UNIVERZA NA PRIMORSKEM
FAKULTETA ZA MATEMATIKO, NARAVOSLOVJE IN
INFORMACIJSKE TEHNOLOGIJE

ZAKLJUČNA NALOGA

ZAKLJUČNA NALOGA

ANONIMNI SPOROČILNI SISTEM NA OSNOVI ZANČNEGA OMREŽJA

ALEKSANDAR TODOROVIĆ

UNIVERZA NA PRIMORSKEM
FAKULTETA ZA MATEMATIKO, NARAVOSLOVJE IN
INFORMACIJSKE TEHNOLOGIJE

Zaključna naloga

Anonimni sporočilni sistem na osnovi zančnega omrežja

(Anonymous messaging system based on mesh networking)

Ime in priimek: Aleksandar Todorović
Študijski program: Računalništvo in informatika
Mentor: doc. dr. Branko Kavšek
Somentor: dr. Jernej Vičič

Koper, september 2013

Ključna dokumentacijska informacija

Ime in PRIIMEK: Aleksandar TODOROVIĆ

Naslov zaključne naloge: Anonimni sporočilni sitem na osnovi zančnega omrežja

Kraj: Koper

Leto: 2013

Število listov: 41 število slik: 8 število tabel:3

Število prilog: 2 število strani prilog: 3 število referenc: 29

Mentor: doc. dr. Branko Kavšek

Somentor: dr. Jernej Vičič

UDK:

Ključne besede: zančno omrežje, anonimni sporočilni sistem, Android

Izvleček:

Zaključna projektna naloga predstavlja anonimno izmenjevanje sporočil v zančnem omrežju s pomočjo mobilnih naprav.
Naloga nudi pregled področja zančnega omrežja in brezžičnih tehnologij. Dodaten poudarek je na brezžičnih zančnih omrežjih, saj zaključna projektna naloga temelji na njih. Podrobneje so opisani WiFi, WiFi Direct in Bluetooth, saj te tehnologije nudijo možnost implementacije brezžičnega zančnega omrežja na mobilnih napravah.
Opisana je metodologija, predstavljeni so algoritmi za delo z identitetami, povezovanje mobilnih naprav v brezžična zančna omrežja, ravnanje s podatki in rangiranje sporočil. Prikazani so rezultati meritev WiFi in meritev hitrosti vzpostavljanja določenih modulov.
Predstavljena je tudi dostopnost ter pregled najpogostejših licenc in njihova uporabnost za licenciranje testnega programa ter dostopnost testnega programa.

Key words documentation

Name and SURNAME: Aleksandar TODOROVIČ

Title of final project paper: Anonymous messaging system based on mesh networking

Place: Koper

Year: 2013

Number of pages: 41 Number of figures: 8 Number of tables: 3

Number of appendices: 2 Number of appendix pages: 3 Number of references: 29

Mentor: doc. dr. Branko Kavšek

Co-Mentor: dr. Jernej Vičič

UDK:

Keywords: mesh network, anonymous messaging system, Android

Abstract:

The project paper presents Anonymous messaging system based on mesh networking. It provides an overview of mesh networks and wireless technologies. An additional emphasis is given on WiFi, WiFi Direct and Bluetooth as those are the most suitable technologies for wireless mesh networking implementation on mobile devices.
Methodology describes algorithms for work with identities, mobile device connections on mesh networks, data handling and message ranking.
Results of WiFi measurements and measurements of establishment of specific modules are shown.
The availability of the pilot program is described, as is the licensing of the program and a brief overview of the most popular licenses and their usability.

Zahvala

Zahvaljujem se mentorju, doc. dr. Branku Kavšku in somentorju dr. Jerneju Vičiču, za strokovno pomoč pri izdelavi zaključne naloge.

Zahvaljujem se tudi družini za podporo in pomoč med študijem.

Kazalo vsebine

1 Uvod **1**
1.1 Motivacija 1

2 Predstavitev domene **2**
2.1 Zančno omrežje 2
2.1.1 Brezžično zančno omrežje 3
2.2 Primeri zančnih omrežij v praksi 5
2.2.1 OLPC 5
2.2.2 Serval Mesh 5
2.2.3 FabFi 5
2.3 Mobilni operacijski sistemi 6
2.4 Možne tehnologije implementacije programa 6
2.4.1 WiFi 6
2.4.2 WiFi Direct 7
2.4.3 Bluetooth 8

3 Metodologija **9**
3.1 Identiteta 9
3.1.1 Ustvarjanje identitete 9
3.1.2 Preverjanje identitete 11
3.2 Povezovanje naprav 12
3.2.1 Asihroni načini prenosa 12
3.2.2 WiFi 12
3.3 Podatki 14
3.3.1 Izgled podatkov 15
3.3.2 Atomarnost podatkov 15
3.3.3 JSON 16
3.3.4 Prenos podatkov 16
3.4 Brisanje sporočil 17
3.5 Rangiranje sporočil 18
3.6 Predviden čas pošiljanja 19

4 Rezultati **20**
4.1 Hitrost vzpostavljanja vroče vstopne točke 20
4.2 Hitrost vzpostavljanja WiFi modula 20
4.3 Hitrost povezovanja na vročo vstopno točko 21

5 Dostopnost **22**
5.1 Licenca izvorne kode 22
5.2 Dostopnost programa izvorne kode 22

6 Zaključek **24**

Literatura **25**

Seznam tabel

4.1 Meritve hitrosti vzpostavljanja vroče vstopne točke 20
4.2 Meritve vzpostavljanja wifi modula 21
4.3 Meritve povezovanja na vročo vstopno točko 21

Seznam slik

2.1 Primer zančnega omrežja: sporočila se prek lokalnih povezav širijo po celotnem povezanem grafu. 3
2.2 Primer topologije WiFi omrežja: mobilne naprave so povezane prek brezžične dostopne točke. 7
2.3 Primer topologije WiFi Direct-a: mobilne naprave so povezane med seboj preko WiFi Direct standarda. 8

3.1 Primer preklapljanja med oddajanjem in sprejemanjem 1: modra vozlišča odjemalci so povezani na oranžna vozlišča - vroče vstopne točke . 14
3.2 Primer preklapljanja med oddajanjem in sprejemanjem 2: vozlišča spremenijo namembnost in se povezave spremenijo 15

1 Zaslonski posnetek 1 29
2 Zaslonski posnetek 2 29
3 Zaslonski posnetek 3 30

Seznam prilog

Zaslonski posnetki testnega programa 28
Pilotni program 29

Seznam kratic

WiFi	Wireless Fidelity
OLPC	One Laptop per child
MANET	Mobile ad hoc network
IEEE	Institution of Electrical and Electronics Engineers
OLPC	One Laptop per child
B.A.T.M.A.N.	Better Approach To mobile Adhoc Networking
API	Application programming interface
BSSID	Basic service set identification
IMEI	International Mobile station Equipment Identity
JSON	JavaScript Object Notation
XML	Extensible Markup Language
RAM	Random access memory
WOT	Web of Trust

1 Uvod

Zaključna projektna naloga predstavlja anonimno izmenjevanje sporočil v zančnem omrežju s pomočjo mobilnih naprav.

Skozi nalogo in posamezna poglavja smo s pomočjo teorije preučili potek izmenjave sporočil v zančnem omrežju ter na praktičnem primeru prišli do rezultatov.

Drugo poglavje nudi pregled področja zančnega omrežja in brezžičnih tehnologij. Dodaten poudarek je na brezžičnih zančnih omrežjih, saj zaključna projektna naloga temelji na njih. Poglavje podrobneje opiše WiFi, WiFi Direct in Bluetooth, saj so vse te tehnologije možne za implementacijo brezžičnega zančnega omrežja na mobilnih napravah.

V tretjem poglavju je opisana metodologija; predstavljeni so algoritmi za delo z identitetami, povezovanje mobilnih naprav v brezžična zančna omrežja, ravnanje s podatki in rangiranje podatkov.

Rezultati WiFi meritev in meritev vzpostavljanja določenih WiFi modulov.

V petem poglavju je predstavljena dostopnost. Pregled najpogostejših izvornih licenc in njihova uporabnost za licenciranje testnega programa ter dostopnost testnega programa.

Šesto poglavje povzame vse pridobljene podatke in zaključi zaključno nalogo.

1.1 Motivacija

V času, ko si življenje brez sodobne tehnologije ne znamo več predstavljati in smo skoraj na vsakem koraku povezani s telekomunikacijskimi storitvami, se dogajajo tudi izredne razmere, ko višje sile (naravne katastrofe, protesti, ...) povzročijo izpad interneta ter ostale komunikacijske infrastrukture.

V nalogi smo želeli raziskati ali v primeru izrednih razmer obstajajo načini, kako izmenjevati sporočila preko zančnih omrežij. Ter ali v primerih, ko gre za občutljive zadeve, lahko pošiljatelj sporočil ostane anonimen.

2 Predstavitev domene

Poglavje opisuje zančna omrežja s poudarkom na brezžičnih zančnih omrežjih ter prednosti in slabosti le-teh.

