Digitalni zapis podataka

Stefan Mišković

7 Otkrivanje i korekcija grešaka

U računaru prilikom prenosa podataka, bilo da se radi o dva udaljena uređaja ili unutar jednog računarskog sistema, može doći do pojave grešaka. Greške se mogu pojaviti na prenosnom putu ili na lokacijama gde se nalaze otpremni i prijemni uređaji. One se manifestuju tako što se jedan bit ili niz susednih bitova poremeti (umesto 0 na određenom mestu stoji 1, ili obratno). Postoje dva pristupa za otkrivanje grešaka:

- Kontrola greške unazad, kod koje se samo može ustanoviti da li je došlo do greške ili ne. Ukoliko jeste došlo do greške, poruka se šalje ponovo. Ovde se uz bitove poruke šalju i određeni dodatni bitovi, koji omogućavaju primaocu da otkrije prisustvo greške. Sa povećanjem poruke, oni i dalje ne zauzimaju mnogo memorijskog prostora. Zbog toga su najčešće pogodni pri komunikaciji između dva udaljena uređaja.

- Kontrola greške unapred, kod koje se može ustanoviti da li je došlo do greške i izvršiti korekcija. Ukoliko jeste došlo do greške, korekcija se vrši na određenom bitu ili bloku bitova. I ovde se uz bitove poruke šalju dodatni bitovi, koji omogućavaju primaocu ne samo da otkrije prisustvo greške, već i mesto na kome se ona nalazi. Sa povećanjem poruke memorijski prostor koji oni zauzimaju brzo povećava. Zbog toga su najčešće pogodni pri komunikaciji unutar jednog računarskog sistema.

Izbor algoritma za otkrivanje grešaka zavisi od tipa greške i broja bitova sa greškom. Tip greške može biti pojedinačan ili proširen. U prvom slučaju se radi o grešci na jednom bitu, a u drugom o grešci na nizu uzastopnih bitova. Broj bitova sa greškom (eng. bits error rate – BER) predstavlja verovatnoću da jedan bit u definisanom vremenskom intervalu ima grešku. Na primer, ako je BER = 10^{-6}, to znači da u proseku 1 od 10^6 bitova ima grešku u datom intervalu vremena.

U nastavku će biti pomenuta tri pristupa za otkrivanje grešaka – kontrola parnosti, kontrola zbira bloka i ciklička provera redundanci. Biće opisan i jedan metod za otkrivanje i korekciju grešaka – Hamingov SEC kod.

7.1 Kontrola parnosti

Kod kontrole parnosti se uz poruku šalje i dodatan bit koji ima vrednost 1 ako je broj jedinica u poruci neparan, a vrednost 0 ako je broj jedinica paran. Na primer, pri slanju poruke 101101 pošiljalac će nadovezati i bit 0, budući da je broj jedinica paran. Poruka koja se šalje je oblika 1011010. Ako nije došlo do greške, primalac će tu poruku i primiti. Neka je, na primer, došlo do greške na prvom bitu. Poruka koju dobija primalac je tada 0011010. Na osnovu dela 001101 primalac zaključuje da ima neparan broj jedinica, a

kodiran bit je 0, čime konstatuje da je došlo do greške. Na ovaj način se može ustanoviti da li je došlo do greške ako je promenjen neparan broj bitova. Ali ako dva bita promene vrednosti, parnost će ostati ista, pa se u tim slučajevima ne može ustanoviti greška.

7.2 Kontrola zbira bloka

Kod kontrole zbira bloka se uz poruku šalje i dodatan broj (zapisan u binarnom sistemu na odgovarajući broj bitova) koji predstavlja broj jedinica u poruci. Na primer, pri slanju poruke 101101 pošiljalac će nadovezati i broj 4, budući da u njoj postoje 4 jedinice. Poruka koja se šalje je oblika 101101|4. Ako nije došlo do greške, primalac će tu poruku i primiti. Neka je, na primer, došlo do greške na trećem i četvrtom bitu. Poruka koju dobija primalac je tada 100001|4. Na osnovu dela 100001 primalac zaključuje da postoje dve jedinice, a kodiran broj je 4, čime konstatuje da je došlo do greške. Na ovaj način se može ustanoviti da li je došlo do greške ako je promenjen broj bitova tako da broj jedinica ne ostaje isti. Ali ako dva bita 0 i 1 promene vrednosti ili se izvrši zamena dva susedna bita, broj jedinica će ostati isti, pa se u takvim situacijama ne može ustanoviti greška.

