User Datagram Protocol (UDP)

Dr Neeraj Mathur
FCS, LMCST

User Datagram Protocol (UDP) is a connectionless Transport layer protocol. For some applications, speed and efficiency are more important than reliability. In such cases, a connectionless protocol can be used. A connectionless protocol doesn't go to the trouble of establishing a connection before sending a packet. Instead, it simply sends the packet.

After UDP has placed a packet on the network (via the IP protocol), it forgets about it. UDP doesn't guarantee that the packet actually arrives at its destination. Most applications that use UDP simply wait for any replies expected as a result of packets sent via UDP. If a reply doesn't arrive within a certain period of time, the application either sends the packet again or gives up.

UDP uses a simple transmission model without implicit handshaking dialogues for providing reliability, ordering, or data integrity. Thus, UDP provides an unreliable service and datagrams (packets) may arrive out of order, appear duplicated, or go missing without notice.

The best-known Application layer protocol that uses UDP is DNS, the Domain Name System. When an application needs to access a domain name such as www.wiley.com, DNS sends a UDP packet to a DNS server to look up the domain. When the server finds the domain, it returns the domain's IP address in another UDP packet.

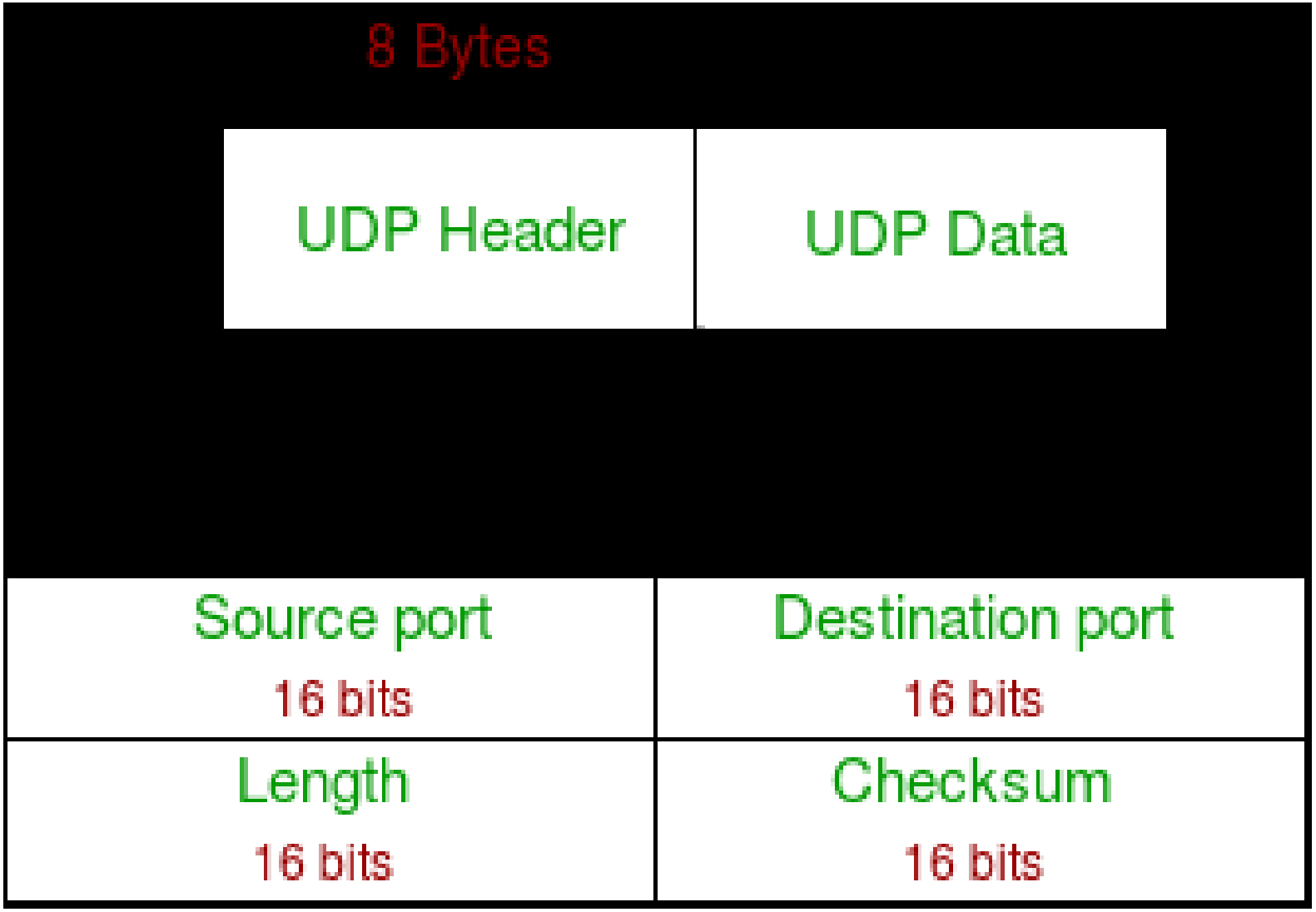
User Datagram Protocol (UDP) is a Transport Layer protocol. UDP is a part of Internet Protocol suite, referred as UDP/IP suite. Unlike TCP, it is unreliable and connectionless protocol. So, there is no need to establish connection prior to data transfer.

