Chapter 11

Data Link Control

- Two main functions of the data link layer are data link control and media access control
- Data link control functions
  - Framing
  - Flow and error control
  - Software implemented protocols for smooth and reliable transmission of frames between nodes

Framing

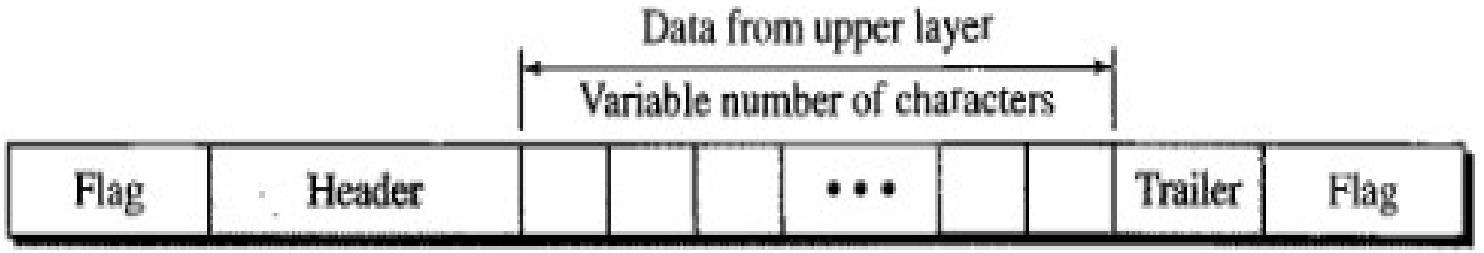
- Separates a message from one source to destination, or from other messages to other destinations by adding a sender address and a destination address
- Destination address tells where the frame is to go
- Sender address helps the recipient acknowledge the receipt
- A very large frame makes flow and error control very inefficient – a single bit error requires retransmission

Framing

- Fixed-size framing – no need to define boundaries of the frames; ex. ATM networks
- Variable-size framing – need to define the end of the frame and beginning of the next; used in most networks
  - Character-oriented
  - Bit-oriented

Character-Oriented Protocols

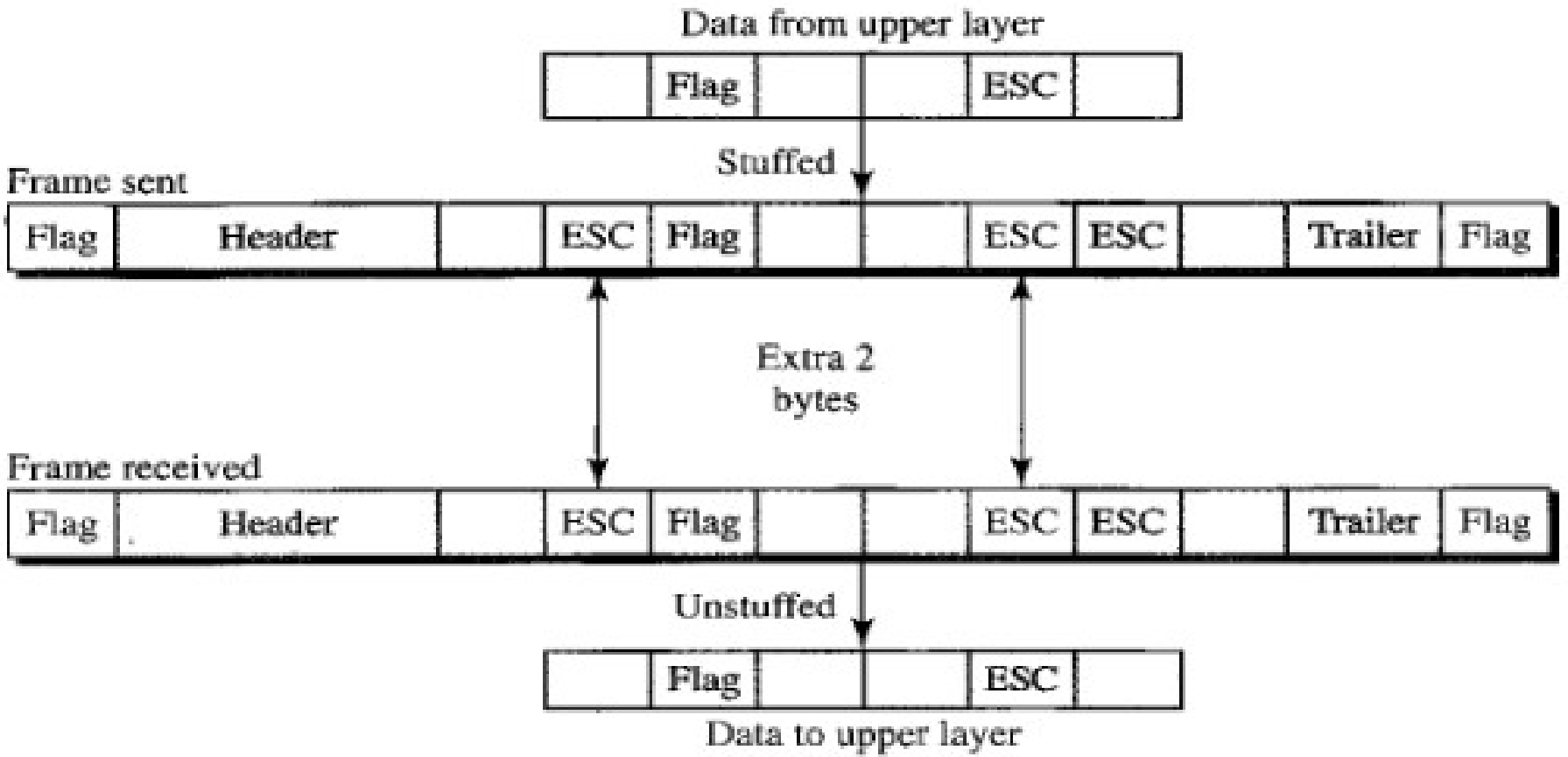
- Data to be carried are 8-bit characters from a coding system such as ASCII
- Header and trailer are multiples of 8 bits
- 8-bit (1 byte) flag is added at the beginning and end of a frame



Character-Oriented Protocols

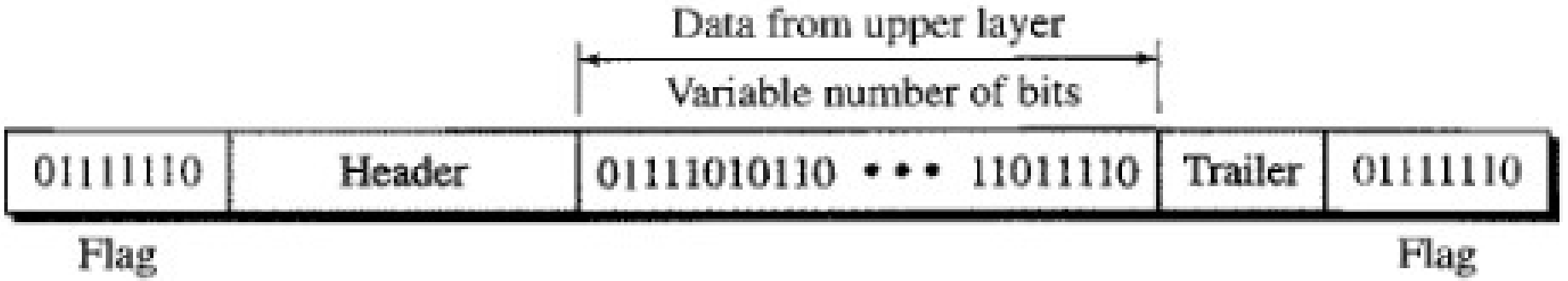
- Byte-Stuffing - Prevents data from being interpreted as flag
- A special byte, called escape character (ESC), is added to the data section of the frame when there is a character with the same pattern as the flag

Byte Stuffing



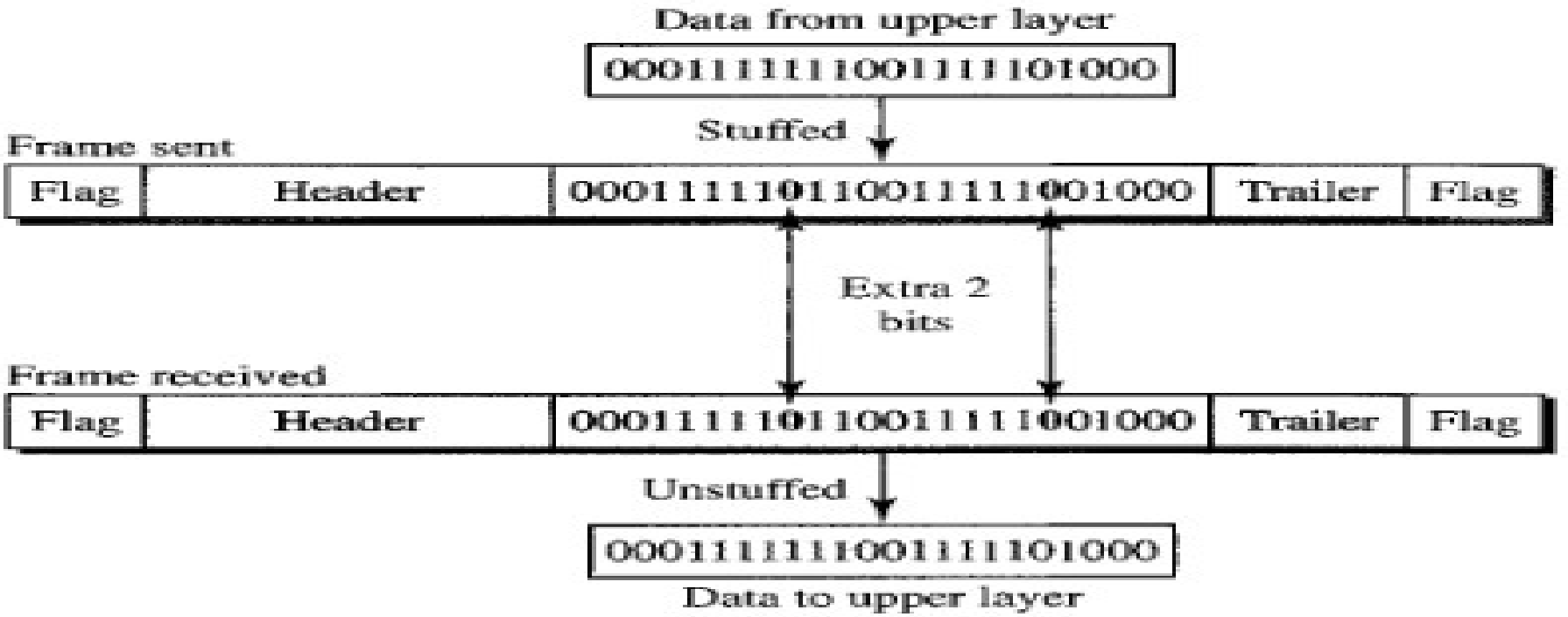
Bit-Oriented Protocols

- Data section of frame is a sequence of bits to be interpreted by the upper layer as text, graphic, audio, video, etc.
- Delimiter is a special bit pattern: 01111110



Bit Stuffing

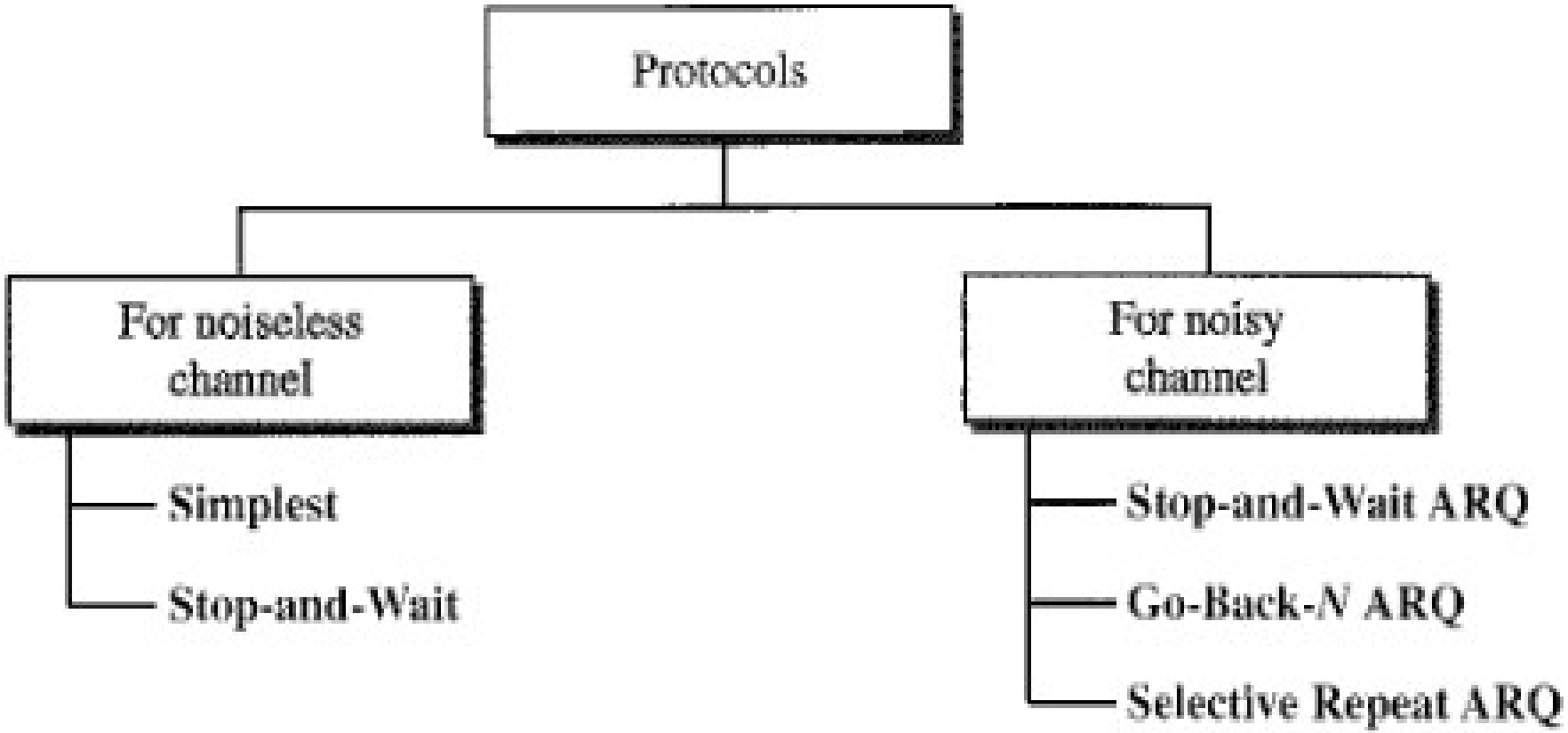
- Adds one extra 0 whenever five consecutive 1s follow a 0 in the data



