

Upotreba Rendezvous protokola za duži životni vek bežične senzorske mreže

Mirko Kosanović

Savremene računarske tehnologije
Visoka tehnička škola strukovnih studija
Niš, Srbija
mirko.kosanovic@open.telekom.rs

Mile Stojčev

Elektronski fakultet/Katedra za elektroniku
Univerzitet u Nišu
Niš, Srbija
mile.stojcev@elfak.ni.ac.rs

Sadržaj—Jedan od najvažnijih zadataka koje projektanti trebaju da reše prilikom projektovanja aplikacija za rad u Bežičnim Senzorskim Mrežama (BSM), je racionalno trošenje električne energije. Senzorski čvorovi (SČ), kao sastavni delovi jedne BSM, napajaju se putem baterija što znači da imaju ograničenu količinu energije na raspolaganju. Ispunjavanje ovog uslova nije lak zadatak, pre svega zbog ograničenih resursa same mreže kao i prirodnih uslova u kojima one rade. U ovom radu predstavili smo jednu grupu protokola, *rendezvous* protokole, koji se koriste u BSM u cilju prevazilaženja nepotrebne potrošnje el. energije. Analizirali smo efekte ovih protokola i dali svoj doprinos u rešavanju nekih problema za koje dosadašnji, nama poznati protokoli, nisu dali adekvatna rešenja. To se pre svega odnosi na smanjivanje aktivnosti kod prijavljivanja novih SČ i rešavanju problema kolizije kod istovremene prijave dva i više SČ.

Ključne riječi - 1; bežične senzorske mreže 2; *rendezvous* protokoli 3; životni vek 4; energetska efikasnost

I. UVOD

Bežične senzorske mreže se sastoje od velikog broja senzorskih čvorova (SČ) koji mogu biti raspoređeni u praktično bilo kom prirodnom okruženju. SČ-ovi su u stanju da potpuno samostalno formiraju jedinstvenu mrežu, preko koje mogu da prikupljaju, obrađuju i prosleđuju podatke dobijene od senzora. Jedan od najbitnijih faktora BSM po kome se one razlikuju od ostalih bežičnih struktura je potrošnja el. energije. Kako je ovde izvor el. energije obično limitiran (<0,5 Ah, 1,2 V) jasno je da je životni vek SČ, a samim tim i cele BSM, direktno zavistan od količine energije kojom on raspolaže. U višeskokovitim BSM ovaj faktor još više dolazi do izražaja. Ovde svaki SČ ima dvostruku ulogu: sa jedne strane potrebno je da prikuplja podatke, a sa druge strane da istovremeno vrši preusmeravanje podataka koje dobija od drugih SČ, tj. Da vrši njihovo preusmeravanje. Ove povećane aktivnosti SČ imaju za posledicu znatno veću potrošnju el.energije što dovodi do prekida rada SČ-ova koji su najviše izloženi neprekidnom radu. Problemi u funkcionisanju nekog od SČ u mreži (do koga može doći zbog gubitka el.energije) može dovesti do velike promene u mrežnoj topologiji, a to zahteva ponovnu reorganizaciju mreže kao i pronalaženje novih puteva rutiranja podataka. Sve to zahteva dodatnu potrošnju el.energije koja opet može proizvesti prekid rada dodatnih SČ, tj. u krajnjoj istanci, do potpunog prekida rada cele mreže. Zato se pred projektantima kao osnovni cilj postavlja projektovanje energetske efikasne aplikacije kako bi se životni vek BSM što

više produžio. Ukoliko su svi delovi jednog SČ stalno aktivni tada bi njihov životni vek bio veoma kratak, svega nekoliko dana. Promena baterije predstavlja jako komplikovan zadatak, ne samo zbog nepristupačnosti lokacije gde se SČ nalaze, već i zbog njihovog velikog broja u BSM (od nekoliko stotina pa do nekoliko hiljada SČ). Sa druge strane, povećanje njenih kapaciteta nije moguće zbog male veličine SČ (nekoliko mm³), a punjenje ovih baterija je dosta otežano, pre svega zbog limitiranih resursa SČ, kako računarskih tako i fizičkih. Tako nam predstoji jedino rešenje a to je da obezbedimo da BSM radi na što manjem energetske nivou primenom energetske efikasne protokola.

