

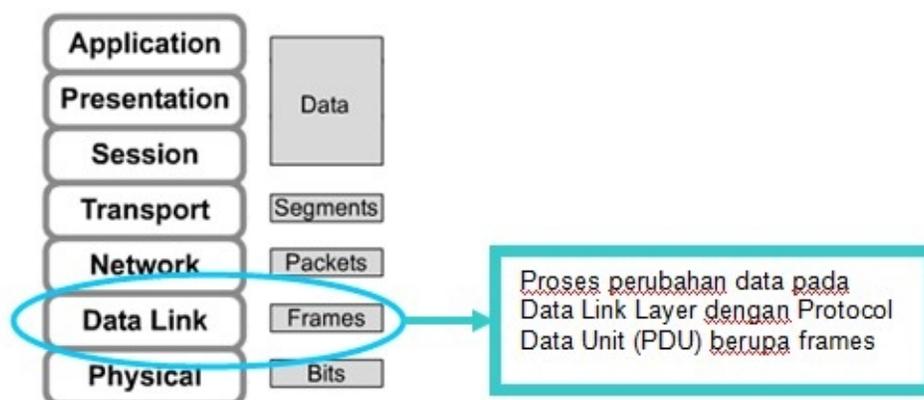
DATA LINK LAYER

Pengertian Data Link Layer

- Menyediakan prosedur pengiriman data antar jaringan
- Mendeteksi dan mengoreksi error yang mungkin terjadi di physical layer
- Memiliki address secara fisik yang sudah di-kode-kan secara langsung ke network card pada saat pembuatan card tersebut (disebut **MAC Address**)
- Contoh: Ethernet, HDLC, Aloha, IEEE 802 LAN, FDDI
- Perangkat yang beroperasi di layer ini adalah bridge dan switch



