



UNIVERZITET U NIŠU
ELEKTRONSKI FAKULTET



Goran S. Nikolić

**VISOKO-POUZDAN PRENOS PODATAKA
KOD BEŽIČNIH SENZORSKIH MREŽA SA
MALOM POTROŠNJOM ENERGIJE
PRIMENOM 2D-SEC-DED TEHNIKE
KODIRANJA**

DOKTORSKA DISERTACIJA

Niš, 2019



UNIVERSITY OF NIŠ
FACULTY OF ELECTRONIC ENGINEERING



Goran S. Nikolić

**HIGH-RELIABLE DATA TRANSFER IN
LOW-POWER WIRELESS SENSOR
NETWORKS USING 2D-SEC-DED
ENCODING TECHNIQUE**

DOCTORAL DISSERTATION

Niš, 2019

Podaci o doktorskoj disertaciji

Mentor: dr Goran S. Jovanović, redovni profesor, Elektronski fakultet Niš, Univerzitet u Nišu

Naslov: Visoko-pouzdan prenos podataka kod bežičnih senzorskih mreža sa malom potrošnjom energije primenom 2D-SEC-DED tehnike kodiranja

Rezime: Ova disertacija se bavi izazovima energetske efikasnosti u sistemima sa ograničenim resursima tipa homogenih i heterogenih bežičnih senzorskih mreža za aplikacije prikupljanja podataka u realnim uslovima. Istraživanje pokriva nekoliko oblasti od optimizacije fizičkog sloja do rešenja za mrežni sloj. Problem koji se rešava sagledan je iz tri različite perspektive: energetske profila čvorova sa posebnim osvrtom na aktivnost bloka senzovanja, mrežnog protokola sa posebnim osvrtom na pronalaženje adekvatne tehnike kodiranja kojom bi se smanjila ili eliminisala potreba za ponovnim slanjem i evaluacija dometa predaje za tako odabranu tehniku kodiranja. Ako energetske efikasnost u bežičnim senzorskim mrežama formulišemo kao problem balansiranja opterećenja tada blok upravljanja napajanjem može, u značajnoj meri, da doprinese smanjenju potrošnje adekvatnim uključivanjem/isključivanjem pojedinih podblokova senzorskog čvora nezavisno od hardverske platforme. Smanjenjem potrošnje energije ostvaruje se produžetak životnog veka senzorskog čvora, a samim tim i cele senzorske mreže. Analizirani energetske profili otkrivaju značajne razlike u potrošnji energije bežičnih senzorskih čvorova u zavisnosti od broja spoljašnjih senzora, rezolucije analogno-digitalnog konvertora, dinamike mrežnog saobraćaja, topologije mreže, primenjenih tehnika kodiranja, režima rada i aktivnosti tokom životnog veka senzorskog čvora i mnogih drugih faktora. U tom smislu predložena je kombinovana primena tehnika svesnih energetske potrošnje (*power aware*) koja obuhvata faktor ispune (*duty-cycling*) na sistemskom nivou i upravljanje napajanjem (*power gating*) na nivou senzorskih elemenata tj. senzora. Procena performansi pokazuje da se, u proseku, može ostvariti smanjenje potrošnje energije za tri reda veličine kada su pomenute dve tehnike implementirane u senzorski čvor. Sa druge strane, u bežičnim senzorskim mrežama izbor šeme kodiranja tj. kanalno kodiranje zavisi od aplikacija i karakteristike-, modela- i tipa-grešaka u bežičnom kanalu. Tako na primer, jedna tehnika kodiranja je poželjnija za upotrebu kada je pojavljivanje grešaka rafalnog tipa (*burst*) dominantni, dok je druga tehnika kodiranja prihvatljivija u situacijama kada šum prouzrokuje slučajne greške koje su po prirodi jednostruke ili dvostruke. Vođeni ovom

konstatacijom uz analizu karakteristika kanala, u disertaciji predložimo tehniku kodiranja poruka kojom proširujemo tradicionalne protokole u cilju poboljšanja njihove energetske efikasnosti, zadržavajući visoku pouzdanost u prenosu podataka i malu latenciju poruka.

Naime, evaluacija kanala u bežičnim senzorskim mrežama zastupljenim u industriji pokazuje da je većina grešaka jednostruke ili dvostruke prirode, a greške tipa *burst* su prisutne, ali retko. U tom kontekstu u disertaciji je razvijena efikasna tehnika za ispravljanje grešaka u hodu ili na odredištu (FEC) zasnovana na *Hamming*-ovoj šemi kodiranja relativno niske složenosti i nazvana je *Two Dimensional-Single Error Correcting-Double Error Detection (2D-SEC-DED)*. Predložena tehnika kodiranja ima za cilj da minimizira ponovno slanje paketa čime se ostvaruje ušteda energije. Evaluacija predložene šeme kodiranja pokazuje da je kôd u stanju da ispravi sve jednostruke greške i 99.99% dvostrukih/višestrukih grešaka. Analiza je izvedena kroz implementaciju, u MATLAB-u, dve verzije *Rendezvous* protokola za dugotrajni život (*Rendezvous Protocol for Long Life - RPLL*) nazvane Modifikovani (*Modified*) RPLL (*M-RPLL*) i Standardni (*Ordinary*) RPLL (*O-RPLL*). Energetska dobit, koja se na ovaj način ostvaruje, iskorišćena je za poboljšanje performansi bežičnog prenosa povećanjem dometa prenosa. Ilustracije radi, za prenos signala u zatvorenom prostoru okarakterisan eksponentom gubitaka na putu $\alpha=4$ na ciljanom $BER = 5 \cdot 10^{-4}$ predložena šema kodiranja je u stanju da poboljša domet za oko 18 m ili jačinu primljenog signala (*RSSI*) za oko 8.5 dBm u poređenju sa bežičnim senzorskim mrežama kod kojih je zastupljena šema kodiranja bez sposobnosti ipravljanja grešaka.

Naučna oblast:	Elektrotehnika i računarstvo
Naučna disciplina:	Elektronika
Ključne reči:	Bežične senzorske mreže, Kodovi za korekciju greške na odredištu, Energetska efikasnost, 2D SEC-DED šema kodiranja, Opseg predaje
UDK:	(004.738.5.057.4:681.586):621.3.016.4
CERIF klasifikacija:	T170 Elektronika
Tip licence kreativne zajednice:	CC BY-NC-ND

Data on Doctoral Dissertation

Doctoral
Supervisor:

dr Goran S. Jovanović, full time professor, Faculty of Electronic Engineering, University of Niš

Title:

High-reliable data transfer in low-power wireless sensor networks using 2D-SEC-DED encoding technique

Abstract:

This dissertation deals with the challenges of energy efficiency in systems with limited resources of homogeneous and heterogeneous wireless sensory networks for data collection applications in real environments. This research covers several fields from physical layer optimization up to network layer solutions. The problem which has to be solved is viewed from three different perspectives: the energy profile of the nodes with a special emphasis on the activity of the sensing block, the network protocol with a special focus on finding an adequate coding technique that need to reduce or eliminate the request for retransmission and evaluating the range of transmission for the proposed encoding technique.

If energy efficiency in wireless sensor networks is formulated as a load balancing problem then the power management unit can significantly contribute to reduction in power consumption. Power management is implemented by switching on/off individual sub-blocks of the sensor node independently of the hardware platform. By reducing energy consumption both an extension of the lifetime of the sensor node and sensor network, is achieved. The obtained energy profiles reveal significant differences in energy consumption of wireless sensor nodes depending in terms of external sensors number, resolution of the analog-to-digital converter, network traffic dynamics, topology of the network, applied coding techniques, operating modes and activities during the lifetime of the sensor node and other factors.

In this sense, the application of combination of power aware techniques, such as the duty-cycling at system-level, and power gating at the level of sensor elements, i.e. sensors, is proposed. An evaluation of the approach shows that energy consumption reduction three orders of magnitude on average can be achieved, when these two techniques are incorporated into the sensor node.

On the other hand, in the wireless sensor networks, the choice of coding scheme, i.e. channel coding depends on the application and characteristics-, model-, type-errors appearing in the wireless channel. For example, one encoding technique is preferred for use when burst errors patterns are dominant, while another coding technique is more acceptable in situations where noise causes random errors that are either single or double in nature. Bearing this in mind, along with the analysis of channel characteristics, in this dissertation, we propose a

new message coding technique by which on extend traditional protocols with aim to improve energy efficiency, while maintaining high reliability in data transmission and low latency of message transfer.

Namely, channel evaluation in wireless sensor networks used in industry shows that most of the errors are of single or double nature, and burst type errors are present, but rarely. In this context, in this dissertation, an effective technique for correcting errors at a destination (FEC) based on Hamming's coding scheme of relatively low complexity, called Two Dimensional-Single Error Correction-Double Error Detection (2D-SEC-DED) was developed. The proposed encoding technique is intended to minimize packet retransmissions, thus saving energy. Evaluation of the proposed encoding scheme shows that the code is able to correct all single errors and 99.99% of double/multiple errors. The analysis was carried out through the implementation, in MATLAB, of two versions of Rendezvous Protocol for Long Life (RPLL), called Modified RPLL (M-RPLL) and Ordinary RPLL (O-RPLL), respectively. The energy gain achieved in this way is used to improve the performance of wireless transmission, such as increasing of the transmission range. As illustration, for indoor environment characterized by the path loss exponent $\alpha = 4$ at the target BER of $5 \cdot 10^{-4}$, the proposed encoding scheme is able to improve the transmission distance by about 18 m, or the received signal strength (RSSI) by about 8.5 dBm compared to wireless sensor networks with encoding schemes without possibility to correct errors.

Scientific Field:	Electrical and Computer Engineering
Scientific Discipline:	Electronics
Key Words:	Wireless sensor networks, Forward error correcting codes, Energy efficiency, 2D SEC-DED encoding scheme, Transmit range
UDC:	(004.738.5.057.4:681.586):621.3.016.4
CERIF Classification:	T170 Electronics
Creative Commons License Type:	CC BY-NC-ND

SADRŽAJ

1	UVOD.....	12
2	BEŽIČNE SENZORSKE MREŽE.....	18
2.1	Klasifikacija senzorskih mreža.....	20
2.2	Primeri aplikacija u svetu.....	21
2.3	Tipovi bežičnih senzorskih mreža.....	24
3	ANATOMIJA SENZORSKOG ČVORA.....	28
4	OPIS PROBLEMA I PREGLED TEKUĆEG STANJA.....	31
4.1	Opis problema.....	32
4.2	Svrha ove teze.....	32
4.3	Povezani radovi.....	33
4.4	Teorijska osnova kontrole grešaka.....	33
4.5	Korekcija grešaka na odredištu.....	34
4.6	Kontrola grešaka u bežičnim senzorskim mrežama.....	35
4.6.1	Izbegavanje retransmisije.....	36
4.6.2	Povećanje broja preskoka.....	36
4.6.3	Upravljanje predajnom snagom.....	36
4.7	Gubici na putu.....	37
4.8	Verovatnoća greške na bit nivou i na nivou paketa.....	37
4.9	Verovatnoća greške na nivou kodne reči.....	38
4.10	Model sistema.....	39
5	VRSTE MEHANIZAMA ZA KONTROLU GREŠAKA.....	41

5.1	Kada je FEC koristan?.....	42
5.2	Vrste kodova za ispravljanje grešaka	43
5.3	Izbor odgovarajućih kandidata kodova za ispravljanje grešaka.....	43
5.4	Tipovi modulacionih šema i njihov značaj za prenos signala.....	44
5.5	Model kanala	45
6	STRUKTURA SENZORSKOG ČVORA	48
6.1	Povezivanje senzorskih elemenata - senzora	50
6.2	Tipovi senzora u bežičnim senzorskim mrežama	50
6.3	Karakterizacija potrošnje energije senzora	51
6.4	Osnove potrošnje energije: faktori, izvori i primeri.....	54
6.5	Tipovi potrošnje energije i snage	55
6.6	Potrošnja struje i njene komponente	56
6.7	Disipacija snage u <i>CMOS</i> tehnologiji	57
6.8	Maksimalna snaga, prosečna snaga, snaga najgoreg slučaja i tipična vrednost snage.....	59
6.9	Hardverske funkcije ugrađene u mikroprocesor i kako ih iskoristiti	60
6.9.1	Gejtovanje napajanja.....	60
6.9.2	Gejtovanje takta	60
6.9.3	Primeri režima rada za uštedu snage kod <i>MSP430FR5969</i> mikrokontrolera	61
6.9.4	Upravljanje taktom i naponom napajanja	62
6.9.5	Optimizacija na nivou periferija i algoritama	62
6.9.6	Komunikacija periferija jezgo.....	63
6.9.7	Algoritamski pristup optimizaciji	63
6.9.8	Sekvence koda male potrošnje i obrasci podataka.....	64
6.10	Izbor tehnika za smanjenje potrošnje energije	65
6.11	Podsistem napajanja	67
6.11.1	LDO vs prekidačka logika	67
6.11.2	Upravljanje opterećenjem	70
7	HARDVERSKA REALIZACIJA SENZORSKOG ČVORA	72
7.1	Podsistem za obradu.....	73
7.2	Radio podsistem	73
7.3	Jedinica napajanja	73
7.4	Senzorski podsistem.....	75
8	ENERGETSKI PROFIL	77
8.1	Tehnika duty-cycling.....	78

8.1.1	Analiza tehnike faktora ispunje.....	78
8.1.2	Gejtovanje napajanja.....	80
8.2	Procena potrošnje energije	85
8.3	Energetski profil tokom aktivnosti <i>initialization i sensing</i>	87
9	TOLERANCIJA NA KVAROVE U BEŽIČNIM SENZORSKIM MREŽAMA.....	93
9.1	Pouzdanost prenosa podataka u bežičnim senzorskim mrežama	93
9.2	Kodovi za ispravljanje grešaka u bežičnim senzorskim mrežama	94
9.3	Osnovni koncepti kodiranja	98
9.4	Osnovni principi kodiranja.....	98
9.5	Blok kodovi i konvolucionni kodovi.....	101
9.6	Efikasnost CRC tehnike kodiranja	102
9.7	Implementacija <i>CRC</i> procesne jedinice	103
9.8	Korekcija greške na strani prijemnika.....	104
9.9	Dobit usled kodiranja	108
9.10	Vreme kašnjenja usled kodiranja.....	109
9.11	Komunikacija u bežičnim senzorskim mrežama tolerantna na višestruke greške bitova pomoću dvo-dimenzionalnih kodova grešaka	110
9.12	Dvo-dimenzionalni kodovi.....	110
9.12.1	Pravougaoni kodovi	110
9.12.2	<i>Hamming</i> produkt kodovi	113
9.12.3	<i>Hamming</i> produkt kodovi sa dodatnim bitom parnosti – (<i>SEC-DED</i>).....	113
9.13	<i>Erasure</i> kodiranje	116
9.14	Dužina kodne reči <i>2D SEC-DED</i> u odnosu na <i>Erasure</i> kodove	117
9.15	Sistematik kodovi	117
9.16	Sistematik kod za <i>2D SEC-DED</i> šemu kodiranja.....	118
10	RENDEZVOUS PROTOKOL.....	120
10.1	Izbor <i>MAC</i> protokola za aplikacije praćenja	120
10.2	Klase <i>Rendezvous</i> šema.....	122
10.3	<i>Rendezvous</i> protokol za dug životni vek senzorskog čvora (<i>Rendezvous Protocol for Long Live Sensor Node - RPLL</i>)	122
10.4	Rad u niskom faktoru ispunje korišćenjem <i>RPLL</i> protokola	123
10.5	Formiranje okvira kod <i>O-RPLL</i>	124
10.6	Prenos podataka bez grešaka i potreba za ponovnim prenosom	125
10.7	Formiranje okvira kontrolnih poruka u <i>M-RPLL</i>	126
10.8	Provera <i>SEC-DED</i> koda	127

10.9	Formiranje okvira poruke <i>DATA</i> kod <i>M-RPLL</i> protokola.....	128
10.10	Sposobnost ispravljanja grešaka <i>2D SEC-DED</i> šeme kodiranja kod <i>M-RPLL</i> protokola.....	129
10.11	Specifičnosti predloženog rešenja	131
11	ANALITIČKI MODEL PRENOSA PODATAKA	133
11.1	Efikasnost kanala kao metrika	136
11.2	Kriterijumi za selekciju šema kodiranja za ispravljanje grešaka.....	139
11.3	Vrednosti metrike <i>BER</i>	142
11.4	<i>BER</i> testeri.....	143
11.4.1	Metrike za procenu kvaliteta linka i energetske efikasnosti radio bloka	145
11.5	Energetski profil	146
11.6	Profil aktivnosti radio podsistema u toku <i>Cu-St</i> faze	147
11.7	Arhitektura Radio bloka	149
11.8	Model potrošnje energije <i>Radio</i> arhitekture	150
11.9	Efekat vremena prelaza i njegov uticaj na sistem.....	152
11.10	Potrebni komunikacioni režijski troškovi između protokola baziranih na <i>NO-FEC</i> i <i>FEC</i> tehnikama kodiranja za postizanje savršene komunikacione veze	153
11.11	Poređenje komunikacijskih režijskih troškova između <i>O-RPLL</i> i <i>M-RPLL</i> protokola za savršen link	156
12	KOMPARATIVNA ANALIZA <i>O-RPLL</i> I <i>M-RPLL</i> PROTOKOLA	160
12.1	Očekivani broj prenosa.....	160
12.2	Energija po bitu pri prenosu podataka po linku bez grešaka.....	162
12.3	Komunikacija preko linka koji nije idealan.....	164
12.4	Analiza performansi <i>O-RPLL</i> i <i>M-RPLL</i> protokola u pogledu broja ponovljenih prenosa	165
12.5	Razumevanje metrike snage kod bežičnih senzorskih mreža.....	167
12.6	Prosečna potrošnja struje Radio podsistema u pogledu uticaja T_{BP} i <i>BER</i> za fiksne vrednosti broja senzorskih elemenata i ADC rezolucije.....	169
12.7	Prosečna struja i energetska dobit <i>Radio</i> podsistema u pogledu uticaja broja senzorskih elemenata ($\#SE$) i rezolucije <i>AD</i> konvertora ($\#ADC$) kao parameter za fiksni <i>beacon</i> period T_{BP} i fiksnu vrednost <i>BER</i>	173
12.8	Softverski režijski troškovi za realizaciju <i>2D SEC-DED</i> šeme kodiranja.....	177
13	EVALUACIJA PERFORMANSI BEŽIČNOG PRENOSA PODATAKA	179
13.1	Gubici na putu i efekat senke	182
13.2	Model propagacije signala u zatvorenom prostoru.....	183
14	MATEMATIČKI MODEL ZA EVALUACIJU DOMETA PREDAJE	186

|

14.1	Metrike evaluacije: faktor oporavka pogrešnih paketa i faktor poboljšanja <i>BER</i> -a	189
14.2	<i>MATLAB</i> model za procenu dometa predaje	192
15	ZAKLJUČAK	196
16	LITERATURA	199
17	PRILOG A	205
18	PRILOG B	209

1 UVOD

Korišćenje bežičnih tehnologija poslednjih godina postaje sve izraženije u svim segmentima života. U cilju obezbeđivanja dostupnosti informacija u svakom trenutku, ostvarivanja mogućnosti upravljanja sa svakog mesta, praćenja različitih procesa aktivnosti kako samog čoveka tako i svega onoga što je vezano za njega, dovelo je do toga da ovo polje elektronike doživljava svih ovih godina stalnu ekspanziju. U okviru toga bežične senzorske mreže postale su jedna od važnih oblasti u polju umrežavanja, posebno kroz povećanje broja aplikacija. Naime, bežična senzorska mreža ima važnu ulogu u aplikacijama kao što su daljinsko nadgledanje životne sredine i praćenje određenih ciljeva. Njihova ekspanzija je omogućena zbog dostupnosti senzora koji su postali manji, jeftiniji, prilagodljiviji i inteligentniji. Senzori opremljeni bežičnim interfejsima putem kojih mogu da komuniciraju jedni sa drugima formiraju bežičnu senzorsku mrežu. Projektovanje bežične senzorske mreže značajno zavisi od aplikacije i tokom tog procesa mora da uzme u obzir brojne faktore kao što su okruženje, ciljevi aplikacionog dizajna, cena, hardver, softver i sistemska ograničenja.

Dizajn bežičnih senzorskih mreža je aplikaciono specifičan tj. promena u aplikaciji često dovodi do promene u dizajnu mreže. Takođe, mrežne topologije ponašaju se rekonfigurabilno ukoliko dođe do nekih fizičkih promena u mreži tokom njenog životnog veka. Dakle možemo da konstatujemo da je ovo jedna izazovna i kompleksna oblast koja od inženjera traži širok spektar znanja i veština.

Cilj uvodnog dela disertacije je da predstavi sveobuhvatan pregled istorije nastanka bežičnih senzorskih mreža, aplikacija koje se u realnim uslovima koriste i literature o različitim aspektima

senzorskih mreža. Probleme, izazove i kompromise sa kojima ćemo se susresti tokom ove analize razvrstaćemo u tri kategorije:

- 1) interna platforma i osnove rada sistema,
- 2) komunikacioni protokol stek, i
- 3) mrežne usluge, topologije i raspoređivanje čvorova

Da bi razumeli procese koje treba preduzeti tokom realizacije današnjih senzorskih mreža korisno je da ukratko sagledamo njihovu istoriju. Kao i kod mnogih drugih naprednih tehnologija poreklo bežičnih senzorskih mreža datira iz vojnih i primena u teškoj industriji, dakle daleko od lake industrije i potrošački orijentisanih aplikacija koje danas prevladavaju. Prvi iskorak u akademske krugove učinjen je 80-tih godina prošlog veka pre svega kroz partnerstvo sa univerzitetima da bi se ubrzo nakon toga odomaćila u akademskim i civilnim naučnim istraživanjima. Ovo je dovelo do ekspanzije u primeni bežičnih senzorskih mreža u aplikacijama kao što su praćenje kvaliteta vazduha, otkrivanje šumskih požara, sprečavanje posledica šumskih požara, sprečavanje posledica prirodnih nepogoda, poboljšana realizacija meteoroloških stanica i dr. Nakon toga inženjeri su počeli da promovišu upotrebu bežičnih senzorskih mreža u aplikacijama kao što su distribucija električne energije, prečišćavanje otpadnih voda i automatizacija procesa proizvodnje u fabrikama.

Iako je tržišna potreba za bežičnim senzorskim mrežama bila sve izraženija, prelazak sa navedenih usko specifičnih i limitiranih aplikacija u praksi se pokazao kao pravi izazov i ne tako jednostavan korak. Naime vojne, naučno-tehnološke i primene u teškoj industriji u prethodnim dekadama bile su zasnovane na velikim i skupim sensorima i uglavnom sopstvenim usko specifično razvijenim mrežnim protokolima.

Kod bežičnih senzorskih mreža u početnom periodu svog nastajanja akcenat je bio stavljen na funkcionalnosti i performanse dok su drugi faktori kao što su troškovi hardvera, standardi umrežavanja i skalabilnost bili zanemareni. Upravo je kombinacija visokih troškova i usko specifičnih primena sprečila široko rasprostranjenu primenu i usvajanje bežičnih senzorskih mreža u drugim oblastima.

Međutim nedavni napredak u tehnologiji poluprovodnika, umrežavanja i tehnologiji materijala doveo je do razmeštaja bežičnih senzorskih mreža velikih razmera. Na taj način stvorena je nova generacija bežičnih senzorskih mreža koja se u velikoj meri razlikuje od bežičnih senzorskih mreža koje su prethodno opisane. Današnje najsavremenije bežične senzorske mreže imaju znatno niže troškove instaliranja i održavanja, traju duže i robusnije su. Zastupljene su u brojnim

aplikacijama u našim domovima, na radnim mestima, donoseći nove izvore informacija, kontrole i pogodnosti kako u privatnom tako i u profesionalnom životu.

Ideja je da u budućnosti svaka aplikacija bežične senzorske mreže, koliko je to moguće, započne od već dobro ustanovljene i utvrđene standardizovane kombinacije hardversko softverskih rešenja.

Međutim ono sa čime se sada suočavaju projektanti bežičnih senzorskih sistema i mreža su razumevanje složenih kompromisa među mnogim različitim aplikacijama uključujući troškove raspoređivanja, hardver i softver, pouzdanost sistema, sigurnost i performanse. Naime, projektanti bežičnih sistema moraju da procene dobit uvedenih kompromisa i da donesu više odluka kao što su izbor tehnologije senzora i baterija, frekvencija nosioca, izlazna snaga, mrežni protokoli i dr.

Cilj svega navedenog je da se dođe do rešenja koje će omogućiti veće prisustvo bežičnih senzorskih mreža kako u industriji tako i u potrošačkoj primeni što će se postići smanjenjem cene realizacije i ugradnje, smanjenjem potrošnje energije po senzorskom čvoru i pojednostavljenjem zadataka razvoja i održavanja.

Smanjenje troškova raspoređivanja bežičnih senzorskih mreža uz povećanje funkcionalnosti podrazumeva ostvarivanje značajnog napretka u četiri ključne oblasti: senzori, *CMOS* zasnovani poluprovodnički uređaji, mrežni protokoli i tehnologije za skladištenje/generisanje energije. Tek kada se učine dodatni naponi i ostvare navedene težnje imaćemo i ostvarenje veće prisutnosti bežičnih senzorskih mreža u raznim oblastima.

U tom kontekstu cilj ove disertacije je da dâ skroman doprinos ovim naporima sagledavajući bežične senzorske mreže kroz prethodno potencirane ključne oblasti projektovanja senzorskog čvora. Posebna pažnja biće posvećena analizi potrošnje senzorskog podsistema realizovanog od različitih tipova senzora i analizi potrošnje modifikovanih verzija postojećih protokola sa stopom grešaka kao parametrom i svim benefitima koje njihova primena može da pruži.

Osnova bilo koje bežične senzorske mreže je u sensorima. Tokom prethodne dekade ubrzano su napredovale tehnologije za realizaciju senzora u vidu:

- 1) mikroelektromehaničkih sistema (*MEMS*) – žiroskop, akcelerometar, magnetometar, senzor pritiska, senzor piroelektričnih efekata, akustični senzor
- 2) senzora baziranih na *CMOS*-u – senzor temperature, vlažnosti, prisustva objekta, hemijskog sastava
- 3) *LED* senzora – senzor ambijentalne svetlosti, senzor rastojanja, senzor hemijskog sastava

Kada se kombinuju u mrežu u vidu homogenih ili heterogenih senzorskih čvorova ovako širokog spektra praćenja fenomena iz okruženja evidentno je da pružaju mogućnost razvoja novih aplikacija. Sa druge strane, napredak u domenu poluprovodničkih tehnologija obezbeđuje hardver potreban za realizaciju bežičnih senzorskih mreža. Komercijalno dostupni mikrokontroleri, koji su maltene sistemi na čipu (*SOC*) uključuju dovoljno perifernih blokova visokih performansi (pojačavače, *ADC*, *DAC*, komunikacione interfejse i dr.) i postojeane memorije putem kojih su u stanju da opsluže i neophodno procesiranje na aplikacionom nivou i mrežni protokol stek dok istovremeno obezbeđuju vezu na mrežu putem *RF*-a.

Na polju mrežnih topologija i protokola takođe je prisutan napredak u skladu sa potrebama aplikacije. Generalno topologije za mrežno povezivanje mogu da se svrstaju u sledeće četiri kategorije: jednosmerne, dvosmerne, tipa zvezda i mrežastog tipa (*mesh* topologija tj. povezan svako sa svakim).

Prvi protokoli za umrežavanje koristili su jednostavne jednosmerne komunikacione linkove koji su i dalje prisutni u nekim aplikacijama. Pošto se pojavila potreba za naprednijim topologijama kao posledica povećanja broja senzorskih čvorova, inženjeri umrežavanja razvili su protokole malih memorijskih zahteva za mrežne topologije tipa dvosmerna, zvezda i *mesh*. Takođe postoji težnja da se na protokolarnom nivou izvrši prelaz sa uskospecifičnih protokola za potrebe lokalnog umrežavanja na standardizovani protokol opšteg tipa. Slična težnja je i u drugim navedenim oblastima značajnim za realizaciju bežičnih senzorskih mreža.

Na polju baterijskih izvora energije prisutan je širok spektar rešenja koji pruža slobodu projektantima sistema da prilagođavaju izvore energije prema potrebama svojih aplikacija. Međutim, iako baterije predstavljaju jeftinu tehnologiju osnovna karakteristika im je relativno kratak životni vek pa se poslednjih godina, prikupljanjem energije iz okruženja, pokušava produžiti životni vek senzorskog čvora sa idejom ili dopune baterije ili kao praktična zamena ukoliko to aplikacija dozvoljava. Primeri za to su: generisanje snage kao posledica temperaturne razlike korišćenjem termo-električnog generatora, piroelektričnih efekata, kinetičkog kretanja piezoelektričnog materijala, fotonaponskih ćelija koje konvertuju sunčevu energiju u električnu ili direktnog konvertovanja *RF* energije u električnu putem specijalnih antena i ispravljača.

Na kraju pogled u budućnost nam ukazuje da će se, kroz napredak u tehnologijama senzora, umrežavanja, poluprovodnika i skladištenja energije, omogućiti kombinovanje bežičnih senzorskih mreža sa ciljem da se formira jedna globalna mreža. Ta globalna mreža objedinjena sa računarstvom

u oblaku i procesiranjem velike količine podataka (*big data*) dovešće do brojnih aplikacija koje će omogućiti ljudima i mašinama da interaguju na jedan potpuno nov način. Neki od brojnih primera su pružanje mogućnosti krajnjim korisnicima da kontrolišu čitav dom kroz jednostavan interfejs, povećanje efikasnosti izvora energije i sistema za distribuciju, proizvodnja veće količine hrane na poljoprivrednim imanjima i farmama i dr. Takođe, životni vek ljudi će se produžiti korišćenjem sistema za praćenje zdravstvenog stanja u kućnom okruženju kao i mnoge druge.

U daljem tekstu disertacije fokus će biti na razvoju bežičnih senzorskih mreža za potrebe komunikacija u industriji. Za razliku od uobičajenih žičanih veza kod kojih se kabliranje izvodi radi ostvarivanja komunikacije u centrima podataka ili međusobnom umrežavanju kancelarija, ključno za industrijsku komunikaciju je postavljanje kablova koji su otporni na brojne izvore agresora kao što su hemikalije, motorna ulja, vlaga, vibracije, abrazivna sredstva i dr. Kao alternativa pojavile su se ideje i praktične realizacije industrijske bežične senzorske mreže spremne da zamene većinu od navedene skupe infrastrukture sa stotinama do hiljade modularnih senzorskih čvorova. Naravno to podrazumeva sagledavanje velikog broja izazova koje treba rešiti kako bi ovu tehnologiju učinili sveprisutnom. Na primer, vrednost stope pogrešno prenetih bitova industrijskih senzorskih čvorova kreće se u opsegu od 10^{-2} do 10^{-6} , dok na primer IEEE 802.3 standardi zahtevaju performanse između 10^{-10} do 10^{-12} . Drugo, ali ne i manje važno, aplikacije koje su vremenski osetljive na umrežavanje sa potrebama za ultra niskim latencijama (kašnjenjima), kao što je automatizacija procesa proizvodnje u fabrici sa brzim robotskim rukama za pouzdanu sinhronizaciju, zahtevaju protokole za sinhronizaciju vremena na nivou ciklusa od milisekunde ili mikrosekunde.

Međutim bežična senzorska mreža je ograničena:

- ✓ količinom energije koja je dostupna od strane baterije malim senzorskim čvorovima,
- ✓ propusnim opsegom, i
- ✓ procesorskom moći

pa korišćenje tradicionalnih kompleksnih šema sinhronizacije kao što je npr. mrežni vremenski protokol (*Network Time Protocol -NTP*) nije održivo zbog navedenih ograničenja.

U ovom slučaju bežične senzorske mreže uključuju male bežične senzorske čvorove instalirane na industrijskoj opremi u cilju praćenja njenih performansi na osnovu parametara kao što su vibracije, temperatura, blizina drugoj opremi, kvalitet napajanja, pritisak i dr. Čvorovi se, u opštem smislu, sastoje od mikrokontrolera, nekoliko senzora, komunikacionih modula, memorije za skladištenje podataka i izvora napajanja. Skup čvorova čini jednu senzorsku mrežu i u zavisnosti od

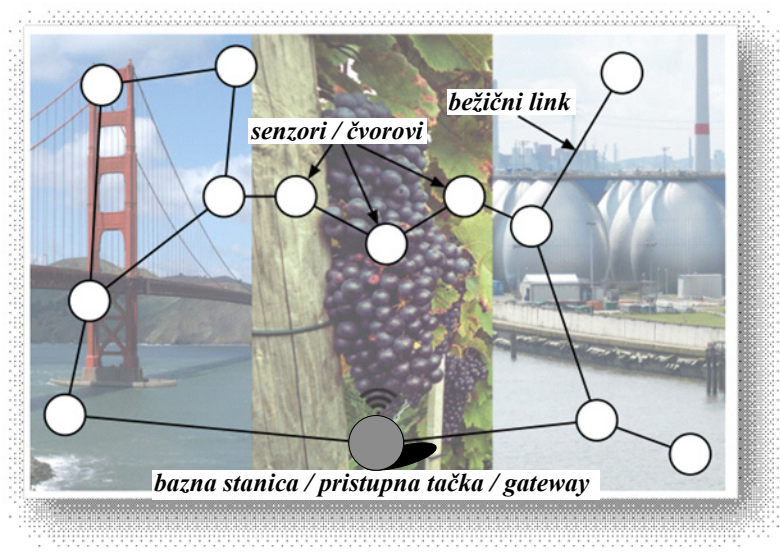
tehnologije pristupa mreža može komunicirati sa eksternim sistemima korišćenjem ili *Ethernet*, *WiFi*, *ZigBee*, *Bluetooth* ili *GPRS* veze. Takođe poznata kao senzorska mreža za prikupljanje podataka, sama mreža može da poseduje jedan čvor nazvan *gateway*, pristupna tačka ili bazna stanica koja bežičnim putem prima podatke i kanališe ih putem Interneta pomoću unutrašnje ožičene veze kao što je npr. *Ethernet*. Ovo eliminiše potrebu da svaki senzorski čvor pojedinačno čuva veliku količinu podataka i da mora da obezbedi sredstva za mrežnu podršku kako bi procesirao i analizirao podatke.

Upravo pouzdanost podataka, sigurnost i lak pristup senzorskim podacima su činjenice koje predstavljaju i najvažnije karakteristike bežične senzorske mreže. Vrlo važno za primenu senzorskih mreža u industriji je i prediktivnost tj. način na koji se deli medijum za prenos. Većina standarda za bežične senzorske mreže koristi kombinaciju *TDMA*, *FDMA* i *CSMA/CA* tehnika za deo protokola koji upravlja pristupom medijumu za prenos. Naime, glavna razlika između žičanih i bežičnih MAC protokola upravo potiče od sposobnosti da detektuju koliziju na medijumu dok se šalju podaci. Pošto to nije moguće preko bežičnog medijuma, jedino rešenje je primena analize kvaliteta usluga (*QoS*) merenjem gubitaka paketa, propusnog opsega i kašnjenja. Neki opšti zaključak je da su *CSMA/CA* protokoli uglavnom podložni bespotrebnom rasipanju energije usled kolizija i nepredvidljivih kašnjenja u komunikacijama s kraja na kraj, tako da se u danas dostupnim standardima koristi *TDMA* mehanizam za pristup kanalu.

Predložena poboljšanja nad prisutnim standardima su u domenu vremenske sinhronizacije sa kratkim vremenskim slotovima gde se sinhronizuju senzorski čvorovi i globalne mrežne operacije. Ovaj i niz drugih aspekata su tema brojnih radova na polju bežičnih senzorskih mreža. Od teorijskih, preko simulacionih do praktičnih realizacija razmatraju se tehnike, metode i protokoli kojima bi se postigao cilj iskazan kroz sintagmu "uradi svoj posao, ali malo troši". U nekima od radova akcenat je stavljen na samo jedan blok senzorskog čvora, drugi pak se fokusiraju na postizanje optimizacije i kompromisa između dva ili više blokova koji su u sadejstvu, a često imamo i oprečne zaključke.

2 BEŽIČNE SENZORSKE MREŽE

Bežična senzorska mreža (u daljem tekstu *WSN – Wirelles Sensor Network*) je zajednički pojam kojim se opisuje nezavistan skup malih računara čiji je glavni cilj da senzuju neka fizička svojstva u svom okruženju kao što su npr. vibracija, vlažnost ili temperatura. Ona se sastoji od nekoliko stotina ili hiljada senzorskih čvorova, koji se često nazivaju samo čvorovi ili samo senzori koji su međusobno povezani putem bežične komunikacije. Tipično, u okviru mreže postoji barem jedan poseban čvor, nazvan *sink*, pristupna tačka, *gateway* ili bazna stanica, koji povezuje senzorsku mrežu sa spoljašnjim svetom (videti Sl. 2-1.).



Sl. 2-1 Tipična senzorska mreža sa nekoliko senzorskih čvorova i jednom baznom stanicom *Izvor:*

Wikipedia

Senzori su međusobno povezani putem bežičnog linka, dok je bazna stanica tipično moćnija po performansama i predstavlja vezu mreže sa spoljašnjim svetom. Oblasti primene i okruženje rada su neograničeni. Postoji nekoliko pretpostavki ili opštih svojstava bežičnih senzorskih mreža koje ih čine drugačijim od drugih tipova bežičnih mreža.

Resursi pojedinačnih senzorskih čvorova su veoma ograničeni: Da bi pokrili velike površine za praćenje pojedinačni senzorski čvorovi moraju da budu jeftini. Da bi bili jeftini, posledično, njihove komponente moraju biti jeftine. Dakle, apsolutni minimum je projektovanje i korišćenje komponenti na senzorskom čvoru, tako da njihov hardver više podseća na jednostavne računare nego na savremene uređaje. Sva svojstva i ograničenja senzorskih mreža posledica su navedenog minimalnog hardverskog dizajna. Na primer, ovo je razlog zašto svaki od senzorskih čvorova ne može biti opremljen *GPS* prijemnikom ili *GPRS* antenom za komunikaciju, ili u najmanju ruku nije poželjno, već umesto toga samo jedan čvor može to sebi da priušti i on je poznat kao *sink* ili bazna stanica.

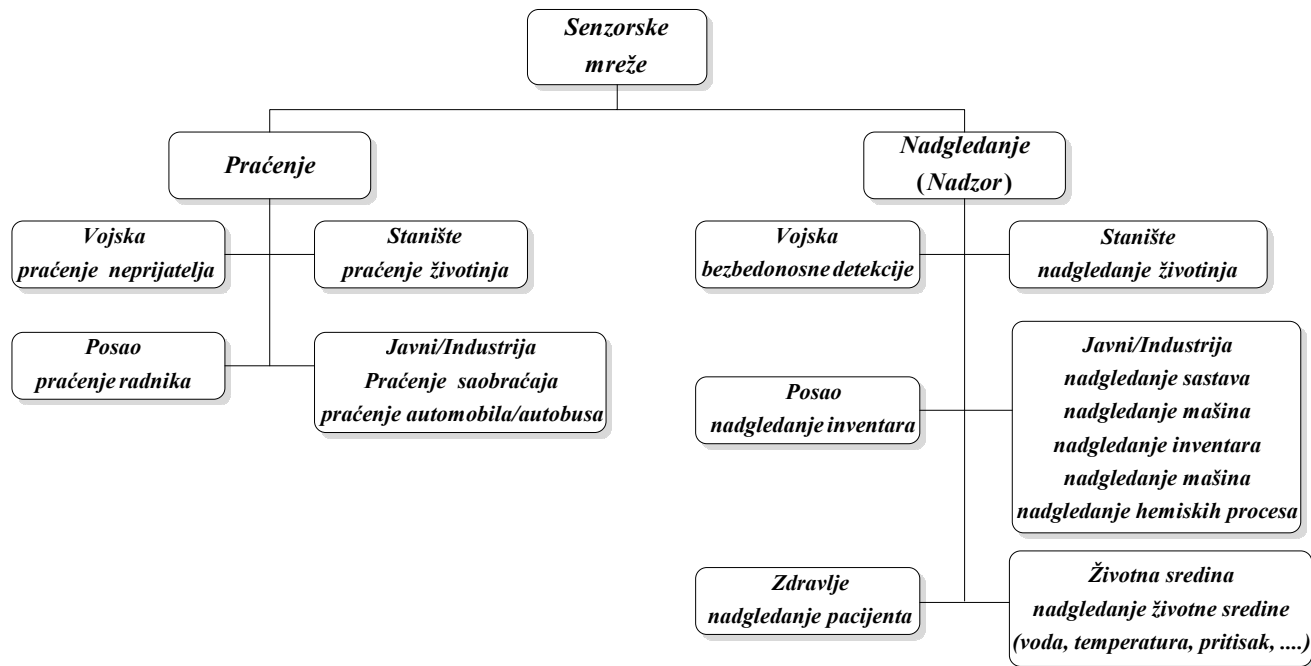
Bežične veze su spontane i nisu planske: Za razliku od drugih bežičnih mreža, kao na primer *Wi-Fi hotspots*, bežične senzorske mreže ne pripadaju grupi mreža koje su pažljivo planirane da savršeno komuniciraju i omogućе određene nivoe kvaliteta usluga. Umesto toga, polazi se od pretpostavke da će svaki senzorski čvor pokušavati da otkrije svoje srodnike i susede i da uspostavi komunikaciju kako bi razmenio neke minimalno tražene podatke sa njima. Prema tome suština je da se bežične senzorske mreže formiraju i raspoređuju brzo i bez nekog prevelikog znanja o okruženju. Postojeća iskustva sa nekim realnim bežičnim senzorskim mrežama i nekim teorijskim postavkama pomažu u postavljanju robusnije i samoodržive mreže koja je jednostavno razbacana po okruženju. Međutim, početna ideja o bacanju senzorskih čvorova iz aviona sa ciljem nadgledanja hiljade kvadratnih kilometara ipak ostaje samo san.

Senzorska mreža senzujе neke fenomene i pojave i treba da prenese podatke do spoljašnjeg korisnika (korisnika koji je izvan mreže): Uvek postoji potreba da se nešto senzujе u okruženju, kao na primer: vlažnost, temperatura, vibracije, ubrzanje, sunčevo zračenje, kiša, hemijske supstance i drugo. Glavni zadatak senzorske mreže je da senzujе neku pojavu i da prenese prikupljene informacije zainteresovanom korisniku, obično aplikaciji koja se nalazi negde izvan područja nadgledanja. Takođe treba imati u vidu da, ograničeni resursi senzorskih čvorova ne dozvoljavaju obimnu obradu informacija lokalno.

Dakle, možemo da zaključimo da su glavne funkcionalnosti senzorskog čvora senzovanje i slanje.

2.1 Klasifikacija senzorskih mreža

Klasifikacija senzorskih mreža sa aspekta primene data je na Sl. 2-2.



Sl. 2-2 Prikaz primena senzorskih mreža

Postoje pet tipova bežičnih senzorskih mreža kategorisane prema mestu gde su postavljene, pa imamo mreže koje su raspoređene na površini zemlje, pod zemljom, pod vodom, multimedijalne i mobilne mreže, što je sa više detalja prikazano u Tab.2-1.

Tab. 2-1 Tipovi bežičnih senzorskih mreža: uporedna analiza sa aspekta tipova senzora

<i>Tipovi WSN-a</i>	<i>Na površini zemlje</i>	<i>Pod zemljom</i>	<i>Pod vodom</i>	<i>Multimedijalni</i>	<i>Mobilni</i>
Primena	<i>Praćenje i senzovanje okruženja</i>	<i>Praćenje poljoprivrednog zemljišta</i>	<i>Praćenje zagađenosti</i>	<i>Unapređenje postojećih aplikacija WSN-a za praćenje i nadgledanje</i>	<i>Nadgledanje Okruženja</i>
	<i>Praćenje industrijskih postrojenja</i>	<i>Upravljanje pejzažima i parkovima</i>	<i>Nadgledanje i istraživanje podvodnog sveta</i>		<i>Nadgledanje životne okoline</i>
	<i>Istraživanje površine zemlje</i>	<i>Praćenje strukture zemljišta</i>	<i>Praćenje radi prevencije od katastrofa</i>		<i>U vojne svrhe</i>
	/	<i>Istraživanje podzemnih bogatstava, zemljišta, vode i minerala</i>	<i>Praćenje seizmičkih aktivnosti</i>		<i>Praćenje neprijatelja</i>
	/	/	<i>Nadgledanje opreme postavljene u vodi i robota</i>		<i>Pretraživanje i spašavanje nakon katastrofa</i>

2.2 Primeri aplikacija u svetu

Monitoring (nadgledanje) vinograda je jedan od najkласičnijih primera korišćenja senzorskih mreža u procesu nadgledanja. Cilj je da se smanji odnosno kontroliše proces navodnjavanja i da se predvidi ili otkrije bolest loze, što je pre moguće. Ovo ne samo da smanjuje troškove uzgajanja grožđa uz manju upotrebu vode, već omogućava i organski razvoj i rast uz nizak stepen upotrebe pesticida. Grupa senzora koja se koristi obuhvata senzor temperature vazduha, vlažnosti vazduha, sunčevog zračenja, pritiska vazduha, vlažnosti zemljišta, vlažnosti prisutnoj u listovima, ultraljubičastog zračenja, pluviometar (sensor kiše) i anemometar (sensor vetra) i dr. Senzori su obično raspoređeni na velikoj površini i dostavljaju svoje informacije spoljašnjoj bazi podataka u kojoj se podaci obrađuju posebnim modelima i algoritmima za zaštitu životne sredine. Dobijeni

rezultati se prikazuju poljoprivrednom stručnjaku ili poljoprivredniku vinogradaru, a mogu se automatski povezati i sa sistemom za navodnjavanje.

Sličan scenario senzorskih mreža koristi se i za mnoge druge primene u poljoprivredi, često nazvane precizna poljoprivreda. Primeri su nadgledanje uzgajanja krompira u Egiptu [1], nadgledanje useva u Malaviju [2] ili rešenje za nadgledanje uzgajanja povrća na farmi organske hrane u Španiji [3]. Svi ovi sistemi imaju jedan zajednički problem: lišće koje se razvija i buja tokom godine. Kada se sistemi prvi put instaliraju polja su obrađena i gotovo bez ikakvog rastinja, tj. biljke su niskog rastinja i male. Međutim kako biljke rastu njihovo lišće počinje da ometa rad sistema, posebno u pogledu komunikacije i senzovanja. Još jedan uobičajeni problem se javlja u vreme berbe, naime senzorski čvorovi su uglavnom dobro pokriveni i skriveni u usevima tako da je vrlo izvesno da će biti oštećeni od strane mašina za berbu. Njihovo ponovno korišćenje predstavlja pravi izazov i ukoliko se ne obnove, mreža postaje neupotrebljiva.

Nadgledanje mostova je slična primena kojom se prati stanje strukture mosta. Opet kao i u prethodnom slučaju prostor je ograničen čak iako je kvalitet komunikacija bolji zbog otvorenog prostora, tj. prenosa signala u slobodnom prostoru. Postoje dva poznata primera mostova koja su nadgledana senzorskim mrežama. Prvi primer je posledica tragedije. Naime, nakon rušenja mosta preko reke Misisipi u Mineapolisu, u procesu ponovne izgradnje most je bio opremljen stotinama senzora za nadgledanje njegovog stanja i signaliziranje ranih upozorenja o nekim nedostacima.

Drugi primer predstavlja most sa šest traka u Grčkoj (*Chariolas Trikoupis*) koji je postavljen nad Korintskim zalivom. Otvoren je za saobraćaj 2003. godine sa sistemom za nadgledanje koji se sastoji od više od 300 senzorskih čvorova opremljenih 3D akcelerometrima, meračima nagiba (*tilt* sensorima), sensorima naprezanja i mnogim drugim. Neposredno nakon otvaranja, senzorska mreža je signalizirala neuobičajene (abnormalne) vibracije konstrukcionih kablova što je navelo inženjere da postave dodatne tegove za stabilizaciju. Nakon toga most nije imao nikakvih dodatnih problema.

Rano otkrivanje požara je presudno za spašavanje života i sprečavanje oštećenja na objektima. Senzorske mreže mogu efikasno da se iskoriste za rano otkrivanje požara i slanje alarma vatrogasnim jedinicima o tačnoj lokaciji nastanka vatre. Ova ideja je realizovana i koristi se širom sveta u zemljama kao što su Španija, Grčka, Australija i Turska. Glavni izazov sa ovom primenom predstavlja sama oprema senzorskog čvora tj. hardverska struktura i njena otpornost na visoke temperature. Sasvim je neupotrebljiva ako mreža senzora, projektovana da detektuje požar, zakaže tokom nešto vrelijeg dana ili sa prvim varnicama vatre. Iz ovih razloga suštinski je bitno da se

projektuju mreže koje su dodatno obezbeđene robustnim hardverom i pameću. Čak šta više, pošto treba da pokriju relativno velika prostranstva, može da se pojavi problem u procesima instaliranja mreže, održavanja i ostvarivanja komunikacije.

Nadgledanje tunela je još jedna oblast u kojoj se koriste senzorske mreže. Tuneli na putevima su opasnost u celom svetu. Ako se nešto desi kao što je saobraćajna nesreća ili požar, od suštinske važnosti je znati koliko je još ljudi u tunelu i gde se oni tačno nalaze. Čak i u uobičajenim radnim uslovima tuneli su sami po sebi opasnost, kao na primer, kada dnevno svetlo na izlasku iz tunela često zaslepi vozača. Projekat TRITon2 u Trentu u Italiji je primer kako bi potencijalno mogli rešiti ovaj problem upotrebom senzorskih mreža. Naime, senzori svetlosti se koriste da adaptivno regulišu svetlost unutar tunela čime postižu dva cilja, smanjuju potrošnju energije za potrebe osvetljenja tunela i umanjuju efekat zaslepljivanja vozača na izlazu iz tunela. Video kamere se koriste da automatski detektuju automobile i prate ih sve dok ne napuste tunel za potrebe spasilačkih operacija. Jedan od glavnih izazova nadgledanja tunela je posledica oblika tunela jer se bežični signali značajno rasipaju u takvom uskom okruženju i komunikacija postaje ometana (opstruirana) i nepouzdana. Pored toga, nema mnogo prostora da se postavi više čvorova i da se na taj način mreža učini robusnijom. Takođe samo postavljanje novih čvorova podrazumeva da je tunel zatvoren za saobraćaj tokom tih aktivnosti što je opet problem.

Nadgledanje životinja je jedna od najstarijih primena senzorskih mreža. Ipak to nisu tipične senzorske mreže jer je glavna pojava koju one prate kretanje životinja. Dakle one funkcionišu više kao aplikacije za praćenje implementirane sa tipičnim senzorskim čvorovima. Primer je farma u Australiji gde se koriste senzorske mreže u procesu uzgoja stoke [4]. Umesto postavljanja električnih ili drugih tipova ograda na velikim površinama, goveda su opremljena ogrlicama koje poseduju sistem pozicioniranja sa zvučnicima i elektro-šokom. Web interfejs omogućuje iscertavanje ograda na virtuelnoj mapi i identifikaciju gde bi stoka trebala da bude, a gde ne. U slučaju da krava pokuša da pređe granicu tog predviđenog "zatvorenog" prostora ona prvo dobije zvučno upozorenje. Ako to ne da rezultat i krava i dalje ostane u zabranjenom području, daje se blagi elektro šok (sličan onom koji se dobija od električnih ograda). Na ovaj način krave se uče da zapamte da nakon zvučnog signala sledi električni šok i da brzo reaguju i napuste zabranjeno područje.

Pored toga praćenje kretanje goveda može da se iskoristi za procenu njihovih vitalnih sposobnosti. Ostali senzori mogu čak da se iskoriste, recimo za praćenje temperature što pruža

moгуćnost uvida u njihovo zdravstveno stanje. Ovo je naročito važno za farme goveda koje su rasprostranjene na velikim površinama kao što su one koje se nalaze u Australiji ili Južnoj Americi.

Prepoznavanje životinja je još jedna oblast nadgledanja životinja. Jedna od prvih primena bežičnih senzorskih mreža razvijena je u Australiji sa ciljem da pronalazi žabe krastače. Populacija žaba krastača se, nakon naseljavanja Australije sa ciljem uništavanja sivih buba, otrgla kontroli pa su lokalni rendžeri bili primorani da ih ubijaju, ali ih je bilo teško pronaći. Iz tog razloga su uvedene senzorske mreže čija se jedinstvenost ogleda u njenom senzoru. On bukvalno sluša zvukove u šumi sa ciljem da prepozna specifično kreketaanje žabe. Nakon što ostvari prepoznavanje, mreža odgovara događaju tako što generiše i šalje, zajedno sa koordinatama lokacije žabe, informaciju rendžerima koji sada mogu fizički da ih pronađu.

Tehnički gledano ovakvu senzorsku mrežu nije trivijalno implementirati. Ona zahteva lokalno audio procesiranje kako bi identifikovala kreketaanje žabe što dovodi do povećanja hardverskih i softverskih resursa u samom čvoru. Ceo sistem je opisan tek nekoliko godina nakon postavljanja mreže od strane *Hu et all.* [5].

Dakle nakon svih navedenih primera, neka globalna konstatacija bi bila da postoji mnogo različitih aplikacija bežičnih senzorskih mreža širom sveta što onemogućava pokrivanje svih njihovih područja primene u ovoj disertaciji. Zainteresovani čitaoci trebalo bi jednostavno da pretražuju Internet za dalje informacije o primenama i primerima bežičnih senzorskih mreža pošto je većina njih dobro opisana.

2.3 Tipovi bežičnih senzorskih mreža

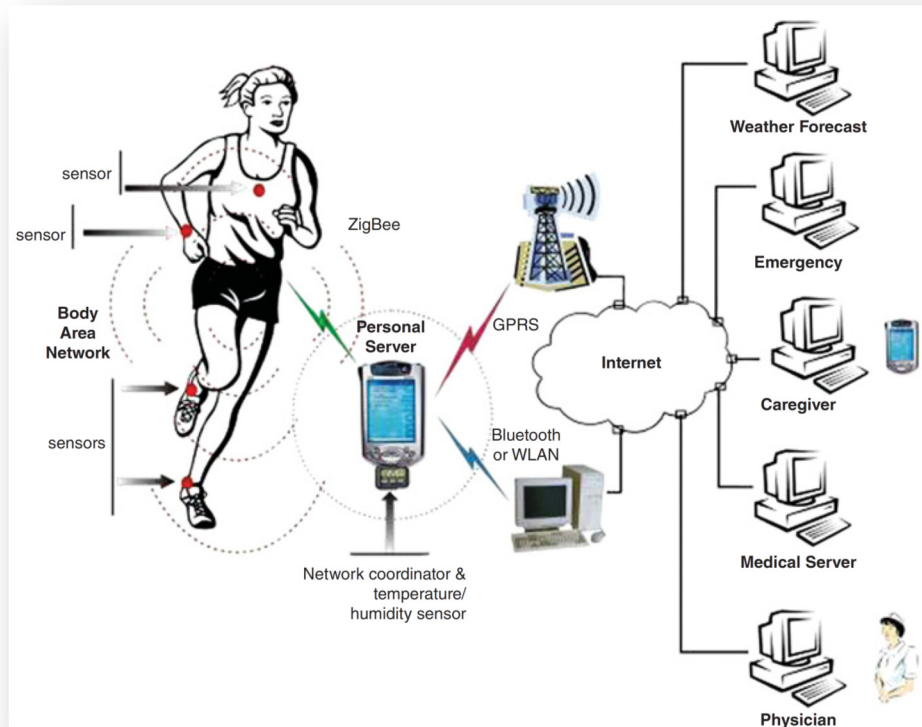
Poslednjih godina su se pojavile različite nove definicije i žargoni u oblasti senzorskih mreža. Ovaj odeljak disertacije ih sa više detalja i sadržajnije opisuje.

Bežična senzorska mreža je i dalje jedan od glavnih pojmova za opis mreže koja se karakteriše međusobnom saradnjom i koordinacijom malih računara koji su u stanju da senzuju neke pojave i fenomene u okruženju i pošalju svoje rezultate centralnom entitetu kao što je baza podataka ili server. Pojam se često koristi za instalacije koje rade na tradicionalan način senzuj i šalji. Međutim, pojam može da se sagleda i u širem kontekstu, zavisno od toga ko koristi termin i gde.

Visokotehnoški-fizički sistem (CPS – Cyber-physical system) je novi pojam za bežičnu senzorsku mrežu. On pokušava da bolje opiše šta zapravo možemo učiniti sa ovim mrežama i njihovim glavnim svojstvima kada se integrišu u fizičko okruženje. Za razliku od drugih računara i

uređaja koji su bez znanja (agnostički) o okruženju odnosno i ne sagledavaju ga, *CPS* su deo okruženja i ograničenih aplikacija. Još jedna važna karakteristika je činjenica da oni mogu da utiču na životnu sredinu putem tzv. aktuatora kao što su oni u automatskim pumpama za navodnjavanje, prekidačima za svetlo, alarmima i regulatorima vlažnosti i temperature. Interesantno je da se pojam bežično više ne koristi u *CPS*-u. Razlog je u tome što su sada mnoge senzorske mreže ožičene pošto je napajanje u zgradama lakše realizovati kroz žičanu vezu nego bežično. Pametne kuće su tipičan primer *CPS* sistema kao i pametne mreže (*smart grids*) i pametni gradovi (*smart cities*).

Mreže senzora na telu se odnose na određenu vrstu mreže koja je projektovana da se nosi na telu (uglavnom se misli na ljudsko telo). Aplikacije obuhvataju nadgledanje zdravstvenog stanja, praćenje i kontrolu težine, vođenje dnevnih aktivnosti i mnoge druge. Postoje i neki neobični primeri poput pametnih cipela ili pametnih majica koji su u stanju da senzuju vaše aktivnosti ili broj otkucaja srca. Većina senzorskih čvorova su mali ili sićušni, ponekad čak i implantabilni. Tekući trend kreće se ka kreiranju jedinstvenog integrisanog uređaja koji će biti u stanju da senzujе sve funkcije, umesto nekoliko različitih senzora na različitim delovima tela kakva je trenutna situacija. Takođe cela senzorska mreža često se svodi na jedan ili nekoliko uređaja pa sam pojam mreža pomalo deluje besmisleno. Na Sl. 2-3 prikazan je primer mreže sa nekoliko senzora na telu.



Sl. 2-3 Mreža senzora tela se sastoji od relativno malog broja senzora na ili u telu čoveka. Izvor:

Wikipedia

Participativno senzovanje, kolaborativno senzovanje ili *crowdsourcing* odnosi se na novi i brzo rastući tip senzovanja u kojem su senzori u osnovi ljudi sa svojim pametnim telefonima. Na primer, ljudi mogu da prate biciklističke staze kojima se kreću, a zatim izvrše procenu u pogledu sigurnosti, buke ili kvaliteta puta. Svi navedeni podaci se prikupljaju u centralnom sistemu i obrađuju u jedinstvenu mapu kvaliteta staza u jednom gradu, koja može da se distribuira svakom zainteresovanom korisniku [6]. Prava vrednost ovih aplikacija je da nije potreban dodatni hardver, već samo prilično jednostavna korisnička aplikacija za pametne telefone. Istovremeno, ovo predstavlja i najveći izazov u ovim aplikacijama, a to je motivisati ljude da se pridruže procesima senzovanja i da dostave visokokvalitetne podatke centralnom sistemu. Privatnost je veliki problem sa ovim aplikacijama zato što zahtevaju od ljudi da dele podatke o svojoj lokaciji.

Internet stvari – (*IoT – Internet of Things*) se često pogrešno smatraju senzorskim mrežama. Glavni koncept *IoT* je da su sve stvari i uređaji, kao što su mašina za pranje veša ili, na primer, radio, povezane sa *Internetom*. *Internet* konekcija ima značajne prednosti kada implementiramo senzorske mreže tako da se može okarakterisati kao obećavajuća tehnologija.

Međutim, cilj može da bude veoma različit, na primer da obezbedi čitanje mail-ova sa mikrotalasne peći ili iz automobila. Termin *Internet* takođe nagoveštava da su obično ove mreže opskrbljene *IP* adresom i samim tim da koriste dobro definisani komunikacioni *stack*. Ovo se može tretirati kao prednost (nema potrebe za ponovnom implementacijom) ili kao nedostatak (velika potrošnja energije, mala fleksibilnost).

Slike Sl. 2-2 i Sl. 2-3 prikazuju neke od prethodno navedenih tipova mreža, visokotehnoloških fizičkih sistema za aplikacije pametna kuća i mrežno polje na telu za nadgledanje zdravstvenog stanja. Zajednička karakteristika svih ovih aplikacija i tipova mreža je da je sistem visoko integrisan sa svojim odgovarajućim okruženjem i ne može se lako prenositi ili generalizovati.

Bez obzira da li koristimo pojam kao na primer bežične senzorske mreže, senzorske mreže, mreže na telu ili visokotehnološki fizički sistem – najvažnije teme prisutne u svim ovim situacijama su senzovanje, komunikacija, upravljanje potrošnjom energijom i raspoređivanje (instaliranje mreže).

Bežične senzorske mreže su savremena tehnologija koja pokriva opšte aplikacije nadgledanja prostora. Obično se sastoje od mnogo (nekoliko do više hiljada) uređaja koji omogućavaju nadgledanje prostora velikih razmera. Glavni zahtevi koji se postavljaju pred mrežama su da budu malih dimenzija i niske cene. Dakle, umesto implementacije složenih aplikacija kroz samu mrežu, senzorska mreža primenjuje dve glavne operacije: senzovanje i komunikaciju senzovanih podataka između čvorova i ka centralnom serveru.

Kao što je već pomenuti, aplikacije bežičnih senzorskih mreža obuhvataju nadgledanje (monitoring) životne sredine i životinja, fabričkih i industrijskih procesa, poljoprivrednih procesa i automatizacije u poljoprivredi, praćenje zdravstvenog stanja i mnoga druga područja. Jedna od glavnih karakteristika bežičnih senzorskih mreža je da su čvrsto povezane sa oblašću primene. Naime, sistem koji je razvijen za nadgledanje poljoprivrednih procesa na jednom mestu teško se može koristiti na nekom drugom području, a gotovo je nemoguće koristiti ga u, na primer, nadgledanju životinja.

3 ANATOMIJA SENZORSKOG ČVORA

Ovaj odeljak istražuje generalni koncept arhitekture senzorskih čvorova i neke opšte primere realizacije. Postoji veliki izbor senzorskih čvorova u opsegu od sićušnih implantabilnih sa namenom za ljudsko telo do pametnih telefona i tableta. Cilj ovog sažetog prikaza je da nam pruži uvid u glavne razlike između senzorskih čvorova i tradicionalnih *IT* sistema.

Bežične senzorske mreže se sastoje od velikog broja prostorno raspoređenih i baterijski napajanih senzorskih čvorova koji imaju mogućnost da senzuju fizičko okruženje, primene potrebna izračunavanja nad dobijenim podacima i komuniciraju korišćenjem radio interfejsa [7]. Senzorski čvor u okviru bežične senzorske mreže je tipično opremljen senzorom (pretvaračem), radio-primopredajnikom, mikrokontrolerskom jedinicom i izvorom napajanja (obično baterija). Senzor (pretvarač) konvertuje senzovane fizičke efekte i fenomene u električne signale. Mikrokontrolerska jedinica obrađuje i čuva senzovane podatke. Radio-primopredajnik je odgovoran za prijem i slanje paketa podataka drugim senzorskim čvorovima ili pristupnoj tački. Senzorski čvor sadrži bateriju veoma malog kapaciteta, kao jedinicu napajanja, i pošto se rasporedi u svom radnom okruženju ona se više ne može puniti ili zameniti, tj. česta zamena baterije nije praktična zbog nepristupačnog i često neuslovnog okruženja kao i velikog broja čvorova u okviru mreže [8].

Životni vek cele mreže zavisi od toga koliko se efikasno koristi životni vek baterije svakog senzorskog čvora pojedinačno. Da bi se racionalno iskoristila dostupna energija baterije mora da se posegne za nekim od algoritama koji su u stanju da uspešno realizuju upravljanje napajanjem koje se svodi na uvođenje nekih senzorskih čvorova u stanje spavanja, u kojem bi trošili vrlo malo energije,

dok preostali čvorovi ostaju u aktivnom režimu, što znači da izvršavaju senzovanje, obradu i prenos podataka [9].

Senzorski čvorovi u okviru bežičnih senzorskih mreža su sklони greškama i otkazima usled gubitka energije baterije, hardverskih otkaza, grešaka u komunikacionim vezama, zlonamernih ataka i oštećenja i tako dalje. Generalno, mreža ima sledeće karakteristike [10]:

- 1) Dve komponente čvora, senzor (pretvarač) i bežični primopredajnik direktno interaguju sa okruženjem koje je predmet različitih fizičkih, hemijskih i bioloških faktora. Pored toga, greške u komunikaciji između čvorova ili čvora i pristupne tačke se javljaju kao posledica interference šuma, slabljenja signala usled fedinga, postojanja različitih prepreka, vremenskih uslova i dr.;
- 2) Koncept koji uvodi senzorski čvor u režim rada spavanja, kako bi se uštedela energija, izaziva dodatne probleme vezane za korektnu vremensku sinhronizaciju i malo komunikaciono kašnjenje;
- 3) Većina aplikacija bežičnih senzorskih mreža ima postavljena ograničenja u realnom vremenu što znači da prijemni čvor treba da bude obavešten pre isteka krajnjeg roka (*deadline*) nakon što se događaj dogodi (*event*);
- 4) Topologija mreže je dinamična

Dakle, sva ova pitanja doprinose unapređenju dizajna pouzdanih mehanizama otpornih na greške u bežičnim senzorskim mrežama [11-14].

Tri glavne aktivnosti na koje senzorski čvor troši energiju su senzovanje, obrada podataka i komunikacija. Za svaku od navedenih aktivnosti koju preduzima, senzorski čvor troši različitu količinu energije. Da bi se smanjila potrošnja energije u senzorskom čvoru kao jedino moguće rešenje nameće se korišćenje kombinacije različitih tehnika za smanjenje potrošnje. Međutim, sve ove tehnike imaju svoje prednosti i nedostatke primenjivali ih pojedinačno ili u kombinaciji. [15-16]

Istraživanja su pokazala da čak i pri kratkom vremenu trajanja operacije radio prenosa (vreme uključenog radio primopredajnika ili faktora popune, *duty-cycle*, oko 1–2%) energija potrebna za realizaciju tog radio prenosa je za oko red veličine veća od energije potrebne za realizaciju svih drugih operacija zajedno [14]. Studije istraživanja takođe pokazuje da je energija potrebna za prenos jednog bita informacije približno ista kao i energija neophodna za izvršenje hiljadu operacija u procesoru tipičnog senzorskog čvora [17]. Ove činjenice nameću potrebu da razvijemo efikasnu

metodu uštede energije koja podrazumeva ograničenje vremena komuniciranja i količine podataka koji se oprenose ili bar smanjenje ovih veličina kod postojećih rešenja.

Sa druge strane, komunikacijske veze između senzorskih čvorova kod bežičnih senzorskih mreža su veoma nestabilne. Dakle, cilj je da se ugradi funkcija tolerancije na greške u bežičnoj mreži, posebno u slučaju kada dođe do greške tokom prenosa podataka po linku. Da bi se obezbedila pouzdana komunikacija sa podacima obično se koriste dve tehnike, jedna je *Automatic Repeat reQuest (ARQ)*, a druga je *Forward Error Correction (FEC)*.

U velikom broju praktičnih primena *FEC* šeme kodiranja obuhvataju *erasure* i *Hamming* kodove. Važna činjenica koju treba uzeti u obzir u odabiru šema *FEC* kodiranja u radio frekventnim komunikacijama je šema grešaka koje se javljaju tokom prenosa. Dok šum obično uzrokuje pojavu slučajnih grešaka, pojava *fading*-a i interference proizvodi pojavu greške *burst* tipa. Naše opservacije, koje su u skladu sa [18] pokazuju da su većina pogrešnih bitova jednostruke ili dvostruke prirode, dok se ređe događaju greške tipa višestruke ili *burst*.

Sva nevedena redundansa sa druge strane ima svoju cenu i dovodi do dodatne potrošnje energije u procesu kodiranja, prenosa redundantnih bitova i procesa dekodiranja. Imajući u vidu da su bežične senzorske mreže energetske ograničene i da zahtevaju pouzdanu komunikaciju podataka, ključni faktor u fazi projektovanja je izbor optimalnog koda za korekciju grešaka koji podrazumeva ostvarivanje visokih performansi (visoka komunikaciona propusnost) i male potrošnje energije.

4 OPIS PROBLEMA I PREGLED TEKUĆEG STANJA

Korišćenje bežičnih senzorskih mreža za kontrolu procesa u industriji privuklo je veliku pažnju poslednjih godina, videti [19-22]. Osnovni razlog za to je činjenica da se *WSN*-ovi mogu lako i efikasno rasporediti po već postojećoj infrastrukturi bez značajnih intervencija, tj. senzori se mogu postaviti i tamo gde žice ne mogu, a samim tim i cena kabliranja i instalacije se u značajnoj meri smanjuje. Međutim, implementacija bežične kontrole sa baterijski pobuđivanim senzorskim čvorovima malog kapaciteta energije zahteva ekstremnu energetska efikasnost. U procesnoj industriji, fabrike često imaju hiljade kontrolnih petlji. Kao primer, razmotrimo industrijski pogon sa četiri hiljade baterijski napajanih senzorskih čvorova. Ako je životni vek baterije unofornno distribuiran između jedne i dve godine, onda će u jednom ustaljenom stanju, u proseku, biti potrebna zamena baterija na oko deset čvorova dnevno. Očigledno ovo nije prihvatljiva opcija u većini situacija. Dakle, iz ove kratke analize nije baš jasno da li je upotreba baterijski napajanih senzorskih čvorova prihvatljiva za procesnu kontrolu i ako jeste pod kojim uslovima je prihvatljiva.

Da bi postigli produžetak životnog veka senzorskog čvora i donekle poboljšali situaciju, jedno od mogućih rešenja je da podsisteme senzorskog čvora (blokove) postavimo u režim rada spavanja koliko god je to moguće često. Međutim ovo je u suprotnosti sa zahtevom za malo kašnjenje u mreži i upotrebu tehnike prekomernog uzimanja uzoraka koji se često pojavljuju u procesnoj industriji. Stoga, evidentno je da je vrlo važno da efikasno koristimo vreme uključivanja pojedinih podsistema i da optimalno izaberemo interval odmeravanja i tehnike predaje.

Dakle, možemo da zaključimo da, između ostalih tehnika, aktivnosti i metoda, odabir šema za kontrolu grešaka koje su energetska efikasne i sa malim kašnjenjem igra važnu ulogu. Kontrola

grešaka se generalno može realizovati pomoću *ARQ* metode, koja se bazira se na zahtevu za ponovnim slanjem poruke, *FEC* metode, koja se bazira se na korekciji grešaka na određitu ili pomoću kombinacije pomenute dve metode koja je poznata kao hibridna *ARQ* metoda. Danas, *ARQ* metoda koja se čisto zasniva na retransmisiji, u velikoj meri je zastupljena u bežičnim senzorskim čvorovima, a primenjena je i u mrežama za kontrolu procesa (*HART Communication Foundation, 2007*) [23]. Iako bazirana na retransmisiji, uslovljenoj gubitkom paketa, poboljšava propusnost, ali uporedo sa tim uvodi i latenciju i prekomernu potrošnju energije, što je često neprihvatljivo posebno u vremensko kritičnim aplikacijama kakve su zastupljene u kontroli procesa proizvodnje.

