

# Energetski efikasan rendezvous protokol

Mirko Kosanović, Miloš Kosanović  
Savremene računarske tehnologije  
Visoka tehnička škola strukovnih studija  
Niš, Srbija  
[mirko.kosanovic@vtsnis.edu.rs](mailto:mirko.kosanovic@vtsnis.edu.rs)  
[milos.kosanovic@vtsnis.edu.rs](mailto:milos.kosanovic@vtsnis.edu.rs)

Mile Stojčev  
Elektronski fakultet  
Univerzitet u Nišu  
Niš, Srbija  
[mile.stojcev@elfak.ni.ac.rs](mailto:mile.stojcev@elfak.ni.ac.rs)

**Sadržaj**—Višegodišnji rad senzorskih čvorova u Bežičnim Senzorskim Mrežama je ključni uslov koji definiše njihov energetski efikasan rad. U praksi se primenjuju nekoliko metoda koji treba da reše ovaj problem. U ovom radu, koristimo jednu jednostavnu, ali efikasnu tehniku, nazvanu modifikovana šema kod *Rendezvous* protokola. Kako je kolizija jedan od najvećih uzroka povećane potrošnje u senzorskom čvoru, u radu smo analizirali kako ova modifikovana šema utiče na smanjivanje broja kolizija kao i racionalnijoj potrošnji električne energije kod rešavanja nastale kolizije. Predloženo rešenje sadrži izmene u odnosu na standardne šeme *Rendezvous* protokola koje se pre svega odnose na jedinstvenu identifikaciju svakog senzorskog čvora kao i smanjivanju vremena potrebnog za uspostavljanje normalnog režima rada nakon nastale kolizije.

**Ključne riječi**—senzorski čvor; racionalna potrošnja; rendezvous protokol; kolizija

## I. UVOD

Bežične senzorske mreže (BSM) se sastoje od velikog broja jevtinih, malih i autonomnih senzorskih čvorova (SČ), koji su sposobni da potpuno samostalno prikupljaju, obrađuju i razmenjuju podatke. Oni su proizvoljno raspoređeni u širokom geografskom području i uglavnom imaju samostalno napajanje u vidu malih baterija. Kako su gabariti jednog SČ, a samim tim i kapacitet baterije, jako mali, njen životni vek bio bi ograničen na svega nekoliko dana ako bi svi njegovi delovi radili neprekidno. Imajući u vidu da je promena baterija u većini slučajeva praktično neizvodljiva, proizilazi da je efikasna potrošnja energije presudni faktor koji utiče na duži životni vek jednog SČ a samim tim i aplikacije u BSM [1].

U velikom broju aplikacija za BSM imamo relativno jako slab saobraćaj između SČ-ova (od 0,01 do 10 paketa/s). Sve te pakete karakteriše veoma mala dužina, obično između 64-128 bajtova. Sa takvim intenzitetom saobraćaja, SČ većinu svog životnog veka provede u neaktivnom stanju, osluškujući kanal (u tom vremenskom periodu prijemnik SČ je uključen iako ne prima nikakve podatke). Na primer: ako pretpostavimo da imamo 5 susednih SČ, i da oni razmenjuju po jedan paket/s, da je dužina paketa 64B i da brzina slanja primo-predajnika 40 kbps, lako možemo odrediti efikasno radno vreme. Svaki SČ troši samo 1,28% od svog aktivnog vremena za slanje paketa i 6,4% vremena za prikupljanje i prijem paketa. Iz ovog primera se vidi da SČ provede u neaktivnom stanju više od 92% svog vremena. Shodno tome, možemo zaključiti da je energija rasipanja jako velika zbog činjenice da SČ troši veliku količinu vremena u stanju osluškivanja kanala. U cilju da se produži

životni vek SČ neophodno je da se ovo vreme skрати, što se postiže primenom neke od mnogobrojnih PMT (*Power Management Technique*) tehnika [2, 3].

Mi ćemo fokusirati naš interes na jednu od najčešće korišćenih tehnika, *duty cycling* (DC) tehniku. Ona predstavlja efikasnu i često korišćenu metodu koja je postala neizostavan deo gotovo svake aplikacije u BSM. Princip rada ove metode je vrlo jednostavan: držati hardverske komponente što duže u režimu smanjene potrošnje (*idle* ili *sleep* režim), a što manje vremena u režimu normalne potrošnje (*active* režim). Međutim, ovaj jednostavan princip nije lako primeniti jer on zahteva i neke dodatne tehnike kako bi se obezbedio pouzdan rad svih SČ-ova u mreži. Tu se pre svega misli na preciznu vremensku sinhronizaciju svih SČ-ova u mreži, koja treba da omogući nesmetanu komunikaciju između SČ-ova. Istraživanja pokazuju da čak i pri DC faktoru od samo 1-2%, postiže se ušteda u potrošnji električne energije za jedan red veličine veći od potrošnje svih ostalih komponentata u SČ [1]. S druge strane, *duty cycling* dovodi do kompleksnih komunikacijskih šema koji uključuju prozivanje SČ (*polling*) i vremensko deljenje komunikacionog kanala. Naime, da bi dva SČ primila ili predala podatke potrebno je da se komunikacija odvija u isto vreme. Ovaj problem naziva se problem susretanja dva SČ (*rendezvous*) [4].

