UNIT 2 - DATA LINK LAYER (DLL)

Learning Objectives:

Upon completion of this unit, the student should be able to:

- Describe briefly the services provided by the data link layer.
- Understand the framing and the reason for its need.
- Understand the different approaches of framing.
- Explains the different data link protocols.
- Describe the High-level Data Link Control (HDLC)

2.1 Introduction : The two main sections of the Data Link Layer are

1. Data Link Control (DLC): It deals with the design and procedures for communication between nodes: node-to-node communication.
2. Media Access Control (MAC): It explains how to share the link.

1. **Data Link Control (DLC*):*** Data link control functions includes
    1. Framing.
    2. Flow Control.
    3. Error Control.

    Above functions are software implemented protocols that provide smooth and reliable transmission of frames between nodes

1. **Framing:** Divide the bit stream in to group of bits and attach checksum called Framing. The data link layer needs to pack bits into frames, so that each frame is distinguishable from another. Framing in the DLL separates a message from one source to a destination, or from other messages to other destinations, by adding a sender address and destination address. The destination address defines where the packet is to go; the sender address helps the recipient acknowledge the receipt. The framing concept as shown in figure 2.1

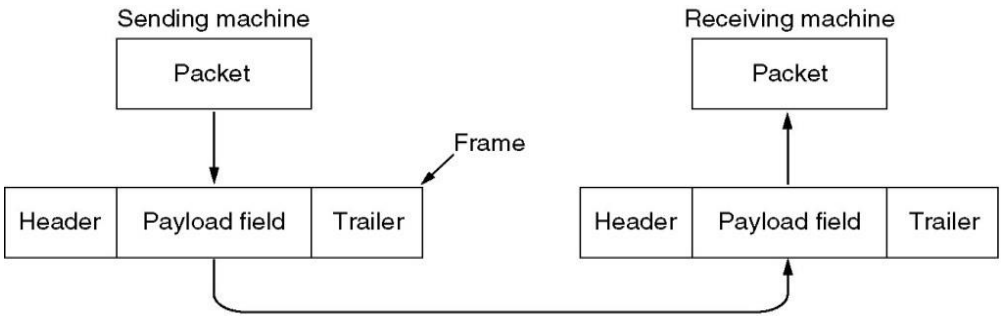


Figure 2.1 The framing concept

Frames of Two Categories:

1. ***Fixed-Size Framing*** :
   - Size of the frame is fixed.
   - No need for defining the boundaries of the frames; the size itself can be used as a delimiter.
   - It is used in the ATM wide-area network, which uses frames of fixed size called cells.
2. ***Variable Size Framing*** :
   - Size of the frame is not fixed.
   - Need to define the start and end of the frame and the beginning of the next.
   - It is used in local area networks.
   - Two approaches were used for this purpose:

     (a). Character-oriented approach

     (b). Bit-oriented approach.

(a).***Character-oriented framing approach****:* Here data to be carried are 8-bit characters from a coding system such as ASCII. The header carries the source and destination addresses and other control information. Trailer carries error detection or error correction redundant bits, are also multiples of 8 bits. To separate one frame from the next, an 8-bit (1-byte) flag is added at the beginning and the end of a frame. The flag, composed of protocol-dependent special characters, signals the start or end of a frame, which shown in figure 2.2

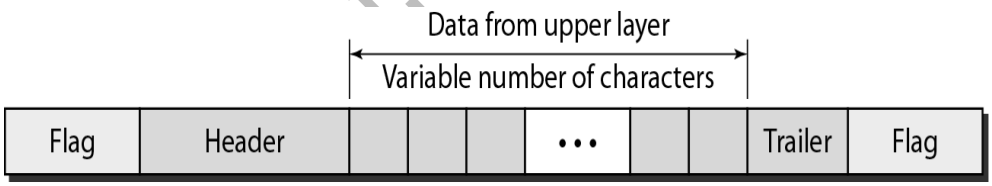


Figure 2.2: Shows the format of a frame in a character-oriented protocol

Advantage:

1 .Simple framing method.

2. Character-oriented framing was popular when only text was exchanged by the data link layers.

3. The flag could be selected to be any character not used for text communication.

Disadvantage:

1. What if the count is garbled?
2. Even if with checksum, the receiver knows that the frame is bad there is no way to tell where the next frame starts.
3. Asking for retransmission doesn't help either because the start of the retransmitted frame is not known. Hence no longer used.

a. Starting and ending character with byte stuffing*:* Byte stuffing is the process of adding 1 extra byte whenever there is a flag or escape character in the text. This is shown in the figure 2.4

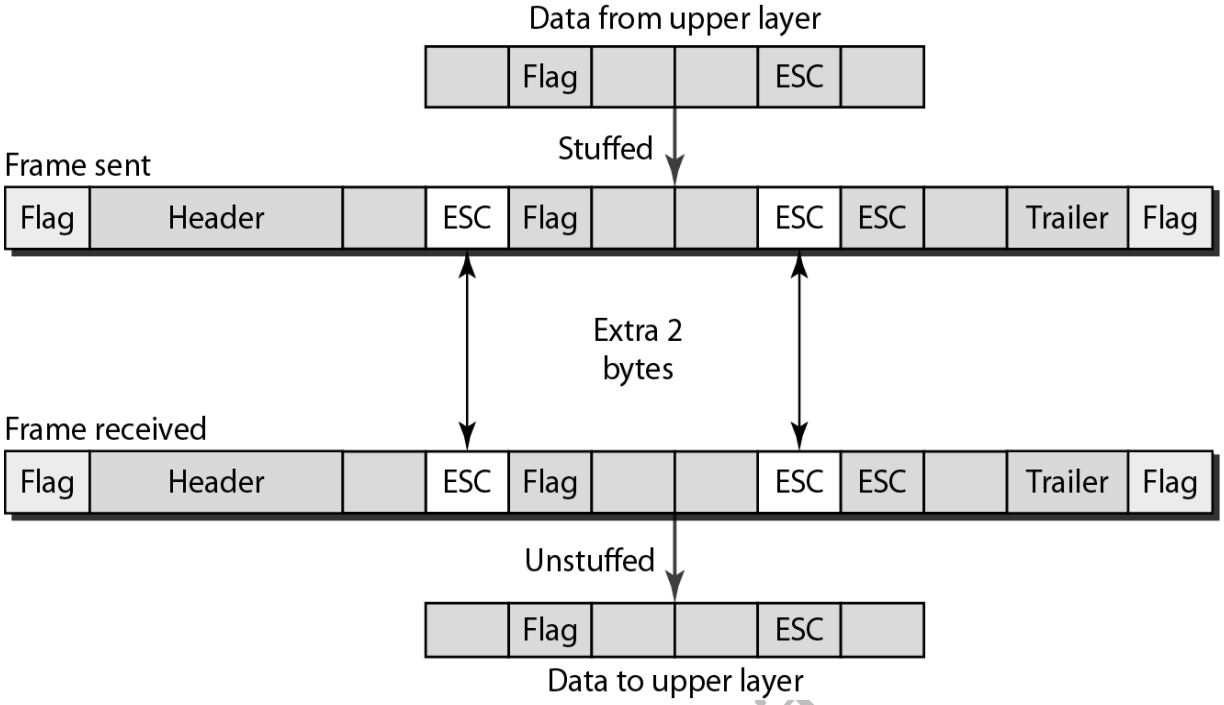


Figure 2.4 Byte stuffing and unstuffing

Limitation of byte stuffing: Fixed character size of 8 bits can't handle heterogeneous environment.

***b*. Bit-Oriented framing approach:** It is the process of adding one extra 0 whenever five consecutive 1's follow a 0 in the data, so that the receiver does not mistake the pattern 0111110 for a flag. Most protocols use a special 8-bit pattern flag 01111110 as the delimiter to define the beginning and the end of the frame, as shown in Figure 2.6. This flag can create the same type of problem. If the flag pattern appears in the data; need to somehow inform the receiver that this is not the end of the frame. We do this by stuffing 1 single bit (instead of 1 byte) to prevent the pattern from looking like a flag. The strategy is called *bit stuffing*.

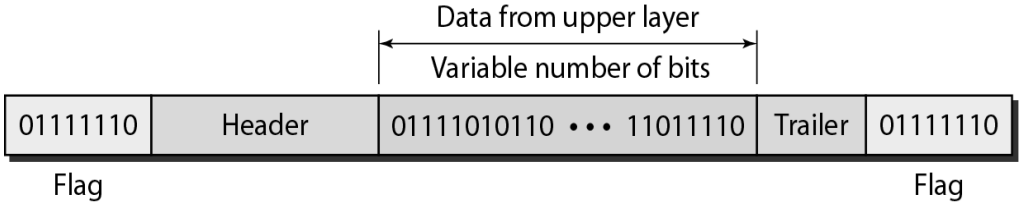


Figure 2.6 A frame in a bit-oriented protocol

Bit stuffing is the process of adding one extra 0 whenever five consecutive 1s follow a 0 in the data, so that the receiver does not mistake the pattern 0111110 for a flag.

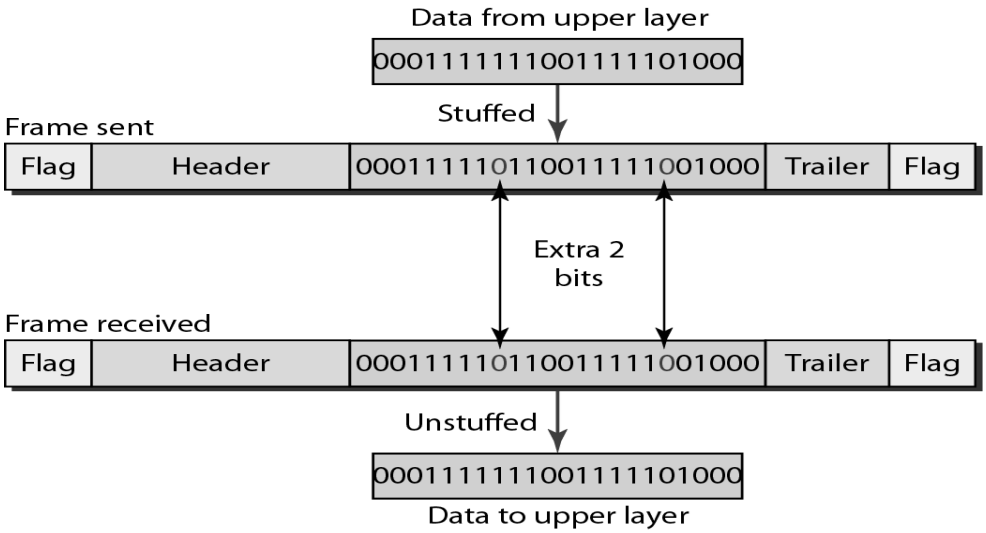


Figure 2.7 Bit stuffing and unstuffing

2. Flow Control: It is a technique for speed matching of transmitter and receiver. It ensures that a transmitting station does not overflow a receiving station with data. It refers to a set of procedures used to restrict the amount of data that the sender can send before waiting for acknowledgment. Two types of flow control: *Feedback based & Rate based flow control*

3. Error Control: It performs both error detection and error correction. It allows the receiver to inform the sender of any frames lost or damaged in transmission and coordinates the retransmission of those frames by the sender. Error control in the data link layer is based on automatic repeat request (ARQ) which is the retransmission of data.

2.2 DATA LINK PROTOCOLS ***:

These protocols in data link layer can combine framing, flow control, and error control to achieve the delivery of data from one node to another. The protocols are normally implemented in software by using one of the common programming languages. Figure 2.9 shows Taxonomy of protocols are used in data link layer

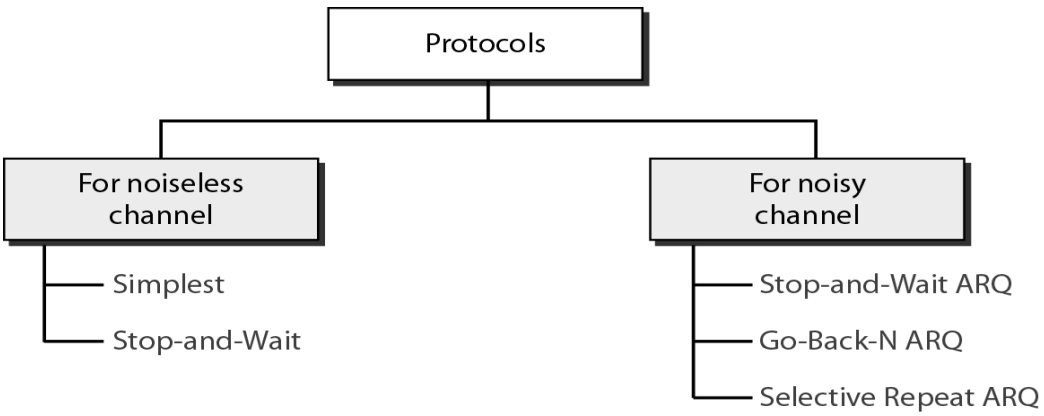


Figure 2.9 Taxonomy of protocols are used in data link layer

1. **Data Link Protocols for Noiseless Channels (Flow Control Protocols):** Two simple protocols are discussed here:
    A. Simplest protocol
    B. Stop and wait protocol

A. Simplest Protocol (unrestricted ideal protocol):

The following assumption has been made for developing the (algorithm) simplex protocol

- The channel is a perfect noiseless channel. Hence no frames are lost, duplicated, or corrupted.
- No flow control and error control used.
- It is a unidirectional protocol in which data frames are traveling in only one direction.
- Both transmitting and receiving network layer are always ready.
- Processing time that is small enough to be negligible.
- Infinite buffer space is available in the transmitter and receiver.

