BENNETT UNIVERSITY
TIMES OF INDIA GROUP

Data Link Layer

Flow Control

FLOW CONTROL

- The most important responsibilities of the data link layer are flow control and error control. Collectively, these functions are known as data link control.

- Flow control refers to a set of procedures used to restrict the amount of data that the sender can send before waiting for acknowledgment.

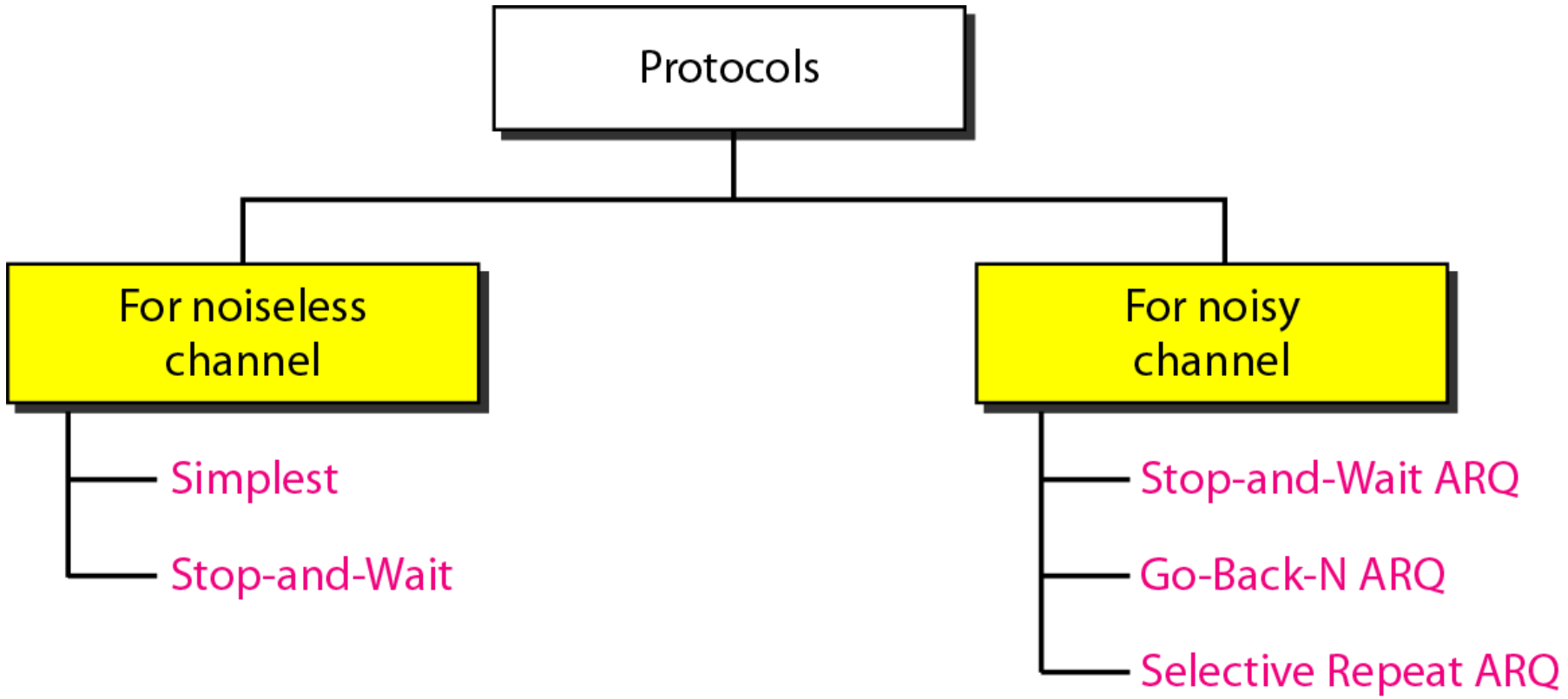
Flow Control Protocols

BENNETT UNIVERSITY
TIMES OF INDIA GROUP

- How the 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.

Types of protocols



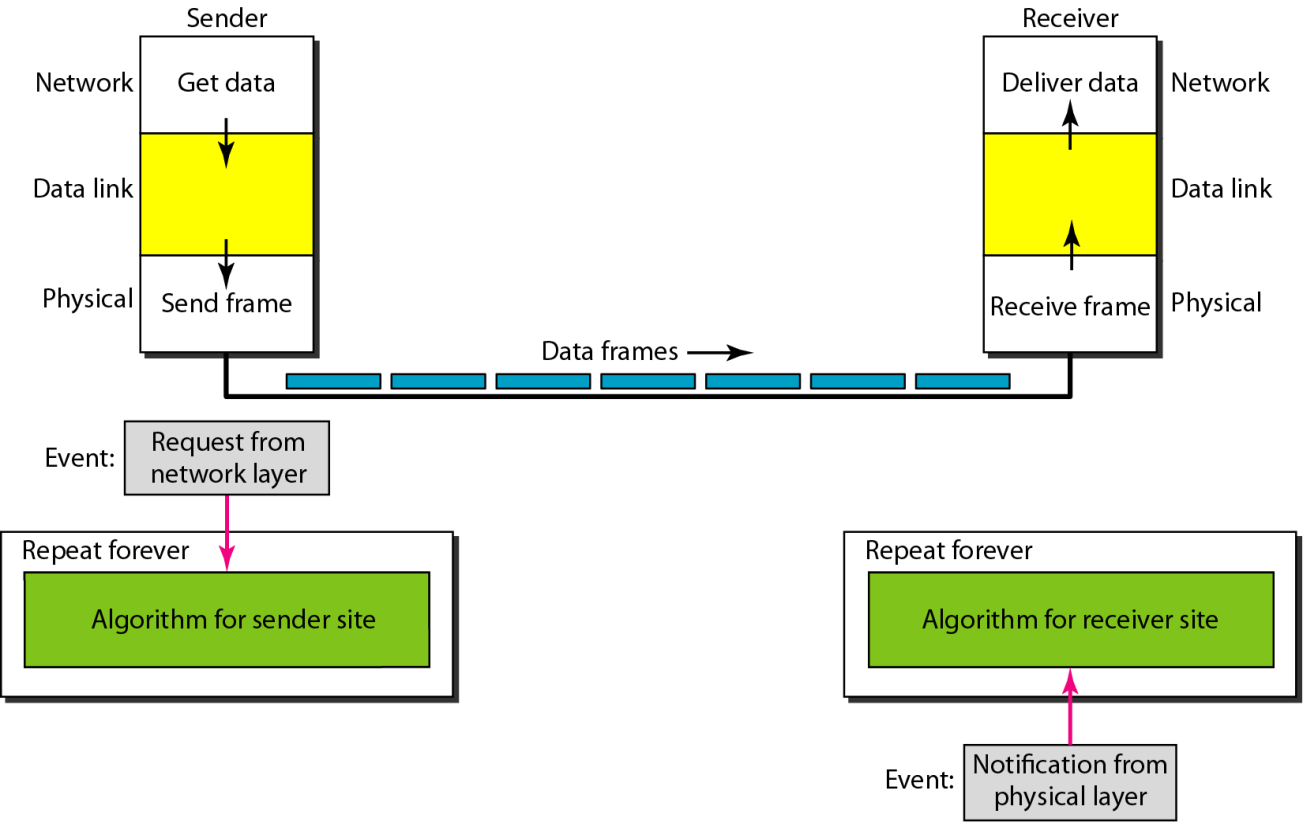
NOISELESS CHANNELS

Let us first assume we have an ideal channel in which no frames are lost, duplicated, or corrupted.

