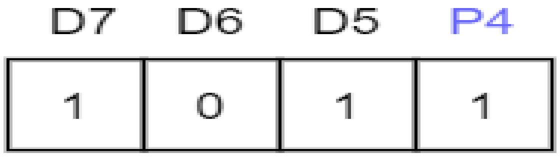
**Step 3:** Check for P4 but while checking for P4, we will use **check four and skip four** method, which will give us the following data bits. But remember since we are checking for P4, so we have started our count from P4(P1 & P2 should not be considered).



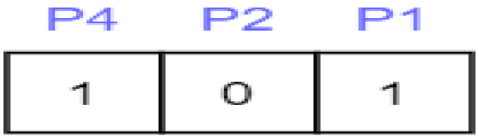
| D7 | D6 | D5 | P4 |
|---|---|---|---|
| 1 | 0 | 1 | 1 |

**Hamming Codes**

- As we can observe that the number of 1's are odd, then we will write the value of **P4 = 1**. This means the error is there.
- So, from the above parity analysis, P1 & P4 are not equal to 0, so we can clearly say that the received hamming code has errors.

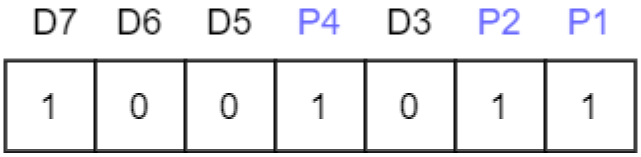
**Hamming Code: Error Correction**

- Since we found that received code has an error, so now we must correct them. To correct the errors, use the following steps:



| P4 | P2 | P1 |
|---|---|---|
| 1 | 0 | 1 |

- Now the error word E will be:
- Now we have to determine the decimal value of this error word **101** which is **5.**
- We get **E = 5**, which states that the error is in the fifth data bit. To correct it, just invert the fifth data bit. So the correct data will be:



| D7 | D6 | D5 | P4 | D3 | P2 | P1 |
|---|---|---|---|---|---|---|
| 1 | 0 | 0 | 1 | 0 | 1 | 1 |

**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** whose size **ranges f**rom a **few hundred** to a **few thousand** bytes.
- **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.

**Types of Data Link Protocols**

- **Data link protocols** can be broadly divided into **two categories**, depending on **whether the transmission channel** is **noiseless** or **noisy.**

**Elementary Data Link Protocols Are:**

1. Simplex Protocol
2. A Simplex Stop-and-Wait Protocol for an Error-Free Channel
3. A Simplex Stop-and-Wait Protocol for a Noisy Channel

**Simplex Protocol**

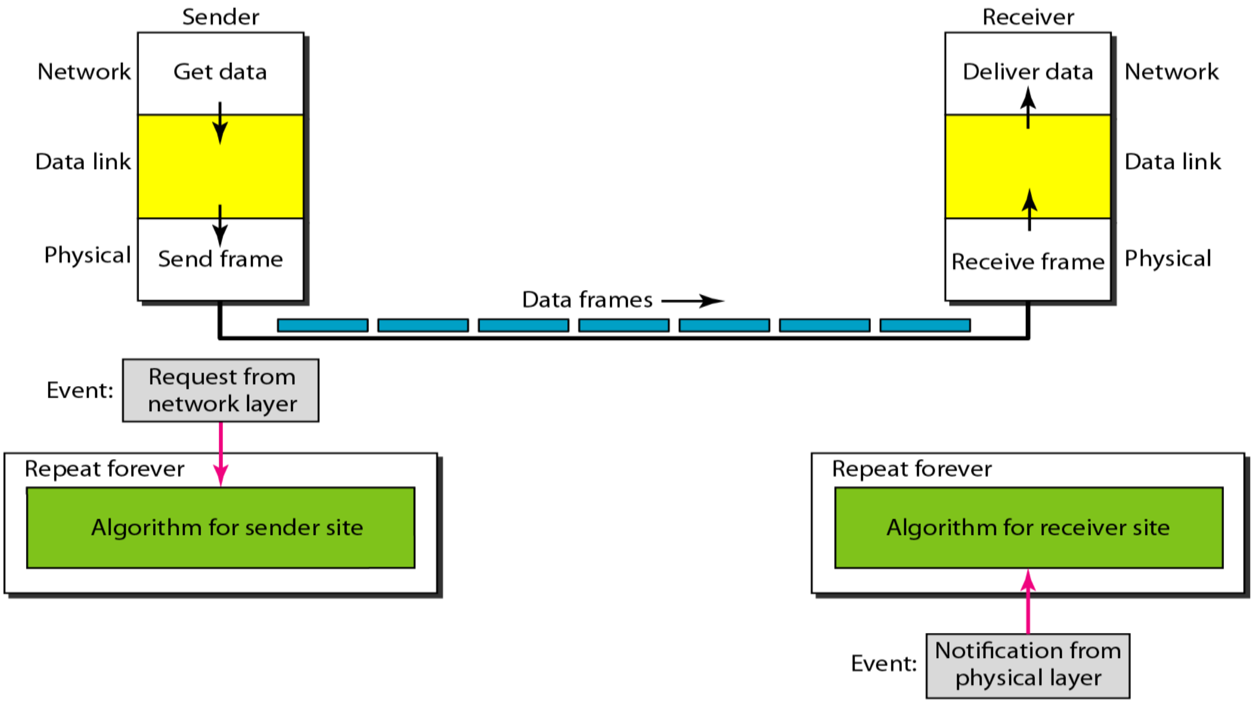
- It is a **unidirectional protocol** in which data frames are traveling in only one direction- from the sender to receiver.
- We assume that the receiver can immediately handle any frame it receives with a **processing time**.
- The data link layer of the receiver immediately removes the header from the frame and hands the data packet to its network layer, which can also accept the packet immediately.

**Simplex Protocol : Design**

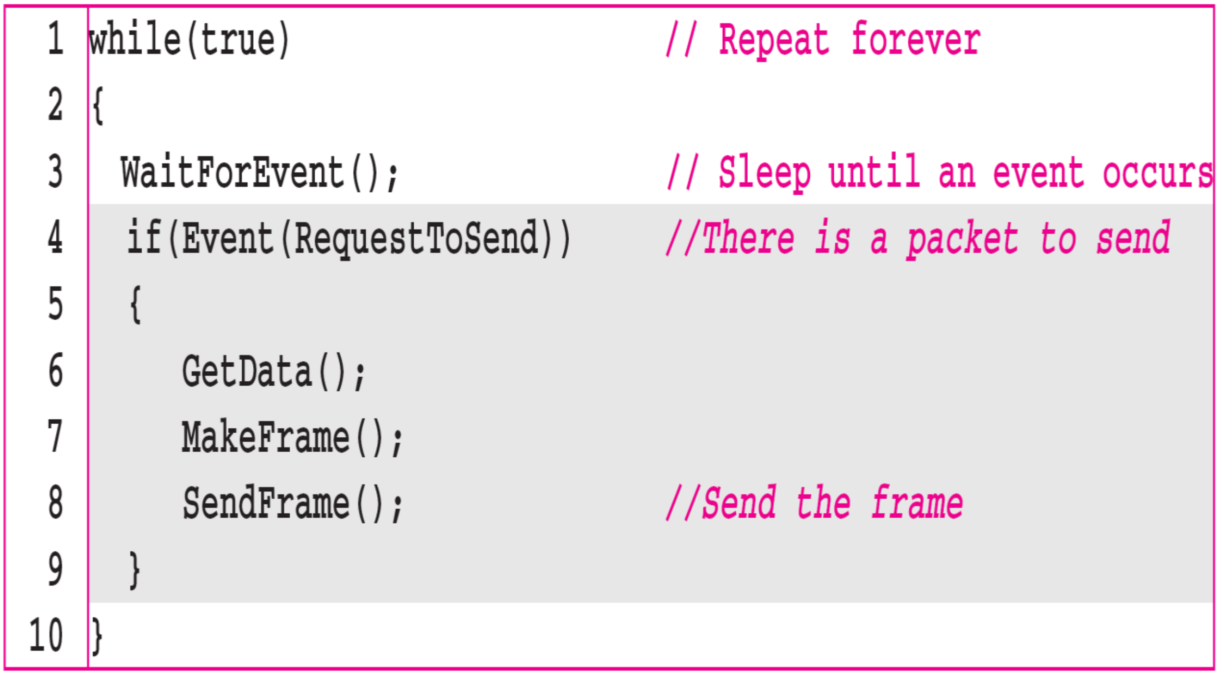
- **Sender Site**: The data link layer at the sender site **gets data from its network layer**, makes a frame out of the data, and sends it.
- **Receiver Site**: The data link layer at the receiver site receives a frame from its physical layer, extracts data from the frame, and delivers the data to its network layer.

**Simplex Protocol : Design**

- The data link layers of the sender and receiver provide transmission services for their network layers. The data link layers use the services provided by their physical layers (such as signaling, multiplexing, and so on) **for the physical transmission of bits**

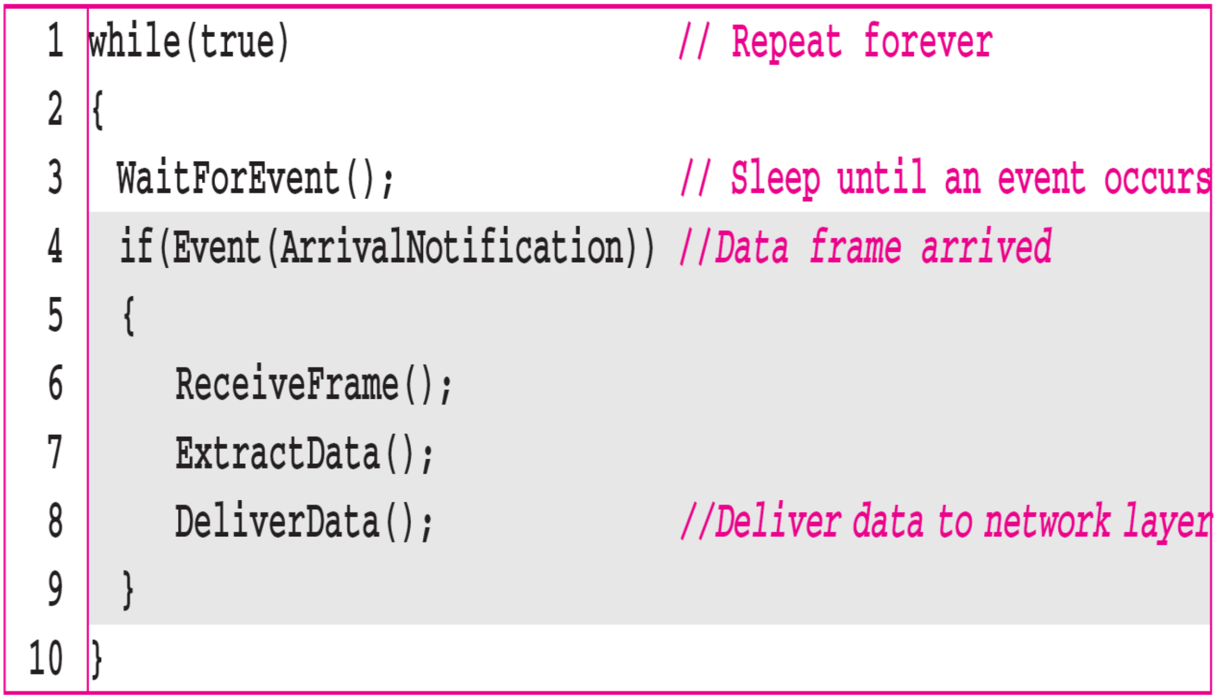


### Algorithm of Simplex Protocol for Sender Site



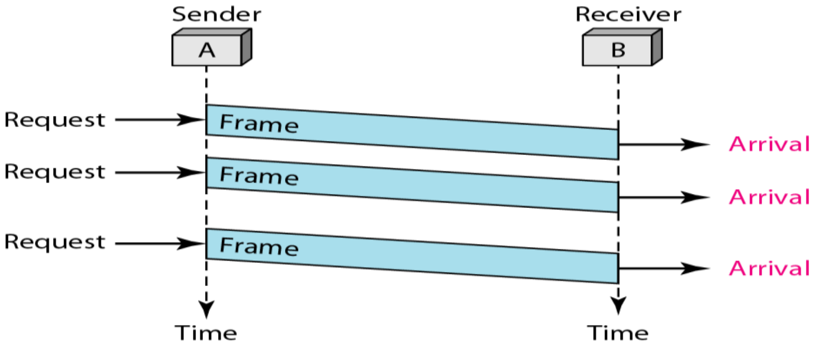
```
while(true)                        // Repeat forever
{
  WaitForEvent();                  // Sleep until an event occurs
   if(Event(RequestToSend))        //There is a packet to send
   {
      GetData();
      MakeFrame();
      SendFrame();                 //Send the frame
   }
}
```

### Algorithm of Simplex Protocol for Receiver Site



```
while(true)                          // Repeat forever
{
  WaitForEvent();                    // Sleep until an event occurs
  if(Event(ArrivalNotification))     //Data frame arrived
  {
     ReceiveFrame();
     ExtractData();
     DeliverData();                  //Deliver data to network layer
  }
}
```

## Simplex Protocol- Flow Diagram



## Simplex Stop – and – Wait Protocol

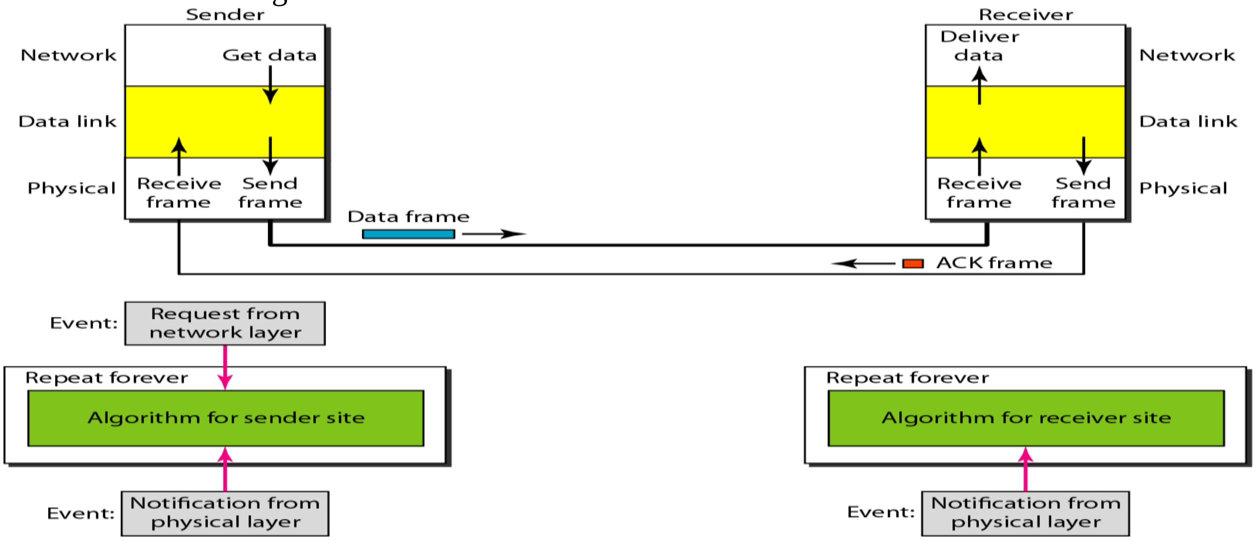
- The sender sends one frame, stops until it receives confirmation from the receiver (okay to go ahead), and then sends the next frame.
- We still have unidirectional communication for data frames, but ACK frames (simple tokens of acknowledgment) travel from the other direction. We add flow control to our previous protocol.

