

**DECENTRALIZOVANO USAGLAŠAVANJE SISTEMSKIH SATOVA U ČVOROVIMA  
AD-HOK BEŽIČNE MREŽE****CONSENSUS CLOCK SYNCHRONIZATION IN DECENTRALIZED WIRELESS  
NETWORKS**

Saša Taloši, *Fakultet tehničkih nauka, Novi Sad*

**Oblast – RAČUNARSTVO I AUTOMATIKA**

**Kratak sadržaj** – Bežične ad-hok mreže danas imaju veoma široku primenu. Zbog prirode upotrebe, sinhronizacija vremena je jedan od bitnih problema sa kojim se suočavaju. U ovom radu ćemo prikazati jedan postojeći algoritam sinhronizacije vremena koji se bazira na usaglašavanju sistemskih satova čvorova.

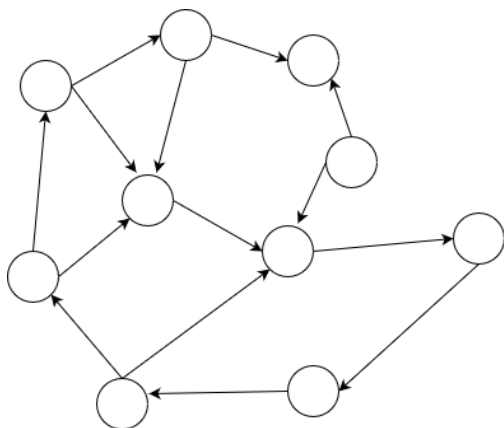
**Ključne reči:** Sinhronizacija satova, usaglašavanje satova, decentralizovane mreže, bežične mreže

**Abstract** – Decentralized low-powered wireless networks have a wide application in science and industry. Because of their distributed nature and differences in environment time synchronization is an important issue that needs to be addressed. In this paper, we will present one algorithm that solves this problem and is based on consensus clock synchronization.

**Keywords:** Clock synchronization, consensus synchronization, decentralized network, wireless network

**1. UVOD**

Kada je bežična mreža decentralizovana nazivamo je ad-hok bežičnom mrežom. U mreži ne postoje čvorovi koji služe za usmeravanje poruka već svaki čvor ravnopravno učestvuje u rutiranju podataka ka drugim čvorovima. Putanja podataka se određuje na dinamički način i zavisi od povezanosti same mreže i algoritma za rutiranje.



Slika 1. Primer bežične ad-hok mreže

Ovakve mreže predstavljaju dobro rešenje za prikupljanje podataka sa velikog broja uređaja. Neki od primera gde bi se ad-hok mreža mogla koristiti su strujna brojila, ulična rasveta, semafori, mreže senzora toplote, vlage, itd. Sem prikupljanja podataka, svaki čvor ima i sposobnost da obradi podatke i da upravlja aktuatorima.

Svaki čvor se sastoji od mikrokontrolera, oscilatora, mrežnog priključka, izvora struje, itd. Izbor mikrokontrolera i njihova primena raste iz godine u godinu, kao što propratno rastu i njihove sposobnosti. Da bi mikroprocesor radio, na čvoru se nalazi i oscilator (interni ili eksterni) koji određuje takt mikrokontrolera. Čvor, takođe, ima i način da komunicira sa mrežom. Komunikacija može biti putem bakra ili radio talasa. Kod bežičnih mreža nam je najbitnije na kojoj frekvenciji se vrši komunikacija. Neko pravilo kojim možemo da se vodimo je da što je veća frekvencija, to je prenos podataka brži, ali mreža postaje manje otporna na smetnje. Neki od primera poznatih tehnologija bežičnih mreža su: Wi-Fi [1], Bluetooth [2], Zigbee [3], LoRaWAN [4] i sada široko popularne bežične mreže tipa 5G i njeni prethodnici 1G, 2G, 3G i 4G.