Gambar. 1: Fungsi dari Data Link Layer



Gambar. 2: PDU pada Data Link Layer berupa Frames



min. length of Data Field = 0 Byte



max. length of Data Field = 8 Byte

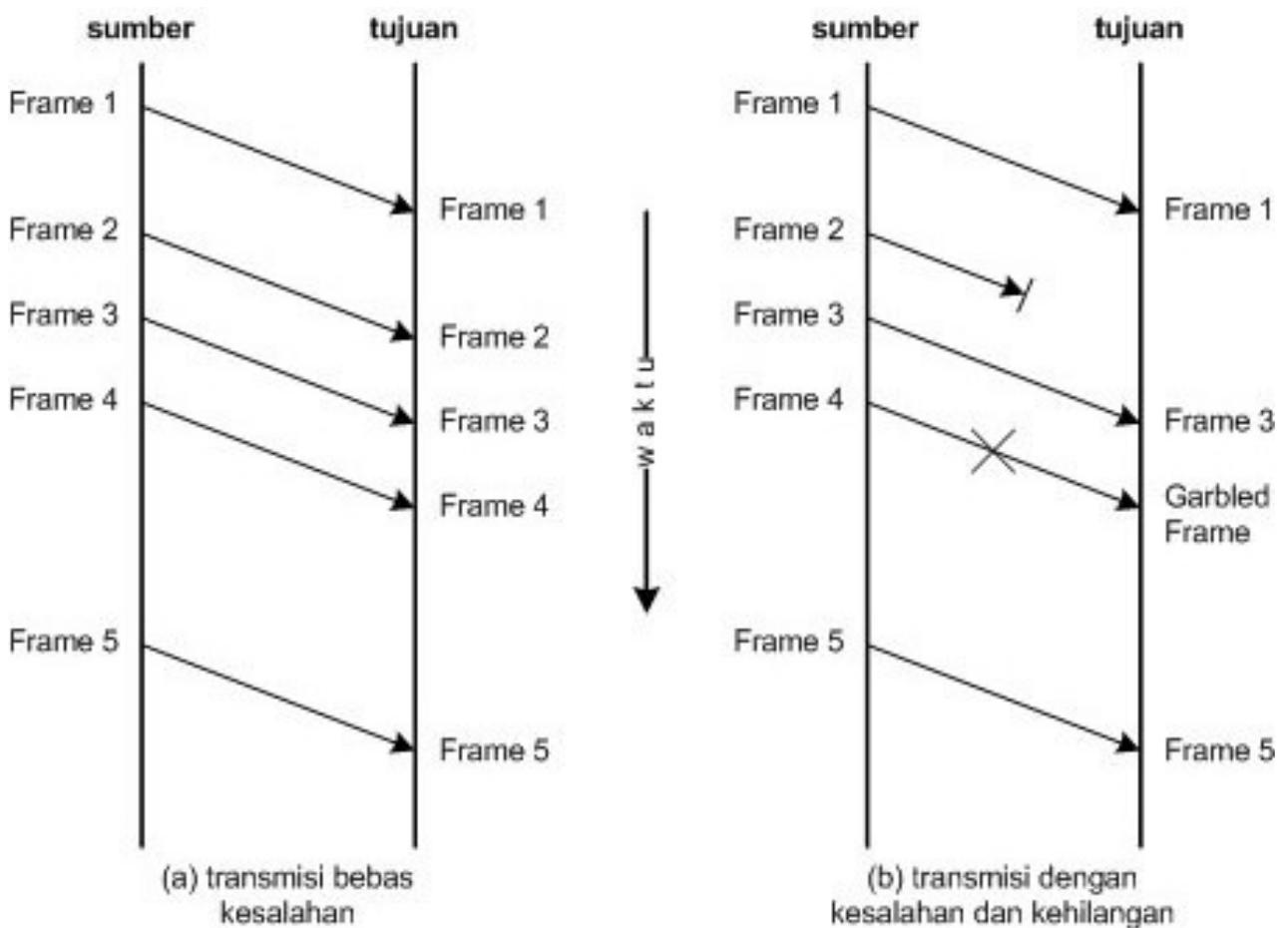
Gambar. 3: Data Frame

Flow Control

Flow Control adalah suatu teknik untuk menjamin bahwa entitas pengirim tidak akan membanjiri data kepada entitas penerima. Entitas penerima secara khusus mengalokasikan *buffer* dengan beberapa kali panjangnya transfer.

Ketika data diterima, receiver harus mengerjakan sejumlah proses tertentu sebelum mengalirkan data ke software dengan level yang lebih tinggi. Dengan tidak adanya flow control maka buffer pada penerima dapat terisi penuh dan melebihi kapasitas, bersamaan pada saat penerima masih memproses data sebelumnya.

Gambar di sebelah kiri menunjukkan mekanisme flow-control dengan tidak adanya error, Sumbu keatas adalah urutan waktu yang akan mempermudah dalam menggambarkan hubungan kirim dan terima yang benar sebagai fungsi waktu. Masing-masing tanda panah (→) menunjukkan satu frame data yang sedang dalam perjalanan diantara dua stasiun. Data dikirimkan dalam urutan frame yang masing-masing frame berisi bagian data dan sejumlah informasi pengontrol.