Two protocols for this type of channel.

- Simplest Protocol
- Stop-and-Wait Protocol

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



Sender-site algorithm for the simplest protocol

```
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 }
```

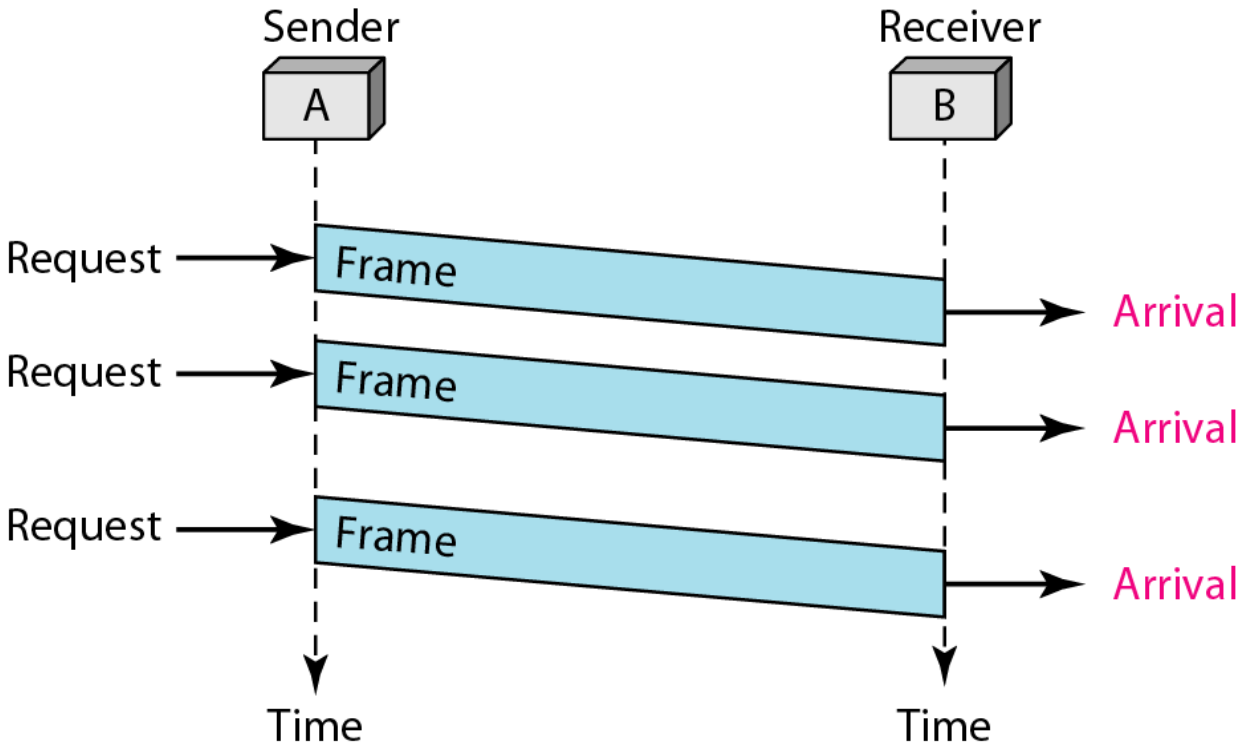
Receiver-site algorithm for the simplest protocol

```
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 }
```

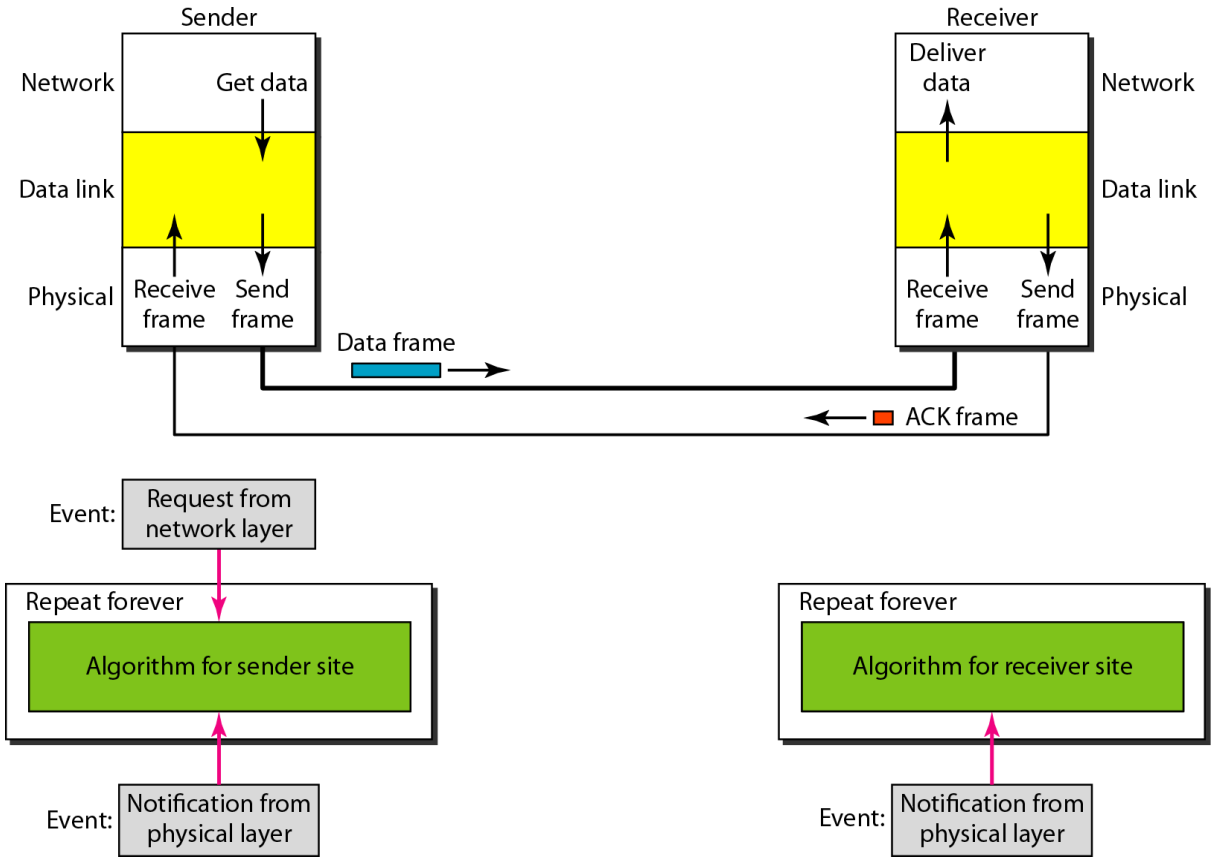
Simplest protocol :Flow diagram

BENNETT UNIVERSITY
TIMES OF INDIA GROUP

- No flow or error control



Design of Stop-and-Wait Protocol



Sender-site algorithm for the stop and wait

```
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 }
```

