezav morda zahteva dinamično spreminjanje oddajne moči, preusmerjanje paketov in podobno. Posledično je potrebno več redundantnih postopkov za ohranjanje odpornosti na napake.
- **Skalabilnost**: V BSO je lahko število senzorskih enot reda sto, tisoč ali še več. Vsak usmerjevalni algoritem bi moral biti sposoben delovati s tako velikim številom senzorskih enot (število enot je odvisno od aplikacije).
- **Povezanost** je odvisna od gostote senzorskih enot na enoto površine. V BSO imamo ponavadi veliko gostoto senzorskih enot na enoto površine in posledično ima vsaka enota veliko število sosedov. Seveda pa ta povezanost s časom izginja (senzorske enote se energijsko izrabijo) zato je potrebno s topologijo vse skozi skrbeti, da ostane povezanost na želenem nivoju.
- **Zlivanje ali združevanje podatkov** je zelo pomembno saj lahko precej razbremeni podatkovne povezave: podatke, ki prihajajo od različnih senzorskih enot lahko združimo. Združevanje podatkov zlije podatke iz različnih virov glede na določeno funkcijo zlitja (maksimum, minimum ali povprečje). V namen zlivanja podatkov večkrat uporabimo metode iz obdelave signalov.
- **Kakovost storitve (QoS)** je v nekaterih aplikacijah zaželena. Govorimo predvsem o aplikacijah, kjer mora biti informacija dostavljena pravočasno ali pa

postane neuporabna ali tam, kjer je pomembnost informacij različna. V nekaterih aplikacijah je torej energija drugotnega pomena. Pomembnejša je zakasnitev, ter hitrost dostave informacije.

Skoraj vse usmerjevalne protokole lahko razdelimo, glede na omrežno strukturo kot enakovredne (flat), hirarhične (hierarchical) in lokacijsko bazirane (location-based). V flat usmerjevalnih protokolih so vsa senzorska vozlišča enakovredna, v hirarhičnih algoritmih usmerjanja pa so nekatere enote višjega ranga kot druge (na primer v gruči, kjer je eno izmed senzorskih vozlišč vedno vodja gruče).

QoS: Quality of Service



Slika 3.0-3

Delitev protokolov glede na omrežno strukturo in operacije.

3.1 Topologija

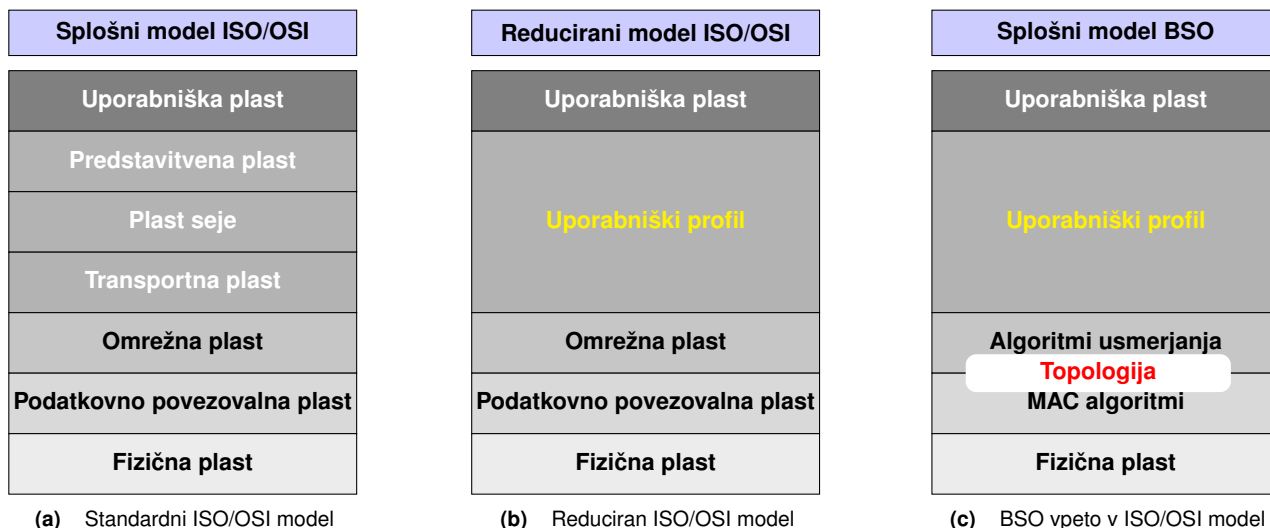
Med dvema napravama, ki komunicirata druga z drugo obstaja povezava, ki je lahko realna (optično vlakno, kabel) ali virtualna (radijska vidljivost med dvema brezžičnima enotama). Množico povezav (tako realnih kot virtualnih) pa imenujemo **topologija**.

V brezžičnih senzorskih omrežjih pa je poleg poznavanje ali uporabe topologije pomemben tudi **nadzor topologije**. Nadzor topologije je v BSO praktično mogoč z manipulacijo radijskega modula senzorske enote (spreminjanje frekvence sprejemnega in oddajnega kanala, spreminjanjem oddajne moči...).

Topologijo lahko definiramo kot **množico komunikacijskih povezav, ki jih algoritmi usmerjanja uporabljajo implicitno in eksplicitno [104]**. Kontrolo topologije najbolje definira [100], ki pravi: **kontrola topologije je umetnost usklajevanja (kooordiniranja) lastnosti vozlišč v nalogi ustvariti ali ohraniti omrežje želenih lastnosti (recimo povezanosti)**. Kontrola topologije in povezanost neposredno vpliva tudi na energijsko porabo senzorske enote, kapaciteto omrežja in drugo...

Hirarhično gledano fizična plast vpliva na topologijo, topologija pa vpliva na protokole usmerjanja prometa.

Kljub zavedanju pomembnosti uporabe nadzora topologije pa še vedno ni čisto točno definirano kam v ISO/OSI model topologija sploh spada [100]. Najbolj smiselno jo je uvrstiti med MAC protokole in protokole usmerjanja, kot prikazuje (Slika 3.1-1c).

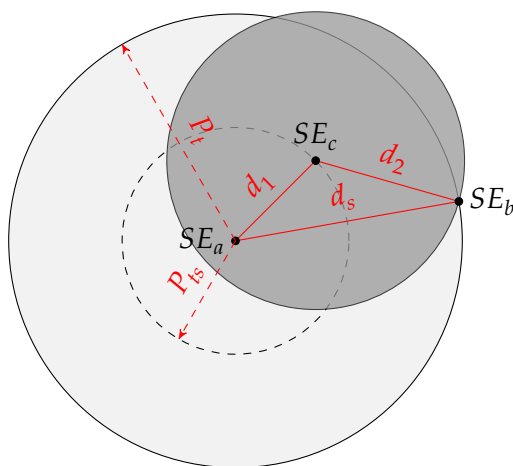


Slika 3.1-1

Umestitev topologije v ISO/OSI model.

3.1.1 Primer uporabe kontrole topologije

Kot primer uporabnosti kontrole topologije navedimo naslednji primer: vzemimo tri senzorske enote SE_a , SE_b in SE_c . Predpostavimo, da je v začetnem stadiju oddajna moč nastavljena na maksimum.



Slika 3.1-2

Primer kontrole topologije.

Iz slike (Slika 3.1-2) je razvidno, da lahko enota SE_a z enoto SE_b komunicira na dva načina:

1. Direktno $SE_a \mapsto SE_b$, kjer predpostavimo, da je energijska poraba enaka d_s^2 .
2. Preko dveh etap $SE_a \mapsto SE_c \mapsto SE_b$, kjer pa energijska poraba znaša $d_1^2 + d_2^2$.

Za lažjo razlago primera predpostavimo, da uporabimo model razširjanja elektromagnetnega valovanja po praznem prostoru (Poglavje 3.1.2), zanima pa nas samo

energijska učinkovitost. Trikotnik $SE_a - SE_c - SE_b$ pogljemo iz geometričnega stališča. Ugotovimo lahko, da velja:

$$d_s^2 = d_1^2 + d_2^2 - 2 \cdot d_1 \cdot d_2 \cdot \cos\alpha \quad (3.1)$$

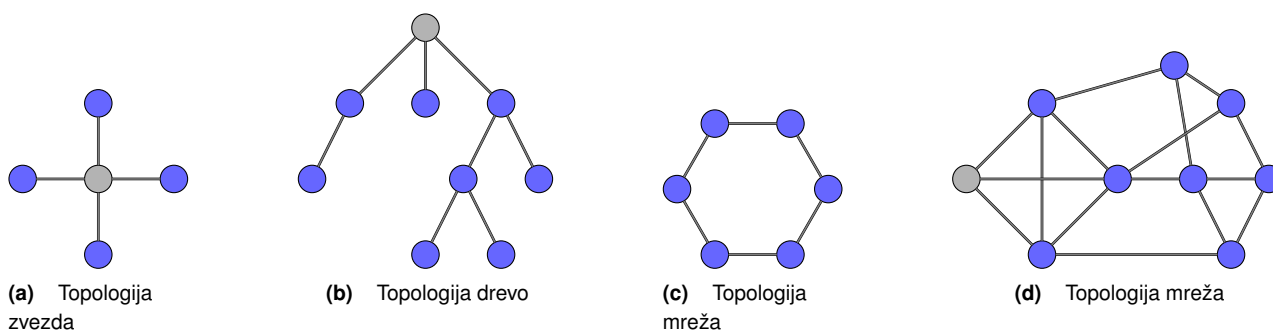
Iz (Slika 3.1) je razvidno, da je vrednost $\cos\alpha \leq 0$ (negativnega predznaka). Iz tega sledi, da velja

$$d_s^2 \geq d_1^2 + d_2^2 \quad (3.2)$$

Iz enačbe 3.2 lahko razberemo, da je iz energijskega stališča bolje uporabiti več krajših etap za komunikacijo med enotami [18].

Kljub različnim delitvam in tipom topologije je v splošnem, glede na povezanost mogoče ločiti štiri tipe topologije:

1. zvezdno topologijo (Slika 3.1-3a), ko je vsaka enota s točno eno povezavo povezana na skupno vozlišče (enoto), število vozlišč = število enot - 1.
2. drevesna topologija (Slika 3.1-3b), ko so enote povezane v drevesno strukturo (brez ciklov).
3. topologija obroča (Slika 3.1-3c), kjer so enote povezane v cikle.
4. mrežna topologija (Slika 3.1-3d), ko so enote povezane na način, ki ne ustreza nobeni od prej naštetih topologij.



Slika 3.1-3
Različne topologije.

3.1.2 Vpliv fizične plasti na topologijo

Naprave v brezžičnih senzorskih omrežjih med seboj komunicirajo z uporabo brezžičnih sprejemno-oddajnih modulov (radio). Radio spreminja vhodne podatke v elektromagnetno valovanje, ki ga oddaja. Elektromagnetno valovanje se razširja čez eter, kjer zaradi določeni fizikalni zakonov slabi v funkciji razdalje. Kako intenzivno je slabljenje in kako se slabljenje obnaša poskušamo opisati z modeliranjem prenosnih kanalov [98].

Radijski enoti lahko komunicirata le takrat, ko je sprejemna moč na sprejemniku višja od določene pragovne vrednosti (sensitivity threshold), sprejeta moč signala

(P_s) na radijski enoti pa je odvisna od oddajne moči (P_o) in izgube na poti S (enačba 3.5).

$$P_s = \frac{P_o}{S_{izgub}(SE_1, SE_2)} \quad (3.3)$$

Modeliranje izgub na poti elektromagnetnega valovanja je gotovo še danes eden največjih izzivov razvoja brezžičnih komunikacij. Mehanizmi, ki v večini vplivajo na izgube pri razširjanju signala po prostoru lahko uvrstimo v tri kategorije: *odboj* (eng.: *reflection*), *lomljenje* (eng.: *diffraction*) in *razsipanje* (eng.: *scattering*) [100], [86].

Hitrost razširjanja elektromagnetnega valovanja lahko matematično opišemo z enačbo:

$$c = f \cdot \lambda \quad (3.4)$$

kjer uporabimo c za hitrost valovanja ($3 \cdot 10^8$ m/s), f za frekvenco valovanja izraženo v *Hz* in λ za valovno dolžino v metrih. V splošnem velja predpostavka, da moč razširjanja signala upada s kvadratom razdalje. V praksi pa je stvar seveda kompleksnejša [61].

Modeli prenosnih kanalov

Najbolj preprost model prenosnega kanala je zagotovo **model razširjanja po praznem prostoru**. Prazen prostor bi lahko definirali kot širok, odprt prostor brez kakršnih koli ovir, od katerih bi se valovanje lahko odbijalo, se lomilo ali razprševalo. Osnovna enačba opisuje odvisnost sprejete moči od razdalje, ojačitve antene, in oddajne moči:

$$P_s = P_o \cdot \frac{G_o \cdot G_s \cdot \lambda^2}{(4 \cdot \pi)^2 \cdot d^n} \quad (3.5)$$

P_s : moč na sprejemni anteni,

P_o : moč na oddajni anteni,

G_s : ojačitev sprejemne antene,

G_o : ojačitev oddajne antene,

λ : valovna dolžina,

d : razdalja med oddajno in sprejemno anteno,

n : faktor okolja.

Faktor okolja (n) v vakuumu znaša $n = 2$, v realnem prostoru pa velja, da je $n = 2 \dots 4$.

Če poenostavimo enačbo 3.5 in predpostavimo, da znašata $G_o = 1$ in $G_s = 1$ lahko ugotovimo, da je sprejeta moč na sprejemni anteni odvisna predvsem od oddajne moči, nosilne frekvence valovanja, razdalje in seveda faktorja okolja (enačba 3.6).

$$P_s = \frac{P_o \cdot \lambda^2}{(4 \cdot \pi)^2 \cdot d^n} \quad (3.6)$$

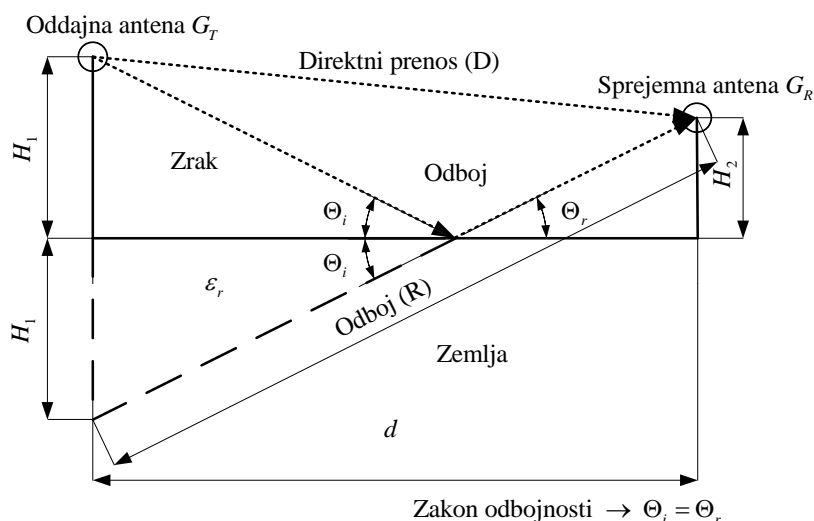
Nadaljevanje bi predstavljal **Log-normalni model kanala**, ki je izpeljanka iz modela kanala razširjanja po praznem prostoru v logaritemskem merilu (enačba 3.7)

$$\bar{P}_s = P_{sr} - 10 \cdot \log\left(\frac{d}{d_r}\right) \quad (3.7)$$

kjer \bar{P}_s predstavlja povprečno sprejeto moč na sprejemniku v dB na razdalji d . P_{sr} predstavlja referenčno sprejeto moč v dB na referenčni kratki razdalji d_r .

Ker so enote senzorskih omrežij navadno bližje tlam je pri radijski vidljivosti potrebno upoštevati tudi odboj EM valovanja od tal. Pri odboju se del energije odbije, del pa absorbira. V najslabšem možnem primeru je odbiti signal na sprejemni anteni prišesteje v proti fazi direktni signalu zato največkrat uporabimo kar preprosto izpeljavo kanala direktnega razširjanja po prostem prostoru, za faktor okolja pa uporabimo $n = 4$.

Delež absorpcije pri odbojnem modelu (Slika 3.1-4) določajo Fresnelovi koeficienti.



Slika 3.1-4
Primer odbojnega modela.

Odbojni model lahko izpeljemo iz geometrije odboja na sliki (Slika 3.1-4):

$$D = \sqrt{(H_2 - H_1)^2 + d^2} \quad (3.8)$$

$$R = \sqrt{(H_2 + H_1)^2 + d^2} \quad (3.9)$$

$$\Theta_i = \arctan\left(\frac{H_1 + H_2}{d}\right) \quad (3.10)$$

Razliko v faznem kotu Δ med direktnim in odbitim signalom lahko definiramo kot:

$$\Delta = \frac{2 \cdot \pi}{\lambda} \cdot (R - D) \cdot \text{sign}(\Gamma) \quad (3.11)$$

$$P_{s-\text{direktni}} = P_o \cdot \frac{G_o \cdot G_s \cdot \lambda^2}{(4 \cdot \pi)^2 \cdot d^n} \quad (3.12)$$

$$P_{s\text{-odbojni}} = P_o \cdot \frac{G_o \cdot G_s \cdot \lambda^2}{(4 \cdot \pi)^2 \cdot d^n} \cdot \Gamma \quad (3.13)$$

$$P_s = P_{s\text{-direktni}} + P_{s\text{-odbojni}} \cdot \cos(\Delta) \quad (3.14)$$

Detajlno izpeljavo in izračun Fresnelovi koeficientov je izpeljal avtor v [86].

Modeli kanalov nam torej omogočajo, da teoretično določimo sprejemno moč in/ali pozicijo brezžičnih senzorskih enot glede na oddajno moč z upoštevanjem določenih zelenih parametrov omrežja (povezanost, propustnost, energijska poraba...).

Model omrežja in prometna uravnoveženost

4.1 Matematični model omrežja

Za nadaljevanje pisanja doktorske teze najprej definirajmo matematični model brezžičnega senzorskega omrežja. Imenujmo ga SPaRCModel. SPaRCModel je definiran kot graf $G = (V, E)$, ali $G = (V, E, W)$, kjer V predstavlja množico senzorskih vozlišč in E množico logičnih povezav med vozlišči. V primeru uteženega grafa G pa W predstavlja množico uteži na povezavah v E . Velja torej, da je število elementov množice uteži enako številu elementov v množici povezav ($|W| = |E|$), elementi v množici pa so prirejeni elementom iz množice E ($E \rightarrow \mathbb{R}_0^+$). Primera, ko bi bila $|W| \neq |E|$ ne bomo obravnavali. Povezave v množici E bodo ne-usmerjene.

Če obstaja povezava (logični link; radijska vidljivost) med senzorskima enotama a in b potem pravimo, da sta enoti sosednji. Če je a sosednja b potem je tudi b sosednja a ; to pomeni, da so povezave neusmerjene.

Brezžično senzorsko omrežje bomo definirali kot množico avtonomnih senzorskih enot, ki zaznavajo različne fenomene in podatke o le teh pošiljajo do uporabnika. Iz trditve je mogoče razbrati, da večina poslanega prometa poteka proti bazni postaji ali postajam (razen v procesu vzpostavitve omrežja).

Kljub temu, da je v brezžičnem senzorskem omrežju lahko prisotnih več baznih postaj bomo v izračunih sklepali, da je prisotna le ena, t.i. imenovana r (*root/koren*). Primer več baznih postaj bomo prevedli v problem ene bazne postaje. To storimo tako, da po potrebi definiramo novo virtualno vozlišče r , ki povezuje vse bazne postaje (Slika 4.1-1b).

4.1.1 Znani matematični modeli omrežij

Po literaturi BSO večinoma modeliramo kot množico omrežnih naprav (senzorskih enot) lociranih na dvo-dimenzionalni ravnini. V splošnem velja predpostavka, da je oddajna moč vseh senzorskih enot enaka [18]. Komunikacijski graf $UDG(V)$ je navadno definiran kot neusmerjen graf na osnovi vozlišč v množici V . Dve enoti $a, b \in V$ sta povezani samo, če je njuna Evklidska razdalja največ 1 ($\|ab\| \leq 1$) [59]. Z naraščanjem števila in gostote senzorskih enot narašča tudi gostota grafa.



Slika 4.1-1

Primer virtualne bazne postaje r .UDG:
Unit-Disk Graph

Gostota grafa pa vpliva na sosednost vozlišč: Večja gostota pomeni tudi več sosednjih vozlišč, kar pa lahko predstavlja problem v brezžičnih senzorskih omrežjih zaradi omejene velikosti pomnilnika senzorskih enot.

S protokoli kontrole topologije spreminjamo začetno topologijo grafa. Nov, z algoritmi kontrole topologije izračunan podgraf je ponavadi uporabljen kot vpet topologija za potrebe usmerjevalnih algoritmov.

Definirajmo razdaljo med dvema vozliščema $a, b \in V(G)$ kot $dist_G(a, b)$ ($dist_G$ predstavlja število etap - povezav od a do b , na najkrajši poti od a do b). Podgraf $H = (V, E')$ grafa $G = (V, E)$, kjer velja $E' \subseteq E(G)$ je t -vpenjalec (ang.: t -spanner) grafa G , če je za vsak $a, b \in V(G)$:

$$dist_H(a, b) \leq t \cdot dist_G(a, b), \quad (4.1)$$

kjer t imenujemo *raztezni faktor* (ang.: stretch factor). Definirajmo uteži na povezavi $(a, b) \in E$ kot $||ab||^\beta$ za konstantni β (navadno $\beta = 2$). Če lahko uporabimo omenjen $||ab||^\beta$ kot zamenjavo za dolžino poti s sumo uteži na tej poti, potem lahko definiramo t -vpenjalec moči (ang.: t -power spanner) grafa G , vrednost t pa v tem primeru imenujemo *raztezni faktor moči* (ang.: power stretch factor). Če postavimo $\beta = 1$, lahko definiramo *vpenjalec dolžine* (ang.: length spanners) in *dolžinski raztezni faktor* (ang.: length stretch factors).

V želji po učinkovitih usmerjevalnih protokolih in usmerjanju prometa je pomembno, da predlagamo algoritme ki izračunavajo vpenjalce s konstantnim raztezni faktorjem. Večino zahtev modeliranja z *UDG* grafi je povzeti v [109]. Vpenjalci moči in dolžine kot **Gabriel-ovi grafi** [24], stopenjsko omejeni **Yao grafi** [79] in **Delaynayeve triangulacije** [77] so bili predstavljeni kot *BPS* strukture [110].

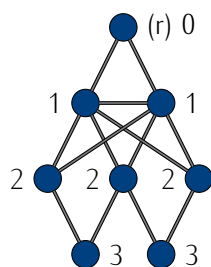
UDG model BSO je popularen in razširjen zaradi svoje preprostosti. Zaradi razmer v komunikacijskem kanalu, razmerja signal-šum (SNR) in podobnih vplivov pa v realnih BSO *UDG* model ni najbolj realističen. Bolj realistični modeli so *QUDG* model [70], SINR model [45] in MIG model [78].

QUGD: Quasi Unit Disk Graph

4.1.2 Drevesna struktura v modelu

Kot smo že dejali je promet iz senzorske enote namenjen bazni postaji (postajam). Komunikacija točka-točka je relativno redka. Ker poskušamo minimizirati časovno zakasnitev sporočil od senzorske enote do bazne postaje se zdi naravna pot optimizacije minimiziranja števila etap od izvora do ponora.

BFS drevesa (Poglavje 2.5) so vpete topologije, ki omogočajo minimalno število etap od enote do bazne postaje (Algoritem 2).



Slika 4.1-2

Rezultat BFS algoritma izvedenega na grafu G .

Na grafu G smo s pomočjo BFS iskanja, začeni v vozlišču r , definirali *nivoje na grafu*. Neposredni sosedi vozlišča r imajo nivo 1, neposredni sosedi teh sosedov nivo 2...

Drevesne strukture omogočajo relativno preproste in učinkovite sheme usmerjanja. Literatura predlaga več kompaktnih usmerjevalnih shem za drevesne topologije [108]: *imensko-neodvisno kompaktno usmerjanje* [72] in *kompaktno usmerjanje v dinamičnih drevesih* [67], ki se ukvarja z usmerjanjem v drevesnih strukturah katerih topologija se spreminja v času (dodajanje ali odvzemanje vozlišč).

4.2 Uravnoteženost prometa

Uravnotežitev ali enakomerna porazdelitev bremena (ang.: Load Balancing) je tehnika s katero enakomerno porazdelimo *delovno obremenitev* med dva ali več virov. V računalništvu izraz uravnotežitev ponavadi pomeni enakomerno razdelitev nalog med več računalnikov, enakomerno porazdelitev podatkov po pomnilniku ali **telekomunikacijskega prometa** po omrežnih povezavah [114].

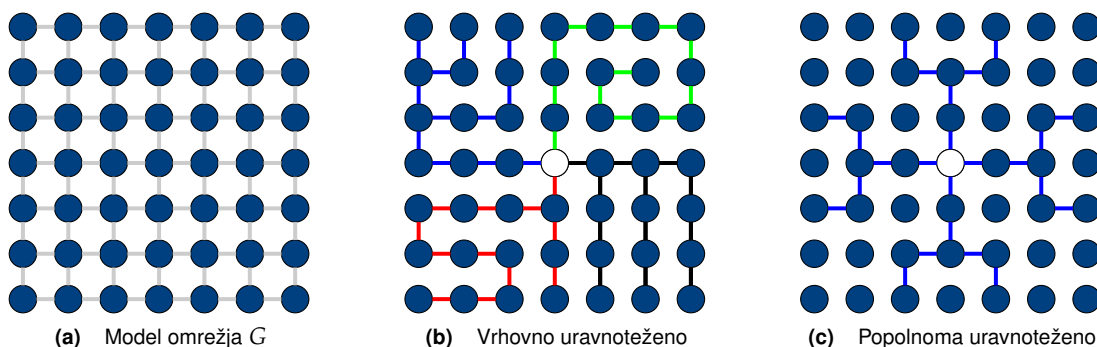
Vzemimo za primer večja podjetja, banke ali ustanove. Zaradi odvisnosti od interneta večina le teh uporablja dva ali več ponudnikov interneta (za primer izpada). V času normalnega delovanja je smiselno, da se internetni promet čimbolj enakomerno porazdeli med vse ponudnike prometa (predpostavimo, da vsi ponudniki ponujajo enako pasovno širino). S tem dosežemo hitrejše delovanje in mogoče celo onemogočimo preobremenitev enega samega ponudnika.

Kot zanimivost omenimo še *nadomestni način delovanja* (ang.: failover). Nadomestni način delovanja je avtomatična sposobnost preklopa delovanja na redundante vire. Če določena organizacija uporablja več ponudnikov interneta in eden od njih izpade, bo nadomestni način delovanja avtomatično preusmeril ves promet na preostale ponudnike, sistem za enakomerno porazdelitev prometa pa bo poskrbel, da bo "višek prometa" enakomerno porazdeljen na preostale delujoče ponudnike.

Osnovna topologija omrežja G (Slika 4.2-1a) omogoča dva različna tipa uravnotežitve: vrhovno uravnotežitev (Slika 4.2-1b) in popolno uravnotežitev prometa (Slika 4.2-1c).

O **vrhovni uravnotežitvi** govorimo takrat, ko je omrežje uravnoteženo na zadnji etapi pred bazno postajo – vsaka povezava zadnje etape je enako obremenjena (v primeru (Slika 4.2-1b) je ta obremenitev enaka 12).

Popolna uravnotežitev pa nastopi, ko je omrežje uravnoteženo v vseh etapah – vse povezave določene etape so enakomerno obremenjene (v primeru (Slika 4.2-1c) zadnje etape obremenitev znaša 6, v primeru predzadnje etape 5...).



Slika 4.2-1
Različni tipi uravnovežitev.

Poleg dveh omenjenih tipov uravnovežitve bi lahko definirali še enega: hirarhičnega [32].

V tezi smo že omenili algoritme za iskanje najkrajših poti po grafih (Poglavje 2.6) in za izračun BFS algoritmov (Poglavje 2.5). Oba algoritma namreč zadostujeta pogoju: najkrajša pot od senzorske enote do ponora oziroma, minimalno število etap od senzorske enote do bazne postaje. Kako pa je z uravnovežitvijo? Algoritmi iskanja najkrajših poti po grafu navadno vrnejo zelo ne-uravnoveženo drevo (Slika 4.2-2a), po drugi strani pa algoritmi za uravnovežitev ne upoštevajo najkrajših poti (Slika 4.2-2b).



Slika 4.2-2
Drevo najkrajših poti in uravnoveženo drevo.

4.3 Kako oceniti uravnoveženost?

Da lahko ocenimo kako dobra je porazdelitev čez vse veje drevesa, lahko kot metriko uravnovežitve uporabimo Chebyshevo sumo [32]. Definicija metrik je naslednja: za vse $a \in \mathbb{R}^n$ in $b \in \mathbb{R}^n$;

$$\{a_1, a_2, a_3, \dots, a_n\}; \quad (4.2)$$

$$\{b_1, b_2, b_3, \dots, b_n\} \quad (4.3)$$

in

$$a_1 \geq a_2 \geq a_3 \geq \dots \geq a_n \quad (4.4)$$

$$b_1 \geq b_2 \geq b_3 \geq \dots \geq b_n \quad (4.5)$$

Posledično velja tudi:

$$n \sum_{k=1}^n a_k b_k \geq \left(\sum_{k=1}^n a_k \right) \cdot \left(\sum_{k=1}^n b_k \right) \quad (4.6)$$

Definirajmo W_{bi} kot utež na i -ti veji vpete topologije drevesa. Definirajmo še vektor uteži:

$$w = \{W_{b1}, W_{b2}, W_{b3}, \dots, W_{bn}\} \quad (4.7)$$

Da lahko ocenimo uravnovežitev vseh vej drevesa predpostavimo še, da velja $a = b = w$. V tem primeru potem velja:

$$n \sum_{k=1}^n W_{bk}^2 \geq \left(\sum_{k=1}^n W_{bk} \right)^2 \quad (4.8)$$

ali drugače

$$1 \geq \frac{\left(\sum_{k=1}^n W_{bk} \right)^2}{n \sum_{k=1}^n W_{bk}^2} \quad (4.9)$$

Faktor uravnoveženja (ang.: balanced factor), ki ga bomo uporabljali za oceno uravnoveženja tako definiramo kot:

$$\theta = \frac{\left(\sum_{k=1}^n W_{bk} \right)^2}{n \sum_{k=1}^n W_{bk}^2} \quad (4.10)$$

Pri popolnoma uravnoveženem drevesu pričakujemo faktor uravnoveženja $\theta = 1$. Izračunajmo zato na primer faktor uravnoveženja θ za (Slika 4.2-1c):

θ : Faktor uravnovežitve (Chebysheva suma)

$$\theta_{(\text{Slika 4.2-1c})} = \frac{(6 + 6 + 6 + 6)^2}{4 \cdot (36 + 36 + 36 + 36)} = 1 \quad (4.11)$$

Dejali smo že, da je drevo (Slika 4.2-2a) neuravnoveženo. Poskusimo to dokazati še s faktorjem uravnoveženja:

$$\theta_{(\text{Slika 4.2-2a})} = \frac{(4 + 1 + 4 + 15)^2}{4 \cdot (16 + 1 + 16 + 225)} = 0.558 \quad (4.12)$$

Chebyshevo sumo lahko uporabimo tudi za vpeto drevesa (Slika 4.3-1) na različnih nivojih. Faktor uravnoveženja izračunamo za vsak nivo posebej, katerega smo predhodno določili (uporaba BFS).

Na prvem nivoju je drevo uravnoveženo, zato pričakujemo faktor uravnoveženja $\theta_{(\text{Slika 4.3-1})-L=0} = 1$:

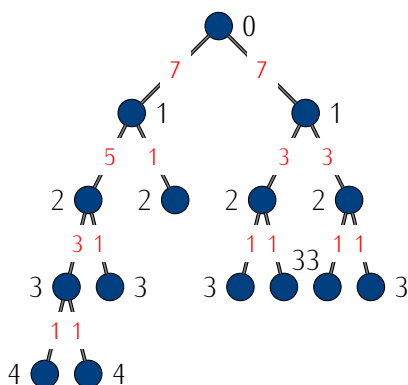
$$\theta_{(\text{Slika 4.3-1})-L=0} = \frac{(7 + 7)^2}{2 \cdot (49 + 49)} = 1 \quad (4.13)$$

Na drugem nivoju drevo več ni uravnoveženo, pričakovan faktor uravnoveženja pa bo manjši od ena ($\theta_{(\text{Slika 4.3-1})-L=1} < 1$), kar potrди tudi izračun:

$$\theta_{(\text{Slika 4.3-1})-L=1} = \frac{(5 + 1 + 3 + 3)^2}{4 \cdot (25 + 1 + 9 + 9)} = 0.818 \quad (4.14)$$

Za tretji nivo znaša faktor uravnovežitve θ :

$$\theta_{(\text{Slika 4.3-1})-L=2} = \frac{(3 + 1 + 1 + 1 + 1 + 1)^2}{6 \cdot (9 + 1 + 1 + 1 + 1 + 1)} = 0.762 \quad (4.15)$$



Slika 4.3-1

Označeno (BFS) uteženo drevo T .