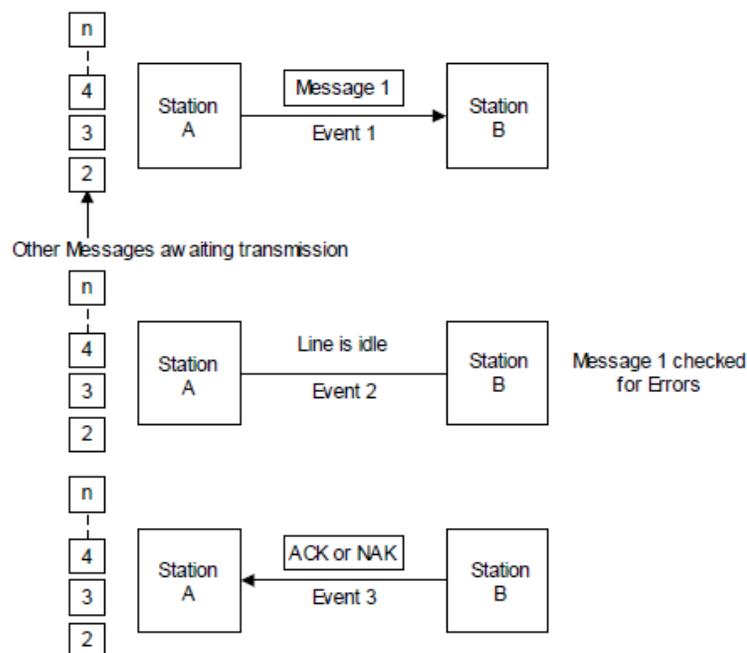
Gambar. 4: Model Transmisi Frame

Diasumsikan bahwa semua frame yang dikirimkan berhasil diterima dengan sukses, tidak ada frame yang hilang dan tidak ada frame yang datang mengalami error. Selanjutnya frame-frame tersebut tiba bersamaan dengan dikirimkannya frame, bagaimanapun juga masing-masing frame yang dikirimkan sebelum diterima akan mendapat delay pada saluran yang besarnya berubah-ubah.

o **Stop And Wait Flow Control**

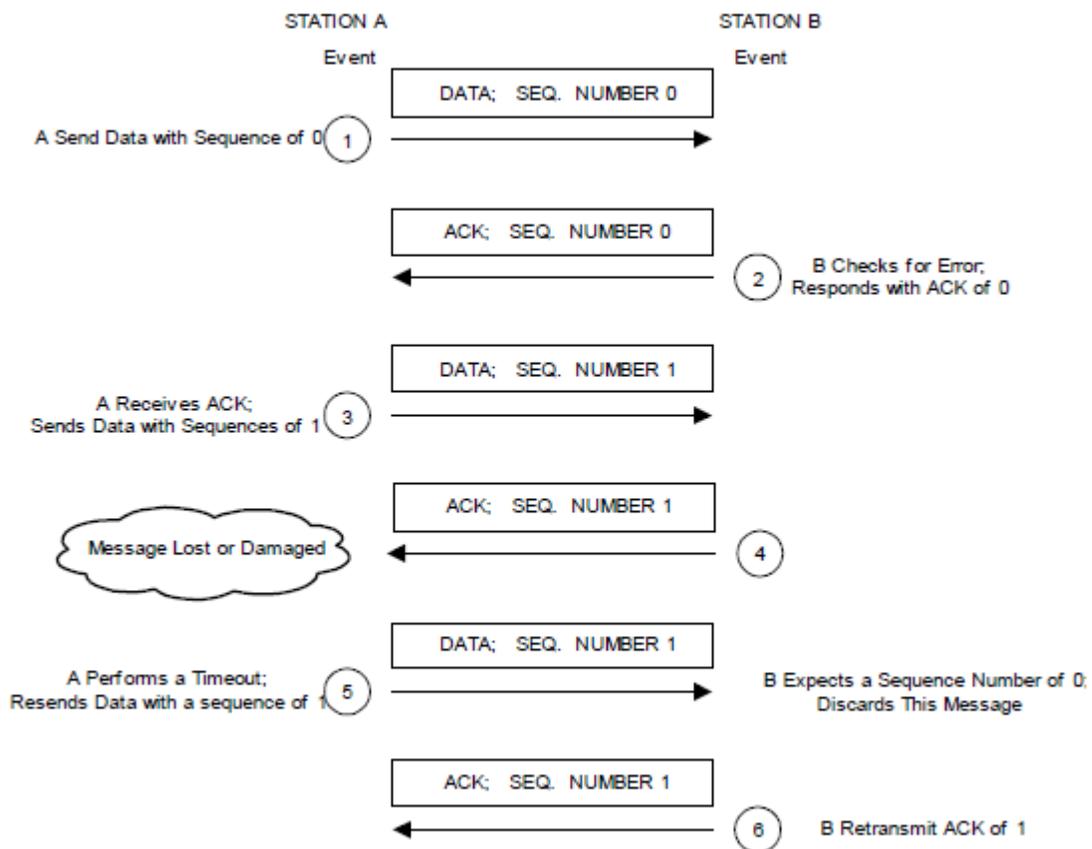
Protokol ini memiliki karakteristik dimana sebuah pengirim mengirimkan sebuah frame dan kemudian menunggu *acknowledgment* sebelum memprosesnya lebih lanjut. Mekanisme *stop and wait* dapat dijelaskan dengan menggunakan gambar di bawah, dimana DLC mengizinkan sebuah message untuk ditransmisikan (event 1), pengujian terhadap terjadinya error dilakukan dengan teknik seperti VCR (*Vertical Redundancy Check*) atau LRC (*Longitudinal Redundancy Check*) terjadi pada even 2 dan pada saat yang tepat sebuah ACK atau NAK dikirimkan kembali untuk ke stasiun pengirim (event 3). Tidak ada messages lain yang dapat ditransmisikan selama stasiun penerima mengirimkan kembali sebuah jawaban. Jadi istilah *stop and wait* diperoleh dari proses pengiriman message oleh stasiun pengirim, menghentikan transmisi berikutnya, dan menunggu jawaban.

