POUZDAN TRANSPORTNI PROTOKOL SA VIŠEADRESNIM SENZORSKIM ČVOROVIMA
RELIABLE TRANSPORT PROTOCOL WITH MULTI ADDRESS SENSOR NODES

Mirko Kosanović, Visoka tehnička škola strukovnih studija u Nišu
Mile Stojčev, Elektronski fakultet u Nišu

Sadržaj – *Zadnjih nekoliko godina bežične senzorske mreže (BSM) postale su veliki izazov za mnoge istraživače. Kako su senzorski čvorovi (SČ) postajali sve manji i jeftiniji, primena BSM je rasla u velikom broju različitih aplikacija. Niska cena hardvera, kao što su CMOS kamere i mikrofoni, omogućio je prikupljanje mnogih multimedijalnih informacija kao što su video i audio podaci iz nadgledanog regiona. Prenos svih ovih podataka od izvora do odredišta zahteva veliku pouzdanost u prenosu tih podataka. Sa druge strane, da bismo omogučili prikupljanje različitih tipova podataka, javlja se potreba za dinamičkim menjanjem softvera u bežičnim čvorovima. Jasno je da će ovaj zahtev biti ispunjen ako se garantuje 100 % pouzdan prenos podataka. U ovom radu, razmatran je problem efikasnog i pouzdanog prenosa podataka u BSM. Predloženo je jedno novo rešenje za prenos informacije u oba smera, nagore (od senzorskog čvora do glavnog čvora) i nadole (od glavnog čvora do senzorskog čvora). Naše rešenje garantuje pouzdan prenos podataka u oba smera, sa znatno povećanim korisnim podacima u paketu u odnosu na nekorisne (smanjeno zaglavlje), u poređenju sa dosadašnjim poznatim rešenjima.*

Abstract – *Wireless Sensor Networks (WSN) has become a big challenging issue for researcher community in the last few years. WSNs have currently deployed in a wide range of applications as a sensor is becoming smaller and production cost is smaller. The low-cost hardware such as CMOS cameras and microphones has required many multimedia data such as video and audio streams, still images, and scalar sensor data from the environment. The transfer for all of this data from source to destination, require being more reliable. On the other side, we have demand for dynamically changes the software in wireless sensor node, to adjust node to be able to collect various type of sense data. This demand required 100% reliable data transport protocol by naturally. In this paper, we look at the problem of efficient and reliable data transport in WSN. We propose a new solution for information delivery in both directions – upstream (from sensor node to master node – sink) and downstream (form master node to sensor node). Our solution ensures the reliable data transport in both directions with much better relation between payload data and header (reduced header) in spite of well known current solutions.*

1. UVOD

Bežične senzorske mreže otvorile su jedno novo, ogromno, a samim tim i neistraženo područje u računarskim mrežama. Veliki razvoj mikro elektro-mehaničkih sistema (*Micro Electro-Mechanical Systems* – MEMS) kao i veliki prodor bežičnih komunikacija, koje zbog svoje cene i jednostavnosti primene preuzimaju primat na polju komunikacija, omogućile su da se sa BSM rešavaju mnogi složeni problemi. Više se njima ne prenose samo jednostavni podaci koje senzori očitavaju: temperatura, vlažnost, pritisak, detekcija prisustva i td. već i mnogi multimedijalni podaci kao slika i zvuk. Očigledno je da se za prenos ovakvih podataka zahtevaju znatno brži i pouzdaniji resursi, jer je količina podataka koja se prenosi višestruko veća. Samim tim povećava se intenzitet toka saobraćaja, koji sada uzrokuje pojavu još većeg broja grešaka koji u znatnoj meri mogu da degradiraju pa i da prekinu rad aplikacije u BSM. Kako je bežična sredina poznata kao sredina sa visokim procentom grešaka u komunikaciji (do 70% [1]), jasno je da pouzdan prenos ovakvih podataka predstavlja jako veliki problem. Sa druge strane, potrošnja el.energije u takvim uslovima se znatno povećava pa se i vek rada pojedinih SČ-ova znatno smanjuje. Ovi problemi ne predstavljaju neki veliki izazov za standardne žičane mreže i uređaje, ali zbog ograničenja kod BSM-a, to predstavlja vrlo ozbiljan i težak zadatak. Ta ograničenja odnose se pre svega na: proizvoljan i nepredvidljiv prostorni raspored, mnogo veću osetljivost na greške pa samim tim i znatno veći broj retransmisija, česte promene topologije senzorskih mreža, kao i velika ograničenja u pogledu napajanja, komunikacione i računarske snage pojedinih SČ-ova [2].