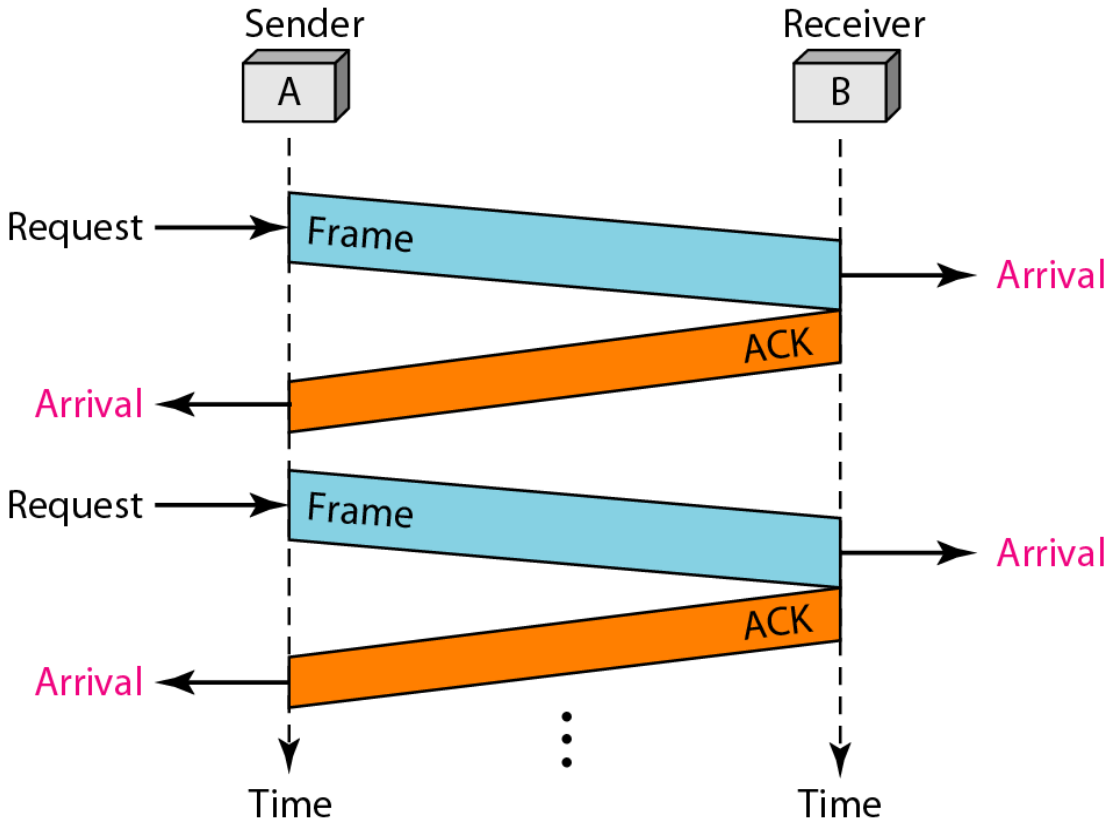
```
1  while(true)                          //Repeat forever
2  {
3    WaitForEvent();                    // Sleep until an event occurs
4    if(Event(ArrivalNotification))     //Data frame arrives
5    {
6       ReceiveFrame();
7       ExtractData();
8       Deliver(data);                  //Deliver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

Receiver-site algorithm for the stop and wait

Stop-and-Wait Protocol :Flow diagram

BENNETT UNIVERSITY
TIMES OF INDIA GROUP

- The sender sends one frame and waits for feedback from the receiver.
- When the ACK arrives, the sender sends the next frame.
- Sending two frames in the protocol involves the sender in four events and the receiver in two events.



