

SMAC protokol u bežičnim senzorskim mrežama

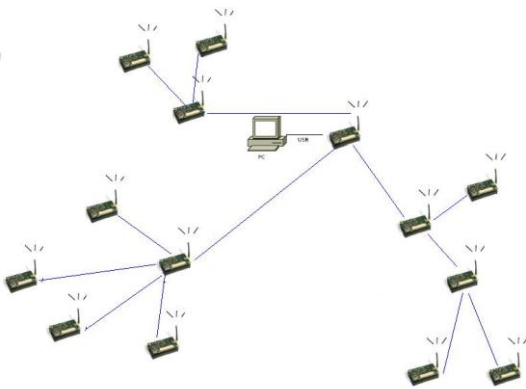
Miloš Rovčanin

Sadržaj - U ovom opisnom radu su predstavljene osnovni koncepti bežičnih senzorskih mreža, konstrukcijske osobine njenih čvorova (motova), kao i protokoli za kontrolu pristupa medijumu za prenos podataka (Medium Access Protocols). Posebna pažnja je posvećena Sensor MAC (SMAC) protokolu. U radu su priloženi rezultati testiranja ovog protokola na Berkeley univerzitetu, SAD.

Ključne reči - Bežične senzorske mreže, mot, TinyOS, SMAC

I. UVOD U BEŽIČNE SENZORSKE MREŽE

Bežične senzorske mreže (WSN – *Wireless Sensor Network*) predstavljaju kategoriju *ad-hoc* bežičnih mreža, koja je tek počela da se razvija. WSN su namenske mreže, sastavljene od velikog broja resursno ograničenih uređaja, motova. Ideja je da se postavljanjem velikog broja ovih uređaja, raspoređivanjem po određenoj šemi uz minimalne troškove, pokrije veliko geografsko područje. Snaga bežičnih senzorskih mreža upravo leži u broju motova koji je čine. Što je njihov broj veći, povezanost i robusnost mreže rastu. WSN su nasledile sve osobine *ad-hoc* mreža i optimizovale njihovu upotrebu kroz *mash* topologiju. Ipak, za razliku od tipične *mash* topologije, koja zahteva potpunu povezanost svih čvorova, u WSN mrežama svaki mot je u kontaktu sa najmanje dva mota, ali ne i sa svima ostalima. Svaki čvor mreže predstavlja jednu pristupnu tačku i pruža usluge rutiranja podataka drugim čvorovima i na taj način formira *multi-hop* mrežu (slika 1).



Slika 1. Topologija *multi-hop* mreže

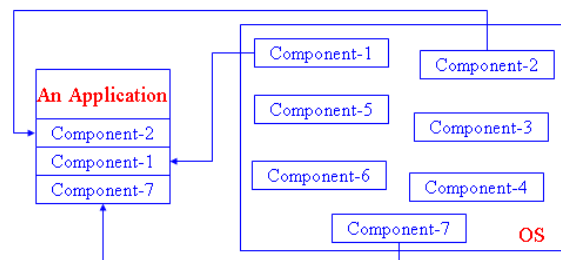
Bitna odlika ovih mreža jeste sposobnost samokonfigurisanja. Čvorovi su u stanju da identifikuju svoju okolinu bez ičije intervencije.

Oni se automatski prilagođavaju promeni topologije što omogućava jednostavno proširenje mreže kada god je to potrebno. Čvorovi su u stanju i da registruju otkaz nekog od suseda, i da pronađu alternativni put za slanje tj. rutiranje paketa do bazne stanice. Cena postavljanja ovakve mreže je daleko ispod cene postavljanja žičanih mreža.

U cilju ostvarivanja željene funkcionalnosti WSN-ova, potrebno je dizajnirati softversku arhitekturu koja će omogućiti funkcionisanje kompletnog sistema uzimajući u obzir ograničenja hardverske platforme. Softver predviđen za WSN mreže mora efikasno da koristi ograničene resurse čvorova, njihovu energiju i procesorsko vreme. Još jedan bitan zadatak ove arhitekture jeste da apstrahuje hardverske komponente i programeru omogući bržu i efikasniju interakciju sa hardverom.

