

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. Keterbatasan 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 Connection-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 tersebut 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 unreliable, 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 ekuivalen 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 dilepaskan, 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 throughput dari data yang berguna. Throughput 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 memiliki informasi yang cukup untuk memperbolehkan penerima mendeteksi bila menemukan kesalahan tetapi tidak lokasinya. Sebuah transmisi kontrol 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 terhadap 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 terdeteksi (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 :

$$P_1 = (1 - P_b)^F$$

$$P_2 = 1 - P_1$$

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.

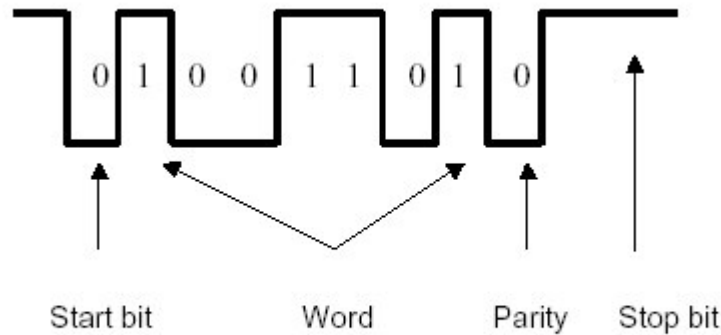
Metode deteksi kesalahan yang dikenal adalah :

- *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 garis 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* diubah 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

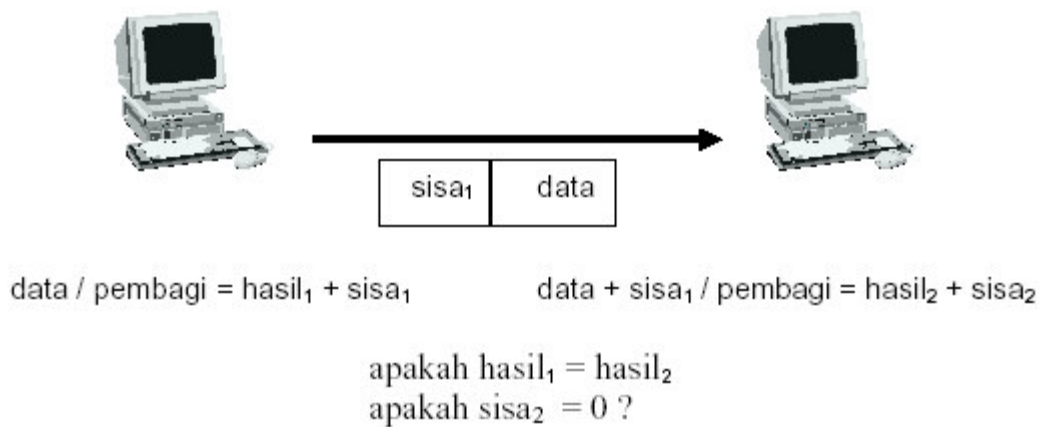
Bit ke :	1	2	3	4	5	6	7	Parity
1	0	1	0	1	1	0	0	1
2	1	1	0	0	1	1	0	0
3	0	1	0	1	1	0	1	0
4	0	0	1	0	1	0	1	1
5	1	0	1	1	0	0	1	0
6	0	1	0	1	0	1	0	1
7	0	1	1	0	1	1	0	0
Parity	1	0	1	0	1	1	1	1

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.



Gambar 5.3 Sistem CRC

Agar dapat mengerti lebih mendetail prosedur pada sistem CRC, maka perlu pula mengetahui proses aritmatik modulo 2 serta konsep untuk menjabarkan deretan bit sebagai polinomial aljabar. Proses aritmatika 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 Eksklusive OR

+	0	1
0	0	1
1	1	0

Pada proses aritmatik 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 diinterpretasikan 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 $c\text{-bits}$, yang menghasilkan urutan biner 110101000. Kemudian membagi $x^3 M(x)$ dengan $G(x)$ menggunakan arithmatika modulo 2, dimana akan menghasilkan 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 :

$$\begin{array}{r}
 \begin{array}{cc} x^3 & +1 \end{array} \overline{) \begin{array}{cccc} & x^5 & +x^4 & +x+1 \end{array}} \\
 \underline{\begin{array}{cccc} x^8 & +x^7 & +x^5 & +x^3 \\ x^8 & & +x^5 & \end{array}} \\
 \begin{array}{cccc} & x^7 & & +x^3 \\ & x^7 & & +x^4 \end{array} \\
 \underline{\hspace{1.5cm}} \\
 \begin{array}{cccc} & & x^4 & +x^3 \\ & & x^4 & +x \end{array} \\
 \underline{\hspace{1.5cm}} \\
 \begin{array}{cccc} & & & x^3+x \\ & & & x^3+1 \end{array} \\
 \underline{\hspace{1.5cm}} \\
 \begin{array}{cc} & x+1 \end{array}
 \end{array}$$

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. Untuk memeriksa error pada saat transmisi, maka receiver menerima blok data (polinomial) dengan deretan polinomial $G(x)$ yang telah dihasilkan pada transmitter menggunakan arithmatika modulo 2, 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 :

$$\begin{array}{r}
 \begin{array}{c} x^3 + 1 \end{array} \overline{) \begin{array}{c} x^5 + x^4 + x + 1 \\ x^8 + x^7 + x^5 + x^3 + x + 1 \\ \underline{x^8 + x^5} \\ x^7 + x^4 + x + 1 \\ \underline{x^7 + x^4} \\ x^4 + x^3 + x + 1 \\ \underline{x^4 + x} \\ x^3 + 1 \\ \underline{x^3 + 1} \\ 0 \end{array} \quad \text{(error-free)}
 \end{array}$$

