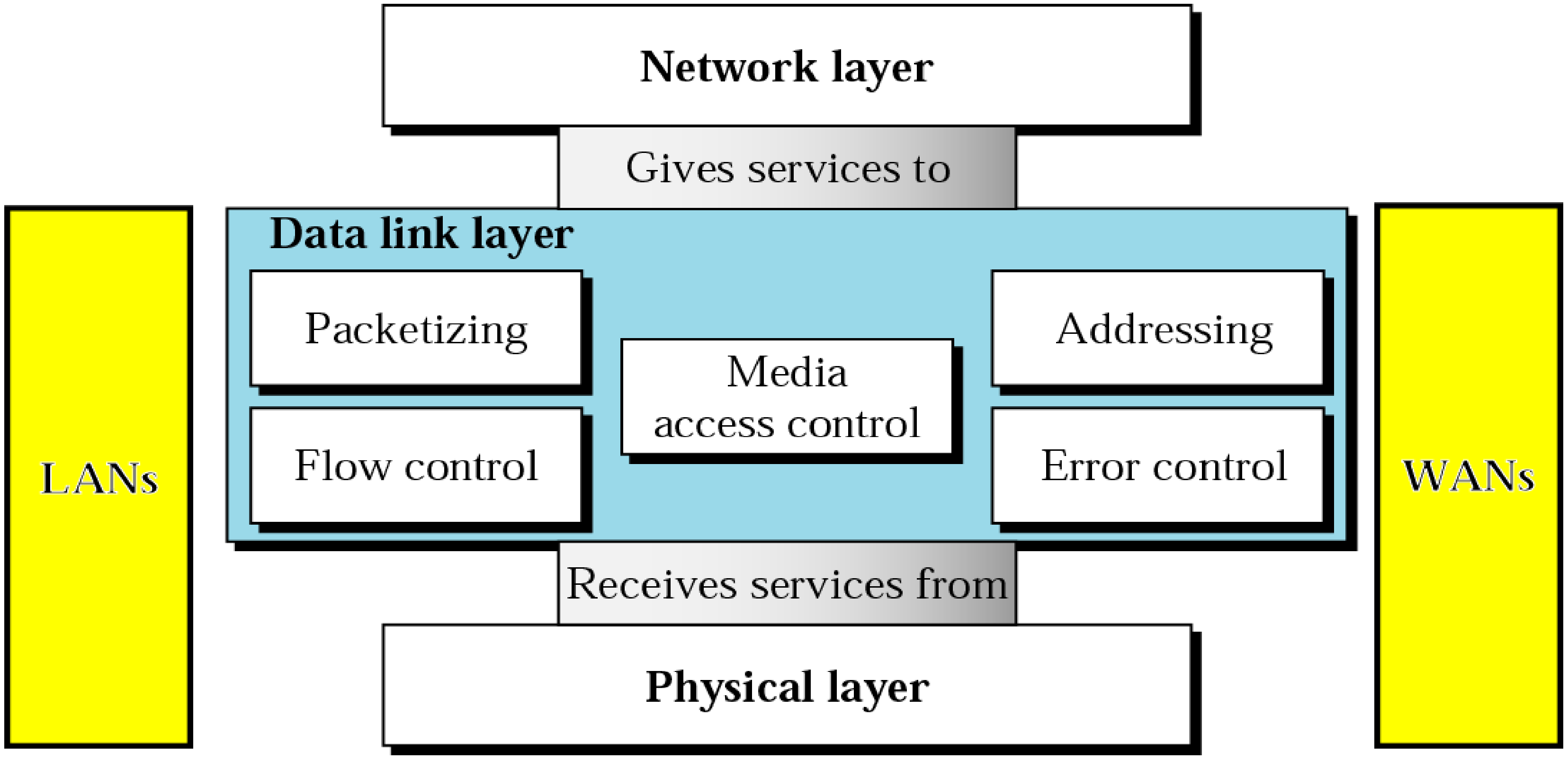
Data Communications and Networks
Unit-2

Outline

- OVERVIEW
- FRAMING
- FLOW AND ERROR CONTROL
- PROTOCOLS
- NOISELESS CHANNELS
- NOISY CHANNELS
- HDLC
- ALOHA

DATA LINK LAYER



Overview

- The two main functions of the data link layer are data link control and media access control.
- Data link control, deals with the design and procedures for communication between two adjacent nodes: node-to-node communication.
- Media access control, or how to share the link.
- Data link control functions include framing, flow and error control, and software implemented protocols.

FRAMING

- The data link layer, needs to pack bits into frames, so that each frame is distinguishable from another.
- Framing in the data link layer separates a message from one source to a destination, or from other messages to other destinations, by adding a sender address and a destination address.
- The whole message could be packed in one frame, that is not normally done.
- One reason is that a frame can be very large, making flow and error control very inefficient.

Fixed-Size Framing

- In fixed-size framing, there is no need for defining the boundaries of the frames; the size itself can be used as a delimiter.
- An example of this type of framing is the ATM wide-area network, which uses frames of fixed size called cells.

Variable-Size Framing

- In variable-size framing, we need a way to define the end of the frame and the beginning of the next.
- Two approaches were used for this purpose:
  - a character-oriented approach
  - a bit-oriented approach

Variable-Size Framing

- **Character-Oriented Protocols**
- 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,
- The 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 (l-byte) flag is added at the beginning and the end of a frame
- Byte stuffing

Variable-Size Framing

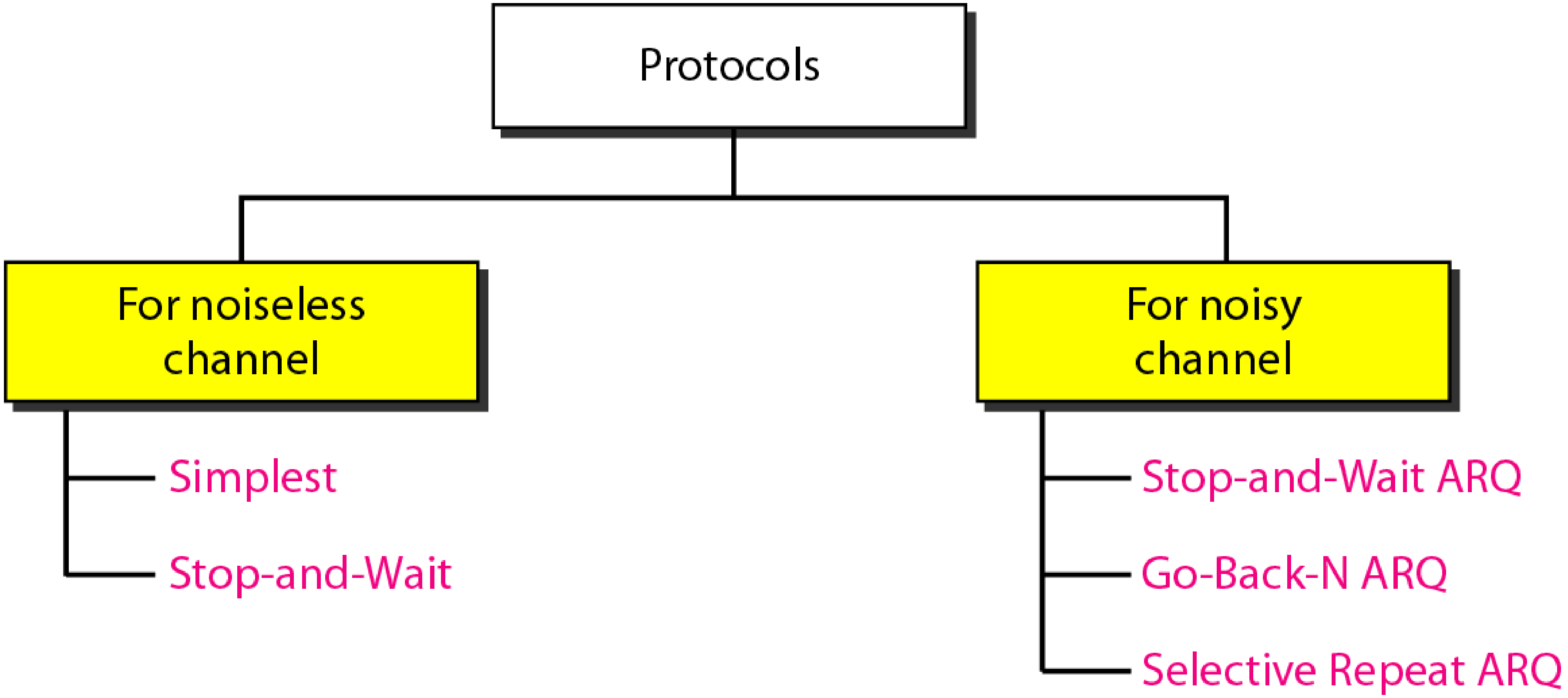
- **Bit-Oriented Protocols**
- In a bit-oriented protocol, the data section of a frame is a sequence of bits to be interpreted by the upper layer as text, graphic, audio, video.
- In addition to headers (and possible trailers), we still need a delimiter to separate one frame from the other.
- Most protocols use a special 8-bit pattern flag 01111110 as the delimiter to define the beginning and the end of the frame.
- Bit stuffing

FLOW AND ERROR CONTROL

- Flow control coordinates the amount of data that can be sent before receiving an acknowledgment and is one of the most important duties of the data link layer.

- Error Control is 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.
- Any time an error is detected in an exchange, specified frames are retransmitted. This process is called automatic repeat request (ARQ).

PROTOCOLS


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

PROTOCOLS

- The protocols are normally implemented in software by using one of the common programming languages.
- We divide the discussion of protocols into those that can be used for
  - noiseless (error-free) channels
  - noisy (error-creating) channels

NOISELESS CHANNELS

- Let us first assume we have an ideal channel in which no frames are lost, duplicated, or corrupted.
- We introduce two protocols for this type of channel. The first is a protocol that does not use flow control; the second is the one use flow control.