TinyOS je real-time operativni sistem koji je na najbolji mogući način zadovoljio postavljene zahteve i postao je najčešće korišćena softverska platforma od strane korisnika i istraživača.

Specijalno za potrebe senzorskih mreža, dizajniran je programski jezik nesC (*Network Embedded System C*), koji podržava programski model koji izvršava aplikacije putem događaja (*event oriented*), a rešava i problem konkurencije. Aplikacije u nesC-u se konstruišu povezivanjem komponenti (slika 2), programskih modula koji predstavljaju apstrakciju hardverskih komponenti.

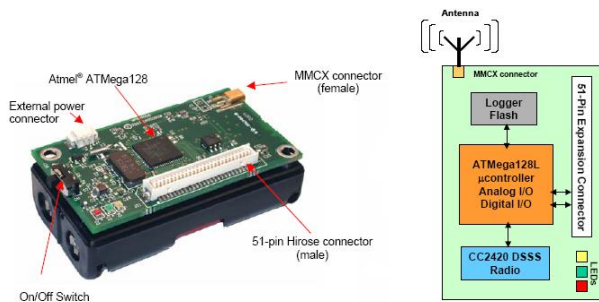


Slika 2. Koncept TinyOS

II. MOT

Mot je čvor bežične senzorske mreže. Jedna od najvećih kompanija koja se bavi proizvodnjom i plasmanom motova na tržište je *Crosboww Technology*, sa sedištem u San Hozeu, USA. Na slici 3 je prikazan izgled mica2 mota i njegova blok šema.

Sastavni delovi svakog mota su: procesor, RF primopredajnik (radio), senzorska ploča, fleš memorija (*EEPROM*) i napajanje (baterije).



Slika 3. Izgled i blok šema mica2 mota

Radio primopredajnik i procesor se nalaze na istoj ploči (radio-procesorska ploča). Mikroprocesor je jezgro mota. On nadgleda senzore od kojih dobija podatke o atributima sredine i radio primopredajnikom ih šalje prema drugim motovima. Na senzorskoj ploči se mogu nalaziti: optički i foto senzor, piezoelektrični senzor, termistor i senzor pokreta. Što će reći, ovim senzorima se mogu dobiti informacije o temperaturi, vlažnosti vazduha, osvetljaju, intenzitetu zvuka. Može se detektovati pokret, ali i meriti ubrzanje. Svaki senzor ima komponentu koja predstavlja njegovu softversku apstrakciju i koja se može uključiti u bilo koju konfiguraciju i koristiti po potrebi programera. Hardverske komponente preko kojih se vrši programiranje motova su programatorske ploče.

Radio primopredajnik je najveći potrošač energije na motu. Kako bi se rasipanje energije svelo na minimum, definisane su razne verzije *Medium Access Control* (MAC) protokola, koje imaju za cilj da, u skladu sa dinamikom razmene poruka u mreži, uključuju i isključuju radio interfejs. Tako čuvaju energiju i produžavaju vek trajanja baterija kojima se mot napaja.

U glavama koje slede biće dat pregled MAC protokola u bežičnim senzorskim mrežama. Posebno će biti opisan SMAC protokol, koji poseduje mehanizme za samokonfiguraciju i prilagođavanje promenama topologije, i kojima se sprečava kolizija paketa i reguliše potrošnja energije u čvorovima mreže.