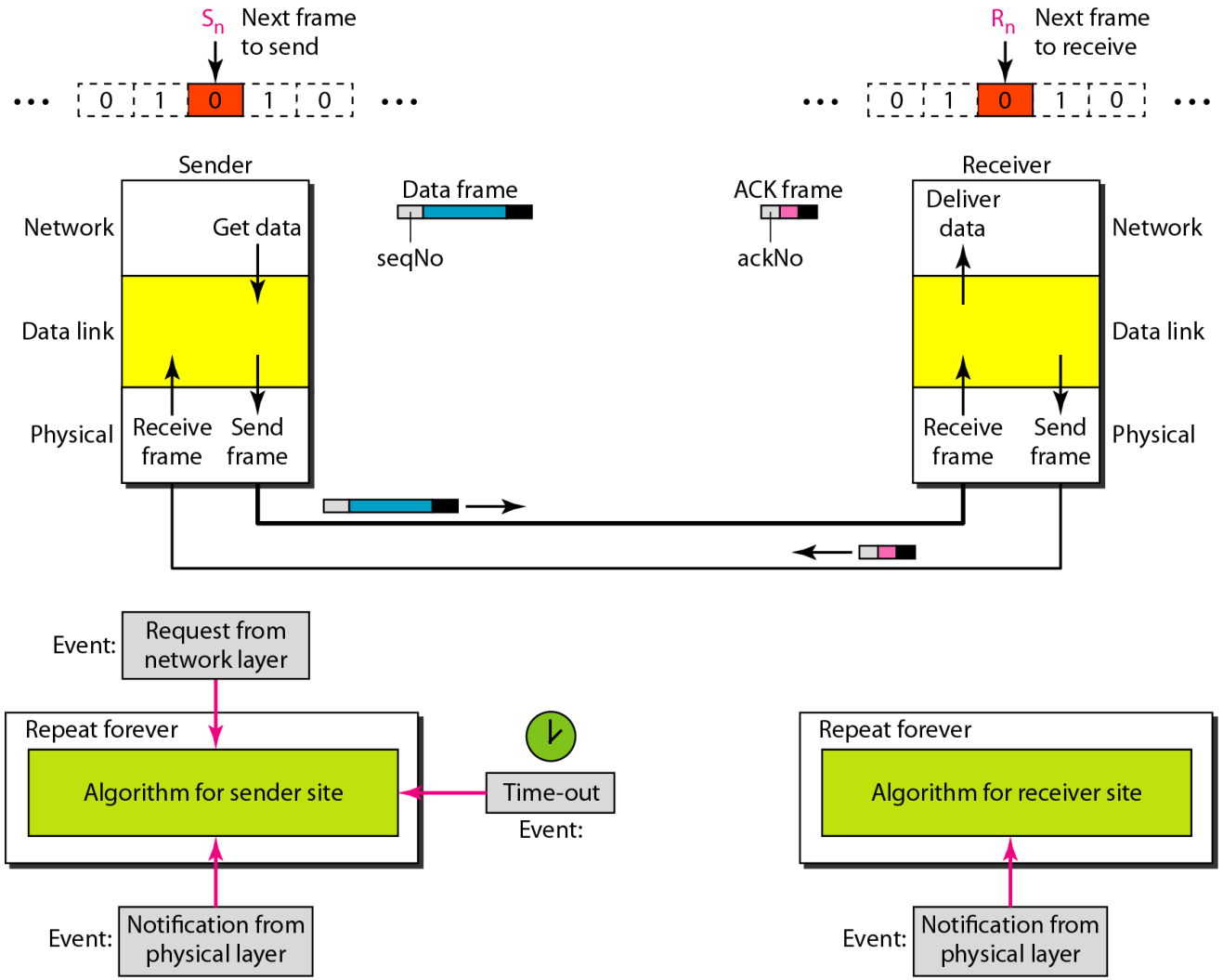
NOISY CHANNELS

- Noiseless channels are nonexistent.
- *Three protocols that use error control.*
  - Stop-and-Wait Automatic Repeat Request
  - Go-Back-N Automatic Repeat Request
  - Selective Repeat Automatic Repeat Request

Stop-and-Wait Automatic Repeat Request

- 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.
- In Stop-and-Wait ARQ, sequence numbers to number the frames is used. The sequence numbers are based on modulo-2 arithmetic.
- In Stop-and-Wait ARQ, the acknowledgment number always announces in modulo-2 arithmetic the sequence number of the next frame expected.

Design of the Stop-and-Wait ARQ Protocol