Flow and Error Control

- Flow control coordinates the amount of data that can be sent before receiving an acknowledgment
- Error control is both error detection and error correction
- Automatic Repeat Request (ARQ) - any time an error is detected in an exchange, specified frames are retransmitted

Protocols

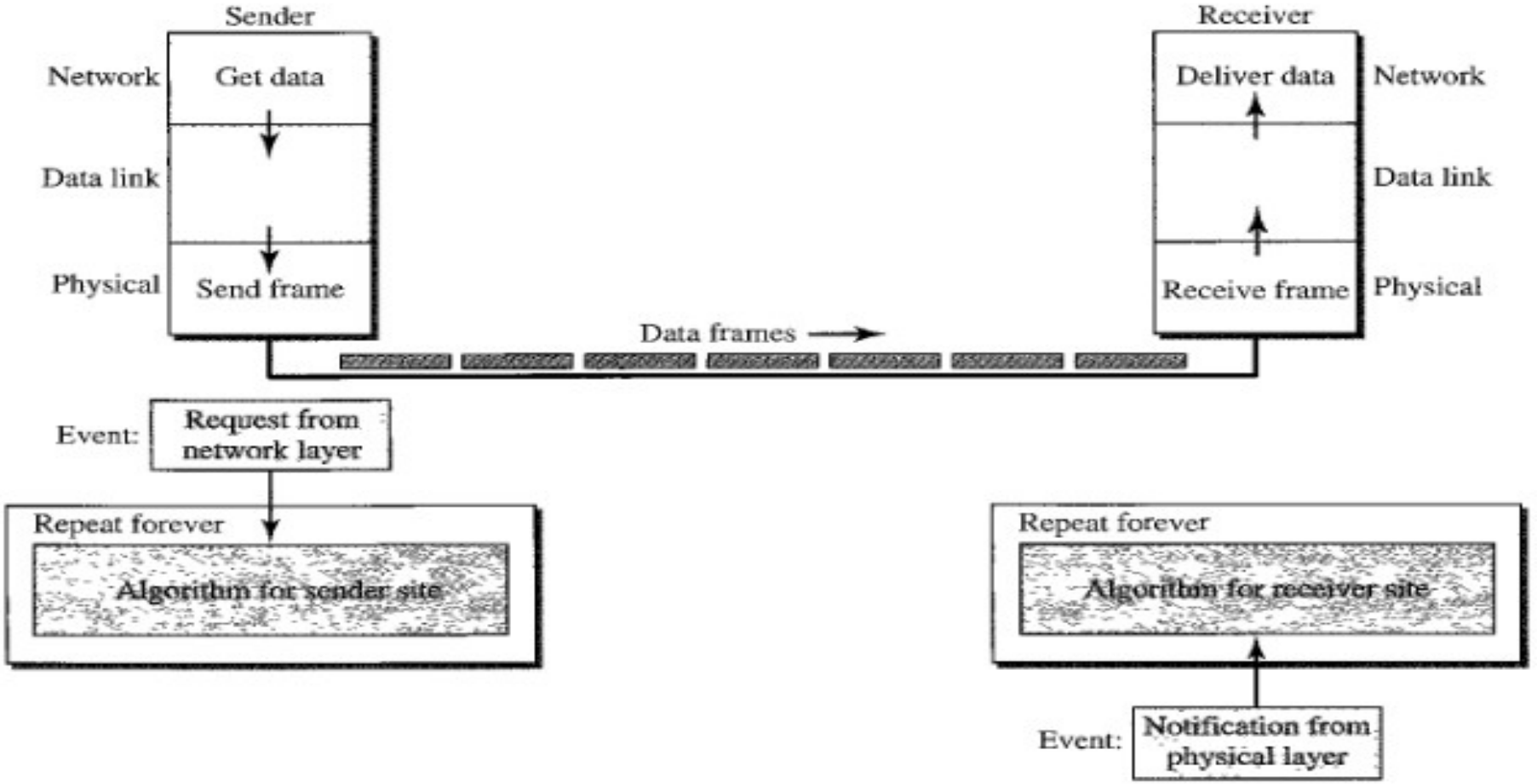

Protocols
For noiseless channel
For noisy channel
Simplest
Stop-and-Wait
Stop-and-Wait ARQ
Go-Back-N ARQ
Selective Repeat ARQ

Protocols

- Focus is on unidirectional transfer
- Specialized frames
  - ACK – acknowledgment
  - NAK – negative acknowledgment
- Piggybacking – NAKs and ACKs included in the data frame
- Implemented in software

Simplest Protocol

- No flow or error control



Simplest Protocol

Algorithm 11.1 *Sender-site algorithm for the simplest protocol*

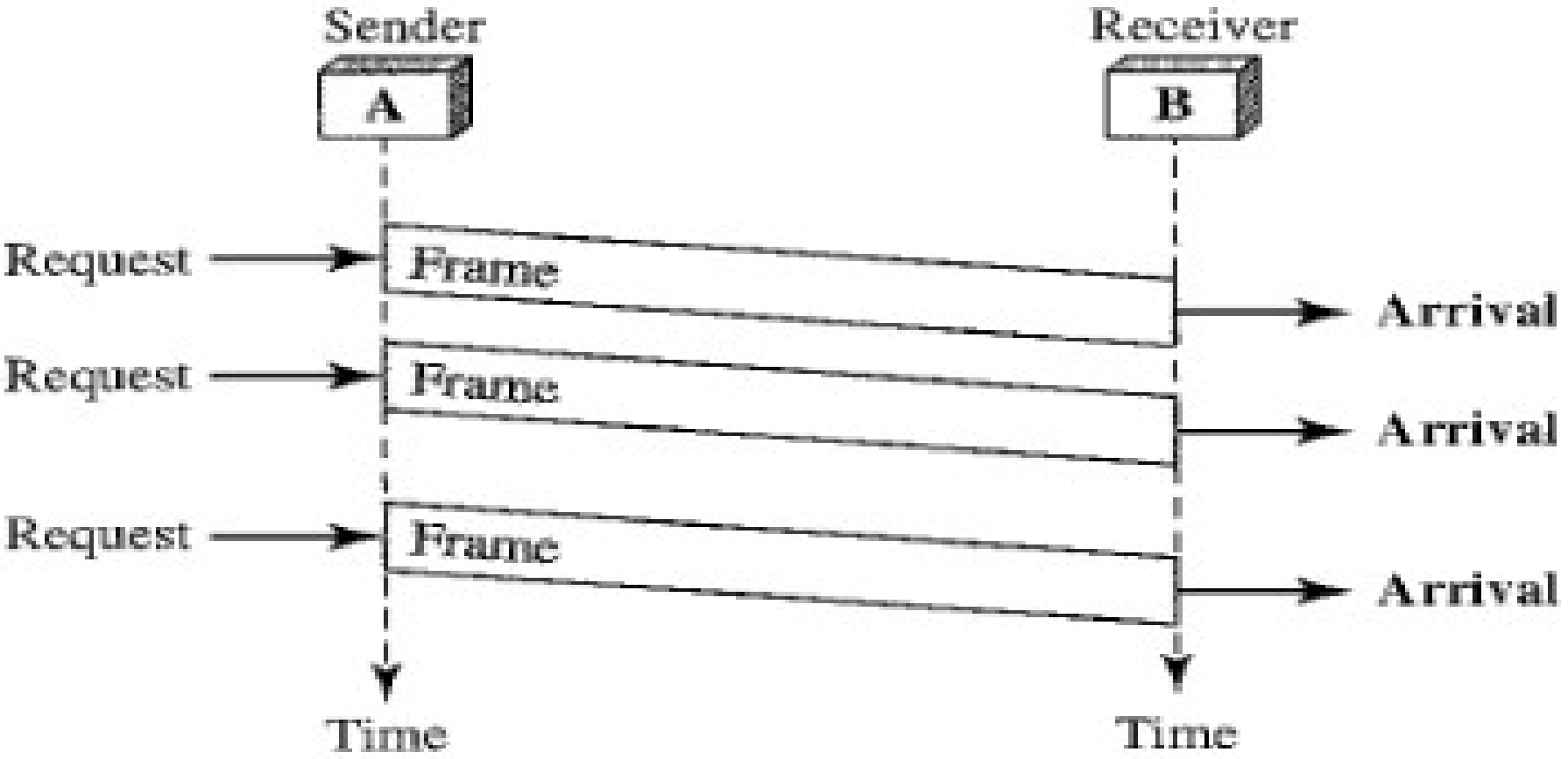
```
1  while(true)                          // Repeat forever
2  {
3    WaitForEvent();                    // Sleep until an event occurs
4    if(Event(RequestToSend))           //There is a packet to send
5    {
6       GetData();
7       MakeFrame();
8       SendFrame();                    //Send the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

Simplest Protocol

Algorithm 11.2 *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 }
```

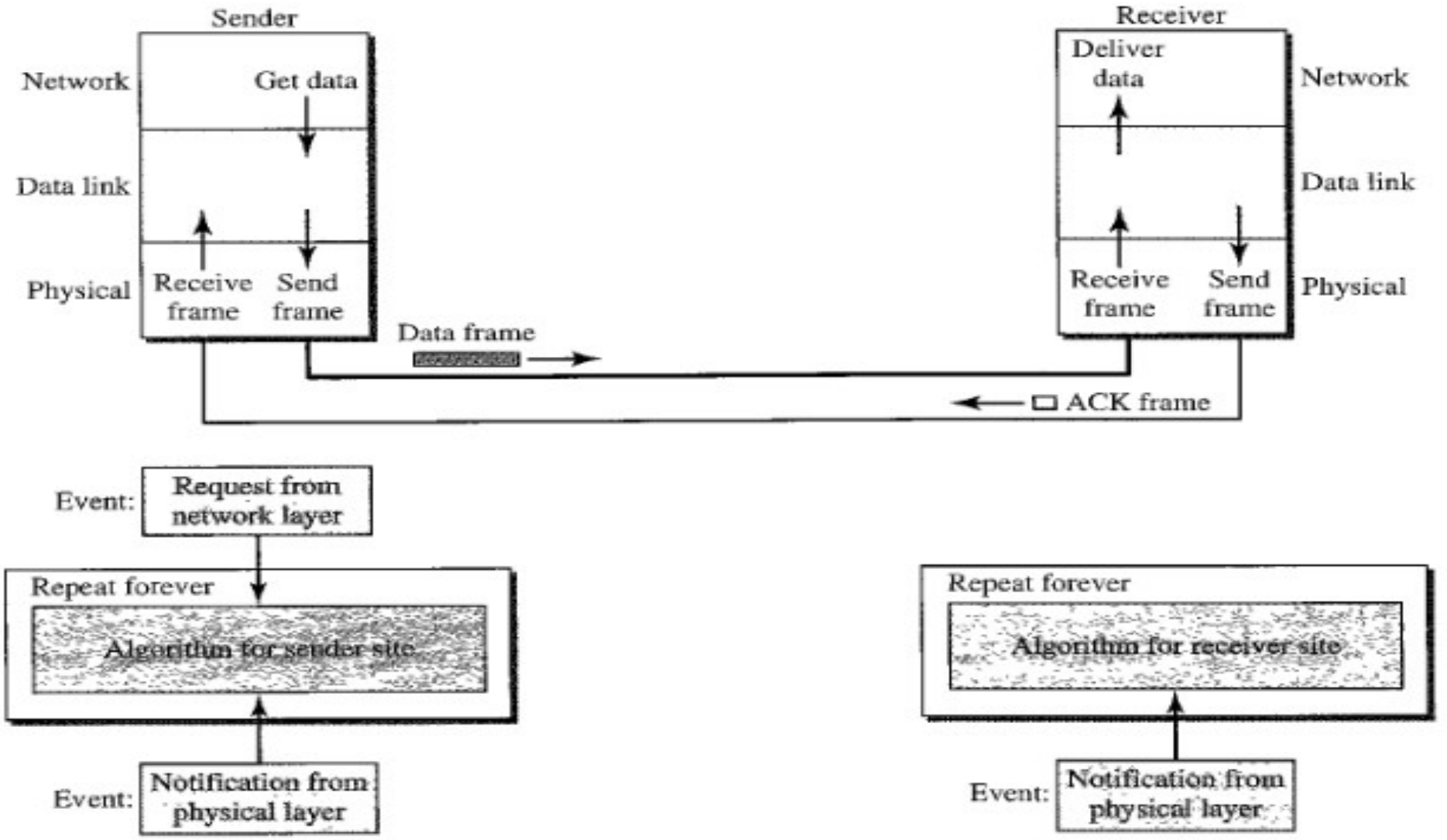
Simplest Protocol



Stop-and-Wait Protocol

- If the data frames arrive at the receiver site faster than they can be processed, the frames must be stored until their use
- Sender sends one frame, until it receives confirmation from the receiver and then sends next frame

