Data Communication

#13 Error Correction

Susmini I. Lestariningati, M.T

Error Correction

- Error correction may generally be realized in two different ways:
  - **Automatic repeat request (ARQ)**
    - (sometimes also referred to as backward error correction): This is an error control technique whereby an error detection scheme is combined with requests for retransmission of erroneous data. Every block of data received is checked using the error detection code used, and if the check fails, retransmission of the data is requested – this may be done repeatedly, until the data can be verified.
  - **Forward error correction (FEC):**
    - The sender encodes the data using an error-correcting code (ECC) prior to transmission. The additional information (redundancy) added by the code is used by the receiver to recover the original data. In general, the reconstructed data is what is deemed the "most likely" original data.
  - ARQ and FEC may be combined, such that minor errors are corrected without retransmission, and major errors are corrected via a request for retransmission: this is called hybrid automatic repeat-request (HARQ).

Forward Error Correction (FEC)

- Any error-correcting code can be used for error detection. A code with minimum Hamming distance, d, can detect up to d – 1 errors in a code word. Using minimum-distance-based error-correcting codes for error detection can be suitable if a strict limit on the minimum number of errors to be detected is desired.
- Codes with minimum Hamming distance d = 2 are degenerate cases of error-correcting codes, and can be used to detect single errors. The parity bit is an example of a single-error-detecting code.

Error Control

- Berfungsi untuk mendeteksi dan memperbaiki eror-eror yang terjadi dalam transmisi frame-frame.
- Ada 2 tipe eror yang mungkin:
  - Frame hilang : suatu frame gagal mencapai sisi yang lain
  - Frame rusak : suatu frame tiba tetapi beberapa bit-bitnya eror

Hamming Code

- Kode Hamming merupakan kode non-trivial untuk koreksi kesalahan yang pertama kali diperkenalkan.
- Kode ini dan variansinya telah lama digunakan untuk kontrol kesalahan pada sistem komunikasi digital.
- Kode Hamming biner dapat direpresentasikan dalam bentuk persamaan:

$$\mathbf{(n,k) = (2^m-1, 2^m-1-m}$$

- Contoh:

  jika m = jumlah paritas = 3

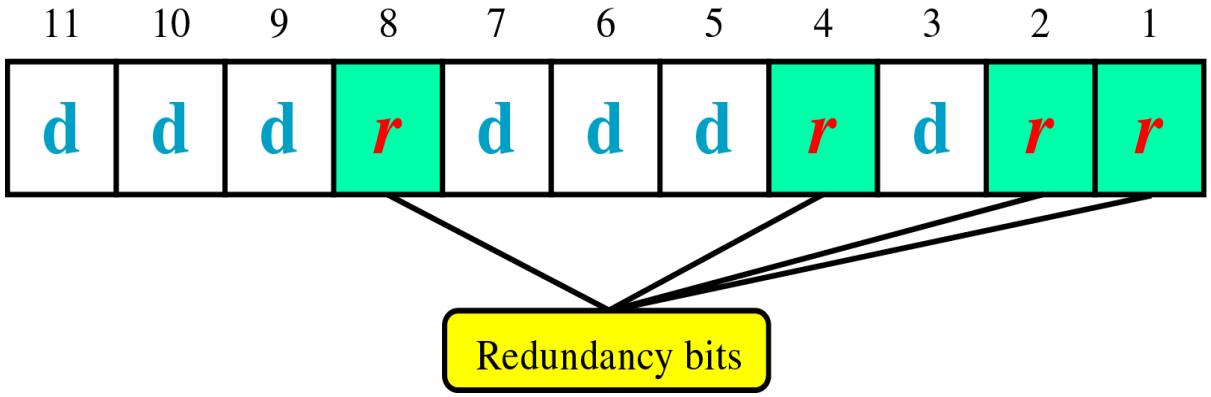
  k = jumlah data = 4

  n = jumlah bit informasi yang membentuk n sandi

  = 7

  maka kode Hamming nya adalah C (7,4) dengan dmin =3

- Error Correcting codes dinyatakan sebagai penerusan koreksi kesalahan untuk mengindikasikan bahwa pesawat penerima sedang mengoreksi kesalahan.
- Kode pendeteksi yang paling banyak digunakan merupakan kode Hamming.
- Posisi bit-bit Hamming dinyatakan dalam 2^n dengan n bilangan bulat sehingga bit-bit Hamming akan berada dalam posisi 1, 2, 4, 8, 16, dst..