Najčešće se koriste kristalni (kvarc) oscilatori zato što su jeftini, mali, mala im je potrošnja i relativno dobro drže frekvenciju u prisustvu smetnji. Međutim, temperatura i vlaga mogu promeniti takt oscilatora u dovoljnoj meri da uvedu grešku u sat mikrokontrolera.

**1.1. Pregled problema**

Vremenska sinhronizacija je veoma bitna za razne primene gde čvorovi moraju da rade istovremeno. Jedan takav primer je prikupljanje podataka. Merne stanice za struju, gas, vlažnost povremeno očitavaju i šalju podatke na jedno centralno mesto gde se obrađuju. Podaci mogu da se obrađuju i na samom čvoru u određenim primenama, ali je i dalje bitno kada je podatak očitao.

Drugi primer gde je bitna sinhronizacija vremena je pri tehnologijama koje se oslanjaju na usaglašenost (sinhronizam) kao ključan faktor. Na primer, slanje paket tokom precizno određenih vremenskih okvira kada su kanali slobodni.

Sinhronizacija čvorova se može iskoristiti i za uštedu energije. Čvorovi koji rade na baterije često se nalaze u uspavanom režimu rada i ako se ne bi probudili na vreme, propustili bi pakete koji se šalju ili bi kasno poslali svoj paket. Ovo bi bilo rešivo ako bi povećali vreme koje je čvor budan, ali bi to skratilo životni vek baterije.

**NAPOMENA:**

Ovaj rad proistekao je iz master rada čiji mentor je bio dr Milan Rapaić, red. prof.

U ovom radu ćemo rešavati dva problema kod unutrašnjih satova u čvorovima. Prvi problem je inicijalna loša postavka sata koja uvodi smicanje u odnosu na realno vreme. Drugi problem je promena brzine oscilatora koja tokom vremena uvećava smicanje sata.

Greška smicanja je razlika u vremenu između vremena čvora i realnog vremena. Najčešće se pojavljuje pri samom postavljanju ili pri lošoj inicijalizaciji memorije čvora. Kada se čvor postavlja, nije moguće postići prikladnu preciznost i uključiti čvor u pravom momentu i ovde nastaje greška. U slučaju da inicijalizacija unutrašnjeg sata zavisi od predefinisane memorije, moguće je da je memorija loše postavljena i nastaje druga vrsta greške.

Kao što smo već naveli, temperatura i vlaga mogu kako ubrzati, tako i usporiti rad oscilatora. Ovo dovodi do toga da sat čvora napreduje ili kasni za realnim vremenom i evidentnije je na lokacijama na kojima su te promene veće.

Algoritam usaglašavanja [5], koji je primenjen u ovom radu, rešava oba prethodno predstavljena problema. Svi čvorovi se usaglašavaju ne na realno, već novo usaglašeno vreme. Postupak usaglašavanja takođe pokušava da koriguje i greške u brzini promene pojedinačnih oscilatora. Algoritam je potrebno periodično izvršavati kako bi se korigovale greške nastale od poslednjeg pokretanja, za šta je poželjno da svi čvorovi u svom dometu imaju što više čvorova koji sa njima komuniciraju.

## 1.2. Organizacija ostatka rada

U poglavlju 2. se prikazuju najznačajniji faktori od uticaja na sinhronizaciju vremena i navode se neki primeri postojećih algoritama. U poglavlju III je uveden matematički model sata, a u poglavlju IV je objašnjena matematička osnova algoritma. Poglavlje V daje opis organizacije simulacije i analizira rezultate simulacije. Nakon toga u poglavlju VI sledi zaključak i diskusija.