Pendekatan *stop and wait* adalah sesuai untuk susunan transmisi *half duplex*, karena dia menyediakan untuk transmisi data dalam dua arah, tetapi hanya dalam satu arah setiap saat. Kekurangan yang terbesar adalah disaat jalur tidak jalan sebagai akibat dari stasiun yang dalam keadaan menunggu, sehingga kebanyakan DLC *stop and wait* sekarang menyediakan lebih dari satu terminal yang on line. Terminal-terminal tetap beroperasi dalam susunan yang sederhana. Stasiun pertama atau host sebagai penanggung jawab untuk peletakkan message diantara terminal-terminal (biasanya melalui sebuah terminal pengontrol yang berada di depannya) dan akses pengontrolan untuk hubungan di bawah dan menjadi masalah yang serius ketika ACK atau NAK hilang dalam jaringan atau dalam jalur. Jika ACK pada event 3 hilang, setelah habis batas waktunya stasiun master mengirim ulang message yang sama untuk kedua kalinya. Transmisi yang berlebihan mungkin terjadi dan menciptakan sebuah duplikasi record pada tempat kedua dari file data pengguna. Akibatnya, DLC harus mengadakan suatu cara untuk mengidentifikasi dan mengurutkan message yang dikirimkan dengan berdasarkan pada ACK atau NAK sehingga harus dimiliki suatu metoda untuk mengecek duplikat message.



Gambar. 5: Stop and wait data link control

Pada gambar di bawah ditunjukkan bagaimana urutan pendeteksian duplikasi message bekerja, pada event 1 stasiun pengirim mengirimkan sebuah message dengan urutan 0 pada headernya. Stasiun penerima menjawab dengan sebuah ACK dan sebuah nomor urutan 0 (event 2). Pengirim menerima ACK, memeriksa nomor urutan 0 di headernya, mengubah nomor urutan menjadi 1 dan mengirimkan message berikutnya (event 3).



Gambar. 6: Stop-and-wait alternating sequence

Stasiun penerima mendapatkan message dengan ACK 1 di event 4. Akan tetapi message ini diterima dalam keadaan rusak atau hilang pada jalan. Stasiun pengirim mengenali bahwa message di event 3 tidak dikenali. Setelah batas waktu terlampaui (*timeout*) stasiun pengirim mengirim ulang message ini (event 5). Stasiun penerima mencari sebuah message dengan nomor urutan 0. Dia membuang message, sejak itu dia adalah sebuah duplikat dari message yang dikirim pada event 3. Untuk melengkapi pertanggungjawaban, stasiun penerima mengirim ulang ACK 1 (event 6).