Uzimajući u obzir sve gore navedene mane BSM, očigledno je, da primena standardnih transportnih protokola kao što su TCP i UDP, nije mnogo isplativa u okruženju BSM-a. Zato se i javlja potreba za razvojem jednog novog pouzdanog transportnog protokola koji će sa jedne strane uzeti u obzir sva ograničenja BSM-a, a sa druge strane će ispuniti sve zahteve jednog sigurnog i pouzdanog transportnog protokola. U nastavku rada, u poglavlju 2 objašnjene su neke bitne osobine koje karakterišu pouzdan prenos podataka u BSM, sa posebnim osvrtom na osobine koje bi jedan transportni protokol trebalo da ima u ovakvom jako limitiranom okruženju. Poglavlje 3 daje nam kratak pregled nekoliko najpoznatijih dosada razvijenih transportnih protokola, koji najbolje rešavaju probleme intezivnog i pouzdanog prenosa podataka. U četvrtom poglavlju je dat

naš predlog jednog novog rešenja pouzdanog transportnog protokola, a analiza njegove uspešnosti u odnosu na dosadašnja rešenja data je u petom poglavlju. Šesto poglavlje zaključuje ovaj rad.

2. KARAKTERISTIKE SIGURNOG PRENOSA

Pre nego što ukažemo na osnovne uslove koje mora da zadovolji jedan siguran, pouzdan transportni protokol, potrebno je da sagledamo neke osnovne karakteristike od kojih zavisi taj protokol. To se pre svega odnosi na:

- smer prenosa - može da bude *downstream* od master SČ (*sink)* ka SČ-ima ili *upstream*- od SČ-ova ka master SČ.
- brzina prenosa od izvora do odredišta predstavlja vrlo bitan faktor kod prenosa nekih podataka. To se narocito odnosi kod prenosa multimedijalnih podataka gde postoje stroge granice dozvoljenog maksimalnog kašnjenja primljenih podataka (*real-time* prenos). Ako uzmemo u obzir velika ograničenja BSM u pogledu komunikacionih uslova,višeskokovite topologije, uskog propusnog opsega, asimetričnih veza i jako osetljivoj komunikaciji koja je podložna mnogim greškama, veoma je teško zadovoljiti ovu karakteristiku [2].
- mehanizam prenosa podataka - može biti direktan od master SČ do senzora tj. svaki SČ direktno komunicira sa master SČ (*end-to-end*) ili indirektan gde se prenos odvija i *multy hop* režimu (*hop-by-hop)*.
- načini za potvrdu prijema podataka-razlikujemo pozitivnu (ACK) i negativnu potvrdu primljenog paketa (NACK).
- vrsta adresiranja SČ-podrazumeva potrebu za korišćenjem *unicast, broadcast* ili *multicast* adresa kod komunikacije.
- tip podataka koji se šalje – vrlo bitna karakteristika kod transportnih protokola kada projektujemo stepen njegove pouzdanost. Nije svejedno da li se šalju neki sistemski ili kontrolni podaci (traži se 100% pouzdanost) ili se radi o podacima koje SČ-ovi sakupljaju i kod koji nije potrebna velika pouzdanost u prenosu.

Razmotrićemo sada neke karakteristike koje moramo imati u vidu kada razvijamo jedan pouzdan transportni protokol za BSM kao i šta on mora da ispuni da bi to bio.

1. Mehanizam za upravljanje intenzititetom saobraćaja - kod BSM najveći intenzititet saobraćaja dešava se oko master SČ, jer se tu sakupljaju podaci koji dolaze od SČ-ova iz BSM-a (*upstream*). Za uspešno rešavanje ovog problema potrebno je razviti tri mehanizma i to: za efektivnu detekciju zagušenja u saobraćaju, za izbegavanje zagušenja kao i za upravljanje intenzititetom saobraćaja u BSM-u. Sva tri mehanizma zasnivaju svoja rešenja na kontroli veličine bafera u SČ-ima, kao i na kontroli frekvencije/brzine slanja podataka (ACC-*Active Congestion Control*).

2. Mehanizam za kontrolu pouzdanog prenosa paketa – pouzdan prenos podataka kod BSM-a u većini slučajeva može imati sasvim drugačije značenje nego što je to kod tradicionalnih žičanih veza, gde je potrebno garantovati ispravan prenos svakog paketa. Za neke aplikacije kod BSM dovoljno je primiti samo jedan ispravan paket od jednog SČ iz nekog regiona, a ne od svih jer većinom oni daju istu informaciju. To nam daje veću slobodu u razvoju transportnih protokola za BSM. Međutim u nekim aplikacijama, kao što su prenos sistemskih podataka, promena ili izmena programa u SČ-ima ili kod prenosa više podataka (više paketa iste poruke) moramo obezbediti potpuni sigurni prenos. Za rešavanje ovog problema bolje je koristiti mehanizam hop-by-hop (HbH) od tradicionalnog end-to-end (EtE) mehanizma, jer se na taj način smanjuje intenzitet saobraćaja a samim tim i ušteda potrošnje je veća.