4.1 Opis problema

FEC tehnika podrazumeva uvođenje redundantnih bitova koji, zbog povećanja dužine paketa, povećavaju i potrošnju energije i latenciju. Sa druge strane, njenom primenom smanjuju se retransmisije, što je pozitivan efekat njene primene. Dakle, neophodno je napraviti kompromis između stope koda i retransmisije. Ova disertacija upravo ima za cilj da, kroz analizu iz svih uglova utvrdi kako bi trebalo da se učini kompromis u industrijskim primenama i gde su prednosti i nedostaci primene *ARQ* i *FEC* tehnike u smislu energetske efikasnosti i latencije. Predložena kodna šema zasnovana je na *FEC* tehnici sa dvodimenzionalnom tehnikom kodiranja baziranoj na *Hamming*-ovom kodiranju. Ovakve šeme su od ranije poznate i korišćenje u memorijama i u računarstvu u oblaku (*cloud computing*) videti na primer [24-27].

Performanse 2D SEC-DED tehnike kodiranja bazirane na *FEC*-u će biti istražene u pogledu potrošnje energije, kašnjenja i gubitaka paketa i upoređene sa predstavnicima šema kodiranja *ARQ* tehnike. Istraživanje i zaključci koji će ovde biti prezentovani zasnivaju se na merenjima, proračunima i iscrpnim simulacijama sprovedenim korišćenjem Matlab softverskog alata.

4.2 Svrha ove teze

Svrha ove teze je da prikaže detaljan opis predložene kodne šeme i mogućnosti njene primene u bežičnim senzorskim mrežama, da pruži bolje razumevanje opisanog kompromisa između stope koda i retransmisije kojim se opravdava njena primena u određenim parametrima kanala i pruži razumevanja koja će se preneti na industrijsku primenu i u nekoj kasnijoj fazi razvoja postati deo standarda bežičnih senzorskih mreža za kontrolu procesa.

4.3 Povezani radovi

U [25] su proučavali energetska efikasnost specifičnih kodova za kontrolu grešaka (*ECC - Error Control Codes*), za nekoliko implementacija dekodera u *WSN*-ovima. Oni zaključuju da razmatrana kodiranja štede energiju čak i na veoma kratkim rastojanjima, ali nisu uzimali u obzir povećane veličine paketa, što dovodi do dužeg vremena prenosa, a samim tim i povećanje potrošnje energije. Razmatrali su samo energiju koja se koristi kod procesa kodiranja i dekodiranja.

U [28] se razmatraju različite tehnike kontrole grešaka, a akcenat je stavljen na određivanje optimalne veličine paketa u bežičnim senzorskim mrežama. U [27] je data procena potrošnje električne energije za tri različita ECC-a, ali za specifične platforme, npr. FPGA i ASIC. U [24] je predstavljena analiza šema kontrole grešaka u bežičnim senzorskim mrežama, koja upoređuje *ARQ*, *FEC* i *HARQ* u smislu potrošnje energije, latencije i stope pogrešno prenetih paketa (*PER*). Videti takođe i referencu [7].

U [29] i [30], razmatrana je samo cena dekodera uključena na strani prijemnika, a ne i ukupni troškovi obrade, koji se povećavaju sa povećanjem veličine paketa. Osim toga, njihova studija je bazirana na konvolucionim kodovima, dok se mi fokusiramo na blok kodove.

4.4 Teorijska osnova kontrole grešaka

U kontroli grešaka baziranoj na *ARQ* metodi, paket se retransmituje ako se utvrdi da ima grešaka. Takvi paketi se ponovo prenose sve dok se ne prime bez grešaka. Detekcija (otkrivanje) grešaka se obično implementira kroz proveru ciklične redundanse (*Cycle Redundance Check - CRC*). Jednostavni kod za detekciju greške se primenjuje na paket pre prenosa, a na prijemnoj strani će se izračunati kontrolna suma (*checksum*) kako bi se proverilo da nije došlo do greške. Ako suma provere ne dovede do prave vrednosti, paket se ponovo šalje (retransmituje). *ARQ* šema koristi pozitivnu potvrdu (*ACK*) ili negativnu potvrdu (*NACK*) kako bi poslala povratne informacije predajniku kojim ga obaveštava da li je prenos bio uspešan ili ne, respektivno. Ako se koristi *ACK* pristup, onda će predajnik ponoviti prenos prethodno poslatog paketa i ukoliko ne dobije bilo kakvu potvrdu u obliku paketne potvrde u okviru predefinisano vremenskog intervala. Ovo čini i glavni nedostatak *ARQ* metode jer se javljaju dodatni troškovi usled retransmisije kako u vidu potrošnje energije, tako i u vidu kašnjenja uz već povećanju potrošnju usled uvođenja redundantnih bitova kao posledice primenjenije tehnike kodiranja.

4.5 Korekcija grešaka na odredištu

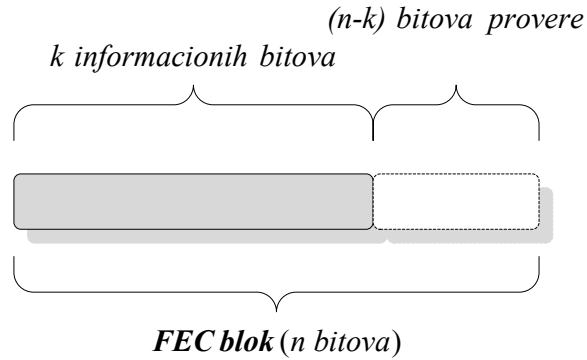
FEC ili kanalno kodiranje je metoda koja se koristi za povećanje performansi kontrole grešaka. Ovo se postiže korišćenjem kodova za korekciju tj. ispravljanje grešaka (*Error Correcting Codes - ECC*) koji dodaju redundantnost paketu što omogućava da se izvestan broj pogrešno prenetih bitova detektuju, a neki od njih i koriguju na strani prijemnika. Glavni nedostatak je redundantnost, kodni bitovi koji povećavaju veličinu paketa. Dodatno, *FEC* uvodi i aktivnosti oko procesa kodiranja i dekodiranja. Prema tome, neki opšti zaključak bi bio da se ovaj metod za kontrolu grešaka tradicionalno koristi u okolnostima i okruženju gde su retransmisije relativno skupe. Postoji mnogo tipova *FEC* kodova koji mogu da se svrstaju u dve kategorije: blok kodovi i konvolucionalni kodovi.

Blok kodovi najpre dele poruku koja treba da se prenese u manje blokove unapred određene dužine. Zatim se tako dobijeni blokovi pojedinačno kodiraju generišući kodne reči. U slučaju konvolucionalnih kodova svaki bit se, za razliku od prethodnog pristupa, kodira kao funkcija unapred definisanog broja prethodnih bitova tj. prati predistoriju bitova informacije što je možda i glavni nedostatak u primeni ovih kodova kod resursno i performansno ograničenih senzorskih čvorova.

Reed-Solomon, *Bose Chadhuri Hockuenghem (BCH)* i *Hamming* kodovi su među najpoznatijim blok kodovima koji su našli veliku primenu u mnogim oblastima elektronike, čuvanja podataka (memorija) i komunikacija. *Hamming*-ov kod poznat je kao jedan od pionirskih kodova koji je 1950. godine razvio *Richard Hamming* i danas se još koristi u aplikacijama kao što su kodovi za korekciju grešaka u memorijama (*ECC memory*). Tokom godina razvijeni su moćniji kodovi kao što su *Reed-Solomon* i *BCH* kodovi.

Blok kodovi su obično predstavljeni trostrukim vektorom u oznaci (n, k, t) gde je n - dužina kodne reči, k - broj informacionih bitova u kodnoj reči i t - je sposobnost korekcije u smislu broja bitova koji se mogu ispraviti. Detaljnije o ovoj problematici pozabavićemo se u kasnijim odeljcima disertacije.

Sl. 4-1 prikazuje kako dodatni bitovi parnosti povećavaju dužinu *FEC* bloka. Infomacioni bitovi dužine k zajedno sa bitovima parnosti čine kodnu reč dužine n , iz čega proističe jedna od najbitnijih karakteristika koda nazvana stopa koda i definisana kao $R_c = k/n$. Stoga, za datu stopu koda R_c i poruku sa m informacionih bitova, kodirani paket bi se sastojao od m/R_c bitova.

Sl. 4-1 Ilustracija *FEC* bloka

Pored tradicionalnih blok i konvolucionih kodova postoje još moćniji kodovi kao što su turbo kodovi i *LDPC* (*Low Density Parity Check*) kodovi. Zbog njihove složenosti i naprednih procesa kodiranja i dekodiranja, ograničeni su na aplikacije koje mogu da opsluže potrebnu kompleksnost proračunavanja.

Kao što je poznato *BCH* kodovi [24] predstavljaju klasu cikličnih lineranih blok kodova. To su moćni kodovi koji, za umereno visok odnos signal-šum *SNR*, uglavnom nadmašuju sve ostale kodove iz grupe blok kodova sa visokom stopom koda (*code rate*).

4.6 Kontrola grešaka u bežičnim senzorskim mrežama

Prilikom dizajniranja šema kodiranja za kontrolu grešaka kod bežičnih senzorskih mreža kritičan parametar i dominantan faktor je energetska efikasnost. Zbog toga dodatni troškovi povećanja potrošnje energije koji nastaju kao posledica *FEC* kodiranja moraju biti upoređeni sa onim što bi moglo da se uštedi na ime smanjenja retransmisije (ostvarena dobit). Glavni razlog povećanja utroška energije koji treba detaljno analizirati je proširenje dužine paketa što dovodi do dužeg radio prenosa. Postoji i dodatni trošak usled procesa kodiranja i dekodiranja. Ovi troškovi, a naročito troškovi dekodiranja pominju se u [24], a u vezi su sa merenjima na *SA-1100* procesoru. Uzimajući u razmatranje procesor male potrošnje, kao što je *MSP430* [31] pretpostavlja se da je "potrošnja energije procesiranja algoritma kodiranja i dekodiranja zanemarljiva u odnosu na druge troškove obrade". Ovo je potvrđeno i rezultatima od strane [25].

Dizajn kontrole grešaka zasnovan na *FEC*-u će imati prednost u tom smislu što može da ispravi određeni broj grešaka u paketu. U suštini, prednost njegove primene je niža stopa grešaka u paketima, označena kao *PER* (*Packet Error Rate*) u poređenju sa *ARQ* tehnikom, ali po cenu povećanja broja bitova u paketu. Ako to razmotrimo na polju mreža sa jednim skokom (*single hop*)

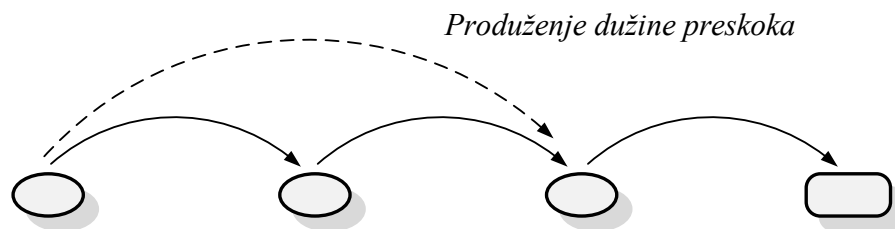
ili više skokova (*multi-hop*) niža vrednost *PER*-a može da se iskoristi na sledeća tri načina čiji opis sledi:

4.6.1 Izbegavanje retransmisije

Niža vrednost *PER*-a postignuta primenom *FEC*-a jednostavno se može koristiti da bi se izbegla retransmisija. Kada se paket prenosi preko kanala sa lošim karakteristikama, *PER* bi se drastično povećavao u slučaju da se primenjuje jednostavna *ARQ* šema što će za posledicu imati čestu retransmisiju. Cena koja se plaća usled ponovnog prenosa u smislu potrošnje energije i kašnjenja mogli bi se izbeći u određenoj meri ako bi se implementirala odgovarajuća šema *FEC*.

4.6.2 Povećanje broja preskoka

U [7] predložena je tehnika nazvana povećanje dužine preskoka, pri čemu se niža vrednost *PER*-a može iskoristiti tako što će se povećati rastojanje u toku jednog preskoka u *multi-hop* mreži u odnosu na standardnu dužinu preskoka. Ovo bi imalo za posledicu da je potreban mali broj preskoka da bi poslati paket došao do cilja. Za uzvrat manji broj preskoka bi doveo do povećanja energetske efikasnosti i smanjenja kašnjenja (niže latencije). Ilustracija tehnike produžetka dužine preskoka prikazan je na Sl. 4-2.



Sl. 4-2 Ilustracija tehnike povećanje dužine preskoka, korišćenjem *FEC* tehnike isprekidanom strelicom ilustrovano je izbegavanje jednog dodatnog preskoka

4.6.3 Upravljanje predajnom snagom

Razmotrimo sistem projektovan sa ciljanom stopom pogrešnih bitova - *Bit Error Rate (BER)*. Primenom *FEC* tehnike predajna snaga može biti smanjena u odnosu na sistem koji koristi *ARQ* metodu, a da i dalje postiže istu stopu *BER*-a. Na ovaj način može da se ograniči potrošnja energije. U teoriji kodiranja ovo je opisano kao dobit kodiranja (*coding gain*) i predstavlja razliku između nivoa odnosa signal-šum (*SNR*) koji su potrebni da bi se postigao isti ciljani *PER*, sa i bez primene tehnike *FEC*-a.

4.7 Gubici na putu

Gubici na putu (ili slabljenje na putu) predstavljaju smanjenje gustine snage elektromagnetnog talasa prilikom njegovog propagiranja kroz bežični radio kanal. Gubici na putu su posledica mnogih efekata kao što su konture terena, različita okruženja (unutrašnji zatvoreni ili spoljašnji otvoreni prostor, u urbanim ili ruralnim područjima), kao i udaljenost između predajnika i prijemnika. Postoji nekoliko modela koji se koriste da aproksimiraju propagaciju signala kao što su na primer model gubitaka na putu u slobodnom prostoru (*free-space path loss*), *ray-tracing propagation*, *log-distance path loss model* i drugi.

Kao što je pomenuto u [32] model *log-distance path loss* je pojednostavljen model koji se obično koristi za opštu kompromisnu analizu različitih dizajna sistema. Teško je napraviti model koji precizno opisuje složenost specifične propagacije signala, ali bez pribegavanju komplikovanim modelima koji opisuju gubitke na putu, model *log-distance path loss* obuhvata suštinu propagacije signala i dat je izrazom:

$$P_r = P_t \cdot K \cdot \left(\frac{d_o}{d} \right)^\gamma \quad (4.1)$$

gde je K - konstanta koja zavisi od karakteristike antene i slabljenja kanala, a d_o - predstavlja referentnu distancu, najčešće je to distanca na tzv. prvom metru. Eksponent gubitaka na putu γ - se određuje na osnovu okruženja kroz koji se elektromagnetni talas prostire. Tipične verednosti za γ u fabričkom okruženju kreće se u opsegu od 1.6 do 3.3 [32].

4.8 Verovatnoća greške na bit nivou i na nivou paketa

Verovatnoća grešaka na bit nivou

Kada propagira kroz bežični kanal signal je podložan poremećajima tj. negativnom uticaju usled prisustva šuma i pojave interference, što može da prouzrokuje promenu vrednosti primljenih bitova. Netačnosti i ograničenja u predajniku ili prijemniku kao što je, na primer, loša sinhronizacija mogu da izazovu isti efekat. Stopa pogrešnih bitova (*BER*) pokazuje koliko je primljenih bitova izmenjeno u toku prenosa u odnosu na ukupan broj prenetih bitova.

Verovatnoća greške na bit nivou se može teorijski proceniti za različite kombinacije kanala i digitalnih modulacionih šema. U slučaju *Additive White Gaussian Noise* - *AWGN* kanala gde je

implementirana *BPSK* - *Binary Phase-Shift Keying* modulaciona šema uz pretpostavku da se radi o savršenoj sinhronizaciji tačan izraz verovatnoće greške na bit nivou je dat kao:

$$p_b = Q\left(\sqrt{\frac{2 \cdot E_b}{N_0}}\right) \quad (4.2)$$

gde je Q skaliran oblik komplementarne *Gauss*-ove funkcije greške

Verovatnoća greške na nivou bita za kvadraturnu faznu modulaciju (*QPSK*) je ista kao i za *BPSK* pošto se *QPSK* modulirani signal može posmatrati kao signal sa *BPSK* modulacijom kako u faznoj tako i u kvadraturnoj komponenti signala. *Offset QPSK* (*OQPSK*) ima istu teorijsku performansu greške na nivou bita kao i *BPSK* i *QPSK* (Za detaljniji prikaz pogledati referencu [33]).

Verovatnoća greške na nivou paketa

Za nekodirani prenos paketa podataka dužine m bitova, verovatnoća grešaka na nivou paketa može da se izračuna kao:

$$p_p = 1 - (1 - p_b)^m \quad (4.3)$$

gde je p_b verovatnoća greške na nivou bita, u ovom slučaju za pomenute *BPSK* i *QPSK* data izrazom (4.2).

4.9 Verovatnoća greške na nivou kodne reči

U slučaju da se prenos podataka vrši sa implementiranom šemom kontrole grešaka koja koristi *FEC* kodove, prethodna jednačina može da se izrazi kao verovatnoća greške na nivou kodne reči

$$p_p = 1 - (1 - p_b)^w \quad (4.4)$$

gde je w broj kodnih reči u paketu. Verovatnoća greške na nivou kodnih reči može, pak, biti izražena kao funkcija verovatnoće greške na nivou bita. Kao što je pomenuto u [34], za specifičan slučaj lineranog binarnog blok koda sa karakteristikama (n, k, t) , dekođer koji koristi *hard-decision* će biti u stanju da koriguje bilo koji broj grešaka do

$$t = \left\lfloor \frac{1}{2} \cdot (d_{\min} - 1) \right\rfloor \quad (4.5)$$

ovo je donja granica sposobnosti korekcije (t) pošto je funkcija minimalne *Hamming*-ove distance (d_{\min}) između bilo koje dve različite kodne reči. Zbog toga, kodna reč može biti korigovana

čak i sa više od t grešaka. Čak štaviše, prema binominalnoj teoremi, verovatnoća da postoji ε grešaka u kodnoj reči od n bitova je

$$p(\varepsilon, n) = \binom{n}{\varepsilon} p_b^\varepsilon \cdot (1 - p_b)^{n-\varepsilon} \quad (4.6)$$

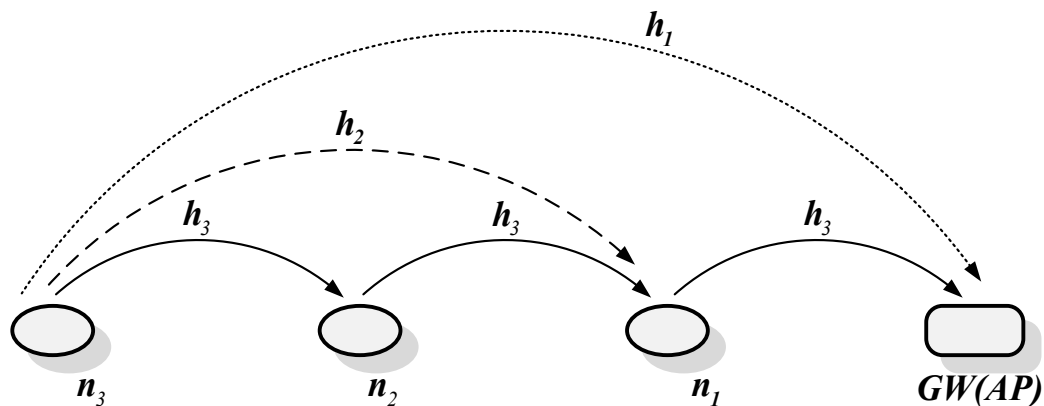
Najzad, gornja granica verovatnoće greške na nivou kodne reči je data kao

$$p_w \leq \sum_{\varepsilon=t+1}^n \binom{n}{\varepsilon} p_b^\varepsilon \cdot (1 - p_b)^{n-\varepsilon} \quad (4.7)$$

što predstavlja sumu verovatnoća za sav nekorigovan (neispravljenj) broj grešaka $\varepsilon > t$.

4.10 Model sistema

U aplikacijama kontrole procesa zahtevi koji se postavljaju u pogledu kašnjenja (latencije) često ograničavaju broj preskoka koji se mogu izvršiti između senzorskih čvorova i pristupne tačke (*access point* ili deljivog *gateway*-a (*GW*)). Uvedimo jednostavan mrežni model koji se sastoji od tri senzorska čvora (n_1, n_2, n_3) koji se nalaze na jednakom rastojanju i mogu biti povezani na *GW*, vidi Sl. 4-3. Pretpostavljamo da svaki čvor koji može preneti pakete i time omogućiti višestruke preskoke, na nivou mreže, poseduje sposobnost ponovnog kodiranja paketa. Analizirajmo situaciju da se paketi šalju do *GW* od najudaljenijeg čvora (n_3) kroz jednostruki skok ili višestruki skok. Postoje tri različite rute (putanje) do čvora *GW*: preko h_1 , preko h_2 i h_3 , ili preko tri preskoka tipa h_3 .



Sl. 4-3 Mrežni model, podatak sa čvora n_3 može da dostigne do *gateway* putem jednostrukog ili višestrukog skoka

Napomena: *GW* - Gateway; *AP* - Acces Point

Da bi pojednostavili analizu, pretpostavićemo da se informacije o rutiranju kao i informacije o tome koje ćemo kodove koristiti postoje u svim čvorovima. U praksi to bi se moglo postići upotrebom centralizovane šeme gde je *GW* odgovoran za kontrolu grešaka, uključujući i rutiranja. Dakle, *GW* bi trebalo da ima potreban *CSI* (*Channel State Information*) u smislu odnosa signal-šum (*SNR*) koji je definisan kao E_b/N_0 iz prethodne mrežne aktivnosti, donosi odluke o tome kojom rutom i kodovima će se koristiti, a takođe i da održi mrežu ažuriranom sa navedenim informacijama tako što će periodično preneti svima sinhronizacioni paket. Drugim rečima, na osnovu *CSI* *GW* će odlučiti o ruti i kodovima za prenos, a takođe i ove informacije preneti na odgovarajuće čvorove. Napomenimo još jednom da je ovo pojednostavljen model koji bi trebao dodatno da se ispita, ali i ovakav i dalje je relevantan i daje važne uvide u to kako koristiti različite *FEC* šeme.

5 VRSTE MEHANIZAMA ZA KONTROLU GREŠAKA

Bežične senzorske mreže zahtevaju jednostavne i proste šeme za kontrolu grešaka zbog zahteva senzorskog čvora za niskim nivoom složenosti. Kontrola grešaka se uobičajeno koristi za otkrivanje i eventualno ispravljanje grešaka u prenosu podataka u okviru bežične senzorske mreže.

U principu, *FEC* kodne šeme se uobičajeno koriste u okolnostima u kojima su ponovna slanja relativno česta što ima za posledicu značajno uvećanje potrošnje energije, tj. prenos se vrši preko linka sa relativno visokom stopom grešaka (reda veličine 10^{-5} ili više). Tokom proteklog perioda glavni cilj u oblasti kodiranja kanala bio je pronalaženje efikasne i praktične šeme kodiranja kojom bi mogli da se približimo teorijskom limitu kapaciteta kanala (takođe poznat kao *Shannon-ov* limit) koji se modelira kao *AWGN* (*Additive White Gaussian Noise*) kanal.

U tu svrhu jasno se izdvajaju dva alternativna pristupa. Prvi se bavi pronalaženjem koda sa dobrim algebarskim osobinama kao što je velika vrednost minimalne *Hamming-ove* distance ili rastojanja. Glavni razlog za postizanje ovog cilja je da maksimiziramo sposobnost korekcije grešaka (takva vrsta kodiranja se naziva algebarsko kodiranje). Kodovi kao što su *Hamming-ov* kod, *Golay* kod, *Reed Solomon* kodovi, *Erasure* kodovi i *Bose-Chadhuri-Horquenghem* (*BCH*) kodovi su glavni predstavnici ovog tipa kodiranja.

Druga, alternativna linija istraživanja i razvoja bila je direktno inspirisana *Shannon-ovim* probabilističkim pristupom kodiranju. Probabilističko kodiranje više se bavi pronalaženjem prosečnih performansi kao funkcija složenosti kodiranja i dekodiranja. U ovu kategoriju spadaju konvolucioni kodovi, kodovi konkatencije (kao što su *Turbo* kodovi), polarni kodovi i *Low-density-Parity-Check*

(LDPC) kodovi. Za detaljniju istoriju nastanka teorije kodiranja kanala i aplikacija pogledati referencu [35].

Korišćenjem ponovnog slanja *ARQ* metoda pruža pouzdanu komunikaciju. Ovakava pristup je energetska neefikasan u tzv. "lošim" kanalima koji dovode do česte pojave retransmisije. *FEC* se ponaša bolje u "lošim" kanalima, dok u situaciji kada su uslovi kanala dobri postojanje redundantnih bitova postaje nepoželjan energetska trošak. Ovo ukazuje na to, da tamo gde je to moguće, kombinacija dve navedene tehnike može delovati efikasnije nego bilo koja od njih pojedinačno. Ova kombinacija *ARQ* i *FEC* šema se naziva *HARK*.

U ovom slučaju *FEC* se koristi za ispravljanje malog broja grešaka (koje se uobičajeno javljaju često) dok se *ARQ* koristi da se zatraži retransmisija (ponovno slanje) u slučaju da *FEC* ne uspe da se izbori sa pojavom većeg broja grešaka koje se, inače, ređe javljaju. Postoje dva oblika *HARK* tehnike, *HARK-I* i *HARK-II*. Za više detalja o *HARK* može se pogledati literatura [36-37]. Generalno, *HARK* je dobar za neka scenarija u bežičnim senzorskim mrežama [24], ali je primenljiv samo na neke specifične aplikacije i troši značajnu količinu energije [38].

5.1 Kada je FEC koristan?

Kao što smo već istakli, energetska efikasnost je od presudnog značaja pri izradi šeme za kontrolu grešaka kod bežičnih senzorskih mreža. Imajući ovo na umu, postojanje dodatnih troškova energije kao posledica uključivanja *FEC* tehnike kodiranja mora da bude upoređeno sa onim što bi moglo da se sačuva kao posledica manjeg broja ili potpunog eliminisanja potrebe za ponovnim slanjem.

Glavni segment povećanja troškova energije je usled dodavanja ekstra bitova odnosno povećavanja dužine paketa što dovodi do povećanja vremena prenosa bitova tj. *Radio-Subsystem* je duže vreme uključen. Postoji i dodatna potrošnja energije usled nešto složenijeg procesa kodiranja i dekodiranja na izvorišnom i odredišnom čvoru, respektivno.

Ono što sigurno predstavlja dobit je da se primenom *ECC* kodova greške smanjuju. Ovakav pristup dovodi do poboljšanja vrednosti *BER*-a nakon dekodiranja, u odnosu na situaciju na linku, pri datom odnosu signal/šum (*Signal-to-Noise Ratio* - *SNR*). Ovo znači da je moguće smanjiti snagu predaje za dati *BER* ponekad po cenu smanjenja propusnog opsega kao i dodatne kompleksnosti procesa kodiranja/dekodiranja na *RF* primopredajniku. U opštem slučaju važi da ako je snaga potrebna za implementaciju *ECC*-a veća od snage koja se štedi usled povećanja dobiti kodiranja

(*coding gain*) nema smisla primenjivati *ECC* odnosno njena primena nije energetski isplativa. Postoji mnogo radova koji obrađuju kompromis između dodatnih troškova (*overhead*) i postignute dobiti na strani potrošnje energije za različite tipove *ECC*-a. U principu, praktična primena kodova bi trebala da ima nisku kompleksnost procesa kodiranja/dekodiranja. Za detaljnije informacije o primeni *ECC* kodova u bežičnim senzorskim mrežama, sa posebnim osvrtom na to kako *ECC* obezbeđuje dobit kodiranja (*coding gain*) što dovodi do uštede energije na strani predajnika po ceni dodatne potrošnje energije usled uvođenja dodatnog dekodera treba pogledati literaturu [39-40].

5.2 Vrste kodova za ispravljanje grešaka

U praksi, skoro svi *ECC* kodovi su linearni. Uzimajući u obzir razliku u strukturi i obradi (procesiranju) informacionih sekvenci, kodovi se mogu podeliti na blok kodove i konvolucione kodove [36-37]. Linearni blok kodovi kodiraju podatke u blokove (poruke) koji su fiksne dužine. Enkoder transformiše poruku od k bitova u veći blok od n bitova poznat kao kodna reč, označen kao $[n, k, d_{\min}]$ sa stopom koda $R = k/n$. Pojam (oznaka) $d_{\min} = n - k + 1$ predstavlja minimalni broj bitova na kojima se razlikuju dve kodne reči. *Hamming*, *Golay*, *Reed-Solomon* i *BCH* kodovi su primeri poznatih blok kodova [36-37]. Pored najčešće korišćenih blok kodova koji su prethodno navedeni, postoji i po karakteristikama znatno moćniji kod poznat kao *Low Density Parity Check (LDPC)* kod, koji takođe spada u klasu blok kodova [40].

Konvolucioni kod primenjuje sledeći algoritam, uzima niz bitova podataka i konvertuje ih u niz bitova koja predaje (prošireni niz bitova) koristeći banku pomeračkih registara kao memorijske registre. Konvolucioni enkoder uzima k -tuple od po m_i bitova poruke kao ulaz i generiše tuple c_i kodnih bitova kao izlaz u datom trenutku i , koji ne zavise samo od ulaznih k -tuple od po m_i bitova poruke u datom trenutku i nego i od prethodnih k -tuple od po m_j prisutnih u trenutku $j < i$. Za više detalja pogledati [37].

5.3 Izbor odgovarajućih kandidata kodova za ispravljanje grešaka

Sa ciljem da izaberemo odgovarajućeg i pogodnog kandidata za *ECC* kod koji ćemo primeniti u aplikacijama bežičnih senzorskih mreža iz obe grupe kodova (blok kodovi i konvolucioni kodovi), neophodno je najpre da ukratko ukažemo na njihove glavne prednosti i nedostatke.

LDPC kodovi, *Turbo* kodovi, konvolucioni kodovi i *Polar* kodovi, kao predstavnici tzv. moćnih kodova, dobro su poznati po svojim izvanrednim performansama koje se odlikavaju u

postizanju izuzetno niskog *BER*-a čak i u situaciji kada je *SNR* izuzetno nizak. Generalno, moćni kodovi obezbeđuju visoku dobit kodiranja (*coding gain*), ali, zbog svoje složenosti, koriste kompleksnije dekodere koji troše puno energije. Na primer, *LDPC* kod se obično dekodira iterativnim procesom. Glavni nedostatak *LDPC* koda, gledano sa aspekta aplikacija male potrošnje, su veliki broj iteracija u procesu dekodiranja, velika latencija, veliki broj jedinica za računanje što sve dovodi do velike potrošnje energije. Konvolucionni kodovi se dekodiraju po principu *trellis*-a (rešetke) ili korišćenjem *Viterbi* dekodiranja, ili *Maximum A Posteriori (MAP)* dekodiranja ili sekvencijalnog dekodiranja. Utvrđeno je da primena konvolucionih kodova zahteva previše kompleksne dekodere za tipične bežične primopredajnike [27], a samim tim ih isključuju iz užeg kruga izbora.

Kompleksnost (složenost) softverske implementacije za različite vrste snažnih kodova za industrijske bežične senzorske mreže proučavana je u literaturi [41]. Autori u pomenutom članku smatraju da su *LDPC* i *Turbo* kodovi suviše složeni za bežične senzorske mreže jer zahtevaju previše memorije i vremena za dekodiranje. U skladu sa prethodno navedenim postavlja se veliki znak pitanja da li snažni kodovi odgovaraju standardnim zahtevima bežičnih senzorskih mreža. Odnosno, za trenutnu *VLSI* tehnologiju, snažni kodovi nisu odgovarajući kandidati za energetska ograničenja koja su postavljena pred bežičnim senzorskim čvorom.

Kao alternativa predlaže se korišćenje nekog od jednostavnih kodova kao što su *Hamming*-ov kod, *Golay* kod, *Reed-Solomon* kod, *BCH* kod ili *Erasure* kod koji, iako pružaju nižu dobit kodiranja (*coding gain*), zahtevaju manje složene dekodere, a samim tim i nižu potrošnju energije. *Hamming*-ovi kodovi se dekodiraju pomoću sindroma ili dekodrom maksimalne verovatnoće. Algebarski kodovi kao što su *Reed-Solomon* kodovi i *BCH* kodovi se dekodiraju složenim rešavanjem polinoma u cilju određivanja lokacije greške. *Erasure* kodovi se dekodiraju relativno jednostavnim matematičkim manipulacijama [42]. Dakle iz prethodno navedenog kratkog pregleda možemo da zaključimo da su jednostavniji kodovi prihvatljiviji izbor za bežične senzorske mreže.

5.4 Tipovi modulacionih šema i njihov značaj za prenos signala

Klasičan pristup smanjenju potrošnje energije kod senzorskog čvora je smanjenje vremena prenosa podataka putem *RF* primopredajnika s obzirom da je komunikacioni segment najveći potrošač energije. Modulacioni metod konstelacije može se prilagoditi da poveća ili smanji broj bitova u simbolu u toku predaje jednog simbola (korišćenjem *M-ary* modulacije). Kada se broj bitova u simbolu poveća da bi se detekcija signala ostvarila uspešno neophodno je da se postigne veći *SNR*.

Drugim rečima, kada se bitska brzina povećava trajanje prenosa po bitu je kraće, što može da dovede do uštede energije pod uslovom da je prenos dovoljno pouzdan. Za slučaj da SNR nije dovoljno visok i ne zadovoljava potrebe odnosa signal-šum na određištu, potrebno je ponovo preneti veliki broj pogrešno primljenih paketa, što dovodi do energetske neefikasnosti.

Napomenimo da upotreba M -ary modulacije rezultira visokom kompleksnošću kola RF prijemnika i nižoj energetskej efikasnosti izlaznog stepena primopredajnika. U procesu bežične komunikacije senzorskog čvora odabir adekvatne modulacione šeme i ECC tehnike kodiranja igra važnu ulogu u postizanju boljeg očuvanja energije [43]. U literaturi su prisutne brojne analize performansi različitih modulacionih tehnika u kombinaciji sa različitim metodama kodiranja i apostrofirane prednosti i nedostaci. Tako na primer, u radu [30] iskazan je uslov kada M -ary modulaciona tehnika postiže manju potrošnju energije u poređenju sa arhitekturom binarne modulacije. U [27] su proučavane i analizirane performanse binarnog BCH i *Reed-Solomon* kodova na osnovu BER performanse, prenošenjem slučajno generisanih podataka korišćenjem $BPSK$ modulacije i $AWGN$ kanala. Autori su zaključili da binarni BCH kodovi u $ASIC$ implementaciji mogu da predstavljaju odgovarajući izbor za bežične senzorske mreže. U [44] autori ukazuju da $GMSK$ modulaciona šema sa *Golay* strategijom kodiranja pod uslovom $AWGN$ kanala postiže značajan dobitak, poboljšava pouzdanost linka i smanjuje energiju signala predajnika u bežičnim senzorskim mrežama. Aktivnosti prezentovane u [45] se fokusiraju na domet predaje, snagu prenosa i optimizaciju nivoa modulacije sa aspekta energetske efikasnosti. Rezultati pokazuju da je izuzetno važno da prilagodimo parametre u skladu sa uslovima kanala i željenog komunikacionog dometa kako bi se postigla minimalna potrošnja energije po bitu u tipičnom primopredajniku bežične senzorske mreže.

5.5 Model kanala

U skladu sa prethodnim navodima, kritično pitanje koje se postavlja, a vezano je za fizički sloj u prenosu podataka kod bežičnih senzorskih mreža, je izbor modulacione koding šeme u kontekstu ograničenih energetske sposobnosti senzorskog čvora. Generalno, ovaj izbor zavisi od karakteristika kanala, odnosno od modela kanala. Većina objavljenih radova [35], [24,38], [25,27] i [46] analizu vrši tako što kanal modeluje *Gauss*-ovim pristupom tj. kao $AWGN$ kanal. $AWGN$ (*Additive White Gaussian Noise*) je osnovni model šuma koji se koristi u teoriji informacija kojim se imitira efekat mnogih slučajnih procesa koji se javljaju u prirodi. Specifične karakteristike iskazane

su modifikatorima: *Additive* zato što se dodaje svaki šum koji može biti utisnut u informacioni sistem; *White* odnosi si se na to da je uniformno raspoređen kroz ceo frekvencijski opseg informacionog sistema. *Gaussian* zato što ima normalnu raspodelu u vremenskom domenu sa srednjom vrednošću vremena oko nule.

Ovakav kanal se karakteriše konstantnim odnosom signal-šum i modulaciona koding šema koja ga pokriva fokusirana je na smanjenje prosečne snage za definisanu pouzdanost prenosa. U ovom pristupu, sa aspekta apstrakcije signala, *AWGN* se često koristi kao pogodan model šuma. Naime, *AWGN* kanal prihvata ulaznu sekvencu realnih brojeva kao svoj ulaz i dodaje beli *Gauss*-ov šum kako bi formirao novu sekvencu realnih brojeva kao svoj izlaz. Beli *Gauss*-ov šum je diskretni slučajni process za koji su komponente nezavisne i identično raspoređene *Gauss*-ove slučajne promenljive. Međutim sa tačke gledišta apstrakcije bita model *Gauss*-ovog šuma nije odgovarajući, jer je neophodno da se vodi računa o nivou signala i da se uoči kako tačno utiče na svaki bit pojedinačno.

Prihvatljiviji model na bit nivou je *Binary Symmetric Channel (BSC)*. *BSC* ima dva karaktera kao ulazne promenljive, na primer 0 i 1 . Verovatnoća greške je $P_Y(Y = I | X = 0) = p_b$. Činjenica da je kanal simetričan znači da je verovatnoća da će se učiniti greška jednaka, nezavisno od toga da li je ulazni bit (simbol) jedan ili nula $P_Y(0|1) = P_Y(1|0) = p_b$. Verovatnoća da bit prolazi nepromenjen je komplementaran događaj onom kada se bit promeni, tako da verovatnoća korektnog bita postaje $P_Y(0|0) = P_Y(1|1) = 1 - p_b$.

To znači da je ukupna verovatnoća u oznaci P_{total} za *BSC* kanal [47]

$$\begin{aligned} P_{total} &= P_X(0) \cdot P_Y(1|0) + P_X(1) \cdot P_Y(0|1) \\ &= P_e \cdot P_Y(1|0) + (1 - P_e) \cdot P_Y(0|1) \end{aligned} \quad (5.1)$$

gde je verovatnoća 0 na ulazu kanala $P_X(0) = P_e$, a verovatnoća 1 na ulazu kanala $P_X(1) = 1 - P_e$, respektivno

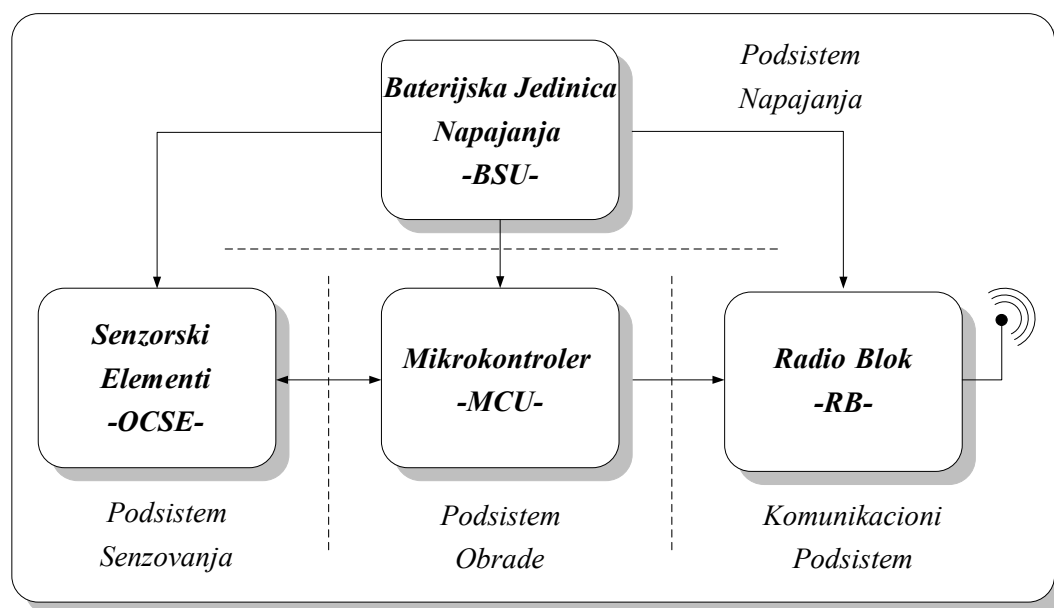
U *BSC* modelu kanala, svaki bit trpi promene nezavisno od ostalih bitova, a verovatnoća oštećenja je ista za sve bitove tako da je proces šuma primer nezavisnog i identično raspoređenog slučajnog procesa. Kao što je već pomenuto, prenos putem bežičnog kanala u industrijskom okruženju je podložan interferenciji što dovodi do čestih pogrešno isporučenih paketa. Kao posledica odnos signal poremećaji postaje slučajna promenljiva što dovodi do optimalne dodele snage. Ovakav

pristup sadrži izbor, zasnovan na merenju kanala, minimalne snage koja bi trebalo da se prenese sa ciljem kompenzacije efekata kanala i time garantuje određeni nivo uspešne isporuke paketa. Takav pristup je zasnovan na *BSC* modelu kanala.

S obzirom da potpuna pokrivenost ove problematike izlazi iz okvira doktorske disertacije, u preostalom delu disertacije fokusiraćemo pažnju na to kako predložena tehnika kodiranja odgovara prenosu u industrijskim bežičnim senzorskim mrežama.

6 STRUKTURA SENZORSKOG ČVORA

Celokupna hardverska struktura senzorskog čvora je prikazana na Sl. 6-1.



Sl. 6-1 Celokupna hardverska struktura senzorskog čvora na blok nivou

Senzorski čvor se sastoji od nekoliko gradivnih blokova koji će biti detaljnije objašnjeni u daljem tekstu:

a) Mikrokontroler (*MCU*) - u našem rešenju označen kao podsystem za obradu, kontroliše rad svih sastavnih podistema unutar senzorskog čvora i vrši obradu podataka. Mikrokontrolerski podsystem obuhvata mikrokontroler i memoriju za lokalnu obradu podataka. Većina postojećih praktičnih realizacija ovog bloka podrazumeva korišćenje sledećih mikrokontrolera *MSP430 Texas Instruments*,

Strong ARM Intel ili *AVR Atmel*. Izbor zavisi od velikog broja faktora proizašlih iz oblasti primene senzorskog čvora. Ovi mikrokontroleri, po svojoj arhitekturi, omogućavaju da neke od svojih unutrašnjih komponenti ili periferija potpuno isključe kada su u stanju mirovanja ili spavanja.

b) Senzorski elementi izvan čipa (*OSCE*) - koji se zove podsistem senzovanja, implementiran kao skup pasivnih i aktivnih senzora (analogni ili digitalni) koji konvertuju ulazne informacije iz spoljašnjeg okruženja u električne signale. U većini aplikacija bežični senzorski čvorovi se koriste za praćenje parametara iz okruženja kao to su svetlost, pritisak, vibracije, protok u cevima, temperatura, ventilacija, električna energija itd. Obično senzorski elementi generišu naponske ili strujne signale na svojim izlazima koji se najpre pojačavaju (kondicioniraju), a zatim se digitalizuju pomoću analogno-digitalnog konvertora nakon čega se obrađuju, memorišu i predaju.

c) Radio blok (*RB*) - implementiran je kao primopredajnik kratkog dometa koji senzorskom čvoru obezbeđuje bežičnu komunikaciju sa pristupnom tačkom ili drugim senzorskim čvorom unutar bežične senzorske mreže. U pogledu potrošnje električne energije ovaj blok predstavlja dominantnog potrošača u okviru senzorskog čvora i kao takav predmet je mnogih pokušaja smanjenja potrošnje i to na:

- 1) nivou kola razvijanjem energetski efikasnih *RF* kola, i
- 2) na sistemskom nivou korišćenjem *RF* komunikacije (ogledaju se skraćivanju dometa komunikacije, minimiziranju količine podataka koji se šalje putem *RF* veze, korišćenjem energetski-efikasnih komunikacionih protokola ili isključivanjem primopredajnika tokom neaktivnih perioda tj. korišćenjem *duty-cycling* koncepta). Za više detalja o ovom problemu pogledajte referencu [48].

d) Jedinica baterijskog napajanja (*PSU*) - deo je podsistema za napajanje koji je odgovoran za obezbeđivanje odgovarajuće vrednosti napona napajanja za svaku pojedinačnu komponentu hardvera u senzorskom čvoru koristeći bateriju kao izvor energije. Takođe, deluje kao kontrolna jedinica koja pojedinačno, po potrebi, uključuje/isključuje napajanje pojedinih ili svih blokova u okviru senzorskog čvora. Ovako koncipiran *PSU* predstavlja veoma važan gradivni blok senzorskog čvora jer, primenom odgovarajućih tehnika zasnovanih na efikasnoj eksploataciji energetskog resursa, produžava životni vek baterije senzorskog čvora, a samim tim i celokupne senzorske mreže. Samu bateriju karakteriše nekoliko nedostataka koji proizilaze iz samog načina funkcionisanja, a mogu da se iskažu kroz sledeće: imaju potrebu da se periodično pune ili/i zamene; velike su i teške u poređenju sa elektronikom (veličina direktno srazmerna kapacitetu). Alternativa koja, donekle,

prevazilazi pobrojane nedostatke je mogućnost da sakuplja energiju iz okruženja kojom bi dopunjavala bateriju ili čak direktno napajala senzorski čvor.

6.1 Povezivanje senzorskih elemenata - senzora

U okviru arhitekture senzorskog čvora senzorski elementi se mogu implementirati kao:

- a) Sastavni delovi na čipu - tipično za buduću generaciju bežičnih dizajna senzorskih čvorova (napredni sistemi na čipu (*SoC - System on Chip*))
- b) Sastavni delovi van čipa - senzorski čvor se sastoji od diskretnih komponenti tipičnih za trenutno dostupne bežične sisteme senzorskog čvora ("na polici").

Naš izbor za dizajn zasniva se na korišćenju komponenti "sa police". Ovo podrazumeva da su senzorski elementi povezani na analogno digitalni konvertor (*ADC*) koji je u našem slučaju sastavni deo mikrokontrolera tj. *MCU* podsistema ili, ako su u pitanju pametni senzori, povezani na *MCU* podsistem putem nekog od serijskih interfejsa ugrađenih u mikrokontroler, najčešće *SPI* ili *I²C*.

Poslednih godina, bežični senzorski čvorovi su pretrpeli značajnu evoluciju od početnih jednostavnih komunikacionih uređaja u efikasne multi senzorske platforme time započinjući novu eru aplikacija koje su svesne sadržaja i servisa. S obzirom da se u sve većoj meri pojavljuju nova i nova kako senzorska tako i računarska sposobnost, upravljanje energetskom efikasnošću senzora postaje centralna tema procesa senzovanje i senzorskog podsistema kod bežičnih senzorskih mreža. U nastavku analiziraćemo različite tipove senzora ugrađene u tipične bežične senzorske čvorove i njihove karakteristike potrošnje energije.

6.2 Tipovi senzora u bežičnim senzorskim mrežama

U savremenim bežičnim senzorskim mrežama može da se nađe na desetine ugrađenih senzora i sa svakom novom generacijom, a u zavisnosti od aplikacije, broj se sve više povećava. Shodno tome, radi jasnije slike i pokušaja da se napravi referentni model, senzore smo grupisali u četiri kategorije na osnovu kriterijuma konteksta primene:

- ✓ Senzori pokreta opisuju orijentaciju i kretanje senzorskog čvora, u slučaju da je čvor mobilan, i tu spadaju akcelerometar, žiroskop i magnetometar. Pored toga, mnogi senzorski čvorovi sadrže virtuelne senzore kao što su senzor rotacije, linearni senzor ubrzanja, senzor gravitacije i senzor pokreta koji obuhvataju specifične aspekte jednog ili više osnovnih harverskih senzora.

- ✓ Bežični senzori šalju i primaju informacije iz spoljašnjih izvora i tu spadaju senzori kao što su *GSM, GPS, Wi-Fi, Bluetooth, NFC (Near Field Communication)* i infracrveni senzor.
- ✓ Senzori ambijenta (okoline, životne sredine) senzuju razne informacije iz neposrednog okruženja čvora i tu spadaju mikrofoni, kamera(e), senzor svetlosti, senzor blizine, pritiska, vlažnosti i temperature.
- ✓ Interno ugrađeni senzori za praćenje stanja glavnih funkcija bežičnog senzorskog čvora kao što su: nivo napunjenosti baterije, napon napajanja i temperatura čvora.

Prethodno navedena kategorizacija izvršena je sa ciljem da grupiše senzore prema senzovanju sličnog konteksta. Senzor pokreta je prirodan izbor za bilo kakve zadatke prepoznavanja aktivnosti senzorskog čvora i može da pruži jednostavno rešenje uz malu potrošnju energije za otkrivanje mobilnih i stacionarnih perioda aktivnosti. Primarna upotreba bežičnih senzora može da se podeli na senzovanje lokacije i senzovanje radi komunikacione razmene podataka, ali su takođe pronašli svoju ulogu i u drugim domenima. Od senzora iz ove grupe, *GPS* senzor treba razlikovati kao poseban slučaj energetski efikasnog senzovanja jer je i jedan od najzastupljenijih senzora, ali, ujedno, i jedan od najvećih potrošača.

Senzori okoline (životnog sredine) dodatno višestrukim korišćenjem kamere i mikrofona pružaju informacije o neposrednom okruženju senzorskog čvora koji se mogu objediniti sa sensorima pokreta u cilju pružanja preciznijih informacija o aktivnostima i promenama lokacije.

Glavna namena internih senzora je da filtriraju i prilagođavaju upotrebu drugih senzora i da obezbede da hardver senzorskog čvora radi u kontrolisanim uslovima rada, a ne na ivici otkaza i operećenja.

6.3 Karakterizacija potrošnje energije senzora

Prilikom izrade strategije za uzorkovanje senzora neophodno je da inženjer dobro razume ponašanje senzora, stimulus iz okruženja koji se putem tog senzora prati kao i koji su ključni faktori potrošnje energije tokom njegove aktivnosti. U tu svrhu korisno je imati bar neku grubu karakterizaciju potrošnje snage obično u formi vrlo jednostavnog determinističkog modela. Model snage senzora opisuje različita stanja, faze aktivnosti senzora kao i pripadajuću potrošnju energije datog senzora. Na primer, u Tab. 6-1 navedene su izmerene vrednosti snage i energije za određene tipove senzora.

Tab. 6-1 Merene vrednosti snage i energije za određene tipove senzora

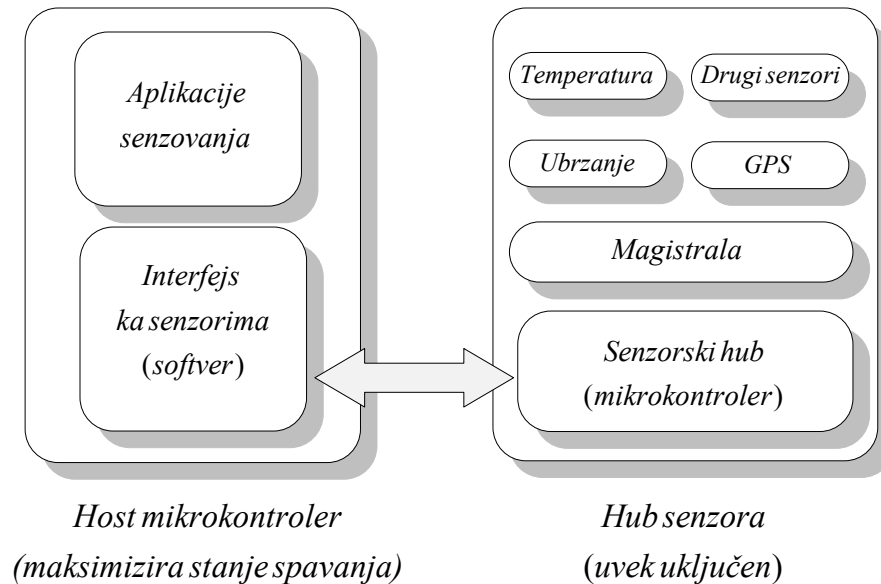
<i>Senzor</i>	<i>uključenje</i>	<i>isključenje</i>	<i>odmeravanje</i>	<i>neaktivnost</i>	<i>pred odmeravanje</i>
<i>Akcelerometar</i>	–	–	<i>21mW</i>	–	–
<i>Gravitacija</i>	–	–	<i>25mW</i>	–	–
<i>Linearni akcelerometar</i>	–	–	<i>25mW</i>	–	–
<i>Magnetometar</i>	–	–	<i>48mW</i>	<i>20mW</i>	–
<i>Orijentacija</i>	–	–	<i>49mW</i>	<i>20mW</i>	–
<i>Rotacija</i>	–	–	<i>50mW</i>	<i>21mW</i>	–
<i>Žiroskop</i>	–	–	<i>130mW</i>	<i>21mW</i>	<i>44mJ</i>
<i>Mikrofon</i>	<i>123mJ</i>	<i>36mJ</i>	<i>101mW</i>	–	–
<i>GPS</i>	<i>77mJ</i>	–	<i>176mW</i>	–	<i>198mW</i>

Hardver bežičnog senzorskog čvora i softver koji ga čini "pametnim" mogu značajno uticati na karakteristike potrošnje energije. Svaki senzor tipično troši neku energiju tokom faze uključivanja i isključivanja i troši snagu tokom faze senzovanja. Takođe, zadržavanjem senzora uključenim, ali ne sve vreme aktivnim, obično troši neku snagu koja je karakteristična za stanje mirovanja.

Granularnost modela može da se proširi tako da uključi različita stanja faze uzorkovanja. Na primer, faza uzorkovanja *GPS* senzora može da se podeli na ostvarivanje *GPS* "uhvati" poziciju (*lock*) (stanje veće potrošnje energije) i prati poziciju (stanje održavanja, manja potrošnja energije). Precizniji modeli mogu se generisati dodavanjem dodatnih informacija modelima (proširenje postojećih modela krupne granularnosti), na primer efektima različitih stopa uzorkovanja (brzine uzorkovanja) na potrošnju energije, standardnim odstupanjima potrošnje energije i vremenom trajanja uključivanja/isključivanja senzora.

Kontinualno senzovanje podrazumeva stalno praćenje senzora na *PCB*-u, kao što su akcelerometar, mikrofon ili kamera. Kao posledica kontinualnog senzovanja javlja se opterećenje procesora i tom prilikom se koristi mnogo energije. Jedno od rešenja je da se prenese proces senzovanja na pomoćni procesor, ukoliko postoji, sa niskim performansama kako bi se uštedela energija. Na sledećoj slici ilustrovan je senzor *hub* koprocesor koji kontinualno prati senzore čime se omogućava da glavni procesor tokom te aktivnosti spava. Senzorski *hub*-ovi predstavljaju efikasan

način da rasterete glavni procesor aktivnosti kao što su prozivanje senzovanih podataka i njihovo naknadno procesiranje kroz namenski projektovano hardversko rešenje (vidi Sl. 6-2).



Sl. 6-2 Prikaz hub senzora

Većina senzorskih aplikacija se sastoje od sekvence stanja koje obrađuju podatke sa senzora. Tipična stanja podrazumevaju sledeće korake:

- ✓ Uzorkovanje i skladištenje (baferovanje), u kojem se signali sa senzora uzorkuju i dobijeni podaci smeštaju u bafer
- ✓ Filtriranje, u kojem se delovi podataka od interesa identifikuju i selektuju za dalju obradu
- ✓ Funkcija ekstrakcije, u kojoj se po nekom parametru izdvajaju delovi iz skupa podataka radi klasifikacije
- ✓ Klasifikacija, u kojoj se podaci klasifikuju na bazi ekstrahovanih osobina korišćenjem metode mašinskog učenja ili metode verovatnoće
- ✓ Post-procesiranje, u kojem aplikacije reaguju na rezultate dobijene tokom aktivnosti senzovanja

Pod pretpostavkom da se aplikacija sastoji od N faza tokom aktivnosti senzovanja, itekako je vazno doneti ispravne odluke o raspoređivanju faza/zadataka (sekvencama izvršenja) uzimajući u obzir energetske karakteristike senzora, procesora, kao i vremena buđenja (*wake-up*).

Cilj ovog dela doktorata da da detaljan prikaz adekvatnih energetskih modela za uštedu energije, razmotri implementaciju tehnika kao što su *duty-cycling* (faktor ispunje) i *power gating*

(upravljanje napajanjem) i ispita kako se i koliko redukuje dinamička i statička potrošnja energije kada se koriste oba načina uštede energije.

Ali pre same analize primene pomenutih tehnika, moramo najpre da sagledamo i izolujemo glavne faktore potrošnje energije.

6.4 Osnove potrošnje energije: faktori, izvori i primeri

Globalno gledano, kada se govori o potrošnji energije, četiri glavna faktora koja je neophodno sagledati, prilikom projektovanja uređaja, su: aplikacija, frekvencija rada, napon napajanja i procesna tehnologija tj. tehnologija izrade, pa prvo što moramo uraditi je da razumemo šta tačno čini navedene faktore toliko važnim.

Dakle, aplikacija je veoma važna, toliko da se profil snage dva uređaja može razlikovati do tačke kada strategije za optimizaciju potrošnje imaju potpuno suprotna stanovišta. Da bismo ovo ilustrovali razmatraćemo dva uređaja: prenosni multimedijalni plejer nasuprot mobilnom telefonu. Prenosni multimedijalni plejer mora da bude u stanju da radi 100% u dužem vremenskom periodu kako bi prikazao video (filmove u punoj dužini) zapise, audio zapise, itd. Opšti profil potrošnje energije ovakvog jednog uređaja očigledno je da treba da se fokusira više na optimizaciju na nivou algoritma i protoka podataka nego, recimo, na efikasnom korišćenju režima rada smanjenje potrošnje.

Uporedimo to sa mobilnim telefonom koji većinu svog vremena provede u stanju mirovanja, a tokom poziva korisnik samo jedan relativno mali deo vremena razgovara. Za ovaj mali deo vremena, jasno je da je procesor izložen velikom opterećenju, vrši kodiranje/dekodiranje govornih podataka i vrši predaju/prijem podataka. U preostalom delu vremena procesor nije u toj meri opterećen, tako da može da izvršava druge aktivnosti kao što su, recimo: slanje paketa podataka o otkucanju srca korisnika, ažuriranje podataka na oblaku, održavanje konekcije sa baznom stanicom i dr.

Za ovaj tip aplikacije očigledno je da će profil potrošnje energije podrazumevati optimizaciju fokusirajući se prvo na maksimiziranje dužine trajanja stanja spavanja procesora kako bi se što više sačuvala snaga, dok se protok podataka/algoritam pristupa razmatraju kasnije.

U slučaju procesne tehnologije, postojeća najnaprednija ugrađena jezgra su zasnovana na *14 nm* tehnologiji, a u bliskoj budućnosti se očekuju i realizacija u *10 nm* pa čak i u *7 nm* tehnologiji. Ono što tekuća i svaka buduća tehnologija manjih dimenzija pruža je manja veličina tranzistora.

Manja veličina tranzistora troši manje energije i oslobađa manju toplotu (disipacija), tako da su očigledno u prednosti u odnosu na svoje prethodnike.

Manja procesna tehnologija generalno omogućuje veću frekvenciju takta, što je takođe pozitivno jer obezbeđuje veću procesorsku moć obrade, ali, sa druge strane, veća frekvencija zajedno sa većim naponom napajanja ostvaruje se po cenu većeg utroška električne energije.

Iako možda deluje previše pojednostavljeno, opšte poznato je da snaga proizvod napona i struje. Dakle ako uređaj zahteva veliki napon napajanja neminovno dolazi do povećanja potrošnje energije. Frekvencija rada je takođe direktni deo ove jednačine jer je struja direktna posledica brzine takta. Naime, kada se naponski izvor primeni na kondenzator, proteći će struja sa izvora prema kondenzatoru sve dok kondenzator ne dostigne ekvivalentan potencijal.

Upravo na takav način taktna mreža u jezgri troši energiju. Dakle na svakoj ivici takta kada se menja potencijal, struja protiče kroz uređaj sve dok ne dostigne sledeće stabilno stanje. Što je brža promena takta to više struje protiče prema tome brži takt podrazumeva veću potrošnju energije ugrađenog procesora. U zavisnosti od uređaja, kola za taktovanje su odgovorna za potrošnju između 50% i 90% dinamičke snage uređaja tako da kontrola i upravljanje taktom predstavlja posebnu temu koja treba biti detaljno analizirana.

6.5 Tipovi potrošnje energije i snage

Ukupna potrošnja električne snage sastoji se od dva tipa snage: dinamičke i statičke (takođe poznata kao statičko curenje) potrošnje, tako da se ukupna snaga potrošnje računa kao:

$$P_{total} = P_{dynamic} + P_{static} \quad (6.1)$$

Kako smo upravo pomenuli tranzicija takta predstavlja veliki deo dinamičke potrošnje. U suštini, u softveru imamo kontrolu nad dinamičkom potrošnjom, ali nemamo kontrolu nad statičkom potrošnjom.

Potrošnja statičke snage: Potrošnja usled struje curenja je snaga koju uređaj troši nezavisno od bilo koje aktivnosti ili zadatka koji jezgro trenutno izvršava, jer čak i u stabilnom stanju postoji mala struja curenja uspostavljanjem puta od napajanja do mase (putem tranzistorske struje tunelovanja, struje curenja inverzno polarizovane diode, i dr.). Jedini faktori koji utiču na potrošnju usled struje curenja su napon napajanja, temperatura i procesna tehnologija.

Već smo u uvodu diskutovali o uticaju napona napajanja i procesnoj tehnologiji. U pogledu temperature, prilično je intuitivno razumeti zašto toplota (zagrevanje) povećava struju curenja.

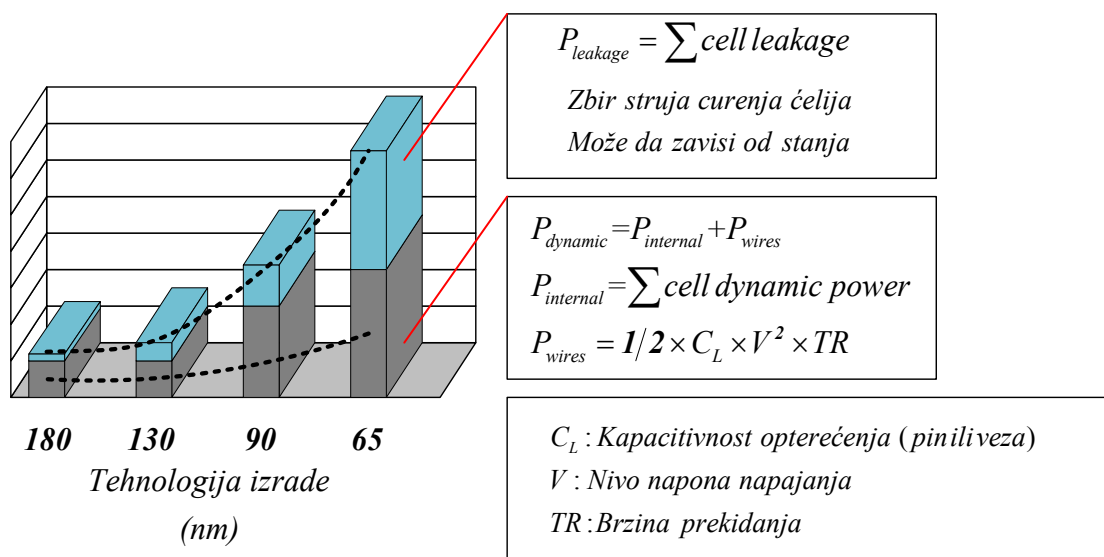
Toplota povećava pokretljivost nosioca naelektrisanja što će dovesti do povećanja protoka elektrona što posleđično dovodi do veće potrošnje statičke snage.

Dinamička potrošnja snage dinamička potrošnja ugrađenih procesora obuhvata snagu koju uređaj koristi aktivno korišćenjem jezgara, podsistema jezgra, perifernih uređaja (blokova) kao što su *DMA*, *I/O* (radio, *Ethernet*, kamera), memorija, *PLL* blokova i takta. Na niskom nivou ovo bi moglo da se prevede na snagu koja se troši od strane prekidačke (komutacione) logike tranzistora kojim se pune i prazne kondenzatori.

Dinamička snaga se povećava kako koristimo više elementa sistema, više jezgara, više aritmetičkih jedinica, više memorije, veće brzine takta ili bilo čega drugog što bi moglo da poveća broj tranzistora koji komutiraju ili brzine na kojima komutiraju. Dinamička potrošnja je nezavisna od temperature, ali zavisi od nivoa napona napajanja.

6.6 Potrošnja struje i njene komponente

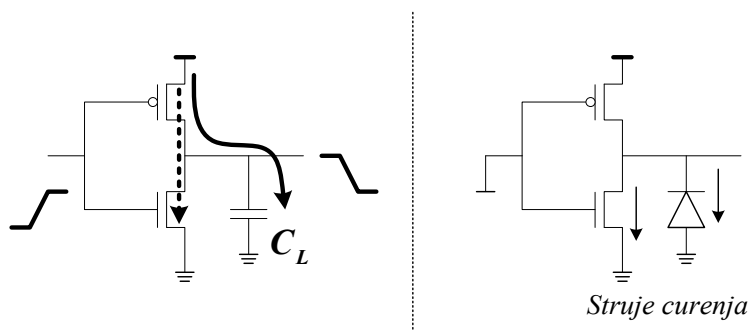
Dakle, kao što smo već istakli, snaga koja se troši u uređaju sastoji se od dva tipa - dinamičke, ponekad nazvane prekidačka ili komutatorska potrošnja snage i statičke, ponekad nazvane snaga potrošnja usled struje curenja. U geometrijama manjim od *90nm* (tehnologija izrade tranzistora) snaga curenja postala je dominantan potrošač snage, dok je za veće geometrije prekidačka imala veći udeo u ukupnoj potrošnji (vidi Sl. 6-3). Strategije smanjenja potrošnje snage mogu da se koriste da minimiziraju oba tipa potrošnje. Da bismo došli do konkretnih strategija razmotrimo sledeće.



Sl. 6-3 Uporedni prikaz potrošnje dinamičke i statičke snage u zavisnosti od tehnologije izrade tranzistora.

6.7 Disipacija snage u CMOS tehnologiji

Ukupna potrošnja snage je funkcija prekidačke aktivnosti, kapacitivnosti, napona i same tranzistorske strukture (vidi Sl. 6-4).



Sl. 6-4 Tipovi struja u CMOS tehnologiji

Ukupna snaga je zbir dinamičke i snage (potrošnje) usled curenja.

$$Total\ Power = P_{switching} + P_{short_switching} + P_{leakage} \quad (6.2)$$

Dinamičke potrošnje je zbir dva faktora: prekidačke potrošnje i potrošnje usled kratkog-spoja (vidi Sl. 6-5). Prekidačka potrošnja se disipira prilikom punjenja ili pražnjenja unutrašnjih i mrežnih kapacitivnosti. Potrošnja usled kratkog spoja je snaga koja se disipira trenutnim uspostavljanjem kratkog spoja između napona napajanja i mase u trenutku kada gejt menja svoje stanje.

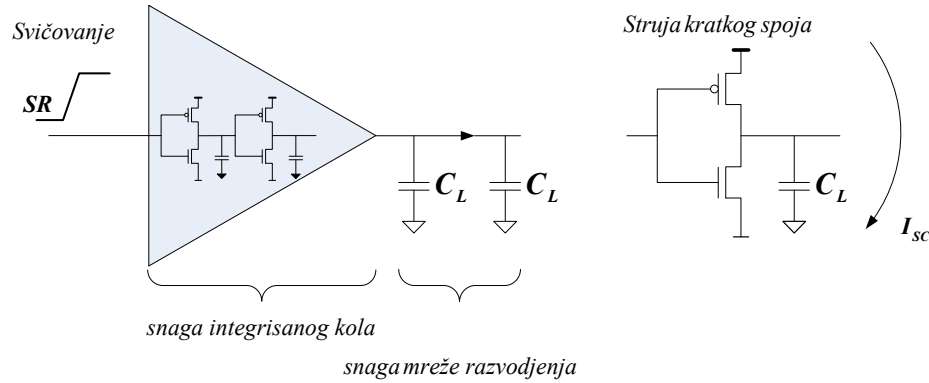
$$P_{switching} = a \cdot f \cdot C_{eff} \cdot V_{dd}^2 \quad (6.3)$$

gde je a - prekidačka aktivnost, f - frekvencija prekidačkog takta (takt komutiranja), C_{eff} - efektivna kapacitivnost i V_{dd} - napon napajanja.

Dok je

$$P_{short-circuit} = I_{sc} \cdot V_{dd} \cdot f \quad (6.4)$$

gde je I_{sc} - struja kratkog spoja u toku promene stanja, V_{dd} - napon napajanja i f - frekvencija takta (komutacije) preklopa.



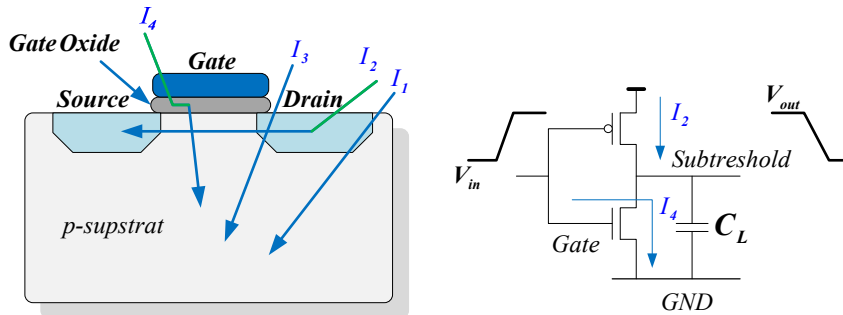
Sl. 6-5 Potrošnja snage usled prekidačke aktivnosti tranzistora i struje kratkog spoja

Dinamička potrošnja se može smanjiti smanjenjem prekidačkih aktivnosti i frekvencije takta što utiče na performanse, kao i smanjenjem kapacitivnosti i napona napajanja. Dinamička potrošnja se takođe može smanjiti odabirom ćelija - naime brže *slew rate* (definiše se kao promena napona ili struje po jedinici vremena) ćelije troše manje dinamičke snage.

Potrošnja usled curenja je funkcija napona napajanja V_{dd} , prekidačkog praga napona V_{th} i veličine tranzistora

$$P_{leakage} = f(V_{dd}, V_{th}, W/L) \tag{6.5}$$

gde je V_{dd} - napon napajanja, V_{th} - napon praga, W - širina tranzistora i L - dužina tranzistora.



Sl. 6-6 Tipovi struje curenja

Od svih mogućih pojava komponenti struje curenja, dominantna je struja curenja pod-praga, vidi Sl. 6-6.

- I_1 struje curenja inverzno polarisane diode (*diode reverse bias current*)
- I_2 struja pod-praga (*sub-threshold current*)
- I_3 struja curenja drejna usled indukovanja gejta

- I_4 struja curenja oksida gejtta

Dok se dinamička snaga disipira samo tokom aktivnosti prekidačke logike, disipacija usled struje curenja je neprekidan (kontinualan) proces i ne zavisi od bilo kakvih aktivnosti.

6.8 Maksimalna snaga, prosečna snaga, snaga najgoreg slučaja i tipična vrednost snage

Prilikom merenja snage ili procenjivanja snage koja je potrebna za normalan rad sistema, postoje četiri glavna tipa snage koje je neophodno da budu sagledane, a to su: maksimalna snaga, prosečna snaga, potrošnja snage u najgorem slučaju i tipična potrošnja snage.

Maksimalna i prosečna (srednja) snaga su opšti pojmovi, koji se koriste za opis samog procesa merenja snage i ne sagledavaju, kroz izmerene vrednosti, uticaj efekata softvera ili drugih varijabli na samu potrošnju snage uređaja.