(A.1) Design: Figure 2.10 shows the design of the simplest protocol with no flow or error control. The data link layer (DLL) at the sender site gets data from its network layer, makes a frame out of the data, and sends it. The DLL at the receiver site receives a frame from its physical layer, extracts data from the frame, and delivers the data to its network layer. The DLL layers of the sender and receiver provide transmission services for their network layers. The DLL link layers use the services provided by their physical layers (such as signaling, multiplexing, and so on) for the physical transmission of bits. *Drawback*: It is an ideal and unrealistic protocol

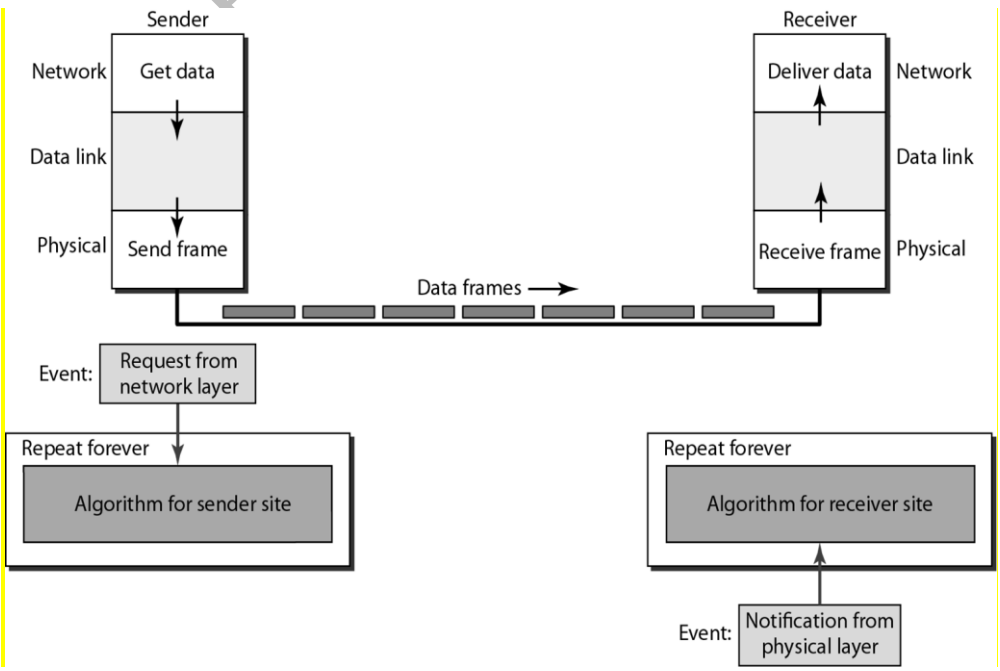


Figure 2.10: The design of the simplest protocol with no flow or error control

(A.2). Algorithms for Simplest Protocol*:* Algorithm 2.1 and 2.2 shows the procedure at the sender and receiver sites

```
1  while(true)                          // Repeat forever
2  {
3    WaitForEvent();                    // Sleep until an event occurs
4    if(Event(RequestToSend))           //There is a packet to send
5    {
6       GetData();
7       MakeFrame();
8       SendFrame();                    //Send the frame
9    }
10 }
```

Algorithm 2.1 Sender-site algorithm for the simplest protocol

```
1  while(true)                          // Repeat forever
2  {
3    WaitForEvent();                    // Sleep until an event occurs
4    if(Event(ArrivalNotification))     //Data frame arrived
5    {
6       ReceiveFrame();
7       ExtractData();
8       DeliverData();                  //Deliver data to network layer
9    }
10 }
```

Algorithm 2. 2 Receiver-site algorithm for the simplest protocol

(A.3). Example for simplex protocol: Figure 2.11 shows an example of communication using this protocol. It is very simple. The sender sends a sequence of frames without even thinking about the receiver. To send three frames, three events occur at the sender site and three events at the receiver site.

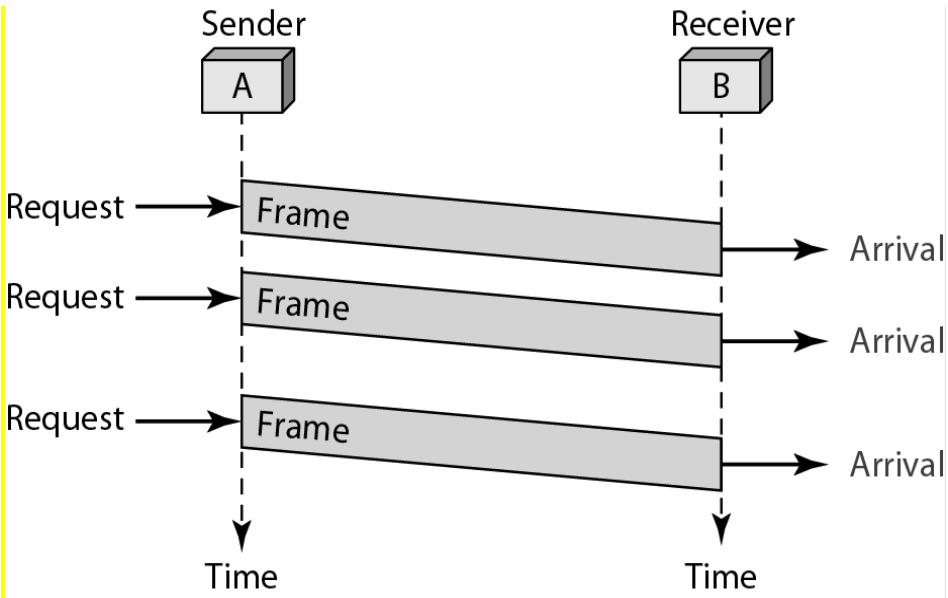
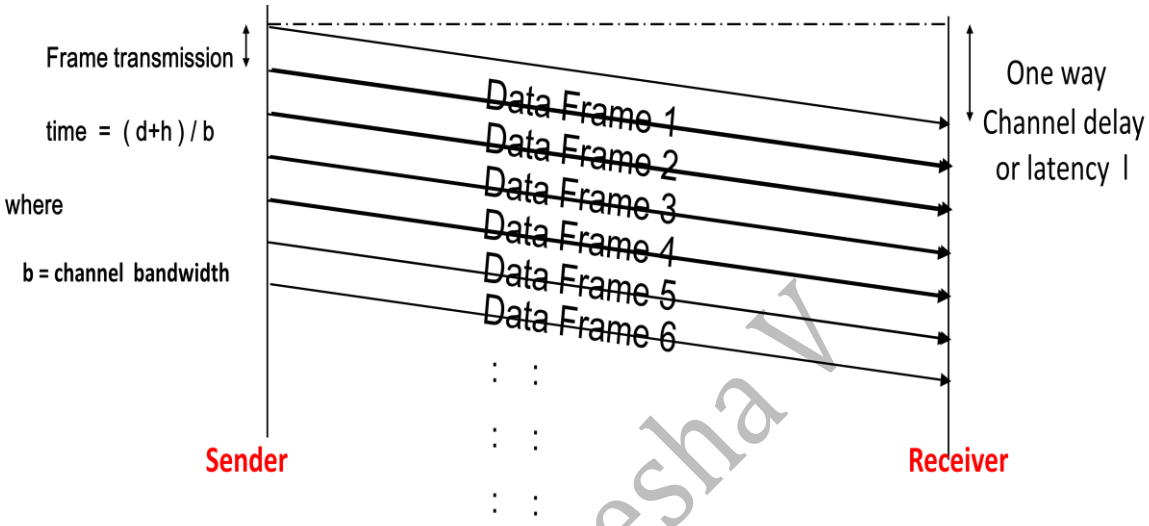


Figure 2.11: Flow diagram for above Example

(A.4).Efficiency analysis*:* If frame has *d* data bits and *h* overhead bits, channel bandwidth *b* bits/second:

- Maximum Channel Utilization = data size / frame size = $d / (d + h)$
- Maximum Data Throughput = $d / (d + h)$ * channel bandwidth

  = $d / (d + h)$ * b



B. Stop-and-Wait Protocol (simplex): The following assumption has been made for developing the (algorithm) Stop-and-Wait Protocol

- The channel is a perfect noiseless channel.
- Flow control used
- It is a bidirectional protocol in which frames are traveling in both direction
- Both transmitting and receiving network layer are always not ready.
- Processing time considerable
- Finite buffer space is available

(B.1). Design*:* Figure 2.12 illustrates the mechanism. Here traffic on the forward channel (from sender to receiver) and the reverse channel. At any time, there is either one data frame on the forward channel or one ACK frame on the reverse channel. It is a half-duplex link.

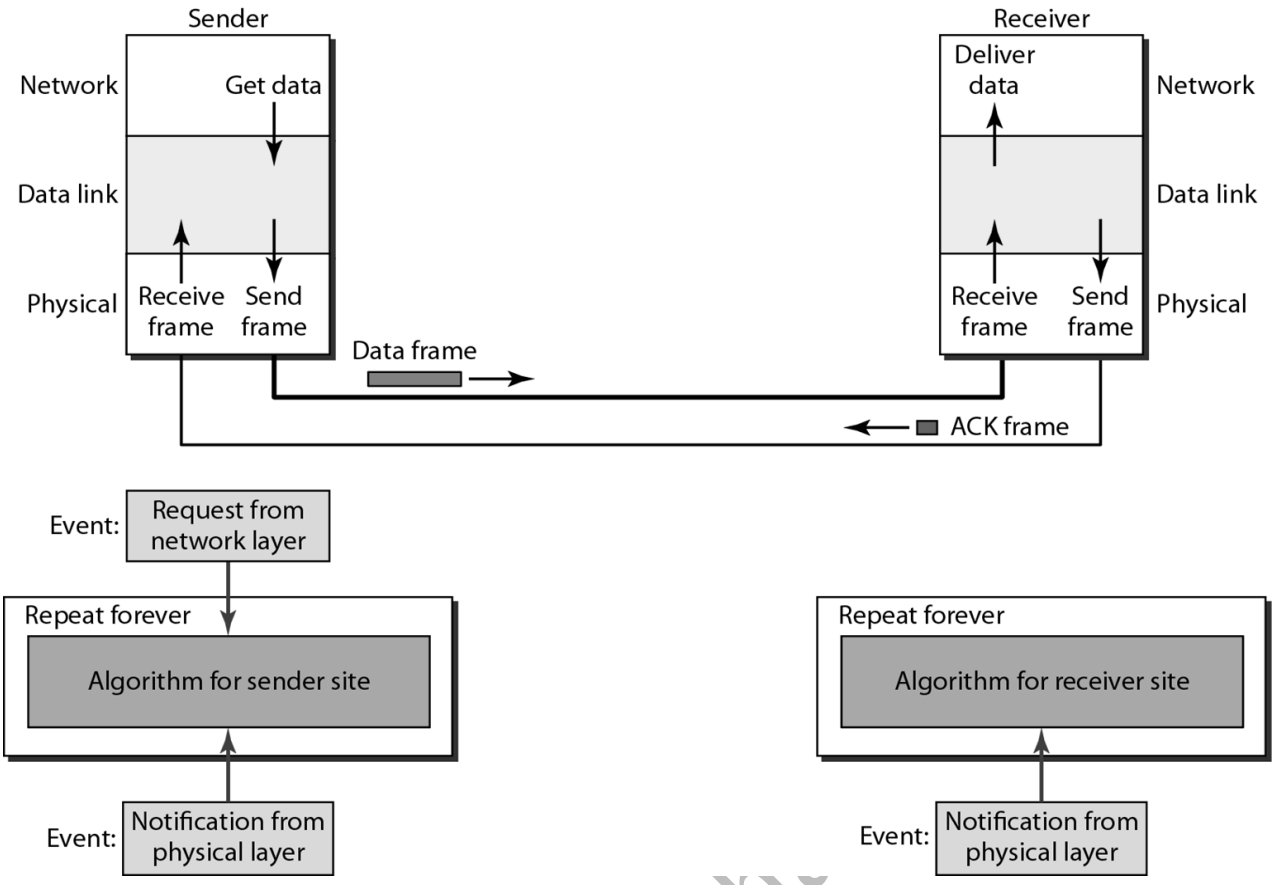


Figure 2.12 Design of Stop-and-Wait Protocol

(B.2). Algorithms*:* Algorithm 2.3 and 2.4 shows the procedure at the sender and receiver sites

```
1  while(true)                          //Repeat forever
2  canSend = true                       //Allow the first frame to go
3  {
4    WaitForEvent();                    // Sleep until an event occurs
5    if(Event(RequestToSend) AND canSend)
6    {
7       GetData();
8       MakeFrame();
9       SendFrame();                    //Send the data frame
10      canSend = false;                //Cannot send until ACK arrives
11   }
12   WaitForEvent();                    // Sleep until an event occurs
13   if(Event(ArrivalNotification) // An ACK has arrived
14    {
15      ReceiveFrame();                 //Receive the ACK frame
16      canSend = true;
17    }
18 }
```

Algorithm 2.3. Sender-site algorithm for the Stop-and-Wait Protocol

```
1  while(true)                          //Repeat forever
2  {
3    WaitForEvent();                    // Sleep until an event occurs
4    if(Event(ArrivalNotification))     //Data frame arrives
5    {
6       ReceiveFrame();
7       ExtractData();
8       Deliver(data);                  //Deliver data to network layer
9       SendFrame();                    //Send an ACK frame
10   }
11 }
```

Algorithm 2.4 Receiver-site algorithm for the *Stop-and-Wait Protocol*

(B.3). Example for Stop-and-Wait protocol

- The sender sends one frame and waits for feedback from the receiver.
- When the ACK arrives, the sender sends the next frame.
- Note that sending two frames in the protocol involves the sender in four events and the receiver in two events, This is shown in figure 2.13

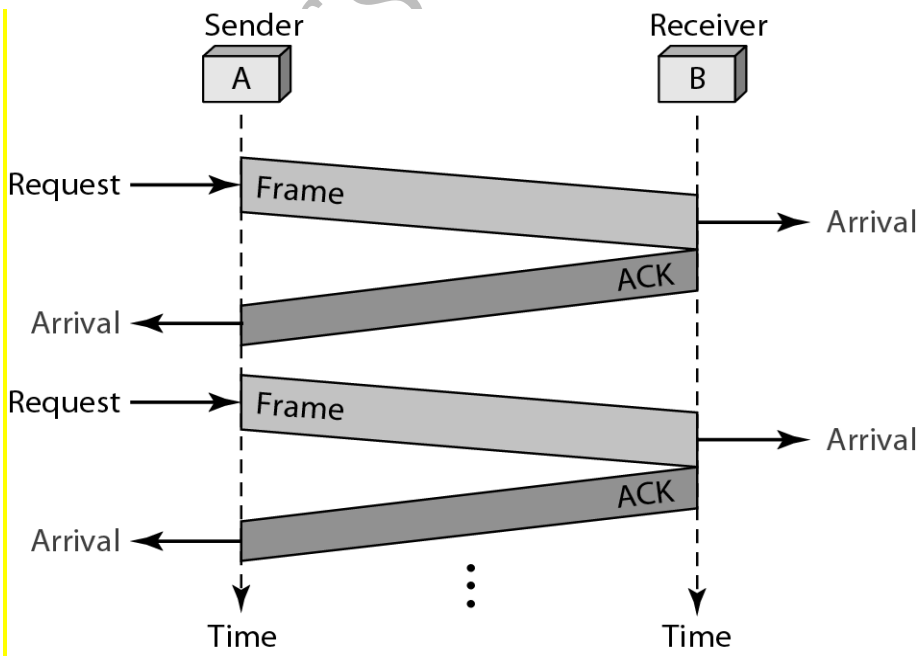


Figure 2.13: Flow diagram for above Example

2. Data Link Protocols Noisy Channels (Flow & Error Control Protocols)

- Stop-and-Wait Protocol gives us an idea of how to add flow control.
- Noiseless channels are nonexistent, hence error occur.
- We discuss three protocols in this section that use error control.
    1. Stop-and-Wait Automatic Repeat Request(ARQ) ***
    2. Go-Back-N Automatic Repeat ReQuest (ARQ) ***
    3. Selective Repeat Automatic Repeat Request (ARQ) ***

1. Stop-and-Wait Automatic Repeat Request (ARQ):

Purpose: To ensure a sequence of information packets is delivered in order and without errors or duplications despite transmission errors & losses. For noisy link, pure stop and wait protocol will break down, and solution is to incorporate some error control mechanism. Stop and wait with ARQ: Automatic Repeat request (ARQ), an error control method, is incorporated with stop and wait flow control protocol.

