

## BAB 7

# KOMUNIKASI DATA DIGITAL DAN DATA LINK CONTROL (DLC)

### 7.1 Pendahuluan

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 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.

## 7.2 Konfigurasi-Konfigurasi Line

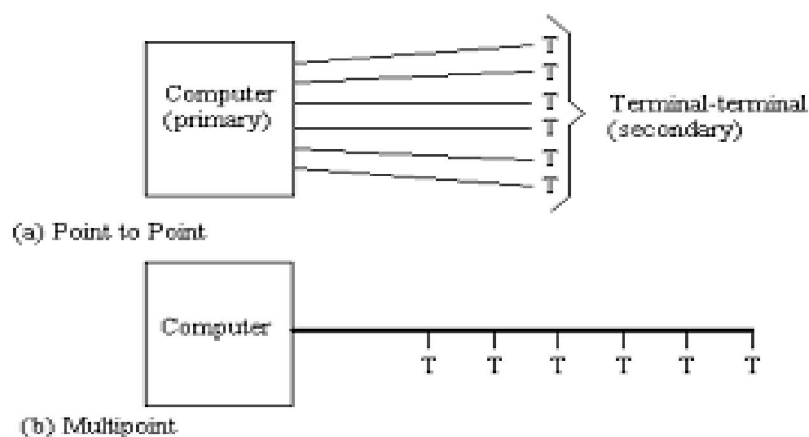
Ada 3 karakteristik yang membedakan berbagai konfigurasi data link, yaitu : topology, duplexity dan line discipline (rancangan tata tertib).

### 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.



Gambar 7.1 Konfigurasi Komputer / terminal tradisional

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

Gambar 7.1, menunjukkan keuntungan konfigurasi multipoint, yaitu : komputer hanya perlu satu tumpukan dan juga hanya memerlukan satu kabel transmisi tunggal sehingga menghemat biaya operasional.

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 satu frekuensi dan menerima pada frekuensi yang lain maka beroperasi dalam mode full-duplex.

#### Line Discipline (Rancangan tata tertib)

Beberapa tata tertib diperlukan dalam penggunaan link transmisi. Pada mode half-duplex, hanya satu stasiun yang dapat mentransmisikan pada suatu waktu. Baik mode half-duplex atau full-

duplex, suatu stasiun hanya mentransmisikan 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 penerima 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 :

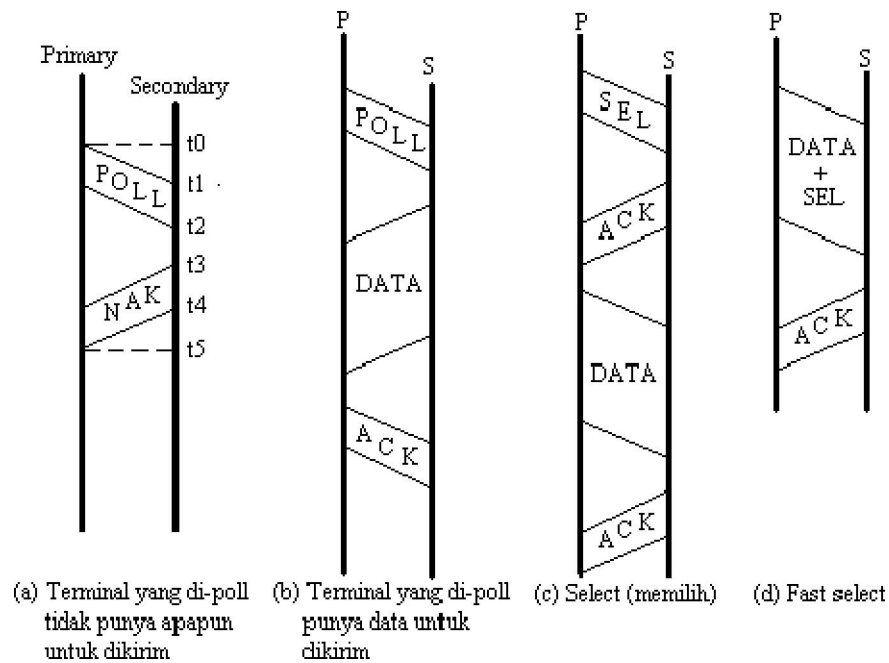
- **Establishment** (penentuan): memutuskan stasiun mana yang transmisi dan mana yang menerima dan apa receiver siap untuk menerima.
- **Data transfer**: data ditransfer dalam satu atau lebih blok-blok *acknowledgment*.
- **Termination** : membatasi koneksi logika (hubungan transmitter-receiver).

### 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 7.2. Serangkaian poll dan select.

Gambar 7.2 memperlihatkan konsep ini. Dalam 7.2a, 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.

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

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

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

$t_{nak}$  = 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 7.2c, dimana ditunjukkan fungsi select.