Jednostavno rečeno, maksimalna snaga predstavlja najveću trenutnu pročitane snagu koja se meri u određenom vremenskom trenutku. Ova vrsta merenja je korisna da pokaže kolika je trenutna snaga koju zahteva uređaj, da bi održao pristojan nivo integriteta signala potreban za pouzdan rad.

Prosečna (srednja) snaga tehnički gledano predstavlja količinu energije koja se potroši u nekom vremenskom intervalu podeljena sa tim intervalom vremena. Inženjerski gledano to se radi tako što se izračunava prosečna struja koja se troši tokom vremena i taj se podatak koristi da bi se odredila snaga. Očitavanja prosečne snage su ono na šta smo fokusirani tokom optimizacije, jer je ovo odlučujući faktor prilikom projektovanja bloka napajanja kojim se iskazuju potrebe sistema, a koristi se i u procesu razumevanja profila toplotne disipacije uređaja.

Vrednosti i snage najgoreg slučaja i tipične snage bazirani su na merenju prosečne (srednje) snage. Na primer, snaga najgoreg slučaja ili profil snage najgoreg slučaja opisuje količinu prosečne snage koju će uređaj koristiti u 100% opterećenju (odnosi se na procesor koji koristi maksimalan broj raspoloživih procesnih jedinica) tokom određenog vremenskog perioda. Ovo se može simulirati stavljanjem jezgara procesora u beskonačnu petlju izvođenjem šest ili više instrukcija po ciklusu (zavisi od broja procesnih jedinica u jezgru), sa više *DMA* kanala koji kontinuirano čitaju ili upisuju iz/u memoriju i periferija koji kontinualno šalju i primaju podatke.

Merenje statičke snage: Potrošnja usled struje curenja na procesoru obično se meri dok je uređaj postavljen u režim rada smanjenje potrošnje, pod pretpostavkom da se u ovom modu isključi takt do

svih jezgara podsistema i periferija. Dodatno, potrošnju struje curenja treba meriti na različitim temperaturama pošto struja curenja varira pod uticajem temperature.

Merenje dinamičke snage: Merenje snage bi trebalo da odredi udeo svakog modula u uređaju pojedinačno kako bi bili u stanju da pružimo informacije o tome kakav uticaj ima određena konfiguracija na potrošnju energije celokupnog sistema. Dinamičku potrošnju snage nalazimo jednostavnim merenjem ukupne snage (na određenoj temperaturi), a zatim oduzimanjem potrošnje usled struje curenja na datoj temperaturi dobijenoj po prethodno opisanoj metodi.

Kada upoređujemo specifične tehnike za optimizaciju snage, upoređujemo rezultate potrošnje energije pre i posle primene svake tehnike kako bismo utvrdili efekat primene te tehnike.

6.9 Hardverske funkcije ugrađene u mikroprocesor i kako ih iskoristiti

Optimizacija hardvera više se fokusira na to kako optimizovati upravljanje taktom i karakteristikama napajanja koje su dostupne u mikroprocesorima i perifernim blokovima. U tom smislu, hardverska podrška ostvarena je putem tzv. režima rada sa smanjenom potrošnjom. Najčešće dostupni režimi rada su gejtovanje napajanja (*power gating*), gejtovanje takta (*clock gating*), skaliranje napona napajanja i skaliranje takta.

6.9.1 Gejtovanje napajanja

Koristi strujni prekidač kojim se prekida veza sa njegovim izvorom za napajanje u toku režima rada mirovanja (*standbay*), da bi eliminisala statičku struju curenja kada se kolo ili blok koji se napaja ne koristi. Korišćenje gejtovanja napajanja dovodi do gubitka stanja i podataka za dato kolo ili blok, što znači da korišćenje ove metode podrazumeva čuvanje potrebnog konteksta/stanja podataka u aktivnu memoriju. Kod embedded procesora koji se sve više i više kreću ka *SoC* rešenjima sa mnogo periferija, u nekim aplikacijama, neke od dostupnih periferija ili većina neće biti potrebne tokom izvršenja određene aplikacije. U tom slučaju opcija sa gejtovanjem napajanja kompletno isključuju nekorišćene periferije u sistemu, a ušteda energije koja se tom prilikom postiže zavisi od specifičnosti periferije.

Važno je napomenuti da se u nekim slučajevima, dokumentacija odnosi na isključivanje periferije putem gejtovanja takta, što se razlikuje od gejtovanja napajanja.

6.9.2 Gejtovanje takta

Kako joj i samo ime kaže, ova tehnika podrazumeva isključivanje takta kolu ili čak dela taktnog stabla u čipu. Kako se dinamička snaga troši tokom procesa komutacije (promene) stanja

pokrenutog promenom takta, ova tehnika omogućava programeru da smanji potrošnju aktiviranjem jedne ili nekoliko instrukcija. Naime, u novije vreme embedded procesori imaju tzv. taktno stablo koji se, od glavnog *PLL* takta, račva u različite taktne domene kako je to zahtevano od strane jezgra, memorije i periferija. Na ovaj način, uz hardversku podršku, omogućeni su nivoi primene gejtovanja takta u cilju prilagođenja potrebama korisnika za uštedu energije

6.9.3 Primeri režima rada za uštedu snage kod *MSP430FR5969* mikrokontrolera

MSP430FR5969 [49] je mikrokontroler firme *Texas Instruments* zasnovan na *FRAM*-u koji različitim aplikacijama embedded inženjera pruža fleksibilnost, bolje performanse i povećanje životnog veka baterije, kroz praktična rešenja kao što su: bežično prikupljanje podataka putem senzora, prikupljanje energije iz okoline, pametne elektro-mreže (*smart grid*), industrijske primene, potrošačka elektronika, automatizacija zgrada i sigurnosni sistemi. Mikrokontroleri *MSP430FR59XX* nude bogati skup funkcija za podršku navedenim aplikacijama i to: do 64kB ugrađenog *FRAM*-a, kontroler direktnog pristupa memoriji (*DMA*), 12-bitni analogno digitalni konvertor (*ADC*), analogni komparator, *IO* kapacitivno osetljivi na dodir i *UART*, *SPI* i *I²C* serijske interfejse. Embedded inženjeri mogu takođe da dodaju svom dizajnu hardvera, *AES* enkripcije, zaštite memorije i hardverski generator slučajnih brojeva realizovan kao akcelerator (*DRGN*) bez zahteva za povećanjem performansi centralne procesorske jedinice (*CPU*), više memorije ili većom snagom.

Tipično je da baterijski napajane aplikacije rade u različitim režimima aktivnosti blokova kako bi produžile trajanje baterije (životni vek baterije). Većina prethodno navedenih aplikacija upravo provodi veći deo svog životnog veka u režimu mirovanja koristeći analogno digitalni konvertor (*ADC*) za merenje signala iz realnog sveta, snimajući podatke u trajnu memoriju za kasnije korišćenje i aktivirajući *CPU* procesiranje radi komunikacije, donošenja odluka i izvršavanja jednostavnih matematičkih operacija. Od ključnog značaja za omogućavanje maksimalnog trajanja baterije je da u svakom od ovih režima imamo ultra nisku potrošnju snage. *MSP430FR59XX* mikrokontroleri upravo *Wolverine* platformom zahtevaju ultra nisku snagu u svakom od navedenih režima, kao na primer: $360nA$ u režimu rada realnog sata (*RTC*), manje od $100\mu A / MHz$ potrošnja snage u aktivnom režimu rada, brzo vreme buđenja od $6.5\mu s$ i visoko precizne periferije kao što su unutrašnje upravljanje napajanjem i 12-bitni *ADC* na $75\mu A$. Nove mikrokontrolere takođe karakteriše ugrađeni *FRAM*, prva u svetu integrisana objedinjena memorijska tehnologija i u industriji sa najnižom snagom postojana memorijska arhitektura koja troši $250\times$ manje energije po

bitu i upisuje brzinom koja je više od $100\times$ veća u odnosu na mikrokontrolere zasnovane na *Flash* i *EEPROM* baziranim memorijama.

Rad sa mikrokontrolerima *MSP430FR59XX* olakšan je dostupnošću hardverskih i softverski alata u vidu resursa koji uključuju 1000+ kodnih primera, *CCS (Code Composer Studio)* softverskog okruženja, *ULP Advisor* alata za analizu softverskog koda, *Grace* softvera za grafičku inicijalizaciju i *MSP430Ware* biblioteka svih periferija u mikrokontroleru.

6.9.4 Upravljanje taktom i naponom napajanja

Neki uređaji imaju mogućnost skaliranja napona i takta, što može pomoći u kreiranju optimizacionih šema snage na nivou uređaja/aplikacije. Skaliranje napona napajanja, kako i samo ime ukazuje, predstavlja proces smanjenja ili povećanja snage. Jedna od metoda je primena *VMR (Voltage Regulator Module)* modula regulatora napajanja pomoću kojeg se kontroliše snaga/napon napajanja prema uređaju.

Uopšteno govoreći, kako se napon napajanja smanjuje žrtvuju se frekvencija/brzina procesora tako da u opštem slučaju napon bi mogao da se smanjuje kada bi se zahtevi prema jezgru ili određenoj periferiji reducirali.

6.9.5 Optimizacija na nivou periferija i algoritama

Ideja ovog segmenta disertacije je da sagledamo kako da minimiziramo potrošnju snage najčešće korišćenjih periferija u embedded procesorima. Prva tema koja se nameće prilikom uključivanja periferije su veličina *burst*-a, brzina prenosa, veličinu transfera i komunikacioni mod. Uobičajene standardne forme periferija u embedded procesorima su *DMA (Direct Memory Access)* pristup, *Ethernet*, *RF* antenski interfejs. *I2C*, *SPI* i *UART* interfejsi su takođe prisutni.

Činjenica je da svi navedeni komunikacioni interfejsi obično zahtevaju sopstvene *PLL*/takt izvore čime povećavaju pojedinačni uticaj na ukupnu potrošnju snage.

Iako je svaki od protokola komunikacije pojedinačnih *I/O* periferija različit, ipak svi oni dele činjenicu da se koriste da čitaju/upisuju podatke na nivou među čipovske komunikacije ili komunikacije sa udaljenim sistemom. Kao takav, jedan od osnovnih ciljeva je da maksimiziramo propusnost dok je periferija u aktivnom režimu rada, a sve u cilju da povećamo efikasnost, ali i vreme koje periferija/uređaj može provesti u stanju smanjenje potrošnje što posledično smanjuje aktivno vreme. Najosnovniji način da ovo postignemo je da povećamo veličinu prenosa/*burst*-a.

6.9.6 Komunikacija periferija jezgo

Sve periferije koriste isti serijski *I/O* hardver tako da se manifestuju i slični problemi. Svi bi mogli da zahtevaju da se drže u aktivnom režimu rada kao forma tzv "buđenja uređaja" ili signalizacije prema jezgru. Ovo je idealno za primenu jer omogućava rad periferije u režimu rada mirovanje (*idle*) ili spavanje (*sleep*). Pitanje je sada kako ostvariti proces buđenja, odnosno kako periferija interaguje sa jezgrom procesora. Postoje tri glavna načina za upravljanje ovim procesom: prozivka (*pooling*), obrada bazirana na vremenu (*time-based processing*) i obrada bazirana na prekidu (*interrupt processing*).

Prozivka: Ovo je ubedljivo najmanje efikasan metod za interakciju jezgro-periferija, jer podrazumeva da je jezgro konstantno budno, tj. u normalnom režimu rada i troši kroz visoko frekventne cikluse takta aktivnu struju samo da bi video da li su podaci spremni. Može da se koristi, ali samo ukoliko potrošnja energije nije od primarnog značaja. U ovom slučaju, prozivka omogućuje jezgru da izbegne preklapanje konteksta koja se javlja tokom obrade prekida čime se štedi nekoliko ciklusa tzv. administrativni režijski troškovi kako bi se brže pristupilo podacima.

Obrada bazirana na vremenu: Efikasna je pod pretpostavkom da će podaci uvek biti dostupni u određenom vremenskom intervalu. Ovaj proces omogućava jezgru da "spava" i koristi tajmer za buđenje (aktiviranje) funkcionalnosti, nakon čega sledi obrada podataka. Loša strana ovog pristupa je složenost i mala fleksibilnost: postavljanje i sinhronizacija zahtevaju mnogo napora od strane programera, a isti efekat se može postići i korišćenjem jednostavne obrade prekida kroz odgovarajuću rutinu.

Obrada bazirana na prekidu: Poslednji komunikacioni mehanizam jezgro-periferija je onaj koji se i najčešće koristi u praksi jer omogućuje procesiranje zasnovano na vremenu bez složenosti softverske arhitekture. Metod za buđenje jezgra iz stanja spavanja podrazumeva sledeće: kada se novi uzorci podataka i paketi prihvate, jezgro se prekida u radu od strane periferije (a može i da se probudi ako je u stanju spavanja) da započne procesiranje novih podataka. Periferija se takođe može koristiti da prekine jezgro kada je spremna za prenos novih podataka tako da jezgro ne mora stalno da vrši prozivku opterećivanjem periferije kako bi video kada su podaci spremni za slanje.

6.9.7 Algoritamski pristup optimizaciji

Od tri glavna područja optimizacije snage koja su ovde navedena, algoritamska optimizacija zahteva najviše napora za postizanje određene dobiti u uštedi snage. Algoritamska optimizacija obuhvata optimizaciju na osnovnom nivou aplikacije, struktuiranje koda, struktuiranje podataka (u

nekim slučajevima, ovo se može smatrati optimizacijom putanje podataka), manipulacija podacima i optimizacija odabranih instrukcija. Navedene aktivnosti se ostvaruju kroz sledeće hardversko-softverske tehnike podrške:

Tehnika odmotavanje petlje podrazumeva izmene u smislu odmotavanja petlji u kodu kako bi optimizovali korišćenje keša. Naime, odmotavanje petlji na ovaj način omogućava kompajleru da u svakoj petlji iteracije koristi više blokova umesto samo jednog čime se povećava paralelizacija procesa i efikasnost koda.

Softverska protočnost Druga tehnika uobičajena za optimizaciju performansi i optimizaciju snage embedded procesora je softverska protočnost. Softverska protočnost je tehnika kojom programer razdvaja skup međusobno zavisnih instrukcija koje bi se obično trebalo izvršiti jedna za drugom tako da na taj način jezgro može započeti procesiranje više instrukcija od jednom u svakom ciklusu.

Eliminacija rekurzije Interesantna tehnika optimizacije je eliminisanje poziva rekurzivnih procedura kako bi se smanjili tzv. režijski troškovi koji se javljaju prilikom poziva funkcije. Pozivi rekurzivnih funkcija zahtevaju da uopšteni kontekst funkcije smesti u magacin (*stack*) sa svakim pozivom. Dodatni problem je ako imamo višestruke varijable za koje treba da izračunamo, na primer faktorijel funkciju od datih varijabli.

Smanjivanje tačnosti Često može da se desi da se vrednosti pojedinih funkcija izračunavaju sa velikom tačnošću bez nekih posebnih razloga što dovodi do komplikovanijih programa koji zahtevaju više funkcionalnih jedinica ili više taktnih ciklusa za dobijanje rezultata.

6.9.8 Sekvence koda male potrošnje i obrasci podataka

Još jedan nivo postizanja uštede ogleda se u specifičnosti instrukcija koje se koriste za pojedine operacije ili realizaciju algoritama. Naime, u nekim situacijama u mogućnosti smo da izvršavamo istu funkciju sa različitim komandama da bi uštedeli snagu, iako je analiza ovakvih kodnih scenarija veoma dugotrajan i detaljima protkan posao. Sve je posledica da različite instrukcije aktiviraju različite funkcionalne jedinice, a samim tim imaju i različite zahteve za napajanjem. Da bismo sve to sagledali od nas se zahteva profilisanje ekvivalentnih instrukcija kako bi razumeli gde postizemo uštedu i kako da postignemo kompromis potrošnje snage.

I na kraju ovog dela disertacije možemo da izvedemo sledeći zaključak. Naime, većina današnjih baterijski napajanih sistema moraju da rade u resursno ograničenim okruženjima, pa zbog toga embedded inženjeri uvek tokom procesa njihovog projektovanja i realizacije moraju da

balansiraju energetska efikasnost i potrošnju električne energije, sa jedne strane, u odnosu na performanse koje njihov dizajn zahteva, sa druge strane.

Takođe tokom poslednje decenije ne samo mobilni telefoni već i širok spektar ostalih embedded uređaja, koji nisu primarno mobilni, su bez "problema" napustili sigurnost tradicionalnog žičanog napajanja i prešli na baterijsko napajanje.

Aplikacije koje koriste bežične senzorske mreže ili neki oblik povezivanje tipa mašina-mašina sada zavise od kapaciteta baterije koji je pre svega određena njenom veličinom, pred kojom se postavljaju dva kontradiktorna zahteva da bude dovoljno mala da se uklapa u takav dizajn i dovoljno velika da daje dovoljno energije.

Neprekidna težnja izražena poslednjih godina ka sveobuhvatnom umrežavanju i bežičnim IoT (*Internet of Things*) stvorila je utisak, kome prethode značajna istraživanja, da smo u stanju da postignemo uspeh na polju prikupljanja energije kako od Sunca putem solarnih ćelija, tako i od oslobođenja disipacije (toplote), vibracija ili nekih drugih ambijentalnih izvora kojom bi mogli da napajamo takve dizajne. Naime, gotovo svaki tehnički dokument ili rad na ovu temu ima kao svoj podtekst izazove napajanja takvih bežičnih uređaja tokom godina ili decenija na predvidljiv i pouzdan način.

Međutim očigledno je da će proći izvesno vreme pre nego što ovi izvori ne postanu praktično upotrebljivi, a do tada embedded inženjeri su i dalje "zaglavljani" u svetu malih baterija u obliku novčića. I dok se pojavljuju nove generacije alternativnih baterija u obliku novčića kao i mikrokontrolera mikro potrošnje (*micro power*) MCU za koje se, od strane proizvođača, kaže da će raditi tokom niza godina, embedded inženjeri imaju dva izazova: prvo da u skladu sa specifikacijom koju dobijaju od strane proizvođača obezbede da mikrokontroleri rade na vrednostima naponima, struje i snage koje su navedene u *data sheet*-u, a drugo da osiguraju da baterije koje koriste pouzdano rade u nizu različitih ambijentalnih uslova.

6.10 Izbor tehnika za smanjenje potrošnje energije

Generalno, prethodnih godina inženjeri su svojim tehnološkim inovacijama, iscrpnim analizama i istraživanjima pronašli brojne načine za uštedu energije tokom procesa dizajna čipa ili dizajna na nivou sistema, ali postoje nekoliko pravila kojima se utvrđuje koji pristupi predstavljaju najbolje rešenje za pojedine dizajne.

U praksi, često se javlja konfuzija o tome koje tehnike gde treba koristiti, koji *IP* ili podsistem je najbolji i kako bi trebalo to sve zajedno upakovati. Odabir uključuje sve od odabira odgovarajućeg nivoa za primenu gejtovanja takta, gejtovanja napajanja ili skaliranja frekvencije.

Izbor zavisi od krajnjeg cilja koji je postavljen u pogledu potrošnje, a svodi se na analizu onih tehnika koje imaju najveću sposobnost zadovoljena zadatih ciljeva u funkciji vremena i energije. Drugim rečima sve se svodi na ispitivanja, testiranja, analize različitih scenarija i dobijanje konkretnih vrednosti brojeva, a onda odluka šta je bolje za ispunjavanje zadatih ciljeva.

Organizacija celokupnog energetski efikasnog dizajna je najvažnija, a dosledna strategija taktovanja i *clock-gating* su takođe nepromenljivo kritične. Pored toga, u tehnikama upravljanja skaliranjem i strujama curenjem na primer, neophodno je da razumemo potrebe aplikacije i da posedujemo *use case* ciljeve kako bismo učinili pravi izbor. *Use case* u ovom kontekstu treba shvatiti kao listu scenarija da bi postigli postavljeni cilj ili kao metodologiju koja se koristi u analizama sistema da identifikuje, pojašni i sistematizuje sistemske zahteve. *Use case* se sastoji od skupa mogućih sekvenci interakcija između sistema i korisnika u određenom okruženju i povezanih sa određenim ciljem.

Dakle, danas postoji mnogo različitih vrsta dizajna, mnogo različitih tehnika se mogu primeniti u praksi, ali to ne znači nužno da primenom svih tehnika postizemo najbolje rezultate. Dakle, sve se svodi na sentencu "analiza pre primene".

Naime, ako sistem skoro nikad ne ide u režim rada isključivanja napajanja, u celini gledano njegova upotreba će potrošiti više energije (neophodno je dodavanje memorije za čuvanje (*retention*) koji troše dodatnu snagu, latencija buđenja (*wakeup*) je velika, itd.). Sa druge strane, ako je sistemu potrebno više snage, ali će biti aktivan samo 10% od ukupnog vremena, to znači da će preostalih 90% vremena moći da se isključi što će dovesti do ukupne uštede energije. Naravno u ovom scenariju moramo da znamo koliko je energija potrebno kada je sistem uključen, trajanje vremena kada je sistem isključen i druge parametre, kako bismo bili u stanju da, u početnoj fazi projektovanja, sagledamo dobit.

Jedan od ključnih napredaka u projektovanju sistema male potrošnje je opšte prihvaćeno prepoznavanje da je u procesu projektovanja snaga globalna. To nam ukazuje da se može uticati na svaki aspekt sistema i na svim nivoima kako bi se uštedelo.

6.11 Podsystem napajanja

U senzorskom čvoru svaki podsistem ili pojedinačno integrisano kolo zahteva različite napone napajanja za svoje funkcionisanje. Na primer, u trenutno najčešće korišćenim dizajnima mikrokontroler i druga digitalna kola mogu da rade na naponu napajanja koji se kreću od $3V$ do $1.8V$. Analogne komponente kao što su RF primopredajnik i senzorski elementi, kako bi se obezbedio pravilan rad i potrebne margine šuma, zahtevaju veće napone napajanja. Baterije koje su ugrađene kao izvor napajanja u senzorskim čvorovima su ograničene u svom izlaznom naponu od strane hemije, a u procesu korišćenja se degradira naponski izlaz.

Budući da tokom životnog veka baterije, napon baterije obično ne odgovara željenim opsezima napona napajanja podsistema/integriranih kola, neophodno je koristiti dodatne konvertore i regulatore realizovane u vidu prekidačkog $dc-dc$ konvertora ili linearnog ldo (*linear low-drop out*) regulatora. Imajući u vidu da je, za tekuća rešenja senzorskih čvorova, potrošnja u opsegu od nekoliko desetina mA (u aktivnom režimu rada) do nekoliko μA (u režimu rada spavanja) elektronika napajanja mora biti posebno dizajnirana za rad u sistemu ultra niske potrošnje.

6.11.1 LDO vs prekidačka logika

Pre nego što se detaljnije pozabavimo suptilnostima linearnih regulatora za bežične senzorske mreže, neophodno je da damo uporedne karakteristike linearnih i prekidačkih tipova regulatora, kao dve topologije ili dva tipa konvertora koje su najzastupljenije na polju $DC-DC$ konvertora, i obrazložimo na bazi kojih argumenata su učinjeni izbori tokom projektovanja i praktične realizacije bežičnog senzorskog čvora.

Naime, prekidački regulatori u nekim slučajevima pružaju značajne pogodnosti u baterijski napajanim sistemima. Na primer, ako visoko-performansni konvertor u prekidačkom režimu rada postigne 90% efikasnost (u transformisanju snage baterije u snagu sistema, tipična efikasnost LDO -a je u opsegu od 60% do 80%, dok je kod prekidačkih regulatora uglavnom u opsegu od 75% do 95%), onda je malo verovatno da će upotreba linearnog regulatora produžiti životni vek baterije, osim u situaciji kada je naponska razlika između napona baterije i izlaznog napona linearnog regulatora mala.

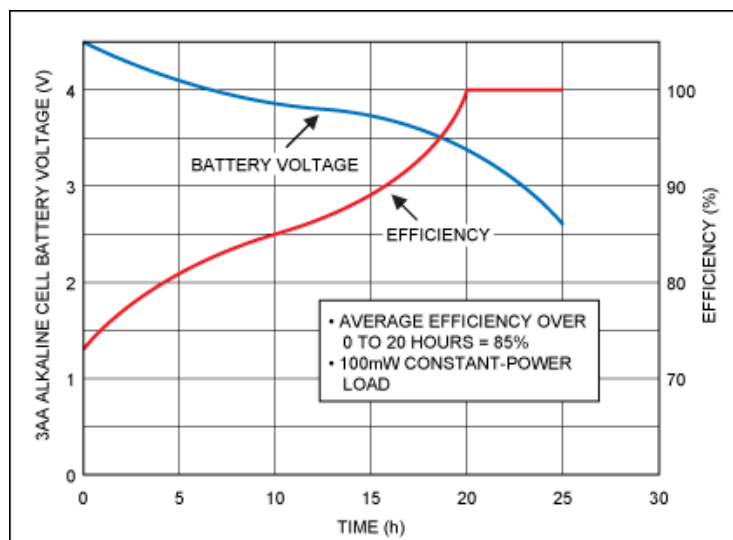
Ovo je jedna od ključnih postavki koja opravdava korišćenje linearnih regulatora.

Naime, stvarni efekat izmerene efikasnosti na životni vek baterije može da bude itekako varljiv. Za mnoge konfiguracije baterija koje su u upotrebi, efikasnost linearnog regulatora je u velikoj meri prihvatljiva, kada se razmatra tokom celog ciklusa procesa pražnjenja baterije.

Za dizajn sa vrlo malom potrošnjom, čak i veliki nedostatak u efikasnosti linearnih regulatora može da bude prihvatljiv. Ako tome dodamo da su jeftini, sa malim brojem spoljašnjih komponenti i malom kompleksnošću kola, onda imamo dosta argumenata za njihov izbor prilikom projektovanja bežičnog senzorskog čvora.

Dobra polazna tačka u projektovanju napajanja za bežične senzorske mreže je da se razmotre i analiziraju rezultati dobijeni korišćenjem realne baterije umesto idealizovanog napajanja na ulazu regulatora. Osnovna karakteristika većina baterija je da imaju ne nultu izlaznu otpornost i relativno brzo opadanje napona do ustaljene vrednosti što je u vezi sa načinom pražnjenja ćelije. Na taj način profil pražnjenja baterije radi u korist linearnog regulatora, jer je disipacija snage u funkciji ulazno izlazne naponske razlike (napona na ulazu i izlazu linearnog regulatora).

Naime, visok napon izvora i loša efikasnost regulatora polazne su karakteristike sveže napunjenje ćelije, ali sa padom napona baterije efikasnost se zapravo poboljšava. Na projektovanom padu napona, kada je napon na ulazu V_{IN} skoro jednak naponu na izlazu V_{OUT} , efikasnost linearnog regulatora je skoro 100%. Ova pojava je suprotna od one tipične za regulatore koji rade u prekidačkom režimu rada. Važna pitanja, međutim, su koji nivo efikasnosti preovladava za produženje trajanja baterije i kakav uticaj ima ovaj profil efikasnosti na životni vek baterije? Opšti zaključak je da čak i ako u startu imamo lošiju efikasnost koja u odnosu na životni vek baterije traje relativno kratko vreme, njen uticaj na trajanje baterije možda neće biti od značaja.



Sl. 6-7 Efikasnost se povećava kako se napon baterije smanjuje za razmatrani 3-čelijski linearno regulisani sistem sa opterećenjem konstantne snage od 100 mW. Prosečna efikasnost neposredno do prestanka rada je 85%. U trenutku samog prestanka rada u bateriji približno ostaje energije za rad još 5 sati (20% od ukupne), što daje ukupnu kumulativnu efikasnost od 68%.

Analizirajući krive sa Sl. 6-7, kao nedostatak koji možemo da uočimo, 20% energije (5 od 25 sati) ostaje u bateriji u trenutku kada regulator dostigne napon kada sistem ne može više da radi tzv. krajni napon pražnjenja (*end of discharge voltage*). Tako da je procenat energije baterije koja se iskoristi $85\% \times 80\% = 68\%$, što i dalje nije tako loše za dizajn sa niskim troškovima. Krive pokazuju da ako sistem može da radi do pada napona na $3V$, povećanje procenta iskorišćenosti energije baterije na $85\% \times 90\% = 76.5\%$ obezbeđuje dodatnih 2.5 časova rada.

Međutim postavlja se pitanje da li postoji neki drugi način poboljšanja efikasnosti?

Najjeftinija metoda za poboljšanje efikasnosti u senzorskim mrežama je smanjenje struje opterećenja što ima za posledicu da će linearni regulator biti u stanju da bolje odrade posao. Iako očigledna, ova prednost ne utiče na stvarnu efikasnost regulatora, koja je po definiciji "snaga na izlazu podeljena sa snagom na ulazu". Ipak, smanjenje struje opterećenja može učiniti više za životni vek baterije nego najsavremeniji dizajn regulatora.

Razmotrimo po kojoj ceni se ostvaruje poboljšanje od 10% u efikasnosti napajanja, a zatim razmislimo o načinu kako da smanjimo opterećenje za istih 10% ili više – odabirom čipova (*IC*) male potrošnje, povećanjem *pull-up* otpornosti ili strožijim upravljanjem napajanjem smanjujući radni napon. Često, bolje je novac potrošiti na strani opterećenja nego na sam regulator. Smanjenje izlaznog napona u linearno regulisanim sistemima potvrđuje ovu tvrdnju. U većini slučajeva, životni

vek baterije se produžava zato što IC kola na strani opterećenja "vuku" manju struju, ali se efikasnost realno smanjuje ($[(V_{OUT} \times I_{OUT}) / V_{IN} \times I_{IN}]$ opada). Drugo pitanje je zašto brinuti o mirnoj struji ?

Druga specifikacija regulatora koja je kritična u aplikacijama koje se baterijski napajaju je "mirna struja", takođe nazvana i "radna struja" ili "struja mase". Ova struja nikada ne dolazi do opterećenja, nego se ona crpe iz baterije i služi da napaja sam regulator. Značaj poznavanja ove specifikacije ogleda se u odnosu magnituda mirne struje u odnosu na magnitudu struje opterećenja. Naime, ako je struja opterećenja $350mA$, a mirna struja $1mA$ njen doprinos neefikasnosti je samo -0.28% . Međutim za opterećenje od $1mA$, procenat gubitaka je mnogo lošiji i iznosi -50% .

Struje opterećenja često variraju u širokom opsegu tako da stvarni efekat mirne struje na životni vek baterije zavisi od kombinacije ova dva krajnja slučaja. Pitanje koje se postavlja je: koja se vrednost struje opterećenja najduže zadržava ? Za slučaj da su struje opterećenja relativno male veći deo vremena, onda moramo da obezbedimo nisku mirnu struju kako bi postigli visoku efikasnost. Ova opaska je posebno kritična u aplikacijama i dizajnima koji se nikada ne isključuju. Naime, čak i u situaciji kada isključimo dugme na uređaju to može da znači da predstavlja režim rada spavanja ili stanja čekanja ili pripravnosti u kojima je sistem napajanja aktivan samo što je struja opterećenja reda mikroampera.

Ponašanje mirne struje tokom pada napona je dosta suptilno, ali važna je razlika između regulatora sa malim padom napon (LDO) realizovanim sa pnp i $pFET$ tranzistorima. Naime, $pFET$ gejtz tranzistora virtualno ne vuče struju, tako da $pFET$ regulatori ne pokazuju gotovo nikakav porast u trenutku dostizanja pada napona. Međutim, kod pnp tranzistora mirna struja raste kako se regulator bori da održi svoj izlazni napon tako što povlači struju baze prema masi. Ovo treba imati u vidu prilikom izbora načina realizacije regulatora sa malim padom napona.

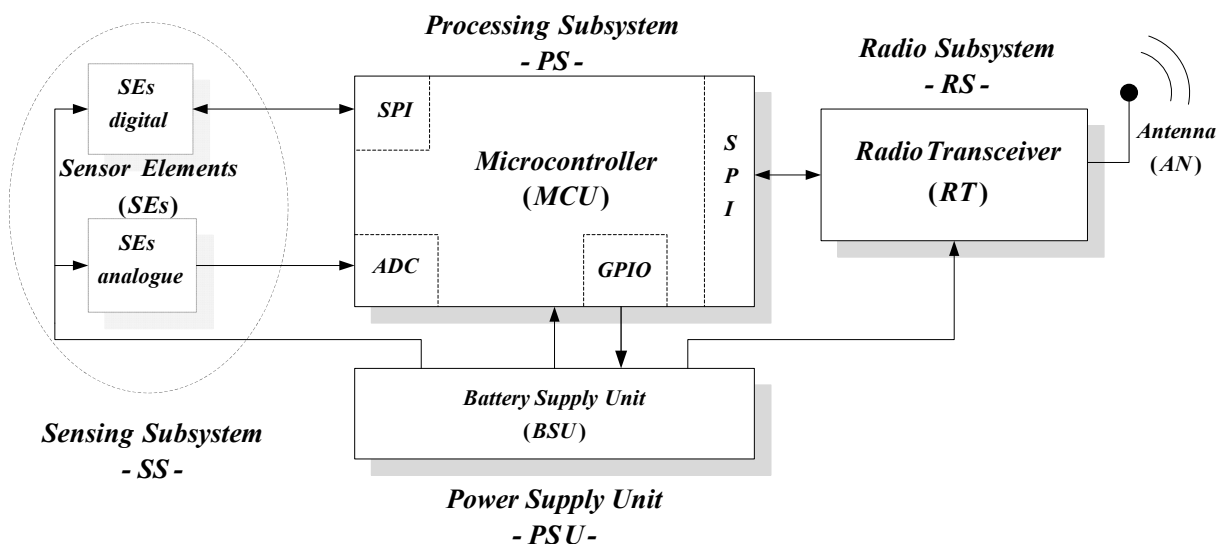
6.11.2 Upravljanje opterećenjem

Da bi se smanjio bespotrebni odliv struje sa baterije, mnogi baterijski napajani sistemi funkcionišu tako što uključuju pojedine svoje unutrašnje blokove samo onda kada su neophodni. Ovaj proces prekidanja se često realizuje sa logikom pobuđivani $pFET$ prekidačima koje prate regulatori napajanja. Kako bi se izbegli regulacioni gubici tokom isporuke struje vršnog opterećenja, otpornost tokom uključivanja mora da bude dovoljno niska kako bi se obezbedilo da napon na strani opterećenja ostane iznad minimalno specificirane vrednosti.

Navedeni problem otpornosti prekidača dodatno se komplikuje u niskonaponskim sistemima sa naponom napajanja od 3.3 V i niže, jer nisko pobudni gejt ne može dovoljno da minimizira otpornost uključenja *FET*-a. Troškovi *FET*-ova sa niskim pragom okidanja gejta opadaju. Međutim u mnogim slučajevima, korišćenje više linearnih regulatora nude bolji pristup. Mnogi novi linearni regulatori imaju mogućnost isključivanja na nivou logike kojim se potpuno isključuje izlaz regulatora što omogućava uređaju da služi i kao regulator i kao prekidač.

7 HARDVERSKA REALIZACIJA SENZORSKOG ČVORA

Na Sl. 7-1 prikazana je arhitektura senzorskog čvora koja se koristi u našem dizajnu. Sastoji se od četiri glavne komponente ili bloka odnosno podsistema: i) Podsystem senzora (*SS*) – obuhvata do šesnaest senzorskih elemenata tj. senzora (*SE*); ii) Podsystem za obradu (*PS*) – sastavljenog od mikrokontrolera (*MCU*), ulazno-izlaznih interfejsa tipa *SPI* i *GPIO*, *ADC* i memorije; iii) Radio podsistema (*RS*) – obuhvata radio primopredajnik (*RT*) i antenu (*AN*), i iv) Jedinice za napajanje (*PSU*). Hardverska struktura senzorskog čvora sa Sl. 7-1 je identična onoj koja je već opisana u delu disertacije Poglavlje 6., ali sada sa mnogo više detalja.



Sl. 7-1 Arhitektura senzorskog čvora

7.1 Podsystem za obradu

MCU, kao ključni gradivni blok u okviru *PS*, obavlja zadatke, obrađuje podatke i upravlja funkcionalnošću ostalih komponenti u okviru senzorskog čvora. Tokom procesa projektovanja naša glavna namera je bila odabrati *MCU* koji će ispuniti, svojom arhitekturom, ograničenja koja se odnose na izvršenje operacija velikom brzinom, ali sa malom potrošnjom snage. Da bi smanjili potrošnju, aktivnosti senzorskog čvora smo organizovali tako da najveći deo svog životnog veka ($>99\%$) provede u stanju spavanja zbog toga što u tom režimu rada "vuče" struju koja je vrlo mala i iznosi $\sim 1 \mu A$. Najadekvatniji izbor od komercijalno dostupnih mikrokontrolera je *MSP430FR5969* sa ugrađenim *FRAM*-om od $64kB$ i *LPM3.5* režimom rada takta realnog vremena *RTC* sa srednjom strujom od $0.25 \mu A$ [49]. Mikrokontrolerski pinovi su deljivog tipa tj. korisnički orijentisani i opslužuju različite funkcije kao što su serijske komunikacije (*SPI*, *UART*), izlaz tajmera, analogni ulazi (*ADC*) i digitalne ulaze/izlaze opšte namene (*GPIO*).

7.2 Radio podsystem

Radio priomopredajnik glavni blok radio podsystema omogućava dvosmernu radio komunikacionu vezu između senzorskog čvora koji šalje (predajnika) i pristupne tačke (prijemnika). U našem slučaju, od komercijalno dostupnih integrisanih kola, odabrali smo *CC110L* [50] kao radio primopredajnik. Navedeni radio primopredajnik može da se nađe u sledeća četiri režima rada: Predaja (*Transmit*), Prijem (*Receive*), Pasivno stanje (*Idle*) i Stanje spavanja (*Sleep*). U našem rešenju *RS* se isključuje kada nije u režimu rada Predaja ili Prijem.

7.3 Jedinica napajanja

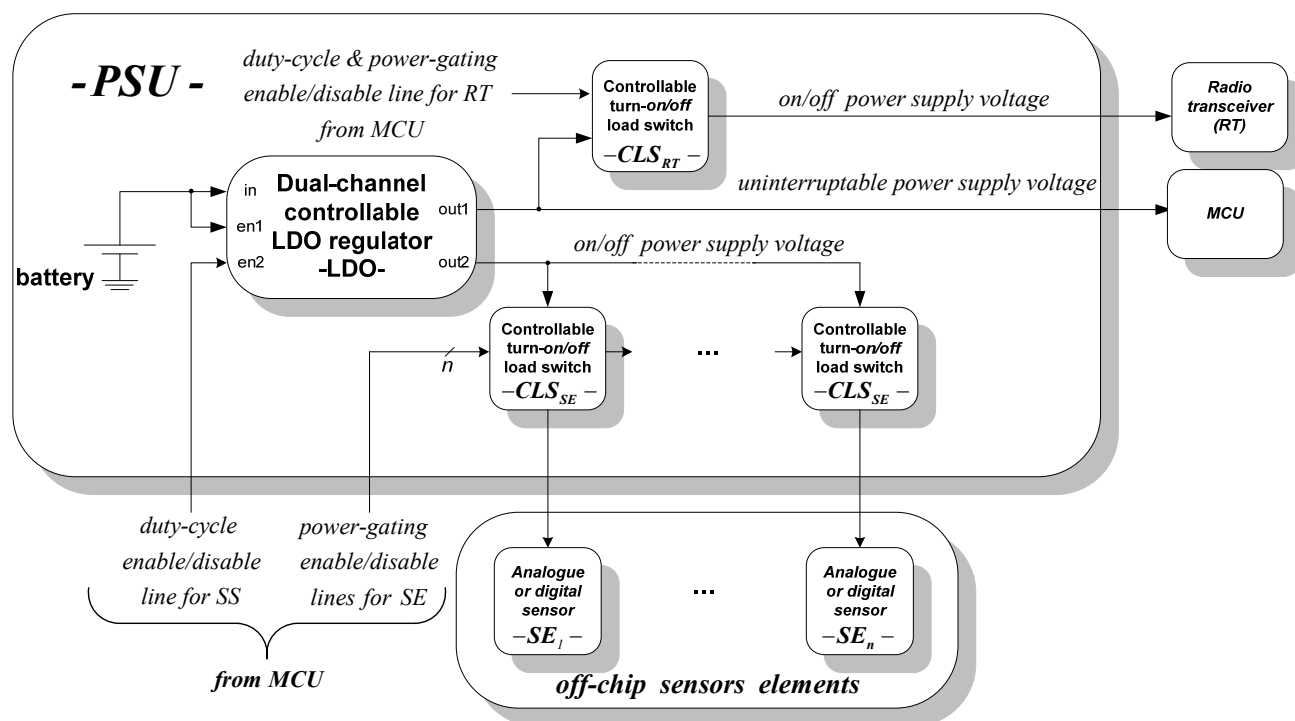
Da bi se postiglo smanjenje potrošnje energije u okviru arhitekture senzorskog čvora, kako smo već istakli u Poglavlju 6, koristimo kombinaciju dve podrobno obrazložene tehnike za uštedu energije: *duty cycling* i *power gating*. *Duty cycling* se koristi za smanjenje potrošnje snage na sistemskom nivou koje se ostvaruje uključivanjem/isključivanjem arhitekturnih blokova senzorskog čvora i time uvođenjem u aktivno/stanje spavanja, respektivno. *Power gating* tehnika se pak koristi na nivou kola sa ciljem da smanji gubitak snage usled struje curenja (smanjenje statičke potrošnje energije).

U predloženom rešenju se *power gating* tehnika koristi da uključi/isključi senzorske elemente (senzore) tokom procesa prikupljanja podataka iz okruženja i uključi/isključi modul radio

primopredajnika tokom aktivnog/stanja spavanja primopredajne arhitekture senzorskog čvora. U skladu sa ovakvim pristupom struktura *PSU* podsistema je realizovana kao na Sl. 7-2. Dakle, kao usko specifično projektovano rešenje jedinica za napajanje deluje kao kontrolna jedinica koja, sem toga što obezbeđuje adekvatne naponske nivoe za svaki od podsistema senzorskog čvora ili pojedinačne hardverske komponente, pojedinačno uključuje i isključuje napajanje u skladu sa potrebama aplikacije tj. primenjuje algoritam upravljanja napajanjem.

PSU se sastoji od:

- a) Baterija - glavni izvor energije za napajanje podsistema senzorskog čvora za Procesiranje, Senzovanje i Radio komunikaciju;
- b) *LDO* regulator sa dva kanala koji se može upravljati - implementiran je sa jednim ulazom (*in1*) i dva izlaza (*out1* i *out2*). Postavljanjem upravljačkih signala za dozvolu rada *en1* i *en2* na logičku jedinicu/nulu ostvaruje se kontrola dozvole/zabrane generisanja napona na izlazima *out1* i *out2*, respektivno. U našem rešenju, na izlazu *out1* napon je uvek prisutan, jer je kontrolni signal $en1 = \{1\}$, dok se napon na izlaznom pinu *out2* može uključiti/isključiti postavljanjem $en2 = \{1/0\}$, respektivno. Primetimo ovde da je cilj ovako definisanog *BSU* realizacija *power management* sa ostvarivanjem funkcije *power gating* (gejtovanja napajanja) na nivou *OSCE* podsistema koje se postiže kontrolom pina *en2*.



SI. 7-2 Struktura jedinice za napajanje

c) Kontrolni prekidači opterećenja za uključivanje/isključivanje (*CLS* - *Controllable Load Switch*) - svaki *CLS* se implementira kao transistor tipa *p*-kanalni ili *n*-kanalni *MOSFET* koji se može uključiti/isključiti putem upravljačkog signala (pina) za dozvolu/zabranu rada. Na ovaj način obezbeđena je osnova za primenu mehanizma *power gating* na lokalnom nivou unutar senzorskog podsistema, naime, mikrokontroler *MCU* individualno uključuje/isključuje napon napajanja za svaki senzorski element postavljanjem odgovarajuće kontrolne linije na logičku jedinicu/nulu. Kontrolni prekidači opterećenja za uključivanje/isključivanje senzorskih elemenata (CLS_{SE}) – svaki *CLS* je implementiran kao *p*-kanalni *MOSFET* transistor koji se može individualno uključiti/isključiti. Na ovaj način se realizuje *power gating* tehnika na nivou kola tj. senzorskog elementa unutar senzorskog podsistema. Kontrolni prekidač opterećenja za uključivanje/isključivanje komunikacionog modula (CLS_{RT}) – implementiran je kao *p*-kanalni *MOSFET* transistor. Na ovaj način se uključuje/isključuje komunikacioni podsistem.

7.4 Senzorski podsistem

U okviru arhitekture senzorskog čvora senzorski elementi se mogu implementirati kao:

- a) Sastavni delovi na čipu - tipično za buduću generaciju bežičnih dizajna senzorskih čvorova (napredni sistemi na čipu (*SoC - System on Chip*))
- b) Sastavni delovi van čipa - senzorski čvor se sastoji od diskretnih komponenti tipičnih za trenutno dostupne bežične sisteme senzorskog čvora ("na polici").

8 ENERGETSKI PROFIL

Predložena bežična senzorska mreža koja predstavlja osnovu za sva dalja istraživanja i analize, sastoji se od nekoliko senzorskih čvorova raspoređenih u ograničenom području. Ovako koncipiran sistem prvenstveno je namenjen za praćenje skalarnih vrednosti raznih parametara iz okruženja. Na ovaj način, a shodno potrebama aplikacije, pretpostavićemo da se skoro sva navedena merenja periodično uzorkuju, ali ne sa nekom velikom brzinom odmeravanja. Na primer, uzimanje jednog uzorka na dva minuta može biti adekvatno za nadgledanje temperature, pritiska, količine svetlosti, vlažnosti i drugih parametara. Kao posledica navedenog nameće se upravljanje napajanjem kao efikasan način za očuvanje energije u bežičnim senzorskim mrežama. Ključna ideja upravljanja potrošnjom je da dinamički, po potrebi, učinimo senzorski čvor neaktivnim da bi smanjili njegovu potrošnju energije, tj. da odlučimo kada senzorski čvor treba da ide u neaktivno stanje i koliko vremena da provede tamo.

Većina strategija upravljanja potrošnjom predloženih u literaturi [51-52] pretpostavlja da se na prikupljanju podataka (aktivnost *sensing*) troši znatno manje energiju u poređenju sa bežičnim prenosom podataka. [15]

Međutim, u velikom broju praktičnih aplikacija ova pretpostavka realno ne stoji, naročito u slučajevima kada se koriste aktivni senzori čija je potrošnja energije uporediva sa onom komunikacionog podsistema. Sličan problem je već razmatran u [53-54].

Kako bi se suočili sa ovim izazovom na efikasan način, predlažemo da koncept upravljanja snagom implementiramo na dva nivoa: a) na nivou sistema; b) na nivou komponenti.

Na prvom nivou bi se primenila tehnika *duty-cycling* putem koje bi se identifikovala neaktivna (*idle*) i aktivna vremena podsistema senzorskog čvora. A na drugom nivou bi se primenila tehnika *power-gating* putem koje bi se isključili svi oni senzorski elementi koji se u datom trenutku ne koriste, a samo analizirani senzorski element bi ostao uključen. Drugim rečima, naš cilj je da se u većem delu vremena tokom koga imamo nepotrebnu potrošnju energije senzorskih elemenata usled ne optimalne konfiguracije hardverskih i softverskih komponenti u značajnoj meri smanjimo.

Napomenimo da je senzorski čvor kao električni sistem vremenski invarijantan tj. ukupna potrošnja energije zavisi samo od potrošnje njegovih pojedinačnih komponenti.

Vođeni ovom konstatacijom u nastavku ćemo posebno analizirati efekte i prednosti implementacije tehnika *duty-cycling* i *power-gating* na potrošnju energije samo na podsistem senzovanja kao sastavnog dela senzorskog čvora.

8.1 Tehnika *duty-cycling*

Duty-cycling je dobro poznata tehnika za minimiziranje potrošnje električne energije u bežičnim senzorskim čvorovima. Glavna ideja koja stoji iza toga je jasna: držimo hardver (podsistem senzovanja, komunikacija, neke delove podsistema napajanja i procesiranja - vidi Sliku 15.) u stanje niske potrošnje tzv. stanje spavanja, osim u slučajevima kada postoji potreba za nekom aktivnošću tokom koje se aktiviraju pojedini ili svi delovi hardvera. Mnoge realizacije *duty-cycling* tehnike omogućavaju čak i da se *MCU* blok postavi u stanje smanjene potrošnje za duži vremenski period, dok njegov interni ili eksterni *RTC* prati vreme sa ciljem da ga u odabranom trenutku vremena (apsolutnog ili relativnog) triger signalom probudi i vrati u normalni režim rada. Vreme buđenja *MCU*-a treba shvatiti kao vreme koje protekne od trenutka generisanja prekidnog signala do početka izvršenja uslužno prekidne rutine. Upravo ovo vreme predstavlja glavni problem primene navedene tehnike zato što je često, zbog svoje dužine trajanja, relativizuje. Napomenimo da sve aktivnosti koje se odnose na primenu navedene tehnike (prelazak u različite režime rada primopredajnika, *MCU*-a, *LDO* regulatora) ostvaruju se pod softverskom kontrolom.

8.1.1 Analiza tehnike faktora ispunje

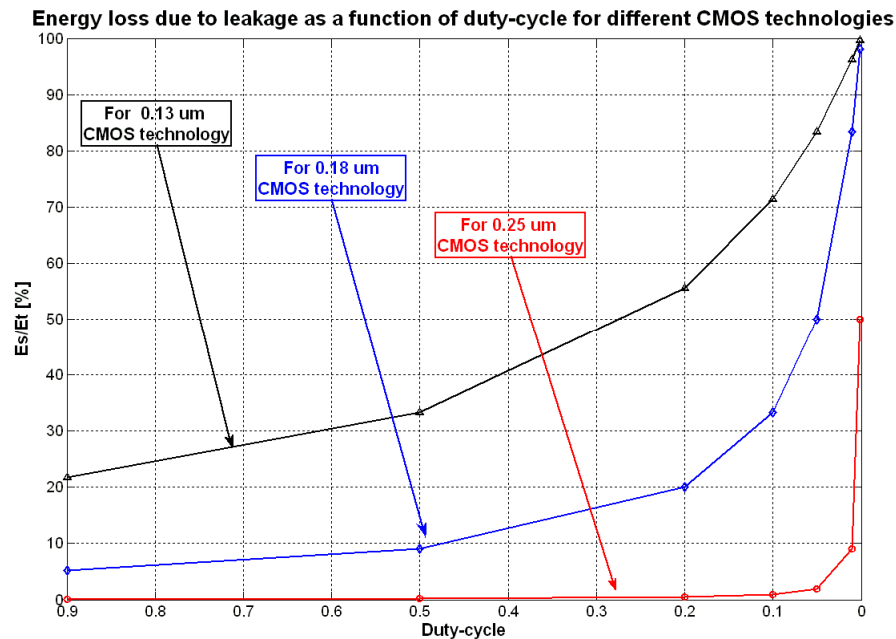
Pretpostavimo da je ukupna energija koja se troši od strane senzorskog čvora, E_{total} i da zavisi od dinamičke (aktivne) energije $E_{dynamic}$ i statičke (pasivne) energije E_{static} , tako da je:

$$\begin{aligned}
 E_{total} &= E_{dynamic} + E_{static} \\
 &= DC \cdot T_{life_time} \cdot P_{dynamic} + T_{life_time} \cdot P_{static}
 \end{aligned}
 \tag{8.1}$$

gde je: DC - *duty-cycling* ($0 < DC < 1$), T_{life_time} - je životni vek senzorskog čvora, a $P_{dynamic}$ i P_{static} odgovaraju dinamičkoj i statičkoj snazi, respektivno. Iz jednačine (8.1) deo energije izgubljen zbog struje curenja može da se iskaže kao:

$$\frac{E_{static}}{E_{total}} = \frac{1}{1 + DC \frac{P_{dynamic}}{P_{static}}}
 \tag{8.2}$$

Odnos $P_{dynamic}/P_{static}$ iskazuje gubitak energije koji zavisi od tehnologije, na koji nemamo uticaj, i proporcionalan je svojstvima kanala *MOS* tranzistora. Slično kao u [55] uzimajući odgovarajući odnos $P_{dynamic}/P_{static}$ za tri različite *CMOS* tehnologije, u stanju smo u da izračunamo uticaj gubitaka energije u odnosu na ukupnu potrošnju u zavisnosti od DC faktora. Dobijeni rezultati su prikazani na Sl. 8-1.



Sl. 8-1 Gubici energije usled struje curenja u funkciji *duty-cycling* za različite *CMOS* tehnologije

Za digitalne komponente senzorskog čvora, slično kao u referenci [56] pretpostavićemo da je navedeni odnos $\approx 1000 @ 0.25 \mu m$, $\approx 20 @ 0.18 \mu m$ i $\approx 4 @ 0.13 \mu m$. Analizom Sl. 8-1 možemo da izvedemo sledeće zaključke:

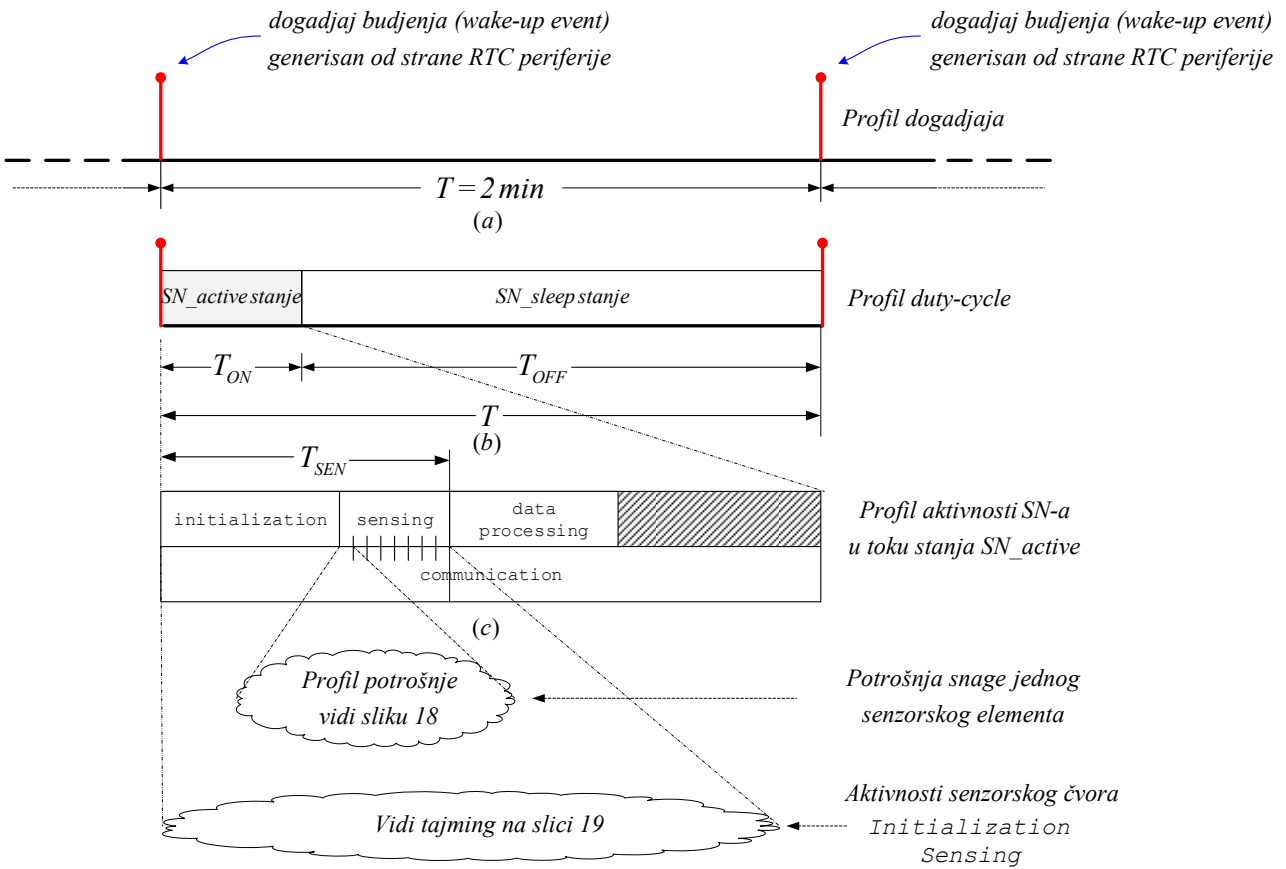
1. Skaliranjem *CMOS* tehnologije, povećava se gubitak energije zbog statičke snage. Drugim rečima, gubici usled struje curenja (statička snaga) su uporedivi sa gubicima usled dinamičke snage [57].

2. U standardnim aplikacijama bežičnih senzorskih mreža *DC*, na nivou senzorskog čvora, je nizak (<1%) što nas dovodi do zaključka da je u ukupnoj snazi sistema dominantan udeo stanja pripravnosti, tj. gubitaka statičke snage usled struje curenja.

3. Teorijski, bolja energetska efikasnost u odnosu na onu koja je postignuta smanjenjem, $E_{dynamic}$, može nadalje da se poboljša smanjenjem *DC* faktora. Međutim, uporedo sa tim postaje kritičan uticaj takta sistema na ukupnu tačnost vremenske sinhronizacije bežične senzorske mreže (*WSN*) [58-59]. Naime, uticaj varijacije temperature okoline na driftovanje takta naglašenije je u visoko izraženim primenama tehnike *duty-cycling* [59].

8.1.2 Gejtovanje napajanja

Sa ciljem da se isključe neaktivni senzorski elementi, a samim tim eliminišu njihove struje curenja, odlučujemo se za implementaciju *power gating* tehnike, koja se inače u projektovanju primarno koristi za smanjenje ukupnih gubitaka statičke snage u kolu [60]. Efikasnost tehnike zavisi od profila aktivnosti komponenti senzorskog čvora. Prilagođavanjem *even-driven* (pobuđivan događajem) upravljačkog mehanizma prvo ćemo predstaviti model aktivnosti senzorskog čvora na opštem nivou, vidi Sl. 8-2, a zatim ćemo, u daljem tekstu, proučiti pitanja potrošnje energije senzorskih elemenata (sastavnih delova podsistema senzovanja - videti Sl. 8-2) koje uključujemo i isključujemo tokom perioda senzovanja i sagledati efekte primene.



Sl. 8-2 Profil aktivnosti senzorskog čvora na opštem nivou

Kao što se može viditi sa Sl. 8-2 *RTC* kolo kao ugrađena periferija *MCU*-a periodično generiše prekidni signal nazvan *wake-up*. Period buđenja signala *wake-up* je T i u našem slučaju, za sprovođenje ove analize, iznosi $T = 2 \text{ min}$. Pojava signala *wake-up* inicijalizuje senzorski čvor i uvodi ga u SN_active stanje tokom kojeg se izvršavaju četiri sekvencijalne aktivnosti inicijalizacija (*initialization*), senzovanje (*sensing*), procesiranje podataka (*data processing*) i komunikacija (*communication*).

Aktivnost *initialization* se odnosi na obnavljanje sadržaja registara *MCU*-a na prethodno SN_active stanje i podešavanje podsistema senzorskog čvora (*LDO* regulator, *CLS* prekidače i primopredajnik, vidi Sl. 7-1 i Sl. 7-2) u odgovarajući režim rada.

Aktivnost *sensing* je odgovorna za prikupljanje informacija iz okoline pomoću senzora i proces konverzije pomoću *ADC* koji je ugrađena periferija u *MCU*. Potrošnja energije tokom ove aktivnosti posledica je više operacija, uključujući uključivanje (isključivanje) senzorskih elemenata,

uzorkovanje signala i analogno-digitalna konverzija ili *SPI* komunikacija, u zavisnosti od toga da li je senzor analognog ili digitalnog (*smart* senzor) tipa.

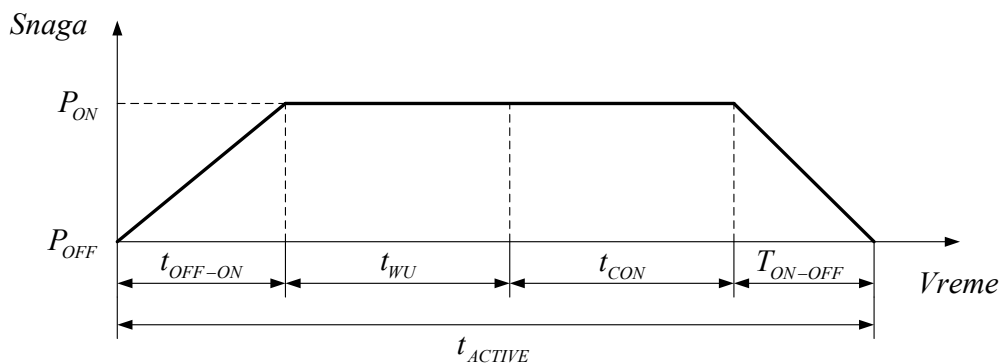
U opštem smislu, pretpostavićemo da imamo n senzorskih elemenata koji su povezani na *MCU* (vidi Sl. 7-2) pa bi u tom slučaju ukupna potrošnja energije senzorskog podsistema u oznaci, E_{Sensor_Total} , mogla da se iskaže kao:

$$E_{Sensor_Total} = N \cdot \sum_{i=1}^n (e_{start_i} + E_{warm_up_i} + E_{conver_i} + e_{stop_i}) \quad (8.3)$$

gde je:

- e_{start_i} - potrošnja energije tokom vremena startovanja senzorskog elementa i
- $E_{warm_up_i}$ - potrošnja energije tokom vremena zagrevanja (*warm-up*) senzorskog elementa i
- E_{conver_i} - potrošnja energije tokom perioda analogno-digitalne konverzije
- e_{stop_i} - potrošnja energije tokom vremena isključivanja senzorskog elementa i
- N - broj *SN_active* stanja tokom životnog veka senzorskog čvora, i
- n - broj senzorskih elemenata u senzorskom čvoru

Profil potrošnje energije senzorskog čvora tokom rada jedne aktivnosti *sensing* na osnovu jednog senzorskog elementa je skiciran na Sl. 8-3.



Sl. 8-3 Individualni profil potrošnje energije senzorskog elementa

Kao što je označeno na Sl. 8-3 t_{off-on} (t_{on-off}) odgovara vremenskom intervalu koji je potreban za prebacivanje senzorskog elementa iz stanja *off* (*on*) u stanje *on* (*off*), T_{WU} predstavlja

vremenski interval zagrevanja senzorskog elementa (*warm-up*), a T_{CON} predstavlja vremenski interval potreban za analogno-digitalnu konverziju.

Osnovni sastavni elementi većine senzorskih elemenata su analogna kola (ulazni i izlazni pojačavači, aktivni filtri i dr.) Kod analognih kola, kolo za gejtovanje napajanja mora da bude uključeno dovoljno dugo pre nego što se uspostavi aktivno radno stanje kako bi se omogućilo kolu da postigne stabilno *dc* stanje. To podrazumeva da i senzorski elementi i njihova sprega sa izvorom stimulusa ne mogu uvek trenutno da odgovore na pobudu.

Naime, senzor se karakteriše vremenski zavisnom karakteristikom i pojavljuje se kašnjenje (latencija) u predstavljanju prave izlazne vrednosti na stimulus. Na Sl. 8-3 ovo kašnjenje odgovara vremenu *warm-up*. U osnovi vreme zagrevanja senzora je vreme koje protekne od trenutka primene napajanja ili eksitacionog signala do trenutka kada sensor može da obavlja svoju namenu u okviru određene tačnosti [60] ili vreme koje protekne od trenutka uključivanja senzora do generisanja stabilne vrednosti na izlazu senzora.

Ovo vreme zavisi od tipa senzora. Mnogi tipovi senzora imaju zanemarljivo kratko vreme zagrevanja (u opsegu od $100\mu s$ do $1ms$) do onih koji rade termičkim ili vlažnim uslovima kao što su termostat ili sensor vlažnosti mogu zahtevati od nekoliko stotina *ms* do sekunde ili minute vremena zagrevanja nakon uključivanja samog senzorskog elementa. Sa aspekta potrošnje energije i primene *power gating* tehnike ova pojava zahteva detaljnu analizu kojom možemo da utvrdimo pod kojim uslovima ima smisla primenjivati navedene tehnike i kakav će uticaj na ukupnu potrošnju imati različita vremena zagrevanja.

Za početak, pretpostavićemo da su svi senzori homogeni. To znači da za $\forall i, i = 1, \dots, n$ važe sledeći uslovi $E_{warm_up_i} = E_{warm_up} = E_w$, $E_{conver_i} = E_{conver} = E_c$ i $e_{start_i} = e_{stop_i} = e_s$, prema tome jednačina (8.3) sada ima oblik:

$$\begin{aligned} E_{Sensor_Total} &= N \cdot \sum_{i=1}^n (e_{start_i} + E_{warm_up_i} + E_{conver_i} + e_{stop_i}) \\ &= N \cdot n \cdot (2 \cdot e_s + E_w + E_c) \end{aligned} \quad (8.4)$$

Ukupna potrošnja energije tokom vremena zagrevanja je:

$$E_{Warm_up_Total} = N \cdot n \cdot E_w \quad (8.5)$$

Ako pretpostavimo da je u proseku $e_s = 0.1 \cdot E_c$ onda je udeo energije zbog zagrevanja u odnosu na ukupnu energiju

$$\frac{E_{Warm_up_Total}}{E_{Sensor_Total}} = \frac{N \cdot n \cdot E_w}{N \cdot n \cdot (2 \cdot e_s + E_w + E_c)} \quad (8.6)$$

$$= \frac{1}{1 + \frac{1.2 \cdot E_c}{E_w}}$$

Ako nadalje pretpostavimo da je $E_w = k \cdot E_c$, gde je k realan broj, možemo da iskažemo udeo izgubljene energije $E_{Warm_up_Total} / E_{Sensor_Total}$ u funkciji E_w za različite vrednosti k kao je to prikazano u Tab. 8-1.

Tab. 8-1 Udeo gubitka energije u funkciji k

k	0	0.1	0.2	0.5	0.8	1	2	5	10	20	50	100	1000	∞
$\frac{E_{TW}}{E_{ST}}$	0	0.077	0.143	0.294	0.400	0.454	0.625	0.800	0.892	0.943	0.976	0.989	0.999	$\rightarrow 1$

Analizirajući rezultate prikazane u Tab. 8-1 možemo da zaključimo da kako se vreme zagrevanja (*warm-up*) povećava kriva $E_{Warm_up_Total} / E_{Sensor_Total}$ se asimptotski približava vrednosti 1. To znači da na donjoj granici vremena zagrevanja $T_{WU} = 0$, ukupna potrošnja energije iznosi

$$E_{Sensor_Total} = N \cdot n \cdot (2 \cdot e_s + E_c) \Big|_{E_w=0} \quad (8.7)$$

a na gornjoj granici vremena zagrevanja $T_{WU} \rightarrow \infty$, ukupna potrošnja energije iznosi

$$E_{Sensor_Total} \sim N \cdot n \cdot E_w \quad (8.8)$$

tj. parametar E_w postaje dominantan.

Generalno, bolje rešenje za dizajn u pogledu $E_{Warm_up_Total}$ je ono u kojem je vreme zagrevanja (*warm-up*) $T_{WU} \rightarrow 0$, ali u ovom slučaju znači da su svi senzorski elementi sve vreme aktivni (vreme zagrevanja se javlja samo jednom, prilikom prvog aktiviranja senzorskog čvora). Kao direktna posledica ove analize možemo da zaključimo sledeće:

1. vreme zagrevanja senzora (*warm-up*) uvodi značajno kašnjenje (latenciju)
2. eliminisanjem ovog vremena potrošnja energije podsistema senzora će biti visoka
3. da bi se efikasno suočili sa ovim problemom uključićemo *power-gating* tehniku

4. u takvom rešenju vreme zagrevanja senzora se ne sme zanemariti kada se razmatra energetski model senzorskog čvora.

8.2 Procena potrošnje energije

U ovom delu disertacije predstavljamo kompletan profil potrošnje energije bežičnog senzorskog čvora tokom aktivnosti *initialization* i *sensing*, samo u okviru *SN_active* stanja [61]. U našem slučaju podsistem senzovanja označen kao *OSCE* (vidi Sl. 7-2) sastoji se od osam senzorskih elemenata (senzora) SE_1, \dots, SE_8 . Senzorski elementi od SE_1 do SE_7 su analognog tipa i pobuđuju *ADC* koji je realizovan kao periferija na čipu mikrokontrolera *MCU* tipa *MSP430FR5969*. Ovi senzorski elementi prate fizičke parametre tipične u bežičnim senzorskim mrežama kao što su temperatura (*LMT87*), vlažnost (*SHT21S*), ubrzanje (*ADXL377*), ambijentalno osvetljenje (*ISL76671*), senzor položaja (*SS345PT*), senzor pokreta (*L3G3250A*) i audio mikrofoni (*MP33AB01*), respektivno. Poslednji senzor (*T5400*) se koristi za merenje pritiska i izmerene vrednosti prenosi *MCU*-u putem *SPI* interfejsa. Za više detalja o vremenskim i električnim specifikacijama senzorskih elemenata čitalac može da pogleda odgovarajuće veb sajtove (na primer *Farnell* [62]). Napon napajanja je $out2 = 3V$ (označen je kao V_{OUT2} izlaz *low-drop out* dvokanalnog naponskog regulatora *TLV716* [63]. *ACLSi* je implementiran kao *p*-kanalni *MOSFET* tranzistor *TPS2290* [64], vidi Sl. 7-2.

Električne i vremenske specifikacije (koristeći podatke iz dokumentacije uređaja i određene direktnim merenjima) i potrošnja energije po senzorskom elementu (utvrđene proračunom i direktnim merenjima) prikazane su u Tab.8-2.