3. Jednostavnost u inicijalnom, početnom prenosu – većina aplikacija u BSM je reaktivnog tipa, gde su SČ-ovi pasivni akteri nekog događaja. Oni posmatraju okolinu, očekujući da se nešto dogodi, i reaguju tek na neku promenu ili prozivku od nadređenog SČ. Sve te promene mogu se smestiti u svega nekoliko paketa koji se šalju nadređenom SČ. Zato je potrebno da inicijalni prenos kod uspostavljanja te veze bude što kraći i jednostavniji.

4. Mali broj ponovljenih slanja paketa (*retransmission*)– vrlo bitna karakteristika koja mora da bude zadovoljena kako bi se izbeglo nepotrebno trošenje el.energije. Ako i dođe do neophodnosti ponovnog slanja paketa treba voditi računa da što manji broj SČ-ova bude uključen u taj prenos. U tom smislu sigurno je da HbH prenos ima prednostu u odnosu na EtE, jer su samo dva SČ uključena u ponovljeno slanje paketa. Mana ovakvog prenosa je da on ne može da nam garantuje pouzdan prenos na nivou cele poruke od predajnika do prijemnika.

5. Mali *header* u odnosu na *payload* podatke – u većini aplikacija, komunikacija u BSM-u se odvija po mehanizmu HbH. Svaki od SČ-ova pored osnovne funkcije, detekcije nekog događaja, treba da zadovolji i funkciju rutera kod preusmeravanja velikog broja paketa koji kroz njega prolaze. Kako se u tim paketima najčešće nalazi veoma mali broj *payload* podataka, od velikog je značaja da *header* tih paketa smanji na najmanju moguću meru kako bi se smanjila ukupna veličina paketa. O tome nam govori podatak da za prenos samo jednog bita informacije trošimo istu količinu el.energije koju potroši procesor u SČ za 1000 instrukcija [3].

6. Svi SČ-ovi moraju da imaju ravnopravan tretman u komunikaciji – kako se radi o mreži sa velikim brojem senzora koji nadgledaju neki region, uvek postoji mogučnost da neki od SČ-ova bude zaspostavljen u komunikaciji. Iz tog razloga informacije koje dolaze do master SĆ mogu biti pogrešne jer nisu kompletne,pa samim tim mogu da dovedu do pogrešnih odluka. Zato treba obezbediti da svi senzori u BSM-u budu ravnopravno tretirani, kako bi se obezbedila pravovremena i realna informacija o promenama u nadgledanom regionu.

7. Saradnja sa susednim čvorovima – poželjno je da postoji međusobna komunikacija između susednih slojeva tj. sa mrežnim slojem. Ako ta komunikacija postoji tada *routing* protokol sa nižeg sloja može da obavesti transportni protokol o nekim problemima koji su se javili u komunikaciji (na primer da obavesti da je gubljenje paketa zbog prekida nekog puta-*route failure* a ne zbog pojačanog saobračaja).

Generalano gledano sve ovo možemo smestiti u tri glavna cilja koji moraju da budu ispunjeni u novom predloženom rešenju a to su: pouzdan protokol, upravljanje intezititetom saobraćaja, kao i laka implementacija tj. minimalni potrebni resursi za realizaciju protokola. Ostvarivanjem ova tri cilja postiže se osnovni zadatak svih protokola u BSM a to je smanjivanje potrošnje el.energije svih SČ-ova, što nam omogućava znatno duži vek rada BSM-a [4], [5].

3. PREGLED DOSADAŠNJIH REŠENJA

Dosadašnji razvijeni transportni protokoli za žičane mreže (TCP, UDP) spadaju u jako zahtevne protokole, jer gotovo svi oni zahtevaju veće resurse za njihovo izvršavanje. Jasno je da bi puna primena ovih protokola u BSM-a bio potpuni promašaj sa gledišta isplativosti i efikasnosti, jer bi morali da za 2-3 bajta korisnih podataka prenosimo čak 30-tak bajtova (samo zaglavlje je kod IPv4-24 bajta, IPv6-40 bajta, UDP-8 bajta, TCP-24 bajta). Zato su razvijeni mnogi drugi alternativni protokoli koji su bili primereni skoromnim resursima kojima SČ-ovi raspolažu [6].

