B A B VI

DETEKSI DAN KOREKSI ERROR

Bahasan ini berhubungan dengan algoritma bagi komunikasi yang reliabel dan efisien antara dua mesin yang berdekatan, yaitu dua mesin yang secara fisik terhubung oleh sebuah saluran komunikasi yang secara konseptual bekerja seperti halnya kabel. Sifat penting sebuah saluran yang membuatnya menyerupai kabel adalah bit-bit diteruskan dalam urutan yang sama dengan
sewaktu bit-bit itu dikirimkan.Rangkaian komunikasi sering membuat kesalahan, memiliki laju data yang terbatas, dan terdapat delay propagasi yang tidak nol antara saat bit dikirimkan dengan saat bit diterima. Keterbatasn ini mempunyai implikasi penting bagi efisiensi pemindahan data.

5.1. MASALAH-MASALAH RANCANGAN DATA LINK LAYER

Data link layer memiliki beberapa fungsi spesifik. Fungsi-fungsi ini meliputi penyediaan interface layanan-layanan baik bagi network layer, penentuan cara pengelompokan bit dari physical layer ke dalam frame, hal-hal yang berkaitan dengan error transmisi, dan pengaturan aliran frame sehingga receiver yang lambat tidak akan terbanjiri oleh pengirim yang cepat.

5.2. LAYANAN YANG DISEDIAKAN BAGI NETWORK LAYER

Fungsi data link layer adalah menyediakan layanan bagi network layer. layanannya yang penting adalah pemindahan data dari network layer di mesin sumber ke network layer di mesin yang dituju. Tugas data link adalah mentransmisikan bit-bit ke mesin yang dituju, sehingga bit-bit tersebut dapat diserahkan ke network layer.

Tiga layanan dari Data Link Layer :
1. Layanan Unacknowledged Connec-tion Less
2. Layanan Acknowledged Connection-Less
3. Layanan Acknowledged Connection-Oriented

1. Layanan Unacknowledged Connectionless

Yaitu dimana mesin sumber mengirimkan sejumlah frame ke mesin yang dituju dengan tidak memberikan acknowledgment bagi diterimanya frame-frame tersebut. Tidak ada koneksi yang dibuat baik sebelum atau sesudah dikirimkannya frame. Bila sebuah frame hilang sehubungan dengan adanya noise, maka tidak ada usaha untuk memperbaiki masalah tersebu di data link layer. Jenis layanan ini cocok bila laju error sangat rendah, sehingga recovery bisa dilakukan oleh layer yang lebih tinggi. Layanan ini sesuai untuk lalu lintas real time, seperti percakapan, dimana data yang terlambat dianggap lebih buruk dibanding data yang buruk. Sebagian besar LAN menggunakan layanan unacknowledgment connectionless pada data link layer.

2. Layanan Acknowledged Connectionless

Layanan inipun tidak menggunakan koneksi, akan tetapi setiap frame dikirimkan secara independent dan secara acknowledgment. Dalam hal ini, si pengirim akan mengetahui apakah frame yang dikirimkan ke mesin tujuan telah diterima dengan baik atau tidak. Bila ternyata belum tiba pada interval waktu yang telah ditentukan, maka frame akan dikirimkan kembali, mungkin saja hilangnya acknowledgment akan

menyebabkan sebuah frame perlu dikirimkan beberapa kali dan akan diterima beberapa kali juga. Layanan ini akan bermanfaat untuk saluran unreliablem, seperti sistem tanpa kabel.

3. Layanan Acknowledged Connection Oriented

Dengan layanan ini, mesin sumber dan tujuan membuat koneksi sebelum memindahkan datanya. Setiap frame yang dikirim tentu saja diterima. Selain itu, layanan ini menjamin bahwa setiap frame yang diterima benar-benar hanya sekali dan semua frame diterima dalam urutan yang benar.Layanan ini juga menyediakan proses-proses network layer dengan ekivalen aliran bit reliabel.Pada layanan connection-oriented dipakai, pemindahan data mengalami tiga fase (tahap). Fase I koneksi ditentukan dengan membuat kedua mesin menginisialisasi variabel-variabel dan counter yang diperlukan untuk mengawasi frame yang mana yang telah diterima dan mana yang belum. Fase II, satu frame atau lebih mulai ditransmisikan. Fase III koneksi dilepaskna, pembebasan variabel, buffer, dan resource lainnya yang dipakai untuk menjaga berlangsungnya koneksi.