Tab. 8-2 Električne i vremenske specifikacije i proračun potrošnje energije po senzorskom elementu

# SE	Tip senzora	out2 [V]	Srednja Struja [mA]	t _{OFF-ON} [ms]	e _{FO} [uJ]	T _{WU} [ms]	E _W [uJ]	T _{CON} * [ms]	E _C [uJ]	t _{ON-OFF} [ms]	e _{OF} [uJ]
1	LMT87	3	0.0041	2.01	0.012	-	n.a.	0.0035	0.000043	0.005	0.000031
2	SHT21		0.1811	150.11	40.778	8000	4346.400		0.001902		0.001358
3	ADXL377		0.3011	5.11	2.308	-	n.a.		0.003162		0.002258
4	ISL76671		0.0361	0.205	0.011	0.350	0.038		0.000379		0.000271
5	SS345PT		3.0011	0.11	0.495	0.0015	0.014		0.031512		0.022508
6	L3G3250A		6.3011	0.11	1.040	0.3	5.671		0.066162		0.047258
7	MP33AB01		0.3011	0.11	0.050	-	n.a.		0.003162		0.002258
8	T5400		0.7911	2.61	3.097	10	23.733		16		37.9728

Napomena: Vreme konverzije u oznaci T_{CON} određeno je tipom SAR-ADC, kao sastavnom periferijom MCU (MSP430FR5969), i za 12-bitnu rezoluciju iznosi $3.5 \mu s$ (identično vreme za sve senzorske elemente); n.a. označava nedostupne podatke iz kataloga

U skladu sa jednačinom (8.4) i podacima prikazanim u Tab. 8-2 pod pretpostavkom da je $N=1$ i $n=8$ procenjena potrošnja energije predloženog dizajna u toku uključivanja napajanja senzorskim elementima (inicijalna faza senzorskih aktivnosti) može se izraziti kao:

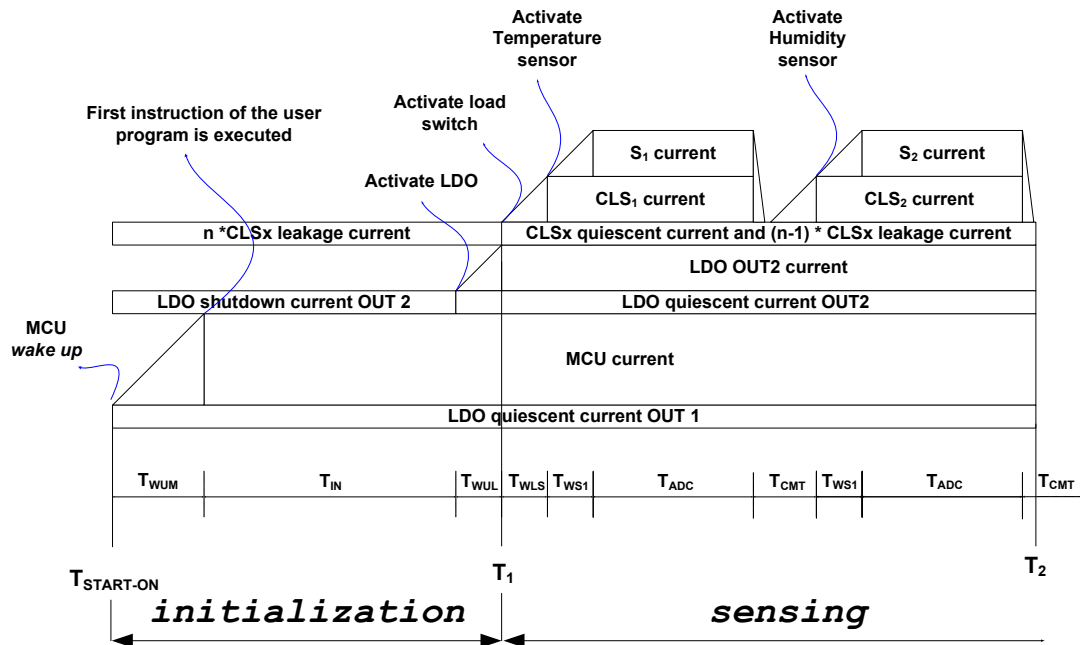
$$\begin{aligned}
 E_{Sensor_Total} &= N \cdot \sum_{i=1}^n (e_{start_i} + E_{warm_up_i} + E_{conver_i} + e_{stop_i}) \\
 &= \sum_{i=1}^n (e_{start_i} + e_{stop_i} + E_{warm_up_i} + E_{conver_i}) \\
 &= 47.791 + .081875 + 4375.856 + 38.07912 \\
 &= 4461.808 \mu J
 \end{aligned} \tag{8.9}$$

Napomenimo da dobijena vrednost odgovara najgorem slučaju potrošnje energije za sve $CLSi$ i SEi tokom aktivnosti `sensing` (Naime, nakon uključivanja napajanja senzorskih elemenata ova se aktivnost dešava samo jednom tokom životnog veka senzorskog čvora. To je tipično za stabilizaciju senzorskih elemenata na uslove okruženja, pa se njegov uticaj, koji se tiče procene potrošnje snage, može zanemariti).

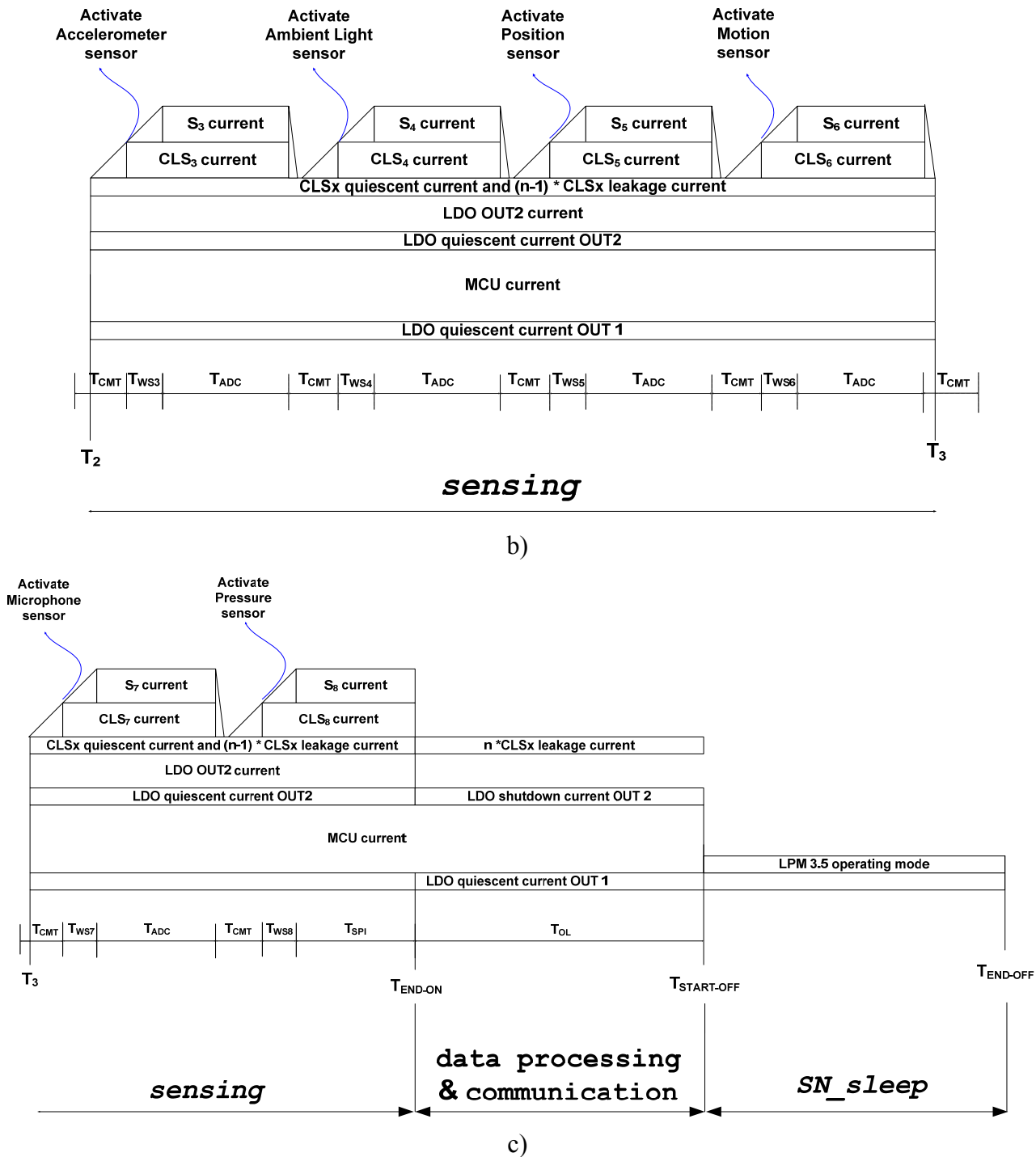
8.3 Energetski profil tokom aktivnosti *initialization* i *sensing*

U cilju određivanja ukupne potrošnje energije tokom aktivnog stanja *active* senzorskog čvora neophodno je uzeti u obzir gubitke energije i drugih blokova senzorskog čvora kao što su *MCU* i *BSU* (vidi Sl. 7-1 i Sl. 7-2) u toku vremenskog perioda T_{SEN} (vidi Sliku 17.).

Detaljniji dijagram vremena u toku aktivnosti *initialization* i *sensing* predstavljen je na Sl. 8-4 i generisan je primenom Matlab softverskog alata.



a)



Sl. 8-4 Profil potrošnje snage u toku aktivnosti sensing

Kao što se vidi sa Sl. 8-4 a) aktivnost initialization počinje u trenutku T_{start_on} i završava se u trenutku T_1 . Aktivnost sensing je prikazan je korišćenjem više dijagrama, tako početni deo ove aktivnosti prikazana je u desnom delu Sl. 8-4 a) sa vremenskim intervalom od T_1 do

T_2 , nastavlja se središnjim delom aktivnosti u vremenskom intervalu od T_2 do T_3 , vidi Sl. 8-4 b), a završava se levim delom aktivnosti u vremenskom intervalu od T_3 do T_{end_on} koji je prikazan na Sl. 8-4 c). Desni deo slike Sl. 8-4 c) obuhvata preostale aktivnosti data processing i communication u vremenskom intervalu T_{end_on} do T_{start_off} i SN_sleep stanje u vremenskom intervalu od T_{start_off} do T_{end_off} . Trajanje ukupnog vremenskog intervala od trenutka T_{start_on} do trenutka T_{end_off} iznosi 2 min.

Detaljniji prikaz trajanja vremenskih intervala svih faza tokom aktivnosti initialization i sensing koji su korišćeni u crtanju dijagrama na Sl. 8-4 uključujući prosečnu struju i potrošnju energije za svaki vremenski podinterval dati su u Tab. 8-3.

Tab. 8-3 Trajanje vremenskih intervala, srednja struja i potrošnja energije u toku aktivnosti initialization i sensing za svaki MCU i OSCE podinterval

<i>Vremenski interval</i>	<i>Trajanje [ms]</i>	<i>Srednja struja [mA]</i>	<i>Energija [μJ]</i>
T_{WUM}	0.3500	0.290	0.1522500
T_{IN}	1.0000	0.495	1.4850000
T_{WUL}	0.9000	0.495	0.6682500
T_{WLS}	0.1600	0.570	0.1368000
T_{WS1}	1.9000	0.590	1.6815000
T_{ADC}	0.0035	0.720	0.0075600
T_{CMT}	0.0050	0.075	0.0005625
	0.1600	0.078	0.0187200
T_{WS2}	150.000	0.729	164.02500
T_{ADC}	0.0035	0.877	0.0092085
T_{CMT}	0.0050	0.075	0.0005625
	0.1600	0.235	0.0564000
T_{WS3}	5.0000	0.869	6.5175000
T_{ADC}	0.0035	1.017	0.0106785
T_{CMT}	0.0050	0.075	0.0005625
	0.1600	0.375	0.0900000
T_{WS4}	0.4450	0.604	0.4031700
T_{ADC}	0.0035	0.752	0.0078960

T_{CMT}	0.0050	0.075	0.0005625
	0.1600	0.110	0.0264000
T_{WS5}	0.0015	3.569	0.0080303
T_{ADC}	0.0035	3.717	0.0390285
T_{CMT}	0.0050	0.075	0.0056250
	0.1600	3.075	0.7380000
T_{WS6}	1.0000	6.869	10.303500
T_{ADC}	0.0035	7.017	0.0736785
T_{CMT}	0.0050	0.075	0.0056250
	0.1600	6.375	1.5300000
T_{WS7}	3.0000	0.869	3.9105000
T_{ADC}	0.0035	1.017	0.0106785
T_{CMT}	0.0050	0.075	0.0056250
	0.1600	0.078	0.0187200
T_{WS8}	10.000	1.359	20.385000
T_{SPI}	16.000	1.373	65.904000
T_{OL}	0.1000	0.496	0.0744000

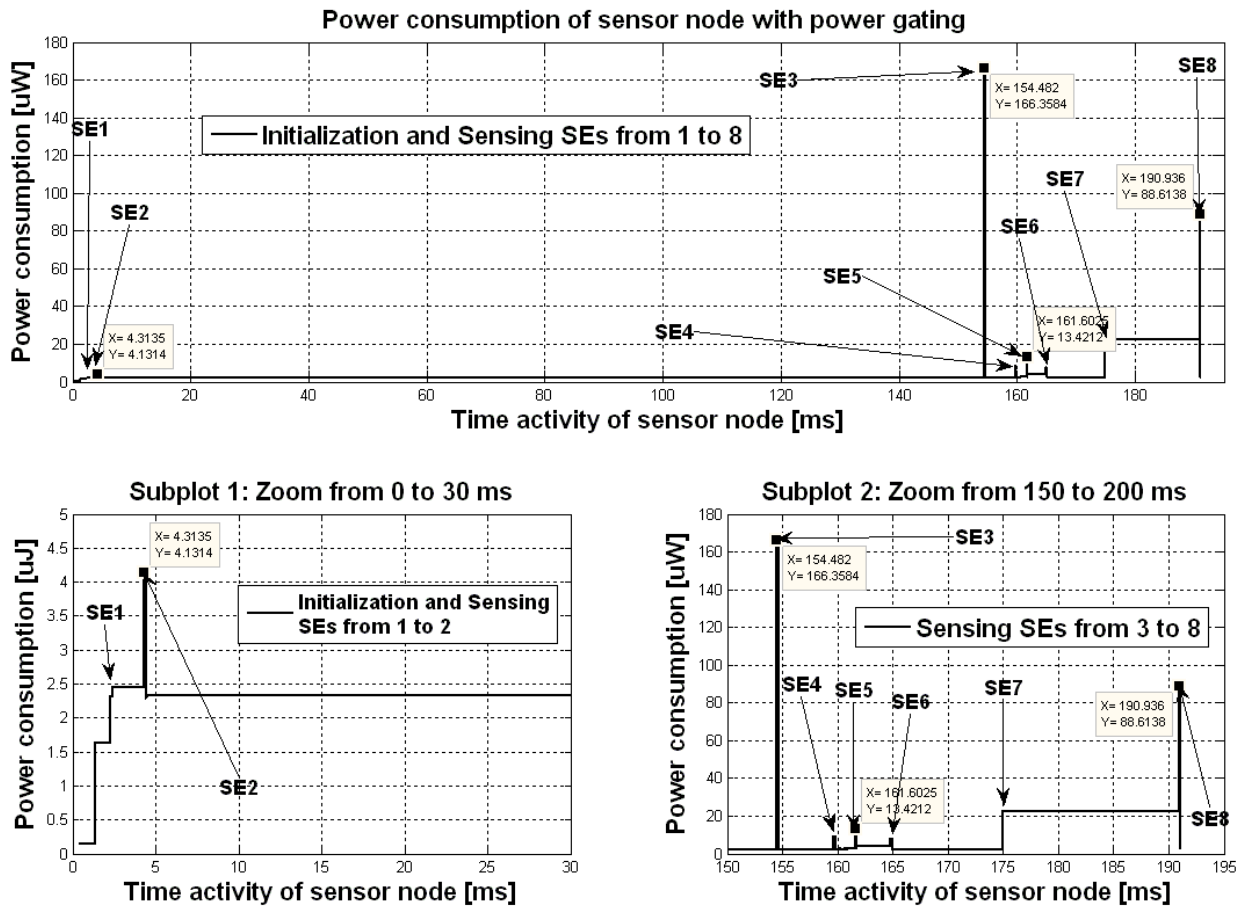
Napomena: Gde je T_{WUM} – vreme buđenja MCU; T_{IN} – vreme inicijalizacije MCU; T_{WUL} – out2 vreme buđenja LDO; T_{WLS} – vreme buđenja CLS; T_{WSx} – vreme zagrevanja senzora $x=\{1,2, \dots,8\}$; T_{ADCx} – vreme konverzije ADC $x=\{1,2, \dots,7\}$; T_{CMT} – vreme komutacije koje uključuje $T_{\text{turn-off}(LSx+Sx)} + T_{\text{turn-on}(LSx)}$; T_{SPI} – vreme prenosa podataka putem SPI; T_{OL} – vreme isključenja LDO

Ukupno vreme trajanja aktivnosti initialization i sensing je $T_{SEN} = 191.036 \text{ ms}$, a odgovarajuća potrošnja energije u ovom vremenskom periodu je $278.31 \mu\text{J}$.

Dijagrami vremena i profil potrošnje energije tokom aktivnosti initialization i sensing (dobijeni korišćenjem MATLAB softverskog alata) prikazani su na Sl. 8-5. Slika Subplot 1 (donji levi deo slike Sl. 8-5) prikazuje aktivnost initialization i prikupljanje podatak sa senzora SE_1 i SE_2 . Slika Subplot 2 (donji desni deo slike Sl. 8-5) odnosi se na prikupljanje podataka sa senzora SE_3 do SE_8 .

Da bismo procenili performanse našeg predloženog dizajna u vezi sa smanjenjem potrošnje energije upoređićemo sledeća dva projektna rešenja:

- a) Prvo, ukupna potrošnja energije senzorskog podsistema u oznaci $E_{dc_ \& _pg}$ tokom aktivnosti *initialization* i *sensing* sa primenjenim tehnikama *duty-cycling* i *power-gating* iznosi $E_{dc_ \& _pg} = 638 \mu J$;
- b) Drugo, ukupna potrošnja energije senzorskog podsistema u oznaci $E_{without_dc_ \& _pg}$ tokom aktivnosti *initialization* i *sensing* bez primene tehnika *duty-cycling* i *power-gating* iznosi $E_{without_dc_ \& _pg} = 3.92 J$;
- c) Napomenimo da se u prvom slučaju aktiviranje senzora ostvaruje primenom *power-gating* tehnike u skladu sa profilom potrošnje energije pojedinačnog senzorskog elementa prikazanog na Sl. 8-2. Takođe navedenih 8 senzorskih elemenata se sekvencijalno sukcesivno uključuju jedan za drugim. Dodatno smanjenje potrošnje energije bi moglo da se ostvari primenom novog algoritma uključivanja senzora u skladu sa njihovim pojedinačnim profilom potrošnje energije što bi za posledicu imalo i skraćenje vremena aktivnosti *sensing*.



Sl. 8-5 Dijagram potrošnje snage u toku initialization and sensing aktivnosti za MCU i OCSE blokove

Napomena: Dijagram je dobije primenom Matlab softverskog alata

Na bazi svega opisanog i procenjenog odnosa $E_{without_dc_ \& _pg} / E_{dc_ \& _pg} = 6146$ možemo da izvedemo zaključak da se opravdava primena obe tehnike za uštedu energije u podsistem senzora (senzovanja) bežičnog senzorskog čvora.

9 TOLERANCIJA NA KVAROVE U BEŽIČNIM SENZORSKIM MREŽAMA

Tehnike tolerancije na kvarove se mogu primeniti na različitim nivoima arhitekture bežičnih senzorskih mreža. Sa stanovišta sistema klasifikujemo toleranciju na kvarove u *WSN* sistemu u sledeća četiri nivoa: hardverski, softverski, mrežno komunikacioni i nivo aplikacije. Za detaljniju analizu koja se odnosi na otkaze u bežičnim senzorskim mrežama, kao i detekciju kvarova i oporavak nakon pojave greške neophodno je konsultovati se sa sledećim referencama [11, 12]. U daljem delu disertacije poseban akcenat će biti stavljen na analizu kvarova na komunikacionom nivou.

9.1 Pouzdanost prenosa podataka u bežičnim senzorskim mrežama

Kroz ovu tezu, izraz pouzdana ukazuje na činjenicu da se podaci moraju preneti bez grešaka ili bez gubitaka. Ponuđena su brojna rešenja za pouzdan prenos podataka kroz bežične senzorske mreže koja uzimaju u obzir fizički, nivo veze, mrežni i transportni nivo kojim obezbeđuje mehanizme pouzdanosti na bit, paket, mrežnom (rutiranje) i transportnom nivou [65]. Sa tačke gledišta pouzdanog prenosa podataka, u daljem tekstu ćemo ukratko dati pregled glavnih zadataka koji se izvršavaju na svakom od nivoa nekog bežičnog komunikacionog protokola.

Fizički nivo: Na ovom nivou se vrše konverzija niza bitova u signale kroz bežični kanal. On je dogovoran za odabir frekvencije, generisanje frekvencije nosioca, modulaciju i šifrovanje podataka (*encryption*). Pouzdanost komunikacije zavisi takođe od hardverskih karakteristika senzorskog čvora, pojačanja antene i osetljivosti (*sensitivity*) prijemnog kola.

Nivo veze: Glavni zadaci ovog nivoa su multipleksiranje/demultipleksiranje podataka, detekcija okvira podataka, pristup medijumu za prenos i kontrola grešaka. Pošto je komunikacioni medijum deljiv, gubici paketa i korupcija (narušavanje integriteta podataka) su uobičajena pojava u bežičnim senzorskim mrežama. Mehanizmi za kontrolu grešaka koji se koriste u bežičnim senzorskim mrežama mogu da se kategorizuju u sledeća tri pristupa:

- 1) Kontrola izlazne snage antene – namenjena da upravlja predajnom snagom koja se može koristiti za postizanje željene stope greške;
- 2) Zahtev za ponovnim slanjem (*Automatic Repeat reQuest - ARQ*) nedostajući paketi se ponovo šalju nakon isteka nekog vremenskog intervala ili eksplicitnog zahteva od strane prijemnika; i
- 3) Korekcija grešaka na odredištu (*Forward Error Correction - FEC*) obično se zasnivaju na korišćenju kodova za otkrivanje i korekciju grešaka.

Mrežni nivo: Glavni cilj je da se odrede efikasne rute za predaju informacija o događajima od izvornog do odredišnog senzorskog čvora u okviru bežične senzorske mreže.

Transportni nivo: Obezbeđuju pouzdane transportne mehanizme koji su u stanju da primene jedinstvene karakteristike *WSN* paradigme.

Pored analize većine pomenutih zadataka, u nastavku ćemo se uglavnom baviti upotrebom *FEC* mehanizma koji se primenjuju na relativno kratak niz bitova, čijim korišćenjem prevazilazimo velika kašnjenja nastala usled primene *ARQ* mehanizma i koji nam omogućavaju rekonstrukciju narušenih podataka na prijemnoj strani bez, ili u ekstremnim slučajevima $< 0.01\%$, sa minimalnom paketnom retransmisijom.

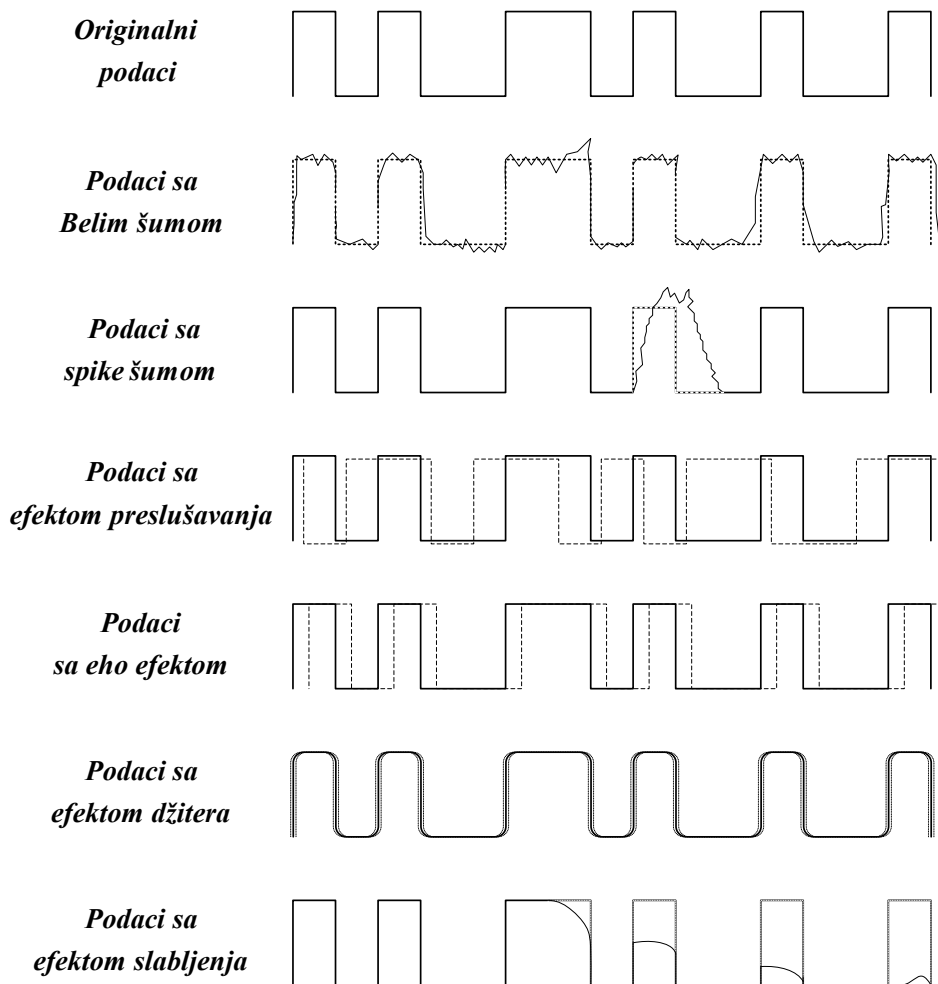
9.2 Kodovi za ispravljanje grešaka u bežičnim senzorskim mrežama

Pre nego što pređemo na konkretan opis predložene *FEC* tehnike, neophodno je da damo kratak pregled izvora problema koji su u vezi pojavom grešaka kod bežičnih senzorskih mreža. Nakon toga, kako bi svaki čitalac bio u stanju da razume predloženu tehniku kodiranja podataka neophodno je da se uputi u osnove iz teorije kodiranja što je i učinjeno u drugom delu Odeljka. Za detaljnije informacije o kodiranju, posebno u vezi sa matematičkim konceptima, pogledajte [36-37].

Kad god se podaci prenose preko komunikacionog kanala, postoji šansa da se jedan ili više bitova "flipuju" (promene svoje stanje). Pojava takvih nekorektnih vrednosti naziva se greška. Kodovi za korekciju grešaka ubacuju pariti bitove provere u sekvencu poruke koji će, na odgovarajući način u zavisnosti od tipa koda, biti korišćeni. Ovi pariti bitovi omogućavaju

prijemniku da ispravi greške u sekvenci poruka ako ih ima, a ima ih usled šuma ili interference u toku prenosa kroz kanal. Tipična metrika za kvantifikovanje pouzdanosti je stopa pogrešnih bitova (*Bit Error Rate - BER*).

Kada se podaci prenose putem nekog medijuma za prenos neminovnost je da se to radi u prisustvu izvora grešaka. Čak i ako se u toku prenosa koriste sve moguće mere za smanjenje grešaka, greška se neprestano pojavljuje i počinje da ometa prenos podataka. Naravno, cilj komunikacione mreže je da dostavi tačne podatke. U tom cilju, detekcija grešaka se primenjuje uglavnom u sloju podataka tj. na nivou veze, ali se takođe vrši i na drugim slojevima. Međutim najčešće mesto gde imamo pojavu grešaka je i dalje nivo veze. U ovom odeljku su opisani mogući i uobičajeni oblici grešaka na ovom nivou i prikazani su na Sl. 9-1.



Sl. 9-1 Uobičajeni oblici grešaka podataka na nivou veze

- ✓ Beli ili *Gauss*-ov šum je kontinualne prirode i zavisi od temperature medijuma za prenos. Beli šum može da promeni sadržaj podataka, kao što je to prikazano na Sl. 9-1. Može da se eliminiše prenosom signala sa šumom kroz skup filtara.
- ✓ *Spike* šum nije kontinualne prirode, ali može potpuno da uništi podatke tako da ne mogu da se oporave.
- ✓ Međusobno preslušavanje je unakrsna (uzajamna) aktivnost preslušavanja između dva aktivna linka. Preslušavanje može da bude električno, kao što je između parova upredenih parica, ili elektromagnetno, u slučaju kada se neželjeni signali pokupe antenom
- ✓ Eho je pojava koja nastaje od uticajem refleksije prenošenog signala. Naime, signal može da se reflektuje od kraja kabla i vrati kroz žicu, interferirajući sa originalnim signalom. Ova greška se događa u *LAN* topologijama tipa magistrala ili bus. Jednosmerni filter, poznat kao *echo canceler*, može se realizovati povezati na link sa ciljem da eliminiše efekat ehoa.
- ✓ Džiter je vremenska nepravilnost koja se javlja na rastuću ili opadajuću ivicu signala uzrokujući grešku. Džiter može da bude rezultat elektromagnetnih smetnji ili uzajamnog preslušavanja i može se smanjiti pravilnom sistemskom zaštitom.
- ✓ Slabljenje bita je gubitak snage bita kako on putuje kroz medijum. Ova vrsta greške se može eliminisati upotrebom pojačavača i repetitora za digitalne sisteme.

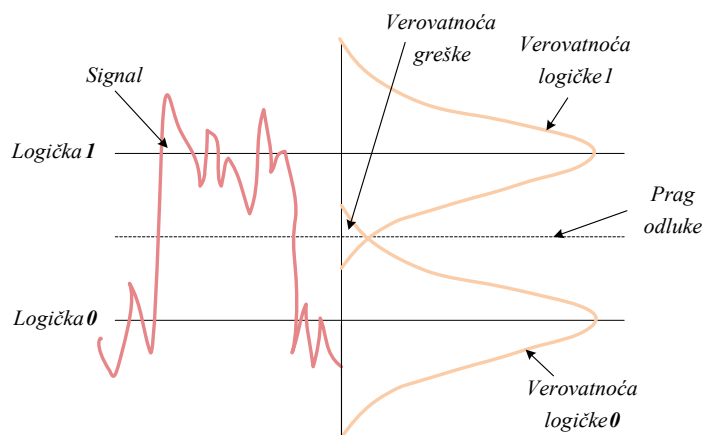
Dakle, nijedna veza nije imuna na greške. Tako na primer, medijum upredene parice, zasnovan na bakarnoj žici, je izložen različitim vrstama interferencije i šuma. Satelitske, mikrotalasne i radio mreže su takođe sklone šumu, interferenciji i međusobnom preslušavanju. Čak i fiber optički kabl može da primi podatke sa greškama iako je verovatnoća pojave grešaka, kod ovog tipa medijuma za prenos, vrlo niska.

Greška na nivou bita se pojavljuje kada električni prijemnik donosi pogrešnu odluku o logičkom nivou bita. Mnogi faktori mogu doprineti *BER*-u, a tu spadaju snaga signala, šum, džiter i *EMI* usled emisije zračenja ili preslušavanja. Prisustvo šuma na mestu gde prijemnik odlučuje o logičkom nivou bita može da dovede do toga da prijemnik pogrešno interpretira taj bit. Loš odnos signal-šum i loš odnos ekstinkcije (prosečan nivo snage bita logičke jedinice (1) u poređenju sa prosečnim nivoom snage bita logičke nule (0)) olakšavaju slučajnom šumu da uzrokuje grešku bita, vidi sliku Sl. 9-2.

Džiter takođe može da bude uzrok greške u bitovima. Promena takta može dovesti do toga da prijemnik izgubi sinhronizaciju sa dolaznim bitovima. Usponska ili silazna ivica koja se javlja u trenutku kada je prijemnik ne očekuje može takođe da uslovi da prijemnik nepravilno intepretira logički nivo. Greške prouzrokovane džiterom mogu se javiti periodično ili samo u određenim obrascima bitova, zavisno od toga šta uzrokuje pojavu džitera.

Emisija zračenja od strane oscilatora kao što su oni u prekidačkim logikama bloka napajanja i mikrop procesorski sistemski takt mogu da izazovu anomalije koje prouzrokuju greške bitova. Osnovna karakteristika im je da se javljaju u periodičnim intervalima.

Preslušavanje se javlja kada se deo signala napajanja prenese na susedne linije koje se koriste za prenos signala i češća je pojava u sistemima prenosa električne prirode. Interferencija može da proizvede promenu nivoa koja može da prouzrokuje da prijemnik pogrešno protumači pristigli bit.



SI. 9-2 Način odlučivanja kod prenosa signala sa prisustvom šuma

Vratimo se sada metodama za utvrđivanje postojanja grešaka u prenosu sa ciljem ostvarivanja pouzdanog prenosa u bežičnim senzorskim mrežama. Kao što je već istaknuto u uvodnom delu disertacije u kontroli grešaka baziranoj na *ARQ* metodi, paket se retransmituje ako se utvrdi da ima grešaka. Takvi paketi se ponovo prenose sve dok se ne prime bez grešaka. Detekcija (otkrivanje) grešaka se obično realizuje kroz proveru ciklične redundanse (*Cycle Redundance Check - CRC*). Sa druge strane, *FEC* ili kanalno kodiranje je metoda koja se koristi za povećanje performansi kontrole grešaka. Ovo se postiže korišćenjem kodova za korekciju (ispravljanje) grešaka (*Error Correcting Codes - ECC*) koji dodaju redundantnost paketu što omogućava da se izvestan broj pogrešno prenetih bitova detektuju, a neki od njih i koriguju na strani prijemnika. Ovakve konstatacije je neophodno obrazložiti i opravdati dubljom analizom tehnika kodiranja koja upravo sledi.

9.3 Osnovni koncepti kodiranja

Kada se digitalni podaci prenose preko kanala u kome je prisutan šum uvek postoji šansa da primljeni podaci sadrže greške. Korisnik generalno utvrđuje stopu grešaka iznad koje primljeni podaci nisu upotrebljivi. U slučaju da primljeni podaci ne zadovolje zahtevanu stopu grešaka, često se koriste kodovi za korekciju (ispravljanje) grešaka pomoću kojih se smanjuju greške prema aplikaciji do nivoa na kojem se mogu tolerisati. U poslednjih nekoliko godina korišćenje kodova za korekciju grešaka postalo je široko rasprostranjeno u cilju rešavanja ovakvih problema.

Od *Shannon*-ovih radova do današnjih dana postalo je jasno da stvarna granica komunikacione brzine nije postavljena kapacitetom kanala već cenom implementacije šema kodiranja. Ograničenja koja su posledica cene javljaju se na polju brzina koje su znatno ispod kapaciteta kanala. U poslednjih nekoliko godina mnogo istraživanja je usmereno na pronalaženju efikasnih i praktičnih kodnih šema za različite tipove kanala sa šumom. Većina napretka u pronalaženju praktičnih šema nastala je u poslednjih dvadesetak godina, a sada je jasno da kodiranje, u mnogim aplikacijama, može da obezbedi značajna poboljšanja u performansama. Postoje brojne aplikacije u kojima je ugrađena logika za kodiranje i prilično uspešno se koristi. Šira primena je usledila kao posledica drastičnog smanjenja troškova, sa jedne strane, i veličine elektronskih uređaja, sa druge strane.

9.4 Osnovni principi kodiranja

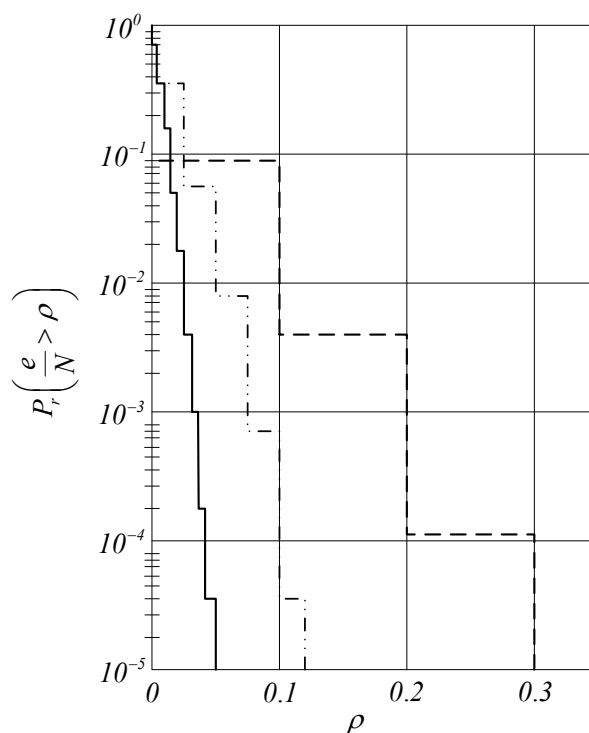
Kodiranje zarad korekcije greške(aka) je u suštini tehnika obrade signala koja se koristi za poboljšanje pouzdanosti komunikacije u digitalnim kanalima. Iako pojedine šeme kodiranja imaju mnogo različitih formi i imaju svoje poreklo u različitim matematičkim disciplinama, sve one, u osnovi, poseduju dve zajedničke komponente ili osobine.

Jedna je upotreba redundantnosti. Naime, sve kodirane digitalne poruke uvek sadrže dodatne ili suvišne (redundantne) simbole. Izbor treba da bude takav da naglasi jedinstvenost svake poruke, tj. da je malo verovatno da će bilo kakav poremećaj u kanalu korumpirati simbole u poruci i time uništiti njihovu jedinstvenost.

Druga je usrednjavanje šuma. Efekat usrednjavanja se ostvaruje tako što redundantni simboli su u direktnoj korelaciji sa rasponom nekoliko informacionih simbola. Neki dragocen uvid u proces kodiranja može se dobiti zasebnom analizom svake od navedenih komponenti.

Prvo, razmotrimo binarni komunikacioni kanal koji je izložen neželjenim smetnjama tako da se greške pojavljuju nezavisno na svakom simbolu i sa prosečnom stopom greške $P_e = 0.01$. Ako bi sa ovako postavljenog kanala ispitali blok podataka sastavljen od 10 simbola bilo bi izuzetno teško da identifikujemo koji su simboli u bloku pogrešni. Sa druge strane, moglo bi da se kaže da blok sadrži tri ili nekoliko grešaka ili samo da bude pogrešan dva puta na svakih milion blokova. Čak štaviše, težina ovakve izjave dobija sve više na značenju kako se veličina bloka povećava. Naime, ne samo da se deo simbola koji su pogrešni u bloku približava prosečnoj stopi greške u kanalu, nego što je još važnije delovi blokova koji sadrže broj grešaka koji se značajno razlikuje od proseka, postaje vrlo mali.

Jednostavnim proračunima može se steći osećaj u kojoj meri je ova izjava tačna. Pretpostavimo da za isti kanala izračunamo verovatnoću da će pogrešan deo simbola premašiti vrednost p i grafički je prezentovati u funkciji nekoliko različitih dužina bloka. Rezultat je prikazan na Sl. 9-3. Obratimo pažnju da "kriva" progresivno napreduje kako se dužina bloka povećava i da se približava step funkciji koja se javlja na abscisi 0.01 .



Sl. 9-3 Verovatnoća da deo simbola sa greškom, e/N , u bloku dužine N prelazi vrednost ρ za zadato $P_e = 0.01$ [66]

Kriva na Sl. 9-3 pokazuje da, ako smo spremni da obradimo simbole u blokovima, a ne jedan po jedan, moguće je smanjiti ukupnu stopu greške. Ono što je neophodno je postojanje šeme "kodiranja" koja je u stanju da toleriše prisustvo greške u nekom delu simbola u bloku bez uništavanja "jedinstvenosti" poruke koju blok prenosi. Slika eksplicitno pokazuje, za nekoliko različitih dužina bloka, koji tačno deo grešaka mora biti korigovan kako bi postigli određenu stopu greške u blokovima. Takođe ukazuje i na to da za fiksnu stopu grešaka deo grešaka koji mora biti korigovan (ispravljen) opada sa povećanjem dužine bloka. Ova razmatranja i analiza ukazuju na potencijal za poboljšanje performansi koji se dobija kroz proces usrednjavanja šuma kao i da ovaj potencijal postaje veći sa povećanjem dužine bloka. Dakle, zaključak je da bi duži blok kodovi trebali biti "efikasniji" od kratkih blok kodova.

Nakon uspostavljanja potrebnih uslova koji nam pružaju mogućnost da ispravimo greške u simbolu sledeće logično pitanje je: kako to možemo da postignemo? Ključ je u redundantnosti.

Neko logičko razmišljanje je da će greške u poruci, koja je predstavljena nizom od n binarnih simbola, biti ispravljene samo pod uslovom ako se ne dozvoli korišćenje svih 2^n mogućih sekvenci kao legitimnih poruka. Naime, ako se svaka moguća primljena sekvenca od n simbola prihvata kao legitimna poruka, onda ne bi imali osnovu za utvrđivanje da li je jedna sekvenca validnija od bilo koje druge sekvence. Ako se sa ovakvim razmišljanjem nastavi dalje, postaje jasno da ako neko želi da ispravi sve obrasce (šablone) od t ili manje grešaka potrebno je i dovoljno je da se svaka legitimna sekvenca poruke razlikuje od svake druge legitimne poruke u najmanje $2t + 1$ pozicija.

Na primer, ako se želi da se isprave sve jednostruke i dvostruke greške u simbolu, neophodno je da se svi parovi sekvenci poruka razlikuju u najmanje pet simbola. Bilo koja primljena sekvenca koja sadrži dve greške i prema tome, razlikuje se od ispravne sekvence na tačno dve pozicije, uvek će se od drugih validnih sekvenci poruka razlikovati na najmanje tri pozicije. Dakle mi se pozivamo na broj pozicija u kojima se dve sekvence jedna od druge razlikuju i označavamo ga kao *Hamming*-ova distanca, d , između dve sekvence. Najmanja vrednost d za sve parove kodnih sekvenci naziva se minimalna distanca koda i označava se sa d_{\min} . Pošto d_{\min} mora uvek da bude za jedan više od dvostruke vrednosti broja grešaka koje treba ispraviti, može se napisati kao:

$$t = \left\lfloor \frac{d_{\min} - 1}{2} \right\rfloor \quad (9.1)$$

gde je $\lfloor \cdot \rfloor$ simbol koji označava manje celi deo. Parametar t označava da se sve kombinacije od t ili manje grešaka u bilo kojoj primljenoj sekvenci mogu ispraviti.

9.5 Blok kodovi i konvolucioni kodovi

Među široko rasprostranjenim blok kodovima u praktičnim realizacijama bežičnih senzorskih mreža spadaju *CRC*, *Hamming*, *SEC-DED* i *erasure* kodovi. *Hamming* kodovi se odlikuju vrlo jednostavnom strukturom i visokom stopom koda i s toga se koriste u *FEC* šemama kada su greške slučajne prirode i njihova stopa grešaka je niska. Ukratko, za bilo koji pozitivan ceo broj $m \geq 3$ moguće je kreirati *Hamming*-ov kod sa sledećim parametrima:

- 1) dužina koda $n = 2^m - 1$;
- 2) broj informacionih bitova $k = 2^m - m - 1$;
- 3) broj bitova za proveru parnosti $n - k = m$;
- 4) sposobnost korekcije grešaka $t = 1$ ($d_{\min} = 3$)

Pošto je *Hamming*-ova distanca tri kod je u stanju da ispravi sve jednostruke greške. *SEC-DED* kod je konstruisan proširenjem *Hamming*-ovog koda dodavanjem dodatnog (ekstra) bita parnosti. *Hamming*-ova distanca *SEC-DED* koda je 4, što pruža mogućnost dekoderu da prepozna razliku između jednostrukih i dvostrukih grešaka. Zbog toga dekodirer može da detektuje i ispravi sve jednostruke greške i istovremeno detektuje, ali ne i da koriguje sve dvostruke greške.

CRC se koristi kao kod za detekciju greške, on ukazuje da su greške prisutne u nizu bitova, ali ne i gde su pozicije tih bitova niti kako da ih ispraviti. *CRC* se najčešće koristi za detekciju grešaka u sistemu koji koristi *ARQ* tehniku.

Erasure kod predstavlja generalizaciju koda parnosti i primenjuje se od strane izvornog čvora. U *erasure* kodovima k blokova izvornog podatka se kodiraju od strane čvora koji šalje kako bi se generisala n blokova kodiranih podataka na način da bilo koji podskup od k kodiranih blokova bude dovoljan da rekonstruiše izvorne podatke. Takav kod se naziva (n, k) kod i omogućava prijemniku da oporavi do $n - k$ izgubljenih blokova u grupi od n kodiranih blokova. Toliki gubitak blokova uslovljen je pojavom grešaka koje su *burst* tipa i u tom slučaju *erasure* kodovi su mnogo efikasniji u smanjenju gubitaka blokova.

9.6 Efikasnost CRC tehnike kodiranja

Metoda *CRC* je prilično dobro proučena, dokazana i primenjena u praksi. Međutim potrebna je analiza kako bi se odgovorilo na pitanje da li je prijemnik uvek u stanju da detektuje oštećene okvire. U tom cilju, razmotrimo generator polinom uopštenog oblika

$$x^{g-1} + x^{g-2} + \dots + 1 \quad (9.2)$$

sa g članova. Očigledno je da je $g-1$ najveći stepen generator polinoma. Sa druge strane, neka je n dužina *burst* greške koja se javlja u primljenoj poruci. U slučaju da je veličina *burst* greške $n < g$ detekcija je 100%. Za sve druge slučajeve, članovi polinoma između x^{g-1} i 1 definišu koji su bitovi pogrešni. Pošto takvih članova ima $g-2$, postoji 2^{g-2} mogućih kombinacija pogrešnih bitova. S obzirom da se sve kombinacije mogu pojaviti sa jednakom verovatnoćom, postoji šansa od $1/2^{g-2}$ da se jedna od kombinacija podudara tačno sa članovima koji postoje u polinomu što će dovesti do tumačenja da je poruka korektno primljena. Dakle, ovo je verovatnoća da pogrešan bit postaje neotkriven. Verovatnoća da se uhvati *burst* greška kroz *CRC* za sve slučajeve je:

$$p = \begin{cases} 1 & \text{za } n < g \\ 1 - \left(\frac{1}{2}\right)^{(g-2)} & \text{za } n = g \\ 1 - \left(\frac{1}{2}\right)^{(g-1)} & \text{za } n > g \end{cases} \quad (9.3)$$

Informacije radi, razmotrićemo sledeći primer primene *CRC* na kome je baziran metod kodiranja podataka u primopredajniku *CC110L*. Naime, u ovom primeru analiziraćemo komunikacioni standard, koji je primenjen u pomenutom primopredajniku, zasnovan na *IBM CRC16* [67] generator polinomu definisanom kao $x^{16} + x^{15} + x^2 + 1$. Za ovaj standard najveći stepen generator polinoma je $g-1=16$, što ima za posledicu da za slučaj da je greška manja od $g=17$ bitova u dužini, primenjena tehnika je u stanju da detektuje postojanje *burst* greške. Pod pretpostavkom da je $n = g = 17$ iz jednačine (9.3) dobijamo

$$p = 1 - \left(\frac{1}{2}\right)^{(17-2)} = 1 - 0.000030517578125 = 0.999969482421875$$

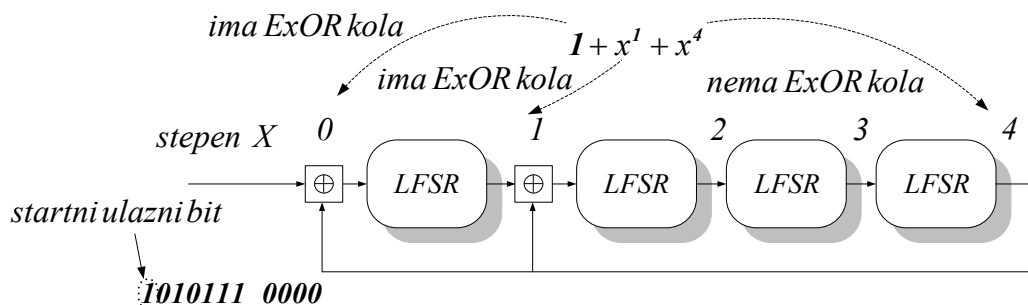
što je približno 1.

Za slučaj da broj grešaka n veći od $g=17$ bitova u dužini na osnovu jednačine (9.3) verovatnoća da će *CRC* detektovati postojanje *burst* greške iznosi:

$$p = 1 - \left(\frac{1}{2}\right)^{(17-1)} = 1 - 0.0000152587890625 = 0.9999847412109375$$

9.7 Implementacija *CRC* procesne jedinice

Hardver u kombinaciji sa softverom predstavlja optimalno rešenje jer može veoma brzo da realizuje algoritam tj. ostvari proces deljenja. Primer osnovnog hardvera koji se koristi u primeni izračunavanja *CRC*-a prikazan je na Sl. 9-4. Hardver uključuje jednostavni registar koji implementira *CRC* proces koji se realizuje pomoću generator polinoma $x^4 + x + 1$. Osim prvog člana (x^4) za svaki od narednih članova određenog stepena, koji postoje u generator polinomu, uključeno je *Ex-OR* kolo, kao što je prikazano na Sl. 9-4. Napomenimo da se notacija vrednosti generator polinoma 10011 sada pojavljuje u formi hardvera prikazanog kombinacijom 1-bitnih pomeračkih registara i ekskluzivnih *OR* kola. Prema tome, na ovaj način, koristeći se blok prikazom primene *CRC* algoritma, u stanju smo da uočimo postojanje članova polinoma tj. polinomski oblik generator polinoma.



Sl. 9-4 Hardverska implementacija *CRC* algoritma

Napomena: U datom primeru msb bit iz povorke informacionih bitova se uvodi prvi u CRC hardver

Inicijalno stanje svih registara jesu 0 bitovi. Svi bitovi podataka 1010111, nakon proširenja sekvencom bitova 0000, počevši od bita najveće težine dolaze sa leve strane i uvode se u hardver, a 1-bitni pomerački registar pomera bitove udesno svaki put kada se unese novi bit. Krajni desni bit u sekciji pomeračkih registara se vraća nazad ka izabranim tačkama, tačke u kojima postoji *ExOR* kolo. U ovim tačkama, vrednost bita se dobija primenom *ExOR* operacije između bita koji se vraća, u datom trenutku, kroz povratnu granu i bita koji se, iz prethodne sekcije pomeračkog registra, pomera

u desno. Tako generisan bit se uvodi u narednu sekciju pomeračkog registra. Kada se svi bitovi podatka zajedno sa bitovima proširenja, u određenom broju ciklusa, provuku kroz hardver *CRC*, sadržaj registara mora da bude tačno onakav kakav je ostatak generisan u procesu matematičke primene algoritma prikazanog na Sl. 9-5.

$$\begin{array}{r}
 \begin{array}{c}
 \textit{Delilac} \\
 \textit{(generator polinom)}
 \end{array} \\
 1 \ 0 \ 0 \ 1 \ 0
 \end{array}
 \begin{array}{c}
 \begin{array}{c}
 \textit{Količnik} \\
 1 \ 0 \ 1 \ 1 \ 1 \ 0 \ 0
 \end{array} \\
 \hline
 1 \ 0 \ 1 \ 0 \ 1 \ 1 \ 1 \ 0 \ 0 \ 0 \ 0 \\
 1 \ 0 \ 0 \ 1 \ 0 \ \downarrow \ \downarrow \ \downarrow \ \downarrow \ \downarrow \\
 \hline
 0 \ 1 \ 1 \ 1 \ 1 \ \downarrow \\
 0 \ 0 \ 0 \ 0 \ 0 \ \downarrow \\
 \hline
 1 \ 1 \ 1 \ 1 \ 1 \ \downarrow \\
 1 \ 0 \ 0 \ 1 \ 0 \ \downarrow \\
 \hline
 1 \ 1 \ 0 \ 1 \ 0 \ \downarrow \\
 1 \ 0 \ 0 \ 1 \ 0 \ \downarrow \\
 \hline
 1 \ 0 \ 0 \ 0 \ 0 \ \downarrow \\
 1 \ 0 \ 0 \ 1 \ 0 \ \downarrow \\
 \hline
 0 \ 0 \ 1 \ 0 \ 0 \ \downarrow \\
 0 \ 0 \ 0 \ 0 \ 0 \ \downarrow \\
 \hline
 0 \ 1 \ 0 \ 0 \ 0 \\
 0 \ 0 \ 0 \ 0 \ 0 \\
 \hline
 1 \ 0 \ 0 \ 0
 \end{array}
 \begin{array}{c}
 \textit{Deljenik} \\
 \textit{(podaci sa dodatim 0-ma)}
 \end{array}
 \end{array}
 \begin{array}{c}
 \textit{Ostatak} \\
 \textit{(CRC)}
 \end{array}$$

Sl. 9-5 Deljenje po modulu-2 za proces primene *CRC* algoritma realizovano na strani predajnika

9.8 Korekcija greške na strani prijemnika

Korekcija greške na strani prijemnika je tehnika koja omogućava prijemniku da ispravi određen broj grešaka u primljenoj poruci. Ovo se postiže dodavanjem redundantnosti poruci podataka kroz blok *FEC* enkodera na strani predajnika u skladu sa određenim unapred definisanim pravilima. *FEC* dekoder, na strani prijemnika, koristi poznavanje ovih pravila kako bi identifikovao i, ako je moguće, korigovao bilo koje greške koje su se pojavile.

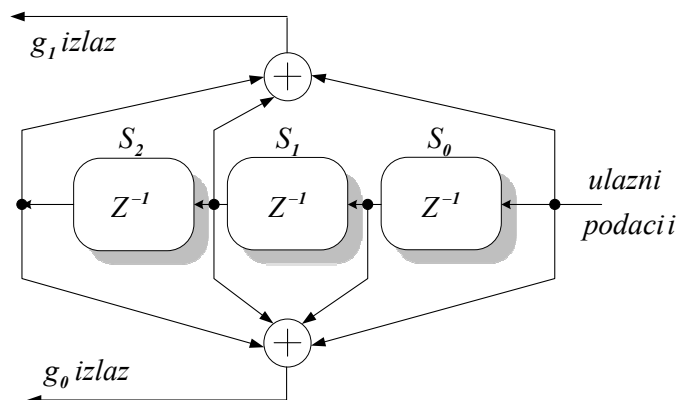
Uopšteno govoreći, postoje dve osnovne klase *FEC* kodova:

- ✓ linearni blok kodovi (*BCH*, *Reed-Solomon*, itd.) i
- ✓ konvolucionni kodovi.

Linearni blok koder (n, k) uzima k -bitni blok poruke podataka i dodaje mu $n-k$ redundantne bitove koji su algebarski povezani sa k -bitovima poruke stvarajući tako n -bitni kodni blok. Jasno je da postoje 2^k validne kodne reči, što je daleko manje od 2^n mogućih kodnih reči, a dobar linearni blok kod je onaj u kojem je *minimalna distanca* u oznaci d_{\min} , tj. minimalni broj bitova koji mora biti promenjen da bi se od jedne kodne reči dobila druga kodna reč, maksimizirana. Da bismo bili u stanju da korigujemo (ispravimo) e pogrešnih bitova neophodno je da važi $d_{\min} > 2e$, tj. nakon e pogrešnih bitova ispravna kodna reč je i dalje ona sa najmanjom distancom do primljene kodne reči. Bezdimenzionalni odnos $r = k/n$ se naziva kodni odnos ili stopa koda (*code rate*).

Konvolucionni enkoder je u osnovi konačni automat (*finite state machine*) sa k -bitnim ulazom i n -bitnim izlazom, tako da je $n > k$ i unutrašnjom M -bitnom memorijom. Važan parameter konvolucionog enkodera je njegova ograničena dužina (*constraint length*) $L = M + 1$ koja specificira preko koliko uzastopnih n -bitnih izlaznih perioda k -bitna ulazna vrednost utiče na izlaz. *FSM* je tako koncipiran da svaki unos sekvence bitova poruke rezultira kodiranim izlaznim nizom bitova koja maksimizira minimalnu distancu u odnosu na onu koja bi se generisala za bilo koju drugu sekvencu ulaznih poruka. Konvoluciono dekodiranje se obično realizuje primenom *Viterbi* algoritma, koji konceptualno, upoređuje primljenu sekvencu sa kodiranom verzijom svih mogućih ulaznih sekvenci enkodera (tabela pretraživanja) i zadržava tabulator o tome koliko je blizu poklapanju sa nekom od sekvenci. Periodično *Viterbi* algoritam prati kroz svoju memoriju i izlazni deo ulazne sekvence, i traži koja, kada je kodirana, najbliže odgovara primljenoj sekvenci kodova.

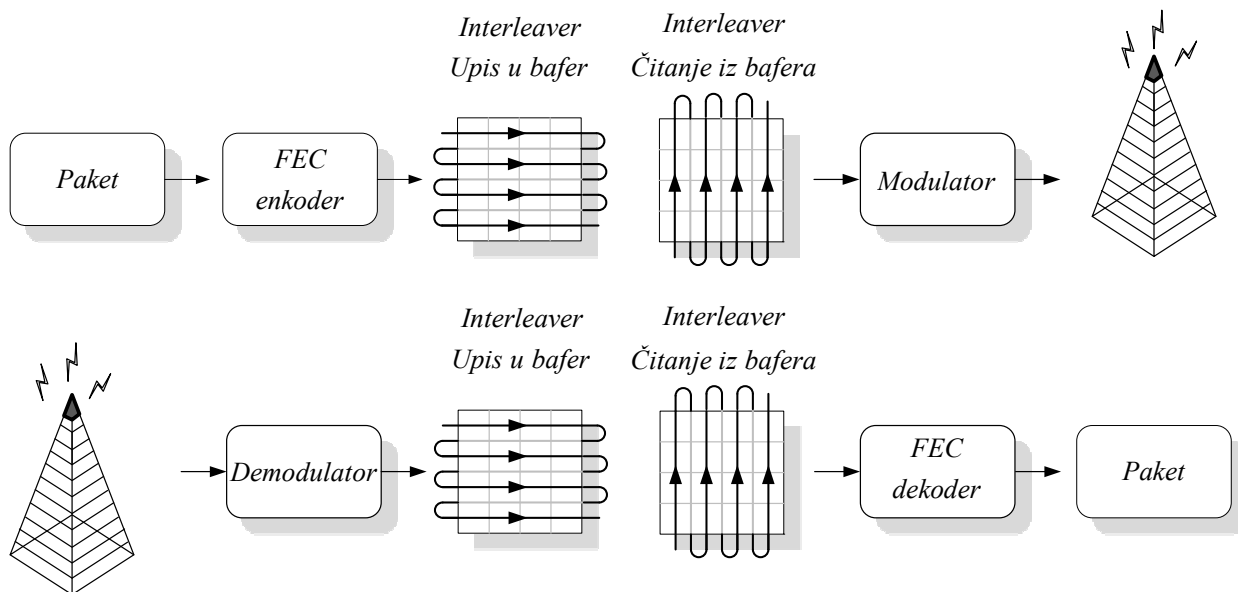
Svi primopredajnici serije *CC* [68], koji koriste konvolucione kodove imaju kodnu brzinu $r = 1/2$ konvolucionog, ne-rekurzivnog enkodera sa ograničenjem dužine $L = 4$ ($M = 3$) implementiranim kao što je prikazano na Sl. 9-6.



Sl. 9-6 Implementacija konvolucionog kodiranja u primopredajniku CC serije

Svaki ulazni bit se kodira sa dva izlazna bita, čime se udvostručuje količina podataka koji se moraju preneti. Dakle, ako se koristi ista brzina prenosa podataka (bitska brzina), moguć je prijem bez grešaka sa manjom jačinom predajnog signala – time se efektivno povećava opseg radio-talasa ili se smanjuje potrošnja energije za fiksni domet. Ako se zahteva izvorna brzina prenosa podataka neophodno je brzina predaje udvostručiti što se može ostvariti ili povećanjem tj. udvostručavanjem brzine modulacije ili primenom *2-ary* ili *4-ary* modulacionog formata. Očigledno, uvođenje ovih izmena će povećati broj grešaka po bitu u primljenoj kodiranoj sekvenci, ali korekcija grešaka u dekomoderu obezbeđuje da dekodirana sekvenca poruka sadrži manje pogrešnih bitova nego ako je sekvenca poruka preneti bez kodiranja.

Konvoluciono kodiranje daje najbolje rezultate ako su pogrešni bitovi ravnomerno (ili bar slučajno) raspoređeni kroz primljenu kodiranu sekvencu. Naravno, usled neminovnosti pojave grešaka tipa *burst* kako zbog prirode mnogih izvora radio interferencija tako i zbog karakteristika demodulatora, verovatnije je da će se pogrešni bitovi pojaviti zajedno jedan za drugim. Da bi se izborili sa ovim problemom vrši se tzv. preklapanje (*interliving*) kodiranih podataka nakon procesa kodiranja na predajnoj strani i de-preklapanje (*de-interliving*) pre procesa dekodiranja na prijemnoj strani, što je prikazano na Sl. 9-7. Svrha primene ovog procesa je da se obezbedi da susedni simboli u kodnoj sekvenci razmaknu (udalje) u prenosnoj sekvenci, tako da se bilo koje grupe pogrešnih bitova u primljenoj sekvenci "rašire" i mnogo ravnomernije rasporede pomoću *de-interliver-a*, dozvoljavajući dekomoderu da radi pod optimalnim uslovima.



Sl. 9-7 Proces *interliving* i *deinterliving* na predajnoj i prijemnoj strani

Koliko pogrešnih bitova *FEC* može ispraviti?

Konvolucioni kod upotrebljen u, na primer, *CC11xx* seriji čipova ima maksimalnu tzv. slobodnu distancu $d_{free} = 6$ bitova. Ovo znači da će promena bilo kog bita u sekvenci poruke promeniti najmanje 6 bitova u kodiranom izlaznom nizu. U skladu sa tim, zahtevaju se najmanje tri pogrešna bita u primljenoj kodnoj sekvenci pre bilo koje druge sekvence poruke da bi jedna ispravna bila verovatna ili verovatnija. Sposobnost da ispravimo dva pogrešna bita u celoj kodiranoj sekvenci možda neće zvučati mnogo. Međutim zbog nemogućnosti preciznijeg predviđanja koliko pogrešnih bitova možemo da ispravimo korišćenjem konvolucionih koda faktor dobrote tj. efikasnost datog koda koji se obično povezuje sa konvolucionim kodovima je njegova asimptotska dobit kodiranja (*asymptotic coding gain*) odnosno smanjenje (redukovanje) *SNR* primljenog signala koji daje ekvivalentan *BER* kao u slučaju nekodiranog signala. Ova dobit može da se iskoristi za povećanje dometa (*range*) ili smanjenje snage predajnika uz postizanje istog rastojanja. Za binarno ulazni *AWGN* kanal (relevantan za *2-arne* modulacione formate na *CCxx00*) asimptotska kodna dobit (asimptotsko kodno pojačanje) je:

$$G_a = 10 \log_{10} (d_{free} r) \text{ dB} \quad (9.4)$$

gde je d_{free} slobodna distanca koda, r je stopa kodiranja. Korišćeni kod ($d_{free} = 6$, $r = 1/2$) ima asimptotsku dobit usled kodiranja od $4.8dB$, iako je postignuta dobit znatno manja za binarne modulacione formate (možda $2 - 3dB$).

9.9 Dobit usled kodiranja

Uobičajeni faktor dobrote za komunikacioni sistem je odnos energije po informacionom simbolu i spektralne gustine šuma (E_b/N_0) koji je potreban da bi se postigla zadata verovatnoća greške. Izraz "dobitak kodiranja" opisuje stepen poboljšanja koji se postiže kada se koristi određena šema kodiranja. Uobičajeni metod za utvrđivanje dobiti kodiranja je da se da grafik zavisnosti verovatnoće greške u funkciji E_b/N_0 za kodirane i nekodirane operacije, a zatim iščitavanje razlike potrebnih vrednosti odnosa E_b/N_0 pri određenoj stopi greške dođe do vrednosti dobiti usled kodiranja.

Dobitak kodiranja je koristan koncept samo u onim situacijama kada se značajno postiže povećanje performansi povećanjem snage. Na primer, ako postoji komunikacioni link koji je izložen pojavi slučajnih nestanka blokova u toku prenosa (*erasures*), onda u slučaju visokih odnosa signal šum postoji ograničenje ili gornja granica performansi koji se ne može prevladati povećanjem snage. Odgovarajuća šema kodiranja, međutim, može u značajnoj meri da smanji nivo ovog ograničenja ili da učini da potpuno nestane. Neko bi možda bio u iskušenju da kaže da je dobitak kodiranja beskonačna. Međutim, željena performansa nikada se ne bi mogla ostvariti bez kodiranja i takva izjava bi bila besmislena.

Asimptotska dobit kodiranja se ponekad koristi kao kvantitativna mera dobiti za određene kodove. Zavisi samo od kodne stope i minimalnog rastojanja i može da se definiše kako za nekvantizovan kanal tako i za binarni kvantizovani kanal.

Metode kontrole grešaka se primenjuju u prenosu digitalnih podataka u slučaju kada je stopa greške bez kontrole neprihvatljiva. Metod kontrole grešaka omogućava korisniku da odabere verovatnoću sa kojom je pojava greške u primljenoj poruci prihvatljiva, a zatim da prenese informacije sa najvećom mogućom dostupnom efikasnošću za kombinaciju tehnike modulacije, brzine prenosa podataka i okruženja u kojem će prenos biti ostvaren.

Hibridna kontrola greške je kombinacija koda za ispravljanje greške (na primer, blok kodovi) uz korišćenje retransmisione kontrole greške. Kao kombinacija dve procedure kodiranja, hibridna

kontrola greška ispoljava i prednosti i nedostatke i jedne i druge kombinacije. Konkretno, u okruženju slučajnih grešaka kod za ispravljanje grešaka na strani odredišta će obezbediti ispravljanje grešaka u okviru blok koda, a retransmisija će eliminisati preostale greške. U okruženju *burst* grešaka velike gustine, retransmisija će se koristiti za korekciju grešaka dok će kontrola greške na prijemnoj strani eliminisati greške u intervalima između pojave *burst*-ova. Ako se gustina grešaka tipa *burst* povećava na ekstremno visok nivo i *burst*-ovi postanu preterano dugi po broju bitova, ili ako se slučajne greške pojavljuju sa stopom grešaka na bit nivou koja je veća od praga (procenat korigovanih (ispravljenih) bitova greške) izvedenog iz primenjenog koda, sistem ulazi u kontinuiranu retransmisiju i srušiće se. U stanju otkaza, kod ispravlja nedovoljan broj grešaka, a dužina *burst*-a prevazilazi dužinu bloka uslovljavajući kontinuiranu retransmisiju. Mađutim, pošto će kodovi biraju iz određene porodice kodova, izborom moćnijih kodova može se sprečiti otkaz. Hibridni sistem za kontrolu grešaka predstavlja kombinaciju retransmisije i kodova za ispravljanje slučajnih i *burst* grešaka u simbolu koji se ponaša kao kontinuirani sistem.

9.10 Vreme kašnjenja usled kodiranja

Jedan od nedostataka korekcionih kodova je kašnjenje koje se uvodi u sistem usled kodiranja pomoću provere parnosti. Da bi se procenilo kašnjenje razmatraju se tri segmenta u kojima se može desiti kašnjenje (tj. kodiranje, prenos i dekodiranje). U procesu kodiranja, kada su informacioni bitovi dostupni od strane izvora počinje proračun simbola provere i postavljanje na izlaz komunikacionog interfejsa. Kada se poslednji informacioni bit postavi na izlaz svi bitovi provere su odmah dostupni. Stoga ne postoji kašnjenje u kodiranju sve dok koder može da procesira bitove brzinom koja je veća od brzine prenosa podataka po kanalu. Na strani dekodiranja, proračuni se ne mogu pokrenuti sve dok nisu svi simboli dostupni. Dakle, postoji kašnjenje od jednog bloka u prenosu. Za trenutno dostupan hardver, kalkulacije u dekodiranju zahtevaju najviše vreme u dužini trajanja jednog bloka. Prema tome, ukupno kašnjenje kao posledica kodiranja za neprekidan prenos podataka je dvostruko veće od dužine bloka.

Za slučaj da su blokovi poslani u razmaknutom diskontinualnom sistemu kašnjenje usled kodiranja bi bilo dvostruka dužina bloka po prenetom bloku. Ako se koristi takav sistem, ovo kašnjenje bi moglo da dominira u imeniocu izraza za proračun efikasnosti čime se drastično smanjuje efikasnost. Budući da bi stvarni sistem mogao imati različita kašnjenja do kašnjenja koje je jednako

dvostrukoj dužini bloka, pretpostavićemo najgori slučaj u našoj analizi performansi i koristiti maksimalnu vrednost kašnjenja u svim slučajevima.

9.11 Komunikacija u bežičnim senzorskim mrežama tolerantna na višestruke greške bitova pomoću dvo-dimenzionalnih kodova grešaka

U mnogim komunikacionim kanalima, koje susrećemo u bežičnim senzorskim mrežama za industrijski nadzor, greške se ne pojavljuju ni nezavisno nasumično niti u dobro definisanim pojedinačnim *burst*-ovima, već se pojavljuju na mešoviti način. Da bi se izborili sa pojavom mešovitih tipova grešaka, korišćenje kodova za korekciju slučajnih grešaka ili kodova za korekciju grešaka koje se pojavljuju u pojedinačnim *burst*-ovima će biti ili neefikasno ili neadekvatno.

Zbog toga je poželjno dizajnirati kodove koji su u stanju da isprave pojavu slučajno generisanih grešaka u prenosu kao i grešaka koje su tipa *burst* pojedinačno ili višestruko manifestovanih. U literaturi postoji nekoliko metoda konstruisanja kodova za istovremenu korekciju kako slučajno generisanih grešaka tako i grešaka tipa *burst* [36-37].

U nastavku disertacije cilj nam je da opravdamo uvođenje metode kodiranja zasnovane na korišćenju dvo-dimenzionalnih kodova kao racionalan i prihvatljiv izbor dizajna.

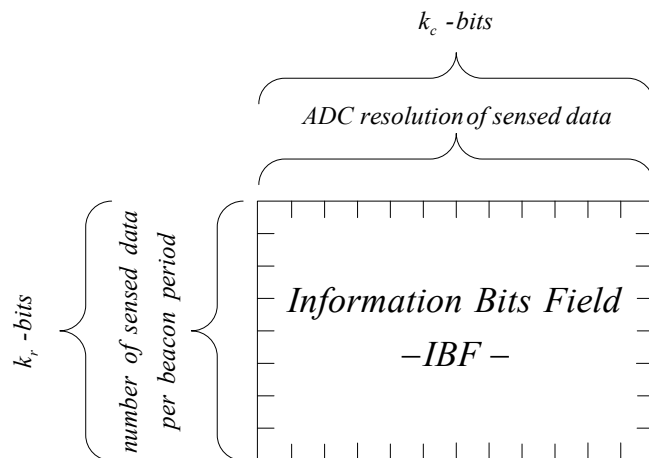
9.12 Dvo-dimenzionalni kodovi

Za početak predstavimo dvo-dimenzionalno (*2D*) kodiranje grešaka kao šemu za detekciju i korekciju grešaka koja koristi kodiranje grešaka u dve dimenzije (horizontalno i vertikalno). Napomenimo da će se kroz naredni tekst disertacije pojam *horizontalno* odnositi na detekciju/korekciju linija reči (po vrsti, terminologijom matrica), a *verikalno* će se odnositi na detekciju/korekciju bit-po-bit po liniji (po koloni, terminologijom matrica). U tekstu koji sledi opisaćemo *2D* kodove za upravljanje greškama koji kombinuju pravougaone kodove i *Hamming* produkt kodove (umnožak kodove).