PSFQ (*Pump Slowly, Fetch Quickly*) - obezbedjuje pouzdan prenos od master SČ ka SČ-ima (*downstream*) sa relativno malom brzinom slanja paketa. Master SČ šalje podatke svim susednim čvorovima (*broadcast*), a oni ih prosleđuju dalje (HbH organizacija). Na nivou jednog preskoka vrši se kontrola prenosa paketa pa je potrebno da svaki SČ baferuje paket koji prosleđuje dalje. Ukoliko dođe do gubitka nekog paketa on se odmah obnavlja na nivou dva susedna SČ između kojih je došlo do gubitka. Kako je priliv novih paketa relativno spor, SČ ima dovoljno vremena da ponovno pošalje paket (uzima ga iz svog bafera) [5].

GARUDA – Predstavlja još jedan *downstream* protokol koji se sastoji iz tri komponente: WFP (*Wait-for First-Packet*) koja garantuje uspešan prenos jednog inicijalnog paketa. Taj paket treba da odredi SČ-ove koji će predstavljati okosnicu puta preko koga će se odvijati pouzdan prenos podataka (*Core sensor nodes*). Za izbor tih senzora zadužena je druga komponenta protokola. Tu važi pravilo da samo oni senzori koji imaju rastojanje 3**i* skoka (*HopCount*), gde je *i* ceo broj, mogu biti kandidati da postanu *Core sensors*. Treća komponenta je zadužena za retransmisiju loših paketa i sastoji se iz dva dela: retransmisije za *Core sensors* i retransmisije za *non-Core sensors* korišćenjem NACK sekvence. Glavni nedostaci ovog protokola su što garantuje pouzdan prenos samo u jednom pravcu od master SČ ka SČ-ima(*downstream*) i nema kontrolu intenzititeta saobraćaja [4].

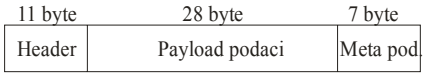
RMST (*Reliable Multi-Segment Transport*) – protokol koji obezbeđuje pouzdan prenos podataka od SČ-ova do master SČ(*upstream*). Oslanja se na protokol rutiranja *Directed Diffusion,* od koga dobija put kojim treba da idu podaci od senzora do master SČ. Za otkrivanje izgubljenih paketa koristi se mehanizam vremenskog tajmera. Kada on istekne, šalje se NACK poruka prethodnom SČ sa označenim brojem paketa koji nije primljen. Moguće je koristi HbH *mode*, ako se vrši keširanje primljenih paketa u svakom SČ,ili EtE *mode* kada se paketi ne pamte u svakom čvoru. Glavni nedostaci ovog protokola su da ne vodi računa o intenzititetu saobraćaja, nema mehanizam za upavljanje potrošenom elek. energijom u SČ-ima kao i što garantuje pouzdan prenos samo za pojedinačne pakete ali ne i za nivo cele aplikacije [5].

ESRT (*Event to Sink Reliable Transport*) – cilj ovog protokola je da obezbedi siguran prenos podataka koje su SČ-ovi prikupili o događaju, do master SČ (*upstream*). Ovde se vrši prilagođavanje brzine prikupljanja tih podataka u zavisnosti od kvaliteta komunikacione veze i zahteva aplikacije. Master SČ periodično vrši proveru pouzdanosti prijema paketa u jedinici vremena, i na osnovu toga vrši prilagođenje brzine slanja, šaljući tu informaciju svim SČ-ima. Kvalitet veze se procenjuje u svakom SČ, bilo nadgledanjem lokalnog reda čekanja ili korišćenjem bita koji ukazuje da je veza do tog čvora loša. Ovaj protokol garantuje pouzdanost samo na nivou aplikacije ali ne i za svaki paket pojedinačno. Nedostaci su da je brzina slanja podataka ista za sve SČ-ove u BSM kao i da se koristi kanal sa pojačanom snagom emitovanja (*one-hop*) [4].

CODA (*Congestion Detection and Avoidance*)–kao što samo ime ovog protokola kaže, njegov osnovni cilj je da detektuje i izbegne sudare (*congestion*) u prenosu podataka. To se postiže na osnovu posmatranja bafera SČ-ova kao i opterećenosti bežičnog kanala. Kada SČ detektuje da je došlo do prekoračenja osnovnog praga tih vrednosti, obaveštava njegov najbliži *upstream* susedni SČ da smanji brzinu prenosa. Predstavlja *upstream* protokol i veoma je sličan ESRT-u. Nedostatak ovog protokola je da ne nadgleda pouzdan prenos podataka već samo intenzitet saobračaja [5].