## Simplex Stop and Wait protocol for an error-free channel: Design

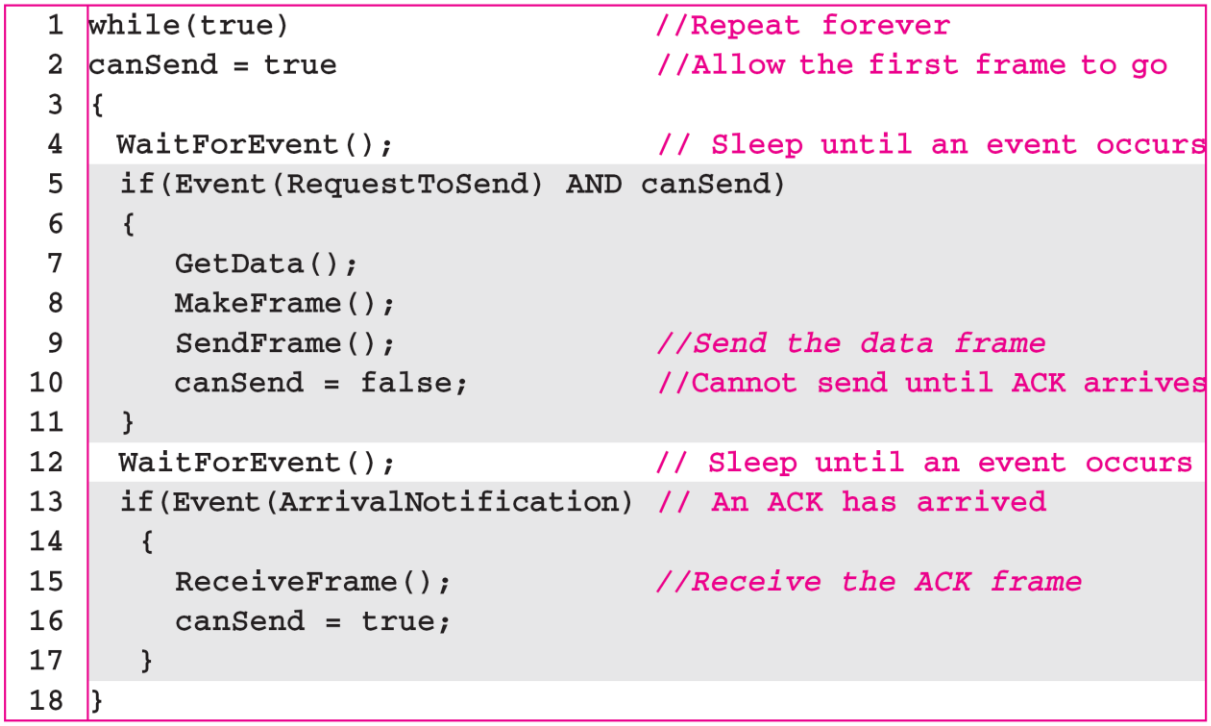
- **Sender Site**: The data link layer in the sender site waits for the network layer for a data packet.
- It then checks whether it can send the frame. If it receives a positive notification from the physical layer, it makes frames out of the data and sends it.
- It then waits for an acknowledgement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

## Simplex Stop and Wait protocol for an error-free channel: Design

- **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.

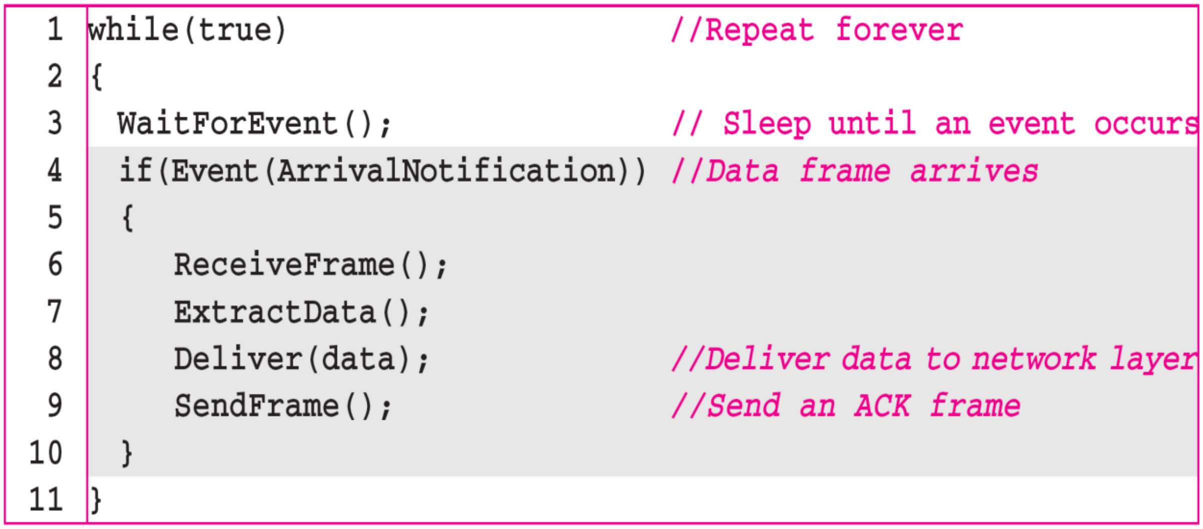


**Sender Site: Algorithm of Simplex Stop – and – Wait Protocol for Noiseless Channel**



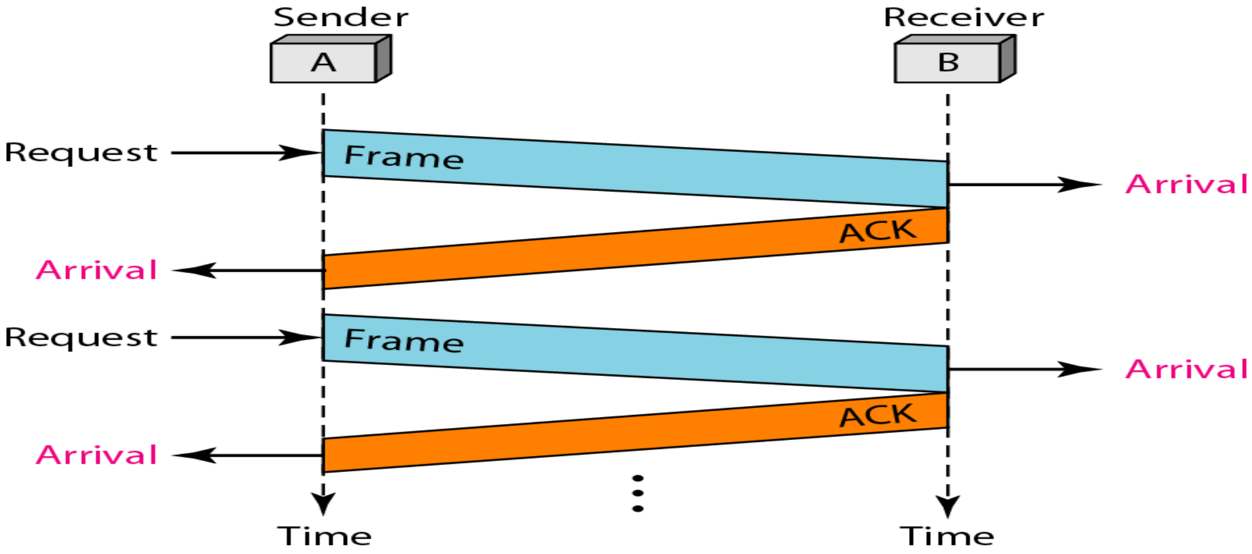
```
while(true)                          //Repeat forever
canSend = true                       //Allow the first frame to go
{
 WaitForEvent();                     // Sleep until an event occurs
 if(Event(RequestToSend) AND canSend)
 {
    GetData();
    MakeFrame();
    SendFrame();                     //Send the data frame
    canSend = false;                 //Cannot send until ACK arrives
 }
 WaitForEvent();                     // Sleep until an event occurs
 if(Event(ArrivalNotification)       // An ACK has arrived
  {
    ReceiveFrame();                  //Receive the ACK frame
    canSend = true;
  }
}
```

**Receiver Site : Algorithm of Simplex Stop – and – Wait Protocol for Noiseless Channel**



```
while(true)                          //Repeat forever
{
 WaitForEvent();                     // Sleep until an event occurs
 if(Event(ArrivalNotification))      //Data frame arrives
 {
    ReceiveFrame();
    ExtractData();
    Deliver(data);                   //Deliver data to network layer
    SendFrame();                     //Send an ACK frame
 }
}
```

### Simplex Stop – and – Wait Protocol for Noiseless Channel



### Simplex Stop – and – Wait 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.

### STOP-AND-WAIT AUTOMATIC REPEAT REQUEST

- **Stop-and-Wait Automatic Repeat Request** (Stop-and Wait ARQ), adds a simple error control mechanism to the Stop-and-Wait Protocol.
- To detect and correct corrupted frames, we need to add redundancy bits to our data frame. When the frame arrives at the receiver site, it is checked and if it is corrupted, it is silently discarded. The detection of errors in this protocol is manifested by the silence of the receiver.
- The received frame could be the correct one, or a duplicate, or a frame out of order. The solution is to number the frames.
- When the receiver receives a data frame that is out of order, this means that frames were either lost or duplicated.
- The completed and lost frames need to be resent in this protocol.
- At the same time, it starts a timer. If the timer expires and there is no ACK for the sent frame, the frame is resent, the copy is held, and the timer is restarted.
- Since the protocol uses the stop-and-wait mechanism, there is only one specific frame that needs an ACK even though several copies of the same frame can be in the network.
- Since an ACK frame can also be corrupted and lost. The ACK frame for this protocol has a sequence number field.
- The protocol specifies that frames need to be numbered. This is done by using sequence numbers. A field is added to the data frame to hold the sequence number of that frame.

- The sequence numbers must be suitable for both data frames and ACK frames, we use this convention: The acknowledgment numbers always announce the sequence number of the next frame expected by the receiver.
-

## ELEMENTARY DATA LINK PROTOCOLS

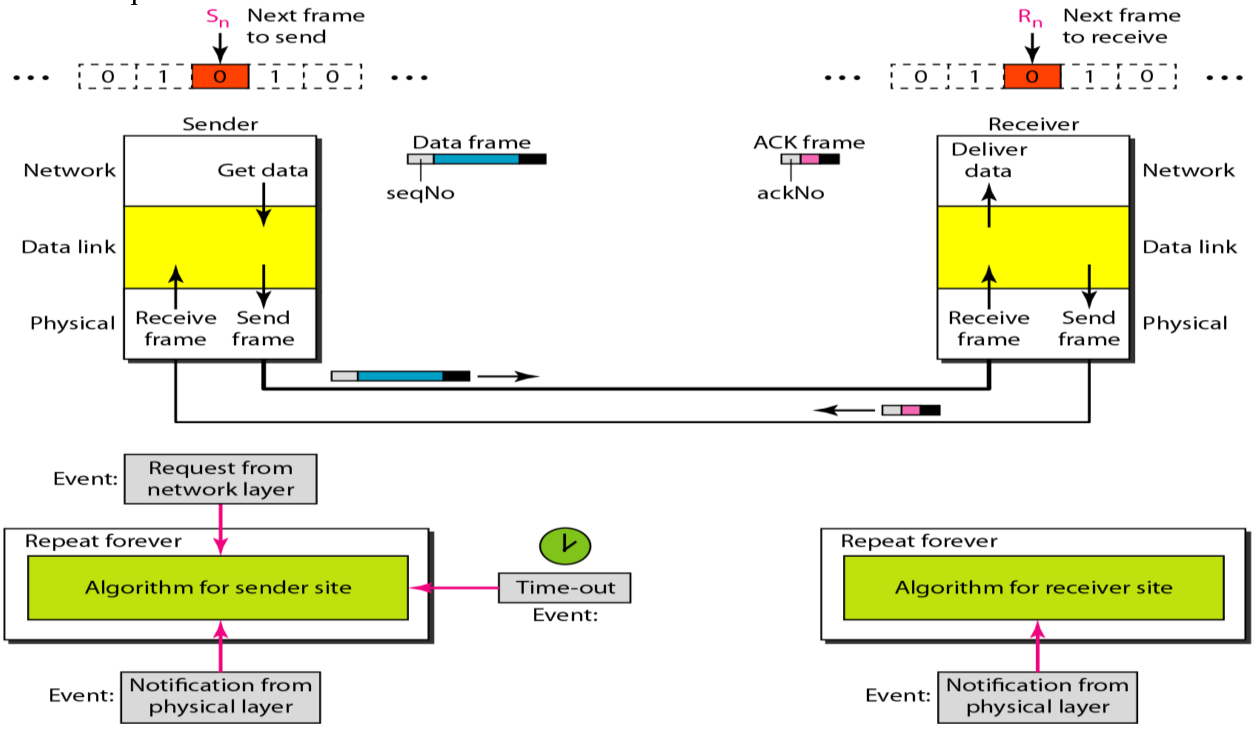
- **Simplex Stop – and – Wait Protocol for Noisy Channel : Design**

- **Sender Site** − 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
-

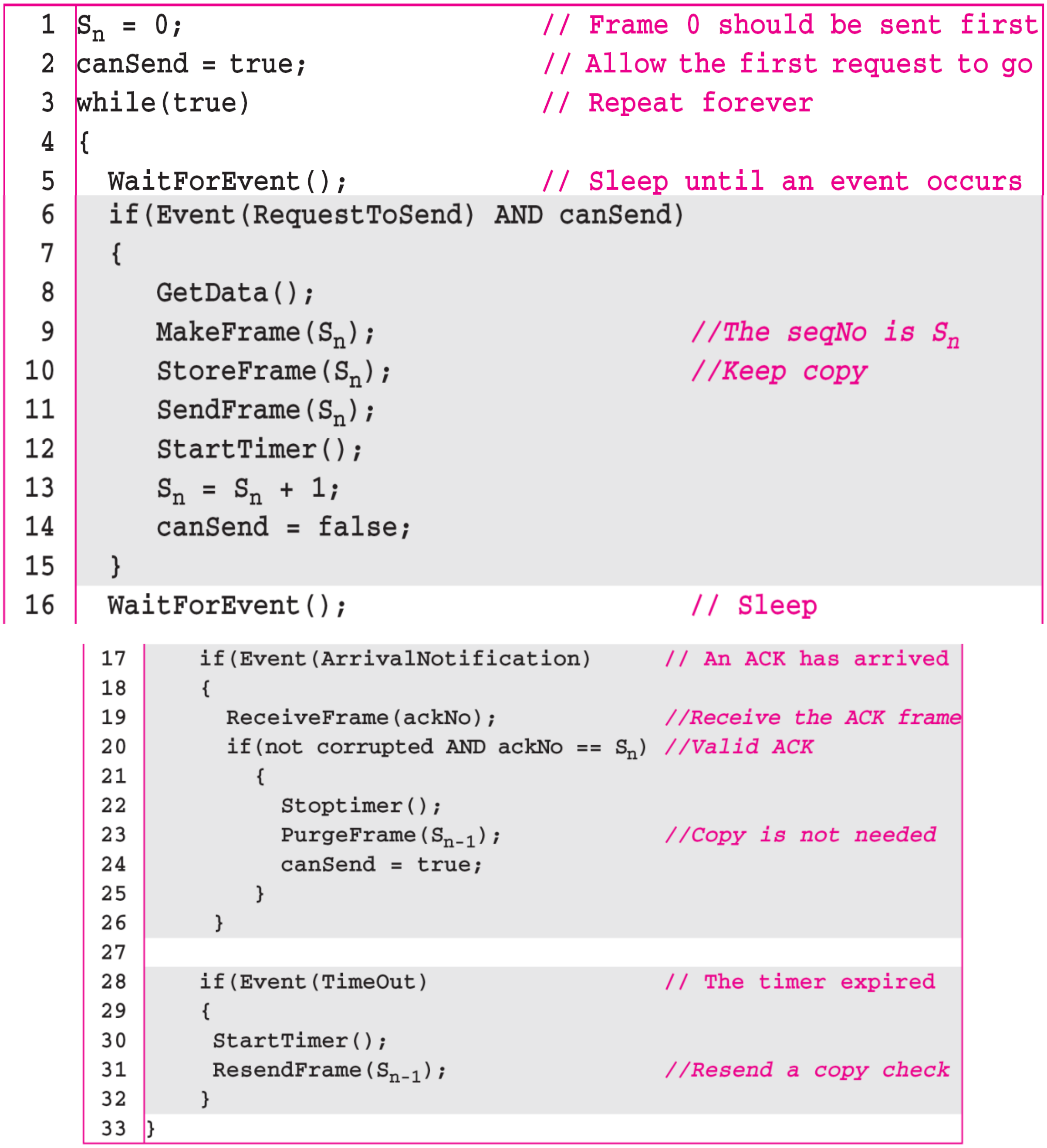
**Simplex Stop – and – Wait Protocol for Noisy Channel : Design**