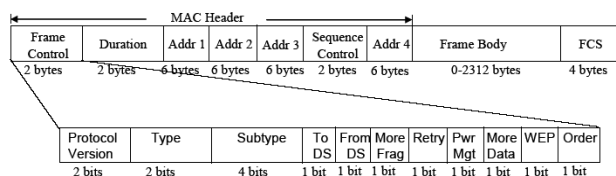
III. MAC PROTOKOLI

Obzirom da se motovi napajaju baterijama, jasno je da će, nakon određenog vremena, biti potrebna njihova zamena. Kako bi se produžio životni vek baterija, osmišljeno je više verzija MAC protokola namenjenih isključivo senzorskim mrežama. Njihov zadatak jeste kontrola radio primopredajnika, najvećeg potrošača energije na motu. Ideja se sastoji u tome da se radio drži uključenim jedino onda kada mot prima pakete ili kada ih šalje tj. da se *idle listening* period skрати što je više moguće. *Idle listening* je stanje kada je radio primopredajnik uključen, osluškuje kanal, a niti prima podatke niti ih šalje. Na žalost, ovo nije jedini izvor rasipanja energije. Kolizija paketa, zaglavlje koje protokol dodaje na paket (*protocol overhead*), prijem i odbacivanje poruka koje su stigle na adresu na koju nisu upućene su dodatni izvori utroška energije.

Kolizija paketa nastaje kada mot pokuša da pošalje poruku preko medijuma koji je već zauzet. Većina

jednokanalnih radio primopredajnika nema tu sposobnost da može da prima i da šalje poruke u isto vreme tako da je jedini način da pošiljalac zaključi da je došlo do kolizije, odsustvo poruke kojom se potvrđuje prijem. Propagacijsko kašnjenje između udaljenih čvorova, vreme koje je potrebno radio primopredajniku da počne sa emitovanjem kada “oseti” da je medijum slobodan (250 do 500 μ s) kao i situacija kada imamo čvorove koji se nalaze van dometa pošiljaoca, ali u dometu primaoca (*hidden neighbours*) su najčešći uzroci kolizije, dok ponovno slanje paketa iziskuje dodatnu potrošnju energije. Kolizija se može izbeći korišćenjem *backoff* (IEEE 802.11) algoritama, TDMA (*Time Division Medium Access*) organizovanom pristupu medijumu kao i primenom tehnike razmene RTS-CTS-DATA-ACK poruka u cilju rezervisanja medijuma za prenos. RTS i CTS poruke u sebi sadrže polje u kom je upisano trajanje razmene. Na taj način, motovi koji ne učestvuju u razmeni poruka, mogu da podese sopstvene NAV tajmere što garantuje da neće pokušati sa slanjem poruka sve dok se tekuća transmisija ne završi. Ova tehnika je implementirana u nekim verzijama MAC protokola i vezuje se za pojam *adaptivnog osluškivanja* o čemu će biti reči kasnije u tekstu.

Zaglavlje protokola (*protocol overhead*) takođe utiče i na potrošnju energije, ali i na propusni opseg. Međutim, ukoliko se želi postići pouzdan prenos podataka, MAC nivo mora da ima implementirane mehanizme koji će omogućiti konfigurisanje pouzdanog sistema za prenos. Stoga, zaglavlje koje MAC dodaje mora da sadrži polje koje ima ulogu pri detekciji (i korekciji) greške, adrese primaoca i pošiljaoca, kao i podatke na osnovu kojih će mot znati kada sme a kada ne sme da pristupa medijumu tj. na osnovu kojih će moći da organizuje svoj “raspored spavanja”.



Slika 4. MAC zaglavlje paketa

Standardni kodovi za korekciju i detekciju greške gotovo udvostručuju količinu podataka koji se prenose dodajući redundantne informacije kako bi kasnije bilo moguće ispraviti grešku, ukoliko do nje dođe. Dodavanjem FCS (*Frame Check Sequence*) polja, umanjuje se broj poruka koje aplikacija može da pošalje u jedinici vremena, ali se i znatno umanjuje verovatnoća potrebe za retransmisijom poruke usled greške pri prenosu, što je naročito bitno u mrežama sa gustim saobraćajem. Sa dužinom od 9 do 25 bajta, zaglavlje MAC protokola je često duže od poruke koja se prenosi.

Još jedan izvor rasipanja energije jeste obrada, a potom odbacivanje poruka (*message overhearing*) koje su namenjene drugim motovima. Ova tehnika je uobičajena u mrežama u kojima ne postoji kontrola utroška energije. Njome se povećava propusna moć mreže a smanjuje kašnjenje isporuke (*latency*), ali se energija troši nemilice, naročito ukoliko prijem poruke zahteva više energije nego

slanje. Postoje dva načina kojima se umanjuju negativne posledice: *rano odbacivanje paketa* i *propuštanje paketa*.