Langkah-langkah Kode Hamming

- Tandai semua posisi bit 2^n sebagai bit redundancy (yaitu posisi 1, 2, 4, 8, 16, 32, 64, …)
- Posisi bit sisanya selain no 1 diatas adalah posisi bit yang akan dipakai (yaitu posisi 3, 5, 6, 7, 9, 10, 11, 12, 13, 14, 15, 17, …)
- Setiap bit paritas: (no.1)
  - Posisi 1 (r1): periksa tiap satu bit dari bit pertama, lalu lompati tiap satu bit, periksa 1 bit berikutnya, lompati 1 bit, dan seterusnya. (1, 3, 5, 7, 9, 11, 13, 15, 17, …)
  - Posisi 2 (r2): periksa tiap dua bit dari bit kedua, lompati 2 bit, periksa lagi bit berikutnya dan seterusnya (2, 3, 6, 7, 10, 11, 14, 15, …)
  - Posisi 4 (r4): periksa tiap 4 bit dari bit keempat, lompati 4 bit berikutnya, cek 4 bit, dan seterusnya (4,5,6,7,12,13,14,15,20,21,22,23, …)
  - Posisi 8 (r8): periksa setiap 8 bit dari bit kedelapan, lompati 8 bit berikutnya, periksa 8 bit berikutnya dan seterusnya (8-15, 24-31, 40-47, …)
  - Posisi 16 (r16): periksa setiap 16 bit dari bit ke-enambelas, lompati 16 bit berikutnya, periksa lagi 16 bit, dan seterusnya (16-31, 48-63, 80-95, …)
  - Posisi 32 (r32): periksa setiap 32 bit dari bit keenambelas, lompati 32 bit berikutnya, periksa lagi 32 bit, dan seterusnya (32-63, 96-127, 160-191, …)
  - Set bit paritas 1 jika total bit 1 ganjil, set bit paritas 0 jika jumlah bit 1 nya adalah genap.

Poses Bit Hamming

Bit position		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	
Encoded data bits		p1	p2	d1	p3	d2	d3	d4	p4	d5	d6	d7	d8	d9	d10	d11	p5	d12	d13	d14	d15	
Parity bit coverage	p1	X		X		X		X		X		X		X		X		X		X		
	p2		X	X			X	X			X	X			X	X			X	X		...
	p3				X	X	X	X					X	X	X	X					X	
	p4								X	X	X	X	X	X	X	X						
	p5																X	X	X	X	X	

Example 1

A byte of data: 10011010

Create the data word, leaving spaces for the parity bits: _ _ 1 _ 0 0 1 _ 1 0 1 0

