

# Seminarska naloga

## Implementacija algoritma za grupiranje in določanje vodja grupe

Danijel Maraž, Nejc Rebernik

Fakulteta za Računalništvo in Informatiko UL dm9929@student.uni-lj.si

**Povzetek** The article covers the work done in the scope of the seminar assignment as part of the subject wireless sensor networks.

**Keywords:** Raft nRF24L01 Leader Consensus Wemos ESP

### 1 Uvod

Cilj projekta je bila implementacija algoritma za določanje vodje gruče brezžičnih naprav. Razvoj in testiranje sva opravljala na ploščici z WEMOS D1 mini kontrolerjem in modulom za komuniciranje nRF24L01. Na ploščici so bili nameščeni še drugi moduli, ki pa jih v obsegu najinega razvoja nisva uporabila. Modul nRF24L01 sprejema in oddaja na frekvenci 2.4GHz z maksimalno hitrostjo prenosa do 2Mbps. Majhne velikosti sporočil uporabljenih v komunikaciji in glede na to relativno velika podatkovna pretočnost sta omogočala hiter razvoj in testiranje ter skalabilnost. Za določanje vodje v gruči sva implementirala protokol RAFT. Programska koda je napisana v jeziku C++ s knjižnico za upravljanje s komunikacijskim modulom nRF24L01.

### 2 Splošna struktura

Odločila sva se, da ne bova imela več kot ene verzije kode in da se bo vsaka naprava lahko priključila na najino omrežje. Posledično je koda že sama po sebi zelo univerzalna, saj vsebuje vse kar napravi omogoča, da opravlja vse vloge RAFT protokola. Osnoven tok dogodkov sestoji iz aktivacije naprave in pogona *LR\_task* in *election\_task* opravil. *LR\_task* opravilo je namenjeno poslušanju aktivnosti na omrežju, ter hkrati pošiljanju sporočil, ko je to potrebno. *Election\_task* je za razliko samo časovnik, ki napravi sporoči, kdaj naj privzeto pošlje sporočilo. Za tako zasnovano sva se odločila, ker je posledica tega večja predvidljivost in determinizem samega programa, saj lahko samo eno opravilo pošilja oziroma sprejema.

### 3 Pošiljanje in poslušanje

Knjižnica za upravljanje z nRF24L01 modulom predvideva za prenos podatkov uporabo programskih abstrakcij z imenom cevi (eng. pipe), ki pa so najin projekt v veliki meri ovirale, saj je imeti hkrati odprto cev za poslušanje in branje nemogoče in pa so sploh cevi namenjene neenakopravni komunikaciji torej bolj v staticni situaciji master-slave. Posledično se ob vsakem klicu posebne funkcije *sendMsg*, ki je namenjena pošiljanju sporočil radio modul ponovno aktivira in odpre nova pisalna cev. Enako se zgodi ob vsaki novi vrnitvi v zanko poslušanja le, da se tokrat na novo odpre poslušalna cev. Cevi imajo prav tako tudi naslove, ki pa jih nisva potrebovala in sva ves čas pisala in poslušala na nekem poljubnem privzetem naslovu. Vredno je omeniti, da sva tako komunikacijo lahko doseгла z nastavljivo delovanja modula, ki ne predvideva uporabe avtomatskih potrditvenih paketov ob prejetju paketka, zato je scenarij v katerem se poslan paketek izgubi mogoč vendor na razdalji nekaj deset centimetrov bistveno manj verjeten.

#### 3.1 Election task

Opravilo ima zelo preprosto zasnova. Njegova glavna naloga je, da nastavi spremenljivko *hastoSend* na *true* in posledično sproži pošiljanje novega sporočila (nove volitve) v glavnem opravilu. Poleg tega s funkcijo *changeRole* nastavi vlogo naprave na to *kandidata*.

#### 3.2 Listen and react task

Ob vstopu v glavno opravilo naprava vsem ostalim članom sporoči, da se je aktivirala, ter takoj za tem začne poslušalno pisalno zanko. Naprava nato:

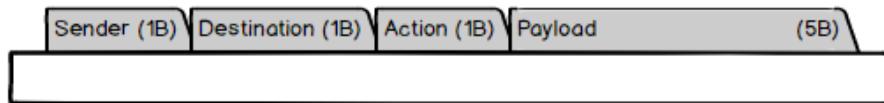
- Pod pogojem, da je *kandidat* preveri, če je dobila več kot polovico glasov. V tem primeru postane novi vodja.
- Začne poslušati in ob primeru, da obstajajo nova sporočila jih prebere ter določi kako se mora odzvati.

V najslabšem primeru bo naprava zaznala promet, ki pa ji ni namenjen (namembnost se preveri s funkcijo *relevantData*) in si bo s klicem funkcij *seenDevice* in *removeInactiveDevs* posodobila interno tabelo naprav.