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 [1]. Svaka od ovih aktivnosti troši različitu količinu energije, međutim količina energije koja se potroši u primopredaji podataka znatno prevazilazi potrošnju pri svim ostalim aktivnostima u SČ. Kao primer može da nam posluži podatak da je energija koja se potroši za prenos samo 1 bita informacije preko komunikacionog kanala, jednaka energiji koja se potroši za izvršavanje 1000 CPU instrukcija [2]. Shodno tome, ako uspemo da smanjimo potrošnju komunikacionog kanala za samo 1-2 %, uspećemo da uštedimo energiju koja je po veličini znatno veća od potrošnje pri svim ostalim aktivnostima SČ. Uzimajući ovu činjenicu u obzir, očigledno je da se najveće uštede u potrošnji el.energije mogu postići primenom energetske efikasne komunikacionih protokola. Ovi protokoli koriste različite tehnike optimizacije utroška energije u svim slojevima tradicionalnog OSI referentnog modela, od fizičkog do aplikacionog [3, 4, 5]. 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 u mreži. Tu se pre svega misli na preciznu vremensku sinhronizaciju svih SČ u mreži, koja treba da omogući nesmetanu komunikaciju između SČ. Istraživanja pokazuju da čak i pri DC faktoru od samo 1-2%, postiže se ušteda u potrošnji el.energije za jedan red veličine veći od potrošnje svih ostalih komponenata u SČ [3]. 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, predlažemo modifikaciju tradicionalne šeme pod nazivom modifikovani *rendezvous* protokol, MRP. Da bi efikasno rešili vremensku sinhronizaciju SČ-ova u slučaju velikih vremenskih odstupanja (veliki *clock drift*) koristili smo tehniku proširenja *duty cycling* tehnike [5]. MRP obezbeđuje sinhronizovano buđenje SČ-ova u potpuno decentralizovanom načinu rada i rešava problem istovremenog prijavljivanja dva ili više SČ, tj. pojavu kolizije. Korišćenjem odgovarajuće analitičke metode, izveli smo eksplicitne izraze na osnovu kojih smo odredili kako izbor odgovarajućih komponenti, pre svega kvarcnog oscilatora, utiče na životni vek SČ, kao i na određivanje maksimalnog broja SČ-ova koji mogu regularno da učestvuju u predloženom *rendezvous* protokolu.

Nakon kratkog uvoda u problem koji ćemo analizirati u nastavku rada, u poglavlju 2. data je osnovna podela svih *rendezvous* protokola gledano sa različitih aspekata: na osnovu broja SČ-ova, u zavisnosti od OSI referentnog nivoa na kome su *rendezvous* tehnike primenjene i u odnosu na način uspostavljanja komunikacije između SČ-ova. U narednom poglavlju analiziran je jedan od standardnih *rendezvous* protokola i na njegovom primeru date su dobre i loše osobine ovih protokola. Poglavlje 4. daje pregled predloženog modifikovanog *rendezvous* protokola, MRP, gde su jasno istaknuti načini rešavanja uočenih problema kod ranijih *rendezvous* šema. Performanse predloženog protokola sa gledišta maksimalnog broja SČ-ova sa odgovarajućim DC faktorom, data su u poglavlju 5. Na kraju poglavlja 6 zaključuje ovaj rad.

II. PREGLED RENDEZVOUS ŠEMA

Sa gledišta broja SČ koji učestvuju u *rendezvous* šemi ove šeme možemo podeliti u dve velike grupe. Prva se odnosi na *rendezvous* protokole koji kontrolišu potrošnju el.energije na nivou cele BSM, formiranjem odgovarajuće putanje za prenos podataka. Naime, izborom odgovarajućeg optimalnog podskupa SČ-ova koji će ostati aktivni i garantovati prenos podataka između SČ-ova u BSM, moguće je smanjiti broj aktivnih SČ i time znatno uštedeti el.energiju na nivou cele BSM. Drugi pristup se koncentriše na upravljanje energijom u svakom SČ. Tokom svog rada, bez obzira na trenutne potrebne putanje za prenos podataka, SČ menja režim rada između aktivnog (*active*) i neaktivnog (*sleep* ili *idle*) [4].

Sa druge strane, u zavisnosti od OSI referentnog nivoa na kome su *rendezvous* tehnike primenjene možemo ih podeliti na dve kategorije: prva koja koristi standardne MAC protokole (nalaze se između prvog i drugog nivoa) i druga koja koristi protokole koji rade na principu aktivne/neaktivne šeme, *sleep/wake* (rade iznad drugog nivoa) [4]. Imajući u vidu da *sleep/wake* protokoli rade na višem nivou, čak i na aplikacionom sloju, oni imaju veću fleksibilnost i mogu se kombinovati sa bilo kojim standardnim MAC protokolom. Po ceni većeg kašnjenja, ali kontrolisanog kašnjenja, oni postižu veći procenat uštede el. energije. Osnovni princip rada sastoji

se u tome da primo-predajnik bude aktivan samo onda kada je potrebno da primi ili pošalje podatak susednom SČ [6, 7].