V poglavju so prav tako prikazani primeri različni implementacij zančnih omrežij in področja njihove uporabe.

Opisuje tudi možne tehnologije za implementacijo anonimnega sporočanje na osnovi zančnega omrežja.

2.1 Zančno omrežje

Zančno omrežje (Mesh network) je način povezave vozlišč v računalniškem omrežju, ki služi za telekomunikacijo napravam povezanim na zančno omrežje in vozliščem med seboj [12] [13].

Za usmerjanje prometa se uporabljata dve metodi usmerjevanja:

- poplavljanje,
- usmerjanje.

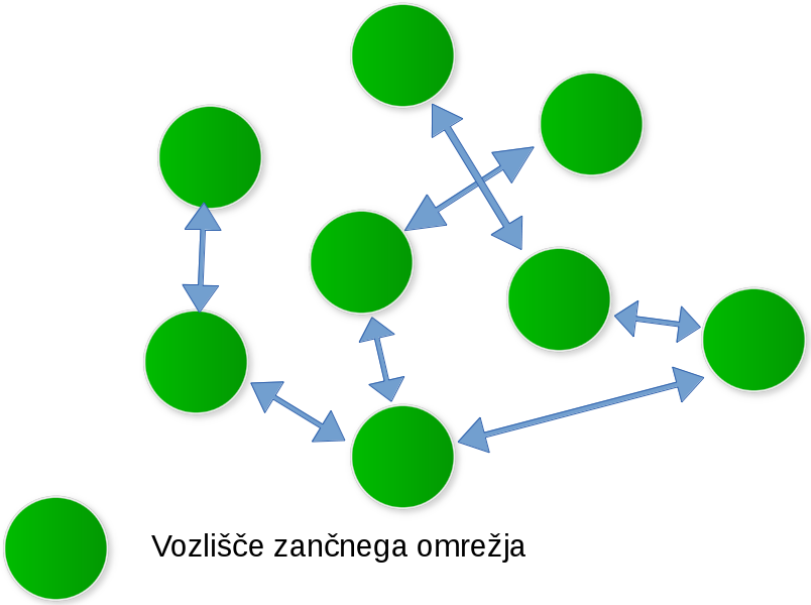
Pri polno povezanem začnem omrežju ni potrebno usmerjati prometa, saj je vsako vozlišče povezano z vsakim. Edini razlog za usmerjanje v takem omrežju je poškodba povezave. Tako omrežje se redko uporablja, saj je iz ekonomskega vidika zelo drago. Večina takšnih omrežij je implementirana z majhno količino vozlišč [12].

Za zanesljivo in hitro usmerjanje služi "Samo celjenje" (Self Healing) [12] [21], ki v zančnem omrežju najde nedelujoče povezave in vozlišča, ter jih odstrani iz usmerjevalnega seznama. Vsako vozlišče dela samo celjenje pri sebi, kar naredi algoritem porazdeljen in učinkovit [12].

Velik problem zančnega omrežja je vstavljanje in popravljanje vozlišč, saj mora biti vsako vozlišče povezano z ostalimi vozlišči [12].

Topologija omrežja omogoča enostavno in lahko odpravljanje napak.

Slika 2.1 prikazuje primer zančnega omrežja.



Slika 2.1: Primer zančnega omrežja: sporočila se prek lokalnih povezav širijo po celotnem povezanem grafu.

2.1.1 Brezžično zančno omrežje

Brezžična zančna omrežja so najbolj pogosta vrsta zančni omrežij [3]. Prvotno so bila razvita za potrebe vojske.

Lahko se povezujejo z različnimi frekvenčnimi pasovi, kar dovoljuje enostavno uporabo hibridnega brezžičnega omrežja.

Zaradi neuporabe kablov je brezžično zančno omrežje občutno cenejše, kot običajno zančno omrežje iz kablov. Zaradi popolne brezžičnosti je tudi lažje vzdrževati vozlišča, saj je dovolj vozlišče samo vgraditi in se bo le-to samo nastavilo. Tej tehnologiji pravimo "Samo nastavljanje" (Self configuration) [21].

V nadaljevanju so opisana pogosta zančna omrežja in pogosti algoritmi za usmerjanje na njih.

IEEE 802.11s

IEEE 802.11s je dopolnitev običajnih 802.11 protokolov za brezžično komunikacijo opisanih v razdelku 2.4.1.

Sam po sebi nima definiranih frekvenc in načina prenosa. Za to je odvisen od IEEE 802.11 a, b, g, n in drugih IEEE 802.11 standardov [26].

IEEE 802.11s definira poleg fizične topologije omrežja, še privzeti usmerjevalni protokol imenovan Hybrid Wireless Mesh Protocol oziroma HWMP. Poleg HWMP protokola je mogoče uporabiti tudi druge usmerjevalne protokole, kot so npr. OLSR in

BATMAN, ki sta opisana v razdelku 2.1.1.

Trenutno je IEEE 802.11s implementiran v operacijskih sistemih Linux od različice jedra 2.6.26 dalje ter pri FreeBSD-u od različice 8.0 dalje.

MANET

MANET oziroma daljše "Mobile ad hoc network" so samo nastavljiva brezžična zančna omrežja. MANET je postal zelo priljubljen z uvedbo WiFi-ja in prenosnih naprav ki uporabljajo le tega.

Glavni cilj MANET-a je samoprilagodljivost. Samoprilagodljivost je izjemnega pomena v MANET-u, saj je v naravi mobilnih naprav, da pogosto menjujejo lokacijo. Brez samoprilagoljivosti bi ob spremembi lokacije mobilne naprave porušile del omrežja, ki je odvisen od le-teh naprav, saj bi usmerjevale tabele vsebovale nedostopna vozlišča [11] [20] [29].

OLSR

OLSR oziroma Optimized Link State Routing Protocol je usmerjevalni protokol za MANET in druga brezžična zančna omrežja, kot je npr. IEEE 802.11s opisana v razdelku 2.1.1. Njegove značilnosti so, da s poplavljanjem na izbranih vozliščih ustvari usmerjevalno tabelo. Dobra lastnost algoritma je, da se z večanjem količine vozlišč ne večajo odvečni podatki usmerjevalne tabele. Slaba lastnost je, da protokol ne zaznava kvalitete povezave [23].

B.A.T.M.A.N.

"Better Approach To Mobile Adhoc Networking" oziroma B.A.T.M.A.N. je usmerjevalni protokol za dinamična zančna omrežja, z namenom zamenjave OLSR protokola. Glavna lastnost B.A.T.M.A.N.-a je porazdeljenost, tako da nobeno vozlišče nima vseh podatkov. S to lastnostjo se zmanjša količina usmerjevalnega prometa v omrežju. Protokol naj bi nudil boljši pristop k mobilnem zančnem omrežju [2] [20].

ZigBee

ZigBee je primer zančnega omrežja zgrajenega na 802.11.15 protokolu. Narejen je za samodejno prepoznavanje in samodejno celjenje omrežja. Torej vozlišča/naprave se lahko dinamično vstavljajo in odstranjujejo iz omrežja, ne da bi to vplivalo na omrežje(omrežje se prilagodi).

ZigBee-jevo omrežje je načrtovano tako, da naprave čim manj pošiljajo podatke in le redko prehajajo iz stanja spanja v stanje pošiljanja, kar močno zmanjša prenos podatkov in zmanjša porabo energije.

Zaradi opisanih lastnosti se ZigBee zančno omrežje pogosto uporablja za zančna omrežja senzorjev [1].

2.2 Primeri zančnih omrežij v praksi

V tem razdelku so opisani primeri uporabe zančnih omrežij.

2.2.1 OLPC

"One Laptop Per Child" oziroma OLPC je projekt poceni prenosnikov za razvijajoči se svet. Projekt je zelo priljubljen saj je do leta 2011 bilo razdeljenih več kot 2 milijona prenosnikov [25].

Posebnost OLPC projekta je, da podpira IEEE 802.11s brezžični zančni povezovalni standard. Prisotnost IEEE 802.11s je pomembna, saj večinoma v razvijajočih se zemljah ni dostopa do interneta [16].

2.2.2 Serval Mesh

Serval Mesh je sistem, ki skuša s pomočjo mobilnih naprav zgraditi zančno omrežje, katero nadomesti navadno mobilno komunikacijsko omrežje z zančnim. Serval Mesh omogoča pošiljanje šifriranih sporočil in vzpostavljanje šifriranih klicev znotraj zančnega omrežja [19].

Serval mesh extender je del projekta Serval project, ki skuša zgraditi poceni napravo za povečevanja dosega zančnega omrežja. Imeli so tudi zbiranje denarja za ustvaritev povečevala dosega, ki pa jim ni uspelo [22].