Faktor uravnoteženosti lahko torej brez težav izračunamo tudi za drevesa, pri čemer lahko izračunamo faktor za vsak nivo posebej. V primeru iskanja popolnoma uravnoteženega drevesa pač težimo k temu, da je faktor uravnotežitve na vseh nivojih čim bližje vrednosti $\theta = 1$.

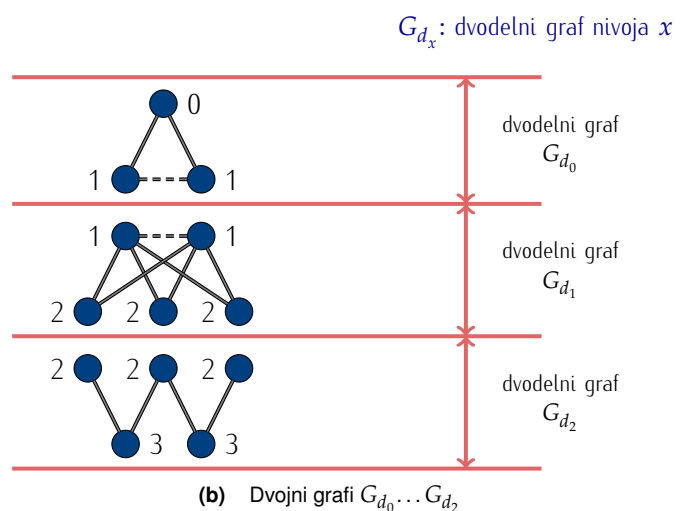
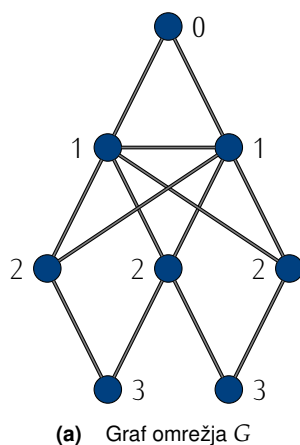
Faktor uravnotežitve θ bomo v nadaljevanju uporabili za oceno uravnoteženosti drevesa.

Predlog algoritma optimalnega polprirejanja

Eden od klasičnih problemov kombinatorične optimizacije je tudi iskanje maksimalnega prirejanja (ujemanja, ang.: matching) v dvodelnih grafih. Problem dvodelnega prirejanja ima mnogo praktičnih aplikacij in znanih algoritmov za izračun le tega [27, 51, 71].

Model BSO je povezan graf $G = (E, V)$ (Slika 5.0-2a), algoritmi dvodelnega prirejanja pa kot vhod uporabljajo dvodelne (bipartitne) grafe.

V poglavju 4.1.2 smo definirali "nivoje" grafa G , par sosednjih nivojev tega grafa pa predstavlja dvodelni graf (Slika 5.0-2b) za potrebe algoritma polprirejanja.



Slika 5.0-2

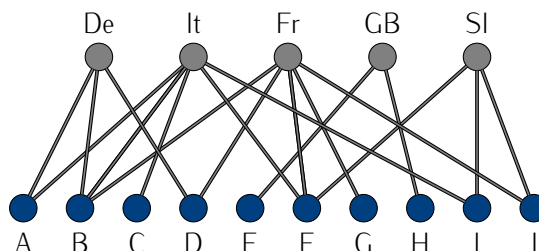
Graf G in izpeljava dvojnih grafov.

5.1 Algoritmi polprirejanja

Problem optimalnega polprirejanja (ang.: Optimal Semi-Matching) bomo poskušali ponazoriti s problemom švicarske banke. Predpostavimo, da v množico ljudi v banki spada 5 bankirjev in 10 strank. Vsak bankir govori drug, en svetovni jezik (Nem-

ško, Angleško, Italijansko, Francosko in Slovensko), vsaka od strank pa enega ali več jezikov. Problem, ki se pojavi je **Določiti vsako stranko ustreznemu bankirju**. Predpostavimo še, da za postrežbo ene stranke potrebujemo eno enoto časa.

Problem lahko najbolje predstavimo kot dvodelni (bipartitni) graf (Slika 5.1-1).



Slika 5.1-1

Problem švicarske banke modeliran z dvojnim grafom G .

Iz (Slika 5.1-1) lahko razberemo, da oseba A govori Nemško in Italijansko (De,It), oseba B Nemško, Italijansko in Francosko, oseba C samo Italijansko ...

Vprašanje, ki si ga zastavimo je **kako razdeliti stranke med bankirje, da bo povprečen čakalni čas stranke najmanjši in obremenitev bankirjev čimbolj enakomerna?**

Na (Slika 5.1-1) definirajmo modra vozlišča kot množico $U = \{A, B, C, D, E, F, G, H, I, J\}$ in siva vozlišča kot množico $V = \{De, It, Fr, GB, SI\}$. Povezave med omenjenimi vozlišči spadajo v množico E . Formalno lahko definiramo dvodelni graf $G = (U \cup V, E)$ v katerem velja $E \subseteq U \times V$.

Algoritmi ali problemi polprirejanja so navadno motivirani z problemi enakomerne porazdelitve virov. Predpostavimo množico opravil in množico strojev, katere lahko zaposlimo z podmnožicami množice opravil. Za dokončanje vsakega opravila potrebujemo eno enoto procesnega časa. Opravila morajo biti med stroje seveda porazdeljena tako, da minimizirajo določeno optimizacijsko zahtevo:

- minimizirati je mogoče maksimalno število opravil določenih kateremukoli stroju,
- minimizirati je mogoče povprečni čas izvedbe opravila,
- maksimirati je mogoče enakomerne porazdelitve opravil vsem strojem ...

Problemi enakomerne porazdelitve virov so lahko statični (kot predpostavljamo mi) ali dinamični, kjer se število opravil spreminja s časom [73]. Problem dinamičnega dodeljevanja virov je lahko rešljiv tudi z algoritmi za statično enakomerno porazdelitev virov, v primeru, da je čas izračunavanja mnogo manjši kot pa čas v katerem se spreminja število opravil.

5.1.1 Predpostavke v algoritmih polprirejanja

Naj bo $G = (U \cup V, E)$ dvodelni graf v katerem imenujemo vozlišča U leva vozlišča in V desna vozlišča. Povezave so definirane kot $E \subseteq U \times V$. Število elementov v množicah U in V definiramo kot $n = |U|$ in $m = |V|$.

Prirejanje v grafu G je množica povezav $M \subseteq E$, kjer je vsako vozlišče v G krajšiče največ ene povezave v M . Vsakemu vozlišču v U je torej prirejeno največ eno vozlišče v V in obratno. *Polprirejanje* v grafu G pa predstavlja množica povezav

U: Leva vozlišča

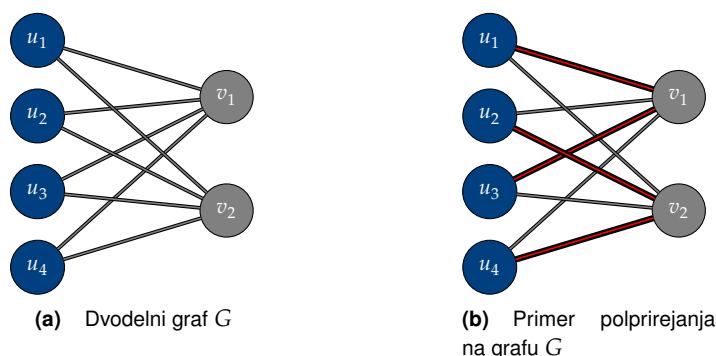
V: Desna vozlišča

M: Prirejanje;
množica povezav E

$M \subseteq E$, kjer je vsako vozlišče $u \in U$ povezano z natanko eno povezavo v V . V tem primeru mora biti valenca vseh vozlišč vsaj 1 ($deg(u) \geq 1$ pri čemer velja, da je $u \in U$).

Že v poglavju (Poglavje 2.1) smo definirali stopnjo vozlišča v kot $deg(v)$. V smislu uravnotežitve opravil pa je stopnja vozlišč tudi obremenitev vozlišča s opravili (število pravil dodeljenih vozlišču). Z $deg_M(v)$ bomo definirali število opravil prirejenih vozlišču v ; $deg_M(v)$ bi lahko imenovali tudi obremenitev vozlišča v .

$deg_M(v)$: Stopnja prirejanja vozlišča v



Slika 5.1-2

Graf G in primer polprirejanja.

Za (Slika 5.1-2b) lahko torej izrazimo $deg(v_1) = 4$ in $deg_M(v_1) = 2$.

Definirajmo ceno prirejanja $cost_M(v)$ za določeno vozlišče $v \in V$ [50]:

$$cost_M(v) = \sum_{i=1}^{deg_M(v)} i = 1 + 2 + \dots + deg_M(v) \quad (5.1)$$

$cost_M(v)$: Cena polprirejanja vozlišča v

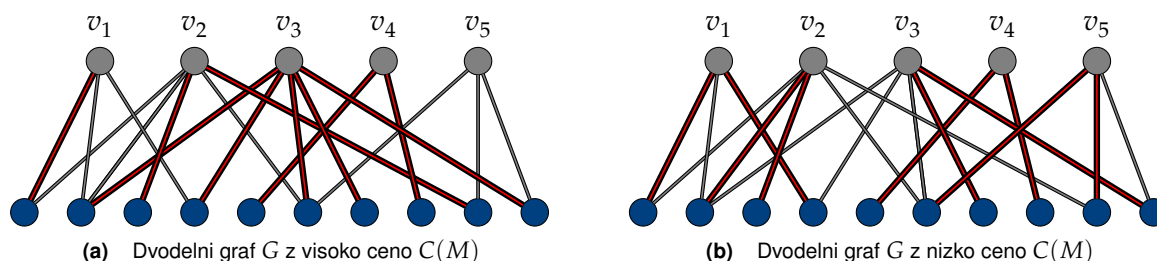
Skupno ceno prirejanja pa definirajmo kot [50]:

$$C(M) = \sum_{v \in V} cost_M(v) \quad (5.2)$$

$C(M)$: Skupna cena polprirejanja

Polprirejanje M je optimalno, če je cena prirejanja minimalna [23, 50].

M je torej optimalno polprirejanje, če je $C(M)$ minimalna možna. Kot ceno pa lahko definiramo tudi leksikografsko razporeditev uteži [23].



Slika 5.1-3

Primer visoke in nizke cene $C(M)$.

V tabeli 5.1-1 imamo primer izračuna cene prirejanja za dva različna dvodelna grafa. Ugotovimo lahko, da je graf G na (Slika 5.1-3b) bolj pravilno uravnotežen

Tabela 5.1-1Izračun cenovne vrednosti $C(M)$ za (Slika 5.1-3a) in (Slika 5.1-3b).

$deg_M(v)$	(Slika 5.1-3a)	(Slika 5.1-3b)
$deg_M(v_1)$	1	3
$deg_M(v_2)$	3	3
$deg_M(v_3)$	15	3
$deg_M(v_4)$	3	3
$deg_M(v_5)$	0	3
$C(M)$	22	15

kot graf na (Slika 5.1-3b). To potrjuje tudi izračun njune cene polprirejanja (Tabela 5.1-1).

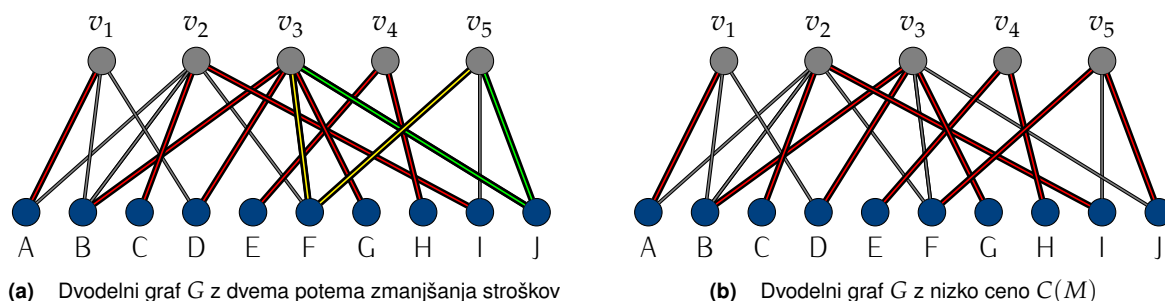
Predstavimo še termin *alternirajoča pot*. Za določeno polprirejanje M grafa G je alternirajoča pot zaporedje povezav :

P: Pot

$$P = (\{v_1, u_1\}, \{u_1, v_2\}, \dots, \{u_{k-1}, v_k\}) \quad (5.3)$$

kjer velja $v_i \in V$, $u_i \in U$ in $\{v_i, u_i\} \in M$ ter $\{u_i, v_{i+1}\} \notin M$ za vsak i . Zaradi priročnosti in krajšega pisanja pa pot včasih definiramo le kot zaporedje vozlišč: $(v_1, u_1, \dots, u_{k-1}, v_k)$.

Kadar obstaja alternirajoča pot za katero velja $deg_M(v_1) > deg_M(v_k) + 1$ potem to pot P imenujemo *pot zmanjšanja stroškov*. V kolikor po tej poti zamenjamo prirejene in ne-prirejene povezave se cena prirejanja zmanjša.

**Slika 5.1-4**

Primer znižanja stroškov z uporabo alternativnih poti.

Z uporabo dveh poti zmanjšanja stroškov (Slika 5.1-4a):

1. $P = (v_3, F, v_5)$ - rumena barva
2. $P = (v_3, J, v_5)$ - zelena barva

smo uspeli zmanjšati skupno ceno polprirejanja iz $C(M) = 22$ na $C(M) = 16$. Za doseg minimalne skupne cene polprirejanja pa bi morali uporabiti še eno alternativno pot (v_3, B, v_1) .

Dokazano je bilo, da je M optimalno polprirejanje, če v M ne obstaja nobena pot zmanjšanja stroškov [23, 50].

5.1.2 Znani algoritmi polprirejanja

V literaturi najdemo nekaj algoritmov, ki se ukvarjajo s tematiko sorodno optimalnemu polprirejanju na neutženih [42, 50] in uteženih [83] grafih. Specifično problematiko optimalnega polprirejanja pa obravnavajo avtorji v [23] in delno tudi v [50]. V nadaljevanju se bomo osredotočili na dva (A_{SM1} in A_{SM2}), predstavljena v [50].

A_{SM1} algoritem za optimalno polprirejanje

Časovna zahtevnost algoritma A_{SM1} znaša $O(|U||E|)$, kar je enako kot pri Madžarski metodi [71, 75].

A_{SM} :
Algoritem polprirejanja

Ob začetku izvajanja algoritma A_{SM1} je množica M prazna, potem pa se v omenjeni množici vse skozi shranjuje delno polprirejanje. V vsaki iteraciji algoritem uporabi določeno vozlišče iz množice U in mu priredi vozlišče v V , kjer velja, da je $deg_M(v)$ čim manjši. V vsaki iteraciji se vrednost $deg_M(v)$ poveča za 1 natanko enemu vozlišču v iz V [50].

Z drugimi besedami: (Algoritem 4) vozlišča iz U dodaja drugega za drugim v M . Pri dodajanju vozlišča u pogledamo vse potencialne sosede in izberemo takega, ki je najmanj obremenjen. če tako vozlišče ne obstaja potem poskušamo z iskanjem alternativnih poti najti vozlišče v_x , ki je najmanj obremenjeno. Če takšna pot obstaja (Algoritem 4 , line 13), vzdolž te poti zamenjamo prirejene in ne-prirejene povezave in tako zmanjšamo obremenitev (Algoritem 4 , line 22) [50].

A_{SM2} algoritem za optimalno polprirejanje

Časovna zahtevnost drugega algoritma A_{SM2} znaša $O(\min\{|U|^{\frac{3}{2}}, |U||V|\} \cdot |E|)$, kar je slabše kot algoritem A_{SM1} ($O(|U||E|)$). Praktično pa, kot bomo videli algoritem A_{SM2} deluje hitreje kot A_{SM1} [50].

Algoritem A_{SM2} spada med t.i. "požrešne algoritme", ker za svoje delovanje v prvi fazi uporablja heuristiko. Algoritem je razdeljen na dva dela dela in sicer:

1. Iskanje začetnega polprirejanja M , z uporabo heuristike Algoritem 5 .
2. Dokler obstajajo poti zmanjševanja stroškov, izvajanje zamenjave prirejenih in ne-prirejenih robov vzdolž teh poti P Algoritem 6 .

Iskanje začetnega polprirejanja pričnemo tako, da razvrstimo vozlišča v U po naraščajoči stopnji vozlišča ($deg(u_x)$). Vsako vozlišče iz U (začnemo pri tistih z najmanjšo stopnjo) nato izbere najmanj obremenjenega soseda iz V ($deg_M(v_x)$), če obstaja več takih vozlišč men njimi izberemo takega, ki ima najmanjšo stopnjo ($deg(v_x)$) (Algoritem 5 , line 4). Podobno strategijo predlaga tudi [15].

Iskanje poti zmanjšanja stroškov je funkcija s katero najprej najdemo še od funkcije neobiskano vozlišče $v \in V$ z maksimalnim $deg_M(v)$. Iz omenjenega vozlišča izvedemo DFS algoritem. Če iskanje iz omenjenega vozlišča ne vrne poti zmanjšanja stroškov to pomeni, da taka pot ne obstaja [50].

Omenjena strategija omogoča, da je že začetno polprirejanje blizu optimalnega. Dobra heuristika in s tem gradnja začetnega polprirejanja M je ključ do hitrega izvajanja algoritma. Bolj kot je začetno polprirejanje optimalno hitrejši je algoritem.

Algoritem 4: Algoritem optimalnega polprirejanja A_{SM1}

Data: Dvodelni graf G
Result: Optimalno polprirejanje M

- 1 Začetni pogoji:
- 2 M ; Množica polprirejanja, na začetku prazna
- 3 Q ; Vrsta, na začetku prazna
- 4 **foreach** $root \in U$ **do**
- 5 $Q.Pocisti()$;
- 6 $Q.Vstavi(root)$;
- 7 **while** $Q \neq \emptyset$ **do**
- 8 $w = Q.Izberi()$
- 9 /* Sosedje, ki niso v množici M */
- 10 **if** $w \in U$ **then**
- 11 $N = NeprirejeniSosedje(w)$
- 12 **else**
- 13 /* Sosedje v množici M */
- 14 $N = PrirejeniSosedje(w)$
- 15 **if** $najboljsiV == \emptyset \parallel deg_M(w) < deg_M(najboljsiV)$
- 16 **then**
- 17 $najboljsiV = w$;
- 18 **foreach** $n \in N$ **do**
- 19 **if** $!S.Vsebuje(n)$ **then**
- 20 $n.Stars = w$;
- 21 $Q.Vstavi(n)$;
- 22 $S = S \cup N$;
- 23 $v = najboljsiV$;
- 24 $u = v.Stars$;
- 25 **while** $u \neq root$ **do**
- 26 $v = u.Stars$;
- 27 $M.Odstrani(\{u, v\})$;
- 28 $u = v.Stars$;
- 29 $M.Dodaj(\{u, v\})$;

Algoritem 5: Izračun začetnega polprirejanja A_{SM2}

Data: Dvodelni graf G
Result: Začetno polprirejanje M

- 1 Začetni pogoji:
- 2 M ; Množica polprirejanja, na začetku prazna
- 3 /* Razvrstimo elemente v U naraščajoče po $deg(u)$ */
- 4 $U.Razvrsti()$; **foreach** $u \in U$ **do**
- 5 $v = MIN(deg(MIN(deg_M(v_x), x \in V)))$;
- 6 $M.Dodaj(\{u, v\})$;
- 7 **end**

Algoritem 6: Iskanje poti znižanja stroškov A_{SM2}

```

1 S; Množica še neobiskanih vozlišč iz V
2 PoisciPotZnizanjaStroskov();
3 S = V;
4 while S ≠ ∅ do
5   koren = MAX(degM(vx), x ∈ S);
6   S.Odstrani(koren);
7   koren.Stars = null;
8   DoDFS(koren)
9 end

```

Algoritem 7: DoDFS A_{SM2}

```

1 DoDFS(vozlisce);
                                     /* Sosedje, v množici M */
2 foreach u ∈ PrirejeniSosedje(vozlisce) do
3   u.Stars = v;
                                     /* Sosedje, ki niso v množici M */
4   foreach w ∈ NeprirejeniSosedje(u) do
5     if w ∉ S Or degM(w) > degM(vozlisce) then
6       continue;
7     end
8     S.Odstrani(w);
9     w.Stars = u;
10    if degM(w) ≤ degM(koren) - 2 then
11      Return:true;
                                     /* Pot znižanja stroškov najdena */
12    end
13  end
14 end

```

5.2 Predlog alternativnega algoritma optimalnega polprirejanja

Največja časovna zahtevnost algoritma A_{SM3} znaša prav tako, kot A_{SM1} $O(|U||E|)$, kar je enako Madžarski metodi [71, 75], vendar se zaradi dodatka hevristične izboljšave računski zahtevnost zmanjša.

Algoritem deluje podobno kot (Algoritem 5): Na začetku algoritma razvrstimo vozlišča v U naraščajoče od tistega z najmanjšo stopnjo (število sosedov) do tistega z največjo (Algoritem 8, line 4). Vozlišča iz U zaporedoma prirejamo vozliščem iz V , kjer izberemo najmanj obremenjeni vozlišče iz V ($deg_M(v_x)$), če pa obstaja več takih vozlišč, med njimi izberemo tistega, ki ima najmanjšo stopnjo ($deg(v_x)$) (Algoritem 8, line 7).

Iz izbranih vozlišč (u_c in v_c) ustvarimo pot P_c (Algoritem 8, line 8). V kolikor je stopnja polprirejanja vozlišča v_c za manj kot dva večja od stopnje najmanj obremenjenega vozlišča v polprirejanju *najboljsiV* (Algoritem 8, line 9), pot P_c dodamo v M . Drugače poiščemo alternativno pot do vozlišča v_x iz V , katerega stopnja polprirejanja je enaka stopnji polprirejanja vozlišča *najboljsiV* (Algoritem 8, line 13). Z dodajanjem vozlišča se stopnja polprirejanja v vsaki iteraciji poveča za 1.

A_{SM3} : Predlagan alternativni algoritem optimalnega polprirejanja

Algoritem 8: Algoritem optimalnega polprirejanja A_{SM3}

```

Vhod: Dvodelni graf  $G$ 
Izhod: Optimalno polprirejanje  $M$ 
1  $i = 0$ ;
2  $M_i = 0$ ,  $najboljsiV.deg_M = 0$ ;
3  $U = \{u_1, u_2, \dots, u_n\}$ ;
4  $U.sortiraj(deg(u_x))$ ;
5 for  $i = 1 \rightarrow i = n$  do
6    $u_c = Izberi(U_i)$ ;
7    $v_c = IzberiUstreznegaSoseda(u_c)$ ;
8    $P_c = \{u_c, v_c\}$ ;
9   if  $v_c.deg_M \leq najboljsiV.deg_M + 2$  then
10     $M = M \cup PolprirejenePovezave(P_c)$ ;
11  end
12  else
13     $PopraviPotP(ref P_c)$ ;
14     $M = M \cup PolprirejenePovezave(P_c)$ ;
15  end
16   $najboljsiV = IzberiNajboljsiV(V)$ ;
17  while  $najboljsiV.deg_M == najboljsiV.deg$  do
18     $V.Odstrani(najboljsiV)$ ;
19     $najboljsiV = IzberiNajboljsiV(V)$ ;
20  end
21 end

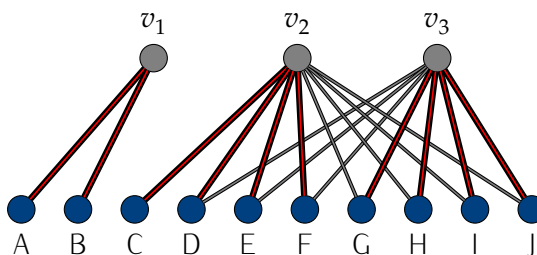
```

Na koncu iteracije izberemo še najmanj obremenjeno vozlišče iz V (Algoritem 8, line 16), to vozlišče pa uporabimo samo, če obstaja praktična možnost njegove nadaljne uporabe (Algoritem 8, line 17). Zakaj je to pomembno?

V kolikor za vozlišče velja, da je njegova stopnja polprirejanja za več kot 1 večja (enačba 5.4) od vozlišča z najmanjšo stopnjo polprirejanja, potem je mogoče, da taka pot obstaja.

$$(deg_M(v_{trenutni}) - \min(deg_M(v_x))) > 1 \quad (5.4)$$

V praksi pa ni nujno, da pot zmanjšanja stroškov dejansko obstaja, kar prikazuje slika 5.2-1. To se zgodi takrat, ko je stopnja vozlišča $deg(v_x)$ enaka stopnji polprirejanja $deg_M(v_x)$.



Slika 5.2-1
Optimalno polprirejanje na dvodelnem grafu G .

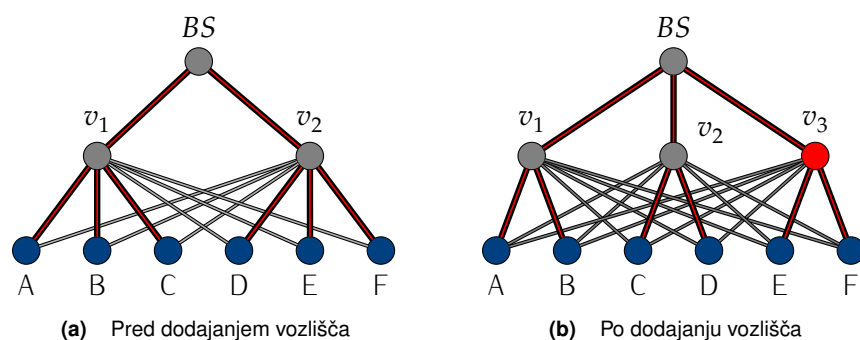
Drevo sestavljamo tako, da postopno prirejamo vozlišča iz množice U vozliščem

v množici V . Med eksperimentiranjem smo poskusili tudi obratno pot: postopno prirejanje V vozlišč vozliščem iz množice U , vendar se je omenjena metoda izkazala za precej slabšo.

5.2.1 Dodajanje in odstranjevanje vozlišč(a)

Prednost predlaganega protokola je tudi v tem, da lahko izjemno preprosto in hitro dodamo ali odvezamo vozlišče iz topologije. To pomeni, da ni potrebe računati celotno drevo znova, pač pa to storimo le za dva nivoja (med katera je bilo vozlišče dodano).

Predpostavimo, da je drevo optimalno polprirejeno preden dodamo vozlišče. Kadar vozlišče dodamo je potrebno na dveh nivojih usmerjevalnega drevesa izvesti algoritem za iskanje poti znižanja stroškov (Algoritem 6). Če dodamo samo eno vozlišče algoritem v samo eni iteraciji "popravi" usmerjevalno drevo na zgornjem nivoju. Na nivoju nižje algoritem potrebuje več iteracij (odvisno od št. vozlišč) – polprirejanje je spet optimalno.



Slika 5.2-2

Primer dodajanja vozlišča.

Za primer si pogledjmo omrežje na sliki (Slika 5.2-2a). Če omrežju dodamo novo vozlišče (v_3) (Slika 5.2-2b) je potrebno izvesti algoritem iskanja poti znižanja stroškov dvakrat. Na višjem nivoju enkrat (dodajanje povezave $v_3 - BS$). Na nižjem nivoju algoritem izvedemo enkrat, algoritem pa potrebuje dvoje iteracij (povezavo $C - v_1$ zamenja $F - v_3$, povezavo $E - v_2$ pa zamenja povezava $F - v_3$).

5.3 Primerjava algoritmov polprirejanja

5.3.1 Primerjava delovanja algoritmov

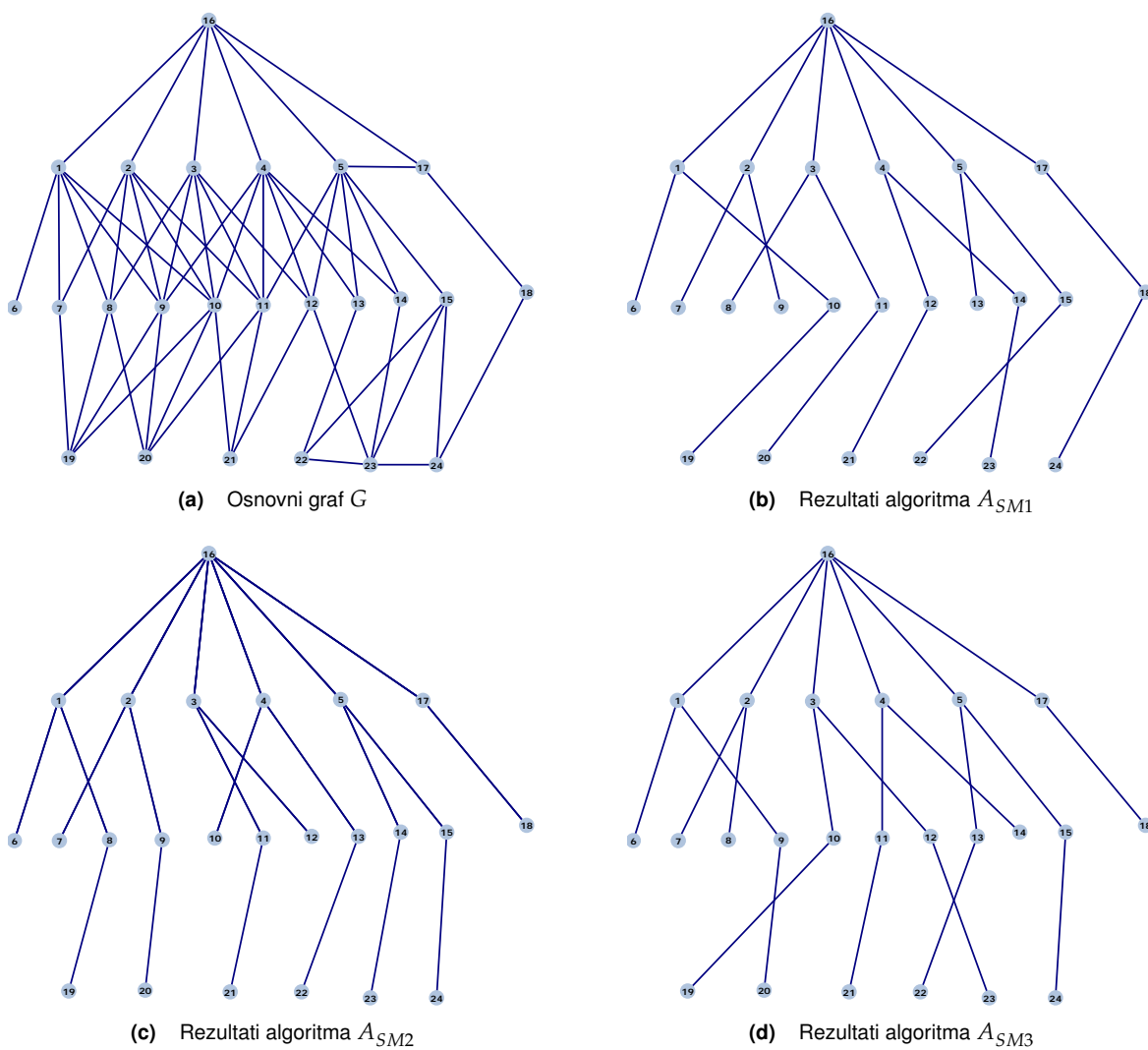
Tabela 5.3-1

Faktorji uravnotežitve θ za grafe (Slika 5.3-1).

