Module III

Data Link layer does this all responsibilities,

Hop to Hop delivery

Error detection

Flow control

Noisy channels

Pg no: 267

Error detection and correction

- Data can be corrupted during the transmission
- Some application requires a mechanism for detection and correction
- Some applications can tolerate a small level of error (eg ; audio, video)

Types of errors

- Errors : unpredictable and predictable changes in transmission .
- Interference cause errors .
- 2 types

1. Single bit error : (1-0, or 0-1) , ie, 1 bit is changing . Least likely occuring type
2. Burst error : 2 or more bits are changing , most likely occuring type

:Figure 10.1 *Single-bit error*

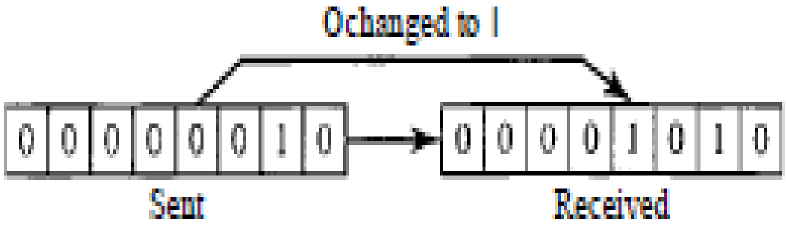
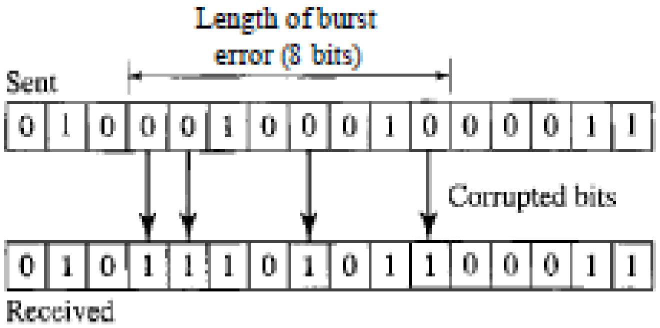


Figure 10.2 *Burst error oflength 8*



Redundancy

- The central concept in detecting or correcting errors is redundancy.
- To be able to detect or correct errors, we need to send some extra bits with our data. These redundant bits are added by the sender and removed by the receiver.
- Their presence allows the receiver to detect or correct corrupted bits.

Detection versus correction

- The correction of errors is more difficult than the detection.
- In error detection, we are looking only to see if any error has occurred. The answer is a simple yes or no.
- Not even interested in the number of errors. single-bit error is the same for us as a burst error.

- In error correction, we need to know the exact number of bits that are corrupted and more importantly, their location in the message.
- The number of the errors and the size of the message are important factors.

Forward Error Correction Versus Retransmission

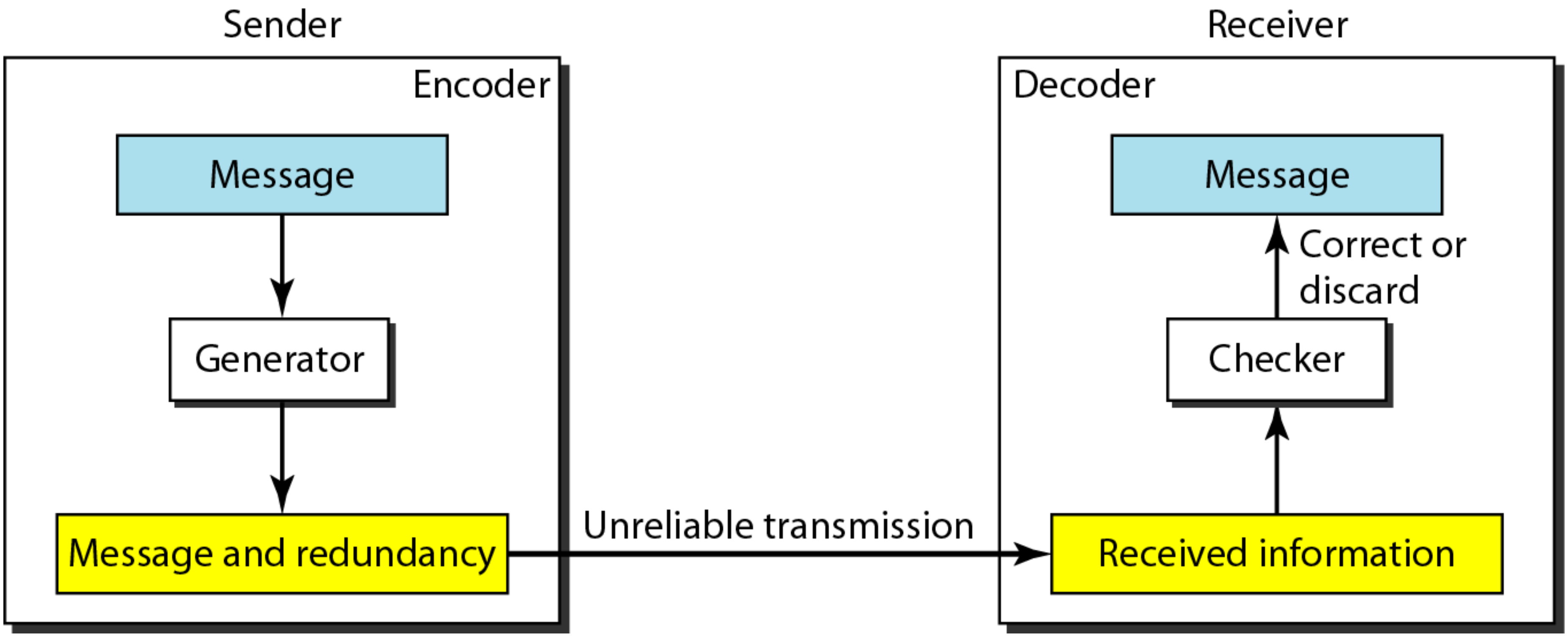
- Forward error correction is the process in which the receiver tries to guess the message by using redundant bits.
- This is possible, if the number of errors is small.
- Correction by retransmission is a technique in which the receiver detects the occurrence of an error and asks the sender to resend the message.
- Resending is repeated until a message arrives that the receiver believes is error-free (usually, not all errors can be detected).

Coding (encoding &decoding)

- Redundancy is achieved through various coding schemes.
- 2 types :

1. Block coding
2. Convolution coding

General idea of coding



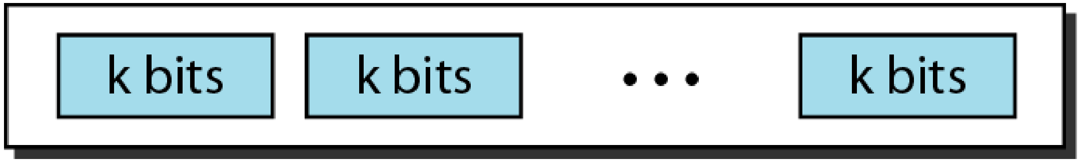
The structure of encoder and decoder

Block coding

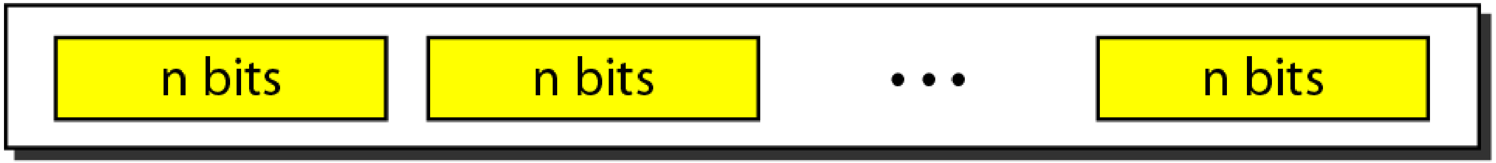
- In block coding, divide our message into blocks, each of k *bits, called* datawords*.We* add r *redundant bits to each block to make the length*

$n = k + r$*.The resulting n-bit blocks* are called codewords

Datawords and codewords in block coding



2^k Datawords, each of k bits



2^n Codewords, each of n bits (only 2^k of them are valid)

Example :

- Let’s assume k=10, r=2

so , n=10+2 =12

- So 2^k data words , 2^{10} = 1024 bits
- 2^n = 2^{12} =4096 bits

Ie, instead of 1024 bits we sends 4096 bits (3072 bits extra)

3072 bits either : used/other purpose/ unused

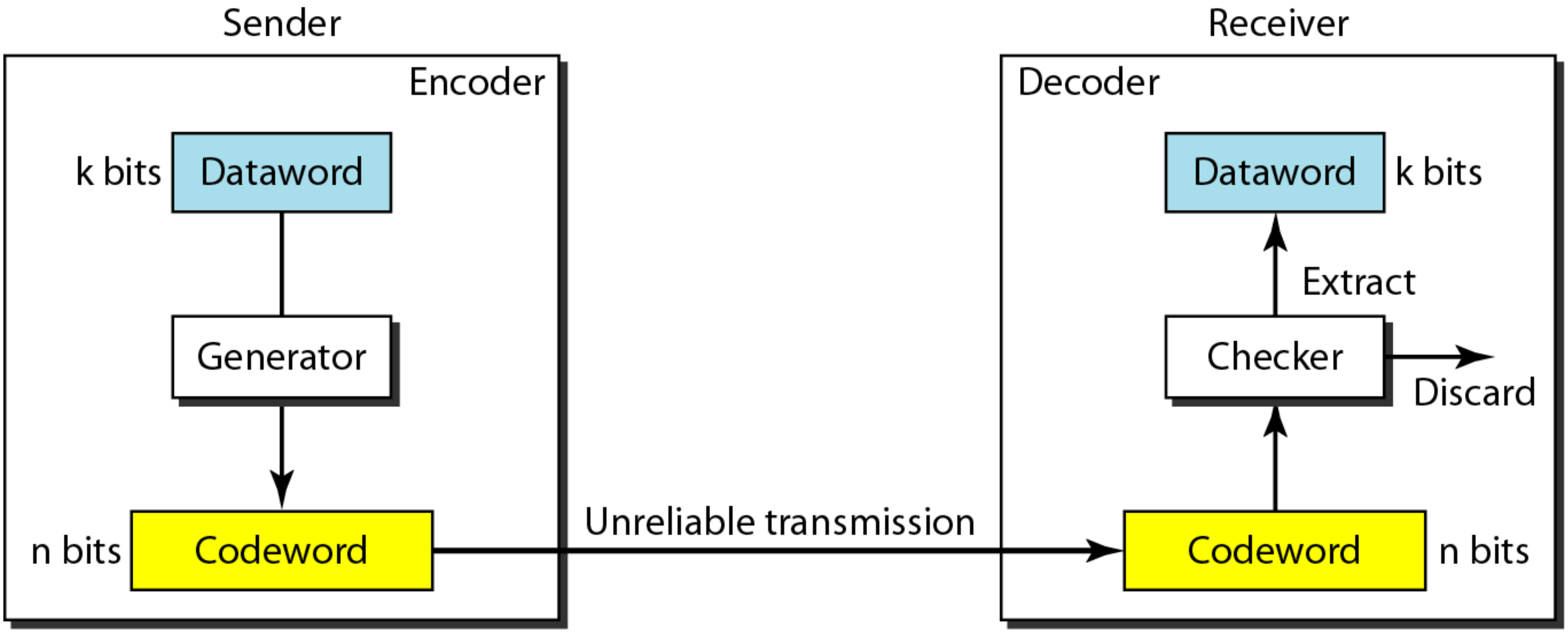
Error detection

- If the following conditions are met , the receiver can detect a change in original code word

1. The receiver has (or can find) a list of valid codeword
2. The original codeword has changed to a invalid one

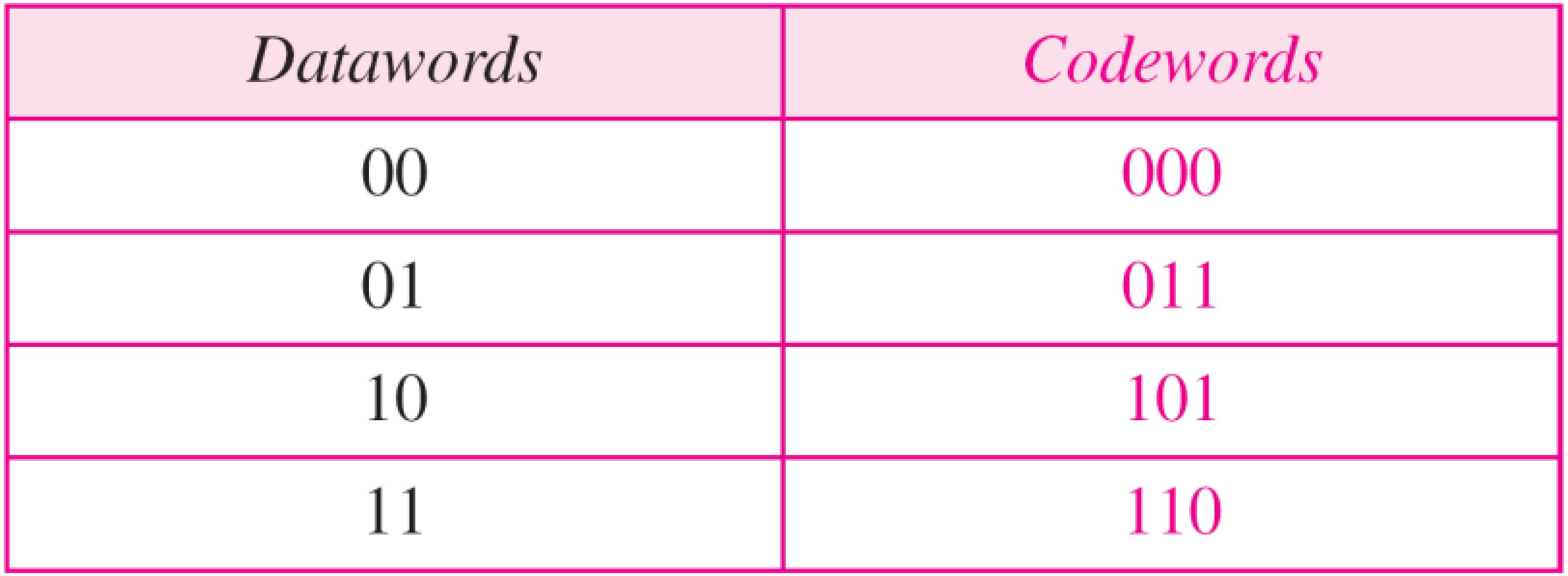
- Enough redundancy is added to detect an error.
- The receiver knows an error occurred but does not know which bit(s) is(are) in error.
- Has less overhead than error correction

Process of error detection in block coding



Example

Let us assume that k = 2 and n = 3. Following Table shows the list of datawords and codewords.