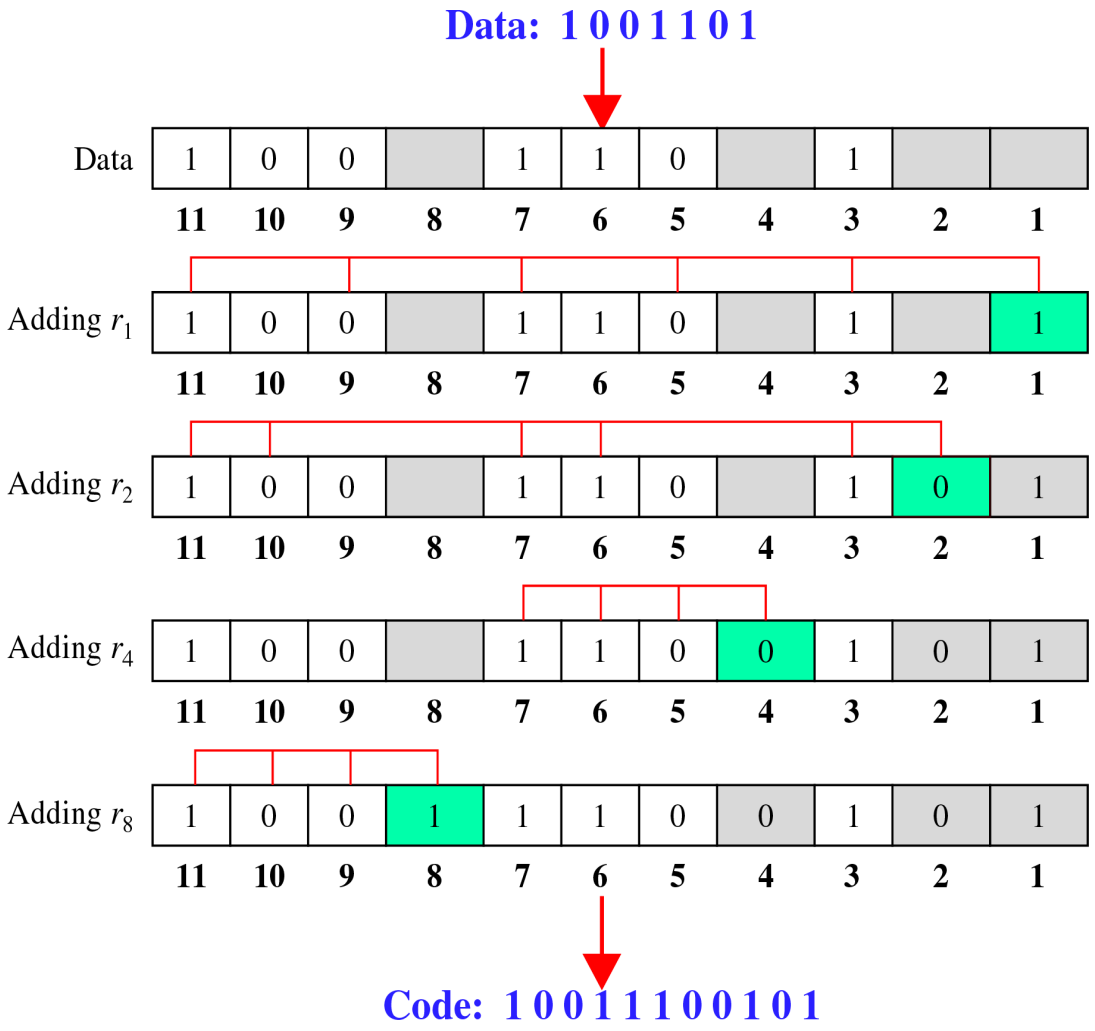
Calculate the parity for each parity bit (a ? represents the bit position being set):

- Position 1 checks bits 1, 3, 5, 7, 9, 11:
  ? _ **1** _ **0** 0 **1** _ **1** 0 **1** 0. Even parity so set position 1 to a 0: 0 _ 1 _ 0 0 1 _ 1 0 1 0
- Position 2 checks bits 2, 3, 6, 7, 10, 11:
  0 **? 1** _ 0 **0 1** _ 1 **0 1** 0. Odd parity so set position 2 to a 1: 0 1 1 _ 0 0 1 _ 1 0 1 0
- Position 4 checks bits 4, 5, 6, 7, 12:
  0 1 1 **? 0 0 1** _ 1 0 1 **0**. Odd parity so set position 4 to a 1: 0 1 1 1 0 0 1 _ 1 0 1 0
- Position 8 checks bits 8, 9, 10, 11, 12:
  0 1 1 1 0 0 1 **? 1 0 1 0**. Even parity so set position 8 to a 0: 0 1 1 1 0 0 1 0 1 0 1 0

- Code word: **0 1** 1 **1** 0 0 1 **0** 1 0 1 0.

