❖ DATA LINK LAYER DESIGN ISSUES :

- Datalink Layer is the second layer in OSI Reference model.
- DLL Supervises the flow of information between adjacent nodes.
- The data link layer uses the services of the physical layer to send and receive bits over communication channels.

It has a number of functions, including:

1. Providing a well-defined service interface to the network layer.

2. Dealing with transmission errors.

3. Regulating the flow of data so that slow receivers are not swamped by fast senders.

Some of the important functions of Data link layer includes well defined service interfaces to the network layer , framing , flow of control, error detection and error control, frame formatting and sequencing.

All these are important functions for reliable communication and plays an important role in designing data link layer.

Fig. 3-1. Frame management forms the heart of what the data link layer does. In the following sections we will examine all the above mentioned issues in detail.

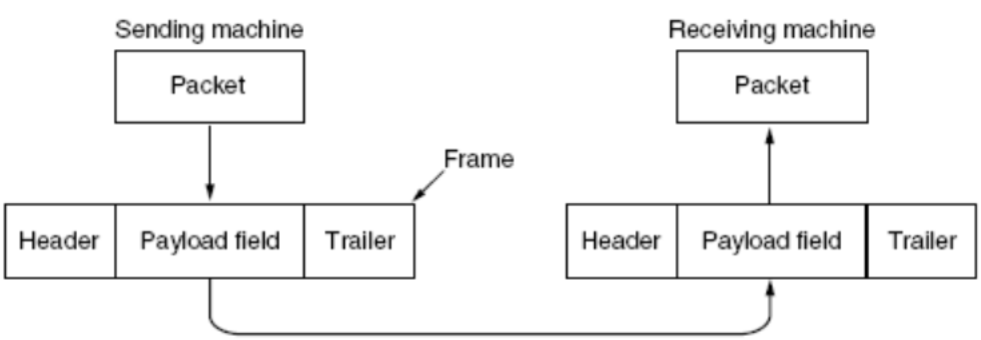


Figure 3-1. Relationship between packets and frames.

1. Services provided to the Network layer:

- The primary function of the data link layer is to **provide services to the network layer.**
- The principal service is **transferring data from the network layer on the source machine to the network layer on the destination machine.**
- The **two Datalink layer communicates with each other by Datalink control protocols.**
- Following are the important services provided by the Data link layer to the Network layer.

1. **Unacknowledged connectionless service.**

2. **Acknowledged connectionless service.**

3. **Acknowledged connection-oriented service.**

1. Unacknowledged connectionless service:

- It is un acknowledgement service of transmission.
- Here the source machine **sends the data to the destination machine without any acknowledgement.**
- For this **no connection is either established or released.**
- If data is loss due to noise or interrupts the lost **data is not even recovered by the layer.**

2. Acknowledged connectionless service:

- It is acknowledgement connection less service of transmission.
- Here **each data frame is acknowledged by the destination machine.**
- If any **data frame is lost or not arrived in time the same message can be transmitted again.**
- In this service no connections are used.

3. Acknowledged connection oriented service:

- It is acknowledgement connection oriented service of transmission.
- Here the **connection is established for data transmission.**
- Each frame is **numbered before transmission and corresponding acknowledgement is also received.**
- Here connection is established before transmission of data.

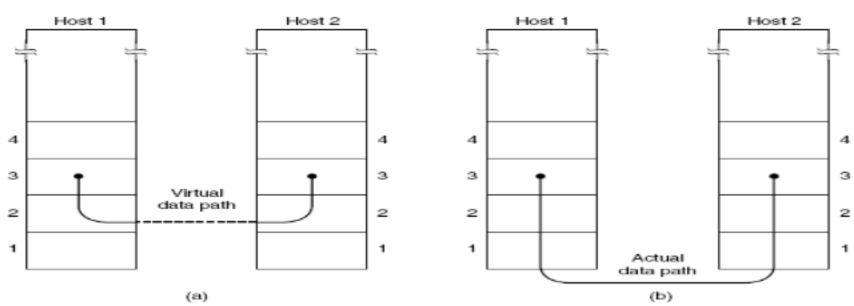


Figure 3-2. (a) Virtual communication. (b) Actual communication.

Fig. 3-2(a). The actual transmission follows the path of Fig. 3-2(b), but it is easier to think in terms of two data link layer processes communicating using a data link protocol.

2 Framing:

- "Breaking of the bit stream by inserting spaces or time gaps is called framing".
- To service the network layer , data link layer uses the service provided to the physical layer.
- The data link layer encapsulates each data packet from the network layer into frames that are then transmitted.

A frame has three parts, namely −

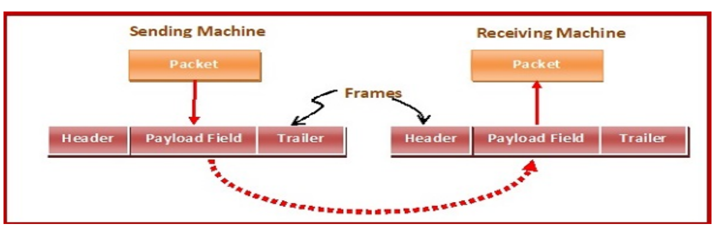
- **Frame Header**
- **Payload field that contains the data packet from network layer**
- **Trailer**

Frame Header: It contains the source and the destination addresses of the frame and the control bytes.

Payload field: It contains the message to be delivered.

Trailer: It contains the error detection and error correction bits (damaged or not). It is also called a Frame Check Sequence (FCS).

Flag: Two flag at the two ends mark the beginning and the end of the frame.



- Physical layer accepts the bit stream and deliver to the destination.
- This bit stream (sequence of bits) may contain errors i.e. number of bits received may not be equal to the number of bits transmitted.
- The data link layer breaks the stream into discrete frames (individual frames) and compute the check sum for each frame.
- So it is difficult and risky to count each frame , so by using the following methods to overcome

I. **Character count**

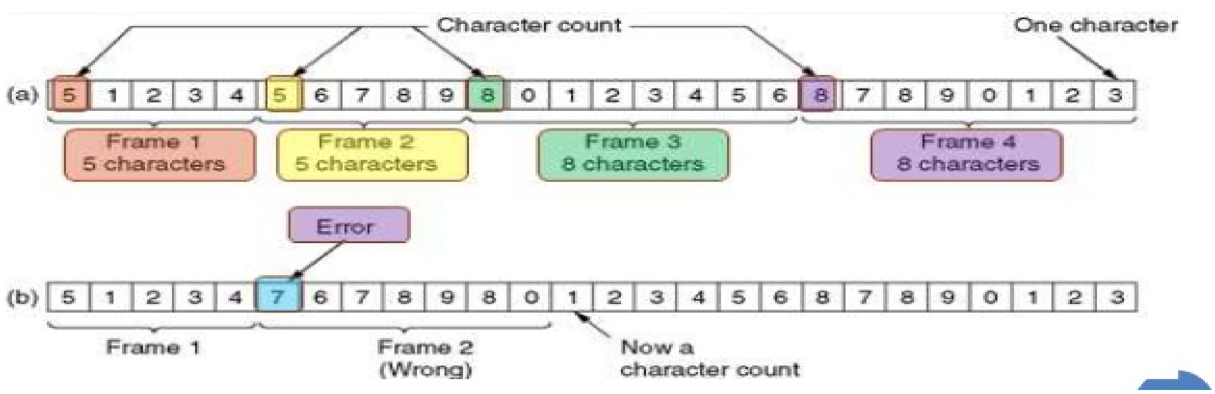
2. Starting and Ending characters with character stuffing.

3. Starting and Ending flags with bit stuffing.

1. Character count:

- The first framing method uses a field in the header to specify the number of characters in the frame. When the data link layer at the destination sees the character count, it knows how many characters follow and hence where the end of the frame is.
- This technique is shown in Fig for four frames of sizes 5, 5, 8, and 8 characters, respectively.
- Figure. A character stream. (a) Without errors. (b) With one error.

Dis advantage For example, if the character count of 5 in the second frame becomes a 7.



2. Starting and Ending characters with character stuffing:

- Each frame starts and ends with a special byte (i.e. FLAG byte).
- If already FLAG byte occurs in the frame Stuff (added) an Extra byte (i.e. ESC).
- If Esc byte is there in the frame add one more Esc in the frame.
- If it consist of both Flag and Esc add one more Esc and Flag bytes in the given frame.