Though Transmission Control Protocol (TCP) is the dominant transport layer protocol used with most of Internet services; provides assured delivery, reliability and much more but all these services cost us with additional overhead and latency. Here, UDP comes into picture. For the real-time services like computer gaming, voice or video communication, live conferences; we need UDP. Since high performance is needed, UDP permits packets to be dropped instead of processing delayed packets. There is no error checking in UDP, so it also save bandwidth.

User Datagram Protocol (UDP) is more efficient in terms of both latency and bandwidth.

UDP Header

UDP header is 8-bytes fixed and simple header, while for TCP it may vary from 20 bytes to 60 bytes. First 8 Bytes contains all necessary header information and remaining part consist of data. UDP port number fields are each 16 bits long, therefore range for port numbers defined from 0 to 65535; port number 0 is reserved. Port numbers help to distinguish different user requests or process.


8 Bytes
UDP Header
UDP Data
Source port
16 bits
Destination port
16 bits
Length
16 bits
Checksum
16 bits

- **Source Port :** Source Port is 2 Byte long field used to identify port number of source.
- **Destination Port :** It is 2 Byte long field, used to identify the port of destined packet.
- **Length :** Length is the length of UDP including header and the data. It is 16-bits field.
- **Checksum :** Checksum is 2 Bytes long field. It is the 16-bit one's complement of the one's complement sum of the UDP header, pseudo header of information from the IP header and the data, padded with zero octets at the end (if necessary) to make a multiple of two octets.

Applications of UDP

- Used for simple request response communication when size of data is less and hence there is lesser concern about flow and error control.
- It is suitable protocol for multicasting as UDP supports packet switching.
- Normally used for real time applications which can not tolerate uneven delays between sections of a received message.
- DNS (Domain Name Service)

Application layer can do some of the tasks through UDP-

- Trace Route
- Record Route
- UDP takes datagram from Network Layer, attach its header and send it to the user. So, it works fast.

DCN - Data-link Control & Protocols

Data-link layer is responsible for implementation of point-to-point flow and error control mechanism.