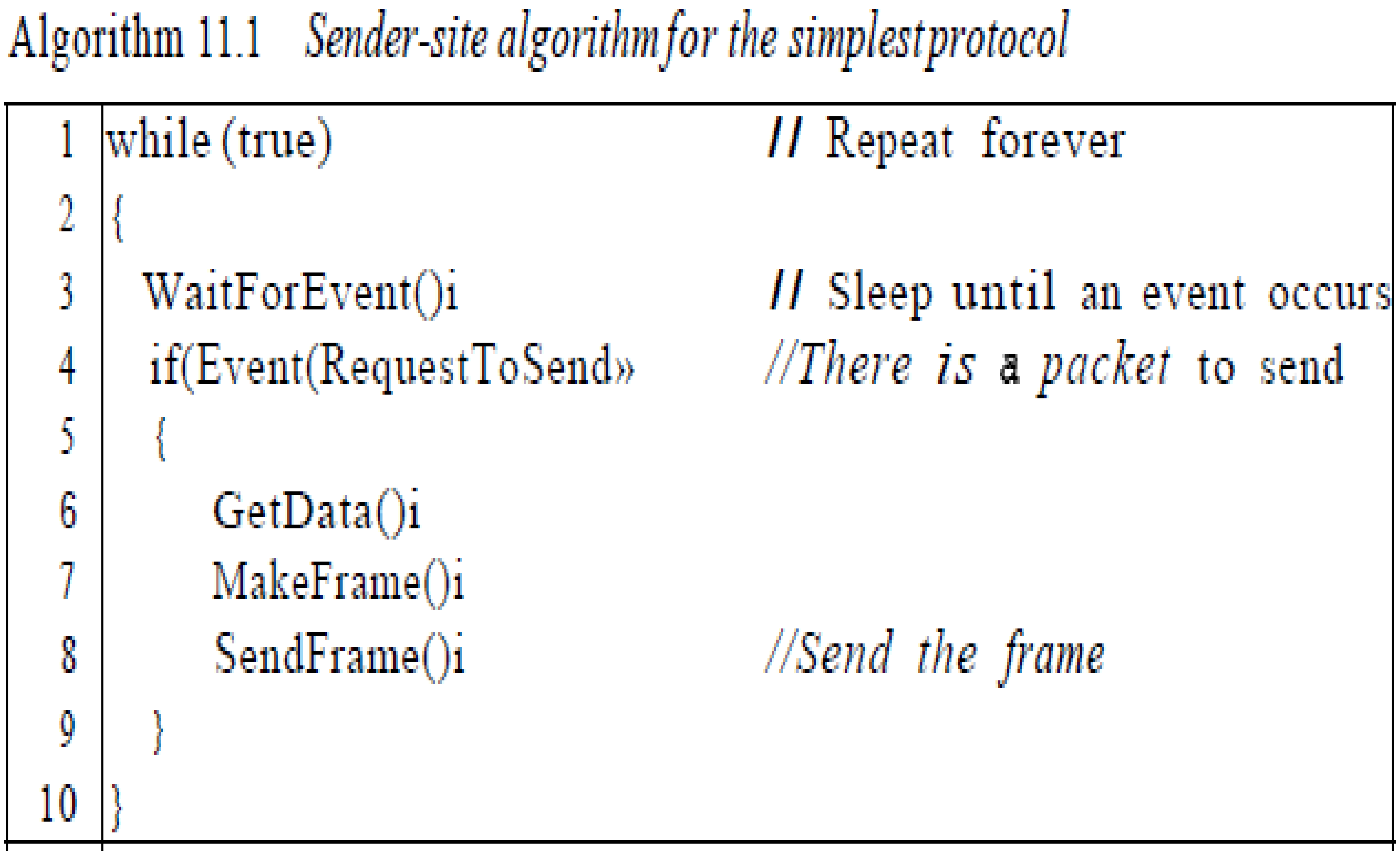
Simplest Protocol

- It is one that has no flow or error control.
- 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.
- The data link layer of the receiver immediately removes the header from the frame and hands the data packet to its network layer
- The receiver can never be overwhelmed with incoming frames.

Design

- The sender site cannot send a frame until its network layer has a data packet to send.
- The receiver site cannot deliver a data packet to its network layer until a frame arrives.
- If the protocol is implemented as a procedure, we need to introduce the idea of events in the protocol.
- The procedure at the sender site is constantly running; there is no action until there is a request from the network layer.
- The procedure at the receiver site is also constantly running, but there is no action until notification from the physical layer arrives

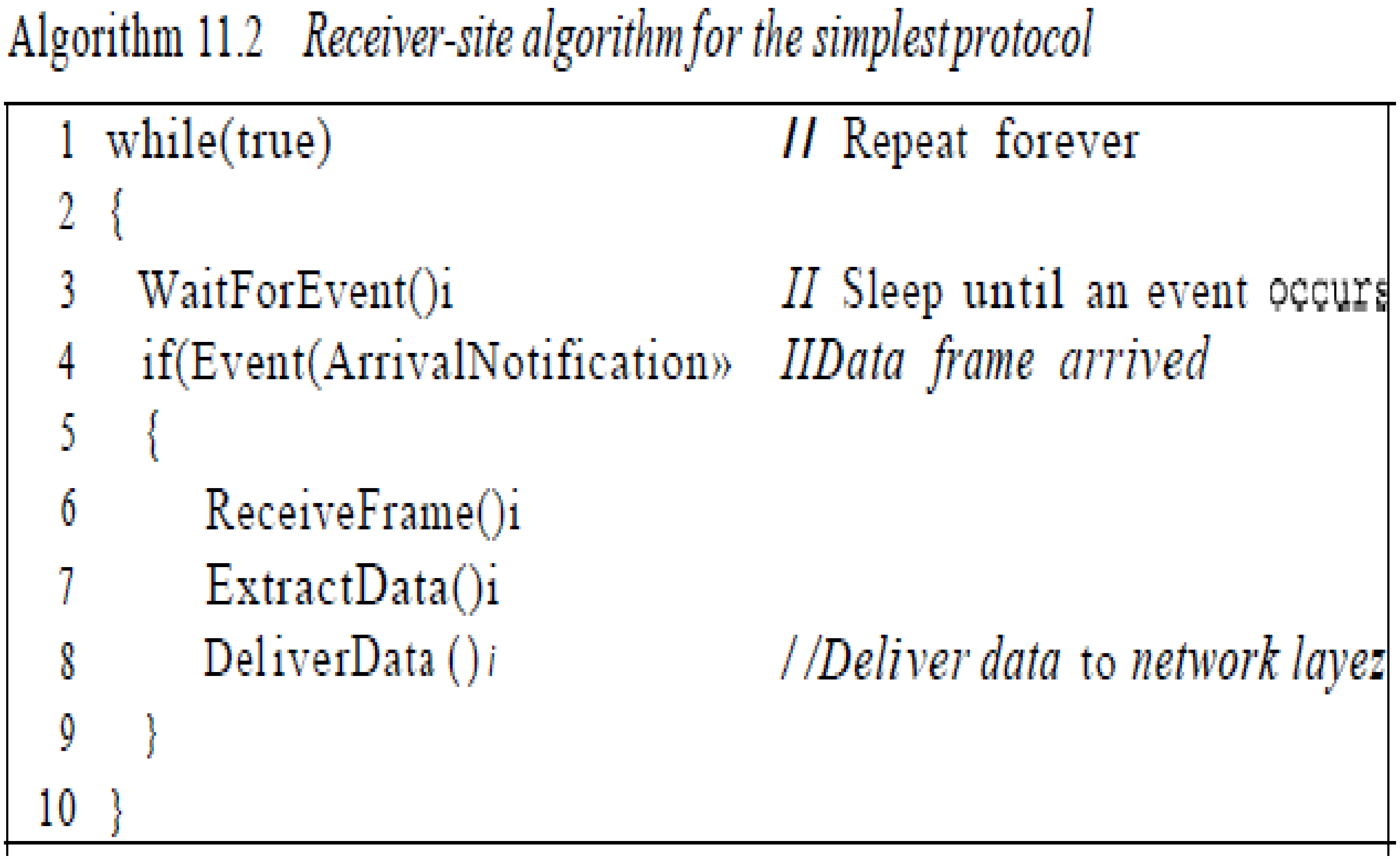
Simplest Protocol



Algorithm 11.1 *Sender-site algorithm for the simplestprotocol*

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

Simplest Protocol



Algorithm 11.2 *Receiver-site algorithm for the simplestprotocol*

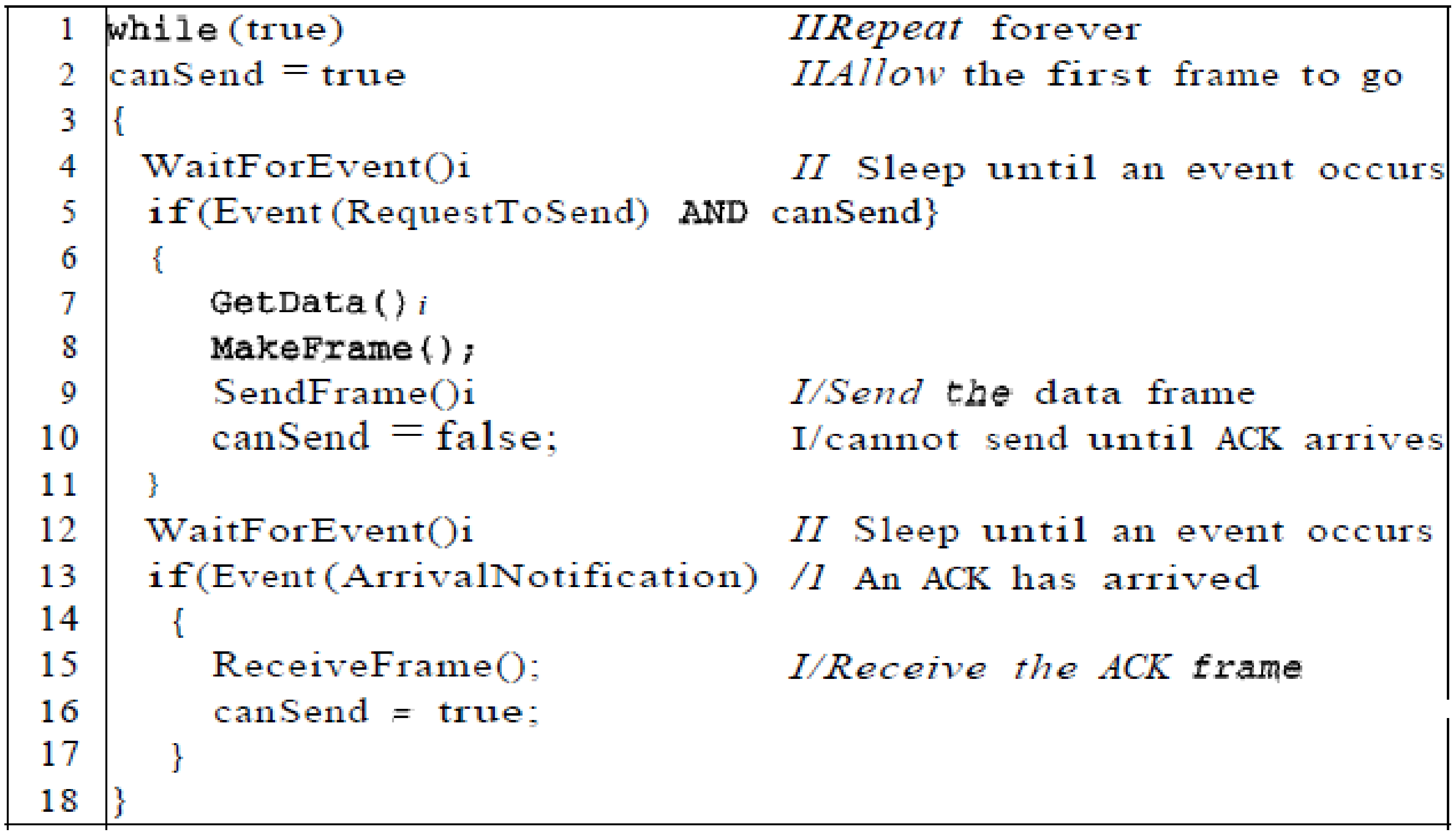
```
while(true)                              // Repeat  forever
{
  WaitForEvent()i                        II Sleep until an event occurs
  if(Event(ArrivalNotification»          IIData  frame  arrived
  {
      ReceiveFrame()i
      ExtractData()i
      DeliverData ()i                    / /Deliver data to network layez
  }
}
```

Stop-and-Wait Protocol

- If data frames arrive at the receiver site faster than they can be processed, the frames must be stored until their use.
- To prevent the receiver from becoming overwhelmed with frames, we need to tell the sender to slow down.
- In this protocol the sender sends one frame, stops until it receives confirmation from the receiver (okay to go ahead), and then sends the next frame.