4. PREDLOŽENO REŠENJE

Prilikom razvoja transportnog protokola pošli smo od činjenice da novi protokol mora pre svega da ispunjava zahteve *energy-aware* tipa protokola, kao i da je dovoljno jednostavan kako bi se implementirao u hardver i softver velikog broja BSM aplikacija. Pored toga pošli smo od činjenice da će u budućnosti, zbog povećane potrebe prenosa većeg broja multimedijalnih podataka, biti potreban protokol koji će omogućavati pouzdan prenos u oba smera. Zato predloženo rešenje podjednako tretira i omogućava pouzdan prenos podataka(*reliable transport data*) u oba smera, na niže od master SČ ka SČ i na više od SČ-ova prema master SČ.



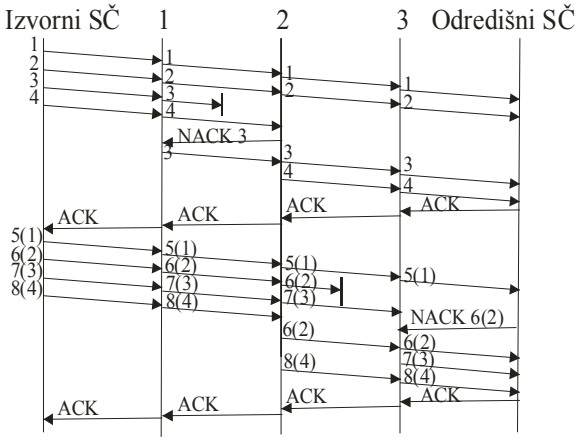
Header:		*Meta podaci*:	
lenght - dužina okvira	(1 byte)	**tx_power** - količina el.energ.	(1 byte)
fct - polje za kontrolu prijem.	(2 byte)	**rssi** - indik.kvaliteta prijema	(1 byte)
dsn - broj okvira	(2 byte)	**lqi** - indikator kvaliteta veze	(1 byte)
destpan - ID aplikacije	(1 byte)	**crc** - CRC kod	(1 byte)
dest.addr. - adresa odredišta	(2 byte)	**ack** - ACK polje okvira	(1 byte)
sour.addr. - adresa izvorišta	(2 byte)	**time** - vreme slanja okvira	(1 byte)
type - tip podataka u okviru	(1 byte)	**rx_interval** - interval prijema	(1 byte)

Slika broj 1. Struktura message_t paketa kod TinyOS

U većini aplikacija koje trebaju da prenesu neku multimedijalnu informaciju, komunikacija se sastoji od velikog broja paketa koje je potrebno poslati za samo jednu informaciju–sliku.Za slanje jedne tipične video slike potrebno nam je oko 25-30 kB (28888 bajta kod JPEG ili 319254 bajta kod BMP) [7]. U najpopularnijem operativnom sistemu, TinyOS, koji je postao gotovo standard kod primene u BSM-a, za komunikaciju između SČ-ova koristi se struktura *message_t* (slika br.1). Kod nje je standardna veličina korisnih podataka 28 bajta a veličina zaglavlja 11 bajta (kod komunikacionog čipa CC2420, standard IEEE 802.15.4 max.veličina paketa je 128 bajta), ali se te veličine mogu programski promeniti [8]. Imajući u vidu da standardni paket koji se šalje zauzima od 28-110 bajta korisnih (*payload*) podataka (*message_t* kod TinyOS), ispada da je broj paketa koje treba poslati za samo jednu sliku jako veliki i prelazi cifru od preko hiljadu paketa. Kako se u tim paketima najčešće nalazi veoma mali broj korisnih podataka u odnosu na podatke koji se odnose na *header* i *metadata* (28/18), od velikog je značaja da se ti ''nekorisni'' podaci smanje na najmanju moguću meru. Na taj način povećali bi se korisni

podaci a ukupna dužina paketa bi ostala ista, što bi dovelo do znatnog smanjivanja broja paketa koji se šalju. Kako se zbog specifične topologije ovih mreža, komunikacija odvija po višeskokovitom mehanizmu (*hop-by-hop*), svaki od SČ-ova pored osnovne funkcije, detekcije nekog događaja, treba da zadovolji i funkciju rutera kod preusmeravanja velikog broja paketa koji kroz njega prolaze. To znači da u jednom slanju podataka učestvuje više SČ-ova, pa ušteda samo jednog bajta u toj višestrukoj komunikaciji se multiplicira sa brojem preskoka. Sa druge strane smanjivanje broja paketa, smanjuje i intenzitet saobraćaja, a time se smanjuje i procenat grešaka. Sve to doprinosi uštedi velike količine el. energije u SČ-ima, što je i osnovni cilj svakog protokola u BSM [3].