**Receiver Site** − 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.

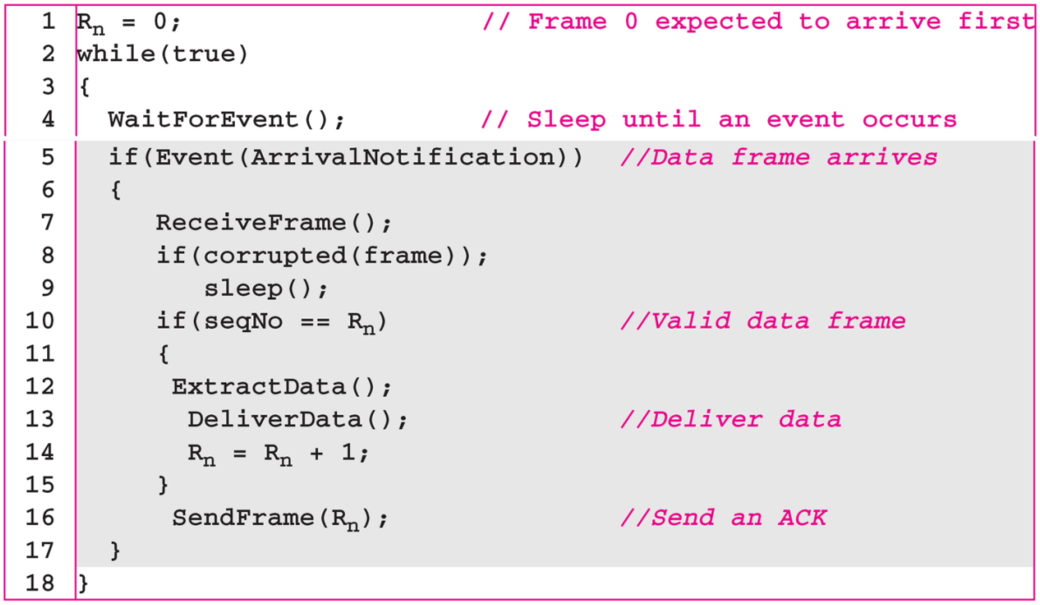


**Sender Site Algorithm of Simplex Stop – and – Wait Protocol for Noisy Channel**



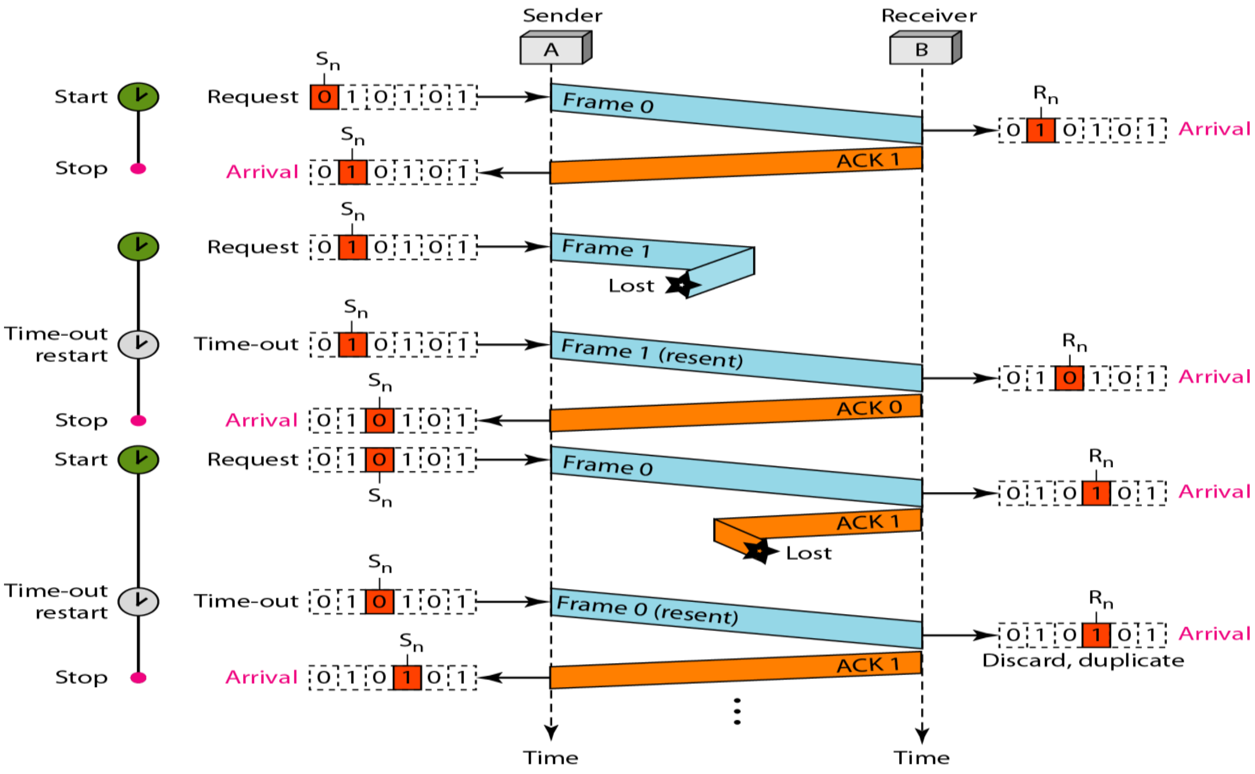
```
S_n = 0;                          // Frame 0 should be sent first
canSend = true;                   // Allow the first request to go
while(true)                       // Repeat forever
{
  WaitForEvent();                 // Sleep until an event occurs
  if(Event(RequestToSend) AND canSend)
  {
     GetData();
     MakeFrame(S_n);                         //The seqNo is S_n
     StoreFrame(S_n);                        //Keep copy
     SendFrame(S_n);
     StartTimer();
     S_n = S_n + 1;
     canSend = false;
  }
  WaitForEvent();                            // Sleep
    if(Event(ArrivalNotification)     // An ACK has arrived
    {
      ReceiveFrame(ackNo);            //Receive the ACK frame
      if(not corrupted AND ackNo == S_n) //Valid ACK
        {
          Stoptimer();
          PurgeFrame(S_{n-1});        //Copy is not needed
          canSend = true;
        }
     }

    if(Event(TimeOut)                 // The timer expired
    {
     StartTimer();
     ResendFrame(S_{n-1});            //Resend a copy check
    }
}
```

**Simplex stop – and – wait protocol for noisy channel – Flow diagram**



```
Rn = 0;                        // Frame 0 expected to arrive first
while(true)
{
  WaitForEvent();              // Sleep until an event occurs
  if(Event(ArrivalNotification))  //Data frame arrives
  {
     ReceiveFrame();
     if(corrupted(frame));
        sleep();
     if(seqNo == Rn)              //Valid data frame
     {
      ExtractData();
       DeliverData();             //Deliver data
       Rn = Rn + 1;
     }
      SendFrame(Rn);              //Send an ACK
  }
}
```

## Simplex stop – and – wait protocol for noisy channel − Flow diagram



## SLIDING WINDOW PROTOCOLS

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.

### Types of Sliding Window Protocols

1. A one-bit sliding window protocol
2. A protocol using Go-Back-N
3. A protocol using Selective Repeat

### A one-bit sliding window protocol

- In one – bit sliding window protocol, the size of the window is 1.
- So the sender transmits a frame, waits for its acknowledgment, then transmits the next frame.
- Thus it uses the concept of **stop and waits** for the protocol. This protocol provides for full – duplex communications.
- Hence, the acknowledgment is attached along with the next data frame to be sent by **piggybacking**.

### A one-bit sliding window protocol: Working Principle

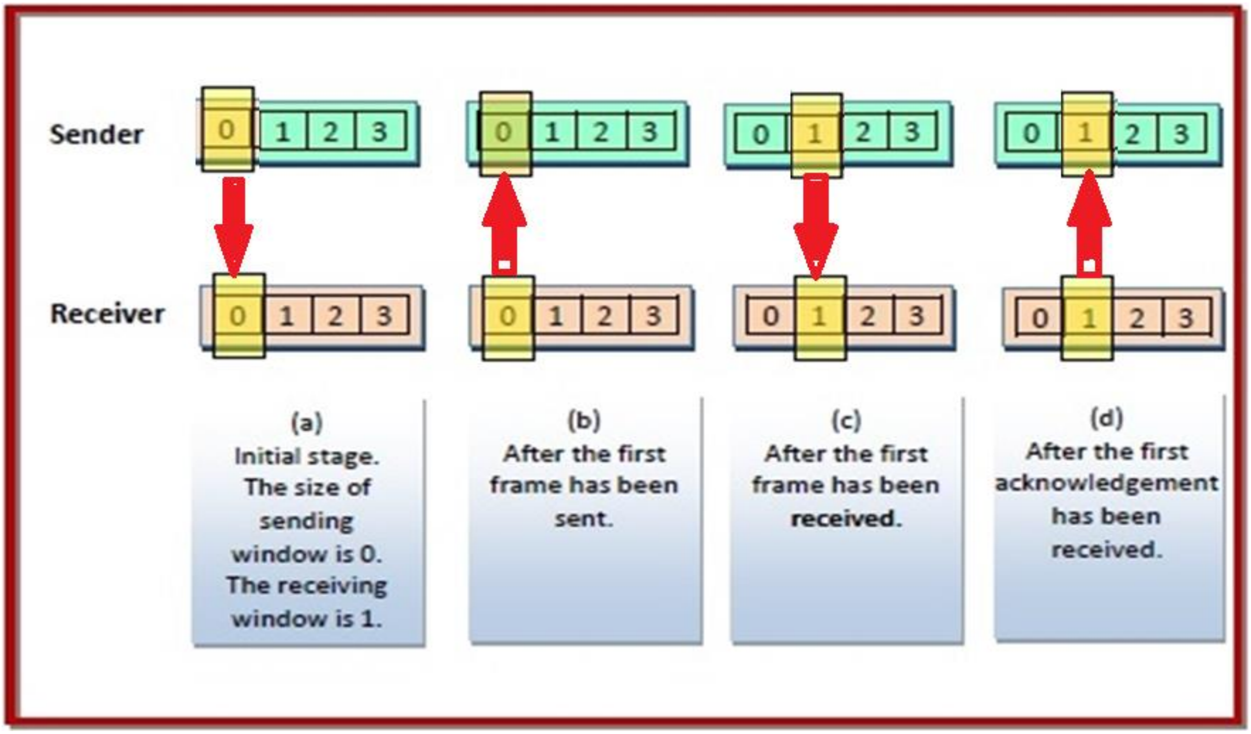
- The data frames to be transmitted additionally have an acknowledgment field, **ack** field that is of a few bits length.
- The **ack** field contains the sequence number of the last frame received without error.

### A one-bit sliding window protocol: Working Principle

- If this sequence number matches with the sequence number of the frame to be sent, then it is **inferred** that there is no error and the frame is transmitted.
- Otherwise, it is **inferred** that there is an error in the frame and the previous frame is retransmitted.

### A one-bit sliding window protocol : Example

- The following diagram depicts a scenario with sequence numbers 0, 1, 2, 3, 0, 1, 2 and so on.
- It depicts the sliding windows in the sending and the receiving stations during frame transmission.
- **A one-bit sliding window protocol : Example**



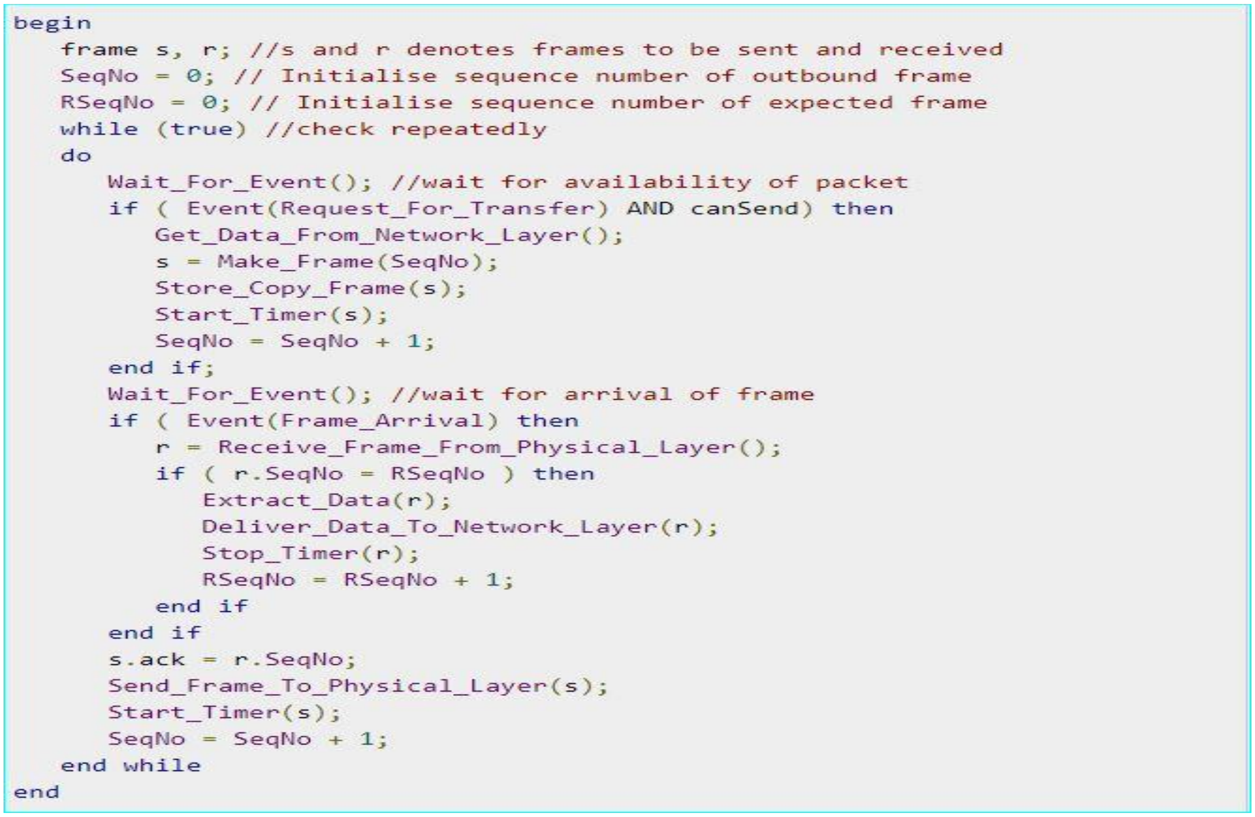
(a)
Initial stage.
The size of
sending
window is 0.
The receiving
window is 1.

(b)
After the first
frame has been
sent.

(c)
After the first
frame has been
received.

(d)
After the first
acknowledgement
has been
received.

### The algorithm of One – bit Sliding Window Protocol



```
begin
   frame s, r; //s and r denotes frames to be sent and received
   SeqNo = 0; // Initialise sequence number of outbound frame
   RSeqNo = 0; // Initialise sequence number of expected frame
   while (true) //check repeatedly
   do
      Wait_For_Event(); //wait for availability of packet
      if ( Event(Request_For_Transfer) AND canSend) then
         Get_Data_From_Network_Layer();
         s = Make_Frame(SeqNo);
         Store_Copy_Frame(s);
         Start_Timer(s);
         SeqNo = SeqNo + 1;
      end if;
      Wait_For_Event(); //wait for arrival of frame
      if ( Event(Frame_Arrival) then
         r = Receive_Frame_From_Physical_Layer();
         if ( r.SeqNo = RSeqNo ) then
            Extract_Data(r);
            Deliver_Data_To_Network_Layer(r);
            Stop_Timer(r);
            RSeqNo = RSeqNo + 1;
         end if
      end if
      s.ack = r.SeqNo;
      Send_Frame_To_Physical_Layer(s);
      Start_Timer(s);
      SeqNo = SeqNo + 1;
   end while
end
```

**A protocol using Go-Back-N**

- In Go-Back-N ARQ protocol, if one frame is lost or damaged, then it retransmits all the frames after which it does not receive the positive ACK.
- It is a case of sliding window protocol having to send window size of N and receiving window size of 1.

**A protocol using Go-Back-N: Working Principle**

- Go – Back – N ARQ provides for sending multiple frames before receiving the acknowledgment for the first frame. The frames are **sequentially numbered** and a finite number of frames.
- The maximum number of frames that can be sent depends upon the size of the sending window.
- If the acknowledgment of a frame is not received within an agreed upon time period, all frames starting from that frame are retransmitted.

