

DATA LINK CONTROL

Pengiriman data melalui *link komunikasi data* yang terlaksana dengan penambahan kontrol layer dalam tiap device komunikasi, dinyatakan sebagai **data link control** atau **data link protocol**.

Data link adalah medium transmisi antara stasiun-stasiun ketika suatu prosedur data link control dipakai.

Keperluan-keperluan dan tujuan-tujuan untuk komunikasi data secara efektif antara dua koneksi stasiun transmisi-penerima secara langsung, untuk melihat kebutuhan bagi data link control:

- ✚ *Frame synchronization* : data dikirim dalam blok-blok yang disebut frame. Awal dan akhir tiap frame harus dapat diidentifikasi.
Memakai variasi dari konfigurasi line : lihat section 5.1.
- ✚ *Flow control* : stasiun pengirim harus tidak mengirim frame-frame pada rate/kecepatan yang lebih cepat daripada stasiun penerima yang dapat menyerapnya.
- ✚ *Error control* : bit-bit error yang dihasilkan oleh sistem transmisi harus diperbaiki.
- ✚ *Addressing* (peng-alamat-an) : pada line multipoint, identitas dari dua stasiun yang berada dalam suatu transmisi harus diketahui.
- ✚ Kontrol dan data pada link yang sama : biasanya tidak diinginkan mempunyai path komunikasi yang terpisah untuk sinyal-sinyal kontrol. Karena itu, receiver harus mampu membedakan kontrol informasi dari data yang sedang ditransmisi.
- ✚ *Link management* : permulaan, pemeliharaan dan penghentian dari pertukaran data memerlukan koordinasi dan kerjasama diantara stasiun-stasiun. Diperlukan prosedur untuk manajemen pertukaran ini.

5.1 Konfigurasi-Konfigurasi Line

Ada 3 karakteristik yang membedakan berbagai konfigurasi data link, yaitu : topology, duplexity dan line discipline (rancangan tata tertib).

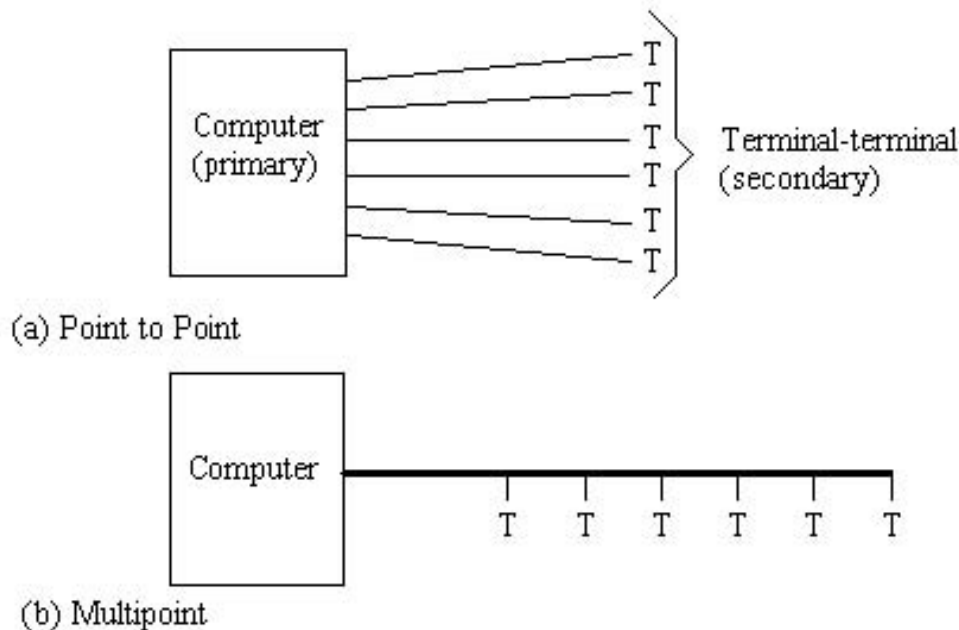
Topology dan Duplexity

Topology dari suatu data link, menyatakan pengaturan fisik dari stasiun pada suatu link.

Ada dua konfigurasi topology :

- ✚ Point to point, jika hanya ada dua stasiun.
- ✚ Multipoint, jika ada lebih dari dua stasiun. Dipakai dalam suatu komputer (stasiun utama/stasiun primary) dan suatu rangkaian terminal (stasiun sekunder/stasiun secondary).

Gambar 5.1, menunjukkan keuntungan konfigurasi multipoint, yaitu : komputer hanya perlu suatu I/O port tunggal dan juga hanya memerlukan suatu kabel transmisi tunggal sehingga menghemat biaya operasional.



Gambar 5.1. Konfigurasi Komputer/terminal tradisional.

Duplexity dari suatu link menyatakan arah dan timing dari aliran sinyal.

Jenis-jenisnya :

- ✚ *Simplex transmission*, aliran sinyal selalu dalam satu arah. Contoh : hubungan komputer dengan printer. Transmisi simplex ini jarang dipakai karena tidak mungkin untuk mengirim error atau sinyal kontrol kembali melalui link ke sumber data.
- ✚ *Half-duplex link*, dapat mentransmisi dan menerima tidak secara simultan.
- ✚ *Full-duplex link*, dua stasiun dapat mengirim dan menerima data satu terhadap yang lain secara simultan.

Pensinyalan digital, dapat memakai full-duplex dan half-duplex link. Untuk pensinyalan analog, penentuan duplexity tergantung pada frekuensi, baik penggunaan transmisi guided atau unguided, dimana bila suatu stasiun transmisi dan penerimaan pada frekuensi yang sama, berarti beroperasi dalam mode half-duplex sedangkan bila suatu stasiun mentransmisi pada suatu frekuensi dan menerima pada frekuensi yang lain maka beroperasi dalam mode full-duplex.

Line Discipline (Rancangan tata tertib)

Beberapa tata tertib diper lukan dalam penggunaan link transmisi. Pada mode half-duplex, hanya satu stasiun yang dapat mentransmisi pada suatu waktu. Baik mode half-duplex atau full-duplex, suatu stasiun hanya mentransmisi jika mengetahui bahwa receiver telah siap untuk menerima.

Point to Point Link

Bila stasiun ingin mengirim data ke stasiun yang lain, maka pertama dilakukan penyelidikan (dinyatakan sebagai **enq/enquiry**) stasiun lain untuk melihat apakah siap menerima. Stasiun kedua merespon dengan suatu positive acknowledge (ack) untuk indikasi telah siap. Stasiun pertama kemudian mengirim beberapa data, sebagai suatu frame.

Setelah beberapa data dikirim, stasiun pertama berhenti untuk menunggu hasilnya. Stasiun kedua menetapkan penerimaan data (ack) yang sukses. Stasiun pertama kemudian mengirim suatu message akhir transmisi (eot) yang menghentikan pertukaran dan mengembalikan sistem seperti semula. Bila terjadi error pada transmisi, suatu negative acknowledgment (nak) dipakai untuk mengindikasikan bahwa suatu sistem tidak siap menerima, atau data yang diterima error. Hal ini diperlihatkan sebagai garis tipis dalam gambar. Jika hal ini terjadi maka stasiun mengulang tindakan akhirnya atau mungkin memulai beberapa prosedur perbaikan error (erp). Garis tebal pada gambar memperlihatkan keadaan normal.

Ada 3 fase dalam prosedur kontrol komunikasi ini :

- ✚ *Establishment* (penentuan) : memutuskan stasiun mana yang transmisi dan mana yang menerima dan apa receiver siap untuk menerima.
- ✚ *Data transfer* : data ditransfer dalam satu atau lebih blok-blok acknowledgment.
- ✚ *Termination* : membatasi koneksi logika (hubungan transmitter-receiver).