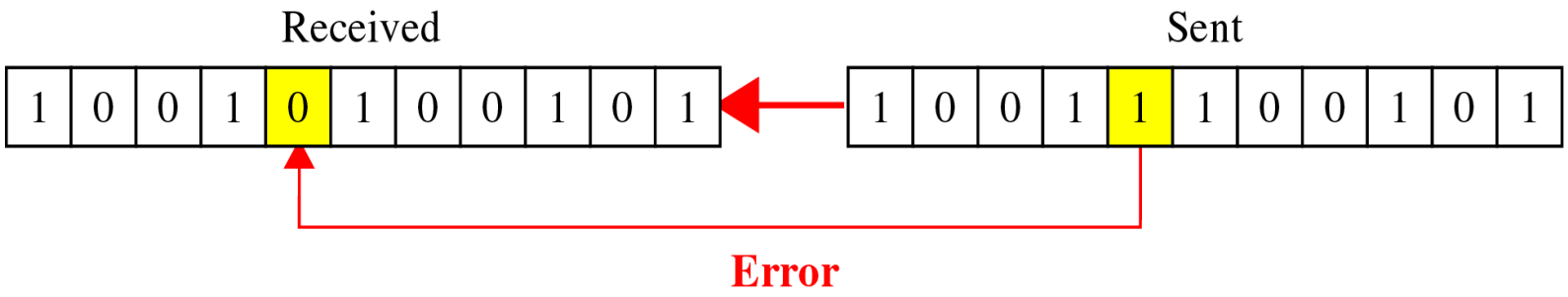
Example 2

- Example of Hamming Code

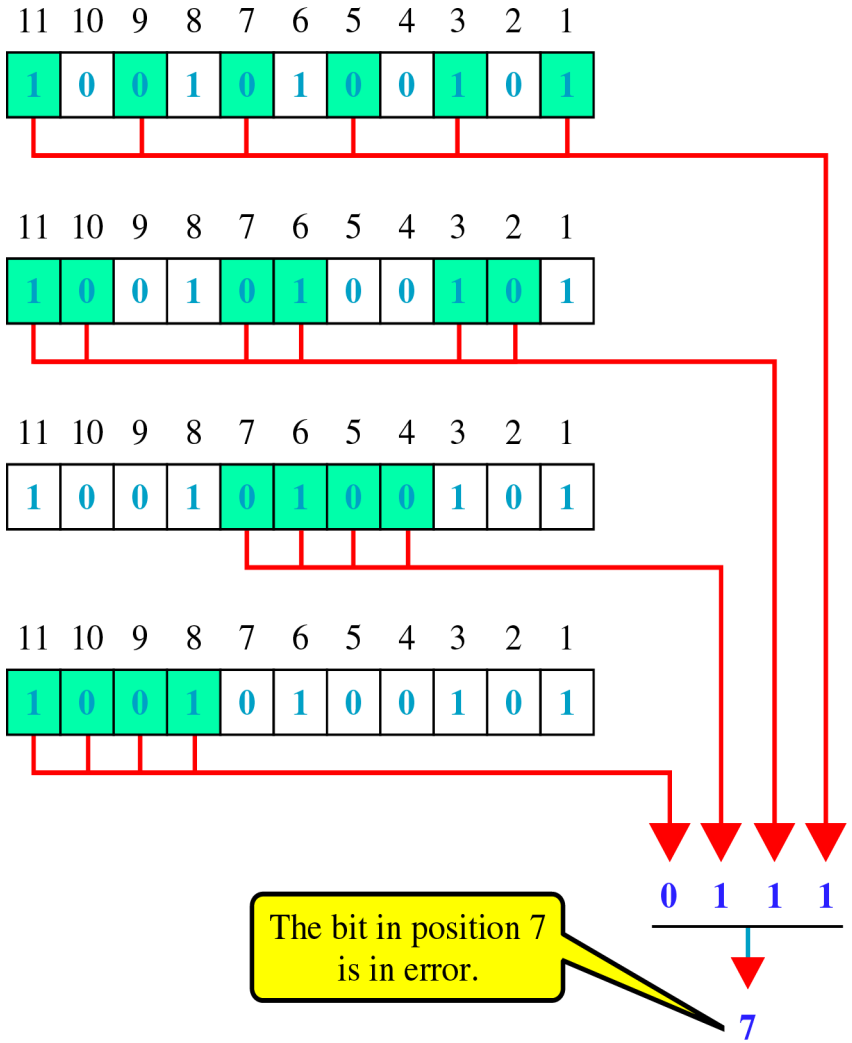


Example 3

- Jika bit yang dikirim dan sampai di penerima mengalami error pada bit tertentu, hitung dengan menggunakan kode Hamming dimana posisi bit yang mengalami kerusakan!