Efek delay propagasi dan kecepatan transmisi

Kita akan menentukan efisiensi maksimum dari sebuah jalur *point-to-point* menggunakan skema *stop and wait*. Total waktu yang diperlukan untuk mengirim data adalah :

$$T_d = T_I + nT_F$$

dimana T_I = waktu untuk menginisiasi urutan

$$T_I = t_{prop} + t_{poll} + t_{proc}$$

T_F = waktu untuk mengirim satu frame

$$T_F = t_{prop} + t_{frame} + t_{proc} + t_{prop} + t_{ack} + t_{proc}$$

t_{proc} = waktu proses

t_{prop} = waktu propagasi

t_{frame} = waktu pengiriman

t_{ack} = waktu balasan

Untuk menyederhanakan persamaan di atas, kita dapat mengabaikan *Termination* (membatasi koneksi logika (hubungan transmitter-receiver)). Misalnya, untuk sepanjang urutan frame, T_1 relatif kecil sehingga dapat diabaikan. Kita asumsikan bahwa waktu proses antara pengiriman dan penerimaan diabaikan dan waktu balasan frame adalah sangat kecil, sehingga kita dapat mengekspresikan T_d sebagai berikut:

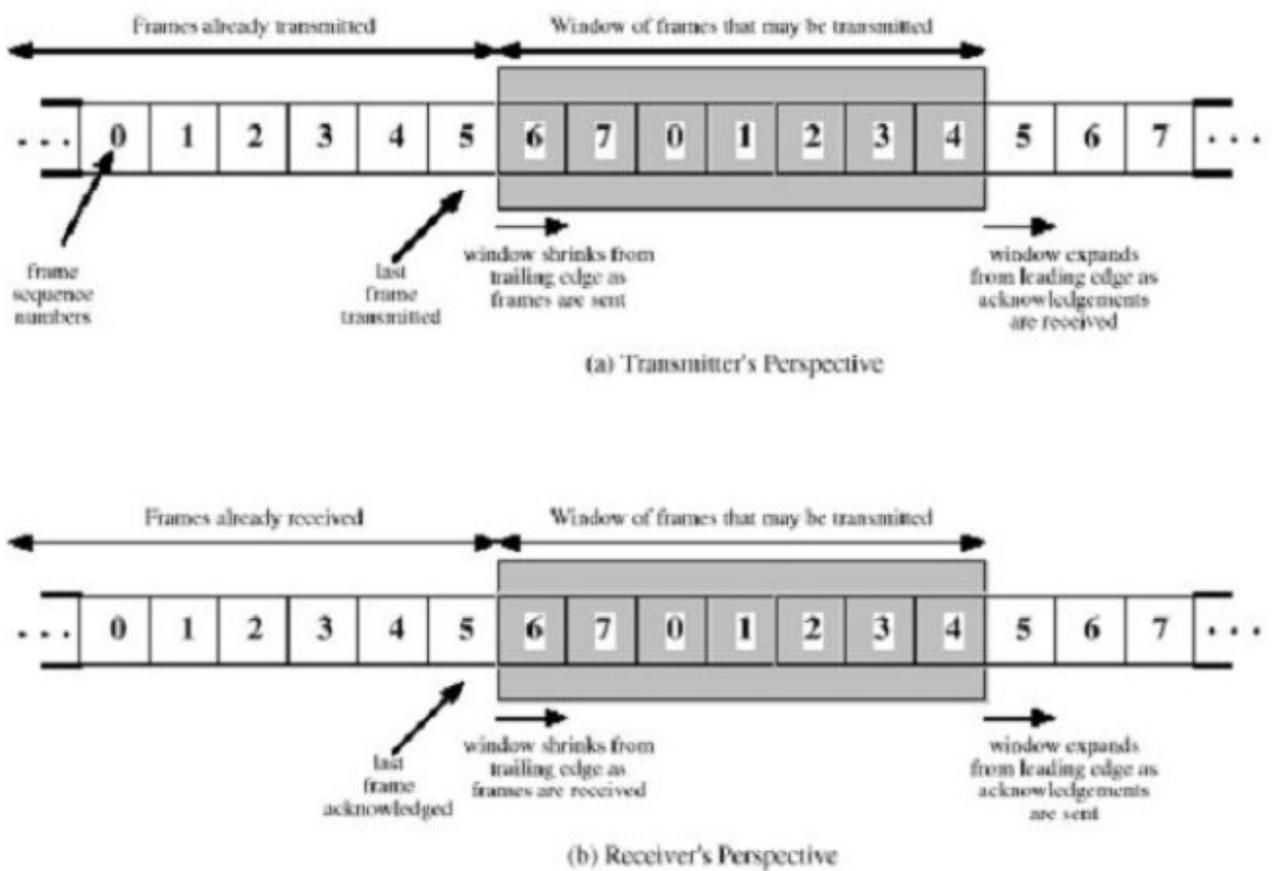
$$T_d = n(2t_{prop} + t_{frame})$$

Dari keseluruhan waktu yang diperlukan hanya $n \times t$ frame yang dihabiskan selama pengiriman data sehingga utilization (U) atau efisiensi jalur diperoleh.

o **Sliding-Window Flow Control**

Masalah utama yang selama ini adalah bahwa hanya satu frame yang dapat dikirimkan pada saat yang sama. Dalam keadaan antrian bit yang akan dikirimkan lebih besar dari panjang frame ($a > 1$) maka diperlukan suatu efisiensi. Untuk memperbesar efisiensi yang dapat dilakukan dengan memperbolehkan transmisi lebih dari satu frame pada saat yang sama. Bila suatu *station A* dan *B* dihubungkan dengan jalur *full-duplex*, *station B* mengalokasikan buffers dengan selebar n frame, yang berarti stasiun B dapat menerima n frame, dan *station A* diperbolehkan untuk mengirim frame