**A protocol using Go-Back-N: Working Principle**

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

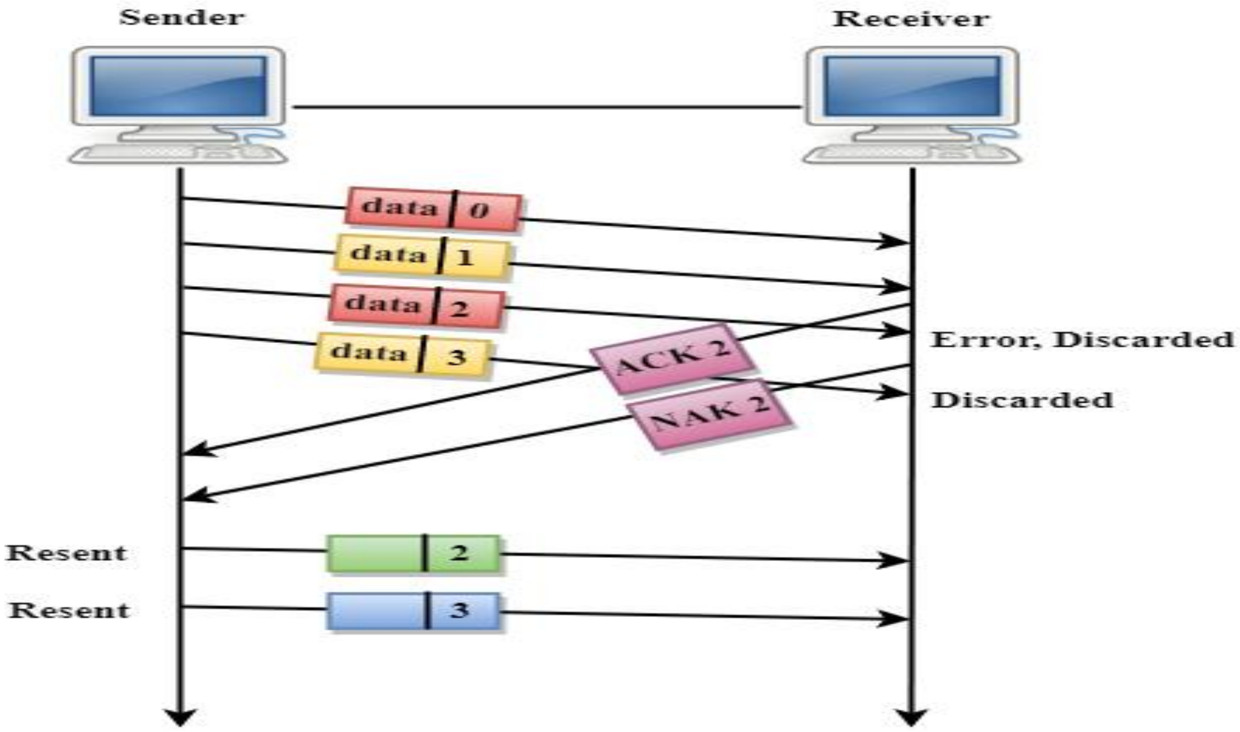
**A protocol using Go-Back-N: Working Principle**

- 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 1.
- **Sender Site Algorithm of Go-Back-N Protocol**

**A Protocol using Go-Back-N**

- In the above figure, three frames have been transmitted before an error discovered in the third frame.
- In this case, ACK 2 has been returned telling that the frames 0,1 have been received successfully without any error.
- The receiver discovers the error in data 2 frame, so it returns the NAK 2 frame.
- The frame 3 is also discarded as it is transmitted after the damaged frame. Therefore, the sender retransmits the frames 2,3.

**Protocol Go-Back-N**



**A Protocol using Go-Back-N**

- **Lost Data Frame:** In Sliding window protocols, data frames are sent sequentially. If any of the frames is lost, then the next frame arrive at the receiver is out of sequence.
- The receiver checks the sequence number of each of the frame, discovers the frame that has been skipped, and returns the NAK for the missing frame.
- The sending device retransmits the frame indicated by NAK as well as the frames transmitted after the lost frame.

**A Protocol using Go-Back-N**

- **Lost Acknowledgement:** The sender can send as many frames as the windows allow before waiting for any acknowledgement.
- Once the limit of the window is reached, the sender has no more frames to send; it must wait for the acknowledgement.
- If the acknowledgement is lost, then the sender could wait forever. To avoid such situation, the sender is equipped with the timer that starts counting whenever the window capacity is reached.
- If the acknowledgement has not been received within the time limit, then the sender retransmits the frame since the last ACK.

- **A protocol using Selective Repeat**
- Selective-Repeat ARQ technique is more efficient than Go-Back-n ARQ.
- In this technique, only those frames are retransmitted for which negative acknowledgement (NAK) has been received.
- The receiver storage buffer keeps all the damaged frames.
- The sender provide searching mechanism that selects only the requested frame for retransmission.

**A protocol using Selective Repeat**

- Selective repeat protocol, also called Selective Repeat ARQ (Automatic Repeat reQuest), is a data link layer protocol that uses sliding window method for reliable delivery of data frames. Here, only the erroneous or lost frames are retransmitted.
- It uses two windows of equal size: a sending window that stores the frames to be sent and a receiving window that stores the frames receive by the receiver.

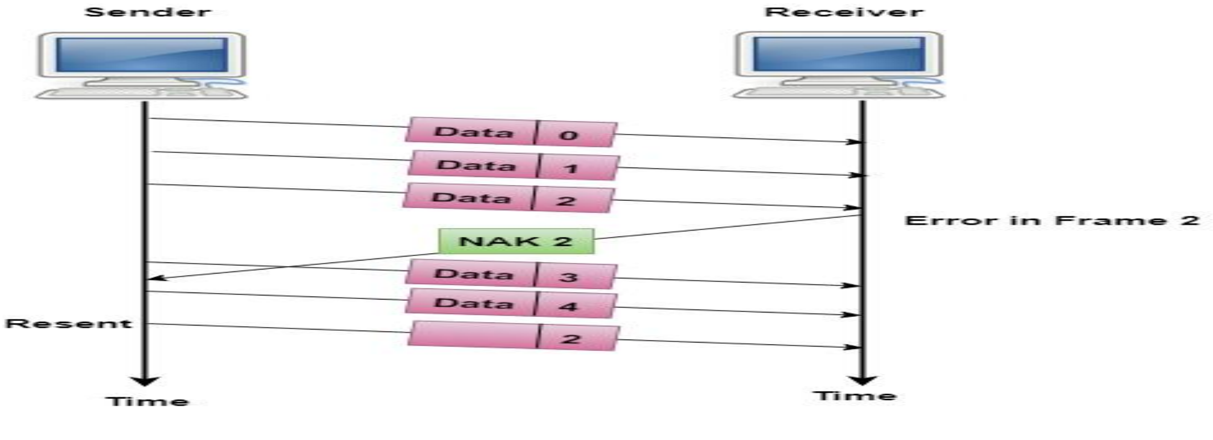
**A protocol using Selective Repeat : Working Principle**

- Selective Repeat protocol provides for sending multiple frames depending upon the availability of frames in the sending window, even if it does not receive acknowledgement for any frame.
- The maximum number of frames that can be sent depends upon the size of the sending window.

**A protocol using Selective Repeat : Working Principle**

- The sender continues to send frames from sending window. Once, it has sent all the frames from the window, it retransmits the frame whose sequence number is given by the Negative acknowledgements.
- The receiver records the sequence number of the earliest incorrect or un-received frame. It sends the sequence number of the missing frame along with every acknowledgement frame.

**A protocol using Selective Repeat**



# Stop and Wait protocol Vs Sliding Window protocol

### Stop and Wait protocol

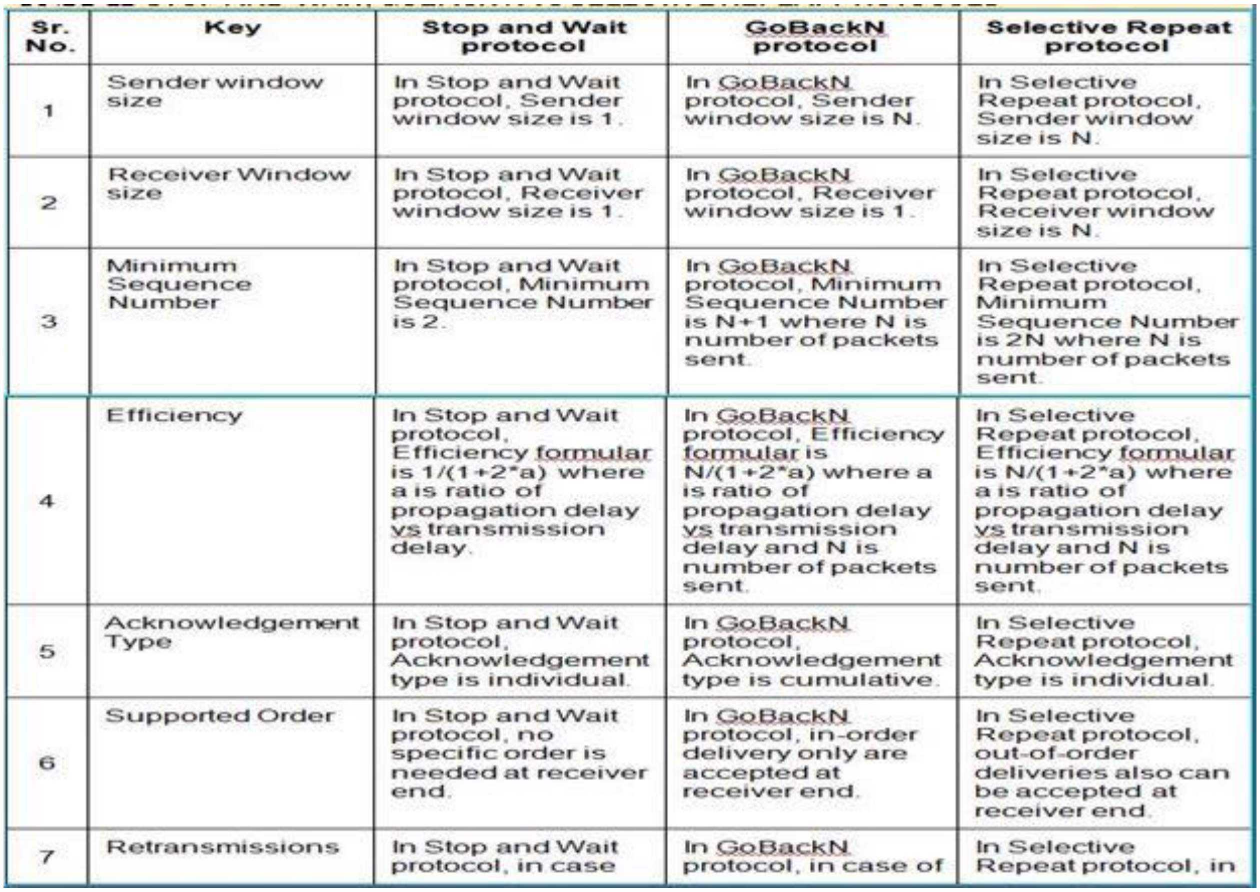
- Stop and Wait protocol is a protocol for flow control mechanism. In this protocol, sender sends one frame at a time and waits for acknowledgment from the receiver. Once acknowledged, sender sends another frame to the receiver.

### Sliding Window protocol

- Stop and Wait protocol is also a protocol for flow control mechanism. In this protocol, sender sends multiple frames at a time and retransmits the frames which are found to be corrupted or damaged.

| **Sr. No.** | **Key** | **Stop and Wait protocol** | **Sliding Window protocol** |
|---|---|---|---|
| 1 | Mechanism | In Stop and Wait protocol, sender sends single frame and waits for acknowledgment from the receiver. | In Sliding window protocol, sender sends multiple frames at a time and retransmits the damamged frames. |
| 2 | Efficiency | Stop and Wait protocol is less efficient. | Sliding Window protocol is more efficient than Stop and Wait protocol. |
| 3 | Window Size | Sender's window size in Stop and Wait protocol is 1. | Sender's window size in Sliding Window protocol varies from 1 to n. |
| 4 | Sorting | Sorting of frames is not needed. | Sorting of frames helps increasing the efficiency of the protocol. |
| 5 | Efficiency | Stop and Wait protocol efficiency is formulated as 1/(1+2a) where a is ratio of propagation delay vs transmission delay. | Sliding Window protocol efficiency is formulated as N/(1+2a) where N is no. of window frames and a is ratio of propagation delay vs transmission delay. |
| 6 | Duplex | Stop and Wait protocol is half duplex in nature. | Sliding Window protocol is full duplex in nature. |

## Stop and Wait, GoBack-N Vs Selective Repeat protocols



| Sr. No. | Key | Stop and Wait protocol | GoBackN protocol | Selective Repeat protocol |
|---|---|---|---|---|
| 1 | Sender window size | In Stop and Wait protocol, Sender window size is 1. | In GoBackN protocol, Sender window size is N. | In Selective Repeat protocol, Sender window size is N. |
| 2 | Receiver Window size | In Stop and Wait protocol, Receiver window size is 1. | In GoBackN protocol, Receiver window size is 1. | In Selective Repeat protocol, Receiver window size is N. |
| 3 | Minimum Sequence Number | In Stop and Wait protocol, Minimum Sequence Number is 2. | In GoBackN protocol, Minimum Sequence Number is N+1 where N is number of packets sent. | In Selective Repeat protocol, Minimum Sequence Number is 2N where N is number of packets sent. |
| 4 | Efficiency | In Stop and Wait protocol, Efficiency formular is 1/(1+2*a) where a is ratio of propagation delay vs transmission delay. | In GoBackN protocol, Efficiency formular is N/(1+2*a) where a is ratio of propagation delay vs transmission delay and N is number of packets sent. | In Selective Repeat protocol, Efficiency formular is N/(1+2*a) where a is ratio of propagation delay vs transmission delay and N is number of packets sent. |
| 5 | Acknowledgement Type | In Stop and Wait protocol, Acknowledgement type is individual. | In GoBackN protocol, Acknowledgement type is cumulative. | In Selective Repeat protocol, Acknowledgement type is individual. |
| 6 | Supported Order | In Stop and Wait protocol, no specific order is needed at receiver end. | In GoBackN protocol, in-order delivery only are accepted at receiver end. | In Selective Repeat protocol, out-of-order deliveries also can be accepted at receiver end. |
| 7 | Retransmissions | In Stop and Wait protocol, in case | In GoBackN protocol, in case of | In Selective Repeat protocol, in |

### EXAMPLE DATA LINK PROTOCOLS

- Here we will examine the data link protocols found on **point-to-point** lines in the Internet in two common situations.
- The **first situation** is when packets are sent over SONET optical fiber links in wide-area networks.
- These links are widely used, for example, to connect routers in the different locations of an ISP's network.
- The **second situation** is for ADSL links running on the local loop of the telephone network at the edge of the Internet. These links connect millions of individuals and businesses to the Internet.
- The Internet needs **point-to-point** links for these **uses**, as well as **dial-up modems, leased lines, and cable modems**, and so on.

- A standard protocol called **PPP (Point-to-Point Protocol)** is used to send packets over these links.

## HDLC (High-Level Data Link Control)

**HDLC** (High-Level Data Link Control) is a bit-oriented protocol that is used for communication over the **point-to-point and multipoint links**. This protocol implements the mechanism of ARQ(Automatic Repeat Request). With the help of the HDLC protocol,full-duplex communication is possible.
**HDLC** is the most widely used protocol and offers reliability, efficiency, and a high level of Flexibility.
In order to make the HDLC protocol applicable for various network configurations, there are three types of stations and these are as follows:

- **Primary Station** This station mainly looks after data like management. In the case of the communication between the primary and secondary station; it is the responsibility of the primary station to connect and disconnect the data link. The frames issued by the primary station are commonly known as commands.
- **Secondary Station** The secondary station operates under the control of the primary station. The Frames issued by the secondary stations are commonly known as responses.
- **Combined Station** The combined station acts as both Primary stations as well as Secondary stations. The combined station issues both commands as well as responses.

### Transfer Modes in HDLC

The HDLC protocol offers two modes of transfer that mainly can be used in different configurations. These are as follows:

- Normal Response Mode(NRM)
- Asynchronous Balance Mode(ABM)
-

Let us now discuss both these modes one by one:

**1. Normal Response Mode(NRM)**
In this mode, the configuration of the station is unbalanced. There are one primary station and multiple secondary stations. Where the primary station can send the commands and the secondary station can only respond.This mode is used for both **point-to-point** as well as **multiple-point links.**

**2. Asynchronous Balance Mode(ABM)**
In this mode, the configuration of the station is balanced. In this mode, the link is point-to-point, and each station can function as a primary and as secondary.
Asynchronous Balance mode(ABM) is a commonly used mode today.