- If error is detected by receiver, it discards the frame and sends a negative ACK (NAK), causing sender to re-send the frame.
- In case a frame never got to receiver, sender has a timer: each time a frame is sent, timer is set. If no ACK or NAK is received during timeout period, it re-sends the frame
- Timer introduces a problem: Suppose timeout and sender retransmits a frame but receiver actually received the previous transmission! Receiver has duplicated copies.
- To avoid receiving and accepting two copies of same frame, frames and ACKs are alternatively labeled 0 or 1: ACK0 for frame 1, ACK1 for frame 0

Note:

1. Link parameter is defined by

    a = propagation time / frame time = R d / V L

    Where R = data rate (bps), d = link distance (m), V = propagation velocity (m/s)

    L = frame Length (bits) .
2. Error correction in Stop-and-Wait ARQ is done by keeping a copy of the sent frame and retransmitting of the frame when the timer expires
3. In Stop-and-Wait ARQ, use sequence numbers to number the frames. The sequence numbers are based on modulo-2 arithmetic.
4. In Stop-and-Wait ARQ, the acknowledgment number always announces in modulo-2 arithmetic the sequence number of the next frame expected.

(1.1) Design of the Stop-and-Wait ARQ Protocol*:* Figure 2.15 shows the design of the Stop-and-Wait ARQ Protocol.

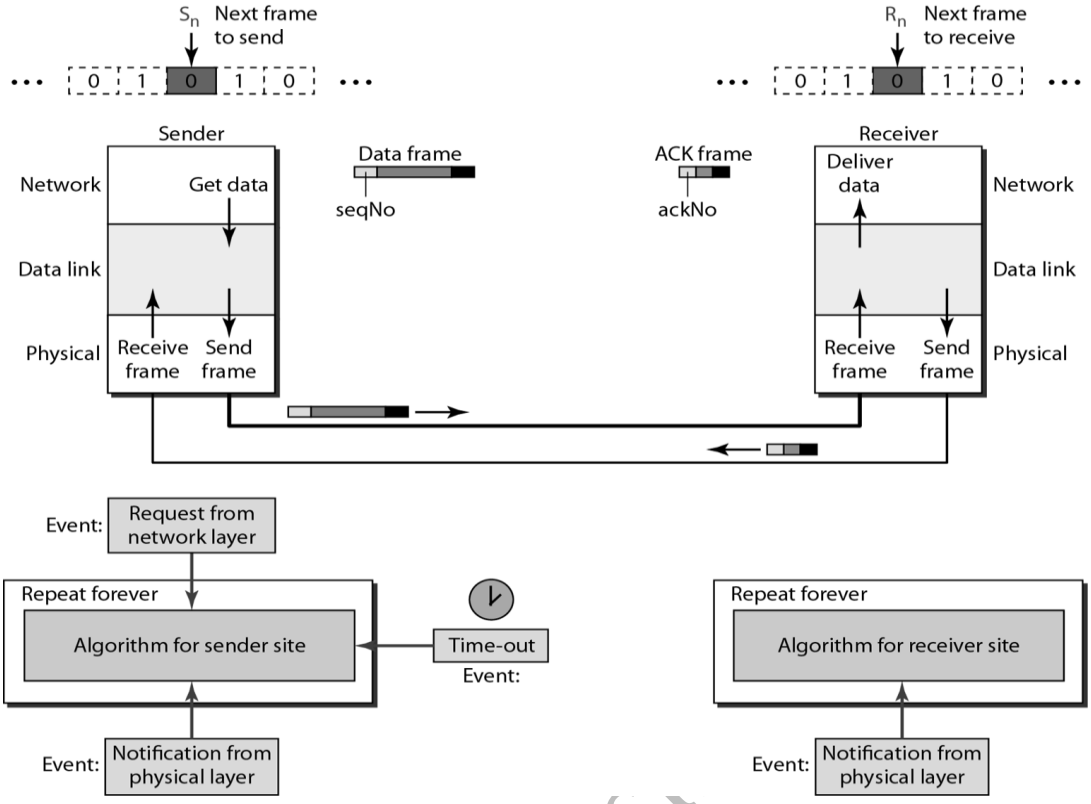


Figure 2.15 Design of the Stop-and-Wait ARQ Protocol

The sending device keeps a copy of the last frame transmitted until it receives an acknowledgment for that frame. A data frames uses a seqNo (sequence number); an ACK frame uses an AckNo (acknowledgment number). The sender has a control variable, which we call S_n (sender, next frame to send), that holds the sequence number for the next frame to be sent (0 or 1).The receiver has a control variable, which we call R_n (receiver, next frame expected), that holds the number of the next frame expected. When a frame is sent, the value of S_n is incremented (modulo-2), which means if it is 0, it becomes 1 and vice versa. When a frame is received, the value of R_n is incremented, which means if it is 0, it becomes 1 and vice versa.

(1.2). Example 1*:* Figure 2.17 shows an example of Stop-and-Wait ARQ.

Event:

1. Frame 0 is sent and acknowledged.
2. Frame 1 is lost and resent after the time-out.
3. The resent frame 1 is acknowledged and the timer stops.
4. Frame 0 is sent and acknowledged, but the acknowledgment is lost.
5. The sender has no idea if the frame or the acknowledgment is lost. so after the time-out, it resends frame0, which is acknowledged.

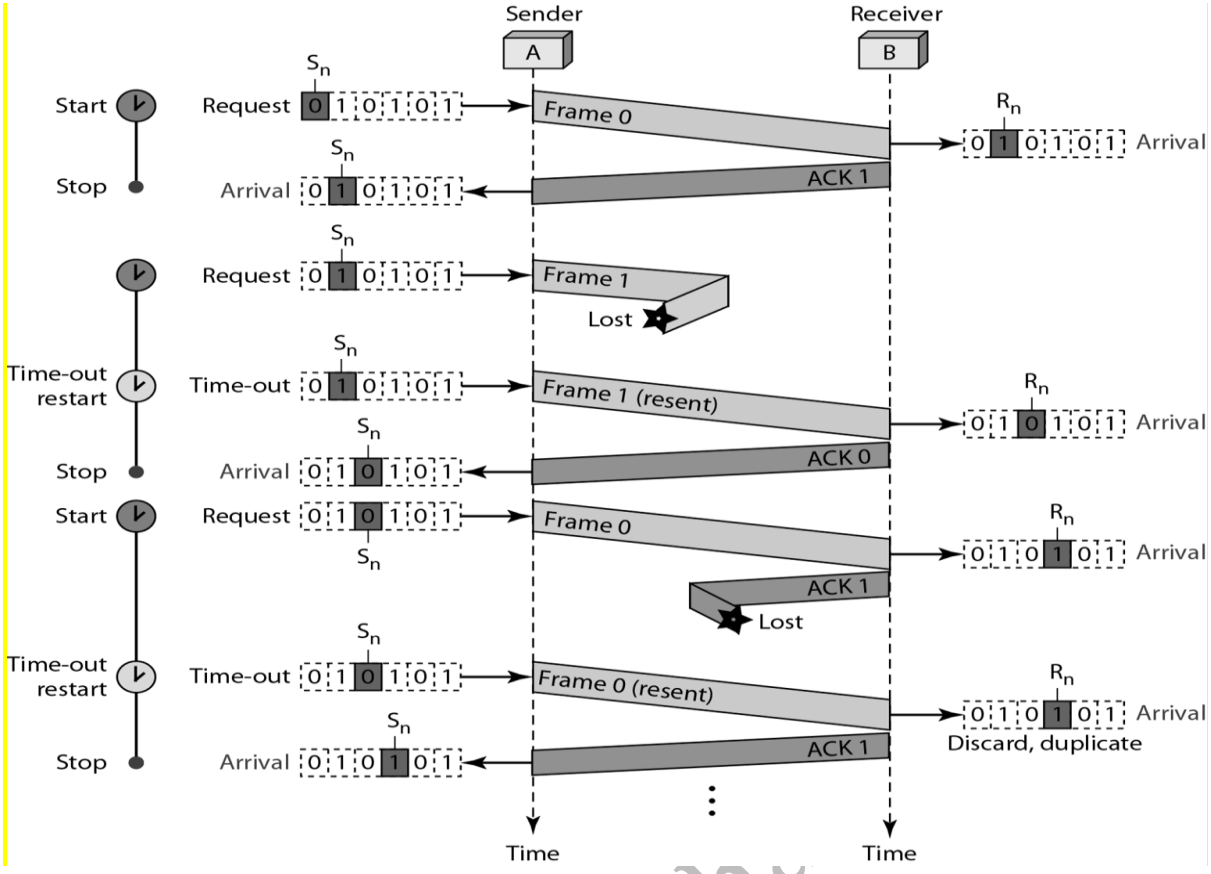


Figure 2.17 Flow diagrams for Example 1

(1.3). Efficiency of Stop-and-Wait ARQ :

Example 1*:* Assume that, in a Stop-and-Wait ARQ system, the bandwidth of the line is 1 Mbps and 1 bit takes 20 ms to make a round trip. What is the bandwidth-delay product? If the system data frames are 1000 bits in length, what is the utilization percentage of the link?

Solution:

The bandwidth-delay product (capacity) is = B.W * Delay per bit

$= (1*10^{6})*(20*10^{-3)} = 20{,}000$ bits

Analysis:

The system can send 20,000 bits during the time it takes for the data to go from the sender to the receiver & then back again. However, the system sends only 1000 bits.

Then link utilization = data size/ bandwidth-delay product (channel capacity)

= 1000 /20,000, or 5 percent.

Note: For this reason, for a link with a high bandwidth or long delay, the use of Stop-and-Wait ARQ wastes the capacity of the link.

***Example 2*:** What is the utilization percentage of the link in Example 1 if we have a protocol that can send up to 15 frames before stopping and worrying about the acknowledgments?

Solution

- The bandwidth-delay product is still 20,000 bits.
- The system can send up to 15 frames or 15,000(1000 bits/frame) bits during a round trip.
- Percentage of utilization is 15,000/20,000, or 75 percent.
- If there are damaged frames, the utilization percentage is much less because frames have to be resent.
- *Remedy for increasing the efficiency is Pipelining*
- Pipelining: several frames can be sent before we receive news about the previous frames.
- Pipelining improves the efficiency of the transmission if the number of bits in transition is large with respect to the bandwidth-delay product

Advantages of stop and wait ARQ

1. It can be used for noisy channels
2. It has both error and flow control mechanism
3. It has timer implementation

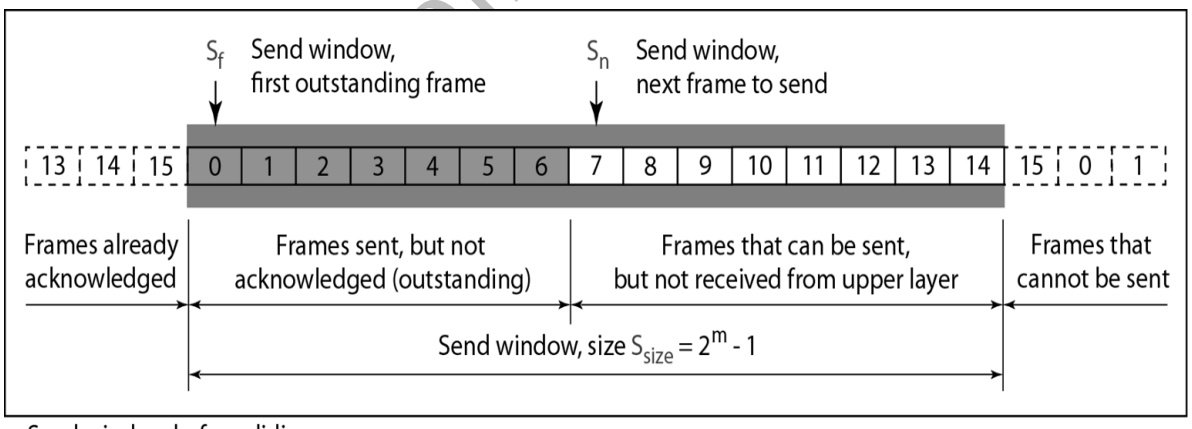
Disadvantages of stop and wait ARQ:

1. Efficiency is very less.
2. Only 1 frame is sent at a time
3. Timer should be set for each individual frame
4. No pipelining
5. Sender window size is 1(disadvantage over Go back N ARQ)
6. Receiver window size is 1(disadvantage over selective repeat ARQ)