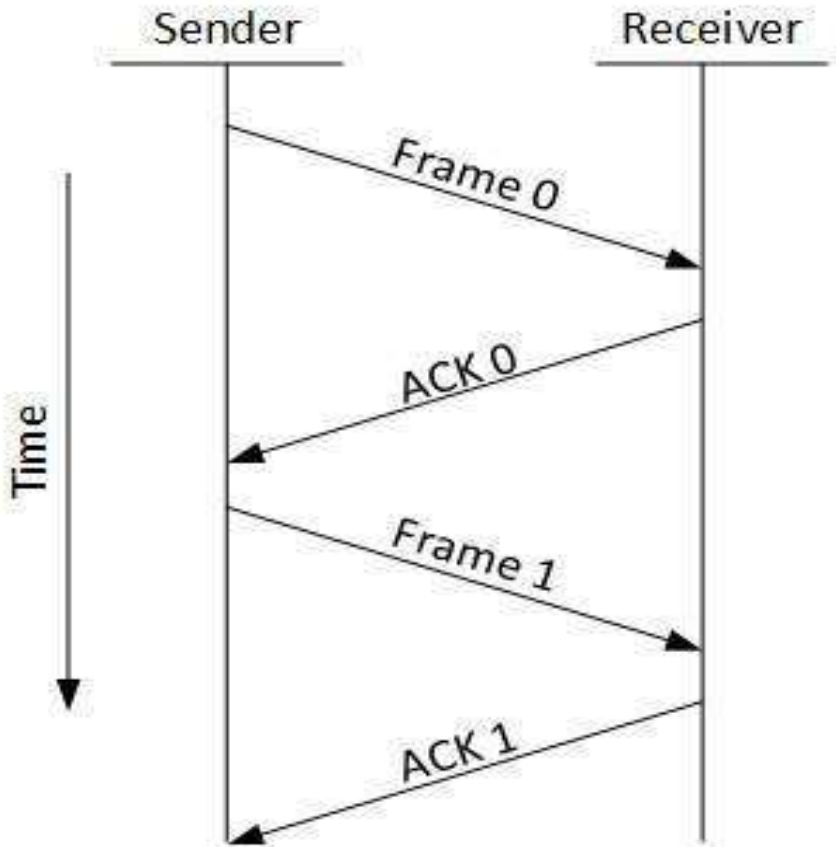
Flow Control

When a data frame (Layer-2 data) is sent from one host to another over a single medium, it is required that the sender and receiver should work at the same speed. That is, sender sends at a speed on which the receiver can process and accept the data. What if the speed (hardware/software) of the sender or receiver differs? If sender is sending too fast the receiver may be overloaded, and data may be lost.

Two types of mechanisms can be deployed to control the flow:

- Stop and Wait

This flow control mechanism forces the sender after transmitting a data frame to stop and wait until the acknowledgement of the data-frame sent is received.