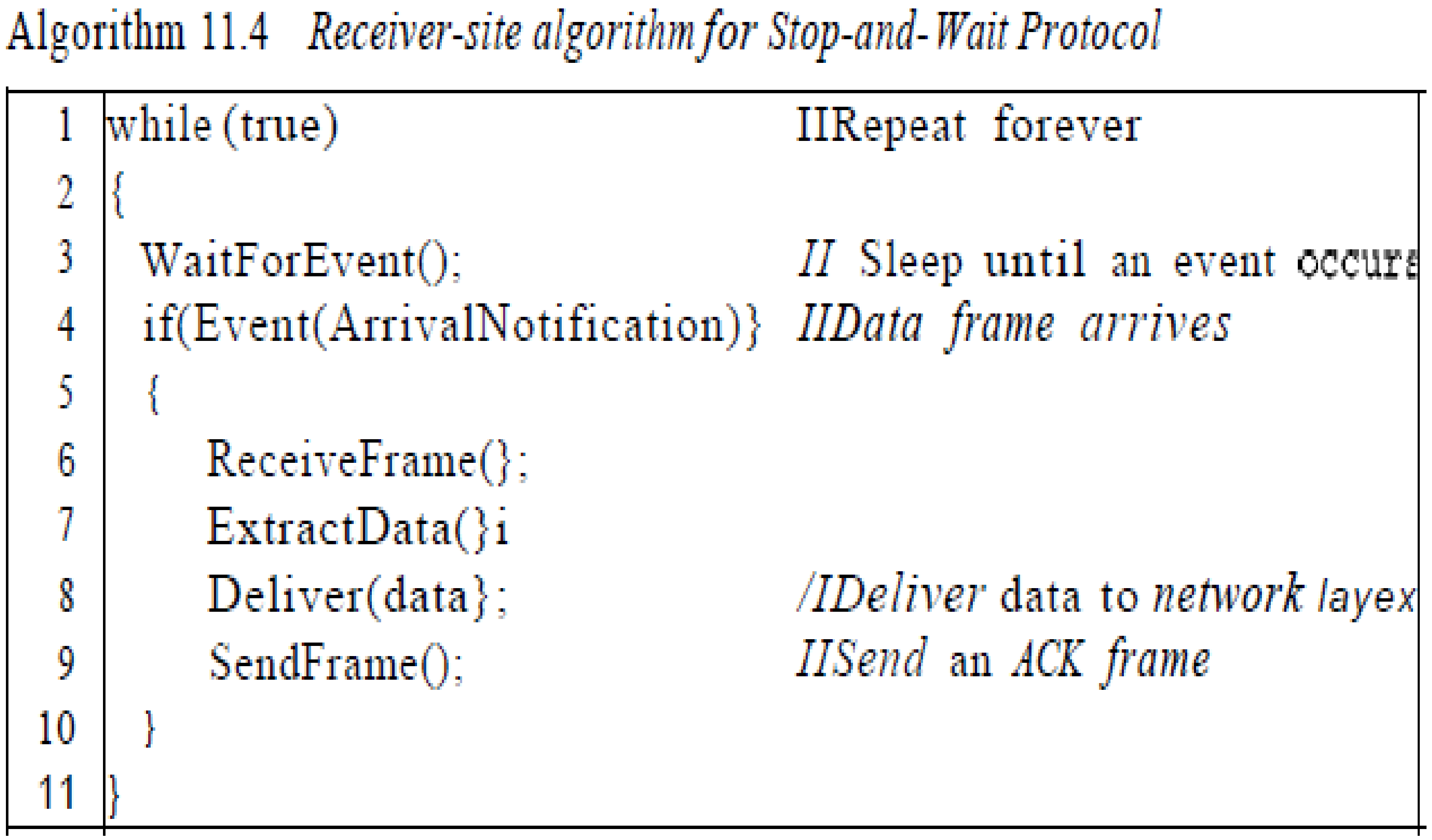
Stop-and-Wait Protocol

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



```
while(true)                                  IIRepeat forever
canSend = true                               IIAllow the first frame to go
{
  WaitForEvent()i                            II Sleep until an event occurs
  if(Event(RequestToSend) AND canSend}
  {
      GetData()i
      MakeFrame();
      SendFrame()i                           I/Send the data frame
      canSend = false;                       I/cannot send until ACK arrives
  }
  WaitForEvent()i                            II Sleep until an event occurs
  if(Event(ArrivalNotification)              /1 An ACK has arrived
   {
      ReceiveFrame();                        I/Receive the ACK frame
      canSend = true;
   }
}
```

Stop-and-Wait Protocol



Algorithm 11.4 *Receiver-site algorithm for Stop-and-Wait Protocol*

```
while (true)                              IIRepeat  forever
{
  WaitForEvent();                         II  Sleep until an event occur
  if(Event(ArrivalNotification)}          IIData frame arrives
  {
      ReceiveFrame():
      ExtractData()i
      Deliver(data}:                      /IDeliver data to network layex
      SendFrame():                        IISend an ACK frame
  }
}
```

NOISY CHANNELS

- Noiseless channels are nonexistent, so we need to add error control to our protocols.
- Stop-and-Wait Automatic Repeat Request protocol add 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.

NOISY CHANNELS

- We discuss three protocols in this section that use error control.
  - Stop & Wait Automatic Repeat Request
  - Go-Back-N Automatic Repeat Request
  - Selective Repeat Automatic Repeat Request

Stop-and-Wait ARQ

- Lost frames are more difficult to handle than corrupted ones.
- 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.

Stop-and-Wait ARQ

- The completed and lost frames need to be resent in this protocol.
- If the receiver does not respond when there is an error, how can the sender know which frame to resend?
- To remedy this problem, the sender keeps a copy of the sent frame. 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.

Stop-and-Wait ARQ

- Since an ACK frame can also be corrupted and lost, it too needs redundancy bits and a sequence number.
- The ACK frame for this protocol has a sequence number field.

Sequence Numbers

- A field is added to the data frame to hold the sequence number of that frame.
- The sequence numbers of course can wrap around. For example, if we decide that the field is m bits long, the sequence numbers start from 0, go to $2^m - 1$, and then are repeated.
- Let us reason out the range of sequence numbers we need. Assume we have used x as a sequence number; we only need to use x + 1 after that. There is no need for x + 2

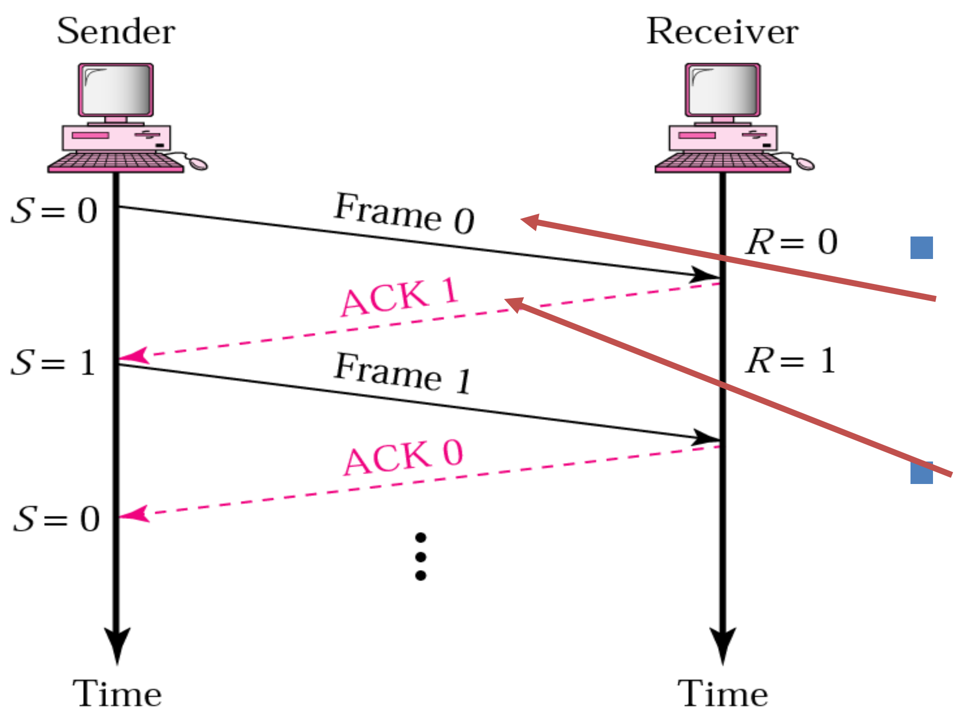
Acknowledgment Numbers

- The acknowledgment numbers always announce the sequence number of the next frame expected by the receiver.
- For example, if frame 0 has arrived safe and sound, the receiver sends an ACK frame with acknowledgment 1 (meaning frame 1 is expected next).
- If frame 1 has arrived safe and sound, the receiver sends an ACK frame with acknowledgment 0 (meaning frame 0 is expected).

Stop-and-Wait Automatic Repeat Request(Summary)

- Simplest flow and error control mechanism
- The sending device keeps a copy of the last frame transmitted until it receives an acknowledgement
  - identification of duplicate transmission (lost or delayed ACK)
- a damaged or lost frame is treated in the same way
- timers introduced
- positive ACK sent only for frames received safe & sound