Graf	A_{SM1} (Slika 5.3-1b)	A_{SM2} (Slika 5.3-1c)	A_{SM3} (Slika 5.3-1d)
θ	0.969	0.948	0.928

Za potrebe testiranja pravilnosti delovanja algoritma smo izbrali tri grafe:

1. Naključno generirano in povezano drevo (Slika 5.3-1).
2. Mrežast graf valence približno 4 (Slika 5.3-2).

**Slika 5.3-1**

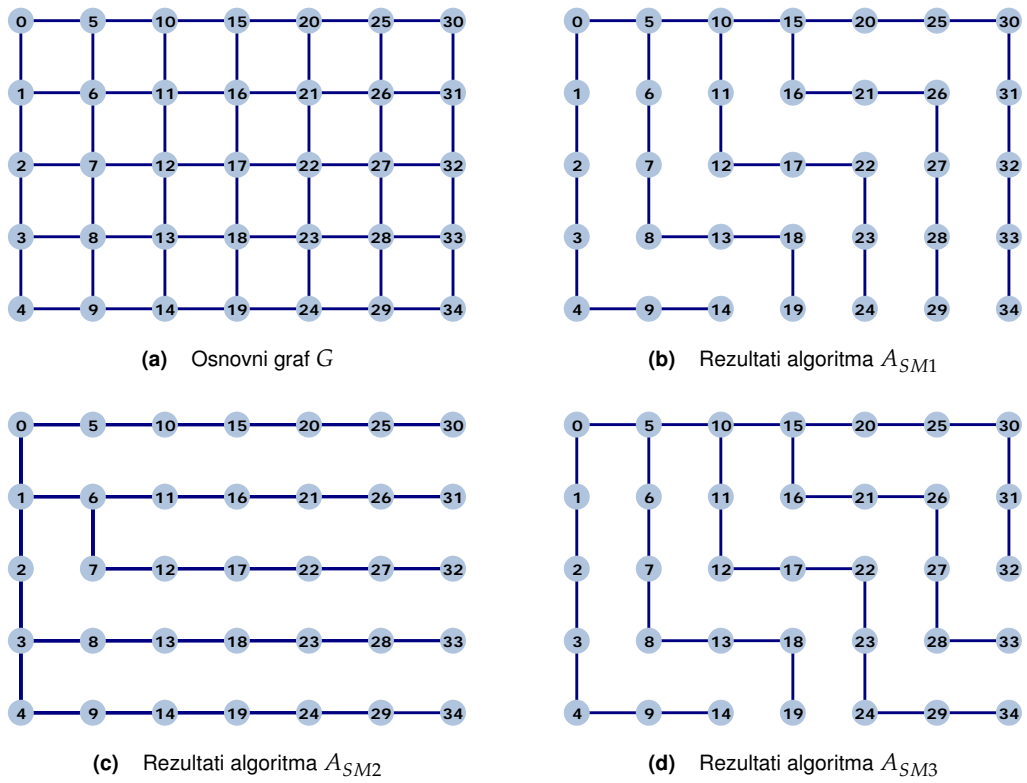
Primerjava delovanja algoritmov za optimalno polprirejanje.

Tabela 5.3-2Faktorji uravnovežitve θ za grafe (Slika 5.3-2).

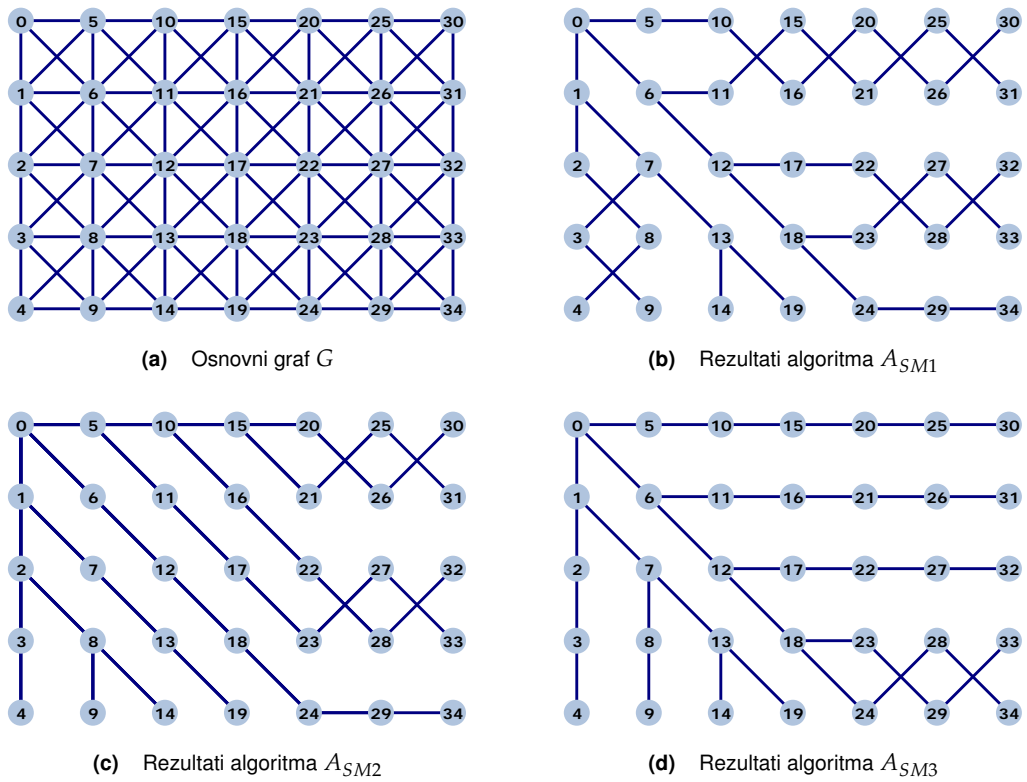
Graf	A_{SM1} (Slika 5.3-2b)	A_{SM2} (Slika 5.3-2c)	A_{SM3} (Slika 5.3-2d)
θ	0.705	0.705	0.705

Tabela 5.3-3Faktorji uravnovežitve θ za grafe (Slika 5.3-3).

Graf	A_{SM1} (Slika 5.3-3b)	A_{SM2} (Slika 5.3-3c)	A_{SM3} (Slika 5.3-3d)
θ	0.838	0.838	0.838



Slika 5.3-2
Primerjava delovanja algoritmov za optimalno polprirejanje.



Slika 5.3-3
Primerjava delovanja algoritmov za optimalno polprirejanje.

3. Mrežast graf valence približno 8 (Slika 5.3-3).

Iz tabele 5.3-1 je razvidno, da najboljše rezultate uravnovežitve, ocenjene po metodi uravnovežitvenega faktorja θ vrne algoritem A_{SM1} . V bistvu pa ni večjih odstopanj med faktorji uravnovežitve. V primeru mrežastih grafov valence 4 in 8 pa v rezultatih sploh ni razlike (tabela 5.3-2 in tabela 5.3-3).

5.3.2 Primerjava hitrosti izvajanja algoritmov

Primerjavo algoritmov smo izvajali na računalniku konfiguracije:

CPU : Intel Core 2 Quad Q9450.

RAM : 8Gb, DDR2 – 1066 Mhz.

GPU : Nvidia GeForce 9600GT.

OS : Windows 7, 64Bit.

Za programiranje smo uporabili Visual Studio 2010 in .NET ogrodje 4.0. Zaradi primerljivosti algoritmov smo v vseh treh algoritmih uporabili enake osnovne (int, string...) in generične podatkovne tipe

Generična vrsta (Queue) : "System.Collections.Generic.Queue<T>".

Generični HashSet : "System.Collections.Generic.HashSet<T>".

Generična Lista : "System.Collections.Generic.List<T>".

Standardna Hash tabela (HashTable) : "System.Collections.HashTable".

Hitrost izvajanja algoritmov bi bila zagotovo večja ob uporabi C++ programskega jezika in implementaciji svojih lastnih podatkovnih tipov, vendar smo se zaradi razvoja SPARCsoft™ programskega okolja odločili, da za jezik C#. V poglavju (Poglavje B.2) bomo okolje SPARCsoft™ na kratko tudi predstavili.

Za testiranje pravilnosti delovanja algoritmov in merjenje časa njihovega izvajanja smo uporabili družine grafov, predlagane že v [28, 50]. Mednje sodijo:

- **Naključni grafi** kjer določimo fiksno število vozlišč n . Povezave med vozlišči definiramo z verjetnostjo p dobljeno s pomočjo Poissonove porazdelitve.
- **Šibko povezani omrežni grafi**, kjer $M \cdot N$ predstavlja število vozlišč, valenca vozlišča pa je štiri (razen za vozlišča na robovih) – (Slika 5.3-2a).
- **Krepko povezani omrežni grafi**, kjer $M \cdot N$ predstavlja število vozlišč, valenca vozlišča pa je osem (razen za vozlišča na robovih) – (Slika 5.3-3a).
- **Vrv** je graf, kjer n definira število vozlišč razdeljenih med N_p poti P , zakoreninjenih v enem vozlišču.

Testiranja najbolje predstavimo z (Algoritem 9). Za testiranje hitrosti smo uporabili štiri osnovne tipe grafov (naključni, omrežni, vrvi) in par pod-tipov grafov (naključni z verjetnostjo povezave $p = 0.075$, naključni z verjetnostjo povezave

Algoritem 9: Algoritem testiranja hitrosti izvajanja algoritmov A_{SM1}, A_{SM2} in A_{SM3}

Result: Rezultat izvajanja algoritmov A_{SM1}, A_{SM2} in A_{SM3}

```

1 Rezultat;
2 foreach TipGrafa in GRAPH TYPES do
3   UstvariNovoDatotekoRezultatov(Res);
4   for Moc = 11 to 19 do
5     for i = 1 to 100 do
6       G = UstvariGraf(TipGrafa, 2Moc);
7       for i = 1 to 10 do
8         Rezultat.ASM1 = IzvediASM1(G);
9         Rezultat.ASM2 = IzvediASM2(G);
10        Rezultat.ASM3 = IzvediASM3(G);
11        ZapisiVDatoteko(Res);
12      end
13    end
14  end
15 end
```

$p = 0.075 \dots$). Vsak tip grafa (vključno z pod-tipi) testiramo z devetimi različnimi števili vozlišč velikosti od $n = 2^{11}$ do $n = 2^{19}$ (Algoritem 9, line 4).

Na podlagi števila vozlišč in tipa grafa $100x$ ustvarimo nov graf G (Algoritem 9, line 5) na katerem katerem desetkrat izvedemo algoritem izračuna A_{SM1}, A_{SM2} in A_{SM3} (Algoritem 9, line 7). Rezultate shranjujemo v datoteko.

Rezultate smo uvozili v Matlab in najprej povrečili rezultate za vsak algoritem posebej. Meritvi, ki sta izmed 10 notranjih najbolj izstopali smo izločili iz povprečenja.

Naključni grafi

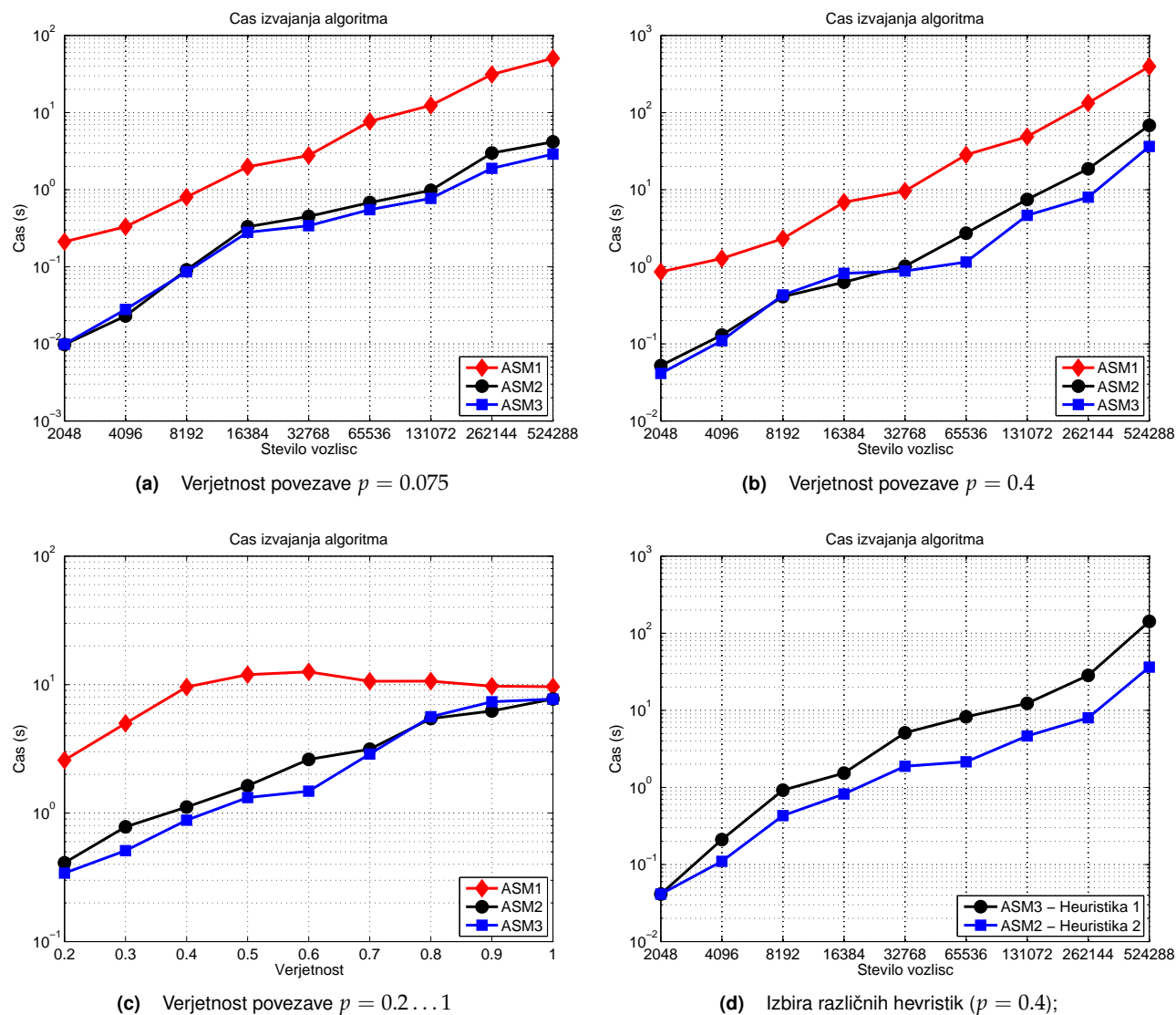
Distribucija brezžičnih senzorskih vozlišč v brezžičnem senzorskem omrežju je navadno naključna [22, 35, 74], zato smo se odločili da bomo največjo pozornost pri testiranju posvetili naključnim grafom.

Iz (Slika 5.3-4a) in (Slika 5.3-4b) vidimo, da je časovno najbolj zahteven algoritem A_{SM1} , algoritem A_{SM2} deluje veliko hitreje kot A_{SM1} , najhitrejši algoritem pa je A_{SM3} .

Če rezultate ovrednotimo procentualno je algoritem A_{SM3} v povprečju 14,5% hitrejši (najboljši rezultat 36,8% pri 2^{18} vozliščih) za (Slika 5.3-4a) in v povprečju 23,8% hitrejši (najboljši rezultat 57,5% pri 2^{16} vozliščih) za (Slika 5.3-4b).

Na (Slika 5.3-4c) lahko vidimo hitrost izvajanja algoritma pri konstantnem številu vozlišč ($n = 2^{15}$). Opazimo lahko, da se hitrost izvajanja algoritma A_{SM1} od verjetnosti $p = 0,4$ ne spreminja bistveno, upada pa hitrost izvajanja algoritmom A_{SM2} in A_{SM3} . Padanje hitrosti algoritma A_{SM2} in A_{SM3} je posledica razvrščanja elementov v listah ($V.Sort()$), ki je uporabljena v algoritmih zaradi izbire sosedu (spomnimo: izberemo najmanj obremenjenega sosedu, če je takih več, izberemo tistega, ki ima najmanjšo stopnjo vozlišča). Algoritme bi lahko še pohitrili z eliminacijo potrebnega razvrščanja.

Primerjava na (Slika 5.3-4d) opisuje uporabo različnih hevrstik za izbiranje sose-



Slika 5.3-4

Čas izvajanja algoritma na naključnem grafu.

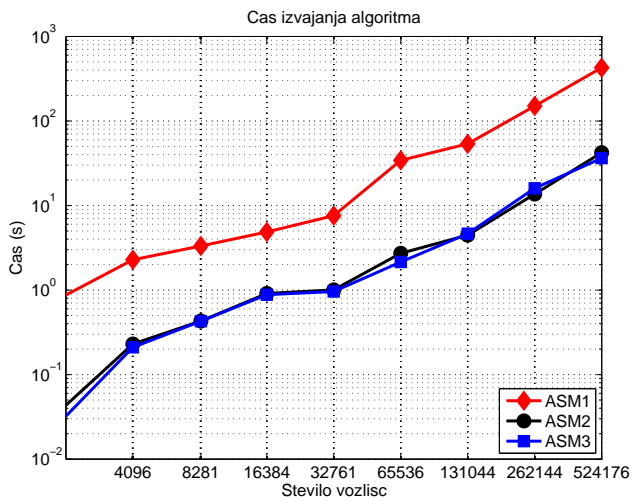
dov, katero uporabljata algoritma A_{SM2} in A_{SM3} . Videti je, da je pri izbiri smotrne hauristike (spomnimo: izberemo najmanj obremenjenega sosedu, če je takih več, izberemo tistega, ki ima najmanjšo stopnjo vozlišča) rezultat v povprečju 54,76 % boljši kot pri izbiri slabe hevristike (recimo: izberemo najmanj obremenjenega sosedu, če je takih več, pa izberemo naključnega med njimi).

Mrežni grafi

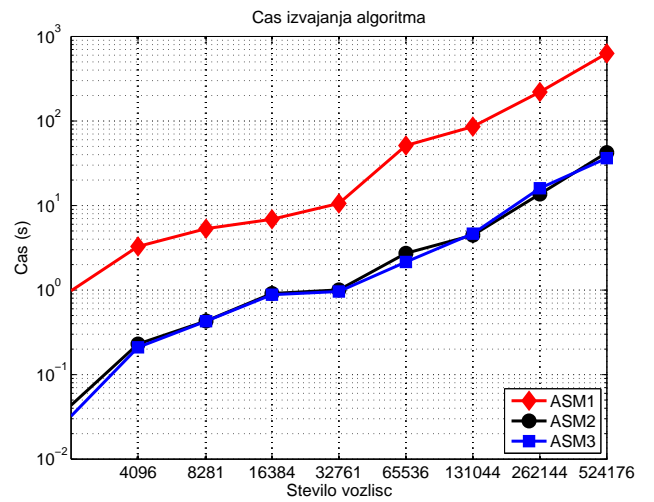
Pri mrežnih grafih valence 4 in 8 ne vidimo posebne razlike med (Slika 5.3-5a) in (Slika 5.3-5b). Iz tega sledi, da valenca mrežnih grafov praktično ne vpliva na hitrost izvajanja algoritmov.

Grafi vrvi

Rezultati pri vrvnih grafih ((Slika 5.3-6a) in (Slika 5.3-6b)) so praktično enako, kar je smiselno, saj ima vsak algoritem v določeni iteraciji le eno možnost izbire (izbrati



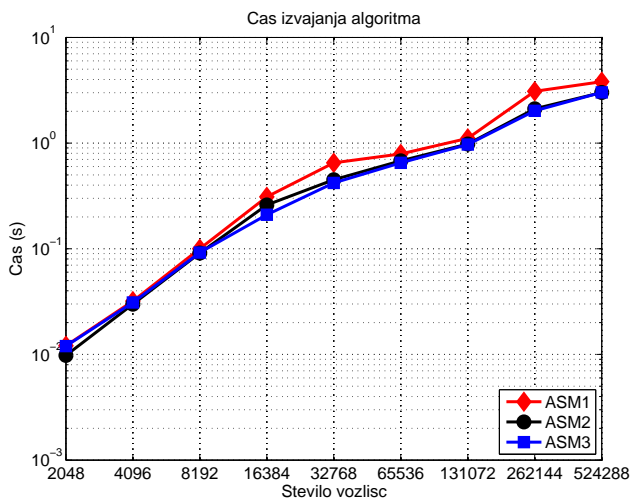
(a) Mrežni graf valence $deg(v) = 4$



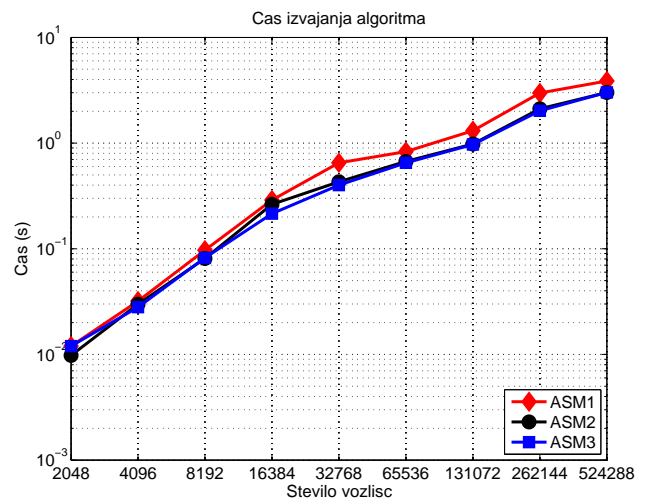
(b) Mrežni graf valence $deg(v) = 8$

Slika 5.3-5

Čas izvajanja algoritma na mrežnem grafu.



(a) Graf $n = 3$ vrvi vpetih v eni točki



(b) Graf $n = 6$ vrvi vpetih v eni točki

Slika 5.3-6

Čas izvajanja algoritma na mrežnem grafu.

soseda na vrvi).

5.4 Algoritmi polprirejanja za utežene grafe

Problem iskanja optimalnega polprirejanja za neutežene dvodelne grafe je dokazano rešljiv v polinomskem času. To je naprej dokazal Harvey et. al. [50], nato pa še potrdil Fakcharoenphol et. al. [42] in Bokal et.al. [23]. Računska zahtevnost algoritma za izračun optimalnega polprirejanja uteženih grafov pa je ostal odprt problem. Leta 2007 je Chor Ping Low predpostavil in dokazoval, da je problem iskanja optimalnega polprirejanja *NP* težak problem [82, 83].

Za rešitev omenjenega problema lahko uporabimo algoritem predlagan s strani [83], ali pa priredimo algoritem iz [76, 103].

Predlog usmerjevalnega protokola: SPaRCRoute

6.1 Predpostavke in uvod

Skozi raziskave in študijo gradiva o usmerjanju v BSO smo prišli do dveh temeljnih vprašanj: ali število etap v BSO in uravnotežitev prometa vplivata na delovanje, zakasnitve in energijsko porabo BSO? V nadaljevanju bomo pokazali, da je temu tako...

Posledično predlagamo nov usmerjevalni algoritem za senzorska omrežja, ki uporablja minimalno število etap (kar dosežemo z algoritmi BFS) in prometno uravnotežitev na vseh nivojih (algoritmi optimalnega polprirejanja).

Predlagan algoritem usmerjanja lahko predstavimo kot hibridni usmerjevalni protokol. Za svoje delovanje uporablja vse tri osnovne principe komuniciranja:

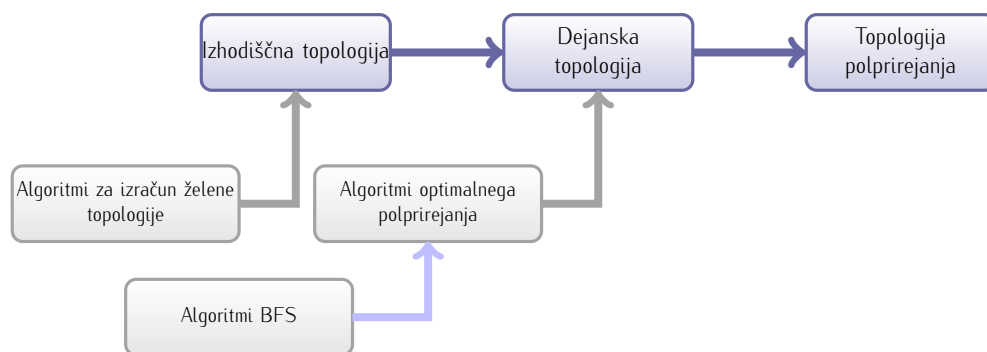
1. *točka-točka; unicast* (ang.: peer-to-peer ali point-to-point), kjer enota točno naslovi drugo enoto.
2. *točka-več točk; multicast*, kjer enota naslovi točno določene druge enote.
3. *točka-vse točke; broadcast*, kjer enota naslovi vse druge enote.

6.2 Začetna in dejanska topologija ter topologija polprirejanja

Ko smo že dejali (Poglavje 3.1) se senzorsko omrežje "vzpostavi" pri maksimalni oddajni moči vseh senzorskih vozlišč. Takemu stanju, ki tvori povezan graf pravimo **začetna topologija** [18].

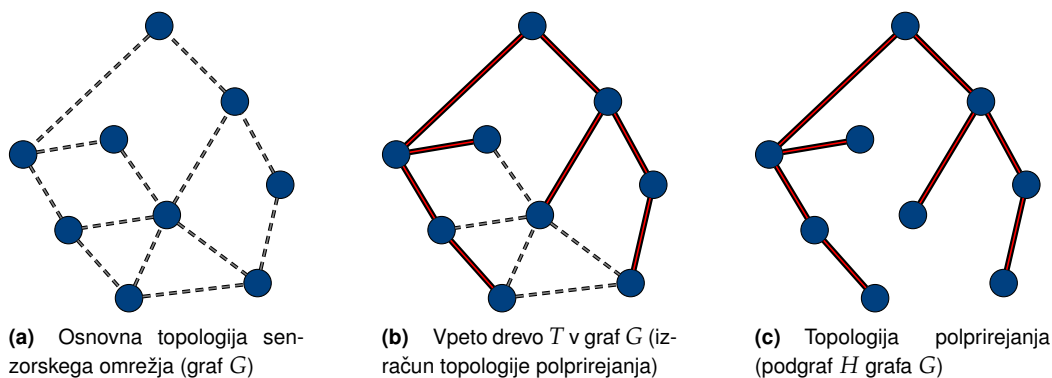
Po vzpostavitvi omrežja uporabimo algoritme za izračun potrebne/želene topologije. Z algoritmi topologije dosežemo, da se omrežje reorganizira in uporablja izračunano topologijo (Slika 6.2-2a). Takšno topologijo imenujemo **dejanska (ali realna) topologija** [18].

Za potrebe usmerjevalnega algoritma pa bomo definirali še tretji tip topologije: **topologija polprirejanja**. Topologija polprirejanja je v našem primeru drevo, vpeto v

**Slika 6.2-1**

Od začetne do topologije polprirejanja.

realno topologijo (Slika 6.2-2b). Topologija polprirejanja je torej podgraf H grafa realne topologije G (Slika 6.2-2c).

**Slika 6.2-2**

Od realne topologije do topologije polprirejanja.

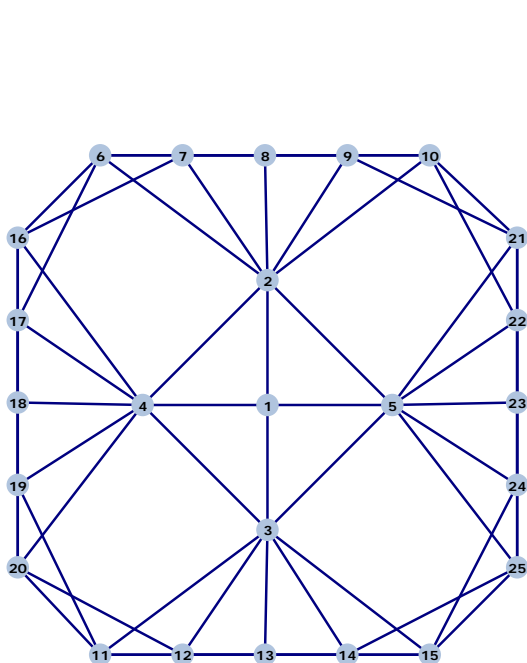
6.2.1 Dejanska topologija in topologija polprirejanja

Izračun topologije polprirejanja, vpete v dejansko topologijo nam omogočajo algoritmi omenjeni v (Poglavje 5). Predstavljamo si preprost povezan graf G , ki predstavlja dejansko topologijo $T_{Dej.}$. Topologijo polprirejanja izračunamo v dveh korakih:

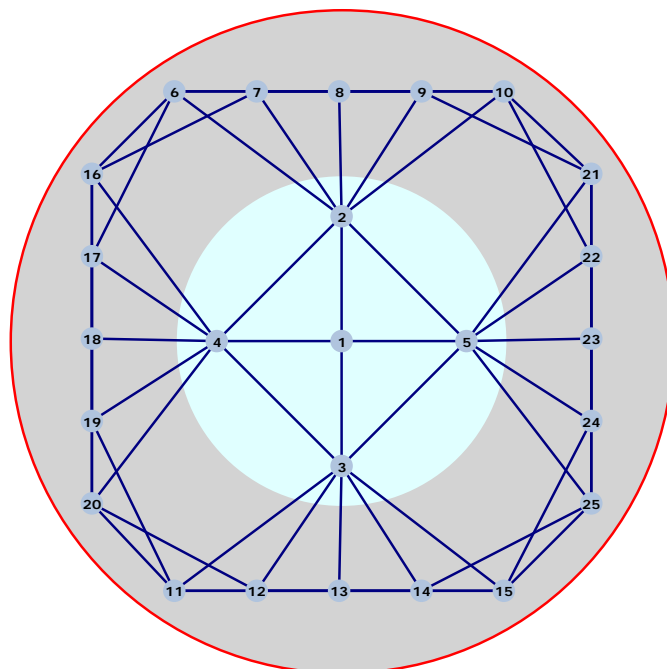
1. Izračun oddaljenosti vsakega vozlišča od bazne postaje (Slika 6.2-3c).
2. Izračun topologije polprirejanja z algoritmi optimalnega polprirejanja (Slika 6.2-3d).

Na (Slika 6.2-3a) vidimo dejansko topologijo primera brezžičnega senzorskega omrežja. Že na pogled vidimo, da ima graf dva nivoja (Slika 6.2-3b). Nivo 1 predstavljajo neposredni sosedi bazne postaje (v tem primeru vozlišče z oznako "1"), neposredni sosedi teh sosedov, ki niso sosedje vozlišča "1" pa predstavljajo drugi nivo.

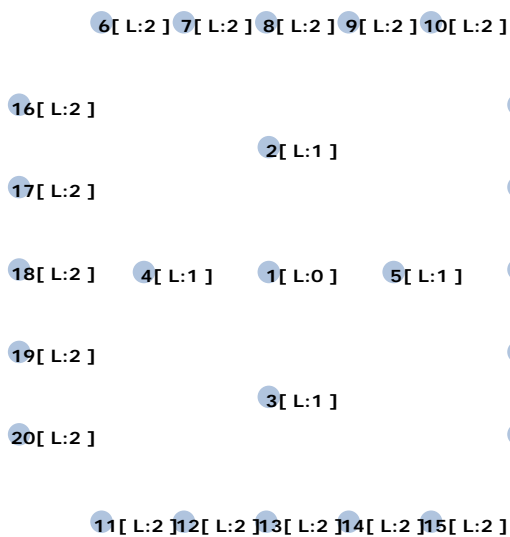
Z izračunom BFS algoritma smo predvidevanja tudi potrdili (Slika 6.2-3c). Algoritem BFS nam omogoča izračun optimalnega polprirejanja (spomnimo, da za algoritme izračuna polprirejanja potrebujemo dvodelne grafe (Poglavje 5.1)).



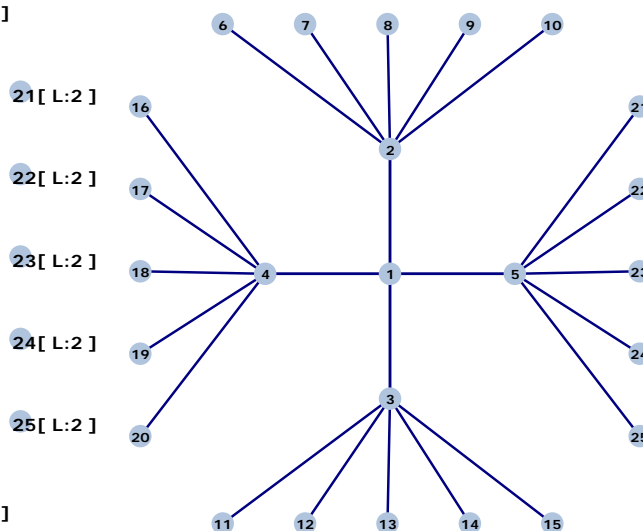
(a) Dejanska topologija T_{Dej} .



(b) Dejanska topologija T_{Dej} , s predvidenimi nivoji grafa



(c) Rezultati algoritma BFS



(d) Rezultat algoritma A_{SM3}

Slika 6.2-3

Računanje topologije polprirejanja.