Rano odbacivanje paketa (early rejection) je tehnika koja omogućava motu da isključi radio primopredajnik kada shvati da poruka nije njemu namenjena (adresa destinacije kod *unicast* slanja poruke nije njegova ili mot ne pripada datoj grupi-kod *broadcast* slanja).

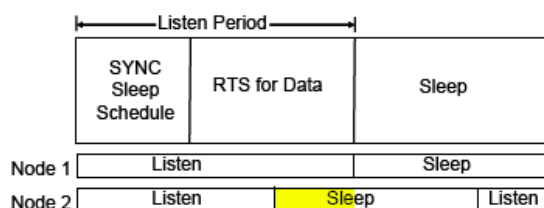
Propuštanje poruka (message passing) je tehnika implementirana u SMAC i TMAC protokolu i ona omogućuje motu da uskladi svoj period "spavanja" i podesi NAV (*Network Allocation Vector*) na osnovu podatka o dužini transmisije, koji je prikupio osluškajući razmenu RTS-CTS poruka neka dva mota.

Danas je u upotrebi više različitih verzija MAC protokola. Ovde su navedene samo neke: SMAC (*Sensor MAC*), BMAC (*Berkeley MAC*), TMAC (*Timeout MAC*), GMAC (*Gateway MAC*) itd.

A. SMAC protokol

SMAC protokol bežičnoj senzorskoj mreži omogućava samokonfigurisanje, uštedu energije (*energy efficient protocol*), ali i adaptivnost prilikom promene topologije (dodavanje novih čvorova mreži ili uklanjanje postojećih). Pored toga, SMAC omogućava pouzdan prenos uz produženi vek trajanja. Umesto da su stalno uključeni, čvorovi tek periodično osluškaju medijum. Na taj način štede znatnu količinu energije jer je radio primopredajnik ubedljivo najveći potrošač na motu. Pored toga, SMAC je iz IEEE 802.11 standarda prisvojio šeme koje definišu pristup medijumu. Ovaj protokol takođe omogućava motovima da poruke koje im nisu namenjene jednostavno propuste (*message passing*) i da ne troše energiju na obradu poruka koje će na posletku odbaciti. Na kraju, SMAC omogućava i efikasnu transmisiju dugih poruka tako sto ih dele u više kraćih i šalju deo po deo (fragmentacija).

Kao što je prikazano na slici 5, SMAC deli raspoloživo vreme na ramove. Ramovi su podeljeni na periode kada mot osluškuje kanal (*listen*) i periode kada je radio isključen (*sleep*). Period kada mot osluškuje kanal je takođe podeljen na dva dela, sinhronizacioni (*SYNC*) i deo namenjen prenosu podataka (*RTS for DATA*).



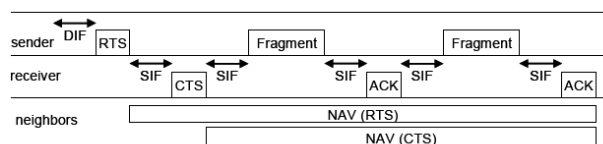
Slika 5. SMAC ram

U sinhronizacionom periodu, motovima je dozvoljeno da emituju sopstveni raspored "spavanja" na taj način što će u SYNC poruci da navedu podatak kada će se prvi sledeći put isključiti ("isključujem se za t sekundi"). Mot koji čuje ovu poruku će usvojiti taj raspored i poruku proslediti dalje. Posledica svega ovoga je formiranje virtuelnih klastera u mreži koje čine motovi koji imaju isti raspored spavanja/osluškivanje. Ukoliko mot za vreme SYNC perioda ne primi SYNC poruku, on će je sam generisati i

poslati u mrežu. Eksperimentalno je pokazano da ukoliko se odnos spavanja/osluškivanje unutar vremenskog frejma podesi na 9/1 (10% duty cycle), postići će se značajna ušteda energije, prvenstveno zahvaljujući skraćivanju *idle listening* perioda.