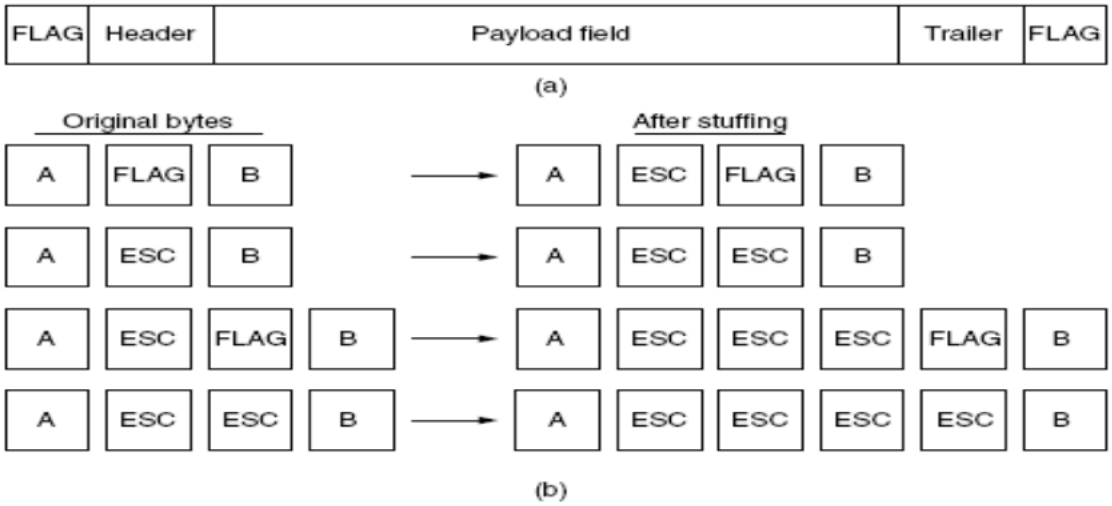
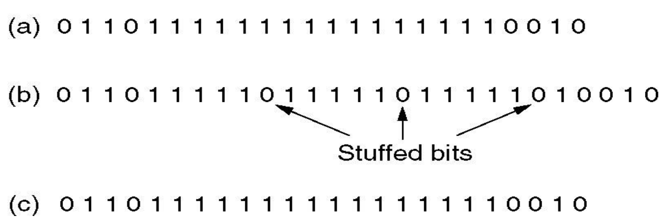


Figure 3-4. (a) A frame delimited by flag bytes. (b) Four examples of byte sequences before and after byte stuffing.

3.Bit Stuffing:

In bit stuffing a specific bit is stuffed into the out going character stream.
In this example if any 5 consecutive 1 s are encountered in the data it is automatically stuffs a 0 bit is replaced in the frame.

Each frame begins &Ends a Special bit pattern 01111110 when even sender sees 5 consecutive 1 s it stuffs 0.



Bit stuffing

(a) The original data.

(b) The data as they appear on the line.

(c) The data as they are stored in receiver's memory after de stuffing.

3.Error Control:

- Only technique in error control is Retransmission.
- To ensure proper sequencing and safe delivery of frames at the destination , An Acknowledgement should be sent by the destination network.
- The receiver sends back positive or negative acknowledgements about incoming frames.
- If the sender receives a positive Acknowledgement it means the frame has arrived safely.
- If the sender receives a Negative Acknowledgement it means the some thing wrong in that frame.
- Phases of Error control
  - Error Detection (find the errors).
  - Acknowledgement (ACK).
    - Positive ACK
    - Negative ACK.
    - Retransmission.

4. Flow Control:

- When **sender is running on Fast machine and Receiver is on slow receiver (overloaded data) some data may loss.**
- To solved the problem of marking the start and end of each frame how to make sure all frames are eventually **delivered to the network layer at the destination and in the proper order.**
- **Both sender and receiver maintain the same speed i.e. same capabilities.**
- This situation can occur when the sender is running on a fast, powerful computer and the receiver is running on a slow, low-end machine.
- Clearly to prevent this situation **Two approaches are commonly used.**

1). In the first one, **feedback-based flow control**, the receiver sends back information to the ***sender giving it permission to send more data***, or at **less data** telling the sender how the receiver is doing.

2). In the second one, **rate-based flow control**, the protocol has a built-in mechanism that **limits the data transfer rate at which senders may transmit data.**

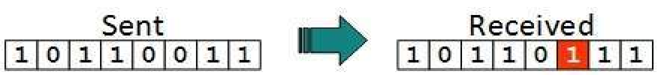
❊Error Correction and Error Detection:

- During transmission of data from the source to a receiver many factors like noise, change some part of data , many types of interrupts in the data.
- So **system to detect and correct such errors.by using these two techniques.**
- **"Error detection"** techniques is ability to **detecting** such **errors due to** causes noise or interrupts etc..
- **"Error Correction"** techniques is enables reconstruction of the original data in many cases.
- **Redundancy :** Error detection uses the concept of Redundancy . Redundancy means adding extra bits for Detecting errors.

Types of Errors:

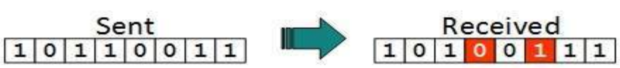
There may be three types of errors:

- **Single bit error :**



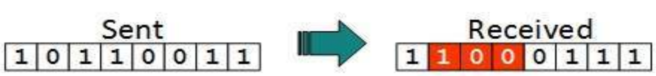
In a frame, there is only one bit, anywhere though, which is corrupt.

- **Multiple bits error:**



Frame is received with more than one bits in corrupted state.

- **Burst error:**



Frame contains more than one consecutive bits corrupted.

Basic approach used for error detection is the use of redundancy bits, where **additional bits are added to facilitate detection of errors.**

Error Detection techniques are:

1. Parity check (1. Simple parity , 2. Two - Dimensional parity).
2. Checksum.
3. Cyclic redundancy check (CRC).

Parity :

Simplest form of error detection is to append a single bit called Parity bit to a string of data. There are 2 forms of parity check methods.

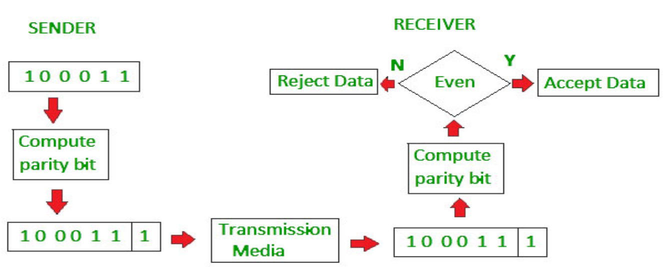
1. **Simple parity**
2. **Two - Dimensional parity**

1.Simple parity:

Blocks of data from the source are subjected to a check bit or parity bit generator form, where a parity of :

- 1 is added to the block if it contains odd number of 1's, (Odd Parity)
- 0 is added if it contains even number of 1's. (Even parity)

This scheme makes the total number of 1's even, that is why it is called even parity checking.



Error Detection –examples:

Example 1 :

Suppose the sender wants to send the word *world*. In ASCII the five characters are coded as

1110111 1101111 1110010 1101100 1100100

The following shows the actual bits sent

11101110 11011110 11100100 11011000 11001001

Example 2 :

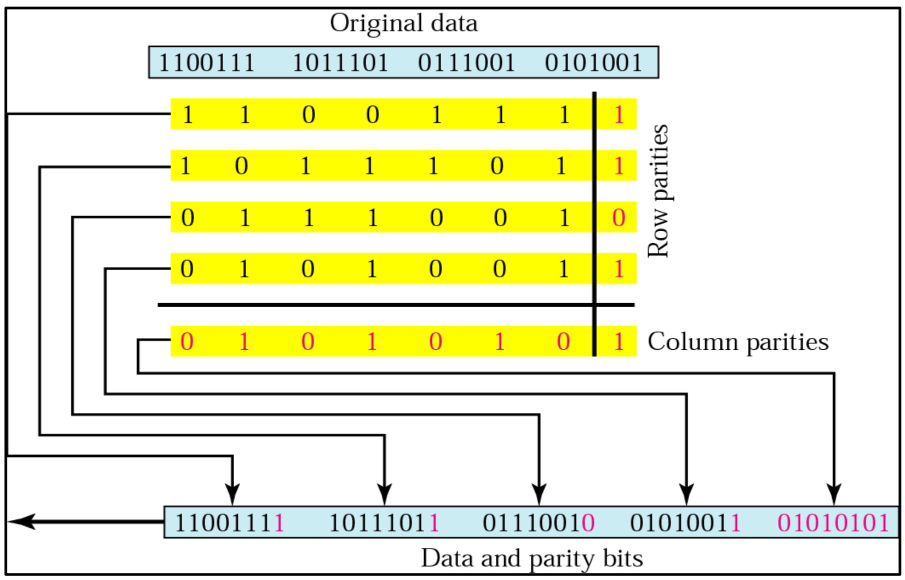
Now suppose the word is corrupted during transmission.

1111110 1101110 1110100 1011000 1100100

The receiver counts the 1s in each character and comes up with even and odd numbers (7, 6, 5, 4, 4). The receiver knows that the data are corrupted, discards them, and asks for retransmission.

2. Two –Dimensional Parity Check:

- In this 2 dimensional parity checking a block of bits is organized in **Rows and Columns (Table).**
- Then parity of each unit is calculated.
- Calculate parity bits for each row and column and attach to the original data and send to the receiver.



2. Check sum:

For error detection by checksums, data is divided into fixed sized frames or segments.