U ovom radu razmatrana je jedna od najrasprostranjenijih tehnika za uštedu energije a to je primena *rendezvous* protokola. U tom smislu analizirana je standardna *rendezvous* šema (DLDC-MAC protokol) [5]. Prikazane su prednosti i mane ovog protokola i u cilju rešavanja uočenih nedostataka predložena je njegova modifikacija. Nakon toga, data je procena standardne potrošnje energije u SČ i na osnovu nje izvršena je uporedna analiza potrošnje električne energije u odnosu na broj kolizija kod standardnog i predloženog tj. modifikovanog rešenja. Kao parametar kod ove analize korišćena je dužina perioda prozivanja SČ-ova (*beacon* perioda).

## II. POTROŠNJA ELEKTRIČNE ENERGIJE U SENZORSKOM ČVORU

U svakom SČ možemo da razlikujemo četiri glavne aktivnosti tokom kojih on troši energiju. To su prikupljanje, obrađivanje, primopredaja (prijem i slanje) i pamćenje podataka [2]. Sa druge strane dobro je poznato da deklarirani kapacitet energije u bateriji, koja se nalazi u SČ-u, nije uvek ekvivalentan energiji koju baterija može da preda. Da bi se izračunala maksimalna količina energije koja može da se

iskoristi iz baterije, potrebno je imati u vidu sledeće dve karakteristike, koje imaju uticaj na ukupnu potrošnju baterije, i to: za svaku bateriju se definiše struja curenja (*leakage current*), koja predstavlja direktnu posledicu samopražnjenja baterije i drugu koja se odnosi na potrošnju energije SČ-a koja se utroši prilikom prelaza iz jednog u drugi način rada (*switching mode*). Međutim, kod bežičnih komunikacija javljaju se i neka dodatna rasipanja energije o čemu treba voditi računa kod projektovanja protokola. To se javlja pre svega zbog sledećih razloga [4]:

a) **Idle listening** – kada SČ ne prima pakete on mora da osluškuje kanal kako bi mogao da detektuje budući dolazeći paket, što uzrokuje rasipanje energije.

b) **Collision** - kada se veći broj SČ-ova nalazi na malom prostoru (u dometu su), preklapanje istovremenog slanja paketa je uobičajena pojava ako se ona efikasno ne kontroliše. Tada se svi preklapljeni paketi odbacuju, pa SČ obično ponovo šalju te iste pakete, što opet izaziva dodatno trošenje energije za njihovo slanje i prijem.

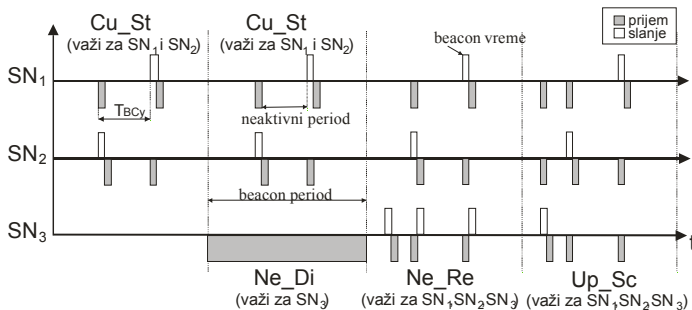
c) **Overhearing** - kada neki SČ šalje neki paket, taj paket primaju svi okolni SČ-ovi, i pored toga što taj paket nije namenjen svim SČ-ima, što predstavlja još jedan nepotreban izvor trošenja energije.

d) **Over-emitting** – SČ šalje pakete drugom SČ-u koji tog trenutka nije u aktivnom stanju ili jednostavno nije u mogućnosti da primi taj paket. Isti paket potrebno je ponovo poslati, što takođe zahteva dodatnu potrošnju energije.

e) **Control packet overhead** - prisustvo dodatnih upravljačkih paketa u BSM-i, kao što su zahtevi za slanje (RTS), potvrda da može da se šalje (CTS), potvrda ispravnog prijema (ACK), slanje *beacon*-a, prenos i prijem dela paketa koji nisu korisni (*payload*) podaci kao što su predajna i prijemna adresa, broj paketa, dužina paketa i kontrolni kod ispravnosti paketa, zahteva dodatnu potrošnju energije. □

### III. KARAKTERISTIKE RENDEZVOUS PROTOKOLA

Kako bi prikazali osnovnu ideju *rendezvous* tehnike, kao i ukazali na njene prednosti i mane, izvršena je analiza jedne standardne *rendezvous* šeme (DLDC-MAC protokol), koja je prikazana u radu [4]. Da bi analiza bila jasnija koristili smo tri SČ-a:  $SN_1$ ,  $SN_2$  i  $SN_3$  koja su prikazana na Slici 1. Međutim, broj SČ-ova uopšte ne utiče na donete zaključke koji važe i za potpuno proizvoljan broj SČ-ova.