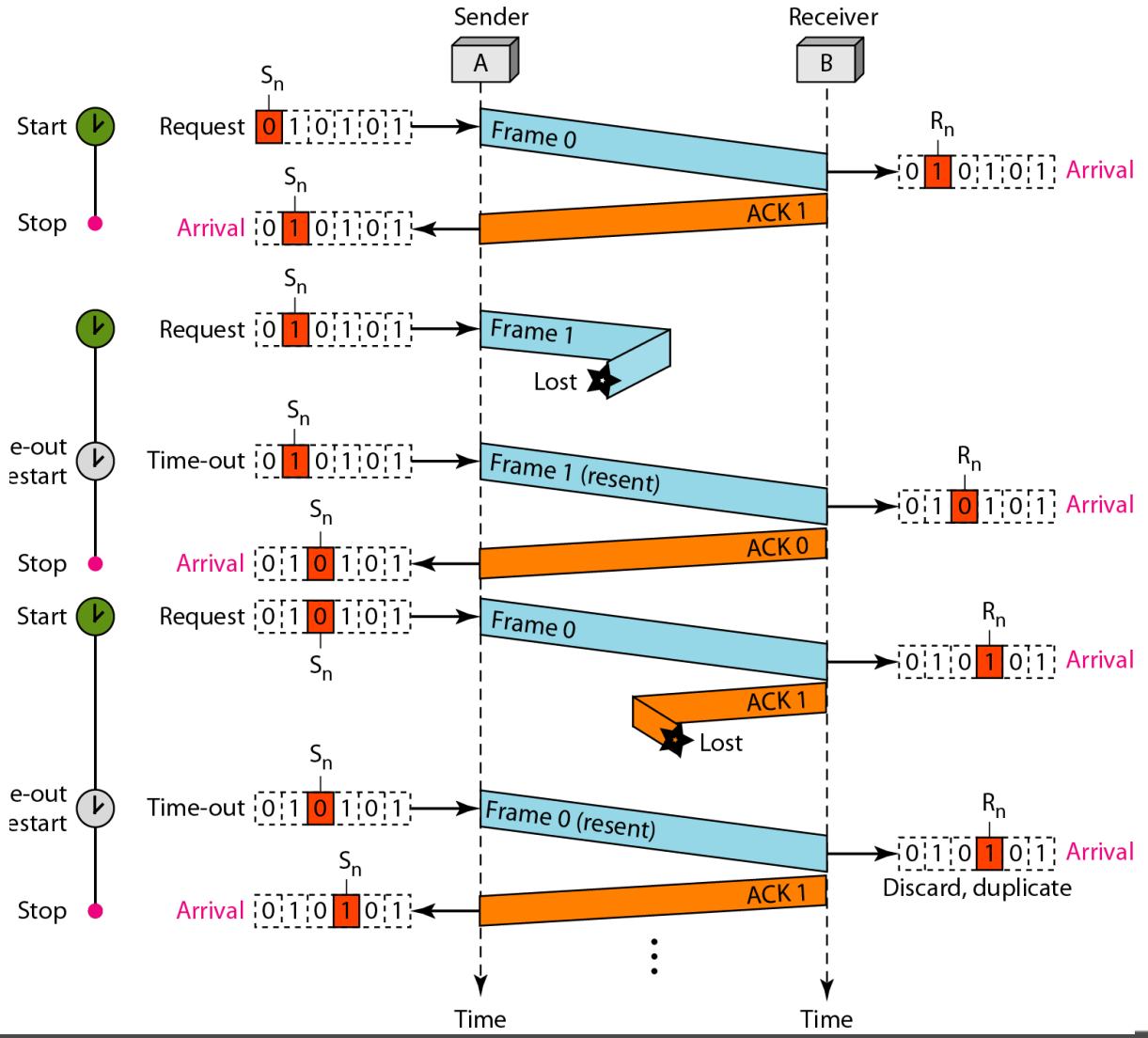


Flow diagram

BENNETT UNIVERSITY
TIMES OF INDIA GROUP

An example of Stop-and-Wait ARQ.

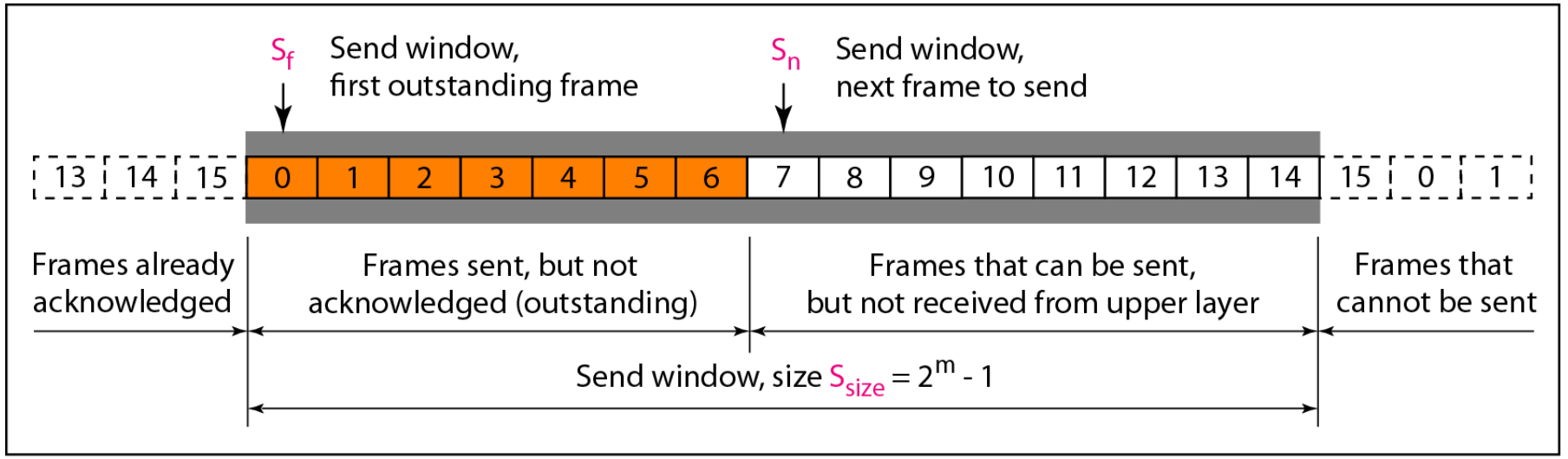
- Frame 0 is sent and acknowledged.
- Frame 1 is lost and resent after the time-out.
- The resent frame 1 is acknowledged and the timer stops.
- Frame 0 is sent and acknowledged, but the acknowledgment is lost.
- The sender has no idea if the frame or the acknowledgment is lost, so after the time-out,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



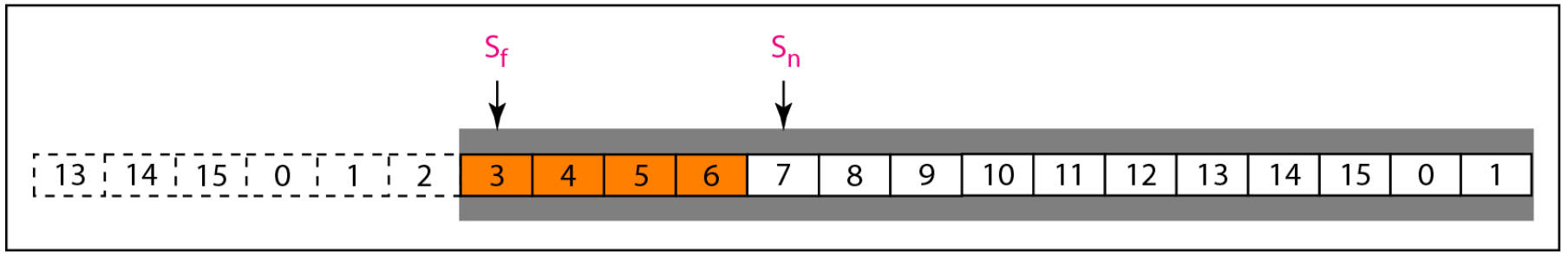
Send window for Go-Back-N ARQ

BENNETT UNIVERSITY
TIMES OF INDIA GROUP

- In the Go-Back-N Protocol, the sequence numbers are modulo 2^m, where m is the size of the **sequence number** field in bits.