Datawords	*Codewords*
00	000
01	011
10	101
11	110

A code for error detection

Assume the sender encodes the dataword 01 as 011 and sends it to the receiver. Consider the following cases:

1. *The receiver receives 011. It is a valid codeword. The receiver extracts the dataword 01 from it.*
2. *The codeword is corrupted during transmission, and 111 is received. This is not a valid codeword and is discarded.*
3. *The codeword is corrupted during transmission, and 000 is received. This is a valid codeword. The receiver incorrectly extracts the dataword 00. Two corrupted bits have made the error undetectable.*

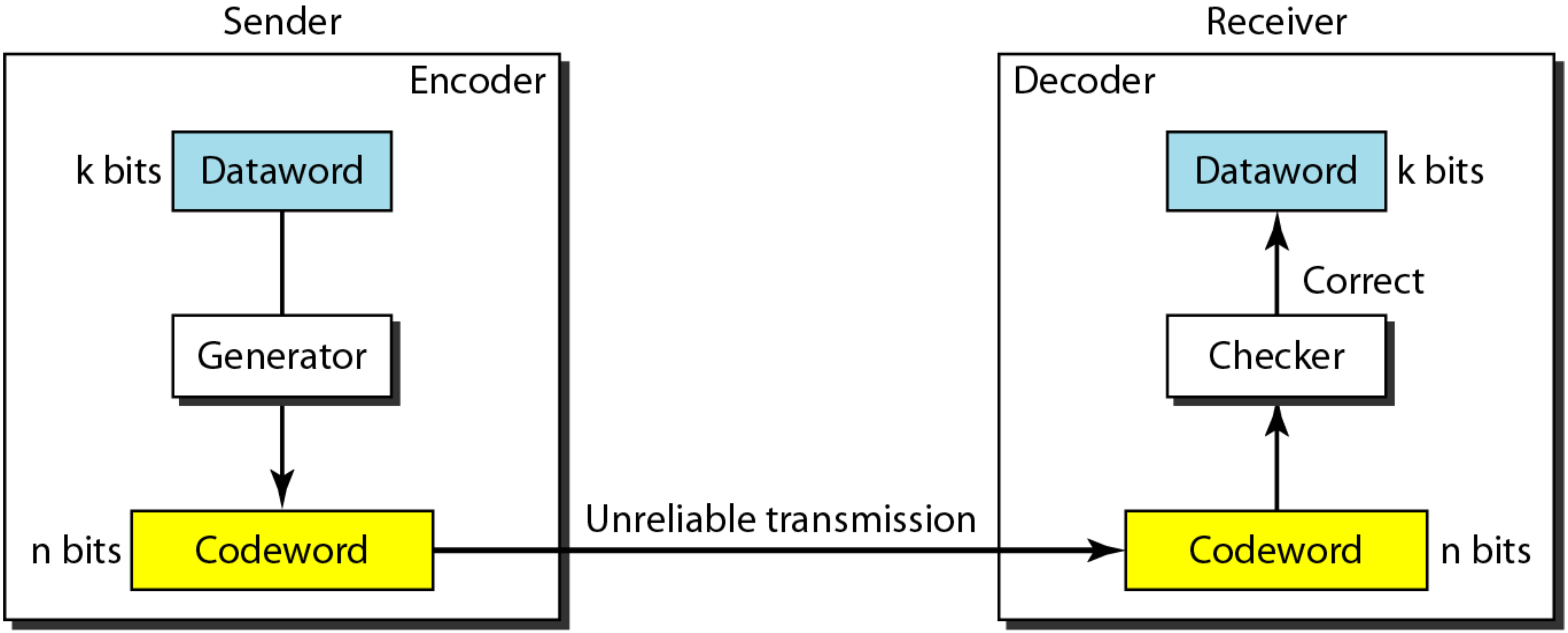
Note

An error-detecting code can detect only the types of errors for which it is designed; other types of errors may remain undetected.

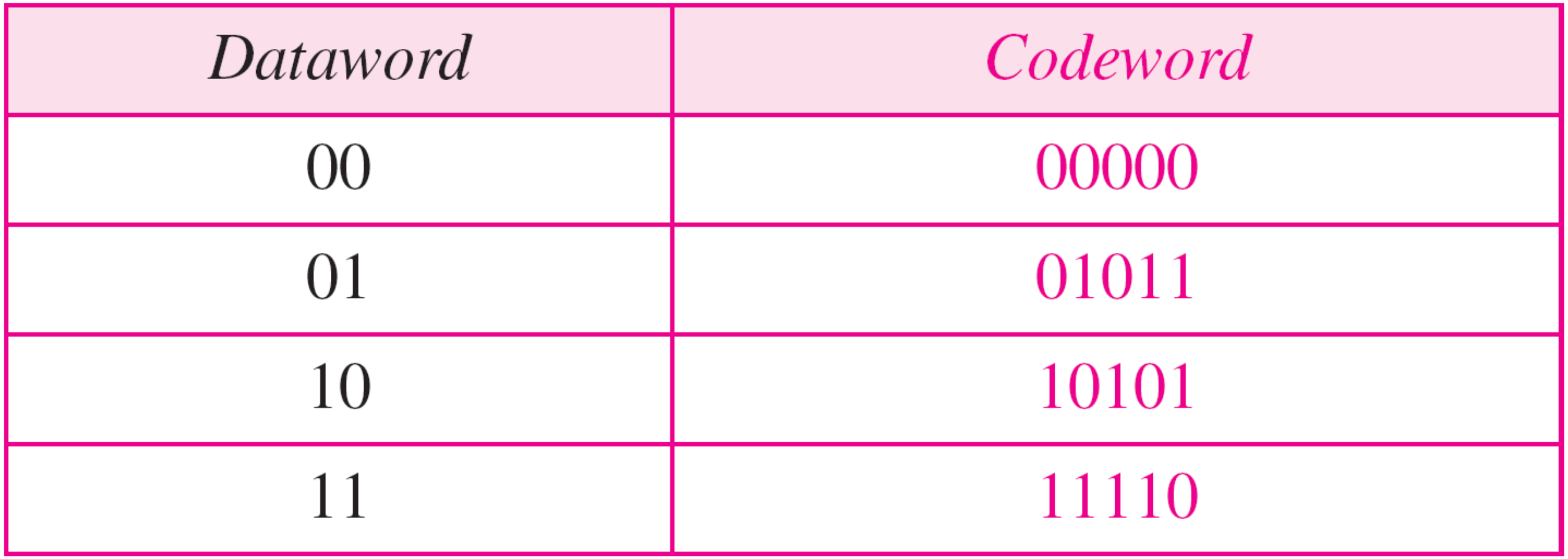
Error correction .

- Error correction is much more difficult than error detection.
- In error detection, the receiver needs to know only that the received codeword is invalid;
- In error correction the receiver needs to find (or guess) the original codeword sent.
- We can see that the idea is the same as error detection but the checker functions are much more complex.

Structure of encoder and decoder in error correction



A code for error correction



Dataword	*Codeword*
00	00000
01	01011
10	10101
11	11110

Example

The sender creates the codeword 01011.
The codeword is corrupted during transmission, and 01001 is received.

- First, the receiver finds that the received codeword is not in the table. This means an error has occurred.

- The receiver, assuming that there is only 1 bit corrupted, uses the following strategy to guess the correct dataword.

1. *Comparing the received codeword with the first codeword in the table (01001 versus 00000), the receiver decides that the first codeword is not the one that was sent because there are two different bits.*

2. *By the same reasoning, the original codeword cannot be the third or fourth one in the table.*

3. *The original codeword must be the second one in the table because this is the only one that differs from the received codeword by 1 bit. The receiver replaces 01001 with 01011 and consults the table to find the dataword 01.*

Framing : Data link layer

- 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 destination address defines where the packet is to go; the sender address helps the recipient acknowledge the receipt.

- The destination address defines where the packet is to go;
- the sender address helps the recipient acknowledge the receipt.
- When a message is carried in one very large frame, even a single-bit error would require the retransmission of the whole message. When a message is divided into smaller frames, a single-bit error affects only that small frame

- 2 types framing

1. Fixed size framing
2. Variable size framing

- **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. Example : 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.

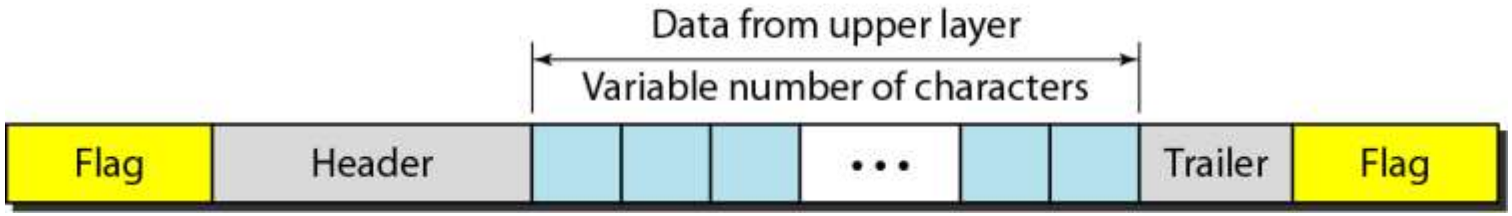
Two approaches were used

1. Character-oriented approach
2. Bit-oriented approach

- a character-oriented protocol, data to be carried are 8- bit characters from a coding system such as ASCII.

- **The header**, which normally carries the source and destination addresses and other control information.
- **The trailer**, which carries error detection or error correction redundant bits, are also multiples of 8 bits.
- **The flag**, composed of protocol-dependent special characters. To separate one frame from the next, an 8-bit (1 byte) flag is added at the beginning and the end of a frame.

A frame in a character-oriented protocol



- In character-oriented framing, only text was exchanged by the data link layers.
- The flag could be selected to be any character not used for text communication.
- Any pattern used for the flag could also be part of the information. If this happens, the receiver, when it encounters this pattern in the middle of the data, thinks it has reached the end of the frame. To solve this problem, a **byte-stuffing** strategy was added.

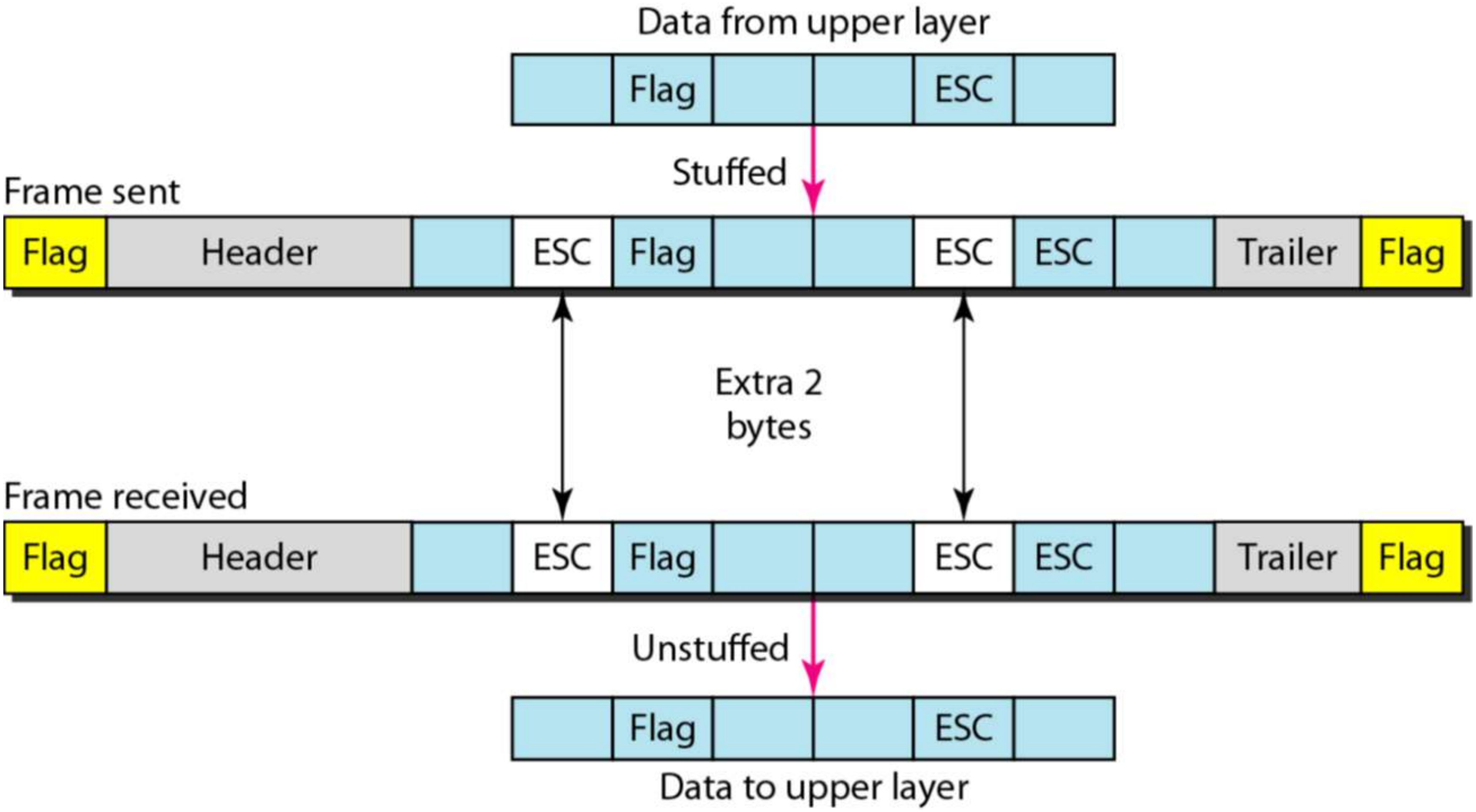
Note

Byte stuffing is the process of adding 1 extra byte whenever there is a flag or escape character (ESC) in the text.

