Computer Science Fundamentals

Channel Coding – CRC Codes

Technische Hochschule Rosenheim
Winter 2021/22
Prof. Dr. Jochen Schmidt

Cyclic Redundancy Check (CRC)

- Linear, cyclic block codes
  - cyclic: circular shifts of a code word result in a valid code word

- Goals:
  - Detection of
    - Single- and double-bit errors
    - Burst errors (several erroneous bits in a row)
  - easy implementation (especially in hardware)

- Used, e.g.
  - Ethernet, USB, Bluetooth, SCSI, Serial ATA, ISDN, DECT (cordless phones), CAN, FlexRay (Automotive)
  - …

CRC – Idea

- Attach a k bit CRC code to an n bit long message
- Interpret message as coefficients of a dyadic polynomial
  - Dyadic = calculate modulo 2 (thus, coefficients can only assume values 0 and 1)
  - CRC is based on polynomial division

- Example
  - Message: 10011010
  - Polynomial N(x) =

$$1 \cdot x^7 + 0 \cdot x^6 + 0 \cdot x^5 + 1 \cdot x^4 + 1 \cdot x^3 + 0 \cdot x^2 + 1 \cdot x + 0 \qquad = x^7 + x^4 + x^3 + x$$

CRC – Idea

- Choose a generator polynomial $C(x)$ of degree k (k = length of attached CRC code)
- Transmit a polynomial $S(x)$
  - derived from $N(x)$
  - such that $S(x)$ is divisible by $C(x)$ without remainder

- Example $k = 3$
  - $C(x) = x^3 + x^2 + 1$
  - transmit: $S(x) = N(x) + k$ Bit

CRC – Sender

Technische Hochschule Rosenheim

Steps

- $T(x) = N(x) \cdot x^k$ $\longrightarrow$ append k zeros to message
- Calculate remainder R(x) from division T(x) / C(x) $\longrightarrow$ T(x) mod C(x)
- Send S(x) = T(x) – R(x)
  - as we have mod 2 $\longrightarrow$ T(x) – R(x) = T(x) + R(x)
  - i.e.: append R(x) to N(x)

Example

- N(x) = 10011010 = $x^7 + x^4 + x^3 + x$
- C(x) = 1101 = $x^3 + x^2 + 1$ $\longrightarrow$ k = 3
- T(x) = 10011010000 = $x^{10} + x^7 + x^6 + x^4$
- R(x) = 101 = $x^2 + 1$
- S(x) = 10011010101 = $x^{10} + x^7 + x^6 + x^4 + x^2 + 1$

CRC – Polynomial Division

- All calculations are mod 2
- Therefore, we have $1 + 1 = 1 - 1 = 0$
- Subtraction can be done by bitwise XOR of coefficients
- Always start with the leftmost coefficient of the message N(x) (or rather its extension T(x))

CRC – Polynomial Division – Example (Sender)

Technische Hochschule **Rosenheim**

$C(x) =$	$x^3 + x^2 + 1$	$= 1101$	Generator
$N(x) =$	$x^7 + x^4 + x^3 + x$	$= 10011010$	Message

T(x) = Message with k zeros appended

```
10011010000
1101          XOR
-----
 1001
 1101
 -----
  1000
  1101
  -----
   1011
   1101
   -----
    1100
    1101
    -----
       1000
       1101
       -----
        101
```

Generator

Remainder T(x) mod C(x)

Transmit message and attached remainder (k bits):

$S(x) = 10011010101$

CRC – Receiver

Steps

- Received polynomial S‘(x)
- Calculate remainder R‘(x) of division S‘(x) / C(x) ⟶ S‘(x) mod C(x)
  - remainder = 0
    - error-free transmission
    - or undetectable error
  - remainder ≠ 0
    - at least 1 bit in message is incorrect
    - message must be re-sent
      Note: error-correction is in principle possible (depending on generator), but rarely used with CRC

CRC – Example (Receiver, error-free)

Technische Hochschule Rosenheim

$C(x) =$	$x^3 + x^2 + 1$	= 1101	Generator
$S'(x) =$	$x^{10} + x^7 + x^6 + x^4 + x^2 + 1$	= 10011010101	Received message

```
 10011010101    <----------  S'(x) = received message incl. CRC
 1101
 ----        XOR
  1001
  1101
  ----
   1000
   1101
   ----
    1011
    1101
    ----
     1100
     1101
     ----
       1101
       1101
       ----
          0
```

Generator

Remainder S'(x) mod C(x)

Result:

- Remainder = 0
- CRC check ok

CRC – Example (Receiver, with errors)

$C(x) =$	$x^3 + x^2 + 1$	= 1101	Generator
$S'(x) =$	$x^{10} + x^7 + x^5 + x^4 + x^2 + 1$	= 10010110101	Received message

```
10010110101     <-  S'(x) = received message incl. CRC
1101
----
 1000
 1101
 ----
  1011
  1101
  ----
   1101
   1101
   ----
      00101
```

Generator

double error

Remainder S'(x) mod C(x)

Result:

- Remainder = 101
- CRC check not ok

CRC – Detectable Errors

- received polynomial $S'(x) = S(x) + F(x)$
  - $F(x)$ is a polynomial that represents the erroneous bits
  - $F(x) = 0 \longrightarrow$ no errors
- all errors can be detected where $F(x)$ is not a multiple of $C(x)$
  $\longrightarrow$ Requirements for generators $C(x)$

- Which errors can be detected?
  - all single-bit errors, if x^k and the constant term 1 exist
  - all double errors, if $C(x)$ has at least three terms, and the size of the data is smaller than the cycle length of $C(x)$
  - all r-bit errors for odd r, if $C(x)$ has an even number of terms; especially if it contains the factor $(x + 1)$
  - all burst errors of length smaller k, if $C(x)$ contains the constant term
  - most burst errors of length $\geq k$

CRC – Some Common Generator Polynomials

Name	Usage	Polynomial
CRC-1	Parity bit	$x + 1$
CRC-4-CCITT	Telecommunication = (15,11) Hamming	$x^4 + x + 1$
CRC-5-USB	USB	$x^5 + x^2 + 1$
CRC-5-Bluetooth	Bluetooth	$x^5 + x^4 + x^2 + 1 = (x^4 + x + 1)(x + 1)$
CRC-8-ITU-T	ISDN	$x^8 + x^2 + x + 1 = (x^7 + x^6 + x^5 + x^4 + x^3 + x^2 + 1)(x + 1)$
CRC-15-CAN	CAN bus	$x^{15} + x^{14} + x^{10} + x^8 + x^7 + x^4 + x^3 + 1= (x^7 + x^3 + x^2 + x + 1) (x^7 + x^3 + 1)(x + 1)$
CRC-32	Ethernet, Serial ATA, …	$x^{32} + x^{26} + x^{23} + x^{22} + x^{16} + x^{12} + x^{11} + x^{10} + x^8 + x^7 + x^5 + x^4 + x^2 + x + 1$

Exercise

For protection during transmission, we want to use a CRC code.

The (binary) **message** to be sent is:

$$1100\ 0110$$

As **generator polynomial** we use:

$$x^6 + x + 1$$

What is the message to be sent, including the attached CRC code?