Multipoint links

Aturan umum yang dipakai dalam situasi ini yaitu **poll** dan **select**.

- ✚ *Poll* : primary meminta data dari suatu secondary.
- ✚ *Select* : primary mempunyai data untuk dikirim dan memberitahu suatu secondary bahwa data sedang datang.

Gambar 5.3 memperlihatkan konsep ini. Dalam 5.3a, primary mem-poll suatu secondary dengan mengirim suatu message "poll". Dalam hal ini, secondary tidak punya apa -apa untuk dikirim dan merespon dengan message "nak". Timing total untuk rangkaian ini :

$$T_N = t_{prop} + t_{poll} + t_{proc} + t_{nak} + t_{prop}$$

Dimana : T_N = total waktu untuk mem-poll terminal dengan tanpa mengirim apapun.

$$t_{prop} = \text{waktu penyebaran} = t_1 - t_0 = t_5 - t_4$$

$$t_{prop} = \text{waktu untuk transmisi suatu poll} = t_2 - t_1$$

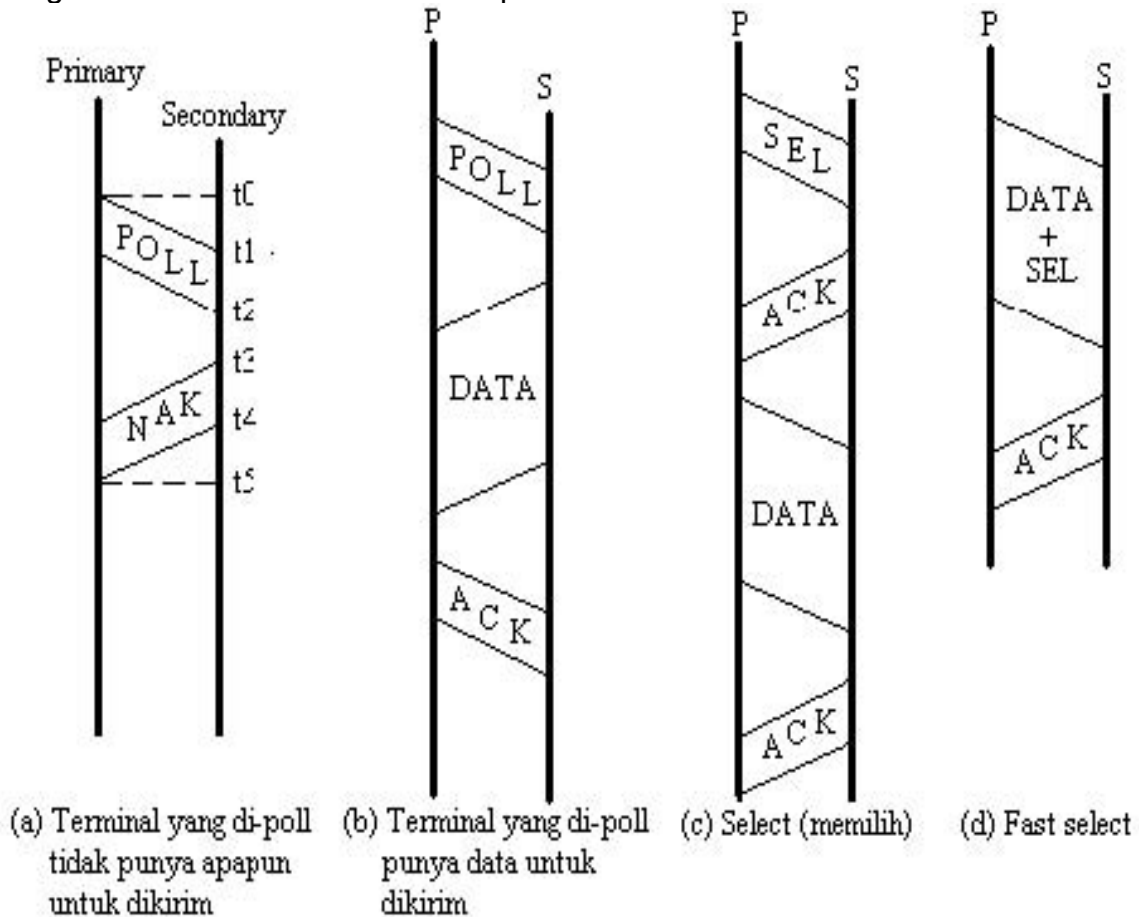
$$t_{proc} = \text{waktu untuk memproses poll sebelum acknowledgment} = t_3 - t_2$$

$$t_{nak} = \text{waktu untuk transmisi suatu negative acknowledgment} = t_4 - t_3$$

Transmisi dari primary harus menunjuk pada secondary yang dipilih; transmisi dari secondary harus menyamakan secondary tersebut.

Gambar 5.3c, dimana ditunjukkan fungsi select.

Gambar 5.3d, menunjukkan suatu teknik alternatif yaitu *fast select*, dimana message select termasuk data yang ditransfer. Teknik ini cocok untuk aplikasi-aplikasi dengan message -message pendek yang seringkali ditransmisi dan waktu transfer untuk message tersebut tidak lebih lama daripada waktu balasan.



Gambar 5.3. Serangkaian Poll dan Select

Bentuk lain dari line discipline, yaitu *contention*, dimana tidak ada primary tetapi hanya suatu kumpulan stasiun-stasiun peer keduanya baik transmitter dan receiver harus diidentifikasi. Stasiun ini dapat mentransmisi jika jalur/line sedang bebas; kalau tidak maka harus menunggu. Teknik ini dapat ditemukan dalam pemakaian secara luas pada local network dan sistem satelit.

Dalam hal ini dapat disimpulkan bahwa :

- ✚ *Point to point* : tidak perlu address.
- ✚ *Primary -secondary multipoint* : perlu satu address, untuk mengidentifikasi secondary.
- ✚ *Peer multipoint* : perlu dua address, untuk mengidentifikasi transmitter dan receiver.