- This extra byte is usually called the escape character (ESC), which has a predefined bit pattern.
- Whenever the receiver encounters the ESC character, it removes it from the data section and treats the next character as data, not a delimiting flag.
- If the text contains one or more escape characters followed by a flag, it creates another problem. The receiver removes the escape character, but keeps the flag, which is incorrectly interpreted as the end of the frame.

- To solve this problem, if the escape character that is part of the text, an extra escape character is added to show that the second escape character is part of the text.

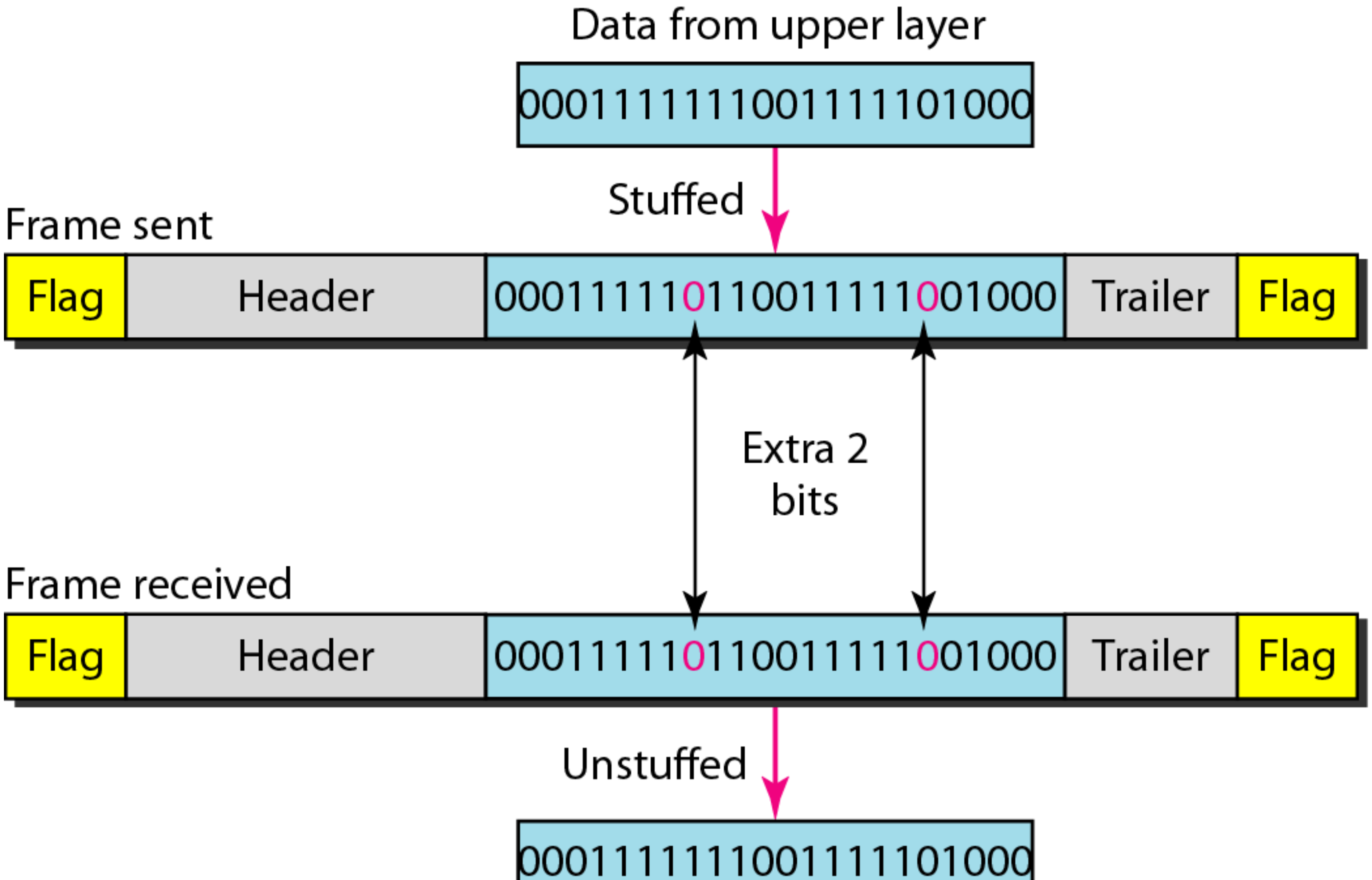
- Another problem of Character-oriented protocols is the universal coding systems, such as Unicode, have 16-bit and 32- bit characters that conflict with 8-bit characters.


Data from upper layer
Flag
ESC
Stuffed
Frame sent
Flag
Header
ESC
Flag
ESC
ESC
Trailer
Flag
Extra 2
bytes
Frame received
Flag
Header
ESC
Flag
ESC
ESC
Trailer
Flag
Unstuffed
Flag
ESC
Data to upper layer

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, and so on. In addition to headers, we still need a delimiter.
- 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 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.



FLOW AND ERROR 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.
- Each receiving device has a block of memory, called a *buffer, reserved for storing incoming* data until they are processed.
- If the buffer begins to fill up, the receiver must be able to tell the sender to halt transmission until it is once again able to receive.

Error Control

- 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.
- The term *error control* refers primarily to methods of error detection and retransmission.
- Any time an error is detected in an exchange, specified frames are retransmitted. This process is called Automatic Repeat Request (ARQ).
- Error control in the data link layer is based on ARQ , which is the retransmission of data.

PROTOCOLS

- Protocol : Rules for communication.
- The protocols are normally implemented in software by using one of the common programming languages.

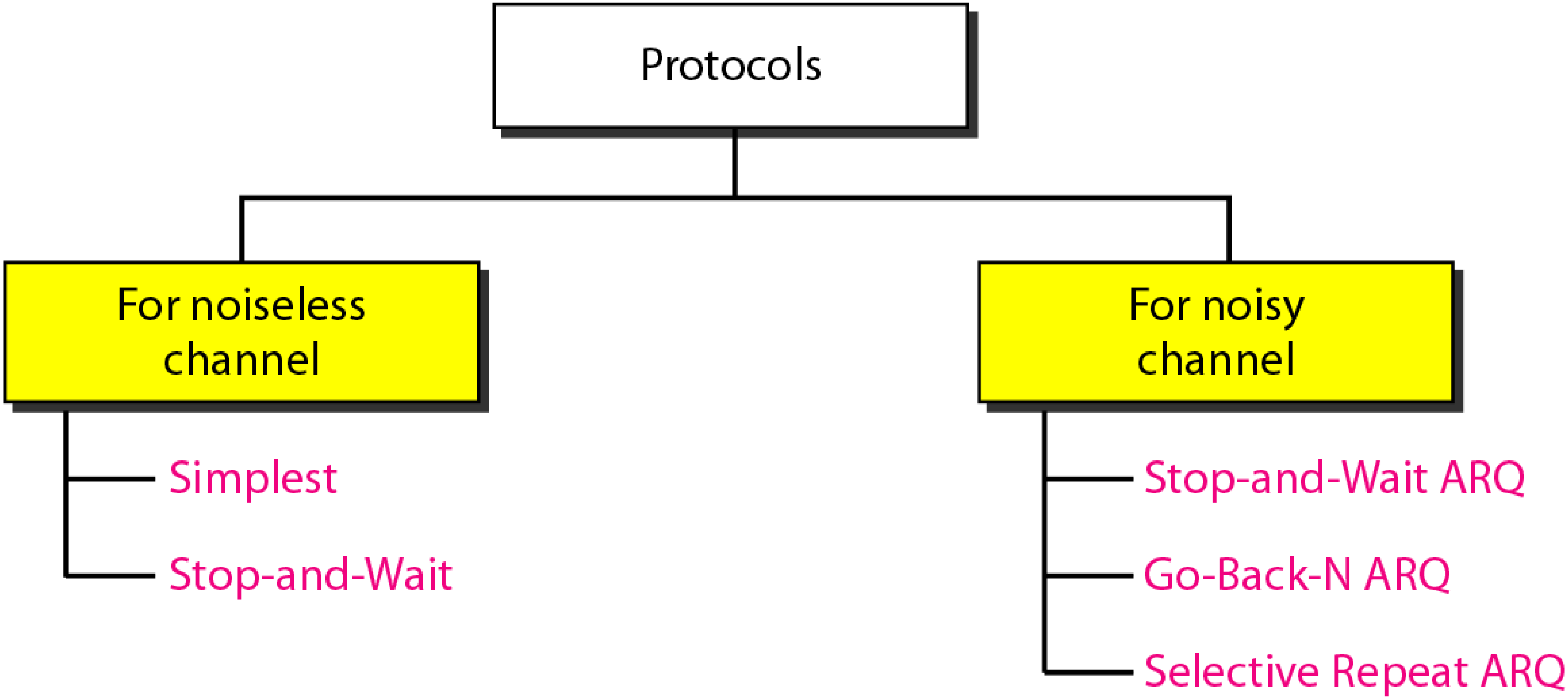
Protocols can be used for

1. Noiseless (error-free) channels

and those that can be used for

2. Noisy (error-creating) channels

Taxonomy of protocols discussed in this chapter



- All the protocols we discuss are unidirectional in the sense that the data frames travel from one node, called the sender, to another node, called the receiver.
- Although special frames, called acknowledgment (ACK) and negative acknowledgment (NAK) can flow in the opposite direction for flow and error control purposes, data flow in only one direction.
- In a real-life network, the data link protocols are implemented as bidirectional; data flow in both directions. In these protocols the flow and error control information such as ACKs and NAKs is included in the data frames in a technique called piggybacking.

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.

Topics discussed in this section:

Simplest Protocol
Stop-and-Wait Protocol

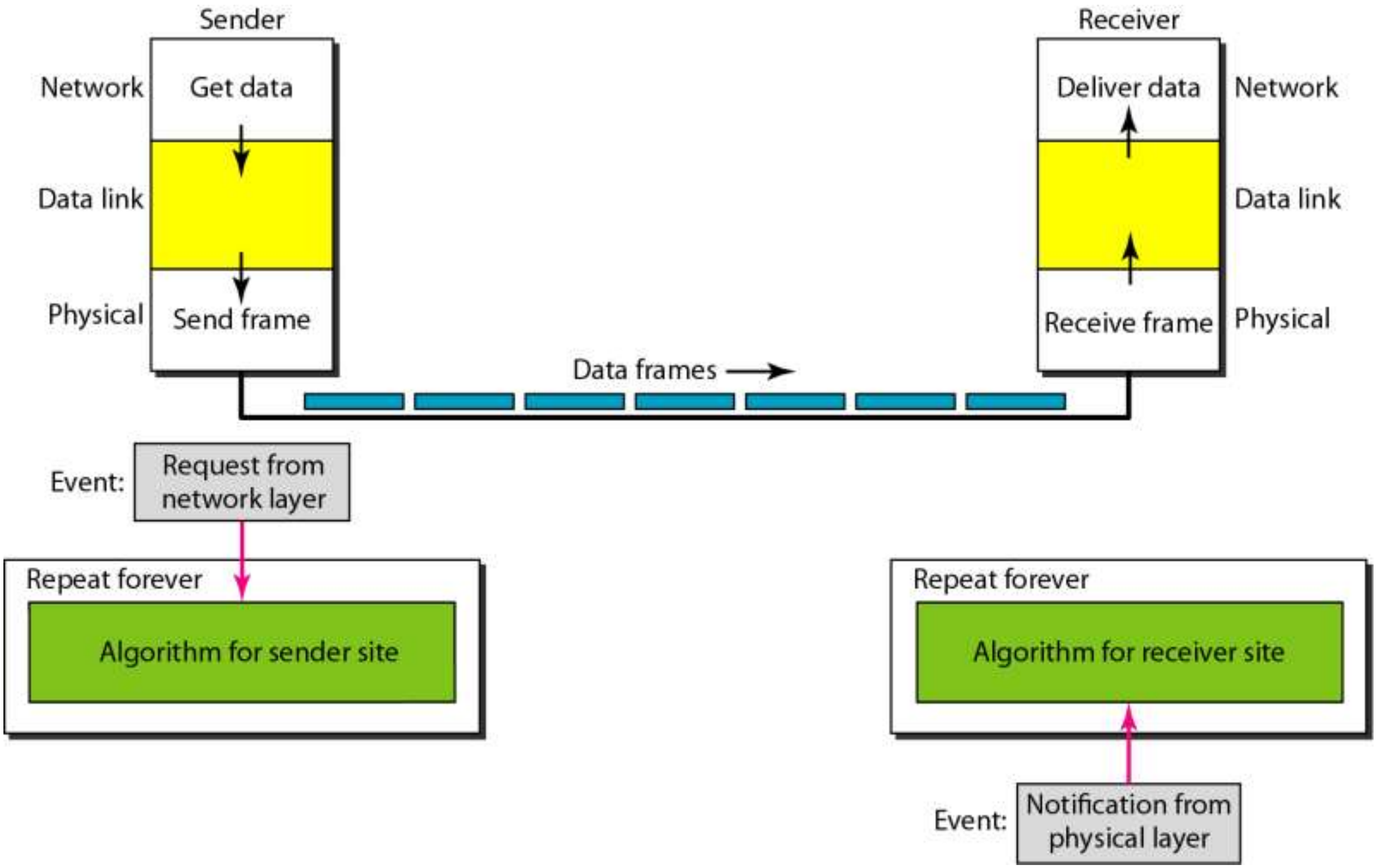
Simplest Protocol

- It 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.
- 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, which can also accept the packet immediately.
- The receiver can never be overwhelmed with incoming frames.

