Měření provedli: Jakub Jíra, Jan Trávníček, Julie Trollerová
Protokol vypracoval: Jakub Jíra
Datum: 7.3.2019

Metalické vedení

Pomůcky:

Počítač se softwarem Matlab

Teorie:

Cyclic redundancy check (CRC) je postup, kterým se dají zkontrolovat data během komunikace. Tedy zda od odesílatele přišla na počítač v pořádku. K tomuto využívá binární polynom, na jehož koeficienty převede data, tedy vznikne datový polynom M(x). Tento polynom M(x) se vydělí polynomem G(x), což je v celé síti stejný generující polynom. Zbytek po tomto dělení, polynom R(x), připojíme za zprávu. V počítači příjemce proběhne stejná kontrola, a pokud je zbytek stejný, algoritmus považuje tato data za správná.

Vypracování:

K vypracování laboratorní práce jsme využili model v programu Matlab simulující síť, do které se dá vložit případný Error vector E(x).

Jako generující polynom jsme používali polynom CRC-3 → $x^3 + x + 1 = 1011$. Hlavně z důvodu praktičnosti při manuálních výpočtech. Hammingova vzdálenost *d* je pro něj 3, tedy dvě nejbližší kódová slova se od sebe liší 3 bity.

1. *Je-li chybový vektor posunem generujícího polynomu (násobení obecnou mocninou), chyba není detekována*

Ano, to jsme potrvdili. Jako příklad jsme si vybrali vektor 10110, což jsme nastavili v softwaru jako Error vector. Vzhledem k tomu, že vektor 10110 je násobkem generujícího polynomu 1011, tak simulator očekávaně žádnou chybu nezachytil

2. *Obecněji, je-li chybový vektor beze zbytku dělitelný generujícím polynomem, chyba není detekována.*

Ano, toto plyne již ze základního vzorce:

$$M(x) = Q(x) \cdot G(x) + R(x)$$

Tedy z toho, pokud je Error vector násobkem generujícího vektoru, tak vyplývá:

$$M(x) + E(x) = (Q(x) + k) \cdot G(x) + R(x)$$

Kde

$$E(x) = k \cdot G(x), \qquad k \in N$$

Takováto chyba je považována CRC algoritmem za správný polynom, protože zbytek vyšel stejný. Stejné je to i u předchozího příkladu.

3. *Každá násobná chyba je detekována, pokud má generující polynom koefcient 1 u absolutního členu, zároveň má alespoň jeden další člen*

Ano, potvrdili jsme to pomocí polynomu $x + 1$ - CRC-1, protože polynom odhalil každou jednonásobnou chybu. Hammingova vzdálenost kódových slov je totiž 2, protože přidáváme na konec jeden paritní bit.

4. *Pokud je generující polynom beze zbytku dělitelný polynomem $x + 1$, pak detekuje jakýkoliv lichý počet chyb*

Ano, využili jsme toho, že polynom $x^3 + x^2 + x + 1$, odpovídající vektoru 1111, je dělitelný polynomem $x + 1$, odpovídající polynomu 0011, beze zbytku. Tento polynom detekuje pouze lichý počet chyb. Nedetekuje však žádnou dvojnásobnou chybu. Obecně se tyto polynomy moc nevyužívají, neboť detekují právě jen lichý výskyt chyb.

5. *Pokud $x^i + 1$ není beze zbytku dělitelný generujícím polynomem pro všechna $i \in \langle 1, n - 1 \rangle$, kde n je délka kódového slova, pak jsou detekovány všechny dvojnásobné chyby*

Zde jsme si pro potvrzení vybrali polynom $x^7 + 1$. Když jsme tedy tento polynom nastavili, tak algoritmus detekoval všechny dvounásobné chyby.

6. *Spolehlivost*

Testovali jsme polynom $x^3 + x + 1$

Pravděpodobnost chyby: 0.0001	detekováno 1119/1126 chyb – úspěšnost 99.38%
Pravděpodobnost chyby: 0.01	detekováno 10553/11155 chyb – $úspěšnost$ 94.60%

Závěr:

CRC algoritmus je obecně využívaný díky své jednoduchosti a jednoduché implementaci. Je ale velice zavislý na volbě generujícího polynomu, který může mít různé vlastnosti. Proto existuje mnoho polynomu, které se používají v různých odvětvích. Používají se i polynomy s vyšším stupněm, které mají větší Hammingovu vzdálenost. Zároveň ale musíme uvažovat nad tím, že zpráva musí obsahovat bity „navíc" a je důležité s tím při přenosu počítat.

CRC je užitečné hlavně při náhodném rušení, ovšem před cílenými útoky bohužel nechrání.