Slika 1. Standardna šema *Rendezvous* protokola

Osnovna ideja sastoji se u tome da svaki SČ periodično šalje kratku poruku, *beacon*, kako bi signalizirao svoje prisustvo u BSM. Takođe, u tačno određenim vremenskim intervalima svaki SČ se aktivira kako bi mogao da primi *beacon*-e od svojih suseda. Na osnovu vremena prijema *beacon*-a, SČ proračunava naredno vreme prijema za svaki susedni SČ, uključujući i odgovarajuća *guard* vremena. *Guard* vreme predstavlja dodatno proširenje aktivnog perioda prijema *beacon*-a, koje je neophodno uzeti u obzir kod proračuna, kako bi se sa većom sigurnošću obezbedila istovremena komunikacija (*rendezvous*) između dva SČ-a. To vreme je posledica nesavršenosti i neujednačenosti kvarcnih oscilatora koji se nalaze u svakom SČ-u (*drift*). Imajući u vidu da SČ zna vremena kada njegovi susedi šalju svoje pakete moguće je da on većinu vremena provede i neaktivnom stanju i da se samo periodično prebacuje u aktivno stanje kada naiđu odgovarajuća vremena prijema *beacon*-a od svojih suseda. Drugim rečima, SČ najviše vremena provodi u *sleep/idle* stanju štedeći svoju potrošnju električne energije. Na Slici 1. prikazan je scenario aktivnosti SČ-ova koji koriste standardni *rendezvous* protokol (DLDC-MAC). Pretpostavka je da je osnovni period, *beacon* period  $T_{BP}$ , unapred definisan i poznat je svim SČ-ima u BSM-i. *Beacon* period se periodično ponavlja i on obuhvata sva neophodna vremena za slanje i prijem paketa ka i od SČ-ova, kao i vreme koje SČ provede u neaktivnom stanju (*sleep*). Sa Slike 1. može da se vidi da postoje četiri različite faze kroz koje prolazi svaki od SČ-ova. To su sledeće faze:

**Tekuća faza ( $Cu\_St$ )** – Svaki registrovani SČ (u navedenom primeru,  $SN_1$  i  $SN_2$ ) šalje periodično kratku poruku, *beacon*, i nakon toga prelazi, jedan kraći period, u režim prijema kako bi mogao da prihvati *beacon*-e od novih susednih SČ-ova koji žele da se priključe BSM-i. *Beacon* period je potpuno identičan za sve SČ-ove. Nakon što prihvati *beacon*, SČ procenjuje svoje sledeće *beacon* vreme uzimajući u obzir trajanje *beacon* perioda, vreme prijema *beacon*-a i *guard* period. Treba imati u vidu da SČ procenjuje *beacon*-e za sve susedne SČ-ove, što mu olakšava da ova procena bude tačnija. Zahvaljujući ovome SČ može da većinu svog *beacon* perioda provede u neaktivnom stanju i da pređe u aktivno stanje samo da bi primio *beacon*-e od suseda i da pošalje svoj *beacon*. Na ovaj način, svaki SČ periodično aktivno učestvuje u komunikaciji jer je uvek svestan svojih susednih SČ-ova sa kojima komunicira, tj. zna *beacon* vremena svih susednih SČ-ova. Treba imati na umu da je svaka od faza ( $Cu\_St$ ,  $Ne\_Di$ ,  $Ne\_Re$  i  $Up\_Sc$ ) podeljena u nekoliko vremenskih slotova, čiji je period trajanja nešto duži, za *guard* period, od potrebnog vremena za slanje ili prijem *beacon*-a od susednih SČ-ova. Faza  $Cu\_St$  je dominantna od svih ostalih faza, jer u njoj svaki SČ provede najveći deo svog životnog veka (preko 99 %).

**Faza pronalaženja ( $Ne\_Di$ )** – predstavlja početnu fazu kroz koju prolazi svaki novi SČ koji želi da se priključi BSM. Naime, pri prvom uključenju, SČ ulazi u  $Ne\_Di$  fazu u kojoj je njegov prijemnik neprekidno uključen ceo jedan *beacon* period. Osnovni cilj ove faze je da novi SČ detektuje sve aktivne SČ-ove iz svoje okoline (dometa) i da registruje njihova *beacon* vremena. Tako u odnosu na Sliku 1., za vreme ovog perioda  $SN_3$  prihvata *beacon*-e od  $SN_1$  i  $SN_2$ .

**Faza registracije ( $Ne\_Re$ )** – služi da bi se novi SČ prijavio susednim SČ-ima. Nakon prijema *beacon* vremena od susednih

SČ-ova, u fazi *Ne\_Re*, novi SČ izračunava, tj. određuje svoje *beacon* vreme (prazan vremenski slot za emitovanje svog *beacon*-a) u okviru *beacon* perioda. Kako je u prethodnoj fazi registrovao *beacon* vremena susednih SČ-ova (*SN<sub>1</sub>* i *SN<sub>2</sub>*), *SN<sub>3</sub>* sada može, u odgovarajućim vremenskim slotovima, da obavesti susedne SČ-ove o svom prisustvu u BSM-i. To je neophodno kako bi već registrovani SČ-ovi, *SN<sub>1</sub>* i *SN<sub>2</sub>*, mogli da novoprijavljeni SČ, *SN<sub>3</sub>*, uključe u okviru svog *beacon* perioda, tj. da uključe prijemni režim u novo formiranom vremenskom slotu koji odgovara *SN<sub>3</sub>*.

**Faza prihvatanja (*Up\_Sc*)** - prema aktivnostima ova faza je vrlo slična fazi *Cu\_St*, sa samo jednim izuzetkom koji se odnosi na dodavanje *beacon* vremena novo prijavljenog *SN<sub>3</sub>* u okviru *beacon* perioda svih SČ-ova koji su u dometu i koji su nesmetano prihvatili poruku o novom SČ-u u mreži.

Neke od glavnih prednosti analiziranog protokola su:

1. Na jedan potpuno decentralizovan način rada, uspešno je sinhronizovano vreme aktiviranja svih SČ-ova u okolini, čak i u sredinama sa vrlo nepouzdanim komunikacionim linkovima.

2. Većina problema vezanih za bežične komunikacije kao što su: *clock drift*, asimetrične veze, skriveni terminal i otežane komunikacije su uspešno prevaziđeni.

3. Primenom ovog protokola znatno se štedi električna energija, pa se zato očekuje znatno produženje životnog veka svih SČ-ova, a samim tim i život cele BSM.