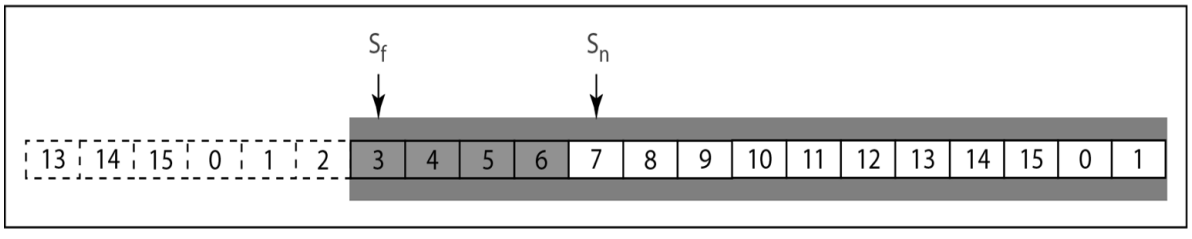
2. Go-Back-N Automatic Repeat reQuest (ARQ*)* *: It improves the efficiency stop and wait-ARQ protocol by transmission multiple frames must be in transition while waiting for acknowledgment. It Keep channel busy by continuing to send frames. Several frames are sending before receiving acknowledgments. It keeps a copy of these frames until the acknowledgments arrive. Use m-bit sequence numbering. Sequence numbers are modulo 2m, where m is the size of the sequence number field in bits. For example, if m is 4, the only sequence numbers are 0 through 15 inclusive. However, we can repeat the sequence. So the sequence numbers are 0, 1,2,3,4,5,6, 7,8,9, 10, 11, 12, 13, 14, 15,0, 1,2,3,4,5,6,7,8,9,10, 11, …

Procedure:

- Introduce a window of size n. Can inject n packets into net before hearing an ACK
- Use Sliding window. Sliding window is an abstract concept that defines the range of sequence numbers that is the concern of the sender and receiver.
- The sender and receiver need to deal with only part of the possible sequence numbers.
- Label each packet with a sequence number
- A window is a collection of adjacent sequence numbers
- The range which is the concern of the sender is called the "send sliding window"
- The range that is the concern of the receiver is called the "receive sliding window".
- The send window is an imaginary box covering the sequence numbers of the data frames which can be in transit.
- The window at any time divides the possible sequence numbers into four regions shown in figure 2.18
  1. *First Region*: It defines the sequence numbers belonging to frames that are already ACKed.
  2. *Second Region*: It defines the range of sequence numbers belonging to the frames that are sent and have an unknown status called these "outstanding frames."
  3. *Third Region*: It defines the range of sequence numbers for frames that can be sent; but not yet been received from the network layer.
  4. *Fourth Region:* It defines sequence numbers that cannot be used until the window slides



a. Send window before sliding



b. Send window after sliding

Figure 2.18 Send windows for Go-Back-N - ARQ

- The send window is an abstract concept defining an imaginary box of size 2^{m-1} with three variables: Sf, Sn, and Ssize.
- The send window can slide one or more slots when a valid acknowledgment arrives. Figure 2.19 : Receive window for Go-Back-N ARQ

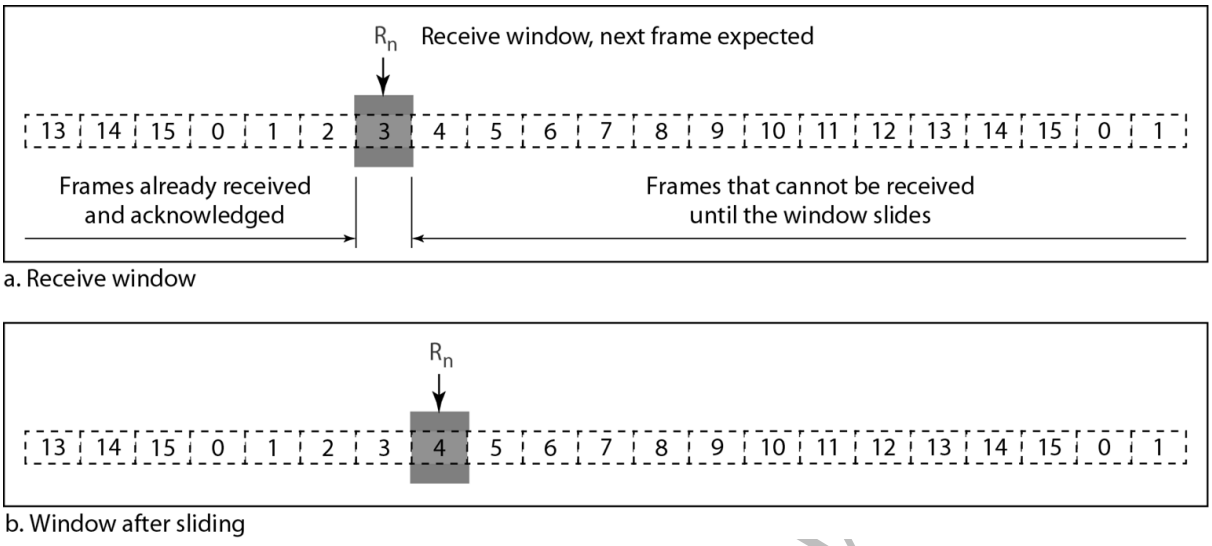


Figure 2.19 Receive windows for Go-Back-N - ARQ

- The receive window is an abstract concept defining an imaginary box of size 1 with one single variable Rn. The window slides when a correct frame has arrived; sliding occurs one slot at a time

Timers:

- Although there can be a timer for each frame that is sent, in this protocol use only one.
- The reason is that the timer for the first outstanding frame always expires first; sends all outstanding frames when this timer expires.

Acknowledgment:

- The receiver sends a positive acknowledgment if a frame has arrived safe and sound and in order.
- If a frame is damaged or is received out of order, the receiver is silent and will discard all subsequent frames until it receives the one it is expecting.
- The silence of the receiver causes the timer of the unacknowledged frame at the sender site to expire. This, in turn, causes the sender to go back and resend all frames, beginning with the one with the expired timer.
- The receiver does not have to acknowledge each frame received. It can send one cumulative acknowledgment for several frames.

Resending a Frame: When the timer expires, the sender resends all outstanding frames. For example, suppose the sender has already sent frame 6, but the timer for frame 3 expires. This means that frame 3 has not been acknowledged; the sender goes back and sends frames 3, 4, 5, and 6 again. That is why the protocol is called *Go-Back-N* ARQ.

(2.1). Design of Go-Back-N ARQ*:* Figure 2.20 shows the design for this protocol. Here multiple frames can be in transit in the forward direction, and multiple acknowledgments in the reverse direction. The idea is similar to Stop-and-Wait ARQ; the difference is that the send window allows us to have as many frames in transition as there are slots in the send window.

Send Window Size*:*In Go-Back-N ARQ, the size of the send window must be less than 2^m ; the size of the receiver window is always 1. why the size of the send window must be less than *2m.* As an example, we choose *m* =2, which means the size of the window can be *2m* - 1, or 3.Figure 2.21 compares a window size of 3 against a window size of 4. If the size of the window is 3 (less than 2^2) and all three acknowledgments are lost, the frame timer expires and all three frames are resent. The receiver is now expecting frame 3, not frame 0, so the duplicate frame is correctly discarded. On the other hand, if the size of the window is 4 (equal to 2^2) and all acknowledgments are lost, the sender will send a duplicate of frame 0. However, this time the window of the receiver expects to receive frame 0, so it accepts frame 0, not as a duplicate, but as the first frame in the next cycle. This is an error.

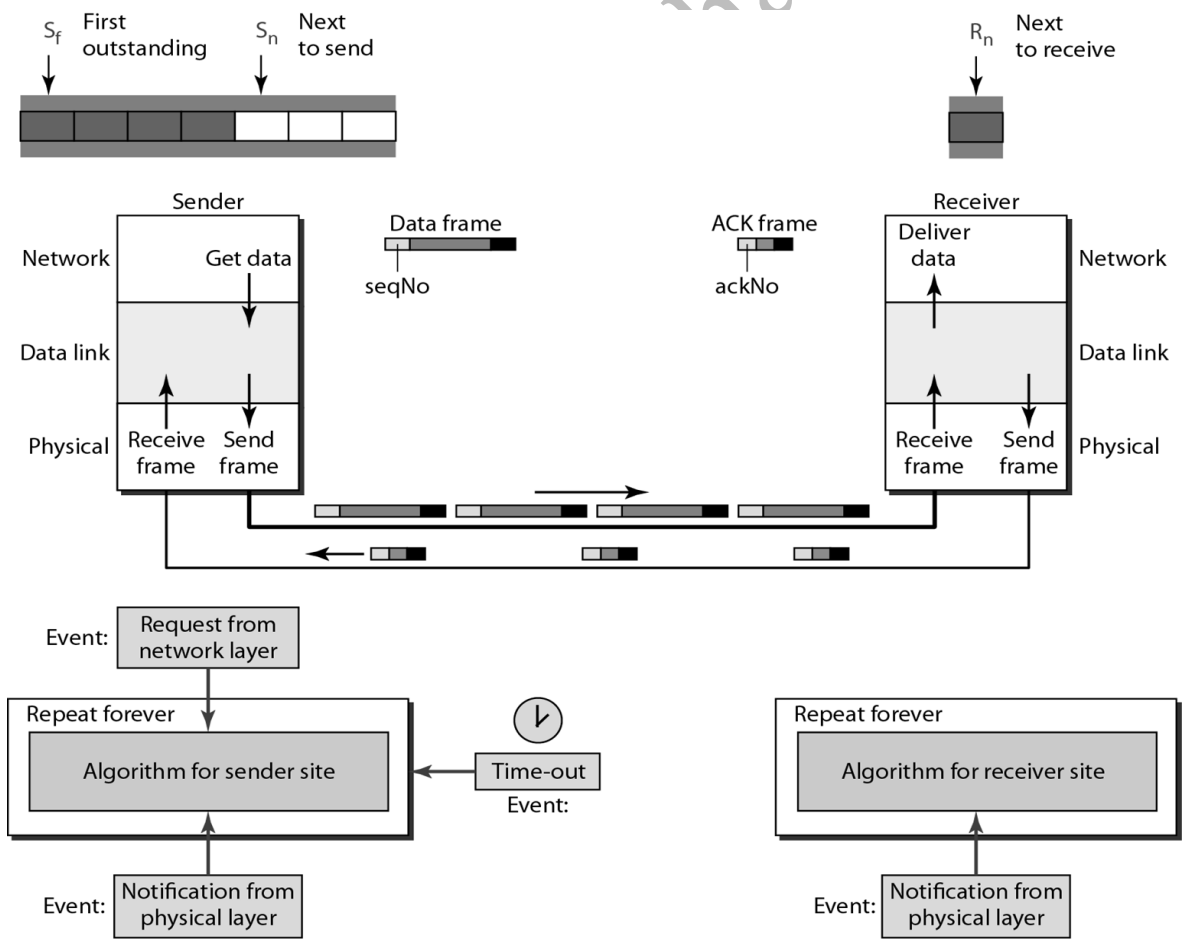


Figure 2.20 Design of Go-Back-NARQ

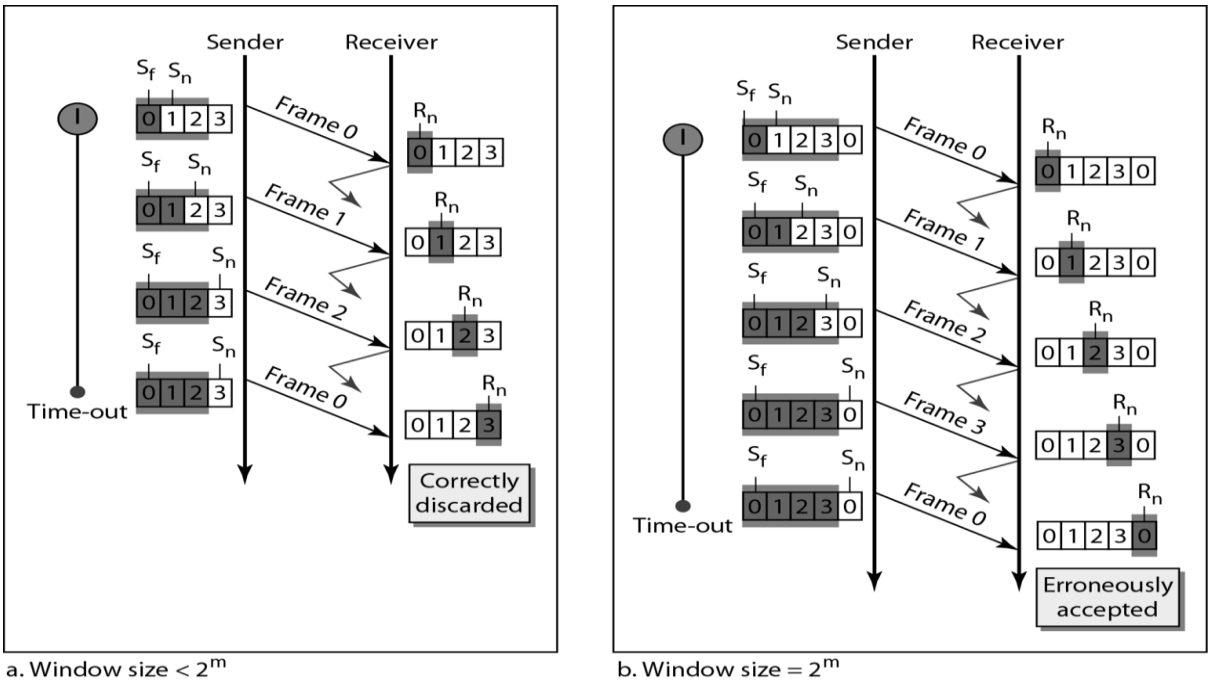


Figure 2.21 Window size for Go-Back-NARQ

Example 1

- Figure 2.22 shows an example of Go-Back-N.