Karena jarak dan peralatan, pengiriman informasi, dapat mengalami perubahan atau melemah. Umumnya interferensi listrik. Kesalahan timbul dalam bentuk burst yaitu lebih dari satu bit terganggu dalam satu satuan waktu.Deteksi error dengan Redundansi, yaitu data tambahan yang tidak ada hubungannya dengan isi informasi yang dikirimkan, berupa bit pariti.Berfungsi menunjukkan ada tidaknya kesalahan data. Yaitu dengan mendeteksi dan mengoreksi kesalahan yang terjadi. Makin banyak redundansi makin baik deteksi errornya. Akibatnya makin rendah troughput dari data yang berguna.
Troughput adalah perbandingan antara data yang berguna dengan data keseluruhan. Banyaknya tambahan pada redundansi sampai 100% dari jumlah bit data.

5.3. Ada dua pendekatan untuk deteksi kesalahan :

1. Forward Error Control

Dimana setiap karakter yang ditransmisikan atau frame berisi informasi tambahan (redundant) sehingga bila penerima tidak hanya dapat mendeteksi dimana error terjadi, tetapi juga menjelaskan dimana aliran bit yang diterima error.

2. Feedback (backward) Error Control

Dimana setiap karakter atau frame memilki informasi yang cukup untuk memperbolehkan penerima mendeteksi bila menemukan kesalahan tetapi tidak lokasinya. Sebuah transmisi kontro digunakan untuk meminta pengiriman ulang, menyalin informasi yang dikirimkan.

Feedback error control dibagi menjadi 2 bagian, yaitu :

1. Teknik yang digunakan untuk deteksi kesalahan
2. Kontrol algoritma yang telah disediakan untuk mengontrol transmisi ulang.

Metode Deteksi Kesalahan :

1. *Echo*

Metode sederhana dengan sistem interaktif .Operator memasukkan data melalui terminal dan mengirimkan ke komputer. Komputer akan menampilkan kembali ke terminal, sehingga dapat memeriksa apakah data yang dikirimkan dengan benar.

2. *Error Otomatis*

Metode dengan tambahan bit pariti.

Terdapat 2 cara :

a. Pariti Ganjil (Odd Parity)

Yaitu bit pariti yang ditambahkan supaya banyaknya bit "1" tiap karakter atau data ganjil.

b. Pariti Genap (Even Parity)

Yaitu bit pariti yang ditambahkan supaya banyaknya bit "1" tiap karakter atau data genap.

Tanpa memperhatikan desain dari sistem transmisi maka, maka akan terdapat error yang menghasilkan perubahan terhapat satu atau lebih dari bit didalam frame yang ditransmisikan. Beberapa kemungkinan adanya error pada pengiriman frame meliputi :

*P*b = propabilitas error bit tunggal, biasanya disebut bit-error-rate
*P*1 = probabilitas frame yang diterima tanpa adanya error
*P*2 = probabilitas frame yang diterima dengan error tidak terdeteksi
*P*3 = probabilitas frame yang diterima dengan error terdeteksi

Jika tidak ada suatu alat yang dapat dipergunakan untuk mendeteksi error, maka probabilitas error yang terderteksi (*P*3) sama dengan 0, Untuk mempercepat menetapkan probabilitas, diasumsikan bahwa probabilitas nenerapa bit yang mengalami error (*P*b) adalah tetap, dan tidak tergantung masing-masing bit., sehingga didapatkan hubungan :

$$P_1 = (1\text{-}P_b)^F$$

$$P_2 = 1 - P_1$$

dimana *F* adalah jumlah bit per frame.

Probabilitas frame yang diterima tanpa error akan berkurang apabila probabilitas dari error bit tunggal bertambah. demikian juga probabilitas frame yang diterima dengan tanpa error bit berkurang dengan bertambahnya panjang frame. maka lebih banyak bit dengan probabilitas yang tinggi dari pada yang terkena error. Tidak ada sistem telekomunikasi data yang bebas dari kesalahan transmisi data, kesalahan ini sering kali disebabkan oleh gangguan pada saluran, sistem switching, radiasi gelombang, *cross-talk*, dll.

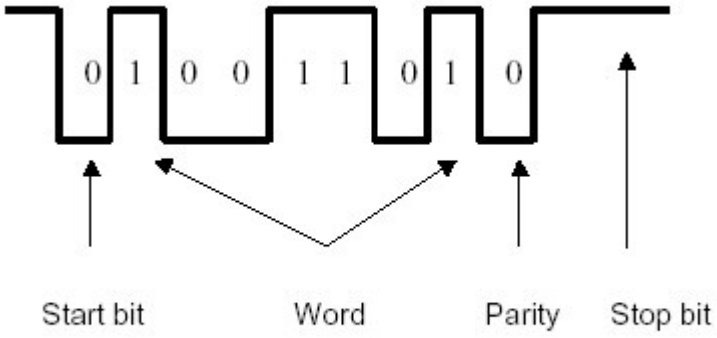
Metode deteksi kesalahan yang dikenal adalah :

- *Vertical-redundancy-checking*
- *Longitudinal-redundancy-checking*
- *Cyclic-redundancy-checking*