Međutim analizirajući ovu šemu možemo videti da postoje i neki nedostaci, koje ovaj protokol ne može da reši, a to su:

1. Određivanje dužine adekvatnog *beacon* perioda kada se dva ili više SČ istovremeno prijave u BSM.

2. Određivanje adekvatnog *beacon* vremena (vremenski slot) za svaki novoprijavljeni SČ.

3. Kako *beacon* vremena nisu unapred definisana (vremenski slotovi) može da dođe do njihovog preklapanja usled razlika u frekvencijama lokalnih satova (*clock drift*). Tada je potrebno da se izaberu nova *beacon* vremena (vremenski slotovi) koji opet mogu da prouzrokuju nova preklapanja i dodatne probleme.

4. Problem rešavanja kolizije koja može da se pojavi prilikom istovremenog prijavljivanja dva ili više SČ-a u BSM.

#### IV. PREGLED MODIFIKOVANE ŠEME

Da bi se prevazišli uočeni problemi kod standardnih *rendezvous* protokola, potrebno je da se izvrše neke promene u odnosu na prethodno opisanu šemu rada protokola DLDC-MAC. Osnovni principi rada tog modifikovanog protokola prikazani su u Ref. [7]. U ovom radu su razmatrane performanse predloženeog protokola koji se odnose na: a) maksimalni broj SČ-ova u okviru BSM u zavisnosti od primenjenog faktora DC (*Duty Cycle*) kao i primenjenog kvarcnog oscilator u SČ; i b) maksimalna dužina *beacon* perioda u zavisnosti od broja istovremeno aktivnih SČ-ova.

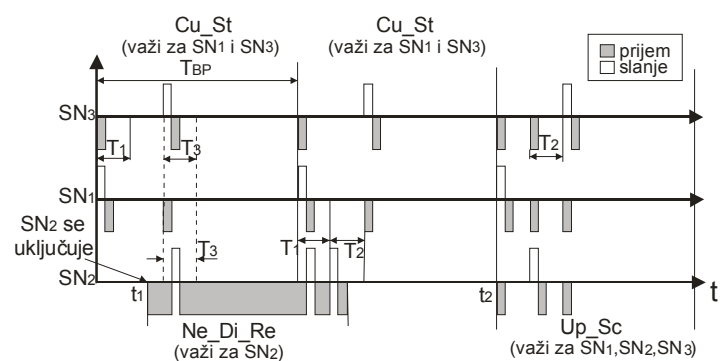
Predloženi protokol je usvojio sve dobre osobine DLDC-MAC protokola koje uspešno rešavaju problem loših linkova, sakrivenog terminala, asimetričnih veza i klok drift-a. Da bi

rešili osnovni problem DLDC-MAC protokola, izbegavanje kolizije kod prijavljivanja novih SČ-ova i određivanje adekvatnih vremenskih slotova za slanje *beacon*-a, u predloženom protokolu se pošlo od činjenice da svi SČ-ovi budu jedinstveno identifikovani putem svog ID broja, koji se dodeljuje svakom SČ u inicijalnom procesu formiranja (ID=1,2,3,...,n, gde je n maksimalan broj SČ u mreži). Uvođenje jedinstvenog ID broja za svaki SČ stvara mogućnost da unapred može da se definiše jedinstveno *beacon* vreme (trenutak kada počinje vremenski slot) za sve SČ-ove u okviru *beacon* perioda. Na osnovu ukupnog broja susednih SČ-ova, određuje se minimalno trajanje *beacon* perioda, koje direktno zavisi od veličine DC faktora koji želi da se postigne. Uvođenje jedinstvenog ID broja omogućilo je da više ne postoji početna faza inicijalizacije pa je postupak formiranja stabilnog režima rada sada ovde smanjen za jednu fazu. Samim tim smanjeno je kašnjenje kod prijavljivanja novih SČ-ova u već postojeću BSM. Scenario uspostavljanja normalnog režima rada prilikom prijavljivanja novih SČ-ova prikazan je na Slici 2. i on se sastoji iz sledećih faza:

**Tekuća faza (*Cu\_St*)** – ova faza je identična kao i već opisana faza *Cu\_St* kod DLDC-MAC protokola.

**Faza pronalaženja i prijavljivanja (*Ne\_Di\_Re*)** – jedinstven ID broj svakog SČ-a unapred definiše tačnu poziciju njegovog vremenskog slota u okviru *beacon* perioda. Svaki novi SČ (*SN<sub>2</sub>* na Slici 2.) koji želi da se prijavi u BSM-u započinje svoju aktivnost sa fazom *Ne\_Di\_Re* u kojoj se on aktivira i nalazi se u režimu prijema. Kada naiđe vremenski slot *T<sub>3</sub>* (*T<sub>1</sub>*), *SN<sub>2</sub>* prihvata *beacon* koji je poslao *SN<sub>3</sub>* (*SN<sub>1</sub>*) i odmah nakon toga obaveštava *SN<sub>3</sub>* (*SN<sub>1</sub>*) o svom prisustvu u BSM-i. U trenutku kada naiđe *time slot T<sub>2</sub>*, *SN<sub>2</sub>* regularno označava svoje prisustvo u BSM-i šaljući svoj *beacon* (ova informacija je potrebna novim SČ-ima koji se i isto vreme prvi put prijavljuju u BSM-u). Faza *Ne\_Di\_Re* je ujedinila faze *Ne\_Di* i *Ne\_Re* u jednu jedinstvenu, čime je, prilikom prijavljivanja novih SČ-ova, smanjeno vreme za uspostavljanje normalnog režima rada za jedan *beacon* period.