Stop-and-Wait Protocol



Stop-and-Wait Protocol

Algorithm 11.3 *Sender-site algorithm for Stop-and-Wait Protocol*

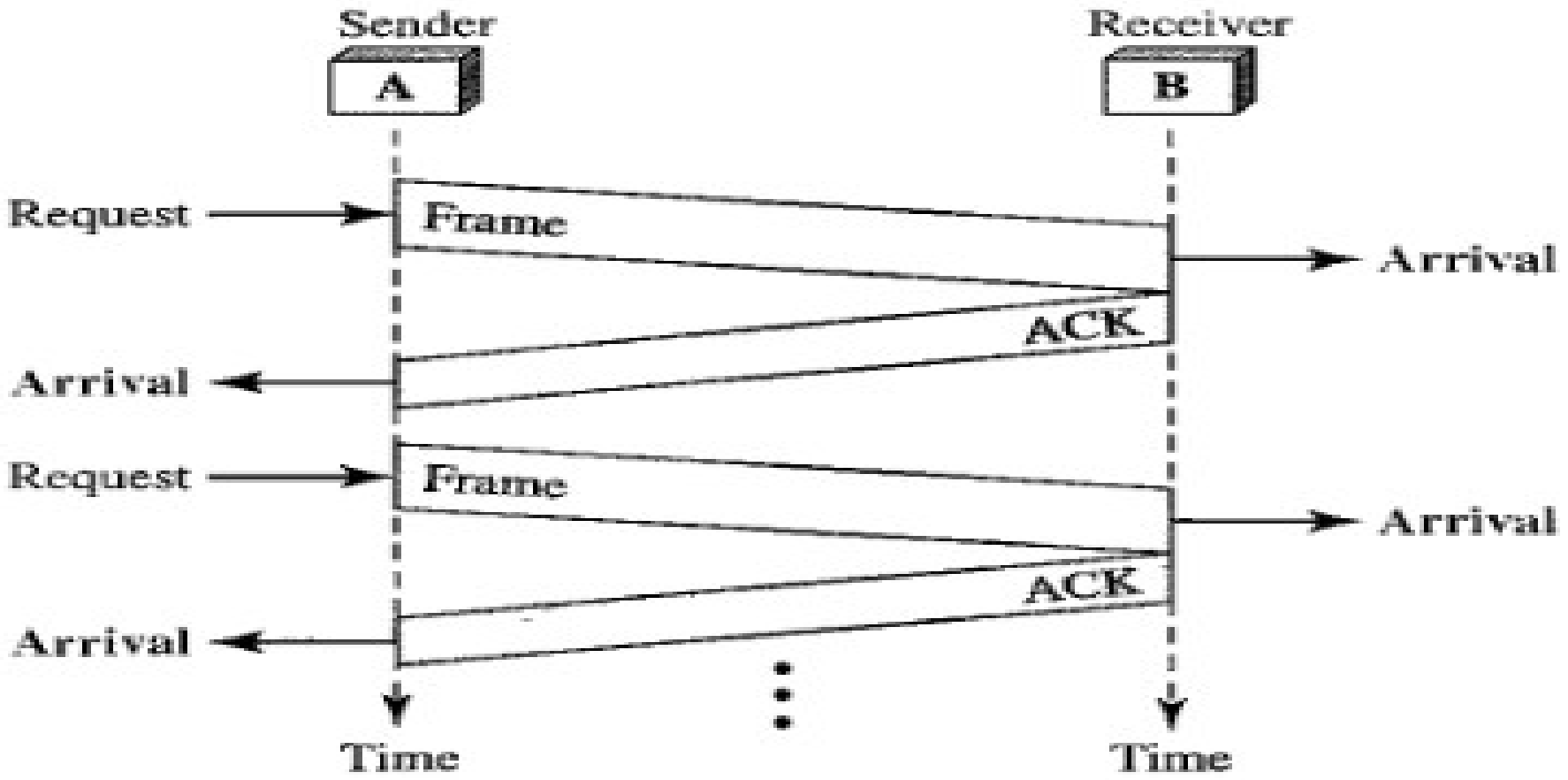
```
1  while(true)                          //Repeat forever
2  canSend = true                       //Allow the first frame to go
3  {
4    WaitForEvent();                    // Sleep until an event occurs
5    if(Event(RequestToSend) AND canSend)
6    {
7       GetData();
8       MakeFrame();
9       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 }
```