U odnosu na način kako SČ uspostavljaju komunikaciju sve *rendezvous* protokole možemo podeliti u tri kategorije [4, 8]:

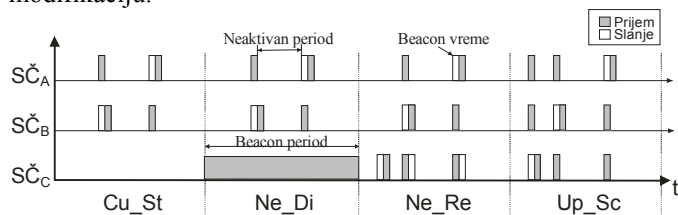
1. na zahtev - potpuno asinhrona šema kod koje izvorni SČ inicira (budi) drugi destinacioni SČ kome želi da pošalje podatak. Komuniciranje između SČ-ova je potpuno asihrono jer se ono uspostavlja samo kada izvorni SČ želi da komunicira sa nekim drugim SČ. Osnovni problem ove šeme je da svi SČ-ovi moraju da koriste još jedan dodatni metod za iniciranje prenosa između SČ-ova. Obično se ovde koriste dva nezavisna kanala: jedan za slanje podataka a drugi za iniciranje tog prenosa. To znači da SČ-ovi moraju da imaju primopredajnike sa najmanje dva kanala što dodatno povećava njihovu cenu i gabarite.
2. zakazan *rendezvous* – potpuno sinhronizovana šema koja zahteva da svi susedni SČ-ovi imaju dogovor o sledećoj sesiji, tj. da se aktiviraju u potpuno isto vreme, kako bi se obavila komunikacija. Glavna prednost ove šeme sastoji se u tome da je moguće slati *broadcast* poruke svim SČ-ima jer su oni svi aktivni u istim vremenskim periodima. Sa druge strane, efikasnost ove šeme zavisi od veoma precizne vremenske sinhronizacije svih SČ-ova u BSM što uopšte nije lako uraditi a naročito u BSM [9].
3. pseudo-asinhrona šema – u osnovi ovo predstavlja šemu u kojoj se SČ-ovi periodično bude. Ona omogućava svakom SČ da se aktivira nezavisno u odnosu na druge SČ-ove koji se nalaze u njegovom susedstvu. Naime, izvorni SČ se prvi aktivira i čeka da se određeni SČ aktivira kako bi mu poslao podatke.

Ove tri kategorije obuhvataju gotovo sve *rendezvous* šeme koje imaju za cilj da omogućе veću energetsку efikasnost u BSM. Međutim to ne sme da bude jedini uslov koji ovi protokoli treba da ispune. Potrebno je da imaju i prihvatljive karakteristike u ostalim aspektima mrežne komunikacije, takođe. Jedan od najvažnijih uslova, koji dolazi do izražaja u mrežnim komunikacijama, odnosi se na kašnjenje primopredaje podataka, tj. latencija podataka. Uključivanje/isključivanje SČ iz aktivnog u neaktivni režim rada, neminovno povećava kašnjenje paketa. Zato je važno da *rendezvous* šema ne dozvoli da to kašnjenje prelazi opseg tolerancije aplikacije koja radi u BSM. Odnosno, potrebno je da u nekom unapred definisanom vremenskom intervalu izvršiti prenos paketa kako bi podaci koji se dobiju bili relevantni i pravilno upotrebljeni.

III. PREDNOSTI I MANE STANDARDNOG RENDEZVOUS PROTOKOLA

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 tehnika (*Power Management Technique*). Mi smo se opredelili za primenu odgovarajuće *rendezvous* šeme. U tom smislu, analizirali smo standardnu *rendezvous* šemu (DLDC-MAC protokol) [10], i ukazali na prednosti i mane ovog protokola. Nakon toga, u cilju rešavanja uočenih nedostataka predložili smo njegovu modifikaciju.