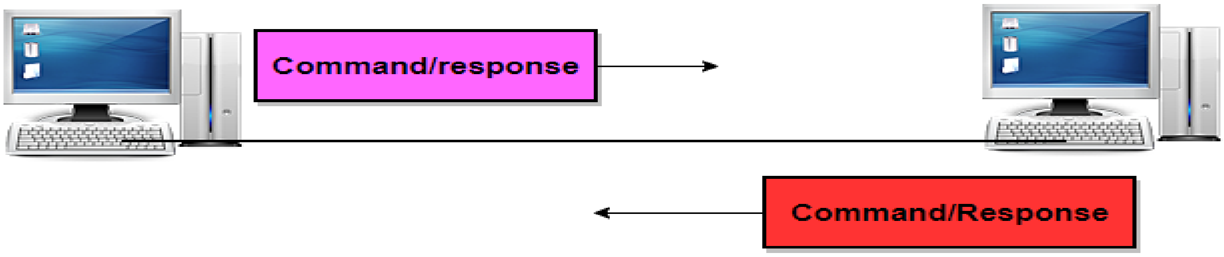


Figure: Asynchronous Balance Mode

**HDLC Frames**

In order to provide the flexibility that is necessary to support all the options possible in the modes and Configurations that are just described above. There are three types of frames defined in the HDLC:

- **Information Frames(I-frames)** These frames are used to transport the user data and the control information that is related to the user data. If the first bit of the control field is 0 then it is identified as I-frame.
- **Supervisory Frames(S-frames)** These frames are only used to transport the control information. If the first two bits of the control field are 1 and 0 then the frame is identified as S-frame
- **Unnumbered Frames(U-Frames)** These frames are mainly reserved for system management. These frames are used for exchanging control information between the communicating devices.

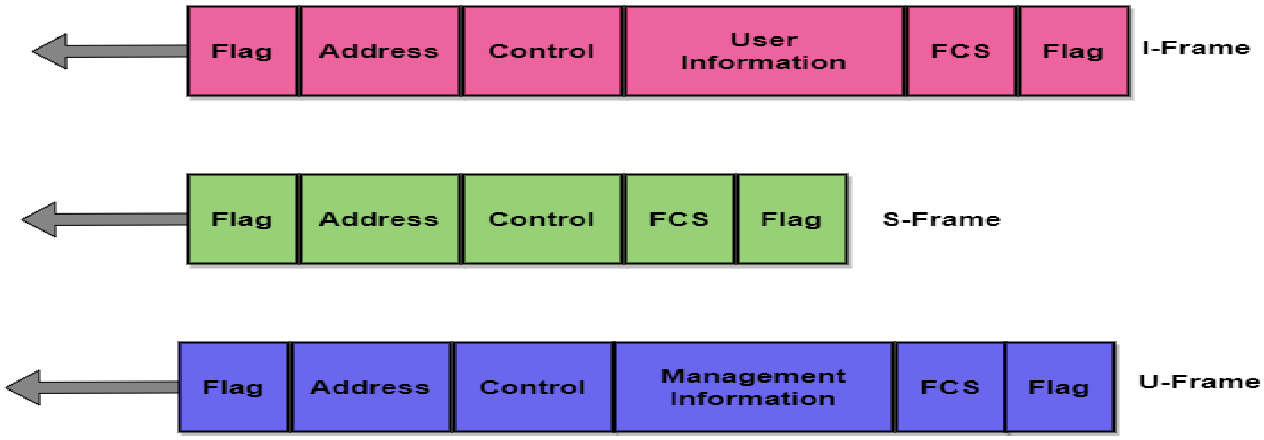
Each type of frame mainly serves as an envelope for the transmission of a different type of message.

**Frame Format**

There are up to six fields in each HDLC frame. There is a beginning flag field, the address field then, a control field, an information field, a frame check sequence field(FCS), and an ending field.

In the case of the multiple-frame transmission, the ending flag of the one frame acts as the beginning flag of the next frame.

Let us take a look at different HDLC frames:



**1. Flag Field**

This field of the HDLC frame is mainly a sequence of 8-bit having the bit pattern 01111110 and it is used to identify the beginning and end of the frame. The flag field mainly serves as a synchronization pattern for the receiver.

**2. Address Field**

It is the second field of the HDLC frame and it mainly contains the address of the secondary station. This field can be 1 byte or several bytes long which mainly depends upon the need of the network. In case if the frame is sent by the primary station, then this field contains the address(es) of the secondary stations. If the frame is sent by the secondary station, then this field contains the address of the primary station.

**3. Control Field**

This is the third field of the HDLC frame and it is a 1 or 2-byte segment of the frame and is mainly used for flow control and error control. Bits interpretation in this field mainly depends upon the type of the frame.

**4. Information Field**

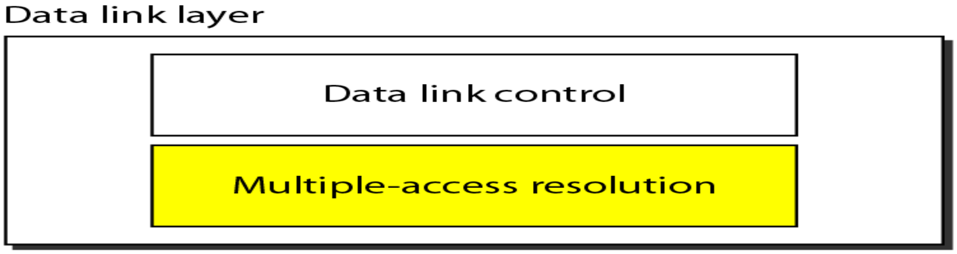
This field of the HDLC frame contains the user's data from the network layer or the management information. The length of this field varies from one network to another.

**5. FCS Field**

FCS means Frame check sequence and it is the error detection field in the HDLC protocol. There is a 16 bit CRC code for error detection.

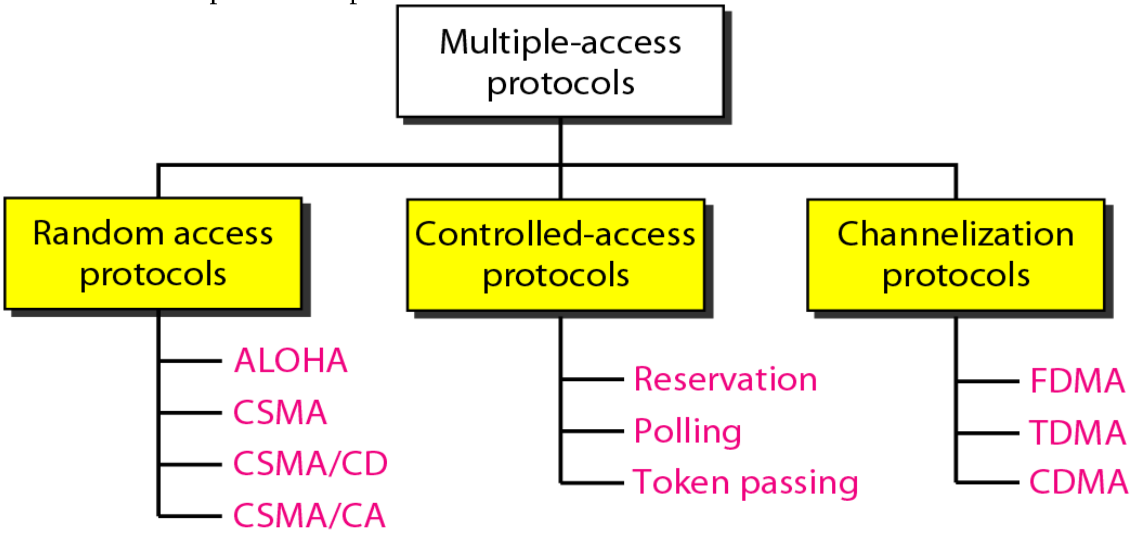
## MULTIPLE-ACCESS PROTOCOLS

- Data link layer as two sub-layers. The upper sub-layer is responsible for data link control, and the lower sub-layer is responsible for resolving access to the shared media. If the channel is dedicated.
- Data link layer divided into two functionality-oriented sub-layers.



- The upper sub-layer that is responsible for flow and error control is called the **logical link control (LLC)** layer
- The lower sub-layer that is mostly responsible for multiple-access resolution is called the media access control (MAC) layer.
- When nodes or stations are connected and use a common link, called a multi-point or broadcast link.

- We need a multiple-access protocol to coordinate access to the link.

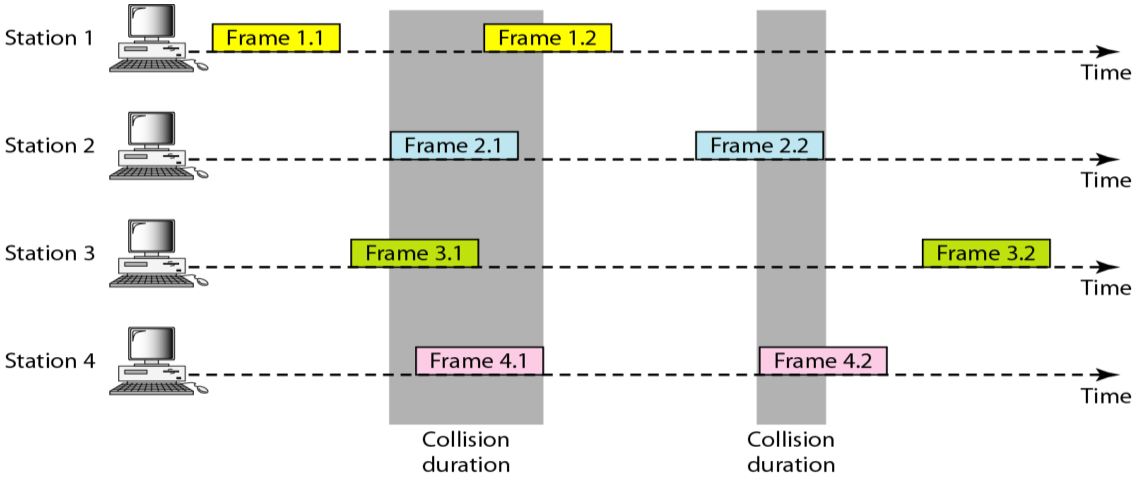


- In a **random access method**, each station has the right to the medium without being controlled by any other station.
- However, if more than one station tries to send, there is an access **conflict-collision**-and the frames will be either destroyed or modified.

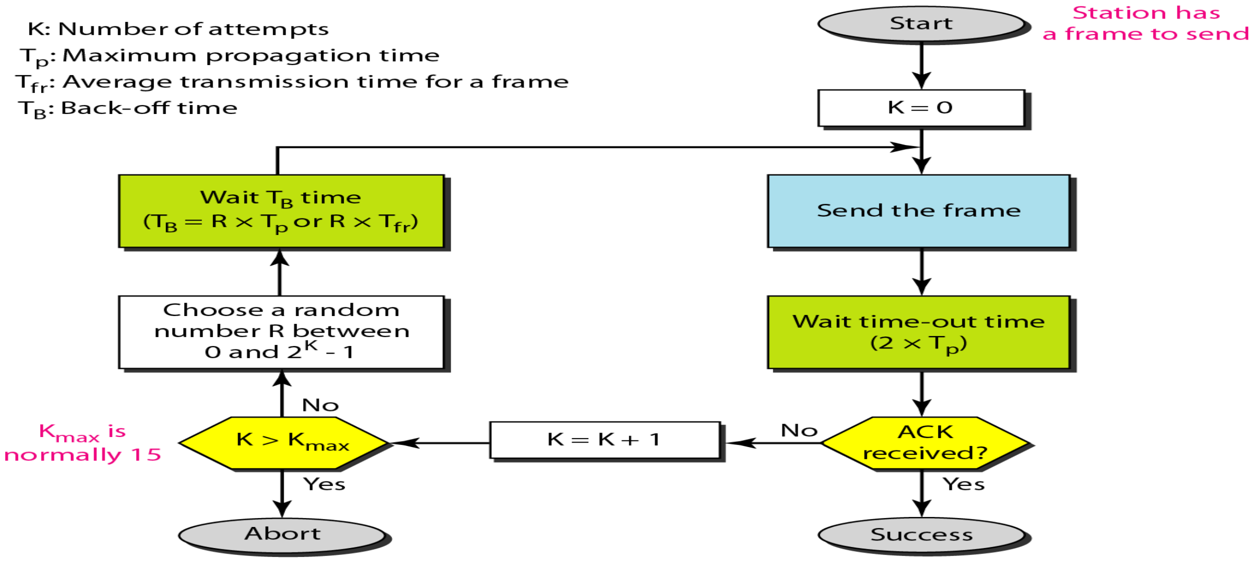
**ALOHA** – **ALOHA** stands for **Additive Links On-line Hawaii Area**. Its developed at the University of Hawaii in **1971**. It was designed for **wireless LAN** but is also applicable for shared medium. In this, multiple stations can transmit data at the same time and it can handle the collisions.

**Pure Aloha:**

- When a station sends data it waits for an acknowledgement. If the acknowledgement doesn't come within the allotted time then the station waits for a random amount of time called back-off time and re-sends the data.
- Since different stations wait for different amount of time.

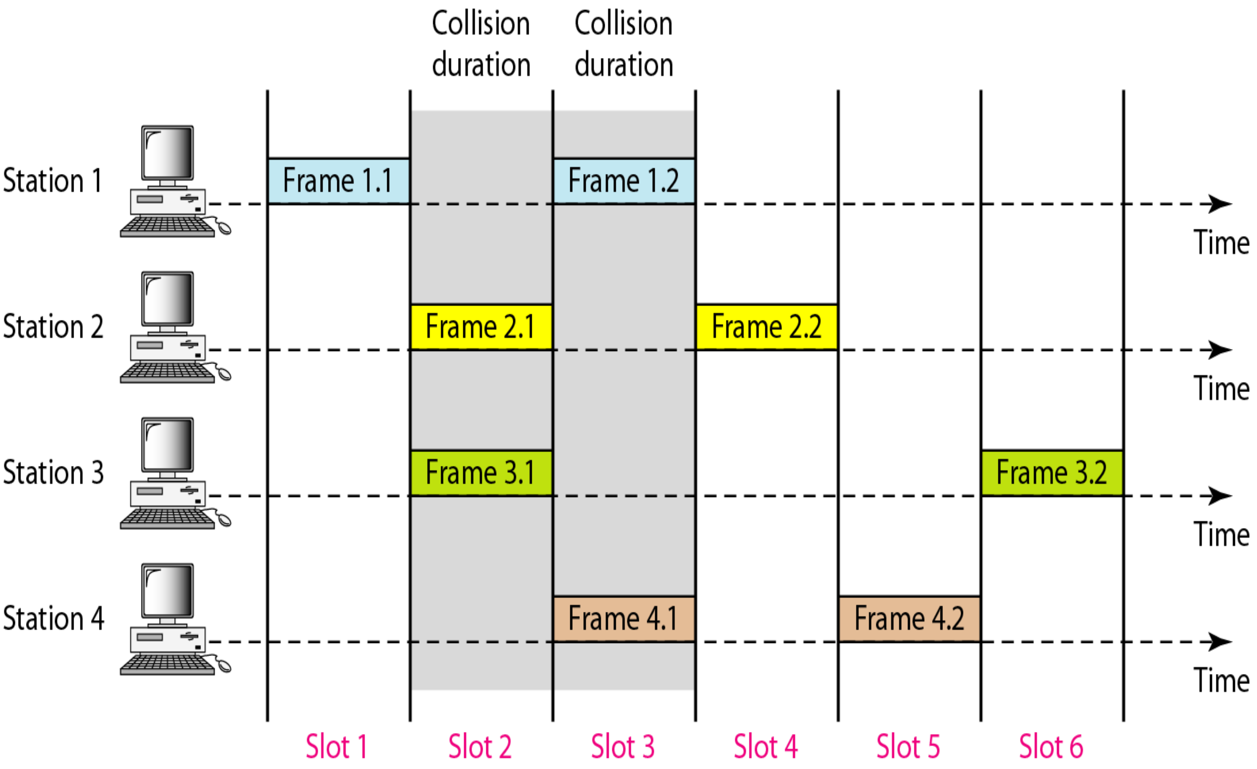


**Pure Aloha: Flow diagram**


K: Number of attempts
$T_p$: Maximum propagation time
$T_{fr}$: Average transmission time for a frame
$T_B$: Back-off time
Start
Station has a frame to send
K = 0
Send the frame
Wait time-out time ($2 \times T_p$)
ACK received?
No
Yes
Success
K = K + 1
$K > K_{max}$
$K_{max}$ is normally 15
Yes
Abort
No
Choose a random number R between 0 and $2^K - 1$
Wait $T_B$ time ($T_B = R \times T_p$ or $R \times T_{fr}$)

### Slotted Aloha:

- It is similar to pure aloha, except that we divide time into slots and sending of data is allowed only at the beginning of these slots.
- If a station misses out the allowed time, it must wait for the next slot. This reduces the probability of collision.