### 3.3 Tabela naprav

Tabela naprav je namenjena shranjevanju seznama naprav in stanja naprav v omrežju. Vsaka naprava, ki se omrežju predstavi s poslanim sporočilom se zapiše v lokalno tabelo naprav. Napravo se iz tabele odstrani, ko je njena zadnja aktivnost bila zaznana pred več kot petimi mandati. Velikost tabele sva iz praktičnih razlogov omejila na samo deset različnih naprav, vendar bi v praksi glede na najin način naslavljanja lahko ta imela do  $2^8 - 1$  naslovov.

## 4 Protokol komunikacij



Slika 1. Osnovna oblika paketa.

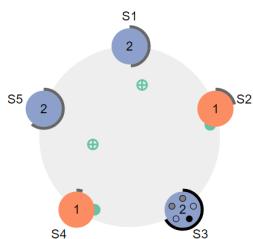
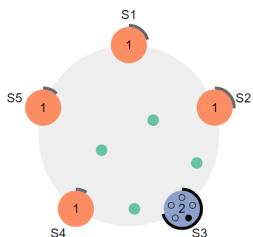
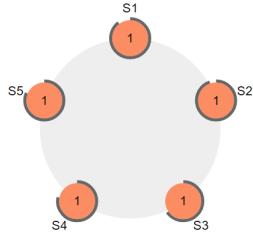
Pošiljanje podatkov na radijskem modulu je omejeno na samo 8 bajtov na paket, zato sva pri implementaciji najinega dodatka aplikacijski plasti poskušala čim več bajtov nameniti koristnim informacijam, saj bi to bilo v realni situaciji zaželeno. Prvi trije bajti so namenjeni naslavljaju paketov ter funkcij Raft protokola in sicer:

- Prvi bajt je namenjen naslovu pošiljatelja paketka. Ta je pridobljen iz zadnjega ASCII znaka MAC naslova naprave.
- Drugi bajt je namenjen naslovu prejemnika paketka ("F" je broadcast).
- Zadnji bajt identificira RAFT dejanje paketka.

Opis možne Raft vsebine:

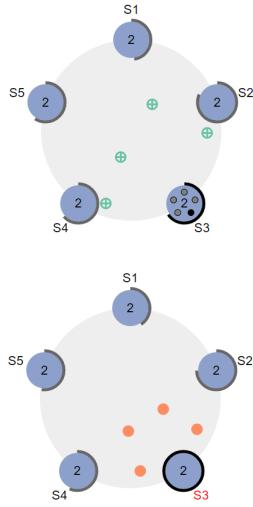
- A (ang. acknowledge) paket služi za potrditev prejema poljubnega sporočila, ter kot tip predstavitvenega paketa.
- E (ang. election) paket poziva ostale naprave k glasovanju za novonastalega kandidata, ki je pošiljatelj paketa.
- V (ang. vote) paket predstavlja glas privrženca, ki ga ta isti pošlje kandidatu in je odziv na E paket.

- H (ang. heartbeat) paket ponastavi časovnik za nove volitve in ga pošlje vodja privržencem. V realnem svetu služi tudi za deljenje informacij med člani skupine (ang. log propagation).



## 5 Raft protokol

Raft (Reliable ,Replicated ,Redundant, And Fault-Tolerant) je algoritem za sklepanje soglasja v porazdeljenih sistemih z več vozlišči. Narejen je bil leta 2014 z misljijo, da bo v veliki meri nadomestil starejši in veliko bolj zapleten Paxos algoritem. Omenjeno soglasje Raft doseže s pomočjo izvolitev vodilnega vozlišča.



**Slika 2.** Shema tipičnega volitvenega cikla.

Naprava ima lahko v Raft skupini eno izmed treh vlog:

- Vodja (ang. leader)
- Privrženec (ang. follower)
- Kandidat (ang. candidate)

Odgovornost vodje je, da privržencem redno pošilja sporočila, da je ta še vedno aktiven (ang. heartbeat). Med tem ima vsak privrženec poljubni časovni interval (navadno med 150 ms - 300 ms), ki mu ob izteku spremeni vlogo v tisto kandidata ter sproži nove volitve. Če privrženec prejme potrditev, da je vodja aktiven pred iztekom časovnega intervala se časovni interval ponastavi. Pri Raftu je važna tudi kvaliteta posameznega vodje, zato se čas beleži v mandatih. En mandat je (enako kot v svetu politike) trajanje aktivnosti enega vodje oziroma potek enih volitev. V idealni situaciji bomo torej imeli samo en mandat, ker se naš vodja ne bo nikoli pokvaril. V praksi temu ni tako in številka mandata se poveča ob začetku vsakih novih volitev.