VERTICAL-REDUNDANCY-CHECKING

Metode ini lebih umum disebut *parity-checking* karena menggunakan sistem pengecekan paritas dan merupakan sistem untuk mencari kesalahan data yang paling sederhana. Dalam satu byte terdapat satu bit *parity*, bit ini nilainya tergantung kepada ganjil atau genapnya jumlah bit satu dalam tiap byte. *Parity-checking* dibagi menjadi dua yaitu *odd-parity* (paritas ganjil) dan *even-parity* (paritas genap). Aturan pada *odd-parity* yaitu jumlah bit satu dalam setiap byte harus ganjil. Komputer selalu mengecek *parity-bit* setiap karakter yang akan dikirim, bila jumlah bit satu dalam 7 bit pertama adalah genap, maka *parity-bit* diubah jadi 1, sebaliknya jika jumlah bit satu dalam 7 bit pertama adalah ganjil, maka *parity-bit* diubah menjadi 0. Dalam *even-parity*, jumlah bit satu dalam setiap

byte garus genap. Sebagai contoh, didalam komunikasi data digunakan sistem *odd-parity*, maka jika huruf ***A*** disusun dalam kombinasi data biner berupa “1000001, dimana jumlah bit satu dalam 7 bit pertama adalah genap, maka *parity-bit* biubah menjadi 1. Sedangkan dalam sistem *even-parity* jika huruf ***M*** disusun dalam kode biner adalah “1001101”, dimana didalam 7 bit pertama jumlah bit satu adalah genap, maka *paritybit* ini diubah menjadi 0, atau dapat dilihat pada gambar 5.1 dibawah.



Gambar 5.1 *Karakter “**M**” Dengan Even-Parity*

Sebenarnya sistem komputer mampu untuk menjalankan *parity-checking* ini, maka bila didalam saluran terjadi gangguan, maka jumlah bit satu dalam karakter yang diterima tidak sesuai, misalnya tadinya berjumlah ganjil kemudian berubah menjadi genap. Tetapi *parity-checking* ini masih mempunyai kelemahan, terutama bila jumlah bit yang rusak jumlahnya genap, maka kerusakan ini menjadi tidak dapat dideteksi. Karakter yang mengandung kesalahan 2 atau 5 bit bila hanya dilihat dari sisi genap ganjilnya jumlah bit satu, maka tidak akan kelihatan kesalahannya.

LONGITUDINAL-REDUNDANCY-CHECKING

Sistem ini sebenarnya digunakan untuk memperbaiki kelemahan yang ada pada VRC (parity-checking). Pada sistem LRC data dikirim secara per blok (frame) berisi 8 byte dan setiap frame terdapat satu parity-bit, fungsi dari bit ini sebagai kontrol seperti pada parity-checking. *Parity-bit* ini memuat 7 *parity-bit* dari byte sebelumnya, sedangkan cara untuk mengubah nilai ketujuh bit ini yaitu dengan melihat jumlah bit satu dari seluruh byte secara vertikal atau dapat dilihat pada gambar 5.2 dibawah :

Bit ke :	1	2	3	4	5	6	7	Parity
1	0	1	0	1	1	0	0	1
2	1	1	0	0	1	1	0	0
3	0	1	0	1	1	0	1	0
4	0	0	1	0	1	1	0	1
5	1	0	1	1	0	1	0	0
6	0	1	0	1	0	1	0	1
7	0	1	1	0	1	1	0	0
Parity	1	0	1	0	1	1	1	1

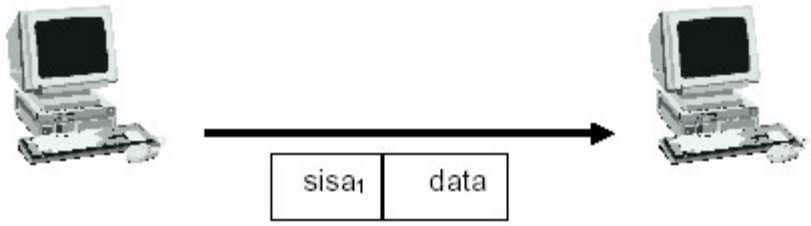
Bit ke :	1	2	3	4	5	6	7	Parity
1	0	1	0	1	1	0	0	1
2	1	1	0	0	1	1	0	0
3	0	1	0	1	1	0	1	0
4	0	0	1	0	1	0	1	1
5	1	0	1	1	0	0	1	0
6	0	1	0	1	0	1	0	1
7	0	1	1	0	1	1	0	0
Parity	1	0	1	0	1	1	1	1

Gambar 5.2 *Kelemahan LRC*

Walaupun masih memiliki beberapa kelemahan namun sistem LRC lebih baik dari VRC, sebab bila ada kesalahan yang tidak terlihat oleh *parity-bit*, maka akan diketahui oleh *parity-byte*. Dalam sistem transmisi data LRC membutuhkan banyak tambahan bit pada setiap data dikirim, misalkan untuk mengirimkan 7 karakter (59 bit) diperlukan tambahan 15 bit sebagai *parity-bit,* sehingga sistem LRC ini tidak banyak dipakai walaupun dapat bermanfaat.