sebanyak n tanpa menunggu adanya jawaban. Untuk menjaga jejak dimana frame yang dikirimkan sedang dijawab maka masing-masing jawaban diberi label dengan nomor yangurut. *Station B* menjawab frame dengan mengirimkan jawaban yang dilengkapi nomor urut dari frame berikutnya yang diinginkan. Jawaban ini juga memiliki maksud untuk memberitahukan bahwa *station B* siap untuk menerima n frame berikutnya, dimulai dengan nomor urut yang telah tercantum. Skema ini juga dapat dipergunakan untuk menjawab lebih dari satu frame. Misalnya *station B* dapat jawaban sampai sampai frame ke 4 tiba, dengan kembali jawaban dengan nomor urut 5, *station B* menjawab frame 2, 3, dan 4 pada satu saat. *Station A* memelihara daftar nomor urutan yang boleh dikirim, sedangkan *station B* memelihara daftar nomor urutan yang siap akan diterima. Masing-masing daftar tersebut dapat dianggap sebagai window dari frame, sehingga prinsip kerjanya disebut dengan pengontrol aliran *sliding-window*. Diperlukan untuk dibuat komentar tambahan untuk masing-masing, karena nomor urut yang dipakai menempati daerah didalam frame, komentar tambahan ini dibatasi oleh terbatasnya tempat yang tersedia. Misalnya untuk daerah dengan panjang 3 bit, maka nomor urut jangkauannya antara 0 s/d 7 saja, sehingga frame diberi nomor dengan modulo 7, jadi sesudah nomor urut 7 berikutnya adalah nomor 0. Pada umumnya untuk daerah dengan panjang k -bit, maka jangkauan nomor urut dari 0 sampai dengan $2^k - 1$, dan frame diberi nomor dengan modulo 2^k . Pada gambar dibawah menggambarkan proses *sliding-windows*, dengan diasumsikan nomor urut menggunakan 3 bit sehingga frame diberi nomor urut 0 s/d 7, selanjutnya nomor yang sama dipakai kembali sebagai bagian urutan frame. Gambar segiempat yang diberi bayangan (disebut *window*)