2.2.3 FabFi

FabFi je odprto-kodni brezžični zančni sistem, ki nudi zastonj dostop do interneta. Zasnovan je za poceni povezovanje. Najdaljša razdalja med vozlišči je 6 km. Ideja FabFija je, da lahko vsak posameznik zgradi vozlišče in razširi območje omrežja. Izdelovanje vozlišča je zelo poceni, saj stane okrog 60 $. Vsa navodila za izdelavo so na voljo na spletni strani projekta. Trenutno FabFi pokriva Afganistan s 50-imi vozlišči in hitrostjo 11.5 Mbps ter Kenijo s 45-imi vozlišči in hitrostjo 30 Mbps. Protokol v omrežju je nek naslednik 802.11 protokolov. Za primer Afganistana in Kenije je to 802.11 g. Usmerjevalnik protokol v tem omrežju je OLSR, ki je opisan v razdelku 2.1.1 [8].

2.3 Mobilni operacijski sistemi

Mobilni operacijski sistemi so operacijski sistemi na mobilnih napravah. Najbolj priljubljeni so s kar Android 74,4 %, iOS 18,2 %, Microsoftovi operacijski sistemi 2,9 %, Research in Motion operacijske sisteme 3 % prodajo v prvem četrtletju leta 2013 [15].

Pri snovanju raziskave predstavljene v nadaljevanju smo izbrali mobilni operacijski sistem Android zaradi mnogih prednosti:

- najbolj priljubljen oziroma operacijski sistem z največjim tržnim deležem saj je bilo v prvem četrtletju prodanih 74,4 % naprav s pred-naloženim mobilnim operacijskim sistemom Android [15],
- je zelo odprt, saj je večina kode odprto-kodna, razen določenih zaprto-kodnih gonilnikov, kar nam omogoča vpogled v varnost in strukturo sistema [5],
- omogoča lažji dostop do sistema preko svojega Application programming interfacea oziroma skrajšano API(Aplikacijskega vmesnika) [4].

2.4 Možne tehnologije implementacije programa

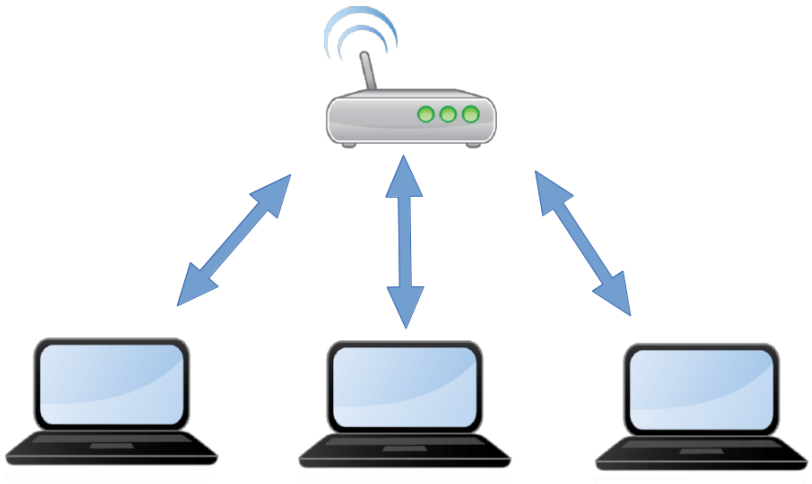
Možne tehnologije za implementacijo programa so vse, ki so dovolj razširjene na mobilnih napravah. V naslednjih razdelkih bomo opisali 3 najbolj razširjene in najbolj primerne tehnologije za implementacijo programa.

2.4.1 WiFi

Wifi je brezžična telekomunikacijska tehnologija z uporabo radijskih valov, ki je nastala leta 1997. Razvija ga "Institute of Electrical and Electronics Engineers" oz IEEE, po katerem je standard dobil ime IEEE 802.11. Slabost prvotnega WiFi protokola je prenos podatkov, saj je bil omejen na 2 Mbps, kar je za današnje potrebe premalo. Zaradi omejenega prenosa podatkov so nastale izboljšave originalnega protokola, sprva 802.11 a in b, kjer je IEEE 802.11 a deloval s hitrostjo 54 Mbps na frekvenci 5 GHz in IEEE 802.11 b 11Mbps na frekvenci 2.4 GHz. Zaradi cene in večjega dometa ter nižje frekvence je IEEE 802.11 b bil prevladujoči standard.

Naslednik 802.11 b protokola je bil 802.11 g, ki je združil prednosti obeh standardov v en standard, hitrost prenosa je bila 54 Mbps, frekvenca pa 2.4 GHz. Združene prednosti so povečale domet, medtem ko je cena bila manjša kot pri IEEE 802.11 a.

Glede na potrebe uporabnikov po dodatni hitrosti in dometu, so razvili naslednika protokola 802.11 g, to je 802.11 n, ki deluje na obeh frekvencah 2.4 GHz in 5 Ghz.



Slika 2.2: Primer topologije WiFi omrežja: mobilne naprave so povezane prek brezžične dostopne točke.

802.11 n uporablja več anten za prenos podatkov, prenos pa je odvisen od modulacije, količine anten ter kodiranja. Največja možna hitrost je 600 Mbps.

Ker se evolucija nenehno spreminja, je nov naslednik protokola 802.11 n, protokol imenovan 802.11 ac, ki pa še ni dokončno definiran.

Povezovanje med WiFi napravami je mogoče le, v kolikor obe napravi podpirata vsaj en skupni protokol. Naprave poleg podprtega protokola, podpirajo še vse njegove prednike, razen 802.11 a, ki je le redko podrt. Ta lastnost naredi WiFi naprave med seboj kompatibilne. Naprava se lahko poveže le na brezžično dostopno točko oz. angleško "wireless access point". Brezžične dostopne točke so ponavadi usmerjevalniki oziroma usmerjevalnikom podobna oprema. Lahko pa s pomočjo programov naredimo vročo dostopno točko s prenosnikom, mobilno napravo oziroma sorodno napravo, saj je le-ta podobna brezžični dostopni točki [27].

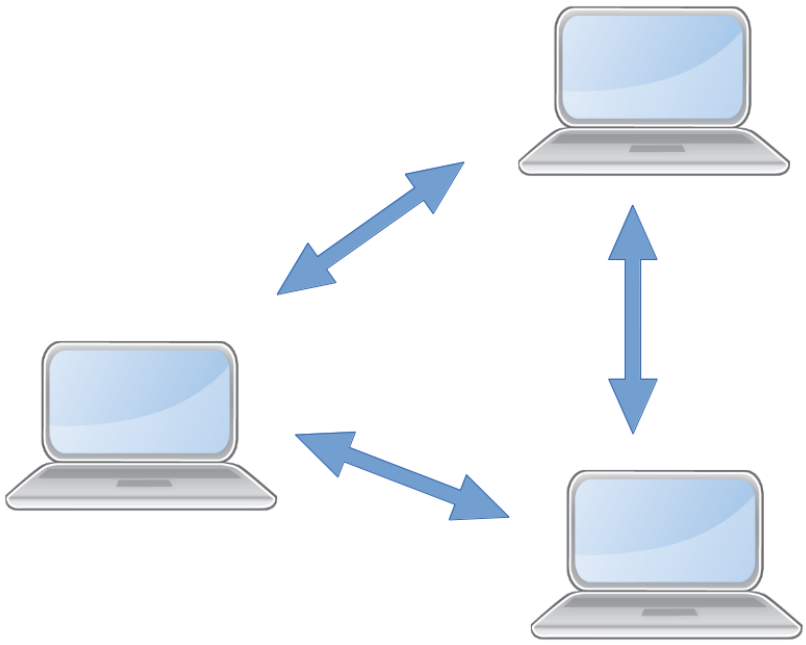
Primer povezovanja naprav v WiFi omrežje je mogoče videti v sliki 2.2.

Glavne prednosti WiFi-ja so velika razširjenost, saj imajo vse mobilne naprave WiFi modul, hiter prenos podatkov do 600Mbps, dober doseg pri frekvenci 2.4 GHz in dokaj hitro vzpostavljanje povezave, kot je prikazano v poglavju 4.

2.4.2 WiFi Direct

WiFi Direct, oziroma prvotno klican WiFi P2P, je standard, ki omogoča povezovanje naprav neposredno med seboj, brez potrebe po brezžični dostopni točki. Primer povezovanja WiFi Direct naprav je mogoče videti na sliki 2.3.

Hitrost WiFi Direct je definirana s hitrostjo 802.11 protokolov, saj promet poteka med napravami, enako kot pri 802.11 protokolih.



Slika 2.3: Primer topologije WiFi Direct-a: mobilne naprave so povezane med seboj preko WiFi Direct standarda.

Wifi Direct je popolnoma kompatibilen s trenutnimi WiFi protokoli, saj nudi povezovanje na običajna 802.11 omrežja. Wifi naprava zazna WiFi Direct napravo, kot brezžično dostopno točko.

WiFi Direct je podrt le od Androidove različice 4.0 dalje, kar ga ne naredi ustreznega za implementacijo testneega programa. Kompatibilnost z WiFi-jem naredi WiFi Direct dobrega kandidata za nadgradnjo [28].

2.4.3 Bluetooth

Bluetooth je ena najstarejših tehnologij za povezovanje mobilnih naprav. Leta 1994 je Ericsson ustvaril Bluetooth.