- The send window is an abstract concept defining an imaginary box of size **2^m – 1** with three variables: Sf, Sn, and S_{size}.
- The send window can slide one or more slots when a valid acknowledgment arrives.



a. Send window before sliding

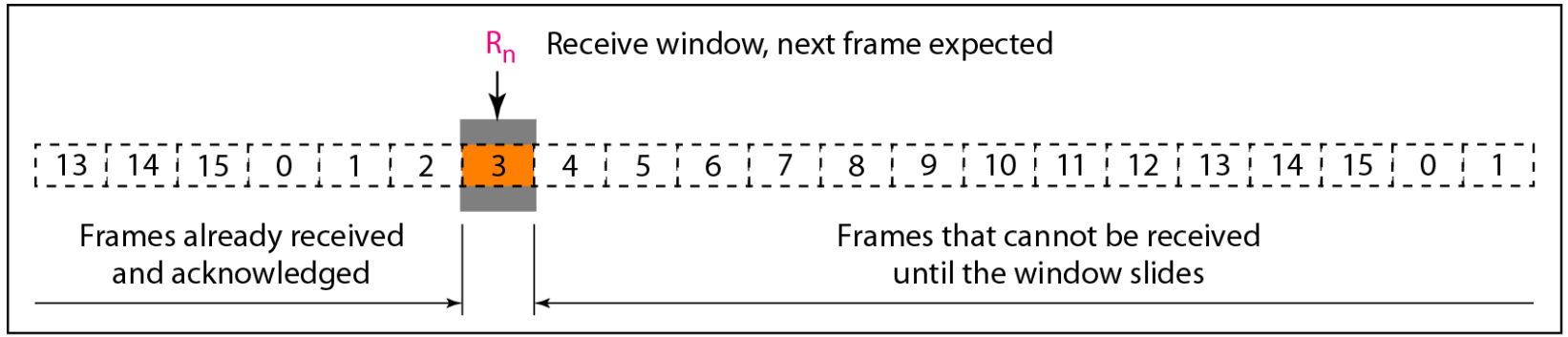


b. Send window after sliding

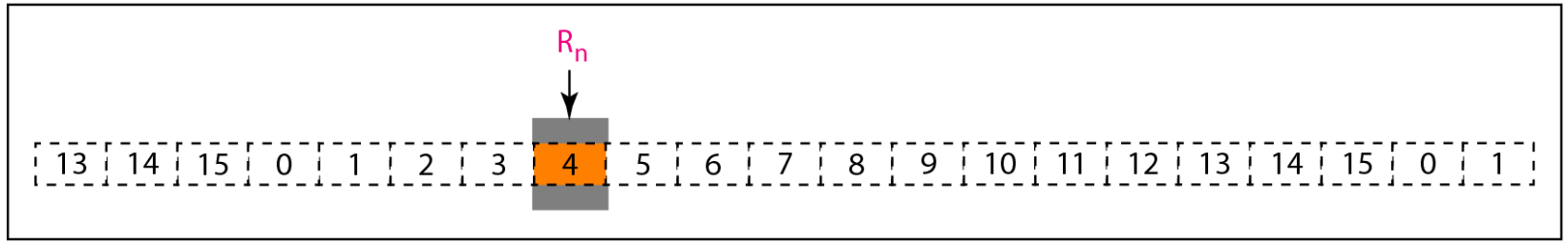
Receive window for Go-Back-N ARQ

BENNETT UNIVERSITY
TIMES OF INDIA GROUP

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

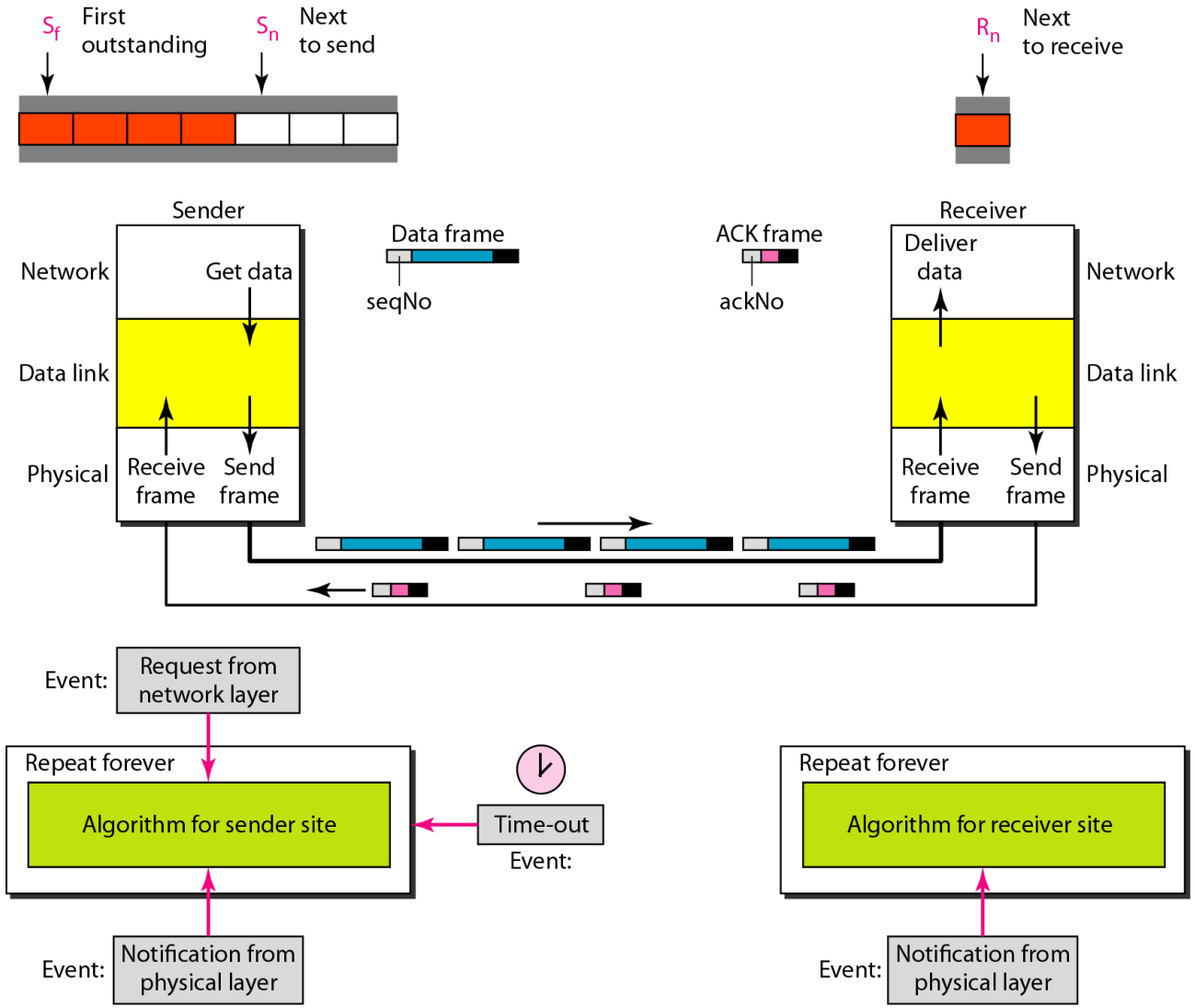


a. Receive window



b. Window after sliding

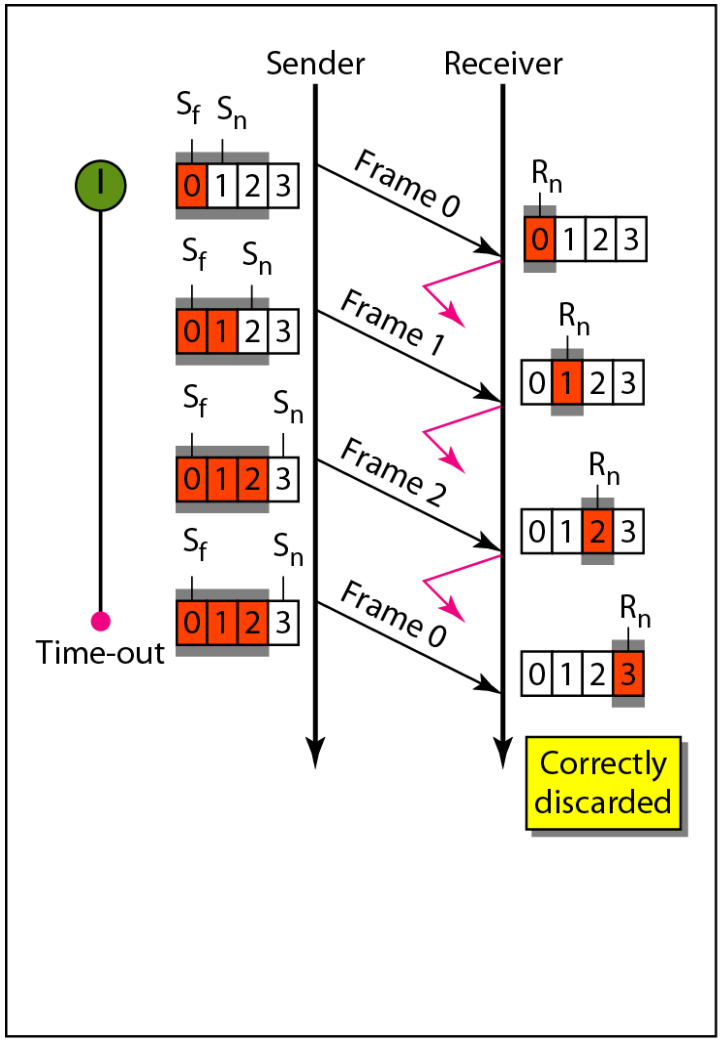
Design of Go-Back-N ARQ