menunjukkan transmitter dapat mengirimkan 7 frame, dimulai dengan frame nomor 7. Setiap waktu frame dikirimkan maka *window* yang digambarkan sebagai kotak dibayangi akan menyusut, setiap waktu jawaban diterima, *window* akan membesar. Ukuran panjang *window* sebenarnya tidak diperlukan sebanyak ukuran maksimumnya untuk diisi sepanjang nomor urut. Sebagai contoh, nomor urut menggunakan 3 bit, stasiun dapat membentuk *window* dengan ukuran 4, menggunakan protokol pengatur aliran *sliding-window*. Sebagai contoh diasumsikan memiliki daerah nomor urut 3 bit dan maksimum ukuran *window* adalah 7 frame. Dimulai dari *station A* dan *B* telah menandai *window* dan *station A* mengirimkan 7 frame yang dimulai dengan frame 0 (F0), sesudah mengirimkan 3 frame (F0, F1, dan F2) tanpa jawaban maka *station A* telah menyusutkan *window*-nya menjadi 4 frame. *Window* menandai bahwa *station A* dapat mengirimkan 4 frame, dimulai dari frame nomor 3 selanjutnya *stasiun B* mengirim *receive-ready* (RR) yang berarti semua frame telah diterima sampai frame nomor 2 dan selanjutnya siap menerima frame nomor 3, tetapi pada kenyataannya disiapkan menerima 7 frame, dimulai frame nomor 3. *Station A* terus mengirimkan frame nomor 3, 4, 5, dan 7, kemudian *station B* menjawab RR7 sebagai jawaban dari semua frame yang diterima dan mengusulkan *station A* mengirim 7 frame, dimulai frame nomor 7.

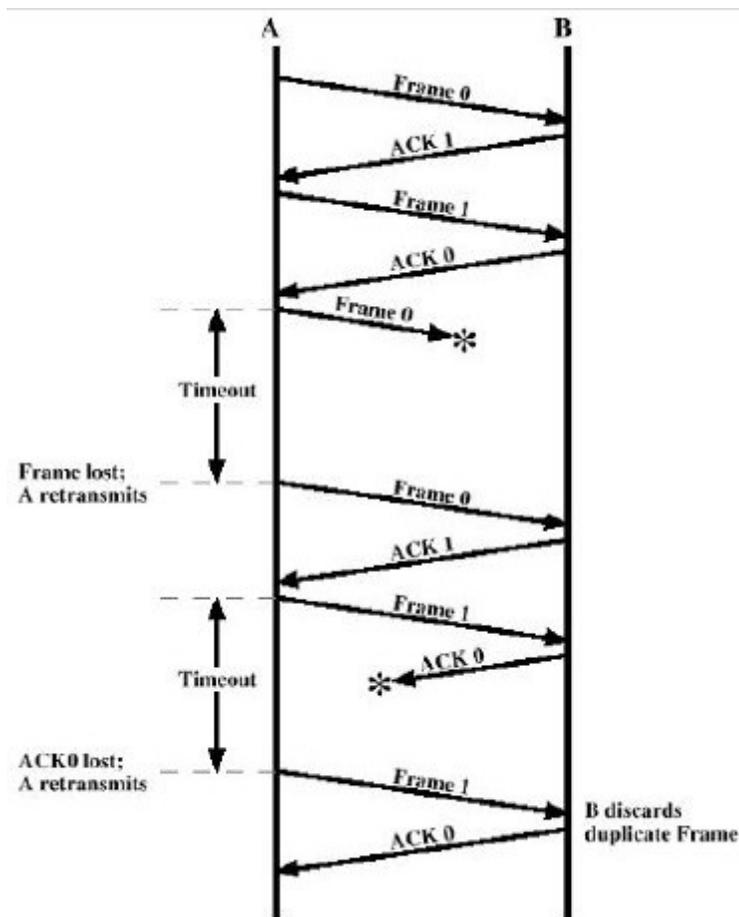


Gambar. 7: *Skema Aliran Sliding-Window*

Receiver harus dapat menampung 7 frame melebihi satu jawaban yang telah dikirim, sebagian besar protokol juga memperbolehkan suatu station untuk memutuskan aliran frame dari sisi (arah) lain dengan cara mengirimkan pesan *receive-not-ready* (RNR), yang dijawab frame terlebih dulu, tetapi melarang transfer frame berikutnya. Bila dua stasiun saling bertukar data (dua arah) maka masing-masing perlu mengatur dua

Stop and Wait ARQ

Berdasarkan pada teknik flow control stop and wait, dan digambarkan dalam gambar dibawah. 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. 8: 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.