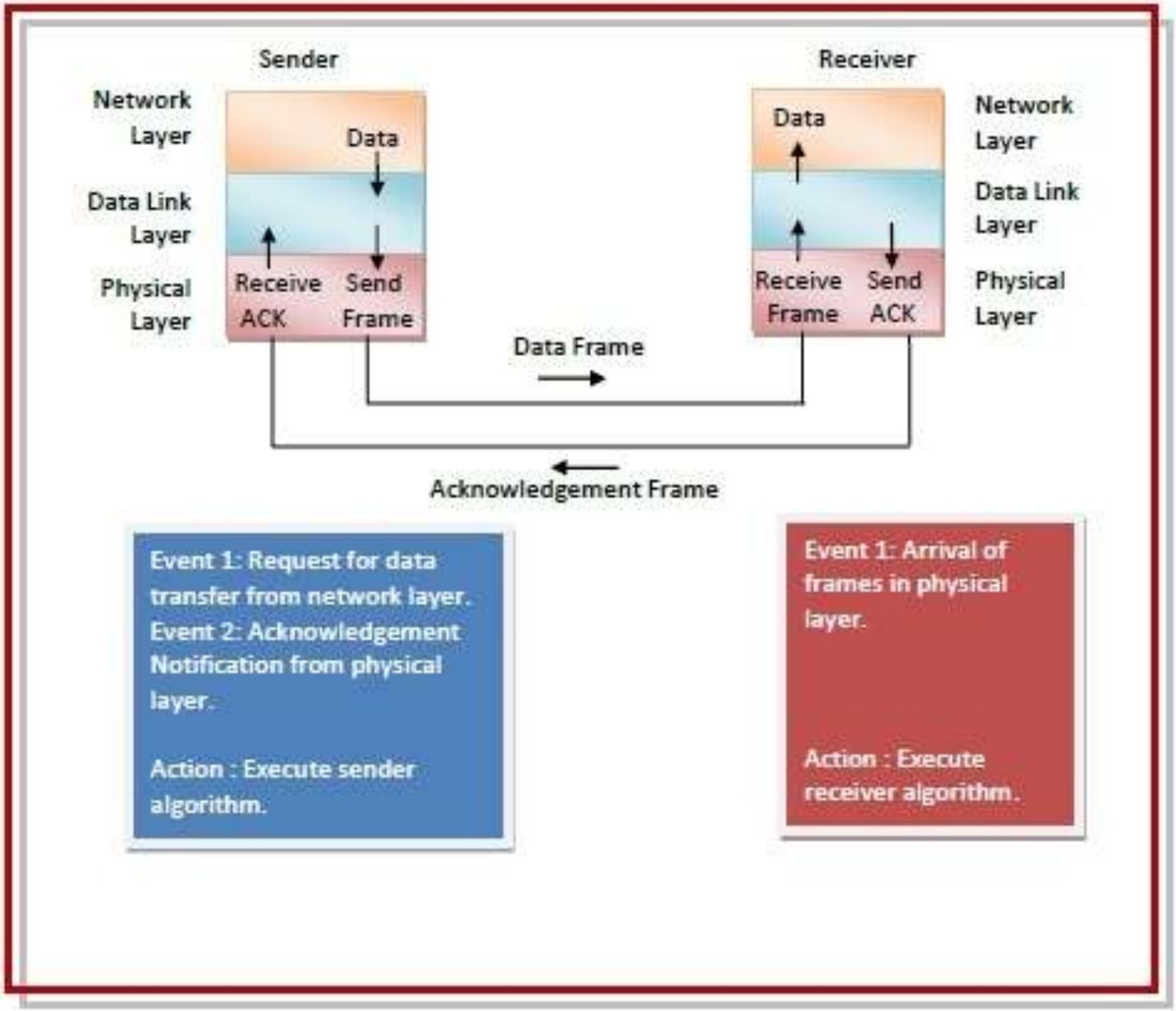


Stop – and – Wait protocol is data link layer protocol for transmission of frames over noiseless channels. It provides unidirectional data transmission with flow control facilities but without error control facilities.

This protocol takes into account the fact that the receiver has a finite processing speed. If data frames arrive at the receiver's end at a rate which is greater than its rate of processing, frames be dropped out. In order to avoid this, the receiver sends an acknowledgement for each frame upon its arrival. The sender sends the next frame only when it has received a positive acknowledgement from the receiver that it is available for further data processing.

Sender Site: The data link layer in the sender site waits for the network layer for a data packet. It then checks whether it can send the frame. If it receives a positive notification from the physical layer, it makes frames out of the data and sends it. It then waits for an acknowledgement before sending the next frame.

Receiver Site: The data link layer in the receiver site waits for a frame to arrive. When it arrives, the receiver processes it and delivers it to the network layer. It then sends an acknowledgement back to the sender.


Sender
Receiver
Network Layer
Data Link Layer
Physical Layer
Data
Receive ACK
Send Frame
Data
Receive Frame
Send ACK
Network Layer
Data Link Layer
Physical Layer
Data Frame
Acknowledgement Frame
Event 1: Request for data transfer from network layer.
Event 2: Acknowledgement Notification from physical layer.
Action : Execute sender algorithm.
Event 1: Arrival of frames in physical layer.
Action : Execute receiver algorithm.

- Sliding Window

In this flow control mechanism, both sender and receiver agree on the number of data-frames after which the acknowledgement should be sent. As we learnt, stop and wait flow control mechanism wastes resources, this protocol tries to make use of underlying resources as much as possible.