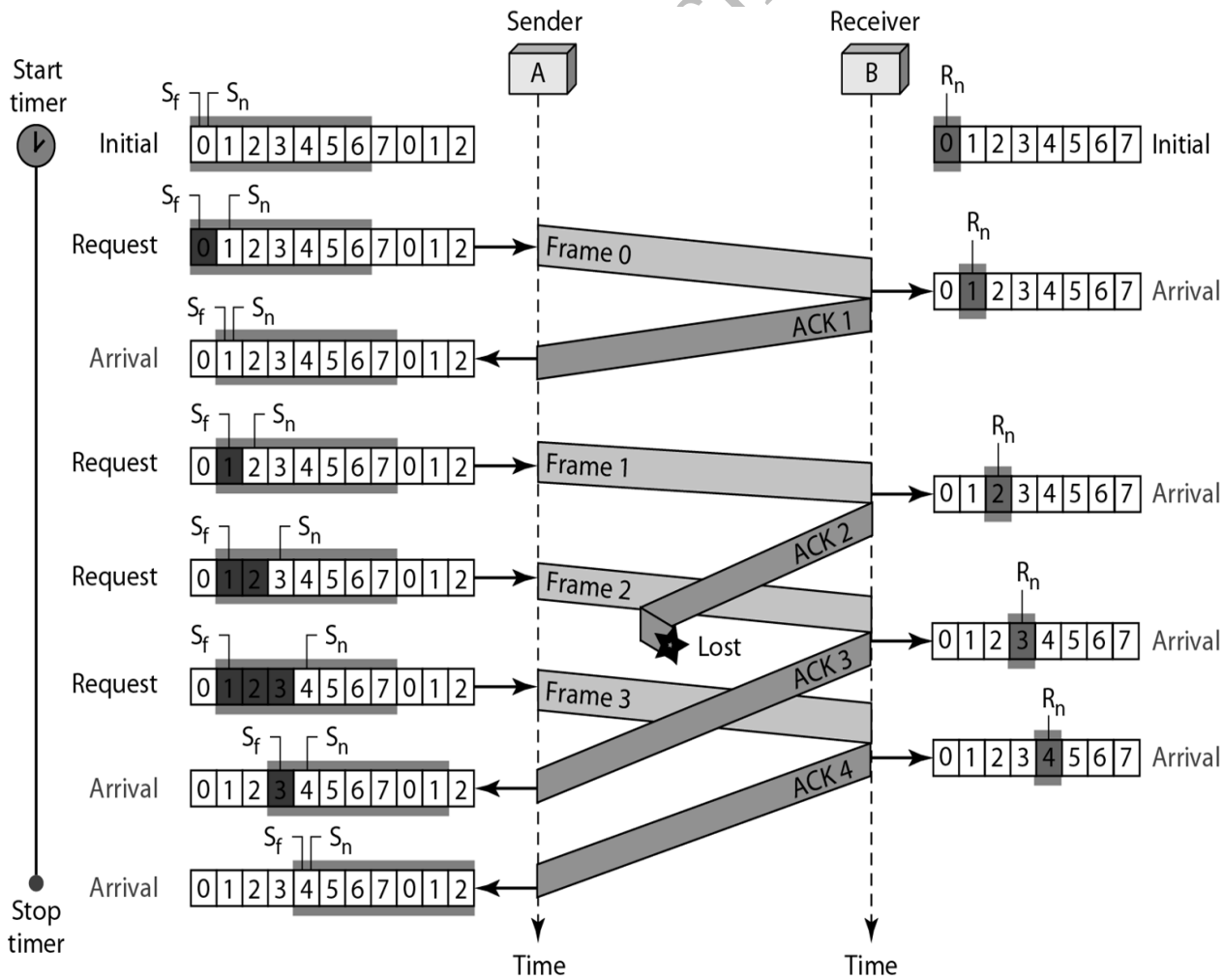


Figure 2.22 Flow diagram for Example 1

- This is an example of a case where the forward channel is reliable, but the reverse is not.
- No data frames are lost, but some ACKs are delayed and one is lost.
- The example also shows how cumulative acknowledgments can help if ACKs are delayed or lost.
- After initialization, there are seven sender events. Request events are triggered by data from the network layer; arrival events are triggered by acknowledgments from the physical layer.
- There is no time-out event here because all outstanding frames are ACKed before the timer expires.
- Note that although ACK 2 is lost, ACK 3 serves as both ACK 2 and ACK 3.

Example 2

- Figure 2.23 shows what happens when a frame is lost.
- Frames 0, 1, 2, and 3 are sent. However, frame 1 is lost. The receiver receives frames 2 and 3, but they are discarded because they are received out of order.
- The sender receives no acknowledgment about frames 1, 2, or 3. Its timer finally expires.
- The sender sends all outstanding frames (1, 2, and 3) because it does not know what is wrong. Note that the resending of frames 1, 2, and 3 is the response to one single event.
- When the sender is responding to this event, it cannot accept the triggering of other events. This means that when ACK 2 arrives, the sender is still busy with sending frame 3.
- The physical layer must wait until this event is completed and the data link layer goes back to its sleeping state.
- We have shown a vertical line to indicate the delay. It is the same story with ACK 3; but when ACK 3 arrives, the sender is busy responding to ACK 2. It happens again when ACK 4 arrives.
- Note that before the second timer expires, all outstanding frames have been sent and the timer stopped.

Advantages:

1. The sender can send many frames at a time.
2. Timer can be set for a group of frames.
3. Only one ACK can acknowledge one or more frames.
4. Efficiency is more.
5. Waiting time is pretty low.
6. We can alter the size of the sender window

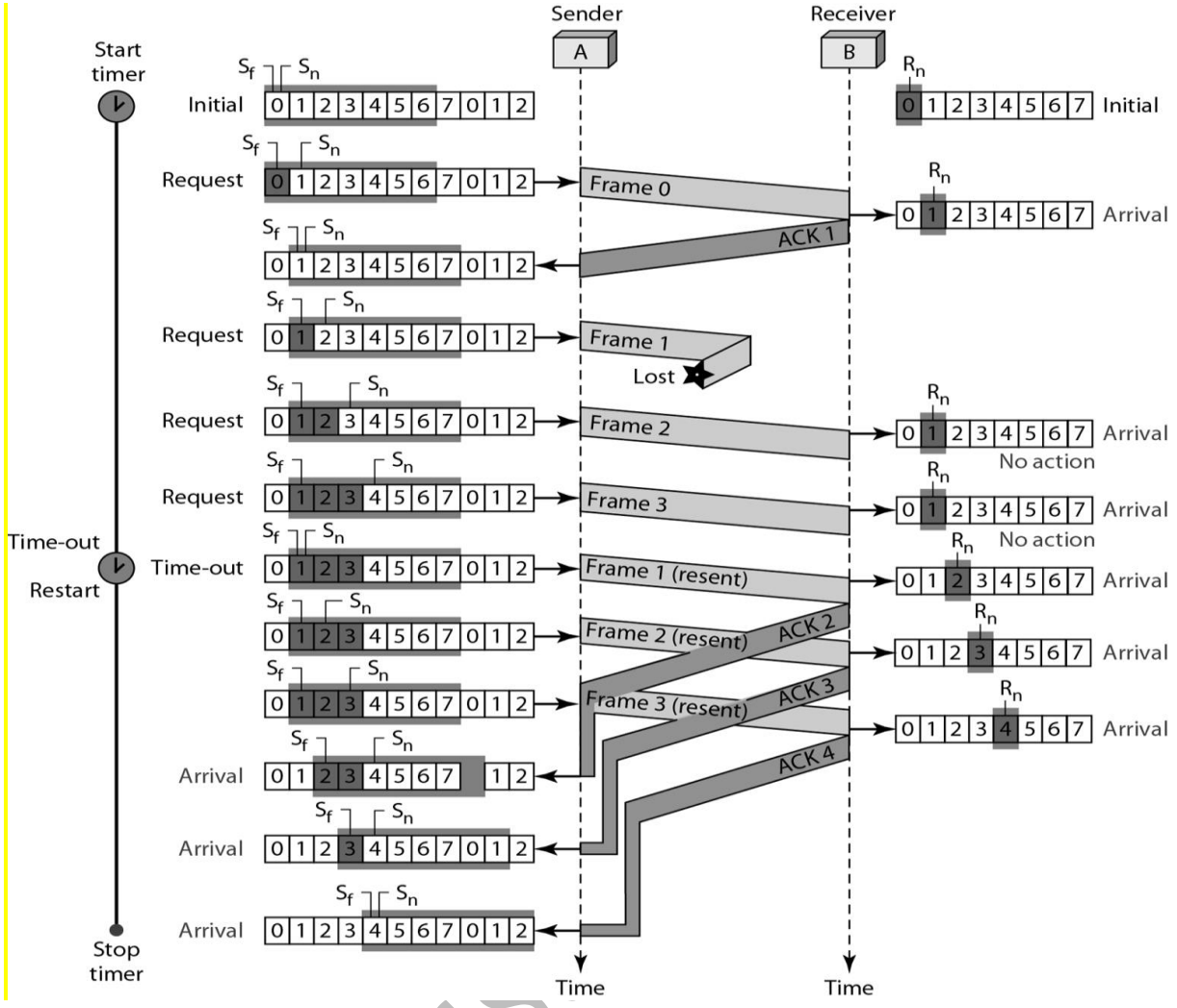


Figure 2.23 Flow diagrams for Example 2

Disadvantages:

1. This protocol is very inefficient for a noisy link.
2. Buffer requirement:
3. Transmitter needs to store the last N packets
4. Scheme is inefficient when round-trip delay large and data transmission rate is high
5. Retransmission of many error-free packets following an erroneous packet
6. When RTT is large: for a high number of NACK, a lot of BW is wasted
7. If NACK is lost, a long time is wasted until re-transmission of all packets (until NACK is sent).

3. Selective Repeat Automatic Repeat Request*:** A mechanism that does not resend N frames when just one frame is damaged; only the damaged frame is resent called "Selective Repeat ARQ." Go-Back-N ARQ inefficient because multiple frames are resent when errors or losses occur. Selective Repeat retransmits only an individual frame. Timeout causes individual

corresponding frame to be resent.NAK causes retransmission of oldest un-ACked frame. It is more efficient for noisy links, but the processing at the receiver is more complex.

3.1 Type of Windows: The Selective Repeat Protocol also uses two windows: *Send window* & *Receive window.* This is shown in figure 2.24 and 2.25

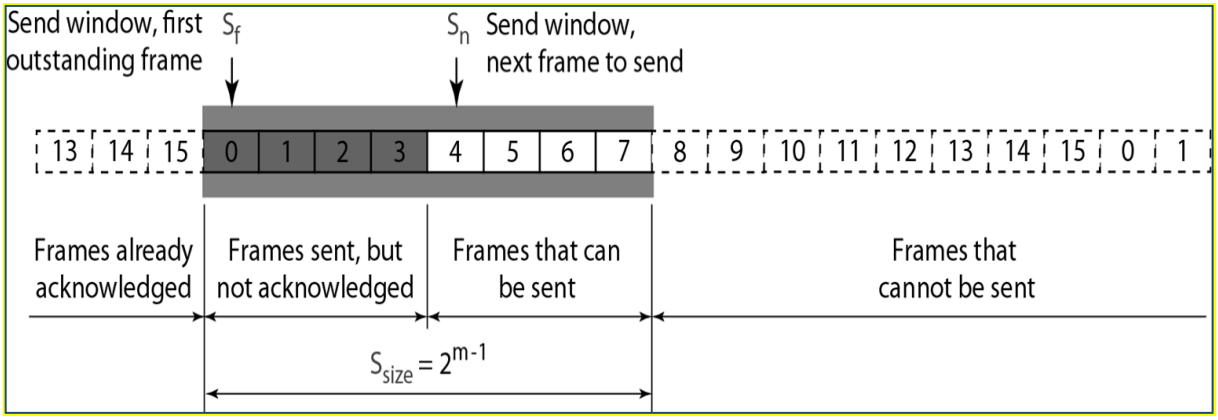


Figure 2.24 Send windows for Selective Repeat ARQ

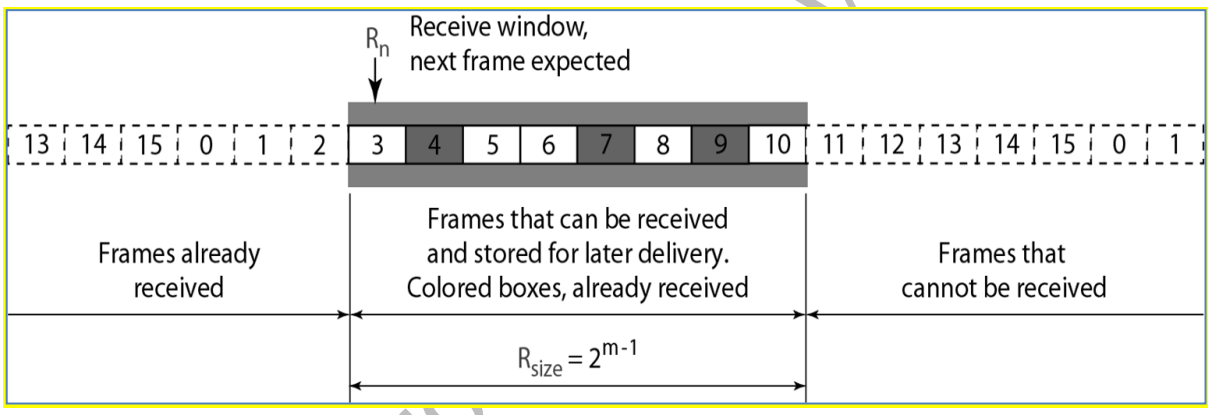


Figure 2.25 Receive windows for Selective Repeat ARQ

Windows Description:

- This Protocol also uses two windows: a send window and a receive window.However, there are differences between the windows in this protocol and the ones in Go-Back-N.
  - First, the size of the send window is much smaller.
  - Second, the receive window is the same size as the send window.
- The smaller window size means less efficiency in filling the pipe, but the fact that there are fewer duplicate frames can compensate for this.
- The protocol uses the same variables as we discussed for Go-Back-N.
- This Protocol allows as many frames as the size of the receive window to arrive out of order and be kept until there is a set of in-order frames to be delivered to the network layer.

- Those slots inside the window that are colored define frames that have arrived out of order and are waiting for their neighbors to arrive before delivery to the network layer.