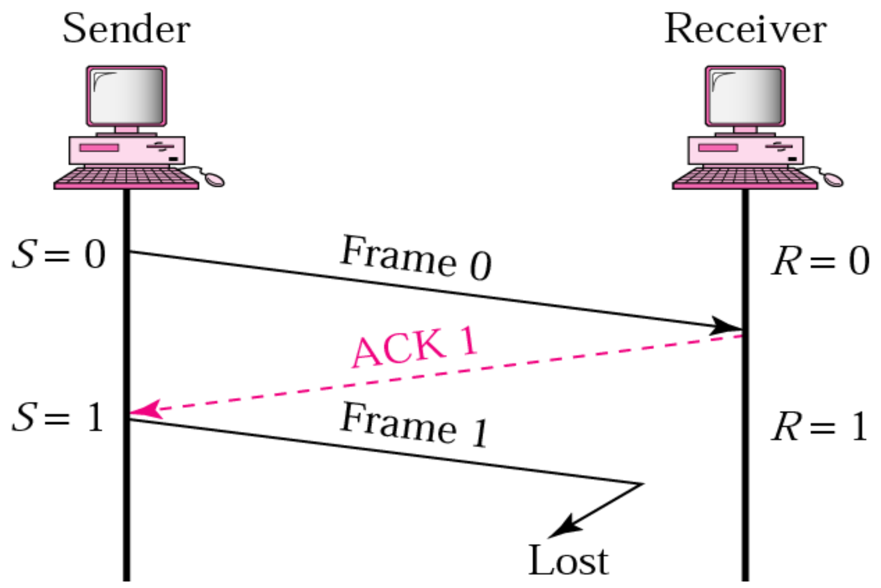
Stop-and-Wait ARQ
-Normal operation-



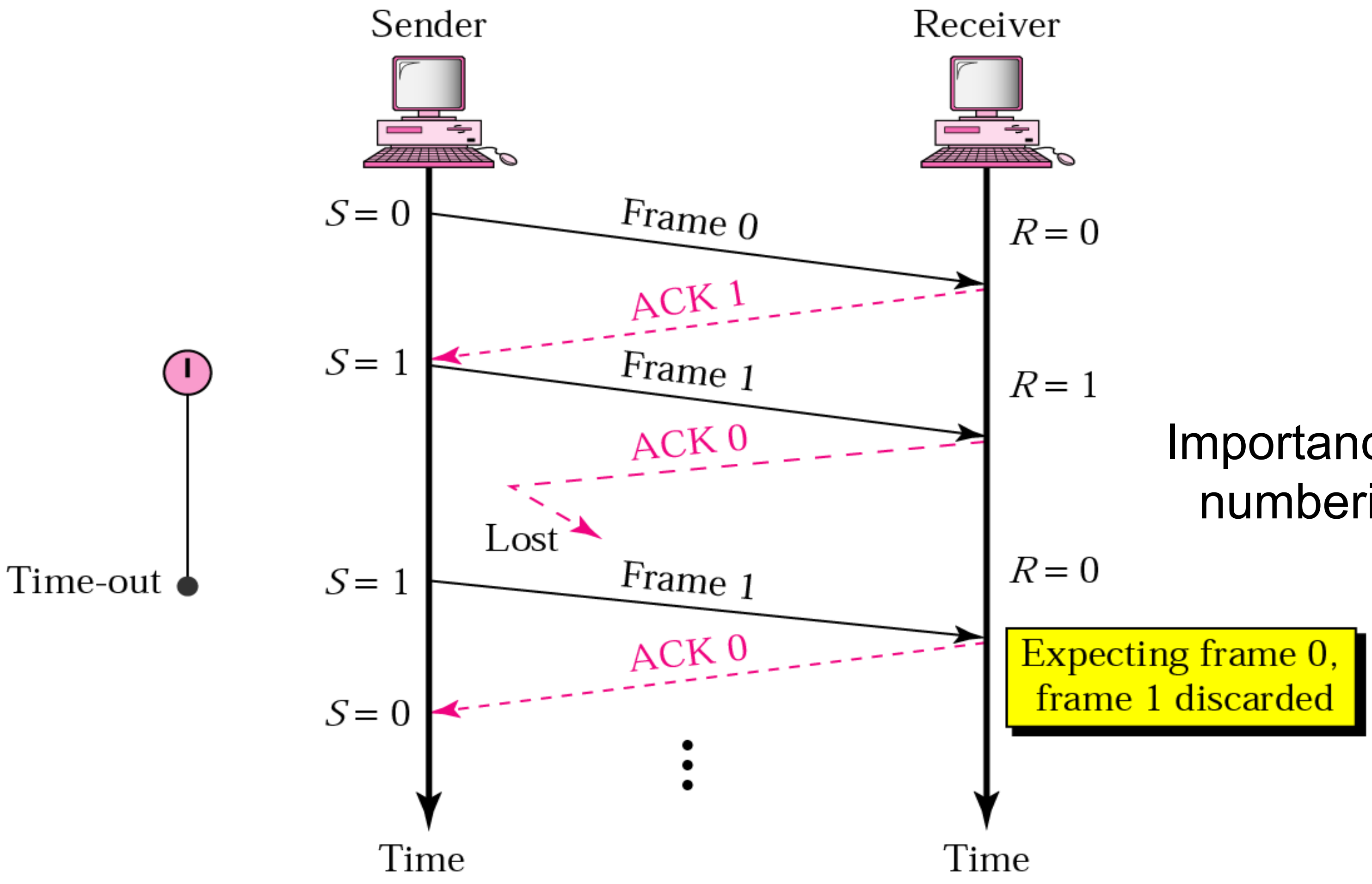
- The sender will not send the next piece of data until it is sure that the current one is correctly received
- sequence number necessary to check for duplicated packets
- No NACK – when packet is corrupted – duplicate ACKs instead

Stop-and-Wait ARQ
-Lost or damaged frame-

roundtrip delay
+
processing in the receiver

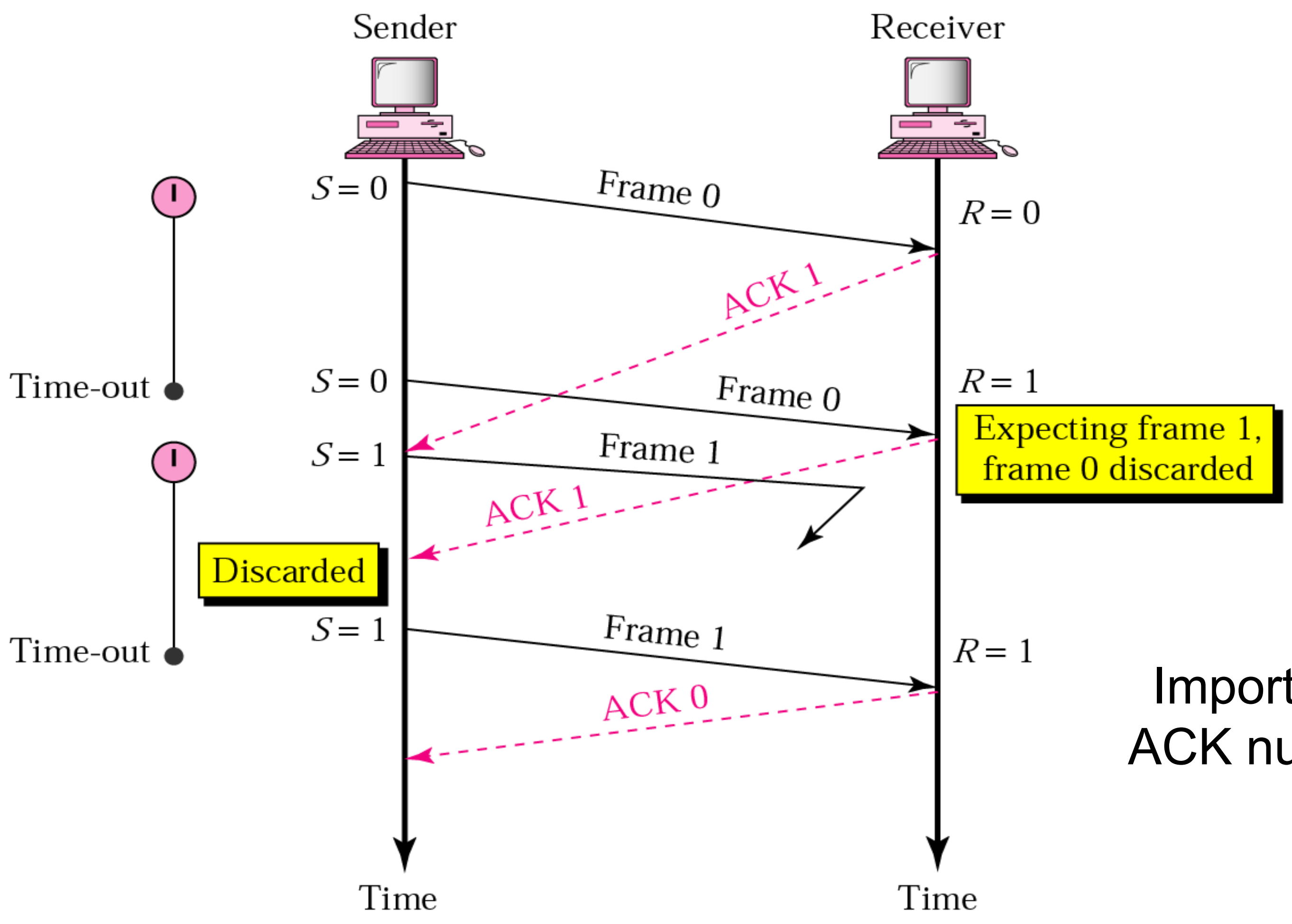


Stop-and-Wait ARQ
-Lost ACK-



Importance of numbering

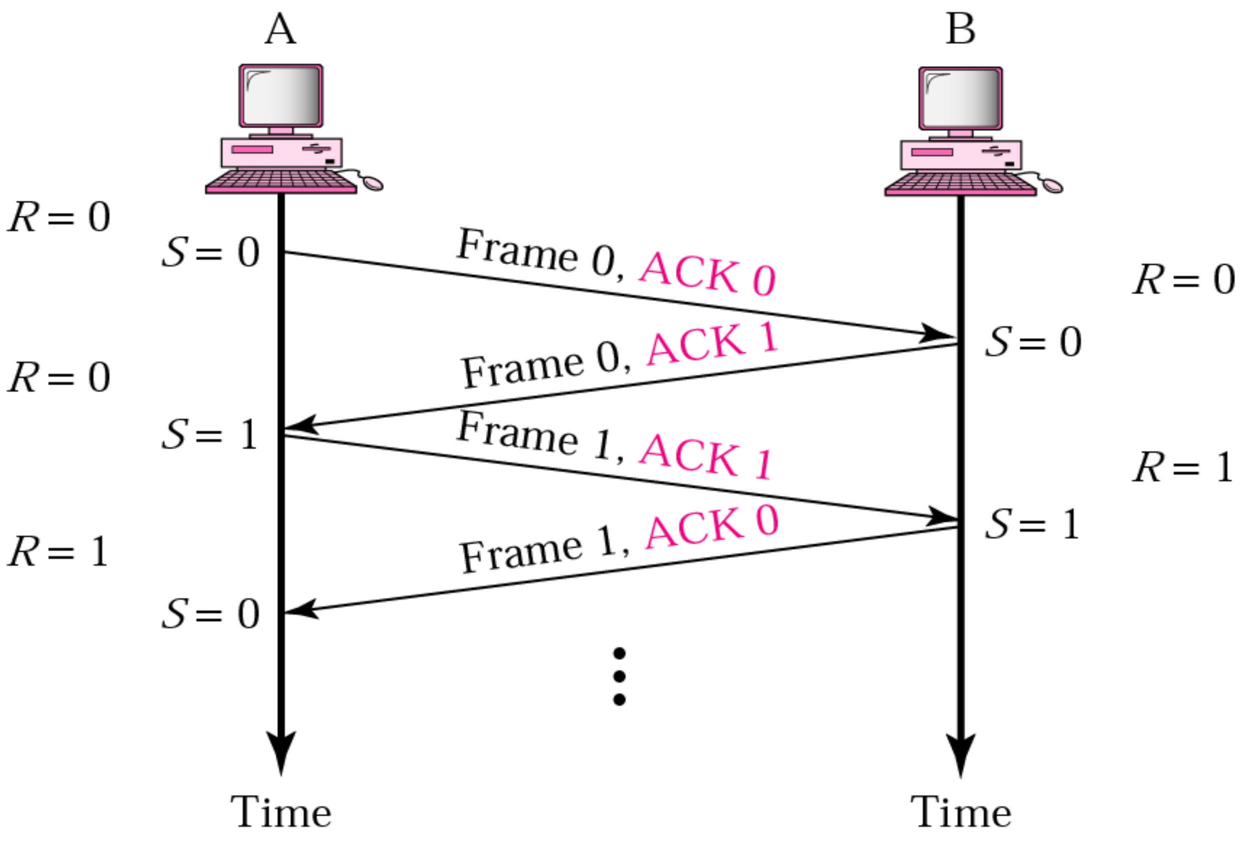
Stop-and-Wait ARQ
-Delayed ACK-



Importance of ACK numbering

Duplex Stop-and-Wait ARQ

- Piggybacking
  - combine data with ACK (less overhead saves bandwidth)