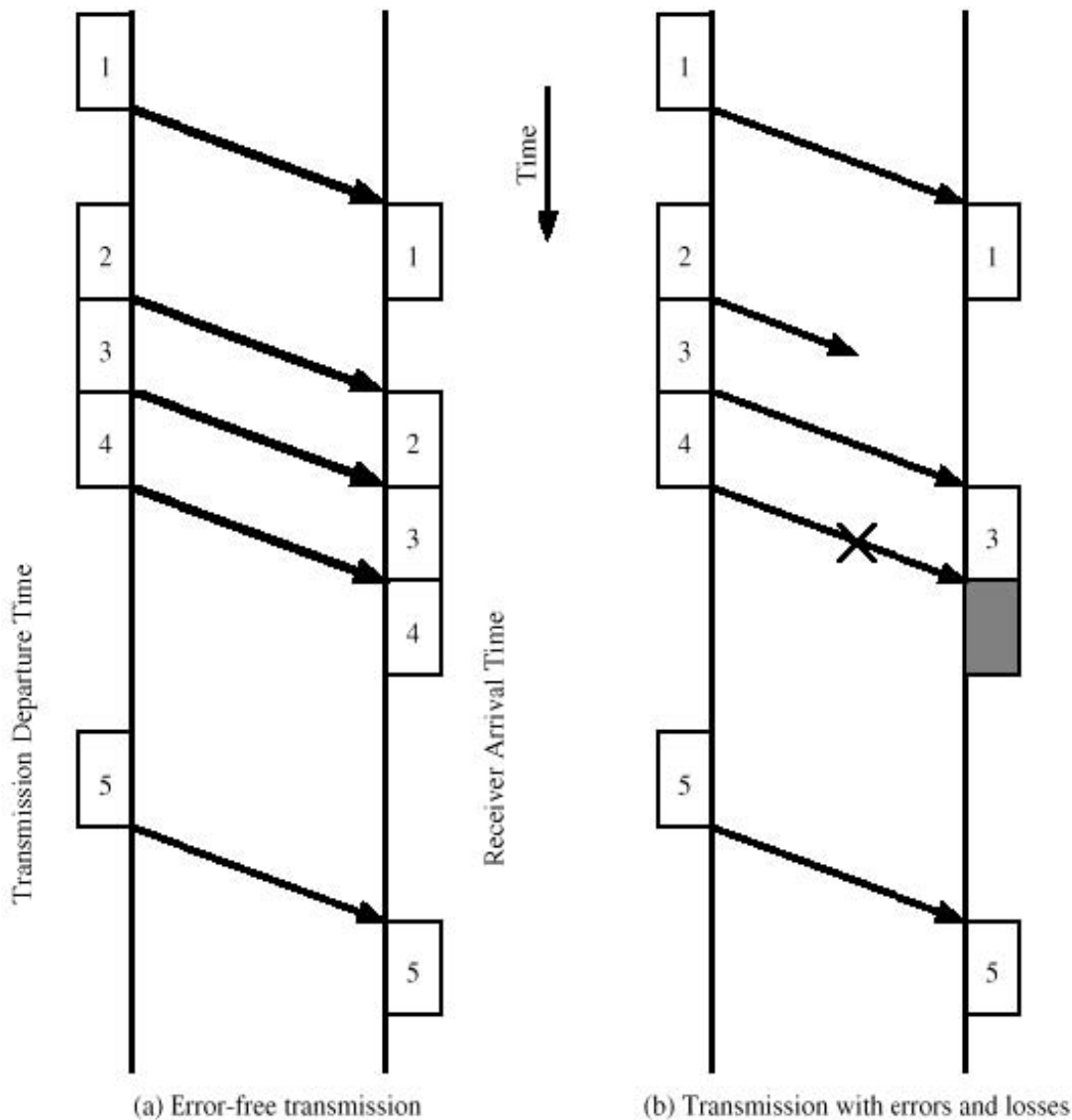
5.2 Flow control

Adalah suatu teknik untuk memastikan/meyakinkan bahwa suatu stasiun transmisi tidak menumpuk data pada suatu stasiun penerima.

Tanpa flow control, buffer dari receiver akan penuh sementara sedang memproses data lama. Karena ketika data diterima, harus dilaksanakan sejumlah proses sebelum buffer dapat dikosongkan dan siap menerima banyak data.

Gambar 5.4a tiap tanda panah menyatakan suatu perjalanan frame tunggal. Suatu data link antara dua stasiun dan transmisinya bebas error. Tetapi bagaimanapun, setiap frame yang ditransmisi semauanya dan sejumlah delay sebelum diterima.

Gambar 5.4b suatu transmisi dengan losses dan error.



Gambar 5.4. Model Dari Transmisi Frame

Bentuk sederhana dari flow control, yaitu **stop-and-wait flow control**.

Cara kerjanya : suatu entity sumber mentransmisi suatu frame. Setelah diterima, entity tujuan memberi isyarat untuk menerima frame lainnya dengan mengirim acknowledgment ke frame yang baru diterima. Sumber harus menunggu sampai menerima acknowledgment sebelum mengirim frame berikutnya. Entity tujuan kemudian dapat menghentikan aliran data dengan tidak memberi acknowledgment.

Untuk blok-blok data yang besar, sumber akan memecah menjadi blok-blok yang lebih kecil dan mentransmisi data dalam beberapa frame. Hal ini dilakukan dengan alasan :

- ✚ Transmisi yang jauh, dimana bila terjadi error maka hanya sedikit data yang akan ditransmisi ulang.
- ✚ Pada suatu multipoint line.
- ✚ Ukuran buffer dari receiver akan terbatas.

Efek dari penambahan delay dan kecepatan transmisi

Misal message panjang yang dikirim sebagai suatu rangkaian frame-frame f_1, f_2, \dots, f_n , Untuk suatu prosedur polling, kejadian yang terjadi :

Stasiun S1 mengirim suatu poll dari stasiun S2.

S2 merespon dengan f_1 .

S1 mengirim suatu acknowledgment.

S2 mengirim f_2 . S1 meng-acknowledgment.

·
·
·

S2 mengirim f_n .

S1 meng-acknowledgment.

Waktu total untuk mengirim data tersebut :

$$T_D = T_I + nT_F$$

Dimana : $T_I =$ waktu untuk memulai rangkaian = $t_{prop} + t_{poll} + t_{proc}$

$T_F =$ waktu untuk mengirim satu frame = $t_{prop} + t_{frame} + t_{proc} + t_{prop} + t_{ack} + t_{proc}$

Bila dianggap T_I relatif kecil dan dapat turun, proses antara transmisi dan penerima diabaikan dan frame acknowledgment sangat kecil; maka :

$$T_D = n(2t_{prop} + t_{frame})$$

Dari waktu itu, hanya $n \times t_{\text{frame}}$ yang sebenarnya dihasilkan pada transmisi data, maka efisiensi dari line :

$$U = \frac{n \times t_{\text{frame}}}{n (2t_{\text{prop}} + t_{\text{frame}})}$$

$$U = \frac{t_{\text{frame}}}{2t_{\text{prop}} + t_{\text{frame}}}$$

Bila $a = t_{\text{prop}}/t_{\text{frame}}$,
maka : $U = 1 / (1+2a)$

Persamaan diatas untuk a yang konstan, bentuk ekspresi lainnya :

$$a = \frac{\text{waktu penyebaran}}{\text{waktu transmisi}}$$

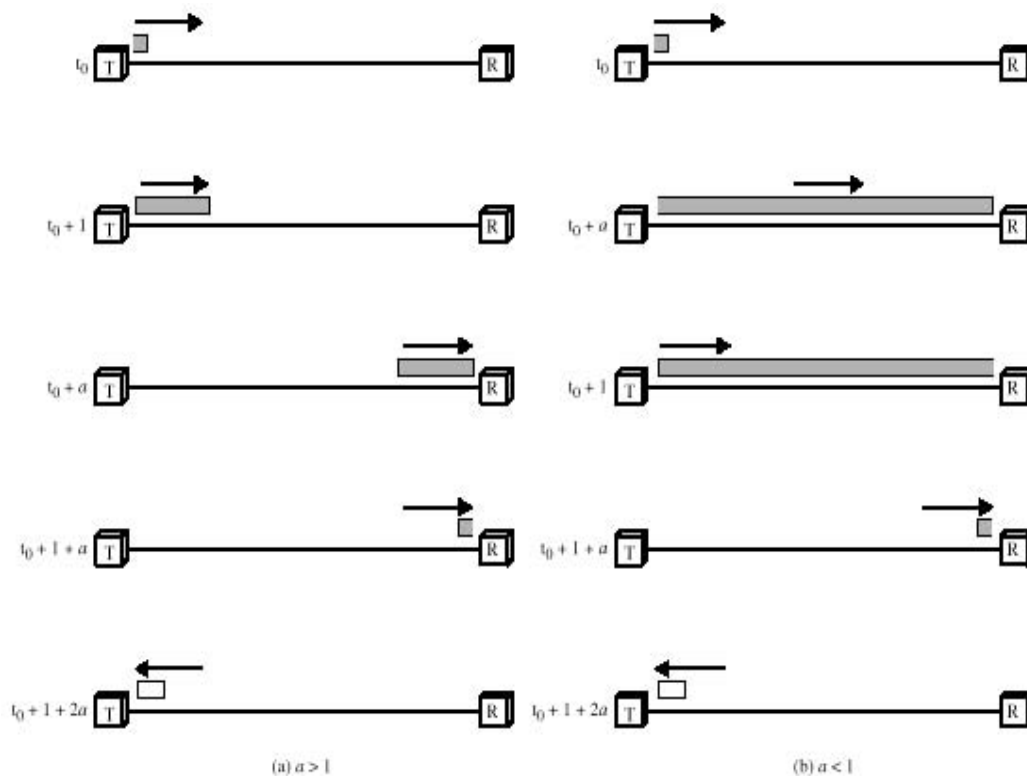
atau :

$$a = \frac{d/v}{L/R} = \frac{Rd}{VL}$$

Dimana : d = jarak link
 V = kecepatan penyebaran
 R = data rate
 L = panjang frame