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

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

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

atau

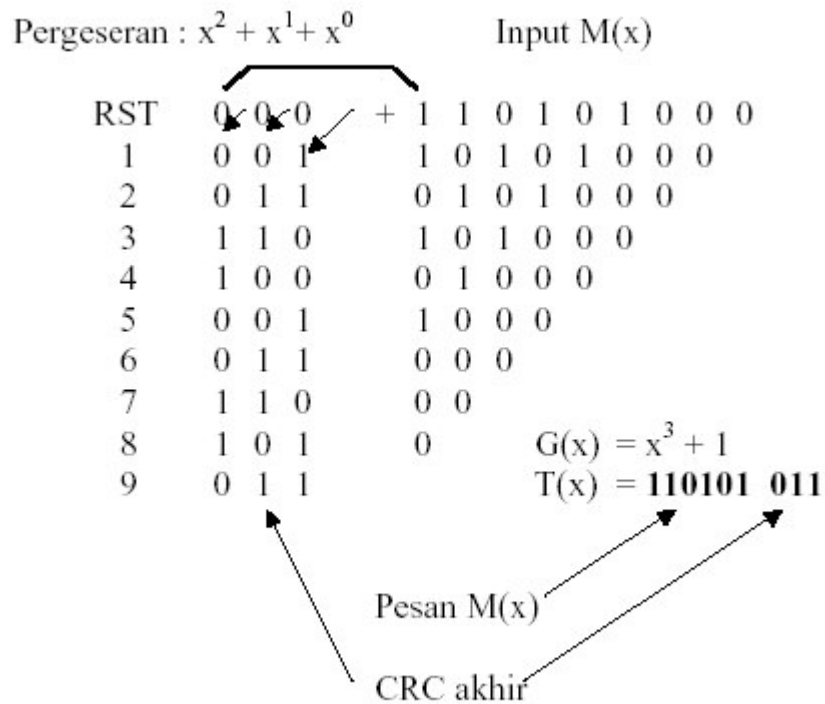
$$\frac{T(x)}{G(x)} = Q(x) + (1 + 1) \left(\frac{R(x)}{G(x)} \right)$$

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

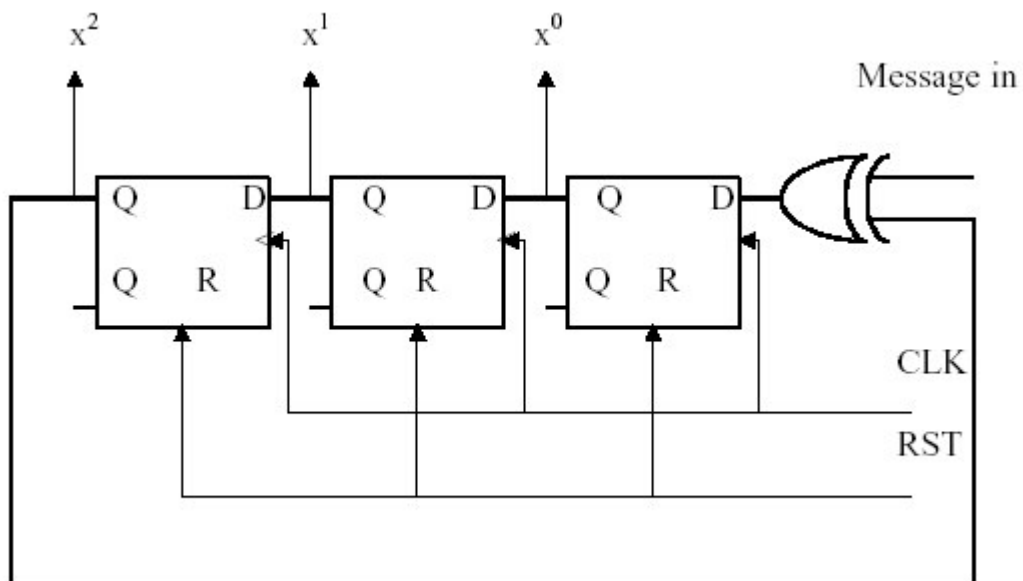
$$\frac{T(x)}{G(x)} = Q(x) \text{ (tanpa sisa)}$$

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 diperlukan sama dengan jumlah c , seperti besarnya pattern $G(x)$ untuk operasi polinomial, atau jumlah bit pada kode CRC.