Stop-and-Wait Protocol

Algorithm 11.4 *Receiver-site algorithm for 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 }
```

Stop-and-Wait Protocol



Stop-and-Wait ARQ

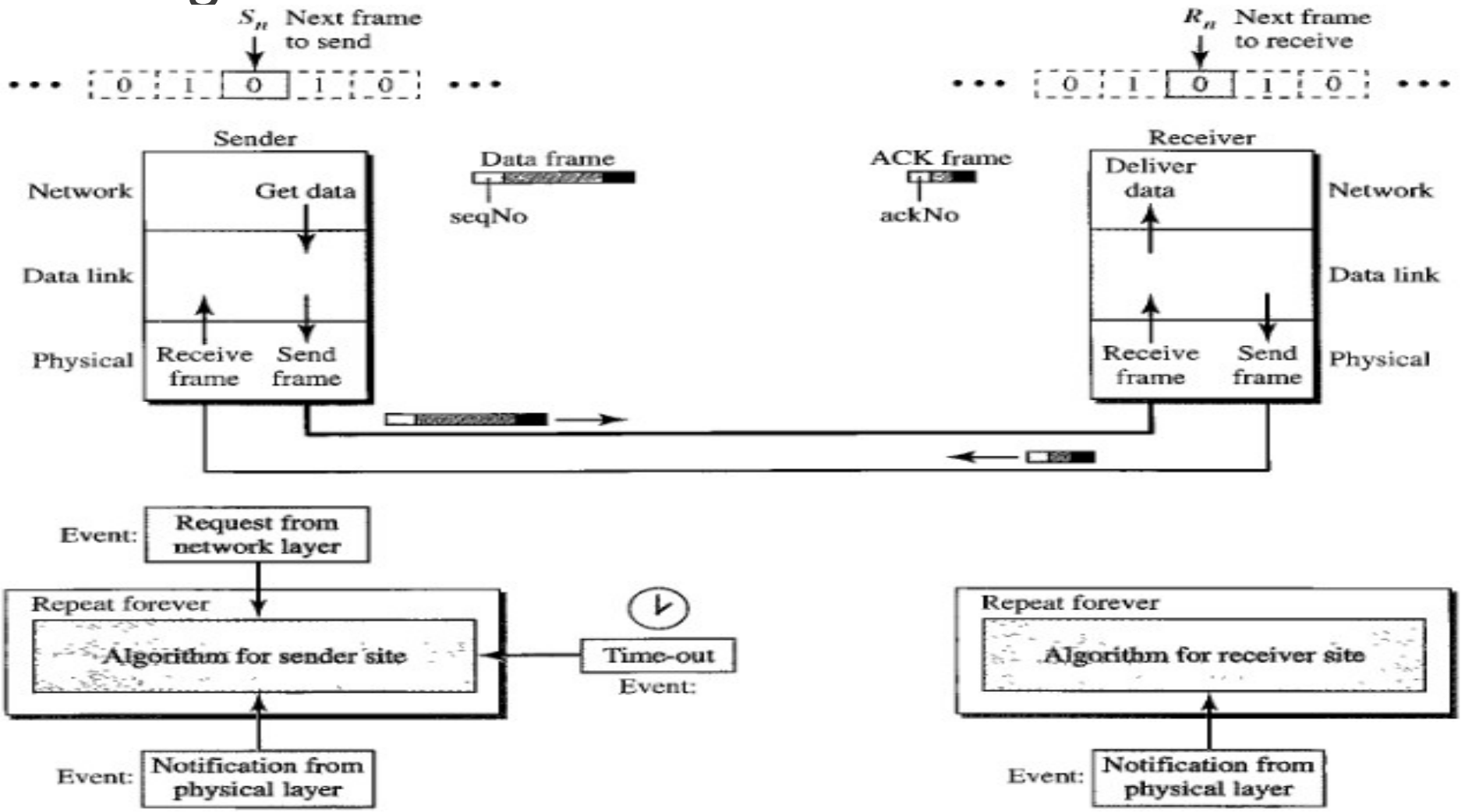
- Adds redundancy bits to data frame
- At the receiver end, if a frame is corrupted, it is silently discarded
- Numbering frames (sequence numbers) allows detection of lost frames
- Corrupted and lost frames must be resent
- Sender maintains a copy of sent frame and starts a timer
- If timer expires and no ACK received, resend
- ACK frames can also be corrupted

Stop-and-Wait ARQ

- Sequence numbers
  - Select the smallest range that provides unambigous communication; also to minimize frame size
  - If m bits long, sequence number start from 0 to 2^m-1 and then are repeated
  - If we have used x as a sequence number; we need only to use x+1 (modulo-2)
- Acknowledgment numbers
  - Announces the next sequence number expected (modulo-2)

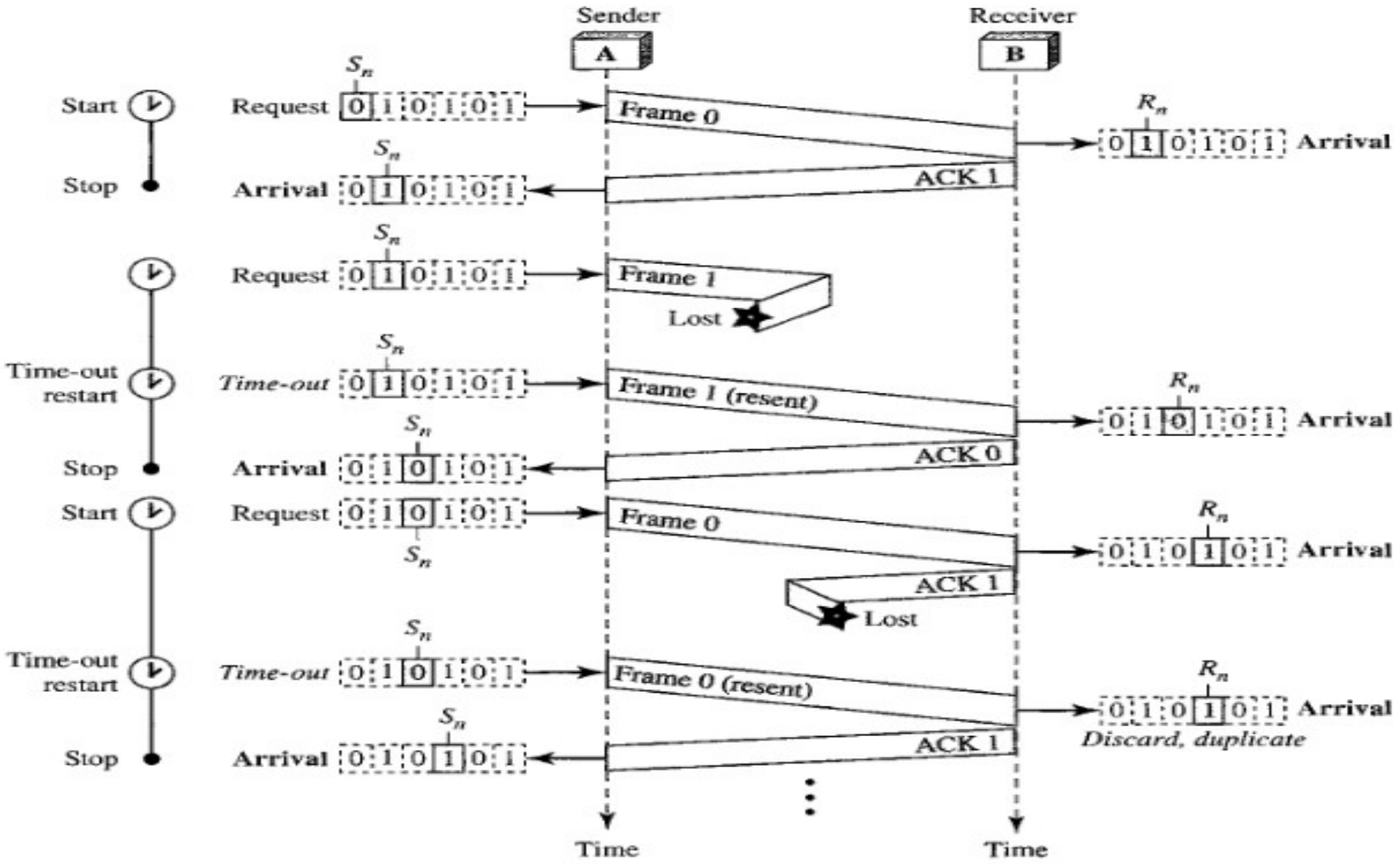
Stop-and-Wait ARQ

- Design



Stop-and-Wait ARQ

- Flow



Stop-and-Wait ARQ

- Inefficient if channel is thick and long: large bandwidth and round-trip delay is long
- Recall: bandwidth-delay product (BDP) – volume of pipe in bits
- BDP is a measure of the number of bits we can send out of our system while waiting for news from the receiver