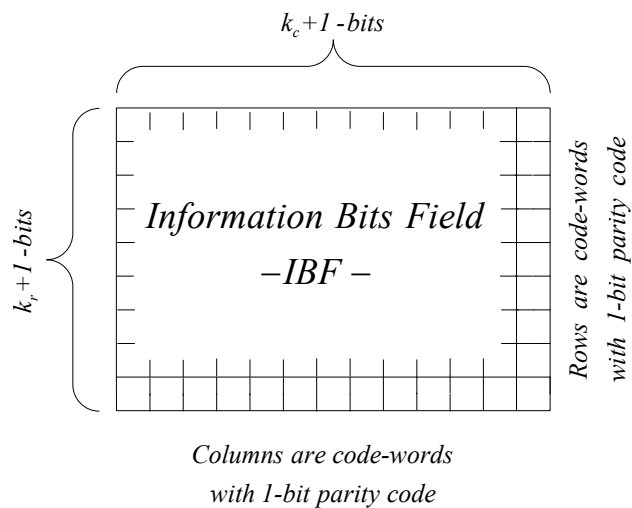
9.12.1 Pravougaoni kodovi

Tokom jednog potpunog *beacon* perioda *Rendezvous* protokola koga smo, kao što je više puta rečeno u prethodnom delu disertacije, usvojili kao tip protokola koji je implementiran u našu bežičnu senzorsku mrežu (a koga ćemo sa više detalja objasniti u narednom Poglavlju 10.) podaci prikupljeni od nekoliko senzorskih elemenata čuvaju se u linearnoj memoriji mikrokontrolera kao matrica bitova reda $k = k_r \times k_c$, tzv. informaciono bit polje koga ćemo obeležiti sa *IBF* (*Information Bit Field*), gde

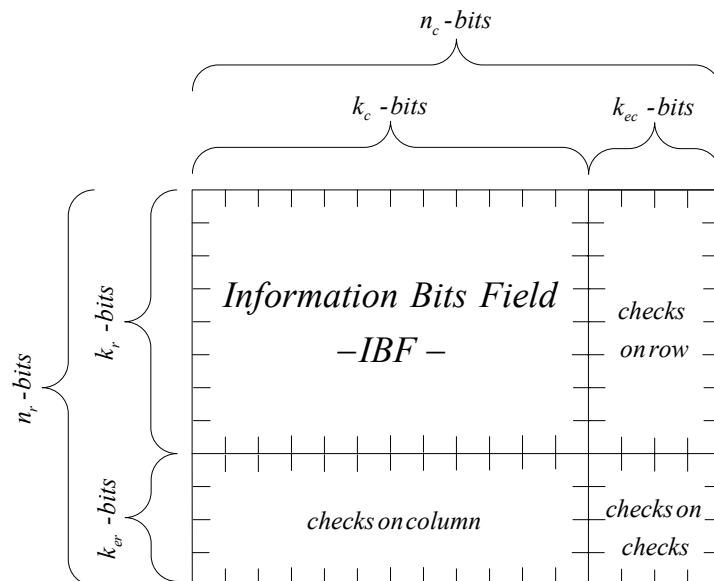
k_r odgovara broju senzorskih elemenata (analognih ili digitalnih senzora) u senzorskom čvoru, dakle podaci koji su senzovani, a k_c odgovara broju bitova po elementu (jednom senzoru, *ADC* rezolucija senzovanih podataka, vidi Sl. 9-8 a))



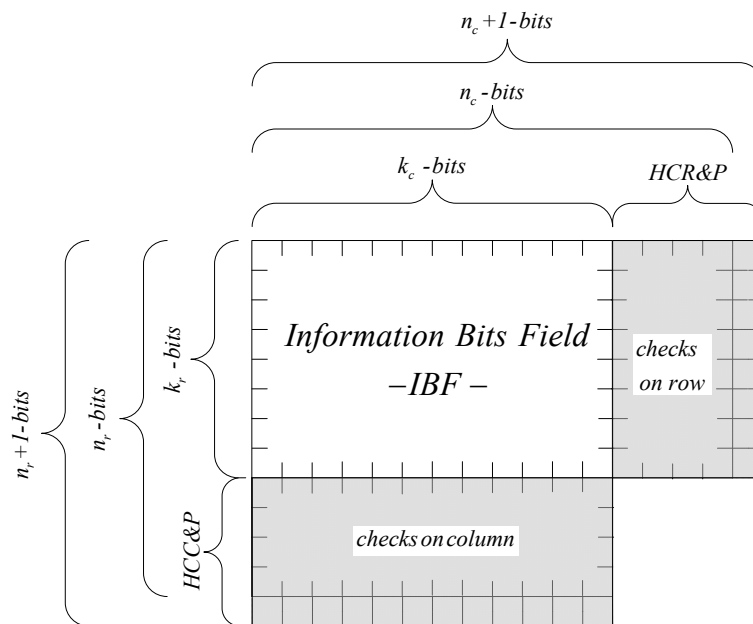
a)



b)



c)



d)

Sl. 9-8 a) Dvo-dimenzionalna (2D) organizacija informacionog bit polja *IBF*; b) Pravougaoni kod; c) *Hamming*-ov umnožak kod; i d) 2D *SEC-DED* kod

Za pravougaone kodove, informacioni bitovi su konceptualno raspoređeni u 2D polje. Svaka vrsta se proširuje tzv. detekcijom po vrsti tj. pariti bitom provere (horizontalni parnost) sa ciljem da detektuje jednostruke greške u vrsti. Slično tome, svaka kolona koristi tzv. detekciju po koloni tj.

pariti bitove provere (vertikalnu parnost) da bi otkrila jednostruke greške po koloni. Donji desni ugao može biti ili pariti provera kolone ili pariti provera vrste (vidi Sl. 9-8 b)), što rezultira uvećanjem polja na red $(k_r + 1) \times (k_c + 1)$. Dužina kodne reči n , za ovako koncipiran kod je jednaka $n = k_r \times k_c + k_r + k_c + 1$. Kodna stopa $R = k/n$ pravougaonih kodova se u tom slučaju može napisati kao $R = k/n = k_r \times k_c / ((k_r + 1) \times (k_c + 1))$. Pravougaoni kod može da koriguje šablon jednostrukih grešaka, jer je takva greška jedinstveno locirana u preseku detektovane greške vrste i detektovane greške kolone. Na primer, ako su dimenzije matrice $k_r = k_c = 11$, kod $(n, k) = (144, 121)$ može da koriguje jednostruke greške koja se nalazi bilo gde u polju od 144 bit pozicija.

Generalno minimalna *Hamming*-ova distanca svih dvo-dimenzionalnih kodova jednaka je proizvodu minimalne *Hamming*-ove distance po vrstama, d_{\min_row} i minimalne *Hamming*-ove distance po kolonama, d_{\min_column} , tj. $d_{\min_2D} = d_{\min_row} \times d_{\min_column}$. Prema tome, *2D* kodovi se ponekad nazivaju i produkt kodovi (umnožak kodovi). Sve vrednosti minimalne *Hamming*-ove distance, koje se odnose na dvodimenzionalne kodove, u daljem tekstu biće date shodno prethodno navedenom izrazu.

9.12.2 *Hamming* produkt kodovi

Hamming produkt kodovi proširuju $k_r \times k_c$ informaciono polje bitova (*IBF*) sa k_{er} i k_{ec} bitova (vidi Sl.9-8 b)). Na svaku vrstu dodaju se *Hamming*-ovi bitovi provere $k_{ec} = n_c - k_c$ za detekciju i korekciju pogrešnih bitova po vrsti kako bi se ispravile jednostruke bit greške po vrsti i na svaku kolonu dodaju se *Hamming*-ovi bitovi provere $k_{er} = n_r - k_r$ za detekciju i korekciju grešaka po koloni kako bi se ispravile jednostruke bit greške po koloni. Donji desni ugao mogu biti ili *Hamming*-ovi bitovi provere po koloni ili po vrsti sa istim svojstvom da mogu da isprave jednostruke bit greške. Zbog toga dužina kodne reči u bitovima, za informaciono polje bitova $k = k_r \times k_c$, iznosi $n = n_r \times n_c$, a stopa koda $R = k/n = (k_r \times k_c) / (n_r \times n_c)$ što je manje u odnosu na stopu koda pravougaonih kodova, čime je iskazana činjenica da su *Hamming* produkt kodovi redundantniji.

9.12.3 *Hamming* produkt kodovi sa dodatnim bitom parnosti – (*SEC-DED*)

Kod izbora koda, generalno, važno je razmotriti koje se vrste grešaka očekuju u sistemu koji bi trebalo zaštititi. *SEC-DED* kod pretpostavlja da postoje jednostruke bit greške, da je pojava

dvostrukih bit grešaka retka, a pojava trostrukih ili višestrukih bit grešaka je tako retka da se mogu ignorisati.

Klasični *Hamming*-ovi kodovi u stanju su da otkriju i isprave jednostruke bit greške, ali nisu u stanju da naprave razliku između dvostruke bit greške u jednoj kodnoj reči od jednostrukih bit grešaka u različitim kodnim rečima (po koloni). *Hamming*-ovi kodovi mogu da se prošire dodatnim bitom parnosti. Generalno, takvi kodovi se zovu *SEC-DED* kodovi.

Ako kombinujemo informaciono bit polje *IBF* sa *Hamming*-ovim kodom za proveru vrste sa dodatnim bitom parnosti po vrsti (*HCR&P*) i *Hamming*-ovim kodom za proveru kolone sa dodatnim bitom parnosti po koloni (*HCC&P*), ali bez provere *checks on checks* bitova sa Sl. 9-8 c) dobijamo šemu kodiranja koja je prikazana na Sl. 9-8 d) koju ćemo označiti kao *2D SEC-DED*. Glavni razlog za izostavljanje *checks on checks* polja (Sl. 9-8 c)) je sledeći: u predlogu koda mi samo detektujemo, ali ne i korigujemo greške koje se javljaju u poljima *HCR&P* i *HCC&P*. Na ovaj način, po analizama koje su sprovedene, nećemo izgubiti značajno na sposobnosti koda, ali ćemo sa druge strane u značajnoj meri smanjiti redundantnost, a time smanjujemo broj bitova koje treba preneti odnosno kod činimo efikasnijim.

Kreiranje 2D SEC-DED koda: *2D SEC-DED* kod je dvo-dimenzionalni kod, dijagram na Sl. 9-9 pokazuje svoj opšti raspored podataka bitova parnosti. U opštem slučaju grafički prikaz *2D SEC-DED* koda možemo da parametrizujemo sledećim izrazom

$$2D SEC - DED_{k_c+h_{op},k_r+v_{op}}^{d_{min}} [k_r, k_c] \quad (9.5)$$

gde je d_{min} - minimalna *Hamming*-ova distance, $k_{ec}(k_{er})$ - broj bitova provere parnosti po vrsti (koloni), k_r - je broj senzorskih elemenata po senzorskom čvoru, k_c - je broj bitova koji odgovara *ADC* rezoluciji senzovanih podataka (ili, za pametne senzore, očitanih putem nekog interfejsa), $h_{op}(v_{op})$ - je broj bitova provere dodatne parnosti po horizontalnom (vertikalnom) smeru.

U opštem smislu, za horizontalni (vertikalni) *Hamming* kod važi sledeća nejednakost

$$2^{code_bits} \geq data_bits + code_bits + 1 \quad (9.6)$$

koja treba da bude ispunjena i predstavlja uslov kojim se definiše potreban i dovoljan broj kodnih bitova. Pod ovim uslovima, za određene $k_r(k_c)$, imamo da važi

$$2^{k_{er}} \geq k_r + k_{er} + 1 \quad (2^{k_{ec}} \geq k_c + k_{ec} + 1) \quad (9.7)$$

gde $k_{er}(k_{ec})$ - odgovara pojmu *code_bits*, a $k_r(k_c)$ - odgovara pojmu *data_bits* u izrazu (9.6).

U posebnom slučaju, ako za bilo koju celobrojnu vrednost $q \geq 2$ generišemo (n, k) kod koji ispunjava sledeća dva uslova $n = 2^q - 1$ i $k = 2^q - q - 1$ onda kažemo da smo kreirali tzv. perfekt (savršen) kod i to važi za sve tipove kodova.

Dodatno pojašnjenje oznaka korišćenih na Sl.9-9 je:

Oznaka $H(i, op)$ predstavlja vrednost XOR operacije nad k_c i k_{ec} bitovima i -te vrste. Tako da za $0 \leq i \leq k_r - 1$ imamo da je

$$H(i, op) = [\bigoplus_{j=0}^{k_c-1} d(i, j)] \oplus [\bigoplus_{j=0}^{k_{ec}-1} H(i, j)] \quad (9.8)$$

Slično prethodnom, oznaka $V(op, j)$ predstavlja vrednost XOR operacije nad k_r i k_{er} bitovima j -te kolone. Tako da za $0 \leq j \leq k_c - 1$ imamo da je

$$V(op, j) = [\bigoplus_{i=0}^{k_r-1} d(i, j)] \oplus [\bigoplus_{i=0}^{k_{er}-1} V(i, j)] \quad (9.9)$$

		kodne reči koje koriste horizontalni SEC_DED								
		k_c kolone podataka				k_{ec}				h_{op}
kodne reči koje koriste vertikalni SEC_DED	k_r data	$d(0,0)$	$d(0,1)$...	$d(0, k_c - 1)$	$H(0,0)$	$H(0,1)$...	$H(0, k_{ec} - 1)$	$H(0, op)$
		$d(1,0)$	$d(1,1)$...	$d(1, k_c - 1)$	$H(1,0)$	$H(1,1)$...	$H(1, k_{ec} - 1)$	$H(1, op)$
		\vdots	\vdots	...	\vdots	\vdots	\vdots	...	\vdots	\vdots
		$d(k_r - 1, 0)$	$d(k_r - 1, 1)$...	$d(k_r - 1, k_c - 1)$	$H(k_r - 1, 0)$	$H(k_r - 1, 1)$...	$H(k_r - 1, k_{ec} - 1)$	$H(k_r - 1, op)$
	k_{er}	$V(0,0)$	$V(0,1)$...	$V(0, k_c - 1)$	\square	\square	...	\square	\square
		$V(1,0)$	$V(1,1)$...	$V(1, k_c - 1)$	\square	\square	...	\square	\square
		\vdots	\vdots	...	\vdots	\vdots	\vdots	...	\vdots	\vdots
		$V(k_{er} - 1, 0)$	$V(k_{er} - 1, 1)$...	$V(k_{er} - 1, k_c - 1)$	\square	\square	...	\square	\square
	h_{ov}	$V(op, 0)$	$V(op, 1)$...	$V(op, k_c - 1)$	\square	\square	...	\square	\square

Napomena: $d(i, j)$ predstavlja bitove podataka; $H(i, j)$ i $V(i, j)$ predstavljaju horizontalne i vertikalne pariti bitove provere generisane korišćenjem Hamming-ovog koda, respektivno; $H(i, op)$ i $V(op, j)$

predstavljaju bitove horizontalne i vertikalne ukupne parnosti, respektivno; simbol \square predstavlja polje nekorišćenih bitova.

Sl. 9-9 Opšti raspored podataka kod $2D SEC - DED_{k_c+h_p, k_r+v_p}^{d_{mn}}$ koda

U nastavku, umesto notacije $2D SEC - DED_{k_c+h_p, k_r+v_p}^{d_{mn}}$ $[k_r, k_c]$ kratko ćemo koristiti notaciju $2D SEC-DED$ kod, a pre same upotrebe ove notacije daćemo vrednost parametara koji su navedeni u indeksu originalne notacije.

9.13 Erasure kodiranje

Erasure kodovi (*ErCo*) se primenjuju radi poboljšanja pouzdanosti uglavnom u velikim sistemima za skladištenje podataka kod kojih je presudno da se zaštite podaci od gubitaka usled otkaza dela sistema ili brisanja podataka u blokovima. *Erasure* kodovi postavljaju osnovu ove zaštite, što omogućava sistemu da rekonstruiše izgubljene podatke kada komponente sistema otkazu. *Erasure* kodovi mogu međutim ispoljiti značajne režijske troškove tokom aktivnosti dve osnovne operacije: kodiranja, kada se podaci o kodiranju računaju iz novo upisanih podataka, i dekodiranja kada se podaci rekonstruišu nakon otkaza. *Erasure* šeme kodiranja prvobitno su razvijene za komunikaciju putem kanal koji se modeluje kao *Binary Erasure Channel - BEC* i kao takve mogu da se koriste i u bežičnom prenosu.

Erasure kod je *FEC* kod za kanale tipa *erasure* koji poboljšavaju pouzdanost prenosa podataka uvođenjem redundanse, međutim, bez opštih režijskih troškova koje bi nametnula striktna replikacija tj. tzv. dvostruka redundansa. *ErCo* predstavlja generalizaciju koda parnosti i primenjuje se od strane izvornog čvora. Sam kod, kao i ostali kodovi, se sastoji od algoritma za kodiranje i algoritma za dekodiranje. Prvo navedeni proširuje grupu od k paketa u n paketa generisanjem $m = n - k$ redundantnih paketa, gde je $k < n$. Svaki podskup od n paketa koji sadrži najmanje k' paketa je dovoljan da uspešno dekodira originalne podatke pod uslovom da je $k \leq k'$. Stopa koda $R = k/n$ opisuje troškove u smislu redundantnih paketa. [42].

Generalno, *ErCo* kodovi se mogu podeliti na optimalne i skoro (približno)-optimalne kodove. Optimalni *ErCo* kodovi, kao što je *Reed Solomon* kod [65], imaju osobine da su bilo kojih k od n paketa dovoljna da uspešno dekodiraju originalne podatke, tj. $k' = k$. Skoro (približno)-optimalni *ErCo* kodovi, na primer *Tornado* kodovi [69] unose neznatni overhead tako da se zahtevaju $k' = (1 + \varepsilon) \cdot k$ paketa da bi se podaci uspešno dekodirali, gde je $\varepsilon > 0$ pa je $k' > k$.

Reed Solomon kodovi se lako implementiraju u bežičnim senzorskim mrežama sa ciljem da minimiziraju retransmisiju.

9.14 Dužina kodne reči *2D SEC-DED* u odnosu na *Erasure* kodove

U cilju korektne procene dužine kodne reči *2D SEC-DED* i *Erasure* kodova koristićemo implementaciju dvo-dimenzionalnog koda sa proverom parnosti koji je uobičajen za obe kodne šeme i omogućava poređenje broja informacionih bitova pod identičnim uslovima. Tokom ovog procesa razmotrićemo sledeća dva tipa *erasure* šema kodiranja koje su zasnovane na šemi koja je prikazana na Sl. 9-8 c).

U opštem slučaju prvi ćemo nazvati kratko E_{-1} , a drugi E_{-2} . Sve tri šeme kodiranja koriste identično informaciono bit polje uređeno u polje matrice reda $k_r \times k_c$. I E_{-1} i E_{-2} kod koriste osobine *Reed Solomon* kodiranja u procesu generisanja vertikalne redundanse i time postižu tolerantnost na gubitak bilo kojih od t paketa (paketa plus bitovi provere) tokom prenosa.

Izgled *2D SEC-DED* šema kodiranja je prikazan na Sl. 9-8 d). Svaka vrsta (simbol) E_{-1} koda se kodira korišćenjem *CRC* koda horizontalne redundanse sa k_{ec} redundantnim bitovima sa ciljem da detektuje bit(ove) greške po vrsti (simbolu) i koda vertikalne redundanse sa $k_{er} = t$ redundantnih paketa (simbola) sa ciljem da se povrate t izgubljenih paketa (simbola) iz polja informacionih bitova.

Neophodno je da istaknimo činjenicu da su za svaki redundantni simbol dodati *CRC* bitovi polja *chack-on-checks*. U zbiru, dužina kodne reči E_{-1} koda je $n_{E_{-1}} = (k_r + t) \times (k_c + k_{ec}|_{CRC})$.

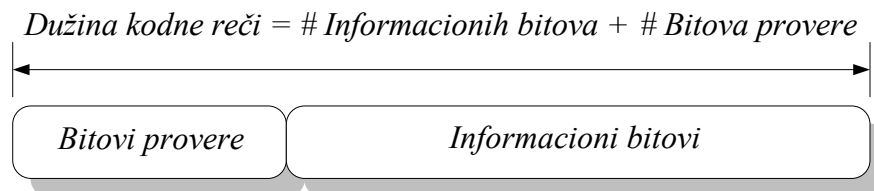
Šema koda E_{-2} za *checks-on-row* i *check-on-checks* polja (vidi Sl. 9-8 c)) umesto *CRC* koristi jedino *SEC-DED* kodnu šemu horizontalne redundanse. Dužina kodne reči E_{-2} koda, u tom slučaju, je $n_{E_{-1}} = (k_r + t) \times (k_c + k_{ec}|_{SEC-DED})$.

Predloženi *erasure* kodovi E_{-1} i E_{-2} su formirani sa ciljem izvođenja uporedne analize sa *2D SEC-DED* kodnom tehnikom koje će biti prezentovana u narednim poglavljima, a tokom kojeg ćemo istaći i pozitivne i negativne karakteristike svih kodova.

9.15 Sistemik kodovi

U principu, kodna reč se sastoji iz dva dela, prvi deo čine informacioni bitovi, a drugi redundantni bitovi (bitovi provere). Ako je redundantnost implicitno ugrađena u kodnu reč, za kod kažemo da je ne sistemik ili ne separabilan. Sa druge strane, ako je redundantnost eksplicitno

dodata informacionim bitovima za kod kažemo da je sistematik (tj. separabilan, kako je to prikazano na Sl. 9-10).



Sl. 9-10 Sistematik kodovi

Generalno, sistematik kodovi su uvek poželjniji u praksi, iz razloga što su informacioni bitovi i bitovi provere razdvojeni tako da predajnik (prijemnik) mogu direktno da pridruže (izdvoje) informacione i bitove provere u kodiranu kodnu reč (iz dekodirane kodne reči). Naime, kada se kodira poruka, ako je deo kodirane poruke sama originalna poruka (informacioni bitovi), moguće je oporaviti originalnu poruku bez dekodiranja tj. manipulacija sa informacionim bitovima, u slučaju da ovaj deo poruke stigne bez oštećenja.

9.16 Sistematik kod za 2D SEC-DED šemu kodiranja

U našem slučaju polje informacionih bitova (vidi Sl. 9-8 a)) označeno u skraćenoj notaciji K je 2D polje reda $k_r \times k_c$. Ideja je da se nad K poljem izvrši dekompozicija u k_r redova. Svaki red sadrži poruku od k_c bitova, ali tako da ti bitovi odgovaraju jednoj originalnoj poruci (u našem slučaju to je podatak sa senzora i određen je rezolucijom ADC konvertora). Princip koji je primenjen je ekvivalentan onome koji je dat u Odeljku 9.8.

Napomenimo da za razliku od klasičnog *interleaving* pristupa gde se originalna poruka dekomponuje u određene "parčiće" u ovom slučaju ti parčići sami za sebe čine originalnu poruku.

Postupci kodiranja primenjuju se u sledeća tri koraka:

Tokom prvog koraka svaka poruka koju ćemo označiti sa k_i , $0 \leq i \leq k_r - 1$, se kodira korišćenjem SEC-DED kodnom šemom, čime dobijamo horizontalnu kodnu reč koju ćemo označiti kao nh_i dužine $k_c + k_{ec} + 1$; tokom drugog koraka, transponujemo polje (matricu) K i dobijamo matricu K^T reda $k_c \times k_r$. Dalje, vršimo dekompoziciju K^T polja u k_c vrsta. Nakon toga, svaka poruka (nije originalna) k_j , $0 \leq j \leq k_c - 1$, se kodira korišćenjem SEC-DED kodnom šemom, čime se dobija vertikalna kodna reč koju ćemo označiti kao nv_j dužine $k_r + k_{er} + 1$. Konačno, treći korak se

bavi formatiranjem paketa, tj. načinom na koji se pakuje u okvir (frejm). Paket se sastoji od k_r kodnih reči. Svaka i -ta kodna reč k_i se formira procesom konkatenacije bitova horizontalne kodne reči (bit ukupne parnosti, *Hamming*-ovi bitovi provere, originalna poruka) sa bitovima provere (bit ukupne parnosti, *Hamming*-ovi bitovi provere) vertikalne kodne reči (nije originalna poruka) na sledeći način (sa brojem bitova kao parametrom)

$$k_i = 1 + 1 + k_{er} + k_{ec} + k_c \quad (9.10)$$

, odnosno po šemi:

$$\left[\begin{array}{l} \text{Ukupna vertikalna parnost} + \text{Ukupna horizontalna parnost} + \\ + \text{Hammingovi horizontalni bitovi} + \text{Hammingovi vertikalni bitovi} + \\ + \text{Originalna poruka} \end{array} \right]$$

Na ovaj način formirana kodna reč poseduje svojstva sistematik koda.

10 RENDEZVOUS PROTOKOL

U ovom delu disertacije najpre ćemo objasniti osnovne principe rada *Rendezvous* protokola (videti Odeljak 10.4) sa posebnim osvrtom na verziju protokola sa originalnim izgledom frejma (okvira) baziranog na *ARQ* metodi koga ćemo nazvati *O-RPLL*. Napomenimo da je izgled strukture frejma *O-RPLL* polazna osnova za sve protokole koji će kasnije biti prezentovani tokom naše analize. Glavna namera je da se identifikuju nedostaci *O-RPLL*-a koji se odnose na povećanu potrošnju energije kao posledica paketne retransmisije. Na kraju, kako bi se izborili sa navedenim nedostatkom, a zadržali osnovne principe rada *Rendezvous* protokola, predložićemo implementaciju modifikovanog *RPLL*-a kao verzije bazirane na *FEC*-u, kojeg ćemo kratko označiti kao *M-RPPL*. Izgledi formata paketa za *O-RPLL* i *M-RPPL* protokole su identični sa ključnom razlikom koja se odnosi na kodiranje podataka i sposobnosti oporavka od grešaka. Detalji o druga protokola označena sa *E1_RPLL* i *E2_RPLL* koji koriste šemu kodiranja *erasure* data su u Odeljku 9.13.

10.1 Izbor *MAC* protokola za aplikacije praćenja

Da bi optimizovali dizajn senzorskog čvora, najpre moramo da raspoložemo modelom potrošnje energije koji će nam omogućiti dobra predviđanja trajanja baterije, a samim tim i životnog veka senzorskog čvora. Naravno cilj je uvek učiniti životni vek senzorskog čvora što je moguće dužim. Da bi produžili životni vek trajanja senzorskog čvora, u literaturi je predložen širok spektar tehnika za minimizaciju potrošnje energije [9]. Neke od njih se bave uštedom energije na nivou *MAC*-a (*Media Access Control*) [70], drugi na protokolima rutiranja [71], treći sa agregacijom podataka ili fuzijom [72], četvrti sa uključivanjem novih arhitektura koje koriste optimizovani radio

blok ili digitalni deo [73], peti koriste primenu tehnike gejtovanja napajanja na čipu kako bi smanjili gubitak usled postojanja struja curenja tj. statičke snage [55].

Imajući u vidu da radio-primopredajnik u senzorskom čvoru (videti Sl. 7-1) troši značajnu količinu energije, a da *MAC* protokol direktno utiče na način njegove upotrebe, očigledno je da treba da usredsredimo našu diskusiju na implementaciju energetski efikasnog *MAC* protokola prvenstveno namenjenog za aplikacije praćenja. Obično takve aplikacije ne moraju da prenose velike količine podataka (tipične brzine su 1–5000 bajtova po sekundi), ali zato prenos mora da bude pouzdan i da obezbedi uspešno isporučivanje podataka od predajnog do prijemnog čvora u okviru senzorske mreže bilo kakve topologije.

Za takve senzorske mreže većina prenetih podataka su posledica očitavanja koja se vrše periodično. Na primer, senzori mogu da prikupljaju podatke o temperature, vlažnosti, pritisku, itd. na svakih 30s, 1, 2, 3 ili 5 minuta, vršeći nadzor na nekom delu terena ili zatvorenog prostora na kome su postavljeni, šaljući samo jedan paket po čvoru. U takvim situacijama u senzorskim mrežama se koriste dva tipa *MAC* protokola: planirani i neplanirani.

Planirani protokoli pokušavaju da smanje potrošnju energije koordinisanjem senzorskih čvorova zajedničkim rasporedom na nivou *MAC*-a. To podrazumeva da se protokolom planiranja definiše koji bi senzorski čvor u kom intervalu trebalo da koristi kanal čime se eliminiše ili svodi na najmanju moguću meru kolizija (sudar) na magistrali, pasivno slušanje ili bespotrebno slušanje od strane čvora.

Najčešći pristup organizaciji planiranja raspoređivanja senzorskih čvorova je da koristi višestruki pristup sa vremenskom raspodelom pri čemu jedan čvor u datom trenutku koristi jedan vremenski slot što znači da svaki čvor može da pristupi bežičnom kanalu samo jedan deo ukupnog vremena predviđenog za sve čvorove u mreži u toku jednog ciklusa.

Protokoli koji nemaju algoritam planiranja pokušavaju da sačuvaju energiju dozvoljavajući senzorskim čvorovima da rade nezavisno sa minimalnom složenošću. Međutim, njih karakteriše veća stopa kolizije na magistrali (sudari), vremensko kašnjenje, nepredvidljivost, pasivno slušanje i nepotrebno slušanje, jer senzorski čvorovi ne koordinišu prenose. Dakle, zaključak bi bio da se *MAC* protokol baziran na vremenskom planiranju lako uklapa u aplikacije za koje projektujemo bežični sistem između ostalog i zbog toga što je jednostavan za upravljanje, veličina paketa je relativno mala i gotovo nepromenljiva, omogućuje determinističko vreme odgovora, poseduje hijerarhijsku strukturu i mogućnost nadogradnje i prediktivno ponašanje cele mreže.

10.2 Klase *Rendezvous* šema

Predloženo je nekoliko ideja za realizaciju *Rendezvous* zadataka u bežičnim senzorskim mrežama kod kojih su aktivnosti sa niskim faktorom ispune. Obično funkcionišu na *MAC* nivou i mogu se svrstati u tri opšte klase [74]:

- i.1. asinhroni
- i.2. pseudo-asinhroni
- i.3. sinhroni

Za nas od interesa su sinhrona rešenja u kojima se senzorski čvorovi vremenski sinhronizuju i uparuju sa određenim komunikacionim vremenskim slotom. Osnovna ideja koja stoji iza sinhronog *Rendezvous* protokola je da se svaki senzorski čvor, koji treba da šalje poruku, probudi u tačno definisanom vremenskom slotu istovremeno sa svojim susednim senzorskim čvorom, koji treba da primi poruku. U ovakvom pristupu to bi značilo da *Rendezvous* protokol podržava komunikaciju u više preskoka (*multihop*). Naš pristup podrazumeva modifikaciju protokola i realizaciju prenosa jednim preskokom (*one-hop*) u tzv. topologiji tipa zvezda, što znači da je prijemni senzorski čvor jedinstven za celu senzorsku mrežu, a kao što smo već naglasili označen je kao pristupna tačka, *gateway* ili *AP* (*Access Point*). Konačno, to znači da se pojedinačni senzorski čvor budi u skladu sa rasporedom buđenja, ostaje aktivan tokom kratkog vremenskog intervala u toku kojeg ostvaruje komunikaciju sa pristupnom tačkom nakon čega i jedan i drugi nastavljaju nezavisno svoje aktivnosti.

10.3 *Rendezvous* protokol za dug životni vek senzorskog čvora (*Rendezvous Protocol for Long Live Sensor Node - RPLL*)

Uz već značajan niz razloga, navedenih u prethodnom tekstu, kojim se opravdava korišćenje *Rendezvous* protokola, neophodno je da istaknemo i sledeće:

- a) Bežična senzorska mreža koju projektujemo primenjivala bi se u aplikacijama za periodično prikupljanje podataka sa senzora malih brzina i drugih retkih sporo-promenljivih pojava iz okruženja od interesa
- b) *RPLL* protokol je pogodan za upotrebu u bežičnim senzorskim mrežama sa umerenim brojem senzorskih čvorova (<255) u topologiji tipa zvezda. Svakom senzorskom čvoru je dodeljen jedinstven 8-bitni identifikacioni broj (*ID*). Bežična senzorska mreža je sastavljena

od jedne pristupne tačke (sa jedinicom za neprekidno napajanje) i određenog broja senzorskih čvorova (uređaja koji su baterijski napajani).

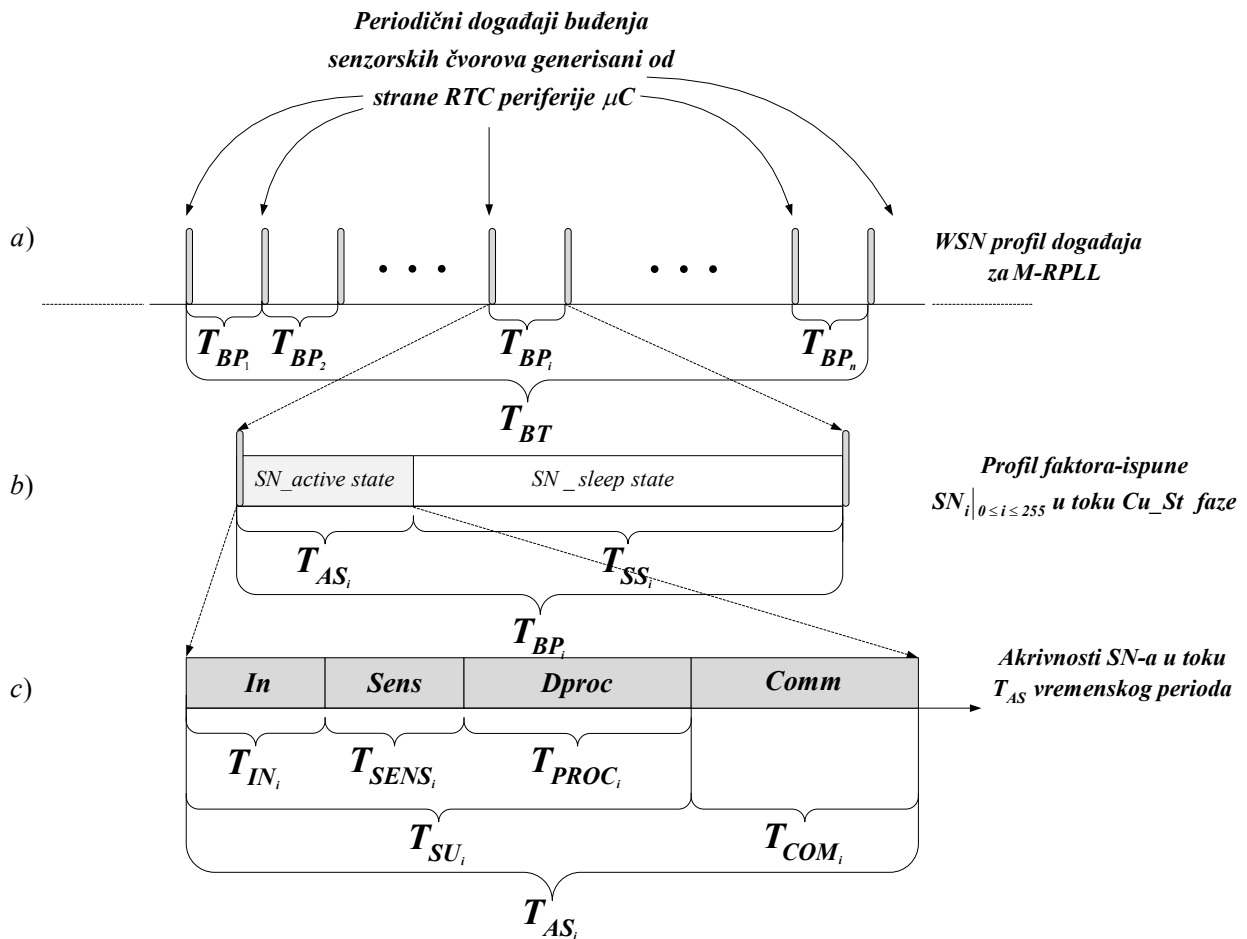
- c) Položaj vremenskog slota, tokom kojeg se senzorski čvor aktivan, u okviru *beacon* perioda je fiksiran i direktno određen *ID* brojem senzorskog čvora. Napomenimo, da je vreme trajanja *beacon* perioda unapred poznato i definisano tokom faze projektovanja tj. u procesu inicijalizacije senzorskog čvora.
- d) Čak i u prisustvu vrlo nepouzdanih veza *RPLL* protokol u stanju je da uspešno sinhronizuje vreme buđenja senzorskog čvora u potpuno decentralizovanom maniru [75].
- e) Problemi nastali usled driftovanja takta, otkaza linka, privremeno asimetrične veze i skrivenih terminala mogu se efikasno preovladati dostupnim mehanizmima protokola.

Ako pretpostavimo da su *M-RPLL* aktivnosti u fazama otkrivanje susednih senzorskih čvorova u okruženju pristupne tačke (*Neighbor-Discovering (Ne-Di)*), registracija svih prisutnih senzorskih čvorova od strane pristupne tačke (*Neighbor-Registration (Ne-Re)*) i ažuriranje planera o dodeli vremenskih slotova u okviru *beacon* perioda (*Update-Scheduler (Up-Sc)*) uspešno izvedene, onda će se aktivnosti faze tekuće stanje (*Current State (Cu-St)*) repetitivno izvršavati u okviru svakog nadolazećeg *beacon* perioda [74-76]. Vreme trajanja svakog *beacon* perioda je identično prema svi senzorskim čvorovima.

10.4 Rad u niskom faktoru ispunje korišćenjem RPLL protokola

Na Sl. 10-1 a) prikazan je obrazac aktivnosti bežične senzorske mreže sa niskim faktorom ispunje. Vreme *beacon* vremena, T_{BT} , sadrži n vremenskih slotova nazvanih *beacon* period, T_{BP} . U skladu sa dodeljenim identifikacionim brojem (*ID*) svaki senzorski čvor se periodično budi na početku svog slota, odnosno *beacon* perioda T_{BP} , obavi određene aktivnosti i nakon toga se isključuje. Postojanje *Real Time Clock* blok periferije na čipu mikrokontrolera (*MCU*) vidi Sl. 7-1, pojednostavljuje realizaciju periodičnog generisanja događaja buđenja (vremenski okidan događaj). To znači da se *RTC*, a samim tim i senzorski čvor, može programirati u cilju generisanja *beacon* perioda od 30s do 300s sa određenim korakom. Pojava signala buđenja (*wake-up*) inicira senzorski čvor da uđe u *SN_active state*, vidi Sl. 10-1 b). Tokom *SN_active state* (vidi Sl. 10-1 c)) sekvencijalno se izvršavaju četiri aktivnosti, *Initialization (In)*, *Sensing (Sens)*, *Data processing (Dproc)*, and *Communication (Com)*. Vreme trajanja *SN_active state* označeno je kao T_{AS}

Aktivnost *In* obuhvata obnavljanje sadržaja *MCU* registara na prethodno *SN_active state*. Vreme trajanja *SN_active state* je $T_{AS} = T_{SU} + T_{COM}$. Vreme pripreme je $T_{SU} = T_{IN} + T_{SENS} + T_{PROC}$, gde T_{IN} , T_{SENS} i T_{PROC} odgovaraju vremenima trajanja aktivnosti *In*, *Sens* i *Dproc*, respektivno. Tokom vremena T_{SS} senzorski čvor se nalazi u stanju *SN_sleep state* (vidi Sl. 10-1 b)). Više detalja u vezi sa aktivnostima koje se obavljaju tokom vremena T_{SU} , već su opisane u Odeljku 8 (vidi Sl. 8-2). Prema tome, u tekstu koji sledi, svu našu pažnju ćemo usmeriti na aktivnost koja se obavlja samo tokom vremena T_{COM} .



Sl. 10-1 Scenario aktivnosti a) generisanje događaja buđenja; b) profil faktora ispunje aktivnosti; c) profilisanje aktivnosti senzorskog čvora tokom *SN-active state*

10.5 Formiranje okvira kod *O-RPLL*

Format *O-RPLL* paketa podataka (vidi Sl. 10-2) se sastoji od sledećih šest segmenata [75].

- ✓ Bitovi preambule – alternativna sekvenca jedinica i nula (1010101010 ...)

- ✓ Sinhronizaciona reč – omogućava sinhronizaciju bajtova dolazećeg paketa. Sastoji se od bajta sinhronizacije i bajta koji označava početno razdvajanje sa pravim podacima
- ✓ Zaglavlje – sastoji se od polja veličine jedan bajt koje nosi informaciju o broju bajtova u polju podataka i polja adrese veličine jedan bajt. Radio primopredajni čip *CC110L* koji koristimo u našem dizajnu, podržava i protokole fiksne dužine paketa i protokole promenljive dužine paketa. Maksimalna dužina paketa može biti podešena od strane projektanta u fazi inicijalizacije i iznosi do 256 bajtova.
- ✓ Polje podataka – odgovara korisnim podacima u okviru frejma (okvira)
- ✓ *CRC* polje – 16-bitno polje provere koje se izračunava nad poljem podataka
- ✓ Završno razdvajanje – ukazuje na završetak paketa koji se prenosi. Napomenimo da može da se analizira rešenje koje ne poseduje ovaj segment.

<i>Preamble bits</i> 101010...1010	<i>Sync</i> word	<i>Header</i>		<i>Data field</i> (payload data)	<i>CRC</i> field	<i>End</i> delimiter
		<i>length</i> field	<i>address</i> field			
2B	2B	1B	1B	nB	2B	1B

Sl. 10-2 Format paketa *O-RPLL* protokola

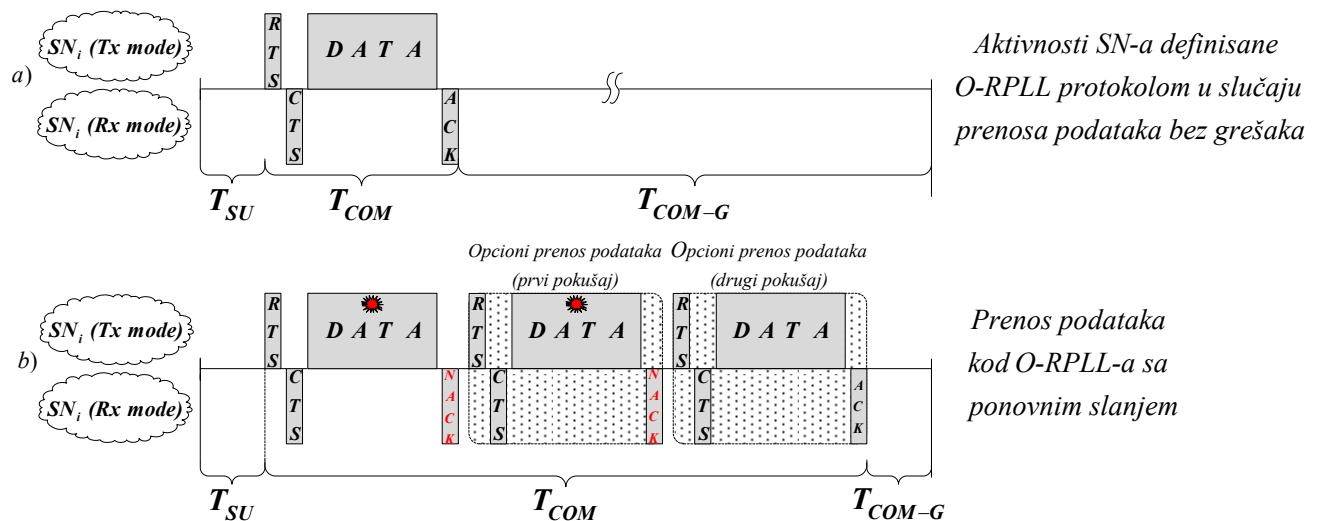
10.6 Prenos podataka bez grešaka i potreba za ponovnim prenosom

Slike Sl. 10-1 b) i Sl. 10-1 c) ilustruju aktivnosti senzorskog čvora označenog SN_i u toku i -og *beacon* perioda. Na kraju vremena T_{SU} , aktivnosti *In*, *Sens* i *Dproc* su već izvršeni i počinje aktivnost T_{COM} . U toku ove aktivnosti koristimo sledeće poruke: *RTS*, *CTS*, *DATA* i *ACK/NACK*, dakle tzv. 4-way proceduru rukovanja. Na početku SN_i šalje *RTS* pristupnoj tački nakon čega čeka da primi poruku *CTS* čime se potvrđuje da je bežični link slobodan i počinje prenos podataka, tj. šalje se poruka *DATA*. Na kraju, pristupna tačka nakon provere ispravnosti odgovara pozitivnom *ACK* porukom za slučaj da je prenos podataka bez grešaka (videti Sl. 10-3 a)). U prethodnoj analizi pretpostavili smo da je komunikacija idealna i da se prilikom prenosa ne pojavljuju greške. U praksi tj. u realnim komunikacijama to nije tačno, jer u proseku samo 30% svih poslatih paketa bude primljeno bez grešaka [77]. Imajući navedeno u vidu neophodno je da uključimo neke modifikacije u proces prenosa podataka po *O-RPLL* protokolu i da pokušamo da rešimo problem realnih komunikacija. U principu postoje dve mogućnosti. Prva je zasnovana na ponovnom slanju

(retransmisiji) oštećenih paketa (videti Sl. 10-3 b)). U *O-RPLL* protokolu, u toku vremena T_{BP} , mogu se tolerisati do tri pokušaja prenosa podataka. U slučaju da tri puta imamo neuspešan prenos, usled pojave grešaka, pretpostavljamo da je link otkazao i odustajemo od novog pokušaja. Uprkos činjenici da ponovno slanje povećava pouzdanost prenosa podataka njeno korišćenje, takođe, podrazumeva povećano kašnjenje i visoke komunikacione režijske troškove.

Drugo rešenje zasniva se na korišćenju efikasne tehnike kodiranja podataka kojom bi, što će biti dokumentovano u daljem tekstu, postigli sledeće ciljeve:

1. Smanjili komunikaciono kašnjenje usled retransmisije i komunikacione režijske troškove, u slučaju kada je stopa gubitaka na linku visoka, i da projektujemo energetski efikasan protokol za dostavu podataka, i
2. Da simuliramo i procenimo performanse predloženog *M-RPLL* protokola



Sl. 10-3 Aktivnosti *O-RPLL* a) prenos podataka bez grešaka; b) prenos podataka sa do tri prenosa (sa do dve retransmisije)

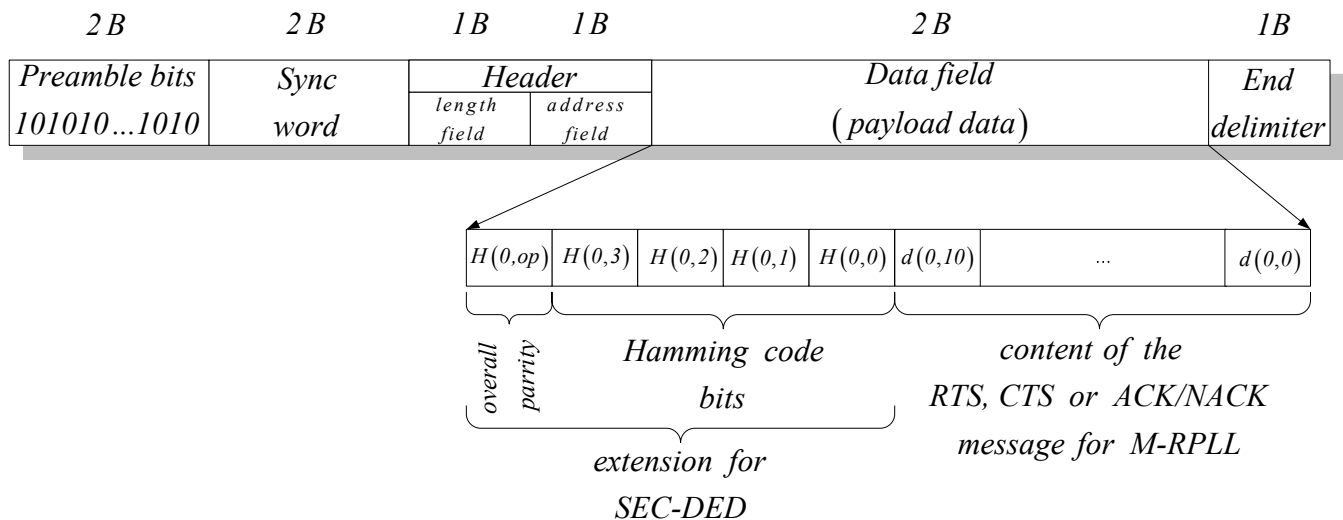
Napomena: T_{COM-G} - komunikaciono zaštitno vreme

10.7 Formiranje okvira kontrolnih poruka u *M-RPLL*

Postoji sledeća razlika između *O-RPLL* i *M-RPLL* verzija protokola. Dužine poruka *RTS*, *CTS*, *DATA* i *ACK/NACK* (videti Sl. 10-2) su modifikovane. Izmena podrazumeva izostavljanje *CRC* polja i umesto njega ubacivanje *SEC-DED/2D SEC-DED* polja tj. šeme kodiranja. Tokom ovog procesa, *RTS*, *CTS* i *ACK/NACK* poruke se kodiraju koristeći *SEC-DED*, dok *DATA* poruka koristi

2D SEC-DED. Ovom modifikacijom tj. uvođenjem nove šeme kodiranja u stanju smo, minimalno, da otkrijemo i korigujemo sve jednostruke greške, posledica sposobnosti kodnih šema, i na taj način drastično smanjimo retransmisiju paketa u *M-RPLL*.

Format kontrolnih poruka (*RTS*, *CTS* i *ACK/NACK*) za *M-RPLL* protokol prikazan je na Sl. 10-4. Razmatrajući jednačinu (9.5) ako odaberemo sledeće vrednosti parametara $k_r = 1$, $k_c = 11$, $k_{er} = 0$, $k_{ec} = 4$, $h_{op} = 1$, $v_{op} = 0$ i $d_{\min} = 4$, predloženi *2D SEC-DED* kod se transformiše u jednodimenzionalni $(16,11)$ *SEC-DED* savršen kod. Tako dobijena šema kodiranja u stanju je da koriguje (ispravi) sve jednostruke greške i da detektuje (otkrije) prisustvo dvostrukih grešaka (*SEC-DED*) u poljima *RTS*, *CTS* i *ACK/NACK*.



Sl. 10-4 Format kontrolnih poruka kod *M-RPLL* protokola

10.8 Provera *SEC-DED* koda

Bežični prijemnik (senzorski čvor ili pristupna tačka) nakon prihvatanja kontrolne poruke proverava *SEC-DED* kod u dva koraka. U Koraku 1, proverava se h_{op} pojedinačni bit parnosti (ukupna parnost). Ako je ukupna parnost pogrešna prijemnik pretpostavlja da je došlo do pojave jednostruke greške i koristeći se drugim bitovima parnosti (bitovima *Hamming*-ovog koda, k_{ec}) u Koraku 2 koriguje jednostruku grešku ili u informacionim bitovima ili u bitovima parnosti, shodno Tab. 10-1. Naravno, može da se desi da su tri ili čak pet bitova pogrešni, međutim u takvim situacijama *SEC-DED* kodna šema je nemoćna čak i u procesu detektovanja takvih grešaka. Za slučaj

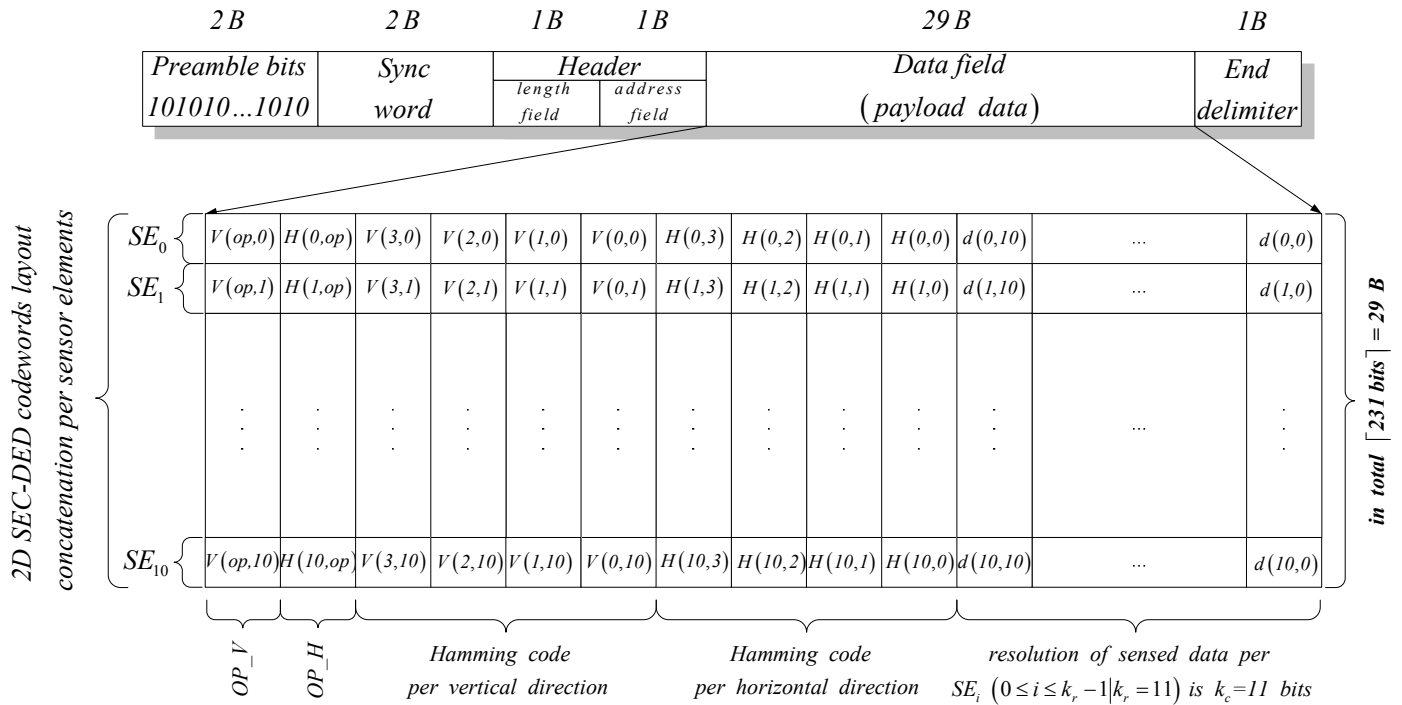
da je ukupna parnost h_{op} dobra, onda ili nema grešaka ili su dva bita pogrešna. Prijemnik zatim prelazi na korak 2 u okviru koga koristi druge bitove parnosti (bitovi *Hamming*-ovog koda, k_{ec}) putem kojih pravi razliku između ova dva slučaja. Naravno ponovo može da dođe do pojave da četiri ili šest bitova budu pogrešni, ali ovaj kod nije u stanju da opsluži ovakve slučajeve.

Tab. 10-1 Markeri (flegovi) za detekciju grešaka

<i>Sindrom</i>	<i>Ukupna parnost</i>	<i>Tip greške</i>	<i>Opis</i>
0	0	<i>no error</i>	Nema grešaka
$\neq 0$	1	<i>single error</i>	Korektivno (vrednost sindroma ukazuje na poziciju pogrešnog bita)
$\neq 0$	0	<i>double error</i>	Nije korektivno
0	1	<i>parity error</i>	Korektivno (dogodila se greška u pariti bitovima)

10.9 Formiranje okvira poruke *DATA* kod *M-RPLL* protokola

Format poruke podataka (*DATA*) za *M-RPLL* protokol prikazan je na Sl. 10-5. Razmatrajući jednačinu (9.5) ako odaberemo sledeće vrednosti parametara $k_r = 11$, $k_c = 11$, $k_{er} = 4$, $k_{ec} = 4$, $h_{op} = 1$, $v_{op} = 1$ i $d_{\min} = 4 \times 4 = 16$, dobijamo (16,11) *SEC-DED* savršeni kod po vrsti i po koloni. Na taj način generišemo produkt kod čime se formira *2D SEC-DED* šema kodiranja, koju predlažemo za korišćenje u bežičnim senzorskim mrežama, prema procedure za generisanje sistemskog koda definisanoj u Poglavlju 9.12.

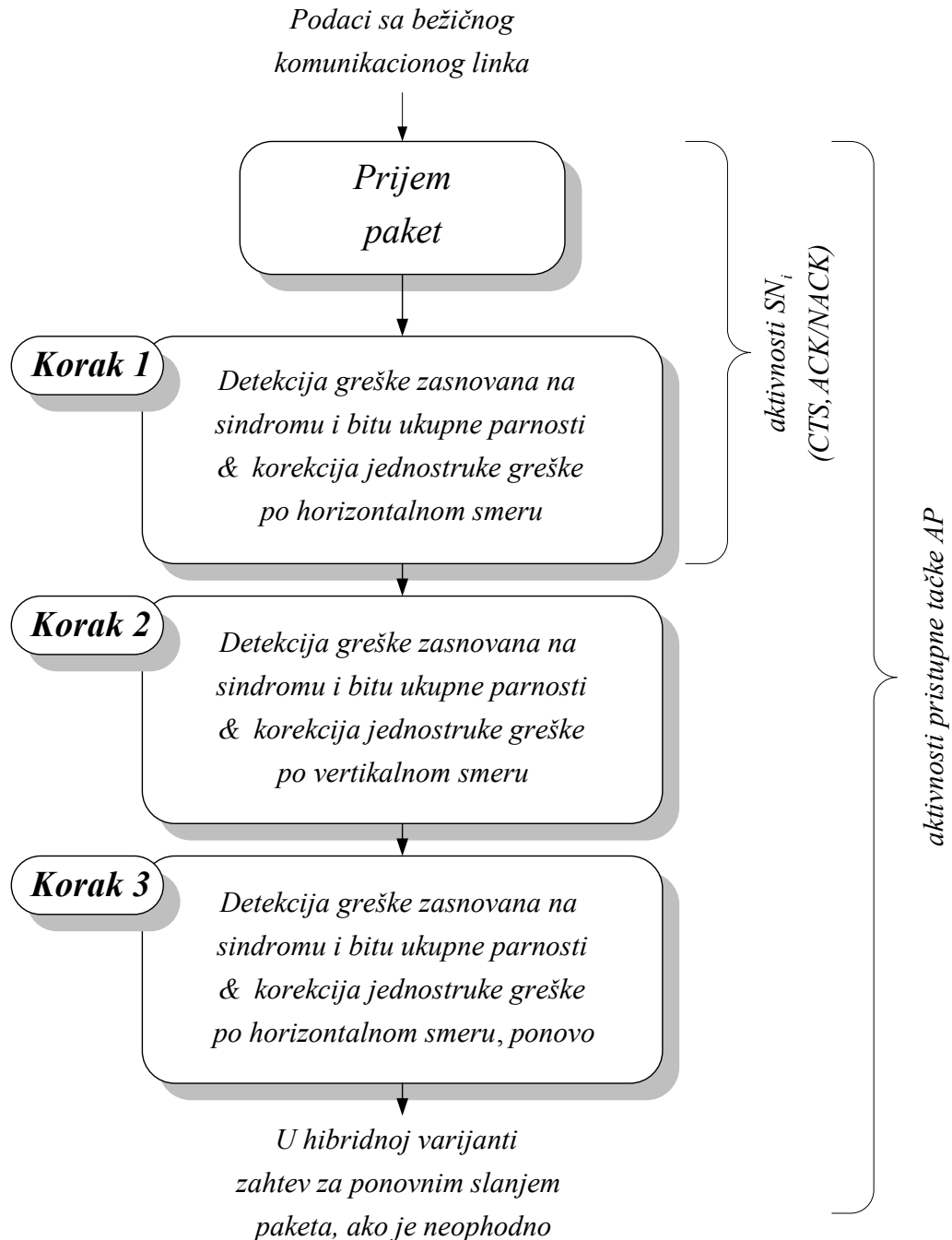


Sl. 10-5 Format okvira (frejma) poruke DATA kod M-RPLL protokola korišćenjem osobine sistematik koda

10.10 Sposobnost ispravljanja grešaka 2D SEC-DED šeme kodiranja kod M-RPLL protokola

Da bi ispravili greške po ceni niske latencije, slično kao u [78], koristimo proces dekodiranja kroz tri makro koraka (vidi Sl. 10-6). Svaki makro korak primenjuje iste aktivnosti koje su već definisane u odeljku 9.16. Korišćenjem navedenog postupka dobijeni efekti su sledeći:

- 1) Makro_Korak_1 - ispravlja jednostruke greške u svakoj vrsti (horizontalni);
- 2) Makro_Korak_2 - ispravlja jednostruke greške u svakoj koloni (vertikalni);
- 3) Makro_Korak_3 - ispravlja jednostruke greške u svakoj vrsti, ponovo (horizontalno);



Sl. 10-6 Procesi detekcije i korekcije sprovedeni od strane senzorskog čvora i pristupne tačke

Kao što se može videti sa Sl. 10-6 predajnik SN_i obavlja samo aktivnosti specificirane Makro_Korakom_1 (bavi se detekcijom i korekcijom grešaka *SEC-DED* kodiranih poruka *CTS* i *ACK/NACK*). Sa druge strane, pristupna tačka izvršava Makro_Korak_1 za *SEC-DED* kodirani *RTS* poruku, a aktivnosti od Makro_Koraka_1 do Makro_Koraka_3 za *2D SEC-DED* kodiranu *DATA* poruku.

Za detaljniju analizu aktivnosti, tokom konkretne primene algoritma pogledati Prilog A.

10.11 Specifičnosti predloženog rešenja

Prethodnih godina učinjeni su značajni napori prezentovani kroz veliki broj radova o najvažnijoj temi sposobnosti otkrivanja/ispravljanja grešaka i odabiru optimalnog *ECC* u bežičnim senzorskim mrežama sa ultra niskom potrošnjom, ali standard još uvek nije postavljen [8, 10-13, 16, 18]. U osnovi, izbor *ECC*-a zavisi od aplikacije, karakteristika greške, u smislu modela greške i šablona pojavljivanja, karakteristika bežičnog kanala. Naše opservacije, u početnoj fazi istraživanja, zasnovane na merenjima, pokazuju da su većina bit grešaka po svojoj prirodi jednostruke ili dvostruke bit greške, a da se *burst*/višestruke greške retko pojavljuju što je u skladu sa [70]. Ovo implicira da *ECC* kodna šema koja je u stanju da ispravi jednostruke i dvostruke greške čime smanjujemo značajan broj grešaka, predstavlja odgovarajući izbor.

Da bismo se izborili sa ovakvom pojavom grešaka i našli odgovarajuću kodnu šemu, pretpostavili smo da korišćenje proizvod koda, koji kombinuje dva *Hamming*-ova blok koda stvara mnogo snažniju tehniku za kontrolu grešaka. Ova polazna pretpostavka, u suštini, bazira se na ideji traženja i analiziranja sposobnosti dugih blok kodova bez povećanja složenosti koja je u vezi sa procesom kodiranja/dekodiranja primenom kraćih i jednostavnijih komponenti koda. Takav dvodimenzionalni linearni blok kod mi zovemo *2D SEC-DED* kod. Celokupni dvodimenzionalni niz (*2D niz*) možemo da sagledamo kao deo koji ima sekciju poruke, sekciju simbola parnosti po vrsti ili bitove provere po vrstama i sekciju simbola parnosti po koloni ili bitova provere po kolonama.

Kao modulacionu tehniku koristimo *Gaussian Frequency Shift Key (GFSK)*. U *GFSK* osnovnom opsegu impulsi (nivoi -1 i $+1$) prvo prolazi kroz *Gauss*-ov filter pre procesa modulacije. Ovo čini impulse uobičenim i time ograničava širinu spektra modulacije. Ovakav izbor modulacione šeme kodiranja omogućava nam da postignemo efikasan prenos sa manjom snagom predajnika i prijemnika.

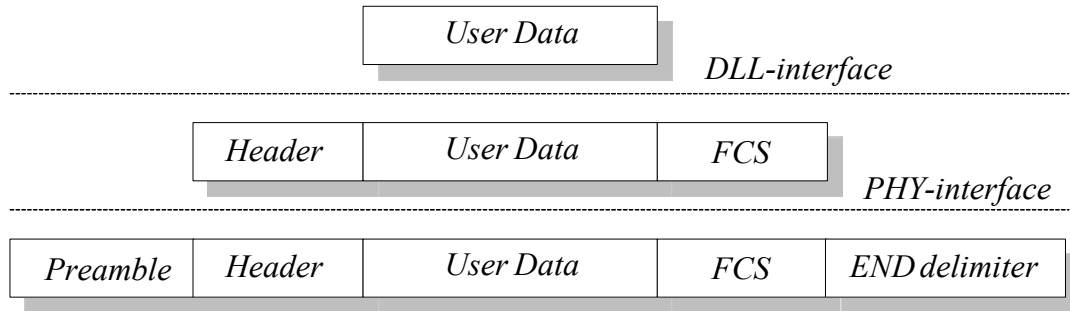
Korišćenjem *2D SEC-DED* tehnike kodiranja smanjujemo ili potpuno eliminišemo broj paketa koje ponovo šaljemo kod bežičnih senzorskih mreža koje su postavljene u industrijskim ili nekim drugim okruženjima koja se karakterišu "surovim" uslovima za rad što znači da je stopa greške na nivou bitova, $BER \geq 10^{-4}$. Ovo znači da ne samo da štedimo energiju nego smo, i na račun te energije, u stanju da povećamo domet predaje (razdaljinu prenosa između predajnika i prijemnika).

U odeljcima disertacije koji slede najpre ćemo opisati analitički model za analizu stope grešaka na nivou paketa (*Packet Error Rate - PER*) u funkciji *BER*-a i odrediti optimalnu veličinu paketa u funkciji overheda (redundantnih bitova u paketu) i *BER*-a kao parametara. Zatim, za dva sinhronizaciona protokola tipa *Rendezvous* (prvi koristi *ARQ*, a drugi *FEC* mehanizam za kontrolu grešaka) pri prenosu podataka bez-grešaka (*error free*) određujemo očekivani broj prenosa paketa u predefinisanim vremenskim intervalima i potrebnu energiju po bitu u tom slučaju. Nakon toga, za prenos podataka (paketa) preko linka koji se karakteriše relativno visokom stopom grešaka procenjujemo performanse u vezi sa brojem ponovnog slanja paketa (retransmisija), dometom predaje, faktorom oporavka paketa sa greškama i faktorom dobiti *BER*-a.

11 ANALITIČKI MODEL PRENOSA PODATAKA

Metodologija za procenu stope pogrešno prenetih paketa (*Packet Error Rate – PER*) metrike, najpre je bazirana na analitičkom modelu prenosa koji predstavlja pojednostavljenu analizu, ali čiji rezultati mogu da nam pruže dobru osnovu za dalje korake u radu. Odnos između *PER*-a i *BER*-a je validan za komunikacioni sistem koji prenosi podatke preko binarno simetričnog kanala sa nekorelisanim šumom. Za takve komunikacije odnos između *PER* i *BER* pretpostavlja da su greške bitova nezavisne i ravnomerno raspoređene. U nastavku ćemo analizirati performansne metrike podrazumevajući da ove pretpostavke važe.

Opšte formatiranje paketa na nivou veze (*Data Link Layer - DLL*) i fizičkom nivou (*PHYSical Layer- PHY*) protokol steka prikazano je na Sl. 11-1. Nivo veze čvora koji šalje (predajnika) prihvata korisničke podatke (*User Data*) sa višeg sloja protokol steka. Nakon obrade, on dodaje polje zaglavlja (*header*) i polje provere okvira (*frame check sequence - FCS*) i isporučuje paket (okvir) fizičkom sloju. Zaglavlje se sastoji iz dva dela. Prvi, nazvan dužina polja sadrži n_{LF} - bitova (ukazuje na broj bitova u paketu), a drugi, koji se naziva adresno polje (pokazuje na adresu odredišnog čvora SN_i) i sadrži n_{AF} - bitova. Polje korisničkih podataka sadrži n_{DATA} - bitova korisničkih podataka, dok je dužina *FCS* polja n_{CS} - bitova.



Sl. 11-1 Opšte formatiranje paketa za komunikaciju preko kanala

Fizički sloj dodaje polje *Preamble* sastavljenu od n_p - bitova preambule i n_s - sinhronizacionih bitova i polje *End Delimiter* koje se sastoji od n_{ED} - bitova (opciono polje, koristi se samo u nekodiranom sistemu).

Nakon ovako formatiranog paketa možemo da, ukupnu dužinu paketa n_{total} , predstavimo na sledeći način:

$$n_{total} = n_p + n_s + n_{LF} + n_{AF} + n_{DATA} + n_{CS} + n_{ED} \quad (11.1)$$

Overhed paketa tj. broj redundantnih bitova u oznaci n_{PO} odgovara sledećoj sumi

$$n_{PO} = n_p + n_s + n_{LF} + n_{AF} + n_{CS} + n_{ED} \quad (11.2)$$

Ova kratka analiza koja može da se prihvati kao opšta kvantitativno izražena forma u formatiranju paketa predstavljaće nam osnovu za dalju analizu i uvođenje analitičkog modela prenosa podataka

Pretpostavimo da je ukupna dužina paketa n_{total} bitova i da je verovatnoća greške po bitu kanala p_b . Svaki paket koji ima jedan ili više pogrešnih bitova u prenosu se odbacuje od strane prijemnika, a verovatnoća P da će paket biti sa greškom je data kao:

$$P = 1 - (1 - p_b)^{n_{total}} \quad (11.3)$$

gde je P stopa greške paketa (*PER*), a p_b stopa greške bitova (*BER*). U tom slučaju, prema prethodnom izrazu, odnos između *PER* i *BER* je dat kao:

$$PER = 1 - (1 - BER)^{n_{total}} \quad (11.4)$$

Napomenimo da je prethodna jednačina samo aproksimacija jer, u realnim uslovima, greške bitova nisu potpuno nezavisne s obzirom da se mogu pojaviti u *burst*-ovima. Pored toga, distribucija greške nije potpuno uniformna. U percepciji koja sledi, umesto *burst* modela greške razmatraćemo

bit model greške zbog sledeće činjenice. *Erasure* kodiranje je poželjnije za upotrebu kada je dominantan *burst* šablon grešaka, dok je *Hamming* kodiranje prihvatljivije kada pojava šuma izaziva slučajne greške. Ovo je u skladu sa već pomenutom konstatacijom da su većina grešaka po svojoj prirodi jednostruke ili dvostruke i da su *burst* greške prisutne, ali ređe.

Sprovedena analiza je bazirana na implementaciji sinhronog protokola tipa *Rendezvous* koji koristi četvorosmerno (*four-way*) rukovanje kao mehanizam za kontrolu protokola. *Rendezvous* protokol u procesu rukovanja koristi sledeće poruke *RTS*, *CTS*, *DATA* i *ACK/NACK*. U tom slučaju *PER* je jednak:

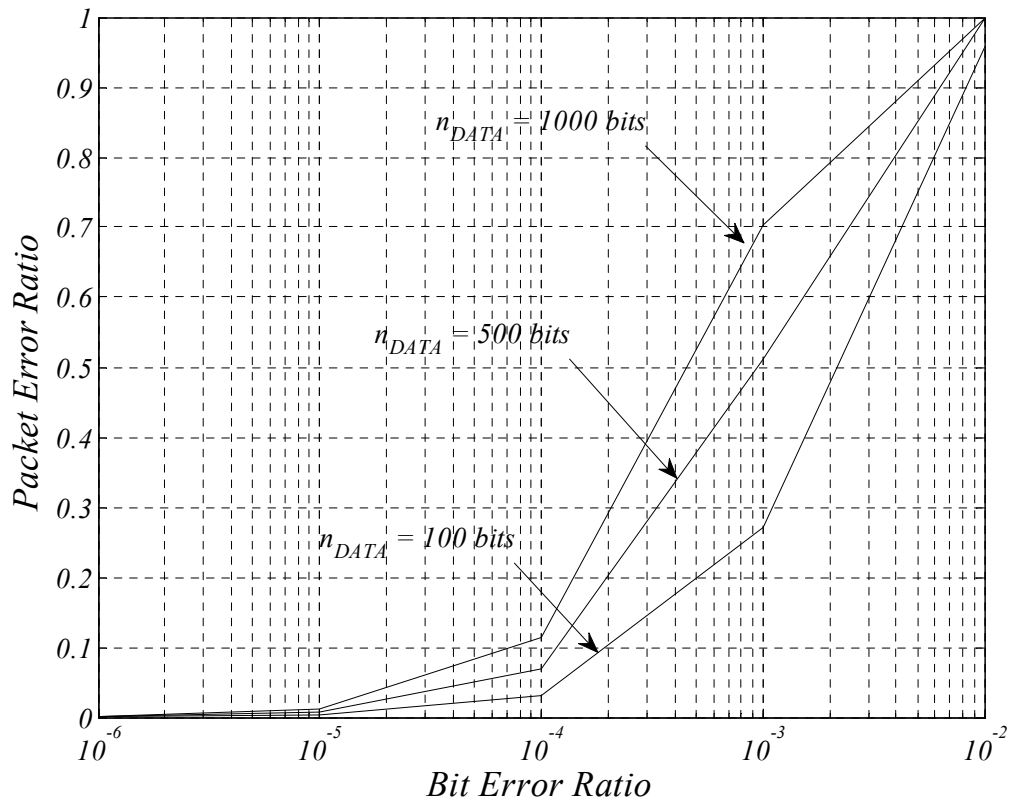
$$PER = 1 - (1 - BER)^{n_{RTS}} \cdot (1 - BER)^{n_{CTS}} \cdot (1 - BER)^{n_{DATA}} \cdot (1 - BER)^{n_{ACK/NACK}} \quad (11.5)$$

gde je:

n_{RTS} , n_{CTS} , $n_{ACK/NACK}$ - odgovara broju bitova u *RTS*, *CTS* i *ACK/NACK* kontrolnim (upravljačkim) porukama, respektivno i one su po protokolu fiksne dužine;

n_{DATA} - odgovara broju bitova u *DATA* poruci podataka koja je po protokolu varijabilne dužine u zavisnosti od primenjene tehnike kodiranja.