Po določevanju nivojev grafa z *BFS* algoritmi imamo vse potrebno za izračun optimalnega polprirejanja. Algoritme za optimalno polprirejanje smo v primeru grafa (Slika 6.2-3a) uporabili dvakrat, kot to pojasnjuje tabela 6.2-1.

Tabela 6.2-1

Zagon A_{SM3} algoritma na grafu (Slika 6.2-3a).

Zagon	U vozlišča	V vozlišča
Prvi zagon	$u_{2...5}$	$v_{6...25}$
Drugi zagon	u_1	$v_{2...5}$

(Slika 6.2-3d) nam torej prikazuje rezultat, **topologijo polprirejanja**, katero bomo uporabili za usmerjevalni protokol.

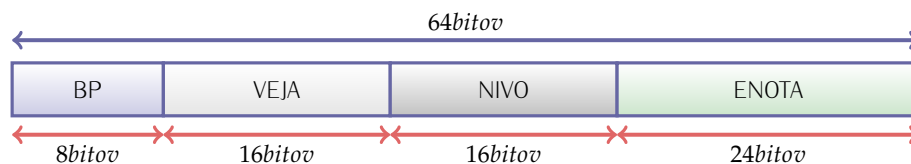
6.3 Identifikacija vozlišča - Naslavljanje

Med vsemi tipi komunikacijskih tehnologij je verjetno najbolj znana IP tehnologija, ki temelji na TCP/IP komunikacijskem modelu. IP v.4 tehnologija uporablja za naslavljanje enot $4 \times 8\text{bitov} = 32\text{bitov}$ dolžino naslova (primer: 192.168.1.1) [4, 13], novejša tehnologija IP v.6 pa za naslavljanje enot postavlja že $8 \times 16\text{bitov} = 128\text{bitov}$ naslovov (primer: 2001:0db8:0:0:0:0:1428:57ab) [13, 21, 47]. Večina stikalnih naprav sicer pretvarja IP naslove v strojne naslove naprav (MAC - ARP tabele) vendar to ne vpliva na osnovno filozofijo naslavljanja. Tudi naprave, ki so del 802.11x standardov (802.11a ... 802.11g) uporabljajo IP temelječe naslavljanje.

Zelo zanimiv način naslavljanja podpira področno vodilo CAN (Controller–area network). Pri vodilu niso naslovljena vozlišča pač pa sporočila [29, 40]. Vodilo je označeno kot realno–časovno zato, ker podpira arbitražni dostop do vodila glede na naslov sporočila (višje prioriteta sporočila imajo prednost) [40, 106].

Najbolj sorodna tehnologija disertaciji je IEEE 802.15.4, ki podpira 64 bitno naslovno polje (2^{64} naprav) vendar je to polje lahko dejansko manjše (odvisno od tipa naslavljanja) [44]. IEEE 802.15.4 podpira dva tipa naslavljanja: naslavljanje znotraj enega PAN (Personal Area Network) omrežja, kjer se uporabljajo kratki naslovi (16bitov) ter naslavljanje med enotami različnih PAN omrežij, kjer se uporabljajo dolgi naslovi vrednosti 64bitov [34, 97].

Za naslavljanje v predlaganem protokolu prav tako predlagamo velikost naslovnega polja vrednosti 64bitov vendar le to razdelimo na sklope, kot to prikazuje (Slika 6.3-1).



Slika 6.3-1

Naslovna struktura predlaganega algoritma.

Naslovno polje je razdeljeno na naslednji način (Slika 6.3-1):

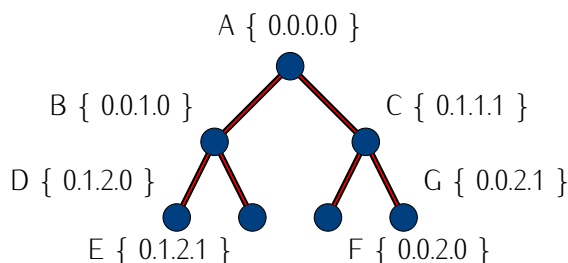
BP : določa naslov bazne postaje (maksimalno 2^8 baznih postaj).

VEJA : določa naslov primarne veje drevesa polprirejanja (maksimalno 2^{16} primarnih vej).

NIVO : določa naslov nivoja v veji (maksimalno 2^{16} nivojev).

ENOTA : določa naslov enote v nivoju (maksimalno 2^{24} enot na nivoju).

Poglejmo še primer naslavljanja na topologiji drevesa (Slika 6.3-2):



Slika 6.3-2

Drevo polprirejanja z naslovi.

Iz (Slika 6.3-2) lahko razberemo, da se recimo enota *E* nahaja v prvi bazni postaji ($bp = 0$), prvi veji ($veja = 0$), na drugem nivoju ($nivo = 2$), naslov vozlišča pa je $enota = 1$. Na podlagi predlaganega načina naslavljanja je mogoče iz naslova enote hitro ugotoviti njene topologijske lastnosti.

6.4 Delovanje protokola

Pri predlaganem protokolu promet razdelimo na 3 podsklope:

1. Promet, katerega prejemnik je bazna postaja (večina prometa).
2. Promet, katerega prejemnik je navadna senzorska enota, ki se nahaja v isti veji kot pošiljatelj.
3. Promet, katerega prejemnik je navadna senzorska enota, zunaj iste veje.

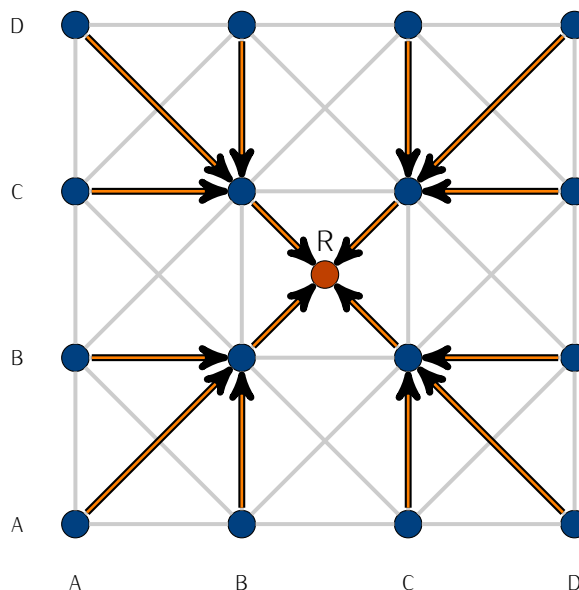
Za usmerjanje prometa predlagamo uporabo dveh algoritmov:

- **Predlagan protokol: SPaRCRoute protokola**, ki ga bomo uporabili za prva dva podsklopa prometnega tipa.
- **ESR (EYES Source Routing Protocol)** [120, 123], predlagan dinamični usmerjevalni protokol skupine EYES (Evropski raziskovalni projekt) [1]. Uporabimo ga za tretji podsklop prometnega tipa.

6.4.1 SPaRCRoute

Za lažjo predstavo definirajmo omrežni graf in izračunajmo optimalno polprirejanje iz vozlišča *R* (Slika 6.4-1). Že na prvi pogled vidimo, da je faktor uravnoteženosti v omenjenem primeru enak $\theta = 1$.

Delovanje usmerjevalnega protokola za pošiljanje podatkov do bazne postaje je izjemno preprosto. Enota po potrebi podatke združi in pošlje naprej nadrejeni enoti.



Slika 6.4-1
Shema delovanja SPaRCRoute.

Na primer: sporočilo iz senzorske enote AD bo za dostavo potrebovalo dve etapi. V prvi etapi bo sporočilo poslano enoti BC , v drugi etapi pa bazni postaji R . Podobno velja za enoto BD ($BD \rightarrow BC \rightarrow R$) (Slika 6.4-1).

Drugi način pošiljanja podatkov, kjer naslovnik ni bazna postaja, pač pa senzorska enota v osnovni veji deluje podobno. Sporočilo je poslano etapno do vozlišča, katerega v veji se nahaja naslovljena postaja. Omenjeno vozlišče potem pošlje sporočilo etapno, do sprejemnika (primer: pošiljatelj: BA , prejemnik: AB , etape pošiljanja $BA \rightarrow BB \rightarrow AB$). Omenjen način pošiljanja podatkov je mogoč zaradi smiselnega naslavljanja vozlišč (Poglavje 6.3).

6.4.2 ESR (EYES Source Routing Protocol)

Za dopolnitev SPaRCRoute protokola smo izbrali ESR (EYES Source Routing Protocol) [120, 123]. Kljub predpostavki, da je komunikacijskega prometa med ključnimi vozlišči izjemno malo smo upoštevali tudi to možnost. So-obstoj dveh usmerjevalnih algoritmov je relativno preprosto izvedljiv saj algoritma ne vplivata drug na drugega, saj smo v protokolu predvideli posebne komunikacijske okvirja (znotraj IEEE 802.15.4 podatkovnih okvirjev [2]), ki podpirajo tovrstno komunikacijo.

Izbira dinamičnega protokola usmerjanja vsekakor ni lahka, saj obstaja ogromno dinamičnih usmerjevalnih protokolov. Metrika, ki smo jo uporabili za izbiro je vsebovala parametre: zakasnitveni časi, overhead prometa, prepustnost omrežja ter energijska učinkovitost. Med vsemi znanimi protokoli smo v ožji izbor postavili tri:

- **DSR** usmerjevalni algoritem, poznan predvsem iz brezžičnih IP povezav [56, 57].
- **AODV** usmerjevalni algoritem, med drugim uporabljen tudi v ZigBee rešitvah [36, 65, 96].
- **ESR** usmerjevalni algoritem, predstavljen kot rezultat evropskega projekta, prirejen posebej za BSO [120, 123].

DSR in AODV, kot najbolj znana kandidata je v magistrskem delu obdelal Akbar [8]. V svojih ugotovitvah je povzel, da je AODV usmerjevalni algoritem bolj primeren za uporabo v BSO kot DSR, saj AODV omogoča večji prepust omrežja, in manjše zakasnitve kot DSR, generira pa malce večji overhead prometa [8].

Naravna nezdružljivost DSR in AODV usmerjevalnih protokolov je bila osnova za predlog alternativnega algoritma usmerjanja imenovanega ESR [120]. Algoritem se je po testiranju izkazal za manj energijsko potrošnega, hitrejšega in z manj overhead-a kot DSR ali AODV. Propustnost omrežja pa je bila za okoli 10 % večja kot pri DSR algoritmu.

Protokol ESR je zasnovan na podobnem principu kot DSR ali AODV: ko je potrebno poslati podatke senzorska enota omrežje preplavi z zahtevo po iskanju poti. Iz odgovorov, ki jih senzorska enota sprejme od omrežja le locira najbolj optimalno pot in po njej pošlje podatke do naslovnika. Zaradi redkega prometa tipa: "senzor – senzor ki ji v isti veji", senzor vzdržuje podatke o le par zadnjih uporabljenih poteh, zato kompleksnega algoritma za vzdrževanje poti ne potrebujemo.

Analiza SPaRCRoute protokola

V omenjenem poglavju bomo analizirali in ovrednotili originalno predlagan del SPaRCRoute protokola.

Analizo in vrednotenje protokola bomo opravili v OPNET simulacijskem orodju. Model brezžičnega senzorskega vozlišča temelji na izdelanem prototipu vozlišča SPaRCMosquito, ki ga bomo na kratko predstavili v poglavju (Poglavje B.1) [62]. Z uporabo omenjenih modelov bomo sestavili model brezžičnega senzorskega omrežja in na njem preizkusili delovanje SPaRCRoute protokola.

7.1 Model vozlišča in omrežja

Za model opisan v nadaljevanju bomo vedno uporabili naslednje simulacijske parametre (če jih bomo spremenili bomo to posebej izpostavili):

Radio : MRF24J40.

CPU : LPC2138.

Napajanje : 2 AA bateriji.

Kapaciteta baterije : 2700 mAh.

MAC okvir : IEEE 802.15.4 okvir.

Hitrost prenosa podatkov : 250 kbit/s.

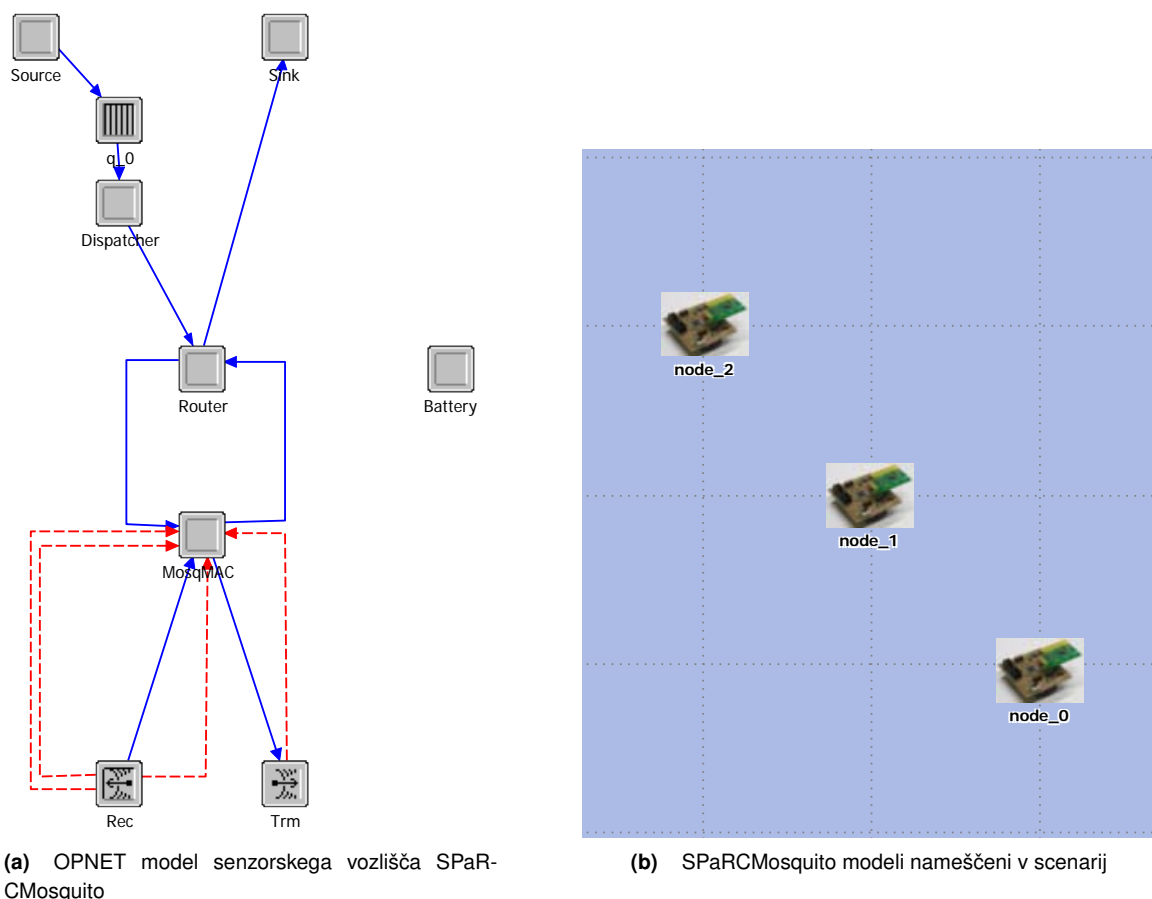
7.1.1 Model senzorskega vozlišča SPaRCMosquito

Za referenco simulacijskega modela smo uporabili v laboratoriju razvit prototip senzorskega vozlišča imenovanega SPaRCMosquito (Poglavje B.2) [86].

Model senzorskega vozlišča smo zasnovali po referenčnem ISO/OSI modelu in zajema naslednje plasti (Slika 7.1-1a):

Fizična plast : Sem spadata sprejmi in oddajni modul (*Rec*, *Trm*).

Podatkovno povezovalna plast : kamor spadajo protokoli MAC.



Slika 7.1-1

OPNET model in scenarij.

Omrežna plast : Kjer se nahaja usmerjevalni proces *Router*.

Aplikacijska plast : Kjer se nahajata procesa generiranja prometa (*Source*) in ponora prometa (*Sink*), ter razdeljevalnik (*Dispatcher*), ki skrbi za dodeljevanje naslovnika (v večini primerov je to bazna postaja).

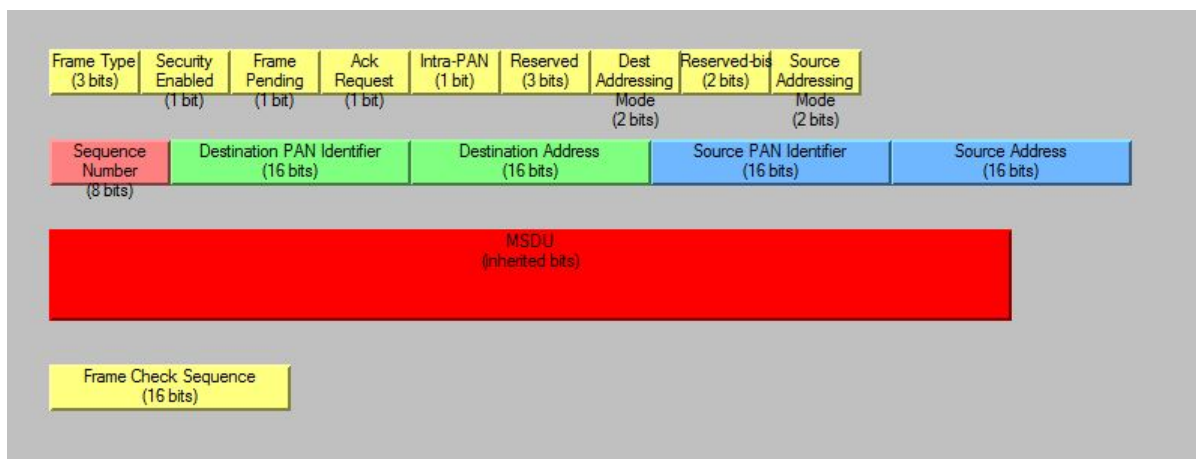
Druge plasti (transportna plast, plast seje in predstavitevna plast) niso implementirane v model.

Posebni proces predstavlja proces baterije (*Battery*), ki ne spada v nobeno plast ISO/OSI referenčnega modela. Proces baterije služi analizi enerjske porabe brezžičnega senzorskega vozlišča.

Model za komunikacijo uporablja nosilno frekvenco **2,4 GHz**, ter podpira 16 kanalov. Celotna fizična plast je implementirana po referenci IEEE 802.15.4 standarda [2].

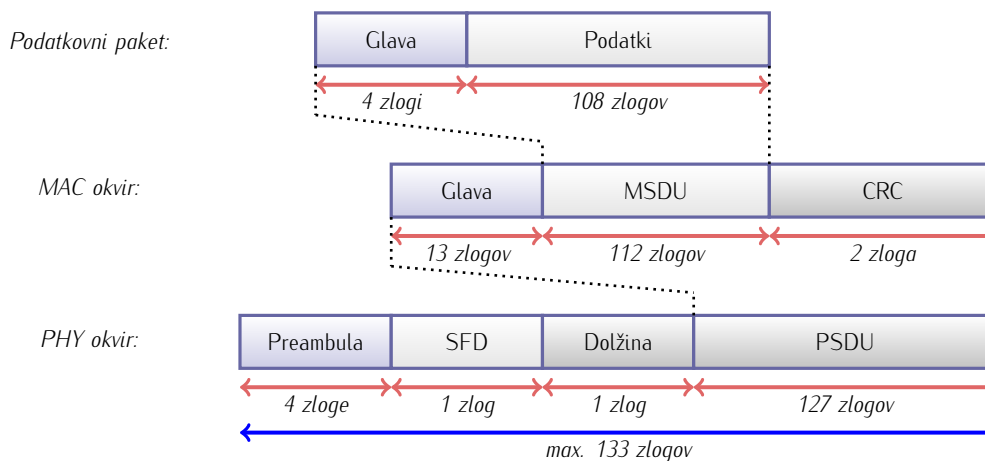
Dolžina podatkovnega paketa znaša do 127 zlogov (spremenljiva podatkovna vsebina – MSDU) (Slika 7.1-2), IEEE802.15.4 okvir pa je maksimalne dolžine 133 zlogov.

V naših simulacijah bomo uporabili konstantno dolge podatkovne okvirje dolžine 112 zlogov. Skupaj z MAC naslovnimi in kontrolnimi zlogi (13 + 2) znaša skupna velikost podatkovnega MAC okvirja 127 zlogov (Slika 7.1-3).



Slika 7.1-2

Osnovni IEEE802.15.4 paket.



Slika 7.1-3

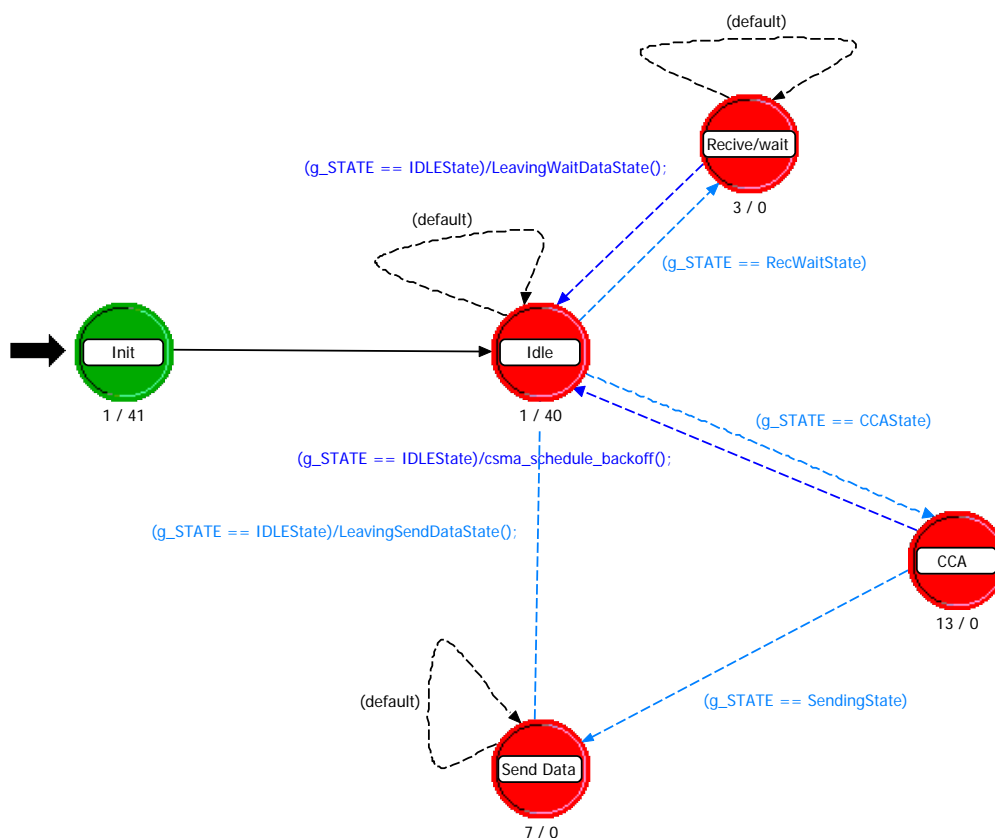
Vpenjanje podatkov v okvir.

Delovanje modela senzorskega vozlišča SPaRCMosquito

Na kratko bi lahko delovanje modela opisali takole: Proces *Source* v modelu generira promet in tega pošilja v čakalno vrsto q_0 . Čakalna vrsta podatke pošlje naprej do *Dispatcher* procesa, ki podatke vpne v paket in izpolni naslovno polje paketa. Podatkovni paket sprejme *Router* usmerjevalnik, ki glede na svoj protokol določi naslednjo etapo sporočila. Paket potuje naprej do *MosqMAC* procesa, ovkiri paket in le tega pošlje naprej na radijski modul *Trm*.

Pri razvoju modela smo veliko pozornosti posvetili *MosqMAC* procesu, saj so rezultati v veliki meri odvisni prav od tega procesa.

SPaRCMosquitoModel uporablja ne vztrajnostni CSMA – CA model komunikacije z vgrajenimi RTS/CTS kontrolnimi paketi [94], kot to predvideva že MACA protokol [63]. Stanje radijskega modula definiramo v 3 stanjih: Oddajanje, sprejemanje in spanje. V simulacijah sklepamo, da je radijski modul popoln, kar pa seveda v praksi ne drži. Tako posplošimo, da je modul v stanju spanja vedno, ko ne oddaja ali sprejema podatkov. Med oddajanjem podatkov modul *MosqMAC* pošlje sporočilo procesu *Battery* za vsak poslan paket. Iz poslanega paketa proces *Battery* izračuna



Slika 7.1-4

MosqMAC proces modela SPaRCMosquitoModel (Slika 7.1-1a).

čas trajanja paketa in potrebno energijo za pošiljanje.

Sprejemanje paketa ni tako enostavno, saj se v primeru trka sporočilo ne prenese iz radijskega modula v *MosqMAC* proces. Za spremljanje stanja sprejemnega radijskega modula tako uporabljamo lokalne statistike "zasedeno" (ang.:bussy), ki nam sporočajo kdaj je modul v stanju branja in kdaj ne.

Zaradi delitve radijskega modula na oddajni in sprejemni del (*Rec* in *Trm*) sprejemni modul sprejema podatke tudi takrat, ko jih oddaja oddajni modul v istem modelu. Tudi tukaj uporabljamo lokalne statistike "zasedeno", da spremljamo stanje radia.

Energijo potrošeno na sprejemnem modulu torej določimo tako, da merimo čas "zasedenosti" sprejemnika, oddajnik pa mora biti ta čas v stanju "prost". Tako se izognemo prisluhom samih sebe (ang.:Echo). Za vsako spremembo stanja podatke pošiljamo na modul *Battery*, ki na podlagi časa zasedenosti sprejemnega modula izračuna energijo potrošeno za poslušanje na kanalu.

Lastnosti modela senzorskega vozlišča SPaRCMosquito

Model brezžičnega senzorskega omrežja temelji na podatkih o strojni opremi procesorja LPC2138 (NXP Arm7 arhitektura) ter MRF24J40 2,4 GHz radijskem modulu. Radijski modul MRF24J40 pozna 3 osnovna stanja in temu primerna je tudi energetska poraba:

1. stanje mirovanja (ang.:sleep): $2\ \mu\text{A}$.
2. stanje oddajanja – TX: $23\ \text{mA}$.
3. stanje sprejemanja – RX: $19\ \text{mA}$.

Za procesor trenutno sklepamo, da se vse skozi nahaja v enem stanju, zato je njegova energijska poraba konstantna ($50\ \text{mA}$).

Tabela 7.1-1

Energijske in časovne razmere pošiljanja enega paketa (teoretično).

Naziv	Velikost (bitov)	Velikost IEEE okvirja (bitov)	Čas paketa ms	Energija TX mWs	Energija RX mWs
RTS	16	120	0.48	0.0265	0.0219
CTS	16	120	0.48	0.0265	0.0219
DATA	912	1016	4.064	0.2243	0.1853
ACK	17	121	0.484	0.0267	0.0221
Σ	961	1377	5.508	0.3040	0.2515



(a) Scenarij za izračun energijskih razmer enega paketa

```

OPNET Simulation Debugger
Type 'help' for Command Summary

ODB> continue
Node: [node_2] with ID = 2 is online...
CPU Current Sleep: 2.000000e-006
On TX Packet: 5, Bits: 120, Energy consumed: 0.000026, Time needed: 0.000480
Path Lost: -106.485978
Node: [node_0] with ID = 0 is online...
CPU Current Sleep: 2.000000e-006
On RX Packet: 5, Bits: 120, Energy consumed: 0.000022, Time needed: 0.000480
On TX Packet: 6, Bits: 120, Energy consumed: 0.000026, Time needed: 0.000480
On RX Packet: 6, Bits: 120, Energy consumed: 0.000022, Time needed: 0.000480
On TX Packet: 1, Bits: 1016, Energy consumed: 0.000224, Time needed: 0.004064
On RX Packet: 1, Bits: 1016, Energy consumed: 0.000185, Time needed: 0.004064
On TX Packet: 2, Bits: 121, Energy consumed: 0.000027, Time needed: 0.000484
On RX Packet: 2, Bits: 121, Energy consumed: 0.000022, Time needed: 0.000484

```

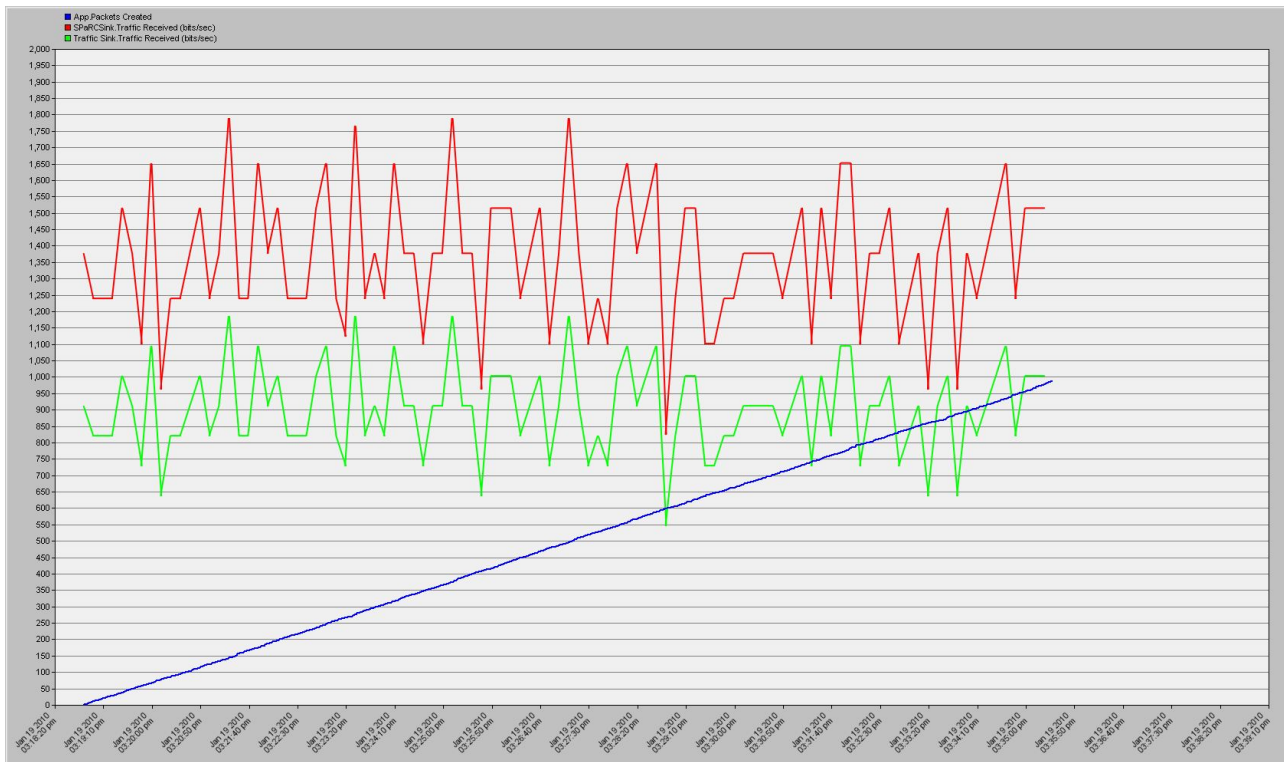
(b) Izsek iz OPNET debugger-ja (energijske in časovne razmere za en poslan paket)

Iz slike (Slika 7.1-5b) je razvidno, da so teoretični izračuni (tabela 7.1-1) in rezultati OPNET simulatorja dovolj podobi. Sklepamo lahko torej, da simulator za model vozlišča pravilno izračunava porabljeno energijo vozlišča.