Gambar 7.2d, 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 sering kali ditransmisikan dan waktu transfer untuk message tersebut tidak lebih lama daripada waktu balasan.

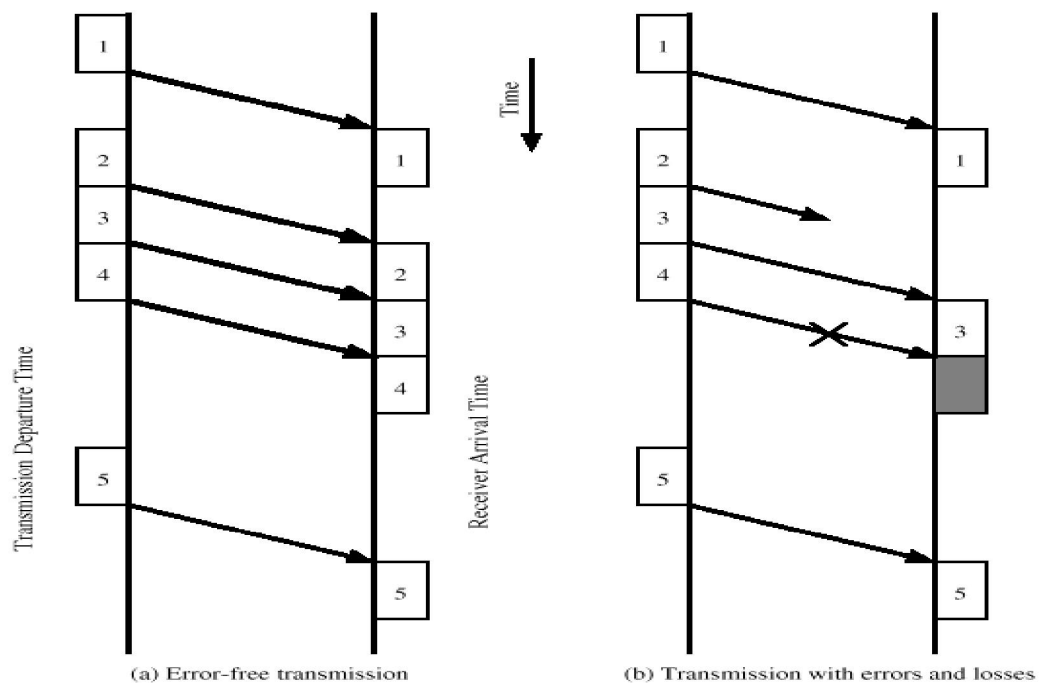
Bentuk lain dari linediscipline, yaitu **contention**, dimana tidak ada primary tetapi hanya suatu kumpulan stasiun-stasiun peer kedua yang baik transmitter dan receiver harus diidentifikasi. Stasiun ini dapat mentransmisi jika jalur/line sedang bebas; kalau tidak maka harus menunggu. Teknik ini dapat 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.

### 7.3 Flow Kontrol

Adalah suatu teknik untuk memastikan/meyakinkan bahwa suatu stasiun transmisi tidak menumpuk data pada suatu stasiun penerima. Tanpa flow control, buffer di 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 tiaptandapanah menyatakan suatu perjalanan frame tunggal. Suatu data link antara dua stasiun dan transmisinya bebas error. Tetapi bagaimanapun, setiap frame yang ditransmisikan akan mengalami sejumlah delay sebelum diterima. Gambar 5.4b suatu transmisi dengan losses dan error.



Gambar 7.3. Model dari transmisi frame

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

Carakerjanya: suatu entity sumber mentransmisikan 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 mentransmisikan data dalam beberapa frame. Hal ini dilakukan dengan alasan :

- Transmisi yang jauh, di mana bila terjadi error maka hanya sedikit data yang akan ditransmisikan 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  $S_1$  mengirim suatu poll dari stasiun  $S_2$ .

$S_2$  merespon dengan  $f_1$ .

$S_1$  mengirim suatu acknowledgment.

$S_2$  mengirim  $f_2$ .



S<sub>1</sub> meng-acknowledgment.

.

.

.

S<sub>2</sub> mengirim f<sub>n</sub>.