Gambar 5.5 menggambarkan efek penggunaan a . Gambar 5.5a ($a < 1$) dimana panjang bit lebih kecil daripada frame. Pada saat t_0 , stasiun mulai mentransmisi suatu frame. Pada t_0+a , leading edge dari frame mencapai stasiun penerima, sementara stasiun pengirim masih melakukan proses transmisi frame. Pada t_0+1 , stasiun pengirim sudah mentransmisi secara lengkap. Pada t_0+1+a , stasiun penerima sudah menerima seluruh frame dan langsung mentransmisi suatu frame acknowledgment yang pendek. Acknowledgment ini tiba kembali di stasiun pengirim pada t_0+1+2a . Jadi total waktu penyebaran : $1 + 2a$. Total waktu transmisi : 1 . Sehingga efisiensi : $U = 1 / (1 + 2a)$

Hasil yang sama dicapai juga dengan $a > 1$, yang digambarkan pada gambar 5.5b.



Gambar 5.5. Efek Dari Utilitasi Stop And Wait

Contoh : pada local network dimana transmisi data digital melalui modem; data rate = 9600 bps, karena range jarak dari 0,1 – 10 Km, dengan data rate 0,1 – 10 Mbps, maka dipakai $V = 2 \times 10^8$ m/s; ukuran frame yang dipakai 500 bit; jika dipakai pada jarak pendek $d = 100$ m, maka $a = 9600 \text{ bps} \times 100 \text{ m} = 9,6 \times 10^{-6}$ dan pemakaiannya efektif. $2 \times 10^8 \text{ m/s} \times 500 \text{ bits}$
Jika dipakai pada jarak yang jauh $d = 5000$ Km, maka $a = 9600 \times 5 \times 10^8 = 0,48$ dan $2 \times 10^8 \times 500$

Efisiensi = 0,5.

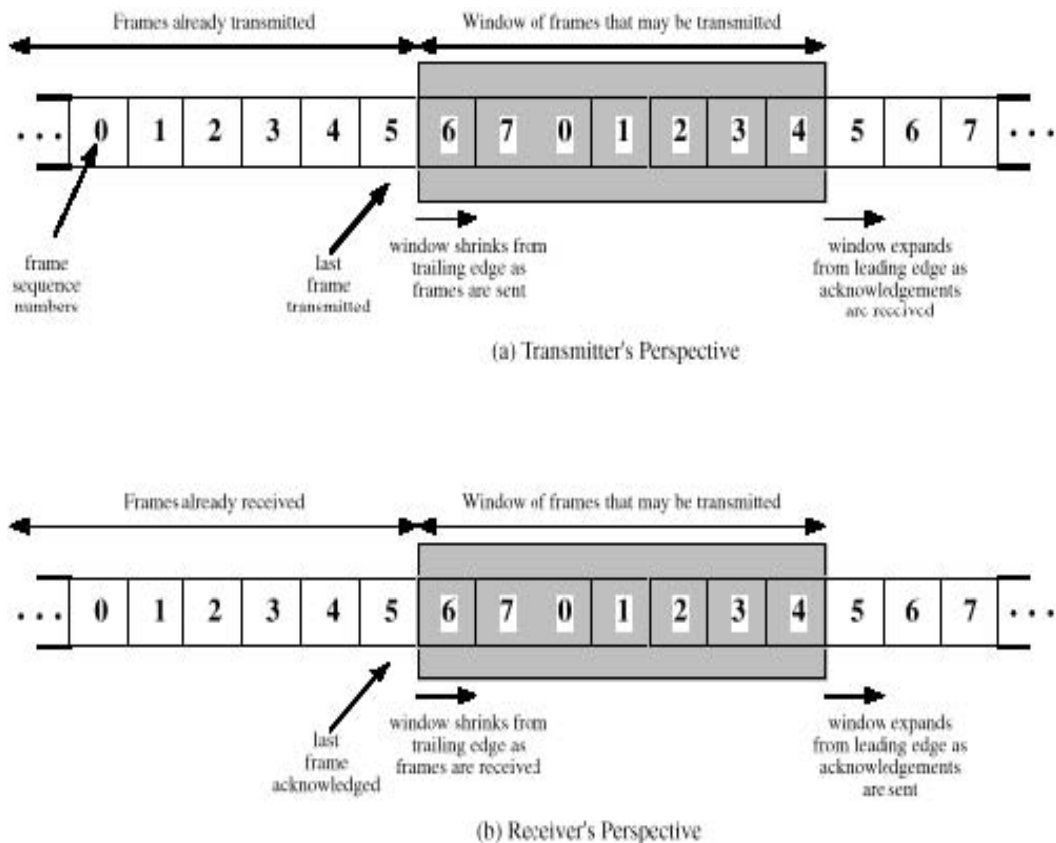
Protocol Sliding Window

Sliding-window flow control dapat digambarkan dalam operasi sebagai berikut :

Dua stasiun A dan B, terhubung melalui suatu link full-duplex. B dapat menerima n buah frame karena menyediakan tempat buffer untuk n buah frame. Dan A memperbolehkan pengiriman n buah frame tanpa menunggu suatu acknowledgement. Tiap frame diberi label nomor tertentu. B mengakui suatu frame dengan n mengirim suatu acknowledgement yang mengandung serangkaian nomor dari frame berikut yang diharapkan dan B siap untuk menerima n frame berikutnya yang dimulai dari nomor tertentu. Skema ini dapat juga dipakai untuk multiple frame acknowledge.

Gambar 5.6 menunjukkan proses sliding-window. Anggap dipakai 3 bit penomoran,

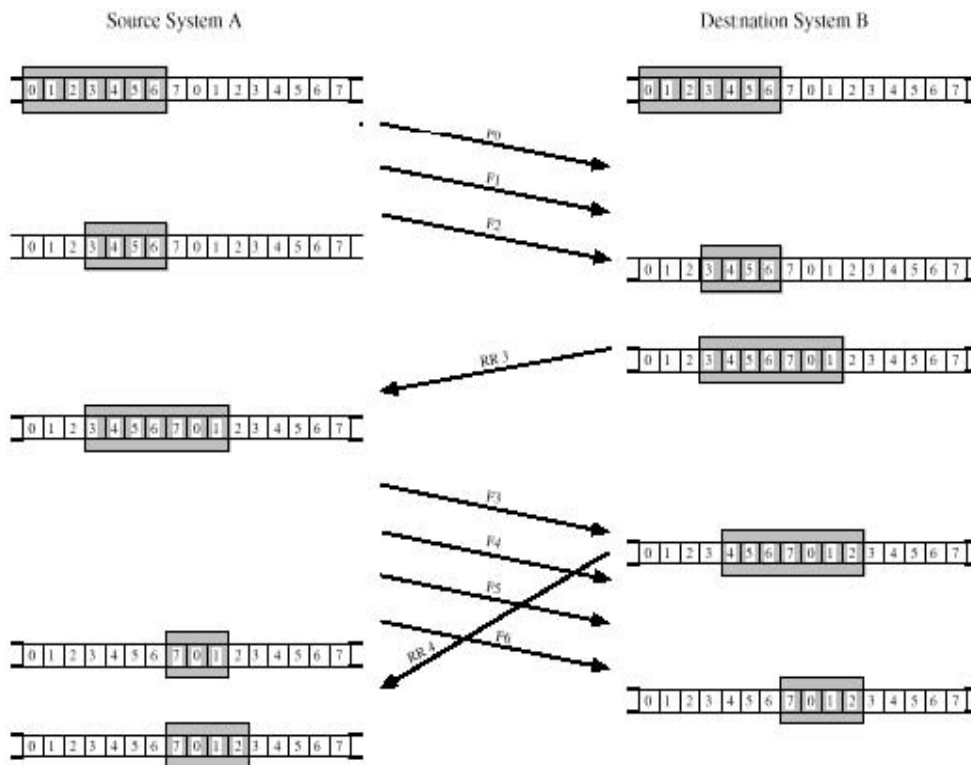
maka terdapat 0-7 nomor. Pada gambar, pengirim dapat mentransmit 7 buah frame, yang dimulai dengan frame ke 6. Setiap kali frame dikirim, daerah dalam kotak akan menyusut; setiap kali sebuah acknowledgment diterima, daerah dalam kotak tersebut akan membesar.