Adaptivno osluškivanje je mehanizam implementiran u SMAC protokolu, koji omogućava motovima da, preslušavajući razmenu RTS-CTS poruka između suseda, podese sopstvene NAV tajmere i isključe radio interfejs za vreme trajanja razmene tj. dok se komunikacioni kanal ne oslobodi. U zaglavlju RTS i CTS poruka se nalazi informacija o dužini trajanja transmisije.

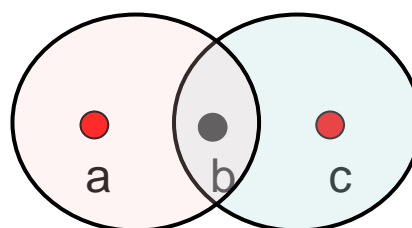
Razmena RTS-CTS poruka pre slanja paketa je metod prevencije od kolizije paketa. Tek kada rezervišu kanal, motovi počinju sa razmenom podataka. Prilikom razmene paketa dužih od 250 bajtova, paket će se, kao što je prikazano na slici 6, izdeliti u više manjih delova i slati deo po deo (*burst*). Bez obzira što se poruka šalje u fragmentima, RTS-CTS poruke će se razmeniti samo jednom, a na svaki primljeni fragment, primalac će poslati potvrdu pošiljaocu (ACK poruka).



Slika 6. Razmena fragmentovane poruke u mreži sa SMAC protokolom

Razmenom RTS-CTS poruka se takođe uspešno rešava problem "skrivenih suseda" (slika 7). U slučaju da se razmena podataka obavlja bez prethodne rezervacije kanala, čvor C bi mogao da počne da šalje podatke čvoru B za vreme razmene podataka između A i B. To bi se dogodilo zato što A nije u dometu C, pa C ne zna da A i B komuniciraju i dolazi do kolizije. U mrežama u kojima je implementiran SMAC protokol i u kojim se kanal rezerviše pre nego što počne transmisija, C će čuti CTS poruku čvora B i podesiti NAV tajmer, tako da neće pokušati komunikaciju sa B sve dok se transmisija između A i B ne završi.

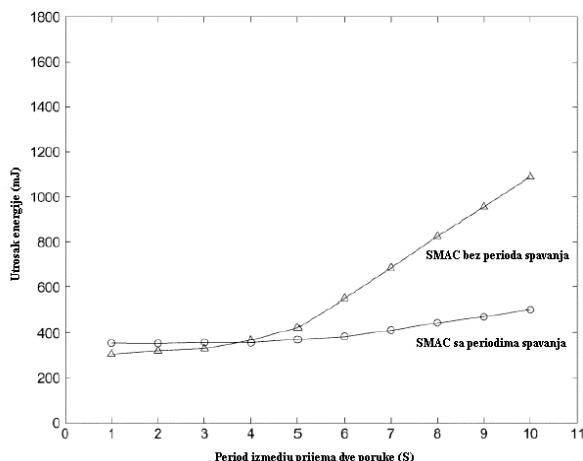
Razmena RTS-CTS poruka kojom se rezerviše kanal se praktikuje isključivo u slučaju *unicast* slanja (slanje na određenu adresu). Pre broadcast slanja poruke se ne vrši razmena RTS-CTS poruka kojim se rezerviše kanal.



Slika 7. Problem "skrivenih suseda" (*hidden neighbours*)

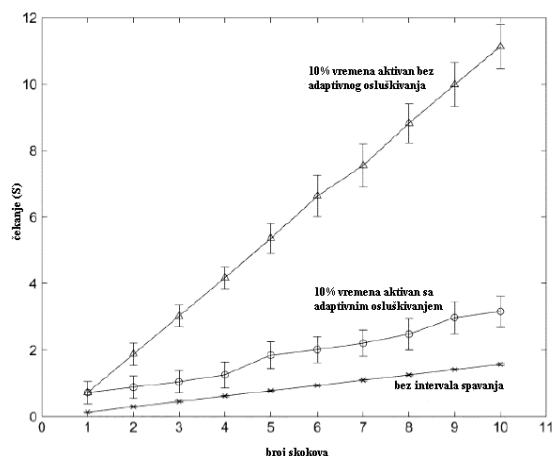
Na grafikonima koji slede su dati rezultati eksperimenata izvršenih na univerzitetu Berkeley, gde je i osmišljen SMAC protokol. Na slici 8 je dat prikaz

potrošnje energije mota u intervalu između prijema dve poruke u slučajevima kada je implementiran SMAC bez perioda spavanja i SMAC sa periodima spavanja:



Slika 8. Zavisnost utroška energije od dužine intervala između prijema dve poruke

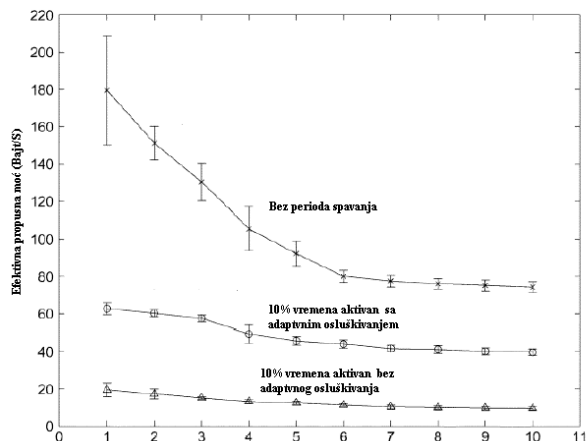
Vidi se da je ušteda značajna tek u situacijama kada je interval između prijema dve poruke veći od 6 sekundi. Na slici 9 je prikazan odnos dužine čekanja (*latency*) i broja skokova potrebnih da poruka dođe do svoje destinacije:



Slika 9. Odnos perioda čekanja i broja skokova potrebnih da paket stigne do destinacije

Sa slike 9 se vidi da se čekanje povećava sa porastom broja skokova, ali i da je najizraženije kod SMAC-a kod kog nije aktivirano adaptivno osluškivanje. To je posledica činjenice da je raspored osluškivanja i spavanja fiksna i da što je veći broj skokova do destinacije raste i verovatnoća da će poruka proći kroz više klastera (različiti rasporedi). Krajnji rezultat je znatno povećanje perioda čekanja u odnosu na mreže gde je implementiran SMAC sa adaptivnim osluškivanjem.

Naposletku, slika 10 prikazuje odnos propusne moći mreže i broja skokova potrebnih da paket stigne do destinacije na koju je poslat:



Slika 10. Odnos efektivne propusne moći mreže i broja skokova potrebnih da paket stigne do destinacije

Vidi se da je u sva tri slučaja prisutan trend opadanja propusne moći sa porastom broja skokova. Jasno se uočava i da propusna moć najmanja u mrežama gde je implementiran SMAC sa adaptivnim osluškivanjem što je i očekivano jer su česte situacije da je medijum slobodan u slučajevima kada određeni motovi, spavaju, a čak i kada ne spavaju, postoji mogućnost da ga neko drugi zauzme pre. To je veoma često posledica brzine propagacije poruke.

LITERATURA

- [1] M.I.Brownfield, "Energy-efficient Wireless Sensor Network MAC Protocol," Dissertation submitted to the Faculty of Virginia Polytechnic Institute and State University
- [2] W. Ye, J. Heidemann, D. Estrin, "An Energy-Efficient MAC Protocol for Wireless Sensor Networks," http://www.isi.edu/~weiye/pub/smac_infocom.pdf
- [3] W. Ye "SMAC Testing Code and Info," http://www.etc.sipfw.edu/~lin/ECET581_CS590/lectures/S_MAC_test_byWeiYe.html

ABSTRACT

This paper presents the most important concepts of Wireless Sensor Networks (WSN), and hardware construction. Specifics of medium access protocols for WSN are elaborated, with emphasis on SMAC (*SensorMAC*) protocol. Results of tests for SMAC protocol are given.

SMAC PROTOCOL IN WIRELESS SENSOR NETWORKS

Miloš Rovčanin