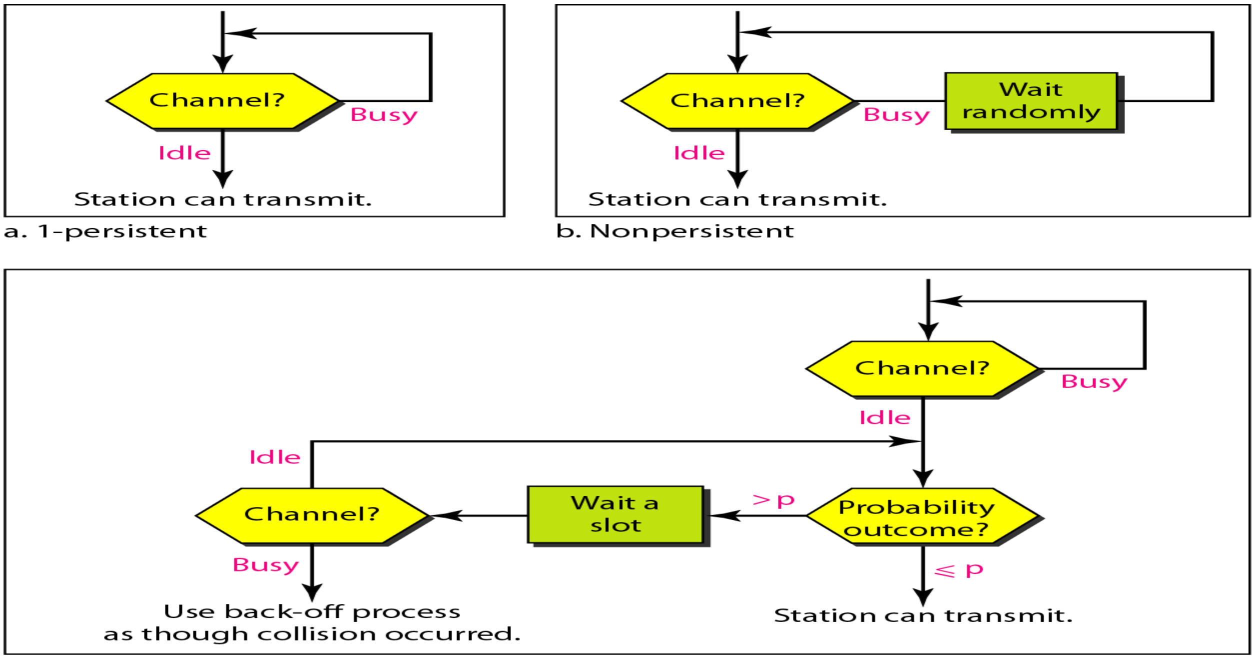
| **PARAMETERS** | **PURE ALOHA** | **SLOTTED ALOHA** |
|---|---|---|
| **Data transmission** | Stations can transmit the data randomly i.e. any number of stations can transmit data at any time. | Here, any random station can transmit the data at the beginning of any random time slot |
| **Time status** | Here, the time is continuous and is not globally synchronized with any other station. | Here, the time is discrete unlike pure ALOHA and is also globally synchronized |
| Vulnerable time | 2*Frame transmission time | Frame transmission time |
| PARAMETERS | PURE ALOHA | SLOTTED ALOHA |
| Probability of successful transmission of a data packet | G*e-2G<br>where, G = no. of stations willing to transmit data | G*e-G |
| Maximum efficiency | 18.4% | 36.8% |
| Collision status | It does not reduce the total number of collisions to half | Here, it reduces the total number of collisions to half and doubles the efficiency of pure ALOHA |

**2. CSMA – Carrier Sense Multiple Access** ensures fewer collisions as the station is required to first sense the medium (**for idle or busy**) before transmitting data.

- If it is idle then it sends data, otherwise it waits till the channel becomes idle. However there is still chance of collision in CSMA due to propagation delay.

**2. CSMA –** For example, if station A wants to send data, it will first sense the medium.

- If it finds the channel idle, it will start sending data. However, by the time the first bit of data is transmitted (delayed due to propagation delay) from station A
- If station B requests to send data and senses the medium it will also find it idle and will also send data. This will result in collision of data from station A and B.



a. 1-persistent

b. Nonpersistent

c. p-persistent

**CSMA access modes-**

- **1-persistent method** is simple and straightforward. In this method, after the station finds the line idle, it sends its frame immediately (with probability 1). This method has the highest chance of collision because two or more stations may find the line idle and send their frames immediately.
- **Non-Persistent:** In the non-persistent method, a station that has a frame to send senses the line. If the line is idle, it sends immediately. If the line is not idle, it waits a random amount of time and then senses the line again. The non-persistent approach
- reduces the chance of collision

**CSMA access modes-**

- **P-persistent:** The p-persistent method is used if the channel has time slots with a slot duration equal to or greater than the maximum propagation time.
- The p-persistent approach combines the advantages of the other two strategies.
- It reduces the chance of collision and improves efficiency.
- In this method, after the station finds the line idle it follows these steps:

1. With probability p, the station sends its frame.
2. With probability q = 1 - p, the station waits for the beginning of the next time slot and checks the line again.

a. If the line is idle, it goes to step 1.

b. If the line is busy, it acts as though a collision has occurred and uses the back-off procedure.

**CSMA/CD** : Carrier sense multiple access with collision detection (CSMA/CD) augments the algorithm to handle the collision.

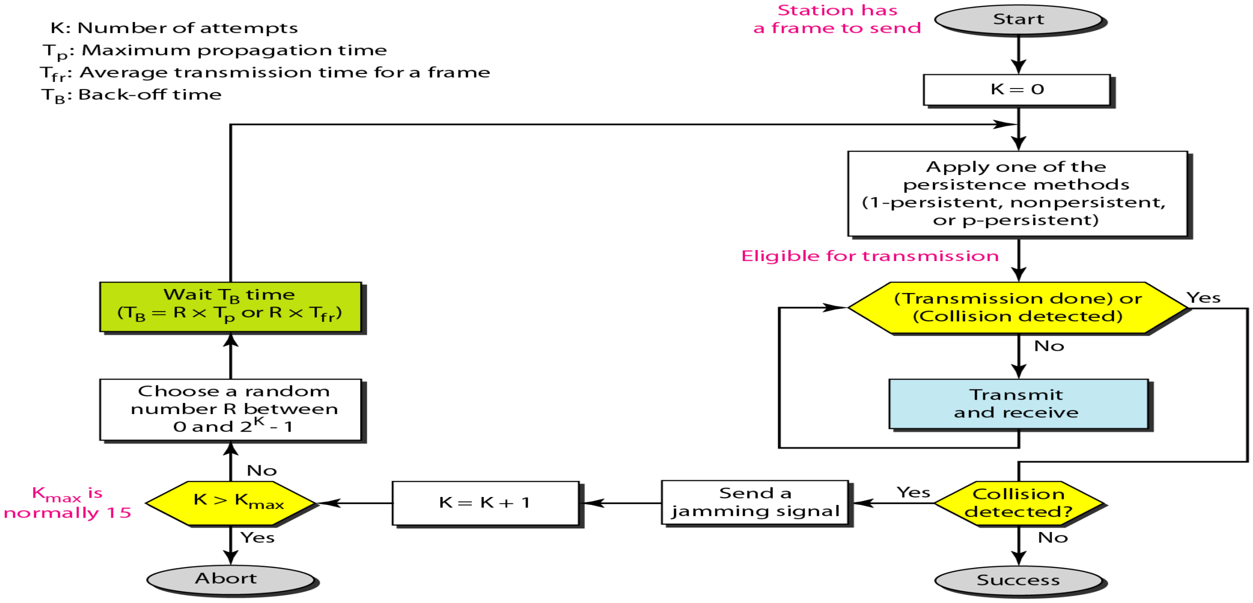
- In this method, a station monitors the medium after it sends a frame to see if the transmission was successful. If so, the station is finished. If, however, there is a collision, the frame is sent again.

**The algorithm of CSMA/CD is:**

- When a frame is ready, the transmitting station checks whether the channel is idle or busy.
- If the channel is busy, the station waits until the channel becomes idle.
- If the channel is idle, the station starts transmitting and continually monitors the channel to detect collision.
- If a collision is detected, the station starts the collision resolution algorithm.
- The station resets the retransmission counters and completes frame transmission.

**The algorithm of Collision Resolution is:**

- The station continues transmission of the current frame for a specified time along with a jam signal, to ensure that all the other stations detect collision.
- The station increments the retransmission counter.
- If the maximum number of retransmission attempts is reached, then the station aborts transmission.
- Otherwise, the station waits for a back-off period which is generally a function of the number of collisions and restart main algorithm.

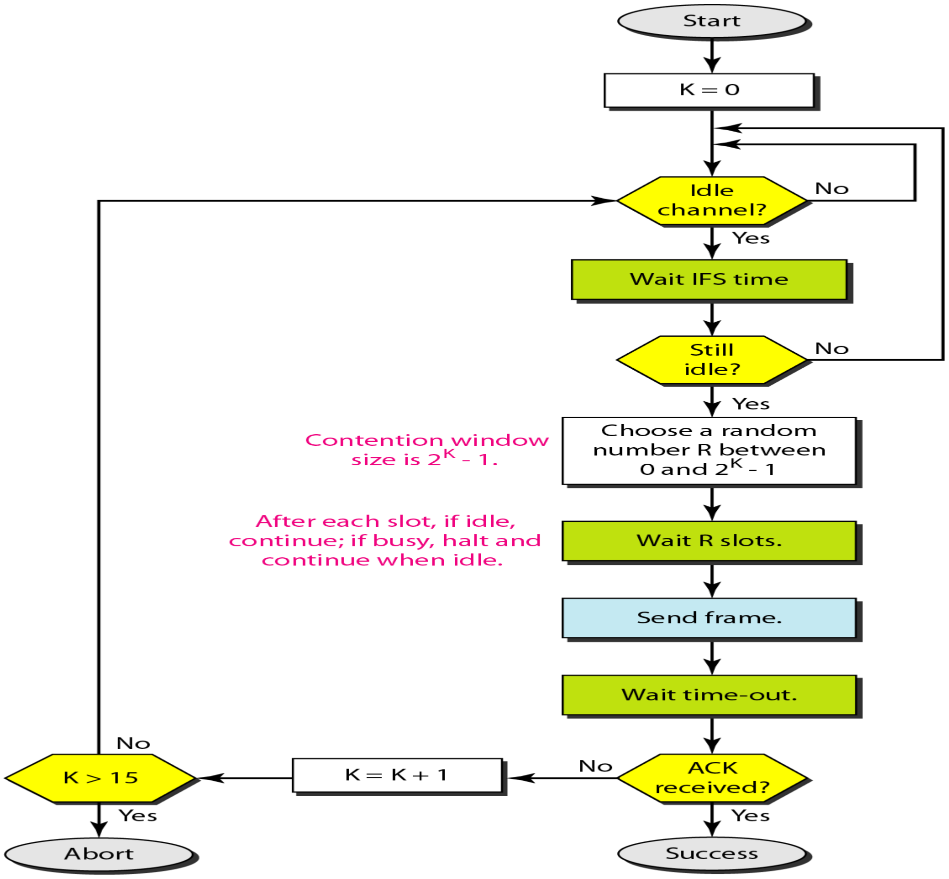

K: Number of attempts
$T_p$: Maximum propagation time
$T_{fr}$: Average transmission time for a frame
$T_B$: Back-off time
Station has a frame to send
Start
K = 0
Apply one of the persistence methods (1-persistent, nonpersistent, or p-persistent)
Eligible for transmission
(Transmission done) or (Collision detected)
Yes
No
Transmit and receive
Collision detected?
Yes
No
Success
Send a jamming signal
K = K + 1
$K > K_{max}$
$K_{max}$ is normally 15
Yes
Abort
No
Choose a random number R between 0 and $2^K$ - 1
Wait $T_B$ time ($T_B = R \times T_p$ or $R \times T_{fr}$)

**CSMA/CA :** In CSMA/CA, if the station finds the channel busy, it does not restart the timer of the contention window; it stops the timer and restarts it when the channel becomes idle.

**MULTIPLE-ACCESS PROTOCOLS- Random Access**

**The algorithm of CSMA/CA is:**

- When a frame is ready, the transmitting station checks whether the channel is idle or busy.
- If the channel is busy, the station waits until the channel becomes idle.
- If the channel is idle, the station waits for an Inter-frame slot (IFS) amount of time and then sends the frame.
- After sending the frame, it sets a timer.
- The station then waits for acknowledgement from the receiver. If it receives the acknowledgement before expiry of timer, it marks a successful transmission.
- Otherwise, it waits for a back-off time period and restarts the algorithm.


Start
K = 0
Idle channel?
No
Yes
Wait IFS time
Still idle?
No
Yes
Contention window size is $2^K - 1$.
Choose a random number R between 0 and $2^K - 1$
After each slot, if idle, continue; if busy, halt and continue when idle.
Wait R slots.
Send frame.
Wait time-out.
No
K > 15
K = K + 1
No
ACK received?
Yes
Yes
Abort
Success

**Advantages of CMSA/CA**

- CMSA/CA prevents collision.
- Due to acknowledgements, data is not lost unnecessarily.

- It avoids wasteful transmission.
- It is very much suited for wireless transmissions.

**Disadvantages of CSMA/CD**

- The algorithm calls for long waiting times.
- It has high power consumption.

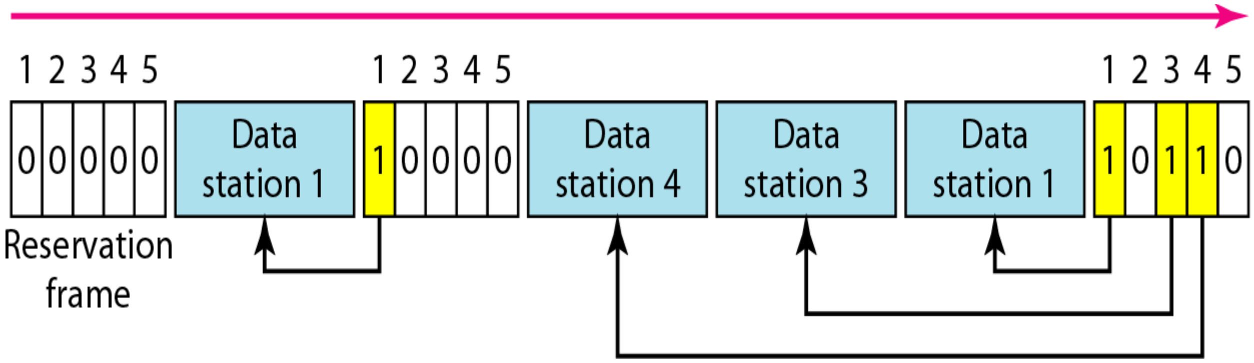
**MULTIPLE-ACCESS PROTOCOLS- CONTROLLED ACCESS**

- In controlled access, the stations consult one another to find which station has the right to send.
- A station cannot send unless it has been authorized by other stations.
- There are three popular controlled-access methods.