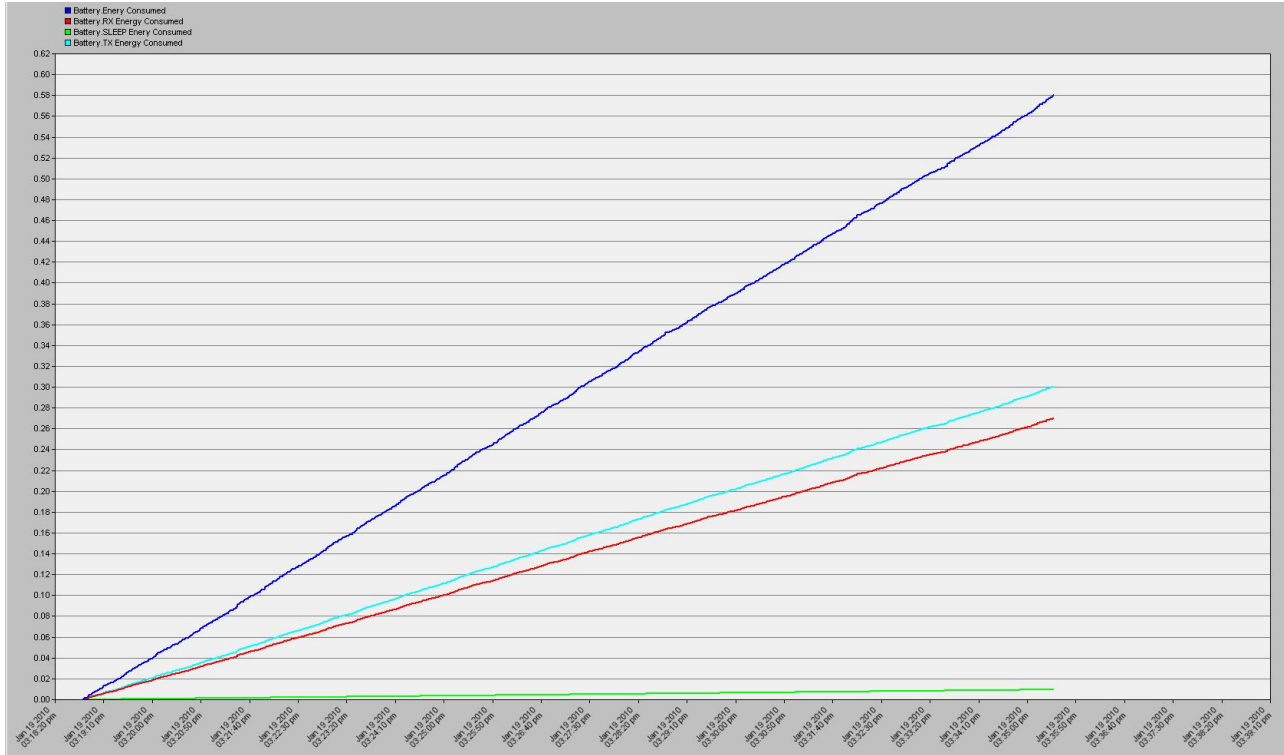
Za primer smo ustvarili scenarij z okoli 980 poslanih paketi (Slika 7.1-5) od enote "node_1" do enote "node_0" (Slika 7.1-5a). Poleg števila poslanih paketov (modra barva) vidimo še povprečno število bitov po sekundi uporabnega prometa (zeleno barva) in skupnega prometa (rdeča barva), ki zajema RTS,CTS,DATA in ACK sporočila.

V scenariju smo spremljali tudi druge parametre:

- Celotna potrošena energija na omrežju (Battery.Energy Consumed) – temno modra barva.
- Energijo pri oddaji paketov (Battery.TX Energy Consumed) – svetlo modra barva.
- Energijo pri sprejemu paketov (Battery.RX Energy Consumed) – rdeča barva.



Slika 7.1-5
Graf ustvarjenih paketov.



Slika 7.1-6
Graf energijske porabe za ustvarjene pakete.

- Energijo pri neaktivnem stanju vozlišča (Battery.SLEEP Energy Consumed) – zelena barva.

Iz grafa je mogoče razbrati, da je količina potrošene energije pri sprejemu in oddaji paketa skoraj enaka, dosti nižja pa je potrošena energija pri neaktivnem stanju vozlišča. Vsota vseh treh energij: oddaje, sprejema in neaktivnosti (Battery.TX Energy Consumed, Battery.RX Energy Consumed, Battery.SLEEP Energy Consumed) je v vsaki x točki seveda enaka skupni potrošeni energiji (Battery.Energy Consumed).

Če analiziramo skupno porabo energije na omrežju in energijo, ki jo posebej potrošita enoti "node_0" in "node_1" (Slika 7.1-7) lahko ugotovimo, da enota "node_1" potroši več energije kot enota "node_0", kar je smiselno, saj enota "node_1" podatke pošilja, "node_1" pa samo sprejema.

Zaradi uporabe RTS, CTS in ACK sporočil nastane v prenosu podatkov določena količina krmilnega (režijskega) prometa. V simulacijah smo ugotovili, da je v naših razmerah približno 66 % uporabnega prometa in 34 % krmilnega prometa (Slika 7.1-8). Iz tega lahko ugotovimo, da je za nasičenje kanala v viru generirati promet velikosti 165 000 bit/s.

Ugotovimo lahko tudi, da je izkoriščenost kanala dobra, približno 99 % (Slika 7.1-8).

Če namesto 165 000 bit/s promet generiramo s 250 000 bit/s ugotovimo, da je kanal še vedno nasičen vendar MAC nivo zavrže določeno število paketov zaradi preplavitve vrste (ang.:overflow) (Slika 7.1-9). Zelena barva prikazuje število sprejetih paketov, temno modra pa število zavrženih paketov.

7.1.2 Vpliv MAC plasti na rezultate simulacij

Skrito vozlišče

V MAC plast modela SPaRCMosquito so integrirani RTS/CTS kontrolni paketi, ki razrešujejo problem skritega vozlišča (hidden node problem). Za raziskovanje lastnosti bomo uporabili tri scenarije:

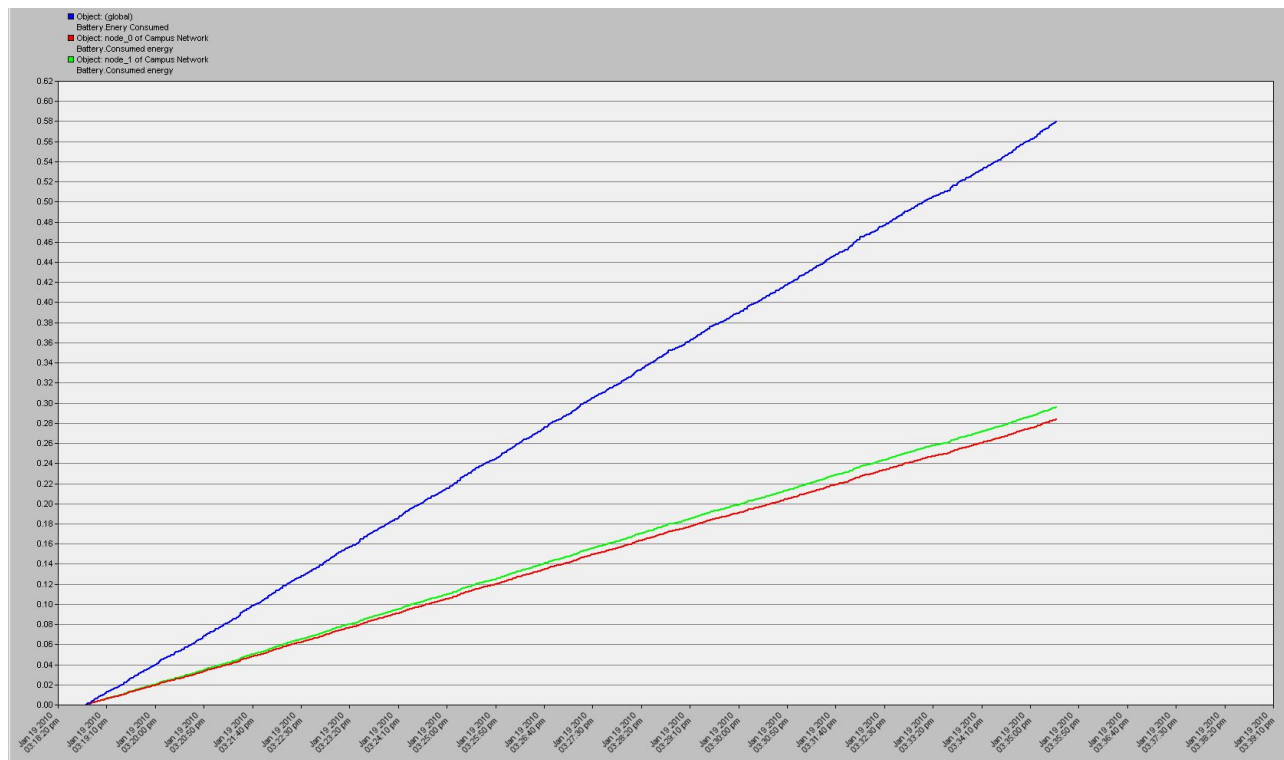
Scenarij 1: 5. vozlišč (bazna postaja in 4. enote, ki generirajo promet) (Slika 7.1-10a).

Scenarij 2: 7. vozlišč (bazna postaja in 6. enot, ki generirajo promet) (Slika 7.1-10b).

Scenarij 3: 9. vozlišč (bazna postaja in 8. enot, ki generirajo promet) (Slika 7.1-10c).

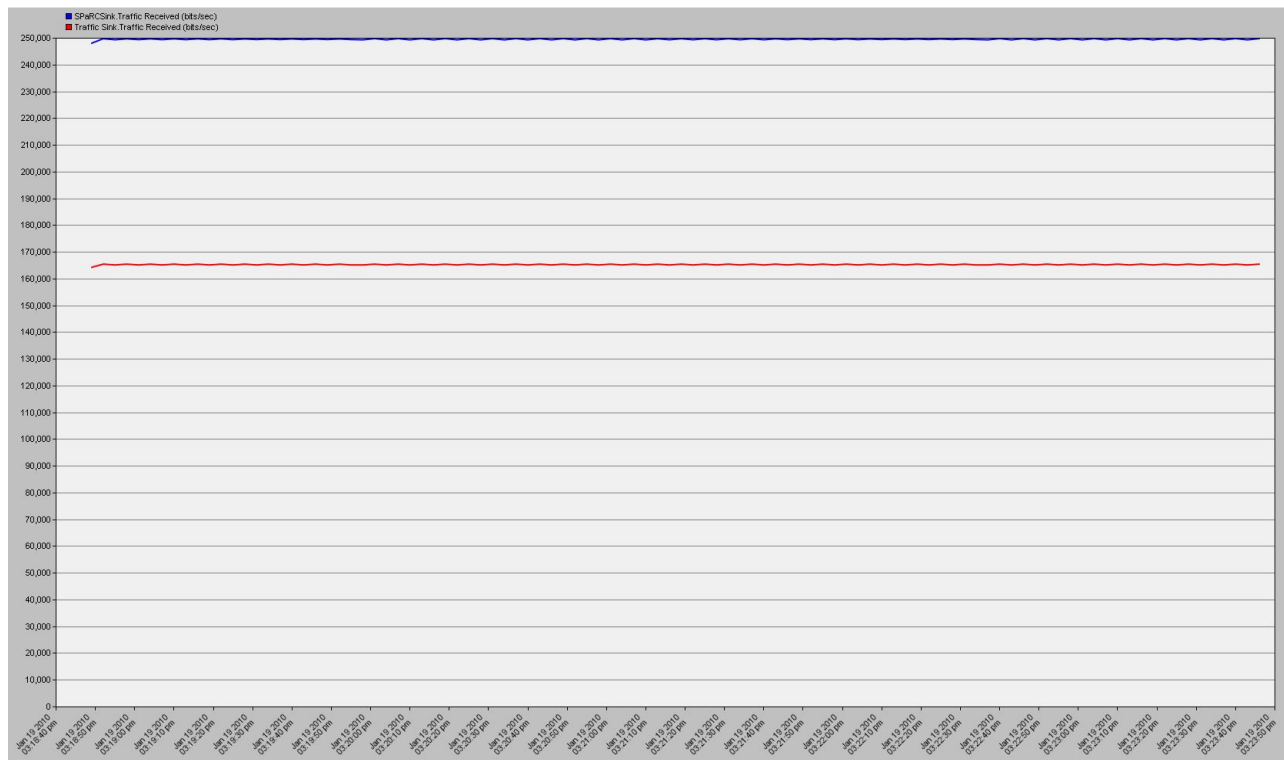
Promet generirajo vse enote, razen bazne postaje. V vsaki simulaciji promet povečamo za 2,5 kbit/s na vseh enotah. Radijska vidljivost enote znaša 75 m, kar pomeni, da je pri prvem scenariju v kolizijski domeni 75 % vozlišč ter 25 % vozlišč skritih, v drugem scenariju je v kolizijski domeni 57 % vozlišč, ter 43 % skritih, v zadnjem, 3. scenariju pa 67 % vozlišč v kolizijski domeni ter 33 % vozlišč skritih.

Za omenjene tri scenarije pogledjmo najprej maksimalni prenos uporabnega prometa pred in po zasičenju kanala (Slika 7.1-11).



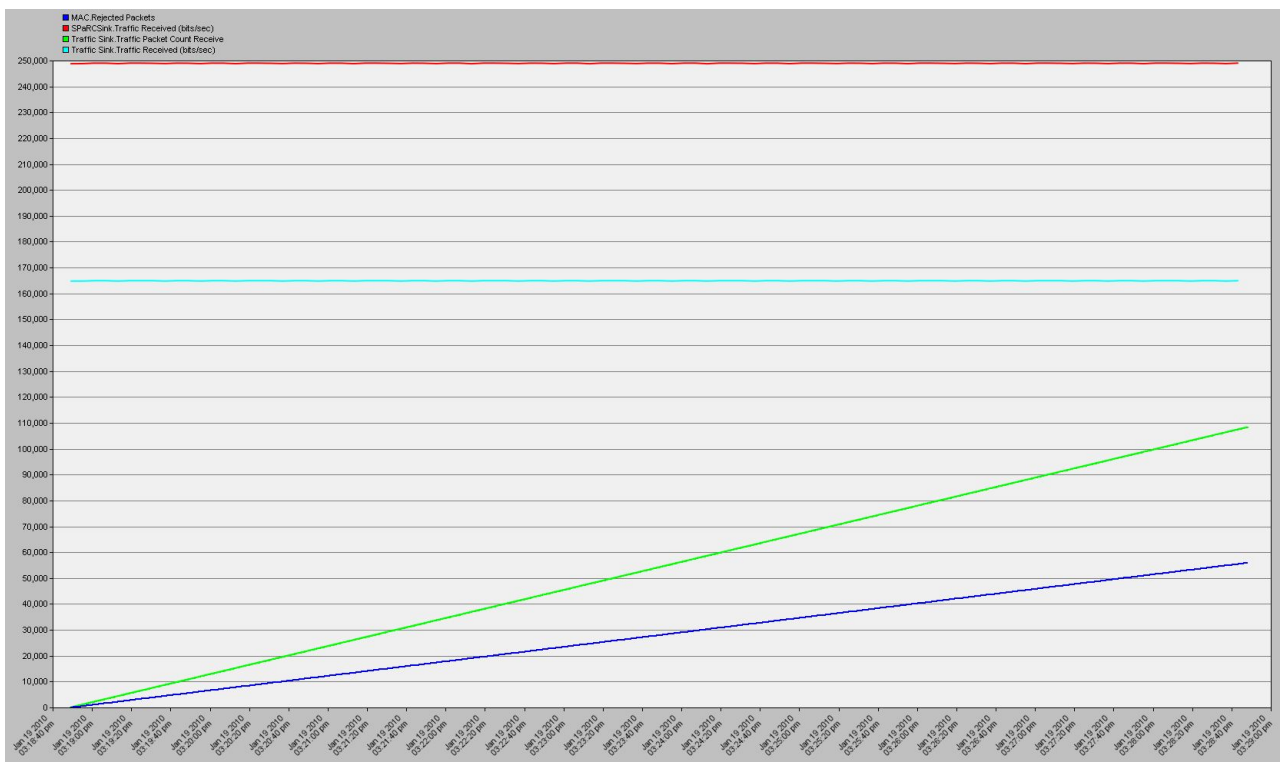
Slika 7.1-7

Graf energijske porabe za ustvarjene pakete: skupna poraba, poraba enote "node_0" in "node_1".

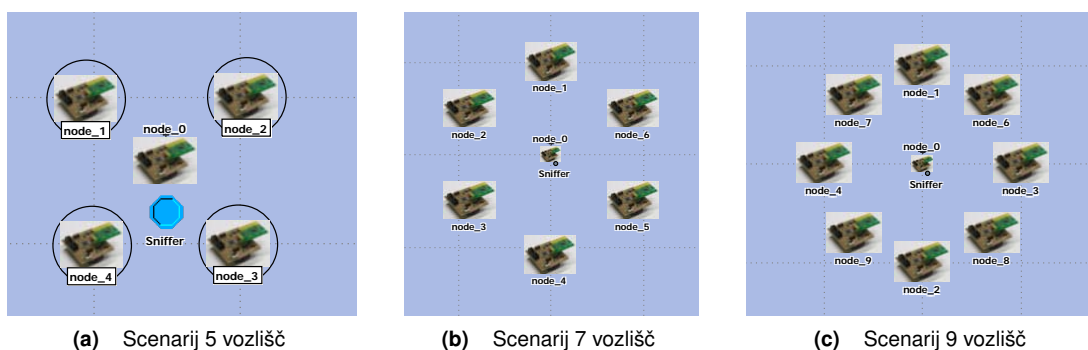


Slika 7.1-8

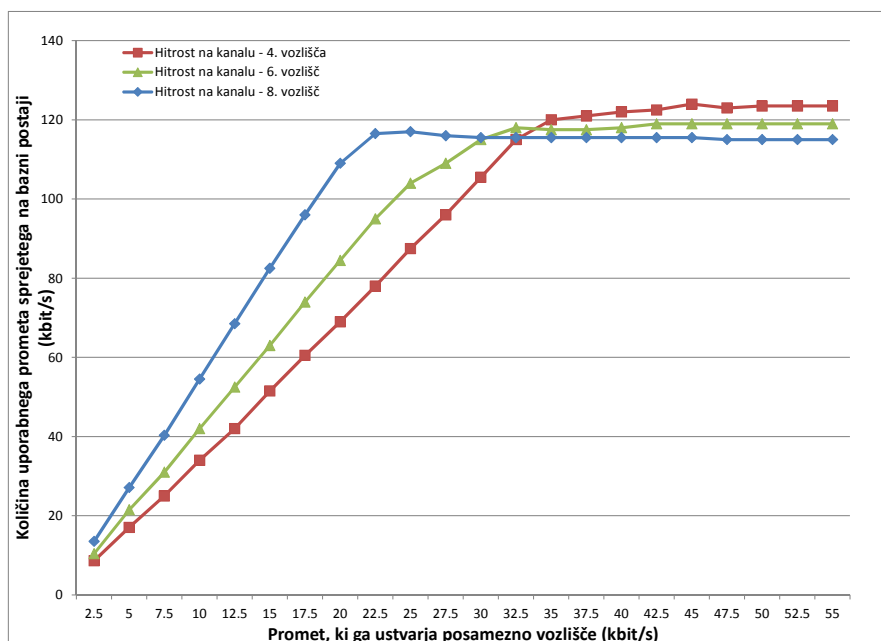
Analiza hitrosti prenosa podatkov za dva vozlišča (Slika 7.1-5a).



Slika 7.1-9
 Analiza hitrosti prenosa podatkov za dva vozlišča in zavrženih paketov (Preplavitev vrste v MAC).



Slika 7.1-10
 Scenarij ocenjevanja razmer zaradi vpliva MAC (RTS/CTS)..



Slika 7.1-11

Analiza količine uporabnega prometa na kanalu.

Rdeča barva predstavlja razmere, ko promet generirajo 4. vozlišča (Slika 7.1-10a). Videti je, da se maksimalni promet, ki se pojavi na kanalu 122 kbit/s, to je tudi točka nasičenja. Z večanjem števila vozlišč točka nasičenja pada (pri 6. vozliščih je ta količina manjša 118 kbit/s), pri 8. vozliščih pa še manjša (115 kbit/s). Sklepamo lahko torej, da bi pri 10. vozliščih nasičenje nastopilo nekje pri ≈ 110 kbit/s... Izkoriščenost kanala (razmerje med poslanim in prejetim prometom) znaša za scenarij 1: 86 %, za scenarij 2: 70 % in scenarij 3: 67 %.

Robne pogoje zasičenja kanala v eni etapi smo spremljali skozi 3 simulacije in sicer: 5 vozlišč (Slika 7.1-10a), 7 vozlišč (Slika 7.1-10b) in 9 vozlišč (Slika 7.1-10c). Vse oddajne postaje uporabljajo enake nastavitve: generiranja prometa s 165 kbit/s ter omogočene RTS/CTS pakete.

Med postajami se nahaja še vozlišče vohljač (ang.: sniffer) (Slika 7.1-10a), ki pa je uporabljen le za spremljanje dejanskega prometa v kanalu.

Tabela 7.1-2

Vpliv MAC plasti: Analiza paketnih razmer.

	5 vozlišč	7 vozlišč	9 vozlišč
Hitrost na kanalu (ves promet) kbit/s	192	181	179
Hitrost na kanalu (uporaben promet) kbit/s	124	121	116
Skupna porabljena energija (mWs)	94,5	98	139
Skupna porabljena energija - stanje oddajanja (mWs)	26,5	24	27,5
Skupna porabljena energija - stanje sprejemanja (mWs)	68	74	111,5

Rezultati izvedenih simulacije smo primerjali s do sedaj znanimi rezultati [58, 68, 69] in ugotovili, da so v skladu z že objavljenimi deli.

Izpostavljeno vozlišče

Razmerje energije, prenosnih hitrosti in uspešnosti paketov na kanalu več etapnega sistema bomo določili s simulacijo v kateri bomo uporabili 7. vozlišč, ki prenašajo sporočila v 6. etapah (Slika 7.1-12). Vozlišča so med sabo oddaljena 50 m, radijska vidljivost znaša 55 m. Točno na sredini med dvema senzorskima vozliščema je nameščen vohljaj (Slika 7.1-12).



Slika 7.1-12
Scenarij testiranja več etap v BSO.

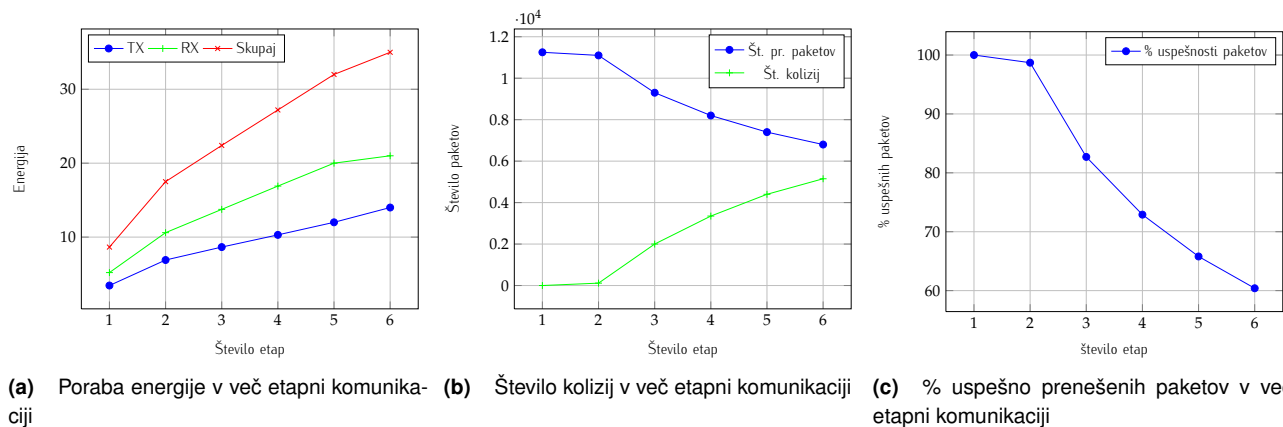
V simulaciji vozlišče generira promet s hitrostjo 16,5 kbit/s, kar znaša $\approx 10\%$ mejne vrednosti nasičenja kanala. Scenarij uporablja CSMA-CA dostop do medija z uporabo RTS/CTS krmilnih paketov. Rezultate simulacij bomo predstavili v tabelarično (tabela 7.1-3 in grafično (Slika 7.1-13)).

Pri analizi energijskega stanja je potrebno upoštevati, da so vedno prisotne vse senzorske enote. V primeru ene etape (promet generira senzorska enota "node_1") energijska analiza zajema tudi prisluhe senzorske enote "node_2", v primeru dveh etap, te prisluhe energijsko upošteva enota "node_3" ... Posledično je energijska poraba zadnje etape nekoliko nižja, kot bi bila, če bi se na scenariju v vrsti, na desni strani enote "node_6" nahajala še ena enota (Slika 7.1-12).

Tabela 7.1-3
Energija, uspešnost paketov in število kolizij v več etapnem sistemu.

	1. etapa	2. etapa	3. etapa	4. etapa	5. etapa	6. etapa
	Kbit/s	Kbit/s	Kbit/s	Kbit/s	Kbit/s	Kbit/s
Energija						
TX	3,45	6,9	8,65	10,3	12	14
RX	5,2	10,6	13,75	16,9	20	21
Skupaj	8,65	17,5	22,4	27,2	32	35
Podatkovni Paketi						
Ustvarjenih	11 250	11 250	11 250	11 250	11 250	11 250
Prejetih	11 250	11 100	9 300	8 200	7 400	6 800
Uspešnost (%)	100	98,7	82,7	72,9	65,8	60,4
Število kolizij	0	115	2 000	3 350	4 400	5 150

Opaziti je mogoče, da je naraščanje skupne porabe energije testnega senzorskega omrežja linearno (če "korigiramo" zadnjo točko zaradi prej omenjene omejitve) (Slika 7.1-13a). Tudi število uspešno poslanih paketov in število kolizij glede na število etap predstavlja bolj ali manj linearno funkcijo (Slika 7.1-13b). Enako seveda, posledično velja tudi za uspešnost prenešenih paketov (Slika 7.1-13c).



Slika 7.1-13

Stanje energije, kolizij in uspešnosti paketov v več etapni komunikaciji.

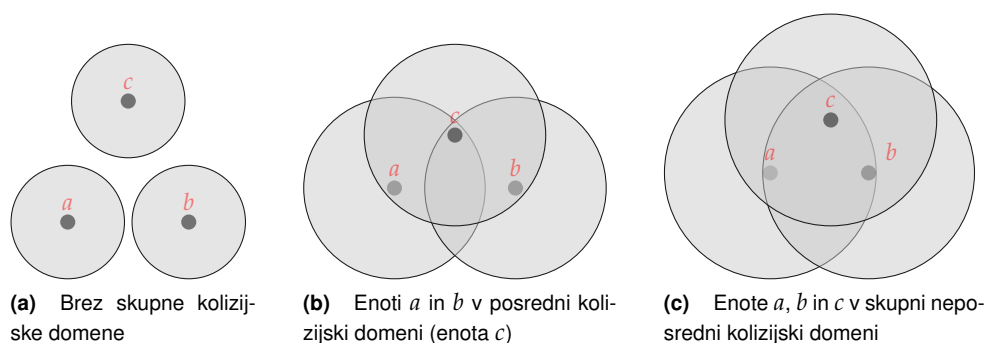
Referenčne slike zajemanja podatkov (1 etapa (Slika C.0-6), 3 etape (Slika C.0-7) in 6 etap (Slika C.0-8)) se nahajajo v poglavju (Poglavje C.0.1).

7.2 Simulacije protokola usmerjanja

7.2.1 Kolizijske domene

V simulacijah bomo pokazali, da je pomembno ali se enoti nahajata v kolizijski domeni ali ne. V ta namen bomo ločili tri pozicije vozlišč:

1. Enote niso v kolizijski domeni (med enotami ni povezav) (Slika 7.2-1a).
2. Enote so v posredni kolizijski domeni (enote so povezane preko drugega vozlišča) (Slika 7.2-1b).
3. Enote so v neposredni kolizijski domeni (enote so povezane neposredno) (Slika 7.2-1c).

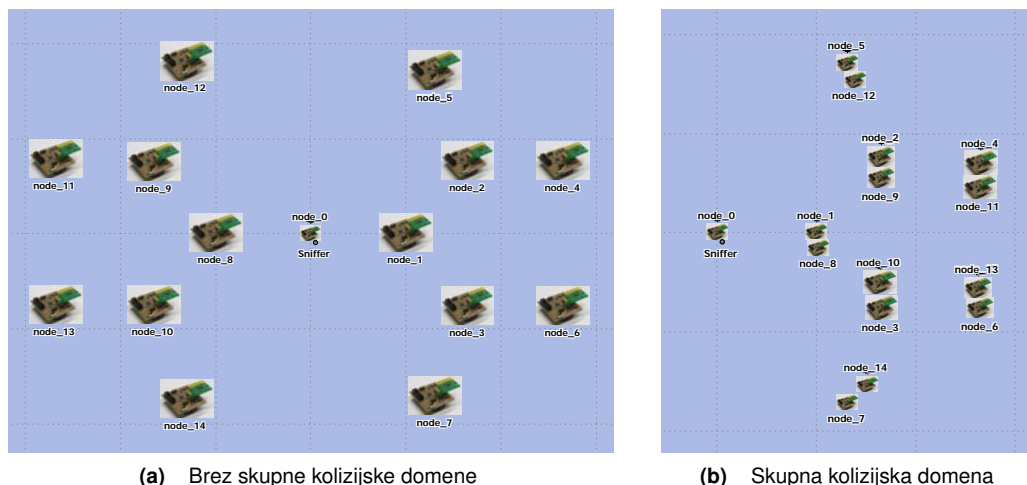


Slika 7.2-1

Kolizijske domene.

V namen testiranja vpliva kolizijskim domen smo pripravili dva scenarija: Scenarij, kjer se veja drevesa ne nahaja v skupni kolizijski domeni (Slika C.0-9) in scenarij, kjer se enote istega nivoja nahajajo v skupni kolizijski domeni (Slika C.0-10).

V v obeh scenarijih smo uporabili enaka simulacijske parametre:



(a) Brez skupne kolizijske domene

(b) Skupna kolizijska domena

Slika 7.2-2

Scenarija testiranja kolizijskih domen.

Radijska vidljivost: 55 m.

Razdalja med vozlišči ki komunicirajo: ≈ 50 m.

Razdalja za ogibanje kolizije: ≈ 70 m.

Generiranje prometa: 1. paket na sekundo z varianco 0,5 s.

Simuliran čas: 3 600 s.

Energijska poraba omrežja v skupni kolizijski domeni je pričakovano večja, kot v omrežju brez skupne kolizijske domene. Pričakovanja potrdijo tudi rezultati simulacij predstavljeni v tabeli 7.2-1. Poraba energije omrežja s skupnimi kolizijskimi domenami je $\approx 62,5\%$ večja od omrežja brez skupnih kolizijskim domen.

Tabela 7.2-1

Energijska analiza scenarija (Slika C.0-9) in (Slika C.0-10).

	Brez skupne domene (Slika C.0-9)	Skupna kolizijska domena (Slika C.0-10)
Oddajanje (TX)	21,5	20
Sprejemanje (RX)	43,5	85
Mirovanje (Sleep)	0,5	0,5
Skupaj	65,5	105,5

Rezultati simulacij paketnih razmer pokažejo, da je število kolizij naraste za več kot 300 %, ko se vozlišča nahajajo v skupni kolizijski domeni (tabela 7.2-2). Uspešnost prve dostave sporočil (brez ponavljanja zaradi kolizij v teku) je v obeh primerih približno enaka (tabela 7.2-2). To pomeni, da večina kolizij nastopi na RTS – CTS izmenjavi kontrolnih paketov, kar je tudi naš cilj!

Iz tabele 7.2-2 je mogoče ugotoviti še, da je čas dostave sporočil v primeru brezkolizijskih domen manjši kot pa, če se postaje nahajajo v skupni kolizijski domeni.

Referenčne slike zajemanja podatkov se nahajajo v poglavju (Poglavje C.0.2) ((Slika C.0-9) in (Slika C.0-10)).

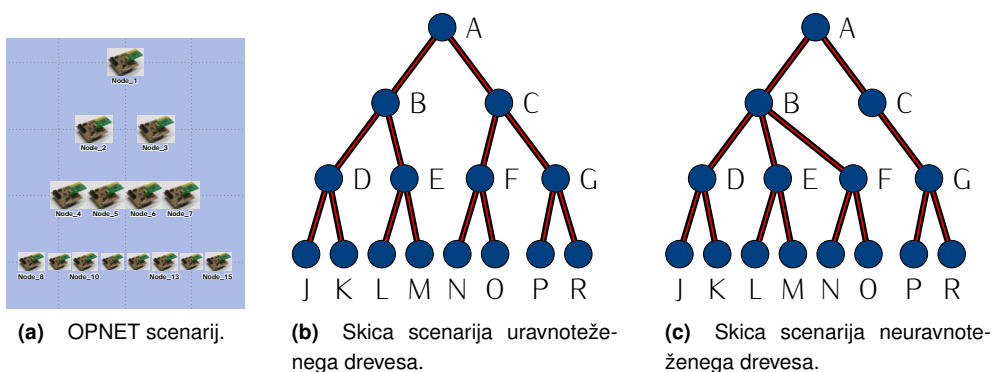
Tabela 7.2-2

Analiza uspešnosti prenosa podatkov scenarijev (Slika C.0-9) in (Slika C.0-10).