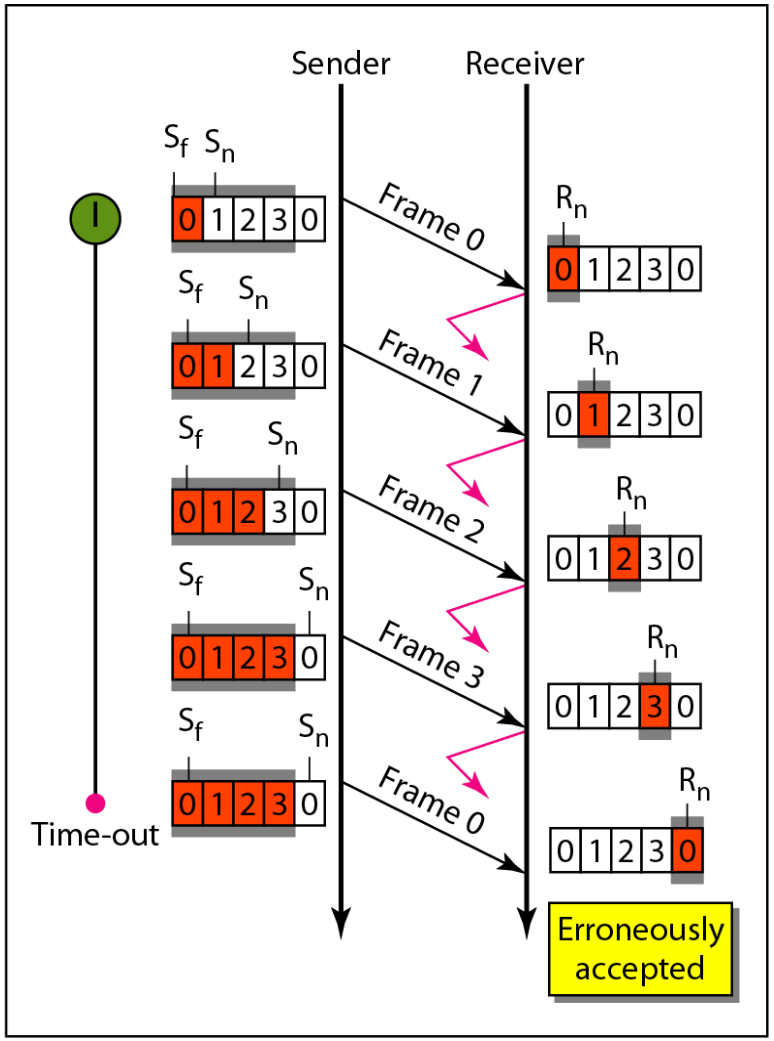
Window size for Go-Back-N ARQ

BENNETT UNIVERSITY
TIMES OF INDIA GROUP

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.

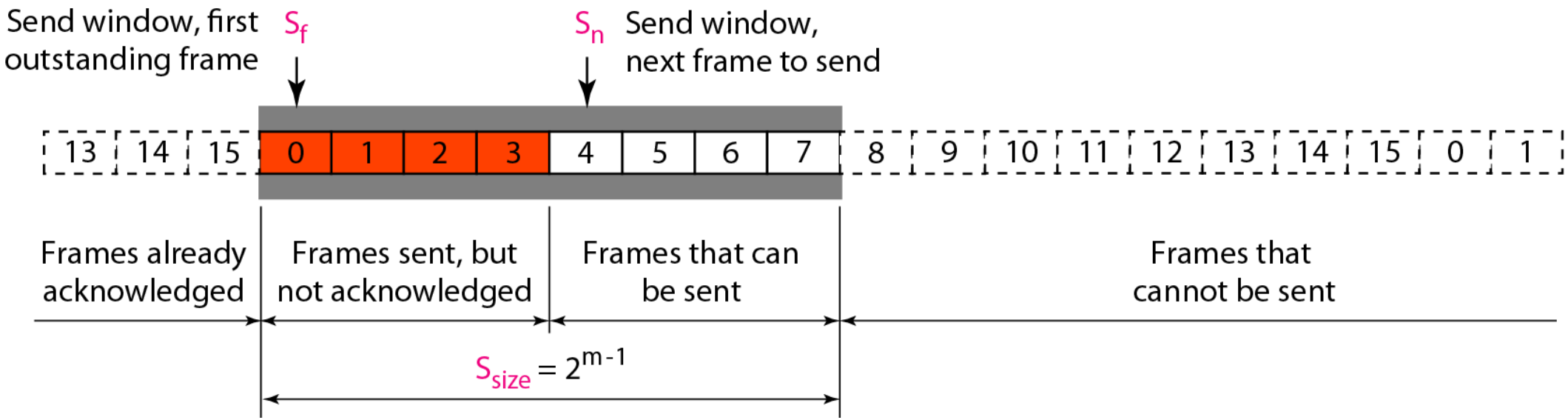


a. Window size < 2^m

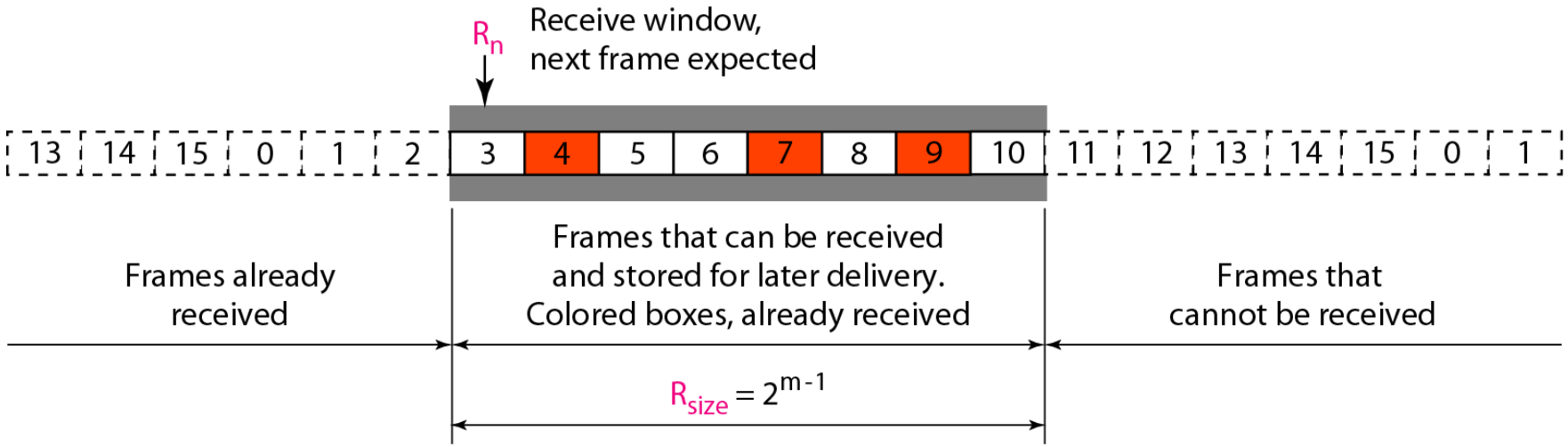


b. Window size = 2^m

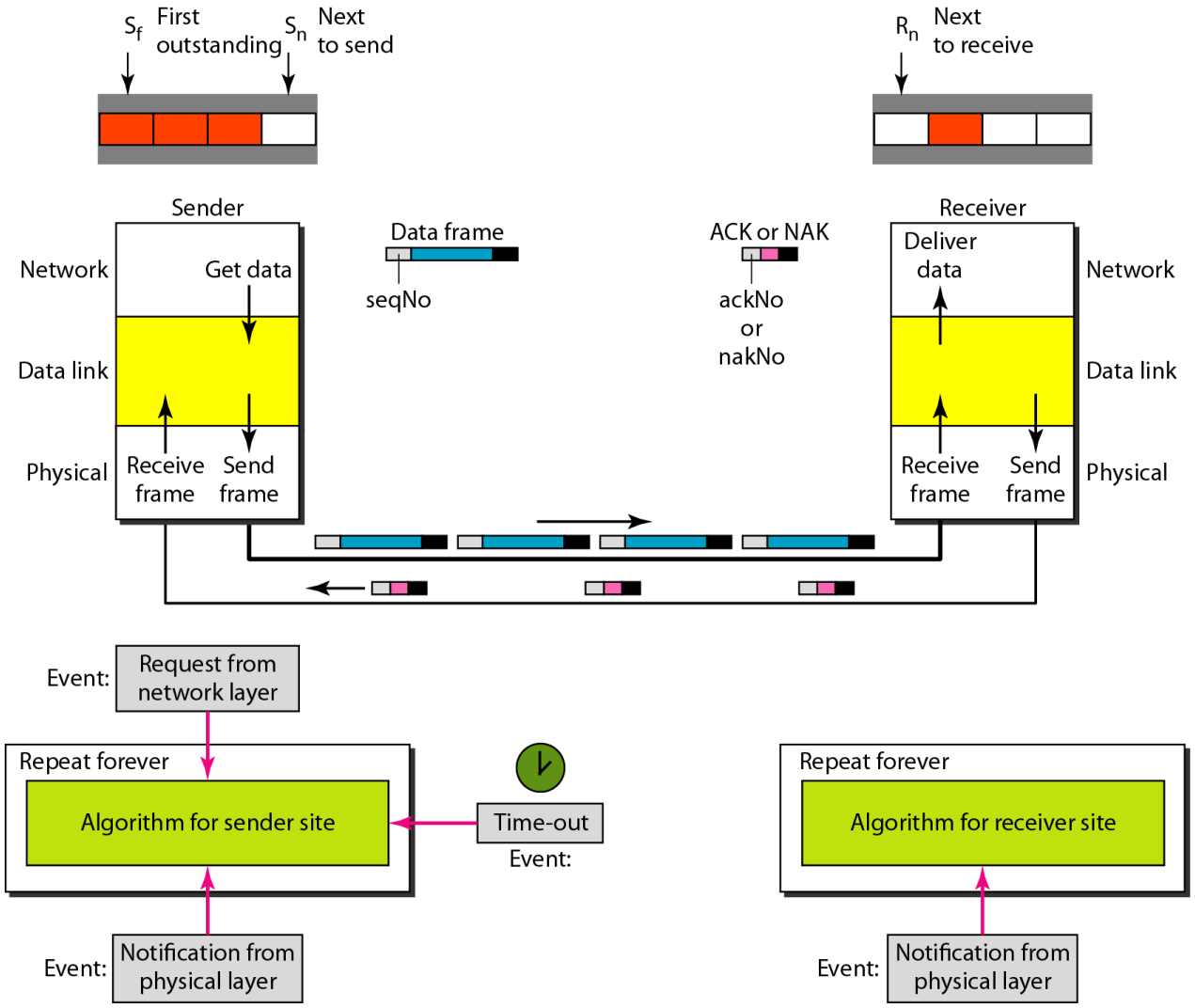
Send window for Selective Repeat ARQ



Receive window for Selective Repeat ARQ



Design of Selective Repeat ARQ

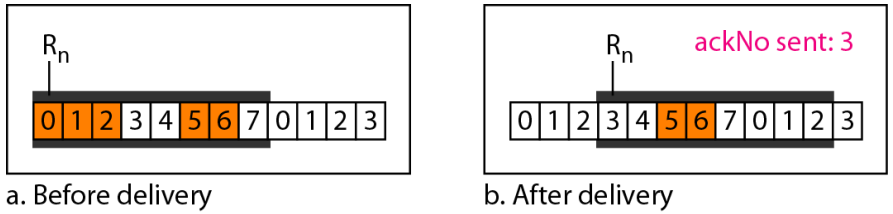


Selective Repeat ARQ, window size

BENNETT UNIVERSITY
TIMES OF INDIA GROUP

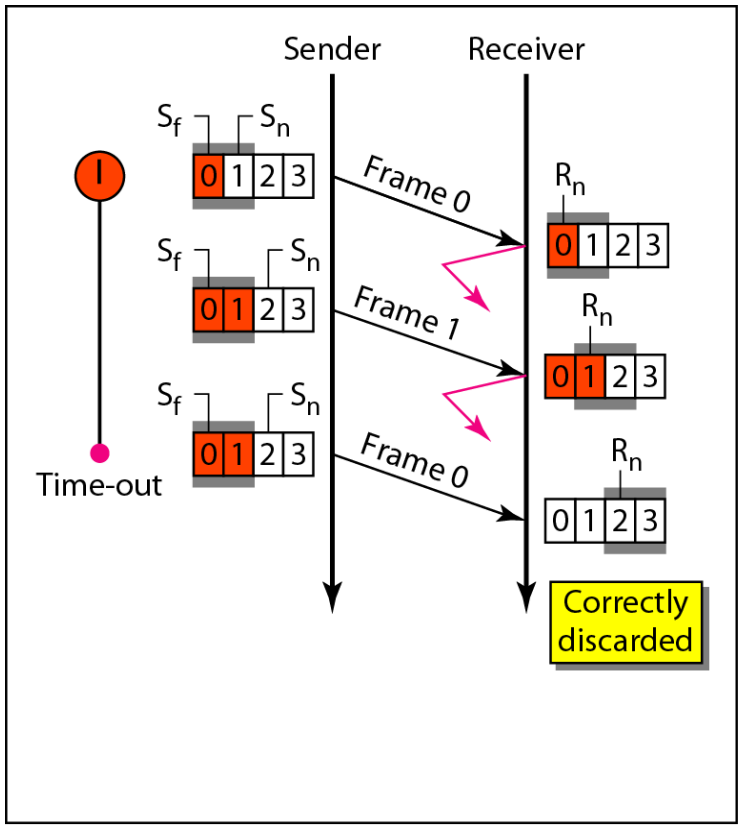
In Selective Repeat ARQ, the size of the sender and receiver window must be at most one-half of 2^m.