### 5.1 Volitve

Za izvolitev mora kandidat pridobiti več kot polovico glasov. Ko naprava postane kandidat voli zase ter sporoči ostalim, da je prišel čas za nove volitve. Privrženci se na ta klic odzovejo ter oddajo svoj glas. Včasih izvolitev kandidata

ne uspe bodisi, ker je prišlo do aktivacije dveh kandidatov (v takem primeru postaneta ponovno privrženca) in posledično delitve glasov ali, ker je imel kandidat manjši števec mandata kot večina privržencev. Privrženci lahko glasujejo samo enkrat na mandat in to naredijo pod pogojem, da je številka mandata kandidata večja ali enaka njihovemu lokalnemu števcu. Vredno je omeniti, da se scenariju dveh sočasnih kandidatov da izogniti z uporabo naključnih števcev za štetje časa pred ponovnim kandidiranjem. Tako je verjetnost, da bo prišlo do dveh sočasnih volitvenih kampanj bistveno manjša.

## 5.2 Naša implementacija

Naša implementacija sledi strukturi Raft protokola z nekaj izjemami:

- Odločila sva se opustiti kvalitativne izbire vodje na podlagi številke mandata, saj pri merjenju le 5 bajtov informacij to ni zelo pomembno in bi vodja v realnem svetu gotovo pošiljal podatke kam drugam. Vodja tudi ne beleži bistveno več informacij kot privrženec.
- Štetje mandatov je prilagojeno in kot mandat štejemo tudi uspešno poslano sporočilo, da je vodja še aktiven, ki resetira časovnike naprav.
- Naprave ob prejetju sporočila, da je vodja še aktiven vsem napravam pošljejo potrditveno sporočilo. Tako lahko vsaka naprava vodi evidenco aktivnih naprav in se naprave v mirovanju, torej tiste, ki prejemajo heartbeat sporočilo ne zбриšejo iz tabele naprav.
- Trajanje volitvenih ciklov je prilagojeno in sicer se privržencev časovnik giba med 2 do 5 sekundami. Za razliko ima vodja frekvenco pošiljanja potrditve-nega paketa 1 sekundo. Za to sva se odločila, ker je tako bistveno lepše razvidno kaj se trenutno dogaja v omrežju.

## 6 One time pad

Sistemu sva dodala nivo varnosti s kriptiranjem sporočil z metodo one-time pad. Metodo sva izbrala zaradi enostavnosti in majhne računske zahtevnosti, kar je pomemben faktor pri procesorjih z nizko močjo in omejitvami napajanja. One-time pad je kriptografska metoda, ki nad biti sporočila pri šifriranju in dešifriranju izvede XOR operacijo s ključem iste dolžine. V implementaciji sva uporabila statičen ključ dolžine 8 bajtov. Za tak pristop sva se odločila zaradi nezaželenih zapletenosti, ki bi jo prineslo dinamično izmenjevanje ključev. To med drugim tudi omogoča visok nivo fleksibilnosti, saj lahko naprava kadarkoli

izpade ter se kasneje priklopi v omrežje brez kakšnegakoli izmenjevanja ključev. Uporaba istega ključa pa bi bila v kakšnem bolj resnem in zaupnem sistemu nepredstavljava, saj lahko napadalec z zajetjem samo dveh spročil dobi XOR vrednost njunih čistopisov. Naš sistem k večjemu zagotavlja zaupnost z nejasnostjo razumevanja sistema (ang. security by obscurity). Zagotavlja pa ne integritete, saj lahko napadalec prav tako spreminja bite kriptiranih sporočil brez, da bi to mi opazili. Prav tako sistemu manjka avtentifikacija, saj lahko napadalec nezaznano pošilja zajete pakete napravam.

$$m1 \oplus key = c1$$

$$m2 \oplus key = c2$$

$$c1 \oplus c2 = m1 \oplus key \oplus m2 \oplus key = m1 \oplus m2$$

## 7 Zaključek

Uspelo nama je implementirati najnim ciljem primerno verzijo Raft protokola. Pri tem sva spoznala razliko med razvijanjem aplikacij za kontrolerje ter običajne višje nivojske sisteme. Lahko rečeva, da je pri prvih delovni tok precej drugačen, saj je tudi način razhroščevanja drugačen, zaradi prisotnosti nizko nivojskih fizičnih elementov kot so radijski oddajniki in interference iz okolja. Sledi nekaj bistvenih dosežkov projekta:

- Delajoč Raft protokol
- Enoličnost kode med napravami (skalabilnost)
- Možnost poljubnega priklopa naprav v omrežje (fleksibilnost)
- Zagotovitev omejene zaupnosti z one-time pad