Design

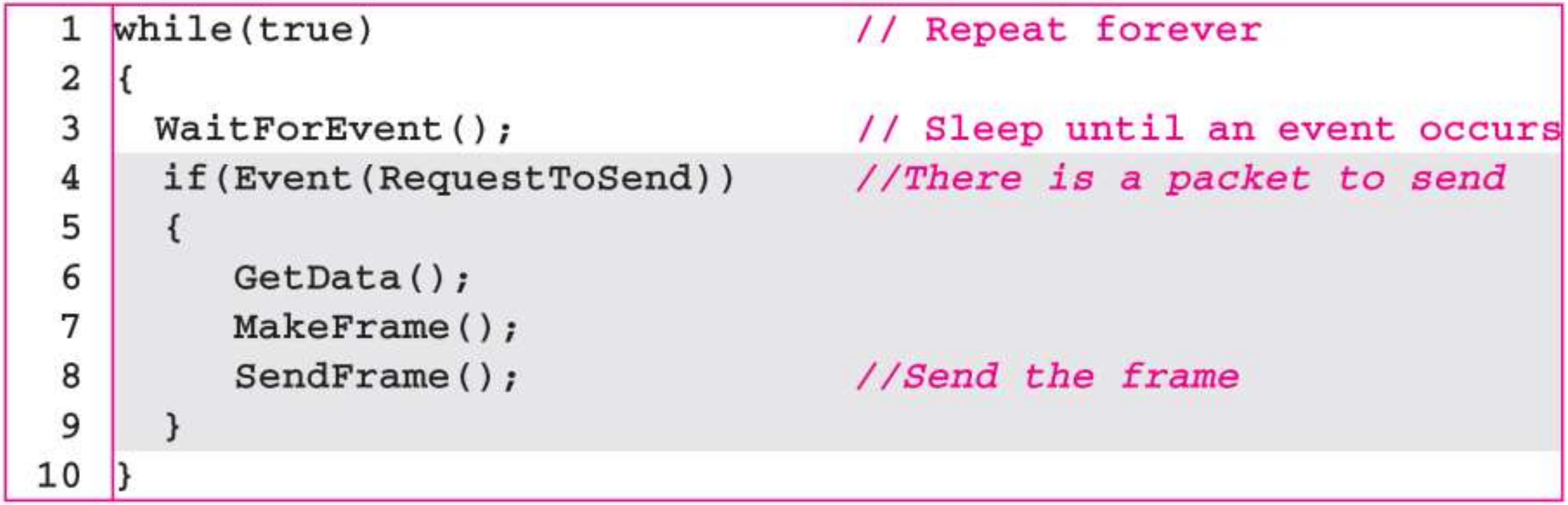
- There is no need for flow control in this scheme.
- The data link layer at the sender site gets data from its network layer, makes a frame out of the data, and sends it.
- The data link layer at the receiver site receives a frame from its physical layer, extracts data from the frame, and delivers the data to its network layer.
- The data link layers use the services provided by their physical layers (such as signaling, multiplexing, and so on) for the physical transmission of bits.

Figure 11.6 *The design of the simplest protocol with no flow or error control*



- 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.
- The protocol is implemented a procedure, 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.
- Both procedures are constantly running because they do not know when the corresponding events will occur.

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

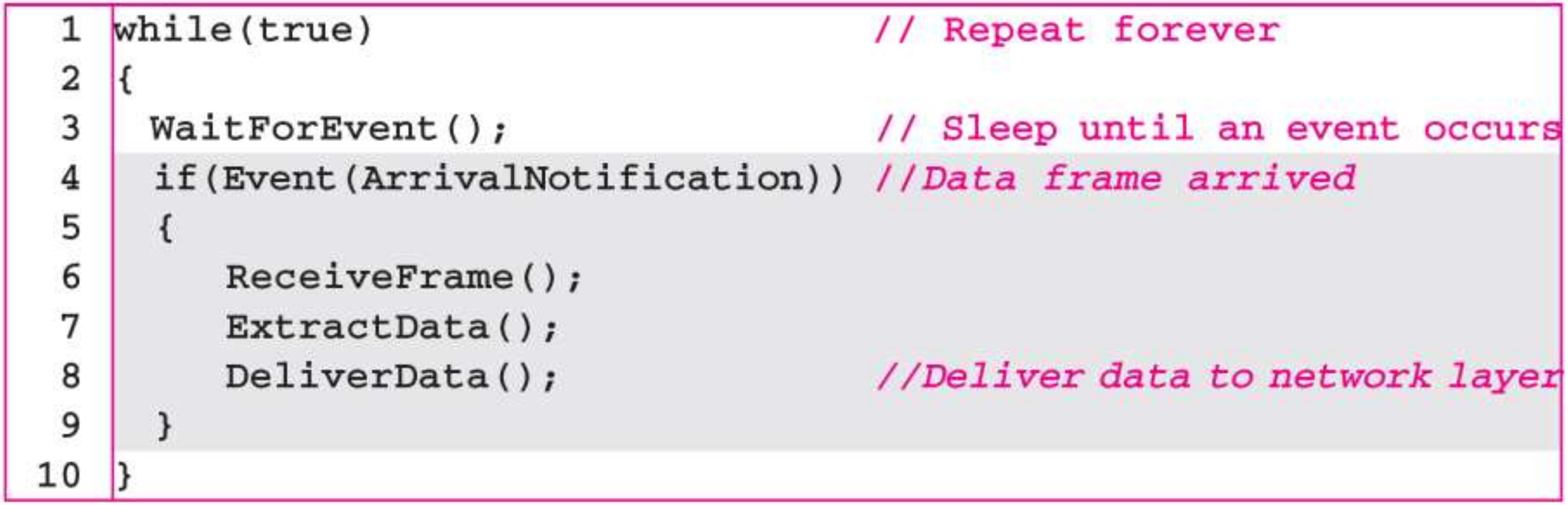


```
while(true)                          // Repeat forever
{
  WaitForEvent();                    // Sleep until an event occurs
  if(Event(RequestToSend))           //There is a packet to send
  {
     GetData();
     MakeFrame();
     SendFrame();                    //Send the frame
  }
}
```

Analysis

- The algorithm has an infinite loop, which means lines 3 to 9 are repeated forever once the program starts.
- The algorithm is an event-driven one, which means that it sleeps (line 3) until an event wakes it up (line 4). This means that there may be an undefined span of time between the execution of line 3 and line 4; there is a gap between these actions.
- When the event, a request from the network layer, occurs, lines 6 though 8 are executed. The program then repeats the loop and again sleeps at line 3 until the next occurrence of the event.

Algorithm 11.2 *Receiver-site algorithm for the simplest protocol*



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

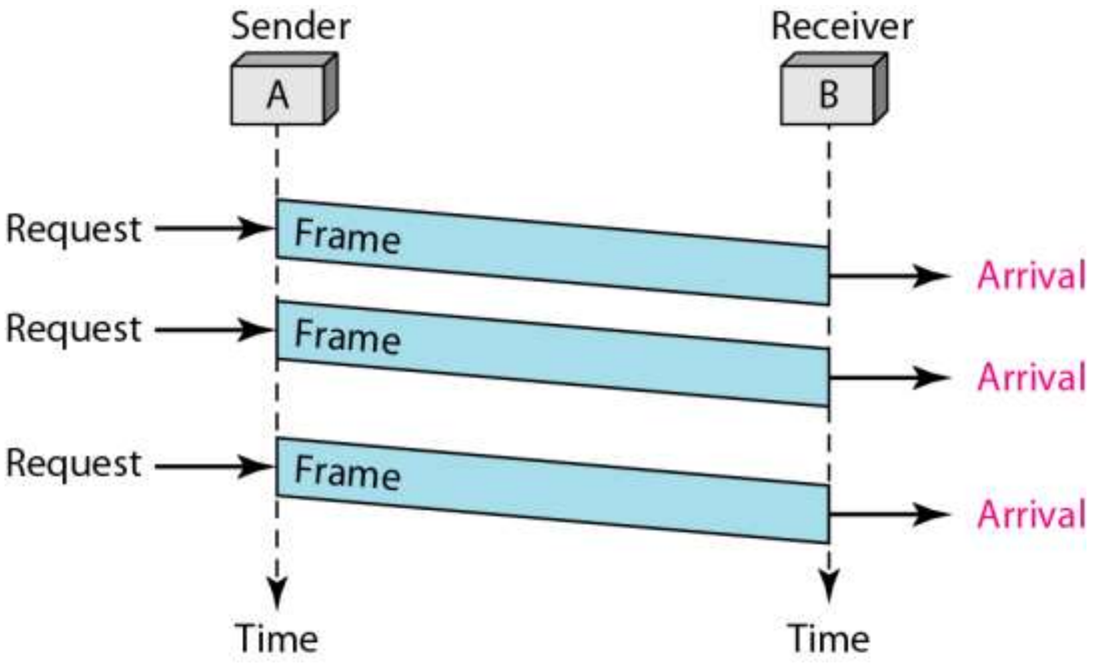
Analysis

- This algorithm has the same format as above Algorithm, except that the direction of the frames and data is upward.
- The event here is the arrival of a data frame. After the event occurs, the data link layer receives the frame from the physical layer using the ReceiveFrame( ) process, extracts the data from the frame using the ExtractData( ) process, and delivers the data to the network layer using the DeliverData( ) process. Here, we also have an event-driven algorithm because the algorithm never knows when the data frame will arrive.

Example

- Figure below 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. The data frames are shown by tilted boxes; the height of the box defines the transmission time difference between the first bit and the last bit in the frame.

Figure 11.7 *Flow diagram for Example 11.1*



Stop-and-Wait Protocol

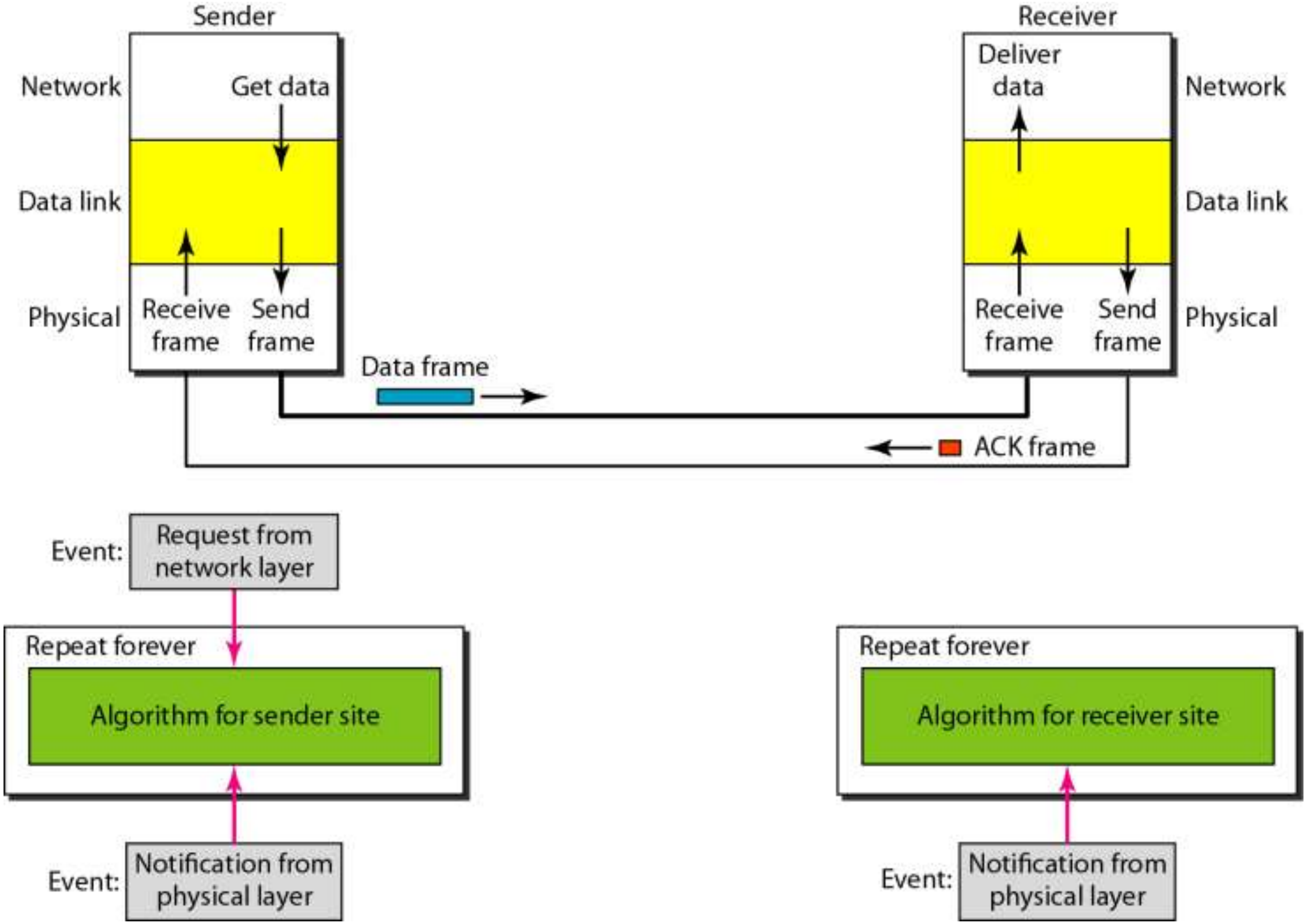
- If data frames arrive at the receiver site faster than they can be processed, the frames must be stored until their use. The receiver does not have enough storage space.
- To prevent the receiver from becoming overwhelmed with frames, need to tell the sender to slow down. There must be feedback from the receiver to the sender.
- The protocol is called the Stop-and-Wait Protocol