## 2. PREGLED LITERATURE

### 2.1. Faktori od uticaja na sinhronizaciju

Prilikom odabira algoritma za sinhronizaciju potrebno je uzeti u obzir nekoliko faktora [1]:

- Vreme pripreme - vreme koje je potrebno da pošiljalac pripremi paket za slanje.
- Vreme pristupa - vreme čekanja na slobodan kanal za slanje.
- Vreme slanja/primanja - vreme koje je potrebno da se poruka pošalje i primi i često je jednake dužine.
- Vreme prenosa - dovoljno malo da možemo da ga zanemarimo.
- Vreme obrade - vreme koje je potrebno da primalac obradi paket.
- Energetska efikasnost - bitan faktor ako je čvor na bateriju.
- Brojnost - bitan ako je algoritam namenjen za male ili velike mreže.
- Potrebna preciznost - preciznost koja se postiže algoritmom.
- Robusnost - opisuje koliko je neki algoritam otporan na to da se aktivni čvorovi mogu menjati kroz vreme.

- Životni vek sinhronizacije - da li sinhronizacija mora da se pokreće više puta ili je dovoljno samo jednom.
- Domet - bitan ako može sinhronizacija da se radi i lokalno, ne uvek na celoj mreži.
- Cena - da li je potrebno dodatnih hardverskih modula da bi algoritam radio.

### 2.2. Postojeći algoritmi

*Globalni pozicioni sistemi* (GPS) se mogu koristiti za sinhronizaciju sa realnim vremenom i pružaju visok stepen preciznosti, čak od 1 do 10 nanosekundi greške [6]. Ovakvi sistemi zahtevaju dodatan hardver koji će komunicirati sa satelitom.

*Precision Time Protocol* (PTP) je algoritam koji sinhronizuje vreme u računarskim mrežama. Na lokalnim mrežama on postiže i preciznost ispod 1 milisekunde [7]. Koristi se gde GPS nije dostupan, finansijskim sistemima i drugim vremenski kritičnim mrežama.

*Network Time Protocol* (NTP) je jedan od najstarijih algoritama za sinhronizaciju računarskih mreža. Dizajniran je da sinhronizuje sve povezane računare na preciznost od oko 1 milisekunde [8].

*The Timing-sync Protocol for Sensor Networks* (TPSN) [9] je algoritam koji je proširenje NTP algoritma i koristi se u bežičnim mrežama. Tokom postavljanja mreže čvorovi pronalaze jedni druge i uspostavljaju međusobne odnose nadređenog i podređenog. Ovako se stvara stablo gde je glavni čvor u korenu. Sinhronizacija kreće od korena i obavlja se niz grane dok svi čvorovi ne budu sinhronizovani.

*The Reference Broadcast Synchronization* [10] (RBS) je algoritam koji pruža sinhronizaciju samo čvorovima koji su u dometu glavnog čvora. On može ili ne mora da ima GPS prijemnik i povremeno pošalje paket sinhronizacije koji ostali čvorovi iskoriste da se sinhronizuju.

U algoritmu *Asynchronous Diffusion* [11] čvorovi potežno šalju sinhronizacioni paket svojim komšijskim čvorovima koji odgovaraju sa svojim vremenom. Prvi čvor usrednji vrednost svih dobijenih vremena i dobijenu vrednost pošalje svima koji su u dometu i svi prihvate poslatu vrednost kao novo vreme. Ovo se ponavlja u celoj mreži dok se ne postigne sinhronizacija.

*Average TimeSync* [12] (ATS) je mešavina dva algoritma usaglašavanja. U prvom koraku, čvorovi šalju svoje lokalno vreme u cilju da izračunaju grešku brzine u odnosu jedan sa drugim. Nakon toga, čvorovi šalju svoju pretpostavku greške brzine i ovo koriste da pretpostave grešku brzine usaglašenog sata. Isti koraci se ponove i za grešku smicanja.

## 3. MODEL SATA

Sat možemo da prikazemo kao funkciju u zavisnosti od realnog vremena

$$S(t) = \alpha t + \beta. \quad (1)$$

Prvi izvod sata  $dS/dt = \alpha$  predstavlja brzinu sata i određuje koliko će se sat uvećati ili umanjiti za određeni period vremena, a  $\beta$  je smicanje sata u odnosu na realno vreme. Idealna brzina sata je 1, a idealno smicanje 0.