Stop-and-Wait ARQ

- Given a bandwidth of 1Mbps and 1 bit takes 20 ms to make a round trip. What is BDP? If the length of frames is 1000 bits in length, what is the percentage utilization of the link?

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

  $PU=1000/20000 = 5\%$

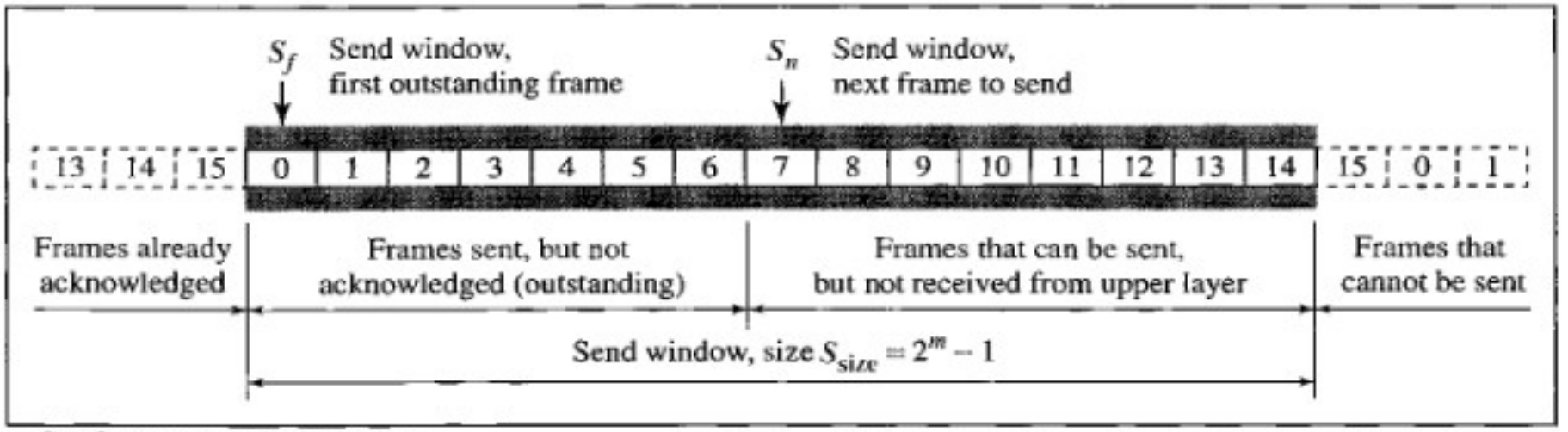
- What if we have protocol that can send up to 15 frames before ACKs are processed? What is the PU?

Go-Back-N ARQ

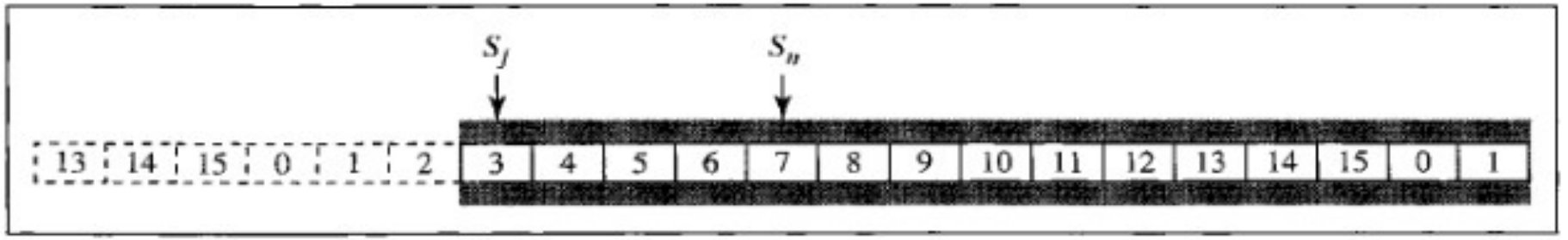
- Multiple frames must be in transit while waiting for ACK - pipelining
- We keep a copy of the frames in transit until acknowledgment arrives
- Sequence numbers are modulo 2^m, where m is the size of the sequence number field in bits
- Sliding window defines the range of sequence numbers that is the concern of the sender and receiver – sender and receiver deals with only a part of the range of sequence numbers
- ACKs are cumulative – more than one frame can be acknowledged by a single ACK

Go-Back-N ARQ

- Send window for Go-Back-N ARQ; S_f, S_n, S_{size}



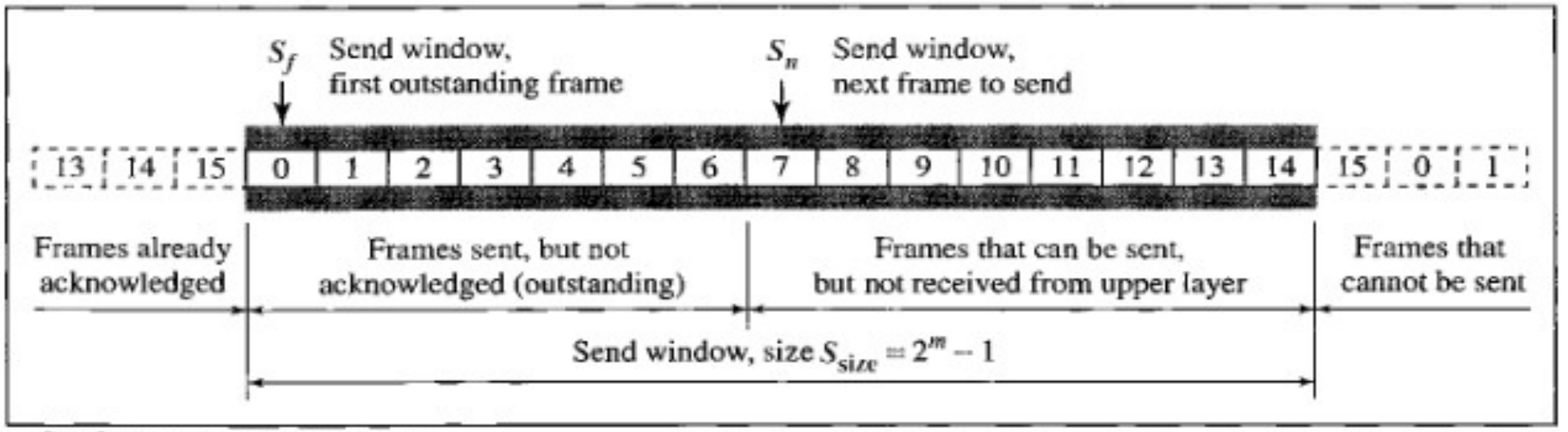
a. Send window before sliding



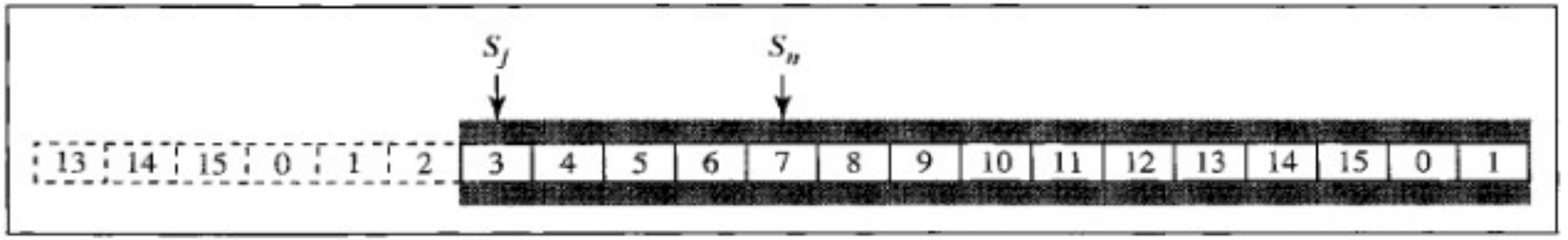
b. Send window after sliding

Go-Back-N ARQ

- Send window for Go-Back-N ARQ; S_f, S_n, S_{size}



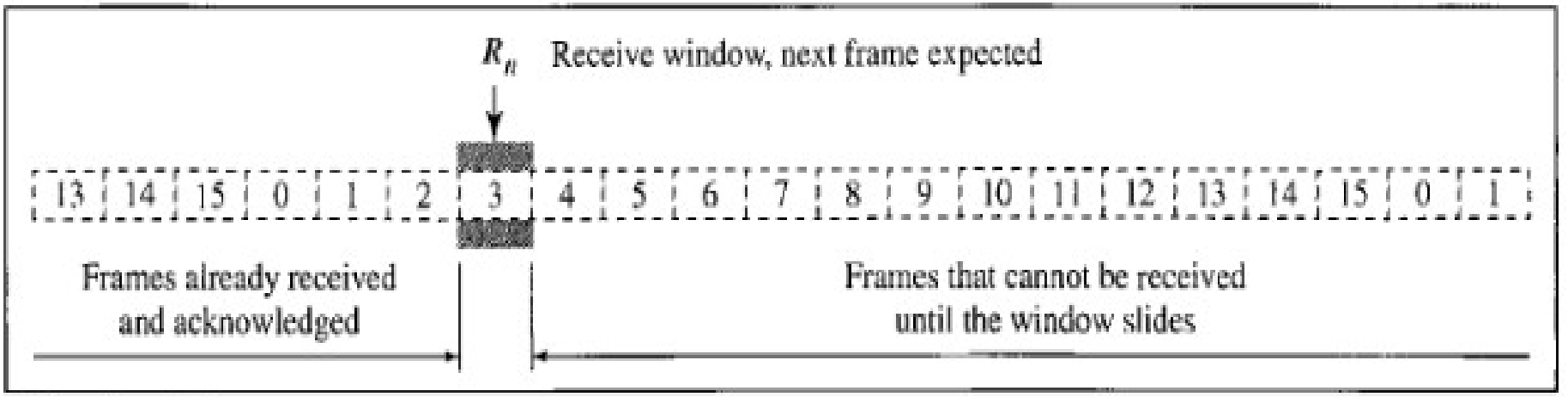
a. Send window before sliding



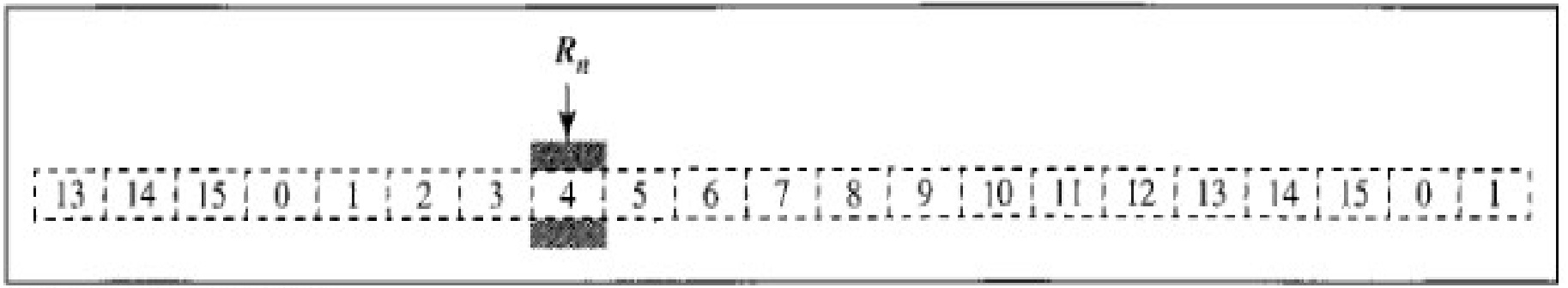