Stop-and-Wait Protocol

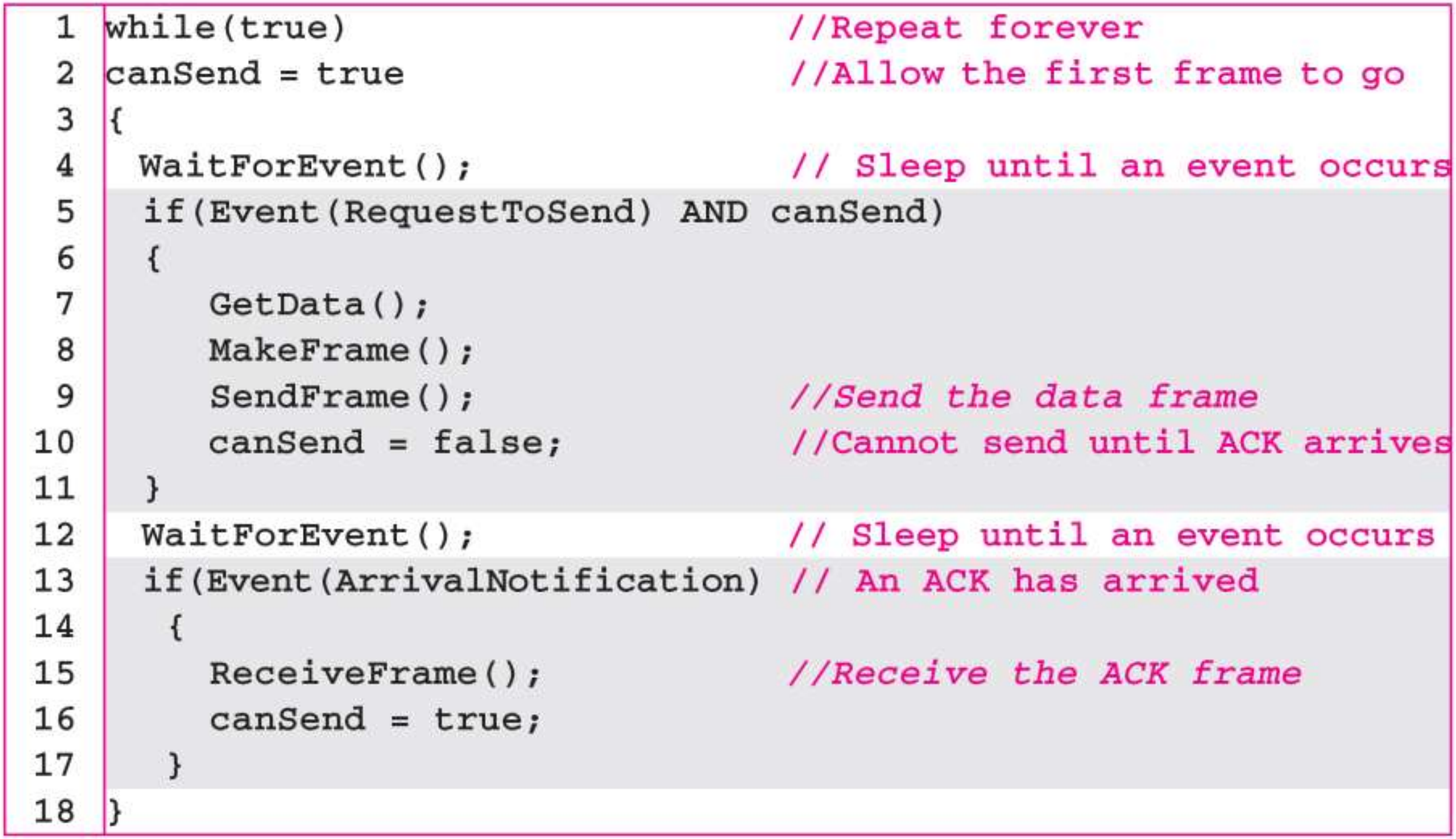
-Sender sends one frame , stops until it receives confirmation from the receiver and sends the next frame

We still have unidirectional communication for data frames, but auxiliary ACK frames (simple tokens of acknowledgment) travel from the other direction.