Bluetooth deluje na frekvencah od 2400 do 2480 MHz. Poraba energije je velika prednost bluetooth standarda, saj le-ta porablja veliko manj energije kot njegov tekmec WiFi. Dobra lastnost bluetootha je možnost pošiljanja več napravam hkrati in biti povezan na več naprav hkrati, kar lahko naredi dobro zančno omrežje. Njegova velika slabost je čas vzpostavljanja, saj je le-ta večji kot pri WiFi-ju. Slabost napram WiFi-ju sta tudi prenos podatkov ter slab domet [6].

Upoštevajoč prednosti in slabosti Bluetooth-a, je Bluetooth primeren za naprave, ki so več časa povezane med seboj in pošiljajo malo podatkov. Raziskava z implementirano pilotsko aplikacijo, ki je predstavljena v poglavjih 5 in 3 zahteva lastnosti, ki jih bluetooth ne omogoča.

3 Metodologija

Poglavje predstavlja pomen identitete v zančnem omrežju ter delo z identitetami. V nadaljevanju so predstavljene metode povezovanja naprav v brezžičnih zančnih omrežjih, podatki ter ravnanje s podatki. Sledi predstavitev brisanja sporočil, rangiranja sporočil in predviden čas pošiljanja sporočil.

3.1 Identiteta

Anonimnost je ključnega pomena, saj preprečuje odkrivanje pošiljatelja v sporočilu. Žal zaradi preprečevanja ponavljanja sporočil, prepisovanja sporočil, brisanja sporočil in drugih možnih nadgradenj, kot sta npr. rangiranje sporočil in prednostno prikazovanje sporočil potrebujemo identiteto pošiljatelja. Ker ima pošiljatelj identiteto ne moremo več zagotoviti popolne anonimnosti. Iz tega razloga je identiteta psevdo-anonimna.

S psevdo-anonimnostjo dosežemo nezmožnost pridobivanja identitete pošiljatelja in zmožnost povezovanja s psevdo-avtorjem, kar pomeni, da vemo da je ista oseba poslala sporočilo, ne vemo pa kdo je poslal sporočilo, saj so nam podatki psevdo-avtorja neznani.

Načini ustvarjanja identitete in preverjanje le-te so opisani v naslednjih podpoglavjih.

3.1.1 Ustvarjanje identitete

Ustvarjanje identitete je mogoče na več načinov, vsem je skupno to, da morajo ustvariti psevdo-anonimno identiteto, ki je ob vsakem zagonu programa drugačna. Namreč identifikator sporočila (številka ki pove, katero po vrsti je sporočilo, ki smo ga napisali) se ob vsakem zagonu po-nastavi na začetno vrednost, ki je enaka 0. Razlog za to je prepisovanje sporočil, saj v primeru ponovnega zagona programa identifikator začne z začetno vrednostjo in pošilja nova sporočila z napačnim identifikatorjem. Kar bi na napravah, ki bi to sporočilo sprejele, sprožilo nedefinirano vedenje (npr. že sprejeto sporočilo bi spremenilo svojo vsebino ali pa se nova sporočila ne bi prikazala, ker sporočila s tem identifikatorjem že obstajajo in se novo sporočilo ne bi shranilo). Zaradi tega smo izbrali ustvarjanje nove identitete vsakič, saj se s tem načinom najlažje

izognemo zgornjim težavam. Izognemo se tudi težavi, kjer bi nekdo odtujil telefon določene osebe, vklopil aplikacijo in pošiljal sporočila v imenu te osebe.

Eden izmed možnih načinov za ustvarjanje identitete je uporaba BSSID (Basic service set identification) vrednosti. Iz imena izhaja, da gre za unikatni identifikator za vsako brezžično napravo ter naključni del, ki ga ustvarimo glede na trenutni čas na mobilni napravi. Te dve vrednosti lahko zgostimo s poljubnim zgoščevalnim algoritmom, tako dobimo skoraj unikatni identifikator. Možnosti, da bi dve napravi v istem zančnem omrežju imele enak zgoščeni zapis so dovolj majhen [10], da lahko to možnost zanemarimo.

Izboljšava zgornje metode je možna z uporabo certifikatov. Zgornjo zgoščeno vrednost, ki smo jo dobili iz BSSID-ja in naključne vrednosti uporabimo, kot ime certifikata, in ta certifikat ustvarimo. Tako vsakemu sporočilu ustvarimo podpis, ki je unikaten glede na certifikat naprave in podpisano sporočilo. Ta podpis dodamo na konec sporočila. Poleg sporočil moramo dodati še javni šifrirni ključ. Te metode bodo opisane v naslednjem podpoglavju.

Možna alternativa BSSID-u za ustvarjanje zgoščene vrednosti je IMEI vrednost (International Mobile Station Equipment Identity), ki je unikatna za vsako mobilno napravo. Slabost BSSID-ja je ta, da je obvezno ob trenutku pridobivanja BSSID-ja imeti usposobljen WiFi modul. Težava se pojavi v nekaterih situacijah, saj ima WiFi določen čas vzpostavljanja, ki je izmerjen v poglavju 4. V kolikor pošiljatelj napiše sporočilo, preden je WiFi vpostavljen, mu program ne more dodeliti identite. V tem pogledu je boljši IMEI, saj imamo do IMEI vrednosti vedno dostop. Ima pa IMEI tudi svojo slabost, saj potrebuje dodatno sledečo pravico v mobilnem operacijskem sistemu android.

```
<uses-permission android:name="android.permission.READ_PHONE_STATE" />
```

```
private void createIdentity() {
        int randomInt = random.nextInt();
        WifiInfo wifiInfo = wifiManager.getConnectionInfo();

        if (wifiInfo.getBSSID() != null) {
                identity = wifiInfo.getBSSID() + randomInt;
        } else {
                identity = telephonyManager.getDeviceId()
                                        + randomInt;
        }

        byte[] byteArray = null;
```

```
try {
        byteArray = identity.getBytes("UTF-8");
} catch (UnsupportedEncodingException e1) {
        e1.printStackTrace();
}

MessageDigest md = null;
try {
        md = MessageDigest.getInstance("MD5");
} catch (NoSuchAlgorithmException e) {
        e.printStackTrace();
}
byte[] digest = md.digest(byteArray);

identity = new String(digest);
}
```

Za potrebe programa smo uporabili zgoščevanje BSSID-ja in naključne številke. V kolikor BSSID ne bo na voljo, bomo uporabili IMEI in naključno številko, kot je razvidno iz priložene izvorne kode [13].

3.1.2 Preverjanje identitete

V kolikor želimo opravljati več storitev, kot je samo prikazovanje podatkov, je preverjanje identitete ključnega pomena. Dober primer je rangiranje, brez zmožnosti točne in zagotove avtentifikacije določene naprave/sporočevalca, ne moremo pravilno glasovati, saj ne moremo biti prepričani v avtentičnost sporočila. Glasovanje in ocenjevanje sporočil je opisano kasneje v razdelku 3.5.

Preverjanje identitete pri načinu, kjer imamo samo zgoščeno vrednost določenih parametrov, je težavno, saj lahko nepridiprav ob spreminjanju izvorne kode, oziroma obratnem inženiringu, namesto ustvarjanja zgoščene vrednosti, sam vnese zgoščeno vrednost. Zaradi tega ne moremo povezovati sporočil z avtorjem. V primeru, ko nas ne zanima kdo je poslal sporočilo, ampak samo sporočilo, nas ta napad ne obremenjuje.

Preverjanje identitete pri načinu s certifikati je bolj zapleteno. Mobilna naprava mora poleg sporočil pošiljati tudi svoj javni ključ. Naprave lahko z javnim ključem pošiljateljeve naprave preverijo avtentičnost sporočila s pripadajočim algoritmom preverjanja pristnosti. V kolikor je rezultat pozitiven, je sporočilo pristno. V nasprotnem primeru se sporočilo zavrže, saj je podpis neveljaven oziroma spremenjen. Globlje se glede podpisovanja sporočil in preverjanja le-teh v tej zaključni nalogi ne bomo spuščali,

saj to ni namen naloge.

3.2 Povezovanje naprav

Način povezovanja naprav je pomemben, saj narekuje način izmenjave podatkov med napravami. Načine povezovanja delimo v 2 skupini: sinhroni in asinhroni.

Sinhroni je način izmenjave podatkov, kjer je lahko mobilna naprava povezana na poljubne mobilne naprave v dometu in se na to napravo povezujejo poljubne mobilne naprave.

Asinhron je način, kjer je naprava povezana na poljubno mobilno napravo, in je ta mobilna naprava povezana samo s povezanimi napravami.

V sinhrono skupino spadajo WiFi Direct, in Bluetooth. V asinhrono pa WiFi.