Slika 1. Blok šema DLDC-MAC protokola

A. Pregled DLDC-MAC protokola

Osnovna ideja ovog rešenja, koje je dato u ref. [10], je sledeće: svaki SČ periodično šalje kratku poruku, pod nazivom *beacon*, kako bi signalizirao svoje prisustvo u BSM. Takođe, u tačno određenim vremenskim intervalima 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 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. Ono je posledica nesavršenosti i neujednačenosti kvarcnih oscilatora koji se nalaze u svakom SČ (*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 vremena prijema *beacon*a od svojih suseda. Drugim rečima, SČ sada najviše vremena provodi u *sleep/idle* stanju štedeći svoju el.energiju. Na sl. 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. On se periodično ponavlja i u okviru njega se nalaze sva neophodna vremena za slanje i prijem paketa ka i od SČ, kao i vreme koje SČ provede u neaktivnom stanju (*sleep*). Sa slike 1 možemo da vidimo da postoje četiri različite faze kroz koje prolazi svaki od SČ. To su:

1. Tekuća faza (Cu_St) – u ovoj fazi svaki SČ je svestan svojih susednih SČ sa kojima komunicira, tj. zna *beacon* vremena svih susednih SČ. U ovoj fazi svaki SČ provede najveći deo svog životnog veka (preko 99 %).
2. Faza pronalazanja (Ne_Di) – predstavlja početnu fazu kroz koju prolazi svaki novi SČ koji želi da se priključi BSM. Naime, pri prvom uključenju, SČ ulazi u ovu 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.

3. 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 i o tome obaveštava susedne SČ-ove kako bi oni mogli da ga private, tj. da uključe svoje prijemnike u novo formiranom vremenskom slotu.
4. 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 SČ u okviru *beacon* perioda,

Navešćemo neke od glavnih prednosti analiziranog DLDC-MAC protokola [10]:

- 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.
- Većina problema vezanih za bežične komunikacije kao što su: *clock drift*, asimetrične veze, skriveni terminal i otežane komunikacije su uspešno prevaziđeni.
- Primenom ovog protokola znatno se štedi utrošak el. energije pa se očekuje višegodišnje produženje životnog veka SČ-ova, a samim tim i život cele BSM.

Međutim analizirajući ovu šemu mi smo uvideli i neke nedostatke koje ovaj protokol ne može da reši, a to su:

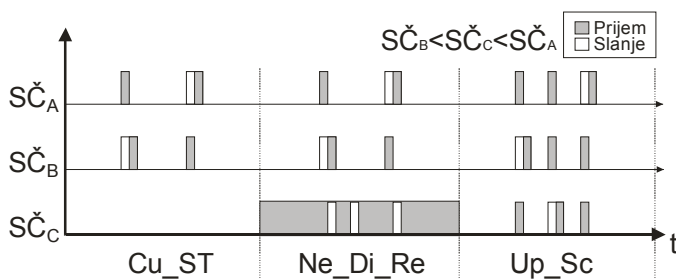
- Određivanje dužine adekvatnog *beacon* perioda kada se dva ili više SČ istovremeno prijavljuju u BSM.
- Određivanje adekvatnog *beacon* vremena (vremenski slot) za svaki novoprijavljeni SČ.
- Kako *beacon* vremena nisu unapred definisana (vremenski slotovi) može da dođe do njihovog preklapanja usled 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.
- Problem rešavanja kolizije koja može da se dogodi ako se dva ili više SČ istovremeno prijave u BSM.

IV. MODIFIKOVANI RENDEZVOUS PROTOKOL

U ovom radu predlažemo neke izmene prethodno pomenutog DLDC-MAC protokola koje eliminišu uočene njegove nedostatke. Taj novi protocol nazvali smo modifikovani *rendezvous* protocol, MRP. U njemu smo usvojili sve dobre osobine DLDC-MAC protokola koje uspešno rešavaju problem loših linkova, skrivenog terminala, asimetričnih veza i klok drift-a. Da bi rešili osnovni problem DLDC-MAC protokola, prijavljivanje novih SČ-ova i određivanje adekvatnih vremenskih slotova za slanje *beacon*-a, mi smo u našem protokolu pošli od činjenice da svi SČ-ovi

budu jedinstveno identifikovani putem jedinstvenog ID broja koji svaki SČ u procesu svog formiranja dobije ($ID=1,2,3,\dots,n$, gde je n maksimalan broj SČ u mreži). Uvođenje jedinstvenog ID broja za svaki SČ omogućuje nam da unapred možemo da definišemo *beacon* vreme (vremenski slot) svakog SČ u okviru *beacon* perioda. Takođe, broj SČ-ova određuje nam i minimalno trajanje *beacon* perioda koje možemo da unapred odredimo u zavisnosti koji DC faktor želimo da imamo. Naš predlog je ujedinio fazu Ne_Di i fazu Ne_Re u jedinstvenu fazu Ne_Di_Re , čime smo smanjili za jedan *beacon* period uspostavljanje normalnog režima rada. Faze Cu_St i Up_Sc ostale su potpuno iste kao i kod DLDC-MAC protokola.