CYCLIC-REDUNDANCY-CHECKING

Sistem ini banyak diterapkan dalam komunikasi data karena prosesnya cukup sederhana dan tidak banyak membutuhkan tambahan bit yang berupa *parity-bit*. Pada sistem CRC data dikirim per frame, dan setiap frame terdiri dari deretan bit yang panjang. Pada akhir blok ditambahkan beberapa *control-bit* untuk menjamin kebenaran data. *Control-bit* dibentuk oleh komputer pengirim berdasarkan perhitungan atas data yang dikirim. Setelah data sampai pada komputer penerima selanjutnya dilakukan perhitungan seperti perhitungan pada komputer pengirim. Hasil perhitungan yang didapatkan dibandingkan dengan *control-bit*, bila sama berarti data dikirim tanpa mengalami kesalahan.



data / pembagi = hasil$_1$ + sisa$_1$ data + sisa$_1$ / pembagi = hasil$_2$ + sisa$_2$

apakah hasil$_1$ = hasil$_2$
apakah sisa$_2$ = 0 ?

Gambar 5.3 *Sistem CRC*

Agar dapat mengerti lebih mendetail prosedur pada sistem CRC, maka perlu pula mengetaui proses arithmatik modulo 2 serta konsep untuk menjabarkan deretan bit sebagai polinomial aljabar. Proses arithmatika yang dilakukan pada sistem CRC seperti sistem penjumlahan bilangan tetapi tanpa menyisakan *(without-carries)* yang dapat dilakukan menggunakan gerbang logika *exclusive-OR,* seperti terlihat pada tabel kebenaran berikut ini :

Tabel Ekslusive OR

+	**0**	**1**
0	0	1
1	1	0

Pada proses arithmatik modulo 2 ini, hanya memperbolehkan menghasilkan 0 atau 1 dan tidak ada hasil negatif, pada proses pengurangan sama seperti proses penjumlahan. Selanjutnya bit-bit dari kode biner dapat diinterprestasikan sebagai polinomial koefisien. Sebagai contoh deretan biner 110101 menjadi :

$$(1)x^5 + (1)x^4 + (0)x^3 + (1)x^2 + (0)x^1 + (1)x^0$$

$$x^5 + x^4 + x^2 + 1$$

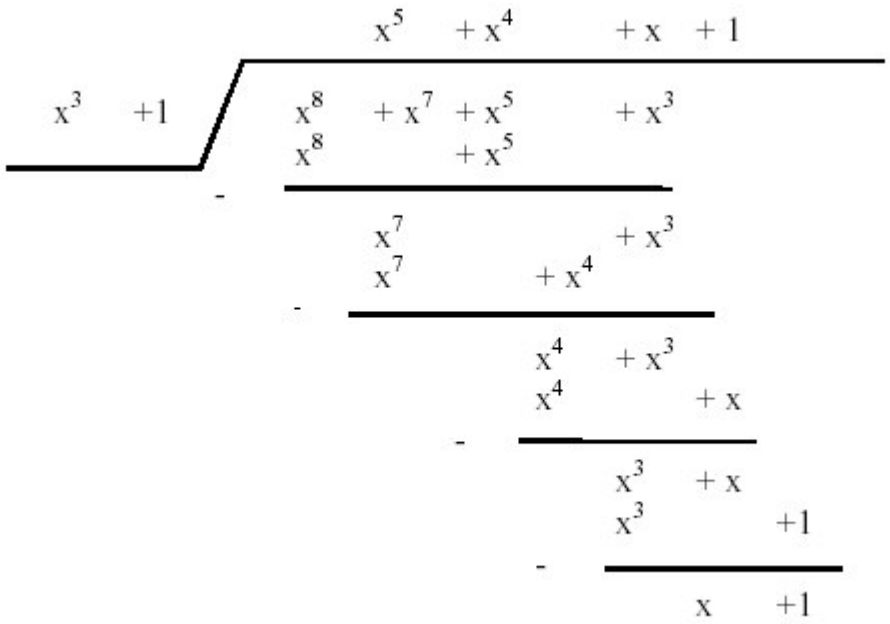
Dengan catatan bahwa untuk kode dengan *n-bit* maka pangkat tertinggi dari polinomial tersebut adalah *n-1*. Untuk melakukan proses perhitungan CRC diassumsikan memiliki sebuah pesan M(x) yang berisi deretan bit yang akan ditransmisikan, pesan tersebut berupa deretan bit 110101, sehingga M(x) = $(1)x^5 + (1)x^5 + (0)x^3 + (1)x^2 + (0)x1 + (1)x^0$. Selanjutnya ditentukan panjang kode *error-checking* G(x) yang akan dipergunakan pada protokol, misalkan kode CRC ditentukan sebagai *c-bits.* Sebagai contoh c = 3, berarti dihasilkan polinomial G(x) = $x^3 + 1$. Kemudian M(x) dikalikan dengan x^c menjadi :