Pada gambar dibawah ini menunjukkan 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 diunjukkan 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 contoh 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 mendeteksi error baik secara hardware maupun software maka ada 3 versi pattern $G(x)$ yang banyak dipakai secara luas, antara lain :

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

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

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

Modulo 2 arithmetic

Menggunakan penjumlahan binary dengan tanpa carry, dimana hanya merupakan operasi *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 pembagiannya adalah binary, remainder 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 :

$$\begin{array}{r}
 \text{P} \rightarrow 110101 \quad \overline{110101011} \leftarrow Q \\
 \left[\begin{array}{r}
 101000110100000 \leftarrow 2M \\
 110101 \\
 \hline
 111011 \\
 110101 \\
 \hline
 111010 \\
 110101 \\
 \hline
 111110 \\
 110101 \\
 \hline
 101100 \\
 110101 \\
 \hline
 110010 \\
 110101 \\
 \hline
 1110 \leftarrow R \\
 1111
 \end{array} \right.
 \end{array}$$

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 :

$$\begin{array}{r}
 110101 \quad \overline{1101010110} \\
 \left[\begin{array}{r}
 101000110101110 \\
 110101 \\
 \hline
 111011 \\
 110101 \\
 \hline
 111010 \\
 110101 \\
 \hline
 111110 \\
 110101 \\
 \hline
 101111 \\
 110101 \\
 \hline
 110101 \\
 110101 \\
 \hline
 00
 \end{array} \right.
 \end{array}$$

Karena tidak ada remainder 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 T_r yang dihasilkan dapat dinyatakan sebagai : $T_r = T + E$

dimana : T = frame yang ditransmisi

E = error pattern dengan 1 dalam posisi dimana terjadi error

T_r = frame yang diterima.

Receiver akan gagal untuk mendeteksi error jika dan hanya jika T_r 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 dijabarkan 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 panjang 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 = $X^{32} + X^{26} + X^{23} + X^{22} + X^{16} + X^{12} + X^{11} + X^{10} + X^8 + X^7 + X^5 + X^2 + X + 1$, membentuk 32 bit FCS.

Shift registers dan gate exclusive-OR

Shift register adalah device penyimpan string 1 bit dimana terdapat sebuah line output, yang mengidikasi 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 $P = 110101$; $P(X) = X^5 + X^3 + X^2 + 1$

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.

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

5.4. Koreksi Kesalahan Transmisi

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

- *Substitusi simbol*

Bila ada data yang rusak maka komputer penerima mengganti bagian itu dengan karakter lain, seperti karakter SUB yang berupa tanda tanya terbalik. Jika pemakai menjumpai karakter ini (pada program word-processor), 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.

BAB VII

MULTIPLEXING

Dua stasiun komunikasi tidak akan memakai kapasitas penuh dari suatu data link untuk efisiensi, karena itu sebaiknya kapasitasnya dibagi. Pembagian ini diistilahkan sebagai ***multiplexing***.

Contoh sederhananya yaitu multidrop line, dimana sejumlah perangkat secondary (misal : terminal) dan sebuah primary (misal : komputer host) saling berbagi pada jalur/line yang sama.

Keuntungannya :

- Komputer host hanya butuh satu port I/O untuk banyak terminal
- hanya satu line transmisi yang dibutuhkan.

Pada chapter ini dibahas 3 teknik multiplexing :

- frequency-division multiplexing (FDM), paling umum dipakai untuk radion atau TV
- time-division multiplexing (TDM) atau synchronous TDM, dipakai untuk multiplexing digital voice.
- peningkatan efisiensi synchronous TDM dengan variasi sebagai berikut :
 - Statistical TDM
 - Asynchronous TDM
 - Intelligent TDM

Gambar 6.1 menyatakan fungsi multiplexing secara umum. Multiplexer mengkombinasikan (me-multiplex) data dari n input dan mentransmisi melalui kapasitas data link yang tinggi. Demultiplexer menerima aliran data yang di-multiplex (pemisahan (demultiplex) dari data tersebut tergantung pada channel) dan mengirimnya ke line output yang diminta.



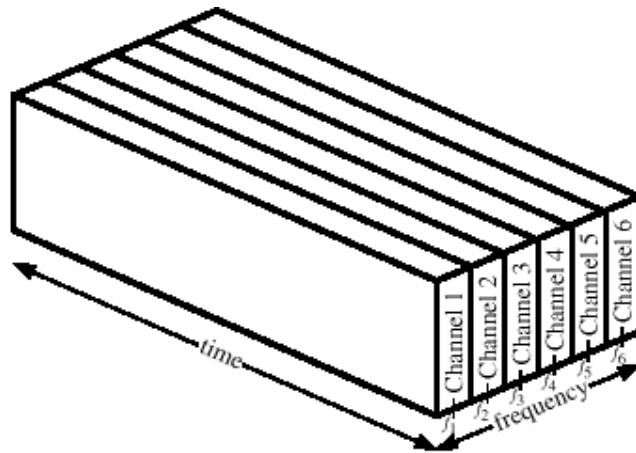
Gambar 6.1 Multiplexing

6.1 Frequency Division Multiplexing

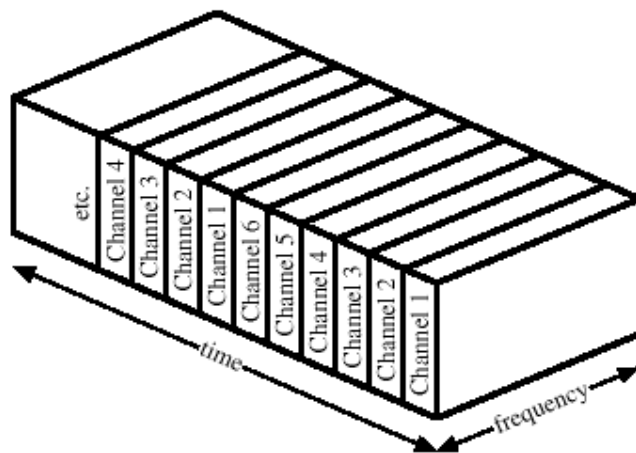
Karakteristik

- Digunakan ketika bandwidth dari medium melebihi bandwidth sinyal yang diperlukan untuk transmisi.