1. Reservation
2. Polling
3. Token Passing

**Reservation**

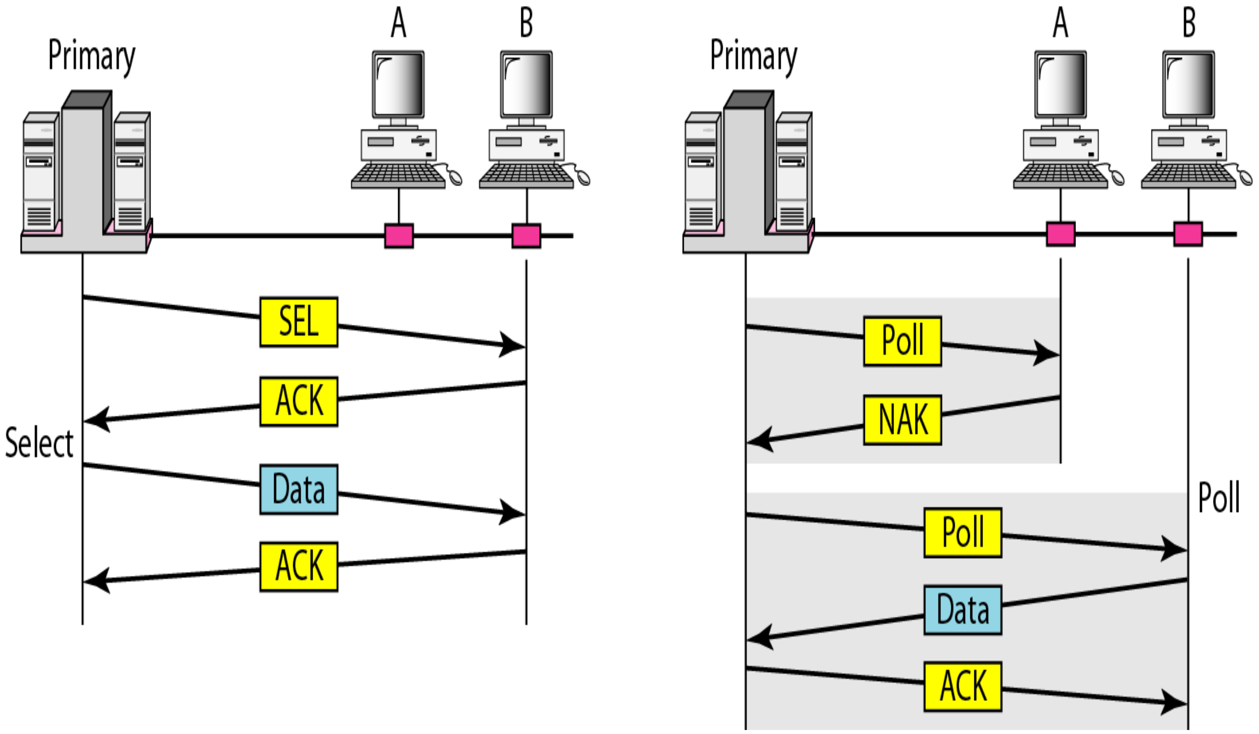
- In the reservation method, a station needs to make a reservation before sending data.
- Time is divided into intervals. In each interval, a reservation frame precedes the data
- frames sent in that interval.
- If there are N stations in the system, there are exactly N reservation minislots in the reservation frame. Each minislot belongs to a station.
- When a station needs to send a data frame, it makes a reservation in its own minislot. The stations that have made reservations can send their data frames after the reservation frame.



**Polling**

- Polling works with topologies in which one device is designated as a primary station and the other devices are secondary stations.

- All data exchanges must be made through the primary device even when the ultimate destination is a secondary device.
- The primary device controls the link; the secondary devices follow its instructions.
- It is up to the primary device to determine which device is allowed to use the channel at a given time.



**Token Passing**

- In the token-passing method, the stations in a network are organized in a logical ring.
- For each station, there is a predecessor and a successor.
- The predecessor is the station which is logically before the station in the ring.
- The successor is the station which is after the station in the ring. The current station is the one that is accessing the channel now.
- The right to this access has been passed from the predecessor to the current station.
- The right will be passed to the successor when the current station has no more data to send.

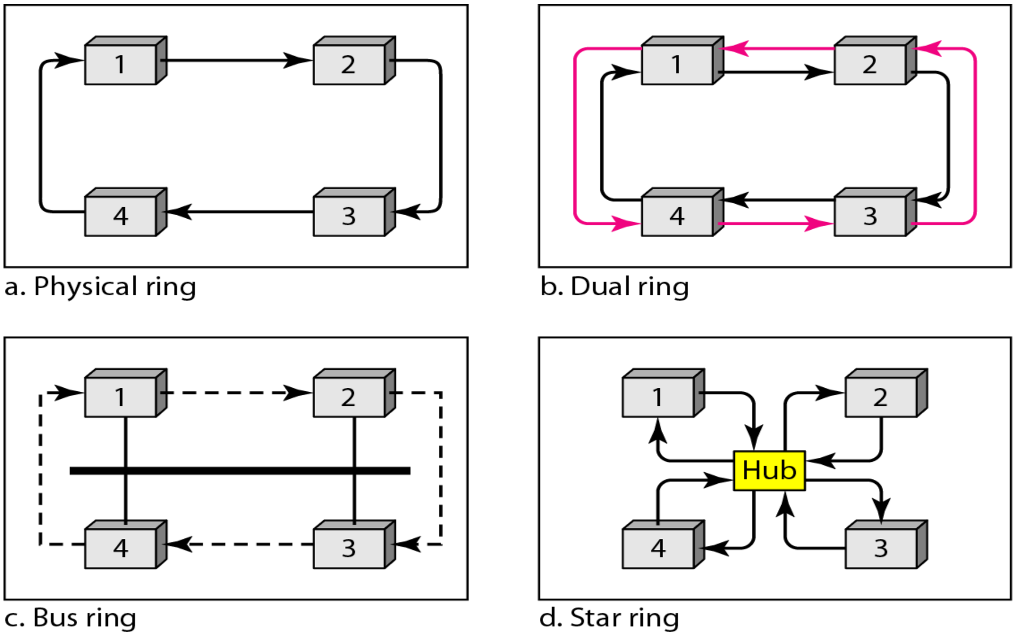
**Token Passing**

- When a station has some data to send, it waits until it receives the token from its predecessor.
- It then holds the token and sends its data. When the station has no more data to send, it releases the token, passing it to the next logical station in the ring.

- The station cannot send data until it receives the token again in the next round. In this process.
- when a station receives the token and has no data to send, it just passes the data to the next station.

**Token Passing :** Token passing accessing methods are

1. Physical ring
2. Bus ring
3. Dual ring
4. Star ring



**Token Passing**

- In the **physical ring topology**, when a station sends the token to its successor, the token cannot be seen by other stations; the successor is the next one in line.
- This means that the token does not have to have the address of the next successor. The problem with
- this topology is that if one of the links-the medium between two adjacent stations fails, the whole system fails.

**Token Passing**

- The **dual ring topology** uses a second (auxiliary) ring which operates in the reverse direction compared with the main ring.

- The second ring is for emergencies only (such as a spare tire for a car). If one of the links in the main ring fails, the system automatically combines the two rings to form a temporary ring.
- After the failed link is restored, the auxiliary ring becomes idle again.

**Token Passing**

- In the **bus ring topology**, also called a token bus, the stations are connected to a single cable called a bus.
- They, however, make a logical ring, because each station knows the address of its successor (and also predecessor for token management purposes).
- When a station has finished sending its data, it releases the token and inserts the address of its successor in the token.
- Only the station with the address matching the destination address of the token gets the token to access the shared media.
- The Token Bus LAN, standardized by IEEE, uses this topology.

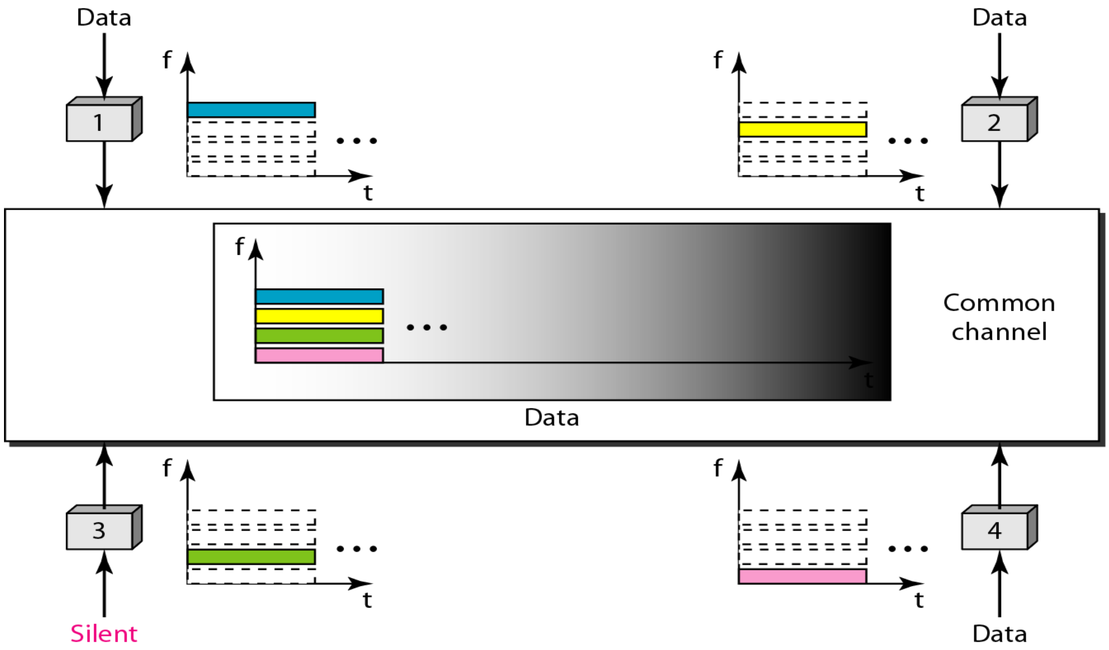
**Token Passing**

- In a **star ring topology**, the physical topology is a star. There is a hub, however, that acts as the connector.
- The wiring inside the hub makes the ring; the stations are connected to this ring through the two wire connections.
- This topology makes the network less prone to failure because if a link goes down, it will be bypassed by the hub and the rest of the stations can operate.
- Also adding and removing stations from the ring is easier.
- This topology is still used in the Token Ring LAN designed by IBM.
-

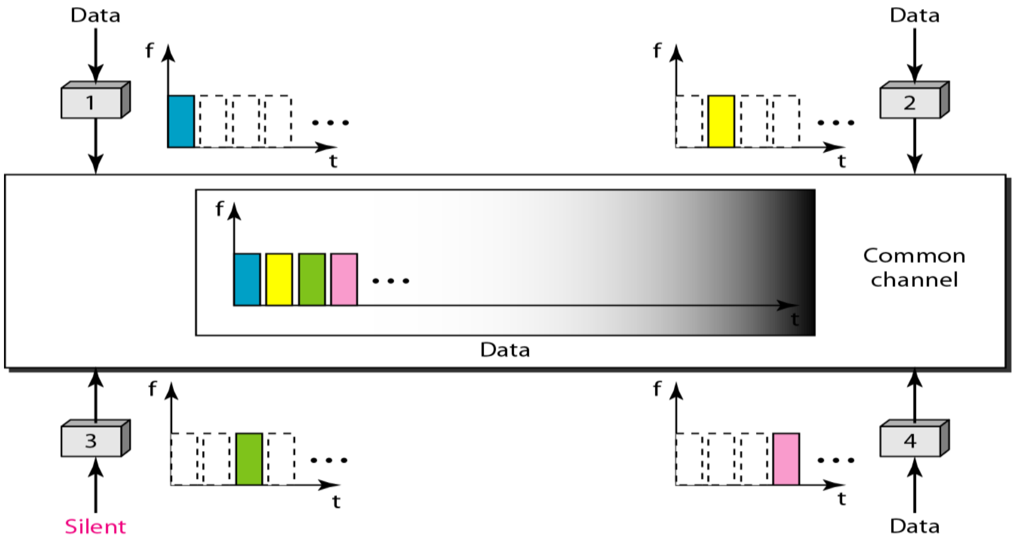
**MULTIPLE-ACCESS PROTOCOLS-CHANNELIZATION**

- Channelization is a multiple-access method in which the available bandwidth of a link is shared in time, frequency, or through code, between different stations.
- There are three channelization protocols: FDMA, TDMA, and CDMA
- **Frequency Division Multiple Access (FDMA) –** In frequency-division multiple access (FDMA), the available bandwidth is divided into frequency bands.
- Each station is allocated a band to send its data. In other words, each band is reserved for a specific station, and it belongs to the station all the time.

- Each station also uses a band-pass filter to confine the transmitter frequencies.
- To prevent station interferences, the allocated bands are separated from one another by small guard bands.

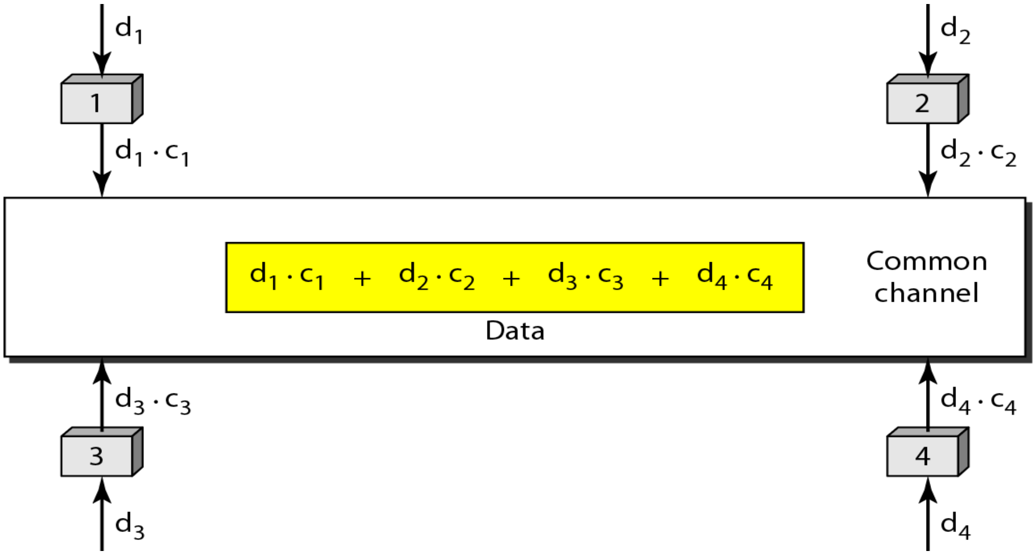


- **Time Division Multiple Access (TDMA) –** In this, the bandwidth is shared between multiple stations. To avoid collision time is divided into slots and stations are allotted these slots to transmit data.
- However there is a overhead of synchronization as each station needs to know its time slot. This is resolved by adding synchronization bits to each slot.
- Another issue with TDMA is propagation delay which is resolved by addition of guard bands.



- **Code Division Multiple Access (CDMA) –** One channel carries all transmissions simultaneously. There is neither division of bandwidth nor division of time.

- For example, if there are many people in a room all speaking at the same time, then also perfect reception of data is possible if only two person speak the same language.
- Similarly data from different stations can be transmitted simultaneously in different code languages.



## Wireless LAN ( IEEE 802.11)

### 802.11Architecture

The 802.11architecture defines two types of services and three different types of stations

### 802.11 Services

The two types of services are

1. Basic services set (BSS)

2. Extended Service Set (ESS)

**1. Basic Services Set (BSS)**

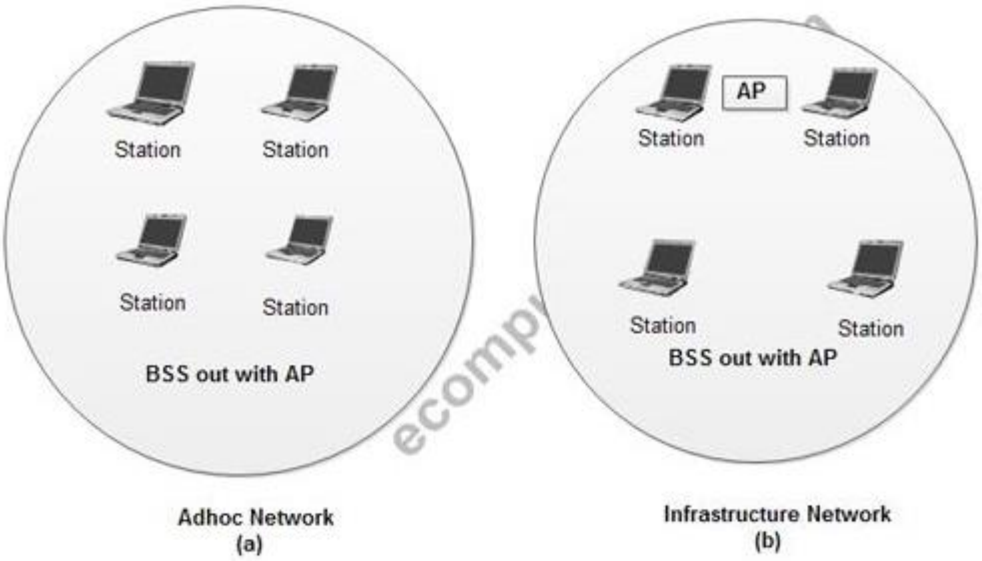
• The basic services set contain stationary or mobile wireless stations and a central base station called access point (AP).

• The use of access point is optional.

• If the access point is not present, it is known as stand-alone network. Such a

BSS cannot send data to other BSSs. This type of architecture is known as adhoc architecture.

• The BSS in which an access point is present is known as an infrastructure network.



**Basic Service Sets**

**2. Extend Service Set (ESS)**

• An extended service set is created by joining two or more basic service sets (BSS) having access points (APs).

• These extended networks are created by joining the access points of basic services sets through a wired LAN known as distribution system.

• The distribution system can be any IEET LAN.