Osnovna ideja našeg rešenja sastoji se u smanjivanju neophodnog zaglavlja (*header*), a povećanju prostora koji prenosi korisne podatke. To je urađeno tako što su iz zaglavlja izbačena 2 bajta koja su služila za označavanje jedinstvenog broja paketa koji se šalje - *dsn* (*data sequence number*). Da bi zadržali jedinstvenu identifikaciju paketa koji se šalju, to je kompenzovano višestrukim različitim adresama koje se dinamički, unapred dodeljuju SČ-ima. Svim adresama u okviru jednog segmenta senzorske mreže raspolaže glavni (*master*) čvor tj. *sink*. Sa gledišta IP adresiranja taj čvor se ponaša kao jedan dinamički DHCP server koji u zavisnosti od intenziteta sabraćaja, kvaliteta veze i smera podataka, dodeljuje te adrese svakom SČ. Jedan čvor u senzorskoj mreži može imati jednu, dve, četri ili više adresa (2^n adresa), koje se dodeljuju na osnovu opterećenosti saobraćaja, broja SČ-ova kojima se poruke šalju, kao i količine podataka koje se šalju. Ukoliko se radi o *downstream* komunikaciji master SČ sve raspoložive adrese dodeljuje SČ-ima. U slučaju *upstream* komunikacije svaki SČ ima samo jednu bazičnu adresu, dok su sve preostale adrese dodeljene master SČ-u. Svaki okvir postaje jedna jedinstvena poruka koja može da se šalje potpuno nezavisno od ostalih, a adresa odredišta predstavlja fizičko mesto gde se ta poruka stvarno nalazi u kompletnoj poruci koja se šalje. Redosled pristizanja poruka na odredištu nije bitan, jer se na osnovu različitih adresa vrši rekonstruisanje stvarne poruke. To je povoljna okolnost jer možemo da koristimo bilo koji do sada razvijen mrežni(*routing*) protokol za BSM, koji podržava istovremeno usmeravanje podataka na više različitih puteva.



Slika broj 2. Redolsed slanja okvira između SČ-ova

Ukoliko je veličina poruke koja se šalje veća od broja okvira tj. raspona adresa koje su dodeljene SČ-ima, tada se poruka deli na blokove gde svaki blok ima onoliko okvira koliko je adresa dodeljeno SČ-ima. Na slici broj 2. prikazan je tok slanja jedne poruke koja se sastoji od dva bloka od po 4 okvira(odredišnom SČ dodeljene su 4 adrese), kao i postupka koji se sprovodi kod lošeg prijema jednog okvira. Sa slike se vidi da se između prolaznih SČ-ova koristi tehnika NACK (slično izloženom rešenju kod PSFQ), što znači da prolazni čvorovi mogu da prihvataju okvire koji imaju sukcesivne adrese u proizvoljnom redosledu. Ukoliko se prihvati neki okvir koji nije u redosledu, NACK poruka se šalje tek posle isteka brojača koji će kasnije biti objašnjen. Svaki od čvorova prihvata okvir, vrši njegovo baferovanje i prosleđuje ga narednom čvoru. Poželjno je da velićina bafera u svakom čvoru bude ista kod svih SČ-ova. Nakon što krajnji odredišni SČ primi sve okvire iz prvog bloka (u našem slučaju sva 4 okvira), on šalje ACK poruku izvornom SČ koji je započeo ovo slanje. Na osnovu toga svaki SČ može da resetuje svoj bafer i pripremi se za prijem novog bloka podataka. Zbog prirode BSM, pre svega zbog velikog broja grešaka u komunikaciji, moguće je da mnogi prolazni SČ ne prime ispravne okvire. Da bi se ipak omogućilo svakom SČ da može da primi sve poslate okvire kao i da resetuje svoj bafer i prima nove okvire sa istim adresama, uvedena su tri *delay* brojača na svakom SČ. Uloga prvog brojača je da kod prijema okvira onemogući trenutno slanje NACK poruke, ukoliko ne dobije neki okvir po redosledu, jer se pretpostavlja da on može i da zakasni zbog višestrukih puteva. Drugi brojač je uveden da omogući svakom SČ dovoljno vremena da može da ponovi slanje traženog okvira. On ima ulogu da kontroliše brzinu slanja okvira a samim tim i intenzitet saobraćaja u BSM-u. Svako detektovanje NACK poruke uslovljava povećanje vrednosti tog *delay* brojača SČ koji je primio tu poruku. U inicijalnom stanju stanje tog brojača je postavljeno na 0,a njegova vrednost se povećava u zavisnosti od broja NACK poruka koje su stigle do njega. Što je veći broj primljenih NACK poruka, veća je i vrednost brojača jer se pretpostavlja da je došlo do pogoršanja u vezi: veća gustina saobraćaja ili smetnje u komunikaciji. Treći brojač ima ulogu da omogući resetovanje prijemnog bafera ukoliko zbog asimetričnih veza i otežanih uslova, nije primljena povratna ACK poruka. Taj brojač se aktivira nakon prijema svih potrebnih okvira iz jednog bloka (4 okvira u našem slučaju). Sada, ukoliko se primi novi set podataka sa istim adresama a vrednost brojača je istekla, moguće je resetovati bafer i primiti novi blok podataka bez obzira da li smo primili ACK potvrdu od odredišnog čvora. Kada ACK poruka stigne do izvorišnog SČ on nastavlja sa slanjem sledećeg bloka podataka koji se smeštaju u okvire koji imaju iste adrese kao i prethodni, ali sa drugim informacijama (sada su to okviri 5,6,7 i 8 a imaju adrese 1,2,3 i 4 respektivno). Postupak se ponavlja sve dok se ne pošalje celokupna poruka.