b. Send window after sliding

Go-Back-N ARQ

- Receive window for Go-Back-N ARQ; R_n



a. Receive window

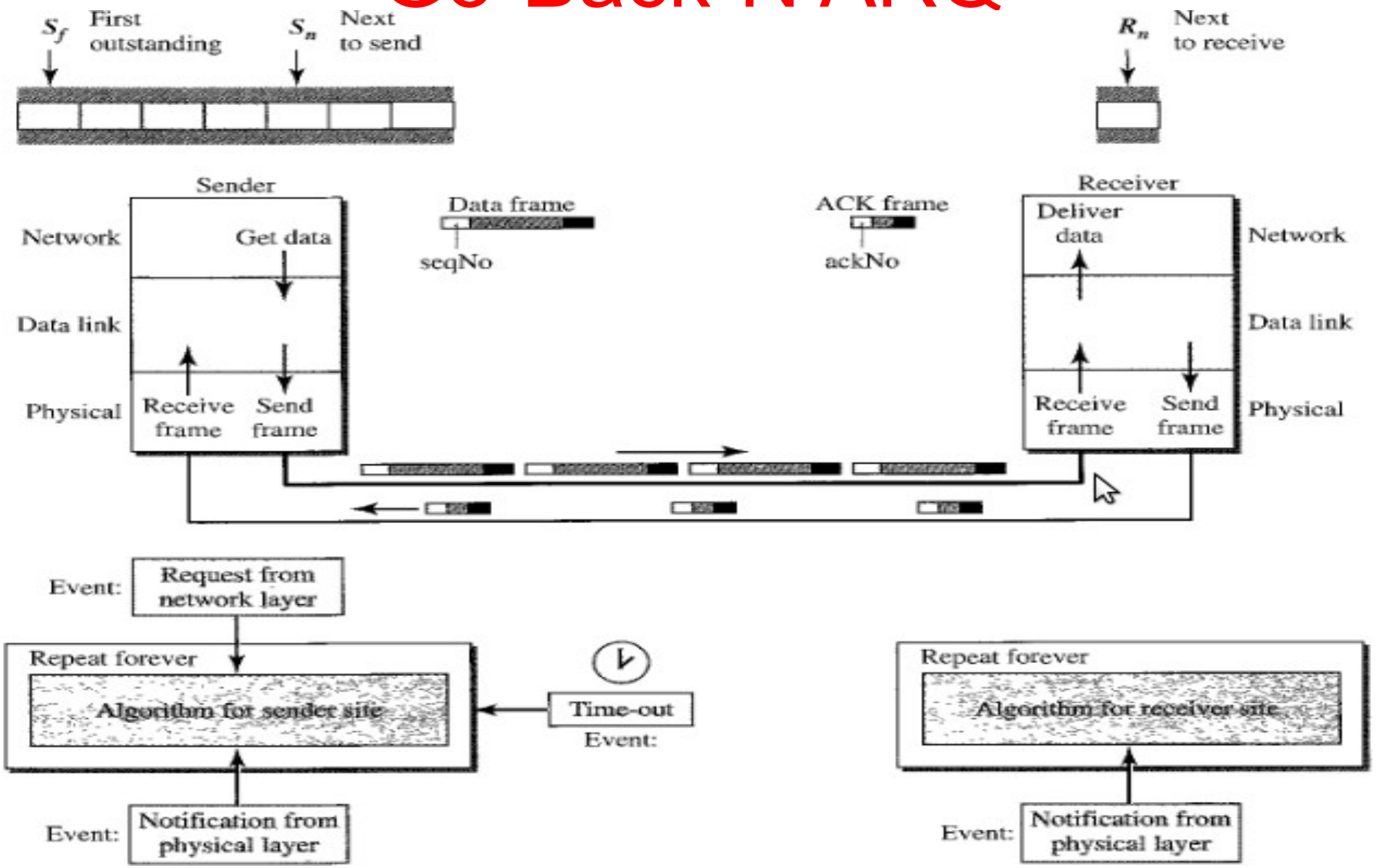


b. Window after sliding

Go-Back-N ARQ

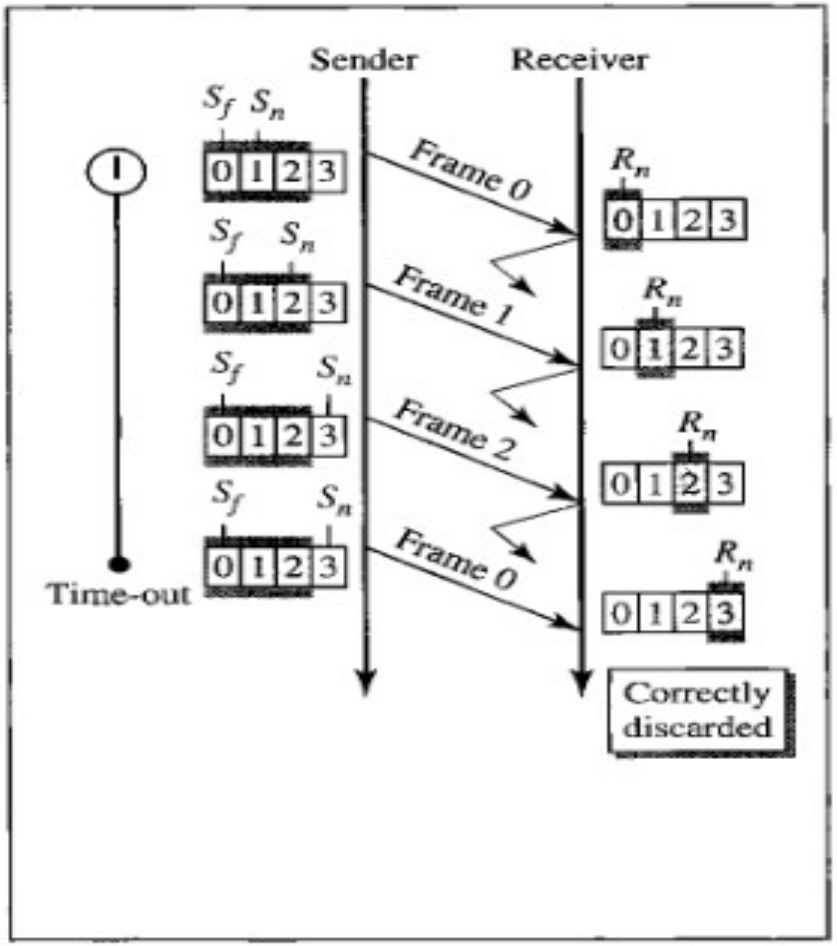
- Only one timer is used since the first outstanding frames always expires first
- Silence of receiver cause the timer of the unacknowledged frame at the sender site to expire
- Sender resends all outstanding frames
  - Ex. Sender has already sent frame 6, but timer for frame 3 expires. Sender must resend 3,4,5,6

Go-Back-N ARQ

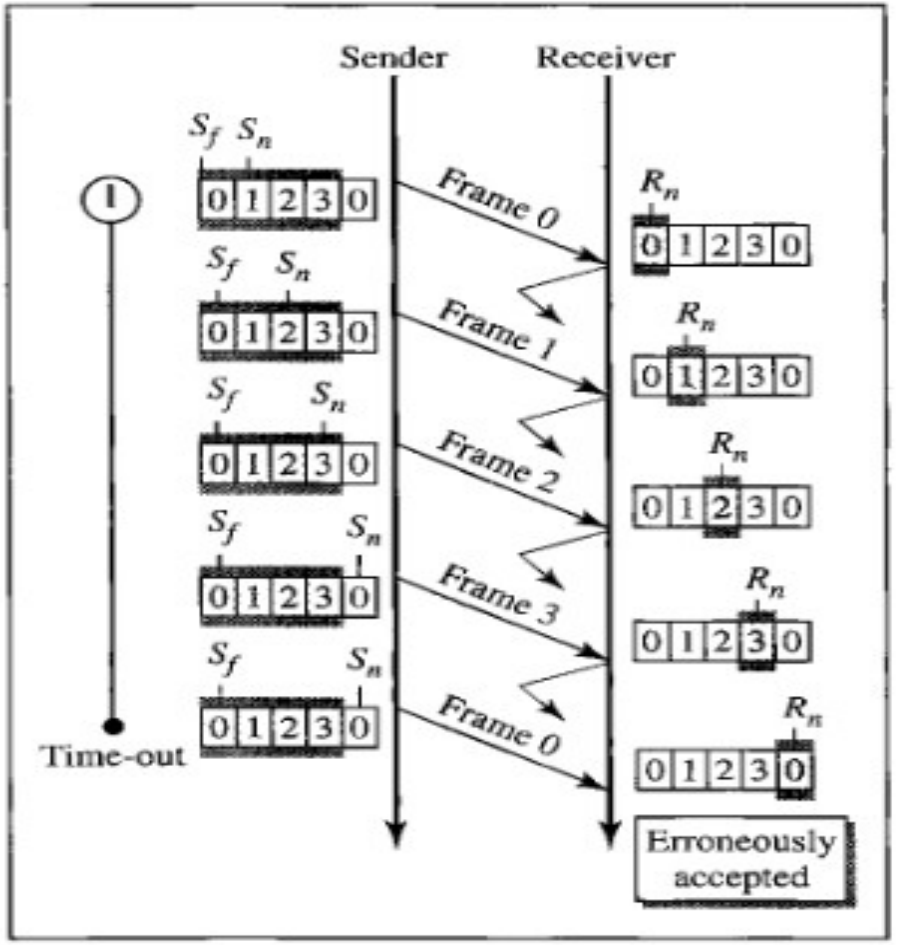

S_f First outstanding
S_n Next to send
R_n Next to receive
Sender
Receiver
Network
Data link
Physical
Get data
Receive frame
Send frame
Deliver data
Data frame
seqNo
ACK frame
ackNo
Event:
Request from network layer
Repeat forever
Algorithm for sender site
Time-out
Event:
Event:
Notification from physical layer
Repeat forever
Algorithm for receiver site
Event:
Notification from physical layer

Go-Back-N ARQ

- Send window size must be less than 2^m; receiver window size is 1



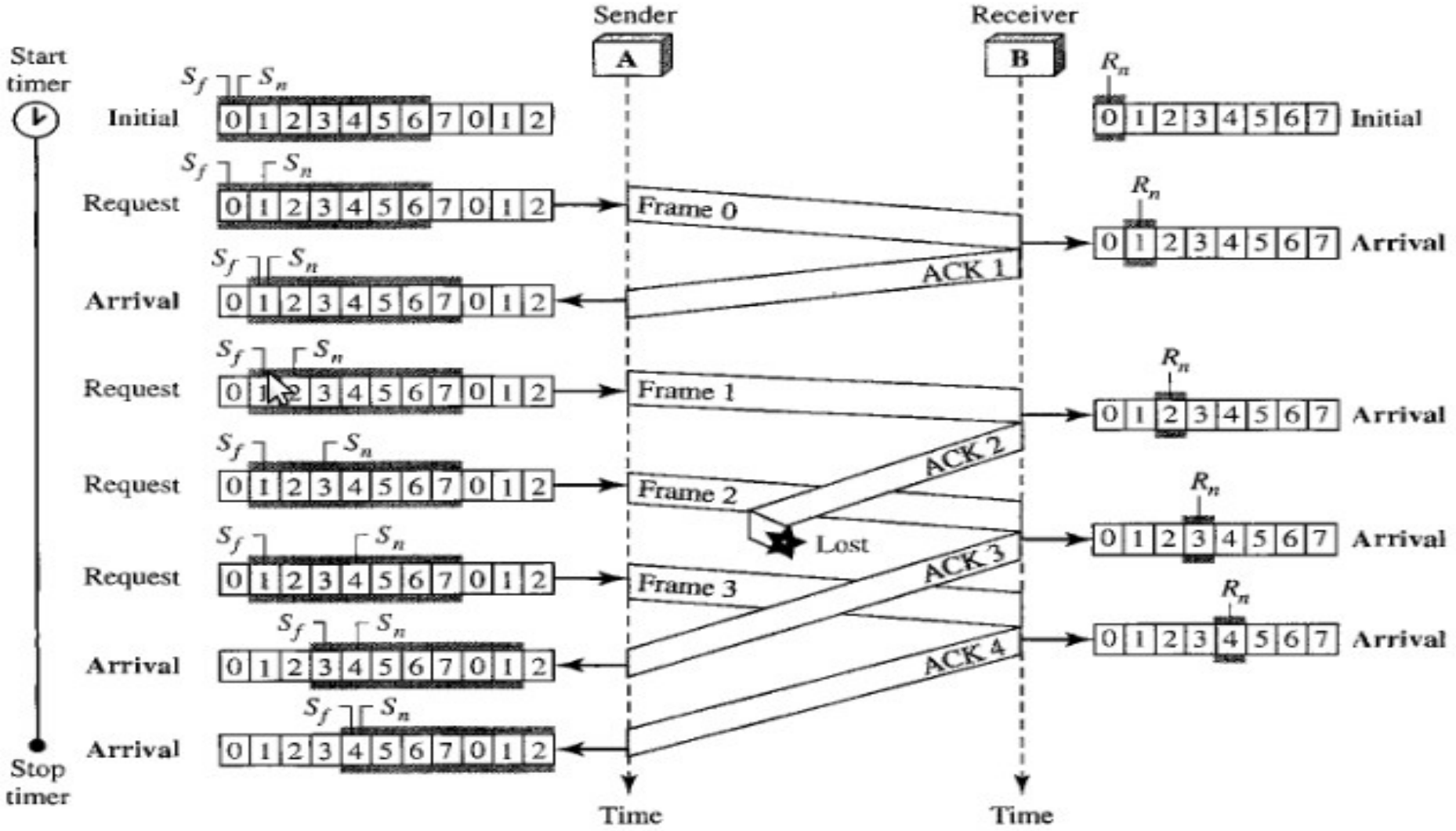
a. Window size < 2^m



b. Window size = 2^m

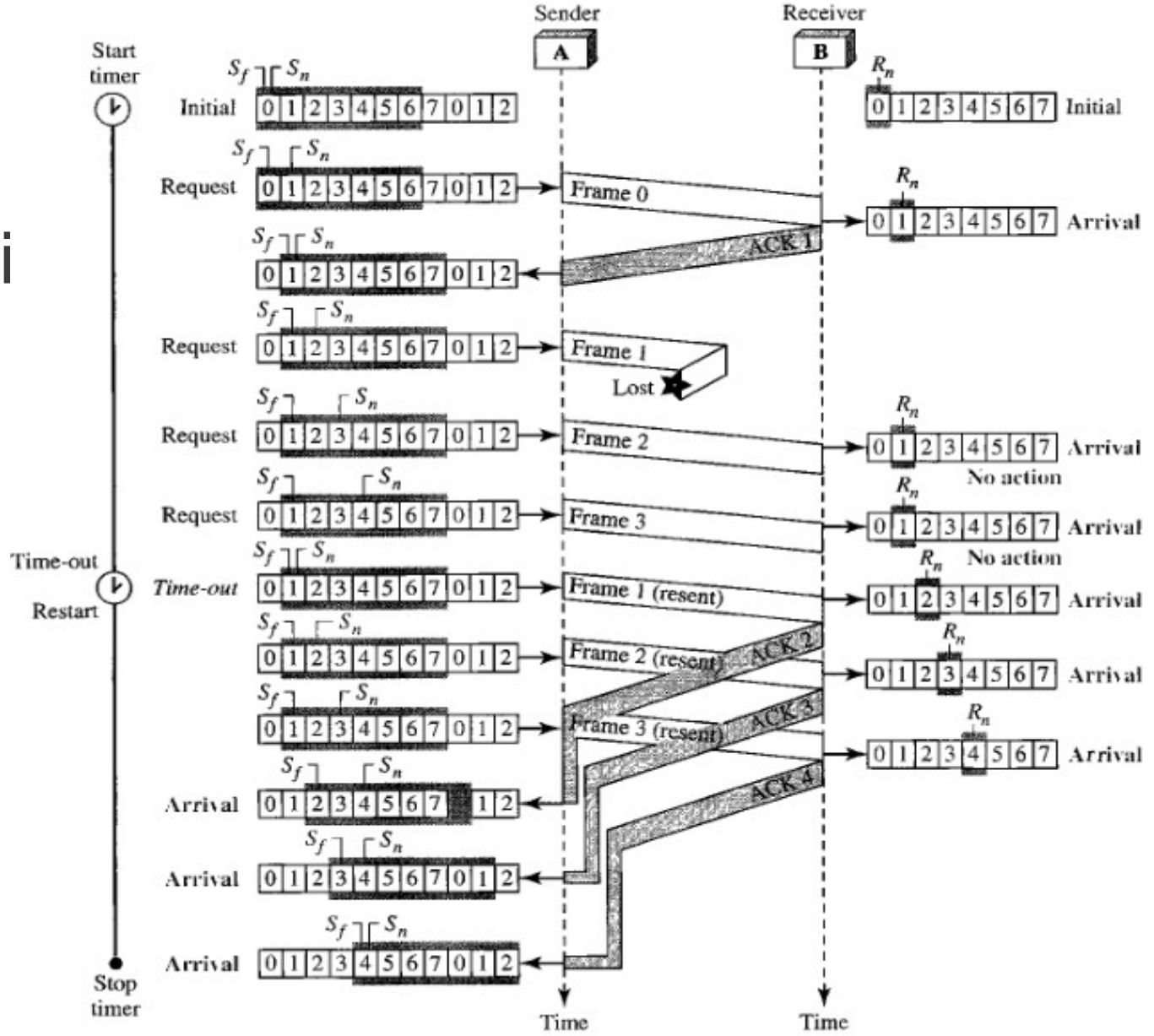
Go-Back-N ARQ

- Example when forward channel is reliable



Go-Back-N ARQ

- Lost frames
- Stop-and-Wai with send window size of 1