3.2.1 Asihroni načini prenosa

Z WiFi Direct, opisanim v razdelku 2.4.2 je mogoče vzpostaviti popolno brezžično zančno omrežje. Glavna prednost uporabe WiFi Direct-a je enostavno pošiljanje in sprejemanje sporočil, medtem ko je slabost WiFi Direct-a v tem, da na veliki količin naprav ne deluje oz. ne deluje pravilno. Zaradi te lastnosti ga ni mogoče uporabiti kot glavni prenosni medij pri projektu. S tehnologijo WiFi-Direct se je mogoče priključiti tudi na vročo vstopno točko, kar pomeni, da je tehnologija združljiva z WiFi-jem. Ta lastnost omogoča ustvarjanje heterogenega zančnega omrežja sestavljenega iz WiFi Direct in običajnih vozlišč. Tako omrežje bi imelo vse prednosti obeh tehnologij.

Z Bluetooth-om, opisanim v razdelku 2.4.3, je podobno kot s tehnologijo WiFi Direct, mogoče vzpostaviti popolno zančno omrežje, saj se vsaka naprava lahko poveže na vsako napravo. Dobra lastnost Bluetooth-a je pošiljanje vsem povezanim napravam naenkrat (multicast). Ta tehnologija poenostavi komunikacijo, saj ni več potrebno vsaki napravi posebej pošiljati podatke. Prednost multicasta je hitrejši prenos podatkov, kot je običajni prenos pri posamezni napravi. Bluetooth ima tudi nekaj slabih lastnost, med drugim tudi kratek domet. Domet je občutno manjši od WiFi-ja, prenos podatkov je zelo majhen, v primeru uporabe multicasta je še vedno manjši od prenosa podatkov WiFi-ja/WiFi Direct-a.

3.2.2 WiFi

WiFi je kot prevladujoči povezovalni standard na mobilnih napravah skoraj popolna rešitev za implementacijo komunikacije pri tem projektu. WiFi-jeva prednost so kot že omenjeno razširjenost, saj je vsebovan na skoraj vsaki napravi, dober domet in hiter

prenos podatkov. Žal ima tudi WiFi nekaj slabih lastnosti. Androididova implementacija WiFi-ja ne dovoljuje istočasnega povezovanja na vročo vstopno točko in oddajanje vroče vstopne točke, kot je že opisano v razdelku 2.4. Zaradi tega je potrebno izmenjevati med oddajanjem vroče vstopne točke in povezovanjem na le-te. V naslednjih podpoglavjih so opisane metode ravnanja z WiFi-jem za doseg prenosa sporočil med napravami.

Vroča vstopna točka

Zaradi Googlovih odločitev je ustvarjanje vroče vstopne točke namenjeno samo operacijskemu sistemu Android in programom sestavljavcev mobilnih naprav zasnovanih z Androidom. Te metode so privatne za razliko od javno dostopnih, so pa še vedno vsebovane v APIju(aplikacijskem vmesniku). Tukaj nam na pomoč vstopi JAVA reflection, ki je del programskega jezika JAVA-e, s katerim lahko spreminjamo in upravljamo z instanco objekta. To nam omogoča dostop do privatnih delov objekta, torej do omogočanja vroče vstopne točke, kot je to prikazano v sledeči kodi [18].

```
wifiManager.setWifiEnabled(false);
wifiConfig = new WifiConfiguration();
wifiConfig.SSID = AP_NAME;
wifiConfig.allowedKeyManagement.set(KeyMgmt.NONE);
Method method;
try {
        method = wifiManager.getClass().
         getMethod("setWifiApEnabled",
           WifiConfiguration.class, boolean.class);
        method.invoke(wifiManager, wifiConfig, true);
} catch (Exception e) {
        Log.e(TAG, "", e);
}
```

Povezovanje na vroče vstopne točke

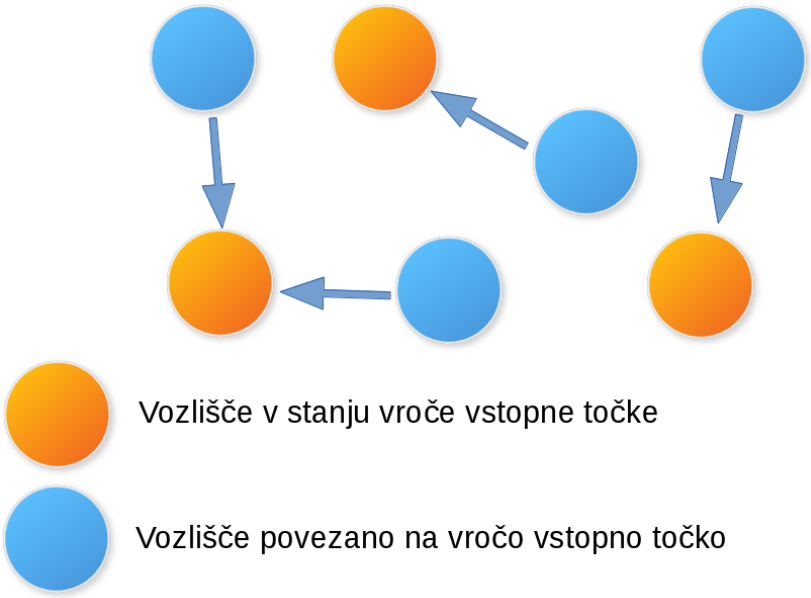
Za izmenjevanje sporočil se naprava mora povezati na pravo vročo vstopno točko. Za povezovanje na vročo vstopno točko mora naprava najprej preiskati omrežje za proste vroče vstopne točke. Nato iz seznama prostih vročih vstopnih točk odstraniti neveljavna omrežja, torej omrežja, ki v svojem imenu ne vsebujejo ime programa in omrežja, ki niso prosta. Iz pridobljenega seznama se naprava poskuša povezati na vsako vročo vstopno točko posebej. V kolikor ji uspe, prične s prenosom sporočil, kot je opisano v razdelku 3.3.4. V kolikor pa ji ne uspe, se poveže na naslednjo vročo

vstopno točko na seznamu. Po končanem povezovanju program preide v stanje vroče vstopne točke, kot je opisano v razdelku 3.2.2.

Preklapljanje med sprejemanjem in oddajanjem

Preklapljane med sprejemanjem WiFi signala in oddajanjem vroče vstopne točke je potrebno zaradi načina implementacije WiFi-ja v operacijskem sistemu Android.

Intervali povezovanja na vročo vstopno točko so deloma naključni, saj v primeru, da mobilne naprave istočasno vključimo, se nobeno zančno omrežje ne bo moglo vzpostaviti. Zaradi tega je preklapljanje med načini naključno. Del časa je definiran s časom vzpostavljanja vroče vstopne točke in vzpostavljanja WiFi-ja ter dodatnim minimalnim časom za pošiljanje sporočil. Časi vzpostavljanja vroče vstopne točke in vzpostavljanje WiFi-ja so opisani v poglavju 4.

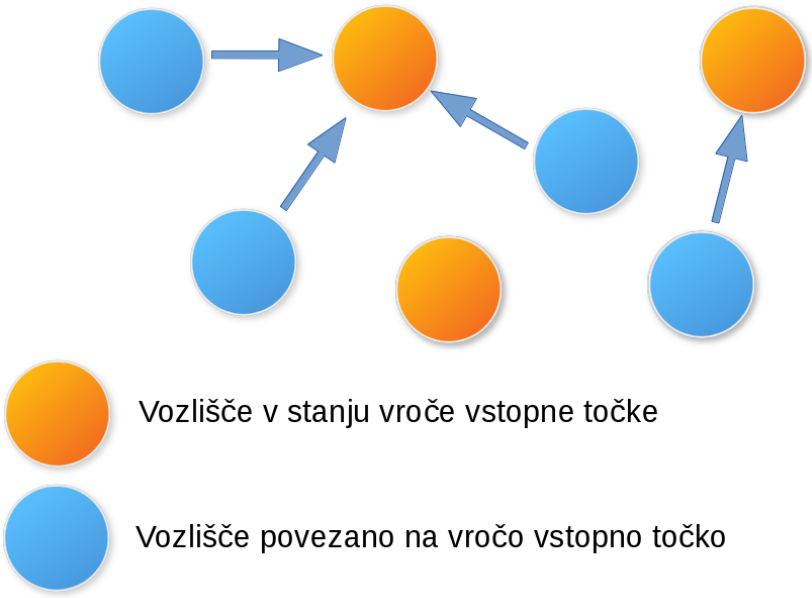


Slika 3.1: Primer preklapljanja med oddajanjem in sprejemanjem 1: modra vozlišča odjemalci so povezani na oranžna vozlišča - vroče vstopne točke

Primer preklapljanja med oddajanjem in sprejemanjem je mogoče videti v sliki 3.1 in 3.2.

3.3 Podatki

V tem poglavju so opisani podatki, njihov izgled, dostop do njih, način kodiranja za pošiljanje ter na koncu še prenos podatkov.



Slika 3.2: Primer preklapljanja med oddajanjem in sprejemanjem 2: vozlišča spremenijo namembnost in se povezave spremenijo

3.3.1 Izgled podatkov

Za predstavitev sporočil potrebujemo vsaj 3 podatke: identifikator sporočila, zaporedno številko sporočila in vsebino sporočila.