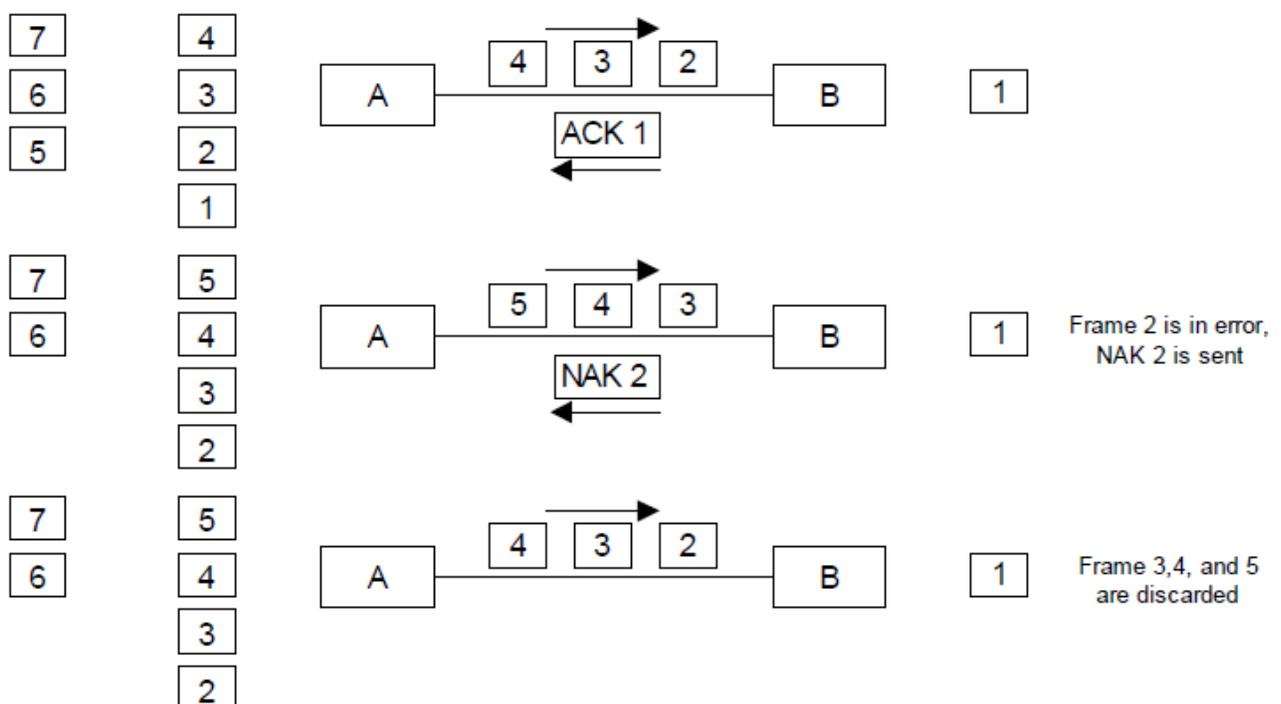
Go Back N ARQ

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 :

- o 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 NAKi.
 - 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.



Gambar. 9: Go-back-N ARQ

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 $2n-1$ sedangkan selective -reject $2n$.

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.

Referensi:

<http://teknik-informatika.com/flow-control/>

<http://missa.staff.gunadarma.ac.id/>

<http://www.rijalfadilah.wordpress.com>

http://www.csi.ucd.ie/Staff/jmurphy/networks/csd8_4-datalink_2.pdf

http://ocw.mit.edu/NR/rdonlyres/Aeronautics-and-Astronautics/16-36Communication-Systems-EngineeringSpring2003/7D64F769-4B55-4184-B8B0-3800DFB7DC31/0/lec15_16.pdf