Koliko često treba raditi sinhronizaciju satova zavisi od promene brzine sata. Što je ta promena veća,

sinhronizaciju treba raditi češće i obrnuto. Da bi se postigla sinhronizacija u mreži, potrebno je da svi čvorovi  $i = 1 \dots N$  savršeno kompenzuju parametre  $\alpha_i$  i  $\beta_i$ , tako da je smicanje 0, i svi satovi kucaju istim taktom.

$$\widehat{S}_i(t) = \frac{\alpha_i}{\widehat{\alpha}_i} t + \beta_i - \widehat{\beta}_i = t, i = 1 \dots N \quad (2)$$

$\widehat{S}_i(t)$  je kompenzovani sat čvora  $i$ , a  $\widehat{\alpha}_i$  i  $\widehat{\beta}_i$  su kompenzacija brzine i kompenzacija smicanja čvora  $i$ . Da bi se zadovoljila jednačina (2), kompenzacioni parametri svakog čvora moraju biti jednaki njegovim parametrima sata,  $\widehat{\alpha}_i = \alpha_i$  i  $\widehat{\beta}_i = \beta_i$ .

#### 4. ALGORITAM USAGLAŠAVANJA

Umesto da pokušamo i sinhronizujemo vreme na realno vreme, ovaj algoritam pokušava da postigne usaglašeno vreme unutar mreže. Sa svakom sinhronizacijom, kompenzacioni parametri se ažuriraju i s vremenom se postiže usaglašavanje, tako da je

$$\lim_{t \rightarrow \infty} \widehat{S}_i(t) = S_u(t), \quad (3)$$

gde je

$$S_u(t) = \alpha_u t + \beta_u \quad (4)$$

usaglašeni sat. Proširivanjem (3) pomoću (2) i (4) dobijamo

$$\lim_{t \rightarrow \infty} \frac{\alpha_i}{\widehat{\alpha}_i} t + \beta_i - \widehat{\beta}_i = \alpha_u t + \beta_u, \quad (5)$$

odakle neposredno sledi

$$\lim_{t \rightarrow \infty} \widehat{\alpha}_i = \frac{\alpha_i}{\alpha_u}, \quad (6)$$

$$\lim_{t \rightarrow \infty} \widehat{\beta}_i = \beta_i - \beta_u. \quad (7)$$

Da bi se postigli kompenzacioni parametri čvorovi moraju da ponavljaju sinhronizaciju koja se sastoji iz dva dela, kompenzacija smicanja i kompenzacija brzine. U svakoj sinhronizaciji, čvorovi razmenjuju svoje lokalne satove da bi postigli jedno zajedničko vreme i poništili grešku smicanja, a u isto vreme upoređuju novo vreme sa vremenom iz prošle sinhronizacije i uklanjaju grešku brzine. Sinhronizacija se vrši u intervalu  $t_{sync}$ , koji je podesiv i može se menjati po potrebi.

##### 4.1. Uklanjanje greške smicanja

U svakoj sinhronizaciji, čvorovi treba da izvrše sledeće korake:

1. Svaki čvor podesi svoj faktor samopouzdanja na
2. Svaki čvor pošalji paket sinhronizacije koji sadrži njegovo trenutno vreme i faktor samopouzdanja.
3. Svaki put kada prime sinhronizacioni paket izračunaju (8) i povećaju svoj faktor samopouzdanja (9).