Drawbacks of Stop-and-Wait ARQ

- After each frame sent the host must wait for an ACK
  - inefficient use of bandwidth

- To improve efficiency ACK should be sent after multiple frames

- Alternatives: Sliding Window protocols
  1. Go-back-*N* ARQ
  2. Selective Repeat ARQ

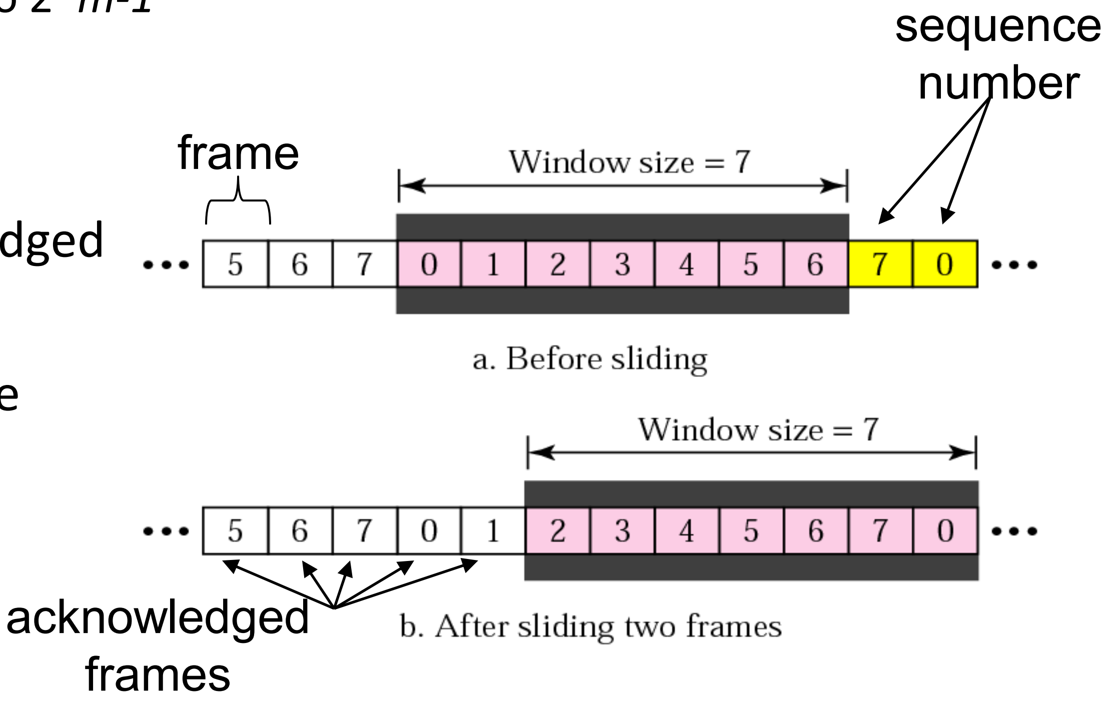
Pipelining

- One task begins before the other one ends
  - increases efficiency in transmission
- There is no pipelining in Stop-and-Wait ARQ

Sliding Window Protocols

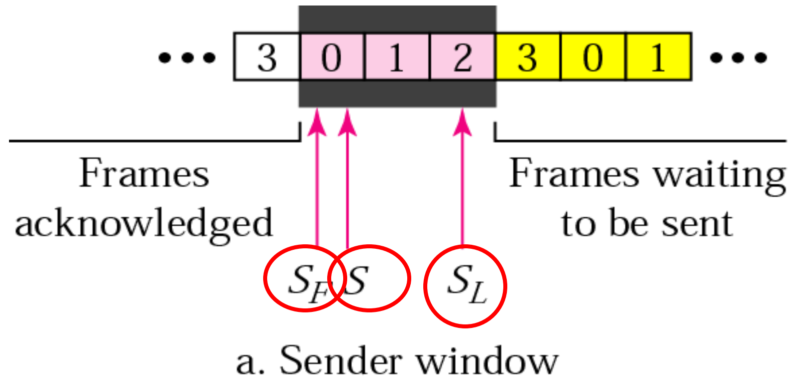
- Sequence numbers
  - sent frames are numbered sequentially
  - number of frames stored in the header
    - if the number of bits in the header is *m* than sequence number goes from 0 to 2^*m-1*

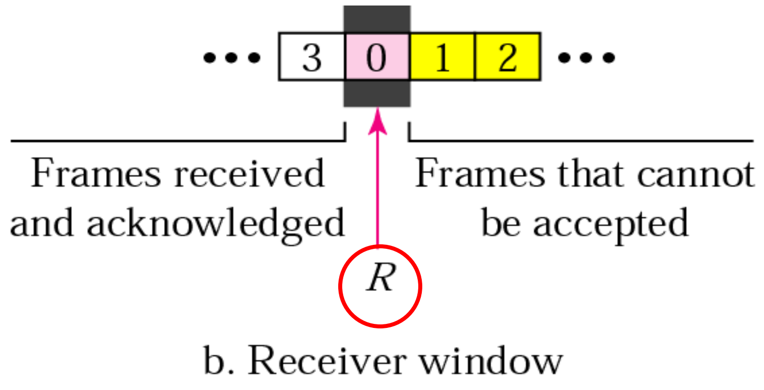
- Sliding window
  - to hold the unacknowledged outstanding frames
  - the receiver window size always 1



Go-back-*N*
-Control variables-

- *S*- holds the sequence number of the recently sent frame
- *SF* – holds sequence number of the first frame in the window
- *SL* – holds the sequence number of the last frame
- *R* – sequence number of the frame expected to be received





3
0
1
2
Frames received and acknowledged
Frames that cannot be accepted
R
b. Receiver window

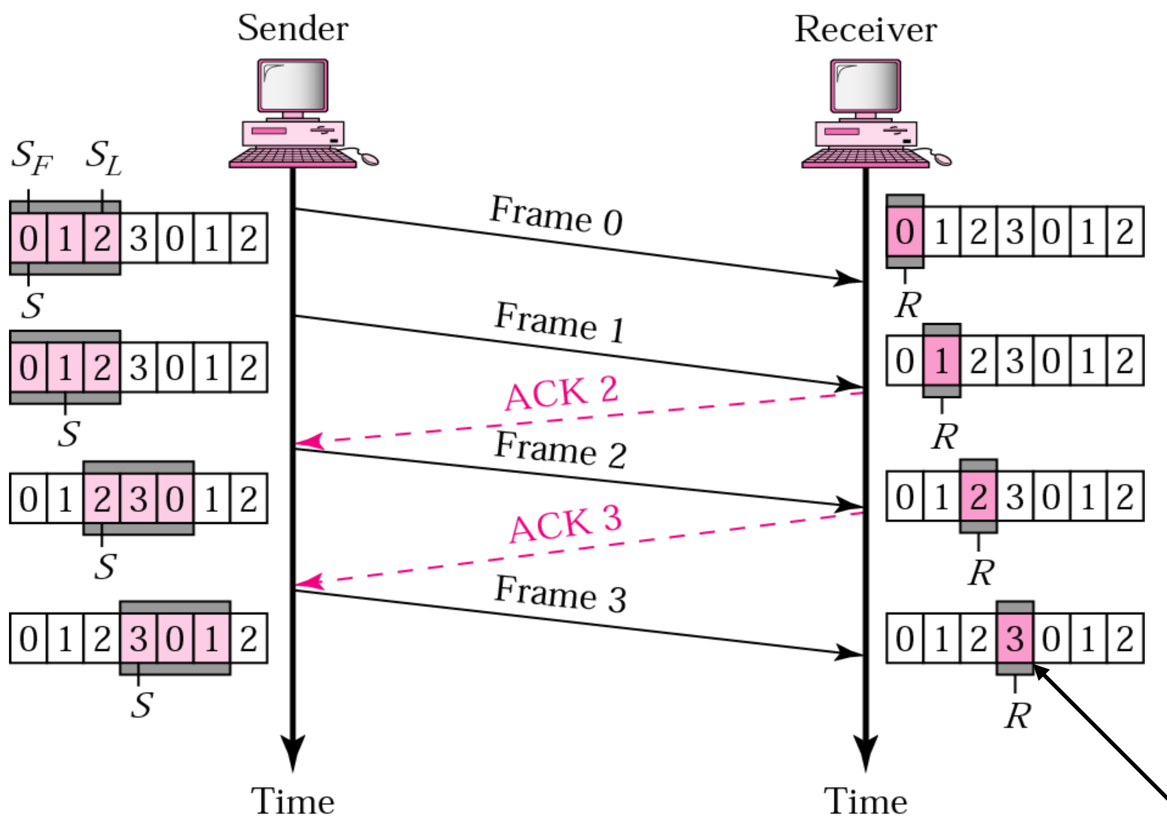
The name of Go-back-*N:* why?

- Re-sending frame
  - when the frame is damaged the sender **goes back** and sends a set of frames starting from the last one ACKn'd
  - the number of retransmitted frames is ***N***

- **Example:**

- The window size is 4.
- A sender has sent frame 6 and the timer expires for frame 3 (frame 3 not ACKn'd). The sender goes back and re-sends the frames 3, 4, 5 and 6.