(3.2). Design*:* The design in this case is to some extent similar to the one we described for the Go Back- N, but more complicated, as shown in Figure 2.26

(3.3).Window size*:* In this the size of the sender and receiver windows must be at most one half of $2^{m.}$ For an example, we choose m = 2, which means the size of the window is $2^m/2$, or 2. Figure 2.27 compares a window size of 2 with a window size of 3.If the size of the window is 2 and all acknowledgments are lost, the timer for frame 0 expires and frame 0 is resent. However, the window of the receiver is now expecting frame 2, not frame 0, so this duplicate frame is correctly discarded. When the size of the window is 3 and all acknowledgments are lost, the sender sends a duplicate of frame 0. However, this time, the window of the receiver expects to receive frame 0 (0 is part of the window), so it accepts frame 0, not as a duplicate, but as the first frame in the next cycle. This is clearly an error.

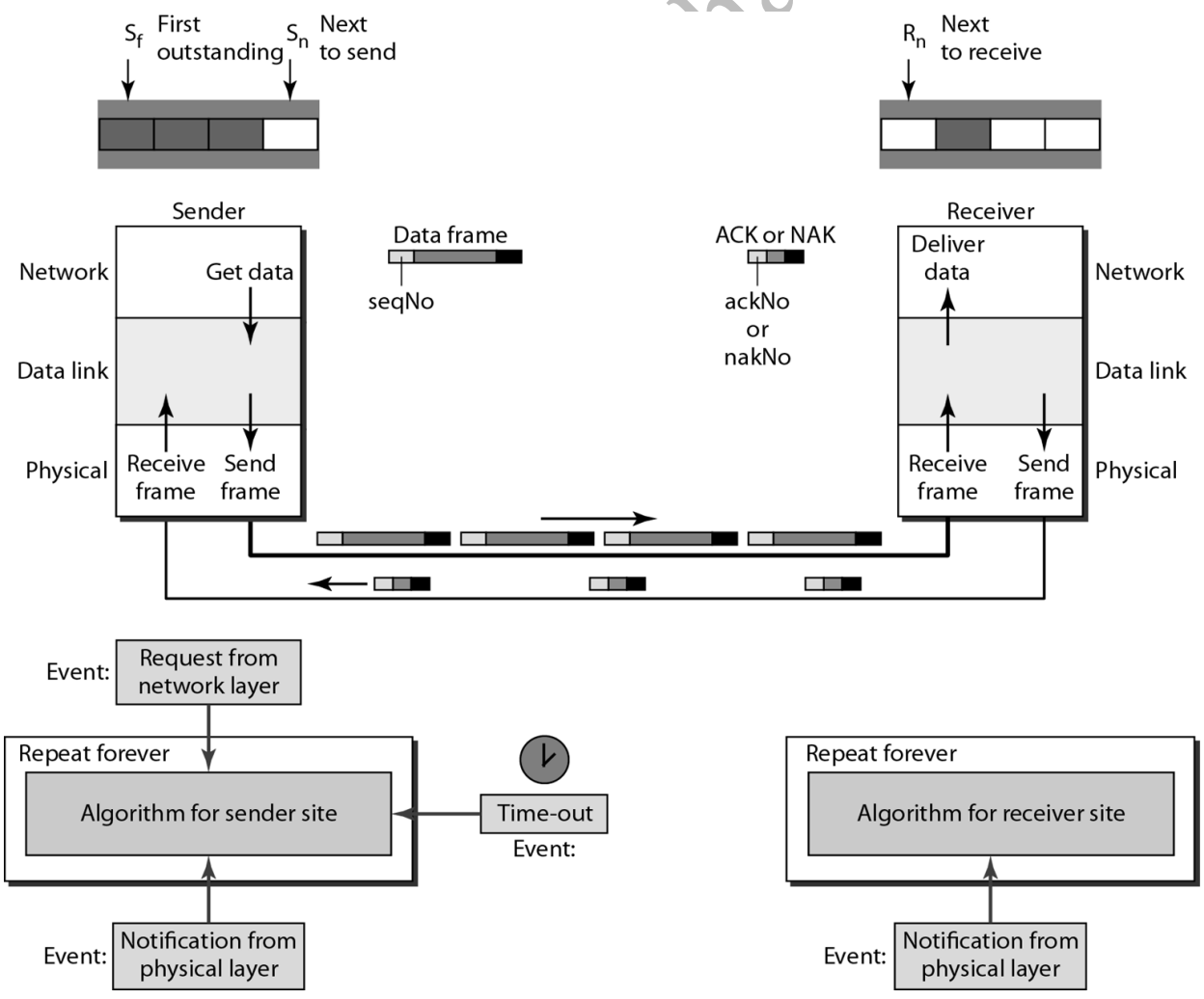


Figure 2.26 Design of Selective Repeat ARQ

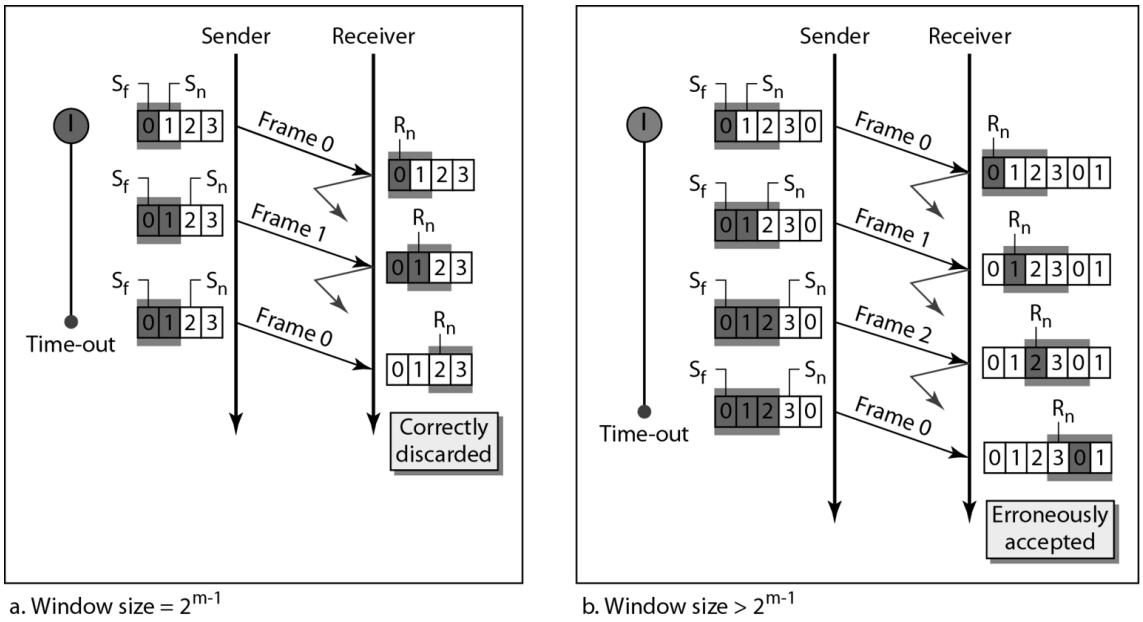


Figure 2.27 Selective Repeat ARQ, window size

(3.4) Analysis:

1. The handling of the request event is similar to that of the previous protocol except that one timer is started for each frame sent.
2. The arrival event is more complicated here. An ACK or a NAK frame may arrive. If a valid NAK frame arrives, we just resend the corresponding frame.
3. If a valid ACK arrives, we use a loop to purge the buffers, stop the corresponding timer and move the left wall of the window.
4. The time-out event is simpler here; only the frame which times out is resent.

Example 1: *Selective Repeat Automatic Repeat Request :* In this example frame 1 is lost. It shows how Selective Repeat behaves in this case. Figure 2.28 shows the situation.

- One main difference is the number of timers. Here, each frame sent or resent needs a timer, which means that the timers need to be numbered (0, 1, 2, and 3).
- The timer for frame 0 starts at the first request, but stops when the ACK for this frame arrives.
- The timer for frame 1 starts at the second request restarts when a NAK arrives, and finally stops when the last ACK arrives. The other two timers start when the corresponding frames are sent and stop at the last arrival event.
- At the receiver site we need to distinguish between the acceptance of a frame and its delivery to the network layer.
- At the second arrival, frame 2 arrives and is stored and marked, but it cannot be delivered because frame 1 is missing.

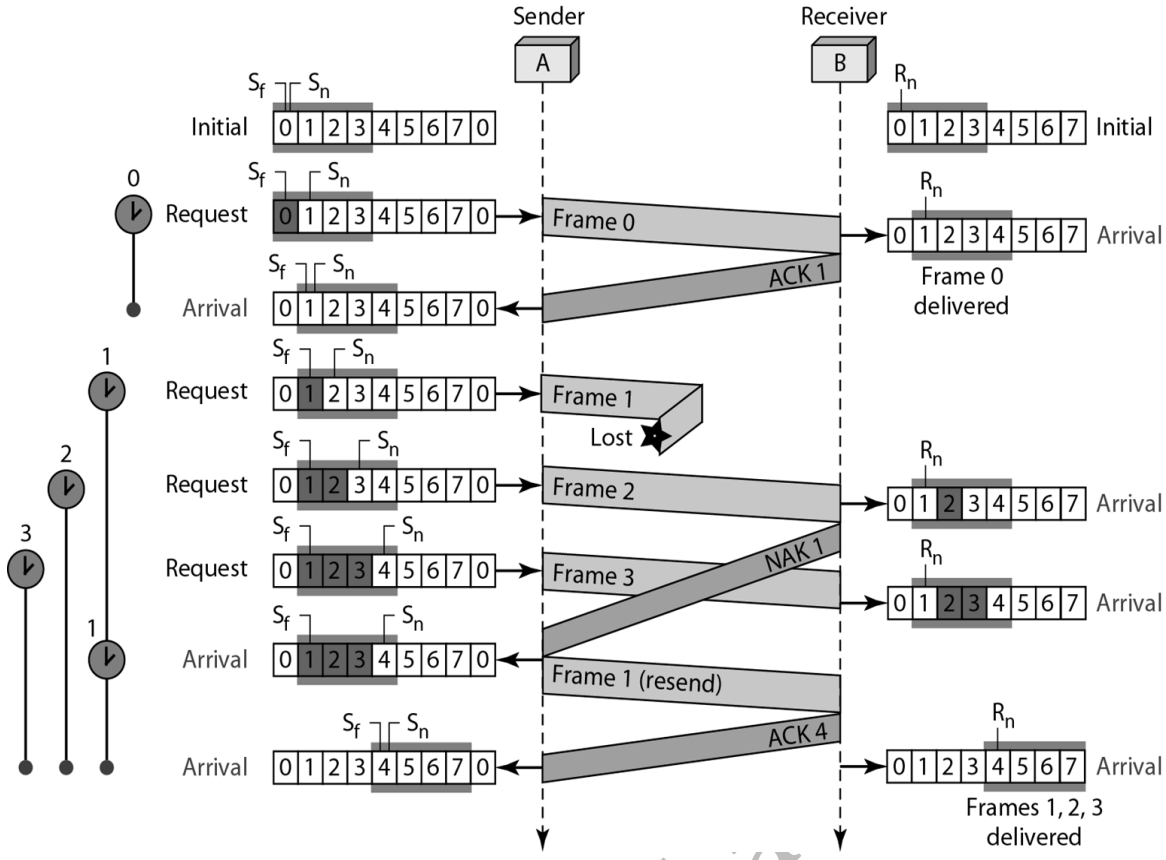


Figure 2.28 Flow diagrams for Example 1

- At the next arrival, frame 3 arrives and is marked and stored, but still none of the frames can be delivered. Only at the last arrival.
- When finally a copy of frame 1 arrives, can frames 1, 2, and 3 be delivered to the network layer.
- There are two conditions for the delivery of frames to the network layer: First, a set of
- consecutive frames must have arrived. Second, the set starts from the beginning of the window. Another important point is that a NAK is sent after the second arrival, but not after the third, although both situations look the same. The reason is that the protocol does not want to crowd the network with unnecessary NAKs and unnecessary resent frames.
- The second NAK would still be NAK1 to inform the sender to resend frame 1 again; this has already been done.
- The first NAK sent is remembered (using the NAK Sent variable) and is not sent again until the frame slides. A NAK is sent once for each window position and defines the first slot in the window. The next point is about the ACKs. Notice that only two ACKs are sent here.
- The first one acknowledges only the first frame; the second one acknowledges three frames.
- In Selective Repeat, ACKs are sent when data are delivered to the network layer. If the data belonging to n frames are delivered in one shot, only one ACK is sent for all of them

Piggybacking:

The three protocols discussed in the previous section are all unidirectional: data frames flow in only one direction although control information such as ACK and NAK frames can travel in the other direction. But in practice data frames are normally flowing in both directions: from node A to node B and from node B to node A. This means that the control information also needs to flow in both directions. A technique called *piggybacking* is used to improve the efficiency of the bidirectional protocols. When a frame is carrying data from A to B, it can also carry control information about arrived (or lost) frames from B; when a frame is carrying data from B to A, it can also carry control information about the arrived (or lost) frames from A.