- Tiap sinyal dimodulasikan kedalam frekuensi carrier yang berbeda dan frekuensi carrier tersebut terpisah dimana bandwidth dari sinyal-sinyal tersebut tidak overlap.



(a) Frequency-division multiplexing



(b) Time-division multiplexing

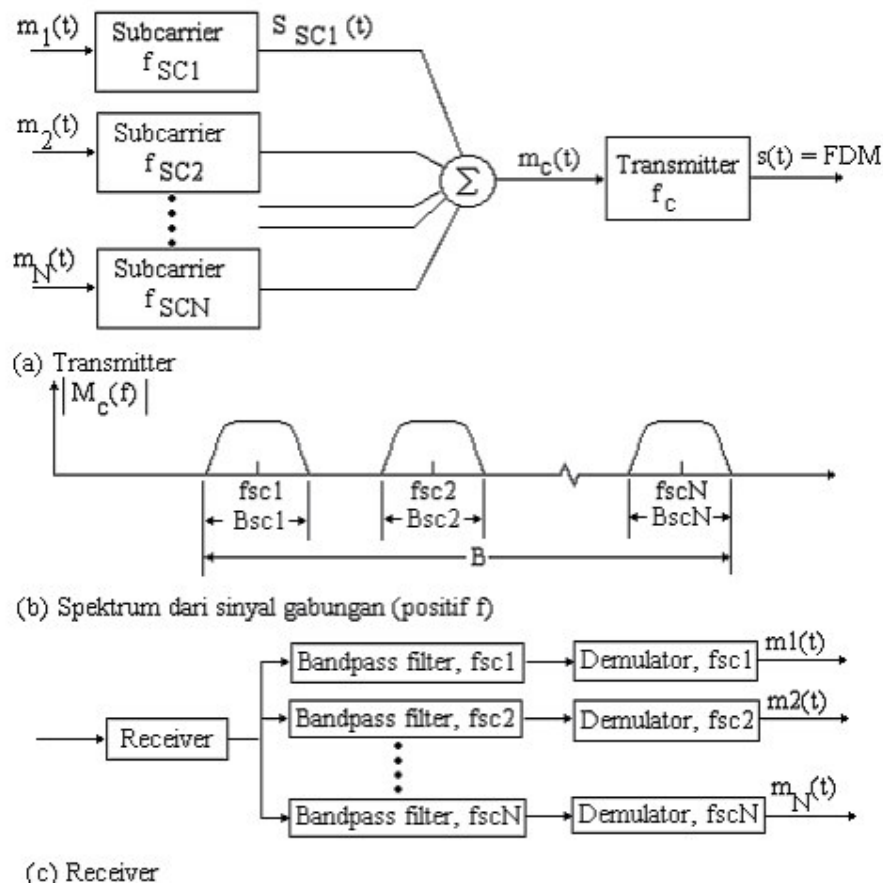
Gambar 6.2 FDM dan TDM

Gambar 6.2a menunjukkan kasus umum dari FDM. Enam sumber sinyal dimasukkan ke dalam suatu multiplexer, yang memodulasi tiap sinyal ke dalam frekuensi yang berbeda (f_1, \dots, f_6). Tiap sinyal modulasi memerlukan bandwidth center tertentu disekitar frekuensi carriernya, dinyatakan sebagai suatu **channel**.

Sinyal input baik analog maupun digital akan ditransmisikan melalui medium dengan sinyal analog.

Contoh sederhana dari FDM yaitu transmisi full-duplex FSK (Frequency Shift Keying). Contoh lainnya yaitu broadcast dan TV kabel.

Gambar 6.3 memperlihatkan transmisi sinyal TV dan bandwidth-nya.



Gambar 6.3 Frequency Division Multiplexing

Sinyal video hitam putih adalah modulasi AM pada sinyal carrier f_{cv} . Karena baseband dari sinyal video = 4 MHz maka sinyalnya sekarang menjadi $f_{cv} - 0,76$ MHz sampai dengan $f_{cv} - 4,2$ MHz. f_{cc} sebagai color subcarrier mentransmisi informasi warna. Sedangkan sinyal audio dimodulasi pada f_{ca} , diluar bandwidth efektif dari 2 sinyal lainnya. Bandwidth audio = 60 KHz. Dengan demikian sinyal TV dapat di-multiplex dengan FDM pada kabel CATV dengan bandwidth = 6 MHz.