Gambar 5.6. Proses Sliding-Window

Gambar 5.7 menunjukkan suatu contoh, dimana di anggap ada 3 bit penomoran dan suatu ukuran window maksimum sebesar 7. A dan B mempunyai window yang mengindikasikan bahwa A boleh mengirim 7 buah frame, dimulai dengan frame ke 0 (f0).

Setelah mengirim 3 buah frame (f0,f1,f2) tanpa acknowledgment, A telah menyusutkan window-nya menjadi 4 buah frame. Window ini menyatakan bahwa A boleh mentransmit 4 buah frame, dimulai dengan frame nomor 3; pada kenyataannya, saya siap menerima 7 frame, yang dimulai dengan frame nomor 3. "Dengan acknowledgment ini, A kembali meminta izin untuk mentransmisi 7 frame masih, diawali dengan frame 3. A mulai mentransmisi frame 3, 4, 5 dan 6. B mengembalikan ACK 4, dimana mengakui frame 3, dan mengizinkan transmisi frame 4 sampai 2. Tetapi, pada waktu acknowledgment mencapai A, A sudah mentransmisi frame 4, 5 dan 6. Kesimpulannya bahwa A hanya boleh membuka window-nya untuk memperkenankan transmisi dari 4 frame, dimulai dengan frame 7.



Gambar 5.7. Contoh dari protocol sliding - windows

Penjelasan-penjelasan diatas untuk transmisi dalam satu arah saja. Jika 2 stasiun menukar data, masing-masing membutuhkan 2 window : satu untuk transmisi data dan yang lain untuk menerima. Teknik ini dikenal sebagai *piggy backing*. Untuk multipoint link, primary membutuhkan masing-masing secondary untuk transmisi dan menerima.

5.3 Error Control

Berfungsi untuk mendeteksi dan memperbaiki error-error yang terjadi dalam transmisi frame-frame. Ada 2 tipe error yang mungkin :

- ✚ Frame hilang : suatu frame gagal mencapai sisi yang lain
- ✚ Frame rusak : suatu frame tiba tetapi beberapa bit-bit-nya error.

Teknik-teknik umum untuk error control, sebagai berikut :

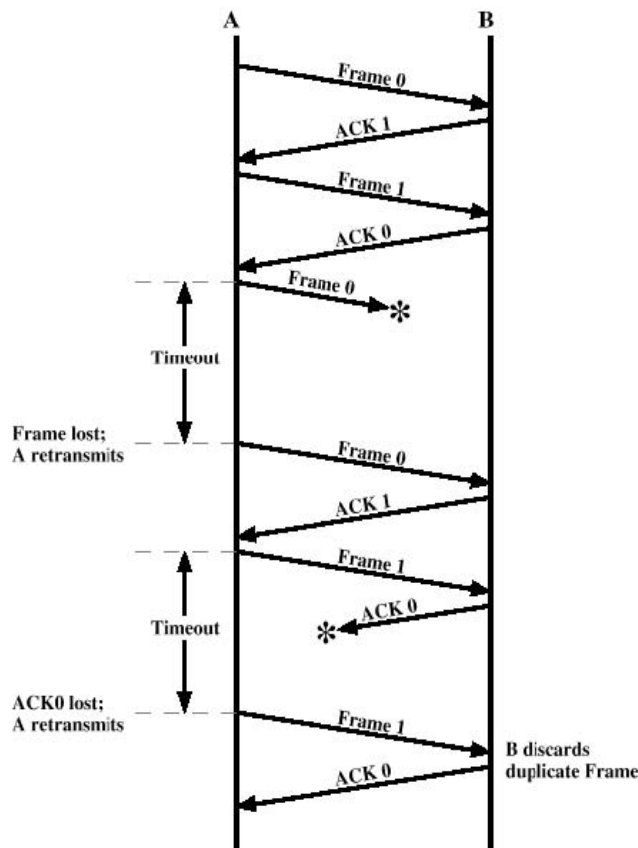
- ✚ Deteksi error : telah dibahas dalam chapter 4; dipakai CRC.
- ✚ Positive acknowledgment : tujuan mengembalikan suatu positif acknowledgment untuk penerimaan yang sukses, frame bebas error.
- ✚ Transmisi ulang setelah waktu habis : sumber mentransmisi ulang suatu frame yang belum diakui setelah suatu waktu yang tidak ditentukan.
- ✚ Negative acknowledgment dan transmisi ulang : tujuan mengembalikan negative acknowledgment dari frame-frame dimana suatu error dideteksi. Sumber mentransmisi ulang beberapa frame.

Mekanisme ini dinyatakan sebagai *Automatic repeat Request (ARQ)* yang terdiri dari 3 versi :

- ✚ Stop and wait ARQ.
- ✚ Go-back-N ARQ.
- ✚ Selective-reject ARQ.

Stop and wait ARQ

Berdasarkan pada teknik flow control stop and wait dan digambarkan dalam gambar 5.10. Stasiun sumber mentransmisi suatu frame tunggal dan kemudian harus menunggu suatu acknowledgment (ACK) dalam periode tertentu. Tidak ada data lain dapat dikirim sampai balasan dari stasiun tujuan tiba pada stasiun sumber. Bila tidak ada balasan maka frame ditransmisi ulang. Bila error dideteksi oleh tujuan, maka frame tersebut dibuang dan mengirim suatu Negative Acknowledgment (NAK), yang menyebabkan sumber mentransmisi ulang frame yang rusak tersebut.



Gambar 5.10. Stop-and-wait ARQ

Bila sinyal acknowledgment rusak pada waktu transmisi, kemudian sumber akan habis waktu dan mentransmisi ulang frame tersebut. Untuk mencegah hal ini, maka frame diberi label 0 atau 1 dan positive acknowledgment dengan bentuk ACK0 atau ACK1 : ACK0 mengakui menerima frame 1 dan mengindikasikan bahwa receiver siap untuk frame 0. Sedangkan ACK1 mengakui menerima frame 0 dan mengindikasikan bahwa receiver siap untuk frame 1.

Go-back-N ARQ

Termasuk continuous ARQ, suatu stasiun boleh mengirim frame seri yang ditentukan oleh ukuran window, memakai teknik flow control sliding window. Sementara tidak terjadi error, tujuan akan meng-acknowledge (ACK) frame yang masuk seperti biasanya.