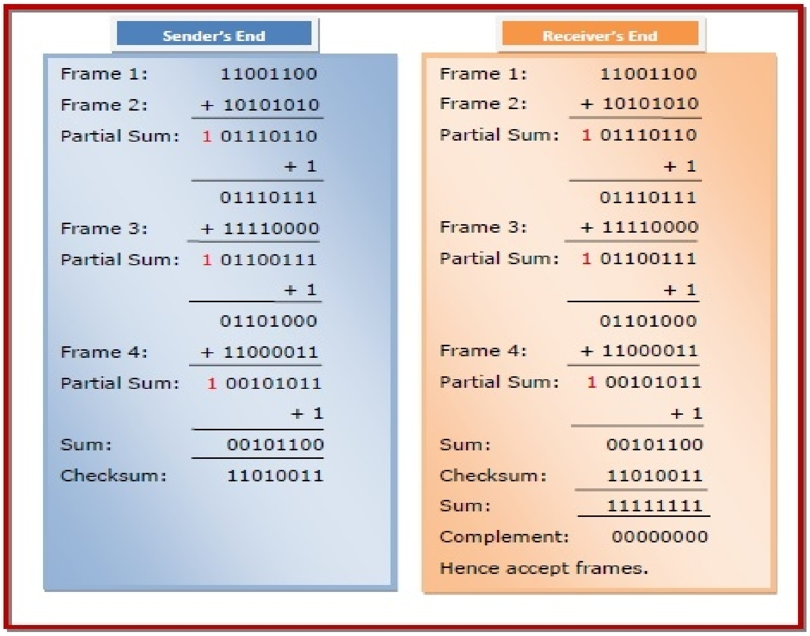
- Sender's End − The sender adds the segments using 1's complement arithmetic to get the sum. It then complements the sum to get the checksum and sends it along with the data frames.
- Receiver's End − The receiver adds the incoming segments along with the checksum using 1's complement arithmetic to get the sum and then complements it.

If the result is zero, the received frames are accepted; otherwise they are discarded.

Example:

Suppose that the sender wants to send 4 frames each of 8 bits, where the frames are 11001100, 10101010, 11110000 and 11000011.

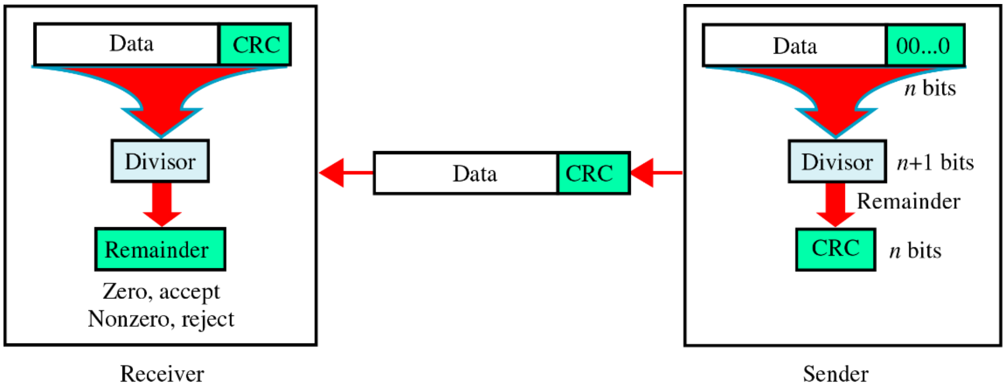
- The sender adds the bits using 1s complement arithmetic. While adding two numbers using 1s complement arithmetic, if there is a carry over, it is added to the sum.
- After adding all the 4 frames, the sender complements the sum to get the checksum, 11010011, and sends it along with the data frames.
- The receiver performs 1s complement arithmetic sum of all the frames including the checksum. The result is complemented and found to be 0. Hence, the receiver assumes that no error has occurred.



3. CRC (Cyclic Redundancy Check): **Multiple bits verifies(brust errors)**

- Data * CRC Generator (find the redundant bits) * CRC bits or redundant (added) bits.
- CRC Generator (**n** bits)
- CRC Bits (n-1 bits)
- CRC is based on binary division instead of addition bits as in parity check.
- In CRC a sequence of redundant bits (CRC Remainder) is appended to the data unit.
- So that it is exactly divisible by a second number .
- At the destination end the received data is divided by same number .

- If any remainder is generated if it is all 0's Success other wise Rejected . xor



Example:

- Data 100100
- CRC generator 1101
- CRC bits (n-1) 4-1=3 (000)initially
- Division XOR

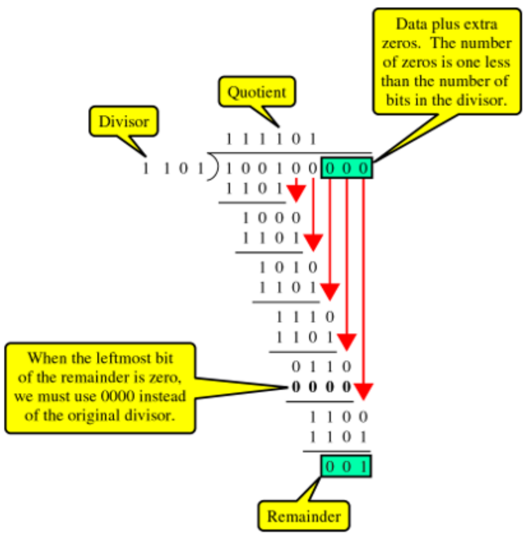
- Data : Polynomial expression

10100111

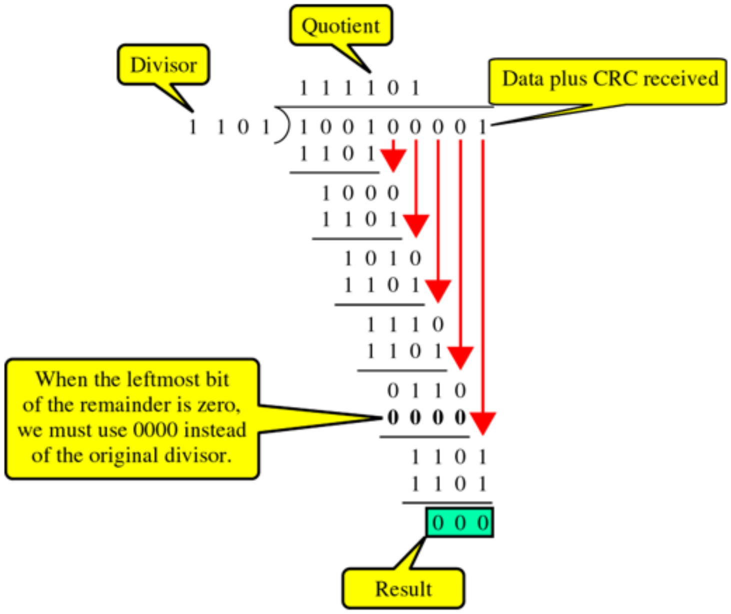
$$x^7 + x^5 + x^2 + x + 1$$

Transmitted Frame: 100100 001

- **Sender Data**



Receiver side verifies



❉ Elementary Data link protocols:

Protocols in the data link layer are designed so that this layer can perform its basic functions:

- Framing
- Error control and
- Flow control.

"Framing" is *the process of dividing bit - streams from physical layer into data frames .*

"Error control" *mechanisms deals with transmission errors and retransmission of corrupted and lost frames.*

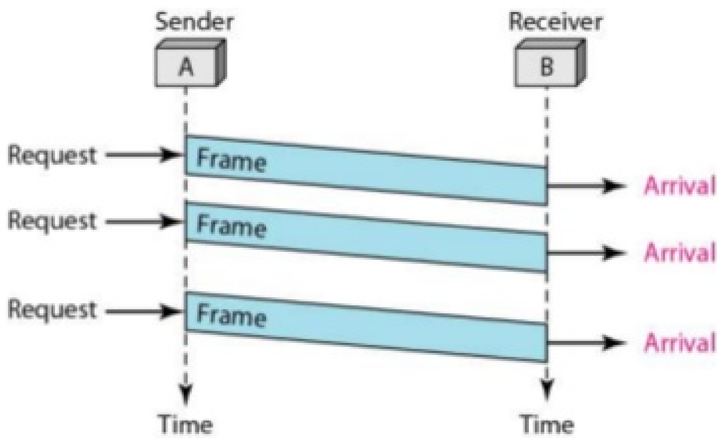
"Flow control" *regulates speed of delivery and so that a fast sender does not drown a slow receiver.*

<u>Types of Data Link Protocols :</u>

1. simplex protocol.
2. Simplex stop – and – wait protocol.
3. Simplex protocol for Noisy channel

❖ 1.Simplex protocol :

- Data are transmitted in one direction only.
- Both transmitting and receiving are always ready state.
- Processing time can be ignored.
- Infinite buffer space available.
- The communication channel b/w sender and receiver never damage or never loss frames.
- No errors occur i.e. No frames damaged and No data loss.
- Advantages : simple protocol.
- Dis Advantage : If Continuous Data Transmits (**flooding** used for flow control) to the receiver congestions may occur data may loss.



❖ 2. Simplex Stop and wait protocol :

- One directional flow of data from sender to receiver.
- Receiver has only finite buffer capacity.
- Communication is error free communication.(no data loss).
- Receiver having finite processing speed.
- Protocols in which the sender sends one frame and then waits for an acknowledgement

before processing is called stop and wait protocol.