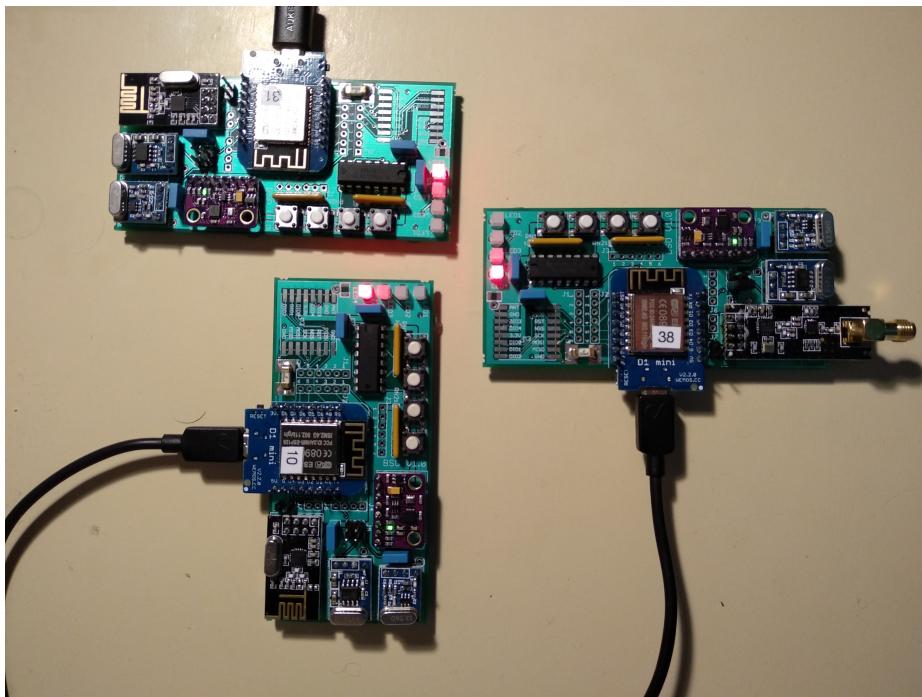
Projekt bi lahko tudi zlahka nadgradili in sicer z dodatkom:

- Izmenjevanja in generacije krptografskih ključev z Diffie-Hellman protokolom
- Pravo Raft implementacijo, ki vključuje izbiro najboljšega možnega vodje
- Funkcionalnostjo pošiljanja dejanskih podatkov z enim izmed senzorjev npr. temperaturo
- Pošiljanje podatkov na zunanji spletni strežnik

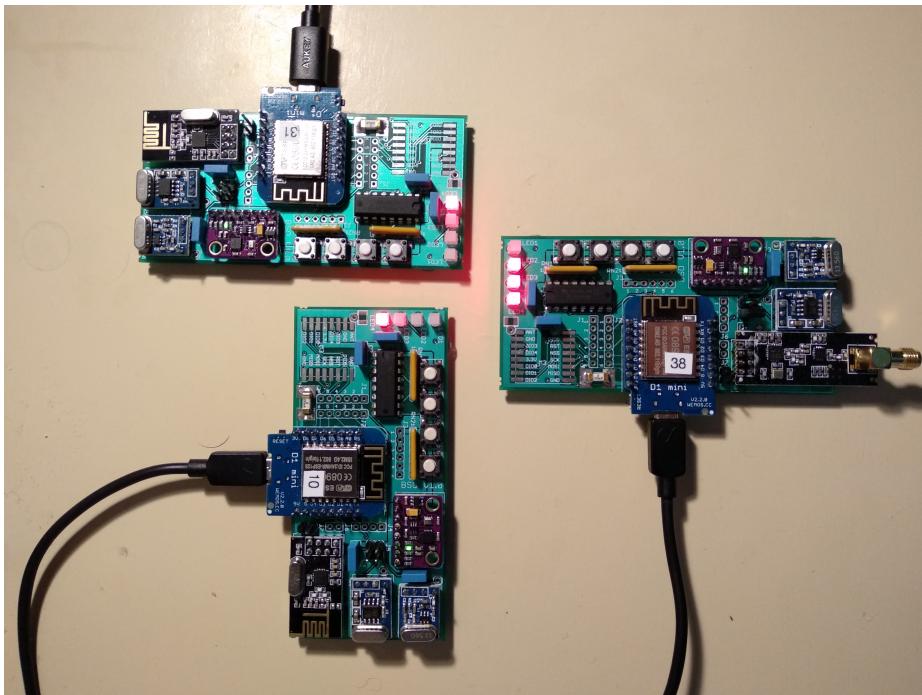
## 8 Primera

Za lažjo predstavo dogodkov sva sistemu dodala užiganje LED diod glede na vlogo naprave.

- Vodja - 3 diode
- Kandidat - 2 diodi
- Privrženec - 1 dioda



**Slika 3.** Brez vodje bodo kmalu sledile nove volitve.



**Slika 4.** Prisotnost vodje.