S<sub>1</sub> 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 antar transmisi dan penerima diabaikan dan frame acknowledgment sangat kecil; maka :

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

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

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

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

$$2t_{prop} + t_{frame}$$

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

Persamaan diatas untuk  $a$  yang konstan, bentuk ekspresi lainnya :

waktu penyebaran

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

waktu transmisi

atau :

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

$$L/R \quad VL$$

Dimana :  $d$  = jarak link

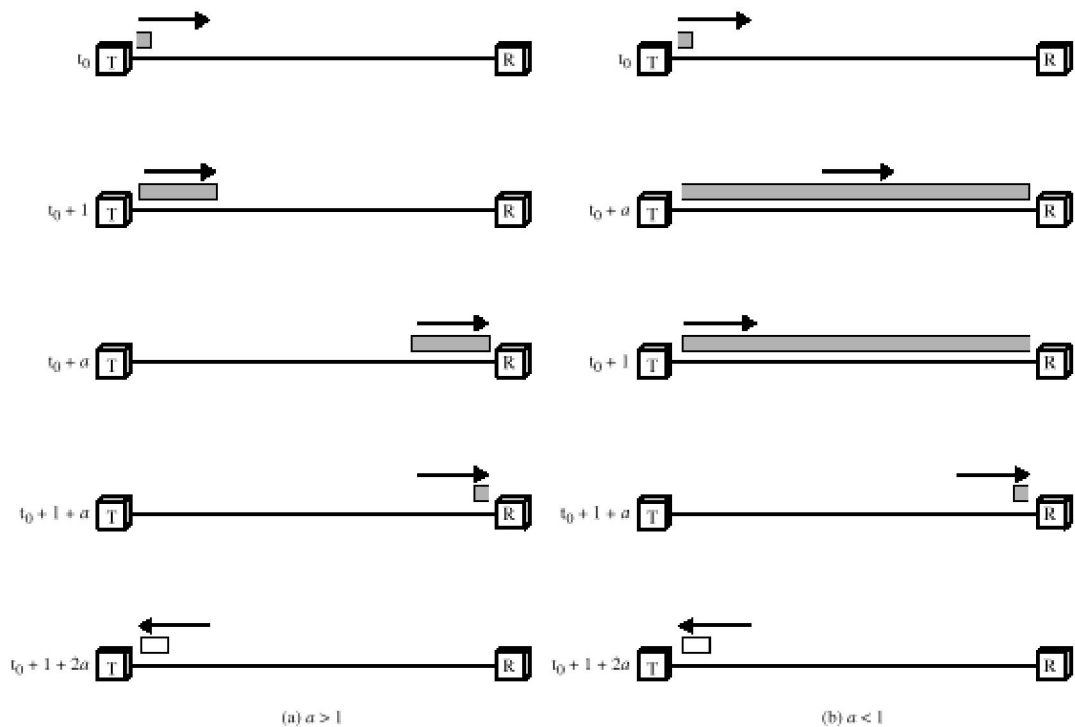
$V$  = kecepatan penyebaran

$R$  = data rate

$L$  = panjang frame

Gambar 7.4 menggambarkan efek penggunaan  $a$ . Gambar 7.4a ( $a < 1$ ) dimana panjang bit

lebih kecil daripada frame. Pada saat  $t_0$ , stasiun mulai mentransmisikan 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 mentransmisikan secara lengkap. Pada  $t_0 + 1 + a$ , stasiun penerima sudah menerima seluruh frame dan langsung mentransmisikan suatu frame acknowledgment yang pendek. Acknowledgment ini tiba kembali di



Gambar 7.4. Efek dari Utilisasi Stop and wait.

stasiun pengirim pada  $t_0+1+2\alpha$ . Jadi total waktu penyebaran:  $1+2\alpha$ . Total waktu transmisi: 1. Sehingga efisiensi:  $U = \frac{1}{1+2\alpha}$

Hasil yang sama dicapai juga dengan  $\alpha < 1$ , yang digambarkan pada gambar 7.4b.

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  $V =$

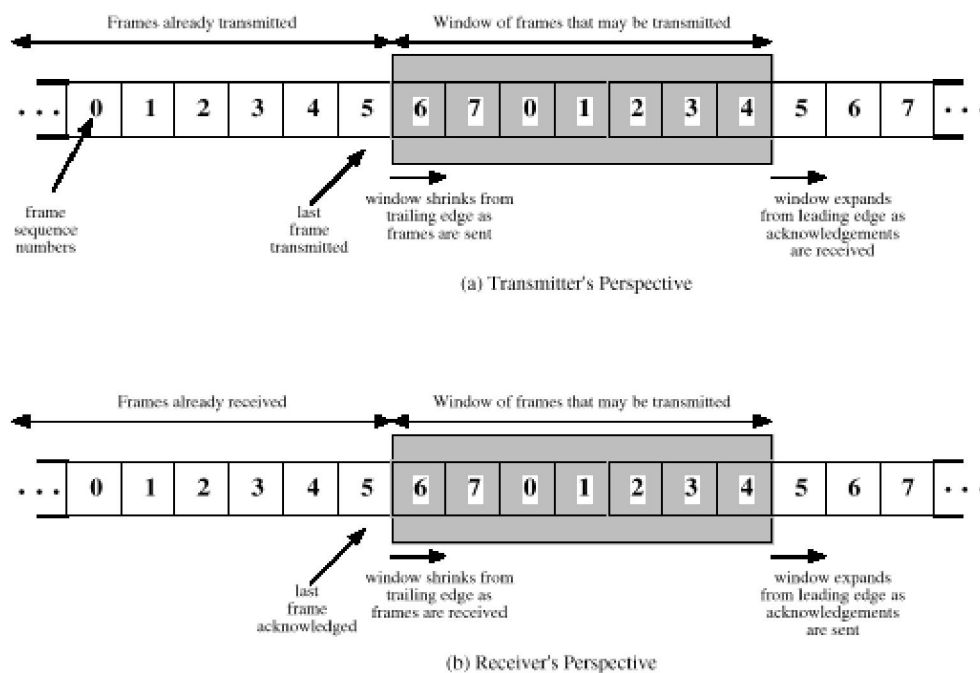
$2 \times 10^8 \text{ m/s}$ ; ukuran frame yang dipakai 500 bit; jika dipakai pada jarak pendek  $d = 100 \text{ m}$ , maka  $a = \frac{9600}{\text{bps} \times 100 \text{ m}} = 9,6 \times 10^{-6}$  dan pemakaiannya efektif.