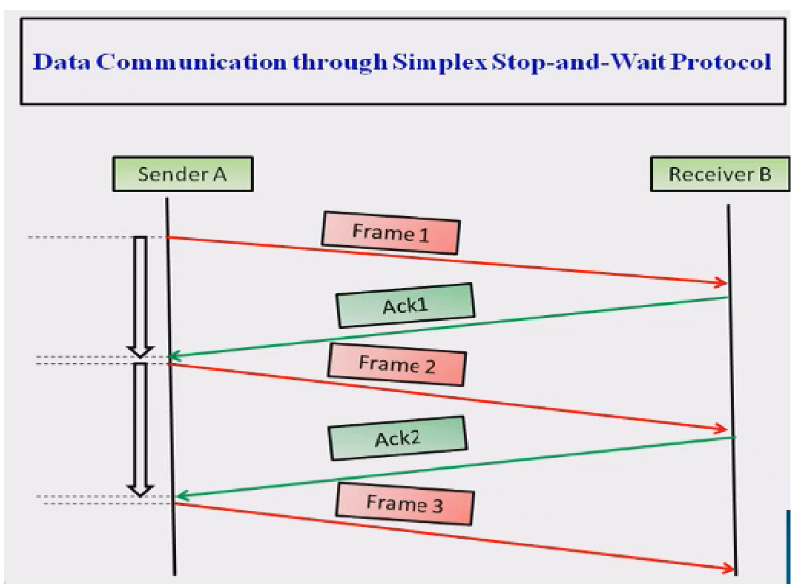
- Such elementary protocols are also called PAR (Positive Acknowledgment with Retransmission) or ARQ (Automatic Repeat reQuest).
- Data frames are transmitted in one direction (simplex protocols) where each frame is individually acknowledge by the receiver by a separate acknowledgment frame.
- The sender transmits one frame, starts a timer and waits for an acknowledgment frame from the receiver before sending further frames.
- A time-out period is used where frames not acknowledged by the receiver are retransmitted automatically by the sender.
- Frames received damaged by the receiver are not acknowledged and are retransmitted by the sender when the expected acknowledgment is not received and timed out.

Advantage :

- *Simple Reliable transmission.(no loss of data).*

Dis Advantage :

- *Sender always wait for Acknowledgement frame to send next frame.(only one to send at a time it takes more time , if acknowledgement receives then only it transmits next frame.*
- *If Acknowledgement frame gets corrupted the sender cannot perform no more trans tractions.*



❊ Simplex protocol for Noisy channel:

Simplex Stop – and – Wait protocol for noisy channel is data link layer protocol for data communications with error control and flow control mechanisms. It is popularly known as Stop – and –Wait Automatic Repeat Request (Stop – and –Wait ARQ) protocol. It adds error control facilities to Stop – and – Wait protocol.

- *Data transfer in only one direction.*
- *Finite processing capacity to the sender and receiver.(same processing speed).*
- *It is a Noisy channel , Errors in the data frame and acknowledgement frame are expects.*
- *Every frame has separate sequence number.*
- *After frame has to be transmitted Timer is started for finite time , Before the timer expires if the Acknowledgement is not received the frame again retransmits.*

* Data frame can be lost
* Ack can be lost

Lost data frames:
This means that sender will never get an ack.
Sender can implement timer for each frame sent. If timer expires retransmit.

But what if the data frame arrived but the ack was lost?

Using above method, sender will retransmit frame that receiver has already received.
Result: *Duplicate frame at sender.*

So receiver needs some way of distinguishing duplicates.
Answer: Use sequence numbers.

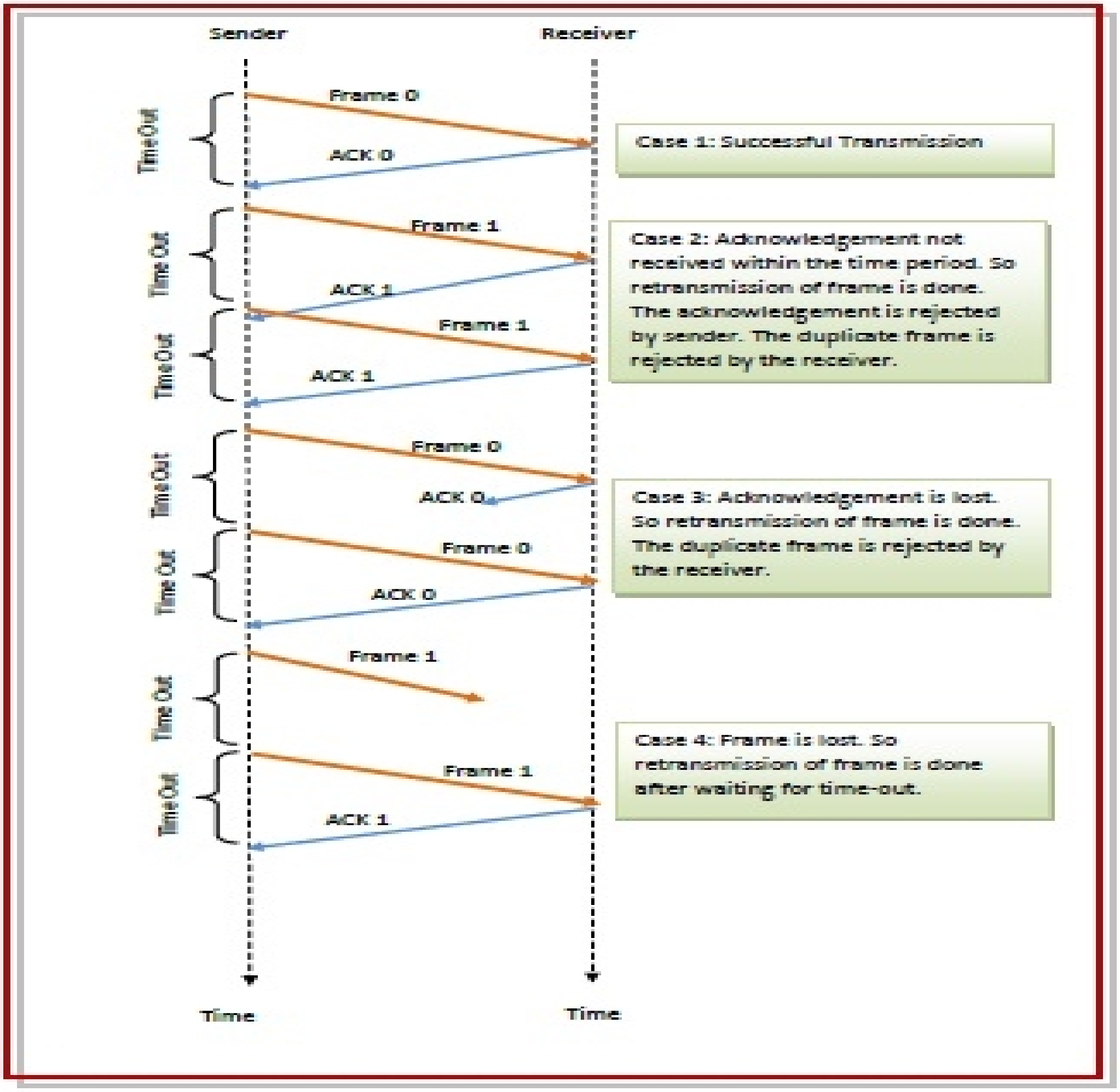
What should be the field size of these sequence numbers ?

- ***Sender Side –***

At the sender site, a field is added to the frame to hold a sequence number. If data is available, the data link layer makes a frame with the certain sequence number and sends it. The sender then waits for arrival of acknowledgment for a certain amount of time. If it receives a positive acknowledgment for the frame with that sequence number within the stipulated time, it sends the frame with next sequence number. Otherwise, it resends the same frame.

- ***Receiver Side –***

The receiver also keeps a sequence number of the frames expected for arrival. When a frame arrives, the receiver processes it and checks whether it is valid or not. If it is valid and its sequence number matches the sequence number of the expected frame, it extracts the data and delivers it to the network layer. It then sends an acknowledgement for that frame back to the sender along with its sequence number.


Sender
Receiver
Time Out
Frame 0
ACK 0
Case 1: Successful Transmission
Frame 1
ACK 1
Case 2: Acknowledgement not received within the time period. So retransmission of frame is done. The acknowledgement is rejected by sender. The duplicate frame is rejected by the receiver.
Frame 0
ACK 0
Case 3: Acknowledgement is lost. So retransmission of frame is done. The duplicate frame is rejected by the receiver.
Frame 1
Case 4: Frame is lost. So retransmission of frame is done after waiting for time-out.
ACK 1
Time
Time

❖ Sliding window protocol:

- Sliding window protocols are data link layer protocols for reliable and sequential delivery of data frames. The sliding window is also used in Transmission Control Protocol.
- In this protocol, **multiple frames can be sent by a sender at a time before receiving an acknowledgment from the receiver.**
- The **term sliding window refers to the imaginary boxes to hold frames.**
- Sliding window method is also known as windowing.

How data is transmitted. **It follows 2 parameters**

1) **sequence number.** (given to each outbound i.e. sending frame) (range 0 to $2^n - 1$).
2) **sliding window.** (imaginary boxes i.e. buffers temporary storage)

Transmitter and receiver side to increase efficiency

- The size of the receiving window is the maximum number of frames that the receiver can accept at a time.
- It determines the maximum number of frames that the sender can send before receiving acknowledgment.