Dakle u jednačini (11.5) promenljivi parametri su *BER* i n_{DATA} pa možemo da izvršimo analizu zavisnosti *PER*-a u funkciji promene njihovih vrednosti. Naime, kada bežična senzorska mreža radi nadgledanje ili praćenje procesa u surovim komunikacionim uslovima vrednost *BER*-a se kreće u opsegu od 10^{-6} do 10^{-2} . Na Sl. 11-2 prikazana je zavisnost *PER*-a u funkciji parametra *BER* za tri različite vrednosti dužine polja podataka *DATA*. Tokom ove analize usvojili smo da su sve kontrolne poruke protokola fiksne dužine tj. $n_{RTS} = n_{CTS} = n_{ACK/NACK} = 80$ bitova. Analizirajući sliku Sl. 11-2 možemo da zaključimo da *PER* raste kako se povećava vrednost *BER*-a i dužina polja *DATA*. Na primer, u slučajevima kada je $BER = 10^{-5} / 10^{-4} / 10^{-3}$ i minimalna (maksimalna) vrednost dužine *DATA* polja $n_{DATA} = 100(1000)$ bitova dobijamo da je $PER[\%] = 0.32 / 3.11 / 27.11(1.21 / 11.45 / 7.40)$. U praksi, smatra se da su bežične senzorske aplikacije kod kojih je $PER < 10\%$ prihvatljive i mogu se tolerisati, što implicira da je optimalan izbor dužine polja *DATA* u opsegu od 100 do 400 bitova.



SI. 11-2 PER u funkciji BER-a i dužine DATA polja kao parametara

Uprkos činjenici da je BER pruža važan uvid u ponašanje bežične senzorske mreže, uzeti u obzir samo vrednosti BER -a nije dovoljno za evaluaciju PER -a. Naime, PER je uglavnom izveden na osnovu jedinstvene hipoteze o uniformnoj distribuciji greške. Prema tome potrebna je mnogo preciznija karakterizacija procesa greške pošto na izlazu fizičkog sloja rezidualne greške nisu uniformno raspoređene. Osim toga, na nivou veze, tokom ove analize nisu uzeti u obzir efekti koda za ispravljanje grešaka (naravno kada je on implementiran).

11.1 Efikasnost kanala kao metrika

Imajući u vidu da realni bežični prenos podataka nije idealan, neophodno je da sada sagledamo i odredimo efikasnost kanala kao kompromis dužine paketa (pod dužinom paketa smatramo korisne informacije i redundantnost paketa) i BER -a. U tom smislu efikasnost kanala u oznaci CE (*Channel Efficiency*) definisaćemo kao:

$$CE = \frac{n_{DATA}}{n_{DATA} + n_{PO}} \quad (11.6)$$

gde n_{DATA} - odgovara broju bitova paketa u delu korisnih informacija, a n_{PO} - odgovara broju bitova u paketu koji predstavljaju redundantne bitove i definisan je jednačinom (11.2).

Tokom ove analize pretpostavljamo da nakon slanja paketa čvor koji šalje čeka na paket potvrde *ACK* (pozitivna) ili *NACK* (negativna) potvrda ili na istek predefinisano vreme ispadanja (*time-out*). Ovakav scenario razmene podataka ukazuje nam da razmatrani tip protokola pripada grupi *Stop-And-Wait (SAW) ARQ* protokola, a njegov izbor opravdava činjenica da su većina trenutno korišćenih pristupa za oporavljanje greške zasnovani na ovoj metodi.

U tom slučaju, a na bazi jednačine (11.6), efikasnost kanala može da se definiše kao:

$$CE = (1 - p_b)^{n_{total}} \cdot \frac{n_{total} - n_{PO}}{n_{total} + n_{ACK} + (t_{proc} + t_{prop}) \cdot R} \quad (11.7)$$

gde je p_b - verovatnoća greške; n_{total} - ukupan broj bitova po paketu; n_{ACK} - broj bitova *ACK* paketa; n_{PO} - broj redundantnih bitova po paketu; t_{proc} - vreme procesiranja (pretpostavljamo da je to identično vreme za čvor pristupna tačka i čvor koji šalje (izvorišni čvor); t_{prop} - vreme propagacije signala; i R - bitska brzina.

Za kanal sa slučajnom distribucijom greške na nivou bitova, definisanom verovatnoćom greške p_b , u stanju smo da odredimo optimalnu dužinu paketa n_{opt} koja odgovara maksimalnoj vrednosti CE nalaženjem prvog izvoda po parametru ukupna dužina paketa i izjednačavanjem sa nulom.

$$\frac{d}{dn_{total}} [CE] = 0 \quad (11.8)$$

$$\frac{d}{dn_{total}} \left[(1 - p_b)^{n_{total}} \cdot \frac{n_{total} - n_{po}}{n_{total} + n_{ACK} + 2 \cdot (t_{proc} + t_{prop}) \cdot R} \right] = 0 \quad (11.9)$$

da bi pojednostavili izračunavanje obeležićemo sa $a = 1 - p_b$ i $b = n_{ACK} + 2 \cdot (t_{proc} + t_{prop}) \cdot R$ tako da imamo da je

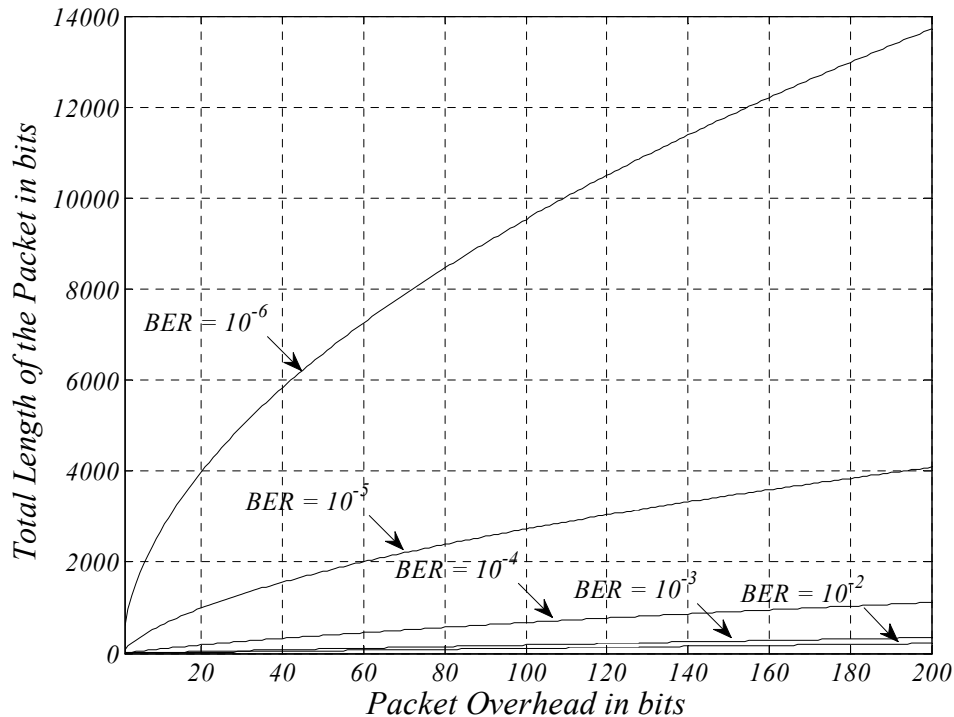
$$\frac{d}{dn_{total}} \left[a^{n_{total}} \cdot \frac{n_{total} - n_{po}}{n_{total} + b} \right] = 0 \quad (11.10)$$

$$n_{total}^2 + n_{total} \cdot (b - n_{po}) - b \cdot n_{po} + \frac{b + n_{po}}{\ln a} = 0 \quad (11.11)$$

pa rešavanjem jednačine (11.11) po n_{total} dobijamo n_{opt}

$$n_{opt} = \frac{n_{po} - b \pm \sqrt{(b - n_{po})^2 - 4 \cdot \left(\frac{b + n_{po}}{\ln a} - b \cdot n_{po} \right)}}{2} \quad (11.12)$$

Bez narušavanja opštosti, u analizi koja sledi, definisaćemo neke vrednosti u skladu sa hardverskim, topološkim i protokol rešenjem koja su primenjena u predloženom projektovanju bežične senzorske mreže. Tako na primer imamo da je bitska brzina $R = 250000bps$, $n_{ACK} = 80$ bitova, $t_{proc} = 10^{-3}s$, (vrednost dobijena za prosečno rastojanje između predajnog i prijemnog čvora od $80m$). Ostale vrednosti u izrazu kao što su verovatnoća greške p_b i redundantnost paketa u bitovima n_{po} , koje su inače promenljive vrednosti i zavise od kvaliteta kanala i odabrane tehnike kodiranja, poslužiće nam kao parametri u analizi. Dakle, pretpostavićemo da verovatnoća greške p_b varira u opsegu od 10^{-6} do 10^{-2} , a da redundantnost paketa uzima vrednost od 64bita do 1024 bitova.



SI. 11-3 Optimalna dužina paketa u funkciji redundantnosti paketa i verovatnoće greške (BER) kao parametara

Na Sl. 11-3 prikazana je optimalna dužina paketa, n_{opt} u funkciji redundantnosti paketa, n_{po} i verovatnoće greške p_b kao parametara. Tako na primer, ako za redundantnost paketa uzmemo vrednost $n_{po} = 72$ bita i analiziramo njen uticaj kroz verovatnoću greške u koracima $p_b = 10^{-6} / 10^{-5} / 10^{-4} / 10^{-3} / 10^{-2}$, dobijamo da je maksimalno prihvatljiv ukupan broj bitova u paketu u koracima $n_{total} = 8001 / 2259 / 521 / 130 / 77$, respektivno prema vrednosti p_b .

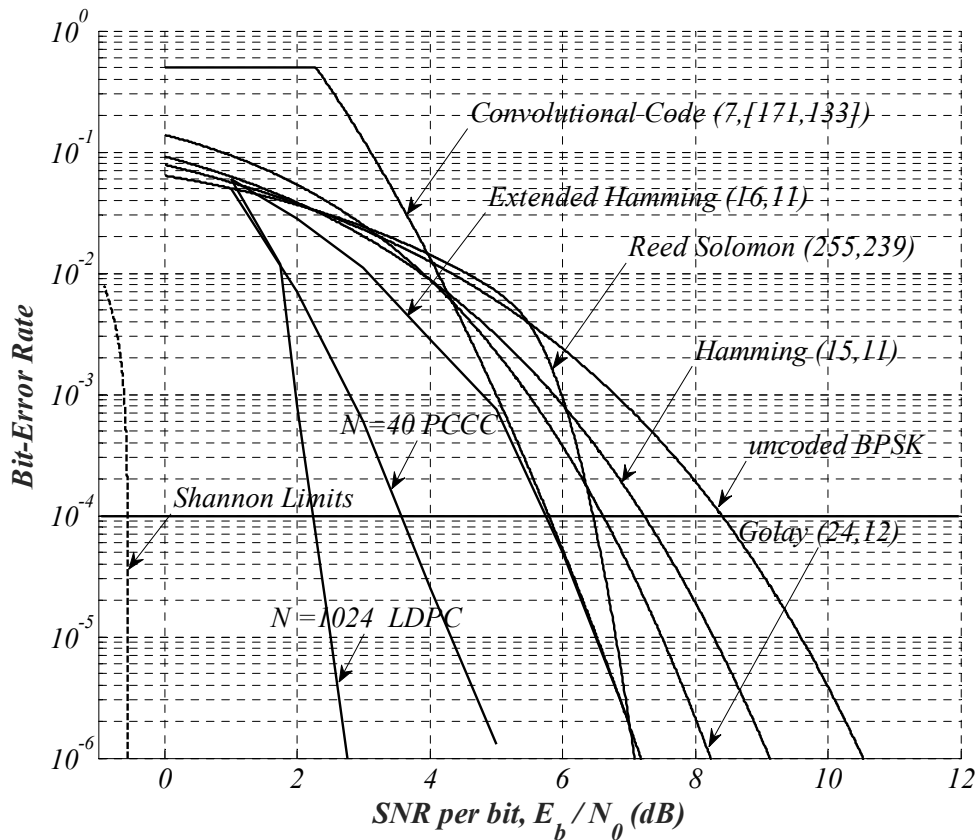
Ukoliko se pak prenos ostvaruje u teškim radnim uslovima okruženja, tipično je da imamo verovatnoću greške $BER \leq 10^{-4}$, što nam ukazuje da bi optimalna vrednost n_{total} trebala da bude u opsegu od 521 do 130 bitova. Sa druge strane, prema jednačinama (11.1) i (11.2) dužina redundantnosti paketa označena kao n_{po} trebala bi biti od 120 do 30 bitova, s čime je u skladu i vrednost sa početka analize.

Konačno, analizom rezultata prikazan na Sl. 11-3 u opštem smislu, možemo da zaključimo sledeće:

- i) Optimalna dužina paketa, n_{opt} je veća za niže vrednosti verovatnoće greške p_b . To znači da se efikasnost kanala povećava kako verovatnoća greške opada, što implicira da je broj paketa koji se ponovo šalje manji;
- ii) Kako se redundantnost paketa, n_{po} povećava, optimalna dužina paketa, n_{opt} se takođe povećava. Međutim, ovaj porast je znatno niži u odnosu na uticaj koji ima verovatnoća greške, p_b .

11.2 Kriterijumi za selekciju šema kodiranja za ispravljanje grešaka

Na bazi zaključaka potenciranih tokom prethodne diskusije očigledno je da je moguće ispraviti greške prisutne u primljenim porukama koje su posledica prenosa preko kanala okarakterisanog prisustvom šuma ako se, kao šeme kodiranja, koristi *ECC* kodovi. Ovo znači da mogu da se obezbede bolje *BER* performanse pri istom odnosu *SNR* u poređenju sa sistemom kod koga nije primenjena tehnika kodiranja. Odnosno posledično, upotreba *ECC*-a nam pruža mogućnost da prenosimo podatke sa nižim *SNR*-om za isti *BER* [79] u odnosu na sistem kod koga nije primenjena tehnika kodiranja ili je korišćena *ARQ* metoda sa retransmisijom. Da bismo mogli da izvršimo procenu dobiti uvođenja *ECC* šeme kodiranja i primenimo uporednu analizu sa drugim šemama kodiranja koristićemo performansnu metriku kodno pojačanje ili dobit kodiranja (*coding gain*).



Sl. 11-4 Performanse BER vs SNR za nekoliko tipova kodova za ispravljanje grešaka

Sl. 11-4 ilustruje BER u zavisnosti od SNR tokom prenosa preko AWGN kanala za nekoliko tipova ECC šema kodiranja. Napomenimo da je ova analiza zasnovana na modelu AWGN kanala i ima za cilj da nam pruži uvid u prednosti i mane nekih od standardnih tipova ECC šema kodiranja, a da je celokupna analiza predložene šeme kodiranja, u ostatku disertacije, bazirana na BSC modelu kanala. Na slici SNR je prikazan u decibelima [dB] i predstavlja SNR po bitu što je u ovom slučaju ekvivalentno E_b/N_0 , gde je E_b energija po bitu, a N_0 spektralna gustina šuma.

Konceptualno gledano SNR i E_b/N_0 imaju isto značenje. Razlika samo postoji što se SNR koristi za prikaz odnosa signala i šuma za analogne signale, dok je E_b/N_0 odnos signala i šuma za digitalni signal za određeni propusni opseg (BW). E_b je energija po jednom bitu, a N_0 je spektralna gustina šuma (proističe iz izraza N/BW , gde je N snaga šuma, a BW propusni opseg).

Opšti izraz (u dB) kojim se definiše relacija između SNR i E_b/N_0 je dat kao:

$$SNR(dB) = E_b/N_0 + 10 \cdot \log_{10}(R_s) + 10 \cdot \log_{10}(k) + 10 \cdot \log_{10}(R) - 10 \cdot \log_{10}(BW_n) \quad (11.13)$$

gde je R_s brzina prenosa simbola, k broj informacionih bitova po simbolu, R je kodni odnos (*code rate*) i BW_n je propusni opseg šuma. Tako na primer, za nekodirani *BPSK* modulacioni sistem imamo da je kodni odnos $R = 1$, broj informacionih bitova po simbolu $k = 1$ i ako izjednačimo R_s sa BW_n što jeste slučaj, onda imamo da važi:

$$SNR(dB) = E_b/N_0 \quad (11.14)$$

Tokom analize korišćenja je *BPSK* modulacija i za nekodirane i za kodirane bitove. U četiri kodirana blok koda spadaju (15,11) *Hamming* kod, (16,11) *Extended Hamming* kod, (24,12) *Golay* kod i (255,239) *Reed-Solomon* kod. Nekoliko moćnih kodova kao što su (7,[171,133]) *Convolution* kod, $N=40$ *PCCC (Parallel Concatenated Convolution Code)* kod i *1024 LDPC* kod su takođe prikazani na slici. *ECC* dobit na ciljanom *BER*-u od 10^{-4} se nalazi duž pune horizontalne linije određivanjem *SNR* razdaljine između tačke preseka sa izabranim *ECC* kodom i sa nekodiranim *BPSK* modulisanim bitovima.

Prilikom izbora *ECC* koda neophodno je napraviti kompromis između nekoliko faktora koji su postavljeni pred projektantom, a dominantni su dobit kodiranja, kompleksnost dekodera i potrošnje snage (kao implementacioni faktori). Da bi uspešno sagledali sve aspekte, istraživači obično koriste *DEB (Decoding Energy per Bit)* metriku da bi korektno izvršili poređenje između performansi *ECC* kodova (metrika zanemaruje potrošnju energije tokom procesa kodiranja) definisanu kao:

$$DEB = \frac{P}{T \cdot V_{dd}^2} \quad (11.15)$$

gde je P potrošnja snage dekodera na protoku T (*Mbps*) i V_{dd} je napon napajanja.

U radu [79] autori, što je u skladu sa našim istraživanjima vidimo na Sl. 11-4, pokazuju da je, na primer:

- a) nepravilni digitalni *N-1024 LDPC* kod obezbeđuje najveću dobit kodiranja $G_{LDPC} = 6.1$, ali troši najviše energije za proces dekodiranja po bitu $DEB_{LDPC} = 0.3nJpb / V_{dd}^2$ i snagu dekodiranja $P_{LDPC} = 690mW$;

- b) za $(255,239)$ Reed-Solomon digitalni dekodirer dobijamo $G_{RS} = 2$, $DEB_{RS} = 0.12nJpb/V_{dd}^2$ i $P_{RS} = 58mW$; i
- c) za *Extended Hamming* kod $G_{EH} = 2.6$, $DEB_{EH} = 0.0027nJpb/V_{dd}^2$ i $P_{EH} = 0.027mW$.

Napomenimo da je *Extended Hamming* kod sa aspekta sposobnosti poznat kao kod koji koriguje jednostruke greške i detektuje dvostruke greške i u literature se često susreće u skraćenom zapisu *SECDED*. Identičan je šemi kodiranja koju koristimo u dizajnu samo pod oznakom *SEC-DED*.

Generalno, u odnosu na snažne kodove (kao što je *LDPC*) manje kompleksni kodovi, kao što je *Hamming*, obezbeđuju nižu dobit kodiranja, ali sa druge strane imaju niži *DEB* i nižu potrošnju potrebnu za proces kodiranja. Napomenimo da je ovo samo jedan aspekt koji se sagledava pri izboru šeme kodiranja, ali nikako ne smemo da zanemarimo i preostale kao što su aplikacija i karakteristike grešaka u bežičnom kanalu.

Jedna od osnovnih uloga bežičnih senzorskih mreža je da prikuplja podatke iz okruženja koje su u većini slučajeva periodične prirode. Tako na primer, svaka dva minuta podaci prikupljeni sa senzora kao što su temperatura, vlažnost, količina svetlosti ili drugo, se šalju od izvorišnog do odredišnog čvora formiranjem jednog paketa, tj. paketi se šalju između dužih vremenskih intervala.

Sa druge strane, prirodu grešaka u bežičnom kanalu možemo da sagledamo analitički, što je i učinjeno u Odeljku 11 i 11.1, ili razvojem mernih sistema i metoda kojim dobijamo bolje predviđanje performansi kanala. Zbog varijabilnosti parametara bežičnog prenosa, razvoj sveobuhvatne metode merenja za procenu performansi bežičnih senzorskih mreža je izuzetno izazovan i gotovo nemoguć poduhvat.

11.3 Vrednosti metrike *BER*

Čak i sa toliko mnogo potencijalnih izvora greške, današnje mreže često moraju da obavljaju svoje aktivnosti sa *BER*-om reda 10^{-12} ili nižim, toliko nižim da ih je teško meriti. *BER* od 10^{-12} se translira na jednu bit grešku na svakih 10^{12} prenetih bitova koji stignu na stranu prijemnika. Vrednosti *BER*-a predstavljaju verovatnoću da prijemnik pogrešno tumači bit, tako da 10^{-12} *BER* znači da svaki bit ima verovatnoću 10^{-12} da će biti pogrešan.

Iako vrednost *BER*-a od 10^{-12} zvuči kao vrlo mala treba imati u vidu da se, na primer, pri brzinama prenosa od 10 Gbps , javlja jedna greška na $100s$. Dakle, kada se sagleda *BER* na ovaj način, može se uvideti zašto je nizak *BER* toliko važan za komunikacioni sistem. Previše grešaka na

nivou bita dovodi do lošeg kvaliteta prenosa, izgubljenih podataka ili retransmisije podataka koja dovodi do neprihvatljive propusnosti podataka.

Postavlja se pitanje šta definiše prihvatljivi *BER*? Standardi definišu granice *BER* performansi, ali korisnici mreže teže nivoima *BER*-a koji su znatno ispod granica navedenih u standardima. Za bežične senzorske mreže standardizovanu vrednost *BER*-a je nemoguće definisati. Naime, širok spektar primene neminovno podrazumeva različite karakteristike kanala pa samim tim i vrednosti *BER*-a se kreću u opsegu od 10^{-12} do 10^{-2} .

11.4 BER testeri

Da bi verifikovali dizajn celokupne mreže, mrežnog podsistema ili mrežne komponente potreban nam je tester *BER*-a. U osnovi, svaki *BER* tester se sastoji iz generatora test sekvenci (obrazac slanja) i brojača grešaka. Brojač broji sve dolazeće bitove, vrši upoređivanje svakog bita sa referentnim obrascem, sračunava broj grešaka i koristeći se sledećom jednačinom izračunava *BER*.

$$BER = \frac{\text{Broj bitova grešaka}}{\text{Ukupan broj primljenih bitova}}$$

Da bi bili u stanju da merimo *BER* prvo moramo slati poznat obrazac (test sekvencu) bitova kroz medijum za prenos ili sistem koji se testira. Potreban nam je poznati obrazac, jer je brojaču grešaka u testeru neophodno da zna koji obrazac da očekuje. U suprotnom tester neće biti u stanju da odredi da li je pristigli bit dobar ili pogrešan.

Pogrešan pristup je da šaljemo obrazac koji ćemo stalno ponavljati kao što je na primer 10101010. Slanje takvog obrasca neće simulirati dinamiku saobraćaja koju će mreža imati tokom uobičajenog rada. Loše je poslati i duge sekvence čiji je uzorak 00001111. Naime ovaj signal izgleda kao takt na jednu četvrtinu bitske brzine prenosa pa u tom slučaju kola za detekciju takta na prijemnoj strani mogu pogrešno da se sinhronizuju sa takvim obrascem.

Dakle, očigledno je da nam je potreban neki izvor koji će biti u stanju da nam obezbedi sve kombinacije 1 i 0 u okviru određenog broja bitova. Većina standarda za potrebe testiranja predlaže korišćenje generatora pseudoslučajnih bit sekvenci (*PRBS*). Test sekvence moraju da budu dovoljno duge da simuliraju slučajne podatke i da se šalju brzinom podataka koja se testira. Naime ako je test sekvencu previše kratka ona će se ponoviti za veoma kratko vreme slanja što opet može da dovede u zabludu kolo za oporavak takta što će osujetiti prijemnik u sinhronizaciji podataka.

Bit testna sekvenca, gotovo u svim standardima, prati format $2^N - 1$ tako na primer, obrazac $2^9 - 1$ sadrži sve 511 moguće 9-bitne kombinacije osim kombinacije sve nule. Sekvenca se ponavlja tek nakon što generator test sekvenci pošalje sve moguće kombinacije.

E sad se postavlja pitanje kako meriti stopu grešaka na nivou bitova, *BER* i stopu grešaka na nivou paketa, *PER* za isti primopredajnik. Problem je u tome što ne postoji test režim rada/direktni režim rada koji nam dozvoljava da direktno merimo *BER*, ali postoje neki načini da dobijemo indikaciju granice osetljivosti i *PER*. Prvo to su neke opšte informacije o tome kako se *BER* ponaša:

1. Granica senzitivnosti je definisana tako da jačina primljenog signala daje *BER* od 0.1%
2. Da bi merili *BER*, amplitude primljenog signala se podešava tako da je stopa greške na nivou bitova 1:1000 (0.1%). Pošto nemamo direktan pristup demodulisanom nizu bitova, najlakši način je da merimo stopu grešaka paketa (*PER*), a onda izračunamo stopu grešaka na bit nivou (zavisi od broja bitova u paketu).

Nešto jednostavniji način da dobijemo indikaciju granice senzitivnosti je da za izlazni snagu od 0dBm i definisani izgled paketa *MCU* na prijemnoj strani čita sve primljene pakete i upoređuje ih sa originalnim paketom. Broj pogrešnih bitova daje prilično dobru tačnost vrednosti *BER*-a. Za relevantno utvrđivanje *BER*-a trebalo bi preneti najmanje 10000 bitova.

U cilju pronalaženja odgovarajućeg rešenja uz korišćenje *CC110L* kao *RF* primopredajnik izvršena su brojna merenja bez korišćenja *ECC* kodne šeme tako što je izvorišni čvor korišćen za kontinualno slanje 15000 paketa sa slučajno generisanim podacima ka odredišnom čvoru koji se nalazio na rastojanju od 60 m.

Ovo je primer simulacije sa tzv. *HIL* (*Hardware-In-the-Loop*) strukturom kojom se pojednostavljuje proces projektovanja složenih i skupih sistema u realnom okruženju sa velikim brojem čvorova koji moraju biti projektovani i testirani pojedinačno, a zatim integrisani u jednu mrežu.

Nakon sprovedenog eksperimenta zaključili smo da najveći udeo u greškama imaju jednostruke bit greške (97%), da se dvostruke bit greške javljaju u znatno manjoj meri negde oko (2.8%), a da su višestruke/*burst* greške prisutne, ali u znatno manjoj meri. Dobijeni rezultati nam ukazuju na upotrebu šeme kodiranja koja može da ispravi sve jednostruke greške i, u najgorem slučaju, značajno smanji ili potpuno eliminiše dvostruke, višestruke/*burst* greške.

11.4.1 Metrike za procenu kvaliteta linka i energetske efikasnosti radio bloka

Uobičajena praksa je da se primenom modela obuhvate osobine i ponašanja pojedinih aspekata senzorskog čvora ili celokupne bežične senzorske mreže. Model, u kontekstu istraživanja sistema bežičnih senzorskih mreža, može se definisati kao apstrakcija funkcionalnog ponašanja bežične senzorske mreže i/ili senzorskog čvora, u obliku pogodnom za simulaciju ili analizu. Termin metrika se koristi da označi aspekte modela koji mogu da se mere ili kvantitativne vrednosti koje model predviđa.

Metrika, kao kriterijum za procenu, koristi se za procenu kvaliteta senzorskog čvora ili bežične senzorske mreže i uključuje: a) energetska efikasnost (prosečna disipacija snage za dati zadatak); b) pravovremenost (latencija ili kašnjenje za datu preduzetu operaciju); c) pouzdanost (koncepti koji se odnose na pouzdanost); i d) bezbednost (troškovi procesiranja usled nasilnog ataka na senzorski čvor ili bežičnu senzorsku mrežu).

Metrike u bežičnim senzorskim mrežama se kreću od: i) metrike za energetska efikasnost radio komunikacionog interfejsa; ii) metrike za kvantitativno iskazivanje performansi protokola za rutiranje; iii) metrike koje karakterišu svojstva cele bežične senzorske mreže (kao na primer pouzdanost); i iv) specifične metrike koje opisuju ponašanje u određenim aplikacijama.

Varijable i konstante koje utiču na ponašanje ili prirodu modela obično se nazivaju parametrima modela. Neki parametri mogu biti konstante dok drugi mogu varirati tokom životnog veka modela senzorskog čvora ili bežične senzorske mreže. Pored toga parametri se mogu kategorizovati kao kontrolisani (obuhvataju stavke kao što su podešavanje režima rada hardvera, parametri protokola ili parametri aplikacije) i kao nekontrolisani tj. oni na koje se ne može uticati (npr. temperature i vlažnost vazduha ambijenta).

Prema prethodno navedenom očigledno je često od interesa prilikom proučavanja modela ponašanja senzorskog čvora ili bežične senzorske mreže ne samo identifikovanje parametara i njihov uticaj na metrike sistema putem modela, već i određivanje korelacije između metrika bežičnih senzorskih mreža zbog zavisnosti između parametara modela.

U ovoj tezi od interesa nam je da tačno odredimo kvalitet bežičnog prenosa podataka u bežičnim senzorskim mrežama pod uslovom energetska efikasnog rada *Radio* bloka. U principu, za sagledavanje ova dva aspekta mreže obično se koriste sledeće pet primarne metrike:

- 1) Indikacija jačine primljenog signala (*RSSI - Received Signal Strength Indication*) - odgovara jačini signala koji stiže do prijemnika

- 2) Odnos signal interferencija plus šum (*SINR - Signal to Interference plus Noise Ratio*) - ukazuje na stepen do kojeg snaga primljenog signala nadmašuje sumu snage šuma plus smetnji na prijemnoj strani
- 3) Stopa ispravno isporučenih paketa (*PDR - Packet Delivery Ratio*) - kao indikator višeg nivoa (krupno zrnaste) strukture prenosa ukazuje na odnos ispravno primljenih paketa na strani prijemnika u odnosu na ukupan broj paketa koje je predajnik poslao.
- 4) Stopa pogrešno prenetih bitova (*BER - Bit Error Ratio*) - kao indikator nižeg nivoa (sitno zrnaste) strukture prenosa ukazuje na ukupan broj bitova koje je prijemnik primio sa greškom podeljen sa ukupnim brojem bitova koji su poslani prijemniku
- 5) Potrošnja energije (*EC - Energy Consumption*) - uglavnom implementirana korišćenjem različitih tehnika upravljanja napajanjem i potrošnjom kao što su faktor-ispune (*duty cycling*), gejtovanje napajanja (*power gating*) i dr. u cilju produženja životnog veka senzorskog čvora, a samim tim i cele bežične senzorske mreže. Za konstantan napon napajanja kao glavni indikator ove metrike koristi se prosečna struja u oznaci I_{ave} .

Generalno gledano nijedna od prethodno navedenih metrika se ne može samostalno koristiti za precizno određivanje kvaliteta bežičnog prenosa podataka uz primarni interes ostvarivanja energetske efikasnosti rada pošto svaka pojedinačno otkriva samo neke osobine kvaliteta bežičnog prenosa podataka. Na primer, *RSSI* ne pruža kvantitativnu informaciju o destruktivnoj interferenciji bežične veze. Zatim, u praksi je gotovo nemoguće odrediti tačnu vrednost *SINR* parametra. Kada veličina paketa i brzina prenosa variraju *PDR* metrika nije u stanju da da jedinstvenu procenu kvaliteta bežičnog prenosa podataka. Za merenje *BER* metrike, koja je u prethodnom delu disertacije detaljnije obrađena, predajnik treba da pošalje poznatu pseudoslučajnu sekvencu bitova što podrazumeva značajne režijske troškove obrade podataka. Tako da je neki prvi predlog upotreba kombinacije prethodno navedenih metrika na odgovarajući način kako bismo imali neki precizniji opis kvaliteta bežičnog prenosa podataka. Sve navedeno nam je potrebno za dalju evaluaciju predloženih rešenja koja sledi.

11.5 Energetski profil

U scenariju kada predajnik SN_i direktno komunicira sa pristupnom tačkom glavne aktivnosti, kao što je već analizirano u odeljku 8.1.2, su "*sense*" (senzovanje), "*data processing*" (procesiranje)

podataka) i "send/receive" (komunikacija). Shodno ovome, ukupna potrošnja energije u oznaci E_{tot} , može se izraziti kao:

$$E_{tot} = E_{SS} + E_{PS} + E_{COM} \quad (11.16)$$

gde su E_{SS} , E_{PS} , E_{COM} potrošnja energije senzorskog (*Sensor-*), procesnog (*Processing*) i radio (*Radio-*) podsistemskih (*Subsystem*) gradivnih blokova prikazanih na Sl. 7-1, respektivno.

Involviranjem *duty-cycling* tehnike, ukupna potrošnja energije predajnog senzorskog čvora je data kao:

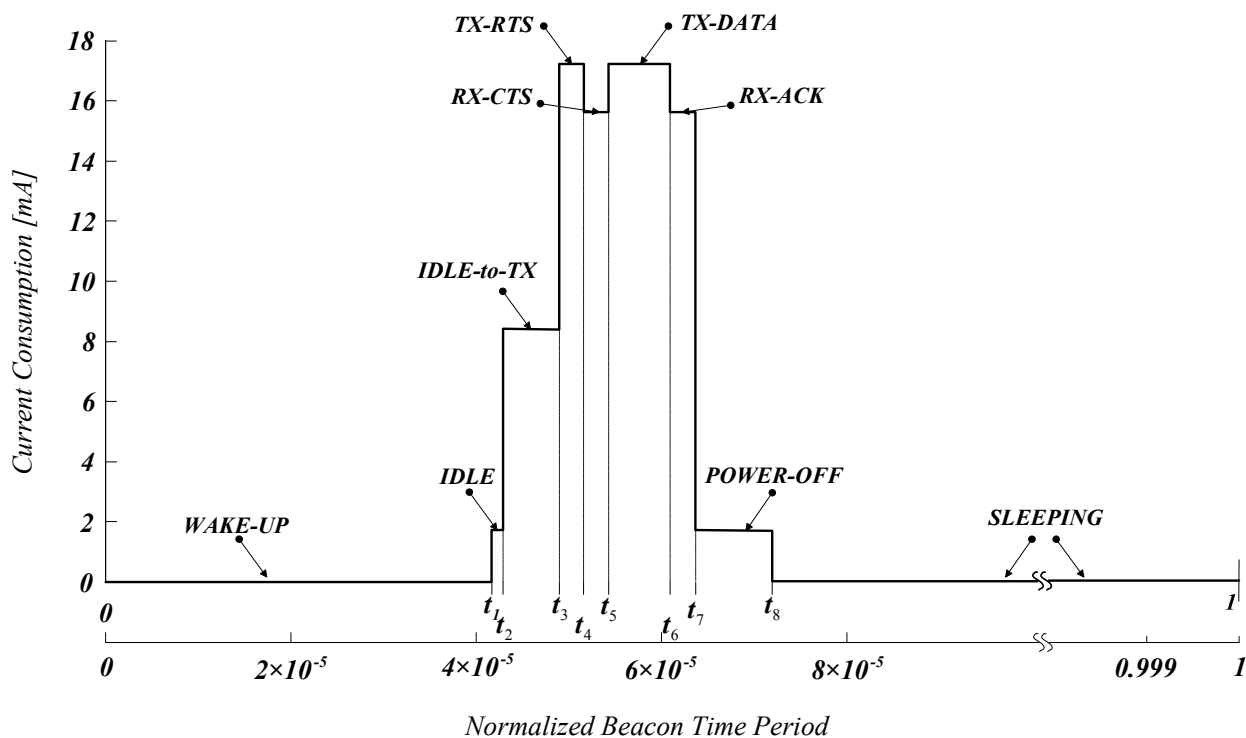
$$E_{tot} = E_{sense} \cdot \delta_{ss} + E_{proc} \cdot \delta_{proc} + (E_{Tx} + E_{Rx}) \cdot \delta_{com} + E_{sleep} \cdot (1 - \delta_{proc}) \quad (11.17)$$

gde su E_{sens} , E_{Tx} , E_{Rx} i E_{proc} potrošnje energije, svaki put kada senzorski čvor SN_i vrši *senzovanje*, *predaju*, *prijem* i *procesiranje podataka (obrada)* u toku vremenskog perioda T_{BP} . E_{sleep} je potrošnja energije senzorskog čvora SN_i u stanju spavanja (*sleep state*). Napomenimo da se vremenski period tokom kojeg je mikrokontrolerska jedinica aktivna preklapa sa zbirom vremenskih perioda aktivnosti *senzovanja* i *komunikacije* ($\delta_{proc} > \delta_{ss} + \delta_{com}$).

11.6 Profil aktivnosti radio podsistema u toku *Cu-St* faze

Kao prvo, u ovom trenutku, pretpostavićemo da su aktivnosti klasičnog *O-RPLL* protokola u fazama otkrivanja suseda (*Neighbor-Discovering (Ne-Di)*), registracija suseda (*Neighbor-Registration (Ne-Re)*) i ažuriranje planera (*Update-Scheduler (Up-Sc)*) već uspešno sprovedene i da se, u ostatku vremena, faza *Current-State (Cu-St)* beskonačno periodično izvršava. Njena perioda ponavljanja odgovara vremenu *beacon* perioda T_{BP} .

Opšti profil aktivnosti gradivnog bloka *Radio* podsistema koji važi za *O-RPLL* i *M-RPLL* protokole i normalizovanu vrednost *beacon* perioda $T_{BP} = 1$ prikazan je na Sl. 11-5. U našem predloženom dizajnu u okviru *Radio* podsistema koristi se *CC110L* [50] kao primopredajna komponenta. U Tab. 11-1 je dat opis aktivnosti *Radio* podsistema za svako stanje. Potrošnja struje za sva stanja, izražena u *mA*, kao i normalizovane vrednosti vremena trajanja pojedinih stanja u odnosu na *beacon* period dobijeni su iz dokumentacije komponente.



Sl. 11-5 Aktivnosti Radio podsistema u toku Cu-St faze

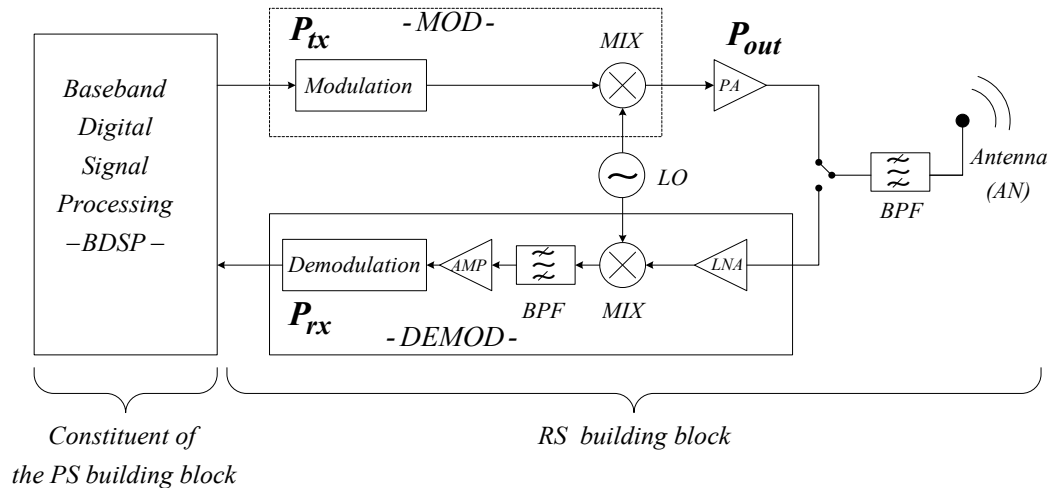
Tab. 11-1 Stanja Radio podsistema

Broj stanja	Naziv stanja	Opis aktivnosti	Vreme trajanja
Stanje 1	WAKE-UP	prelazak iz <i>POWER-OFF</i> u <i>IDLE</i> stanje, uključivanje CLS_{RT} (vidi Sl. 7-2)	$T_{WU} = t_1 - 0$
Stanje 2	IDLE	uključivanje digitalnog segmenta i startovanje lokalnog oscilatora radio primopredajnika <i>RT</i> (vidi Sl. 7-2)	$T_{ID} = t_2 - t_1$
Stanje 3	IDLE-to-TX	uključivanje sintetizatora frekvencije primopredajnika <i>RT</i> (vidi Sl. 7-2)	$T_{IT} = t_3 - t_2$
Stanje 4	TX-RTS	režim rada predaje, slanje poruke <i>RTS</i> (vidi Sl. 10-3)	$T_{RTS} = t_4 - t_3$
Stanje 5	RX-CTS	režim rada prijema, prijem poruke <i>CTS</i> (vidi Sl. 10-3)	$T_{CTS} = t_5 - t_4$
Stanje 6	TX-DATA	režim rada predaje, slanje poruke <i>DATA</i> (vidi Sl. 10-3)	$T_{DATA} = t_6 - t_5$
Stanje 7	RX-ACK	režim rada prijema, prijem poruke <i>ACK</i> (vidi Sl. 10-3)	$T_{ACK} = t_7 - t_6$
Stanje 8	POWER-OFF	isključivanje primopredajnika <i>RT</i> (vidi Sl. 7-2)	$T_{PO} = t_8 - t_7$
Stanje 9	SLEEPING	isključivanje CLS_{RT} (vidi Sl. 7-2)	$T_{SL} = 1 - t_8$

Napomena: Razlika između O-RPLL i M-RPLL protokola ogleda se u vremenu trajanja aktivnosti TX-RTS, RX-CTS, TX-DATA i RX-ACK kao posledica korišćenja različitih šema kodiranja podataka. Komutacija između režima rada predaje i režima rad prijema kao aktivnost postoji i nije zanemarljiva.

11.7 Arhitektura Radio bloka

Na narednoj Sl.11-6 dat je jednostavan pogled na arhitekturu *Radio* bloka visokog nivoa.



Sl. 11-6 Pogled visokog nivoa na radio arhitekturu

Slično kao i u [30] radio arhitektura u našem dizajnu se sastoji od dva relevantna dela, bloka *Radio* podsistema i *Baseband DSP (BDSP)* logike koja predstavlja hardversko/softverske komponente podsistema bloka procesiranja. *BDSP* logika obavlja proces kodiranja/dekodiranja podataka za slanje/prijem. Sastavni delovi bloka *Radio* podsistema su blokovi za modulaciju/demodulaciju (*MOD/DEM*), pojačavač snage (*PA*) i antena. Čak i sa ovako jednostavnom šemom arhitekture *Radio* podsistema na relativno visokom nivou, u stanju smo da primenimo nekoliko tehnika čime bismo smanjili potrošnju energije.

U tom cilju pokušaćemo da ukupnu potrošnju energije smanjimo tako što ćemo smanjiti potrošnju energije koju svaki deo, sa Sl. 11-6, troši samostalno ne narušavajući ukupnu funkcionalnost bloka.

Naime u *BDSP* najčešće se primenjuju tehnike nad kojima male potrošnje kao što su *clock/power gating*, dovtljivi pristupi u arhitekturi i algoritmima koji uključuju kodiranje kanala itd. U bloku *Radio* podsistema primenjuju se različite modulacione tehnike (binarne, *M*-arne, itd) u kombinaciji sa različitim arhitekturama predajnika i prijemnika (direktni, heterodinski, itd) za

implementaciju izabrane modulacione šeme. Pored toga, smanjenje potrošnje energije može se postići korišćenjem različitih arhitektura pojačavača snage (pojedinačni pojačavač snage, višestruki pojačavač snage). U našem predloženom dizajnu, strategija kodiranja/dekodiranja kontrole grešaka se implementira u *BDSP* bloku. Naravno, za korišćenje ovog koncepta potrebna je veća energija u *BDSP*-u, ali ovo povećanje omogućava nam da smanjimo energiju koja se zahteva od strane pojačavača snage za fiksnu stopu grešaka na nivou bitova (*BER*).

11.8 Model potrošnje energije *Radio* arhitekture

Pretpostavimo da je tokom vremenskog perioda od jedne godine T_Y *Cu-St* faza izvršena N_Y puta. Pod ovom pretpostavkom potrošnja energije radio arhitekture E_{radio} je data kao:

$$E_{radio} = N_y \cdot \left[\begin{array}{l} (P_{WU} \cdot T_{WU} + P_{ID} \cdot T_{ID} + P_{IT} \cdot T_{IT}) \\ + (P_{tx} + P_{out}) \cdot (T_{RTS} + T_{DATA}) \\ + P_{rx} \cdot (T_{CTS} + T_{ACK}) \\ + P_{PO} \cdot T_{PO} \\ + P_{BDSP} \cdot T_{BDSP} \\ + P_{SL} \cdot (1 - \delta_{com}) \cdot T_{BP} \end{array} \right] \quad (11.18)$$

gde je: P_{tx} / P_{rx} potrošnja snage predajnika/prijemnika, P_{out} - izlazna predajna snaga, P_{WU} - snaga buđenja (*wake-up*) radio primopredajnog bloka, P_{ID} - snaga u *idle* stanju, P_{PO} - snaga isključenja (*switching-off*) radio primopredajnog bloka, P_{BDSP} - snaga potrebna za proces kodiranja/dekodiranja polja podataka *Data field*, P_{SL} - snaga disipacije u *SLEEPING* stanju, $(T_{RTS}, T_{DATA}) / (T_{CTS}, T_{ACK})$ - su vremena tokom stanja predaja/prijem (realno vreme predaje/prijem podataka), T_{WU} - vreme uključenja (*turn-on*) CLS_{RT} (vidi sliku Sl. 7-2), T_{ID} - vreme provedeno u *idle* stanju, T_{IT} - vreme uključenja (*turn-on*) sintetizatora frekvencije, T_{PO} - vreme potrebno da se isključi (*turn-off*) primopredajni blok, T_{BDSP} - vreme potrebno za proces kodiranja/dekodiranja polja podataka *Data field*.

Kao posledica jednačine (11.18) imamo da je faktor ispunje bloka *Radio* podsistema:

$$\delta_{com} = \frac{T_{WU} + T_{ID} + T_{IT} + T_{RTS} + T_{CTS} + T_{DATA} + T_{ACK} + T_{PO}}{T_{BP}} \quad (11.19)$$

Napomenimo da u okviru vremena T_{proc_act} postoje sledeće aktivnosti bloka podsistema za procesiranje (obradu):

- i) Prikupljanje podataka od podsistema senzora;
- ii) Procesiranje podataka (kodiranje/dekodiranje, opsluživanje protokola, itd.); i
- iii) Komunikacija sa radio primopredajnim blokom putem *SPI* interfejsa kao digitalne komunikacione periferije mikrokontrolera.

Tokom ovih aktivnosti, samo deo vremena od ukupnog aktivnog vremena T_{proc_act} označen kao T_{BDSP} koji se odnosi na kodiranje/dekodiranje i komunikaciju sa blokom *Radio* primopredajnika.

U našem dizajnu približno važi da je $T_{BDSP} = 0.1 \cdot T_{proc_act}$.

Ukupna potrošnja energije radio bloka, za period od jedne godine, je data kao:

$$E_{radio} = V_{cc} \cdot I_{ave} \cdot T_y = V_{cc} \cdot I_{ave} \cdot N_y \cdot T_{BP} = N_y \cdot E_{beacon_period} \quad (11.20)$$

gde je I_{ave} prosečna struja radio bloka u toku vremenskog perioda T_{BP} , a V_{cc} je napon napajanja.

Polazeći od prethodne jednačine sada se jednačina (11.18) može napisati kao:

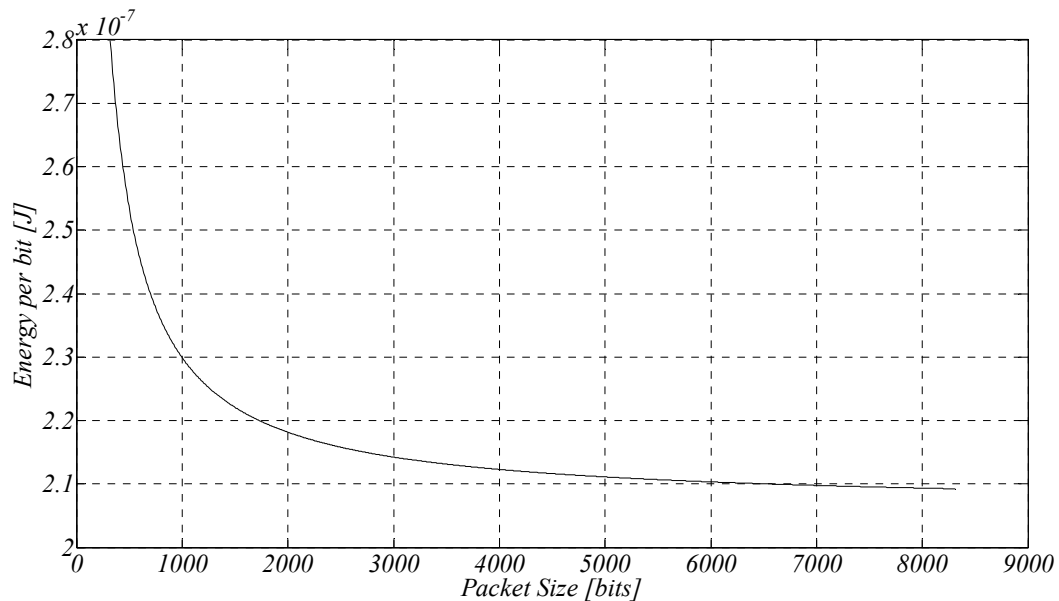
$$E_{radio} = N_y \cdot \left[\begin{array}{l} (V_{cc} \cdot I_{WU} \cdot T_{WU} + V_{cc} \cdot I_{ID} \cdot T_{ID} + V_{cc} \cdot I_{IT} \cdot T_{IT}) \\ + (V_{cc} \cdot I_{tx} + V_{cc} \cdot I_{out}) \cdot (T_{RTS} + T_{DATA}) \\ + V_{cc} \cdot I_{rx} \cdot (T_{CTS} + T_{ACK}) \\ + V_{cc} \cdot I_{PO} \cdot T_{PO} \\ + V_{cc} \cdot I_{BDSP} \cdot T_{BDSP} \\ + V_{cc} \cdot I_{SL} \cdot T_{BP} \end{array} \right] = \frac{T_y}{T_{BP}} V_{cc} \cdot \left[\begin{array}{l} (I_{WU} \cdot T_{WU} + I_{ID} \cdot T_{ID} + I_{IT} \cdot T_{IT}) \\ + (I_{tx} + I_{out}) \cdot (T_{RTS} + T_{DATA}) \\ + I_{rx} \cdot (T_{CTS} + T_{ACK}) \\ + I_{PO} \cdot T_{PO} \\ + I_{BDSP} \cdot T_{BDSP} \\ + I_{SL} \cdot (1 - \delta_{com}) \cdot T_{BP} \end{array} \right]$$

$$= T_y \cdot V_{cc} \cdot \left\{ \frac{1}{T_{BP}} \cdot \left[\begin{array}{l} (I_{WU} \cdot T_{WU} + I_{ID} \cdot T_{ID} + I_{IT} \cdot T_{IT}) \\ + (I_{tx} + I_{out}) \cdot (T_{RTS} + T_{DATA}) \\ + I_{rx} \cdot (T_{CTS} + T_{ACK}) \\ + I_{PO} \cdot T_{PO} \\ + I_{BDSP} \cdot T_{BDSP} \\ + I_{SL} \cdot (1 - \delta_{com}) \cdot T_{BP} \end{array} \right] \right\} = T_y \cdot V_{cc} \cdot I_{ave} \quad (11.21)$$

U našem predloženom dizajnu struja *idle* stanja $I_{SL} = 1\mu A$ je određena hardverskom strukturom CLS_{RT} (vidi Sl. 7-2) sastavnog dela bloka radio podsistema. S obzirom na to da je vreme T_{BP} za četiri reda veličine veće u odnosu na vreme aktivnog stanja T_{AS} (videti Sl. 10-1 i Sl. 11-5) uticaj člana $T_y \cdot V_{cc} \cdot I_{SL} \cdot (1 - \delta_{com})$ koji se odnosi na energiju spavanja se nikako ne sme zanemariti tokom procene vrednosti E_{radio} .

11.9 Efekat vremena prelaza i njegov uticaj na sistem

Efekat vremena prelaza pod direktnim uticajem je konkretnog hardverskog rešenja i za naš predlog dizajna tj. odabrani modul radio primopredajnika $CC110L$ imaćemo da je prosečna energija po prenetom bitom, E_b u funkciji veličine paketa za sledeće radne parametre: $R = 250\text{ kbps}$, $P_{out} = 0\text{ dBm}$, $f_c = 915\text{ MHz}$, $V_{cc} = 3\text{ V}$, $I_{tx} = 17.2\text{ mA}$, $I_{rx} = 15.6\text{ mA}$, $T_{ID} = 150\text{ }\mu\text{s}$, $T_{IT} = 724\text{ }\mu\text{s}$ i $T_{PO} = 1\text{ ms}$ je prikazana na Sl. 11-7.



Sl. 11-7 Efekat vremena prelaza

Posmatrajući krivu na Sl. 11-7 možemo izvesti sledeće opservacije. Vreme pokretanja (*start-up*) $T_{SU} = T_{ID} + T_{IT}$ i vreme isključivanja (*turn-off*) T_{PO} imaju veliki uticaj na E_b , pošto senzorski čvorovi u mrežama imaju tendenciju da komuniciraju koristeći kratke pakete. Generalni zaključak je da kako se dimenzija paketa smanjuje potrošnjom energije dominira vreme prelaza $T_{TT} = T_{SU} + T_{PO}$, a

ne vreme aktivne komunikacije $T_{CT} = T_{RTS} + T_{CTS} + T_{DATA} + T_{ACK}$. Ovo znači da kada kreiramo energetske efikasne komunikacione protokole poželjno je da šaljemo pakete većih veličina, ali sa većom veličinom paketa povećava se, za fiksnu stopu pogrešnih bitova, broj pogrešno prenetih paketa, pa je cena koja se plaća zbog ponovnog slanja (retransmisije) velikih paketa znatno veća.

Pored toga, u cilju uštede energije obično se preduzima isključivanje radio primopredajnika tokom perioda neaktivnog vremena predaja/prijem $T_{IN} = T_{WU} + T_{TT} + T_{SLEEPING}$. Kao posledica toga, kada je radio primopredajnik ponovo potreban troši se velika količina energije da bi se blok ponovo uključio tj. probudio.

11.10 Potrebni komunikacioni režijski troškovi između protokola baziranih na *NO-FEC* i *FEC* tehnikama kodiranja za postizanje savršene komunikacione veze

U aplikacijama bežičnih senzorskih mreža, kao što je praćenje stanja životne sredine, neophodno je stvarne podatke preneti sa malom verovatnoćom greške i predvidljivom latencijom (kašnjenjem). Tokom ovog procesa, ključni parametar od interesa za korisnike je pouzdanost veze između predajnika i prijemnika. Pouzdani prenos podataka se može postići ili povećanjem izlazne snage predaje P_{out} radio predajnika ili uvođenjem *FEC* tehnike kodiranja nad podacima. Sa jedne strane, uvođenje *FEC* tehnike kodiranja povećava veličinu paketa, zahteva dodatno procesiranje podataka i dodatnu energiju na strani predajnika i prijemnika za predaju i prijem dodatnih bitova. Sa druge strane, povećanje izlazne snage predajnika P_{out} je krajnje neefikasno zato što povećava potrošnju snage senzorskog čvora čime drastično smanjuje životni vek baterije, a samim tim i samog senzorskog čvora.

Imajući u vidu sve prethodno pomenuto, dobar izbor predstavljala bi primena *FEC* tehnike kodiranja u odnosu na povećanje izlazne snage predajnika iz dva razloga. Prvi je već pojašnjen u prethodnoj diskusiji, a drugi je posledica sposobnosti korekcije grešaka u paketu, a time drastično smanjenje ponovnog slanja paketa.

Međutim izbor pravog *FEC* algoritma, iz skupa velikog broja poznatih ili kreiranje novog, i njegovo implementiranje u čvorove bežičnih senzorskih mreža ograničenog resursa nije trivijalan zadatak. U nastavku analiziraćemo i proceniti nekoliko *FEC* tehnika kodiranja u funkciji njihovih performansi oporavaka od grešaka i komunikacioni režijski troškovi. Da bismo procenili dobit

uključivanja *FEC* tehnike kodiranja u odnosu na *NO-FEC*, implementirali smo četiri tipa sinhronih *Rendezvous* protokola.

Prvi je već pomenuti *O-RPLL* i on koristi *CRC* kodnu šemu tj. baziran je na *NO-FEC* tehnici. Drugi je *M-RPLL* i on koristi *SEC-DED* kodnu tehniku u okviru produkt koda. Treći je *E_1-RPLL* i on koristi *CRC* kodnu šemu na nivou bitova i *erasure* kodiranje na nivou paketa. Poslednji protokol je *E_2-RPLL* i on koristi *SEC-DED* kodnu tehniku na nivou bitova i *erasure* kodiranje na nivou paketa. Napomenimo da su *M-RPLL*, *E_1-RPLL* i *E_2-RPLL* protokoli sa odgovarajućim kodnim šemama i tehnikama bazirani na *FEC* tehnici kodiranja.

Zahtevi za niskom potrošnjom energije, minimiziranje ponovnog slanja paketa, stepen kašnjenja, balansirano procesiranje i pouzdan prenos podataka u bežičnim senzorskim mrežama su ciljevi od primarnog značaja za nas. U analizi koja sledi razmotrićemo uticaj komunikacijskih režijskih troškova (*overhead*) po paketu kao i stopu koda (*code rate*) kao primarne metrike na osnovu kojih ćemo odabrati najbolje rešenje sa aspekta potrošnje energije senzorskog čvora.

U Tab. 11-2 prikazani su analitički dobijeni rezultati koji se odnose na komunikacione troškove i stopu koda za sve prethodno navedene tipove protokola. Komunikacioni režijski troškovi se računaju kao apsolutna razlika između ukupnog broja bitova neophodnih za prenos pojedinačnog paketa i dužine (broja) informacionih bitova ($k_r \times k_c$) u toku *beacon* perioda T_{BP} (videti Sl. 10-3 a)).

Analizirajući rezultate predstavljene u Tab. 11-2, možemo da zaključimo da najmanje komunikacione režijske troškove ima *M-RPLL* protokol. To podrazumeva da će za *M-RPLL* potrebna snaga za prenos pojedinačnog paketa po *beacon* periodu T_{BP} biti najmanja. Pored toga, stopa kodiranja *M-RPLL* protokola je najveća što znači da je ostvaren najbolji odnos između brzine prenosa podataka i propusnog opsega.

Napomenimo da je tokom analize korišćen *NO-FEC* za detekciju i *FEC* kod za detekciju/korekciju grešaka pri granularnosti na nivou bita koje se javljaju tokom čitavog prenosa podataka. Pod uslovima izražene interferencije paket mora više puta ponovo da se prenese što dovodi do dodatnog utroška energije za senzorski čvor koji koristi *NO-FEC* tehniku kodiranja i stoga je izbor *FEC* tehnike kodiranja pogodniji jer prevazilazi gubitke paketa. Ako sada odemo korak niže glavni problem je odabir efikasnije *FEC* šeme kodiranja među tri pomenute koja je implementirana u *M-RPLL*, *E_1-RPLL* i *E_2-RPLL* protokole, respektivno.

SEC-DED kod je vrlo efikasan u pogledu cene usled povećanog procesiranja i generisanja i memorijskih režijskih troškova (*overhead*), ali ima slabe performanse u pogledu bit grešaka tipa

burst (naime pojava ovakvih grešaka rezultira u pojavi više od jedne greške u okviru kodirane vrste, čime se onemogućuje korektno oporavljanje grešaka, posledica nemogućnosti korigovanja više od jedne greške po vrsti). Sa druge strane, *erasure* kodovi mogu tolerisati gubitak k pogrešnih uzastopnih vrsta, ali su previše procesorski i memorijski zahtevni za ograničene resurse senzorskog čvora.

Na kraju, uključivanjem *2D SEC-DED* kodiranja u *M-RPLL* protokol činimo dobar kompromis između stepena sposobnosti koda za korekcijom grešaka, potrošnjom energije, složenosti proračunavanja i kapaciteta memorije.

Tab. 11-2 Komunikacioni režijski troškovi i stopa koda za protokole bazirane na *NO-FEC* i *FEC* tehnikama kodiranja

Tip protokola	Tehnika kodiranja	Dim. IBF polja [bits]	Proš. po koloni k_{er} [bits]	Proš. po vrsti k_{ec} [bits]	Kom. overhead po T_{BP} [bits]	Code Rate	Hamming distanca na bit nivou	Sposobnost korek. greške na nivou simbola
<i>O-RPLL</i>	<i>CRC</i> (27,11)	$k_r = 11$ $k_c = 11$	0	16	416	0.407	$d^h = 6$	–
<i>M-RPLL</i>	<i>SEC-DED</i> (16,11)		5	5	326	0.524	$d^h = 4$ $d^v = 4$	$t_{\max} = k_r$
<i>E_1-RPLL</i>	<i>CRC</i> (27,11)		5	16	623	0.280	$d^h = 6$	$t = 2$
<i>E_2-RPLL</i>	<i>SEC-DED</i> (16,11)		5	5	399	0.473	$d^h = 4$	$t = 2$

Napomena: Ako je bit na poziciji $d(i, j)$, $d(i, j+p)$, $d(i+t, j)$ i $d(i+t, j+p)$ (vidi Sl. 10-5), za $1 \leq t \leq k_c - 2$, $1 \leq p \leq k_r - 2$, $0 \leq i \leq k_c - 2$ i $0 \leq j \leq k_r - 2$ "flipovan" (predajnik šalje bit, a prijemnik prima inverznu vrednost) onda je nemoguće, korišćenjem *2D SEC-DED* tehnike kodiranja, da korigujemo ovakve šablone greške. Napomenimo da bi korišćenjem *erasure* šeme kodiranja ovakve šablone grešaka efikasno eliminisali.

11.11 Poređenje komunikacijskih režijskih troškova između *O-RPLL* i *M-RPLL* protokola za savršen link

Imajući u vidu da većina trenutno komercijalnih radio primopredajnika ima direktnu hardversku podršku za izračunavanje *CRC* (proces kodiranja i dekodiranja) u ostatku disertacije analiziraćemo performanse *O-RPLL* (koristi *CRC* kodiranje kao standardno rešenje) i *M-RPLL* (koristi *2D SEC-DED* kodiranje kao predlog rešenja) protokola i izvesti zaključke o dobitima uključivanja *2D SEC-DED* tehnike kodiranja.

Tokom ovog koraka pretpostavićemo sledeće vrednosti parametara od kojih su neki posledica odabranog radio primopredajnika *CC110L*:

- a) brzina prenosa podataka modula radio primopredajnika je 250 kbps tj. vreme trajanja bita je $4\mu\text{s}$
- b) format paketa *O-RPLL* je definisan na Sl. 10-2, a paketa *M-RPLL* na Sl.10-4 i Sl. 10-5.
- c) prenos podataka je bez grešaka, tj. kanal je perfektan (idealizovan kanal)
- d) broj senzorskih elemenata (senzora) postavljen u senzorski čvor posmatra se kao parametar i varira u rasponu od 2 do 16.
- e) rezolucija kojom se senzuju senzorski elementi posmatra se kao parametar i varira u rasponu od 8 bitova do 16 bitova (zavisi od izbora *ADC*)
- f) *Beacon* period T_{BP} posmatra se kao parametar i može biti 30 s , 60 s , 120 s , 180 s ili 300 s .
- g) koristi se *Gaussian frequency-shift keying (GFSK)* modulaciona šema

Komunikacioni režijski troškovi izraženi kao broj dodatnih redundantnih bitova u odnosu na informacione bitove za *O-RPLL* i *M-RPLL* protokole je dat u Tab. 11-3.

Tab. 11-3 Komunikacijski režijski troškovi

<i>Rezolucija ADC</i>	<i># Senzora</i>	<i># bitova po T_{BP} za O-RPLL</i>	<i># bitova po T_{BP} za M-RPLL</i>	<i>Kom. overhead # dodatnih bitova</i>
8	2	328	330	2
	4	344	356	12
	6	360	390	30
	8	376	416	40
	10	392	442	50
	12	408	476	68
	16	440	528	88
10	2	332	342	10
	4	352	372	20
	6	372	412	40
	8	392	442	50
	10	412	472	60
	12	432	512	80
	16	472	572	100
12	2	336	356	20
	4	360	392	30
	6	384	440	60
	8	408	476	68
	10	432	512	80
	12	456	560	104
	16	504	632	128
16	2	344	380	36
	4	376	424	48
	6	408	484	76
	8	440	528	88
	10	472	572	100
	12	504	632	128
	16	568	720	152

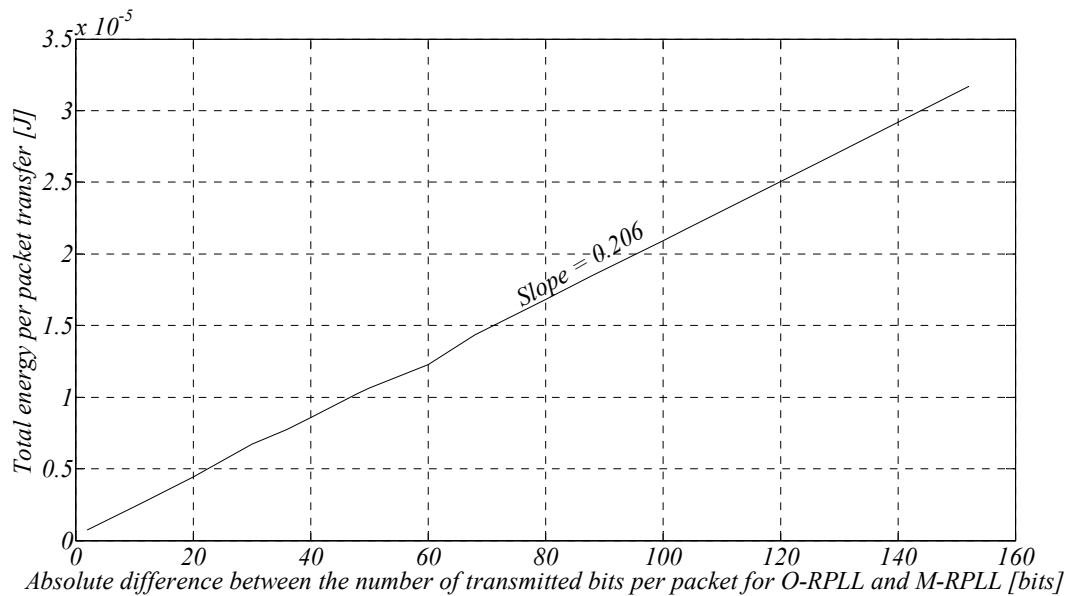
Napomena: komunikacijski režijski troškovi odgovaraju dodatnom broju bitova po prenetom paketu u okviru vremenskog intervala T_{BP} poredeći *M-RPLL* i *O-RPLL* protokole

Analizirajući dobijene rezultate, prikazane u Tab. 11-3, možemo zaključiti sledeće: Kada je prenos podataka bez grešaka, *O-RPLL* protokol je superiorniji u odnosu na *M-RPLL*. Niže performanse su posledica prenosa redundantnih bitova uključenih u *2D SEC-DED/SEC-DED* kodna polja (implementirano u *M-RPLL*) u poređenju sa fiksnim *CRC* poljem (koristi se u *O-RPLL* protokolu). U krajnim granicama komunikacijski režijski troškovi variraju u rasponu od 2 do 152 bita što dovodi do povećanja energije u predaji paketa. Ovo opravdava prethodno iskazanu tezu o odnosu dva protokola za slučaj kada je link savršen.

Takođe, prema podacima datim u Tab. 11-2, za uslov dat kao

$$2 \text{ bita} \leq \text{razlika_u_velicini_paketa} \leq 152 \text{ bita} \quad (11.22)$$

ukupna energija komunikacijskih režijskih troškova po prenetom paketu u funkciji apsolutne razlike između broja prenesenih bitova po paketu za *O-RPLL* i *M-RPLL* protokol prikazana je na Sl. 11-4.



Sl. 11-8 Komunikacijski režijski troškovi po prenetom bitu kao apsolutnoj razlici između *O-RPLL* i *M-RPLL* protokola

Analizirajući rezultate prikazane na Sl. 11-8 možemo zaključiti sledeće:

- 1) U predloženom dizajnu, ukupna energija komunikacijskih režijskih troškova je u granicama od $750nJ$ za 2 bita redundantnosti do $37.7\mu J$ za 152 bita redundantnosti

- 2) ukupna energija komunikacijskih režijskih troškova se linearno sporo povećava kako se povećava i broj redundantnih bitova. Zaključak je da se energija po prenetom bitu inkrementira za 0.206 i manja je od 1 što ukazuje da je efekat prelaznog vremena veoma izražen.

12 KOMPARATIVNA ANALIZA O-RPLL I M-RPLL PROTOKOLA

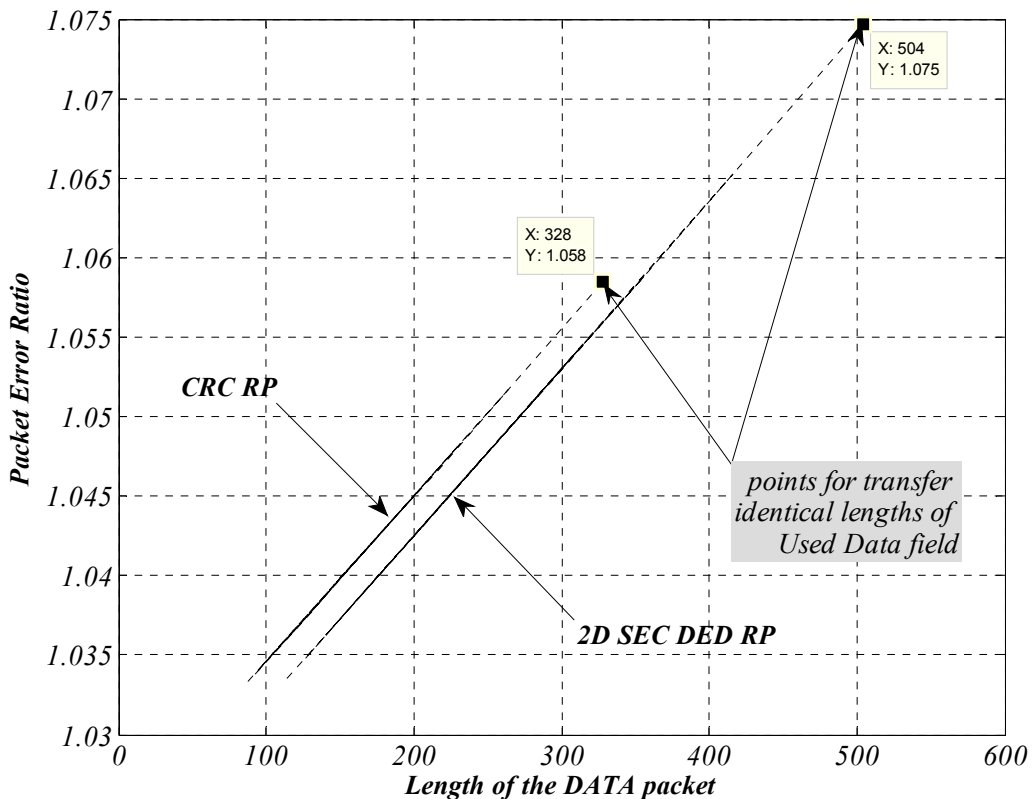
Pored već sprovedenih analiza od interesa je da utvrdimo, prvo kakav je uticaj dužine *DATA* paketa za fiksni *BER* na broj prenosa i , drugo potrebnu energiju po bitu kada je prenos podataka bez grešaka(e). Na bazi ovoga imaćemo još bolji i detaljniji pogled na predloženi sistem i biti u stanju da iz dodatnih uglova opravdamo primenu predloženog sistema.