- Dengan menggunakan kode Hamming

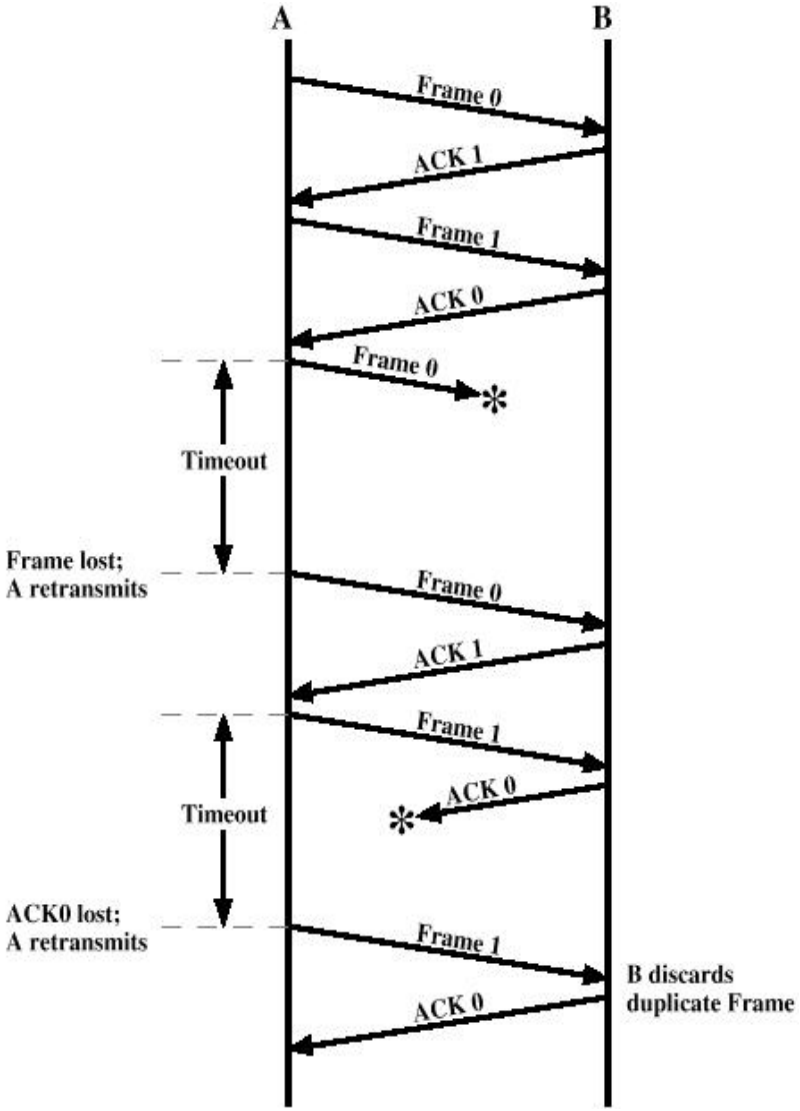


Automatic Repeat Request (ARQ)

- Terdapat 3 versi dari ARQ, yaitu:
  - Stop and wait ARQ
  - Go Back N ARQ
  - Selective Repeat ARQ