Figure 11.8 *Design of Stop-and-Wait Protocol*



Sender-site algorithm for Stop-and-Wait Protocol



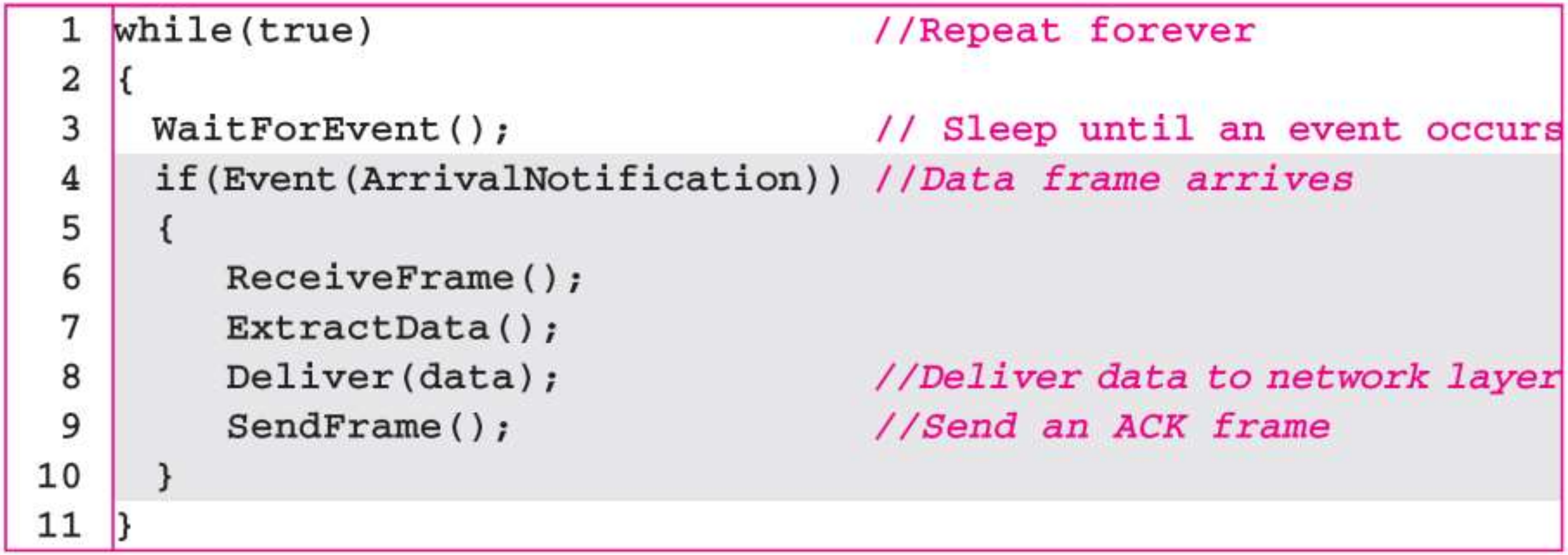
```
while(true)                              //Repeat forever
canSend = true                           //Allow the first frame to go
{
  WaitForEvent();                        // Sleep until an event occurs
  if(Event(RequestToSend) AND canSend)
  {
     GetData();
     MakeFrame();
     SendFrame();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

Analysis

- Here two events can occur: a request from the network layer or an arrival notification from the physical layer
- After a frame is sent, the algorithm must ignore another network layer request until that frame is acknowledged.
- We know that two arrival events cannot happen one after another because the channel is error-free and does not duplicate the frames.

- We need somehow to prevent the immediate sending of the data frame. We have used a simple ***canSend variable*** that can either be true or false. When a frame is sent, the variable is set to false to indicate that a new network request cannot be sent until **canSend** is true.
- When an ACK is received, *canSend* is set to true to allow the sending of the next frame.

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



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

Analysis

- This is very similar to sender-site Algorithm with one exception.
- After the data frame arrives, the receiver sends an ACK frame (line 9) to acknowledge the receipt and allow the sender to send the next frame.

NOISY CHANNELS

Stop-and-Wait Automatic Repeat Request

This protocol adds a simple error control mechanism to the Stop-and-Wait Protocol.

To detect and correct corrupted frames, we need to add redundancy bits to our data frame. 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.

- The corrupted and lost frames need to be resent in this protocol. The sender can know which frame to resend. For this purpose, 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.
- Since the protocol uses the stop-and-wait mechanism. Since an ACK frame can also be corrupted and lost, it too needs redundancy bits and a sequence number

Sequence Numbers

- The protocol specifies that frames need to be numbered. This is done by using sequence numbers. A field is added to the data frame to hold the sequence number of that frame.
- We want to minimize the frame size, we use smallest range sequence numbers. 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 2m - 1, and then are repeated.

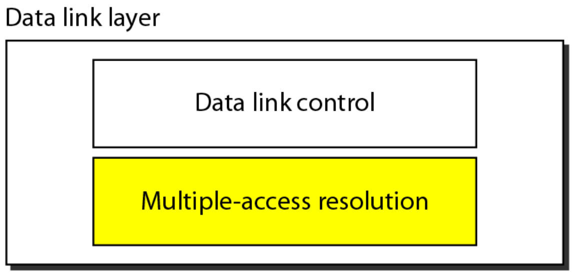
- Assume we have used x as a sequence number; we only need to use x + 1 after that. There is no need for x + 2. Three things can happen.
- 1. The frame arrives safe and sound at the receiver site; the receiver sends an acknowledgment. The acknowledgment arrives at the sender site, causing the sender to send the next frame numbered x + 1.
- 2. The frame arrives safe and sound at the receiver site; the receiver sends an acknowledgment, but the acknowledgment is corrupted or lost. The sender resends the frame (numbered x)

- after the time-out. Note that the frame here is a duplicate. The receiver can recognize this fact because it expects frame x + 1 but frame x was received.
- 3. The frame is corrupted or never arrives at the receiver site; the sender resends the frame (numbered x) after the timeout. There is a need for sequence numbers x and x + 1 because the receiver needs to distinguish between case 1 and case 2.

In Stop-and-Wait ARQ we use sequence numbers to number the frames. The sequence numbers are based on modulo-2 arithmetic.

MULTIPLE ACCESS

The data link layer as two sublayers. The upper sublayer is responsible for data link control, and the lower sublayer is responsible for resolving access to the shared media.

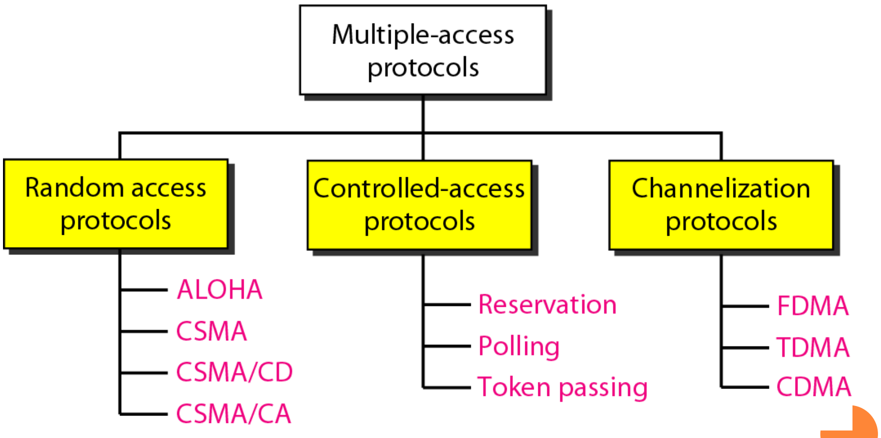


The upper sublayer that is responsible for flow and error control is called the logical link control (LLC) layer; the lower sublayer that is mostly responsible for multiple access resolution is called the media access control (MAC) layer.

When nodes or stations are connected and use a common link, called a multipoint or broadcast link, we need a multiple-access protocol to coordinate access to the link.

- When more than two nodes send at the same time , the transmitted frames collide.
- All collide frames are lost and the bandwidth of the broadcast channel will be wasted.
- We need multiple –access protocol to coordinate access to multipoint or broadcast link.
- Multiple access protocols are needed in wire and wireless LANs and satellite networks.

Multiple-access protocols are categorzed into three groups.



RANDOM ACCESS

In random access or contention methods, no station is superior to another station and none is assigned the control over another.

Two features give this method its name.

1. There is no scheduled time for a station to transmit. Transmission is random among the stations. That is why these methods are called random access.
2. No rules specify which station should send next. Stations compete with one another to access the medium. That is why these methods are also called contention methods.

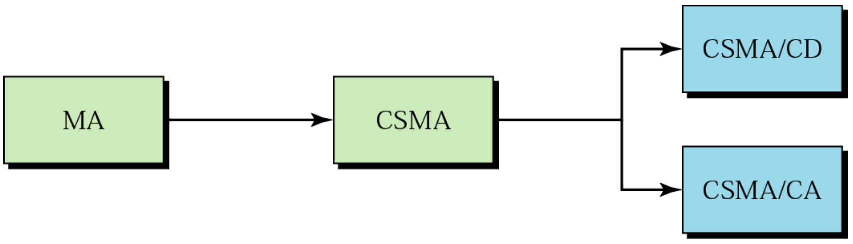
Random access protocols

ALOHA :- The earliest random access method, which used a very simple procedure called Multiple Access (MA).

CSMA(Carrier Sense Multiple Access) :- To sense the medium before transmitting.

CSMA/CD(Carrier Sense Multiple Access with Collision Detection) :- Tells the station what to do when a collision is detected.