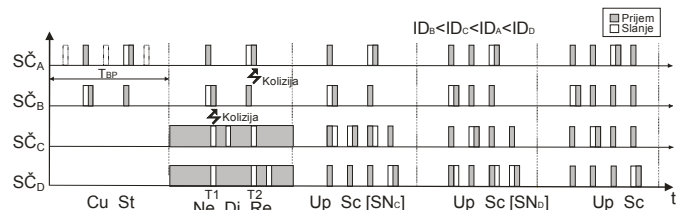
Na slici br.2 prikazan je postupak prijavljivanja novog SČ u okviru već postojeće BSM. Kao što se sa slike vidi, svaki novi SČ koji se prijavljuje (u našem slučaju to je SČ_C), na samom početku ulazi u *receive* mode (faza Ne_Di_Re) kako bi mogao da prihvati *beacon*-e od susednih SČ-ova. *Receive* mode 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 izbegavamo moguću koliziju a ujedno obezbeđujemo da svaki novi SČ prihvati *beacon* od svih susednih SČ-ova. U trenutku kada prihvati neki *beacon* od susednih SČ-ova, novi SČ ulazi u *send* mode i odmah šalje informaciju o svom prisustvu (svoj ID broj) tom istom SČ. Na osnovu te informacije stari SČ-ovi sada ubacuju novi interval osušivanja na mesto u *beacon* periodu (odgovarajući vremenski slot) koji je definisano primljenim ID brojem od novoprijavljenog SČ i istovremeno šalju potvrdu novoprijavljenom SČ (SČ_C) da je uspostavljena i povratna veza između njih. Ovo je potrebno kako bi svaki SČ eliminisao nepotrebno oslušivanje i trošenje energije za SČ-ove sa kojima ne postoji veza (asimetrične veze, SČ 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* modu (svi ostali SČ-ovi su tada u *receive* modu) a n puta u *receive* modu gde n predstavlja ukupan broj SČ-ova u susedstvu. *Receive* mode koji sledi odmah nakon *send* moda je neophodan kako bi se prihvatili ID brojevi novih SČ-ova iz okoline. Problem kolizije je na ovaj način 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 u okviru unapred definisanog *beacon* perioda T_{BP} .



Slika 2. Blok šema modifikovanog rendezvous protokola

Rešavanje ovog problema prikazano je na slici br.3 gde je prikazano istovremeno prijavljivanje dva nova SČ, SČ_C i SČ_D. Ako u periodu Ne_Di_Re dođe do kolizije kod slanja inicijalnih ID, prijava prisustva dva ili više SČ, novoprijavljeni

SČ 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 slota 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). 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.



Slika 3. Blok šema modifikovanog rendezvous protokola sa kolizijom

V. PERFORMANSE PREDLOŽENOG PROTOKOLA

Analiziraćemo sada kako dužina *beacon* perioda, T_{BP} , utiče na energetske efikasne rad svakog SČ kao i koliko SČ-ova može da se prihvati u prethodno definisanom periodu T_{BP} (koliko vremenskih slotova može da bude u periodu T_{BP}), a da DC faktor ostane u granicama od 0,1-1 % (0.001-0.01). Ako se pogleda šema predloženog protokola (slika br.3) vidi se da većinu vremena SČ provede u fazi Cu_St . U okviru ove faze svaki SČ mora da se nađe u predajno/prijemnom modu samo jednom i $n-1$ puta u prijemnom modu. Kad god je on u prijemnom modu neki drugi SČ iz njegovog susedstva mora da je u predajno/prijemnom modu. Za određivanje max. broja susednih SČ-ova odlučujuću ulogu ima dužina perioda T_{BP} , kao i DC faktor koji želimo da postignemo. Vrednost DC faktora ima odlučujuću ulogu u određivanju životnog veka SČ. Tako, što je DC faktor manji to je ušteda energije veća, pa je samim tim i životni vek BSM veći. Odredimo sada maksimalan broj SČ-ova koji mogu regularno da funkcionišu za zadati *beacon* period, T_{BP} , a da ispune unapred definisani DC faktor.