$$x^3\ M(x) = x^8 + x^7 + x^5 + x^3$$

Secara analogi, hal ini sama saja dengan menggeser urutan bit pesan G(x) ke kiri sepanjang *c-bits*, yang menghasilkan urutan biner 110101000. Kemudian membagi x^3 M(x) dengan G(x) menggunakan arithmatika modulo 2, dimana akan mengasilkan hasil bagi/quotient Q(x) dan sisa pembagian/remainder R(x).

$$x^3\ M(x)/G(x) = Q(x) + R(x)$$

Sebagai contoh pembagian modulo 2 ini adalah :



Dari perhitungan diatas menghasilkan Q(x) = $x^5 + x^5 + x + 1$ dan R(x) = x +1. Selanjutnya dinambahkan sisa pembagian R(x) kepada pesan M(x) untuk menggeser polinomial dari pesan yang dikirim, kemudian akan didapatkan deretan polinomial yang dikirim T(x), sebagai berikut :

$$T(x) = x^c\ M(x) + R(x)$$

Sehingga didapatkan :

$$T(x) = x^8 + x^7 + x^5 + x^3 + x + 1$$

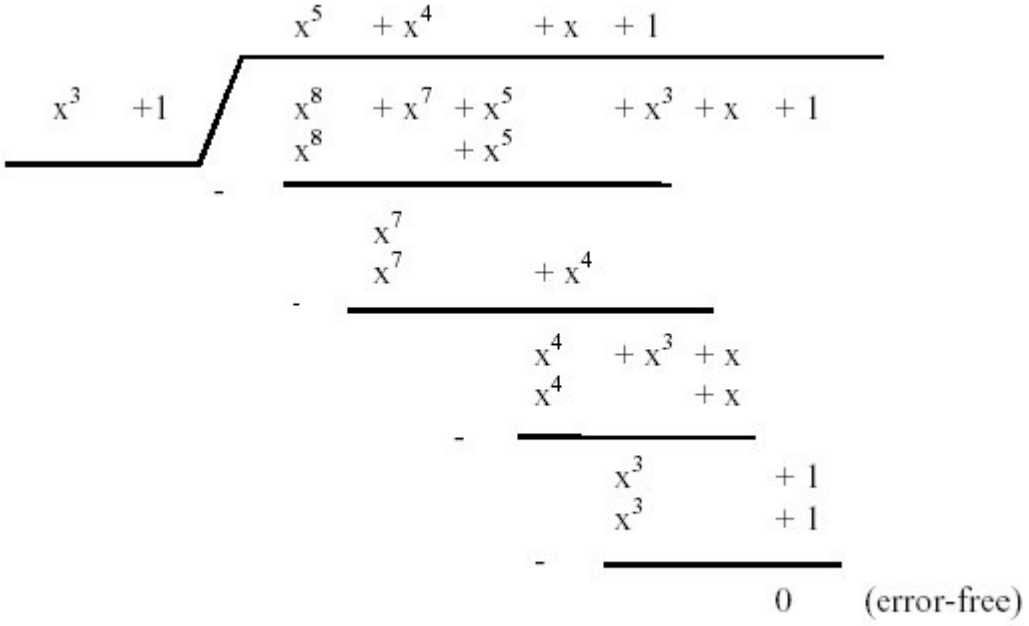
Konversi T(x) menjadi deretan kode biner yang akan ditransmisikan dapat dilihat dari susunan sebagai berikut :

$$T(x) = (1)x^8 + (1)x^7 + (0)x^6 + (1)x^5 + (0)x^4 + (1)x^3 + (0)x^2 + (1)x^1 + (1)x^0$$

$$T(x) = 110101011$$

6 bit pertama adalah pesan asli M(x), sedangkan 3 bit terakhir adalah bentuk kode *error-checking* CRC.mUntuk memeriksa error pada saat transmisi, maka receiver menerima blok data (polinomial) dengan deretan polinomial G(x) yang telah dihasilkan pada transmitter menggunakan arithmatika modulo2, pada contoh ini menggunakan $x^3 + 1$. Jika menghasilkan sisa 0 dari hasil pembagian antara T(x) dan G(x) maka dapat dikatakan pesan M(x) yang dikirim bebas dari error. Bila sisa pembagian menghasilkan selain 0 maka dapat dikatakan terjadi error.