5. ANALIZA USPEŠNOSTI PROTOKOLA

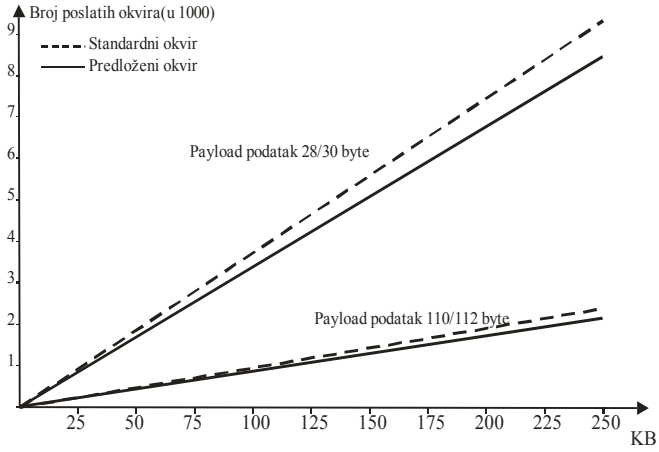
Glavna ušteda kod dodeljivanja višestrukih adresa svakom SČ dobija se u smanjivanju potrebnih bajtova u *header*-u okvira koji se šalje. Sada nema potrebe da se šalje redni broj okvira jer je on jednoznačno određen sa adresom SČ. Analiziraćemo sada uštedu koju postižemo korišćenjem predloženog protokola. Pretpostavićemo da senzorska mreža radi u idealnim uslovima tj. da nema dodatnih retransmisija i analiziraćemo ukupan broj korisnih bajtova koji se pošalje od master SČ do SČ, u zavisnosti od broja okvira koji se šalju. Analiziračemo slučaj kada je veličina okvira standardne dužine od 46 bajta (11 bajta *header*, 28 bajta *payload* podaci

i 7 bajta *meta* podaci) i max. Dužina okvira od 128 bajta (11 bajta *header*, 110 bajta *payload* i 7 bajta *meta data*). Neka je:

p- ukupna dužina poruke koja se šalje
d_p- dužina *payload* podataka u okviru
n_i – broj SČ-ova u u putanji od izvornog do odredišnog SČ
S_s – ukupan broj okvira koji se šalju postojećim rešenjima
S_n – ukupan broj okvira koji se šalju novim protokolom
S_u – broj okvira koji se manje šalje novim protokolom

$$S_u = S_s - S_n = p * n_i / d_p - p * n_i / (d_p + 2) = p * n_i / d_p * (d_p + 2) \quad [1]$$

Iz jednačine [1] sledi da je ušteda u našem predloženom rešenju direktno srazmerna veličini poruke koja se šalje kao i broju SČ kroz koje ona prolazi a obrnuto srazmerna veličini korisnog dela podataka u okviru. Što je korisni deo podataka u okviru manji, imamo veći broj okvira pa samim tim i veću uštedu. Zbog prirode BSM-a i dobro poznatih ograničenja u resursima SČ, veličina okvira koji se šalje između SČ-ova je veoma mala (od 46 do 128 bajta kod TinyOS messages_t). To se odlično uklapa sa predloženim protokolom koji naročito dolazi do izražaja kod slanja većeg broja okvira. Dva su osnovna razloga: povećani broj okvira koje treba poslati da bi se ta vrsta informacije prenela i veliki stepen pouzdanosti prenosa tih okvira. Ako uzmemo u obzir i jako loše uslove u kojima ove mreže rade, samim tim i povećani broj ponovnog slanja okvira – retransmisija, jasno je da ovaj protokol još više dobija na svojoj uštedi.