	Brez skupne domene (Slika C.0-9)	Skupna kolizijska domena (Slika C.0-10)
Ustvarjeni paketi	49 500	49 500
Sprejeti paketi	29 000	28 000
Kolizije	13 500	42 500
Čas dostave (točka-točka)	0.265	0.34
Uspešnost (%)	58,6%	56,6%

7.2.2 Enakomerna porazdelitev prometa

Za ocenjevanje razmer enakomerno porazdeljenega in neporazdeljenega prometa smo pripravili dva scenarija: (Slika 7.2-3b) in (Slika 7.2-3c).

**Slika 7.2-3**

Scenarij simulacije uravnoveženega in neuravnoveženega drevesa.

V scenariju (Slika 7.2-3b) se nahaja popolnoma uravnoveženo drevo (uravnovežitveni faktorji med nivoji so predstavljeni v tabeli 7.2-3), v scenariju (Slika 7.2-3c) pa je to drevo na prvih dveh nivojih neuravnoveženo.

Tabela 7.2-3Uravnovežitveni faktor θ za scenarija (Slika 7.2-3b) in (Slika 7.2-3c).

	Nivo 1	Nivo 2	Nivo 3
$\theta_{(Slika\ 7.2-3b)}$	1	1	1
$\theta_{(Slika\ 7.2-3c)}$	0,72	0,8	1

Simulirali smo eno uro delovanja senzorskega omrežja obeh scenarije z naslednjimi parametri:

Radijska vidljivost: 60 m.

Razdalja med nivoji vozlišč: 50 m.

Generiranje prometa: 1. paket na sekundo z varianco 0,5 s.

Simuliran čas: 3 600 s.

Iz tabele 7.2-4 je mogoče razbrati, da je uspešnost prenosa paketa brez ponavljanja večja pri scenariju s enakomerno porazdeljenimi vozlišči. Čeprav v tem primeru nastopi več kolizij, te niso tako problematične kot kolizije v drugem scenariju. Uspešnost prenosa paketa v prvem poskusu (brez ponavljanja) je tako $\approx 12,5\%$ večja, kot v scenariju neenakomerne porazdelitve. Tudi povprečni čas dostave paketov je v primeru enakomerne porazdelitve manjši, kar da sklepati, da je protokol uspešnejši.

Tabela 7.2-4

Paketne in časovne razmere za scenarija (Slika 7.2-3b) in (Slika 7.2-3c).

	Št. ustvarjenih paketov	Št. sprejetih paketov (1. poskus)	Št. Kolizij	Uspešnost (%)	Čas
Scenarij 1 (Slika 7.2-3b)	93 500	55 750	115 000	59,6 %	0,125
Scenarij 2 (Slika 7.2-3c)	93 500	44 000	97 000	47 %	0,135

Zaradi večjega števila kolizij je posledično večja tudi porabljena energija v scenariju enakomerne porazdelitve, vendar pa izračun energije na en prenešen bit podatkov pokaže, da je energijsko bolj učinkovit scenarij enakomerne porazdelitve (tabela 7.2-5).

Tabela 7.2-5

Energijske razmere za scenarija (Slika 7.2-3b) in (Slika 7.2-3c).

	En. pošiljanja paketov (TX)	En. sprejemanja paketov (RX)	En. mirovanja (Sleep)	En. Skupaj	Energija na prenešen bit
Scenarij 1 (Slika 7.2-3b)	30,5	178	0,5	209	$3,75 \cdot 10^{-3}$
Scenarij 2 (Slika 7.2-3c)	31	143	1	175	$3,98 \cdot 10^{-3}$

Referenčne slike zajemanja podatkov se nahajajo v poglavju C.0.2 ((Slika C.0-11) in (Slika C.0-12)).

Posamezna vozlišča pri enakomerni porazdelitvi prometa

Tabela 7.2-6

Energijske razmere za scenarija (Slika 7.2-3b) in (Slika 7.2-3c), vozlišča 1., 2. in 3..

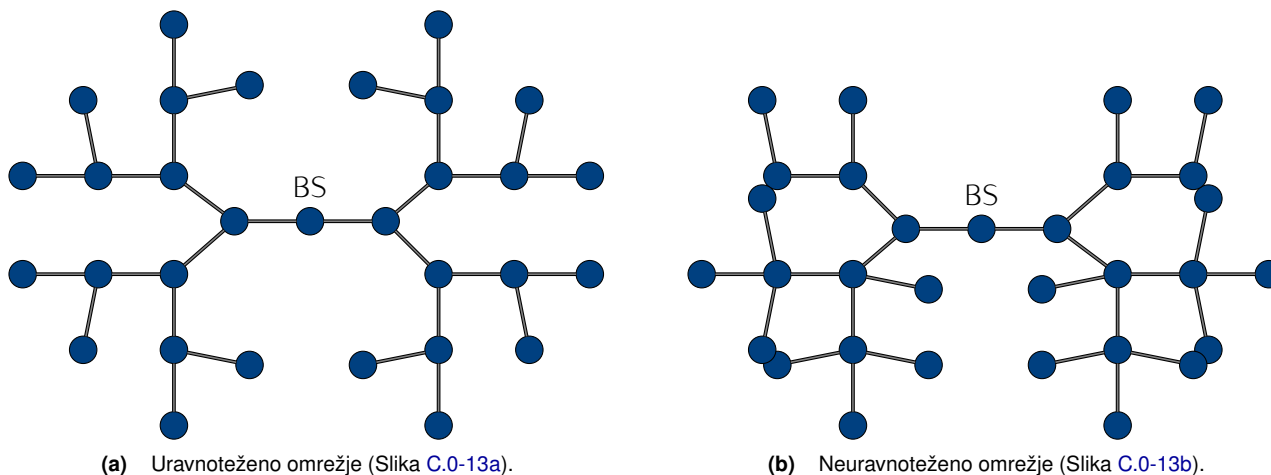
	Vozlišče 1 "Node 1"	Vozlišče 2 "Node 2"	Vozlišče 3 "Node 3"
Scenarij 1 (Slika 7.2-3b)	16,6	22,6	22,6
Scenarij 2 (Slika 7.2-3c)	13	19	16,5

Iz tabele 7.2-6 je razvidno, da je v primeru enakomerne porazdelitve prometa energijska poraba na vozliščih "Node 2" in "Node 3" enaka, v primeru neenakomerne porazdelitve pa je poraba na vozlišču "Node 2" $13,15\%$ večja (s časom naraščajoča) kot na vozlišču "Node 3".

Iz omenjenega sledi, da bo vozlišče "Node 2" veliko prej porabilo vso razpoložljivo energijo kot vozlišče "Node 3", kar pa posledično pomeni, da bodo vse enote, ki so odvisne od vozlišča "Node 2" odrezane od omrežja.

7.2.3 Simulacija dveh manjših senzorskih omrežij

V namen zadnjega testiranja smo v OPNET-u ustvarili dve simulaciji: Simulacijo uravnoteženega omrežja (Slika 7.2-4a) in simulacijo neuravnoteženega omrežja (Slika 7.2-4b).



Slika 7.2-4

Simulacijski scenarij uravnoteženega in ne-uravnoteženega omrežja).

Simulirali smo eno uro (3600 sekund) delovanja senzorskega omrežja. V obeh scenarijih so uporabljeni naslednji parametri:

Radijska vidljivost: 51 m.

Oddaljenost: 50 m.

Generiranje prometa: 2. paketa na sekundo z varianco 0,25 s.

Simuliran čas: 3 600 s.

Poglejmo najprej uravnoteženost v zadnji etapi in predzadnji etapi (tabela 7.2-7).

Tabela 7.2-7

Uravnotežitveni faktor θ za scenarija (Slika 7.2-3b) in (Slika 7.2-3c).

	Nivo 1	Nivo 2	Nivo 3
$\theta_{\text{(Slika 7.2-4a)}}$	1	1	1
$\theta_{\text{(Slika 7.2-4b)}}$	1	0,845	0,758

Rezultate simulacij paketnih razmer predstavljamo v tabeli 7.2-8.

Tabela 7.2-8

Paketne in časovne razmere za scenarija (Slika 7.2-3b) in (Slika 7.2-3c).

	Št. ustvarjenih paketov	Št. sprejetih paketov (1. poskus)	Št. Kolizij	Uspešnost (%)	Čas
Scenarij 1 (Slika 7.2-4a)	200 000	97 500	108 000	48,75 %	0,171
Scenarij 2 (Slika 7.2-4b)	200 000	85 000	135 000	42,5 %	0,183

Iz tabele 7.2-8 je razvidno, da v primeru uravnoveženega scenarija algoritem bolj zanesljiv in hitrejši.

Tabela 7.2-9

Energijske razmere za scenarija (Slika 7.2-3b) in (Slika 7.2-3c).

	En. pošiljanja paketov (TX)	En. sprejemanja paketov (RX)	En. mirovanja (Sleep)	En. Skupaj	Energija na prenešen bit
Scenarij 1 (Slika 7.2-4a)	101	224	0	325	$3,33 \cdot 10^{-3}$
Scenarij 2 (Slika 7.2-4b)	108	271	0	379	$4,46 \cdot 10^{-3}$

Iz tabele 7.2-9 je razvidno, da je uravnovežitev prometa energijsko bolj učinkovita. V primeru simulacij in modela baterije je energijska poraba linearna funkcija, zato je mogoče izračunati smerni koeficient premice in predvidevati kakšna bo energijska poraba čez 3.,4., ali 5. ur ...

Faktor premice energijske porabe neuravnoveženega scenarija (Slika 7.2-4a) je seveda večji kot faktor uravnoveženega scenarija (Slika 7.2-4b), kar pomeni, da bo tudi razlika v energijski porabi s časom delovanja omrežja povečevala.

Podobno je mogoče sklepati tudi za paketne razmere na omrežju.

Referenčne slike zajemanja podatkov se nahajajo v poglavju C.0.2((Slika C.0-13) in (Slika C.0-14)).

Doktorska naloga z naslovom "Prometno uravnoteženi algoritmi za brezžična senzorska omrežja" je rezultat študijskih raziskav na področju brezžičnih senzorskih omrežij prepletenih s teorijo grafov. V nalogi smo si zadali predstaviti nov, preprost algoritem za protokol usmerjanja v brezžičnih senzorskih omrežjih. Algoritem smo testirali in ovrednotili s profesionalnim orodjem OPNET, v katerem smo razvili model senzorskega vozlišča SPaRCMosquito. Med pomembnejše metrike delovanja modela brezžičnega senzorskega omrežja smo šteli: potrošeno energijo celotnega omrežja in izkoristek (energijo na prenešen bit), latentne čase in propustnost omrežja.

V raziskavah, ki smo jih opravili smo ocenili, da je pristop k modeliranju in razvoju novih algoritmov usmerjanja prometa često preveč komunikacijsko ali matematično usmerjen, zato smo predpostavili prednost multidisciplinarnega pristopa: dobro poznavanje teorije komunikacij, nadgrajeno z intenzivno študijo teorije grafov.

Brezžična senzorska omrežja so aplikacijsko odvisna, vse prevečkrat pa jih tudi raziskovalna srenja enači ali prepleta s MANET omrežji. Kot že ime samo pove: "brezžična **senzorska** omrežja", govorimo o omrežjih katerih primarna naloga je zaznavati različne fenomene in sporočanje le teh uporabniku z uporabo več etapnega brezžičnega komunikacijskega sistema. Posledično smo napovedali omejitve BSO: Omejili se bomo na statična brezžična senzorska omrežja, katerih vozlišča ne spreminjajo svoje lege ali pa je intenziteta spreminjanja tako nizka, da je zanemarljiva. Zaradi omenjenega izbranega tipa BSO smo predpostavili, da bomo razvijali **centraliziran** usmerjevalni algoritem, voden iz BP.

Iz dosedanje študije smo lahko sklepali, da je verjetnost uspešnosti dostave paketa v prvem poskusu odvisna predvsem od treh faktorjev:

- števila etap sporočila.
- kolizijskih domen.
- nasičenja kanala.

Sporočila uspešno poslana v prvem poskusu (brez potrebe ponavljanja zaradi kolizij ali sorodnih težav) porabijo najmanj energije in so najhitreje na cilju.

Teoretično lahko vplivamo na vse tri faktorje: zmanjšamo število etap in kolizijskih domen ter preprečimo zasičenje kanala.

Zmanjševanje števila kolizijskih domen je problem MAC protokola, pa še tam je kolizijske domene večkrat nemogoče zmanjšati ali pa jih lahko zmanjšamo le v manjšem obsegu s spremembo lokacije (kolizijske domene so namreč odvisne od postavitve senzorjev, ki pa je lahko čisto naključna), uporabo drugačne tehnologije komunikacije (CDMA...) ali uporabo več kanalov (omejeno število). Tako je jasno, da moramo rešitev iskati v usmerjevalnih algoritmih, ki zadostijo preostalima dvema pogojema: zmanjšanje števila etap (verjetnost uspešnosti dostave paketa v prvem poskusu pada s večanjem števila etap v sistemu, kar smo pokazali s simulacijami v poglavju 7.1.2) od vsakega vozlišča do bazne postaje (ponora) ter preprečiti zasičenje kanala oziroma, enakomerno porazdeliti promet na različnih kanalih.

Algoritmi in metode iskanja, kako enakomerno porazdeliti promet in istočasno doseči minimalno število etap so po našem poznavanju vse NP težki problemi. Posledično smo se zato lotili iskanja rešitve, ki da rezultate v P času. Postavili smo tezo, da je to mogoče doseči z algoritmi BFS in A_{SM} . Omrežje smo modelirali kot povezan graf $G = (V, E)$ (UDG model), v katerem smo za vsa vozlišča določili najkrajšo pot od le teh do BP. Z metodo smo izračunali vpeto drevo, ki ga je potrebno uravnotežiti.

Za uravnotežitev števila vozlišč med starše na vseh nivojih smo uporabili pohitren algoritem optimalnega polpriprejanja. Algoritem polpriprejanja uporabi vpeto drevo in na vseh nivojih uravnoteži drevo. Rezultat je uravnoteženo vpeto drevo. Algoritem smo primerjali še z drugimi podobnimi algoritmi. Ocenjevali smo hitrost izvajanja algoritmov in rezultate le teh. Izkazalo se je, da je predlagan algoritem med vsemi najhitrejši, vsi preizkušeni algoritmi pa vrnejo optimalno polpriprejeno drevo. Prednost algoritma je tudi v hitrem dodajanju in odvzemanu vozlišča ter hitrem določanju alternativnih optimalno polpriprejenih dreves.

Postavlja pa se zanimivo vprašanje: kako algoritem prevesti na problem uteženega drevesa?

Uravnotežitev nivojev smo ocenjevali po metodi Chebyshejeve sume, vendar se postavi vprašanje ali je ocena po Chebyshejevi sumi dovolj natančna in primerna?

Predlog celotnega protokola smo obravnavali v poglavju 6, kjer so definirali način delovanja, naslavljanja in podobne parametre.

V OPNET simulacijskem okolju smo razvili model brezžičnega senzorskega vozlišča SPaRCMosquito. Model radijskega modula smo poskušali približati modulu MRF24J40 (IEEE 802.15.4) z dodatkom RTS/CTS paketov.

Za potrditev uporabnosti protokola smo izbrali par simulacijskih scenarijev in jih simulirali v OPNET okolju. Izkazalo se je, da je predlagan protokol imenovan **SPaRCRoute** doprinese k hitrejši, bolj zanesljivi in manj energijsko potrošnji prenosa podatkov (kot na primer MST).

Po analizi rezultatov in simulacij smo ugotovili tudi, da CSMA-CA z uporabo RTS/CST okvirjev verjetno ni najbolj primeren MAC protokol za SPaRCRoute algoritem zato v nadaljevanju raziskovanja predlagamo izbiro drugega, bolj primernega MAC protokola.

Dodatek

O kandidatu

Kratek CV kandidata:

Kontaktni podatki	Karl Benkič Kumen 56, 2344 Lovrenc na Pohorju e-mail: karl.benkic@uni-mb.si
Osební podatki	Rojen: 16.7.1981 v Mariboru, Slovenija Državljanstvo: Slovensko
Zaposlen	Trenutno zaposlen kot raziskovalec na fakulteti za elektrotehniko, računalništvo in informatiko Maribor.
Trenutna izobrazba	Magister znanosti smer Telekomunikacije
Izobraževanje	
1988 – 1996	Osnovna šola Lovrenc na Pohorju (tedaj OŠ Šcercerjeve brigade).
1996 – 2000	Srednja kovinarska, strojna in metalurška šola Maribor – zaključil z Maturó.
2000 – 2005	Fakulteta za strojništvo, univerze v Mariboru. Študij na smeri Mehatronike.
2005	Zaključil univerzitetni študij z zagovorom diplome: "X-by-wire" pod mentorstvom prof.dr. Ž. Čučej in prof.dr. J. Flašker.
2005	Izbran na razpisu ARRS-a za mladega raziskovalca pod mentorstvom prof.dr. Ž. Čučej.
2005	Vpisal podiplomski študij na UM-FERI smer Elektrotehnika.

2008	<i>Magistriral na fakulteti za elektrotehniko, računalništvo in informatiko Maribor, pod mentorstvom prof.dr. Ž. Čučej in somentorstvom doc.dr. J. Mohorko z nalogo: "Modeliranje CDMA brezžičnih senzorskih omrežij".</i>
Delovne izkušnje	<i>Alchrom d.o.o. LANCom d.o.o. Ultra d.o.o. Piktronik d.o.o. Slovenska vojska.</i>

SPaRCMosquito in SPaRCSoft

B.1 SPaRCMosquito

SPaRCMosquito senzorsko vozlišče je prototip brezžičnega senzorskega vozlišča razvitega v laboratoriju za obdelavo signalov in daljinska vodenja (SPaRC). Prototip vozlišča je namenjen testiranju in ocenjevanju delovanja brezžičnega senzorskega omrežja kot tudi razvoju in testiranju algoritmov ter protokolov za BSO [18, 86].

Druga verzija SPaRCMosquito senzorskega modula je trenutno v razvojni fazi. Osnovne razlike med vozlišči bomo povzeli v tabeli B.1-1:

Tabela B.1-1

SPaRCMosquito verzija 1. in verzija 2..

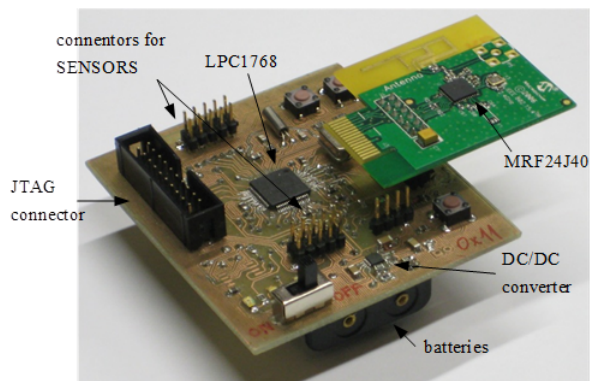
	SPaRCMosquito v.1.	SPaRCMosquito v.2.
Procesor	NXP LPC2138 Arm 7 jedro	LPC1768 Cortex M jedro
Radijska enota	MRF24J40 IEEE 802.15.4	MRF24J40 IEEE 802.15.4

SPaRCMosquito verzija 2. uporablja zmogljivejši a varčnejši procesor. Namesto Arm 7 procesorja sedaj uporabljamo zmogljiv a mnogo varčnejši Cortex M procesor. Spremenili smo tudi izgled. Na žalost nam ni uspelo obdržati "pin kompatibilnosti" verzije 1.

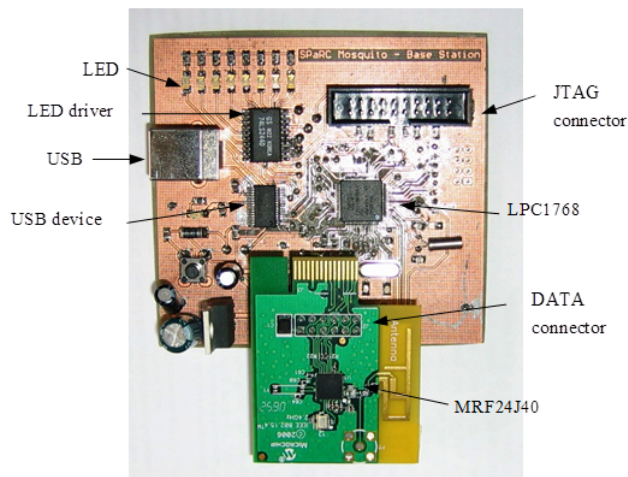
Senzorsko vozlišče omogoča zamenjavo strojne programske opreme na daljavo. Modularna zgradba omogoča uporabo različnih standardnih senzorjev združljivih z AD, SPI, CAN in I2C vodilom ter izvedbo lastnih senzorjev [18, 86].

SPaRCMosquito je trenutno izveden v dveh verzijah: verzija "bazne postaje" (Slika B.1-1b) in "navadnega vozlišča" (Slika B.1-1a).

Bistvena razlika med vozliščem in bazno postajo je v napajanju in priklopu na računalnik. Bazna postaja ne predvideva svojega napajanja, pač pa je napajana iz USB vmesnika na računalniku. Bazna postaja (Slika B.1-1b) je najprej delovala kot splošna bazna postaja (pobiranje podatkov iz omrežja), kasneje pa dopolnili strojno programsko opremo (ang.:Firmware) s katero se BP lahko obnaša kot vohljač (podobno kot Microchip ZENA modul) ali pa kot standardno vozlišče z možnostjo diagnostike omrežja.

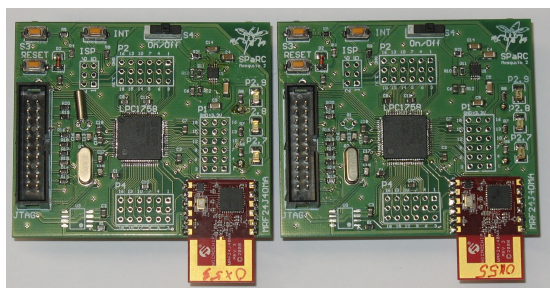


(a) SPaRCMosquito senzorsko vozlišče

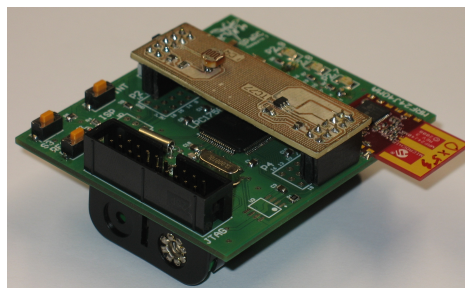


(b) SPaRCMosquito bazna postaja

Verzija 2. (Slika B.1) več ne predvideva ločenega vozlišča in bazne postaje. Priklop bo možen preko vmesnika, ki ga bomo na vozlišče namestili preko I/O razširitvenih modulov.



(a) SPaRCMosquito v.2 senzorski vozlišči



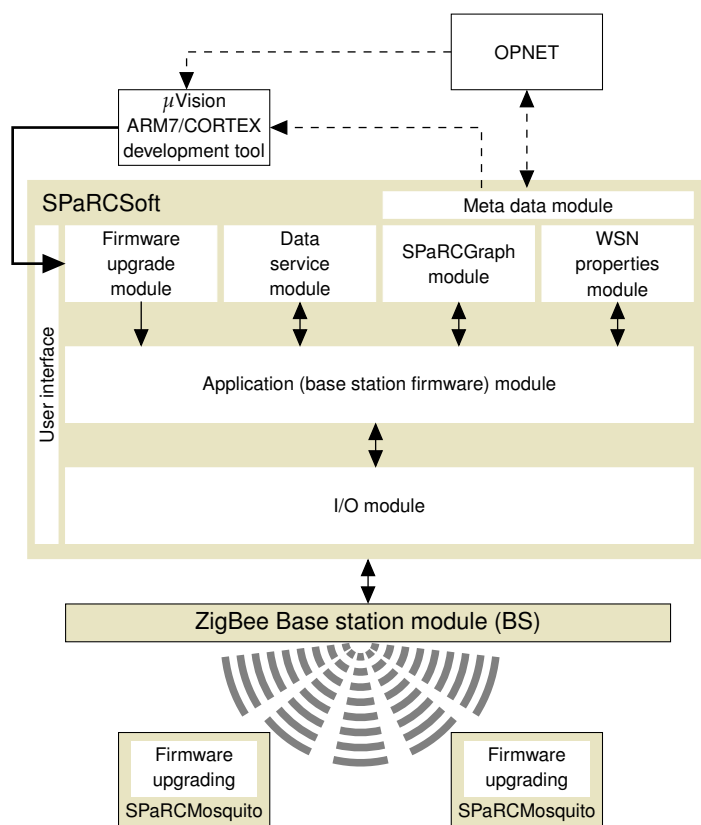
(b) SPaRCMosquito v.2 senzorsko vozlišče

Velika prednost SPaRCMosquito modulov je zamenjava strojne programske opreme na daljavo in zmogljivost procesorja kar omogoča hiter razvoj algoritmov in protokolov.

B.2 SPaRCSoft in druga programska oprema

Programsko opremo senzorskega omrežja SPaRCSeNet (SPaRC Sensor Net) lahko razdelimo na tri večje skupine [19]:

1. Programska oprema za senzorsko vozlišče (ang.:sensor node firmware).
2. Programska oprema bazne postaje (ang.:base station firmware).
3. SPaRCSoft, višje nivojska programska oprema za bazno (postaja / vohljač / vozlišče) in komunikacijski protokol z bazno postajo.



Slika B.2-1
Moduli programa SPaRCSoft.

SPaRCSoft omogoča modeliranje, načrtovanje in ocenjevanje brezžičnih senzorskih omrežij. Programska oprema je grajena modularno, iz več med seboj odvisnih ali neodvisnih modulov (Slika B.2-1). Najpomembnejši moduli so:

I/O module je vmesnik za povezavo na SPaRCMosquito bazno postajo, ki trenutno deluje preko USB virtualnega serijskega vmesnika.

Application module vsebuje višje programske opremo za bazno postajo. Preko aplikacijskega modula je mogoče bazni postaji nastavljati parametre, način delovanja in podobno.

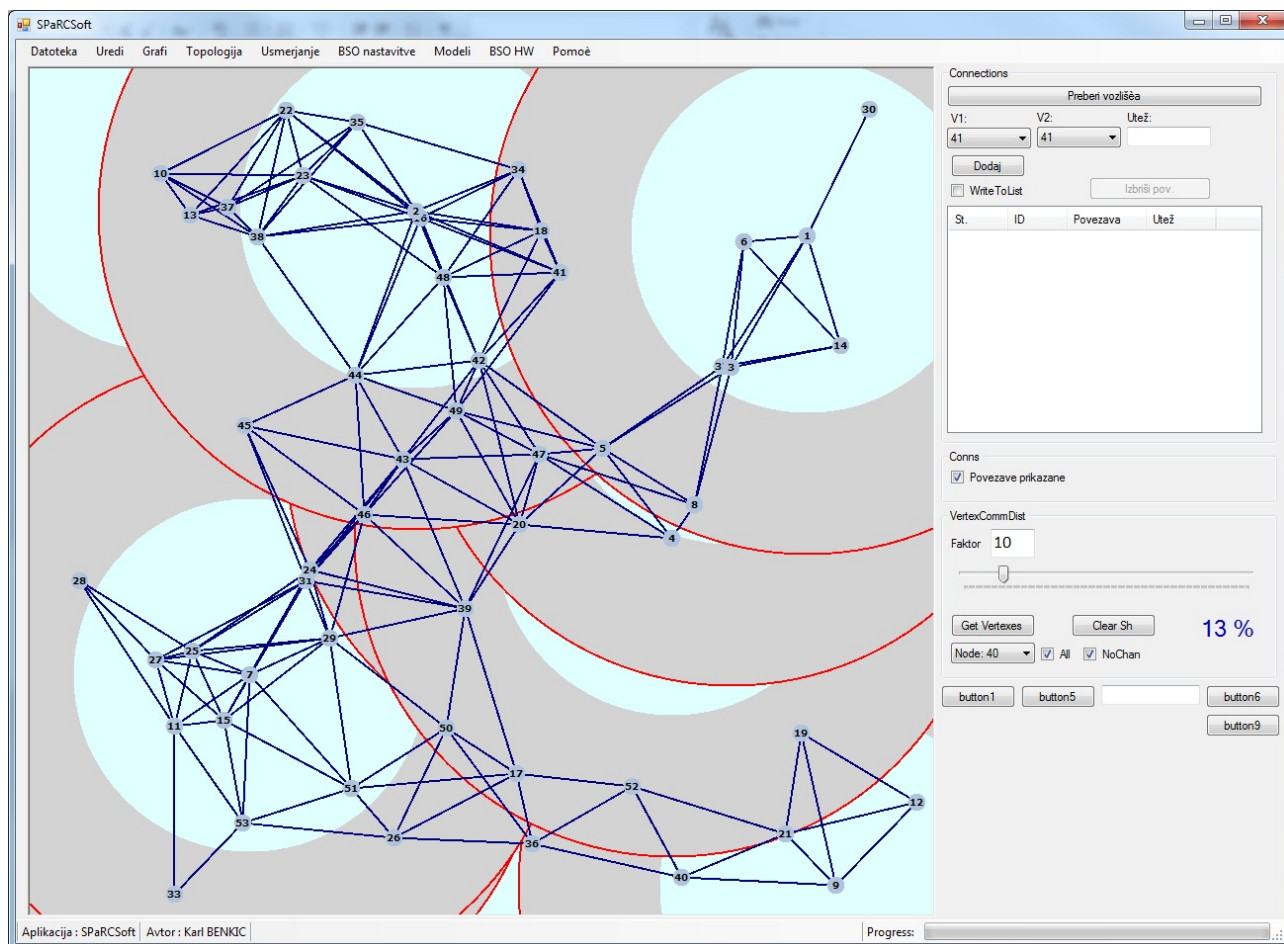
Remote firmware upgrade module je namenjen daljinski nadgradnji programske strojne kode na SPaRCMosquito senzorskih vozliščih.

WSN properties module je modul za ocenjevanje delovanja protokolov. Poleg ocenjevanja služi tudi kot osnovni modul za "vohljač" način delovanja.

SPaRCGraph module je uporabljen za izračun topologij, algoritmov in drugih preračunov. Modul prenese podatke o izračunih po celotnem omrežju ali pa pripravi datoteko za OPNET simulacije.

Meda data module skrbi za pripravo in izmenjavo podatkov in informacij s simulacijskim okoljem OPNET ali razvojem strojne programske kode.

Osnovna podatkovna struktura, ki jo SPaRCSoft uporablja je neusmerjen graf $G = (V, E, W)$ z vozlišči (senzorska vozlišča), povezavami (povezave med vozlišči) in utežmi (lastnosti kanala, parametri...).