The design for a Go-Back-N ARQ using piggybacking shown in Figure 2.29

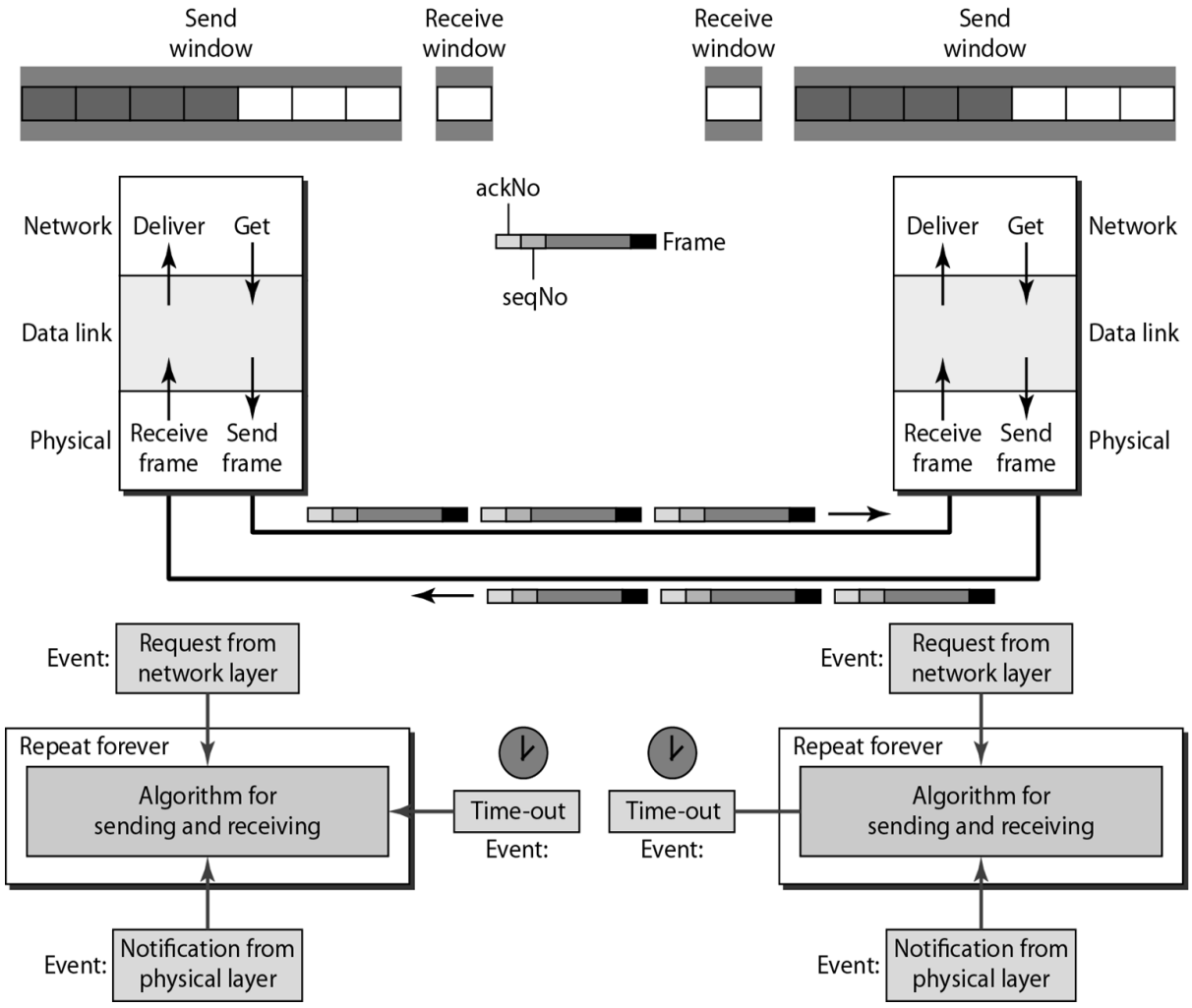


Figure 2.29 Design of piggybacking in Go-Back-NARQ

In this each node now has two windows: one send window and one receive window. Both so needs to use a timer. Both are involved in three types of events: request, arrival, and time-out. However, the arrival event here is complicated; when a frame arrives, the site needs to handle control information as well as the frame itself. Both of these concerns must be taken care of in

one event, the arrival event. The request event uses only the send window at each site; the arrival event needs to use both windows. An important point about piggybacking is that both sites must use the same algorithm. This algorithm is complicated because it needs to combine two arrival events into one.

1. Comparison table for noisy channel protocols

Protocol	**Go-Back-N**	**Stop And Wait**	**Selective Repeat**
Bandwidth utilization	Medium	Low	High
Maximum sender Size Window	2^m-1	N.A	2^(m-1)
Maximum receiver Size Window	1	N.A	2^(m-1)
Pipelining	Implemented	Not Implemented	Implemented
Out of order Frames	Discarded	Discarded	Accepted
Cumulative ACK	Applicable	N.A	Applicable
NAK	N.A	N.A	Applicable

2.3. HDLC (High-Level Data Link Control) ****

(A).<u>Introduction:</u> It is Old and still heavily used protocol. It is a classical bit-oriented protocol for communication over point-to-point and multipoint links. It implements the ARQ mechanisms. HDLC is Mother of many LAN and WAN DLC protocols. This protocol based on some principles.

1. All are bit oriented.
2. All uses bit stuffing for data transferency

(*B).<u>Configurations and Transfer Modes of HDLC</u>*: HDLC provides two common transfer modes that can be used in different configurations:

1. Normal response mode (NRM)
2. Asynchronous balanced mode (ABM).

1. Normal response mode (NRM)

a. Used with unbalanced configuration shown in figure 2.30
b. Unbalanced: one primary station, one or more secondary stations
c. Primary initiates data transfer; secondary can only reply
d. The NRM is used for both point-to-point and multiple-point links
e. Primary station : sends data, controls the link with commands
f. Secondary station : receives data, responds to control messages

g. Combined station : can issue both commands and responses

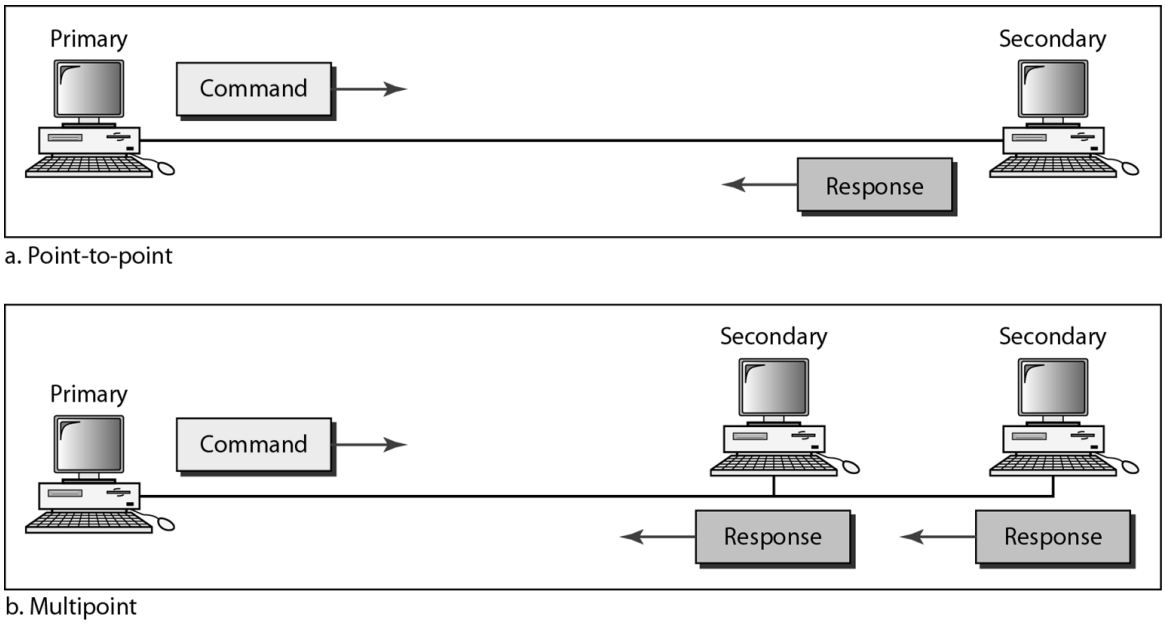


Figure 2.30 Normal response mode

2. Asynchronous balanced mode (ABM):

a. Used with balanced configurations shown in figure 2.31

b. Balanced: two combined stations.

c. Either side may send data at any time.

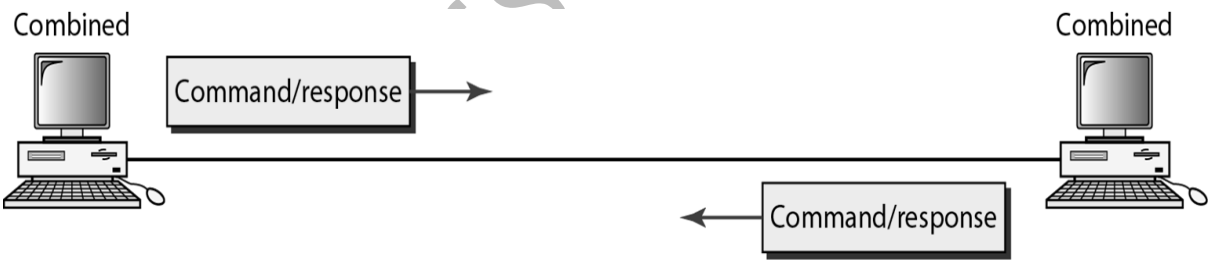


Figure 2.31 Asynchronous balanced mode

4.1. HDLC Frames **:** To provide the flexibility necessary to support all the options possible in the modes and configurations, HDLC defines three types of frames:

(1) ***Information frames (I-frames):*** It is used to transport user data and control information relating to user data.

(2) ***Supervisory frames (S-frames):*** It is used only to transport control information-frames are reserved for system management

(3) ***Unnumbered frames (U-frames*)**: It is intended for managing the link itself.

HDLC frame format:

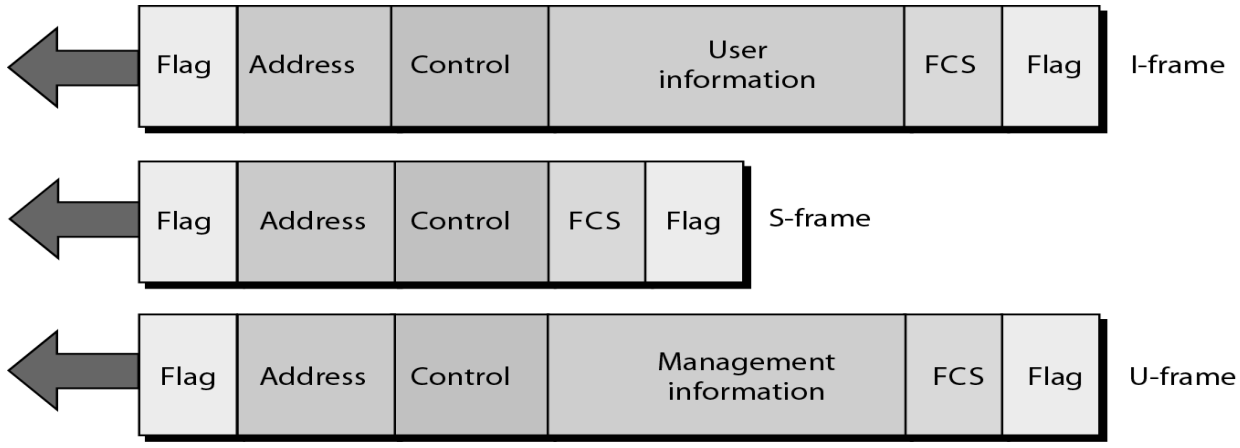


Figure 2.32 HDLC frames

HDLC Frames Description: Each frame in HDLC may contain up to six fields, they are

1. Flag Field.
2. Address Field.
3. Control Field.
4. Information Field.
5. Frame Check Sequence (FCS) Field.
6. Ending Flag Field.

1. Flag field

- The flag field of an HDLC frame is an 8-bit sequence with the bit pattern 01111110 that identifies both the beginning and the end of a frame and serves as a synchronization pattern for the receiver.

2. Address field

- The second field of an HDLC frame contains the address of the secondary station.
- If a primary station created the frame, it contains a *to address*.
- If a secondary creates the frame, it contains *a from* address.
- An address field can be 1 byte or several bytes long, depending on the needs of the network.
- One byte can identify up to 128 stations.
- Larger networks require multiple-byte address fields, shown in figure 2.33
- If the address field is only 1 byte, the last bit is always a 1.
- If the address is more than 1 byte, all bytes but the last one will end with 0; only the last will end with 1.

- Ending each intermediate byte with 0 indicates to the receiver that there are more address bytes to come.
- If the link is strictly point-to-point, the value of the field will be 10000000, as the address is not relevant. An address of 11111111 represents “all”

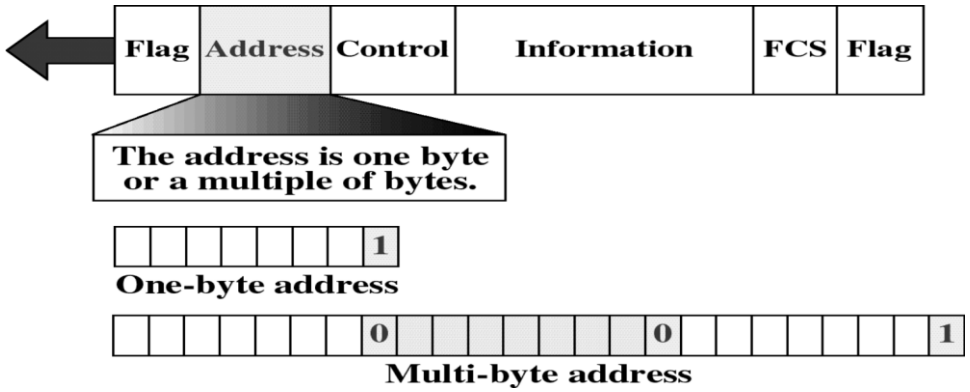


Figure 2.33 Address field in HDLC frame

3. Control field

- The control field is a 1- or 2-byte segment of the frame used for flow and error control.
- The interpretation of bits in this field depends on the frame type.

4. Information field

- The information field contains the user's data from the network layer or management information.
- Its length can vary from one network to another.

5. FCS field

- The frame check sequence (FCS) is the HDLC error detection field. It can contain either a 2- or 4-byte ITU-T CRC.