12.1 Očekivani broj prenosa

Za datu verovatnoću P , definisanu jednačinom (11.5) očekivani broj prenosa N_t je definisan kao:

$$N_t = \frac{I}{1-P} = \frac{I}{1 + \left(1 - (1 - BER)^{n_{RTS}} \cdot (1 - BER)^{n_{CTS}} \cdot (1 - BER)^{n_{DATA}} \cdot (1 - BER)^{n_{ACK/NACK}}\right)} \quad (12.1)$$

U surovim uslovima okruženja sa aspekta prenosa podataka imamo da je tipično $BER = 10^{-4}$. Sada, u skladu sa prethodnom jednačinom moguće je odrediti očekivani, teorijski, broj prenosa za *O-RPLL* (baziran na *CRC*) i *M-RPLL* (baziran na *2D SEC-DED*) protokol u funkciji dužine paketa *DATA* koju tretiramo kao parametar. Tokom ovog proračuna, ostali parametri u jednačini su fiksirani, a to su dužina kontrolnih poruka $RTS = CTS = ACK / NACK$ za navedene protokole iznose 80 bitova za *O-RPLL* (*CRC*) i 72 bita za *M-RPLL* (*2D SEC-DED*). Dužina paketa *DATA* u oba protokola je promenljiva i određena je brojem senzorskih elemenata ($\#SE$) i rezolucijom *ADC*, dakle posledica je konkretnih aplikativnih potreba senzorskog čvora. Kako bi u analizi pokrili sve aspekte realizacije senzorskog čvora pretpostavićemo da broj senzorskih elemenata varira od 2 do 16, a rezolucija *ADC* je u rasponu od 8 do 16 bitova.



Sl. 12-1 Očekivani broj prenosa u funkciji dužine paketa *DATA* za *O-RPLL* i *M-RPLL* protokol

U cilju ostvarivanja korektnog poređenja, pretpostavićemo da oba protokola obavljaju isti posao, tj. prenose identičan broj bitova u polju *User Data* korisničke podatke (videti Sl. 11-1). Napomenimo da će se, kao posledica različitih šema kodiranja, njihova polja za proveru ispravnosti prenetih paketa razlikovati što će implicirati da su njihovi *DATA* paketi različitih dužina.

Dobijeni rezultati, prikazani na Sl. 12-1, odnose se na očekivani broj prenosa u funkciji dužine paketa *DATA*. Analizirajući grafik možemo da izvedemo sledeće zaključke:

- 1) kako se dužina paketa povećava broj prenosa se takođe povećava što za posledicu ima da se povećava *PER* (*Packet Error Rate*) stopa pogrešno prenetih paketa
- 2) zbog kraće dužine paketa broj prenosa za *O-RPLL* (*CRC*) protokol je uvek manji u odnosu na *M-RPLL* (*2D SEC-DED*). Na primer, kada polje *User Data* sadrži podatke senzorskog čvora sa 16 senzorskih elemenata (senzora) i *ADC* rezolucije od 16 bitova, dužina *DATA* paketa za *O-RPLL* će biti 328 bitova, dok će za *M-RPLL* biti 504 bita.

- 3) prema analitičkom modelu, zbog različitih dužina paketa, očekivani broj prenosa za *O-RPLL* (u najgorem slučaju) je manji za najviše 1,62% u odnosu na *M-RPLL*.

12.2 Energija po bitu pri prenosu podataka po linku bez grešaka

U uslovima prenosa podataka, po linku bez grešaka, za oba protokola (*O-RPLL* i *M-RPLL*) izvršićemo procenu energije po bitu (E_{pb}) u funkciji bitske brzine prenosa podataka (R), kao parametra, u opsegu od 600 bps do $500\,000 \text{ bps}$. Tokom ove analize, za *RF* primopredajni čip *CC110L* (kao gradivni blok *RS* modula), usvojicemo sledeće radne parametre (podaci su preuzeti iz dokumentacije): napona napajanja $V_{cc} = 3V$, dužina polja podataka $n_{DATA} = 128b$ (analiza sprovedena nad poljem od $\lceil (k_r \times k_c) \rceil_2 = \lceil (11 \times 11) \rceil_2 = \lceil 121 \rceil_2 = 128$), frekvencija nosioca $f_c = 915 \text{ MHz}$ i izlazna predajna snaga $P_{out} = 0 \text{ dBm}$ (potrošnja struje u predaji i prijemu je direktno zavisna od navedenih parametara i u predajnom režimu rada iznosi 17.2 mA , a u prijemnom vrednost struje pored navedenih parametara zavisi i od vrednosti bitske brzine i data je u [50]).

Tokom komunikacije prenosa podataka koristi se tradicionalna *RTS/CTS/DATA/ACK* četvorosmerna procedura rukovanja. Kako bi utvrdili kolika je energija po transferu neophodno je da izračunamo koliki je ukupan broj bitova koji se tokom jedne sesije prenosa razmeni.

Što se tiče dužine kontrolnih poruka $RTS = CTS = ACK$ su kreirane sa 80 bitova za *O-RPLL* i 72 bita za *M-RPLL* protokol, dok je *DATA* paket dužine 200 bita za *O-RPLL* i 288 bitova za *M-RPLL* protokol, što znači da za jedan transfer podataka tokom četvorosmerne procedure rukovanja imamo da je $n_{total_CRC} = 440$, a $n_{total_2D_SEC_DED} = 504$ bitova. Dobijeni rezultati su prezentovani na Sl. 12-2 i analizom možemo da izvedemo sledeće zaključke:

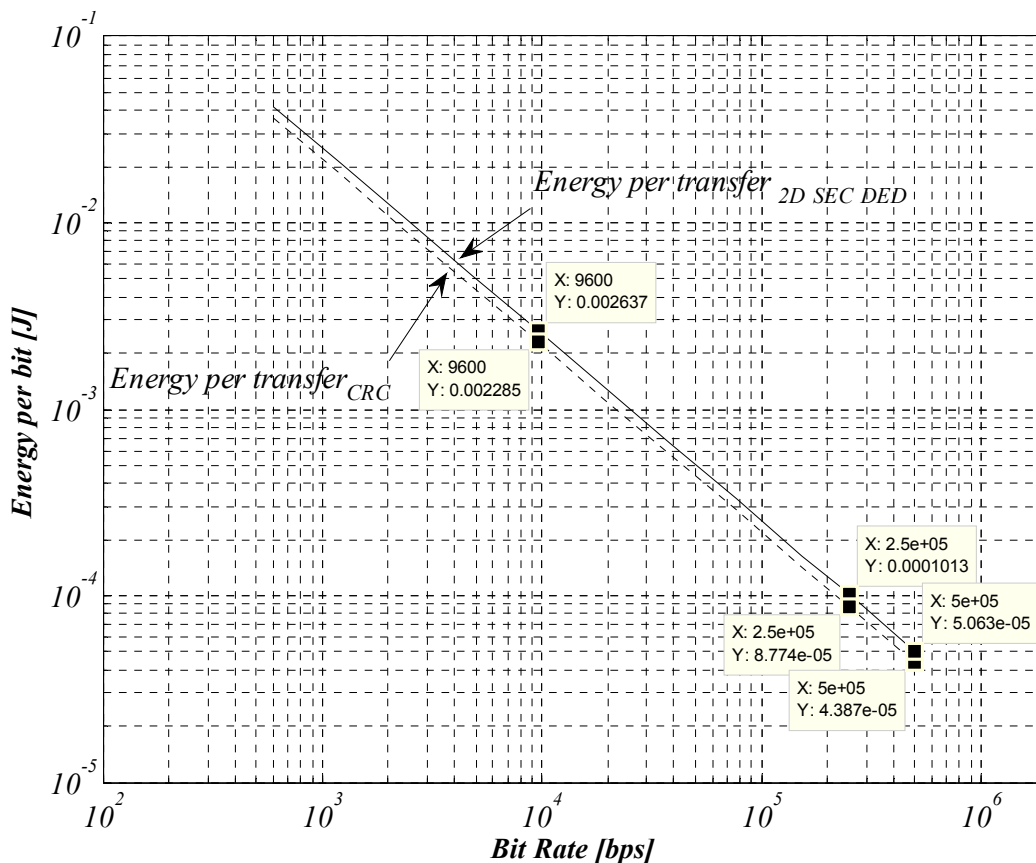
- Energija po transferu *O-RPLL* je uvek manja u poređenju sa energijom po transferu *M-RPLL* protokola. Niža vrednost je posledica manjeg broja bitova po pojedinačnom paketu prenosa za identičnu brzinu prenosa podataka.
- Energija po bitu, a samim tim i energija po transferu opada kako raste brzina prenosa podataka, važi za oba protokola.
- Ako označimo energiju po transferu za oba protokola kao, E_{T_CRC} za *O-RPLL* i E_{T_2D} za *M-RPLL* možemo definisati dobit G_{E_T} , kao novu performansnu metriku, na sledeći način:

$$G_{E_T} = \frac{E_{T_2D} - E_{T_CRC}}{E_{T_2D}} \cdot 100 \text{ [\%]} \quad (12.2)$$

Primenom predložene metrike na predloženo rešenje dobijamo da je faktor dobiti po transferu G_{E_T} konstantan i približno jednak 13.35%, nezavisno od bitske brzine R .

Nakon svih prethodno sprovedenih analiza u situaciji smo da iznesemo neke globalne zaključke.

Kao prvo, do sada smo samo diskutovali o implementaciji *O-RPLL (CRC)* i *M-RPLL (2D SEC-DED)* protokola, tj. *CRC* i *2D SEC-DED* kodnih šema, respektivno i utvrđivali performansnu metriku vezanu za efikasnost kanala. Međutim tokom ovih analiza korišćenjem analitičkog modela, efekti ponovnog slanja paketa tj. retransmisije nisu uzeti u obzir (pretpostavlja se da je prenos podataka bez greške – idealni uslovi). Napomenimo da u realnim uslovima čak i jedan pogrešan bit prisutan u paketu, sa primenjenom *CRC* šemom kodiranja, zahteva ponovno slanje paketa (retransmisiju) što sa aspekta potrošnje energije direktno povećava energetska neefikasnost senzorskog čvora. Kako nam je cilj da senzorski čvor učinimo maksimalno efikasnim, u nastavku ćemo proceniti performanse predloženog rešenja sa aspekta komunikacije i ukazati na prednosti i nedostatke korišćenja *2D SEC-DED* umesto *CRC* šeme kodiranja u okviru *RPLL* protokola.



Sl. 12-2 Energija po bitu u pogledu brzine bitova

12.3 Komunikacija preko linka koji nije idealan

Dakle, u realnim uslovima kao posledica različitih poremećaja u karakteristikama linka (šum, interferencija, itd.) komunikacija se ostvaruje preko linka koji nije idealan. Za precizno određivanje kvaliteta linka sada ćemo uključiti sledeća dva indikatora, koja su u međusobnoj direktnoj zavisnosti. Prvi, nazvan stopa odbačenih paketa (*PRJR – Packet Rejection Ratio*) je definisan kao:

$$PRJR [\%] = 100 [\%] - PDR[\%] \tag{12.3}$$

i odnosi se na aspekt linka sa prijemne strane gde *PRJR* ukazuje na procenat odbačenih paketa. Drugi indikator, *PDR* odgovara stopi uspešno isporučenih paketa sa aspekta predajne strane. Koristeći se ovim indikatorima za link možemo da damo sledeći opis stanja:

- ✓ link je "mrtav" ako ima stopu uspešno primljenih paketa (ekvivalentno stopi uspešno isporučenih paketa) od 0%

- ✓ link je loš ako ima stopu uspešno primljenih paketa (ekvivalentno stopi uspešno isporučenih paketa) $< 10\%$
- ✓ link je srednjeg kvaliteta ako ima stopu uspešno primljenih paketa (ekvivalentno stopi uspešno isporučenih paketa) između 10% i 90% , i
- ✓ link je dobar ako ima stopu uspešno primljenih paketa (ekvivalentno stopi uspešno isporučenih paketa) između 90% i 100%
- ✓ link je savršen ako ima stopu uspešno primljenih paketa (ekvivalentno stopi uspešno isporučenih paketa) od 100%

Detaljnija objašnjenja su sadržana u [80]. Takođe u [81] su prezentovani rezultati koji ukazuju da je u aplikacijama monitoringa životne sredine stopa odbačenih paketa (*PRJR*) u opsegu od 40% do 70% , što ukazuje da će primenom *CRC* šeme kodiranja procenat ponovnog slanja paketa, bar jednom, biti u opsegu od 60% do 30% . Ova činjenica nam sugeriše da je potrebno *ARQ* tehniku razmene podataka zameniti nekom od *FEC* tehnika kodiranja.

Imajući u vidu pak da *2D SEC-DED* šema kodiranja, kao predloženo rešenje, uključuje dodatnu potrošnju energije u procesima kodiranja podataka, prenošenju redundantnih bitova i dekodiranju podataka, cilj je sada da se odredi pod kojim uslovima je korišćenje *2D SEC-DED* šeme kodiranja prikladnije i opravdanije sa aspekta potrošnje energije u odnosu na *CRC* kodiranje za usvojeni model greške. Model greške, koji se koristi tokom procesa simulacije, treba da odslikava poznavanje procesa greške i njene statističke karakteristike jer samo na taj način u stanju smo da optimizujemo bežični prenosni sistem kako na nivou protokola tako i na nivou mehanizama za kontrolu grešaka i da verifikujemo robusne tehnike i koncepte koje smo predložili.

12.4 Analiza performansi *O-RPLL* i *M-RPLL* protokola u pogledu broja ponovljenih prenosa

Imajući u vidu da je prenos podataka sklon greškama, sada ćemo izvršiti procenu performansi oba navedena protokola u modeliranim realnim uslovima. Modelira se prenos podataka po jednom linku između senzorskog čvora i pristupne tačke. Za proces kodiranja podataka, ubacivanja (injektovanja) grešaka, detektovanja (otkrivanja) i korigovanja (ispravljanja) grešaka koristi se softverski simulacioni model kreiran u *MATLAB* softverskom alatu. Kvantitet jednostrukih grešaka i dvostrukih/višestrukih grešaka u okviru polja podataka određen je *MATLAB randerr* funkcijom korišćenjem modela binomne verovatnoće za generisanje stope grešaka na nivou bitova kao

parametrom [82]. U suštini, binomna funkcija se bavi celobrojnomo diskretnom funkcijom i stoga predstavlja odgovarajući model greške na nivou bitova.

Verovatnoća $P(i, m)$ implicira da postoji i grešaka u kodu ili paketu veličine m , tako da važi da je $i < m$, i data je binomnom funkcijom verovatnoće mase definisane kao:

$$P(i, m) = C_i^m \alpha^i \cdot (1 - \alpha)^{m-i} \quad (12.4)$$

gde je $\alpha = BER$, $C_i^m = \binom{m}{i} = \frac{m!}{i!(m-i)!}$, $i \in \{0, 1, 2\}$, $i = 0$ odgovara transferu koji nema grešaka,

$i = 1$ odgovara transferu koji sadrži jednostruku grešku na nivou bita, i $i = 2$ odgovara transferu koji sadrži dvostruke/višestruke greške na nivou bita. Napomenimo da, u opštem slučaju, promenljiva i može da uzme vrednosti od 0 do m . U našem slučaju i uzima vrednost do 2 jer ćemo sve višestruke greške (dve ili više) tretirati kao deo iste grupe.

U osnovi formula može da se shvati na sledeći način $P(i, m) = C_i^m \alpha^i \cdot (1 - \alpha)^{m-i}$ je verovatnoća i uspešnih događaja sa verovatnoćom α^i od m pokušaja i $(m - i)$ neuspešnih događaja sa verovatnoćom $(1 - \alpha)^{m-i}$, gde je m broj pokušaja i α verovatnoća uspešnosti za dati broj pokušaja.

Funkcija *randerr* [83] generiše binarnu matricu čiji su elementi nule i jedinice koju tumačimo tako da prisustvo bita jedinice na određenoj poziciji elementa u matrici označava da imamo grešku u prenosu i kao takvu je koristimo za testiranje sposobnosti kodnih šema za kontrolu greške. *MATLAB* komanda $x = randerr(N, M, [0, 1, 2; P_0, P_1, P_2])$ generiše $N \times M$ binarnu matricu koja ima svojstva da svaka vrsta sadrži nula '1' (ne sadrži jedinice) sa verovatnoćom P_0 , jednu '1' sa verovatnoćom P_1 , dve '1' sa verovatnoćom P_2 , itd.

Verovatnoća greške generisana na bazi prethodno navedenog modela korišćenjem jednačine (12.4) prikazana je u Tab. 12-1.

Tab. 12-1 Verovatnoća greške generisana na bazi modela

<i>BER</i>	<i>Probability P_i per row [%]</i>		
	P_0 (<i>error-free</i>)	P_1 (<i>single-bit error</i>)	P_2 (<i>double-/multiple-bits error</i>)
$\alpha = 10^{-3}$	81.131	16.973	1.896
$\alpha = 10^{-4}$	97.931	2.046	0.023
$\alpha = 10^{-5}$	99.791	0.208	0.001
$\alpha = 10^{-6}$	99.979	0.020	~ 0.001

Generalno, pošto *2D SEC-DED* šema kodiranja ima sposobnost da detektuje i koriguje sve jednostruke bit greške i delimično, zavisi od šablona pojave, dvostruke/višestrukе bit greške, moguće je smanjiti procenat ponovo poslatih paketa tj. retransmisiju za bilo koju vrednost *BER*-a. U skladu sa jednačinom (12.3) možemo da definišemo procenat retransmisija N_r kao:

$$N_r = (N_t - 1) \cdot 100 \text{ [%]} \quad (12.5)$$

gde je N_t , kako je već napomenuto, dato jednačinom (12.1).

12.5 Razumevanje metrike snage kod bežičnih senzorskih mreža

Naglasimo u startu da nije moguće upoređivati potrošnju energije senzorskog čvora za različite protokole korišćenjem jedne metrike. Na primer, ponekad se poređenje vrši po "maksimalnoj vrednosti struje u piku" tzv. vršnoj struji. Dok vršna struja igra ulogu u ukupnoj potrošnji snage, senzorski čvor na kome se izvršava protokol stek trošiće struju na nivou maksimalne vrednosti samo dok je u stanju predaje podataka. Čak i kod sistema koji imaju visoku propusnost, senzorski čvor se nalazi u stanju predaje samo mali procenat ukupnog vremena aktivnosti, tako da navedena metrika ne daje pravu sliku potrošnje senzorskog čvora tokom njegovog životnog ciklusa.

Pored stanja predaje senzorski čvor će, tokom velikog broja ciklusa u svom životnom veku, proći i kroz nekoliko drugih stanja kao što su stanje prijema, stanje spavanja, stanje buđenja iz spavanja, itd. Čak i ako je poznata trenutna potrošnja struje senzorskog čvora u svakom od navedenih stanja to još uvek nije dovoljna informacija da utvrdimo ukupnu snagu koja se potroši od strane čvora. Različiti nivoi protokol steka zahtevaju određenu među nivovosku obradu kako bi bili enkapsulirani i poštovali specifikacije protokola. *MCU* treba vremena da izvrši ovu obradu i tokom

tog vremena, struju troši procesorska jedinica, a ne primopredajni blok. Pored toga postoji i prelazno vreme prilikom prebacivanja čvora iz jednog u drugo stanje sa određenom potrošnjom struje. Dakle, sve navedeno se mora uzeti u obzir kako bi se dobilo tačno merenje ukupne potrošnje struje.

Tab. 12-2 Radni režimi mikrokontrolera MSP430FR5969 [49]

MODE	AM		LPM0	LPM1	LPM2	LPM3	LPM4	LPM3.5	LPM4.5	
	Active	Active, FRAM OFF	CPU Off	CPU OFF	STAN DBY	STAN DBY	OFF	RTC ONLY	SHD WITH SVS	SHD WITH OUT SVS
Max sys clk	16 MHz		16 MHz	16 MHz	50kHz	50kHz	0	50kHz	0	
tip. potroš nja struje	103 $\mu A / MHz$	65 $\mu A / MHz$	70 $\mu A @ 1MHz$	35 $\mu A @ 1MHz$	0.7 μA	0.4 μA	0.3 μA	0.25 μA	0.2 μA	0.02 μA
tip. vreme buđen ja	N/A		trajanje	6 μs	6 μs	7 μs	7 μs	250 μs	250 μs	1000 μs
događ aj buđen ja	N/A		svi	svi	LF IO Comp	LF IO Comp	IO Com p	RTC IO	IO	

U tipičnim aplikacijama senzorskih mreža, senzorski čvor, na kome se izvršava odgovarajući protokol stek, će najviše vremena provoditi u stanju spavanja, što nam je i primarna tehnika u uštedi energije, između događaja koji su posledica komunikacije. Naime, kada mikrokontroler MSP430FR5969 prelazi u optimizovani režim rada ultra male potrošnje pod nazivom *Real-Time Clock (LPM3.5)* onda je naponski regulator PMM prema jezgru potpuno isključen čime se dodatno obezbeđuje ušteda snage. Pošto nema napajanja prema V_{CORE} u toku LPM3.5 CPU i svi digitalni blokovi, uključujući i RAM, su isključeni. Ovo suštinski znači da je čitav uređaj neaktivan, a samim

tim i sadržaj registara i *RAM* memorije su izgubljeni. Da bi se ovo sprečilo, sve bitne vrednosti bi trebalo uskladištiti u *FRAM* pre ulaska u *LPC* režim rada. Dodatno pojašnjenje režima rada dato je u Tab. 12-2. Naime, *MSP430FR5969* ima jedan aktivni režim rada i sedam softverski selektabilnih korisnički orijentisanih režima rada male potrošnje. Pojava događaja koji izaziva prekid može da probudi mikrokontroler iz režima rada male potrošnje (*LPM0* do *LPM4*), opsluži zahtev i vrati se nazad, u režim male potrošnje, iz uslužno prekidne rutine. Režimi rada male potrošnje kao što su *LPM3.5* i *LPM4.5* onemogućuju napajanje jezgra da bi minimizirali potrošnju i budi ih događaj sa *RTC* ili *I/O* pina.

12.6 Prosečna potrošnja struje Radio podsistema u pogledu uticaja T_{BP} i *BER* za fiksne vrednosti broja senzorskih elemenata i ADC rezolucije

Da bi opravdali i validirali prednost primene predloženog *M-RPLL* u odnosu na *O-RPLL* protokol najpre ćemo izvršiti procenu prosečne struje *Radio* podsistema u pogledu uticaja T_{BP} i *BER*, pošto I_{ave} direktno određuje potrošnju energije radio bloka.

Tokom analize pretpostavićemo sledeće:

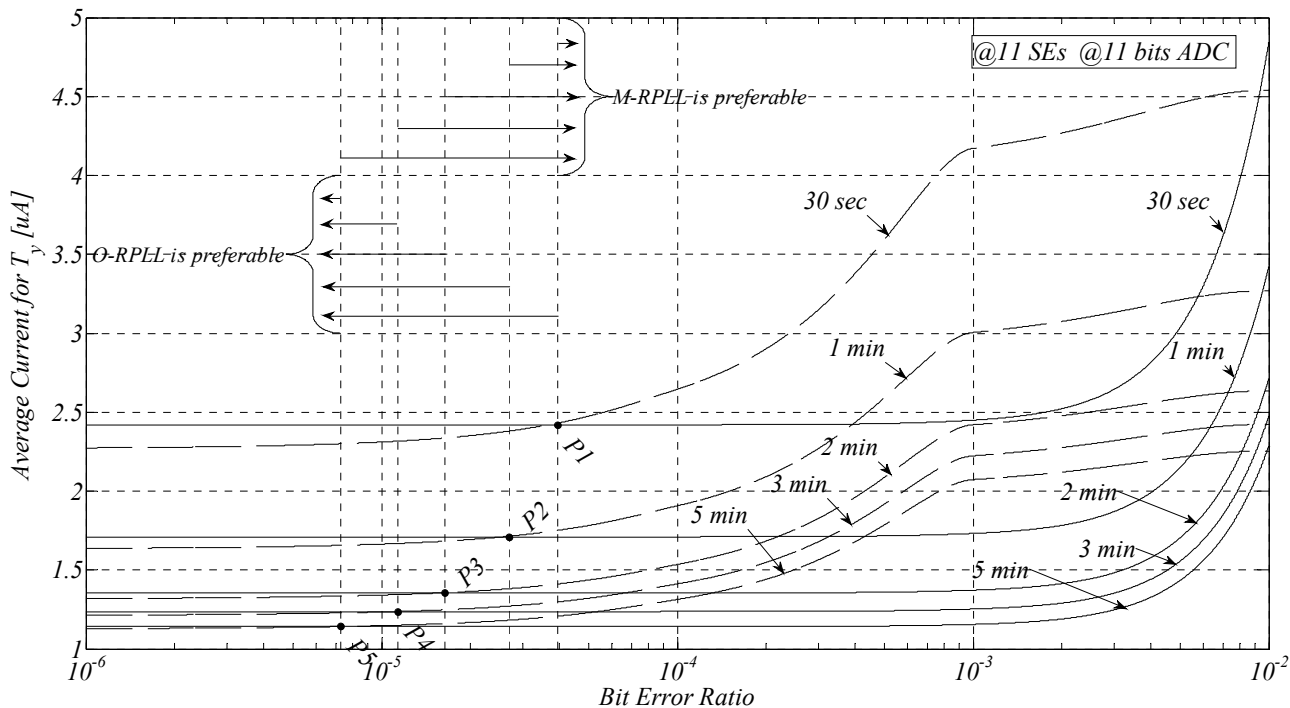
- Aktivnosti senzorskog čvora koji šalje u okviru vremenskog intervala T_{BP} definisane su na Sl. 11-5 (tj. *Cu-St* faza se izvršava na repetitivan način tokom vremenskog perioda T_y)
- Napon napajanja V_{cc} *Radio* podsistema je $3V$
- Ilustracije radi bez narušavanja opštosti pretpostavićemo da senzorski čvor koji šalje ima 11 senzorskih elemenata (senzora) i da se svaki senzorski element senzuje sa *ADC* koji ima rezoluciju 11 bitova, što nam omogućava da koristimo tzv. savršeni kod *SEC-DED* ((15,11) *Hamming* savršen kod proširen dodatnim bitom parnosti). To znači da u izrazu (9.5) imamo da važi $k_r = 11$, $k_c = 11$, $k_{er} = 4$, $k_{ec} = 4$, $h_{op} = 1$ i $v_{op} = 1$.
- Čvor koji šalje kodira polje podataka *Data* koristeći *CRC* šemu kodiranja za *O-RPLL* i *2D SEC-DED/SEC-DED* šemu kodiranja za *M-RPLL* protkol. Format paketa za *O-RPLL* skiciran je na Sl. 10-2, a za *M-RPLL* na Sl. 10-4 i na Sl. 10-5, respektivno.
- Prosečna struja I_{ave} , za vremenski period od T_y u pogledu uticaja *beacon* perioda (vremena buđenja) T_{BP} se izračunava.

- f) Tokom analize koristi se softverski simulacioni model kreiran u *MATLAB*-u koji je opisan u poglavlju 12.4. Kao što se može videti iz Tab. 12-1 većina grešaka na nivou bitova je jednostruke bit greške ili dvostruke bit greške dok su višestruke/*burst* greške prisutne, ali mnogo ređe. Stoga je verovatno da šema kodiranja koja ima sposobnost da ispravi jednostruke i dvostruke greške na nivou bitova može smanjiti značajan deo prisutnih grešaka. Naravno cilj je da ovo i dokažemo korišćenjem simulacionog modela.
- g) Pristupna tačka dekodira podatke korišćenjem algoritma predstavljenog na Sl. 10-6. Napomenimo još jednom da pristupna tačka nije baterijski napajana pa samim tim nije limitirana sa aspekta potrošnje snage.
- h) Osnovna relacija koja određuje performanse propusnosti *O-RPLL* protokola zavisi od verovatnoće P_r da je greška detektovana i da se generiše zahtev za ponovnim slanjem istog paketa (retransmisija), tako da očekivani broj prenosa T_r proračunava kao:

$$\begin{aligned}
 T_r &= (1 - P_r) + 2 \cdot P_r \cdot (1 - P_r) + 3 \cdot P_r^2 \cdot (1 - P_r) + \dots \\
 &\quad + k \cdot P_r^{k-1} \cdot (1 - P_r) + \dots \\
 &= (1 - P_r) \cdot \sum_1^{\infty} k \cdot P_r^{k-1} = \frac{1}{1 - P_r}
 \end{aligned}
 \tag{12.6}$$

S obzirom na sposobnosti *2D SEC-DED/SEC-DED* šeme kodiranja da koriguje greške, vrednost P_r *M-RPLL* protokola je značajno niža u poređenju sa vrednošću dobijenu za *O-RPLL* protokol kod koga se samo ostvaruje detekcija greške. Napomenimo da softverski simulacioni model kreiran u *MATLAB*-u sprovodi analizu i upoređivanje dva protokola za slučaj kada postoji samo jedno paketna retransmisija.

Dobijeni rezultati koji se odnose na prosečnu struju *Radio* podsistema u pogledu uticaja parametra *BER* za pet različite vrednosti parametra *beacon* period T_{BP} od (*30 s, 1 min, 2 min, 3 min i 5 min*) prikazani su na Sl. 12-3.



Sl. 12-3 Prosečna struja *Radio* podsistema u pogledu uticaja parametara BER i T_{BP}

Napomena: Krive označene punom linijom odnose se na M-RPLL, a krive označene isprekidanom linijom odnose se na O-RPLL.

Sledeća zapažanja proističu iz analize rezultata prikazanih na Sl. 12-3:

- a.1) Kako se povećava vrednost *beacon* perioda T_{BP} prosečna vrednost struje I_{ave} se smanjuje što je posledica činjenice da je u tom slučaju senzorski čvor veći deo vremena u spavanju
- a.2) Pošto M-RPLL protokol koristi šemu kodiranja 2D SEC-DED/SEC-DED koja se karakteriše sposobnošću ispravljanja grešaka njen broj retransmisija je znatno manji ili ne postoji i kao posledica toga prosečna struja I_{ave_M-RPLL} sporije raste u poređenju sa prosečnom strujom I_{ave_O-RPLL}
- a.3) Oba protokola troše istu količinu energije u tačkama preseka označenim kao P_1, P_2, P_3, P_4 i P_5 koje odgovaraju parametru *beacon* perioda T_{BP} (30 s, 1 min, 2 min, 3 min i 5 min), respektivno. Prosečne vrednosti struje koje odgovaraju tačkama preseka su date u Tab. 12-3. ($I_{ave-M-RPLL} = I_{ave-O-RPLL}$).

a.4) Za datu vrednost T_{BP} u oblasti sa nižim vrednostima BER -a (levo od bilo koje tačke preseka) potrošnja O - $RPLL$ protokola je manja (sa jedne strane, uzrok tome je manji broj retransmisija usled manje stope pogrešnih bitova, a sa druge strane veća redundantnost M - $RPLL$ protokola dovodi do veće potrošnje energije). Nasuprot tome, u oblasti sa višim vrednostima BER -a (desno od bilo koje tačke preseka) M - $RPLL$ protokol je bolje rešenje (dolazi do izražaja veliki broj retransmisija usled povećane stope pogrešnih bitova). Primera radi za presečnu tačku P_3 imamo da je $I_{ave} = 1.3551 \mu A$ i $BER = 1.64 \cdot 10^{-5}$. Ovo znači da je, za vrednosti $BER < 1.64 \cdot 10^{-5}$, O - $RPLL$ protokol bolji, dok je za $BER > 1.64 \cdot 10^{-5}$ bolji M - $RPLL$ protokol.

Tab. 12-3 Tačke preseka i prosečne vrednosti struje

Tačke preseka	BER	$I_{ave} [\mu A]$	T_{BP}
P_1	$3.94 \cdot 10^{-5}$	2.42	30 sec
P_2	$2.71 \cdot 10^{-5}$	1.71	1 min
P_3	$1.64 \cdot 10^{-5}$	1.36	2 min
P_4	$1.14 \cdot 10^{-5}$	1.24	3 min
P_5	$7.23 \cdot 10^{-6}$	1.14	5 min

Prema svemu što je prethodno navedeno, u prilici smo da izvedemo prve zaključke o predloženom protokolu. Naime, kada bežične senzorske mreže rade u surovim uslovima okruženja (BER vrednosti su $\geq 10^{-4}$) na osnovu prethodne analize opravdano je koristiti M - $RPLL$ protokol umesto O - $RPLL$ protokola, tj. $2D$ SEC - DED šema kodiranja, sa aspekta potrošnje energije, produžava životni vek baterije, a samim tim i senzorskog čvora. Generalno, na ovaj način mi povećavamo procenat PDR (*Packet Delivery Ratio*) za bilo koju fiksnu vrednost BER -a. Tokom iscrpne analize sprovedene na PC korišćenjem softverskog simulacionog modela kreiranog u $MATLAB$ -u primenom algoritma "brutalne" sile (*brute-force*) generisani su rezultati prikazani u Tab.

12-4. Dobijeni rezultati se odnose na *PDR* metriku, za *O-RPLL* i *M-RPLL* protokole, za vrednosti *BER*-a kao parametra u opsegu od 10^{-2} do 10^{-6} i za fiksni vremenski interval $T_{BP} = 2$ min.

Tab. 12-4 Stopa isporučenih paketa za *O-RPLL* i *M-RPLL* protokole

		<i>BER</i>				
		10^{-2}	10^{-3}	10^{-4}	10^{-5}	10^{-6}
T_{BP} 2 min	Ukupan broj poslatih paketa po godini, N_y	262800				
	# paketa sa jednostrukom greškom	262642	213288	43063	4745	434
	# paketa sa dvostrukom/višestrukum greškom	259279	31081	345	4	0
	<i>PDR</i> [%] za <i>O-RPLL</i>	0.06	18.84	83.62	98.20	99.84
	<i>PDR</i> [%] za <i>M-RPLL</i>	0.07	98.75	99.9	99.999	~100

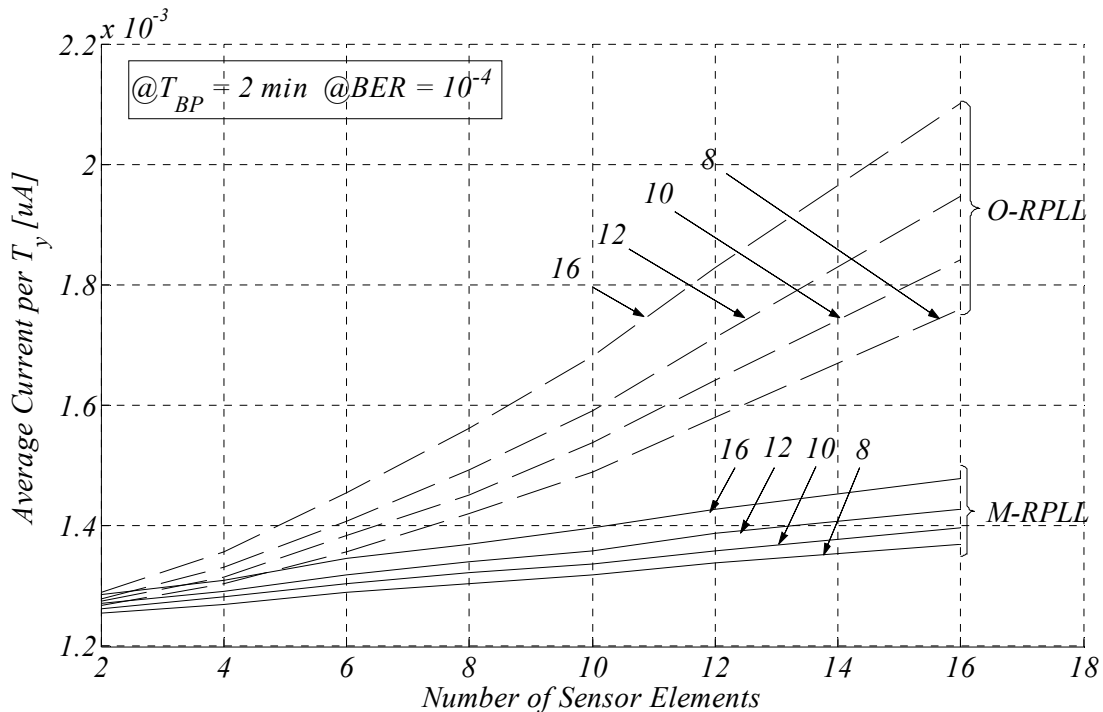
Rezultati pokazuju da se korišćenjem *FEC* tehnike kodiranja umesto *ARQ* tehnike kodiranja u velikoj meri povećavaju performanse *PDR*-a i, u tom kontekstu, da je predložena šema kodiranja *2D SEC-DED* prihvatljivo rešenje za bežične senzorske mreže. Naravno, nikako ne smemo zanemariti činjenicu da ovo podrazumeva prisustvo značajno veće redundanse u broju bitova koji se prenose i da u konačnoj primeni moramo da sagledamo i *BER* parametar.

12.7 Prosečna struja i energetska dobit *Radio* podsistema u pogledu uticaja broja senzorskih elemenata (#*SE*) i rezolucije *AD* konvertora (#*ADC*) kao parameter za fiksni *beacon* period T_{BP} i fiksnu vrednost *BER*

Naša iscrpna analiza, nakon brojnih simulacija i merenja, pokazuje da su za različite vrednosti *beacon* perioda u rasponu $30\text{sec} \leq T_{BP} \leq 5\text{min}$ i *BER* u okviru opsega $10^{-6} \leq BER \leq 10^{-2}$ presečne tačke locirane u oblasti koja je oivičena vrednošću *BER*-a od $10^{-5} < BER < 10^{-4}$ (videti Sl. 12-3). Imajući u vidu ovu činjenicu, sva naša dalja analiza koja se odnosi na potrošnju energije zasnivaće se na najgorem slučaju tj. za fiksnu vrednost $BER = 10^{-4}$. Treba napomenuti da je to vrednost koja odgovara uslovima prenosa u industrijskom okruženju za koje mi i predložimo novu kodnu šemu odnosno nadograđujemo postojeći *Rendezvous* protokol.

U nastavku ćemo izvršiti procenu prosečne struje *Radio* podsistema I_{ave} u zavisnosti od broja senzorskih elemenata (#*SE*) (to znači da će parameter k_r imati vrednosti $k_r = 2/4/6/8/10/12/16$)

(videti izraz (9.5)) sa fiksnim vremenskim *beacon* periodom $T_{BP} = 2$ minuta. Ova vrednost *beacon* perioda je proistekla iz analize dinamike signala parametara koji se prate u okruženju i u skladu sa tim je definisana potrebna brzina odmeravanja. Takođe, tokom ove analize pretpostavićemo da su senzorski elementi (analognog tipa) senzovani od *ADC*-a sa $k_c = 8-10-12-16$ -bitnom rezolucijom (videti izraz (9.5)). Vrednosti za $k_{er}(k_{ec})$ su sračunate u skladu sa uslovima koji su definisani izrazom (9.7), dok je proširena parnost $h_{op} = v_{op} = 1$. Napomenimo da su preduslovi koji su iskazani u napomenama a), b), d), f), g) i h) u odeljku 12.6 važeći i za ovu analizu. Za fiksnu vrednost stope pogrešnih bitova $BER = 10^{-4}$ i vremenski period od jedne godine T_y dobijeni rezultati su prikazani putem grafika na Sl. 12-4.



Sl. 12-4 Prosečna struja *Radio* podsistema u pogledu uticaja broja senzorskih elemenata (#SE) i rezolucije *ADC*-a

Analizom rezultata datih na Sl. 12-4, možemo zaključiti sledeće:

- b.1) Prosečna struja $I_{ave} = f(\#Sensor\ Elements, ADC\ resolution)$ protokola *M-RPLL* u poređenju sa protokolom *O-RPLL* je uvek niža, bez obzira na broj senzorskih elemenata i rezoluciju *ADC*-a. Primera radi, prosečna struja I_{ave} je jednaka $1.291\ \mu A$ za *M-RPLL* i $1.332\ \mu A$ za *O-RPLL*

protokol, za senzorski čvor u kombinaciji sa 4 senzorska elementa sa 12-bit rezolucijom ADC-a, što je vrlo čest slučaj u realizaciji senzorskih čvorova. Pored toga interesantna je realizacija senzorskog čvora sa 8 senzorskih elemenata koji se senzuju 10-bitnom rezolucijom ADC-a i u kom slučaju imamo da je prosečna struja I_{ave} jednaka $1.323 \mu A$ za M-RPLL i $1.452 \mu A$ za O-RPLL protokol. U ekstremnom slučaju za senzorski čvor sa 16 senzorskih elementa sa 16-bit ADC rezolucijom imamo da je prosečna struja I_{ave} za M-RPLL protokol $1.479 \mu A$ i $2.102 \mu A$ za O-RPLL protokol.

b.2) U cilju procene uticaja broja senzorskih elemenata (#SE) i rezolucije ADC-a u pogledu uticaja BER-a, a za beacon period od $T_{BP} = 2$ minuta, iskoristićemo novu metriku koju predlažemo i koju smo nazvali dobit u energiji u oznaci EG (Energy Gain) izraženu u procentima [%]. Dobit u energiji odgovara relativnoj razlici potrošnje energije između O-RPLL i M-RPLL protokola za period od godinu dana T_y .

$$EG = \frac{E_{radio_O-RPLL} - E_{radio_M-RPLL}}{E_{radio_O-RPLL}} * 100 [\%] \quad (12.7)$$

Rezultati koji se odnose na dobit u energiji za tri različite vrednosti stope pogrešnih bitova BER prikazani su u Tab. 12-5.

Tab. 12-5 Dobit u energiji – procena uticaja broja senzorskih elemenata (#SE) i rezolucije ADC-a u pogledu uticaja BER-a, a za beacon period od $T_{BP} = 2$ minuta

BER	ADC	Energetska dobit [%]						
		# Senzorskih elemenata (Senzora)						
		2	4	6	8	10	12	16
10^{-3}	8	9.73	21.76	32.66	39.45	43.79	46.15	47.63
	10	11.18	23.99	35.07	41.19	45.03	46.71	47.52
	12	12.43	26.56	37.23	42.80	45.80	46.90	47.17
	16	14.60	29.79	39.80	44.44	46.41	46.91	46.82
10^{-4}	8	1.01	2.59	5.01	8.16	11.43	15.31	22.25
	10	0.97	2.60	5.78	8.88	13.15	17.22	24.18
	12	0.49	3.10	6.37	10.18	14.54	19.05	26.60
	16	0.23	3.53	7.49	12.37	16.94	21.88	29.64
10^{-5}	8	-0.03	-0.29	-0.67	-0.73	-0.61	-0.70	0.11
	10	-0.29	-0.75	-1.01	-1.21	-0.55	-0.69	0.16
	12	-1.00	-1.04	-1.51	-1.60	-1.13	-1.05	0.17
	16	-1.67	-1.60	-2.34	-1.77	-1.56	-1.22	0.59

Analizirajući rezultate prikazane u Tab. 12-5 možemo da zaključimo sledeće:

- 1) Dobit u energiji je pozitivan u neosenčenim (belim) poljima Tab. 12-5, a negativan u osenčenim poljima. Pozitivna vrednost, izražena u procentima, dobiti u energiji implicira da *M-RPLL* protokol ima bolje performanse, dok je za negativne vrednosti *O-RPLL* protokol superiorniji.
- 2) Pozitivna vrednost dobiti varira u granicama od 0.23% do 47.63%, dok negativna od 0.03% do 2.34%. Napomenimo da je procenat pozitivnih vrednosti dobiti u energiji, u aritmetičkom proseku, značajno veći (reda veličine $\sim \sum_{i=1}^N w_i / N = 22.19\%$) u odnosu na negativnu dobit u energiji, odnosno gubitak energije (reda veličine $\sim \sum_{j=1}^M s_j / M = 1.04\%$).

Napomenimo da $w_i(s_i)$ odgovara vrednostima dobiti u energiji koje se nalaze u belim (osenčenim) poljima (ćelijama), a $N(M)$ broju ćelija u belim (osenčenim) poljima, respektivno. Ova činjenica dodatno naglašava naš prethodni zaključak da je opravdano koristiti *M-RPLL* protokol u ambijentima kada je $BER \geq 10^{-4}$, dok je za $BER < 10^{-4}$ energetski isplativije koristiti *O-RPLL* protokol.

12.8 Softverski režijski troškovi za realizaciju *2D SEC-DED* šeme kodiranja

Režijski troškovi ili overhead se tipično odnosi na količinu potrebnih dodatnih resursa (memorija, procesor, vreme i dr.) koje koriste različiti algoritmi programa za realizaciju određenih aktivnosti. Upravo ova činjenica može da bude odlučujući faktor u projektovanju softvera u odnosu na strukturu, korekciju grešaka i uključivanje pojedinih funkcija biblioteke.

Kao što smo već napomenuli radio arhitektura data na Sl. 11-6 se sastoji iz dva dela, bloka *Radio* podsistema i bloka *Baseband DSP* kao sastavnog dela podsistema za procesiranje. *Baseband DSP* u našem predlogu dizajna je namenjen za softversko generisanje *Data field* polja u okviru frejma formata paketa za *M-RPLL* protokol koristeći *2D SEC-DED* šemu kodiranja (videti Sl. 10-5). Pored ovog rešenja postoji i opcija hardverske realizacije *2D SEC-DED* šeme kodiranja u vidu akceleratora koji bi mogao da bude ili digitalna periferija mikrokontrolera ili sastavni deo primopredajnog bloka. O ovoj temi više u dodatku disertacije, a sada da se osvrnemo detaljnije na prvi predlog. Naime potrošnja energije za obavljanje aktivnosti softverskog generisanja koda u toku vremenskog *beacon* perioda T_{BP} , koju ćemo označiti sa E_{BDSP} , je definisana parametrima $E_{BDSP} = V_{cc} \cdot I_{BDSP} \cdot T_{BDSP}$ datim u jednačini (11.21). Napomenimo da vrednost energije E_{BDSP} odgovara energetskim troškovima režije *M-RPLL* protokola u odnosu na *O-RPLL* protokol. U verziji *O-RPLL* protokola, koji koristi *CRC* šemu kodiranja, postoji podrška na čipu za automatsko proračunavanje *CRC* nad *Data field* poljem i na predajnoj i na prijemnoj strani.

Da bismo odredili vrednost E_{BDSP} pretpostavićemo sledeće parametre radnog okruženja:

1. Mikrokontroler obavlja svoje aktivnosti na sistemskoj frekvenciji $f_{sys} = 16MHz$, naponu napajanja $V_{cc} = 3V$ i u aktivnom režimu rada radi sa prosečnom strujom od $103 \mu A / MHz$.
2. Pretpostavićemo da su podaci koji se dobijaju sa senzora, u ovoj analizi 11 senzorskih elemenata, već smešteni u memorijski bafer i organizovani u vidu strukture polja bitova veličine 11×16 korišćenjem desnog poravnjanja podataka na nivou reči od 16 bitova.

U našem predlogu dizajna, koristeći se tehnikom softverskog generisanja koda, vreme neophodno za aktivnost generisanja matrice koja odgovara *2D SEC-DED* šemi kodiranja (videti Sl. 10-5) je jednako $T_{p-2D} = 491.5625 \mu s$. Vremenski interval T_{p-2D} se proračunava korišćenjem profajlera alata *Code Composer Studio* kao sastavnog dela softverskog projektnog okruženja (*IDE*). Energija koju mikrokontroler troši za obavljanje ove aktivnosti je jednaka $E_{BDSP} = 2.431 \mu J$ i u poređenju sa *O-RPLL* protokolom je približno deset puta niža u odnosu na energiju koja je potrebna za pojedinačnu retransmisiju paketa i koja je reda veličine $28.07 \mu J$.

13 EVALUACIJA PERFORMANSI BEŽIČNOG PRENOSA PODATAKA

Ključni parameter tokom aktivnosti komunikacije, koji je od primarnog interesa prilikom projektovanja bežične senzorske mreže je postizanje pouzdane veze između senzorskih čvorova, onog koji šalje i onog koji prima podatke, uz dodatni uslov minimizacije potrošnje energije. Uopšteno govoreći, pouzdani prenos podataka se može postići ili povećanjem izlazne predajne snage (P_{out}) *radio* podsistema, što za posledicu ima veću potrošnju energije, ili dodavanjem *FEC* tehnike kodiranja polju podataka, što za posledicu ima veću redundantnost, a samim tim i veću potrošnju energije. Naravno u ovakvom odnosu zavisnosti parametara neminovnost je postizanje kompromisa.

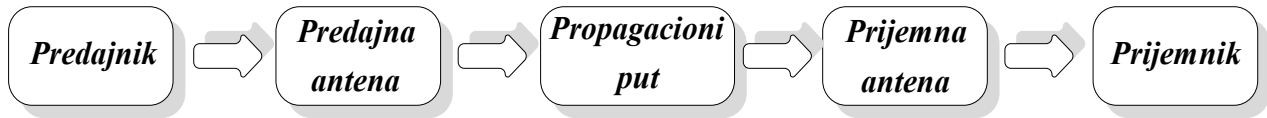
Dakle, ako zadržimo vrednost parametra P_{out} fiksnom, uz korišćenje predložene tehnike kodiranja *SEC-DED/2D SEC-DED* umesto na primer *CRC* kodiranja, koje je najčešće u praktičnoj primeni, možemo ostvariti sledeće:

- 1) smanjiti stopu pogrešno prenetih paketa, *PER*, i/ili
- 2) povećati domet tj. rastojanje između čvora koji šalje i prijemnog čvora

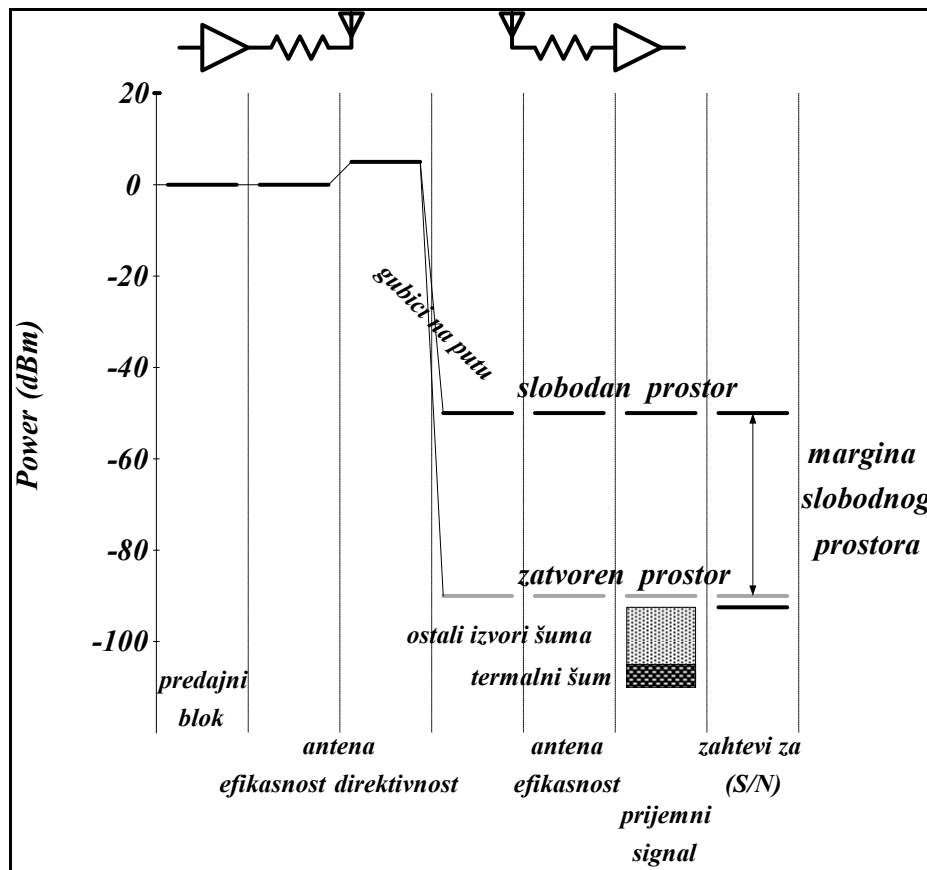
Međutim, kao što je već napomenuto, primena *FEC* tehnike zahteva dodatnu obradu od strane mikrokontrolera i time dodatnu energiju na strani predajnika i prijemnika. U našem rešenju, implementacija *FEC* algoritma u okviru mikrokontrolera čvora koji šalje i čvora koji prima (pristupna tačka *AP*) softverskim pristupom, zahteva zanemarljivo povećanje P_{out} zato što energija koja se troši od strane mikrokontrolera, za generisanje *FEC* šeme kodiranja je za red veličine niža u odnosu na energiju koja je potrebna za ponovno slanje jednog *CRC* kodiranog paketa. Imajući u vidu

da prijemni čvor nije baterijski napajan, a samim tim ni energetski ograničen, njegova potrošnja energije za obradu *FEC* algoritma nije predmet analize u ovoj tezi.

Analiza podrazumeva procenu snage koja će biti dostupna prijemnom čvoru na nekom rastojanju d . U tu svrhu možemo koristiti različite metode za predviđanje snage na prijemnoj strani antene za dati opseg.



a)



b)

SI. 13-1 Bežični prenos podataka: a) Prikaz na blok nivou parametara koji utiču na kvalitet signala; b)

Primer procene budžeta linka (*link budget*) za radio link sa izlaznom snagom od 0dBm

Metode se baziraju na modelima prenosa i ograničenjima koja se unapred usvajaju za dati tip. Najčešće korišćena, ali i najjednostavnija, je *Frii's* jednačina ili jednačina slobodnog prostora. Ova

jednačina se koristi za izračunavanje snage na prijemnoj strani anteni za date parametre na sledeći način [84]:

$$P_r = P_{out} \cdot G_t \cdot G_r \cdot \left(\frac{\lambda}{4 \cdot \pi \cdot d} \right)^\alpha \quad (13.1)$$

gde je P_{out} predajna snaga, $G_t(G_r)$ pojačanje predajne (prijemne) antene, d je rastojanje (domet prenosa), λ je talasna dužina i α je eksponent gubitaka na putu ($\alpha = 2$ za propagaciju u slobodnom prostoru).

Sa aspekta dometa prenosa *Frii's* jednačina može da se napiše kao:

$$20 \log d = G_r(dB) + G_t(dB) + P_{out}(dBm) - P_r(dBm) + 10 \cdot \alpha \log\left(\frac{\lambda}{4 \cdot \pi}\right) \quad (13.2)$$

Iz navedene jednačine proizilazi da kako se rastojanje povećava tako se prijemna snaga smanjuje, što znači da će primljeni paketi biti podložniji greškama kako se rastojanje povećava, jer se na taj način narušava zahtevani odnos signal šuma.

Vrednost α zavisi od propagacionog okruženja: za propagacije koje približno prate model slobodnog prostora ili model dva zraka α je obično sa vrednošću 2 ili 4, respektivno. Vrednost α za složenije sredine prostiranja može da se dobije pomoću minimalne srednje kvadratne greške kojom se postiže približavanje empirijskim merenjima. Alternativno vrednost α može da se dobije i iz modela koji su bazirani na empirijskim merenjima koji uzimaju u obzir frekvenciju i visinu antene. Sumarni prikaz vrednosti za različita unutrašnja (zatvorena) i spoljašnja (otvorena) okruženja i visine antena na 900 MHz i 1.9 GHz je dat u Tab.13-1.

Tab. 13-1 Tipične vrednosti eksponenta gubitaka na putu

Okruženje	opseg α vrednosti
Gradska (urbana) makroćelija	3.7 – 6.5
Gradska (urbana) mikroćelija	2.7 – 3.5
Poslovna zgrada (isti sprat)	1.6 – 3.5
Poslovna zgrada (više spratova)	2 – 6
Prodavnica	1.8 – 2.2
Fabrika	1.6 – 3.3
Kuća	3

Vrednosti eksponenta gubitaka na putu na višim frekvencijama imaju tendenciju da budu viši, dok sa antenama koje su višlje imaju tendenciju da budu niži. Napomenimo da širok spektar vrednosti ekponenta gubitaka na putu dobijenih empirijskim putem za propagaciju u unutrašnjem (zatvorenom) prosotoru uslovljen je slabljenjima usled prisustva podova, objekata u prostoru i postojanju pregrada.

Sa druge strane minimalna predajna snaga P_{out} neophodna za postizanje minimalnog odnosa signal šum (ispod koje prijemnik ne može da radi pouzdano) za slučaj kada je rastojanje između predajnika i prijemnika d u slobodnom prostoru i bez interferencije može se iskazati sledećom relacijom:

$$P_{out} = \eta \cdot \frac{E_b}{N_0} \cdot m \cdot k \cdot T \cdot B \left(\frac{4 \cdot \pi \cdot d}{\lambda} \right)^2 \quad (13.3)$$

gde je λ talasna dužina predajnog signala ($\lambda = c/f$, f je frekvencija nosioca i c brzina elektromagnetnog talasa), E_b je zahtevana minimalna energija po bitu na strani prijemnika, N_0 je spektralna gustina snage šuma, B je propusni opseg, η je odnos informacione brzine i propusnog opsega (tj. spektralna gustina), m je konstanta proporcionalnosti šuma (obično $m \geq 1$), k je Boltzmann-ova konstanta i T je apsolutna temperatura u Kelvin.

Prema jednačini (13.3) izlazna snaga predajnika P_{out} je proporcionalna kvadratu rastojanju d^2 i inverzno proporcionalna kvadratu talasne dužine λ^2 . Ovo znači da predaja na višim frekvencijama zahteva veću izlaznu predajnu snagu P_{out} .

13.1 Gubici na putu i efekat senke

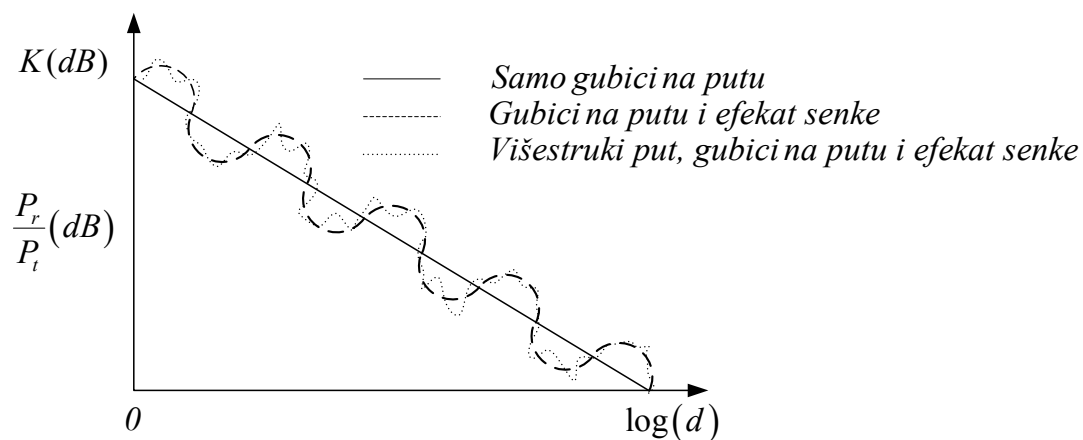
Bežični radio kanal kao medijum za prenos predstavlja ozbiljan izazov ostvarivanju pouzdane komunikacije velikih brzina. Razlog tome nije samo njegova osetljivost na šum, podložnost interferenciji i pojavama drugih prepreka u kanalu, nego što se te prepreke menjaju tokom vremena na nepredvidljiv način zbog kretanja korisnika. Cilj ovog dela teksta je da se izvrši karakterizacija varijacija u snazi signala na prijemu u funkciji rastojanja usled gubitaka na putu i efekata senke. Gubitak na putu je uzrokovan disipacijom snage koju zrači predajnik kao i efektima propagacije kanala.

Model gubitaka na putu generalno pretpostavlja da su gubici na putu isti na datoj udaljenosti između predajnika-prijemnika tj. ne uključuju efekat senke. Efekat senke je izazvan preprekama

između predajnika i prijemnika koji apsorbiraju snagu. Kada prepreka apsorbira svu snagu kažemo da je signal blokiran. Varijacije usled gubitaka na putu javljaju se na veoma velikim rastojanjima (100 do 1000 metara) dok se varijacije usled efekata senke javljaju na rastojanjima proporcionalnim dužini objekta koji ometa signal (10 do 100 metar na otvorenom ili manje u zatvorenom prostoru). Pošto se varijacije zbog gubitaka na putu ili efekata senke javljaju na relativno velikim rastojanjima, ove varijacije se ponekad označavaju i kao efekti propagacije na velikim rastojanjima ili lokalno srednje slabljenje.

Varijacije usled *multipath*-a (višestrukog puta) se dešavaju na veoma kratkim rastojanjima i posledica su talasne dužine signala, tako da se varijacije ponekad nazivaju efekti propagacije na malim (kratkim) rastojanjima ili *multipath fading* (bleđenje (nestajanje) signala usled višestrukog puta).

Sl. 13-2 prikazuje odnos primljenje-predajne snage u *dB* u odnosu na rastojanje tj. distancu koja je prikazana u log razmeri za kombinovane efekte gubitaka na putu, efekata senke i višestrukog puta.



Sl. 13-2 Gubici na putu, efekat senke i višestruki put u funkciji rastojanja

Posle kratkog uvoda i opisa našeg modela signala, prezentovaćemo najjednostavniji model propagacije signala poznat kao gubici na putu kroz slobodni prostor. Signal koji propagira između dve tačke bez efekata slabljenja (atenuacije) ili refleksije odslikava suštinu zakona propagacije kroz slobodni prostor.

13.2 Model propagacije signala u zatvorenom prostoru

Okruženja zatvorenog prostora se međusobno razlikuju u pogledu materijala koji se koriste za zidove i podove, rasporedu soba, hodnika, prozora i otvorenih prostora, položaju i materijalu objekata

koji ometaju signal, veličini svake sobe i broju spratova. Svi navedeni faktori imaju značajan uticaj na gubitke na putu u zatvorenom prostoru. Prema tome, teško je naći opšti model koji bi se primenio sa velikom tačnošću u cilju određivanja gubitka na putu u specifičnim uslovima zatvorenog prostora.

Modeli gubitaka na putu u zatvorenom prostoru moraju precizno da obuhvate efekte slabljenja na ostom spratu, zbog pregradnih zidova, kao i između spratova. Merenja sprovedena na (za) širokom opsegu karakteristika zgrade i frekvencija signala ukazuju na to da je slabljenje po spratu najveće za prvi sprat kroz koji se prođe i smanjuje se sa svakim narednim prolaskom kroz sprat. Konkretno, merenja u [85-88] ukazuju na to da je na frekvenciji od 900MHz , kada su predajnik i prijemnik odvojeni jednim spratom, slabljenje u rasponu od $10-20\text{dB}$, dok su naredna slabljenja od $6-10\text{dB}$ po spratu za sledeća tri sprata, a zatim nekoliko dB po spratu za više od četiri spratova. Na višim frekvencijama gubici usled slabljenja po spratu su tipično veći. Smatra se da se slabljenja po spratu smanjuju kako se broj spratova povećava zbog efekata rasipanja (*scattering*) od viših delova zgrade i pojava refleksija od susednih zgrada. Materijali na preprekama i dielektrične osobine prepreka u velikoj meri variraju, a samim tim i dovode do navedenih gubitaka. Merenja gubitaka usled postojanja prepreka na različitim frekvencijama za različite tipove prepreka datih u Tab. 13-2 ukazuju na nekoliko primera gubitaka na preprekama merenih na $900-1300\text{MHz}$. Gubici usled postojanja prepreka koji se dobijaju od različitih istraživača za isti tip prepreke na istoj frekvenciji često variraju.

Tab. 13-2 Tipične vrednosti gubitaka na preprekama od različitim materijalima

Tip prepreke	Gubici na preprekama [dB]
prepreka od tkanine	1.4
dvostruki zidovi od gipsanih ploča	3.4
Izolacija od folije	3.9
Betonski zid	13
Aluminijumom pokriveni zidovi	20.4
Cela metalna	26

Eksperimentalni podaci za gubitke kroz spratove i usled prepreka mogu da se ugrade u jednostavan model gubitaka na putu kao:

$$P_r = P_t + K - 10 \cdot \alpha \cdot \log_{10} \left[\frac{d}{d_0} \right] - \sum_{i=1}^{N_f} FAF_i - \sum_{i=1}^{N_p} PAF_i \quad (13.4)$$

gde se:

α - dobija iz gubitaka na putu za merenja sprovedena na istom spratu (na primer iz Tab. 13-1), FAF_i predstavlja faktor slabljenja po spratnosti za i -ti sprat kroz koji prolazi signal, a PAF_i predstavlja faktor slabljenja usled prepreka povezan sa i -tom preprekom kroz koju je prošao signal. Broj spratova i prepreka kroz koje prolazi signal su N_f i N_p , respektivno.

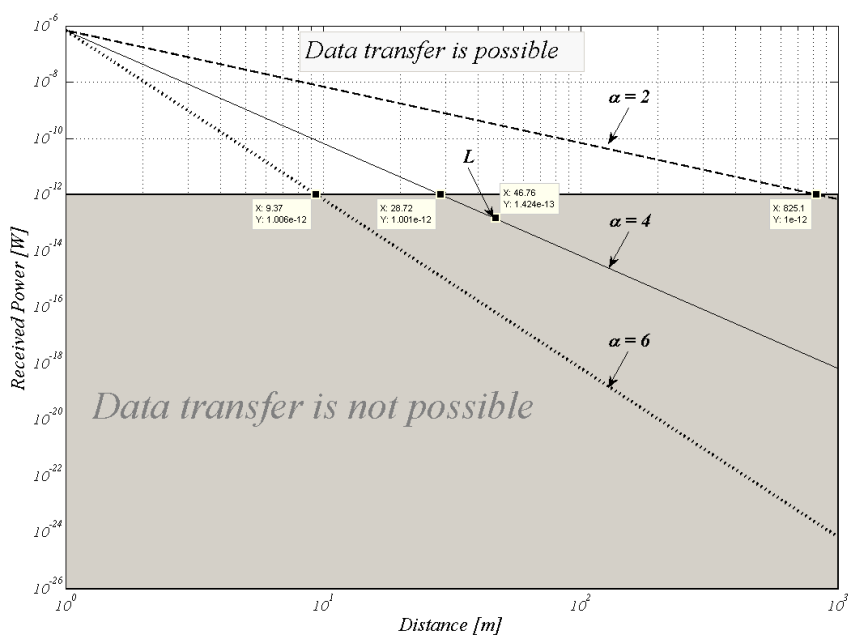
Još jedan važan faktor za sisteme u zatvorenom prostoru za slučaj kada je predajnik lociran izvan zgrade predstavlja gubitak usled penetracije (prodiranja) kroz zgradu. Sprovedena merenja ukazuju da gubitak usled penetracije kroz zgradu funkcija frekvencije, visine zgrade i građevinskog materijala od kojeg je zgrada sagrađena. Gubitak usled penetracije zgrade u prizemlju se obično kreće od $8-20\text{ dB}$ u opsegu od 900 MHz do 2 GHz . Gubitak usled penetracije se blago smanjuje sa povećanjem frekvencije, a takođe se smanjuje za oko 1.4 dB po spratu na spratovima iznad prizemlja. Ovo poboljšanje je tipično posledica smanjenja nereda na višim spratovima i veće verovatnoće da postoji optička vidljivost. Tip i broj prozora u zgradi takođe ima značajan uticaj na gubitke usled penetracije. Naime, merenja koja su sprovedena sa predajnikom iza prozora imaju oko 6 dB manje gubitaka usled penetracije u odnosu na merenja iza spoljašnjih zidova. Osim toga, zidne ploče od stakla imaju slabljenje od oko 6 dB , dok ploče od stakla oivičene olovom imaju slabljenje između 3 i 30 dB .

14 MATEMATIČKI MODEL ZA EVALUACIJU DOMETA PREDAJE

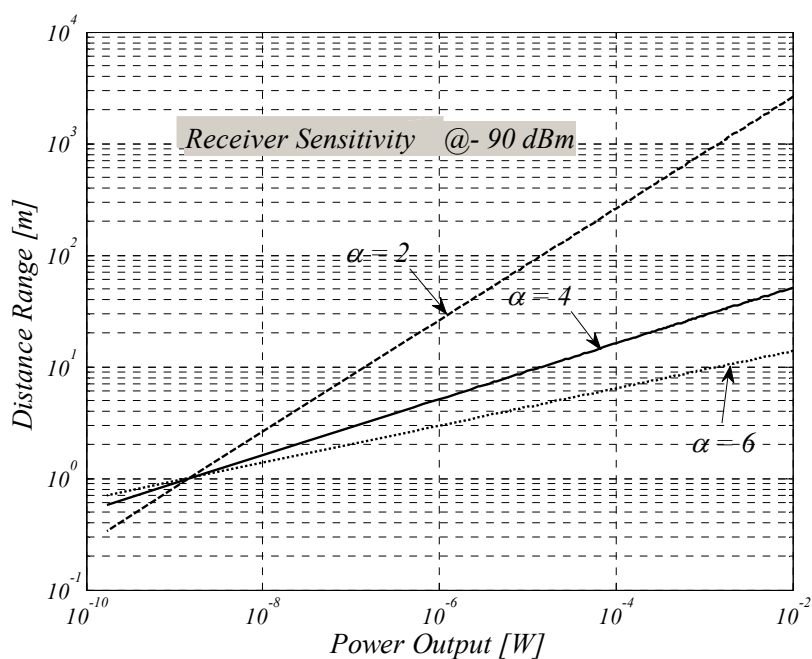
U ovom delu disertacije cilj nam je da odredimo dobit korišćenja *2D SEC-DED* šeme kodiranja u pogledu rastojanja, u poređenju sa *CRC* kodiranjem, odnosno da odredimo koliko povećanje rastojanja smo u stanju da ostvarimo. U eksperimentu ćemo koristiti sledeće vrednosti parametara i postulate:

- 1) Između predajnika i prijemnika postoji bežična veza tipa *tačka-ka-tački*
- 2) Za prenos podataka koristi se *Rendezvous* protokol
- 3) *M-RPLL* i *O-RPLL* protokol koriste *GFSK* modulacionu šemu na frekvenciji nosioca $f_c = 915\text{MHz}$
- 4) Bitska brzina je $R = 250\text{kbps}$
- 5) Predajna izlazna snaga je $P_{out} = 0\text{dBm}(1\text{mW})$
- 6) Predajnik i prijemnik koriste idealne izotropne antene tj. pojačanje antena je $G_t = G_r = 1$
- 7) *BER* se koristi kao indikator kvaliteta kanala sa vrednošću koja varira od 10^{-3} do 10^{-6}
- 8) Senzitivnost prijemnika je $P_{RS} = -90\text{dBm}$ (tipično za *CC110L* primopredajnik [50]). Za navedenu vrednost senzitivnosti ekperiment se ponavlja za različita rastojanja između predajnika i prijemnika koja su u opsegu od 0m do 1000m .
- 9) U bežičnim komunikacijama gubici na putu se karakterišu eksponentom gubitaka na putu čija je vrednost u opsegu od 2 do 6 (vrednost $\alpha = 2$ karakteriše propagaciju u slobodnom

prostoru, $\alpha = 4$ okruženje sa relativno izraženim gubicima, dok vrednosti u opsegu od $\alpha = 4 - 6$ karakterišu sredine kao što su zgrade, stadioni i druge zatvorene sredine).



a)



b)

Sl. 14-1 Snaga na prijemnoj strani i domet signala: a) Snaga na prijemnoj strani u pogledu rastojanja sa eksponentom gubitaka na putu kao parametrom; i b) Domet signala u pogledu predajne izlazne snage sa eksponentom gubitaka na putu kao parametrom

Na Sl. 14-1 a) prikazana je detektovana snaga na prijemnoj strani antene P_r u zavisnosti od rastojanja (dometa predaje) d i eksponenta gubitaka na putu α kao parametara, za predajnu snagu od $P_t = 1mW$ i frekvenciju nosioca $f_c = 915MHz$. Kada je senzitivnost prijmnika $P_{RS} = -90dBm$ (odgovara snazi na strani prijmnika od $P_r = 1pW = 10^{-12}W$) onda je za vrednosti eksponenta $\alpha = 2/4/6$ maksimalna moguća vrednost rastojanja između predajnika i prijmnika jednaka $824.9m / 28.72m / 9.37m$, respektivno.