Proses pembagian dapat dijabarkan sebagai berikut :



Sisa pembagian sama dengan 0 yang menunjukkan bebas dari error, sedangkan hasil pembagian menjadi $Q(x) = x^5 + x^5 + x + 1$, yang ternyata sama dengan hasil pembagian pada sisi transmitter. Bisa dimengerti bahwa yang ditunjukkan diatas merupakan proses yang terjadi pada transmitter dan receiver, pada receiver T(x)/G(x) selalu harus menghasilkan quotient Q(x) dengan sisa pembagian (remainder) harus 0. Untuk melengkapi arithmatik modulo 2 didapatkan persamaan :

$$T(x) = x^c M(x) + R(x)$$

dan

$$\frac{x^c M(x)}{G(x)} = Q(x) + \frac{R(x)}{G(x)}$$

$$\frac{T(x)}{Gx)} = \frac{x^c M(x)}{G(x)} + \frac{R(x)}{G)} = Q(x) + \frac{R(x)}{G(x)} + \frac{R(x)}{G(x)}$$

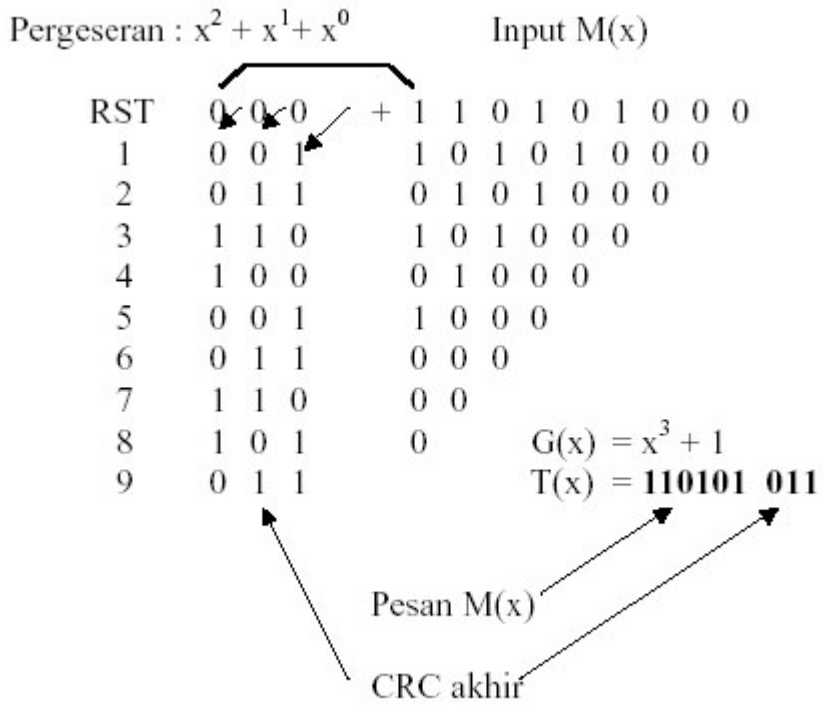
atau

$$\frac{T(x)}{G(x)} = Q(x) + (1 + 1)\left(\frac{R(x)}{G(x)}\right)$$

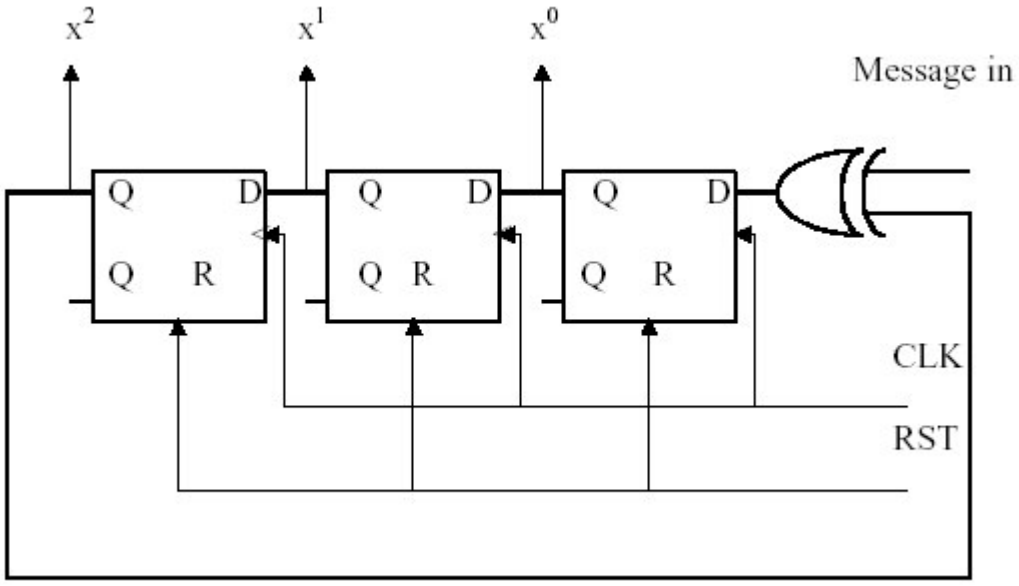
akan tetapi pada proses arithmatikan modulo 2 ini (1 + 1) = 0, maka akan didapatkan persamaan :