Stop and Wait ARQ

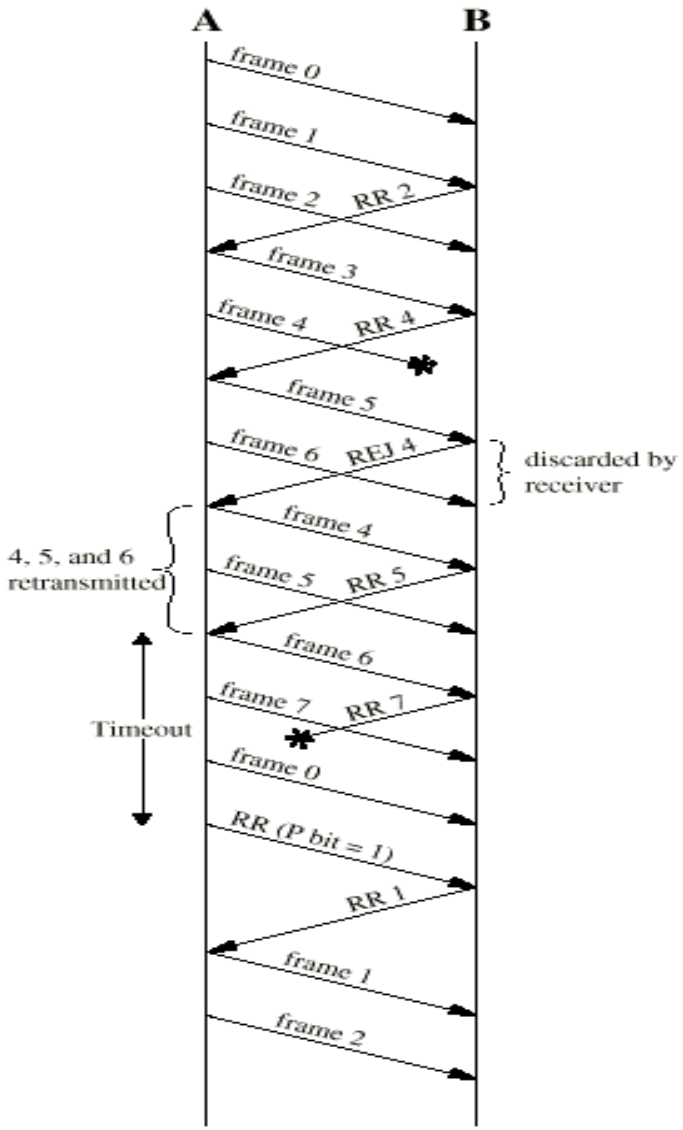
- Stasiun sumber mentransmisi suatu frame tunggal dan kemudian harus menunggu suatu acknowledgment (ACK) dalam periode tertentu. Tidak ada data lain dapat dikirim sampai balasan dari stasiun tujuan tiba pada stasiun sumber. Bila tidak ada balasan maka frame ditransmisi ulang. Bila error dideteksi oleh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



- 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 mengindikasi bahwa receiver siap untuk frame 0. Sedangkan ACK1 mengakui menerima frame 0 dan mengindikasi bahwa receiver siap untuk frame 1.