Working Principle:

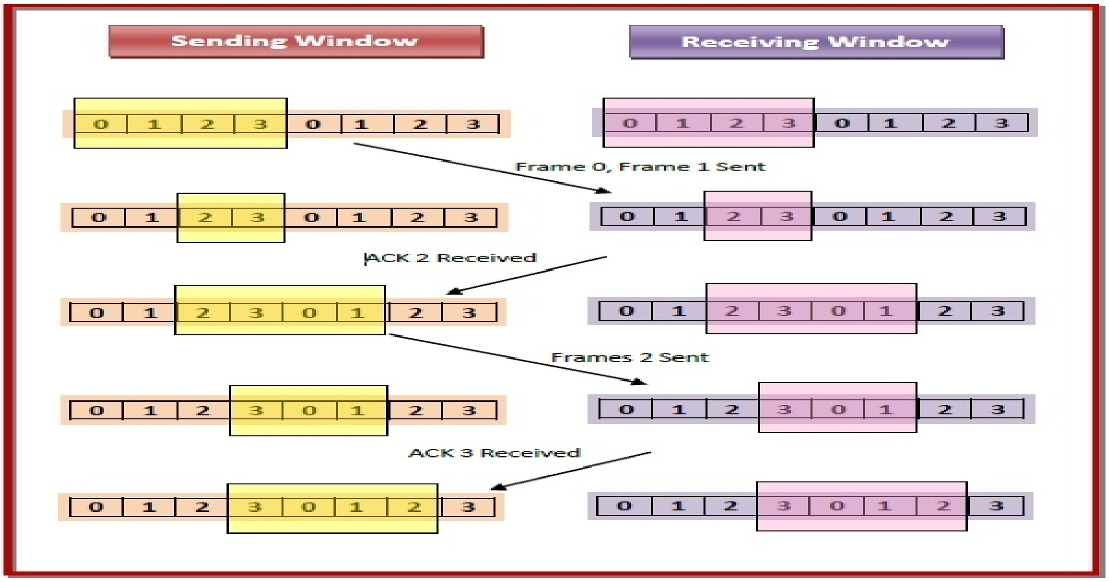
- In these protocols, the sender has a buffer called the sending window and the receiver has buffer called the receiving window.
- The size of the sending window determines the sequence number of the outbound frames.
- If the sequence number of the frames is an **n**-bit field, then the range of sequence numbers that can be assigned is **0 to $2^n - 1$**. Consequently, the size of the sending window is $2^n - 1$.
- **For example:**

if the sending window size is 4, then the sequence numbers will be 0, 1, 2, 3, 0, 1, 2, 3, 0, 1, and so

on. The number of bits in the sequence

Example:

Suppose that we have sender window and receiver window each of size 4. So the sequence numbering of both the windows will be 0,1,2,3,0,1,2 and so on. The following diagram shows the positions of the windows after sending the frames and receiving acknowledgments.

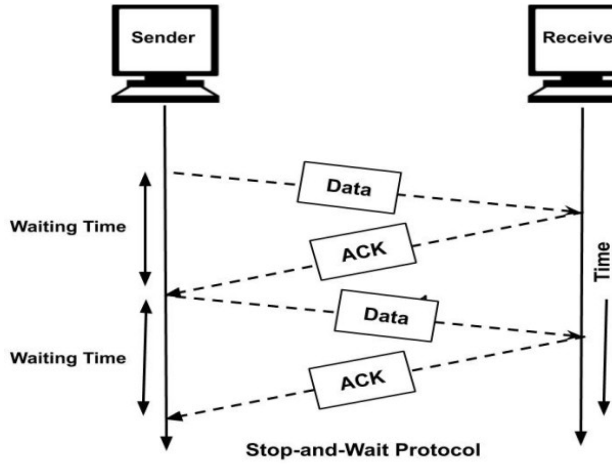


❈ One-Bit Sliding window protocol:

- This is **'Simplex protocol' with sequence number and Acknowledgement frame indicating the sequence number of the next frame expected.**
- **In this sliding window protocol the maximum window size is I. such a protocol uses stop-and-wait since the sender transmits a frame and waits for its Acknowledgement before sending next frame.**
- One bit sliding window protocol is also called as Stop-and-wait ARQ (automatic repeat request).
- It performs Error control and flow control mechanisms.
    a) **Normal Operation.**
    b) **Frame lost operation.**
    c) **Acknowledgment lost operation.**
    d) **Acknowledgement delay operation.**

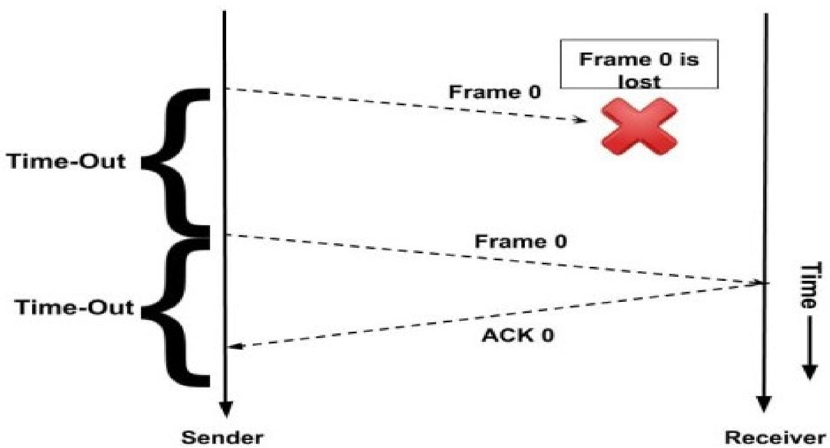
a) Normal Operation:

It is the simplest flow control method. In this, the sender will send one frame at a time to the receiver. The sender will **stop and wait** for the acknowledgment from the receiver. This time(i.e. the time between message sending and acknowledgement receiving) is the waiting time for the sender and the sender is totally idle during this time. When the sender gets the acknowledgment(ACK), then it will send the next data packet to the receiver and wait for the acknowledgment again and this process will continue as long as the sender has the data to send.



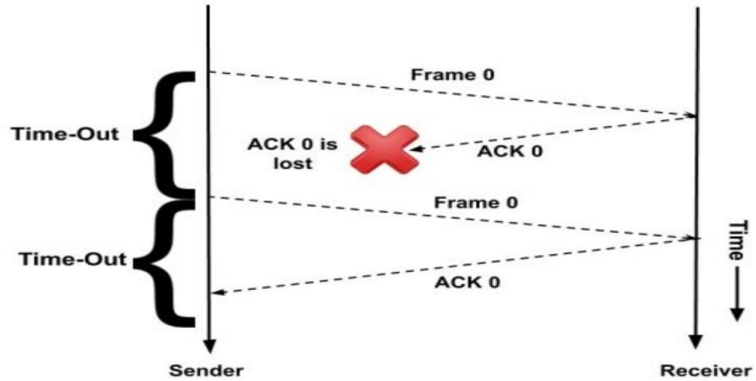
b) Frame loss operation :

Suppose if any frame sent is not received by the receiver and is lost. So the receiver will not send any acknowledgment as it has not received any frame. Also, the sender will not send the next frame as it will wait for the acknowledgment for the previous frame which it had sent. So a deadlock situation arises here. To avoid any such situation there is a **time-out** timer. The sender waits for this fixed amount of time for the acknowledgment and if the acknowledgment is not received then it will send the frame again.

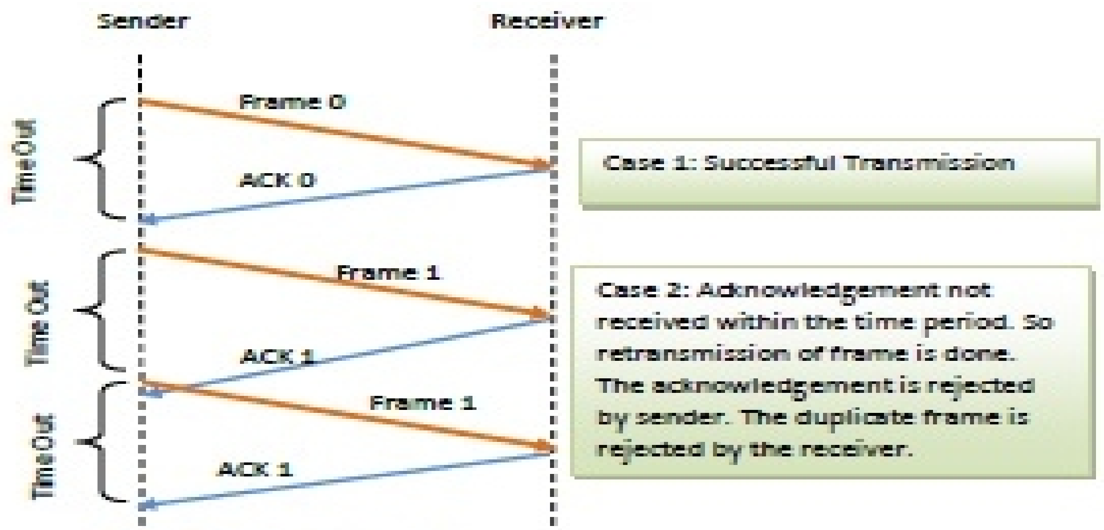


c) Acknowledgement loss :

Consider a situation where the receiver has received the data and sent the acknowledgment but the ACK is lost. So, again the sender might wait till infinite time if there is no system of **time-out** timer. So, in this case also, the time-out timer will be used and the sender will wait for a fixed amount of time for the acknowledgment and then send the frame again if the acknowledgement is not received.



d) Delay Acknowledgement :



Types of Sliding Window Protocols:

The Sliding Window ARQ (Automatic Repeat request) protocols are of two categories −

Go – Back – N ARQ:

- Go – Back – N ARQ provides for sending multiple frames before receiving the acknowledgment for the first frame. It uses the concept of sliding window, and so is also called sliding window protocol. The frames are sequentially numbered and a finite number of frames are sent. If the acknowledgment of a frame is not received within the time period, all frames starting from that frame are retransmitted.

Selective Repeat ARQ:

- This protocol also provides for sending multiple frames before receiving the acknowledgment for the first frame. However, here only the erroneous or lost frames are retransmitted, while the good frames are received and buffered.