$$\frac{T(x)}{G(x)} = Q(x) \text{ (tanpa sisa)}$$

Untuk mengimplementasikan CRC, maka kode CRC dibangkitkan dengan menggunakan software arithmatik. Akan tetapi untuk menambah kecepatan dan penggunaan microprocessor secara efisien, umunya kode CRC ini dibangkitkan dan dilakukan pengecekan menggunakan hardware. Arithmatik modulo 2 dan operasi penggeseran dapat dilakukan menggunakan shift register yang memiliki gerbang exclusive-OR yang dihubungkan secara umpan balik. Jumlah shift register yang yang diperlukan sama dengan jumlah c, seperti besarnya pattern G(x) untuk operasi polinomial, atau jumlah bit pada kode CRC.
Pada gambar dibawah ini menujukkan rangkaian yang membangkitkan kode CRC sebesar 3-bit, seperti yang diuraikan pada contoh diatas. Pesan M(x) yang besarnya 6-bit berupa 110101 di geser pada register, seperti dijunjukkan dibawah terdapat 3 nol. Sesudah 9 kali pergeseran *register-shift-time* (RST) maka register berisi bit CRC yaitu 011.



Pada gambar 5.4 dibawah ditunjukkan suatu rangkaian yang menghasilkan kode CRC 3-bit untuk sontoh kalkulasi di atas dengan bentuk pattern : $G(x) = x^3 + 1$.



Gambar 5.4 *Pembangkit CRC Untuk $G(x) = x^3 + 1$*

Pada implementasi CRC yang digunakan untuk mendeteksian error baik secara hardware maupun software maka ada 3 versi pattern G(x) yang banyak dipakai secara luas, antara lain :

CRC - 16 : $x^{16} + x^{15} + x^2 + 1$

CRC - CCITT : $x^{16} + x^{12} + x^5 + 1$

CRC - 32 : $x^{32} + x^{26} + x^{23} + x^{16} + x^{12} + x^{11} + x^{10} + x^8 + x^7 + x^5 + x^4 + x^2 + x + 1$

Modulo 2 arithmetic

Menggunakan penjumlahan binary dengan tanpa carry, dimana hanya merupakan perasi *exclusive-OR.*

Untuk kepentingan ini didefinisikan :

T = (k + n) bit frame untuk ditransmisi, dengan n < k
M = k bit message, k bit pertama dari T
F = n bit FCS, n bit terakhir dari T
P = pattern dari n+1 bit.

Dimana :

$$T = 2^n M + F$$

$$\frac{2M}{P} = Q + \frac{R}{P}$$

Karena pembaginya adalah binary, remaider selalu kurang dari 1 bit dibanding pembagi. Maka :

$$T = 2M + R$$

atau

$$\frac{T}{P} = \frac{2M + R}{P}$$

$$\frac{T}{P} = Q + \frac{R}{P} + \frac{R}{P}$$

$$\frac{T}{P} = Q + \frac{R + R}{P} = Q$$

Contoh : 1. Diketahui : message M = 1010001101 (10 bit)
pattern P = 110101 (6 bit)
FCS R = dikalkulasi (5 bit)

2. Message M dikalikan dengan 2^5 , maka : 101000110100000
3. Kemudian dibagi dengan P :

```
                    110101011  <-- Q
P --> 110101  ⌈  101000110100000  <-- 2 M
                 110101
                  ----------
                  111011
                  110101
                  ---------
                     111010
                     110101
                     ---------
                        111110
                        110101
                        ---------
                           101100
                           110101
                           ----------
                           110010
                           110101
                           -----------
                               1110  <-- R
                               1111
```

4. Remainder (R = 01110) ditambahkan ke 2^n M untuk mendapatkan T = 10100011010110, yang ditransmisi [$T = 2^n M + R$].
5. Jika tidak ada error, maka receiver menerima T secara utuh. Frame yang diterima dibagi dengan P :

```
                    1101010110
110101  ⌈  101000110101110
           110101
           ----------
            111011
            110101
           ---------
              111010
              110101
              ---------
                 111110
                 110101
                 ---------
                    101111
                    110101
                    ---------
                     110101
                     110101
                     ---------
                           00
```

Karena tidak ada remaider maka dianggap tidak ada error.