Teknik Go-back-N ARQ yang terjadi dalam beberapa kejadian :

- ✚ Frame yang rusak. Ada 3 kasus :
 - ✓ A mentransmisi frame i. B mendeteksi suatu error dan telah menerima frame (i-1) secara sukses. B mengirim A NAK_i, mengindikasikan bahwa frame i ditolak. Ketika A menerima NAK ini, maka harus mentransmisi ulang frame i dan semua frame berikutnya yang sudah ditransmisi.
 - ✓ Frame i hilang dalam transmisi. A kemudian mengirim frame (i+1). B menerima frame (i+1) diluar permintaan, dan mengirim suatu NAK_i.
 - ✓ Frame i hilang dalam transmisi dan A tidak segera mengirim frame -frame tambahan. B tidak menerima apapun dan mengembalikan baik ACK atau NAK. A akan kehabisan waktu dan mentransmisi ulang frame i.

- ✚ ACK rusak. Ada 2 kasus :
 - ✓ B menerima frame i dan mengirim ACK (i+1), yang hilang dalam transmisi. Karena ACK dikumulatif (contoh, ACK6 berarti semua frame sampai 5 diakui), hal ini mungkin karena A akan menerima sebuah ACK yang berikutnya untuk sebuah frame berikutnya yang akan melaksanakan tugas dari ACK yang hilang sebelum waktunya habis.
 - ✓ Jika waktu A habis, A mentransmisi ulang frame I dan semua frame -frame berikutnya.

- ✚ NAK rusak. Jika sebuah NAK hilang, A akan kehabisan waktu (time out) pada serangkaian frame dan mentransmisi ulang frame tersebut berikut frame-frame selanjutnya.

Selective-reject ARQ

Hanya mentransmisi ulang frame-frame bila menerima NAK atau waktu habis. Ukuran window yang perlu lebih sempit daripada go-back-N. Untuk go-back-N, ukuran window $2^n - 1$ sedangkan selective-reject 2^n .

Skenario dari teknik ini untuk 3 bit penomoran yang mengizinkan ukuran window sebesar 7 :

1. Stasiun A mengirim frame 0 sampai 6 ke stasiun B.
2. Stasiun B menerima dan mengakui ketujuh frame-frame.
3. Karena noise, ketujuh acknowledgment hilang.
4. Stasiun A kehabisan waktu dan mentransmisi ulang frame 0.
5. Stasiun B sudah memajukan window penerimanya untuk menerima frame 7,0,1,2,3,4 dan 5. Dengan demikian dianggap bahwa frame 7 telah hilang dan bahwa frame nol yang baru, diterima.

Problem dari skenario ini yaitu antara window pengiriman dan penerimaan. Yang diatasi dengan memakai ukuran window max tidak lebih dari setengah range penomoran.

Performa

Go-back-N dan selective-reject lebih efisien daripada stop and wait.

Pemakaian maksimum (U) untuk masing-masing teknik :

Stop and wait :

$$U = \begin{cases} 1 & N > 2a+1 \\ \frac{N}{2a+1} & N < 2a+1 \end{cases}$$

Selective reject :

$$U = \begin{cases} 1-p & N > 2a+1 \\ \frac{N(1-p)}{2a+1} & N < 2a+1 \end{cases}$$

Go-back-N :

$$U = \begin{cases} \frac{1-p}{1+2a} & N > 2a+1 \\ \frac{N(1-p)}{(2a+1)(1-p+Np)} & N < 2a+1 \end{cases}$$

dimana : a = waktu penyebaran

N = ukuran window

p = probabilitas transmisi suatu frame dengan sukses.

5.4 Protokol-Protokol Data Link Control

Protokol-protokol bit-oriented didisain untuk memenuhi variasi yang luas dari kebutuhan data link, termasuk :

- ✚ Point to point dan multipoint links.
- ✚ Operasi Half-duplex dan full-duplex.
- ✚ Interaksi primary-secondary (misal : host-terminal) dan peer (misal : komputer-komputer).
- ✚ Link-link dengan nilai a yang besar (misal : satelit) dan kecil (misal : koneksi langsung jarak pendek).

Sejumlah protokol-protokol data link control telah dipakai secara luas dimana-mana :

- ✚ High-level Data Link Control (HDLC).
- ✚ Advanced Data Communication Control Procedures.
- ✚ Link Access Procedure, Balanced (LAP-B).
- ✚ Synchronous Data Link Control (SDLC).

Karakteristik-karakteristik Dasar

HDLC didefinisikan dalam tiga tipe stasiun, dua konfigurasi link, dan tiga model operasi transfer data.

Tiga tipe stasiun yaitu :

- ✚ Stasiun utama (*primary station*) : mempunyai tanggung jawab untuk mengontrol operasi link. Frame yang dikeluarkan oleh primary disebut **commands**.
- ✚ Stasiun sekunder (*secondary station*) : beroperasi dibawah kontrol stasiun utama. Frame yang dikeluarkan oleh stasiun-stasiun sekunder disebut responses. Primary mengandung link logika terpisah dengan masing-masing stasiun secondary pada line.
- ✚ Stasiun gabungan (*combined station*) : menggabungkan kelebihan dari stasiun-stasiun primary dan secondary. Stasiun kombinasi boleh mengeluarkan kedua-duanya baik commands dan responses.

Dua konfigurasi link, yaitu :

- ✚ Konfigurasi tanpa keseimbangan (*unbalanced configuration*) : dipakai dalam operasi point to point dan multipoint. Konfigurasi ini terdiri dari satu primary dan satu atau lebih stasiun secondary dan mendukung transmisi full-duplex maupun half-duplex.
- ✚ Konfigurasi dengan keseimbangan (*balanced configuration*) : dipakai hanya dalam operasi point to point. Konfigurasi ini terdiri dari dua kombinasi stasiun dan mendukung transmisi full-duplex maupun half-duplex.

Tiga mode operasi transfer data, yaitu :

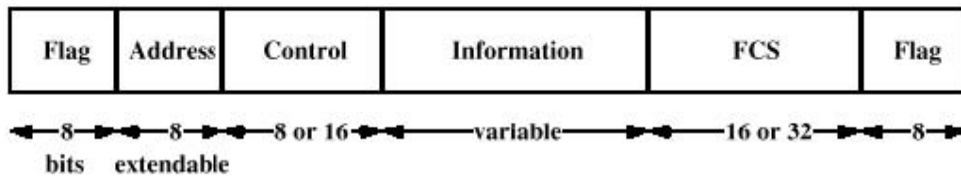
- ✚ *Normal Response Mode* (NRM) : merupakan unbalanced configuration. Primary boleh memulai data transfer ke suatu secondary, tetapi suatu secondary hanya boleh mentransmisi data sebagai response untuk suatu poll dari primary tersebut.
- ✚ *Asynchronous Balanced Mode* (ABM) : merupakan balanced configuration. Kombinasi stasiun boleh memulai transmisi tanpa menerima izin dari kombinasi stasiun yang lain.
- ✚ *Asynchronous Response Mode* (ARM) : merupakan unbalanced configuration. Dalam mode ini, secondary boleh memulai transmisi tanpa izin dari primary (misal : mengirim suatu respon tanpa menunggu suatu command). Primary masih memegang tanggung jawab pada line, termasuk inisialisasi, perbaikan error dan logika pemutusan.

Struktur frame

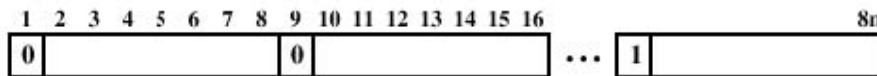
HDLC memakai transmisi synchronous. Gambar 5. 13 menunjukkan struktur dari frame HDLC. Frame ini mempunyai daerah-daerah :

- ✚ Flag : 8 bit
- ✚ Address : satu atau lebih oktaf.
- ✚ Control : 8 atau 16 bit.
- ✚ Informasi : variabel.
- ✚ Frame Check Sequence (FCS) : 16 atau 32 bit.
- ✚ Flag : 8 bit.