❈ Go – Back – N ARQ:

- **N indicates number of frames to be send.**
- **The difference b/w stop & wait protocol and Go-back-N protocol Stop &wait sends only one frame at a time but in this protocol sends several frames Before receiving Acknowledgment.**
- **It is Error control and Detect error is Data link layer.**

Go – Back – N ARQ provides for sending multiple frames before receiving the acknowledgment for the first frame.

It uses the concept of sliding window, and so is also called sliding window protocol.

The frames are sequentially numbered and a finite number of frames are sent.

If the acknowledgment of a frame is not received within the time period, all frames starting from that frame are retransmitted.

The Receiver Receives Damaged frame the receiver sends Negative Acknowledgement with Sequence number of Damaged frame i.e. N values ,Again it sends all frame from damaged frame to until last frame once again.

Damaged Frame:

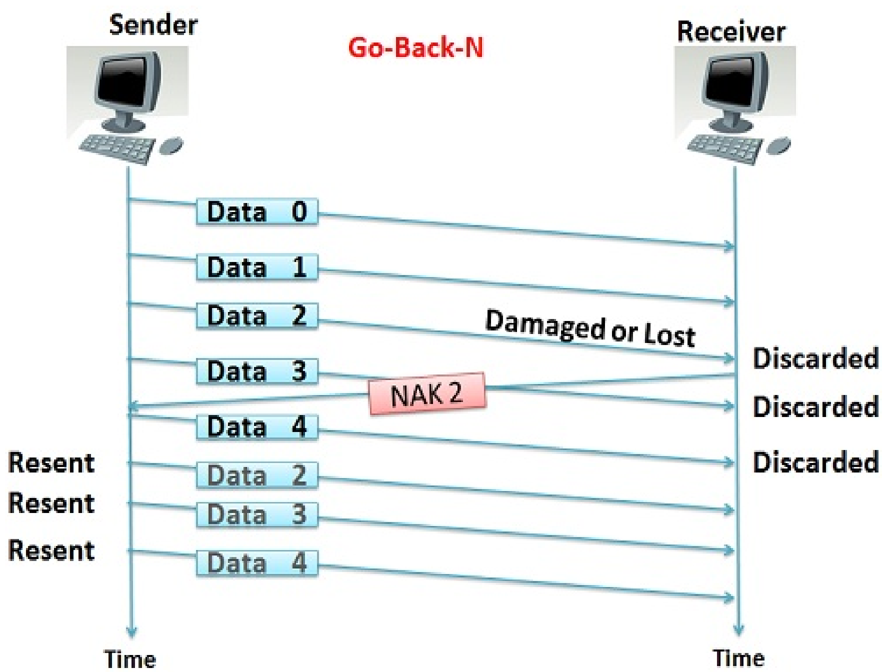
- **Receiver** sends Negative Acknowledgement along with Frame Sequence number.
- After Negative Acknowledgement receives Discard all frames.

Loss Frame:

- Checks the number on each frame.
- Verifies the sequence number i.e. easily identifies which frame is missing.

Loss Acknowledgement:

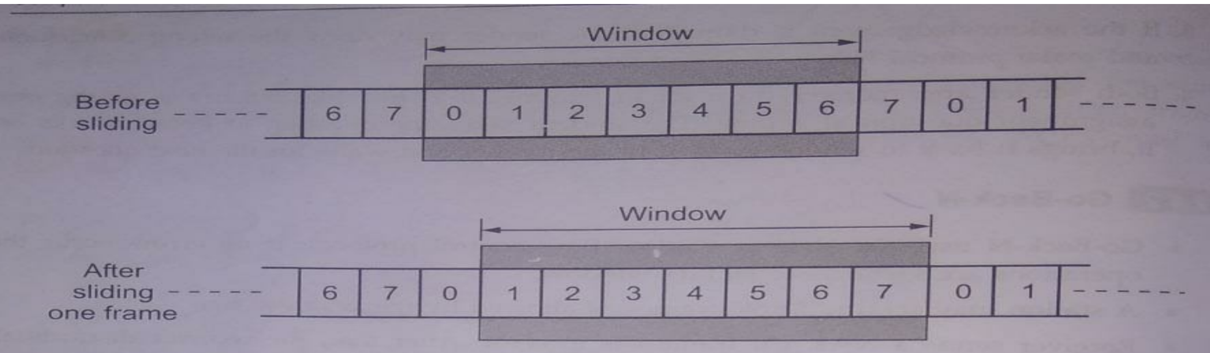
- If the sender does not receives Acknowledgement within a time so agian it transmits .



Additional features of Go-Back-N ARQ

1. Sequence numbers : Sequence numbers of transmitted frame are maintained in the header of each frame. If k is the number of bits for sequence number, then the numbering can range from 0 to $2^k - 1$ e.g. for k = 3. Sequence numbers are 0 to 7 $(2^3 - 1)$.

2. Sender sliding window : Window is a set of frames in buffer waiting for acknowledgement. This window keeps on sliding in forward direction. The window size is fixed. As the ACK is received, the respective frame goes out of window and new frame to sent come into window. Fig. 2.4.7 illustrates sliding of window for window size = 7.



Window size = 7

Fig. 2.4.7 Sender sliding window

3. Receiver sliding window : In the receiver side the size of the window is always one. The receiver is expecting to arrive frames in specific sequence. Any other frame received which is out of order is discarded. The receiver slides over after receiving the expected frame. Fig. 2.4.8 shows receiver sliding window.

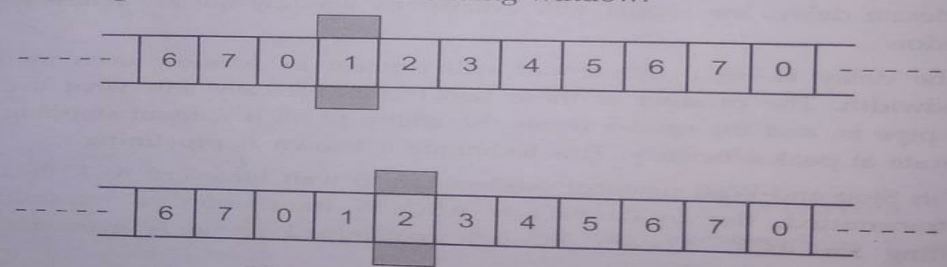


Fig. 2.4.8 Receiver sliding window

4. Control variables :

a) Sender variables

- The sender deals with three different variables.

  S → sequence number of recently sent frame.

  S_F → sequence number of first frame in window.

  S_L → sequence number of last frame in window.

  ∴ Window size $W = S_L - S_F + 1$

  e.g. In previous feature, $W = 7 - 0 + 1 = 8$

b) Receiver variable

- The receiver deals with one variable only.

  R → sequence no. of frame expected.

  If the number matches, then the frame is accepted otherwise not.

5. Timers

- The sender has a timer for each transmitted frame. The receiver don't have any timer.

6. Acknowledgement

- The receiver responds for frames arriving safely by positive acknowledgements. For damaged or lost frames receiver does not reply, the sender has to retransmit it when timer of that frame elapsed.
- The receiver may acknowledge once for several frames.

7. Resending of frames

- If the timer for any frame expires, the sender has to resend that frame and the subsequent frames also, hence the protocol is called Go-Back-N ARQ.

Operation

a) Normal operation

- The sender sends frames and update the control variables i.e. S_F, S, S_L and receiver updates variable R. Fig. 2.4.9 shows normal operation.

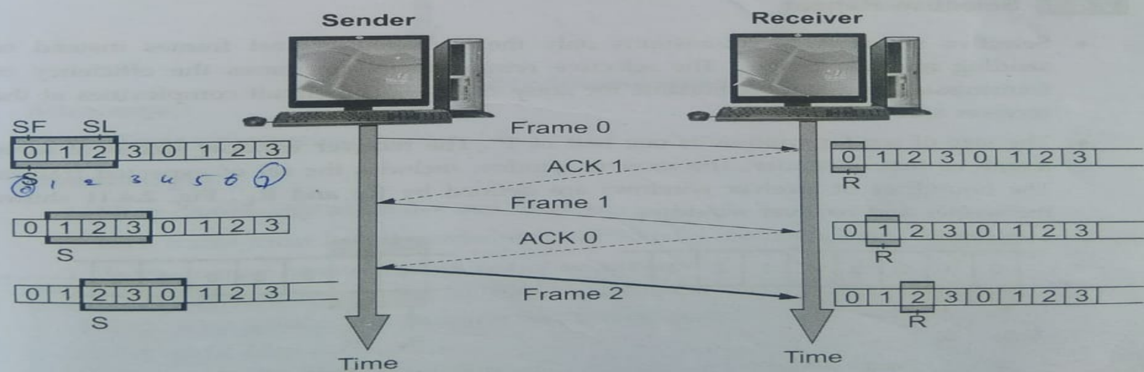


Fig. 2.4.9 Go-Back-N ARQ, Normal operation

b) Damaged or lost frame

- Suppose frame 2 is damaged or lost and if receiver receives frame 3, it will be discarded since it is expecting frame 2. Sender retransmits frame 2 and frame 3. Fig. 2.4.10 shows this process.