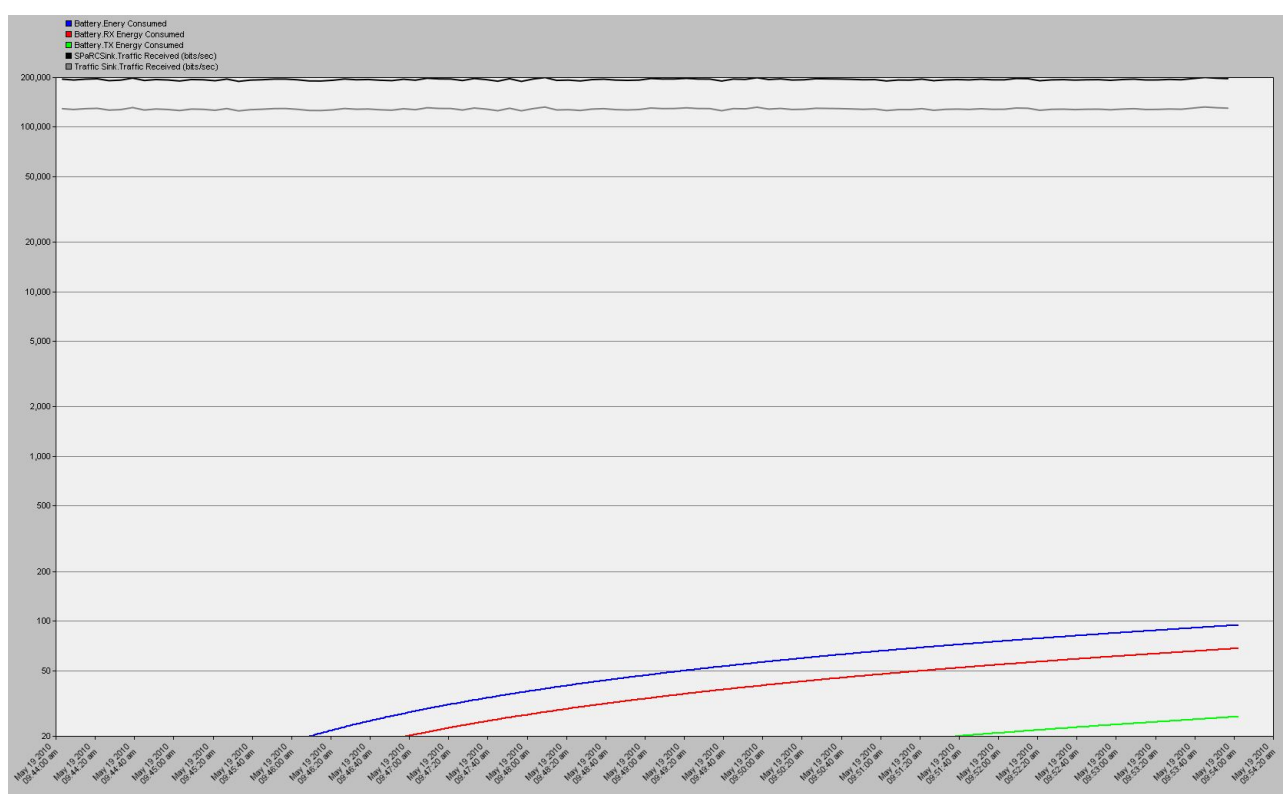
Slika B.2-2
SPaRCSoft programsko orodje.

Programska oprema SPaRCSoft (Slika B.2-2) je bila skozi čas študija razvita namensko za potrebe raziskovanja BSO in teze.

Rezultati simulacij

C.0.1 Simulacije: Vpliv MAC plasti na rezultate simulacij

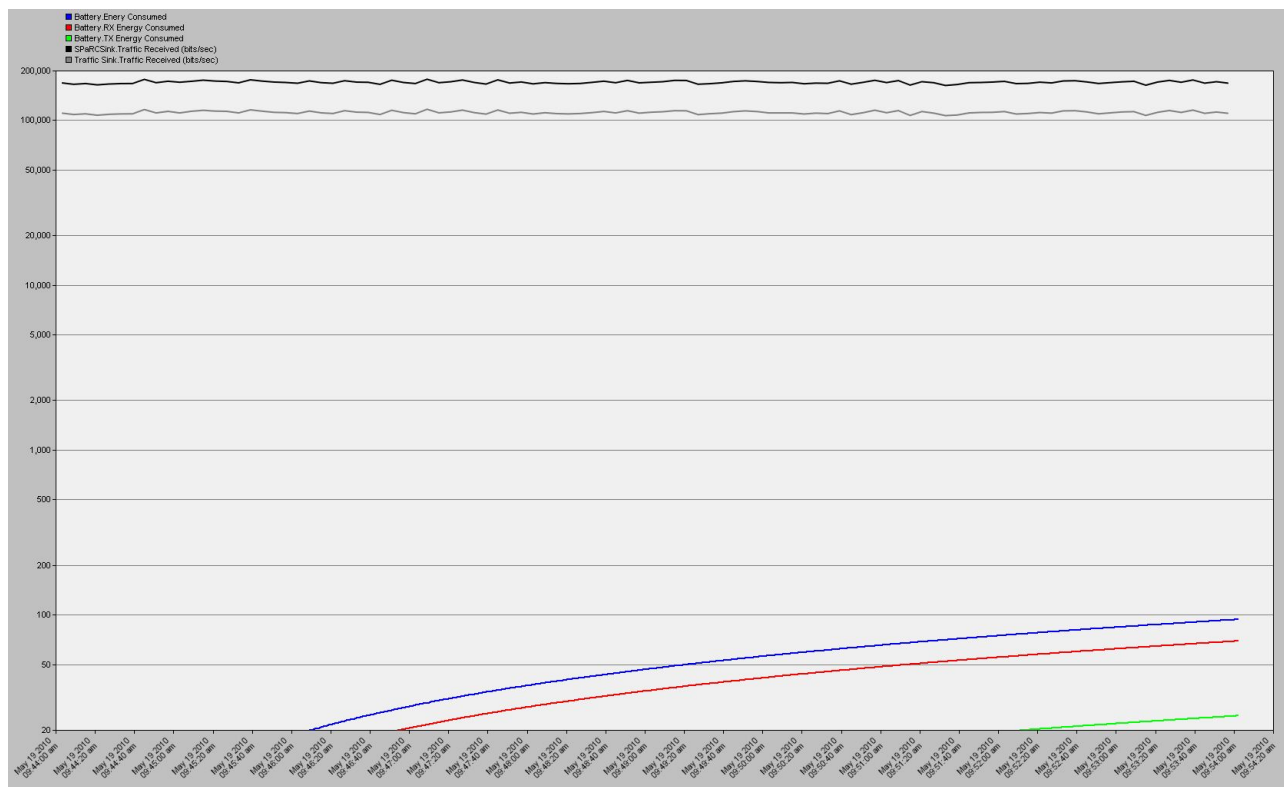
Vpliv skritega vozlišča



Slika C.0-3

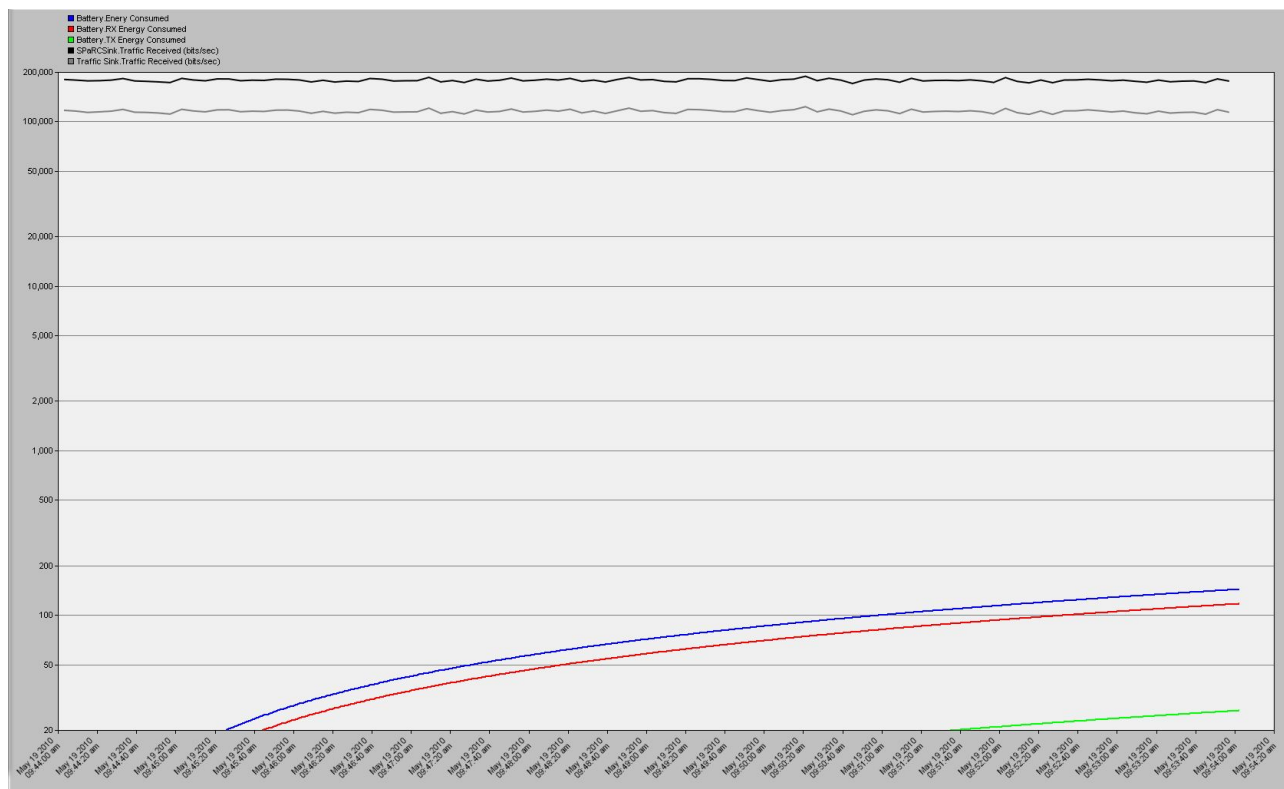
Vpliv MAC plasti, paketne in energijske razmere: 5 vozlišč.

Vpliv izpostavljenega vozlišča



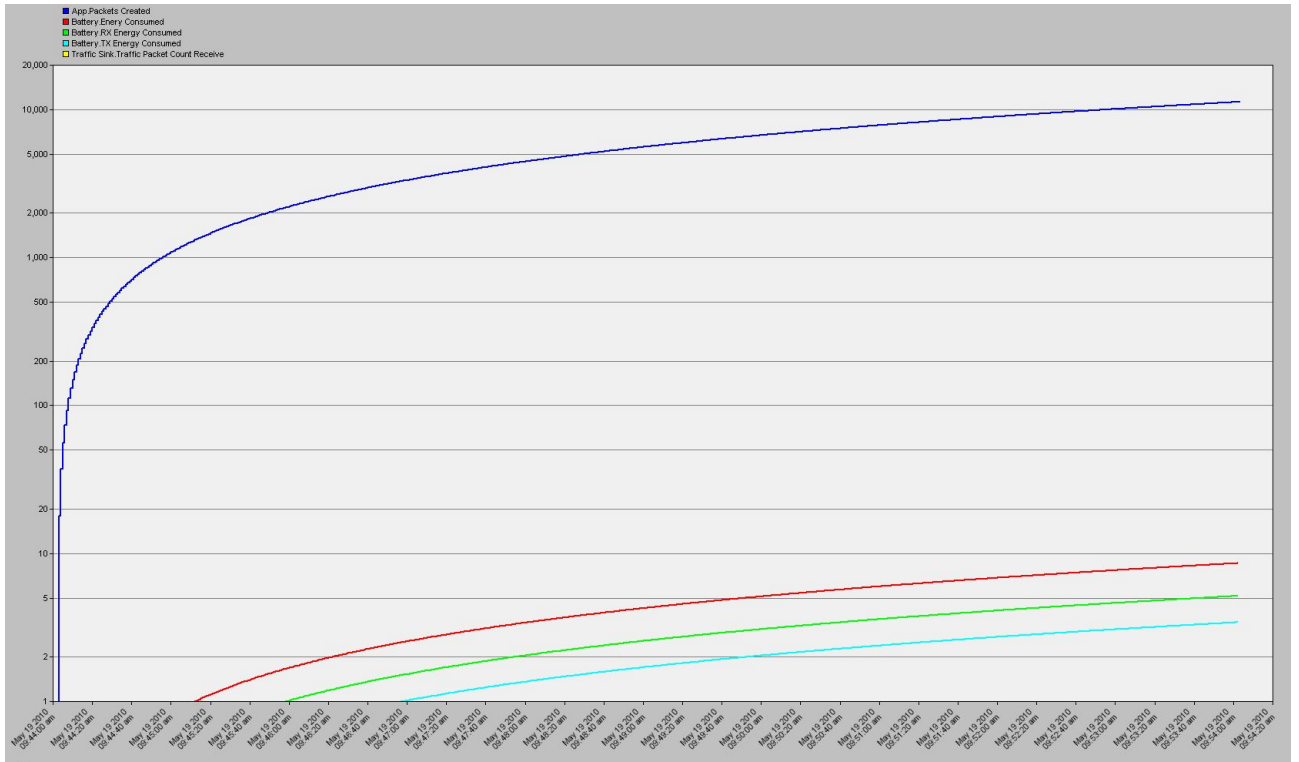
Slika C.0-4

Vpliv MAC plasti, paketne in energijske razmere: 7 vozlišč.



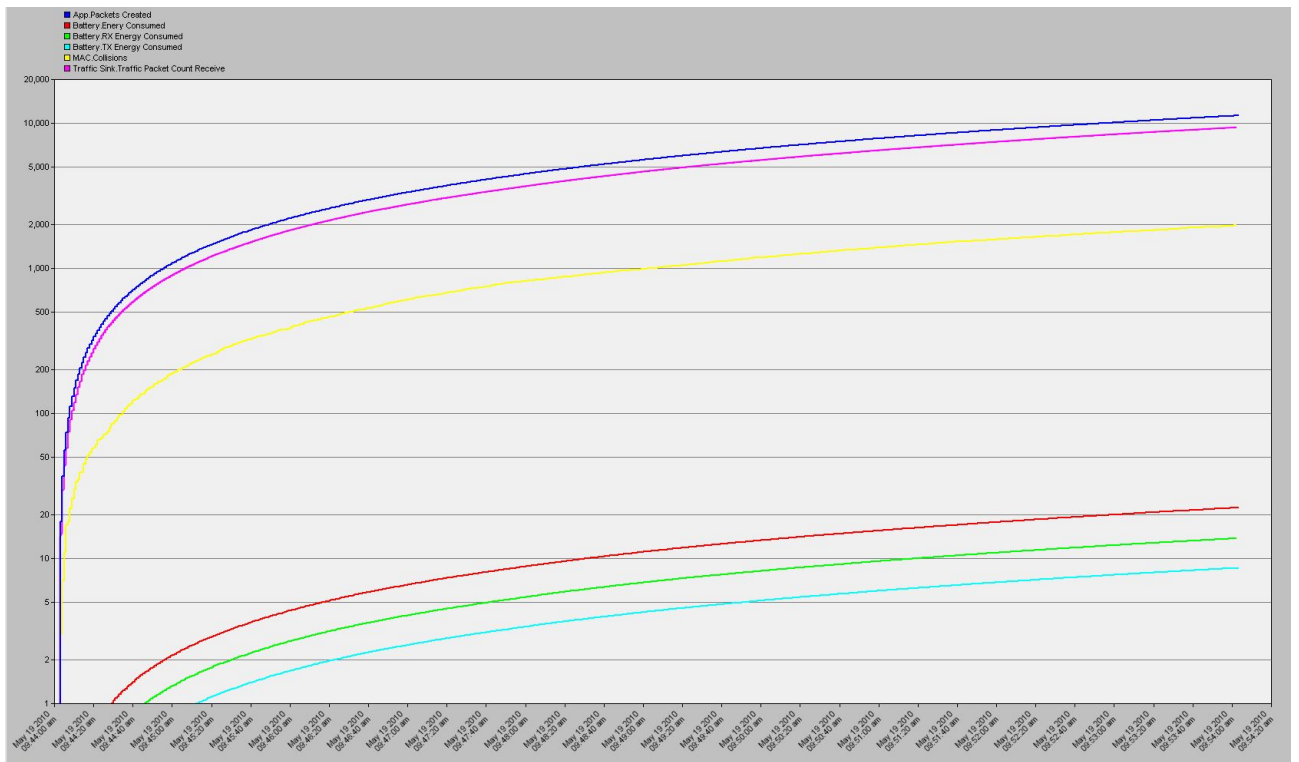
Slika C.0-5

Vpliv MAC plasti, paketne in energijske razmere: 9 vozlišč.



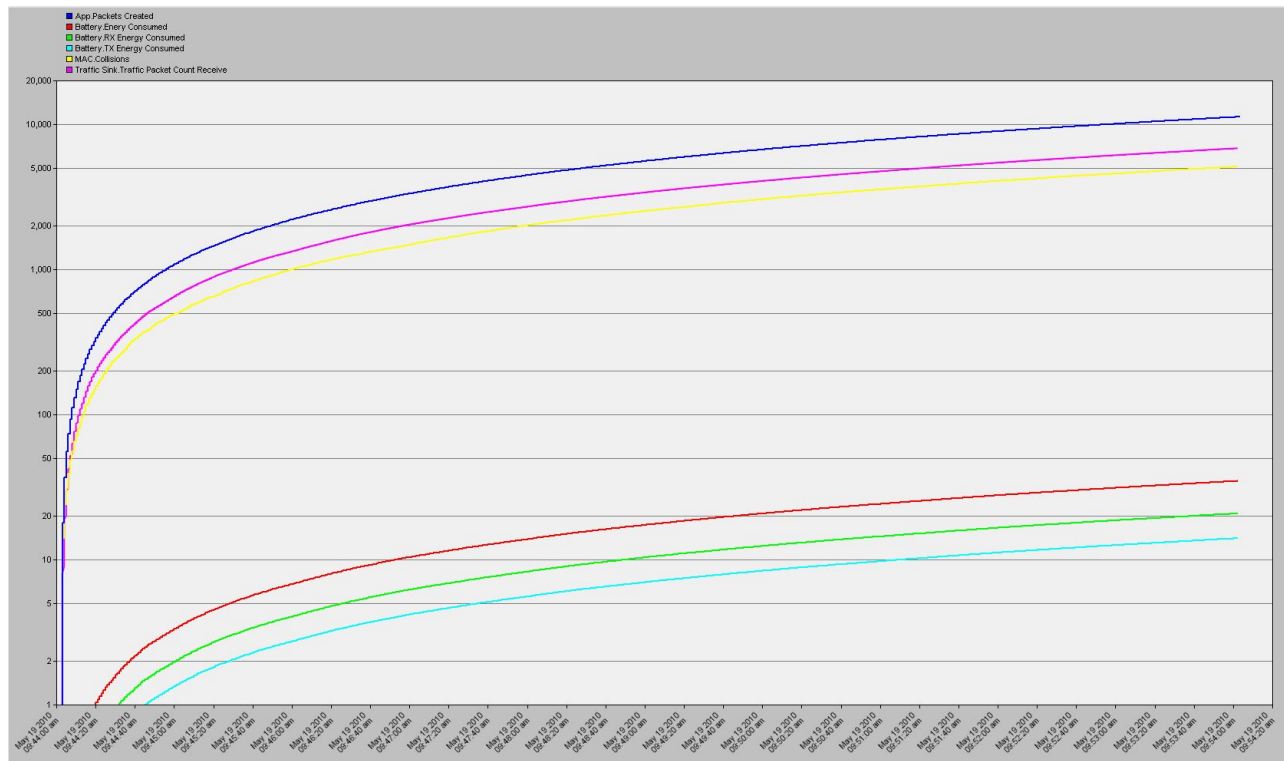
Slika C.0-6

Vpliv izpostavljenega vozlišča: 1 etapa.



Slika C.0-7

Vpliv izpostavljenega vozlišča: 3 etape.



Slika C.0-8

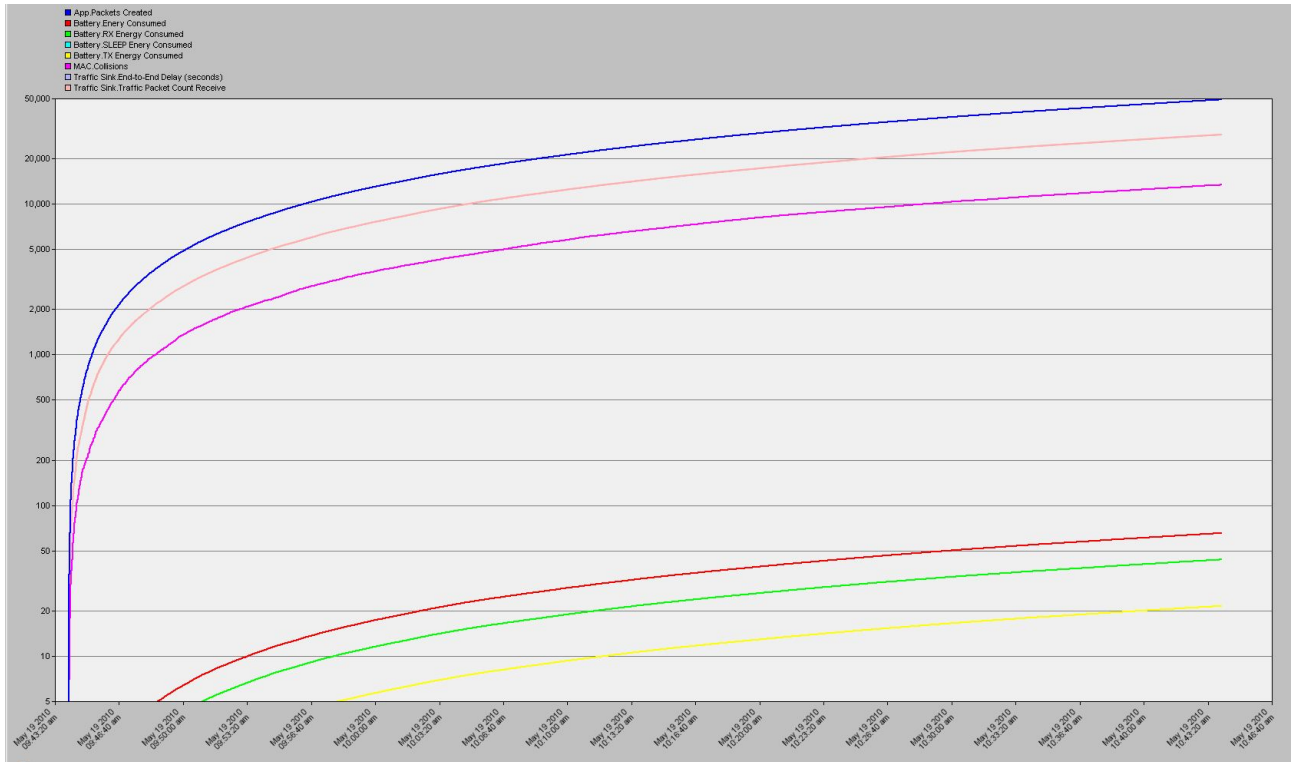
Vpliv izpostavljenega vozlišča: 6 etap.

C.0.2 Simulacije protokola usmerjanja

Simulacije: Vpliv kolizijskih domen na rezultate simulacij

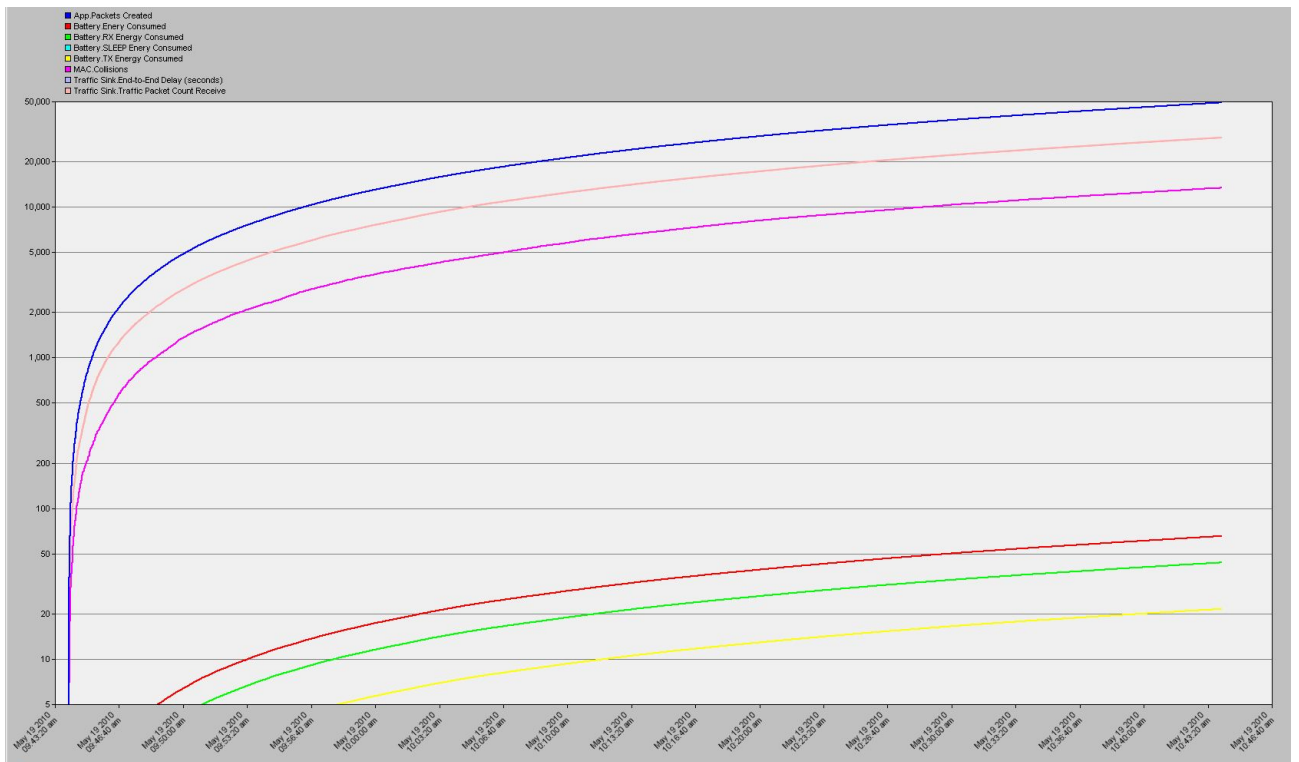
Uravnoteženo in neuravnoteženo drevo

Simulacija dveh manjših senzorskih omrežij



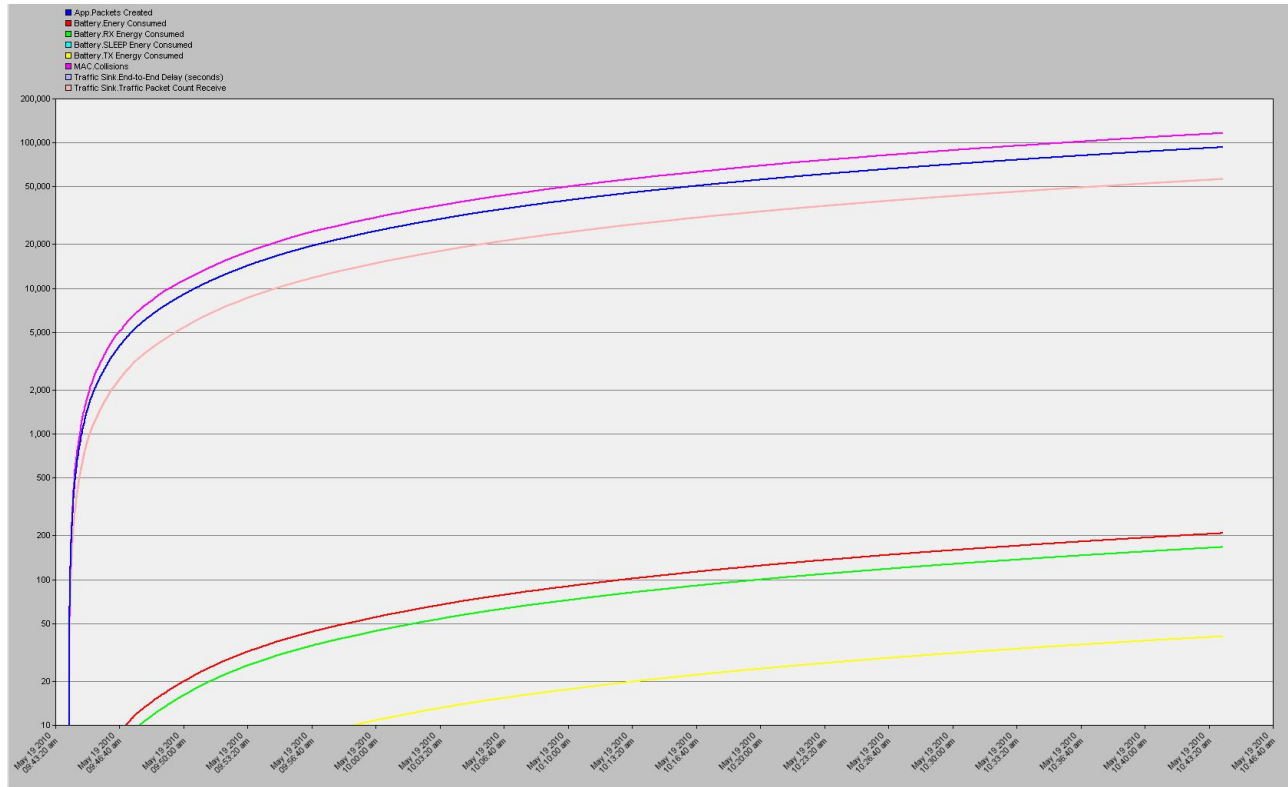
Slika C.0-9

Kolizijske domene v uravnoteženem drevesu: brez skupne kolizijske domene.

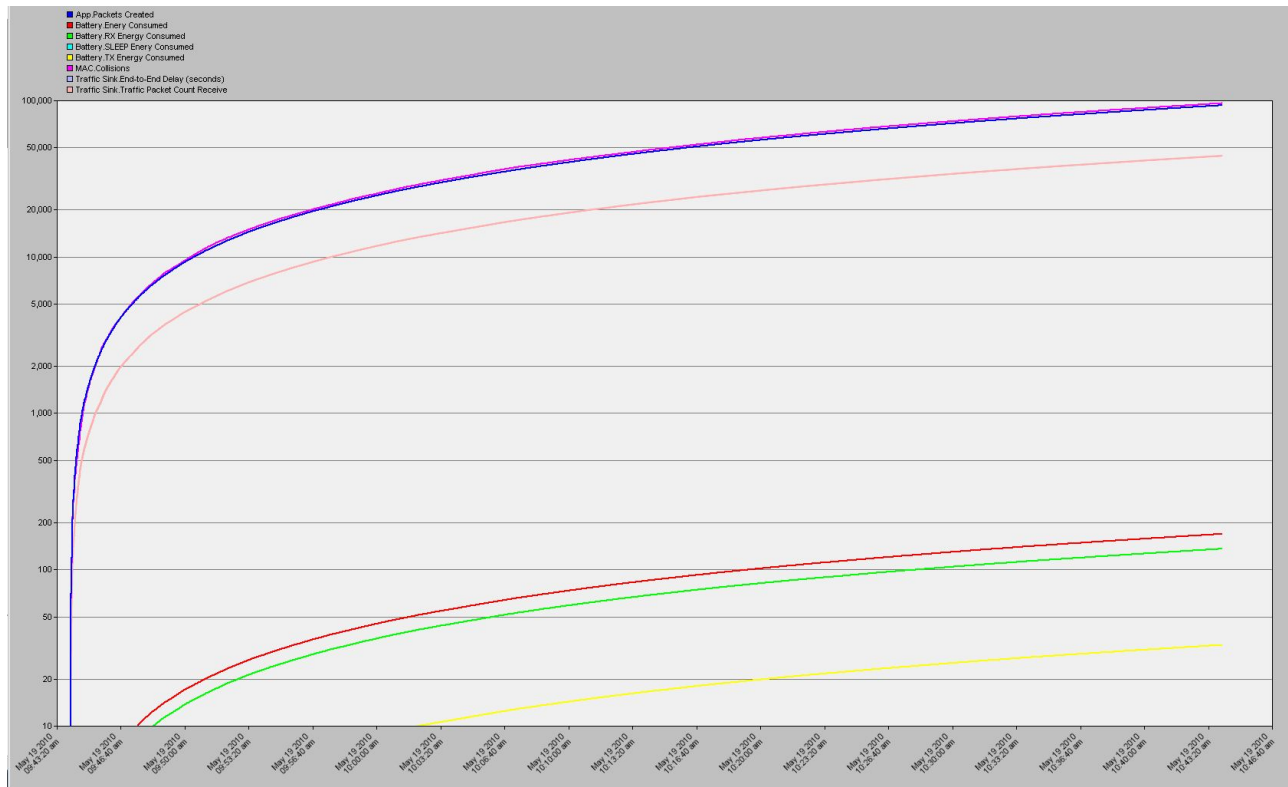


Slika C.0-10

Kolizijske domene v uravnoteženem drevesu: skupna kolizijska domena.



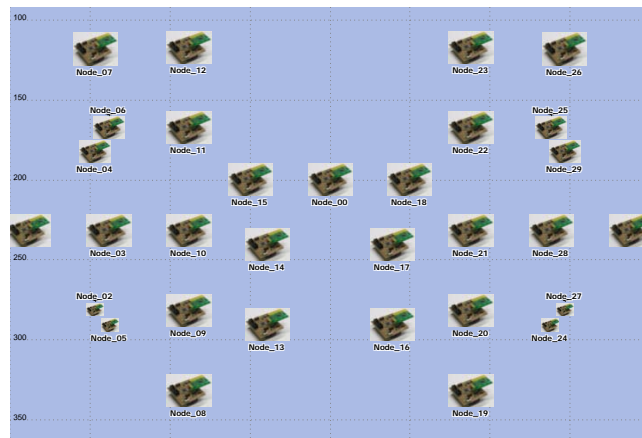
Slika C.0-11
 Simulacijski rezultati uravnoveženega drevesa.