V primeru implementacije certifikatov, kot je opisano v razdelku 3.1.1, bi potrebovali še podpis v sporočilu in dodatne podatke z javnimi ključi naprav.

Za implementacijo rangiranja sporočil, bi potrebovali vse podatke, ki so potrebni za običajno predstavitev sporočila ter identitete s certifikati. Dodatni podatki so podatki, kjer bi s pomočjo ocen oziroma rangiranja ostalih mobilnih naprav, dobili rang podatkov. Več o rangiranju podatkov je napisano v razdelku 3.5.

Identifikator naprave oziroma pošiljatelja potrebujemo, saj le tako se lahko izognemo problemu atomarnosti podatka.

3.3.2 Atomarnost podatkov

Atomarnost podatkov je pomembna, saj se program izvaja vzporedno. Najlepši primer tega je, kjer program bere sporočila in istočasno uporabnik vpiše novo sporočilo. Podoben primer se zgodi, ko program dobi sporočila in preveri ali ima ta sporočila že shranjena ter uporabnik vpiše podatke v program. Dodaten primer je, ko program vpiše nove podatke pridobljene iz zančnega omrežja istočasno, ko uporabnik vpiše novo sporočilo.

Problem je mogoče rešiti na več načinov. Eden izmed načinov je dovoljevanje samo

ene operacije naenkrat, kar pomeni, da lahko samo en proces dostopa do podatkov oziroma jih spreminja. Ta način je najbolj omejen.

Boljši način za reševanje tega problema je, problem bralca in pisalca. Rešitev je v tem, da lahko vsi berejo, ko nobeden ne piše. Piše pa lahko samo eden naenkrat, ko noben ne bere. Z uporabo tega algoritma se poveča prepustnost podatkov, in posledično poveča hitrost programa [24].

Androidov programski jezik (JAVA) ima to že implementirano [9], kar je mogoča nadgradnja, saj trenutno ima testni program svojo implementacijo rešitve.

3.3.3 JSON

Podatke za potrebe sporočanja zakodiramo v standardizirane formate, da jih lahko naprava razvozla. Podatke lahko zakodiramo na poljuben način. Med najbolj priljubljenima spadata XML oziroma daljše "Extensible Markup Language" in JSON "JavaScript Object Notation", kjer podatke zakodiramo na človeku prijazen način. V testnem programu smo uporabili JSON, saj je uporaba hitrejša od XML-ja [14].

Podatke smo v JSON-u predstavili tako, da smo vsa sporočila vstavili v eno JSON polje oz. ang. JSON array. Sporočila smo predstavili z atributi identifikatorja, zaporedne številke sporočila in vsebine sporočila. V primeru uporabe certifikatov, kot je opisano v razdelku 3.1.1, bi potrebovali še dodaten atribut za certifikat in dodat JSON array za javne ključe posameznih naprav. Podobno velja za rangiranje.

3.3.4 Prenos podatkov

Prenos podatkov med napravami je odvisen od uporabljene tehnologije za brezžično komunikacijo. Za vse načine povezovanja (opisane v razdelku 3.2), razen za WiFi, velja da lahko mobilna naprava oziroma vozlišče v omrežju istočasno pošilja in oddaja podatke.

V primeru WiFi-ja to ne velja, saj kot je že bilo omenjeno v 3.2, je v Androidovi implementaciji WiFi-ja nemogoče istočasno prejemati in oddajati signal. Podobno velja za podatke. Vroča vstopna točka ne more sprejemati podatkov. Iz tega razloga mora vroča vstopna točka poslati vsem povezanim napravam, vse podatke, katere vsebuje. Razlog za to, je neobstoj povratne informacije. Način za optimizacijo pošiljanja je hraniti podatke o že poslanih sporočilih na določeno mobilno napravo. Ta način je lahko problematičen, v kolikor mobilna naprava resetira program. V tem primeru mobilna naprava nima več podatkov in ji nobeno vozlišče ne more več posredovati starih podatkov, saj imajo shranjeno, da ta naprava že vsebuje te podatke.

3.4 Brisanje sporočil

Brisanje sporočil postane potrebno, ko se medij za shranjevanje podatkov polni, oziroma dosega maksimalno kapaciteto. Za ta problem obstaja več rešitev.

Najenostavnejši način reševanja prevelike količine sporočil, je brisanje sporočil. Pomembno je paziti katera sporočila se brišejo. Pri tem načinu poznamo več algoritmov, na podlagi katerih izvedemo brisanje. Pomembno pri brisanju je pustiti identifikatorje izbrisanih sporočil, saj se bodo v nasprotnem primeru, izbrisana sporočila ponovno pojavila.

Brisanje na podlagi starosti, je algoritem kjer program briše vsa sporočila starejša od določenega števila sekund oziroma minut dalje. Izboljšave tega algoritma so možne z ne brisanjem svojih sporočil, kar pomeni, da bi program v nedogled pošiljal svoja sporočila in onesnaževal oziroma zasičeval omrežje s starimi sporočili.

Brisanje na podlagi ocene je drugi način brisanja sporočil, kjer program, glede na oceno sporočil, briše sporočila z majhno oceno. Za uporabo tega algoritma je potrebno implementirati certifikate, kot je opisano v razdelku 3.1.1 in glasovanje, kot je opisano v razdelku 3.5. Prednost tega algoritma je, da se slaba sporočila brišejo in dobra še naprej pošiljajo po omrežju, kar poveča možnost širjenja dobro ocenjenih sporočil.

Težava zgornjih dveh algoritmov je, da so v nekaterih primerih nepravični. Na primer brisanje po starosti, je nepravično do sporočil z dobro vsebino, saj bodo te vsebine izbrisane preden dosežejo oddaljena vozlišča. Podobno velja za algoritem, ki temelji na ocenah, določena sporočila ne bodo nikoli izbrisana iz omrežja. Zaradi teh težav je bolje uporabiti mešanico obeh algoritmov. Torej algoritem, kateri briše sporočila glede na čas in oceno, le s takim algoritmom lahko dosežemo najbolj pravilno brisanje, ob pravičnem glasovanju, opisano v razdelku 3.5.

Drugi način za reševanje težave prevelike količine podatkov je stiskanje podatkov. S stiskanjem podatkov, lahko pridobimo prostor tako, da stisnemo del, kjer se nahaja sporočilo, saj so ostali deli naključni, in jih ni mogoče enostavno stiskati. Stiskanje nenaključnih besed ni težavno, saj imajo dokaj nizko entropijo. Zaradi te lastnosti se pridobi nekaj prostora.

Še en način reševanja težave brez brisanja, je premeščanje podatkov. Vsi podatki do sedaj so bili shranjeni v sistemski pomnilnik(RAM), ki pa je omejen in veliko manjši kot podatkovni sistem naprave oziroma spominska kartica. Izbiranje podatkov, ki bodo shranjeni na nosilec lahko izvedemo s poljubnim algoritmom za brisanje, z razliko, da te podatke ne brišemo, ampak jih samo premestimo na spominsko kartico. Mogoče je tudi združiti brisanje in premeščanje v primeru, ko je podatkov preveč in bi napolnili RAM in nosilec. Zaradi pogosto uporabljanja aktivnih podatkov, je bolje te podatke hraniti v pomnilniku, saj je dostop do podatkovnega nosilca počasnejši.

3.5 Rangiranje sporočil

Z rangiranjem sporočil je mogoče doseči neenakopravnost sporočil.

Neenakopravnost sporočil pomeni, da so nekatera sporočila bolj pomembna od drugih, kar lahko dosežemo z glasovanjem posameznih sporočil.

Prednosti rangiranja je mnogo, med najbolj pomembne spadajo izboljšano brisanje sporočil, kot je opisano v 3.4 in prioritetno prikazovanje sporočil. S prioritetnim prikazovanjem sporočil je v teoriji mogoče prikazovati le pomembna in resnična sporočila, saj se sporočila s slabim rangom sploh ne prikažejo.

Za implementacijo rangiranja je potrebno implementirati identiteto pošiljateljev s certifikati, kot je opisano v 3.1. Identiteta s certifikati je potrebna, saj za pravilno rangiranje potrebujemo identiteto, ki je ni mogoče ponarediti. Drugi načini ustvarjanja identitet ne more zagotoviti identiteto brez ponarejanja.

Rangiranje je mogoče implementirati na več načinov oziroma nivojev.

Eden od načinov implementacije je, da se samo sporočila ocenjujejo. Dobra stran tega načina je, da je za uporabnika dokaj enostavno, saj ima ob sporočilu še gumb za pozitivno in negativno glasovanje.

Pri tem načinu je mogoče rang izračunati na 2 načina.

Prvi način, je način kjer se rang izračuna samo na podlagi povezanih vozlišč. Je dokaj enostaven za implementacijo, ima pa slabost, da se upošteva samo lokalno mnenje množice, kar je slabo v primeru, ko je mobilna naprava obkrožena s slabimi oziroma sumljivimi vozlišči.