Pretpostavićemo da se naša BSM sastoji od maksimum n SČ. Za vreme faze Cu_St (ili Up_Sc) postoje dve različite aktivnosti i to: 1) prijemna aktivnost koja traje T_{RBC} , i 2) predajno/prijemna aktivnost koja traje T_{TRBC} . Za vreme prijemne aktivnosti SČ aktivira prijemnik, period T_{RB} . U okviru ovog perioda on prima *beacon*-e od relevantnog SČ, i nakon toga ulazi u neaktivni režim, koji traje T_{OFF} . Na osnovu ovoga sledi da je $T_{RBC}=T_{RB}+T_{OFF}$. Kod predajno/prijemne aktivnosti SČ aktivira predajnik da bi poslao svoj *beacon* drugim SČ-ima, period T_{TB} , zatim se aktivira prijemnik, period T_{RB} , i nakon toga prelazi u neaktivni režim rada T_{OFF} , što znači da je $T_{TRBC}=T_{TB}+T_{RB}+T_{OFF}$. U našem slučaju smatraćemo da su periodi T_{TB} i T_{RB} približno identični. Na osnovu ovoga definišaćemo DC faktor za SČ_k, $k=1,\dots,n$, kao:

$$DC_k = \frac{T_{ONk}}{T_{BP}} = \frac{\sum_{i=1}^n a_i T_{RBi} + T_{TBk}}{T_{BP}}, \quad (1)$$

gde promenljiva $a_i \in \{0,1\}$, $i=1,\dots,n$, ukazuje na činjenicu da li je odgovarajući SČ aktivan ($a_i=1$) ili neaktivan ($a_i=0$), tj. da li postoji link između tog SČ i susjednih SČ-ova.

Iz jednačine (1) sledi,

$$T_{ONk} = \sum_{i=1}^n a_i T_{RBi} + T_{TBk} \quad (2)$$

Definišimo sada da je za vreme faze Cu_St (ili Up_Sc faze) broj SČ-ova koji međusobno mogu da komuniciraju, tj. broj aktivnih susjednih SČ jednak p , gde je $p < n$, tako da sada važi sledeće:

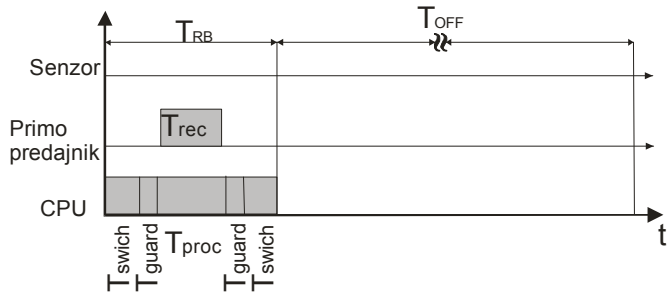
$$\sum_{i=1}^n a_i T_{RBi} = p * T_{RB} \quad (3)$$

Ako za jedan aktivni SČ važi da je $T_{RBi}=T_{Tbi}=T_{RB}$, za $i=1,\dots, n$, tada jednačina (2) može da se transformiše u:

$$T_{ONk} = (p+1) * T_{RB} \quad (4)$$

Zamenom jednačine (4) u jednačinu (1) dobijamo da je:

$$DC_k = \frac{(p+1) * T_{RB}}{T_{BP}} \quad (5)$$



Slika 4. Aktivnosti SČ za vreme prijemne faze T_{RBC}

U okviru svog rada, svaki SČ obavlja različite aktivnosti koje su podeljene po sastavnim delovima jednog SČ. Na slici 4 prikazane su te aktivnosti (senzor, primopredajnik i procesor), u odnosu na vremena trajanja, i u okviru prijemnog ciklusa, T_{RBC} . Posmatrajući sliku 4, vremensko trajanje perioda T_{RB} može se prikazati sa:

$$T_{RB} = T_{switch} + T_{guard} + T_{proc} + T_{guard} + T_{switch} \quad (6)$$

gde T_{switch} predstavlja vreme koje protekne kada se CPU prebacuje iz aktivnog u neaktivni režim rada i obrnuto; T_{proc} predstavlja vreme koje je potrebno procesoru da prihvati podatke koje su mu poslali drugi susjedni SČ-ovi; i T_{guard} predstavlja guard vreme. Ovo vreme je posledica različitih vremena koje imaju SČ-ovi u BSM. Naime, svi SČ beleže lokalno vreme na osnovu svojih lokalnih kvarcnih oscilatora.