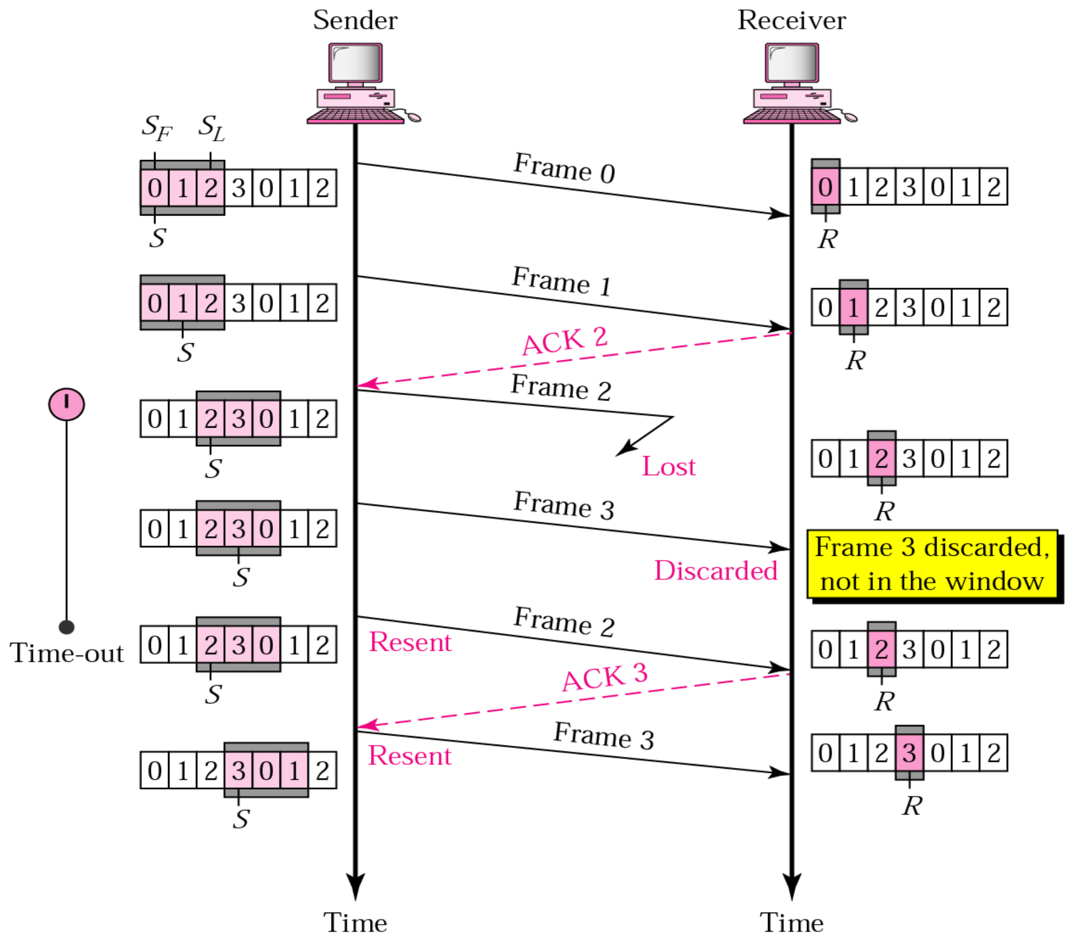
Go-back-*N*
-normal operation-



expected sequence number

- How many frames can be transmitted without acknowledgment?
- ACK1 – not necessary if ACK2 is sent
  - Cumulative ACK

Go-back-*N* -damaged or lost frame-

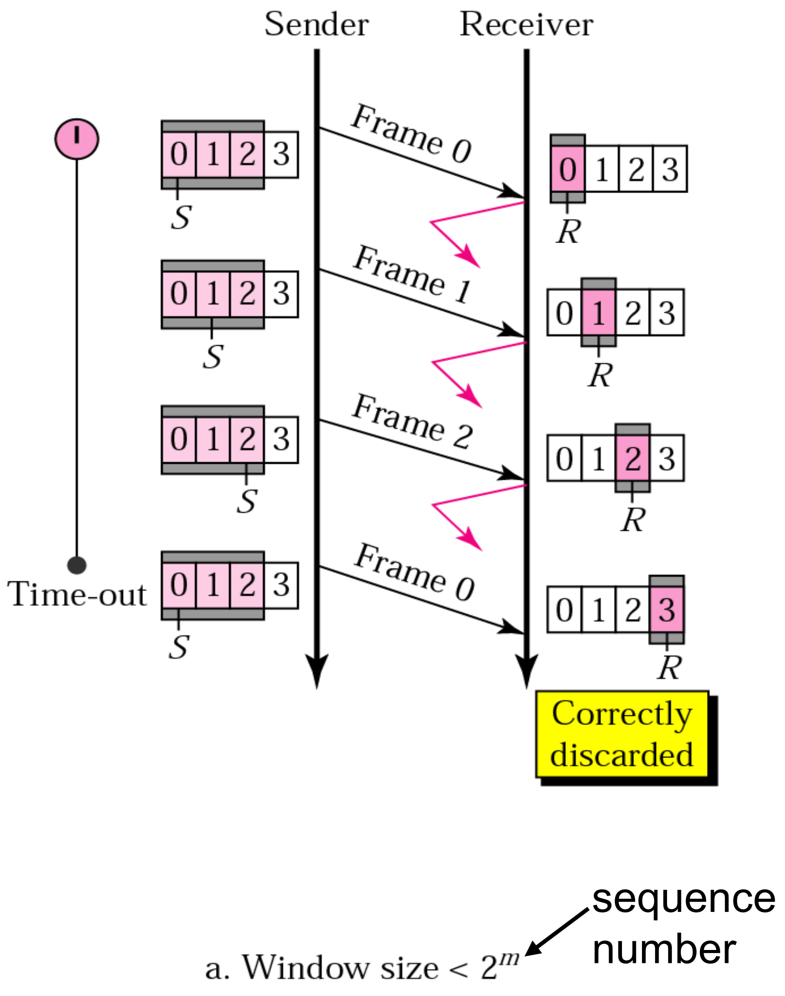


damaged frames are discarded!

Why are correctly received packets not buffered?

What is the disadvantage of this?

Go-back-*N*
-sender window size-



a. Window size < 2^m

Go-back-*N*

- Inefficient
  - all out of order received packets are discarded
- This is a problem in a noisy link
  - many frames must be retransmitted -> bandwidth consuming

- Solution
  - re-send only the damaged frames

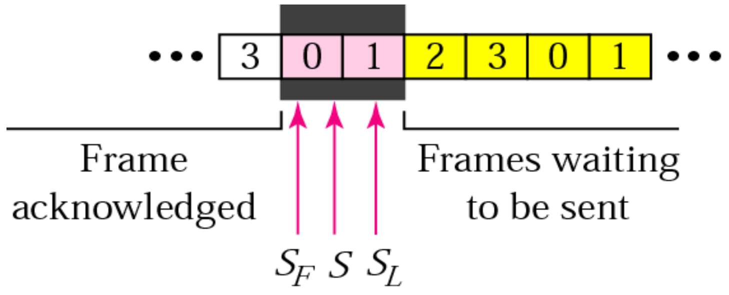
- Selective Repeat ARQ
  - avoid unnecessary retransmissions

Note

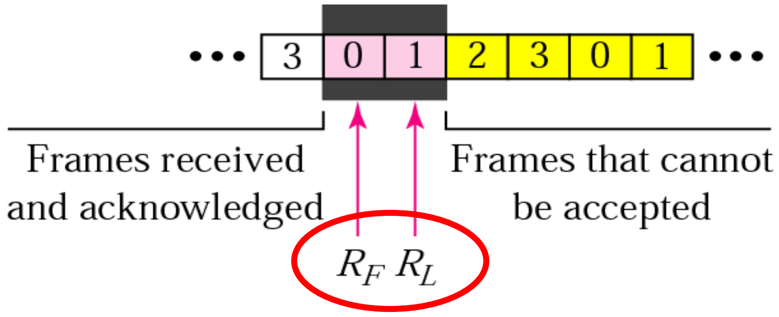
Stop-and-Wait ARQ is a special case of Go-Back-N ARQ in which the size of the send window is 1.

Selective Repeat ARQ

- Processing at the receiver more complex
- The window size is reduced to one half of 2^m
- Both the transmitter and the receiver have the same window size
- Receiver expects frames within the range of the sequence numbers



a. Sender window

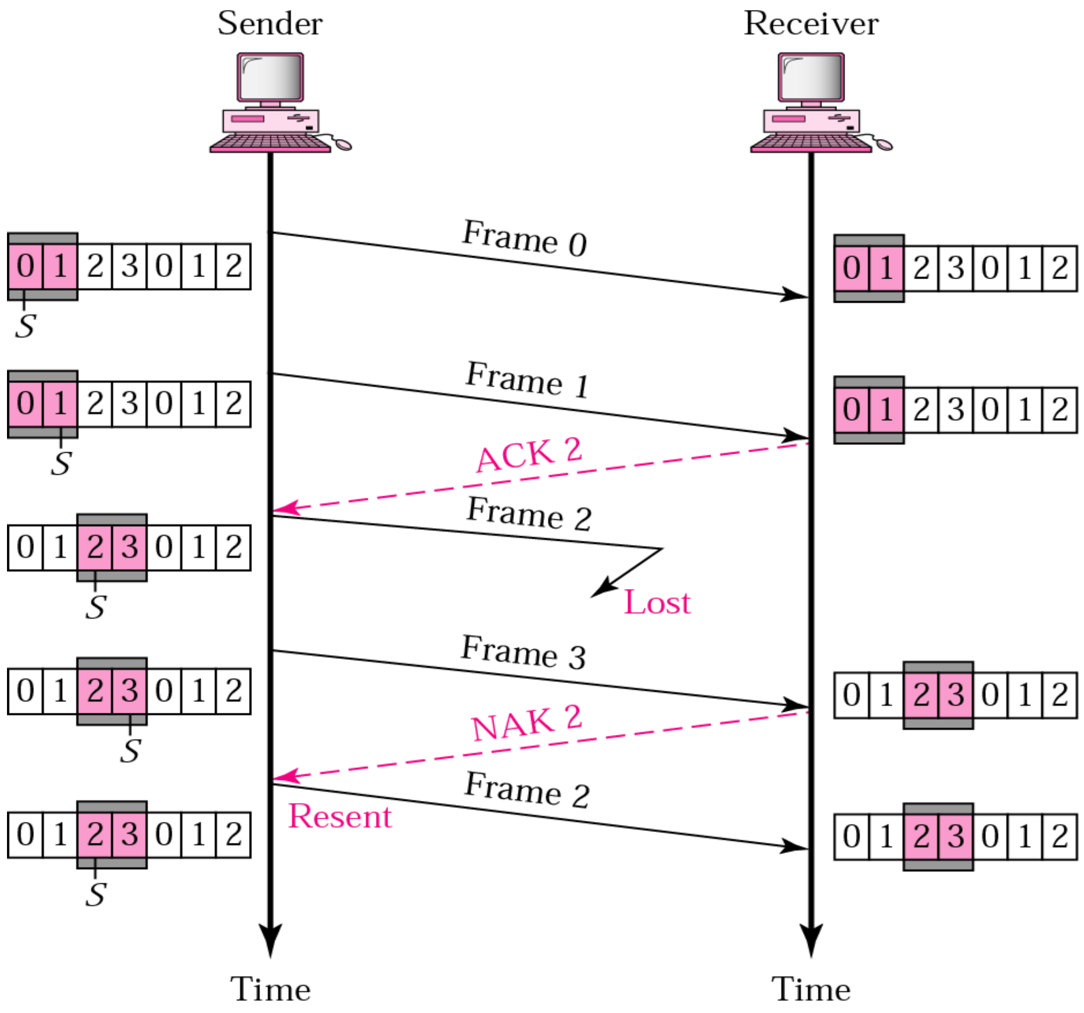


3
0
1
2
3
0
1
Frames received and acknowledged
Frames that cannot be accepted
R_F R_L