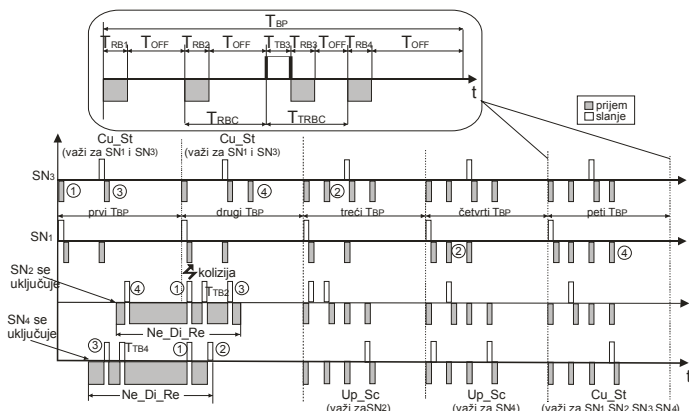
**Faza prihvatanja (*Up\_Sc*)** – ova faza je identična fazi *Up\_Sc* kao kod DLDC-MAC protokola.



Slika 2. Modifikovana šema bez kolizija

Kao što se sa Slike 2. vidi, svaki novi SČ koji se prijavljuje (*SN<sub>2</sub>*), na samom početku ulazi u **receive** režim rada (faza *Ne\_Di\_Re*) kako bi mogao da prihvati *beacon*-e od susednih

SČ-ova. **Receive** režim rada traje ceo jedan *beacon* period, koji je unapred definisan. U okviru njega svaki susedni SČ emituje po jedan *beacon*, i to u periodu (vremenski slot) koji je u direktnoj srazmeri sa ID brojem koji svaki SČ dobija pri početnoj inicijalizaciji sistema. Na taj način izbegava se moguća kolizija, a ujedno obezbeđuje se da svaki novi SČ prihvati *beacon*-e od svih susednih SČ-ova. U trenutku kada prihvati neki *beacon* od susednih SČ-ova, novi SČ ulazi u **send** režim rada i odmah šalje informaciju o svom prisustvu (svoj ID broj) tom istom SČ-u. Na osnovu te informacije stari SČ-ovi sada ubacujuju novi interval osluškivanja na odgovarajuće mesto u *beacon* periodu (odgovarajući vremenski slot), koje je definisano primljenim ID brojem od novoprijavljenog SČ-a i istovremeno šalju potvrdu novoprijavljenom SČ ( $SN_2$ ) da je uspostavljena i povratna veza između njih. Ovo je potrebno kako bi svaki SČ eliminisao nepotrebno osluškivanje i trošenje energije za SČ-ove sa kojima ne postoji obostrana veza (asimetrične veze, SČ-ovi koji ne rade ili ne šalju informacije). Već u sledećem *beacon* periodu, uspostavlja se normalni režim rada, u okviru koga se svaki SČ nalazi samo jednom u **send** režimu rada (svi ostali SČ-ovi su tada u **receive** režimu rada) a  $n$  puta u **receive** režimu rada, gde  $n$  predstavlja ukupan broj SČ-ova u susedstvu. **Receive** režim rada koji sledi odmah nakon **send** režima rada je neophodan kako bi se prihvatili ID brojevi novih SČ-ova iz okoline (susednih SČ-ova).



Slika 3. Modifikovana šema sa kolizijama

Uvođenjem jedinstvenog ID za svaki SČ i definisanjem vremenskog slot, u kome samo jedan SČ može da šalje podatke, problem kolizije je izbegnut kod uspostavljanja normalnog režima rada, ali taj problem ostaje i dalje u inicijalnom režimu, jer je moguće da se istovremeno prijave dva ili više SČ-a u okviru unapred definisanog *beacon* perioda  $T_{BP}$ . Rešavanje ovog problema prikazano je na Slici 3. gde je prikazano istovremeno prijavljivanje dva nova SČ,  $SN_2$  i  $SN_4$ . Ako u periodu *Ne\_Di\_Re* dođe do kolizije kod slanja inicijalnih ID, istovremena prijava prisustva dva ili više SČ-a, novoprijavljeni SČ-ovi neće u tom periodu primiti potvrdu da je njihov ID prihvaćen od susednih SČ-ova. Neprihvatanje potvrde može da znači ili postojanje asimetrične veze ili kolizije. U tom slučaju svaki od novoprijavljenih SČ-ova nastavljaju da emituju *beacon*-e u okviru svog vremenskog slot i istovremeno počinju da broje *beacon* periode. Kada

dođe period  $T_{BP}$  (ID), koji je jedinstven za svaki SČ, SČ šalje svoj ID broj susednim SČ-ima sa kojima nije uspostavio obostranu vezu (SČ od kojih nije dobio potvrdu da su prihvatili njegov ID). Na taj način novoprijavljeni SČ-ovi ponavljaju inicijalnu fazu ali ovog puta izbegavaju pojavu kolizije jer je period  $T_{BP}$  (ID) jedinstven za svaki SČ (period  $Up\_Sc(ID)$  na Slici 3.). Nakon toga uspostavlja se normalni režim rada u kome bi u narednom periodu  $Up\_Sc(ID+1)$  SČ-ovi poslali novom SČ poruku prihvatanja njegovog ID. Ukoliko ni sada novi SČ ne dobije ovu poruku to je znak da postoji asimetrična veza između njih.