$$\widehat{S}_i^+(t_j) = \frac{\gamma_i \widehat{S}_i(t_j) + \gamma_j \widehat{S}_j(t_j)}{\gamma_i + \gamma_j} \quad (8)$$

$$\gamma_i^+ = \gamma_i + 1, i = 1 \dots N, \forall j \in R_i \quad (9)$$

$N$  je broj čvorova u mreži, a  $R_i$  su čvorovi koji su u dometu čvora  $i$ .  $\gamma_i$  i  $\gamma_j$  su faktori samopouzdanja čvorova  $i$  i  $j$ , a  $t_j$  je vreme kada je čvor  $j$  poslao svoj sinhronizacioni paket. Jednačina (9) takođe može da se izrazi kao (10) što je lakše mikrokontrolerima da izračunaju jer ima manje operacija množenja i deljenja.

$$\widehat{S}_i^+(t_j) = \widehat{S}_i(t_j) + \frac{\gamma_j}{\gamma_i + \gamma_j} (\widehat{S}_j(t_j) - \widehat{S}_i(t_j)) \quad (10)$$

Faktor samopouzdanja daje veći značaj čvorovima koji su primili sinhronizacione pakete jer su oni bliži usaglašenom vremenu.

##### 4.1. Uklanjanje greške brzine

Nakon što se sinhronizaciona runda završila, čvor  $i$  ima bolju estimaciju usaglašenog vremena,  $\widehat{S}_i^+(t_n + \tau)$ .  $t_n$  je vreme početka sinhronizacije, a  $\tau$  je trajanje sinhronizacije. Ako izračunamo parametar kompenzacije brzine,  $\widehat{\alpha}_i^+$ , kao rezultat ćemo sledeću sinhronizaciju započeti sa preciznijim satom.

$\widehat{\alpha}_i^+$  se može izračunati bez da čvor zna svoju grešku brzine. Vreme čvora u momentu  $t_n$  se može izraziti kao funkcija prethodnog vremena plus posmatrano vreme koje je prošlo (11).

$$\widehat{S}_i(t_n) = \widehat{S}_i(t_{n-1} + \tau) + \frac{\alpha_i}{\widehat{\alpha}_i} (t_n - t_{n-1} - \tau) \quad (11)$$

Isto važi i za vreme nakon sinhronizacije (12).

$$\widehat{S}_i(t_n + \tau) = \widehat{S}_i(t_{n-1} + \tau) + \frac{\alpha_i}{\widehat{\alpha}_i^+} (t_n + \tau - t_{n-1} - \tau) \quad (12)$$

Apsolutno vreme koje je prošlo između dve sinhronizacije može da se izrazi kao (13).

$$t_n - t_{n-1} - \tau = \frac{\widehat{\alpha}_i}{\alpha_i} (\widehat{S}_i(t_n) - \widehat{S}_i(t_{n-1} + \tau)) \quad (13)$$

Ako ubacimo (13) u (12), dobijemo

$$\widehat{S}_i(t_n + \tau) = \widehat{S}_i(t_{n-1} + \tau) + \frac{\widehat{\alpha}_i}{\alpha_i} (\widehat{S}_i(t_n) - \widehat{S}_i(t_{n-1} + \tau) + \tau) \quad (14)$$

pri čemu je

$$\widehat{\alpha}_i^+ = \frac{\widehat{\alpha}_i (\widehat{S}_i(t_n) - \widehat{S}_i(t_{n-1})) + \alpha_i \tau}{\widehat{S}_i(t_n + \tau) - \widehat{S}_i(t_{n-1} + \tau)}. \quad (15)$$

S obzirom da ne znamo  $\alpha_i$  i  $\tau$  nismo u mogućnosti da računamo  $\widehat{\alpha}_i^+$  na način opisan jednačinom (15). Međutim, ako  $\widehat{\alpha}_i^+$  računamo svaki put kada primimo paket, možemo zanemariti  $\tau$  jer je ono prvenstveno suma vremena čekanja na slanje paketa, a vreme slanja se pretpostavlja da je zanemarljivo. Na taj način dobijamo