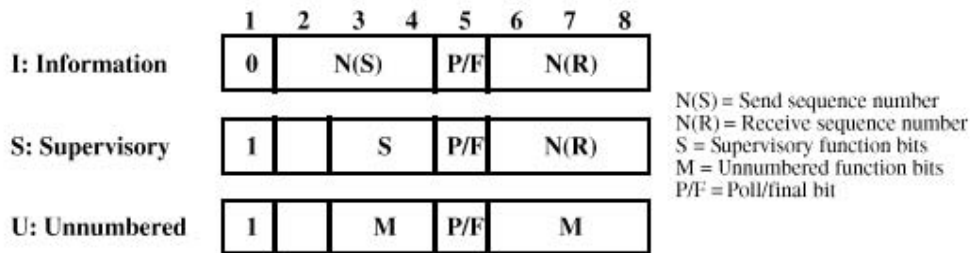
Flag address dan control dikenal sebagai **header**, FCS dan flag dinyatakan sebagai **trailer**.



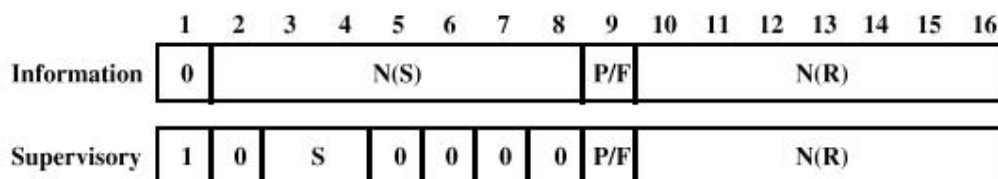
(a) Frame format



(b) Extended Address Field



(c) 8-bit control field format



(d) 16-bit control field format

Gambar 5.13. Struktur Frame HDLC

Daerah-daerah Flag

Membatasi frame dengan pola khusus 01111110. Flag tunggal mungkin dipakai sebagai flag penutup untuk satu frame dan flag pembuka untuk berikutnya. Stasiun yang terhubung ke link secara kontinu mencari rangkaian flag yang digunakan untuk synchronisasi pada start dari suatu frame. Sementara menerima suatu frame, suatu stasiun melanjutkan untuk mencari rangkaian flag tersebut untuk menentukan akhir dari frame.

Apabila pola 01111110 terdapat didalam frame, maka akan merusak level frame synchronisasi. Problem ini dicegah dengan memakai **bit stuffing**. Transmitter akan selalu menyisipkan suatu 0 bit ekstra setelah 5 buah rangkaian '1' dalam frame. Setelah mendeteksi suatu permulaan flag, receiver memonitor aliran bit. Ketika suatu pola 5 rangkaian '1' timbul, bit ke enam diperiksa. Jika bit ini '0', maka akan dihapus. Jika bit ke 6 dan ke 7 keduanya adalah '1', stasiun pengirim memberi sinyal suatu kondisi tidak sempurna.

Dengan penggunaan bit stuffing maka terjadi **data transparency** (=transparansi data).

Gambar 5.14 Menunjukkan suatu contoh dari bit stuffing

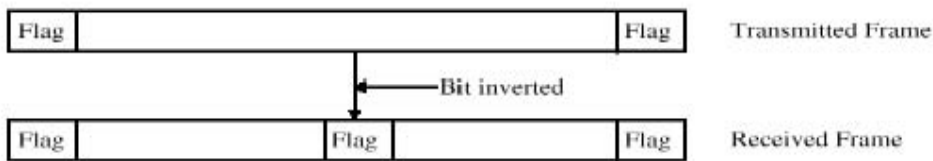
Original Pattern:

111111111111011111101111110

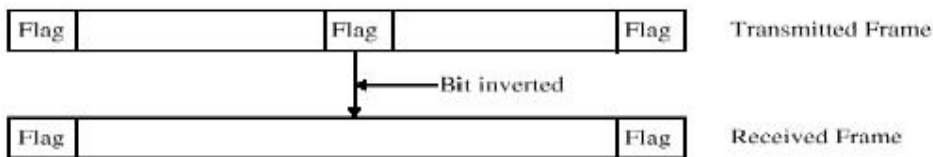
After bit-stuffing

1111101111101101111101011111010

(a) Example



(b) An inverted bit splits a frame in two



(c) An inverted merges two frames

Gambar 5.14. Bit stuffing.

Daerah Address

Dipakai untuk identitas stasiun secondary yang ditransmisi atau untuk menerima frame. Biasanya formatnya dengan panjang 8 bit, tetapi dengan persetujuan lain boleh dipakai dengan panjang 7 bit dan LSB dalam tiap oktet adalah '1' atau '0' bergantung sebagai akhir oktet dari daerah address atau tidak.

Daerah Control

HDLC mendefinisikan tiga tipe frame :

- ✚ *Information frames* (I-frames) : membawa data untuk ditransmisi pada stasiun, dikenal sebagai user data, untuk control dasar memakai 3 bit penomoran, sedangkan untuk control yang lebih luas memakai 7 bit.
- ✚ *Supervisory frames* (S-frames) : untuk kontrol dasar memakai 3 bit penomoran, sedangkan untuk control yang lebih luas memakai 7 bit.
- ✚ *Unnumbered frames* (U-frames) : melengkapi tambahan fungsi kontrol link.

Gambar 5.13b dan d, satu atau 2 bit pertama dari daerah kontrol menunjukkan tipe frame.

Daerah Informasi

Ditampilkan dalam I-frames dan beberapa U-frames.

Panjangnya harus merupakan perkalian dari 8 bit.

Daerah Frame Check Sequence (FCS)

Dipakai untuk mengingat bit-bit dari frame, tidak termasuk flag-flag. Biasanya panjang FCS adalah 16 bit memakai definisi CRC-CCITT. 32 bit FCS memakai CRC 32.

Operasi

Operasi dari HDLC terdiri dari pertukaran I-frames, S-frames, dan U-frames antara sebuah primary dan sebuah secondary atau antara dua primary.

Information Frames

Tiap I-frame mengandung serangkaian nomor dari frame yang ditransmisi dan suatu poll/final (P/F) bit. Poll bit untuk command (dari primary) dan final bit (dari secondary) untuk response.

Dalam Normal response mode (NRM), primary menyebabkan suatu pull yang memberi izin untuk mengirim, dengan mengeset poll bit ke '1', dan secondary mengeset final bit ke '1' pada akhir respon I-frame -nya.

Dalam asynchronous response mode (ARM) dan Asynchronous balanced mode (ABM), P/F bit kadang dipakai untuk mengkoordinasi pertukaran dari S- dan U-frames.

Supervisory Frame

S-frame dipakai untuk flow dan error control.

Unnumbered Frames

U-frame dipakai untuk fungsi kontrol. Frame ini tidak membawa rangkaian nomor-nomor dan tidak mengubah flow dari penomoran I-frame.

Frame-frame ini dikelompokkan menjadi kategori-kategori :

- ✚ *Mode-setting commands and responses* ; mode-setting command ditransmisi oleh stasiun primary/kombinasi untuk inialisasi atau mengubah mode dari stasiun secondary/kombinasi.
- ✚ *Information transfer commands and responses*; dipakai untuk pertukaran informasi antara stasiun-stasiun.
- ✚ *Recovery commands and responses* ; dipakai ketika mekanisme ARQ yang normal tidak berkenan atau tidak akan bekerja.
- ✚ *Miscellaneous commands and responses*.