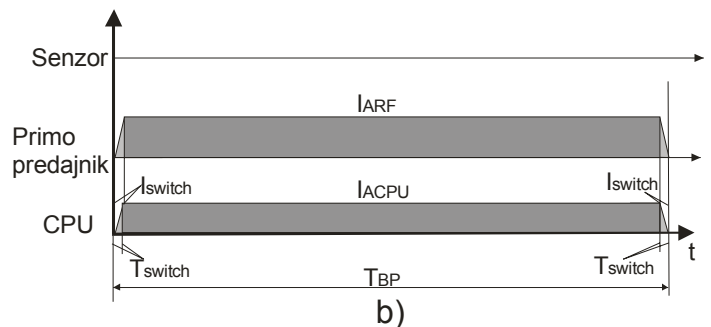
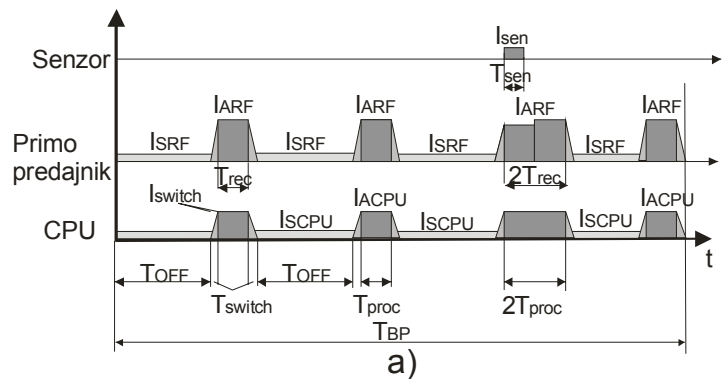
#### V. PROCENA POTROŠNJE ENERGIJE U SENZORSKOM ČVORU

Ukupna potrošnja energije za SČ može se prikazati sledećom jednačinom

$$E_{SČ} = \int_0^T I_{AVR}(t)V(t) dt \quad (1)$$

gde  $E_{SČ}$  predstavlja utrošenu energiju za vreme životnog veka SČ-a;  $I_{AVR}$  prosečna struja tokom životnog veka SČ-a;  $V$  je napon napajanja a  $T$  životni vek SČ-a.

U nameri da se pojednostavi analiza potrošnje SČ pretpostavićemo da je za vreme životnog veka SČ napon napajanja konstantan. Da bi mogli tačno da izračunamo  $E_{SČ} = I_{AVR} * V * T$  neophodno je da znamo koliko iznosi prosečna struja  $I_{AVR}$  i to pojedinačno po svakoj komponenti u SČ-u. Na Slikama 4a. i 4b. respektivno, dati su profili potrošnje (jačina struje) po pojedinim komponentama SČ-a (procesor, primopredajnik i senzor), za vreme faza *Cu\_St* i *Ne\_Di\_Re*.



Slika 4. Profil potrošnje el.energije a) *Cu\_St*, b) *Ne\_Di\_Re*

Na osnovu poznavanja principa rada protokola, možemo da pišemo da važi sledeće:

$$I_{AVR} = hI_{AU} + (1-h)I_{AD} + I_{SD} \quad (2)$$

gde je:  $I_{AU}$  – prosečna struja u SČ-u za vreme faze  $Up\_Sc$ ;  $I_{AD}$  – prosečna struja u SČ-u za vreme faze  $Ne\_Di\_Re$ ;  $I_{SD}$  – struja curenja baterije; i  $h$  – vremenski faktor koji nam pokazuje koliko dugo je SČ tokom svog životnog veka proveo u fazi  $Ne\_Di\_Re$  u odnosu na fazu  $Cu\_St$ . Treba imati u vidu da je prosečna potrošnja za vreme faze  $Cu\_St$  i faze  $Up\_Sc$  približno ista, kao i da se tokom celog životnog veka SČ-a, obično faza  $Ne\_Di\_Re$  dogodi najmanje jednom (kod SČ-ova koji se prvi put prijavljuju u BSM-i ili nakon reseta SČ-a), tako da važi da  $\lim_{t \rightarrow \infty} h \rightarrow 1$ .

Prosečna struja  $I_{AU}$  za vreme faza  $Cu\_St$  i  $Up\_Sc$  (vidi Sliku 4.a), je jednaka:

$$I_{AU} = I_{AUCPU} + I_{AURF} + I_{AUsen} \quad (3)$$

Gde je:

$$I_{AUsen} = I_{sen} \frac{T_{sen}}{T_{BP}}, \quad (4)$$

$$I_{AURF} = nI_{SRF} \frac{T_{OFF}}{T_{BP}} + 2nI_{switch} \frac{T_{switch}}{T_{BP}} + (n+1)I_{ARF} \frac{T_{rec}}{T_{BP}}, \quad (5)$$

$$I_{AUCPU} = nI_{SCPU} \frac{T_{OFF}}{T_{BP}} + 2nI_{switch} \frac{T_{switch}}{T_{BP}} + (n+1)I_{ACPU} \frac{T_{proc} + T_{guard}}{T_{BP}}. \quad (6)$$

gde  $n$  predstavlja ukupan broj susednih SČ-ova sa kojima se komunicira.