CSMA/CA(Carrier Sense Multiple Access with Collision Avoidance) :- Tries to avoid the collision.

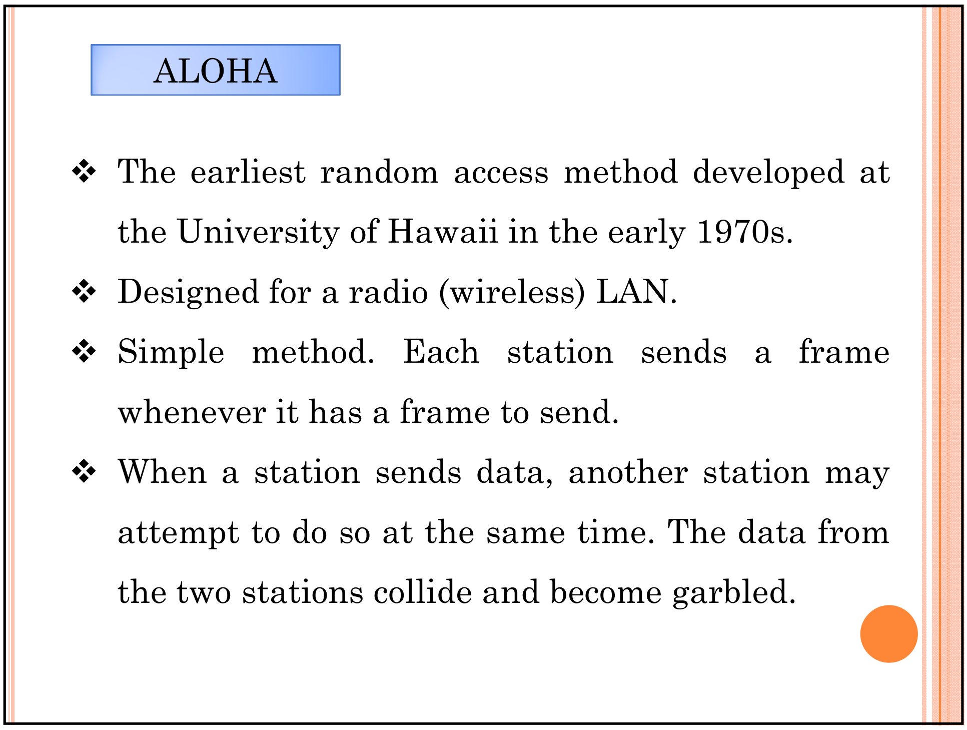


MA

CSMA

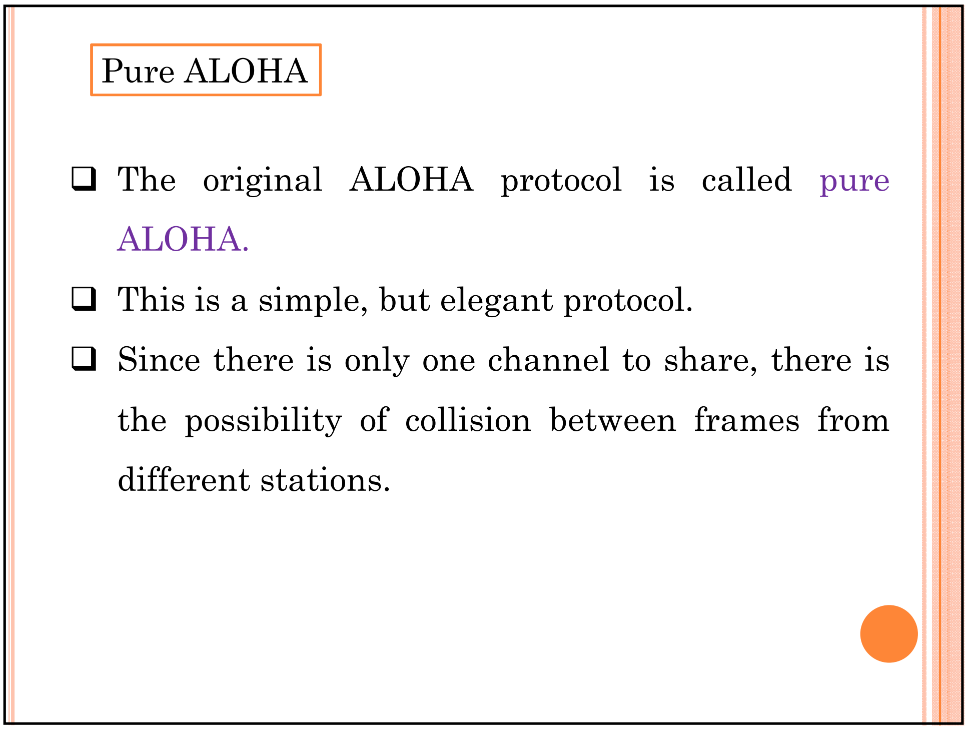
CSMA/CD

CSMA/CA



ALOHA

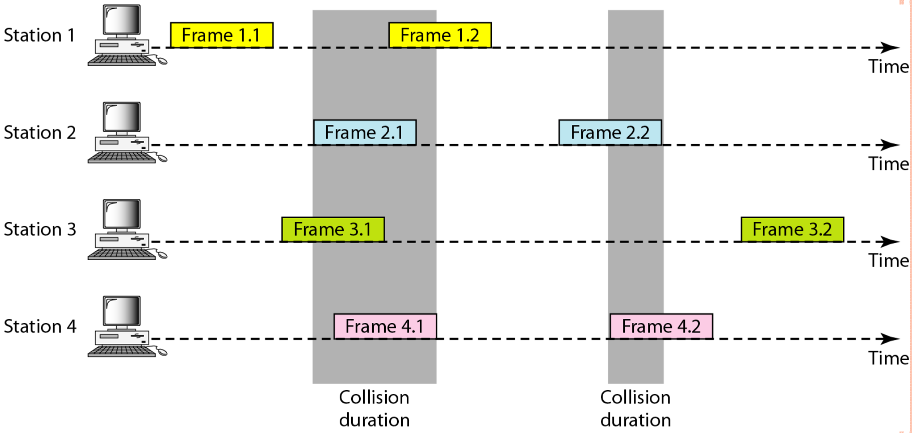
- ❖ The earliest random access method developed at the University of Hawaii in the early 1970s.
- ❖ Designed for a radio (wireless) LAN.
- ❖ Simple method. Each station sends a frame whenever it has a frame to send.
- ❖ When a station sends data, another station may attempt to do so at the same time. The data from the two stations collide and become garbled.



Pure ALOHA

- ❑ The original ALOHA protocol is called pure ALOHA.
- ❑ This is a simple, but elegant protocol.
- ❑ Since there is only one channel to share, there is the possibility of collision between frames from different stations.

Figure : Frames in a pure ALOHA network



The figure shows that each station sends two frames; there are a total of eight frames on the shared medium. Some of these frames collide because multiple frames are in contention for the shared channel.

- Each station sends a frame whenever it has a frame to send.
- It relies on acknowledgments from the receiver.
- If the ACK dose not arrive after a time-out period, the station resend the frame.
- time-out is equal the max possible round trip time = $2 \times T_p$.

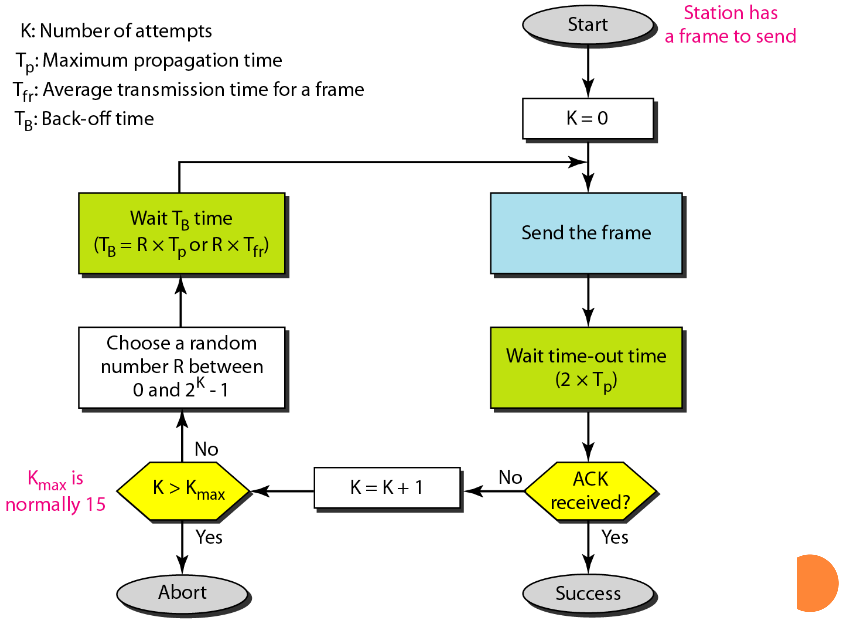
Figure : Procedure for pure ALOHA protocol

K: Number of attempts

T_p: Maximum propagation time

T_{fr}: Average transmission time for a frame

T_B: Back-off time


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

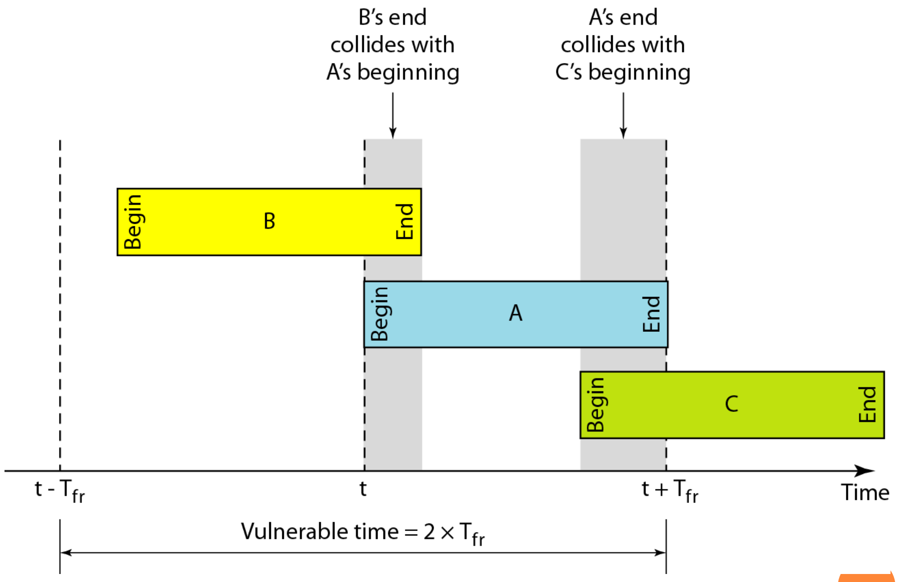
Note: R range from 0 to 2^k-1

- ❖ T_p(max propagation time) time required to send a frame between the most widely separated station
- ❖ To minimize collisions, each station waits a random amount of time (back-off time T_B)before resending its frame.
- ❖ T_B is a random value that depend on K (the number of attempted unsuccessful transmission) .
- ❖ The formula of T_B is the binary exponential back-off.
- ❖ After a max number of retransmission attempts K_{max}, a station must give up and try later to prevent congestion.

Vulnerable time

Vulnerable time is the length of time in which there is a possibility of collision. We assume that the stations send fixed-length frames with each frame taking Tf_r s to send.

Figure below shows the vulnerable time for station A.



B's end collides with A's beginning
A's end collides with C's beginning
Begin
B
End
Begin
A
End
Begin
C
End
t - T_{fr}
t
t + T_{fr}
Time
Vulnerable time = 2 × T_{fr}

Station A sends a frame at time t. Station B has already sent a frame between t-T_{fr} and t. This leads to a collision between the frames from A and B. On the other hand, station C sends a frame between t and t + T_{fr} . Here, there is a collision between frames A and C. The vulnerable time, during which a collision may occur in pure ALOHA, is 2 times the frame transmission time.

Pure ALOHA vulnerable time = 2 x T_{fr}

Throughput

The rate of successful transmissions of frames is called throughput.

The throughput for pure ALOHA is S =G x e^{-2G}.

G is the average number of frames generated by the system during one frame transmission time

The maximum throughput S_{max} =0.184 when G = (1/2).

In other words, if one-half a frame is generated during one frame transmission time (in other words, one frame during two frame transmission times), then 18.4 percent of these frames reach their destination successfully.

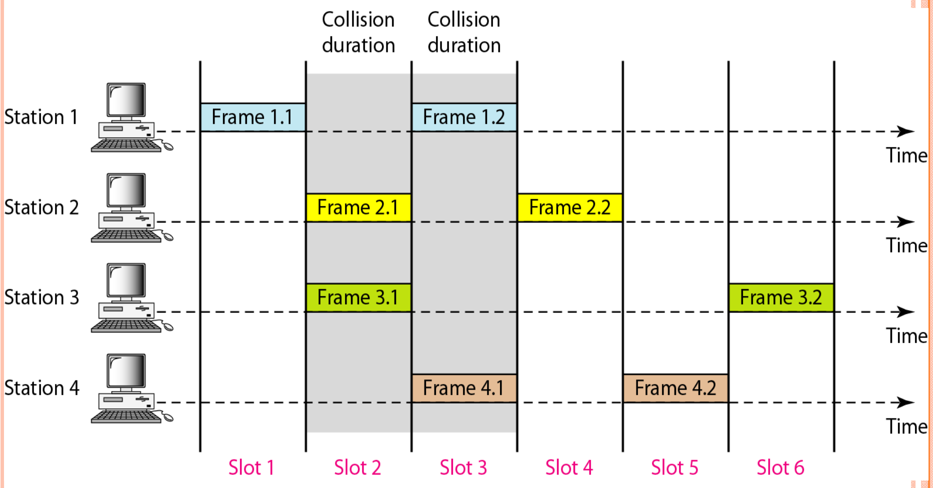
Slotted ALOHA

Slotted ALOHA was invented to improve the efficiency of pure ALOHA.

In slotted ALOHA we divide the time into slots of T_{fr} s and force the station to send only at the beginning of the time slot.

Figure below shows an example of frame collisions in slotted ALOHA.

Figure : Frames in a slotted ALOHA network



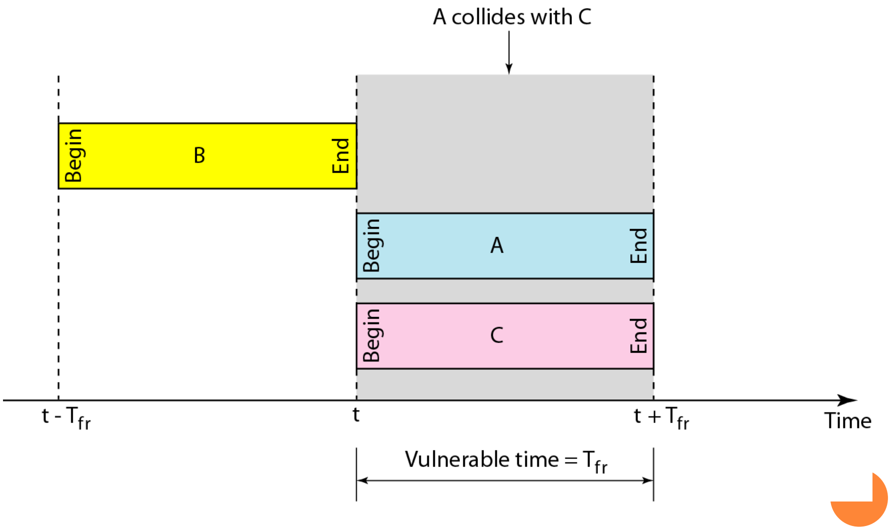
Because a station is allowed to send only at the beginning of the synchronized time slot, if a station misses this moment, it must wait until the beginning of the next time slot. This means that the station which started at the beginning of this slot has already finished sending its frame.

Slotted ALOHA vulnerable time = T_{fr}

The throughput for slotted ALOHA is $S = G \times e^{-G}$.

The maximum throughput $S_{max} = 0.368$ when $G=1$.

Figure : Vulnerable time for slotted ALOHA protocol

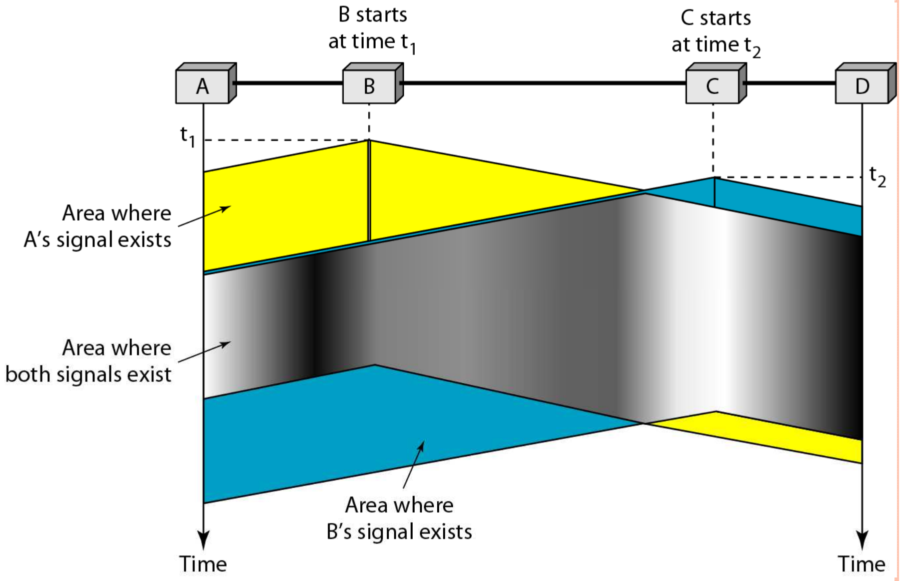


Carrier Sense Multiple Access (CSMA)

To minimize the chance of collision and, therefore, increase the performance, the CSMA method was developed.

1. Sense the carrier before transmit : "sense before transmit“ or “listen before you talk”.
2. CSMA can reduce the possibility of collision, but it can not eliminate it because of the propagation delay. (a station may sense the medium and find it idle, only because the first bit of a frame sent by another station has not been received).

Figure : Space/time model of the collision in CSMA


B starts at time t1
C starts at time t2
A
B
C
D
t1
t2
Area where A’s signal exists
Area where both signals exist
Area where B’s signal exists
Time
Time

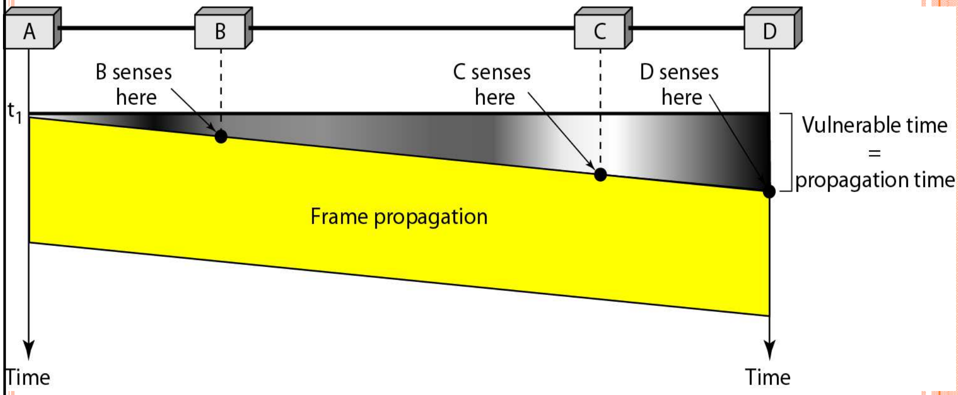
At time t_1 station B senses the medium and finds it idle, so it sends a frame. At time t_2 ($t_2 > t_1$) station C senses the medium and finds it idle because, at this time, the first bits from station B have not reached station C. Station C also sends a frame. The two signals collide and both frames are destroyed.