Međutim svi kvarcni oscilatori ne generišu istu frekvenciju pa dolazi do razlika (*drift*), pa samim tim i do različitih vremena u SČ-ima. Da bi se obezbedila istovremena komunikacija između SČ-ova, potrebno je da se perioda aktivnosti proširi sa vremenom T_{guard} i na taj način obezbedi sigurno poklapanje, tj. *rendezvous*, između SČ-ova. Uzima se da T_{guard} linearno zavisi od dužine perioda T_{BP} i da je on jednak maksimalnom *drift*-u. Ako se sa s_x ($\Delta f/f$) označi maksimalna kvarcna nestabilnost, tada je:

$$T_{guard} = 2 * T_{BP} * s_x \quad (7)$$

Ako sada zamenimo jednačinu (5) u jednačini (4), uzimajući u obzir jednačine (6) i (7) dobićemo da je:

$$T_{BP} = \frac{(p+1) * (2 * T_{switch} + T_{proc})}{DC_k - 2 * s_x * (p+1)} \quad (8)$$

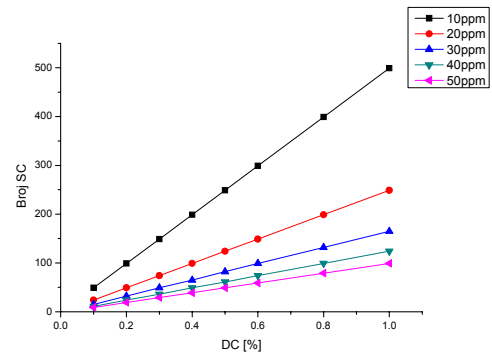
Kako iz jednačine (8) T_{BP} mora uvek da bude pozitivna vrednost, tj. važi da je $T_{BP} > 0$, sledeći uslov koji sledi iz jednačine (8) može da se izvede:

$$DC_k - 2 * s_x * (p+1) > 0 \quad (9)$$

tj. :

$$p < \frac{DC_k}{2 * s_x} - 1 \quad (10)$$

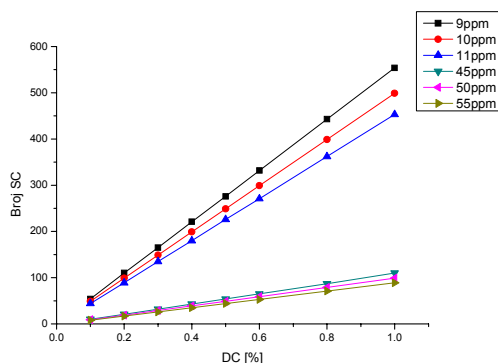
Na osnovu jednačine (10), sada možemo da na osnovu unapred zadatog DC faktora, definišemo maksimalan broj SČ-ova u okviru jedne BSM. Za vreme ove analize, nestabilnost kvarcnog oscilatora, s_x , kao parametar, biće razmatran. Na slici br. 5 prikazan je grafik koji nam prikazuje kako se menja maksimalan broj SČ-ova u funkciji DC faktora, za različite vrednosti stabilnosti kvarcnog oscilatora (od 10 do 50 ppm).



Slika 5. Maksimalan broj SČ-ova u funkciji DC faktora

U realnim uslovima se često postavlja pitanje: Koju vrstu oscilatora, u odnosu na njegovu stabilnost definisanu u ppm, treba ugraditi u SČ? Obično, svaki proizvođač nominalno deklarise stabilnost kvarcnih oscilatora koje on nudi i ona obično iznosi od 10 do 50 ppm. Međutim, nominalna stabilnost nikada nije u tim granicama, pre svega zbog ambijentalnih uslova kojima je taj kvarc izložen kao što su: temperatura, pritisak, vlažnost, i td. U našem razmatranju uzeli smo da dodatne varijacije iznose do $\pm 10\%$ od nominalne vrednosti

stabilnosti kvarcnog oscilatora. Kao direktna posledica ovih razlika dobija se da različiti broj aktivnih SČ-ova, u okviru jednog *beacon* perioda, može regularno da funkcioniše. Na slici broj 6 prikazan je grafik zavisnosti maksimalnog broja aktivnih SČ-ova u odnosu na različiti DC faktor za dva različita kvarcna oscilatora koji imaju stabilnost od 10 ppm i 50 ppm.