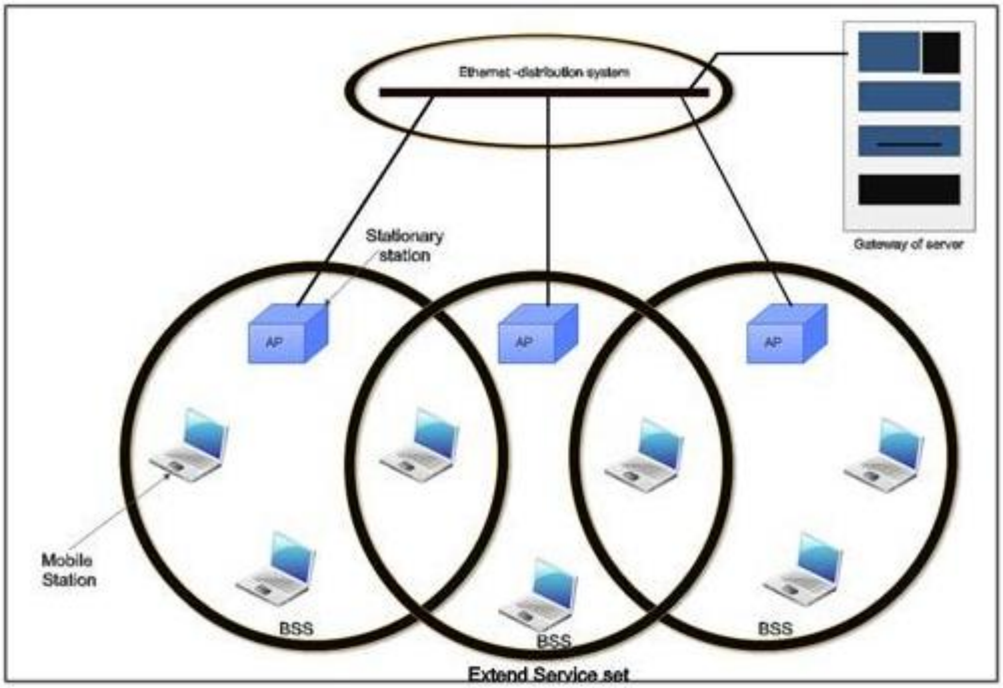
• There are two types of stations in ESS:

(i) **Mobile stations**: These are normal stations inside a BSS.

(ii) **Stationary stations**: These are AP stations that are part of a wired LAN.

• Communication between two stations in two different BSS usually occurs via two APs.

• A mobile station can belong to more than one BSS at the same time.



**802.11StationTypes**

IEEE 802.11 defines three types of stations on the basis of their mobility in wireless LAN. These are:

1. No-transition Mobility

2. BSS-transition Mobility

3. ESS-transition Mobility

1. **No-transition .Mobility**: These types of stations are either stationary *i.e.* immovable or move only inside a BSS.

2. **BSS-transition mobility**: These types of stations can move from one BSS to another but the movement is limited inside an ESS.

3. **ESS-transition mobility**: These types of stations can move from one ESS to another. The communication mayor may not be continuous when a station moves from one ESS to another ESS.

**MAC sublayer Functions**

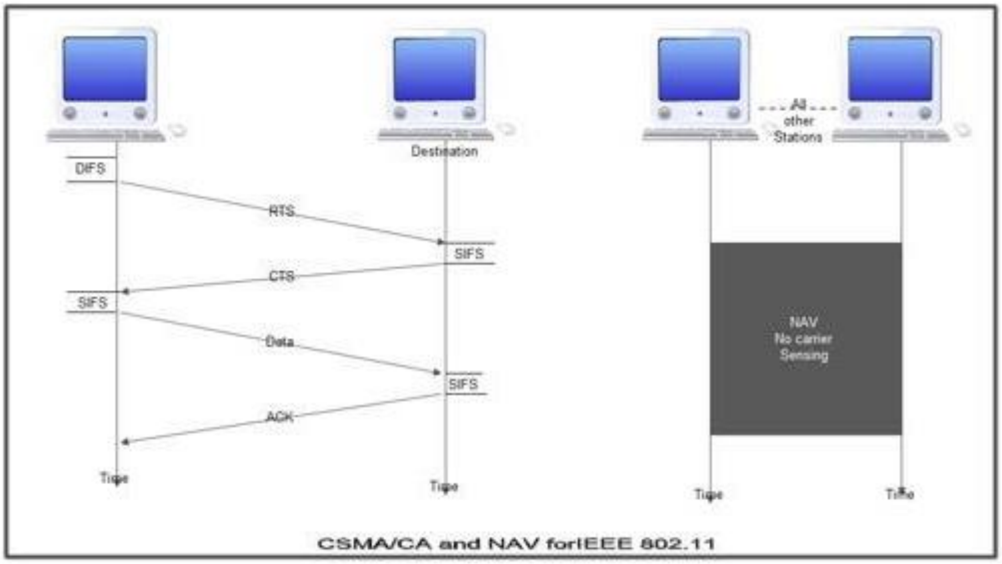
802.11 support two different modes of operations. These are:

1. Distributed Coordination Function (DCF)

2. Point Coordination Function (PCF)

**1. Distributed Coordination Function**

- The DCF is used in BSS having no access point.
- DCF uses CSMA/CA protocol for transmission.
- The following steps are followed in this method.



CSMA/CA and NAV forIEEE 802.11

1. When a station wants to transmit, it senses the channel to see whether it is free or not.

2. If the channel is not free the station waits for back off time.

3. If the station finds a channel to be idle, the station waits for a period of time called distributed interframe space (DIFS).

4. The station then sends control frame called request to send (RTS) as shown in figure.

5. The destination station receives the frame and waits for a short period of time called short interframe space (SIFS).

6. The destination station then sends a control frame called clear to send (CTS) to the source station. This frame indicates that the destination station is ready to receive data.

7. The sender then waits for SIFS time and sends data.

8. The destination waits for SIFS time and sends acknowledgement for the received frame.

**Collision avoidance**

- 802.11 standard uses Network Allocation Vector (NAV) for collision avoidance.
- The procedure used in NAV is explained below:

1. Whenever a station sends an RTS frame, it includes the duration of time for which the station will occupy the channel.

2. All other stations that are affected by the transmission creates a timer caned network allocation vector (NAV).

3. This NAV (created by other stations) specifies for how much time these stations must not check the channel.

4. Each station before sensing the channel, check its NAV to see if has expired or not.

5. If its NA V has expired, the station can send data, otherwise it has to wait.

• There can also be a collision during handshaking *i.e.* when RTS or CTS control frames are exchanged between the sender and receiver. In this case following procedure is used for collision avoidance:

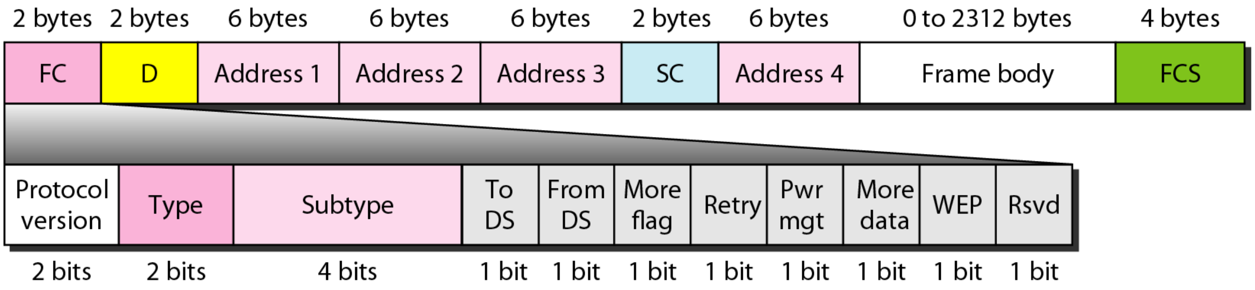
1. When two or more stations send RTS to a station at same time, their control frames collide.

2. If CTS frame is not received by the sender, it assumes that there has been a collision.

3. In such a case sender, waits for back off time and retransmits RTS.

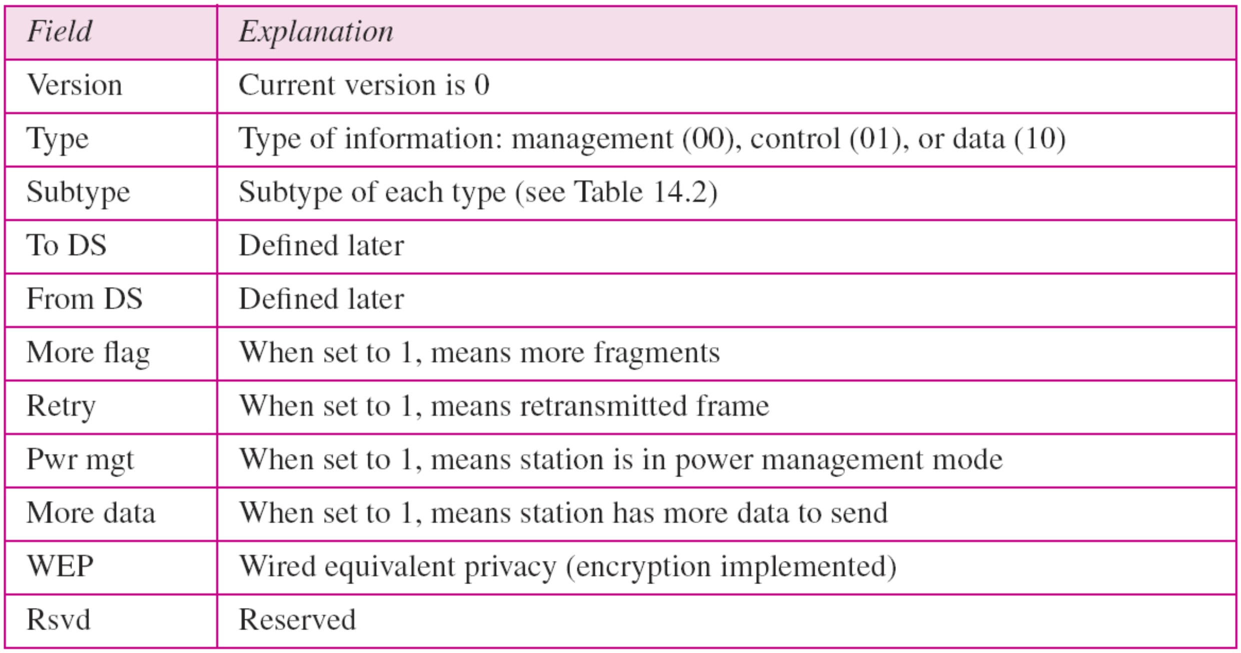
## Frame Format of 802.11

The MAC layer frame consists of nine fields.



- The main fields of a frame of wireless LANs as laid down by IEEE 802.11 are −
- **Frame Control** − It is a 2 bytes starting field composed of 11 subfields. It contains control information of the frame.
- **Duration** − It is a 2-byte field that specifies the time period for which the frame and its acknowledgment occupy the channel.
- **Address fields** − There are three 6-byte address fields containing addresses of source, immediate destination, and final endpoint respectively.
- **Sequence** − It a 2 bytes field that stores the frame numbers.
- **Data** − This is a variable-sized field that carries the data from the upper layers. The maximum size of the data field is 2312 bytes.
- **Check Sequence** − It is a 4-byte field containing error detection information.

Sub Fields:



| *Field* | *Explanation* |
|---|---|
| Version | Current version is 0 |
| Type | Type of information: management (00), control (01), or data (10) |
| Subtype | Subtype of each type (see Table 14.2) |
| To DS | Defined later |
| From DS | Defined later |
| More flag | When set to 1, means more fragments |
| Retry | When set to 1, means retransmitted frame |
| Pwr mgt | When set to 1, means station is in power management mode |
| More data | When set to 1, means station has more data to send |
| WEP | Wired equivalent privacy (encryption implemented) |
| Rsvd | Reserved |

**IEEE 802.11 Frame types**

There are three different types of frames:

1. Management frame

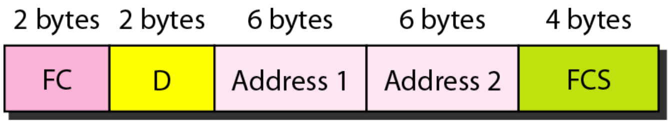
2. Control frame

3. Data frame

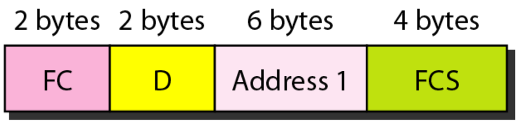
1. **Management frame**. These are used for initial communication between stations and access points.

2. **Control frame**. These are used for accessing the channel and acknowledging frames. The control frames are RTS and CTS.

3. **Data frame**. These are used for carrying data and control information.



RTS



CTS or ACK

### 802.11 Addressing

• There are four different addressing cases depending upon the value of *To DS And from* DS subfields of FC field.

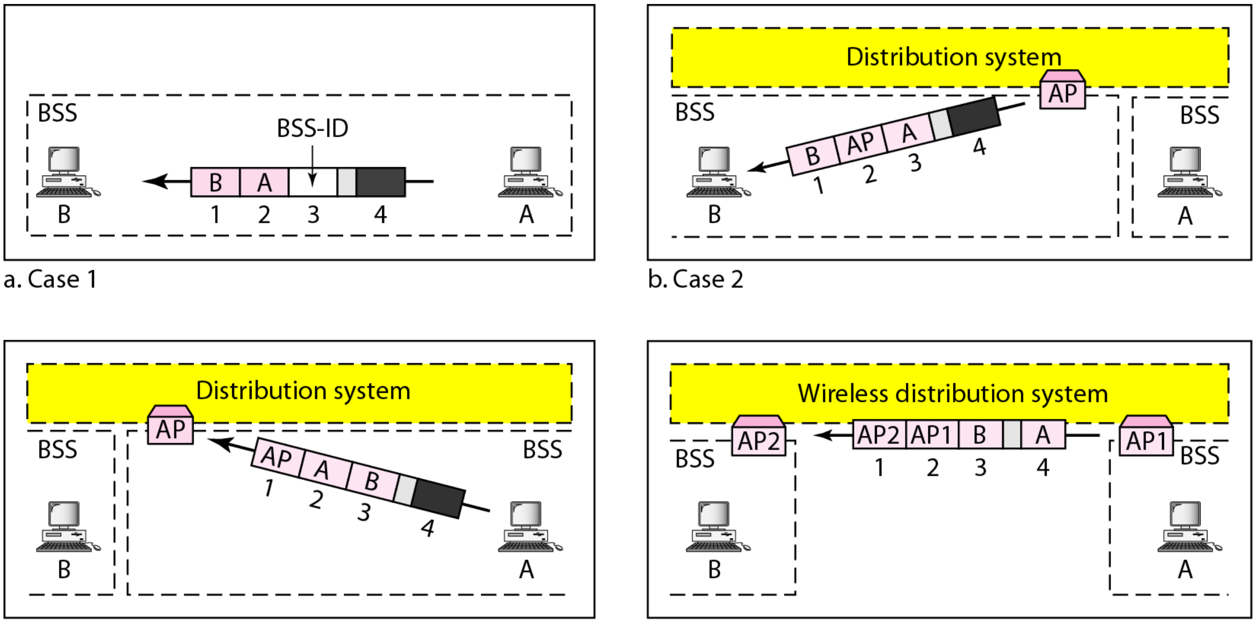
• Each flag can be 0 or 1, resulting in 4 different situations.

1. If *To* DS = 0 and *From* DS = 0, it indicates that frame is not going to distribution system and is not coming from a distribution system. The frame is going from one station in a BSS to another.

2. If *To* DS = 0 and *From* DS = 1, it indicates that the frame is coming from a distribution system. The frame is coming from an AP and is going to a station. The address 3 contains original sender of the frame (in another BSS).

3. If *To* DS = 1 and *From* DS = 0, it indicates that the frame is going to a distribution system. The frame is going from a station to an AP. The address 3 field contains the final destination of the frame.

4. If *To* DS = 1 and *From* DS = 1,it indicates that frame is going from one AP to another AP in a wireless distributed system.



a. Case 1

b. Case 2

c. Case 3

d. Case 4

| *To DS* | *From DS* | *Address 1* | *Address 2* | *Address 3* | *Address 4* |
|---|---|---|---|---|---|
| 0 | 0 | Destination | Source | BSS ID | N/A |
| 0 | 1 | Destination | Sending AP | Source | N/A |
| 1 | 0 | Receiving AP | Source | Destination | N/A |
| 1 | 1 | Receiving AP | Sending AP | Destination | Source |