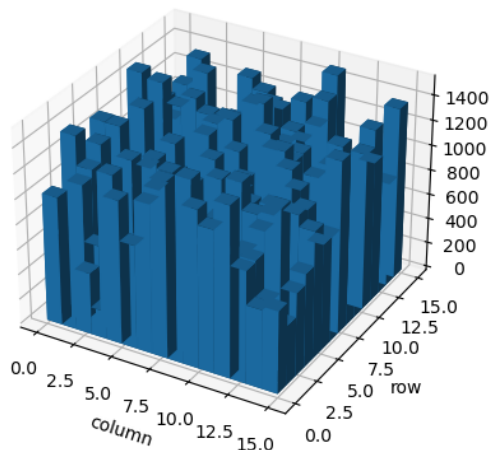
$$\widehat{\alpha}_i^+(t_{n,j}) = \widehat{\alpha}_i(t_{n,j}) \frac{\widehat{S}_i(t_{n,j}) - \widehat{S}_i(t_{n-1})}{\widehat{S}_i^+(t_{n,j}) - \widehat{S}_i(t_{n-1})}. \quad (16)$$

Vrednost  $\widehat{S}_i^+(t_{n,j})$  predstavlja ažurirani sat u trenutku  $t_{n,j}$  izračunato koristeći (10), dok je  $\widehat{S}_i(t_{n-1})$  je vreme nakon poslednje sinhronizacije. Podsećamo da je  $\widehat{\alpha}_i$  nije poznato pre prve sinhronizacije te u prvoj sinhronizaciji on iznosi  $\widehat{\alpha}_i = 1$ . radova.

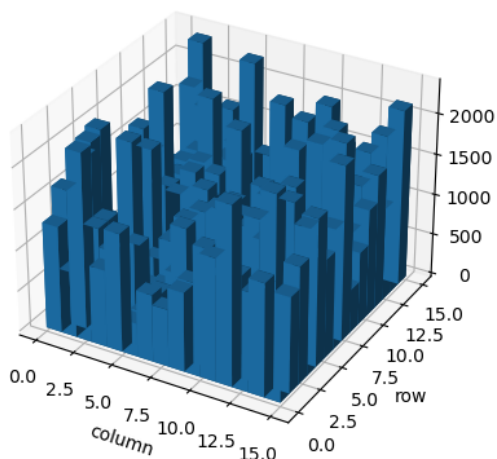
#### 5. REZULTATI SIMULACIJE

Za simulaciju se koristila mreža od 255 čvorova, dimenzija 15 puta 15 i svaki čvor je imao domet 3. Ako je par čvorova van dometa, nisu u mogućnosti da se čuju. Parametri satova su nasumično postavljeni na vrednosti u rasponima  $0.2 < \alpha < 1.2$  i  $-1500 < \beta < 1500$ , gde je  $\beta$  izraženo u milisekundama. Vreme u simulaciji je izraženo u milisekundama. Slika 2 je prikaz apsolutne greške svakog čvora. Nakon toga je u simulaciji vreme napredovalo 10 minuta. Slika 3 je prikaz apsolutne greške nakon 10 minuta simulacije, bez sinhronizacije. Slika 4 je

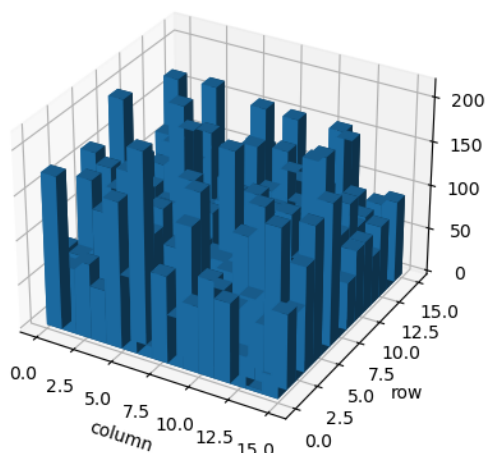
prikaz nakon sinhronizacije. Srednja apsolutna greška na početku iznosila je 749.6 milisekundi, nakon 10 minuta je iznosila 932.7 milisekundi i posle sinhronizacije iznosi 86.95 milisekundi. U toku simulacije primećeno je da vreme klizi naspram realnog vremena. Ovo bi moglo da se reši tako što bi se u mrežu dodao čvor sa realnim vremenom koji bi imao visok faktor samopouzdanja.