Go Back N ARQ

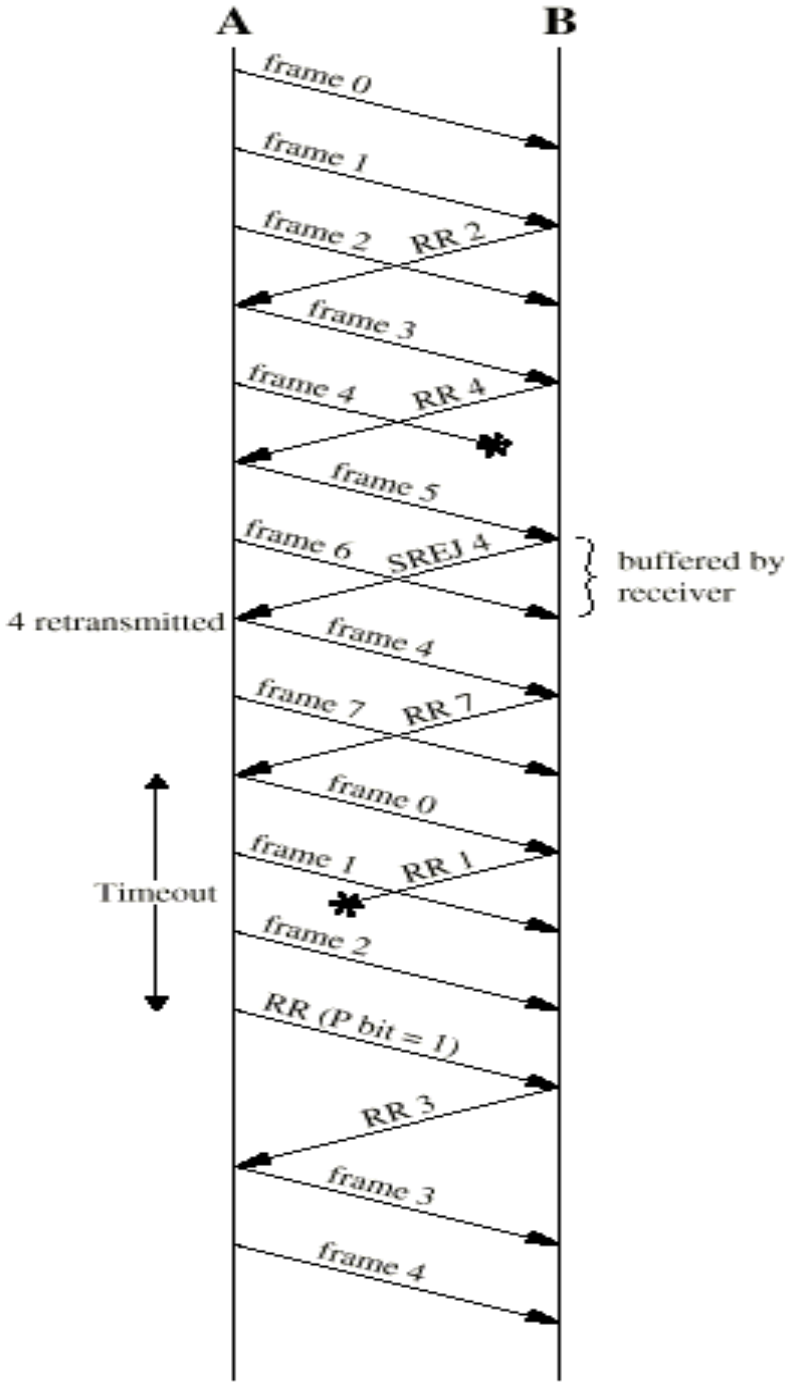
- Teknik Go-back-N ARQ yang terjadi dalam beberapa kejadian :
- Frame yang rusak. Ada 3 kasus :
  - A mentransmisi frame i. B mendeteksi suatu error dan telah menerima frame (i-1) secara sukses. B mengirim A NAKi, mengindikasi bahwa frame i ditolak. Ketika A menerima NAK ini, maka harus mentransmisi 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 dikomulatif (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 Repeat

- 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 -repeat 2^n.



- Skenario dari teknik ini untuk 3 bit penomoran yang mengizinkan ukuran window sebesar 7 :
  - Stasiun A mengirim frame 0 sampai 6 ke stasiun B.
  - Stasiun B menerima dan mengakui ketujuh frame-frame.
  - Karena noise, ketujuh acknowledgment hilang.
  - Stasiun A kehabisan waktu dan mentransmisi ulang frame 0.
  - 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.

Applications

- Applications that require low latency (such as telephone conversations) cannot use Automatic Repeat reQuest (ARQ); they must use forward error correction (FEC). By the time an ARQ system discovers an error and re-transmits it, the re-sent data will arrive too late to be any good.
- Applications where the transmitter immediately forgets the information as soon as it is sent (such as most television cameras) cannot use ARQ; they must use FEC because when an error occurs, the original data is no longer available. (This is also why FEC is used in data storage systems such as RAID and distributed data store).
- Applications that use ARQ must have a return channel; applications having no return channel cannot use ARQ. Applications that require extremely low error rates (such as digital money transfers) must use ARQ. Reliability and inspection engineering also make use of the theory of error-correcting codes.