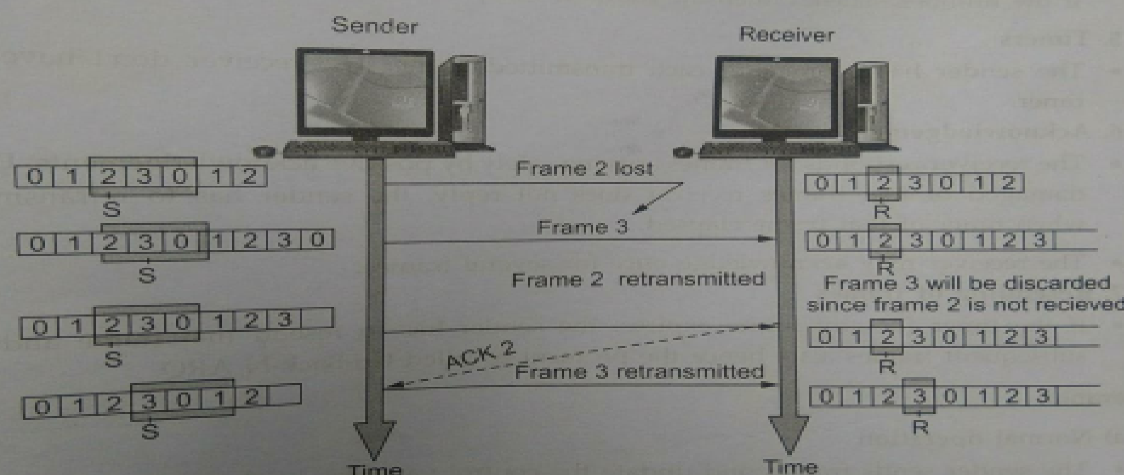
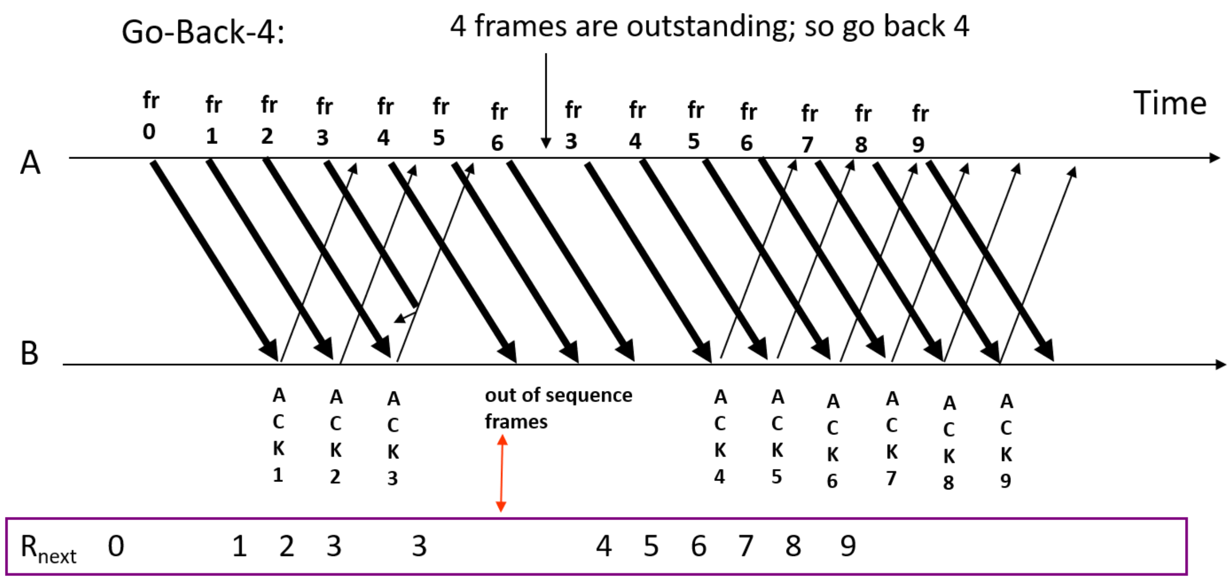
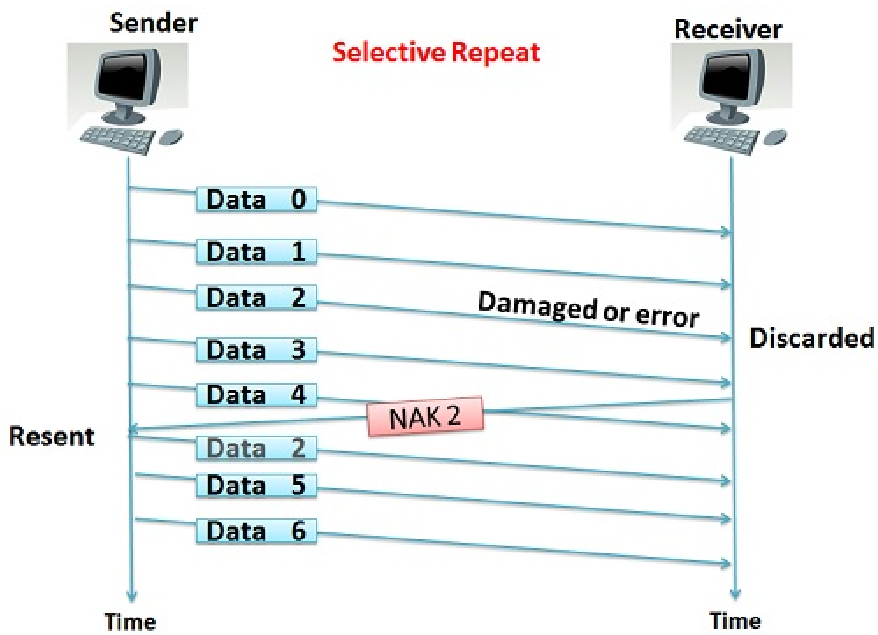


Fig. 2.4.10 Go-Back-N, Damaged frame



❈ Selective Repeat ARQ:

- *This protocol also provides for sending multiple frames before receiving the acknowledgment for the first frame.*
- *However, here only the lost frames are retransmitted, while the good frames are received and buffered.*
- *It Retransmits only Damaged or Loss frame only.*
- *The receiver is capable of Sorting the frame proper sequence manner.*
- *The sender must be capable of searching frame for which NAK has been received.*
- *It requires Less Window size as compare to Go-Back-N Protocol.*



Operation

- In sequential transmission of frame 0, 1, 2, 3, suppose frame 2 is lost and the next frame 3 is already received then receiver sends NAK 2 frame to sender. Then sender retransmits frame 2 only. Fig. 2.4.12 shows operation of selective repeat ARQ.

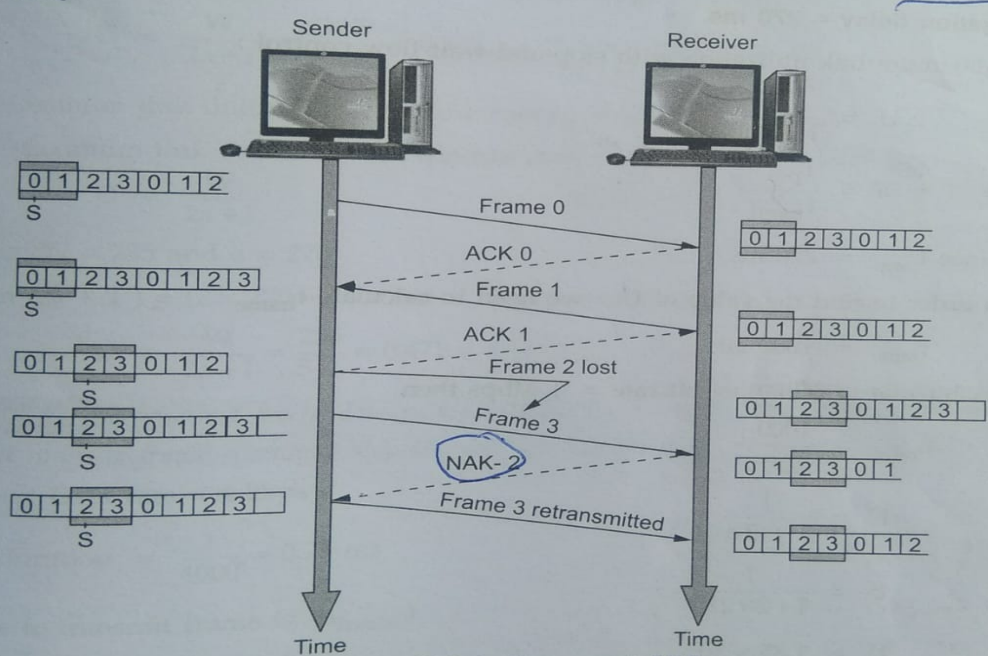
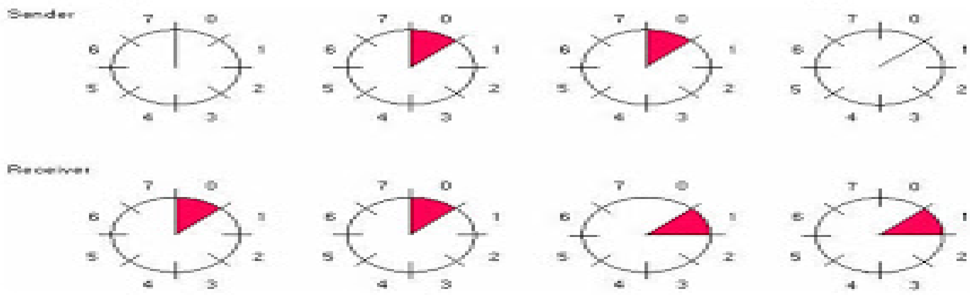