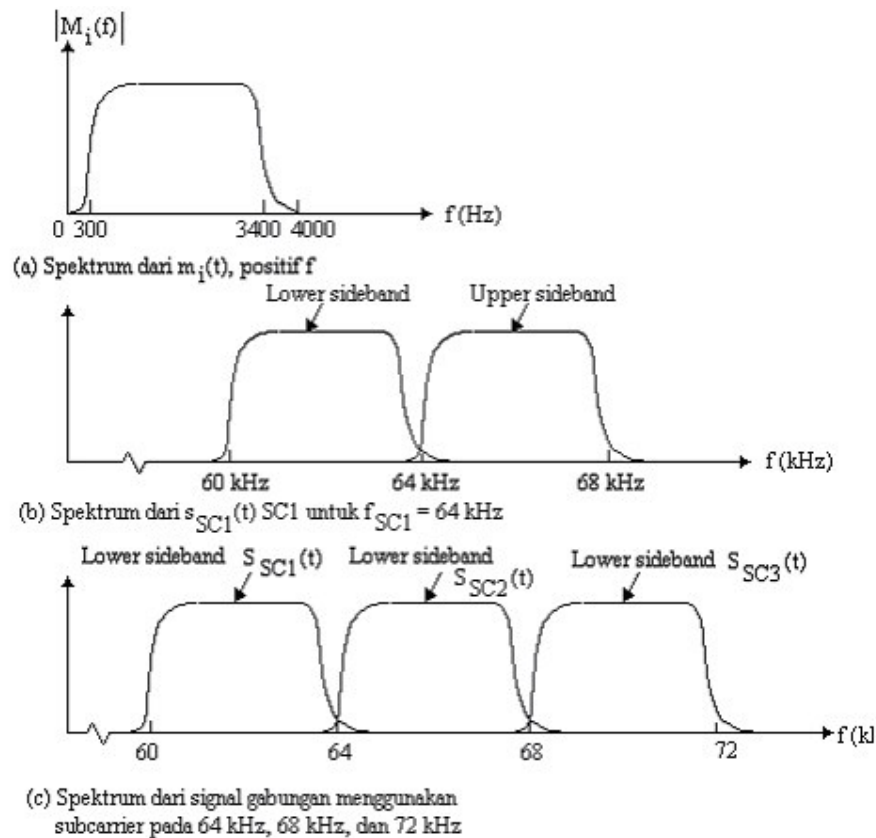
Gambar 6.3 memperlihatkan sistim FDM secara umum. Sejumlah sinyal digital atau analog $[m_i(t), i = 1, N]$ di-multiplex ke dalam medium transmisi yang sama. Tiap sinyal $m_i(t)$ dimodulasi dalam carrier f_{sci} ; karena digunakan multiple carrier maka masing-masing dinyatakan sebagai **sub carrier**. Modulasi apapun dapat dipakai. Kemudian sinyal termodulasi dijumlah untuk menghasilkan sinyal gabungan $m_c(t)$. Gambar 6.3b menunjukkan hasilnya.

Sinyal gabungan tersebut mempunyai total bandwidth B , dimana

$$B > \sum_{i=1}^N B_{si}$$

Sinyal analog ini ditransmisikan melalui medium yang sesuai. Pada akhir penerimaan, sinyal gabungan tersebut lewat melalui N bandpass filter, dimana tiap filter berpusat pada f_{sci} dan

mempunyai bandwidth B_{sci} , untuk $1 < i < N$. Dari sini, sinyal diuraikan menjadi bagian-bagian komponennya. Tiap komponen kemudian dimodulasi untuk membentuk sinyal asalnya. Contoh sederhananya : transmisi tiga sinyal voice (suara) secara simultan melalui suatu medium.



Gambar 6.4 FDM dari 3 sinyal band suara

Gambar 6.4a menggambarkan spektrum sinyal suara dari 300 sampai 3400 Hz. Bila suatu sinyal diamplitudo modulasi pada carrier 64 KHz maka gambar spektrumnya seperti gambar 6.4b. Sinyal termodulasi mempunyai bandwidth 8 KHz dari 60 sampai 68 KHz. Tetapi yang digunakan hanya lower sideband-nya sehingga didapat gambar 6.4c, dimana ketiga sinyal voice tersebut dipakai untuk memodulasi carrier pada 64, 68 dan 72 KHz. Sinyal suara ini ditransmisi melalui modem dan sudah cukup memakai bandwidth 4 KHz. Tetapi masalahnya jika melalui jarak yang jauh maka akan timbul intermodulasi noise dan efek nonlinear dari amplifier pada salah satu channel yang akan menghasilkan komponen-komponen frekuensi pada channel-channel yang lain.

Gambar 6.4 menunjukkan tiga level pertama dari definisi hierarki AT&T, dimana 12 channel voice dikombinasikan untuk menghasilkan suatu **group** sinyal dengan bandwidth $12 \times 4 \text{ KHz} = 48 \text{ KHz}$ dalam range 60 – 108 KHz. Kemudian dibentuk blok dasar berikutnya 60 channel supergroup, yang dibentuk oleh FDM lima group sinyal. Sinyal yang dihasilkan antara 312 sampai 662 KHz.

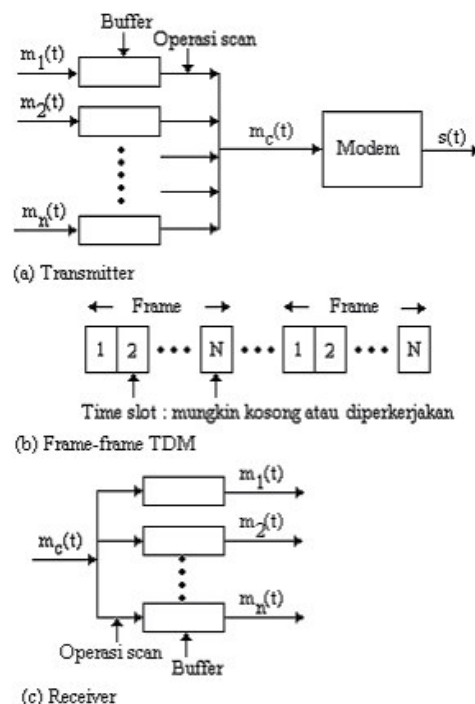
Variasi lainnya, yaitu dengan kombinasi 60 channel voice band langsung dalam suatu supergroup, dimana akan mengurangi biaya karena interface dengan group multiplex tidak diperlukan. Hierarki dari level berikutnya adalah **master group** dengan 10 supergroup input.