Prosečna struja,  $I_{AD}$ , za vreme faze  $Ne\_Di\_Re$  (vidi Sliku 4b.), je jednaka:

$$I_{AD} = I_{ACPU} + I_{ARF} + 2I_{switch} \frac{T_{switch}}{T_{BP}} \quad (7)$$

Sa Slike 4a. možemo da vidimo da  $I_{ACPU}$ ,  $I_{ARF}$ ,  $I_{sen}$  odgovaraju trenutnoj vrednosti potrošnje struje koja je u procesoru, primopredajniku i senzoru, respektivno, kada je SČ u aktivnom režimu rada, dok  $I_{SCPU}$ ,  $I_{SRF}$  odgovara potrošnji struje u procesoru i primopredajniku za vreme kada su oni u neaktivnom režimu rada. Kod procesora MSP430F123 potrebno je  $T_{switch}=6 \mu s$  da on iz aktivnog režima rada pređe u neaktivni režim LPM3 (*Low Power Mode 3*). Takođe važi pretpostavka da se struja za vreme perioda  $T_{switch}$  linearno menja i da se može izračunati sledećom formulom:

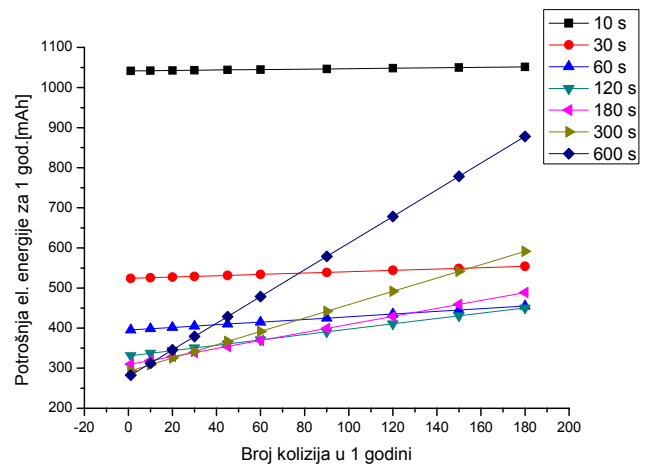
$$I_{switch} = \frac{(I_{ACPU} + I_{ARF} - I_{SCPU} - I_{SRF})}{2} \quad (8)$$

gde je:  $I_{ACPU}$  ( $I_{SCPU}$ ) i  $I_{ARF}$  ( $I_{SRF}$ ) odgovaraju potrošnji struje u procesoru i primopredajniku za vreme aktivnog (neaktivnog) stanja (vidi Sliku 4a.). Zamenom jednačina (4), (5), (6) i (7) u

jednačini (2) dobija se formula preko koje može da se izračuna prosečna potrošnja struje  $I_{AVR}$ . Ostale vrednosti koje su korišćene prilikom izračunavanja prosečne potrošnje iznose:  $I_{ACPU}=300 \mu A$ ,  $I_{ARF}=19700 \mu A$ ,  $I_{SCPU}=1.6 \mu A$ ,  $I_{SRF}=1 \mu A$  i  $I_{sen}=1000 \mu A$ . Za nepunjivu *lithium-ion* bateriju uzeto je da struja samopražnjenja,  $I_{SD}$ , prouzrokuje gubitak kapaciteta baterije od 2% po godini.

## VI. ANALIZA PERFORMANSI MODIFIKOVANOG PROTOKOLA

Da bi smo prikazali energetske efikasnost predloženih modifikacija kod modifikovane *rendezvous* šeme, analizirali smo koliko je porasla potrošnja energije u SČ u toku jedne godine prilikom rešavanja kolizija. Pretpostavili smo da svaki SČ u okolini ima 10 SČ-ova sa kojima stalno komunicira u okviru jednog *beacon* perioda. Prvo smo na Slici 5. prikazali kako zavisi potrošnja električne energije u zavisnosti od broja kolizija i to za različite *beacon* periode. Analiza je rađena za period od jedne godine i na osnovu prosečne potrošnje SČ koja je objašnjena u prethodnom poglavlju.



Slika 5. Potrošnja energije u zavisnosti od broja kolizija za različite *beacon* periode

Analizirajući dobijene grafike sa Sl.5. možemo zaključiti sledeće:

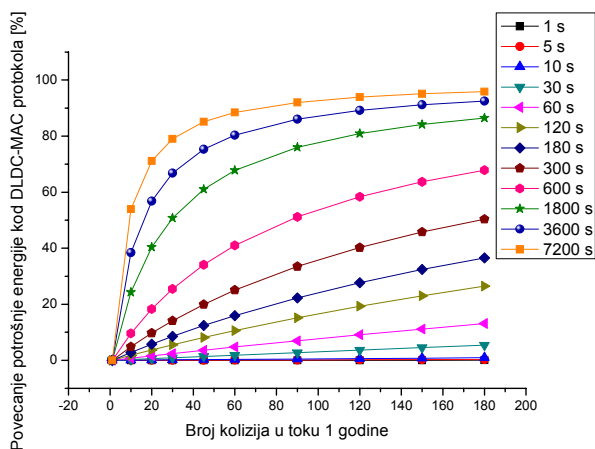
a) Povećavanje trajanja *beacon* perioda smanjuje potrošnju energije u SČ ali se javlja problem što tada imamo veliko kašnjenje u detektovanju pojedinih događaja, jer je potrebno više vremena da protekne dok se ne prozovu svi SČ-ovi u okruženju. Samim tim ovakvi sistemi nisu pogodni za rad u *real-time* okruženju.

b) Za sve *beacon* periode potrošnja električne energije raste ako raste broj kolizija što se i očekivalo.

c) Sa porastom trajanja *beacon* perioda potrošnja električne energije brže raste kako raste broj kolizija. To je i očekivano jer je za razrešavanje svake kolizije neophodno da SČ pokrene fazu pronalaženja i prijavljivanja SČ-ova (faza  $Ne\_Di\_Re$ ). Kako je potrošnja u ovoj fazi direktno srazmerna dužini *beacon* perioda sasvim je razumljivo da ona brže raste kod dužih nego kod kraćih *beacon* perioda.