Drugi način, je izračunati rang glede na vsa vozlišča v množici. V tem primeru se izognemo slabemu rangiranju lokalne množice. Ta način pa ima slabo lastnost - težjo implementacijo, saj program potrebuje podatek o vseh vozliščih v omrežju in podatek o njihovih glasovanjih za posamezna sporočila. To bi močno vplivalo na zaseden prostor. Reševanje tega problema je opisano v 3.4.

Slabost implementacije, kjer se izmenjujejo samo sporočila je, da je vsako vozlišče enakovredno, kar je problematično, kadar določena vozlišča nepravilno glasujejo in znižujejo rang pomembnih sporočil.

Drugi način implementacije je, da se poleg sporočil ocenjujejo tudi avtorji sporočil. Način je dokaj podoben prejšnjemu, z razliko, da se tukaj ocenjuje tudi avtor. Avtorja lahko ocenjujemo posredno oziroma neposredno. Pri neposrednem ocenjevanju ocenjujemo avtorja in sporočilo posebej. Drugi način, je način kjer ocena vsakega sporočila vpliva na oceno avtorja. Načini implementacije tega načina so enaki kot za implementacijo prvega načina. Slabosti in prednosti so prav tako enake.

Tretji način implementacije je poleg ocenjevanja sporočil in avtorjev, ocenjevanje oziroma rangiranje vozlišč. Ocenjevanje avtorjev in sporočil je pri tej metodi izvedeno,

kot v prejšnjih načinih. Rangiranje vozlišč je možno implementirati na več načinov. En način je, da vsako vozlišče posebej rangiramo, glede na to koliko zaupamo vozlišču. To je najboljši način, saj je v tem primeru rang sporočila najbolj pravilen. Problem pri tem načinu je implementacija, saj tega ni mogoče implementirati. Drugi način je, da glede na rang avtorja vozlišče, dobi vozlišče utež s katero ocenjujemo ocene. Ta princip rangiranje se imenuje "Web of Trust" oziroma skrajšano WOT [7].

V testnem programu rangiranje ni uporabljeno, saj ne služi namenu testnega programa.

3.6 Predviden čas pošiljanja

Predviden čas pošiljanja podatkov je zelo pomemben, saj nam pove v povprečju koliko časa potrebuje sporočilo da doseže povezano napravo oziroma drugo napravo.

Čas pošiljanja je odvisen od naslednjih dejavnikov:

- hitrost vzpostavljanja WiFi modula, meritve prikazane v 4.2,
- hitrost vzpostavljanja vroče vstopne točke, meritve prikazane v 4.1,
- hitrost povezovanja na vročo vstopno točko, meritve prikazane v 4.3,
- verjetnost, da je pošiljajoča naprava vroča vstopna točka in prejemajoča naprava odjemalec,
- hitrost WiFi komunikacije, ki je definirana z največjim skupnim IEEE 802.11 protokolom.

4 Rezultati

V tem poglavju so zbrani rezultati meritev hitrosti vzpostavljanja vroče vstopne točke, vzpostavljanja WiFi modula, in hitrosti povezovanja na vročo vstopno točko.

Vse meritve so bile opravljene na mobilni napravi Sony Ericsson Xperia Arc S z različico Android jedra 4.1.2. Čas je bil merjen z ročno uro.

4.1 Hitrost vzpostavljanja vroče vstopne točke

Hitrost vzpostavljanja vroče vstopne točke je pomembna, saj vpliva na skupni čas prenosa podatkov.

Test je bil opravljen tako, da se je meritev začela ob vklopu vroče vstopne točke na mobilni napravi, končala se pa je, ko se je pojavila ikona za deljenje povezave preko vroče vstopne točke.

Povprečen merjen čas je bil 2.93 s s standardnim odklonom 0.18886.

Tabela 4.1: Meritve hitrosti vzpostavljanja vroče vstopne točke

Meritev	1	2	3	4	5	6	7	8	9	10
Rezultat	2,8 s	3 s	3,2 s	2,9 s	2,7 s	2,6 s	3,1 s	3,1 s	3 s	2,9 s

4.2 Hitrost vzpostavljanja WiFi modula

Hitrost vzpostavljanja WiFi modula je podobno kot pri vzpostavljanju vroče vstopne točke, zelo pomembna pri skupnem času prenosa podatkov.

Test je bil opravljen tako, da se je meritev začela ob vklopu WiFi modula, končala pa se je, ko je indikator ob WiFi ikoni prikazal vklopljeno.

Povprečen čas meritev je bil 1.92 s s standardnim odklonom 0.13166.

Tabela 4.2: Meritve vzpostavljanja wifi modula

Meritev	1	2	3	4	5	6	7	8	9	10
Rezultat	1,8 s	2,0 s	2,1 s	1,9 s	1,8 s	1,9 s	2,1 s	1,7 s	1,9 s	2,0 s

4.3 Hitrost povezovanja na vročo vstopno točko

Poleg časa povezovanja na vročo vstopno točko, se šteje tudi čas vzpostavljanja WiFi modula. Da bi pridobili pravilen čas, je potrebno od tega odbiti še povprečen čas vzpostavljanja WiFi modula.

Test je bil opravljen tako, da se je meritev začela ob vklopu WiFi modula, končala pa se je, ko se je prikazala ikona za povezavo z brezžično dostopno točko.

Povprečen čas je bil 7.95 s, od tega je potrebno odšteti še povprečen čas za vzpostavljanje modula, tako da dobimo povprečje 6.03 s. Standardni odklon je 0.20138.

Tabela 4.3: Meritve povezovanja na vročo vstopno točko

Meritev	1	2	3	4	5	6	7	8	9	10
Rezultat	8,2 s	8 s	7,8 s	8,1 s	7,7 s	8,0 s	7,9 s	7,6 s	8,2 s	8,0 s

5 Dostopnost

V tem poglavju so opisane najbolj pogoste licence izvorne kode ter njihove lastnosti. Opisana je tudi dostopnost izvorne kode.

5.1 Licenca izvorne kode

Licenca izvorne kode, odprto-kodnega programa je pomembna, saj nam pove v katere namene in kako lahko uporabimo to kodo.

Med najbolj pogoste licence spadajo

- copyleft,
- permissive.

Pod izrazom permissive spadajo BSD(Berkley Software Distribution) in MIT Licence(Massachusetts Institute of Technology) licence in njim podobne licence. Te licence so najmanj zahtevne in so tudi najkrajše (majhno število točk). Permissive licence dovoljuje dodajanje izvorne kode, ki ni pod isto licenco. Dovoljuje tudi spreminjanje licence izvorne kode, kar naredi te licence dokaj primerne za odprto kodne projekte s ciljem čim večjega trga.

Pod copyleft spadajo GPL(General Public Licence) in podobne licence. Copyleft za razliko od permissive licenc, ne dovoljuje spreminjanja licence izvorne kode in dodajanja izvorne kode pod drugo licenco, kar zelo omeji uporabo kode [17].

Za projekt smo izbrali licenco BSD, saj dopušča največjo svobodo uporabnikom.

5.2 Dostopnost programa izvorne kode

Izvorna koda programa in izvršilna datoteka programa sta javno dostopna na spletnem portalu githubu prek spletne povezave `https://github.com/aleksandar0todorovic/AlternateWifiMeshMessaging`.

Github je izbran za gostovanje izvorne kode, ker je odprt za javnost(vsakdo lahko pobere izvorno kodo) ter zastonj za odprto kodne projekte. Ker je zasnovan na osnovi git-a, lahko tudi vsak posameznik začne svojo vejo programa in se tako program razvija dalje.

Projekt je potrebno klonirati z orodjem git na sledeči način:

```
git clone https://github.com/aleksandar0todorovic/AlternateWifiMeshMessaging
```

6 Zaključek

V zaključni nalogi smo si najprej ogledali osnovne pojme zančnih omrežij, nato smo si ogledali naprednejše pojme ustvarjanja zančnega omrežja, delo s podatki in identitetami. V nadaljevanju smo si ogledali rezultate meritev osnovnih WiFi operacij, ki vplivajo na hitrost delovanja pilotnega programa. Za tem smo si ogledali licenco in dostopnost pilotnega programa.

Začetna predpostavka, da je mogoče na osnovi zančnega omrežja zgraditi anonimni sporočilni sistem se je izkazala za pravilno. V pilotnem programu smo pokazali, kako je mogoče na preprost način ustvariti anonimni sporočilni sistem. Predstavljene so bile tudi izboljšave oziroma nadgradnje metod uporabljenih v pilotnem programu.

Trenutna implementacija anonimnega sporočilnega sistema na osnovi zančnega omrežja v pilotnem programu je zadovoljiva za testne primere. Za normalno uporabo pa pilotni program ni primeren, saj je prepočasen in ne omogoča vseh potrebnih funkcij, kot je npr. rangiranje. S predstavljenimi izboljšavami je mogoče pilotni program izboljšati toliko, da je tudi primeren za uporabo v resničnem svetu.

Brezžična zančna omrežja so prihodnost za področja, kjer je nemogoče oziroma predrago ustvariti navadno omrežje, kjer je telekomunikacija onemogočena ali pa kjer je potrebno komunicirati anonimno brez nadzora oblasti.