Catatan : suara asal atau sinyal data mungkin dimodulasi berulang kali. Tiap tingkatan dapat mengubah data asal; hal ini misalnya jika modulator/multiplexer mengandung non linearitas atau menghasilkan noise.

6.2 Synchronous Time-Division Multiplexing

Karakteristik

- Digunakan ketika data rate dari medium melampaui data rate dari sinyal digital yang ditransmisi.
- Sinyal digital yang banyak (atau sinyal analog yang membawa data digital) melewati transmisi tunggal dengan cara pembagian (=interlaving) porsi yang dapat berupa level bit atau dalam blok-blok byte atau yang lebih besar dari tiap sinyal pada suatu waktu.



Gambar 6.5 system synchronous TDM.

Gambar 6.5a, sejumlah sinyal digital ($m_i(t)$, $i = 1, N$) di-multiplex ke dalam medium transmisi yang sama. Data yang masuk dari masing-masing sumber disimpan dalam buffer yang biasanya berukuran 1 bit atau 1 karakter. Buffer tersebut di-scan secara sequential untuk membentuk komposisi aliran data digital $m_c(t)$ yang dapat ditransmisi langsung atau melalui modem, biasanya transmisi synchronous. Operasi scan tersebut berjalan cepat dimana buffer terlebih dulu dikosongkan untuk dapat menerima data. Dengan demikian data rate $m_c(t)$ harus sama dengan jumlah data rate $m_i(t)$.

- Gambar 6.5b memperlihatkan format data yang ditransmisi. Data-data tersebut dikumpulkan dalam frame-frame. Tiap frame mengandung cycle dari time slot dimana tiap slot mewakili tiap sumber data.

Channel adalah serangkaian slot-slot yang mewakili satu sumber, dari frame ke frame.

Panjang slot sama dengan panjang buffer transmitter yaitu 1 bit atau 1 karakter.

Dalam hal ini dipakai 2 teknik interleaving :

- Character-interlaving :
 - Dipakai dengan sumber asynchronous.
 - Tiap time slot mengandung 1 karakter dari data.
- Bit-interlaving :
 - Dipakai dengan sumber synchronous dan boleh juga dengan sumber asynchronous.
 - Tiap time slot mengandung hanya 1 bit.

Gambar 6.5c, pada receiver, data $m_c(t)$ di-demultiplex dan diarahkan ke buffer tujuan yang sesuai. Untuk tiap sumber input $m_i(t)$, ada sumber output identik yang akan menerima data input pada kecepatan yang sama dengan pada waktu ditimbulkan.

Synchronous TDM :

- Disebut synchronous karena time slot-time slot-nya di-alokasikan ke sumber-sumber dan tertentu dimana time slot untuk tiap sumber ditransmisi. Biar bagaimanapun sumber mempunyai data untuk dikirim.
- Dapat mengendalikan sumber-sumber dengan kecepatan yang berbeda-beda.

TDM Link Control

Mekanisme kontrolnya tidak diperlukan protokol data link maka aliran data yang ditransmisikan tidak mengandung header dan trailer.

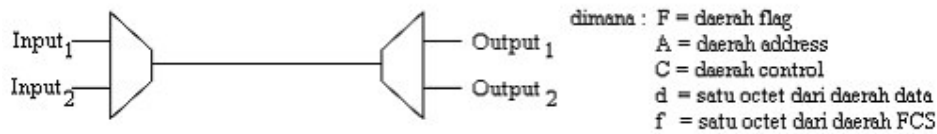
Ada 2 kunci mekanisme kontrol data link : flow control dan error control. Tetapi flow control tidak diperlukan bila multiplexer dan demultiplexer dihubungkan seperti gambar 6.1, data rate dari multiplexer tetap dan keduanya beroperasi pada kecepatan tersebut. Bila dihubungkan ke line output yang tidak dapat menerima data, maka untuk sementara, channel akan membawa slot-slot kosong, tetapi frame-frame keseluruhan akan mempertahankan kecepatan transmisi yang sama.

Untuk error control, transmisi ulang hanya dilakukan pada satu channel dimana terjadi error jadi error control ada per-channel.

Agar flow control, error control dapat dilengkapi per basis channel, dipakai protokol data link misalnya HDLC per basis channel.

Lihat gambar 6.6, dua sumber data, masing-masing memakai HDLC. Yang satu mentransmisi frame-frame HDLC yang mengandung 3 octet data, yang lain mengandung 4 octet data. Kita memakai multiplexing interleaving karakter. Maka octet-octet dari frame-frame HDLC dari 2 sumber dicampur aduk bersama untuk transmisi melalui line multiplex. Operasi multiplexing/demultiplexing adalah transparant untuk mencapai stasiun; untuk tiap pasang stasiun komunikasi, mempunyai link tersendiri.

Pada akhir kedua line perlu suatu kombinasi multiplexer/demultiplexer dengan line full duplex diantaranya. Kemudian tiap channel terdiri dari 2 set slot, satu menuju ke masing-masing arah.



Gambar 6.6 Konfigurasi Pemakaian DLC pada Chanel TDM

Framing

Frame TDM tidak memakai karakter SYNC atau flag untuk synchronisasi frame tetapi **added-digit framing**.