$2 \times 10^8 \text{ m/s} \times 500 \text{ bits}$

Jika dipakai pada jarak yang jauh  $d = 5000 \text{ Km}$ , maka  $a = \frac{9600 \times 5 \times 10^6}{2 \times 10^8 \times 500} = 0,48$  dan

$2 \times 10^8 \times 500$

Efisiensi = 0,5.



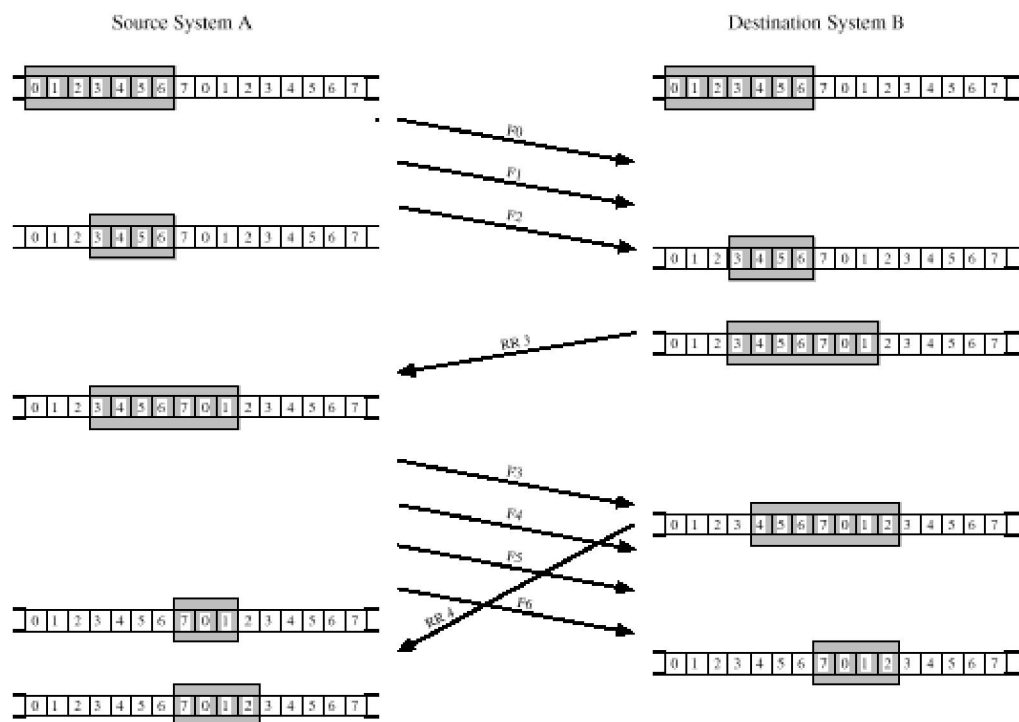
Gambar 7.5. Proses Sliding-window.

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 mengirim suatu acknowledgement yang mengandung serangkaian nomor dari frame berikutnya yang diharapkan dan B siap untuk menerima frame

berikutnya yang dimulai dari nomor tertentu. Skema ini dapat juga dipakai untuk multiple frame acknowledge. Gambar 7.5 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 7.6. Contoh dari protokol sliding-window.

Gambar 7.6 menunjukkan suatu contoh, di mana dianggap ada 3 bit penomoran dan suatu ukuran window maksimum sebesar 7. Ada B mempunyai window yang mengindikasikan bahwa A boleh mengirim 7 buah frame, dimulai dengan frame ke 0 ( $f_0$ ). Setelah mengirim 3 buah

frame ( $f_0, f_1, f_2$ ) 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, sayasiapmenerima 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 ACK4, 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.

Penjelasan-penjelasan di atas 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.