Literatura

[1] S. Ahamed, The role of zigbee technology in future data communication system, *Journal of Theoretical and Applied Information Technology* (2009), 129 – 135. *(Citirano na strani 5.)*

[2] C. Aichele in S. Wunderlich, A. Neumann et al., Better Approach To Mobile Ad-hoc Networking (B.A.T.M.A.N.), *The Wall Street Journal* (2008), . *(Citirano na strani 4.)*

[3] E. Alotaibi in B. Mukherjee, A survey on routing algorithms for wireless Ad-Hoc and mesh networks, *Computer Networks* (2012), 940–965. *(Citirano na strani 3.)*

[4] Android, *Android developer spletna stran*, https://developer.android.com/reference/packages.html. *(Citirano na strani 6.)*

[5] AOSP, *Android Open Source Project spletna stran*, http://source.android.com. *(Citirano na strani 6.)*

[6] bluetooth, *Bluetooth spletna stran*, http://www.bluetooth.com/Pages/Bluetooth-Home.aspx. *(Citirano na strani 8.)*

[7] G. Caroni, Walking the Web of trust, *Proceedings IEEE 9th International Workshops on Enabling Technologies Infrastructure for Collaborative Enterprises WET ICE 2000* (2000), 153–158. *(Citirano na strani 19.)*

[8] FabFi, *FabFi spletna stran*, http://fabfi.fablab.af/. *(Citirano na strani 5.)*

[9] JAVAapi, *Java API spletna stran*, http://docs.oracle.com/javase/6/docs/api/. *(Citirano na strani 16.)*

[10] J. Preshing, *Hash Collision Probabilities*, http://preshing.com/20110504/hash-collision-probabilities. *(Citirano na strani 10.)*

[11] R. Khodr, Ad-hoc network on Android, *Mathematical Modelling* (2010), . *(Citirano na strani 4.)*

[12] S. Madden in P. Levis, Pro Smartphone Cross-Platform Development: iPhone, Blackberry, Windows Mobile and Android Development and Distribution, *IEEE Internet Computing* (2008), 9–11. *(Citirano na strani 2.)*

[13] S. Misra in S. C. Misra, I. Woungang, *Guide to Wireless Mesh Networks* Springer London (), 2009.231–254 *(Citirano na straneh 2 in 11.)*

[14] N. Nurseitov in M. Paulson, R. Reynolds et al., Comparison of JSON and XML Data Interchange Formats: A Case Study, *Scenario* (2009), 157–162. *(Citirano na strani 16.)*

[15] Q. Kennemer, *Gartner: Android devices account for 75% of worldwide smartphone sales in Q1 2013*, http://phandroid.com/2013/05/14/gartner-q1-2013/. *(Citirano na strani 6.)*

[16] V. Rastogi in V. Riberio, A. Nayar, Measurements in OLPC mesh networks, *2009 7th International Symposium on Modeling and Optimization in Mobile Ad Hoc and Wireless Networks* (2009), 1–6. *(Citirano na strani 5.)*

[17] R. Lawrence, *Open Source Licensing Software Freedom and Intellectual Property Law*, Prentice Hall PTR. *(Citirano na strani 22.)*

[18] N. Russler, *Android Wifi Hotspot Manager Class*, http://www.whitebyte.info/android/android-wifi-hotspot-manager-class. *(Citirano na strani 13.)*

[19] Serval Project, *Spletna stran Serval Projekta*, http://www.servalproject.org/, 2013. *(Citirano na strani 5.)*

[20] L. Sicard in M. Markovics, An Ad-hoc Network of Android Phones Using BATMAN, *blogitudk* (2010), 1–8. *(Citirano na strani 4.)*

[21] M. Siddiqui in S. Amin, C. Hong, An Efficient Mechanism for Network Management in Wireless Mesh Network, *2008 10th International Conference on Advanced Communication Technology* (2008), 301–305. *(Citirano na straneh 2 in 3.)*

[22] T. Simonite, *Project Aims to Set Smartphones Free From Cellular Networks*, http://mashable.com/2013/07/12/serval-project/. *(Citirano na strani 5.)*

[23] H. Sinky in B. Handaoui, Implementation and performance measurement and analysis of OLSR protocol, *IWCMC 2010 Proceedings of the 6th International Wireless Communications and Mobile Computing Conference* (2010), 286–290. *(Citirano na strani 4.)*

[24] A. Tanendbaum, *Modern Operating Systems*, Prentice Hall international editions TS - GBV - Gemeinsamer Bibliotheksverbund. *(Citirano na strani 16.)*

[25] s. Verma, OLPC: are we there yet?, *Linux Journal* (2011), . *(Citirano na strani 5.)*

[26] X. Wang in A. Lim, IEEE 802.11s wireless mesh networks: Framework and challenges, *Ad Hoc Networks* (2008), 970–984. *(Citirano na strani 3.)*

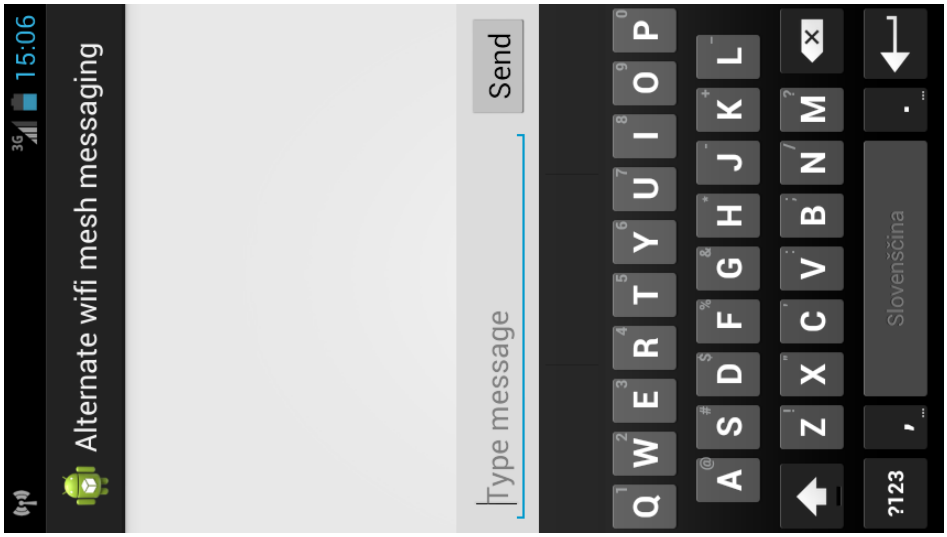
[27] Webopedia, *What is Wi-Fi?*, http://www.webopedia.com/TERM/W/Wi_Fi.html. *(Citirano na strani 7.)*

[28] Wi-Fi Alliance, *Wi-Fi Direct^{TM} spletna stran*, http://www.wi-fi.org/discover-and-learn/wi-fi-direct. *(Citirano na strani 8.)*

[29] Z. Xu in Y. Wang, J. Zhu, A Reliable Multicast Routing Protocol for High-speed Mobile Ad Hoc Networks: R-ODMRP, *Journal of Software* (2010), 20–27. *(Citirano na strani 4.)*

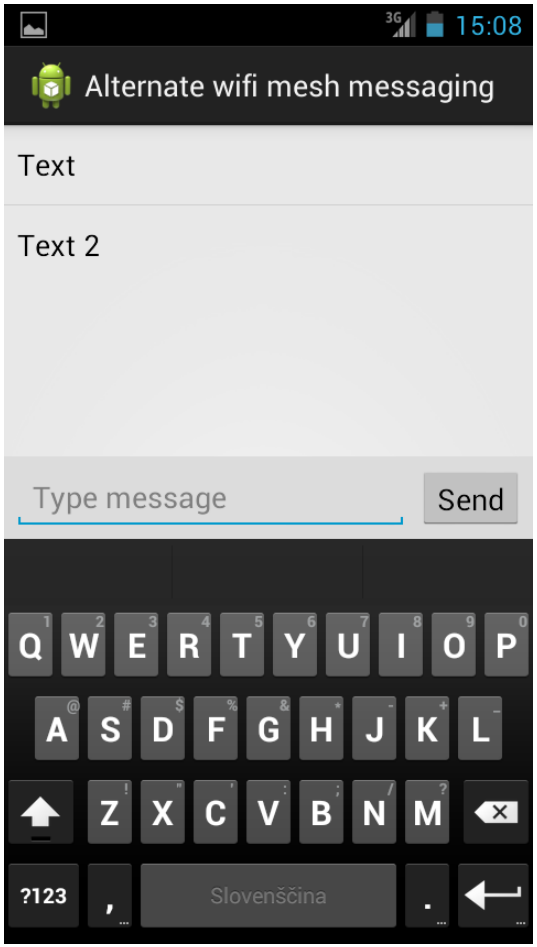
Priloge

Zaslonski posnetki testnega programa



Slika 1: Zaslonski posnetek 1

Slika 2: Zaslonski posnetek 2



Slika 3: Zaslonski posnetek 3

Pilotni program (zgoščenka)