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.

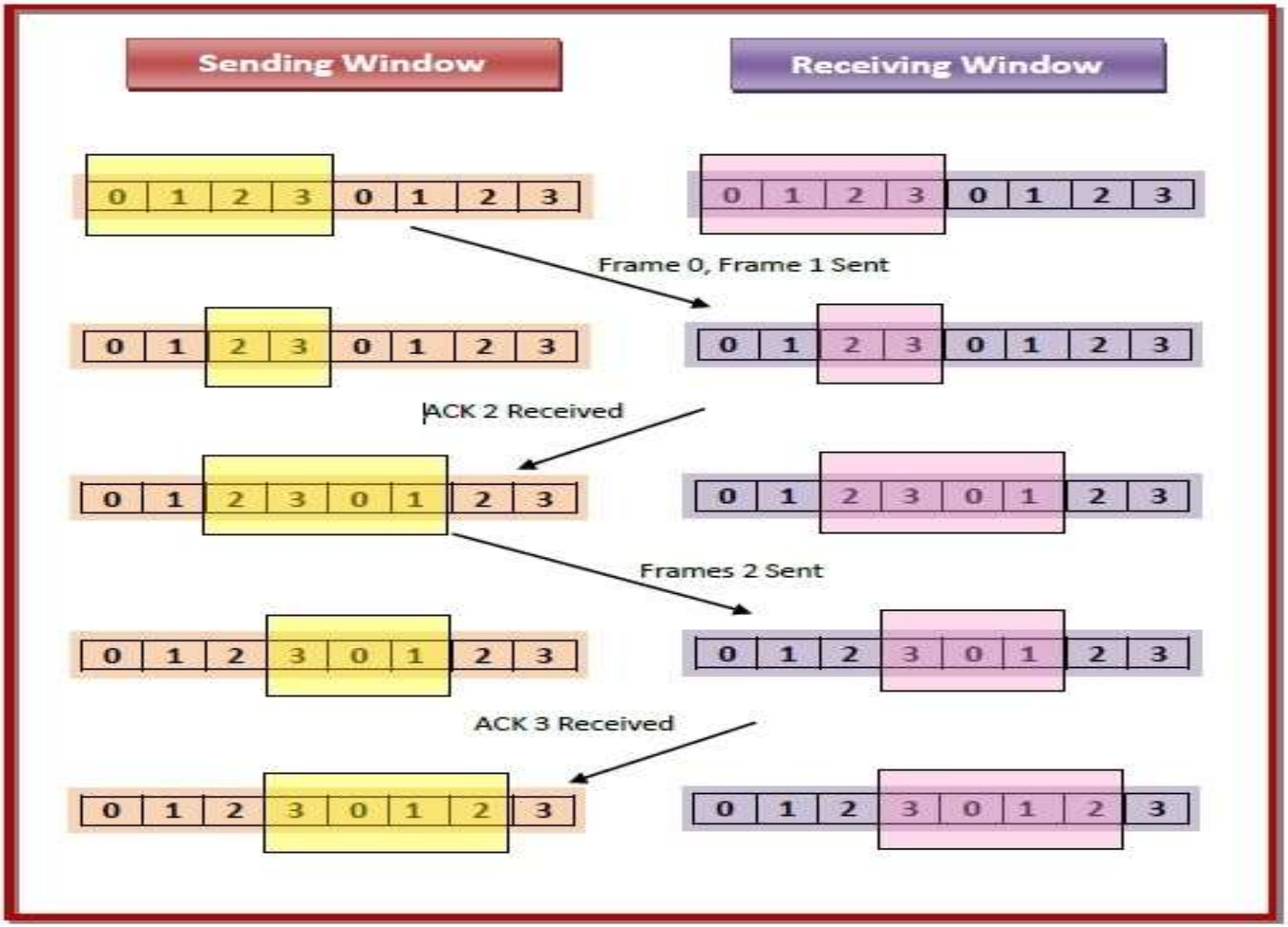
In these protocols, the sender has a buffer called the sending window and the receiver has buffer called the receiving window.