Delivery of data in Selective Repeat ARQ

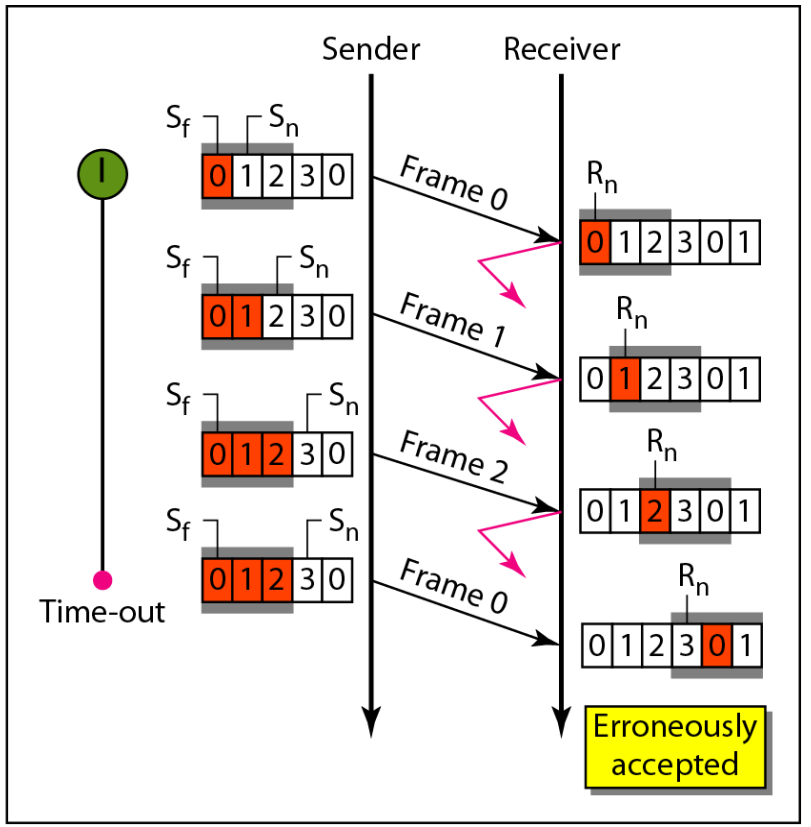


a. Before delivery

b. After delivery



a. Window size = 2^{m-1}



b. Window size > 2^{m-1}

Design of piggybacking in Go-Back-N ARQ

BENNETT UNIVERSITY
TIMES OF INDIA GROUP

- In real life, data frames are normally flowing in both directions: from node A to node B and from node B to node A.
- The control information also needs to flow in both directions. A technique called **piggybacking** is used to improve the efficiency of the bidirectional protocols
- Each node now has two windows: one send window and one receive window. Both also need to use a timer.
- Both are involved in three types of events: request, arrival, and time-out.