Pada cara ini, satu kontrol bilangan ditambahkan ke tiap frame TDM. Juga memakai pola bit identitas dari frame ke frame. Synchronisasi dilakukan dengan cara, receiver membandingkan bit-bit yang masuk dari posisi satu frame untuk memperoleh pola. Jika polanya tidak sama, posisi bit berurutan di cari sampai pola didapat. Sekali synchronisasi frame tercapai, receiver melanjutkan memonitor channel framing bit. Jika pola terputus, receiver harus masuk lagi ke mode framing search.

Pulse Stuffing (= pulsa pengisi)

Dipakai untuk mengatasi problem :

- Jika tiap sumber mempunyai clock yang terpisah, variasi antar clock-clock akan menyebabkan hilangnya synchronisasi.
- Data rate dari input data tidak bertalian dengan angka rasional sederhana.

Sehingga :

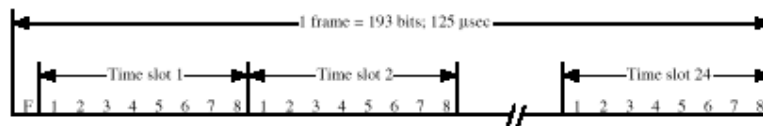
- Data rate yang keluar dari multiplexer, termasuk framing bit, lebih tinggi daripada jumlah maximum kecepatan yang masuk.
- Kapasitas ekstra dipakai oleh **stuffing extra dummy bit-bit** atau pulsa-pulsa ke dalam tiap sinyal yang masuk sampai kecepatannya naik ke clock sinyal yang dibangkitkan.
- Pulsa-pulsa stuffing dimasukkan ke lokasi yang tertentu didalam format frame multiplexer sehingga dapat dikenali dan dipindah ke demultiplexer.
- Sumber 1 : Analog, bandwidth 2 KHz
- Sumber 2 : Analog, bandwidth 4 KHz
- Sumber 3 : Analog, bandwidth 2 KHz
- Sumber 4-11 : Digital, synchronous 7200 bps

Sumber 1, 2, 3 di-scan untuk diambil sinyal PAM-nya dimana kecepatan scan 4 KHz untuk sumber 1 dan 3, untuk satu sampel PAM, dan 2 sampel PAM untuk sumber 2 per scan. Keempat sampel ini diubah ke 4 bit sampel PCM. Dengan demikian total 16 bit dihasilkan dengan kecepatan 4000 kali per detik, untuk komposisi bit rate 64 Kbps. Untuk sumber-sumber digital, **pulse stuffing** dipakai untuk menaikkan kecepatan masing-masing sumber

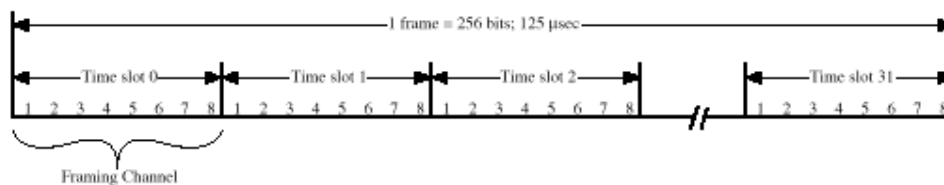
menjadi 8 Kbps, untuk kumpulan data rate 64 Kbps. Sebuah frame dapat terdiri dari beberapa cycle dari 32 bit, yang masing-masing mengandung 16 bit PCM dan 2 bit dari masing-masing ke delapan sumber digital tersebut.

Sistim-sistim Carrier

Dasar dari hierarki TDM adalah format transmisi DS-1 yang memultiplex 24 channel. Tiap frame mengandung 8 bit/channel plus framing bit untuk $24 \times 8 + 1 = 193$ bit.



(a) Interface at 1.544 Mbps



(b) Interface at 2.048 Mbps

Gambar 6.7 Format Transmisi DS 1

Untuk transmisi suara (voice), dimana bandwidth voice = 4 KHz sehingga diperlukan 8000 sampel/detik. Dengan panjang frame 193 bit, maka data rate-nya = $8000 \times 193 = 1,644$ Mbps. Untuk lima dari enam frame, dipakai 8 bit PCM. Untuk setiap bit ke enam tiap channel mengandung 7 bit PCM plus bit pensinyalan.

Untuk data digital, dipakai data rate yang sama dengan voice yaitu 1,644 Mbps. Untuk data disediakan 23 channel. Channel ke 24 disimpan untuk byte SYNC khusus yang menyebabkan lebih cepat dan framing ulang yang lebih baik untuk suatu framing error. Untuk tiap channel, 7 bit/channel dan tiap channel diulang 8000 kali/detik, maka data rate/channel = 66 Kbps. Untuk data rate yang lebih rendah dipakai teknik subrate multiplexing dimana bit tambahan diambil dari tiap channel untuk indikasi speed subrate multiplexing yang sedang dipakai sehingga kapasitas total per channel = $6 \times 8000 = 48$ Kbps.