Slika 2. Apsolutne greške čvorova na početku simulacije



Slika 3. Apsolutne greške čvorova posle 10 minuta, bez sinhronizacije



Slika 4. Apsolutne greške čvorova posle 10 minuta, posle jedne sinhronizacije

## 6. ZAKLJUČAK

Prikazan je algoritam koji dosta zavisi od vremena pripreme, pristupa, slanja/primanja i obrade. Oslanja se na to da su sva vremena zbirno što manja da bi algoritam postigao bolju preciznost. Takođe je primećeno da preciznost raste sa porastom dometa između čvorova. Energetska efikasnost zavisi od toga koliko često se pokreće algoritam, što u velikoj meri zavisi od promenljivosti same mreže. Algoritam nije ograničen brojem čvorova i može da se modifikuje da se pokreće na lokalnom domenu. Izdrživ je i ne zavisi od promena čvorova. S obzirom da je u pitanju programsko rešenje, cena je mala jer ne zahteva nikakvu dodatnu elektroniku.

Pošto je u toku simulacije primećeno da usaglašeno vreme krene da se udaljava od realnog vremena, predlaže se da pojedini čvorovi budu prikazani sa GPS modulima koji bi davali precizno realno vreme i ovim čvorovima dati visok faktor samopouzdanja.

## 7. LITERATURA

- [1] IEEE 802.11TM WIRELESS LOCAL AREA NETWORKS, IEEE, [Online]. <https://www.ieee802.org/11/>
- [2] Bluetooth Specification, Bluetooth, [Online]. <https://www.bluetooth.com/specifications/specs>
- [3] 802.15.4 vs ZigBee, David Gascon, [Online]. <http://sensor-networks.org/index.php?page=08231231506>. 2012.
- [4] What Is LoRa?, Semtech, [Online]. <https://www.semtech.com/lora/what-is-lora>
- [5] Michael Kevin Maggs, Steven G O'Keefe, David Victor Thiel, Consensus Clock Synchronization for Wireless Sensor Networks, IEEE Sensors Journal, 6. 6. 2012.
- [6] GPS Time Transfer, National Institute of Standards and Technology, [Online]. <https://tf.nist.gov/time/commonviewgps.htm>
- [7] IEEE-1588 Standard for a Precision Clock Synchronization Protocol for Networked Measurement and Control Systems, NIST, [Online]. <https://www.nist.gov/system/files/documents/el/isd/ieee/tutorial-basic.pdf>
- [8] David L. Mills, Computer Network Time Synchronization: The Network Time Protocol, CRC Press, 24. 3. 2006.
- [9] S. Ganeriwal, R. Kumar, and M. Srivastava, Timing-sync protocol for sensor networks, in Proc. 1st ACM Conf. Embed. Netw. Sensor Syst., Nov. 2003.
- [10] J. Elson, L. Girod, and D. Estrin, Finegrained network time synchronization using reference broadcasts, in Proc. 5th Symp. Operat. Syst. Design Implement., vol. 36. Dec. 2002.
- [11] Q. Li and D. Rus, Global clock synchronization in sensor networks, in Proc. IEEE Conf. Comput. Commun., vol. 1. Hong Kong, China, Mar. 2004.
- [12] L. Schenato and G. Gamba, A distributed consensus protocol for clock synchronization in wireless sensor network, in Proc. 46th IEEE Conf. Decis. Control, New Orleans, LA, Dec. 2007.

### Kratka biografija:



Saša Taloši rođen je 1993. u Novom Sadu. Završio osnove akademske studije na smeru Računarstvo i automatika na katedri za Računarsku tehniku i računarske komunikacije.