Slika 6. Broj SČ-ova u funkciji DC faktora za $s_x=10$ i 50 ppm $\pm 10\%$

Gledajući grafik sa slike broj 6 možemo da zaključimo sledeće:

1. Za isti DC faktor maksimalni broj SČ-ova uvek opada ako koristimo kvarcni oscilator sa manjom stabilnošću, tj. ako se s_x povećava.
2. Nezavisno od upotrebljenog kvarcnog oscilatora uvek kada raste DC factor, raste i broj SČ-ova koji mogu regularno da rade u istom *beacon* periodu.
3. U svim slučajevima samo mala frekventna devijacija (od $\pm 10\%$) prouzrokuje velike varijacije u maksimalnom broju aktivnih SČ. Što je nestabilnost kvarcnog oscilatora veća to je i ova razlika veća. Tako, na primer za DC=1 i $s_x=10$ ppm ($\pm 10\%$) ova razlika može da iznosi $554-453=101$ SČ-ova dok za isti DC=1 ali za $s_x=50$ ova razlika je znatno manja $110-89=21$ SČ.

VI. ZAKLJUČAK

Uspešnost rada BSM sve više se meri dužinom njenog trajanja. Racionalan utrošak energije postao je osnovni i neophodni zadatak koji treba da ispune gotovo sve aplikacije koje rade u BSM. Međutim zbog mnogih ograničavajućih faktora sredine u kojoj te aplikacije rade, kao i jako ograničavajućih resursa SČ, nije jednostavno pronaći efikasnu metodu koja će sve ovo zadovoljiti. U radu je prikazana jedna od tehnika, *rendezvous*, koja po opštem mišljenju nezaobilazna u svim aplikacijama u BSM. Ovaj rad analizirao je kako različiti DC faktori utiču na potrošnju električne energije. Da bi smo pokazali uspešnost ove metode uveli smo koeficijent uspešnosti, K_u , koji je pokazao superiornost ove tehnike. U našem primeru dobili smo da on ide od 1,99 do 1620,93 u zavisnosti od koeficijenta DC faktora koji smo primenili. Analiza je pokazala da što je DC faktor manji to je i potrošnja SČ manja, pa je samim tim i njihov životni vek duži. Rad je potvrdio koliko je tehnika *duty cycling* superiorna u odnosu na

aplikacije koje ovu tehniku ne primenjuju. Ta nadmoć je toliko velika da se više uopšte ne postavlja pitanje njene primene jer je ona postala sastavni deo gotovo svake aplikacije u BSM koja želi da efikasno koristi strogo limitirane količine električne energije.

LITERATURA

- [1] P. K. Dutta, D. E. Culler, "System Software Techniques for Low Power Operation in Wireless Sensor Networks", In Proceedings of the ICCAD'05, 2005, pp.925-932
- [2] P.Dutta, D.Culler, S.Shenker, "Procrastination Might Lead to a Longer and More Useful Life", <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.93.5780>, acc. 14.12.2012.
- [3] Raghunathan V., Ganerival S., Srivastava M., "Emerging techniques for long lived wireless sensor networks", IEEE Communication Magazine, Vol. 44, No. 4, 2006, pp. 108
- [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.Kosanovic, M.Stojcev, "Sensor Node Lifetime Prolonging", 20 TELFOR, Belgrade 20-22.11.2012
- [6] J. Elson, L. Girod, and D. Estrin, "Fine-grained network time synchronization using reference broadcasts", In Proc. Symposium on Operating Systems Design and Implementation (OSDI), volume 36, pages 147-163, Boston, MA, USA, Dec. 2002.
- [7] 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, 121-128
- [8] G.Anastasi, M.Conti, M.D.Francesco, A.Passarelle, "Energy Conservation in Wireless Sensor Networks: A Survey", <http://info.iet.unipi.it/~anastasi/papers/adhoc08.pdf>, acc.10.12.2012
- [9] Anton Ageev, "Time Synchronization and Energy Efficiency in Wireless Sensor Networks", DISI - University of Trento, PhD thesis, March 2010
- [10] 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

ABSTRACT

One of the most important tasks, that designers need to address when designing applications to operate in Wireless Sensor Networks (WSN), is the rational usage of energy. Sensor Nodes (SN), as components of a WSN, are powered by batteries, which mean that they have a limited amount of available energy. The fulfillment of this condition is not an easy task, due to the limited resources of the sensor nodes as well as the natural conditions in which they work. In this paper we present a protocol group, *rendezvous* protocols, which are used in the WSN to overcome the problem of unnecessary power energy consumption. We analyzed the effects of these protocols in order to solving some of the problems that the previous ones, chosen protocols, are not an adequate solution. This is primarily related to the reduction of activity in registering a new SNs, and solving the problem of collision which is appeared when simultaneously register two or more SNs.

USING RENDEZVOUS PROTOCOLS FOR LONGLIFE WIRELESS SENSOR NETWORKS

Mirko Kosanović, Mile Stojčev