Control Field Description*:* The control field determines the type of frame and defines its functionality. The format is specific for the type of frame, as shown in figure 2.34

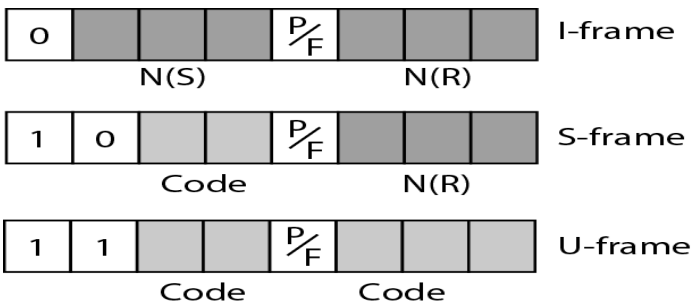


Figure 2.34: Control field format for the different frame types

1. Control Field for I-Frames

- I-frames are designed to carry user data from the network layer.
- They can include flow and error control information (piggybacking).
- The subfields in the control field are used to define following functions.
    - The first bit defines the type.
    - First bit of the control field = 0
- N(S) Sequence Number of the Sent Frame
    - 3 Bits Used.
    - Sequence number between 0 and 7.
    - Extension format, control field is 2 bytes
- N(R) Receive Sequence Number
    - The last 3 bits used
    - It corresponds to the acknowledgment number when piggybacking is used.
- The P/F bit
    - The single bit between N(S) and N(R) is called the P/F bit.
    - Poll/Final is a single bit with two names. It is called *Poll* when set by the primary station to obtain a response from a secondary station. *Final* when set by the secondary station to indicate a response or the end of transmission.In all other cases, the bit is clear.

2. Control Field for S-Frames

- If the first 2 bits of the control field is 10, this means the frame is an S-frame.
- Supervisory frames are used for flow and error control whenever piggybacking is either impossible or inappropriate frames do not have information fields.
- The last 3 bits, called N(R), corresponds to the acknowledgment number (ACK) or negative acknowledgment number (NAK) depending on the type of S-frame.
- The 2 bits called code is used to define the type of S-frame itself. With 2 bits, we can have four types of S-frames, as described below:

a) Receive ready (RR).

- If the value of the code subfield is 00, it is an RR S-frame.
- This kind of frame acknowledges the receipt of a safe and sound frame or group of frames.
- In this case, the value *N(R)* field defines the acknowledgment number.

b) Receive not ready (RNR).

- If the value of the code subfield is 10, it is an RNR S-frame.
- This kind of frame is an RR frame with additional functions.
- It acknowledges the receipt of a frame or group of frames, and it announces that the receiver is busy and cannot receive more frames.
- It acts as a kind of congestion control mechanism by asking the sender to slow down.
- The value of N(R) is the acknowledgment number.

c) Reject (REJ).

- If the value of the code subfield is 01, it is a REJ S-frame.
- This is a NAK frame, but not like the one used for Selective Repeat ARQ.
- It is a NAK that can be used in Go-Back-N ARQ to improve the efficiency of the process by Informing the sender, before the sender time expires, that the last frame is lost or damaged.
- The value of NCR) is the negative Acknowledgment number.

d) Selective reject (SREJ).

- If the value of the code subfield is 11, it is an SREJ S-frame.
- This is a NAK frame used in Selective Repeat ARQ.
- Note that the HDLC Protocol uses the term *selective reject* instead of *selective repeat.*
- The value of *N(R)* is the negative acknowledgment number.

3. Control Field for V- Frames

- Unnumbered frames are used to exchange session management and control information between connected devices.
- Unlike S-frames, U-frames contain an information field, but one used for system management information, not user data.
- As with S-frames, however, much of the information carried by U-frames is contained in codes included in the control field.
- U-frame codes are divided into two sections: a 2-bit prefix before the P/F bit and a 3-bit suffix after the P/F bit. Together, these two segments (5 bits) can be used to create up to 32 different types of U-frames as shown in table 2.1

Table 2.1 U~frame control command and response

Code	*Command*	*Response*	*Meaning*
00 001	SNRM		Set normal response mode
11 011	SNRME		Set normal response mode, extended
11 100	SABM	**DM**	Set asynchronous balanced mode or **disconnect mode**
11 110	SABME		Set asynchronous balanced mode, extended
00 000	UI	**UI**	Unnumbered information
00 110		**UA**	**Unnumbered acknowledgment**
00 010	DISC	**RD**	Disconnect or **request disconnect**
10 000	SIM	**RIM**	Set initialization mode or **request information mode**
00 100	UP		Unnumbered poll
11 001	RSET		Reset
11 101	XID	**XID**	Exchange ID
10 001	FRMR	**FRMR**	Frame reject

SUMMARY OF UNIT TWO: DATA LINK LAYER

- Data link control deals with the design and procedures for communication between two adjacent nodes: node-to-node communication.
- Framing in the data link layer separates a message from one source to a destination, or from other messages going from other sources to other destinations.
- Frames can be of fixed or variable size. In fixed-size framing, there is no need for defining the boundaries of frames; in variable-size framing, we need a delimiter (flag) to define the boundary of two frames.
- Variable-size framing uses two categories of protocols: byte-oriented (or character oriented) and bit- oriented.
- In a byte-oriented protocol, the data section of a frame is a sequence of bytes; in a bit oriented protocol, the data section of a frame is a sequence of bits.
- In byte-oriented (or character-oriented) protocols, we use byte stuffing; a special byte added to the data section of the frame when there is a character with the same pattern as the flag.
- In bit-oriented protocols, we use bit stuffing; an extra 0 is added to the data section of the frame when there is a sequence of bits with the same pattern as the flag.
- Flow control refers to a set of procedures used to restrict the amount of data that the sender can send before waiting for acknowledgment. Error control refers to methods of error detection and correction.
- For the noiseless channel, we discussed two protocols: the Simplest Protocol and the Stop and-Wait Protocol. The first protocol has neither flow nor error control; the second has n error control.
- In the Simplest Protocol, the sender sends its frames one after another with no regards to the receiver.
- In the Stop-and-Wait Protocol, the sender sends one frame, stops until it receives confirmation from the receiver, and then sends the next frame.
- For the noisy channel, three protocols: Stop-and-Wait ARQ, G0Back-N, and Selective Repeat ARQ.
- The Stop-and-Wait ARQ Protocol, adds a simple error control mechanism to the Stop-and-Wait Protocol.
- In the Go-Back-N ARQ Protocol, we can send several frames before receiving acks, improving the efficiency of transmission.
- In the Selective Repeat ARQ protocol we avoid unnecessary transmission by sending only frames that are corrupted.

- Both Go-Back-N and Selective-Repeat Protocols use a sliding window. In Go Back- N ARQ, if m is the number of bits for the sequence number, then the size of the send window must be less than 2m; the size of the receiver window is always 1. In Selective Repeat ARQ, the size of the sender and receiver window must be at most one-half of 2m.
- A technique called piggybacking is used to improve the efficiency of the bidirectional protocols. When a frame is carrying data from A to B, it can also carry control information about frames from B; when a frame is carrying data from B to A, it can also carry control information about frames from A.
- High-level Data Link Control (HDLC) is a bit-oriented protocol for communication over point-to-point and multipoint links. However, the most common protocols for point-to-point access are the Point-to-Point Protocol (PPP), which is a byte-oriented protocol.

UNIT -2 DATA LINK LAYER QUESTION BANK

1. Briefly describe the services provided by the data link layer.

2. Define framing and the reason for its need.

3. Compare and contrast byte-oriented and bit-oriented protocols. Which category has been popular in the past (explain the reason)? Which category is popular now?

4. Compare and contrast byte-stuffing and bit-stuffing. Which technique is used in byte oriented protocols? Which technique is used in bit-oriented protocols?

5. Compare and contrast flow control and error control.

6. What are the two protocols we discussed for noiseless channels in this chapter?

7. What are the three protocols we discussed for noisy channels in this chapter?

8. Explain the reason for moving from the Stop-and-Wait ARQ Protocol to the Go Back- NARQ Protocol.

9. Compare and contrast the Go-Back-NARQ Protocol with Selective-Repeat ARQ.

10. Compare and contrast HDLC with PPP. Which one is byte-oriented; which one is bit-oriented?

11. Define piggybacking and its usefulness.

12. Which of the protocols described in this chapter utilize pipelining?

15. Design two simple algorithms for byte-stuffing. The first adds bytes at the sender; the second removes bytes at the receiver.

16. Design two simple algorithms for bit-stuffing. The first adds bits at the sender; the second removes bits at the receiver.

17. A sender sends a series of packets to the same destination using 5-bit sequence numbers. If the sequence number starts with 0, what is the sequence number after sending 100 packets?

18. Using 5-bit sequence numbers, what is the maximum size of the send and receive windows for each of the following protocols?

(a). Stop-and-Wait ARQ (b). Go-Back-NARQ (c). Selective-Repeat ARQ

19. Design a bidirectional algorithm for the Simplest Protocol using piggybacking. Note that both parties need to use the same algorithm.

20. Design a bidirectional algorithm for the Stop-and-Wait Protocol using piggybacking. Note that both parties need to use the same algorithm.

21. Design a bidirectional algorithm for the Stop-and-Wait ARQ Protocol using piggybacking. Note that both parties need to use the same algorithm.

22. Design a bidirectional algorithm for the Go-Back-N ARQ Protocol using piggybacking. Note that both parties need to use the same algorithm.

23. Design a bidirectional algorithm for the Selective-Repeat ARQ Protocol using piggybacking. Note that both parties need to use the same algorithm.

UNIT 2- DATA LINK LAYER - SOLVED EXAMPLES

1. A channel has a bit rate of 4 kbps and a propagation delay of 20 msec. For what range of a frame size does stop and wait protocol give an efficiency of at least 50%.

Solution:

The bandwidth-delay product (capacity) is = B.W * Delay per bit

$$= (4*10^3)*(20*10^{-3}) = 80 \text{ bits}$$

Efficiency of stop and wait protocol = Frame size/ bandwidth-delay product

0.5 = Frame size /80.

Frame size = 40 bits

Therefore to achieve 50% efficiency in stop and wait protocol frame size should be 40bits

2. Perform bit stuffing on the given bit stream 01101111101111101111111010. Assume flag as 01111110.

Solution:

Given bit stream = 01101111101111101111111010

Stuffed data stream = 011011111**0**0111111**0**1011111**0**11010

3. In a stop-and-wait ARQ system the bandwidth of the line is 1 mbps and 1bit takes 20 ms to make a round trip. What is the bandwidth delay product? If the system data frames are 1000 bits in lengths, what is the percentage utilization of the link?

Solution:

The bandwidth-delay product (capacity) is = B.W * Delay per bit

$$= (1*10^6)*(20*10^{-3}) = 20{,}000 \text{ bits}$$

Data size (frame size) = 1000 bits.

Then link utilization = Data size/ bandwidth-delay product

= 1000 /20,000

Percentage utilization of the link = **5 %**

4. Assume that in a stop-and-wait ARQ system the bandwidth delay product is 10000 bits. If 1 bit takes 20 ms to make round trip. What is the bandwidth of the line? Find the utilization of the link If the system send 8 data frames of 1000 bits length are send before stopping and worrying about the acknowledgment.

Solution:

Given bandwidth*Delay product = 10000 bits

Bandwidth*20ms = 10000 bits

Bandwidth = $10000 /20*10^{-3}$

= $10000 /20*10^{-3}$

= 500Khz or 0.5MHz

Percentage utilization of single frame = 1000/10,000

= 10%

Percentage utilization of 10 frames = 8*1000/10,000

= 8*1000/10,000

Percentage utilization of the link = **80 %**

UNIT 2- DATA LINK LAYER

UNIVERSITY QUESTIONS AND ANSWER

1. a). Explain the selective repeat sliding window protocol with necessary figures.
(**Dec 09/ Jan 10 – 10marks –Answer refer page no. 19)**

b). A channel has a bit rate of 4 kbps and a propagation delay of 20 msec. For what range of a frame size does stop and wait protocol give an efficiency of at least 50%.
(**Dec 09/ Jan 10 – 06marks –Answer refer page no. 35)**

c). Perform bit stuffing on the given bit stream 01101111101111101111111010. Assume flag as 01111110. (**Dec 09/ Jan 10 – 04 marks –Answer refer page no. 35)**

2. a). What is framing? How frames can be classified? Explain bit stuffing with the help of an example. (**May/ June 10 – 06marks –Ans refer page no.2 to 4)**

b). What is the meaning of datalink control? Explain stop-and-wait ARQ, using a suitable block diagram. (**May/ June 10 – 10marks –Ans refer page no.10)**

c). In a stop-and-wait ARQ system the bandwidth of the line is 1 mbps and 1bit takes 20 ms to make a round trip. What is the bandwidth delay product? If the system data frames are 1000 bits in lengths, what is the percentage utilization of the link?
(**May/ June 10 – 04marks –Ans refer page no.35)**

3 a. Explain the stop-and -wait protocol, for noisy channels. (**December 10 – 10marks –Answer refer page no.10)**

b. What are the three types of frames in HDLC protocol? Explain each of them briefly.
(**December 10 – 10marks –Answer refer page no.26)**

4. a. Explain byte stuffing and unstuffing and bit stuffing and unstuffing, with necessary diagrams. (**June/July 2011 – 10marks –Answer refer page no.3)**

b. With a neat diagram, explain three different types of HDLC frames. (**June/July 2011 – 10marks –Answer refer page no.26)**

5. a) In stop and wait ARQ systems, the bandwidth of the line is 1Mbps and it takes 20ms to make round trip. What is the bandwidth delay product? If the system data frames are of 1000bit length. What is the percentage of link? What is the channel utilization percentage of link if the protocol that can send up to 15 frames before stopping and worrying about the acknowledgement? Write the comment. (**December 2011 – 05marks –Answer refer page no.35)**

b). Explain briefly the bit and character stuffing. (**Dec 2011 – 05marks –Ans refer page no.2).**

c). With a neat diagram, explain the HDLC frame form. (**Dec 2011 – 05marks –Ans refer pg 26).**

6.a. Differentiate between character stuffing and bit stuffing with examples.

(December 2012 – 05 marks –Answer refer page no.2 and 5)

b. Explain different HDLC frames. **December 2012 – 05 marks –Answer refer page no.26)**

c. What are sliding windows protocols? Explain Go- Back-N protocol for noisy channel.

(December 2012 – 10 marks –Answer refer page no.13)

7. (a). Explain the mechanism of selective repeat ARQ with diagram showing send window and receive window. .(**June/July 2013 – 10marks –Ans refer page no. 19).**

b. With suitable block diagram, explain the stop and wait protocol, for noise less channels. Also write the sender site logarithm. .(**June/July 2013 – 06 marks –Ans refer page no. 7)**

c. Perform bit stuffing and unstuffing on the given bit stream: 0001111111001111101000. Assume flag as 0111110. .(**June/July 2013 – 04marks –Ans refer page no. 35)**