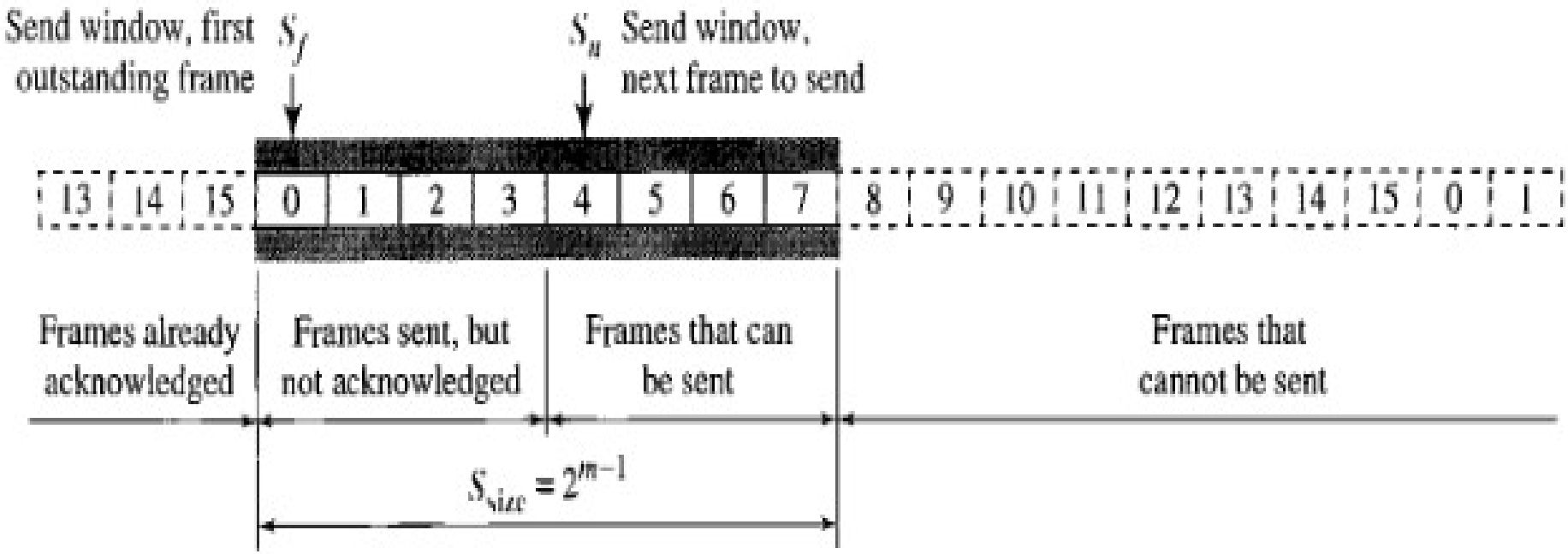


Selective Repeat ARQ

- Only damaged frames are resent
- More efficient for noisy links
- Processing at the receiver is more complex
- Send window max size is 2^{m-1}; same size in receive window;smaller than GBN
  - Less efficiency but fewer duplicate frames

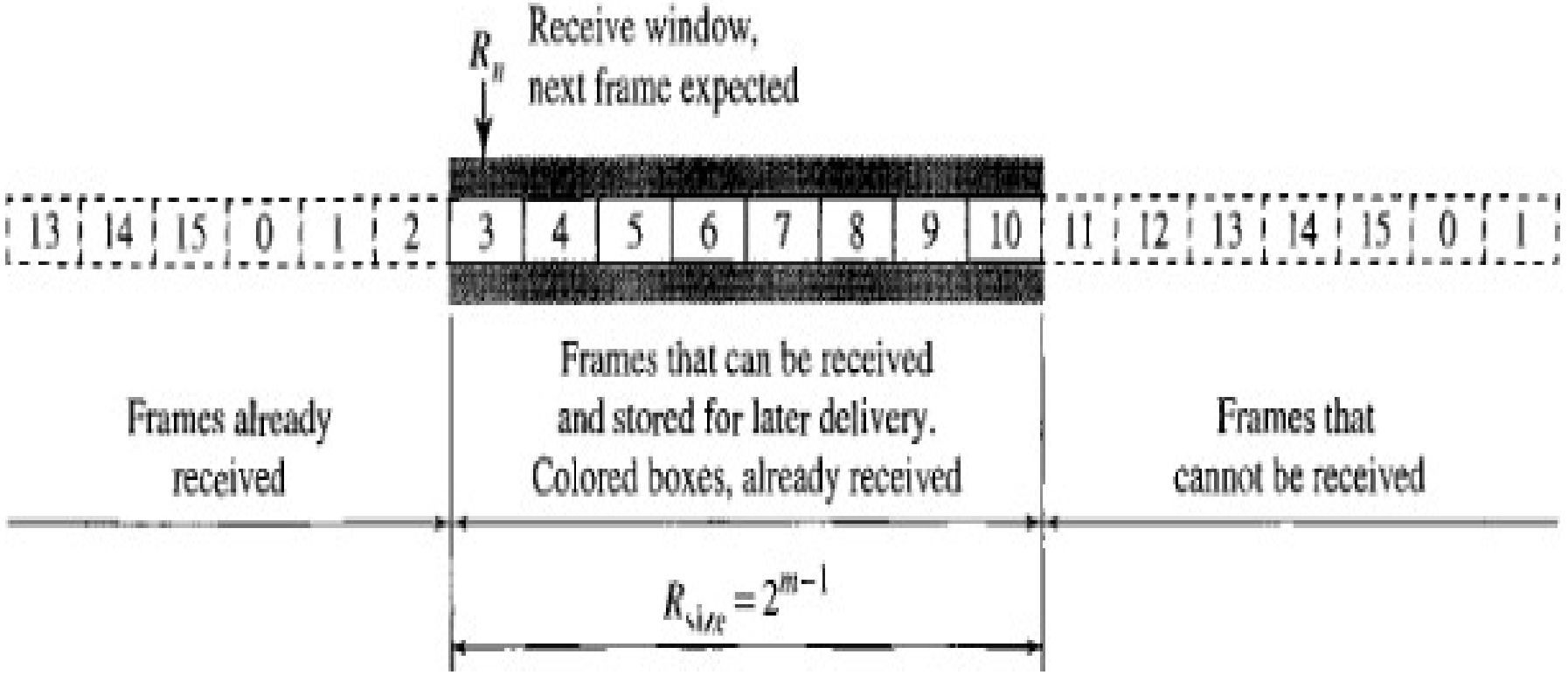
Selective Repeat ARQ

- Send window

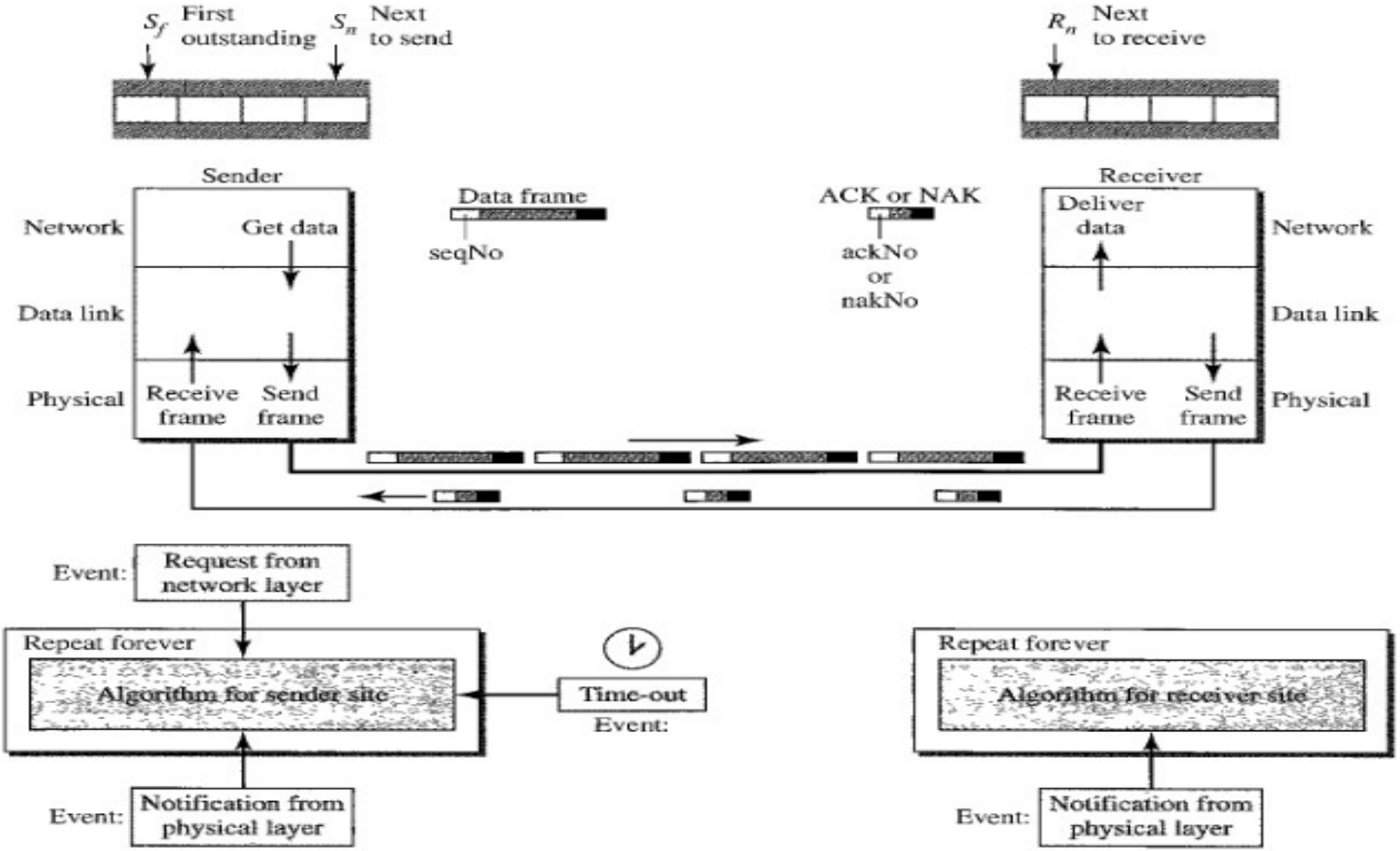


Selective Repeat ARQ

- Receive window

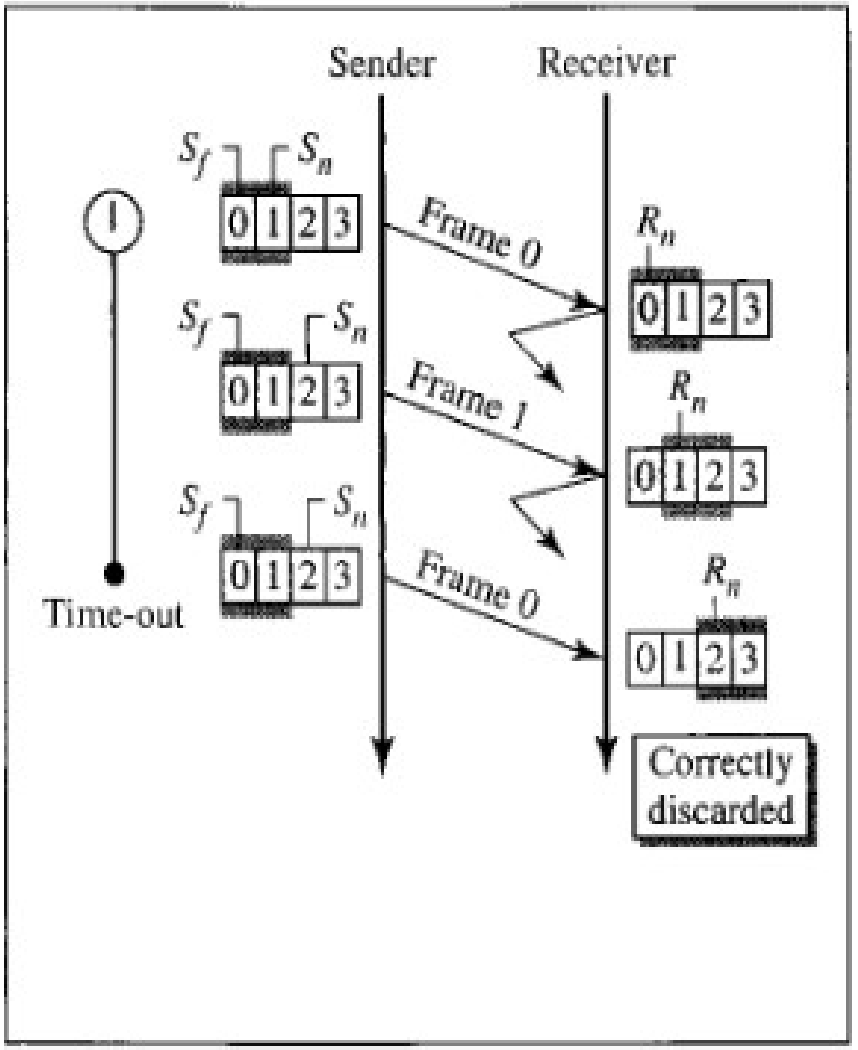


Selective Repeat ARQ

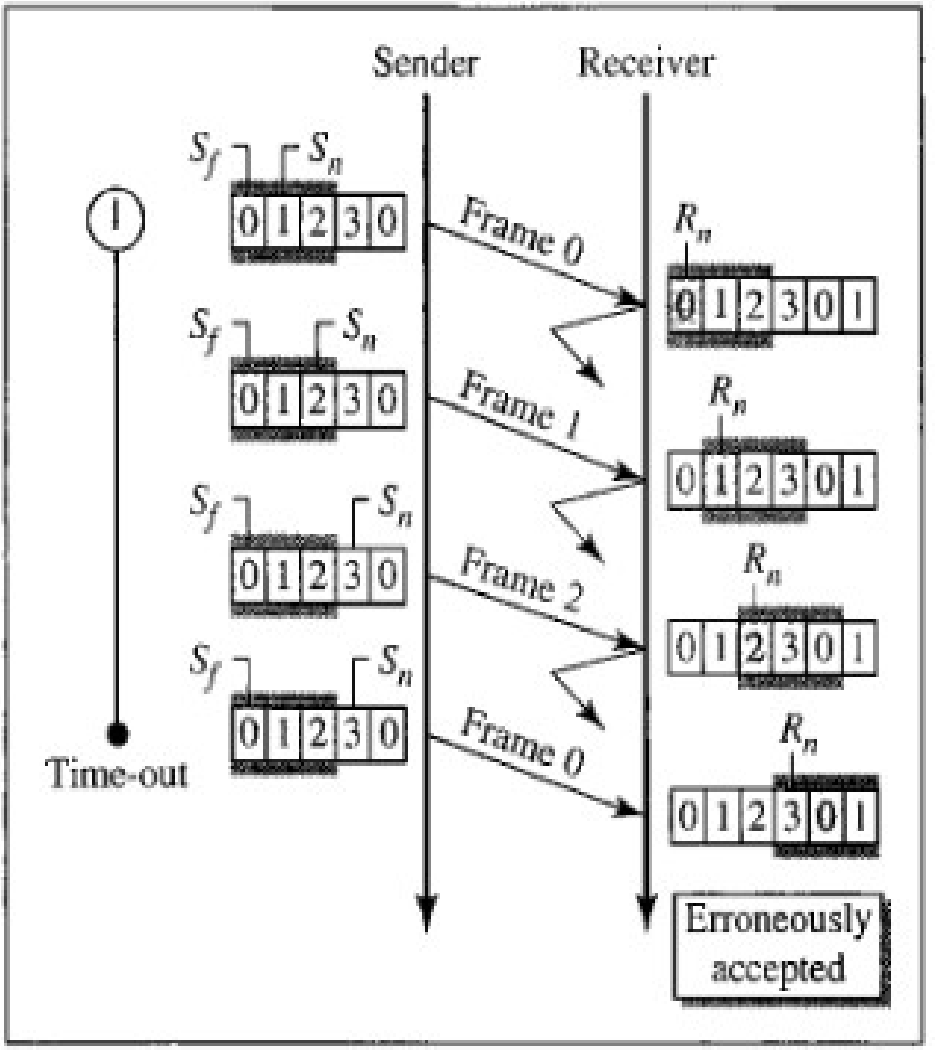


Selective Repeat ARQ

- Window size must be at most one-half of 2^m

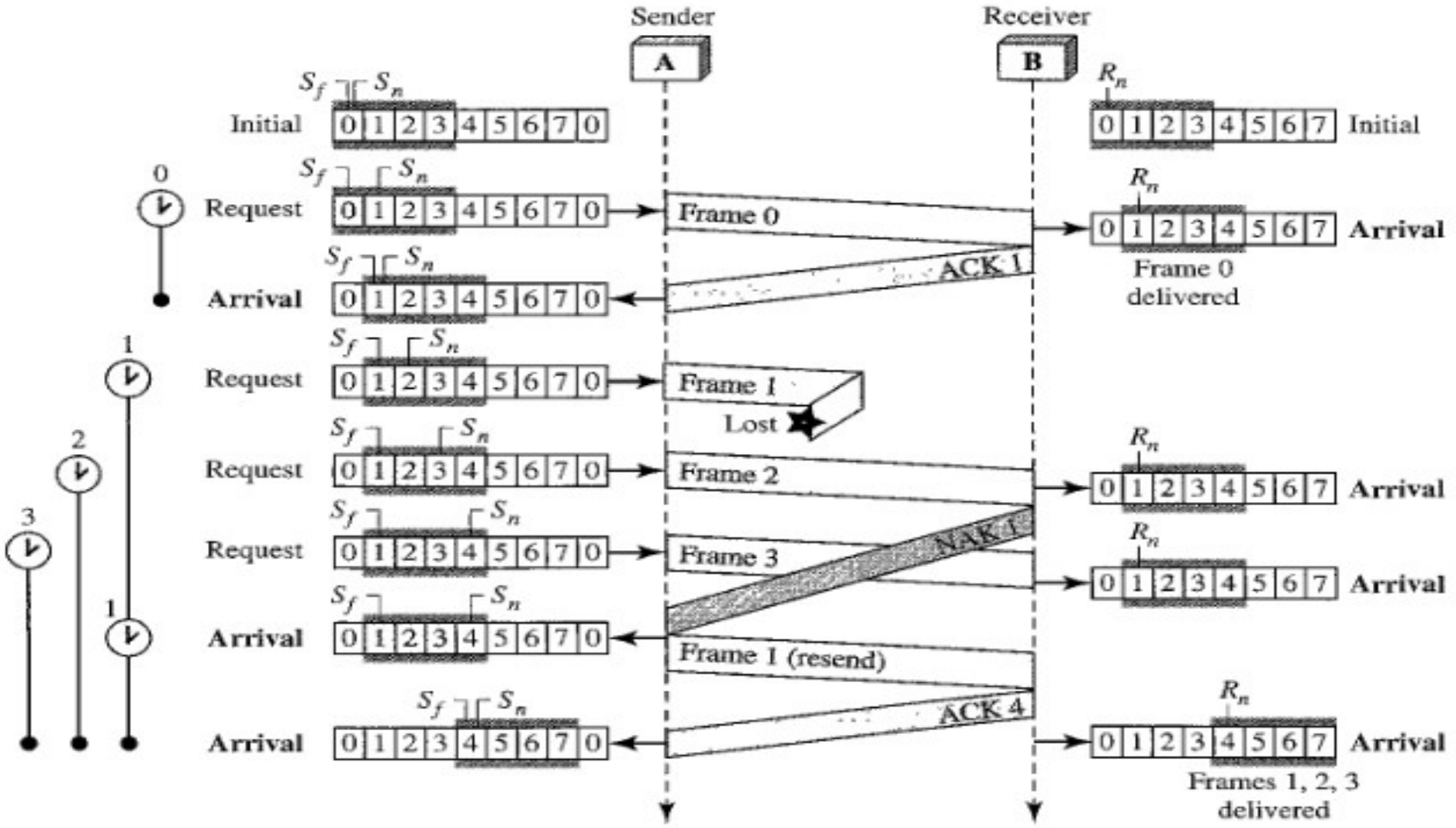


a. Window size = 2^{m-1}



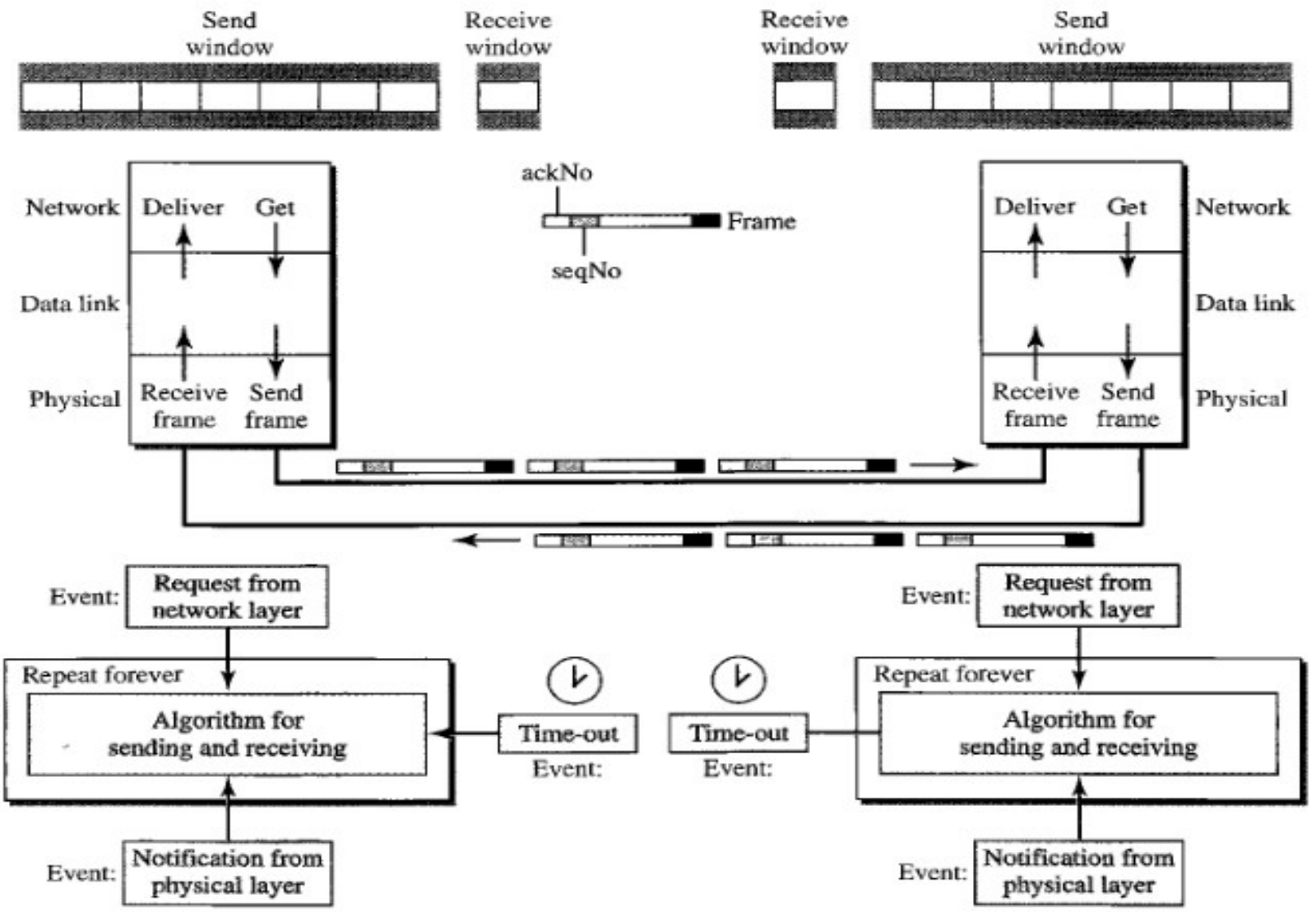
b. Window size > 2^{m-1}

Selective Repeat ARQ



Piggybacking

- GBN



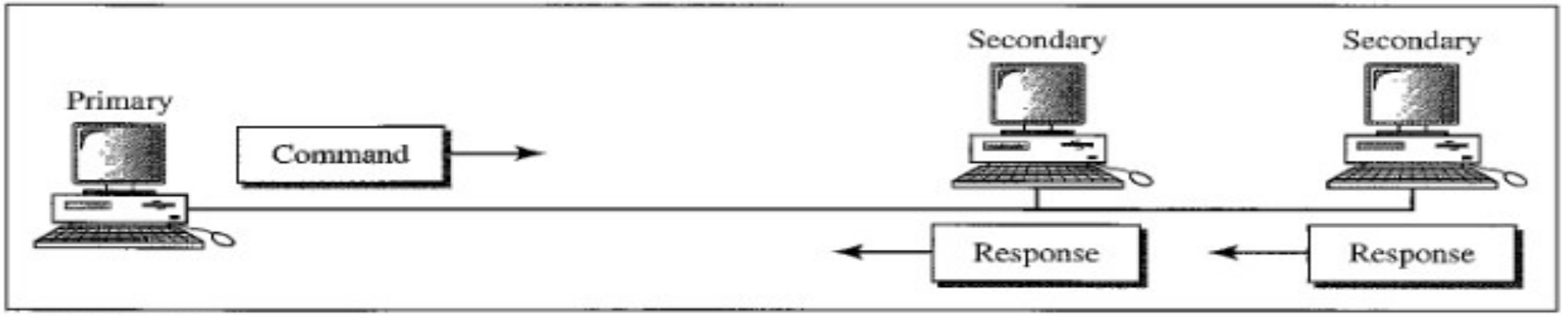
High-level Data Link Control (HDLC)

- A bit-oriented protocol for communication over point-to-point and multiple links
- Implements the ARQ mechanisms discussed

Transfer Modes

- Normal Response Mode (NRM) – One primary stations and multiple secondary stations

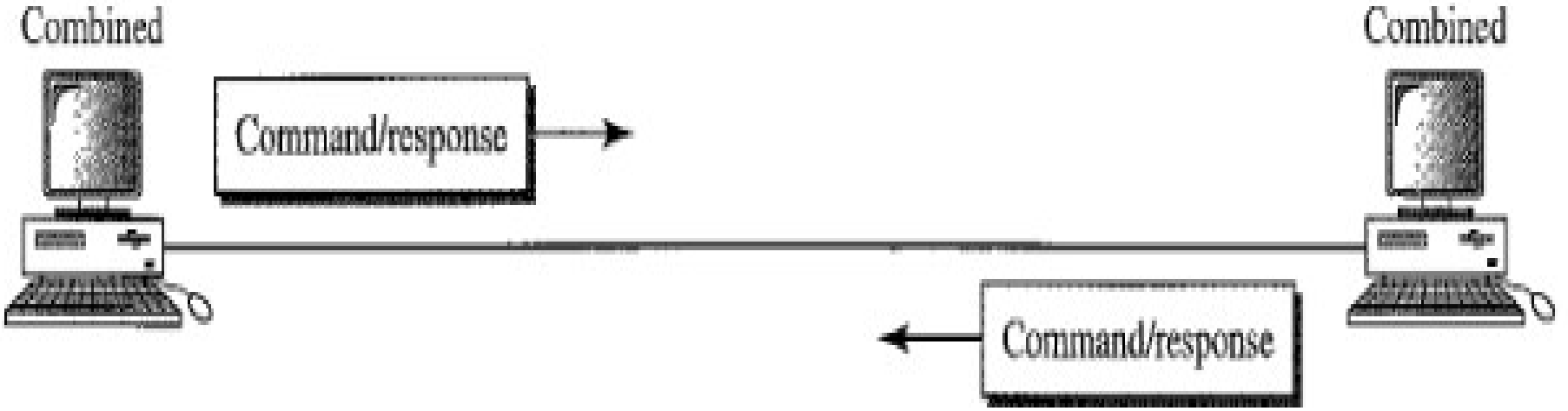
a. Point-to-point



b. Multipoint

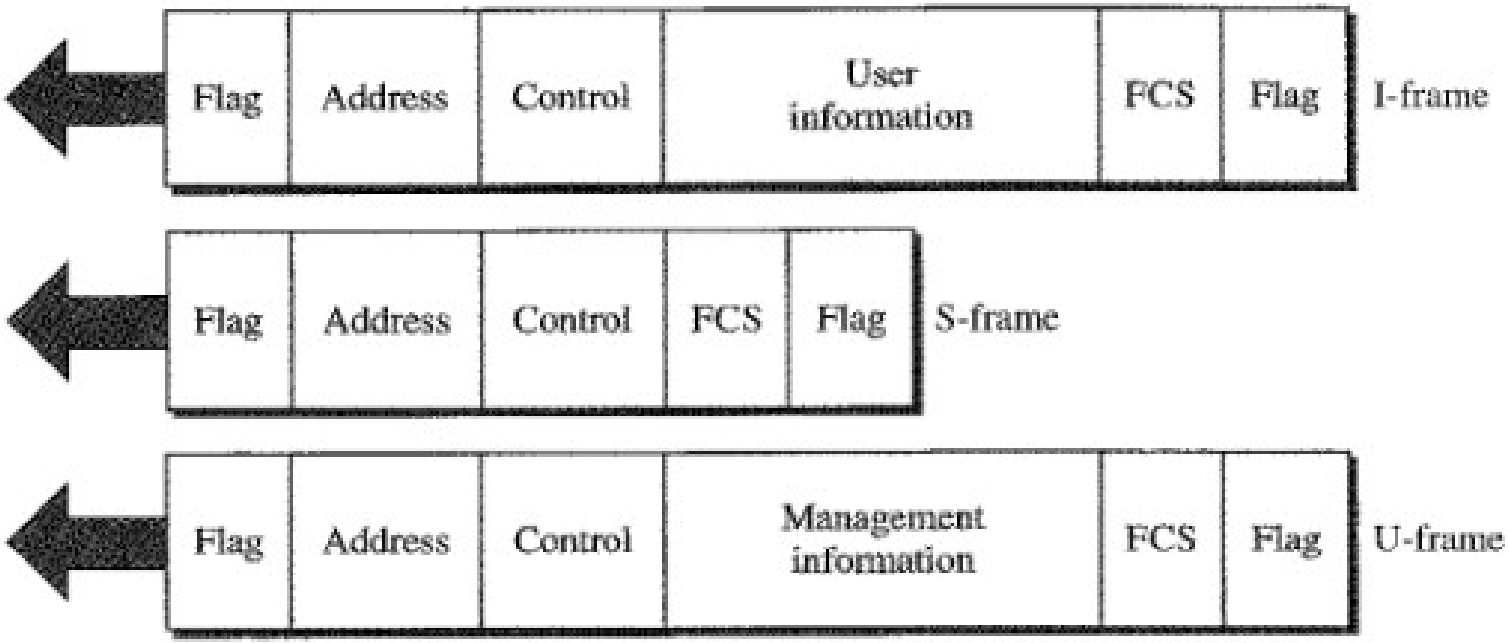
Transfer Modes

- Asynchronous Balanced Mode (ABM) – each station can function as primary and secondary