7.3 Ciklička provera redundanci

Kod cikličke provere redundanci (alternativni naziv je CRC metoda) pošiljalac primaocu šalje kodiranu poruku i polinom generator (koji će u nastavku biti objašnjen). Primalac na osnovu kodirane poruke i oblika polinoma generatora može ustanoviti da li je došlo do greške. Ukoliko nije došlo do greške, može se iz kodirane poruke izdvojiti početna poruka, a ako je došlo do greške, primalac javlja pošiljaocu da poruku ponovo pošalje.

Neka je poruka koju pošiljalac treba da pošalje primaocu 11100110. Pošiljalac sam bira polinom generator $G(x)$, pri čemu je to bilo koji polinom čiji koeficijenti mogu da budu samo 0 ili 1. Neka je odabran polinom $G(x) = x^4 + x^3 + 1$. Budući da su koeficijenti ovakvog polinoma binarne cifre, svaki polinom generator može da ima svoj binarni zapis koji se dobija izlistavanjem koeficijenata počev od najvišeg stepena. U našem slučaju se dobija da je

$$G(x) = x^4 + x^3 + 1 = 1 \cdot x^4 + 1 \cdot x^3 + 0 \cdot x^2 + 0 \cdot x^1 + 1 \cdot x^0,$$

pa je binarni zapis polinoma $G(x)$ dat sa 11001. Na poruku koja se šalje je potrebno dodati onoliko nula koliki je stepen polinoma generatora (u ovom slučaju 4), a zatim izvršiti deljenje tog broja i binarnog ekvivalenta. Pritom je deljenje ta dva binarna broja nešto jednostavnije od klasičnog deljenja, budući da se umesto oduzimanja u svakom koraku vrši ekskluzivna disjunkcija, i samim tim nema ni pozajmica. U našem slučaju se na početnu poruku 11100110 nadovezuje niska 0000, a dobijeni binarni zapis se deli sa 11001. U osnovi, ovde se početna poruka takođe može zameniti polinomom, a nadovezivanje k nula (gde je k stepen polinoma $G(x)$) predstavlja množenje te poruke sa x^k. Deljenje sa binarnim zapisom polinoma je zapravo deljenje sa polinomom $G(x)$. Međutim, u našem slučaju će sve biti zapisano binarnim brojevima, jer je tako zapis znatno jednostavniji. Kod ovog deljenja nije od značaja količnik, već samo ostatak. Postupak deljenja je sledeći:

```
111001100000 / 11001
11001
  10111
  11001
   11100
   11001
     10100
     11001
      11010
      11001
        110
```

Poruka koja se šalje primaocu je početna poruka na koju je nadovezan ostatak. Ostatak je potrebno dopuniti vodećim nulama, tako da bude zapisan na onoliko mesta koliki je stepen polinoma. Dakle, ostatak postaje 0110, a poruka koja se šalje je 111001100110. Uz takvu kodiranu poruku treba poslati i polinom generator $G(x) = x^4 + x^3 + 1$.

Razmotrimo sada kroz primer i obrnut scenario. Neka je do primaoca stigla kodirana poruka 1100101101 i polinom generator $G(x) = x^2 + 1$ koji je izabrao pošiljalac. Primalac najpre ustanovljava da je binarni zapis polinoma $G(x)$ dat sa 101, budući da je $G(x) = 1 \cdot x^2 + 0 \cdot x^1 + 1 \cdot x^0$. Primalac sada vrši deljenje kodirane poruke i binarnog zapisa polinoma, na analogan način kao u prethodnom slučaju. U osnovi tog deljenja se vrši deljenje dva polinoma – kodirane poruke koja se može predstaviti polinomom sa koeficijentima 0 i 1 i polinoma generatora $G(x)$. Postupak deljenja izgleda ovako:

```
1100101101 / 101
101
 110
 101
  111
  101
   100
   101
     111
     101
      100
      101
        11
```

Ako je ostatak 0, poruka je uspešno primljena, a inače to nije slučaj. Ovde je ostatak 11, pa poruka nije uspešno primljena. Primalac tada sugeriše pošiljaocu da poruku ponovo pošalje. Da poruka jeste uspešno primljena, bilo bi potrebno odrediti njen polazni oblik. Ako je stepen polinoma generatora k, polazni oblik poruke se dobija odbacivanjem njenih poslednjih k bitova.