Pattern P dipilih 1 bit lebih panjang daripada FCS, dan bit pattern dipilih tergantung tipe error yang diinginkan. Pada keadaan minimum keduanya baik tingkat high atau low bit dari P harus 1. Frame Tr yang dihasilkan dapat dinyatakan sebagai : Tr = T + E

dimana : T = frame yang ditransmisi

E = error pattern dengan 1 dalam posisi dimana terjadi error

Tr = frame yang diterima.

Receiver akan gagal untuk mendeteksi error jika dan hanya jika Tr dapat dibagi dengan P, yang jika dan hanya jika E dapat dibagi dengan P.

Polynomials

Dalam bentuk variabel x dengan koefisien-koefisien binary. Koefisien-koefisien tersebut berhubungan dengan bit-bit dalam binary sehingga proses CRC-nya dapat dijabaran sebagai :

1. $\frac{X\,M(X)}{P(X)} = Q(X) + \frac{R(X)}{P(X)}$

2. $T(X) = X\,M(X) + R(X)$

Error E(X) hanya tidak akan terdeteksi bila dapat dibagi dengan P(X). Error- error yang dapat dideteksi yang tidak dapat dibagi oleh P(X) :

1. Semua error bit tunggal.
2. Semua error bit ganda, sepanjang P(X) mempunyai faktor paling sedikit 3 syarat.
3. Jumlah error genap apapun, sepanjang P(X) mengandung faktor (X + 1).
4. Burst error apapun dengan panjang burst lebih kecil daripada panjan FCS.
5. Burst error yang paling besar.

Empat versi dari P(X) yang dipakai secara luas :

CRC-12 = $X^{12} + X^{11} + X^3 + X^2 + X + 1$, dipakai untuk transmisi dari 6 bit karakter dan membentuk 12 bit FCS.

CRC-6 = $X^{16} + X^{15} + X^2 + 1$, } umum untuk 8 bit karakter dan keduanya

CRC-CCITT = $X^{16} + X^{12} + X^5 + 1$, } menghasilkan 16 bit FCS.

CRC-32 = $X^{32} + X^{26} + X^{23} + X^{22} + X^{16} + X^{12} + X^{11} + X^{10} + X^8 + X^7 + X^5 + X^5 + X^2 + X + 1$, membentuk 32 bit FCS.

Shift registers dan gate exclusive-OR

Shift register adalah device penyimpan string 1 bit dimana terdapat sebuah line output, yang mengidikasikan nilai yang dimuat, dan sebuah line input.

Seluruh register di-clock secara simultan, yang menyebabkan 1 bit bergeser sepanjang seluruh register . Sirkuit ini dapat dipenuhi sebagai berikut :

1. Register mengandung n bits, sama dengan panjang FCS.
2. Ada lebih dari n gate exclusive-OR.
3. Keberadaan dan ketiadaan suatu gate tergantung pada keberadaan atau ketiadaan dari suatu syarat dalam polynomial pembagi, P(X).

Dalam contoh, Message M = 1010001101 ; $M(X) = X^9 + X^7 + X^3 + X^2 + 1$

Pembagi P = 110101 ; $P(X) = X^5 + X^5 + X^2 + 1$

Pada receiver, tiap bit M yang tiba, disisipi ke dalam shift register. Jika tidak ada error, shift register akan memuat bit pattern untuk R pada akhir dari M. Bit R yang ditransmisi sekarang mulai tiba dan efeknya yaitu me-nol-kan register pada akhir penerimaan, register memuat semua nol.

$P(X) = \sum_{i=0}^{n} a_i X^i$ dimana $a_0 = a_n = 1$ dan semua a yang lain sama baik 0 atau 1.

5.4. Koreksi Kesalahan Transmisi

Bila dijumpai kesalahan pada data yang telah diterima, maka perlu diadakan tindakan perbaikanatau diusahakan agar kesalahan ini jangan sampai memberikan dampak yang besar. Metode koreksi ini diantaranya adalah :

- *Subtitusi simbol*

Bila ada data yang rusak maka komputer penerima mengganti bagian itu dengan karakter lain, sepertu karakter SUB yang berupa tanda tanya terbalik. Jika pemakai menjumpai karakter ini (pada program word-prossessor), maka berarti data yang diterima telah mengalami kerusakan, selanjutnya perbaikan dilakukan sendiri.

- *Mengirim data koreksi*

Data yang dikirim harus ditambah dengan kode tertentu dan data duplikat. Bila penerima menjumpai kesalahan pada data yang diterima, maka perbaikan dilakukan dengan mengganti bagian yang rusak dengan data duplikat, tetapi cara ini jarang dilakukan.

- *Kirim ulang*

Cara ini merupakan cara yang paling simpel, yaitu bila komputer penerima menemukan kesalahan pada data yang diterima, maka selanjutnya meminta komputer pengirim untuk mengirim mengulangi pengiriman data.