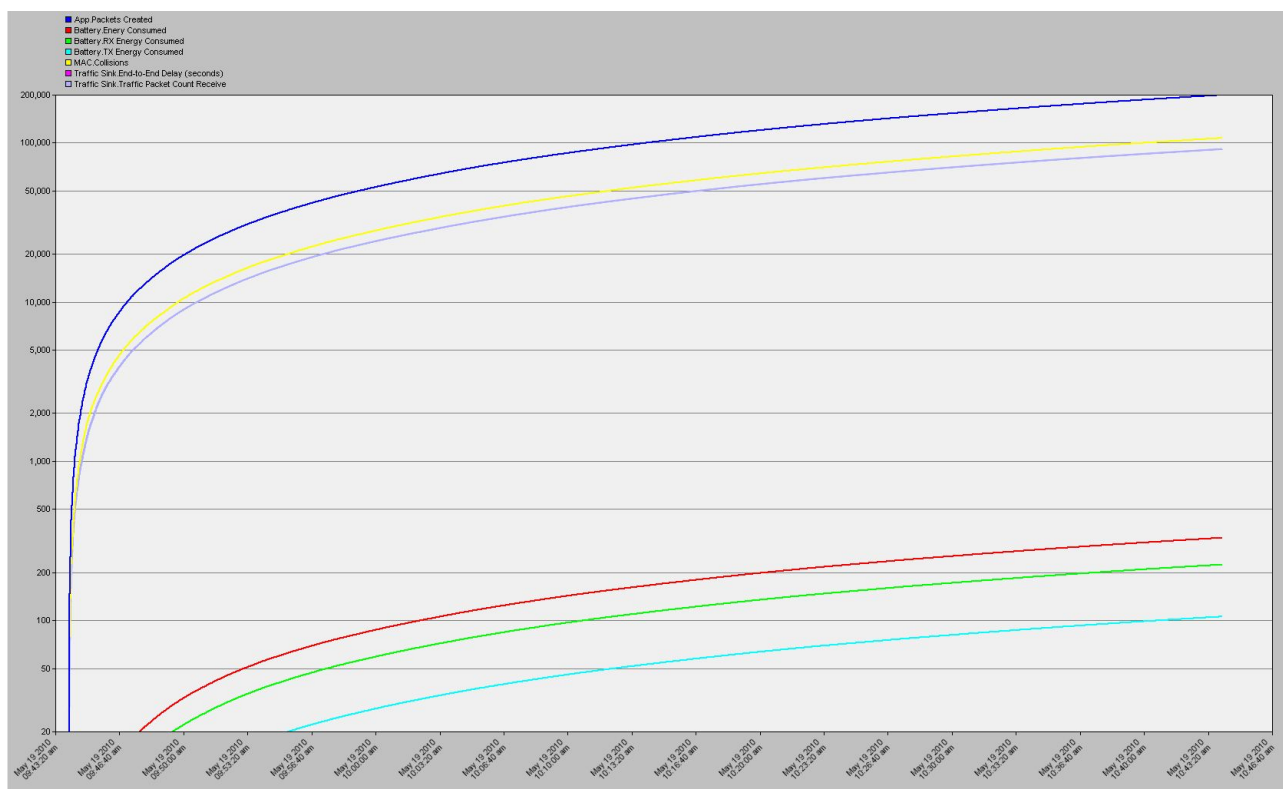
Slika C.0-12
 Simulacijski rezultati neuravnoveženega drevesa.



(a) Uravnoteženo omrežje.

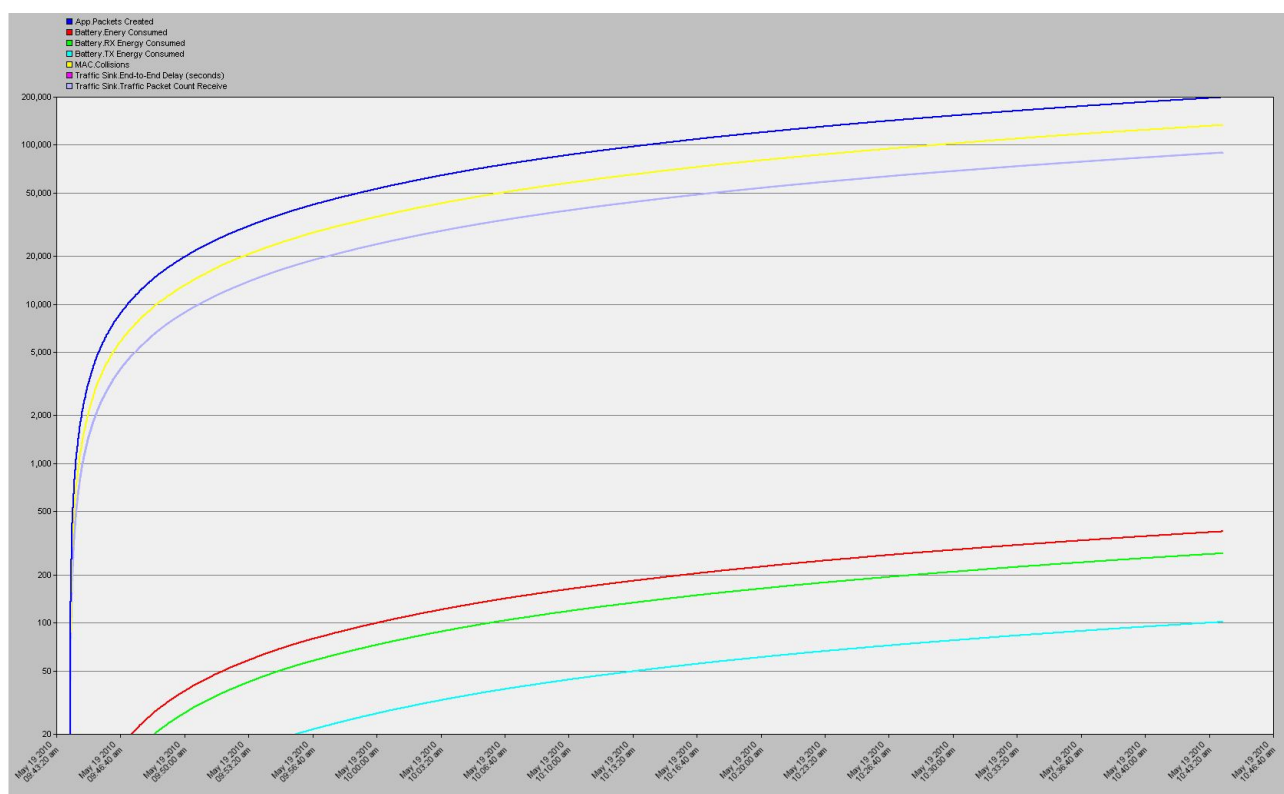


(b) Neuravnoteženo omrežje.



Slika C.0-13

Simulacijski rezultati uravnoteženega scenarija.



Slika C.0-14

Simulacijski rezultati neuravnoteženega scenarija.

Literatura

- [1] European eyes project (ist-2001-34734). <http://www.eyes.eu.org/>. 63
- [2] leee 802.15 wpan task group 4 (tg4). <http://www.ieee802.org/15/pub/TG4.html>. 64, 68
- [3] *The Research of Network Architecture in Warehouse Management System Based on RFID and WSN Integration*, September 2008. 6
- [4] 3Com. Understanding ip addressing: Everything you ever wanted to know. http://www.3com.com/other/pdfs/infra/corpinfo/en_US/501302.pdf. 62
- [5] Xiaoxia Huang A, Jianfeng Wang B, and Yuguang Fang A. Achieving maximum flow in interference-aware wireless sensor networks with smart. *Ad Hoc Networks*, 5(6), 2007. 10, 11
- [6] C-C Han A. Sawides and M. Srivastava. Wireless sensor networks: A survey on ultra-low power-aware design. In *Seventh ACM Annual International Conference on Mobile Computing and Networking (MobiCom)*, 2001. 28
- [7] Adel Ali Ahmed and Norsheila Fisal. A real-time routing protocol with load distribution in wireless sensor networks. *Comput. Commun.*, 31(14):3190–3203, 2008. 11
- [8] Zeeshan Akbar. Evaluation of aodv and dsr routing protocols of wireless sensor networks for monitoring applications. Master's thesis, Blekinge Institute of Technology, 2009. 65
- [9] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci. Wireless sensor networks: a survey. *Computer Networks*, 38(4):393–422, 2002. 2, 6
- [10] Jamal N. Al-Karaki and Ahmed E. Kamal. Routing techniques in wireless sensor networks:a survey. *IEEE Wireless Communications*, 2004. 29
- [11] L. Alazzawi and A. Elkateeb. Performance evaluation of the wsn routing protocols scalability. *Journal of Computer Systems, Networks, and Communications*, 2008. 11
- [12] B. Ames. Electronics are central to 21st century warfare tactics. In *military and Aerospace Electronics*, Januar 2004. 6, 7
- [13] John J. Amoss and Daniel Minoli. *Handbook of IPv4 to IPv6 Transition: Methodologies for Institutional and Corporate Networks*. Taylor & Francis Group, 2008. 62

- [14] ZHAN An-dong, XU Tian-yin, CHEN Gui-hai, YE Bao-liu, and LU Sang-lu. A survey on real-time routing protocols for wireless sensor networks. <http://cs.nju.edu.cn/dislab/xuty/papers/cwsn08.pdf>. 11
- [15] B. Awerbuch, Y. Azar, E.F. Grove, Ming-Yang Kao, P. Krishnan, and J.S. Vitter. Load balancing in the L_p norm. *Foundations of Computer Science, Annual IEEE Symposium on*, 0:383, 1995. 47
- [16] Lichun Bao and J.J. Garcia-Luna-Aceves. Stable energy-aware topology management in ad hoc networks. *Ad Hoc Networks*, 2009. In Press, Corrected Proof. 11, 12
- [17] L. Benini, E. Farella, and C. Guiducci. Wireless sensor networks: Enabling technology for ambient intelligence. *Microelectronics Journal*, 37(12):1639–1649, 2006. 6
- [18] Karl Benkič. Modeliranje cdma brezžičnih senzorskih omrežij. Master's thesis, FER Faculty, University of Maribor, 2008. 32, 37, 59, 91
- [19] Karl Benkič, Marko Malajner, and Žarko Čučej. Aewsn: Academic educational wireless sensor network. *MIDEM*, 41, 2011. manuscript, waiting to be published 2010. 93
- [20] Karl Benkič, Marko Malajner, Peter Planinšič, and Žarko Čučej. On line measurements and visualisation of distances in wsn with rssi parameter. In *International Conference on Systems, Signals and Image Processing*, 2009. 28
- [21] Marc Blanchet. *Migrating to IPv6: A Practical Guide for Mobile and Fixed Networks*. John Wiley & Sons, 2005. 62
- [22] Jan Blumenthal, Frank Reichenbach, and Dirk Timmermann. Position estimation in ad-hoc wireless sensor networks with low complexity. In *in Joint 2nd Workshop on Positioning, Navigation and Communication 2005 (WPNC 05) & 1st Ultra-Wideband Expert Talk 2005 (05)*, pages 41–49, 2005. 55
- [23] Drago Bokal, Bostjan Brešar, and Janja Jerebic. A generalization of Hungarian method and Hall's theorem with applications in wireless sensor networks. manuscript, 2009. 45, 46, 47, 58
- [24] Prosenjit Bose, Pat Morin, Ivan Stojmenovic, and Jorge Urrutia. Routing with guaranteed delivery in ad hoc wireless networks. *Wireless Networks*, 7(6):609–616, 2001. 38
- [25] K. M. Sivalingam C. S. Raghavendra and T. Znati Eds. *Wireless sensor networks*. Kluwer Academic, 2004. 8
- [26] Roger Carr. Simulated annealing. internetni vir, From MathWorld—A Wolfram Web Resource, created by Eric W. Weisstein, 2010. <http://mathworld.wolfram.com/SimulatedAnnealing.html>. 24
- [27] Boris V. Cherkassky, Andrew V. Goldberg, Paul Martin, Joao C. Setubal, and Jorge Stolfi. Augment or push: a computational study of bipartite matching and unit-capacity flow algorithms. *J. Exp. Algorithmics*, 3:8, 1998. 43

- [28] B.V. Cherkassky, A.V. Goldberg, P. Martin, J.C. Setubal, and J. Stolfi. Augment or push? a computational study of bipartite matching and unit capacity flow algorithms, 1998. Izvorna koda na voljo na spletu: <http://www.avglab.com/andrew/soft.html>. 54
- [29] CiA. Can in automation. <http://www.can-cia.org/>. 62
- [30] CodeProject. Introduction to graph with breadth first search(bfs) and depth first search(dfs) traversal implemented in java. internetni vir, 2010. <http://www.codeproject.com/KB/java/BFSDFS.aspx>. 24
- [31] Cavin D., Y. Sasson, and A. Schiper. On the accuracy of manet simulators. In *Proceedings of the second ACM international workshop on Principles of mobile computing*, 2002. 11
- [32] Hui Dai and Richard Han. A node-centric load balancing algorithm for wireless sensor networks. In *Global Telecommunications Conference*, 2003. 11, 40
- [33] Samir R. Das and Charles E. Perkins. Performance comparison of two on-demand routing protocols for ad hoc networks. pages 3–12, 2000. 10
- [34] Digi. Xbee zigbee addressing. <http://www.digi.com/support/kbase/kbaseresultdet1.jsp?kb=191>. 62
- [35] L. Doherty, K.S.J. pister, and L. El Ghaoui. Convex position estimation in wireless sensor networks. In *in INFOCOM 2001. Twentieth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, pages 1655–1663, 2001. 55
- [36] Sinem Coleri Ergen. Zigbee/ieee 802.15.4 summary. <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.124.6333&rep=rep1&type=pdf>. 64
- [37] Janez Žerovnik. *Osnove teorije grafov in diskretne optimizacije*. Tiskarna tehničnih fakultet, 2005. 16, 19, 21, 23, 24, 25
- [38] D. Estrin, R. Govindan, J.S. Heidemann, and S. Kumar. Next century challenges: Scalable coordination in sensor networks. *Mobile Computing and Networking*, pages 263–270, 1999. 6
- [39] *Instrumenting the world with wireless sensor networks*, 2001. 2
- [40] Konrad Etschberger. *Controller Area Network (Hardcover)*. IXXAT Automation GmbH, 2001. 62
- [41] Eyes. Energy efficient sensor network. internetni vir, 2010. <http://www.eyes.eu.org/>. 5
- [42] Jittat Fakcharoenphol, Bundit Lekhanukit, and Danupon Nanongkai. A faster algorithm for finding optimal semi-matching, 2007. 47, 58
- [43] J. Antonio Garcia-Macias and Javier Gomez-Castellanos. *Sensor Networks and Configuration: Fundamentals, Standards, Platforms, and Applications*, chapter MANET versus WSN. Springer-Verlag, 2006. 4
- [44] Drew Gislason. *ZigBee Wireless Networking*. Elsevier, 2008. 62

- [45] P. Gupta and P. R. Kumar. The capacity of wireless networks. *Information Theory, IEEE Transactions on*, 46(2):388–404, 2000. 38
- [46] Zygmunt J. Haas and Marc R. Pearlman. Providing ad-hoc connectivity with the reconfigurable wireless networks. In *Proceedings of the ACM SIGCOMM '98*, pages 166–177. Addison Wesley Longman, 1998. 10
- [47] Silvia Hagen. *IPv6 Essentials, Second Edition*. O'Reilly Media, 2006. 62
- [48] Professor Bernd Hamann. Visualization of great duck island sensor network data. internetni vir, 2010. <http://www.greatduckisland.net/>. 7
- [49] Tero Harju. Lecture notes on graph theory. internetni vir, 2007. Lecture Notes published on: <http://users.utu.fi/harju/graphtheory/graphtheory.pdf>. 20
- [50] Nicholas J. A. Harvey, Richard E. Ladner, László Lovász, and Tami Tamir. Semi-matchings for bipartite graphs and load balancing. *J. Algorithms*, 59(1):53–78, 2006. 45, 46, 47, 54, 58
- [51] John E. Hopcroft and Richard M. Karp. An $n^{5/2}$ algorithm for maximum matchings in bipartite graphs. *SIAM Journal on Computing*, 2(4):225–231, 1973. 43
- [52] International Telecommunication Union ITU. Radiocommunication sector (itu-r). <http://www.itu.int/ITU-R/index.asp?category=information&rlink=rhome&lang=en>. 2
- [53] Aitzol Zuloaga Itziar Marín, Eduardo Arceredillo and Jagoba Arias. Wireless sensor networks: A survey on ultra-low power-aware design. In *WORLD ACADEMY OF SCIENCE, ENGINEERING AND TECHNOLOGY*, 2005. 27
- [54] Bruck Jehoshua, Gao Jie, and Jiang Anxiao. Map: Medial axis based geometric routing in sensor networks. In *Wireless Networks, Special Issue: Selected Papers on MOBICOM 2005*, pages 835–853, 2007. 10
- [55] Bruck Jehoshua, Gao Jie, and Jiang Anxiao. Map: Medial axis based geometric routing in sensor networks. *Wireless Networks*, 13:835–853, 2007. Selected Papers on MOBICOM 2005. 11
- [56] D. Johnson. The dynamic source routing protocol (dsr) for mobile ad hoc networks for ipv4. <http://tools.ietf.org/html/rfc4728>. 64
- [57] David B. Johnson, David A. Maltz, and Josh Broch. Dsr: The dynamic source routing protocol for multihop wireless ad hoc networks, chapter 5, 2001. 10, 64
- [58] Sudarev J.V., White L.B., and Perreau S. Performance analysis of 802.11 csma/ca for infrastructure networks under finite load conditions. In *LANMAN*, 2005. 76
- [59] Benkič K., Brešar B., Čučej Ž., Galčik F., Katrenič J., and Semanišin G. Delay and traffic based load balanced routing protocol in cdma wireless sensor networks. manuscript, 2010. 37
- [60] Romer K. and Mattern F. The design space of wireless sensor networks. *IEEE Wireless Communications*, 11(6), 2004. 8

- [61] K. Kim A. Medour K. Sarkar, Z. Ji and M. Salazar-Palma. A survey of various propagation models for mobile communication. *IEEE Antennas and Propagation Magazine*, pages 51–82, 2000. 33
- [62] BENKIČ Karl, MALAJNER Marko, PEŠOVIĆ Uroš, and ČUČEJ Žarko. Simplified opnet node model for developing load balanced routing protocol in wsn. In *XVII Telecommunications Forum - TELFOR 2009*, pages 210–213, 2009. 67
- [63] Phil Karn. Maca—a new channel access method for packet radio. In *ARRL/CRRL Amateur Radio 9th Computer Networking Conference (1990)*, 1990. 69
- [64] Brad Karp. Gpsr: Greedy perimeter stateless routing for wireless networks. pages 243–254, 2000. 11
- [65] Luke Klein-Berndt. A quick guide to aodv routing. http://w3.antd.nist.gov/wctg/aodv_kernel/aodv_guide.pdf. 64
- [66] Denes Konig. *Theorie der Endlichen und Unendlichen Graphen: Kombinatorische Topologie der Streckenkomplexe*. Chelsea Publishing Company, 1936. 15
- [67] Amos Korman. Improved compact routing schemes for dynamic trees. In *PODC '08: Proceedings of the twenty-seventh ACM symposium on Principles of distributed computing*, pages 185–194, New York, NY, USA, 2008. ACM. 39
- [68] Dariusz Koscielnik. Influence of the hidden stations and the exposed station for the throughput of the lr-wpan. In *proceedings of the IEEE International Symposium on Industrial Electronics (ISIE 2009)*, 2009. 76
- [69] Anis Koubaa, Mário Alves, and Eduardo Tovar. A comprehensive simulation study of slotted csma/ca for ieee 802.15.4 wireless sensor networks. In *proceedings of the 5th IEEE International Workshop on Factory Communication Systems (WFCS'06)*, 2006. 76
- [70] Fabian Kuhn and Aaron Zollinger. Ad-hoc networks beyond unit disk graphs. In *DIALM-POMC '03: Proceedings of the 2003 joint workshop on Foundations of mobile computing*, pages 69–78, New York, NY, USA, 2003. ACM. 38
- [71] H. W. Kuhn. The Hungarian method for the assignment problem. *Naval Research Logistic Quarterly*, 2:83–97, 1955. 43, 47, 49
- [72] Kofi A. Laing. Name-independent compact routing in trees. *Inf. Process. Lett.*, 103(2):57–60, 2007. 39
- [73] Tak-Wah Lam, Hing-Fung Ting, Kar-Keung To, and Wai-Ha Wong. On-line load balancing of temporary tasks revisited. *Theoretical Computer Science*, 270(1–2):325 – 340, 2002. 44
- [74] Koen Langendoen and Niels Reijers. Distributed localization in wireless sensor networks: a quantitative comparison. *Computer Networks*, 43:499–518, 2003. 55
- [75] Eugene Lawler. *Combinatorial Optimization: Networks and Matroids*. Dover Publications, 2001. 47, 49

- [76] J.K. Lenstra, D.B. Shmoys, and E.Tardos. Approximation algorithms for scheduling unrelated parallel machines. *Math. Programming*, 46:259–271, 1990. 58
- [77] Xiang-Yang Li, Gruia Calinescu, and Peng-Jun Wan. Distributed construction of a planar spanner and routing for ad hoc wireless networks. In *IEEE INFOCOM*, pages 1268–1277, 2002. 38
- [78] Xiang-Yang Li, Wen-Zhan Song, and Yu Wang. Efficient topology control for ad-hoc wireless networks with non-uniform transmission ranges. *Wireless Networks*, 11(3):255–264, May 2005. 38
- [79] Xiang-Yang Li, Peng-Jun Wan, and Yu Wang. Power efficient and sparse spanner for wireless ad hoc networks. In *Int. Conf. on Computer Communications and Networks, ICCCN01*, pages 564–567, 2001. 38
- [80] Ben Liang, Zygmunt J. Haas, and Senior Member. Hybrid routing in ad hoc networks with a dynamic virtual backbone. In *Cornell University. He joined the Department of Electrical and Computer Engineering at the University of Toronto*, page 1392, 1997. 10
- [81] Liu, Mark Yarvis, W. Steven Conner, and Xingang Guo. Guaranteed on-demand discovery of node-disjoint paths in ad hoc networks. *Computer Communications*, 30:2917–2930, 2007. 11
- [82] Chor Ping Low. On load-balanced semi-matchings for weighted bipartite graphs. In *Lecture Notes in Computer Science – Theory and Applications of Models of Computation*. 58
- [83] Chor Ping Low. An approximation algorithm for the load-balanced semi-matching problem in weighted bipartite graphs. *Information Processing Letters*, 100(4):154–161, 2006. 47, 58
- [84] Fyffe M., Min-Te Sun, and Xiaoli Ma. Traffic-adapted load balancing in sensor networks employing geographic routing. In *Wireless Communications and Networking Conference, 2007.WCNC 2007. IEEE*, pages 4389–4394, 2007. 11
- [85] Abinash Mahapatra, Kumar Anand, and Dharma P. Agrawal. Qos and energy aware routing for real-time traffic in wireless sensor networks. *Comput. Commun.*, 29(4):437–445, 2006. 11
- [86] Marko Malajner. Lokalizacijske tehnike v brezžičnih senzorskih omrežjih. Master's thesis, FER Faculty, University of Maribor, 2009. 33, 35, 67, 91
- [87] Marko Malajner, Karl Benkič, Peter Planinšič, and Žarko Čučej. The accuracy of propagation models for distance measurement between wsn nodes. In *International Conference on Systems, Signals and Image Processing*, 2009. 28
- [88] Melanie Mitchell. *An Introduction to Genetic Algorithms*. MIT Press, 1998. 24
- [89] Shree Murthy and J. J. Garcia luna aceves. Baltzer journals an efficient routing protocol for wireless networks, 1996. 10

- [90] E. Keith Lloyd Norman L. Biggs and Robin J. Wilson. *Graph Theory 1736–1936*. Oxford University Press, 1998. 15
- [91] University of California at Berkeley. Robotics and intelligent machines research in eecs. internetni vir, 2010. <http://robotics.eecs.berkeley.edu/wiki/pmwiki/pmwiki.php>. 5
- [92] The Institute of Electrical and Inc. Electronics Engineers. Ieee 802.15 wpan low rate alternative phy task group 4a (tg4a). <http://www.ieee802.org/15/pub/TG4a.html>. 2
- [93] Yongsuk Park and Eun-Sun Jung. Plus-tree: A routing protocol for wireless sensor networks. In *Lecture Notes in Computer Science – Advances in Hybrid Information Technology*, pages 638–646. Springer Berlin / Heidelberg, 2007. 10
- [94] Uroš Pešovič. Mehanizmi reševanja problema skritega vozlišča v IEEE 802.15.4/zigbee brezžičnih senzorskih omrežjih. Master's thesis, FERl Faculty, University of Maribor, 2010. 69
- [95] Charles E. Perkins. Highly dynamic destination-sequenced distance-vector routing (dsv) for mobile computers. In *In Proceedings of ACM SIGCOMM94 Conference*, pages 234–244, 1994. 10
- [96] Charles E. Perkins and Elizabeth M. Royer. Ad-hoc on-demand distance vector routing. In *IEEE WORKSHOP ON MOBILE COMPUTING SYSTEMS AND APPLICATIONS*, pages 90–100, 1999. 64
- [97] Rabbit. An introduction to zigbee. <http://www.rabbit.com/documentation/docs/manuals/ZigBee/Introduction/ZigBeeIntro.pdf>. 62
- [98] Theodore Rappaport. *Wireless Communications: Principles and Practice, 2nd Edition*. Prentice Hall PTR, 2001. 32
- [99] Thomas L. Magnanti Ravindra K. Ahuja and James B. Orlin. *Network Flows: Theory, Algorithms, and Applications (Hardcover)*. Prentice Hall, 1993. 21
- [100] Paolo Santi. *Topology Control in Wireless Ad Hoc and Sensor Networks*. John Wiley & Sons, Ltd, 2005. 11, 30, 33
- [101] Paulo Sérgio Sausen, Marco Aurélio Spohn, and Angelo Perkusich. Broadcast routing in wireless sensor networks with dynamic power management and multi-coverage backbones. *Information Sciences*, 180:653–663, 2010. 11
- [102] David Schneider. Dash7 wireless networking gains momentum. *IEEE spectrum*, 37(12):1639–1649, 2010. 6
- [103] Evgeny V. Shchepin and Nodari N. Vakhania. Approximation algorithms for scheduling unrelated parallel machines. In *Lecture Notes in Computer Science – Theoretical Computer Science: Exploring New Frontiers of Theoretical Informatics*. 58
- [104] Gaurav Srivastava, Paul Boustead, and Joe F. Chicharo. A comparison of topology control algorithms for ad-hoc networks. In *Proceedings of the 2003 Australian Telecommunications, Networks and Applications Conference*, 2003. 30

- [105] Clausen T., Jacquet P., Laouiti A., Minet P., Muhlethaler P., Qayyum A., and Viennot L. Op-timized link state routing protocol (olsr). internetni vir, draft, 2010. <http://www.ietf.org/rfc/rfc3626.txt>. 10
- [106] Žarko čučej. *Komunikacije v sistemih daljinskega vodenja*, volume 1. Univerza v Mariboru, Fakulteta za Elektrotehniko, računalništvo in informatiko, Maribor, Slovenija, 1998. 62
- [107] Raziskave brezžičnih senzorskih omrežij. internetni vir, 2010. <http://www.research.rutgers.edu/~mini/sensornetworks.html>. 8
- [108] Jan van Leeuwen and Richard B. Tan. Compact routing methods: A survey. In *SIROCCO*, pages 99–110, 1994. 39
- [109] Yu Wang. Topology control for wireless sensor networks. In *Wireless Sensor Networks and Applications*, pages 113–147. 2008. 38
- [110] Yu Wang and Xiang-Yang Li. Localized construction of bounded degree and planar spanner for wireless ad hoc networks. *Mob. Netw. Appl.*, 11(2):161–175, 2006. 38
- [111] R. Want, A. Hopper, V. Falcao, and J. Gibbons. The active badge location system. *Acm Transactions on Information Systems*, 10(1):91–102, 1991. 2
- [112] Douglas B. West. *Introduction to Graph Theory (2nd Edition)*. Prentice Hall, 2001. 21, 25
- [113] K. Whitehouse and D. Culler. Calibration as parameter estimation in sensor networks. In *roceedings of ACM International Workshop on Wireless Sensor Networks and Applications (WSNA'02)*, September 2002. 5
- [114] Wikipedia. Load balancing (computing). internetni vir, 2010. [http://en.wikipedia.org/wiki/Load_balancing_\(computing\)](http://en.wikipedia.org/wiki/Load_balancing_(computing)). 39
- [115] Wikipedia. Predstavitev notacije velikega o. internetni vir, 2010. http://en.wikipedia.org/wiki/Big_O_notation. xxv, 23, 24
- [116] Wikipedia. Predstavitev teorije grafov. internetni vir, 2010. http://en.wikipedia.org/wiki/Graph_theory. 15
- [117] Wikipedia. Queue (data structure). internetni vir, 2010. [http://en.wikipedia.org/wiki/Queue_\(data_structure\)](http://en.wikipedia.org/wiki/Queue_(data_structure)). 24
- [118] Robin J. Wilson and John J. Watkins. *Uvod v teorijo grafov*. DMFA–založništvo, 1997. 20, 21
- [119] WiSeNts. Embedded wisents – project fp6-004400. internetni vir, 2010. <http://www.embedded-wisents.org/>. 5
- [120] J. Wu, P. Havinga, S. Dulman, and T. Nieberg. Eyes source routing protocol for wireless sensor networks. In *European Workshop on Wireless Sensor Networks EWSN*, pages 67–71, 2004. 63, 64, 65
- [121] M. Younis and K. Akkaya. Strategies and techniques for node placement in wireless sensor networks: A survey. *Ad Hoc Networks*, 6(4):621–655, June 2008. 11

-
- [122] ZebraNet. The zebranet wildlife tracker. internetni vir, 2010. <http://www.princeton.edu/~mrm/zebranet.html>. 7
- [123] Yang Zhang, Jian Wu, and Havinga P.J.M. Implementation of an on-demand routing protocol for wireless sensor networks. In *13th International Conference on Telecommunications*, pages 1–4, 2006. 63, 64



Univerza v Mariboru



Fakulteta za elektrotehniko,
računalništvo in informatiko
2000 Maribor, Smetanova 17

mag. Karl Benkič
Maribor, 8. julij 2010

Izjava

Izjavljam, da je doktorska disertacija pod mentorstvom prof.dr. Žarka Čučēja in somentorstvom prof.dr. Boštjana Brešarja rezultat lastnega raziskovalnega dela ter, da so podatki korektno navedeni.

mag. Karl Benkič

UNIVERZA V MARIBORU

Fakulteta za elektrotehniko, računalništvo in informatiko
(ime fakultete)

IZJAVA O ISTOVETNOSTI TISKANE IN ELEKTRONSKE VERZIJE ZAKLJUCNEGA DELA IN
OBJAVI OSEBNIH PODATKOV AVTORJA

Ime in priimek avtorja (avtorice): Karl Benkič
Vpisna številka: 95029183
Študijski program: FERI-E DR ELEKTROTEHNIKA EDOK
Naslov zaključnega dela: Prometno uravnoreženi usmerjevalni algoritmi za
brežžična senzorska omrežja

Mentor: Prof. dr. Žarko Čučej
Somentor: Prof. dr. Boštjan Brešar

Podpisani-a Karl Benkič izjavljam, da sem za potrebe arhiviranja oddal-a elektronsko verzijo zaključnega dela v Digitalno knjižnico Univerze v Mariboru. Zaključno delo sem izdelal-a sam-a ob pomoči mentorja. V skladu s 1. odstavkom 21. člena Zakona o avtorskih in sorodnih pravicah (Ur. l. RS, št. 16/2007) dovoljujem, da se zgoraj navedeno zaključno delo objavi na portalu Digitalne knjižnice Univerze v Mariboru.

Tiskana verzija zaključnega dela je istovetna elektronski verziji, ki sem jo oddal-a za objavo v Digitalno knjižnico Univerze v Mariboru. Podpisani-a izjavljam, da dovoljujem objavo osebnih podatkov, vezanih na zaključek študija (ime, priimek, leto in kraj rojstva, datum zagovora, naslov zaključnega dela) na spletnih straneh in v publikacijah UM.

Kraj in datum: 7.8.2010

Podpis avtorja (avtorice): 