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 $2n-1$. Consequently, the size of the sending window is $2n-1$. Thus in order to accommodate a sending window size of $2n-1$, a n-bit sequence number is chosen.

The sequence numbers are numbered as modulo-n. 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 number is 2 to generate the binary sequence 00, 01, 10, 11.

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.

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



Error Control

When data-frame is transmitted, there is a probability that data-frame may be lost in the transit or it is received corrupted. In both cases, the receiver does not receive the correct data-frame and sender does not know anything about any loss. In such case, both sender and receiver are equipped with some protocols which helps them to detect transit errors such as loss of data-frame. Hence, either the sender retransmits the data-frame or the receiver may request to resend the previous data-frame.

Requirements for error control mechanism:

- Error detection - The sender and receiver, either both or any, must ascertain that there is some error in the transit.

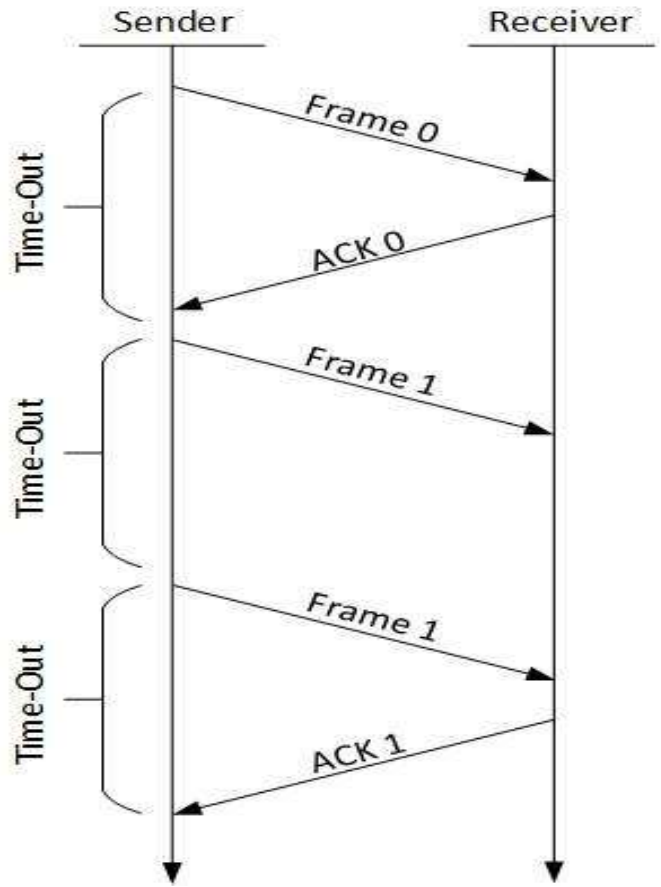
- Positive ACK - When the receiver receives a correct frame, it should acknowledge it.

- Negative ACK - When the receiver receives a damaged frame or a duplicate frame, it sends a NACK back to the sender and the sender must retransmit the correct frame.

- Retransmission: The sender maintains a clock and sets a timeout period. If an acknowledgement of a data-frame previously transmitted does not arrive before the timeout the sender retransmits the frame, thinking that the frame or it's acknowledgement is lost in transit.

There are three types of techniques available which Data-link layer may deploy to control the errors by Automatic Repeat Requests (ARQ):

- Stop-and-wait ARQ



The following transition may occur in Stop-and-Wait ARQ:

- The sender maintains a timeout counter.
- When a frame is sent, the sender starts the timeout counter.
- If acknowledgement of frame comes in time, the sender transmits the next frame in queue.
- If acknowledgement does not come in time, the sender assumes that either the frame or its acknowledgement is lost in transit. Sender retransmits the frame and starts the timeout counter.
- If a negative acknowledgement is received, the sender retransmits the frame.