7.4 Hamingov SEC kod

Hamingov SEC kod (SEC kod je skraćenica od engleskog termina single error detection code), za razliku od prethodnih algoritama, može da otkrije postojanje greške i da izvrši korekciju na bitu na kome se greška pojavila. Algoritam će biti objašnjen na primeru

konkretne poruke koja ima 12 bitova. Svaka poruka koja će biti razmatrana će u našem slučaju imati isto toliko bitova.

Pretpostavimo da je kodirana poruka koju je pošiljalac poslao primaocu 101001100110. Ovo znači da prvih 8 bitova predstavlja bitove poruke, a poslednja 4 kontrolne bitove. Primalac će sada na osnovu bitova poruke generisati svoje kontrolne bitove i uporediti ih sa kontrolnim bitovima koju mu je prosledio pošiljalac. Na osnovu toga će zaključiti da li je došlo do greške na nekom bitu, i ako jeste odraditi potrebne korekcije. Neka su bitovi poruke označeni sa $m_8, \ldots, m_1$, a kontrolni bitovi sa $c_4, \ldots, c_1$ na sledeći način:

m_8	m_7	m_6	m_5	m_4	m_3	m_2	m_1	c_4	c_3	c_2	c_1
1	0	1	0	0	1	1	0	0	1	1	0

Sada je potrebno formirati tablicu Hamingovih SEC kodova. U prvoj koloni su dekadni brojevi od 12 do 1, u drugoj koloni odgovarajući binarni brojevi zapisani pomoću 4 bita, a u trećoj se svakom broju (od 12 do 1) dodeljuje neki od bitova $m_8, \ldots, m_1, c_4, \ldots, c_1$. Bitovi $c_4, \ldots, c_1$ se dodeljuju onim brojevima koji su stepeni dvojke (brojevima 1, 2, 4 i 8), a na preostala mesta se dodaju bitovi $m_8, \ldots, m_1$ na način prikazan u sledećoj tabeli:

12	1100	m_8
11	1011	m_7
10	1010	m_6
9	1001	m_5
8	1000	c_4
7	0111	m_4
6	0110	m_3
5	0101	m_2
4	0100	c_3
3	0011	m_1
2	0010	c_2
1	0001	c_1

Zatim se izračunavaju kontrolni bitovi $c'_4, \ldots, c'_1$ preko bitova $m_8, \ldots, m_1$ tako što se redom u četvrtoj, trećoj, drugoj i prvoj koloni svih binarnih zapisa iz tabele u odgovarajuću formulu ubace oni bitovi m_i čija je vrednost 1 i izvrši se njihova ekskluzivna disjunkcija. Drugim rečima, važi:

$$
\begin{aligned}
c'_4 &= m_5 \oplus m_6 \oplus m_7 \oplus m_8 = 0 \oplus 1 \oplus 0 \oplus 1 = 0,\\
c'_3 &= m_2 \oplus m_3 \oplus m_4 \oplus m_8 = 1 \oplus 1 \oplus 0 \oplus 1 = 1,\\
c'_2 &= m_1 \oplus m_3 \oplus m_4 \oplus m_6 \oplus m_7 = 0 \oplus 1 \oplus 0 \oplus 1 \oplus 0 = 0,\\
c'_1 &= m_1 \oplus m_2 \oplus m_4 \oplus m_5 \oplus m_7 = 0 \oplus 1 \oplus 0 \oplus 0 \oplus 0 = 1.
\end{aligned}
$$

Konačno, izračunava se vrednost $c_4c_3c_2c_1 \oplus c'_4c'_3c'_2c'_1 = 0110 \oplus 0101 = 0011$. U tabeli, u redu gde je broj 0011, nalazi se bit m_1. To znači da je došlo do greške baš na tom bitu i da je tu vrednost bita potrebno invertovati. Zbog toga je ispravna poruka 10100111. Ako je u pravcu odgovarajućeg broja u tabeli neka od vrednosti $c_4, \ldots, c_1$ ili ako je rezultat poslednje ekskluzivne disjunkcije broj koji se ne nalazi u tabeli (dekadna vrednost mu je manja od 1 ili veća od 12), ne dolazi do greške.