Frames

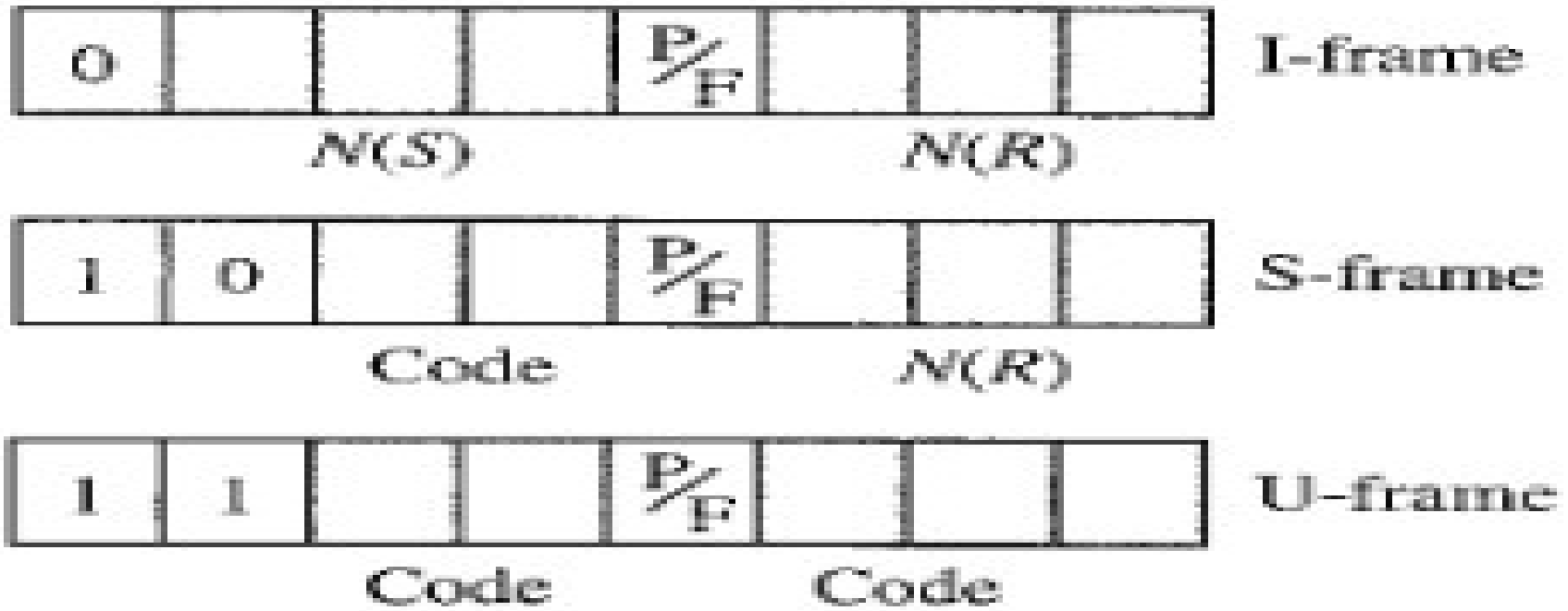
- I-frames – information frames
- S-frames – supervisory frames
- U-frames – unnumbered frames



Frame Fields

- Flag – 8-bit pattern: 01111110
- Address – to secondary from primary; from secondary to primary;
  - can be 1 byte or several bytes
  - 1 byte can identify up to 128 stations
  - The last bit of the last byte is always ends with 1;signals the last byte of the address field
- Control field – 1- or 2-byte segment used for flow and error control; depends on the type of frame
- Information field – data
- Frame Check Sequence – 2- or 4-byte ITU-T CRC

Control Field



Control Field: I-Frames

- First bit: 0
- N(S) – sequence number of the frame
- N(R) – acknowledgment number when piggybacking is used
- P/F bit – Poll or Final bit

Control Field: S-Frames

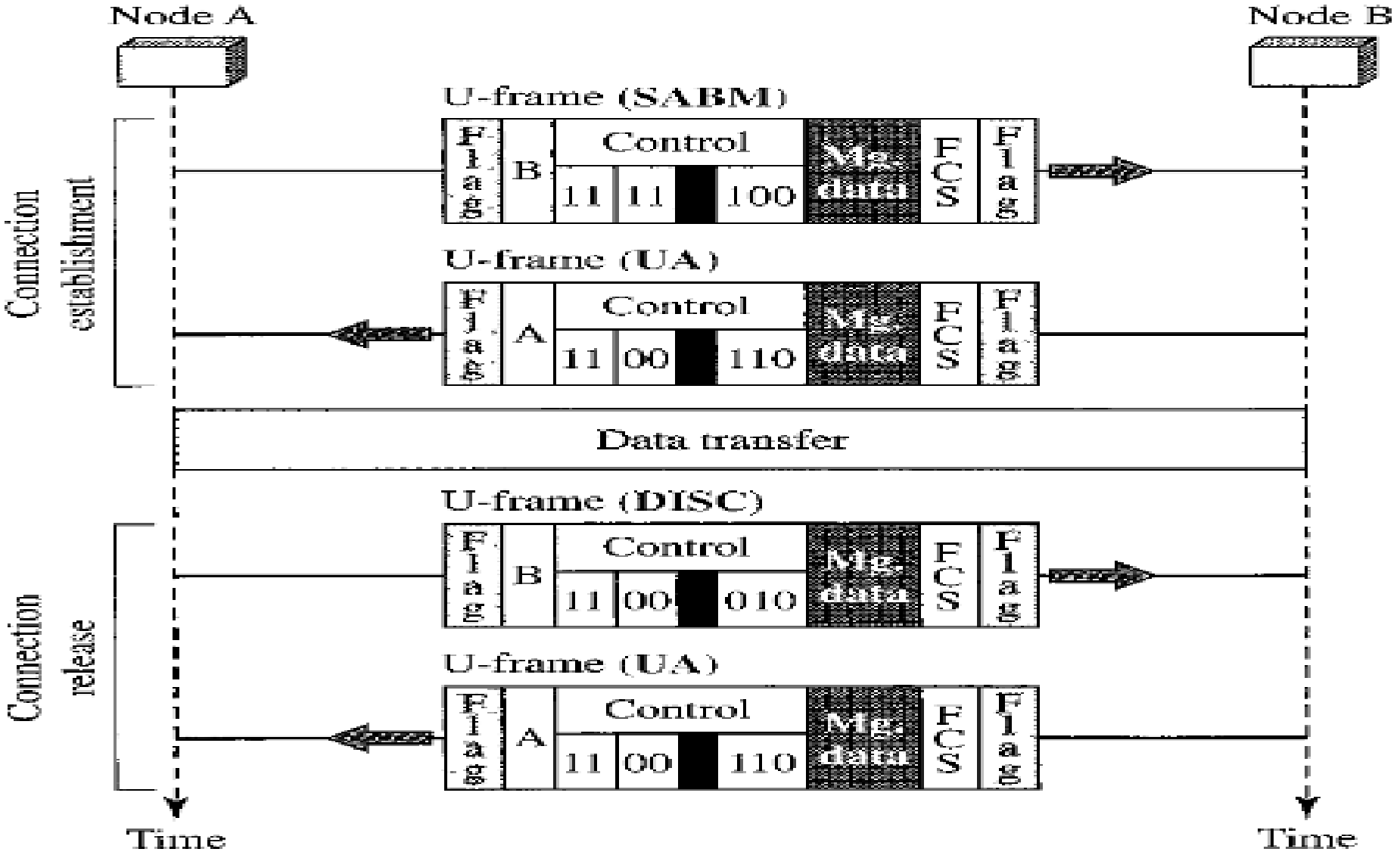
- First two bits: 10
- Control Codes
  - Receive ready (RR) - 00
  - Receive not ready (RNR) – 10 ; receiver is busy
  - Reject (REJ) – 01; NAK used in GBN
  - Selective reject (SREJ) – 11; NAK used in SR

Control Field: U-Frames

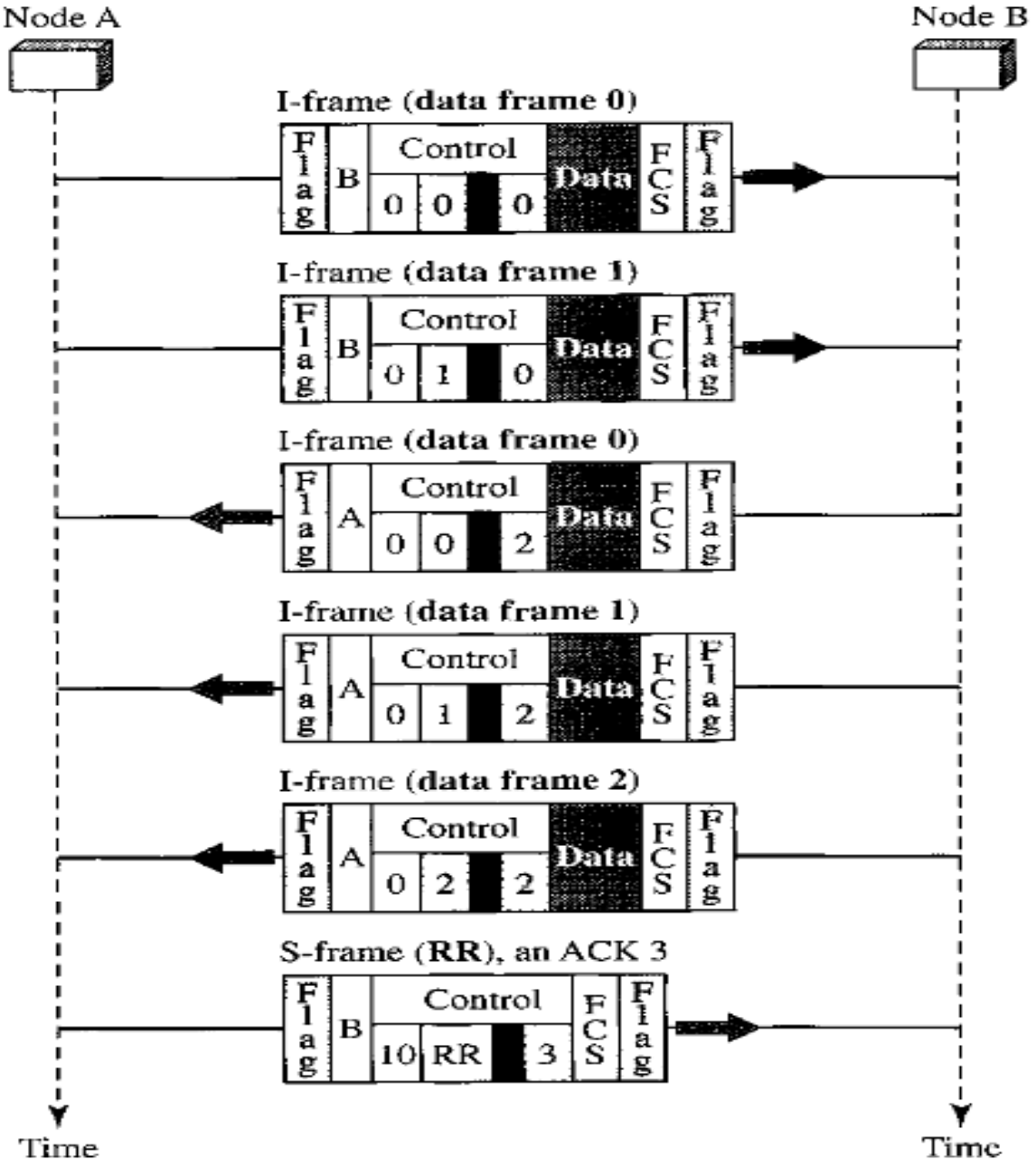
- First two bits: 11

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

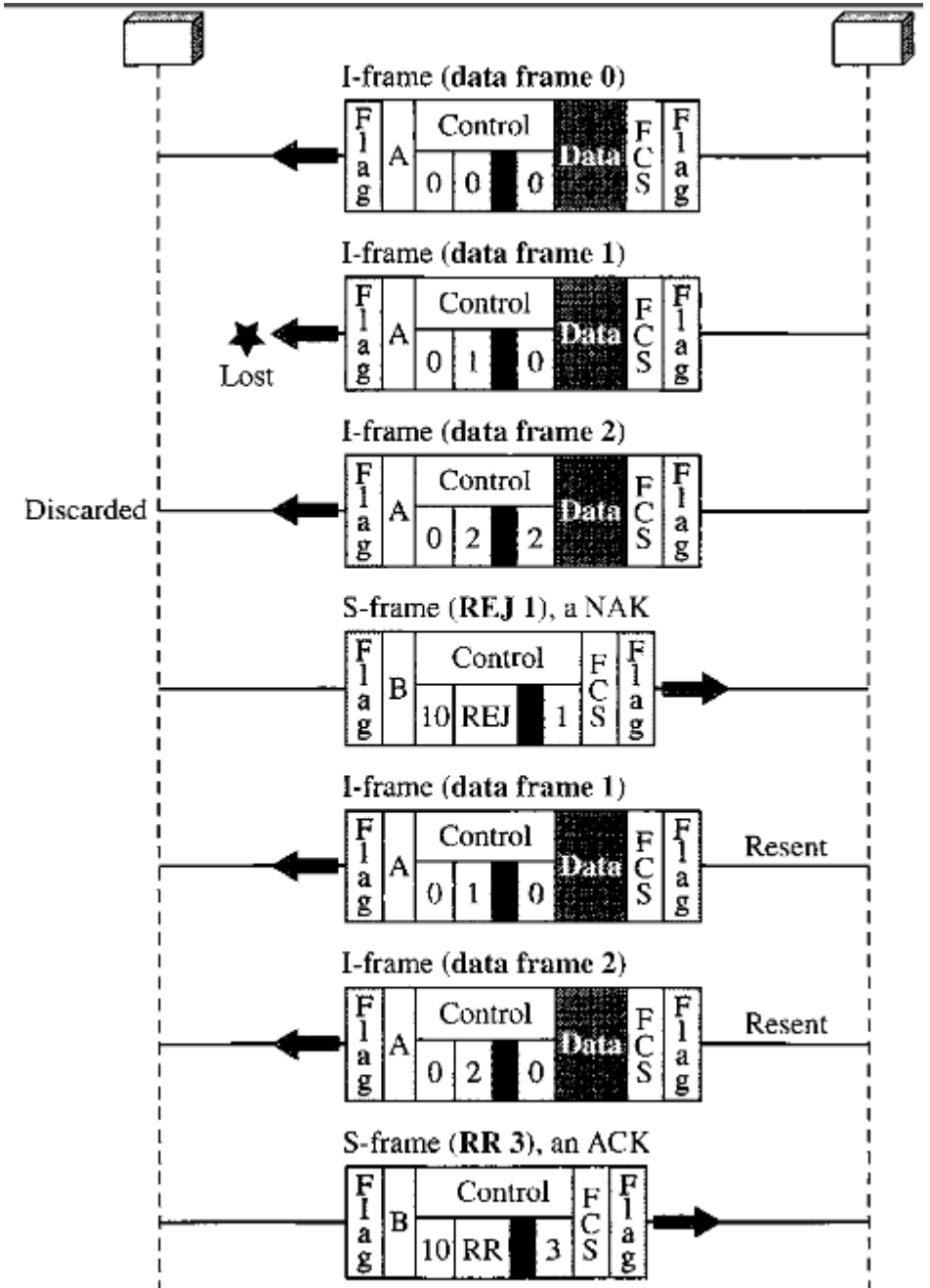
Connection and Disconnection



Piggybacking without Error



Piggybacking with Error



Enjoy! :)