Fig. 2.4.12 Selective repeat ARQ

A Sliding Window Protocol of Size 1 with a 3-bit Sequence Number



(a) Initial state

(b) After the first frame has been sent

(c) After the first frame has been received

(d) After the first acknowledgment frame has been received

S.NO	GO-BACK-N PROTOCOL	SELECTIVE REPEAT PROTOCOL
1.	In Go-Back-N Protocol, if the sent frame are find suspected then all the frames are re-transmitted from the lost packet to the last packet transmitted.	In selective Repeat protocol, only those frames are re-transmitted which are found suspected.
2.	Sender window size of Go-Back-N Protocol is N.	Sender window size of selective Repeat protocol is also N.
3.	Receiver window size of Go-Back-N Protocol is 1.	Receiver window size of selective Repeat protocol is N.
4.	Go-Back-N Protocol is less complex.	Selective Repeat protocol is more complex.
5.	In Go-Back-N Protocol, neither sender nor at receiver need sorting.	In selective Repeat protocol, receiver side needs sorting to sort the frames.
6.	In Go-Back-N Protocol, Out-of-Order packets are NOT Accepted (discarded) and the entire window is re-transmitted.	In selective Repeat protocol, Out-of-Order packets are Accepted.
7.	In Go-Back-N Protocol, if Receives receives a corrupt packet, then also, the entire window is re-transmitted.	In selective Repeat protocol, if Receives receives a corrupt packet, it immediately sends a negative acknowledgement and hence only the selective packet is retransmitted.

❈Error Detection and Error correction:

Hamming Code:

- Hamming code is a set of error-correction codes that can be used to **detect and correct the errors** that can occur when the data is moved or stored from the sender to the receiver.
- In this method System to identify where Error is occur and Rectify that Error.
- It is **technique developed by R.W. Hamming for error correction.**

Redundant bits –

- Redundant bits are extra binary bits that are generated and added to the information-carrying bits of data transfer to ensure that no bits were lost during the data transfer.

The number of redundant bits can be calculated using the following formula:

$$2^r \geq m + r + 1$$

where, r = redundant bit, m = data bit

Suppose the number of data bits is 7,

then the number of redundant bits can be calculated using: = $2^4 \geq 7 + 4 + 1$
Thus, the number of redundant bits= **4**

Data (M) -----→ Any Length

(parity bits) Redundant Bits (r) ----------→ Find Length of r by using below equation.

$$2^r \geq m + r + 1$$

where, r = redundant bit, m = data bit

Sender ----→ send the message **M + r**

Parity bits –

A parity bit is a bit appended to a data of binary bits to ensure that the total number of 1's in the data is even or odd. Parity bits are used for error detection.

There are two types of parity bits:

Even parity bit:

In the case of even parity, for a given set of bits, the number of 1's are counted. If that count is odd, the parity bit value is set to 1, making the total count of occurrences of 1's an even number. If the total number of 1's in a given set of bits is already even, the parity bit's value is 0.

Odd Parity bit :

In the case of odd parity, for a given set of bits, the number of 1's are counted. If that count is even, the parity bit value is set to 1, making the total count of occurrences of 1's an odd number. If the total number of 1's in a given set of bits is already odd, the parity bit's value is 0.

General Algorithm of Hamming code –

The Hamming Code is simply the use of extra parity bits to allow the identification of an error.

1. Write the bit positions starting from 1 in binary form (1, 10, 11, 100, etc i.e 1,2,3,4......).
2. All the bit positions that are a power of 2 are marked as parity bits (1, 2, 4, 8, etc).
3. All the other bit positions are marked as data bits.
4. Each data bit is included in a unique set of parity bits, as determined its bit position in binary form.

   a. Parity bit 1 covers all the bits positions whose binary representation includes a 1 in the least significant position (1, 3, 5, 7, 9, 11, etc).

   b. Parity bit 2 covers all the bits positions whose binary representation includes a 1 in the second position from
   the least significant bit (2, 3, 6, 7, 10, 11, etc).

   c. Parity bit 4 covers all the bits positions whose binary representation includes a 1 in the third position from the least significant bit (4–7, 12–15, 20–23, etc).

   d. Parity bit 8 covers all the bits positions whose binary representation includes a 1 in the fourth position from
   the least significant bit bits (8–15, 24–31, 40–47, etc).

   e. In general, each parity bit covers all bits where the bitwise AND of the parity position and the bit position is
   non-zero.

5.Since we check for even parity set a parity bit to 1 if the total number of ones in the positions it checks is odd.

6. Set a parity bit to 0 if the total number of ones in the positions it checks is even.

Position	R8	R4	R2	R1
0	0	0	0	0
1	0	0	0	1
2	0	0	1	0
3	0	0	1	1
4	0	1	0	0
5	0	1	0	1
6	0	1	1	0
7	0	1	1	1
8	1	0	0	0
9	1	0	0	1
10	1	0	1	0
11	1	0	1	1

R1 -> 1,3,5,7,9,11
R2 -> 2,3,6,7,10,11
R3 -> 4,5,6,7
R4 -> 8,9,10,11

Example :

Haming Code

Data (M) $\rightarrow$ Any Length

(parity bits) Redundant bits (r) $\rightarrow$ length of 'r' depends on

$$\boxed{2^r \geq m+r+1}$$

Sender Sends $\rightarrow$ $m + r$ Information.

$\downarrow$ Placed at powers of '2'.

$r_1 \rightarrow 2^0$

$r_2 \rightarrow 2^1$

$r_3 \rightarrow 2^2$

$r_4 \rightarrow 2^3$

(r_1 placed at 2^0, $r_1 \rightarrow 2^1$ - - - -)

$\boxed{r = 4}$ (parity bits)

P_8	P_4	P_2	P_1
$\downarrow$	$\downarrow$	$\downarrow$	$\downarrow$
2^3	2^2	2^1	2^0

$\rightarrow$ positions of parity bits.

Either we can follow "Even parity" (or) "ODD parity".

Even Parity : Ex :

Data (M) = 1001 $\leftarrow$ 4 bits.

(Redundant) parity bits (r) = $\boxed{2^r \geq m+r+1}$ satisfies the Condition.

$2^1 \geq 4 + 1 + 1$	$2^2 \geq 4 + 2 + 1$	$2^3 \geq 4 + 3 + 1$
$2 \geq 6$ ×	$\cancel{4} \geq 7$ ×	$8 \geq 8$ ✓
		satisfies $r = 3$ ✓

$\therefore$ (r) = 3 bits

P_4	P_2	P_1

2020-9-20 12:08

$(r) = 3$ bits

P_4	P_2	P_1

→ powers of '2' $(2^0, 2^1, 2^2)$
1, 2, 4 position

Length of Data $= M + r$
$= 4 + 3$
$= 7$ bits total.

Now calculate powers of '2'

7	6	5	4	3	2	1
1	0	0	$\boxed{P_4}$	1	$\boxed{P_2}$	$\boxed{P_1}$

(i.e) 1 0 0 P_4 1 P_2 P_1

then sender Identify the P_1, P_2, P_4 bits.

then sender Identify the P_1, P_2, P_4 bits.

Now Calculate All possible Binary Representation of '3' bits

(P_4) (P_2) (P_1)

1 → 0 0 1
2 → 0 1 0
3 → 0 1 1
4 → 1 0 0
5 → 1 0 1
6 → 1 1 0
7 → 1 1 1
(Binary Representation).

→ Now find position of 1's in Binary Representation (Even parity)

P_1 → 1, 3, 5, 7 → P_1 1 0 1 → $P_1 = 0$
P_2 → 2, 3, 6, 7 → P_2 1 0 1 → $P_2 = 0$
P_4 → 4, 5, 6, 7 → P_4 0 0 1 → $P_4 = 1$

values of particular position in Dataframe

P_1 101 → 0 ...

P_1 101 → 0101 → Even parity
P_2 101 → 0101 → Even parity
P_4 001 → 1001 → Even parity.

Now Here $\boxed{P_1 = 0,\ P_2 = 0,\ P_4 = 1}$ ⟹ Replace these values in original Data.

7	6	5	4	3	2	1
1	0	0	1	1	0	0
			P_4		P_2	P_1

∴ there All Calculations done at "Sender" Level.

→ Now the frame $\boxed{1001100}$ is send to the Receiver.

$\boxed{\text{Receiver}}$ Receives Corrupted data: ←

7	6	5	4	3	2	1
1	1	0	1	1	0	0

(With Error)

2020-9-20 12:09

Receiver Receives Corrupted data: ← 1 1 0 1 1 0 0
(With Error)

(find Positions)
$P_1 \rightarrow 1, 3, 5, 7 \rightarrow 0101 \rightarrow P_1 = 0$
$P_2 \rightarrow 2, 3, 6, 7 \rightarrow 0111 \rightarrow P_2 = 1$
$P_4 \rightarrow 4, 5, 6, 7 \rightarrow 1011 \rightarrow P_4 = 1$

Consider Even parity

if we get $\boxed{P_1, P_2, P_4 \text{ all 0's}}$ then it is Original Data

Now find Error

P_4	P_2	P_1
1	1	0

→ Decimal 6 So 6th position Error.

7	6	5	4	3	2	1
1	~~1~~ → 0	0	1	1	0	0

↑ 6th position Now change '1' to '0'