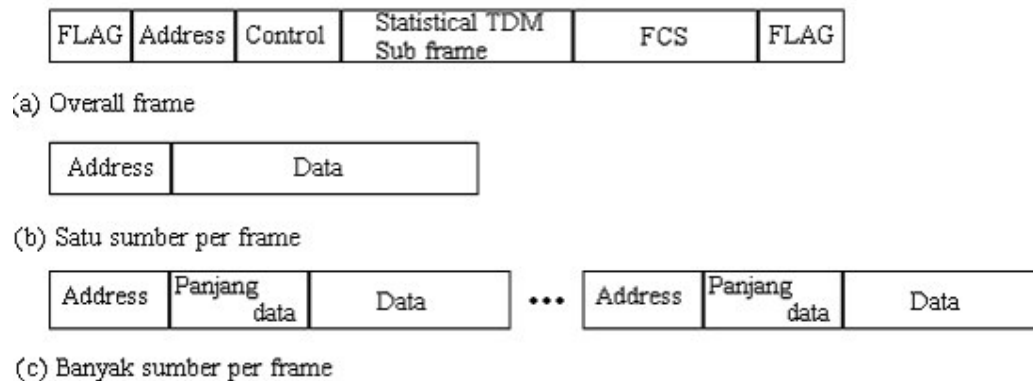
6.3 Statistical Time-Division Multiplexing

Karakteristik

- Statistical TDM yang dikenal juga sebagai asynchronous TDM dan intelligent TDM, sebagai alternative synchronous TDM.
- Mempunyai sejumlah line I/O pada satu sisi dan line multiplex kecepatan tinggi pada sisi lainnya. Dimana ada n line I/O, tetapi hanya k ($k < n$) time slot yang sesuai pada frame TDM.
- Untuk input, fungsi multiplexer ini untuk men-scan buffer-buffer input, mengumpulkan data sampai penuh, dan kemudian mengirim frame tersebut.

- Untuk output, multiplexer menerima suatu frame dan mendistribusikan slot-slot data ke buffer output tertentu.
- Data rate pada line multiplex lebih rendah daripada jumlah data rate dari device masukan sehingga statistical multiplexer dapat menggunakan data rate yang rendah untuk mendukung sebanyak device yang sama dengan synchronous multiplexer.
- Gambar 6.7 membandingkan statistical dan synchronous TDM dimana terdapat 4 sumber data dan menunjukkan produksi data dalam 4 time epoch (t_0, t_1, t_2, t_3). Untuk synchronous multiplexer, selama tiap epoch, data dikumpulkan dari keempat sumber dan dikirim. Misal pada epoch pertama, sumber C dan D tidak menghasilkan data, maka 2 dari 4 time slot yang ditransmisi oleh multiplexer adalah kosong.

Untuk statistical multiplexer, tidak mengirim slot-slot kosong jika ada data untuk dikirim sehingga selama epoch pertama, hanya slot A dan B yang dikirim. Karena data tiba dan didistribusikan ke line I/O yang tidak di perkirakan, sehingga informasi address diperlukan untuk meyakinkan pengiriman. Maka dari itu overhead atau slot untuk statistical TDM lebih banyak karena tiap slot membawa sebuah address.



Gambar 6.8 Format frame dari statistical TDM

- Struktur framenya padat.
- Sistemnya membuahi synchronous protokol seperti HDLC dimana data frame harus mengandung bit-bit kontrol untuk operasi multiplexing. Untuk (a) hanya 1 sumber data yang dimasukkan per frame. Sumber diidentifikasi oleh suatu address. Panjang daerah data adalah variabel dan diakhiri oleh akhir dari overall frame. Cara ini dapat bekerja baik dibawah beban yang ringan, tetapi kurang efisien untuk beban yang berat.

Untuk efisiensi :

- Dengan menggunakan multiple data source yang dibentuk dalam suatu frame tunggal.
- Daerah address dapat dikurangi dengan memakai pengalamatan relatif dimana tiap address menunjukkan sumber aliran relatif terhadap sumber terdahulu.
- Memakai 2 bit label untuk panjang daerah [SEID78].

Performansi

Data rate dari output statistical multiplexer lebih rendah daripada jumlah data rate input. Hal ini dimungkinkan karena rata-rata jumlah dari input kurang daripada kapasitas line multiplex. Tetapi masalah yang timbul yaitu terjadinya periode peak ketika input melampaui kapasitas.

Solusinya : dengan memasukkan suatu buffer dalam multiplexer untuk menahan sementara kelebihan input.

Contohnya dapat dilihat pada tabel 6.6 dimana untuk tiap kapasitas, ketika input melampaui output, pembentukkan back lop yang harus disimpan. Tabel ini menunjukkan sejumlah bit-bit input dari 10 device tiap milisecond dan output dari multiplexer.

Pertimbangan ukuran buffer dan data rate dari line ditentukan untuk menentukan waktu respon sistim dan kecepatan line multiplex. Semakin besar buffer, delaynya semakin panjang.

Parameter-parameter untuk statistical TDM :

N = jumlah dari sumber input

R = data rate tiap sumber, bps

M = kapasitas efektif dari line multiplex, bps = kecepatan maksimum dimana bit-bit data dapat ditransmisikan

α = waktu tengah tiap sumber yang sedang transmisi, $0 < \alpha < 1$

$k = M/(NR)$ = ratio kapasitas line multiplex terhadap total input maksimum, $\alpha \leq k \leq 1$ = ukuran kompresi oleh multiplexer

$k=1$ berhubungan dengan synchronous TDM

$k < \alpha$ input akan melampaui kapasitas multiplexer

$\lambda = \alpha N R$ = rata-rata kecepatan untuk tiba, bps

	1	
$S =$	----	dipakai untuk transmisi 1 bit, sec
	M	

$p = \lambda S = \alpha N R / M = \alpha / k = \lambda / M$ = pemakaian total kapasitas link

t_q = ukuran rata-rata delay oleh sumber input