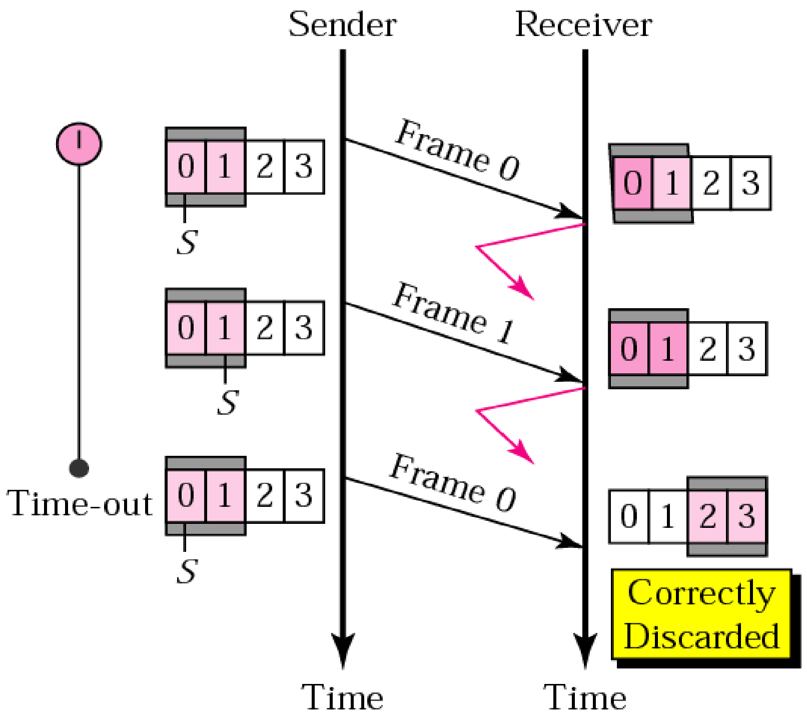
b. Receiver window

Selective Repeat ARQ
-lost frame-

Note:
retransmission
triggered with NACK
and not with
expired timer



Selective Repeat ARQ
-sender window size-



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

HDLC

- High-level Data Link Control (HDLC) is a bit-oriented protocol for communication over point-to-point and multipoint links.
- HDLC provides two common transfer modes that can be used in different configurations:
  - Normal response mode (NRM)
  - Asynchronous balanced mode (ABM)

Normal Response Mode

- In normal response mode (NRM), the station configuration is unbalanced.
- We have one primary station and multiple secondary stations. A primary station can send commands; a secondary station can only respond.
- The NRM is used for both point-to-point and multiple-point links.

Asynchronous Balanced Mode

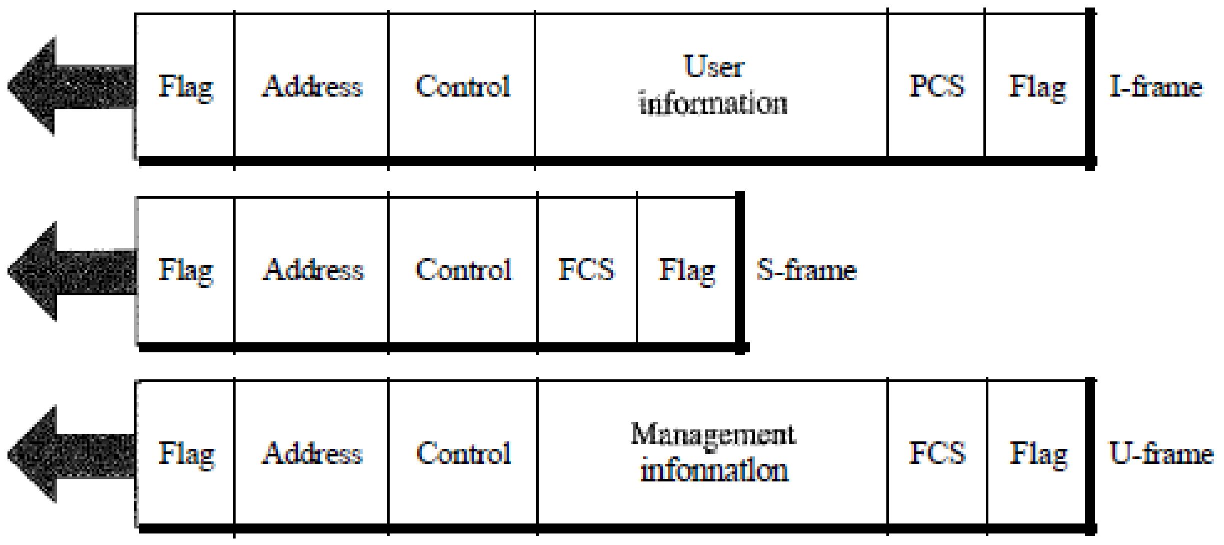
- In asynchronous balanced mode (ABM), the configuration is balanced. The link is point-to-point, and each station can function as a primary and a secondary

Frames

- HDLC defines three types of frames: information frames
- (I-frames), supervisory frames (S-frames), and unnumbered frames (V-frames).
- I-frames are used to transport user data and control information relating to user data .
- S-frames are used only to transport control information.
- V-frames are reserved for system management.

Frame Format

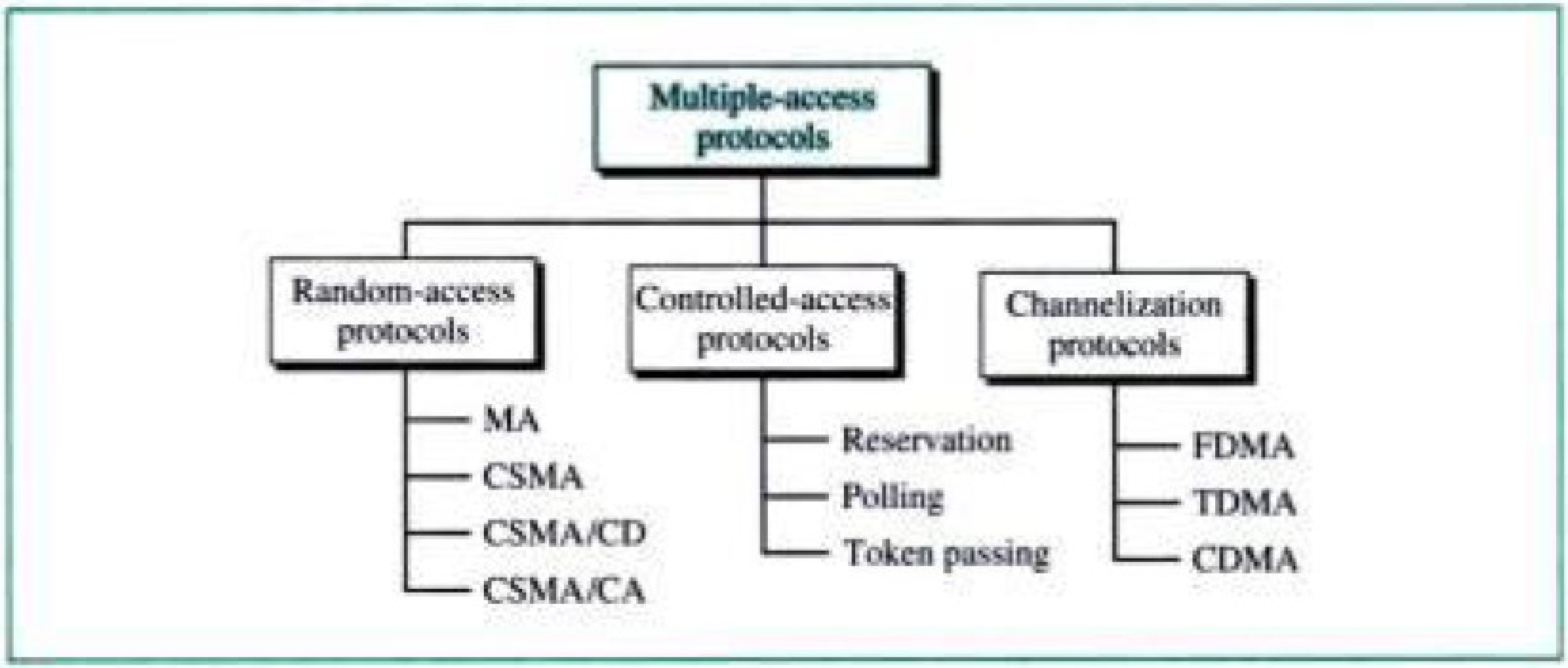
Figure 11.27 *HDLC frames*



Multiple Access (Chapter 12)

- In the protocols we described, we assumed that there is an available dedicated link (or channel) between the sender and the receiver.
- This assumption mayor may not be true.

Multiple Access



RANDOM ACCESS

- In random access no station is superior to another station and none is assigned the control over another.
- At each instance, a station that has data to send uses a procedure defined by the protocol to make a decision on whether or not to send.
- This decision depends on the state of the medium (idle or busy)
- 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, the earliest random access method
- The idea is that each station sends a frame whenever it has a frame to send. However, since there is only one channel to share, there is the possibility of collision.
- The pure ALOHA protocol relies on acknowledgments from the receiver. When a station sends a frame, it expects the receiver to send an acknowledgment. If the
- acknowledgment does not arrive after a time-out period, the station assumes that the frame has been destroyed and resends the frame.

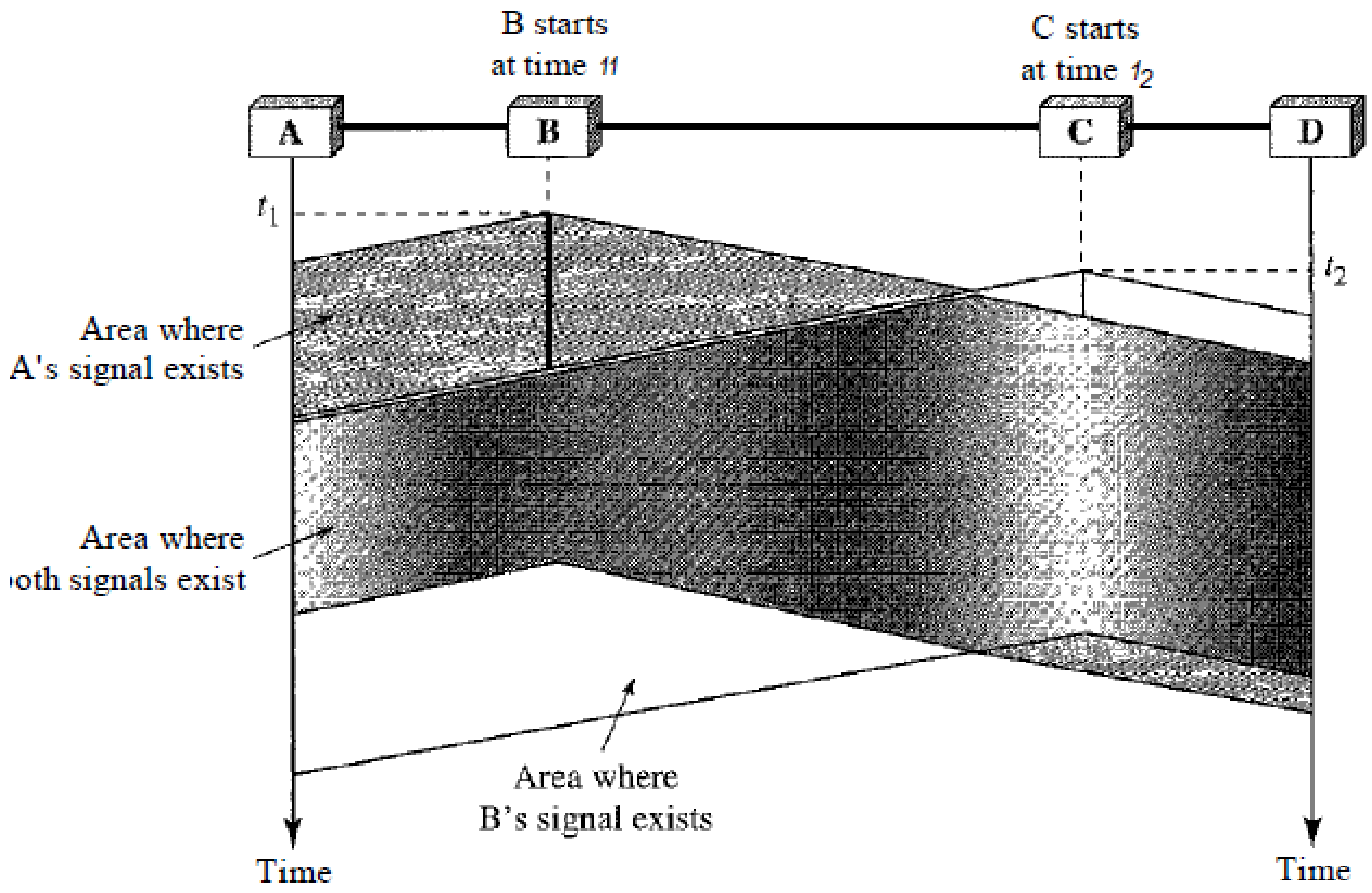
ALOHA

- Pure ALOHA dictates that when the time-out period passes, each station waits a random amount of time before resending its frame.
- The randomness will help avoid more collisions. We call this time the back-off time