Contoh-contoh Operasi

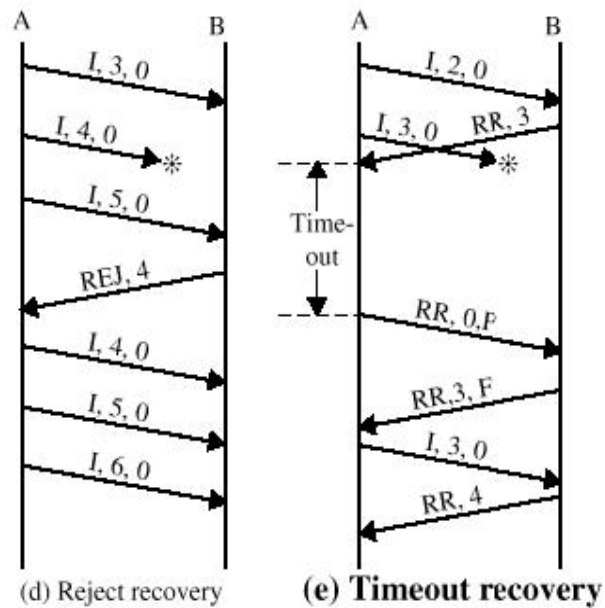
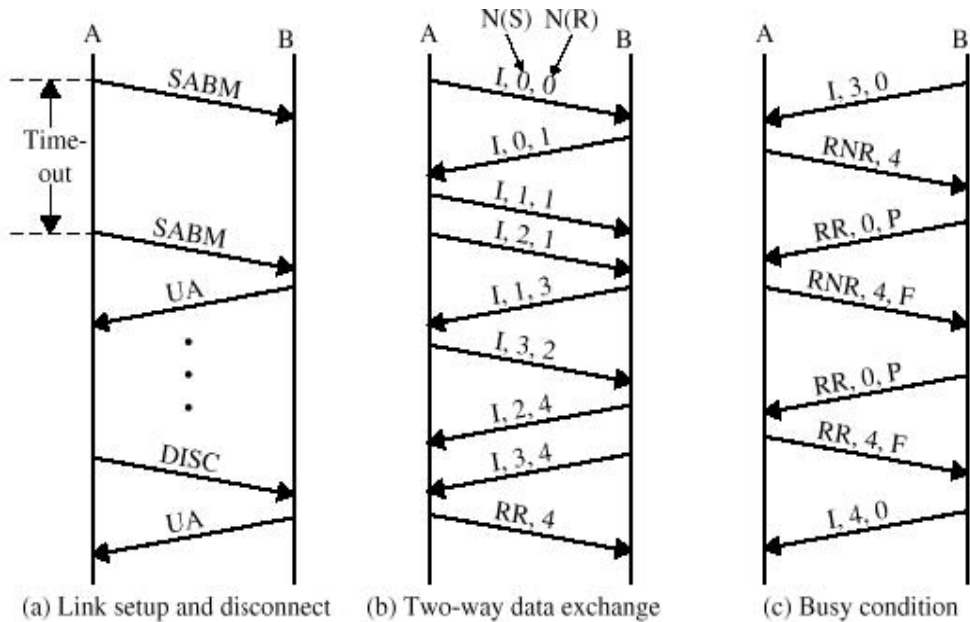
Gambar 5.15 menampilkan beberapa contoh operasi HDLC.

Gambar 5.15a menunjukkan frame-frame yang terlihat dalam link setup dan disconnect. Entity HDLC untuk satu sisi mengeluarkan command SABM untuk sisi yang lain dan memulai timer. Sisi yang lain, setelah menerima command SABM, mengembalikan respon UA dan mengeset variabel lokal dan counter ke nilai inialisasinya. Entity awal menerima respon UA, mengeset variabelnya dan counter-counter, dan menghentikan timer. Koneksi logika sekarang aktif, dan kedua sisi boleh mulai mentransmisi frame-frame. Sewaktu timer selesai tanpa suatu respon, A akan mengulang SABM. Hal ini akan diulang sampai UA atau DM diterima.

Penggambaran yang sama untuk prosedur pemutusan (disconnect). Satu sisi mengeluarkan command DISC dan yang lain merespon dengan respon UA.

Gambar 5.15b menggambarkan pertukaran full-duplex dari I-frames. Ketika suatu entity mengirim suatu nomor I-frame dalam suatu anak panah dengan tanpa penambahan data, kemudian serangkaian nomor yang diterima diulang (misal I,1.1;I,2.1 dalam arah A ke B). Ketika suatu entity menerima suatu nomor I -frame

dalam suatu anak panah dengan tanpa frame yang keluar, kemudian serangkaian nomor yang diterima dalam frame yang keluar berikutnya harus mencerminkan aktivitas kumulatif (misal I,1.3 dalam arah B ke A). Catatan, sebagai tambahan untuk I-frames, pertukaran data boleh melibatkan S-frames.



Gambar 5.15. Contoh dari operasi HDLC

Gambar 5.15c menunjukkan suatu operasi untuk kondisi yang sibuk. Beberapa kondisi dapat meningkat karena entity HDLC tidak mampu memproses I-frames secepat I-frame tersebut tiba, atau maksud user tidak mampu menerima data secepat mereka tiba dalam I-frames. Buffer dari entity penerima akan terisi dan harus menghentikan flow I-frame yang masuk dengan memakai command RNR. Dalam contoh ini, stasiun mengeluarkan RNR, yang memerlukan sisi yang lain untuk

menahan transmisi I-frames. Stasiun yang menerima RNR akan mem-poll stasiun yang sibuk pada beberapa interval period dengan mengirim RR dengan set P bit. Hal ini memerlukan sisi lainnya untuk merespon dengan RR ataupun RNR. Ketika kondisi sibuk telah jelas, B mengembalikan RR, dan transmisi I-frame dari NT dapat mulai lagi.

Gambar 5.15d suatu contoh error recovery memakai command REJ. Dalam contoh ini, A mentransmisi I-frame nomor 3,4 dan 5. Nomor 4 terjadi error. B mendeteksi error tersebut dan membuang frame tersebut. Ketika B menerima I-frame nomor 5, maka frame ini dibuang karena diluar permintaan dan mengirim REJ dengan N(R) dari 4. Hal ini menyebabkan A untuk melakukan transmisi ulang dari semua I -frame yang sudah dikirim, dimulai dengan frame 4. Dan kemudian dapat melanjutkan untuk mengirim frame tambahan setelah frame yang ditransmisi ulang.

Gambar 5.15e menunjukkan error recovery memakai time out. Dalam contoh ini, A mentransmisi I-frame nomor 3 sebagai akhir dalam rangkaian I -frames. Frame tersebut mengalami error. B mendeteksi error tersebut dan membuangnya. Bagaimanapun, B tidak dapat mengirim REJ. Hal ini karena tidak ada cara untuk mengetahui bila ini adalah suatu I-frame. Jika suatu error dideteksi dalam suatu frame, semua bit-bit ini dari frame tersebut disangsikan, dan receiver tidak mempunyai cara untuk bertindak atas hal tersebut. A, bagaimanapun, memulai suatu timer begitu frame ditransmisi. Timer ini mempunyai panjang durasi yang cukup untuk merentang respon waktu yang diharapkan. Ketika timer berakhir, A melaksanakan tindakan pemulihan. Hal ini biasanya dilakukan dengan mem-poll sisi lain dengan command RR dengan set P bit, untuk menentukan status dari sisi lain tersebut. Karena poll membutuhkan suatu respon, entity akan menerima suatu frame yang mengandung N(R) dan mampu memproses. Dalam kasus ini, respon mengindikasikan bahwa frame 3 hilang, dimana A mentransmisi ulang.