Slika broj 3. Zavisnosti broja okvira od veličine poruke

Na slici broj 3. prikazan je uporedni prikaz zavisnosti ukupnog broja okvira u odnosu na veličinu informacije koja treba da se šalje kod standardne i predložene veličine okvira. Linearna zavisnost jasno pokazuje da se sa povećanjem broja okvira tj. sa većom količinom podataka koji se šalju, razlika poslatih okvira se povećava u korist predloženog rešenja. Treba istaći još jednu karakteristiku predloženog protokola a to je broj adresa koje se dodeljuju SČ-ima jer one imaju ulogu u poboljšanju prenosa podataka. Što je taj broj veći bolje su i karakteristike predloženog protokola, jer je moguće poslati veći broj okvira u okviru jedne grupe pa se samim tim smanjuje broj kontrolnih ACK povratnih poruka. Protokolom je predviđeno da se dva bajta rezervišu za adresiranje SČ-ova, što nam omogućava ukupno 2^{16} različitih adresa. Kako je to veliki broj adresa mi smo viši adresni bajta podelili na dva dela. Prvih 4 bita iskoristili smo za informaciju o kom skoku (*hop*-u) pripadaju SČ-ovi (max. 16 skokova). Na taj način omogučili smo da SČ-ima iz istog skoka dodelimo 2^{12} (4096) adresa. Sledeći skok takođe će imati 2^{12} istih adresa ali koje će biti jednoznačno određene zahvaljujući različitom skoku kome pripadaju. To znači da nam na raspolaganju stoji mogućnost adresiranja ukupno 16 skokova i u svakom od njih imamo 4096 različitih adresa ili 8 skokova po 8192 adresa. Treba istaknuti još jednu prednost koju nam predloženi protokol daje a to je smanjena potreba za velikim baferima kod SČ-ova, što nije zanemarujuća činjenica kada su u pitanju skromni memorijski resursi kojima SČ-ovi raspolažu. Predloženi protokol deli sve poruke na više blokova, u okviru kojih imamo onoliko okvira koliko je adresa dodeljeno SČ kome je ta poruka upućena. Zato svaki SČ ne mora da rezerviše veliki broj bafera koji je potreban kod prenosa velikih poruka-okvira. Kao kod protokola klizajućih prozora i ovde imamo prihvatanje okvira koji su jednoznačno određeni različitim adresama. Kada se prozor napuni tj. pristignu svi okviri sa adresama iz dodeljenog raspona adresa, to znači da je primljen kompletan jedan blok poruke. Nakon toga odredišni SČ ACK porukom vrši 'resetovanje' svih bafera koji su učestvovali u prenosu tog bloka, i na taj način omogučuje prijem novog bloka podataka koji se baferuju na istim memorijskim mestima kao i prethodni primljeni blok.

6. ZAKLJUČAK

Velika razlika u karakteristikama žičanih i bežičnih medijuma, uslovila je razvojem velikog broja različitih protokola. Očigledno je, da je zbog otežanih uslova rada, vrlo teško napraviti mrežne protokole koji bi garantovali 100% pouzdan prenos informacija. Ako tome dodamo i veoma siromašne resurse kojima raspolaže BSM, izazov postaje još veći. Dalji rad na ovom protokolu zahtevaće praktične provere na stvarnom fizičkom modelu, kao i eksperimentisanju sa veličinom okvira koji se šalju kao i brojem tih okvira tj. broja višestrukih adresa koje se dodeljuju SČ-ima.

LITERATURA:

[1] Z.Rosberg, R.P.Liu, A.Y.Dong, L.D.Tuan,S.Jha, ''ARQ with Implicit and Explicit ACKs in WSN'', http://www.ict.csiro.au/staff/ren.liu/publications/2008-rosberg-glob.pdf , posećen 20.01.2010

[2] Anna Hac, ''Wireless Sensor Networks Designs'', John Wiley & Sons Ltd, 2003

[3] J.Jones, M.Atiquzzaman, ''Transport Protocols for WSN: State-of-the-Art and Future Directions'', International Journal of Distributed Sensor Networks, 3: 119-133, 2007

[4] C.Wang, M.Daneshmand, B.Li, K.Sohraby, ''A Survey of Transport Control Protocols for Wireless Sensor Networks'', IEEE Network Magazine, Special Issue on WSN, 20(3):page 34-40, 2006

[5] H.Karl, A.Willig, ''Protocols and Architectures for WSN'', John Wiley & Sons, Ltd, 2005

[6] A.Dunkels, ''Towards TCP/IP for Wireless Sensor Networks'', Malardalen University Licentiate Thesis No.45, Swedish Institute of Computer Science, March 2005.

[7] H.Öztarak, A.Yazici, D.Aksoz, R. George,''Multimedia Processing in WSN'', IEEE 4th International Conference Innovations in Information Technology, page78-82, 2007.

[8] P.Levis, TEP 111, message_t, http://www.tinyos.net/tinyos-2.x/doc/html/tep111.html, posećen 20.01.2010