Carrier Sense Multiple Access (CSMA)

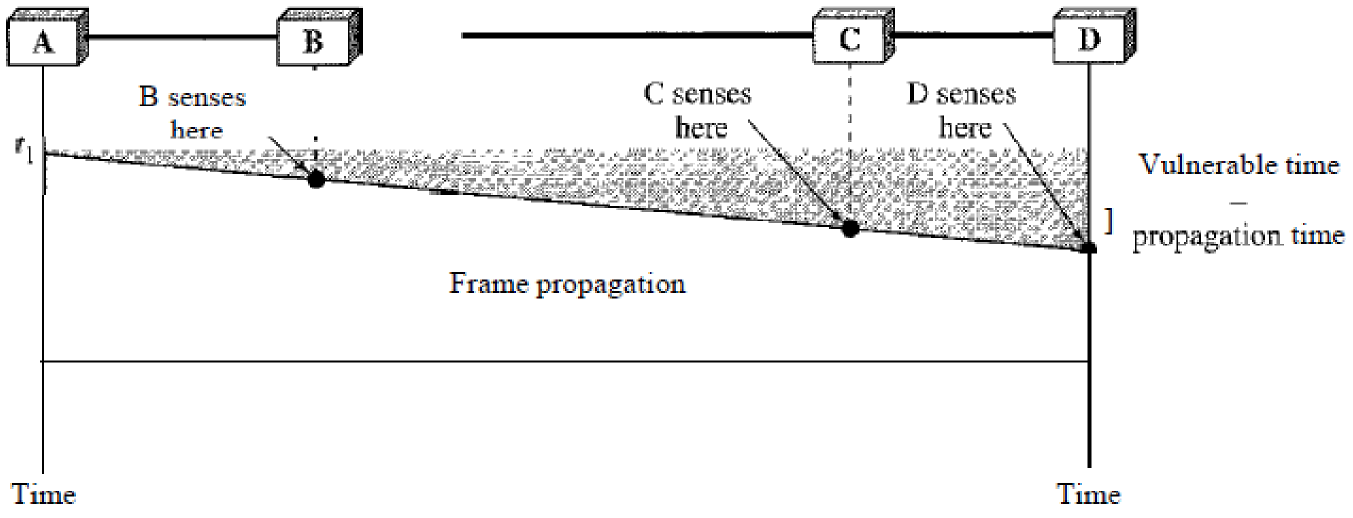
- To minimize the chance of collision and, therefore, increase the performance, the CSMA method was developed.
- The chance of collision can be reduced if a station senses the medium before trying to use it.
- CSMA can reduce the possibility of collision, but it cannot eliminate it.

Carrier Sense Multiple Access (CSMA)



Vulnerable Time for CSMA

- The vulnerable time for CSMA is the propagation time Tp
- This is the time needed for a signal to propagate from one end of the medium to the other



Carrier Sense Multiple Access with Collision Detection (CSMA/CD)

- The CSMA method does not specify the procedure following a collision.
- Carrier sense multiple access with collision detection (CSMA/CD) augments the algorithm to handle the collision.

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
- Reservation
  - a station needs to make a reservation before sending data.
- Polling
  - its works with topologies in which one device is designated as a primary station.
  - All data exchanges must be made through the primary device even when the ultimate destination is a secondary device

CONTROLLED ACCESS

- Token Passing
  - The stations in a network are organized in a logical ring.
  - a special packet called a token circulates through the ring. The possession of the token gives the station the right to access the channel and send its data.

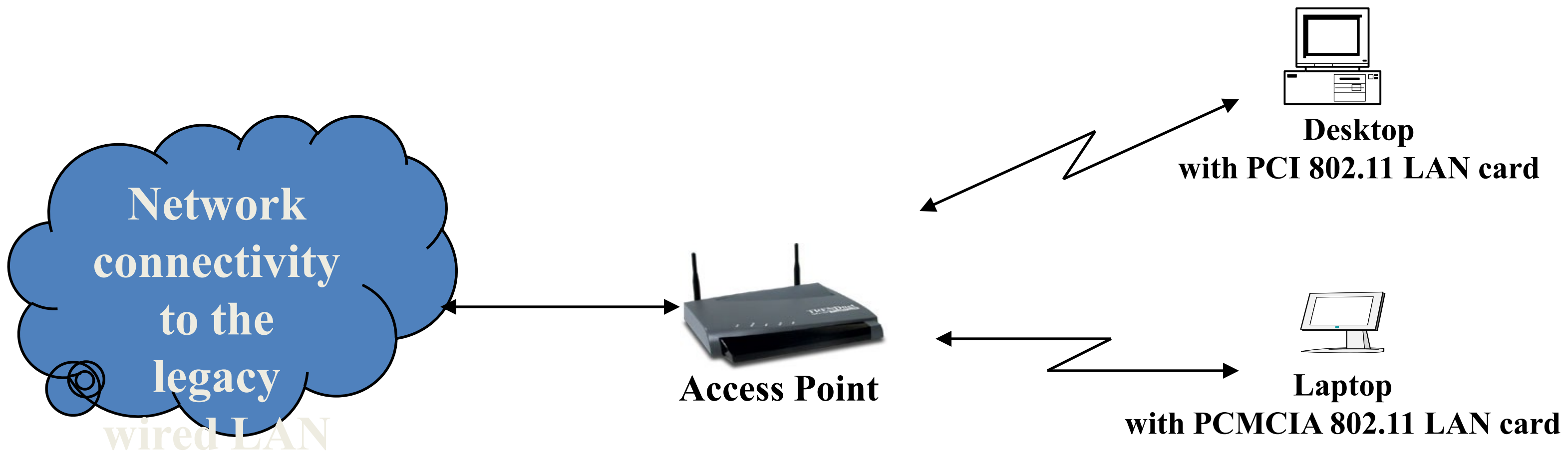
CHANNELIZATION

- 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.
- Frequency-Division Multiple Access (FDMA)
  - Each station is allocated a band to send its data.
- Time-Division Multiple Access (TDMA)
  - Each station is allocated a time slot during which it can send data.

CHANNELIZATION

- Code-Division Multiple Access (CDMA)
  - It differs from TDMA because all stations can send data simultaneously; there is no timesharing.
  - CDMA simply means communication with different codes

802.11 Wireless LAN

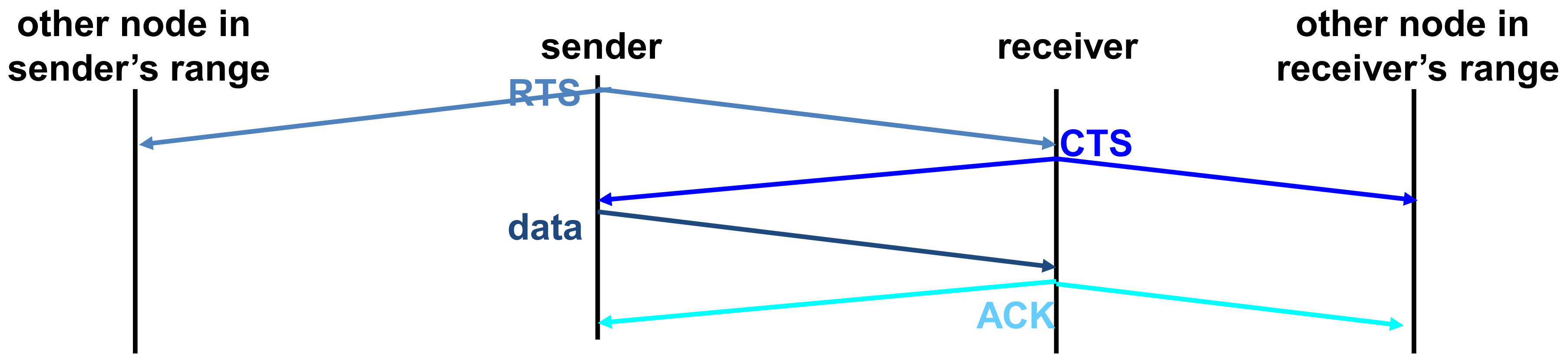

Network connectivity to the legacy wired LAN
Access Point
Desktop
with PCI 802.11 LAN card
Laptop
with PCMCIA 802.11 LAN card

- Provides network connectivity over wireless media
- An Access Point (AP) is installed to act as Bridge between Wireless and Wired Network
- The AP is connected to wired network and is equipped with antennae to provide wireless connectivity

802.11 Wireless LAN

- Range (Distance between Access Point and WLAN client) depends on structural hindrances and RF gain of the antenna at the Access Point
- To service larger areas, multiple APs may be installed with a 20-30% overlap
- A client is always associated with one AP and when the client moves closer to another AP, it associates with the new AP (Hand-Off)
- Three flavors:
  - 802.11b
  - 802.11a
  - 802.11g

Multiple Access with Collision Avoidance (MACA)



- **Before every data transmission**
  - Sender sends a Request to Send (RTS) frame containing the length of the transmission
  - Receiver respond with a Clear to Send (CTS) frame
  - Sender sends data
  - Receiver sends an ACK; now another sender can send data
- **When sender doesn't get a CTS back, it assumes collision**

WLAN : 802.11b

- The most popular 802.11 standard currently in deployment.
- Supports 1, 2, 5.5 and 11 Mbps data rates in the 2.4 GHz ISM (Industrial-Scientific-Medical) band

WLAN : 802.11a

- Operates in the 5 GHz UNII (Unlicensed National Information Infrastructure) band
- Incompatible with devices operating in 2.4GHz
- Supports Data rates up to 54 Mbps.

WLAN : 802.11g

- Supports data rates as high as 54 Mbps on the 2.4 GHz band
- Provides backward compatibility with 802.11b equipment