Osenčeni deo na Sl. 14-1 a) ukazuje na slučaj kada bežični prenos podataka nije pouzdan (nivo snage primljenog signala je ispod praga, tj. $< 1pW$).

Tab. 14-1 Tipične vrednosti dometa predaje za različite vrednosti izlazne snage i eksponenta gubitaka na putu

	Eksponent gubitaka na putu		
	2	4	6
Izlazna snaga [dBm]	Rastojanje [m]		
10	2609	51	14
0	864	29	10
-10	258	16	6

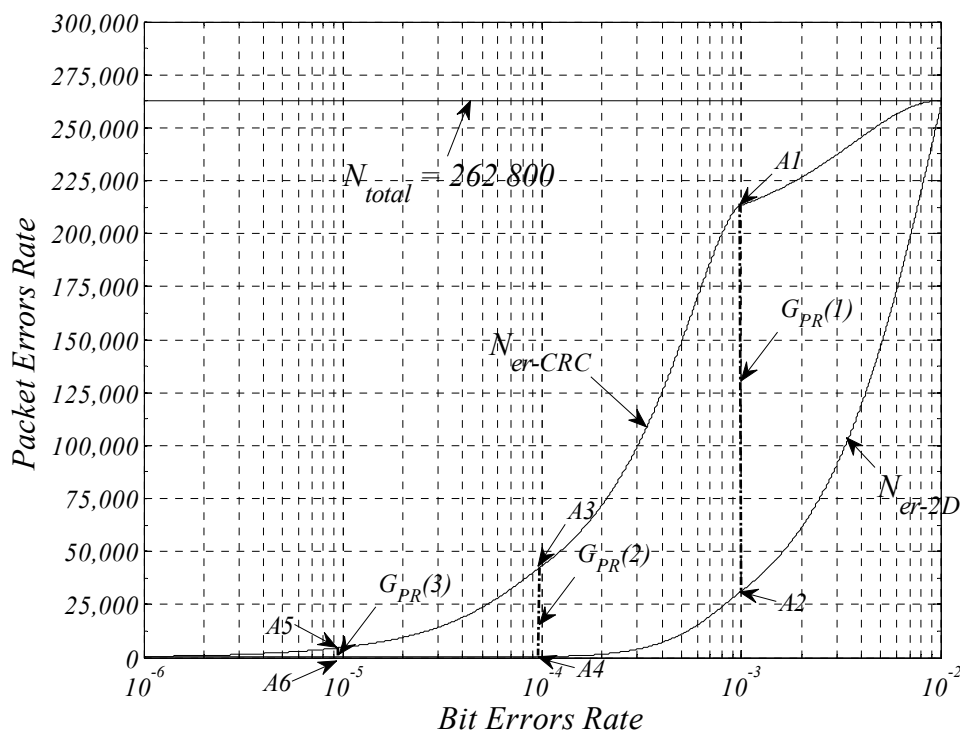
Na Sl. 14-1 b) prikazane su dometi predaje u zavisnosti od snage koju je izračila predajna antena P_{out} sa eksponentom gubitaka na putu kao parametrom. Napomenimo da korisnik senzorskog čvora ima mogućnost da softverskim putem postavi nivo izlazne snage u opsegu od $-67.7dBm$ ($\sim 170pW$) do $10dBm$ ($\sim 10mW$) [49]. Opšti zaključak, što se vidi i sa Sl. 14-1 b) je da za istu vrednost snage P_{out} dužina dometa drastično opada kako eksponent gubitaka raste, odnosno kako se uslovi okruženja prenosa pogoršavaju, što je i opšte poznata činjenica. Neke tipične vrednosti su date u Tab. 14-1.

14.1 Metrike evaluacije: faktor oporavka pogrešnih paketa i faktor poboljšanja *BER*-a

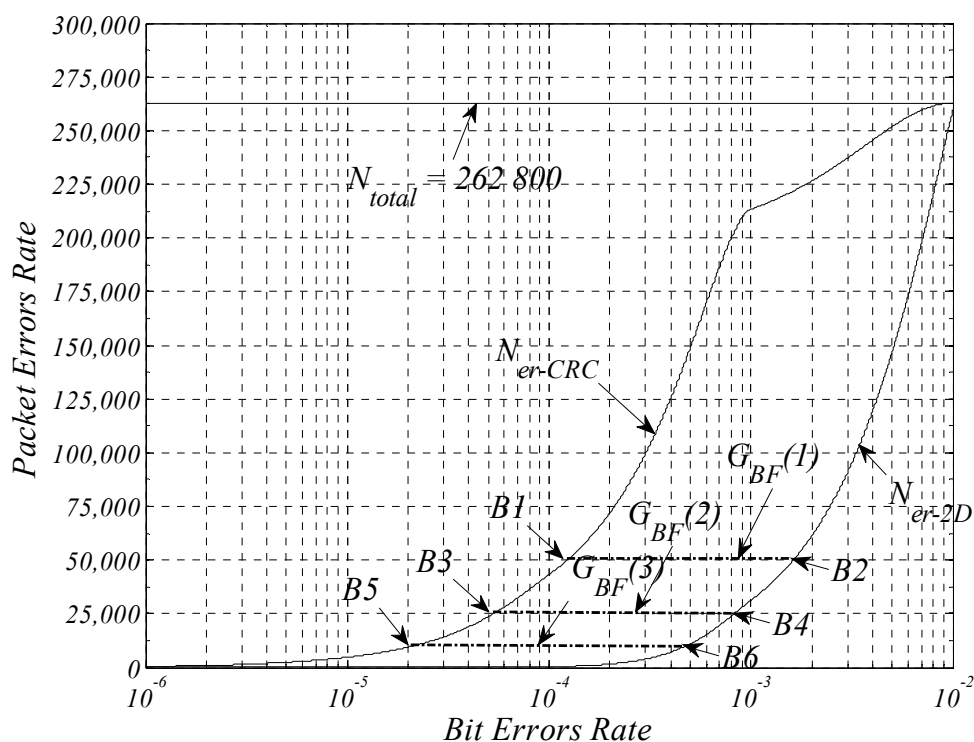
Ovaj deo disertacije ima za cilj da utvrdi prednosti involviranja *2D SEC-DED* šeme kodiranja u odnosu na *CRC* šemu kodiranja, koje će nam poslužiti da izvedemo konačan zaključak po pitanju dobiti u pogledu dometa. Da bismo sproveli ovu analizu korišćićemo rezultate analiza datih u prethodnim poglavljima koje ćemo suziti sledećim pretpostavkama:

- 1) Ukupan broj poslatih paketa, za vremenski *beacon* period $T_{BP} = 2$ minuta, u toku perioda od jedne godine je $N_{total} = 262800$
- 2) Suma jednostrukih, dvostrukih/višestrukih grešaka na nivou bitova u okviru paketa specificirana je *MATLAB randerr* funkcijom korišćenjem modela *Binomial* raspodele verovatnoće [82-83]

Na Sl. 14-2 ilustrovan je broj prenešenih paketa sa greškom za slučaj kada je korišćenja *CRC* šema kodiranja, N_{er_CRC} , i za slučaj kada je korišćenja *2D SEC-DED* šema kodiranja, N_{er_2D} (nakon korišćenja procedure ispravljanja grešaka) u zavisnosti od *BER* za fiksni broj paketa od $N_{total} = 262800$.



a)



b)

SI. 14-2 Broj pogrešnih paketa u prenosu sa CRC i 2D SEC-DED šemom kodiranja: a) metrika evaluacije $G_{PR}(i)$; b) metrika evaluacije $G_{BF}(i)$

Napomena:

- 1) Parovi tačaka ($A_1 = 213290, A_2 = 31082$), ($A_3 = 43068, A_4 = 345$) and ($A_5 = 4745, A_6 = 4$) se koriste za proračunavanje vrednosti $G_{PR}(i)$, $i = 1, 2, 3$, respektivno;
- 2) Parovi tačaka ($B_1 = 0.001626931, B_2 = 0.000123769$), ($B_3 = 0.000831034, B_4 = 0.000053299$) i ($B_5 = 0.000478711, B_6 = 0.000021304$) se koriste za proračunavanje $G_{BF}(i)$, $i = 1, 2, 3$, respektivno.

Neka je ukupan broj prenešenih paketa za vremenski period od jedne godine N_{total} . U realnom scenariju broj pogrešno primljenih paketa u oznaci N_{er} će biti $N_{er} < N_{total}$. Za CRC šemu kodiranja broj paketa koji se prenesu bez greške će biti $N_{CRC} = N_{total} - N_{er-CRC}$, dok je za 2D SEC-DED šemu kodiranja jasno da će ova vrednost biti veća jer će neki paketi od navedenog broja N_{CRC} biti uspešno oporavljeni, tako da će broj paketa primljenih sa greškom u ovom slučaju iznositi $N_{er-2D} < N_{er-CRC}$. Sa druge strane, ovo implicira da će za 2D SEC-DED šemu kodiranja broj ispravno primljenih paketa, obuhvata pakete bez grešaka i uspešno oporavljene pakete, biti

$N_{2D} = N_{total} - N_{er_2D}$. Razlika, u oznaci N_{diff} , predstavlja broj uspešno oporavljenih paketa i iskazana je kao $N_{diff} = N_{2D} - N_{er_CRC}$.

Imajući ovo u vidu sada smo u prilici da uvedemo novu metriku koju smo nazvali faktor oporavljanja paketa, G_{PR} , definisan kao odnos razlike između primljenih paketa sa greškom i neuspešno oporavljenih paketa u odnosu na ukupan broj paketa poslat u toku jedne godine i može se izraziti kao:

$$G_{PR} = \frac{N_{2D} - N_{er_CRC}}{N_{total}} \cdot 100\% \quad (14.1)$$

Za fiksni $BER = 10^{-5} / 10^{-4} / 10^{-3}$ (videti Sl. 14-2 a)) dobijamo $G_{PR} = 1.84\% / 16.32\% / 62.20\%$ respektivno, što ukazuje da je $2D SEC-DED$ šema kodiranja superiornija za veće vrednosti BER -a, tj. kada je bežična senzorska mreža aktivna u sredini koja je izrazito šumna, jer u tom slučaju CRC šema kodiranja detektuje veliki broj paketa sa greškom.

Za fiksni broj paketa sa greškom možemo uvesti novu metriku koju smo nazvali faktor dobiti BER -a, u oznaci G_{BF} , (videti Sl. 14-2 b)) kao odnos između vrednosti BER -a za $2D SEC-DED$ i za CRC šemu kodiranja kao:

$$G_{BF} |_{N_e=const} = \frac{BER_{2D}}{BER_{CRC}} \cdot 100\% \quad (14.2)$$

Kao metrika evaluacije (procene) G_{BF} ukazuje na to koliko je $2D SEC-DED$ šema kodiranja rezistivnija na BER u odnosu na CRC šemu kodiranja.

Tako na primer, kada je broj neoporavljenih paketa $N_{er} = 50000 / 25000 / 10000$ (videti Sl. 14-2 b)) dobijamo $G_{BF} |_{N_e=const} = 13.14\% / 15.59\% / 22.47\%$, respektivno.

Ova činjenica ukazuje da uključivanjem predložene šeme kodiranja sistem postaje rezistivniji na greške bitova što indirektno znači da su potrebe za ponovnim slanjem redukovane.

Dakle sve prethodne analize ukazuju da primenom $2D SEC-DED/SEC-DED$ šeme kodiranja možemo:

- ✓ smanjiti verovatnoću pogrešnih bitova (p_b) za fiksnu vrednost predajne izlazne snage P_{out}
- ✓ smanjiti predajnu izlaznu snagu P_{out} za isti domet

- ✓ zadržati predajnu izlaznu snagu P_{out} na početnu vrednost, pre primene CRC kodne šeme $2D SEC-DED/SEC-DED$ kodnom šemom, kao i snagu osetljivosti prijemnika P_{RS} , a kao posledica dobiti primenom predložene šeme kodiranja povećati komunikaciono rastojanje (domet) između predajnika i prijemnika.

Konačno, sve ovo ćemo potvrditi u narednom delu disertacije i odrediti konkretne vrednosti dobiti rastojanja.

14.2 MATLAB model za procenu dometa predaje

Iako ARQ tehnika prenosa zahteva jednostavniji dekodier i značajno manji broj redundantnih bitova, pojava i samo jednog pogrešnog bita u paketu koji je kodiran CRC kodnom šemom nužno zahteva ponovno slanje paketa (retransmisiju) što direktno utiče na energetska efikasnost senzorskog čvora. Ako ostanemo pri primeni ARQ tehnike prenosa neophodno rešenje je da povećamo izlaznu snagu predajnika P_{out} kako bismo smanjili ponovno slanje paketa. Sa druge strane, korišćenjem FEC tehnike prenosa, zbog većeg broja redundantnih bitova, u startu nam je potrebna veća energija za njihovo slanje.

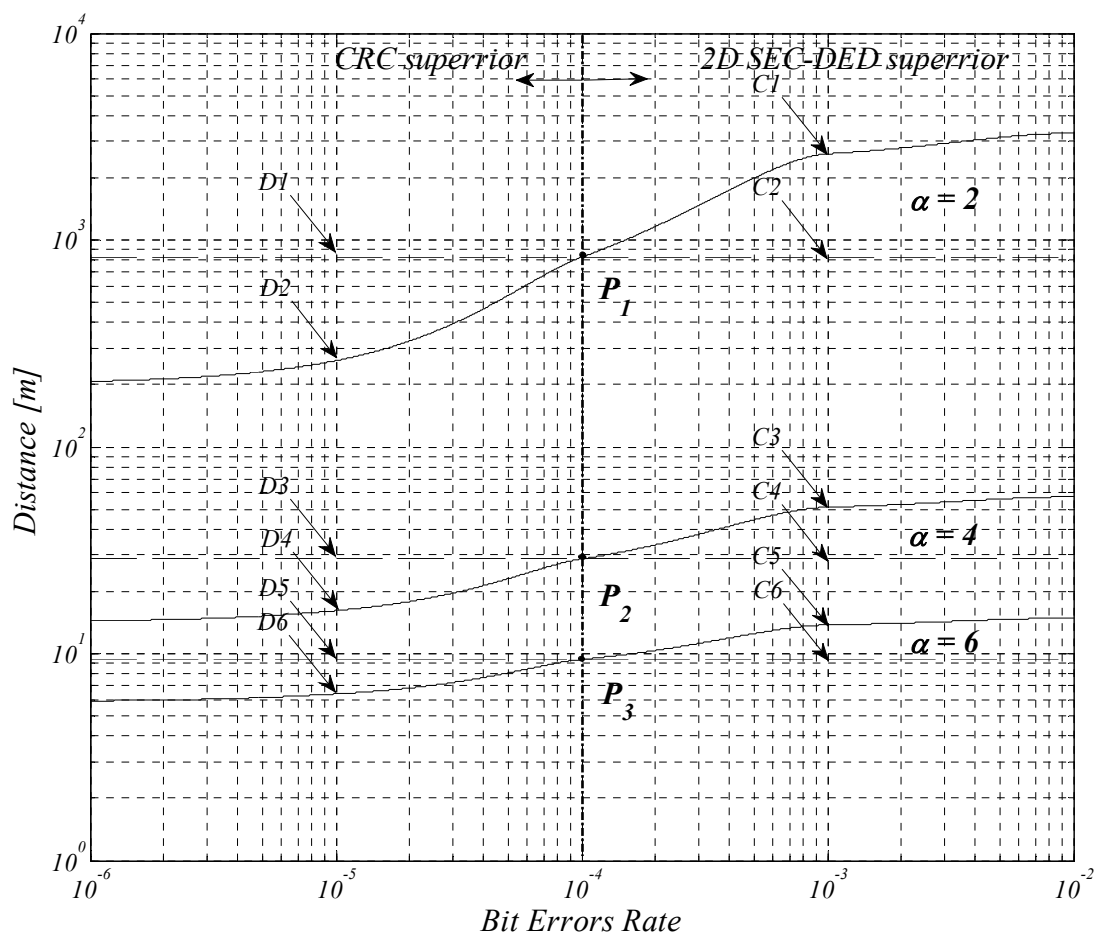
Ali kada je okruženje u kome se ostvaruje prenos takvo da postoji potreba za retransmisijom odgovor koji je od interesa za nas je pod kojim uslovima tj. za koje vrednosti BER -a je jedna od dve navedene tehnike superiornija.

Tokom ovog istraživanja pretpostavljamo da su važeći uslovi podešavanja i parametrizacije modela od 1 do 9 sa početka ovog poglavlja. Zatim, u svrhu upoređivanja kao referentni nivo rastojanja koristićemo gornju granicu maksimalno mogućeg dometa tokom koga je ostvaren uspešan prenos paketa, dostupan za datu snagu predaje i aperture antene.

Da bismo sproveli analizu uvešćemo novu metriku koju ćemo nazvati dobit u rastojanju u oznaci G_D definisanu kao:

$$G_D = \frac{D_{2D} - D_{CRC}}{D_{CRC}} \cdot 100[\%] \quad (14.3)$$

gde je $D_{2D}(D_{CRC})$ postignuta rastojanja kada se koristi $2D SEC-DED$ (CRC) šema kodiranja, respektivno.



Sl. 14-3 Opseg predaje u funkciji BER za tri različite vrednosti eksponenta gubitaka na putu kao parametara

Napomena:

- 1) Parovi tačaka ($C_1 = 825.10, C_2 = 2606.00$), ($C_3 = 28.72, C_4 = 51.10$) i ($C_5 = 9.40, C_6 = 13.80$) se koriste za proračunavanje $G_D(i)$, $i = 1, \dots, 3$ za $BER = 10^{-3}$, respektivno ;
- 2) Parovi tačaka ($D_1 = 825.10, D_2 = 261.30$), ($D_3 = 28.72, D_4 = 16.20$) i ($D_5 = 9.40, D_6 = 6.40$) se koriste za proračunavanje $G_D(i)$, $i = 1, \dots, 3$ za $BER = 10^{-5}$, respektivno.

Analizom rezultata datih na Sl. 14-3 možemo da izvedemo sledeće zaključke:

- 1) Tačke preseka P_1 , P_2 i P_3 odgovaraju slučajevima kada je potrošnja energije za obe šeme kodiranja jednaka, pri istoj vrednosti $BER = 10^{-4}$, pri postizanju istog rastojanja između predajnog i prijemnog čvora, a za pakete koji su definisani u odeljku 10.6, 10.8 i 10.10. Ovo ukazuje da je poželjnije koristiti CRC šemu kodiranja za vrednosti $BER < 10^{-4}$. Nasuprot

tome za vrednosti $10^{-4} < BER < 10^{-2}$ (tipično za *harsh* okruženje) primena *2D SEC-DED* šeme kodiranja je superiornija. Za izrazito šumne kanale koji se karakterišu vrednošću $BER > 10^{-2}$ ponovno slanje paketa za obe šeme kodiranja predstavlja jedino moguće rešenje.

- 2) Generalno za slanje N_y paketa, broj paketa za vremenski period od jedne godine, potrebna energije je:

$$E_{t_y} = P_{out} \cdot t_y = P_{out} \cdot t_b \cdot n_{bp} \cdot (N_y + \Delta N_y) \quad (14.4)$$

ili gledano sa drugog aspekta

$$E_{t_y} = N_b \cdot E_{pb} = N_b \cdot U \cdot I \cdot t_b = N_b \cdot P_{out} \cdot t_b = P_{out} \cdot t_b \cdot n_{bp} \cdot (N_y + \Delta N_y) \quad (14.5)$$

gde je P_{out} - predajna izlazna snaga; t_y - vreme potrebno za uspešno slanje svih paketa u toku vremenskog perioda od jedne godine; $t_b = 1/R$ - vreme potrebno za slanje jednog bita; n_{bp} - odgovara broju bitova po paketu (uključuje poruke *RTS*, *CTS*, *DATA* i *ACK/NACK*); N_y - broj pokušaja slanja paketa u toku vremenskog perioda od jedne godine (direktna posledica vremena koje protekne između slanja dva susedna paketa tzv. *beacon* perioda i u konkretnom slučaju ono iznosi 2 minuta); ΔN_y - broj retransmitovanih paketa u toku vremenskog perioda od jedne godine; N_b - ukupan broj bitova koji se šalje u vremenskom period od jedne godine.

Sada ćemo razmotriti slučaj kada je ispunjen uslov jednakosti potrošnje energije:

$$E_{total} = E_{CRC} = P_{out_CRC} \cdot t_{y_CRC} = E_{2D} = P_{out_2D} \cdot t_{y_2D} \quad (14.6)$$

Kao posledica smanjenog ili potpuno eliminisanog broja retransmisija vreme potrebno za slanje svih paketa (u najboljem slučaju jednako broju N_y) kodiranih šemom kodiranja *2D SEC-DED* je manje ($t_{y_2D} \ll t_{y_CRC}$) pošto za ovu šemu kodiranja važi da $\Delta N_y \rightarrow 0$. Ova činjenica nam pruža mogućnost značajne uštede energije i jedna od posledica je mogućnost povećanja izlazne predajne snage P_{out_2D} na račun smanjenje ili eliminisane retransmisije što implicira da na ovaj način povećavamo opseg prenosa (domet).

- 3) Za ciljani $BER = 10^{-3}$ i eksponent gubitaka na putu vrednosti $\alpha = 2/4/6$ kao parametar (videti Sl, 14-3) dobijamo $G_D = 216\%/78\%/47\%$, a za ciljani $BER = 10^{-5}$ dobijamo $G_D = -68\%/-44\%/-32\%$. Pozitivna vrednost metrike dobit u rastojanju ukazuje da je šema kodiranja *2D SEC-DED* superiornija (u području za vrednosti $BER > 10^{-4}$), dok negativna

vrednost metrike ukazuje na činjenicu da je šema kodiranja *CRC* poželjnija (u području za vrednosti $BER < 10^{-4}$). Za identičnu potrošnju energije u periodu od jedne godine pozitivna dobit se može tumačiti kao rast (uvećanje) koje indirektno dovodi do povećanja dometa tj. opsega prenosa.

- 4) Još jedno interesantno područje primene bežičnih senzorskih mreža je zatvoren prostor tzv. *indoor* okruženje u kome važi da je eksponent gubitaka na putu $\alpha = 4$ i više. U takvom okruženju i za ciljanu vrednost stope pogrešno prenetih bitova (*BER*) od $10^{-3} / 5 \cdot 10^{-4} / 3 \cdot 10^{-4} / 10^{-4}$ kao parametar, predložena šema kodiranja *2D SEC-DED* u stanju je da poboljša rastojanje predaje (domet) za oko $22.41m / 18.00m / 12.68m / 0m$ ili da poboljša *RSSI* za oko $10dBm / 8.47dBm / 6.35dBm / 0dBm$, respektivno, u poređenju sa bežičnim senzorskim mrežama bez sposobnosti korekcije grešaka tokom kodiranja. Na primer, za ciljane vrednosti $BER = 5 \cdot 10^{-4}$, $\alpha = 4$ i $R_s = -90dBm$ korišćenjem *CRC* šeme kodiranja maksimalno rastojanje koje se postiže je $d_{CRC} = 28.72m$. Pod istim uslovima, kada se koristi *2D SEC-DED* šema kodiranja maksimalno rastojanje koje se postiže je $d_{2D} = 46.76m$ (tačka *L* na Sl. 14-1 a)). Razlika u rastojanju koja se na ovaj način ostvaruje je $\Delta d = d_{2D} - d_{CRC} \approx 18m$. Virtuelno poboljšanje senzitivnosti prijemnika od kataloške vrednosti $R_s = -90dBm$ na $R_s = -98.47dBm$, uz istovremeno održavanje istih performansi greške je direktna posledica korišćenja *2D SEC-DED* šeme kodiranja.

15 ZAKLJUČAK

Tokom protekle dve decenije bežične senzorske mreže su privukle veliku pažnju u istraživačkim i naučnim krugovima, uglavnom iz razloga rapidnog povećanja broja njihovih aplikacija. Brzi porast broja aplikacija bežičnih senzorskih mreža, koje su uglavnom baterijski napajane, postavio je značajne izazove pred projektante. Glavni izazovi su dizajn senzorskog čvora male potrošnje, visoka pouzdanost pri određenoj brzini isporuke paketa, dugoročno senzovanje, izbor kodova za detekciju ili kodova za ispravljanje grešaka koje postižu nisku stopu grešaka u prenosu bitova pri umerenim vrednostima SNR i postizanje dugog životnog veka bežične senzorske mreže. Životni vek bilo koje bežične senzorske mreže direktno zavisi od efikasnog korišćenja izvora energije senzorskih čvorova koji je čine. Snaga se prvenstveno troši tokom aktivnosti bežične predaje i prijema, ali analiza je pokazala da i ostali blokovi imaju itekako značajan uticaj na potrošnju i, u najmanju ruku, nikako ih ne smemo zanemariti. U cilju odgovora na navedene izazove uglavnom se koriste agresivne tehnike upravljanja potrošnjom energije uz podrazumevano ostvarivanje aktivnosti prikupljanja podataka, njihovo procesiranje i komunikacije podataka do centralnog čvora.

U prvom delu, disertacija je pružila sveobuhvatan uvid u aspekte potrošnje energije senzorskog podsistema u okviru senzorskog čvora. Da bismo postigli smanjenje potrošnje energije tokom aktivnosti senzovanja informacija iz okruženja predložili smo korišćenje kombinacije dve tehnike za uštedu energije. Prvu, nazvanu *duty-cycling* koristimo za smanjenje potrošnje snage na nivou sistema tj. upravljamo arhitekturom senzorskog čvora između aktivnog i stanja spavanja. Drugu, nazvanu *power-gating* ili upravljanje napajanjem koristimo za uključivanje i isključivanje senzorskih elemenata tj. senzora u okviru senzorskog podsistema tokom aktivnosti pribavljanja

informacija iz spoljašnjeg okruženja. Dobijeni rezultati na osnovu analize, merenja i validacije korišćenjem *MATLAB* softverskog okruženja pokazuju da se, primenom pomenute dve tehnike, u proseku može postići smanjenje potrošnje energije za tri reda veličine. Tako na primer, za vremenski period od dva minuta potrošnja energije kada se primenjuju obe tehnike je $\approx 638 \mu J$ u poređenju sa $\approx 3.92 J$ u slučaju kada se senzorski čvor nalazi sve vreme u aktivnom stanju i svi senzori su uključeni.

Takođe jedan od načina da se sačuva energija senzorskog čvora je da se što više minimizira ponovno slanje paketa na nivou veze, a koje je posledica prisustva grešaka u nepouzdanj bežičnoj komunikaciji, što predstavlja drugi deo disertacije. Problem se rešava korišćenjem efikasne *FEC* tehnike kodiranja kanala kojom se povećava brzina isporuke paketa. *FEC* tehnika kodiranja pruža mogućnost prijemniku da ispravlja greške bez potrebe za povratnim kanalom putem koga bi zahtevao ponovno slanje podataka, ali ta prednost je na račun povećanja fiksne propusnosti kanala kao i povećane potrošnje energije. Izbor *FEC* koda ima direktan uticaj na uštedu energije senzorskog čvora, a u korelaciji je sa sposobnošću otkrivanja i ispravljanja grešaka.

U cilju ispunjenja navedenih zahteva u navedenoj disertaciji predložena je nova šema kodiranja komunikacionih kanala u primeni senzorskih čvorova u aplikacijama praćenja okruženja u "teškim" radnim uslovima koji se karakterišu stopom pogrešnih bitova $BER \geq 10^{-4}$, aplikacijama logistike, itd. Predložena implementacija tehnike kanalnog kodiranja nazvana *2D SEC-DED/SEC-DED* minimizira potrebu za ponovnim slanjem paketa i u velikoj meri pomaže u smanjenju potrošnje energije. Osnovna ideja, kao što je prezentovano, sastoji se u korišćenju dvodimenzionalnog kontrolnog koda. Naime, kombinovanjem mogućnosti otkrivanja grešaka pravougaonih kodova i mogućnosti ispravljanja grešaka *Hamming*-ovog produkt koda proširenog ukupnom parnošću po horizontalnim i vertikalnim pravcima, ostvarujemo značajna poboljšanja. Takođe, uvođenje nešto veće redundanse manifestuje se relativno malim povećanjem latencije u toku prenosa paketa kroz komunikacioni link, a glavna dobit ogleđa se u smanjenoj potrošnji energije senzorskog čvora.

Rezultati dobijeni direktnim merenjima i iscrpnim simulacijama pokazuju sledeće:

- 1) Ispravljeno je 100% prisutnih jednostrukih grešaka i $99,99\%$ prisutnih dvostrukih/višestrukih grešaka. Napomenimo da sa tačke gledišta tolerantnosti sistema na grešku uvođenje *2D SEC-DED/SEC-DED* tehnike kodiranja je ekvivalentno implementaciji *TMR* šeme redundantnosti sa glasanjem na nivou reči, tj. umesto da koristimo trostruku repliku paketa mi koristimo samo jedan paket sa ne tako velikom redundantnošću.

- 2) Kada bežična senzorska mreža koristi sinhroni protokol (kao što je *Rendezvous* protokol) i radi u "teškim" uslovima praćenja okruženja koje se karakteriše stopom pogrešnih bitova $BER \geq 10^{-4}$, opravdano je koristiti *2D SEC-DED/SEC-DED* tehniku kodiranja (bazirana na *FEC*-u) umesto široko zastupljene i u protokolu korišćene *CRC* tehnike kodiranja (bazirana na *ARQ*-u). Ova teza, koja je dokazana u disertaciji, posledica je činjenice da *FEC* kodiranje pruža prijemniku sposobnost da otkrije i ispravi greške sa minimalnim zahtevima za povratnim kanalom putem koga bi zahtevao ponovno slanje paketa. Kvantitativno je iskazana kroz potrošnju energije za različite vrednosti *BER*-a.
- 3) Uopšteno govoreći, kada se *beacon* period povećava u opsegu $30\text{ s} \leq T_{BP} \leq 5\text{ min}$ potrošnja energije radio bloka čvora koji šalje, E_{radio} , se smanjuje u oblasti kada *2D SEC-DED/SEC-DED* ima superiornije performanse u odnosu na *CRC* šemu kodiranja prema nižim vrednostima stope pogrešno prenetih bitova u opsegu od $10^{-4} \geq BER \geq 7 \cdot 10^{-6}$.
- 4) Za heterogene hardverske strukture senzorskih čvorova (heterogenost je prvenstveno izražena kroz broj senzorskih elemenata i *ADC* rezoluciju) i $T_{BP} = 2\text{ min}$ u radnoj oblasti $BER > 10^{-5}$ dobit energije u oznaci *EG*, kao performansna metrika, je $0.23\% \leq EG \leq 47.63\%$. To podrazumeva da je upotreba *2D SEC-DED/SEC-DED* šeme kodiranja poželjnija u odnosu na *CRC*. Nasuprot tome, u radnoj oblasti $BER < 10^{-5}$ kada se koristi *CRC* šema kodiranja imamo da je $0.03\% \leq EG \leq 2.34\%$.
- 5) Predložena implementacija *2D SEC-DED/SEC-DED* šeme kodiranja je efektivna kada vrednost stope grešaka *BER* nije visoka, a većina grešaka su jednostruke prirode. Kada je većina grešaka tipa *burst* efikasnost predložene šeme kodiranja u smanjenju gubitaka paketa direktno zavisi od šablona pojavljivanja grešaka i detaljnije je objašnjena u Dodatku A.

Na bazi rezultata koji su proistekli iz predložene tehnike kodiranja objašnjenje u drugom delu disertacije, u trećem delu je dat jedan od načina iskorišćenja dobiti poboljšanjem performansi bežičnog kanala. Naime, analiza za prenos signala u zatvorenom prostoru je pokazala da za eksponent gubitaka na putu $\alpha = 4$ na ciljanom $BER = 5 \cdot 10^{-4}$ predložena šema kodiranja je u stanju da poboljša domet za oko 18 m ili indikaciju jačine primljenog signala (*RSSI*) za oko 8.5 dBm u poređenju sa *O-RPLL* protokolom kod koga je ugrađena *CRC* tehnika kodiranja kojom se ostvaruje *ARQ* mehanizam oporavka pogrešno primljenih paketa.

16 LITERATURA

- [1] Sherine M. Abd El-kader and Basma M. Mohammad El-Basioni, Precision Farming Solution in Egypt Using the Wireless Sensor Network Technology, *Egyptian Informatics Journal*, 14(3):221–233, 2013. ISSN 1110-8665.
- [2] M. Mafuta, M. Zennaro, A Bagula, G. Ault, H. Gombachika, and T. Chadza, Successful deployment of a Wireless Sensor Network for precision agriculture in Malawi, In *IEEE 3rd International Conference on Networked Embedded Systems for Every Application (NESEA)*, pages 1–7, Dec 2012.
- [3] J. A. Lopez Riquelmea, F. Sotoa, J. Suardiaza, P. Sanchezza, A. Iborraa, and J. A. Verab, Wireless Sensor Networks for precision horticulture in Southern Spain, *Computers and Electronics in Agriculture*, 68(1), 2009.
- [4] Swain. D., G. Bishop-Hurley, and J. Griffiths, Automatic cattle control systems –grazing without boundaries, *Farming Ahead*, June 2009.
- [5] W. Hu, N. Bulusu, C.T. Chou, S. Jha, A. Taylor, and V.N. Tran, Design and Evaluation of a Hybrid Sensor Network for Cane Toad Monitoring, *ACM Transactions on Sensor Networks*, 5(1), 2009.
- [6] S. Verstocket, V. Slavkovikj, P. De Potter, and R. Van de Walle, Collaborative bike sensing for automatic geographic enrichment: Geoannotation of road/terrain type by multimodal bike sensing, *Signal Processing Magazine, IEEE*, 31(5):101–111, Sept 2014. ISSN 1053–5888.
- [7] I.F. Akyildiz, M.C. Vuran, *Wireless Sensor Networks*, JohnWiley & Sons Ltd., Chester UK, 2010.
- [8] F. Koushanfar, M. Potkonjak, A. Sangiovanni-Vincentelli, Fault Tolerance in Wireless Sensor Networks, in: *Handbook of Sensor Networks*, by I. Mahgoub and M. Ilyas (eds.), Section VIII, Ch. 36, pp. 36.1-13, CRC Press, Boca Raton, 2004.
- [9] G. Anastasi, M. Conti, M. Di Francesco, A. Passarella, Energy conservation in wireless sensor networks: A survey, *Ad Hoc Netw.* 7 (2009), pp. 537–568.
- [10] F. Yang, *Reliable and Time-Constrained Communication in Wireless Sensor Networks* (PhD thesis, Ecole doctorale) Informatique et Mathématiques, INSA de Lyon, 2011.
- [11] H. Liu, A. Nayak, I. Stojmenović, Fault-Tolerant Algorithms/Protocols in Wireless Sensor Networks, in: S. Misra, et al., (Eds.), *Guide to Wireless Sensor Networks*, Computer Communications and Networks, Springer-Verlag, London 2009, pp. 261–291 (Ch. 10).

- [12] L. M. Sá de Souza, H. Vogt, M. Beigl, A Survey on Fault Tolerance in Wireless Sensor Networks, Interner Bericht, Universität Karlsruhe Fakultät für Informatik, 2007, https://books.google.rs/books/about/A_survey_on_fault_tolerance_in_wireless.html?id=Yh0ZnwEACAAJ&redir_esc=y, 2015, (accessed 20.09.2015)
- [13] C. Jeevanandam, K. Ramalakshmi, D. Priya, A Survey on Energy Efficient Fault Tolerance Mechanisms in Wireless Sensor Networks, International Journal of Emerging Technology and Advanced Engineering, Volume 4, Issue 1, January 2014, pp. 330-334.
- [14] S. Aggarwal, R. Rani, A review of fault detection techniques for wireless sensor networks, Int. J. Comput. Sci. Issues 10 (4), July 2013, pp. 127–138 (No 1.).
- [15] V. Raghunathan, S. Ganerival, and M. Srivastava, "Emerging Techniques for Long Lived Wireless Sensor Networks", IEEE Communication Magazine, 2006, Vol.41, No. 4, pp. 130-141
- [16] M. Stojcev, M. Kosanovic, Lj. Golubovic, Power Management and Energy Harvesting Techniques for Wireless Sensor Nodes, In Proc. of 9th TELSIXS Conference, IEEE, Nis, Serbia, 7-9, October 2009, pp. 65-72
- [17] G. Pottie, W. Kaiser, Wireless integrated network sensors, Comm. ACM 43 (5), 2000, pp. 51–58.
- [18] J. Jeong, C.-T. Ee, Forward Error Correction in Sensor Networks, Proceedings of the First International Workshop on Wireless Sensor Networks (WWSN), Marrakesh, Morocco, 4–8, June, 2007, pp. 1–6.
- [19] Gungor, V. C., & Hancke, G. P. (2009). Industrial Wireless Sensor Networks: Challenges, Design Principles, and Technical Approaches. IEEE Transactions on Industrial Electronics.
- [20] Song, J., Han, S., & Mok, A. K. (2008). WirelessHART: Applying Wireless Technology in Real-Time Industrial Process Control. IEEE Real-Time and Embedded Technology and Applications Symposium, (pp. 377-386). St. Louis.
- [21] De Biasi, M., Snickars, C., Landernäs, K., & Isaksson, A. J. (2008). Simulation of Process Control with WirelessHART Networks Subject to Packet Losses. 4th IEEE Conference on Automation Science and Engineering, (pp. 548-553). Washington DC.
- [22] Saifullah, A., Xu, Y., Lu, C., & Chen, Y. (2010). Real-Time Scheduling for WirelessHART Networks. 2010 31st IEEE Real-Time Systems Symposium, (pp. 150-159). San Diego. 27
- [23] HART Communication Foundation. (2007). 2.4GHz DSSS O-QPSK Physical Layer Specification. Austin: HART Communication Foundation.
- [24] M. Vuran, and I. Akyildiz, Error Control in Wireless Sensor Networks: A Cross Layer Analysis, IEEE/ACM Transactions on Networking, Vol. 17, No.4, August 2009, pp. 1186-1199
- [25] S. Horward, C. Schiegel, and K. Iniewski, Error Control Coding in Low-Power Wireless Sensor Networks: When is ECC Energy Efficient?, EURASIP Journal on Wireless Communications and Networking, Vol. 2006, No. 2, 2006, pp.1-14,
- [26] Kleinschmidt, J. H., Borelli, W. C., & Pellenz, M. E. (2007). An analytical model for energy efficiency of error control schemes in sensor networks. International Conference on Communications 2007, (pp. 3895-3900). Glasgow.
- [27] G. Balakrishnan, M. Jang, Y. Jiang, and Y. Kim, Performance Analysis of Error Control Codes for Wireless Sensor Networks, in Proceedings of the 4th International Conference on Information Technology (ITNG), 2-4 April 2007, Las Vegas, NV, USA, 2007, pp. 876-879
- [28] Sankarasubramaniam, Y., Akyildiz, I. F., & McLaughlin, S. W., Energy efficiency based packet size optimization in wireless sensor networks. IEEE Int. Workshop on Sensor Network Protocols and Applications, 2003, pp. 1-8.

- [29] Min, R., Bhardwaj, M., Cho, S.-H., Ickes, N., Shih, E., Sinha, A., et al. (2002). Energy-Centric Enabling Technologies for Wireless Sensor Networks. *IEEE Wireless Communications*, pp. 28-39.
- [30] Shih, E., Cho, S., Lee, F. S., Calhoun, B. H., & Chandrakasan, A. (2004). Design Considerations for Energy-Efficient Radios in Wireless Microsensor Networks. *Journal of VLSI Signal Processing* 37, pp. 77–94.
- [31] Björnemo, E. (2009). Error correcting codes in static channels. In *Energy Constrained Wireless Sensor Networks*. Uppsala.
- [32] Goldsmith, A. (2005). *Wireless Communications*. New York: Cambridge University Press.
- [33] Sklar, B. (2003). *Modulation and Coding Trade-Offs*. In *Digital Communications*. New Jersey: Prentice Hall PTR.
- [34] Proakis, J. G. (2001). *Block and Convolutional Channel Codes*. In *Digital Communications*. New York: McGraw-Hill.
- [35] J. Costello, and J. Forney, Channel Coding: The Road to Channel Capacity, in *Proceedings of the IEEE*, Vol. 95, No. 6, June 2007, pp. 1150- 1170
- [36] S. Lin, D. J. Costello, *Error Control Coding*, 2nd ed. Upper Saddle River, New Jersey: Pearson Prentice Hall; 2004.
- [37] T. K. Moon, *Error Correction Coding Mathematical Methods and Algorithms*, Hoboken, New Jersey, John Wiley & Sons; 2005
- [38] R. Agarwal, E. Popovici, and B. O’Flynn, Adaptive Wireless Sensor Networks: A System Design Perspective to Adaptive Reliability, in *Conference Proceedings of Wireless Communications and Sensor Networks (WCSN’06)*, 2006
- [39] N. Alrajeh, U. Marwal, B. Shams, and S. Shah, Error Correcting Codes in Wireless Sensor Networks: An Energy perspective, *Applied Mathematics & Information Systems, An International Journal*, Vol. 9, No. 2, 2015, pp. 809-818
- [40] Z. Wang, Z. Cui, and J. Sha, VLSI Design for Low-Density Parity-Check (LPDC) Decoding, *IEEE Circuits and Systems Magazine*, Vol. 11, No.2, First Quarter 2011, pp. 52-69
- [41] Y. Yitbarek, K. YU, J. Akerberg, M. Gidlund, and M. Bjorkman, Implementation and Evaluation of Error Control Schemes in Industrial Wireless Sensor Networks, in *Proceedings of the IEEE International Conference on Industrial Technology (ICIT)*, 26 Feb.-1 March 2014, Busan, South Korea, 2014, pp. 730-735
- [42] M. Mutschlechner, B. Li, and F. Dressler, Using Erasure Codes to Overcome Reliability Issues in Energy-Constrained Sensor Networks, 11th IEEE/IFIP Conference on Wireless on Demand Networks and Services (WONS 2014), April 2014, Obergurgl, Austria, 2014, pp. 41-48
- [43] R. Anane, R. Bouallegue, and K. Raoof, Extensive Simulation Performance Evaluation of MSK Scheme with Channel Coding in Wireless Sensor Networks, *The 29th IEEE International Conference on Advanced Information Networking and Applications (AINA-2015)*, March 25-27, Korea, 2015, pp. 400-405
- [44] R. Anane, K. Raoof, and R. Bouallegue, On the Evaluation of GMSK Scheme with ECC Techniques in Wireless Sensor Networks, *International Journal of Wireless & Mobile Networks (IJWMN)*, Vol. 7, No.2, April 2015, pp. 17-28
- [45] M. Holland, T. Wang, B. Tavli, A. Seyedi, and W. Heinzelman, Optimizing Physical-Layer Parameters for Wireless Sensor Networks, *ACM Transactions on Sensor Networks*, Vol. 7, No. 4, pp. 28:1-28.20
- [46] S. Kasnavi, S. Kilambi, B. Crowley, K. Iniewski, and B. Kaminska, Application of Error Control Codes (ECC) in Ultra-Low Power RF Transceivers, *Architecture, Circuit and*

- Implementation of SoCs, in Proceedings of the 2005 Dallas/CAS Workshop, 10-13 October, Dallas, Texas, USA, 2005, pp. 195-198
- [47] T. Öberg, Modulation, Detection, and Coding, John Wiley & Sons, LTD, Chichester, UK, 2001
- [48] G. Chen, S. Hanson, D. Blaauw, and D. Silvester, "Circuit Design Advances for Wireless Sensing Applications", Proceedings of the IEEE, Vol. 98, No. 11, November 2010, (pp. 1808-1826)
- [49] Texas Instruments, datasheet for Ultra-Low-Power Microcontroller MSP430FR5969, <http://www.ti.com/product/msp430fr5969>, 2015 (accessed 20.08.2015).
- [50] Texas Instruments, datasheet for Value Line Transceiver CC110L, available at <http://www.ti.com/product/cc110l>, 2015 (accessed 20.08.2015).
- [51] W. Dargie, "Dynamic Power Management in Wireless Sensor Networks: State-of-the-Art", Sensors Journal, IEEE, Vol. 12, No. 5, 2012, pp. 1518 - 1528
- [52] F. Juan, B. Lian, and Z. Hongwei, "Hierarchically Coordinated Power Management for Target Tracking in Wireless Sensor Networks", International Journal of Advanced Robotic Systems, Feb. 2013, Vol. 10, pp. 1 - 14
- [53] V. Jeličić, "Power Management in Wireless Sensor Networks with High-Consuming Sensors", Technical project report, April 2011, University of Zagreb, Faculty of Electrical Engineering and Computing, pp. 1-9, av. February 2014 at <http://www.ztel.fer.unizg.hr/download/repository/VJelicic,KDI.pdf>
- [54] H. Joe, J. Park, C. Lim, D. Woo, and H. Kim, "Instruction-Level Power Estimator for Sensor Networks", ETRI Journal, Vol. 30, No. 1, February 2008, (pp. 47 - 58)
- [55] G. Panić, Z. Stamenković, R. Kraemer, Power Gating in Wireless Sensor Networks, Wireless Pervasive Computing, ISWPC2008, 3rd International Symposium on, Santorini, Greece, May 2008, pp. 499-503.
- [56] Leibniz Institute for High Performance Microelectronics – IHP, Frankfurt (Oder), Germany, <http://www.ihp-microelectronics.com>
- [57] N.S. Kim, T. Austin, D. Baauw, T. Mudge, K. Flautner, J.S. Hu, M.J. Irwin, M. Kandemir, and V. Narayanan "Leakage Current: Moore's Law Meets Static Power ", IEEE Computer, Dec. 2003, Vol.36, No. 12, pp. 68 - 75
- [58] M. Kosanovic, M. Stojcev, "RPATS – Reliable Power Time Synchronization Protocol", Microelectronics Reliability, Vol. 54. No. 1, 2014, pp.303-315
- [59] T. Schmid, R. Shea, Z. Charbiwala, J. Friedman, M. Srivastava, and Y. Cho, "On the Interaction of Clocks, Power, and Synchronization in Duty-Cycled Embedded Sensor Nodes", ACM Transactions on Sensor Network, Vol. 7, No. 3, 2010, (pp. 1-19), article No. 24
- [60] J. Fraden, "Handbook of Modern Sensors: Physics, Designs, and Applications", Fourth Edition, Springer New York, 2010
- [61] G. Nikolic, G. Panic, Z. Stamenkovic, G. Jovanovic and M. Stojcev, "Implementation of External Power-Gating Technique During Sensing Phase in Wireless Sensor Networks", 29th International Conference on Microelectronics MIEL 2014, Belgrade, Serbia, 12-15 May 2014, accepted for presentation
- [62] Online catalogue of www.farnell.com av. at January. 2014
- [63] Texas Instruments, Low-Dropout Voltage Regulator, av. at www.ti.com/lit/gpn/tlv716120275p, January. 2014
- [64] Texas Instruments, Low Ron Load Switch, av. at <http://www.ti.com/lit/ds/symlink/tps22908.pdf>, January. 2014

- [65] S. Kim, R. Fonseca, D. Culler, Reliable Transfer on Wireless Sensor Networks, SECON (Sensor and Ad Hoc Communications and Networks), Santa Clara, California 4–7 October 2004, pp. 449–459.
- [66] George C. Clark, Jr. and J. Bibb Cain, Error-Correction Coding for Digital Communications, Harris Corporation, Melbourne, Florida, 1981, pp.3
- [67] Application Note, AN207, CRC-16-CCITT, <https://www.maximintegrated.com/en/app-notes/index.mvp/id/27> (accessed 2017)
- [68] Robin Hoel, Texas Instruments Design Note DN504, SWRA113A, FEC Implementation, <http://www.ti.com/lit/an/swra113a/swra113a.pdf> (accessed 2017)
- [69] M.G. Luby, M. Mitzenmacher, M.A. Shokrollahi, D.A. Spielman, Efficient erasure correcting codes, IEEE Trans. Inf. Theory 47 (2) (2001) 569–584
- [70] M. R. Ahmad, E. Dutkiewicz, X. Huang, A Survey of Low Duty Cycle MAC Protocols in Wireless Sensor Networks, Ch. 5, pp. 69 – 90, in Emerging Communications for Wireless Sensor Networks, by A. Foerster and A. Foerster, eds, InTech, Rijeka, Croatia, 2011.
- [71] J. N. Al-Karaki, A. E. Kamal, Routing Techniques in Wireless Sensor Networks: A Survey, IEEE Wireless Communications, Vol. 11, No. 6, (2004), pp. 6-28.
- [72] A. Boulis, S. Ganeriwal, and M. Srivastava, Aggregation in Sensor Networks: An Energy Accuracy Trade-Off, Ad Hoc Networks, Vol. 1, (2003), pp. 317–331.
- [73] B. H. Calhoun, D. C. Daly, N. Verma, D. Finchelstein, D. D. Wentzloff, A. Wang, S.-H. Cho, A. P. Chandrakasan, Design Considerations for Ultra-Low Energy Wireless Micro-Sensor Nodes, IEEE Trans. Computers, Vol. 54, No. 6, (2005), pp. 727-740
- [74] E. Y. Lin, A Comprehensive Study of Power-Efficient Rendezvous Schemes for Wireless Sensor Networks, PhD thesis, University of California, Berkeley, 2005.
- [75] M. R. Kosanović, M. K. Stojčev, RPLL - Rendezvous Protocol for Long-Living Sensor Node, Facta Universitatis Series: Electronics and Energetics Vol. 28, No 1, (2015), pp. 85 – 102.
- [76] M. Brzozowski, K. Piotrowski, P. Langendoerfer, A Cross-Layer Approach for Data Replication and Gathering in Decentralized Long-Living Wireless Sensor Networks, ISADS 2009. In: Proc. of the 9th international symposium on autonomous decentralized system, Athens; 2009. pp. 49–54.
- [77] Z. Rosberg, R.P. Liu, A. Y. Dong, L.D. Tuan, S. Jha, ARQ with Implicit and Explicit ACKs in WSN, In: Proc. of the Global Telecommunications Conference IEEE GLOBECOM 2008, New Orleans, LO; 2008, pp. 1–6.
- [78] B. Fu, P. Ampadu, On Hamming Product Codes with Type-II Hybrid ARQ for On-Chip Interconnects, IEEE Trans. Circuits Syst. I, Vol. 56, (2009), pp. 2042–2054.
- [79] N. Sadaghi, S. Howard, S. Kasnavi, K. Iniewski, V. Gaudet, and G. Schlegel, Analysis of Error Control Code Use in Ultra-Low-Power Wireless Sensor Networks, Proceedings. 2006 IEEE International Symposium on Circuits and Systems, (ISCAS 2006), 21-24 May 2006, Island of Kos, Greece, 2006, 4 pp.-3561
- [80] K. Srinivasan, P. Dutta, A. Tavakoli, P. Levis, An Empirical Study of Low Power Wireless, ACM Transactions on Sensor Networks, Vol. 6, No. 2, Feb. 2010, Article No. 16.
- [81] Z. Rosberg, R.P. Liu, T.L. Dinh, Y.F. Dong, S. Jha, Statistical Reliability for Energy Efficient Data Transport in Wireless Sensor Networks, Wireless Networks, Vol. 16, No. 7, (2010), pp. 1913 – 1927.
- [82] C. T. Bhunia, Information Technology Network and Internet, New Age International Publisher, New Delhi, (2005), pp. 173

- [83] Mathworks, *randerr*, Generate bit error patterns, at: <http://www.mathworks.com/help/comm/ref/randerr.html?requestedDomain=www.mathworks.com>, 2016, (accessed 20.2.2016)
- [84] Yi Huang, Kevin Boyle, *Antennas from Theory to Practice*, John Wiley & Sons, 2008, UK
- [85] S. Y Seidel and T. S. Rappaport, "914 MHz path loss prediction models for indoor wireless communications in multifloored buildings," *IEEE Transactions on Antennas and Propagation*", pp. 207–217, Feb. 1992.
- [86] A.J. Motley and J.M.P. Keenan, "Personal communication radio coverage in buildings at 900 MHz and 1700 MHz," *Electronic Letters*, pp. 763–764, June 1988.
- [87] A. F. Toledo and A.M.D. Turkmani, "Propagation into and within buildings at 900, 1800, and 2300 MHz," *Proc. IEEE Vehicular Technology Conference*, pp. 633-636, May 1992.
- [88] F.C. Owen and C.D. Pudney, "Radio propagation for digital cordless telephones at 1700 MHz and 900 MHz," *Electronic Letters*, pp. 52-53, Sept. 1988.

17 PRILOG A

Ilustrativni primer primene 2D SEC-DED koda prikazan matricno

Izgled vektora nakon ubacivanja bitova slučajnih grešaka kao šum (prisustvo jedinice ukazuje na poziciju bita greške).

																				# grešaka	
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1
0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	2
0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0	1
0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1	0	0	0	2
0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	2
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	2
0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0	0	0	0	0	0	2
0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1
0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Matrica 21×11 predstavlja 11 vektora sa po 21-im bitom koji su kreirani po sledećem principu

$$\left[\begin{array}{l} \text{Ukupna vertikalna parnost} + \text{Ukupna horizontalna parnost} + \\ + \text{Hammingovi horizontalni bitovi} + \text{Hammingovi vertikalni bitovi} + \\ + \text{Originalna poruka} \end{array} \right]$$

za istu matricu generator slučajnih grešaka generiše 16 bit pozicija, označenih jedinicom i crvenom bojom, koje predstavljaju pozicije grešaka u paketu koji se prenosi. Dakle u ovom primeru imamo jedan od mogućih šablona grešaka čija je stopa pogrešno prenetih bitova izrazito visoka i iznosi $BER = 16 / (21 \times 11) = 16 / 231 \approx 7 \times 10^{-2}$ i kodnim odnosom $n = (10 \times 11) / (21 \times 11) = 110 / 231 = 0.476$.

Korak_1: Horizontalno dekodiranje - proces se odvija u skladu sa algoritmom koji je prezentovan u disertaciji gde najpre utvrdimo koji je sindrom za svaki od vektora, a zatim poređenjem sindroma kolona S i bita parnosti kolona P utvrđujemo koliko pogrešnih bitova po vektoru imamo i na kraju određujemo koju akciju da preduzmemo.

																				S	P	Tip greške	Akcija	#vrste
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Nakon primene Korak_1 tj. sprovedene korekcije izgled vektora u matrici 11×11 je sledeći

																					S	P	Tip greške	Akcija	#vrste	
																					0	0	1	single	koriguj	1
																					■	■	0	double	sledeća	2
																					0	0	1	single	koriguj	3
																					0	0	0	double	sledeća	4
																					0	0	0	double	sledeća	5
																					0	0	1	single	koriguj	6
																					0	0	0	double	sledeća	7
																					0	0	0	double	sledeća	8
																					0	0	1	parity	ništa	9
																					0	0	0	no error	ništa	10
																					0	0	0	no error	ništa	11

Zaključak je da su samo vektori koji imaju dvostruku ili višestruku grešku ostali nekorigovani.

Napomenimo da u okviru pripreme za naredni korak prethodno prikazanu matricu 11×11 najpre treba da transponujemo, a zatim da joj pridružimo četiri kolone iz matrice 21×11 koji predstavljaju vertikalne *Hamming*-ove bitove. Sa tako pripremljenom matricom ulazimo u Korak_2: Vertikalno dekodiranje koje se svodi na iste aktivnosti kao i u Korak_1

Korak_2:

																			<i>S</i>	<i>P</i>	<i>Tip greške</i>	<i>Akcija</i>	<i># vrste</i>	
		0	0	0	0				0	♣	0	0	0	0	0	0	0	0	0	≠0	1	<i>single</i>	<i>koriguj</i>	<i>1</i>
		0	0	0	0				0	♣	0	0	0	0	0	0	0	0	0	≠0	1	<i>single</i>	<i>koriguj</i>	<i>2</i>
		0	0	0	0				0	0	0	0	0	0	0	0	♣	0	0	≠0	1	<i>single</i>	<i>koriguj</i>	<i>3</i>
		0	0	0	0				0	0	0	0	0	0	0	0	0	0	0	0	0	<i>no err</i>	<i>ništa</i>	<i>4</i>
		0	0	0	0				0	0	0	0	0	0	0	0	0	0	0	0	0	<i>no err</i>	<i>ništa</i>	<i>5</i>
		♣	0	0	0				0	0	0	0	0	0	0	♣	♣	0	0	≠0	1	<i>triple</i>	<i>sledeća</i>	<i>6</i>
		0	0	0	0				0	0	0	♣	0	0	♣	0	0	0	0	≠0	0	<i>double</i>	<i>sledeća</i>	<i>7</i>
		0	0	0	0				0	0	0	0	♣	0	0	0	0	0	0	≠0	1	<i>single</i>	<i>koriguj</i>	<i>8</i>
		0	0	0	0				0	0	0	0	0	0	0	0	0	0	0	0	0	<i>no err</i>	<i>ništa</i>	<i>9</i>
		0	0	0	♣				0	0	0	♣	0	0	0	0	0	0	0	≠0	0	<i>double</i>	<i>sledeća</i>	<i>10</i>
		♣	0	0	0				0	0	0	0	0	0	0	0	0	0	0	0	1	<i>parity</i>	<i>ništa</i>	<i>11</i>

Nakon primene Korak_2 tj. sprovedene korekcije izgled vektora u matrici 11×11 je sledeći

																				<i>S</i>	<i>P</i>	<i>Tip greške</i>	<i>Akcija</i>	<i># vrste</i>
									0	0	0	0	0	0	0	0	0	0	0	≠0	1	<i>single</i>	<i>koriguj</i>	<i>1</i>
									0	0	0	0	0	0	0	0	0	0	0	≠0	1	<i>single</i>	<i>koriguj</i>	<i>2</i>
									0	0	0	0	0	0	0	0	0	0	0	≠0	1	<i>single</i>	<i>koriguj</i>	<i>3</i>
									0	0	0	0	0	0	0	0	0	0	0	0	0	<i>no error</i>	<i>ništa</i>	<i>4</i>
									0	0	0	0	0	0	0	0	0	0	0	0	0	<i>no error</i>	<i>ništa</i>	<i>5</i>
									0	0	0	0	0	0	♣	♣	0	0	0	≠0	1	<i>triple</i>	<i>sledeća</i>	<i>6</i>
									0	0	0	♣	0	0	♣	0	0	0	0	≠0	0	<i>double</i>	<i>sledeća</i>	<i>7</i>
									0	0	0	0	0	0	0	0	0	0	0	≠0	1	<i>single</i>	<i>koriguj</i>	<i>8</i>
									0	0	0	0	0	0	0	0	0	0	0	0	0	<i>no error</i>	<i>ništa</i>	<i>9</i>
									0	0	0	♣	0	0	0	0	0	0	0	≠0	0	<i>double</i>	<i>sledeća</i>	<i>10</i>
									0	0	0	0	0	0	0	0	0	0	0	0	1	<i>parity</i>	<i>ništa</i>	<i>11</i>

Priprema za naredni korak podrazumeva da se prethodno prikazana matrica 11×11 transponuje, a zatim opet proširi sa horizontalnim *Hamming*-ovim bitovima, koji su već korišćeni u Korak_1, iz matrice 21×11 . Sa tako pripremljenom matricom ulazi se u Korak_3 u okviru koga se ponovo primenjuje horizontalno otkrivanje i ispravljanje grešaka.

Korak_3

																		S	P	Tip greške	Akcija	# vrste	
				0	0	0	0	0	0	0	0	0	0	0	0	0	0			no err	ništa	1	
				0	0	0	0	0	0	0	0	0	0	0	0	0	0			no err	ništa	2	
				0	0	0	0	0	0	0	0	0	0	0	0	0	0			no err	ništa	3	
				0	0	0	0	0	0	0	0	0	0	■	0	0	■	0			double	pogrešan	4
				■	0	0	0	0	0	0	0	0	0	0	0	0	0			no err	ništa	5	
				0	0	0	0	0	0	0	0	0	0	0	0	0	0			no err	ništa	6	
				0	0	0	0	0	0	0	0	0	■	■	0	0	0			double	pogrešan	7	
				0	0	0	0	0	0	0	0	0	■	0	0	0	0			single	koriguj	8	
				0	■	0	0	0	0	0	0	0	0	0	0	0	0			no err	ništa	9	
				0	0	0	0	0	0	0	0	0	0	0	0	0	0			no err	ništa	10	
				0	0	0	0	0	0	0	0	0	0	0	0	0	0			no err	ništa	11	

Nakon Korak_3 zaključujemo da u okviru paketa imamo dva vektora koji su pogrešno primljeni, dakle primenom predloženog koda nismo uspeli da izvršimo korekciju svih grešaka u paketu. U hibridnoj varijanti metode prenosa uputili bi zahtev za ponovnim slanjem. Treba napomenuti da u ovom primeru stopa pogrešno prenetih bitova izuzetno visoka i samo nam je poslužila da ilustrujemo algoritam primene koda.

																		S	P	Tip greške	Akcija	# vrste
				0	0	0	0	0	0	0	0	0	0	0	0	0						1
				0	0	0	0	0	0	0	0	0	0	0	0	0						2
				0	0	0	0	0	0	0	0	0	0	0	0	0						3
				0	0	0	0	0	0	0	0	■	0	0	■	0					pogrešan	4
				0	0	0	0	0	0	0	0	0	0	0	0	0						5
				0	0	0	0	0	0	0	0	0	0	0	0	0						6
				0	0	0	0	0	0	0	0	■	■	0	0	0					pogrešan	7
				0	0	0	0	0	0	0	0	0	0	0	0	0						8
				0	0	0	0	0	0	0	0	0	0	0	0	0						9
				0	0	0	0	0	0	0	0	0	0	0	0	0						10
				0	0	0	0	0	0	0	0	0	0	0	0	0						11

18 PRILOG B

Skraćenice

<i>ARQ</i>	<i>Automatic Repeat Request – Zahtev za ponovnim slanjem</i>
<i>AP</i>	<i>Access Point – Pristupna tačka</i>
<i>AWGN</i>	<i>Additive White Gaussian Noise – Beli Gausov šum</i>
<i>ASIC</i>	<i>Application Specific Integrated Circuit – Aplikaciono specifična integrisana kola</i>
<i>ACK</i>	<i>Acknowledgement – Paket potvrde</i>
<i>BER</i>	<i>Bit Error Rate – Stopa pogrešno prenetih bitova</i>
<i>BCH</i>	<i>Bose-Chaudhuri-Hocquenghem – BCH kod</i>
<i>BFSK</i>	<i>Binary Frequency Shift Key – Binarna frekventna digitalna modulacija</i>
<i>BPSK</i>	<i>Binary Phase Shift Key – Binarna fazna digitalna modulacija</i>
<i>BSC</i>	<i>Binary Symmetric Channel – Binarno simetričan model kanala</i>
<i>Burst</i>	<i>Greška rafalnog tipa</i>
<i>CRC</i>	<i>Cycle Redundancy Check – Cilična redundansa provere</i>
<i>CTS</i>	<i>Clear To Send – Dozvola slanja</i>
<i>Comm</i>	<i>Communication – Komunikacija</i>
<i>DLL</i>	<i>Data-Link Layer – Nivo veze</i>
<i>DATA</i>	<i>Data – polje podataka u okviru koji se prenosi</i>
<i>Dproc</i>	<i>Data processing – Procesiranje podataka</i>
<i>DEB</i>	<i>Decoding Energy per Bit – Energija dekodiranja po bitu</i>
<i>ECCs</i>	<i>Error Correcting Codes – Kodovi za korekciju (ispravljanje) grešaka</i>
<i>EC</i>	<i>Energy Consumption – Metrika potrošnje energije</i>
<i>ES</i>	<i>Energy Source – Izvor energije</i>
<i>FEC</i>	<i>Forward Error Correction – Korekcija (ispravljanje) grešaka na određistu</i>
<i>FCS</i>	<i>Frame Check Sequence – Sekvenca za proveru ispravnosti okvira</i>
<i>GMSK</i>	<i>Gaussian Minimum Shift Key – GMSK digitalna modulacija</i>
<i>GFSK</i>	<i>Gaussian Frequency Shift Key – GFSK digitalna modulacija</i>
<i>HARQ</i>	<i>Hybrid-ARQ – Hibridna ARQ metoda razmene podataka</i>
<i>In</i>	<i>Initialization – Proces inicijalizacije senzorskog čvora</i>

<i>LDPC</i>	<i>Low Density Parity Check – LDPC kod</i>
<i>MCU</i>	<i>Microcontroller – Mikrokontroler</i>
<i>MAC</i>	<i>Media Access Control – Upravljanje pristupom medijumu za prenos</i>
<i>M-RPLL</i>	<i>Modified-RPLL – Modifikovani RPLL</i>
<i>O-RPLL</i>	<i>Ordinary- RPLL – Standardni RPLL</i>
<i>PCCC</i>	<i>Parallel Concatenated Convolutional Code – PCCC kod</i>
<i>PER</i>	<i>Packet Error Rate – Stopa pogrešno prenetih paketa</i>
<i>PDR</i>	<i>Packet Delivery Ratio – Stopa uspešno dostavljenih paketa</i>
<i>PHY</i>	<i>PHYSical layer – Fizički nivo</i>
<i>PSU</i>	<i>Power Supply Unit – Jedinica napajanja</i>
<i>RPLL</i>	<i>Rendezvous Protocol for Long Life - Rendezvous protokola za dugotrajni život</i>
<i>RF</i>	<i>Radio Frequency – Radio frekventni</i>
<i>RSSI</i>	<i>Received Signal Strength Indication – Indikacija jačine primljenog signala</i>
<i>RTS</i>	<i>Request To Send – Zahtev za slanjem</i>
<i>RS</i>	<i>Radio Subsystem – Radio podsistem</i>
<i>CRC_RP</i>	<i>Rendezvous Protocol with CRC encoding scheme (O-RPLL)</i>
<i>2D SEC-DED_RP</i>	<i>Rendezvous Protocol with 2D SEC-DED encoding scheme (M-RPLL)</i>
<i>SN</i>	<i>Sensor Node – Senzorski čvor</i>
<i>Sens</i>	<i>Sensing - Senzovanje</i>
<i>SPI</i>	<i>Serial Peripheral Interface – Serijski periferalni interfejs</i>
<i>SINR</i>	<i>Signal to Interference plus Noise Ratio – Odnos signal šuma sa interferencijom</i>
<i>SS</i>	<i>Sensor Subsystem – Senzorski podsistem</i>
<i>TDMA</i>	<i>Time Division Multiple Access – Višestruki pristup medijumu na bazi vremena</i>
<i>2D- SEC-DED</i>	<i>Two Dimensional Single Error Correction and Double Error Detection</i>
<i>UD</i>	<i>User Data – Korisnički podaci</i>
<i>WSNs</i>	<i>Wireless Sensor Networks – Bežična senzorska mreža</i>

BIOGRAFIJA AUTORA

Kandidat Goran S. Nikolić rođen je 14. maja. 1973 godine u Leskovcu. Stalno nastanjen u Nišu. Oženjen i ima jedno dete. Kandidat Goran S. Nikolić završio je Osnovnu školu "Radovan Kovačević" u Lebanu i Gimnaziju "Stojan Ljubić" u Lebanu. Za uspehe postignute u osnovnoj školi i gimnaziji dobitnik je diplome "Vuk Karadžić". Diplomirao je na Elektronskom fakultetu Univerziteta u Nišu, smer Elektronika i telekomunikacije, 2003. godine sa prosečnom ocenom 8.16 (osam i 16/100) i ocenom 10 (deset) na diplomskom ispitu sa temom "**Projektovanje Data Logger-a**" čime je stekao zvanje Diplomirani inženjer elektrotehnike za elektroniku i telekomunikacije. Školske 2003/2004 upisao je poslediplomske studije na smeru za elektroniku, naučna oblast: Projektovanje i inženjering mikrosistema. Iste je završio odbranom magistarske teze pod nazivom "**Jedan pristup projektovanju umreženih *embedded* sistema velike sigurnosti**" 2010. godine čime je stekao zvanje Magistar tehničkih nauka. Školske 2010/2011 upisao je doktorske studije na smeru za Elektroniku. Naučno-stručno veće za tehničko-tehnološke nauke Univerziteta u Nišu na sednici održanoj 13.02.2017. godine dalo je saglasnost na Odluku o usvajanju teme doktorske disertacije mr Gorana Nikolića, koja je usvojena dana 29.12.2016. godine na Elektronskom fakultetu u Nišu, pod naslovom "**Visoko-pouzdan prenos podataka kod bežičnih senzorskih mreža sa malom potrošnjom energije primenom 2D-SEC-DED tehnike kodiranja**".

IZJAVA O AUTORSTVU

Izjavljujem da je doktorska disertacija, pod naslovom

"Visoko-pouzdan prenos podataka kod bežičnih senzorskih mreža sa malom potrošnjom energije primenom 2D-SEC-DED tehnike kodiranja"

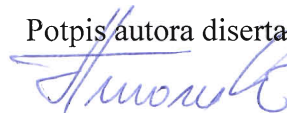
koja je odbranjena na Elektronskom fakultetu Univerziteta u Nišu:

- rezultat sopstvenog istraživačkog rada;
- da ovu disertaciju, ni u celini, niti u delovima, nisam prijavljivao na drugim fakultetima, niti univerzitetima;
- da nisam povredio autorska prava, niti zloupotrebio intelektualnu svojinu drugih lica.

Dozvoljavam da se objave moji lični podaci, koji su u vezi sa autorstvom i dobijanjem akademskog zvanja doktora nauka, kao što su ime i prezime, godina i mesto rođenja i datum odbrane rada, i to u katalogu Biblioteke, Digitalnom repozitorijumu Univerziteta u Nišu, kao i u publikacijama Univerziteta u Nišu.

U Nišu, 22.05.2019

Potpis autora disertacije:



Goran S. Nikolić

IZJAVA O ISTOVETNOSTI ELEKTRONSKOG I ŠTAMPANOG OBLIKA DOKTORSKE
DISERTACIJE

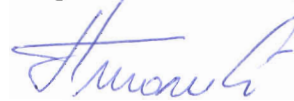
Naslov disertacije:

**"Visoko-pouzdan prenos podataka kod bežičnih senzorskih mreža sa malom potrošnjom
energije primenom 2D-SEC-DED tehnike kodiranja"**

Izjavljujem da je elektronski oblik moje doktorske disertacije, koju sam predao za unošenje u
Digitalni repozitorijum Univerziteta u Nišu, istovetan štampanom obliku.

U Nišu, 22.05.2019

Potpis autora disertacije:



Goran S. Nikolić

IZJAVA O KORIŠĆENJU

Ovlašćujem Univerzitetsku biblioteku „Nikola Tesla“ da u Digitalni repozitorijum Univerziteta u Nišu unese moju doktorsku disertaciju, pod naslovom:

"Visoko-pouzdan prenos podataka kod bežičnih senzorskih mreža sa malom potrošnjom energije primenom 2D-SEC-DED tehnike kodiranja"

Disertaciju sa svim priložima predao sam u elektronskom obliku, pogodnom za trajno arhiviranje.

Moju doktorsku disertaciju, unetu u Digitalni repozitorijum Univerziteta u Nišu, mogu koristiti svi koji poštuju odredbe sadržane u odabranom tipu licence Kreativne zajednice (Creative Commons), za koju sam se odlučio.

1. Autorstvo (CC BY)
2. Autorstvo – nekomercijalno (CC BY-NC)
3. Autorstvo – nekomercijalno – bez prerade (CC BY-NC-ND)
4. Autorstvo – nekomercijalno – deliti pod istim uslovima (CC BY-NC-SA)
5. Autorstvo – bez prerade (CC BY-ND)
6. Autorstvo – deliti pod istim uslovima (CC BY-SA)

U Nišu, 22.05.2019

Potpis autora disertacije:



Goran S. Nikolić