Na Slici 6. prikazani su grafici koji prikazuju relativan odnos potrošnje električne energije između DLDC -MAC i

predloženog protokola za period od jedne godine u funkciji broja kolizija i to za različite *beacon* periode,  $T_{BP}$ , kao parametar. Analizom rezultata prikazanih na Slici 6. možemo zaključiti sledeće :



Slika 6. Relativan odnos potrošnje energije u SČ kod standardnog i modifikovanog rešenja u funkciji broja kolizija

a) Bez obzira na broj kolizija potrošnja kod DLDC -MAC protokola je uvek veća u odnosu na predloženi protokol.

b) Najbolje karakteristike modifikovani protokol pokazuje kod manjeg broja kolizija, do 50 kolizija. Sa povećanjem broja kolizija potrošnja energije kod standardnog i modifikovanog rešenja se približavaju tj. relativni odnos teži jedinici.

c) Za oba protokola sa povećanjem broja kolizija povećava se potrošnja energije. Međutim, predloženi protokol pokazuje bolje rezultate sa povećanjem broja kolizija, tako na primer: za  $T_{BP} = 180s$  i 10 kolizija ušteda potrošnje energije iznosi 2,82%, dok je za  $T_{BP} = 180s$  i 60 kolizija ušteda je 15,95 %.

d) Za duže *beacon* periode  $T_{BP}$  predloženi protokol troši manje energije. Tako na primer za  $T_{BP} = 10s$  i 30 kolizija ušteda je 0,15%, dok je za  $T_{BP} = 300 s$  i isti broj kolizija ušteda 14,11 % .

## VII. ZAKLJUČAK

Primarni cilj kojem se teži kod razvijanja aplikacije za BSM je efikasna energetska potrošnja. Kako postoje različiti nepotrebni odlivi energije u BSM, postoje i veliki broj tehnika koje nam pomažu da većinu ovih odliva svedemo na što manju meru. U ovom radu analizirali smo i modifikovali jednu od najprimenjivijih tehnika za međusobnu komunikaciju SČ-ova u BSM po imenu *rendezvous* šema. U cilju uštede potrošnje električne energije izvršili smo neke modifikacije u okviru standardne šeme a koje su se odnosile pre svega na uvođenje jedinstvenog ID broja za svaki SČ. To nam je omogućilo smanjivanje aktivnosti svakog SČ prilikom

njegovog prijavljivanja u već postojeću BSM, kao i izbegavanje mogućih kolizija prilikom istovremenog prijavljivanja većeg broja SČ-ova. Zahvaljujući uvedenim izmenama potrošnja električne energije u modifikovanom protokolu je smanjena, i taj procenat smanjenja raste kako raste dužina *beacon* perioda, kao i ako se broj kolizija povećava. Samim tim, dobijeni rezultati u potpunosti opravdavaju uvođenje predloženih modifikacija u *rendezvous* šemu jer se postiže značajna ušteda u potrošnji električne energije u SČ..

## LITERATURA

- [1] Raghunathan V., Ganerival S., Srivastava M., Emerging techniques for long lived wireless sensor networks, IEEE Communication Magazine, Vol. 44, No. 4, 2006, pp. 108-114
- [2] P. K. Dutta, D. E. Culler, System Software Techniques for Low Power Operation in Wireless Sensor Networks, In Proceedings of the ICCAD'05, San Jose, California, USA, 2005, pp. 925-932
- [3] M. Kosanovic, M. Stojcev, "Sensor Node Lifetime Prolonging", TELFOR 2012, Beograd, 20-22.11.2012, pp.178-181
- [4] En Yi Lin, A Comprehensive Study of Power-Efficient Rendezvous Schemes for Wireless Sensor Networks, PhD thesis, University of California, Berkeley, 2005
- [5] M.Brzozowski, K.Piotrowski, P.Langendoerfer, "A Cross-layer approach for data replication and gathering in decentralized long-living wireless sensor networks", ISADS 2009, The 9th International Symposium on Autonomous Decentralized System, 2009
- [6] D. Christmann, R. Gotzhein, M. Krämer, M. Winkler, Flexible and energy-efficient duty cycling in wireless networks with MacZ, Proc. 10th Annual Int New Technologies of Distributed Systems (NOTERE) Conf, IEEE, 2010, pp. 121-128
- [7] M.Kosanovic, M.Stojcev, "Energy Efficient Rendezvous Protocol for Wireless Sensor Networks", 2nd Mediterranean Conference on Embedded Computing, MECO – 2013, Budva, Montenegro, pp. 215-218

## ABSTRACT

Many years of experience of sensor nodes in wireless sensor networks is a key condition that defines their energy efficient operation. In practice, apply several methods to solve this problem. In this paper, we use a simple but effective technique, called the modified scheme at Rendezvous protocol. As the collision one of the biggest causes of increased spending, in this paper we discuss how this modified scheme affects reducing the number of collisions as well as reducing electricity consumption that is necessary for solving the resulting collision. The modified scheme contains changes to the standard scheme rendezvous protocol which primarily relate to the unique identification of each of the sensor nodes and reducing the period of establishing normal mode after the collision occurred.

## ENERGY EFFICIENT RENDEZVOUS PROTOCOL

Mirko Kosanović, Miloš Kosanović, Mile Stojčev