Vulnerable Time

Vulnerable time: time in which there is a possibility of collision.

Vulnerable time for CSMA is the max propagation time T_p needed for a signal to propagate from one end of the medium to the other.

Figure : Vulnerable time in CSMA



Persistence Methods

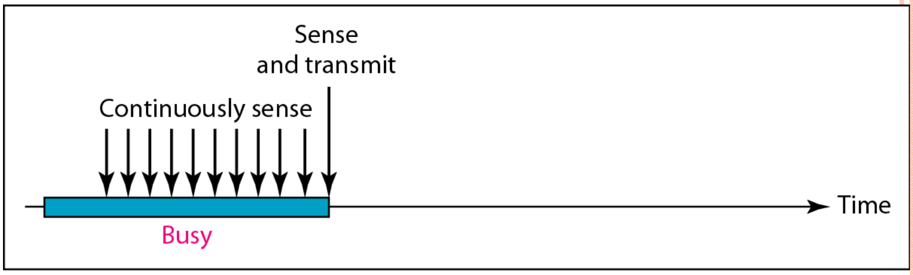
If the channel is busy or idle, the station use three methods to solve these problems.

The 1-persistent method

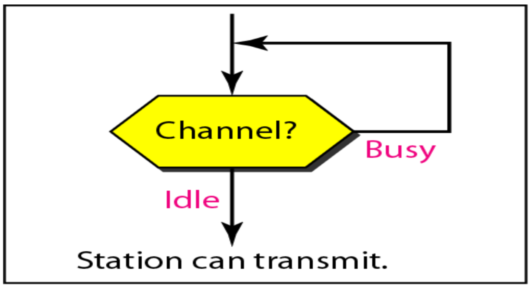
The nonpersistent method

The p-persistent method.

1-persistence methods



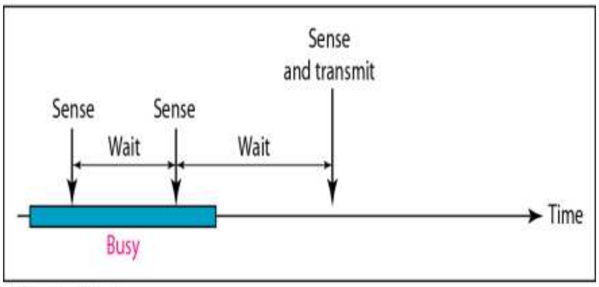
a. 1-persistent



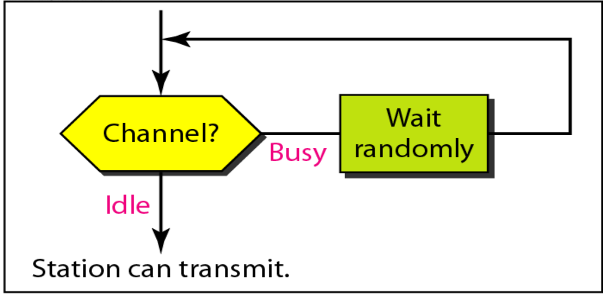
a. 1-persistent

1. The 1-persistent method is simple and straight forward.
2. When the sender (station) is ready to transmit data, it checks if the medium is busy. If so, it senses the medium continually until it becomes idle.
3. If line is idle, sends the frame immediately (with probability of 1).
    - Chances of collision is high.

non-persistence methods



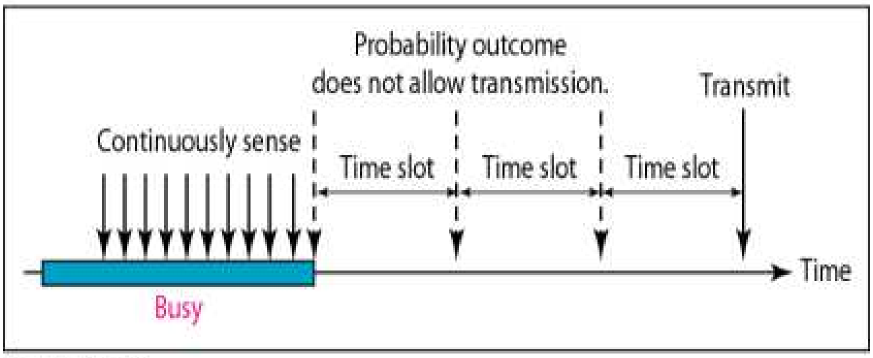
b. Nonpersistent



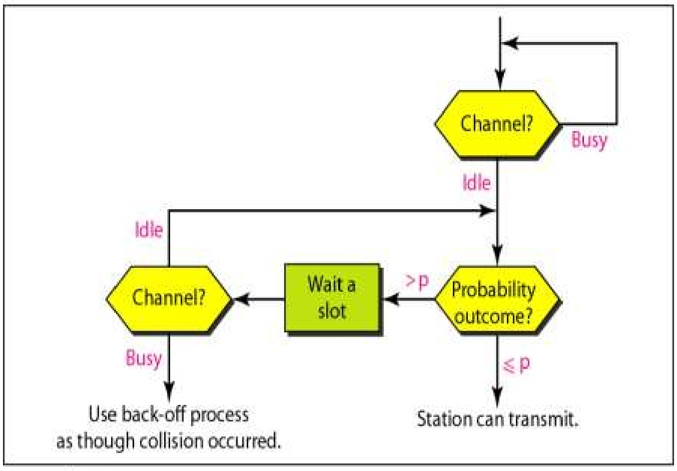
b. Nonpersistent

- If a station has a frame to send, it senses the line.
- If line is idle, the station sends the frame immediately.
- If line is not idle, the station waits a random period of time and then senses the line again.
  - Chances of collision is reduced.
  - Reduces efficiency of the network (because the medium remain idle when there may be stations with frames to send.

P-Persistence



c. p-persistent



c. p-persistent

- It used if the channel has time slots with a slot duration equal to or grater than the maximum propagation time.
- When the sender (station) is ready to transmit data, it checks if the medium is busy. If so, it senses the medium continually until it becomes idle.
- If line is idle it may or may not send. It sends with probability p.

  - ✓ Reduces the chance of collision and improves the efficiency by combining the other two strategies

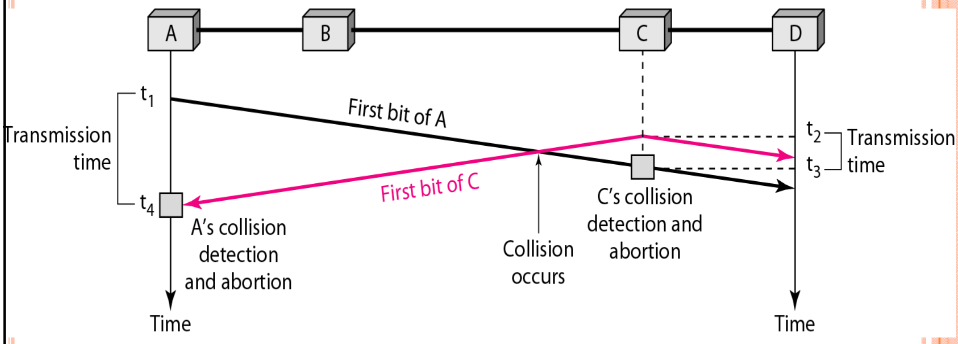
1. With probability p, the station sends its frame.
2. With probability q = 1 - p, the station waits for the beginning of the next time slot and checks the line again.
   - If the line is idle, it goes to step 1.
   - If the line is busy, it acts as though a collision has occurred and uses the backoff procedure.

Carrier Sense Multiple Access collision detection (CSMA/CD)

"Listen-while-talk" protocol. A host listens even while it is transmitting, and if a collision is detected, stops transmitting

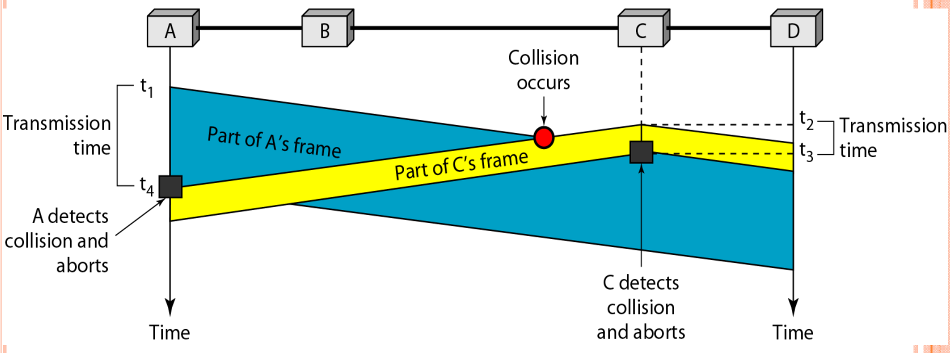
To better understand CSMA/CD, let us look at the first bits transmitted by the two stations involved in the collision. Although each station continues to send bits in the frame until it detects the collision, we show what happens as the first bits collide.

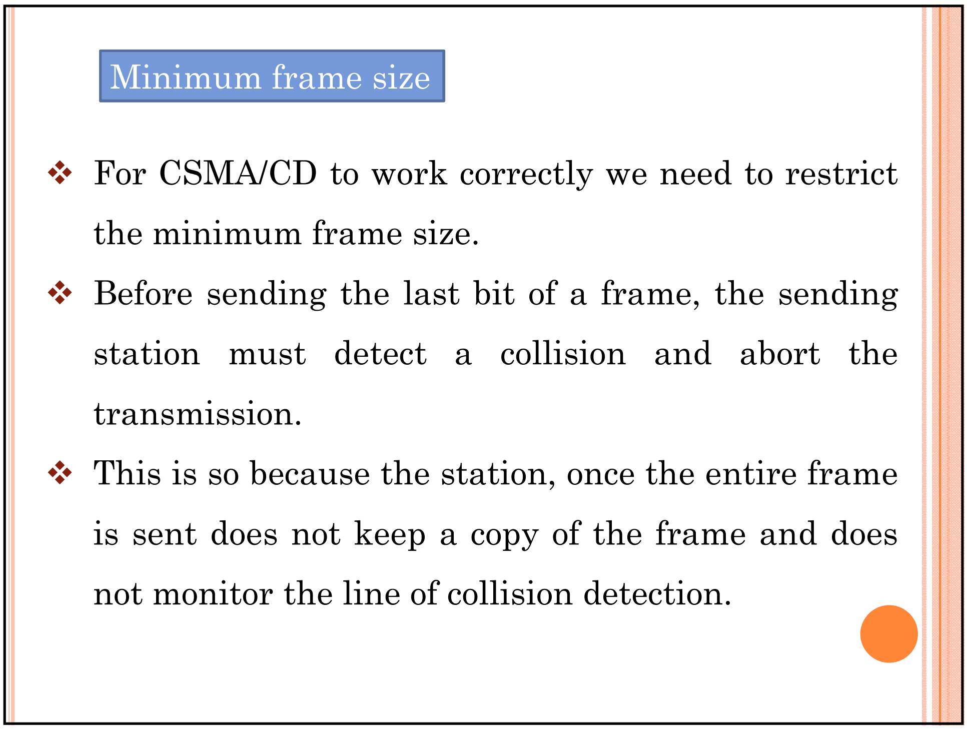
Figure : Collision of the first bit in CSMA/CD



At time t_1, station A has executed its persistence procedure and starts sending the bits of its frame. At time t_2, station C has not yet sensed the first bit sent by A. Station C executes its persistence procedure and starts sending the bits in its frame, which propagate both to the left and to the right. The collision occurs sometime after time t_2. Station C detects a collision at time t_3 when it receives the first bit of A's frame. Station C immediately aborts transmission. Station A detects collision at time t_4 when it receives the first bit of C's frame; it also immediately aborts transmission.

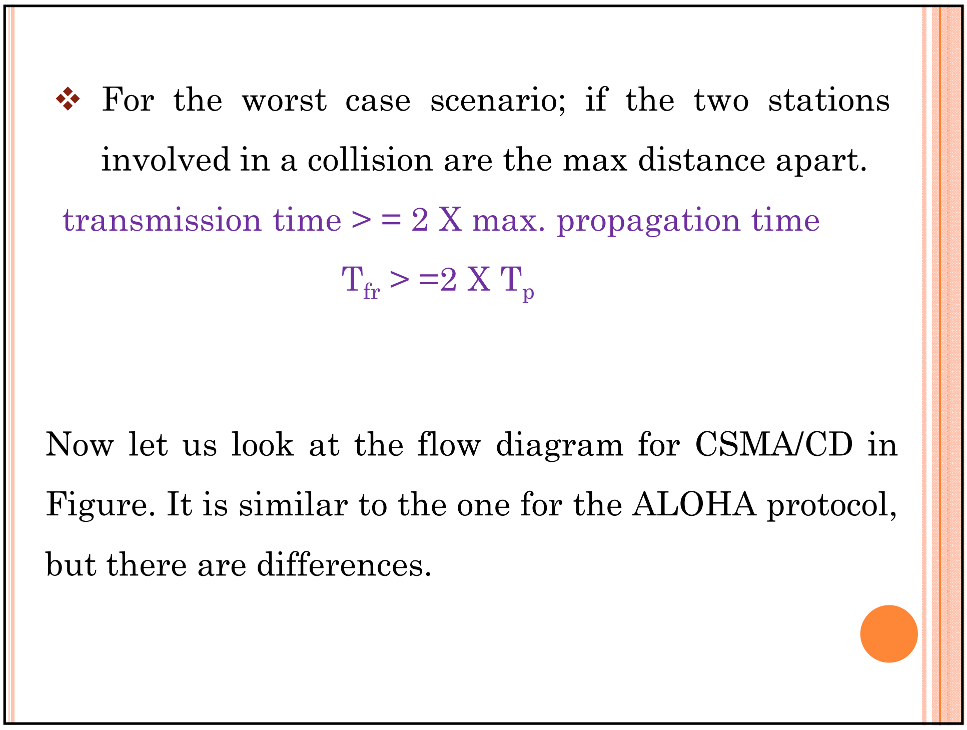
Figure : Collision and abortion in CSMA/CD





Minimum frame size

- ❖ For CSMA/CD to work correctly we need to restrict the minimum frame size.
- ❖ Before sending the last bit of a frame, the sending station must detect a collision and abort the transmission.
- ❖ This is so because the station, once the entire frame is sent does not keep a copy of the frame and does not monitor the line of collision detection.



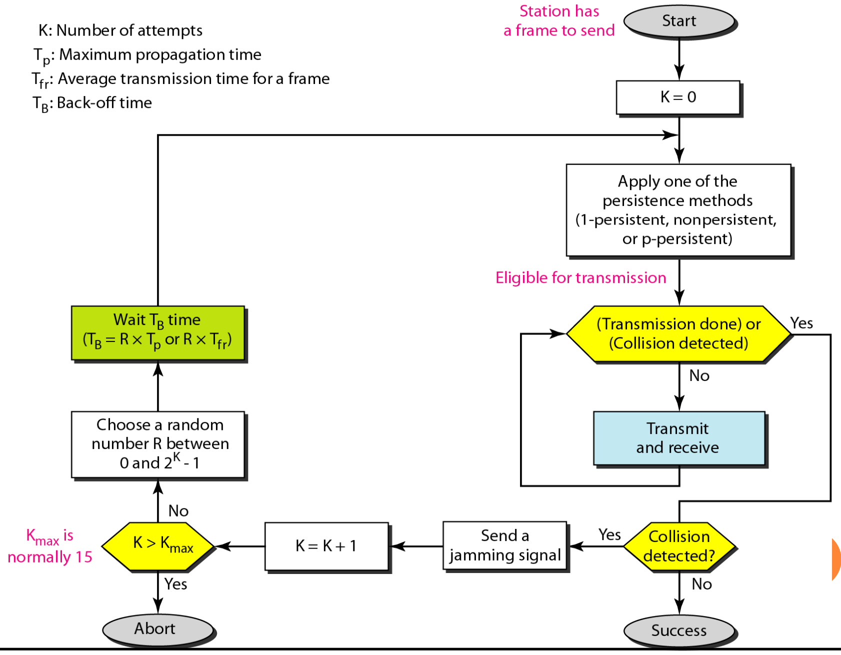
- ❖ For the worst case scenario; if the two stations involved in a collision are the max distance apart.

transmission time > = 2 X max. propagation time

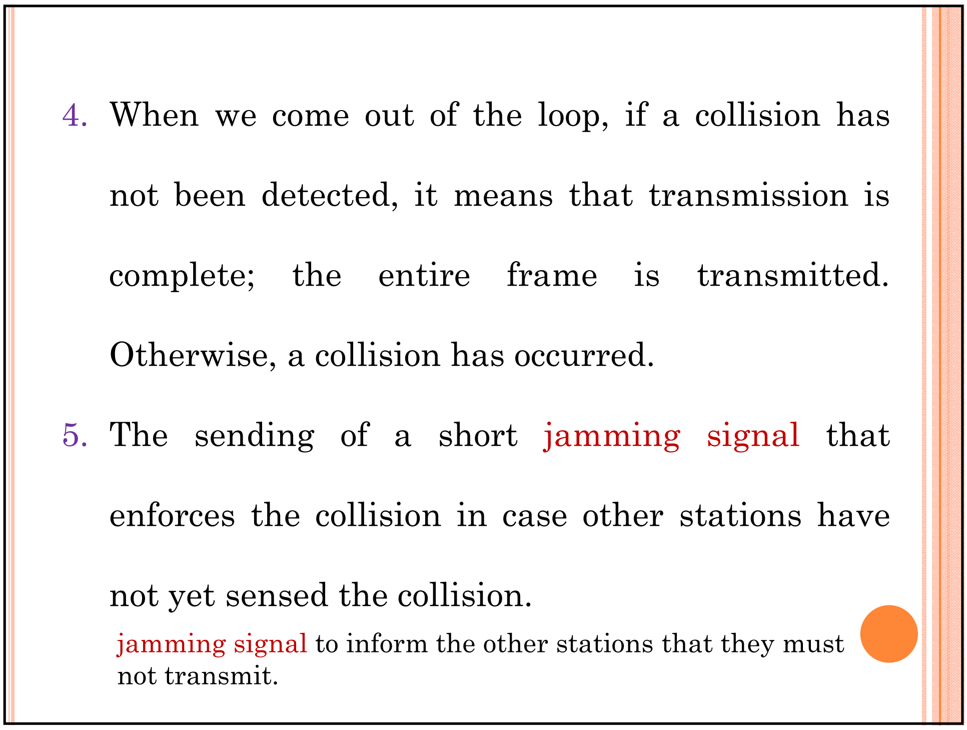
$$T_{fr} > = 2 \text{ X } T_p$$

Now let us look at the flow diagram for CSMA/CD in Figure. It is similar to the one for the ALOHA protocol, but there are differences.

Figure : Flow diagram for the CSMA/CD

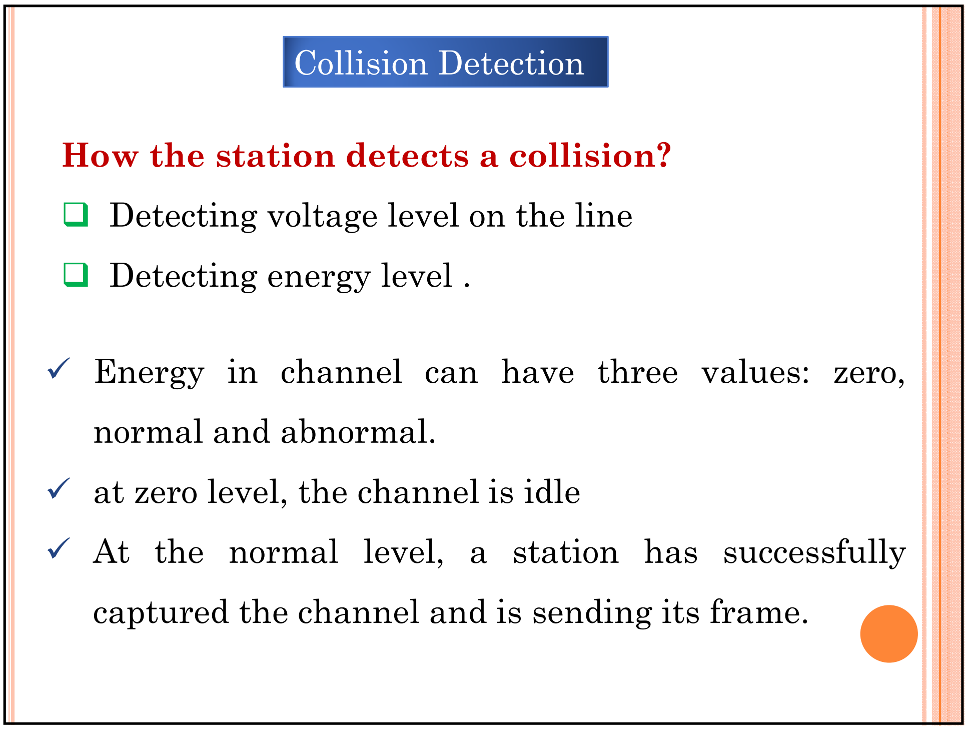

K: Number of attempts
T_p: Maximum propagation time
T_{fr}: Average transmission time for a frame
T_B: Back-off time
Station has a frame to send
Start
K = 0
Apply one of the persistence methods (1-persistent, nonpersistent, or p-persistent)
Eligible for transmission
(Transmission done) or (Collision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

1. The addition of the persistence process.
2. Transmission and collision detection is a continuous process. In ALOHA, first transmit the entire frame and then wait for an acknowledgment.
3. We constantly monitor in order to detect one of two conditions: either transmission is finished or a collision is detected.



4. When we come out of the loop, if a collision has not been detected, it means that transmission is complete; the entire frame is transmitted. Otherwise, a collision has occurred.
5. The sending of a short jamming signal that enforces the collision in case other stations have not yet sensed the collision.

jamming signal to inform the other stations that they must not transmit.



Collision Detection

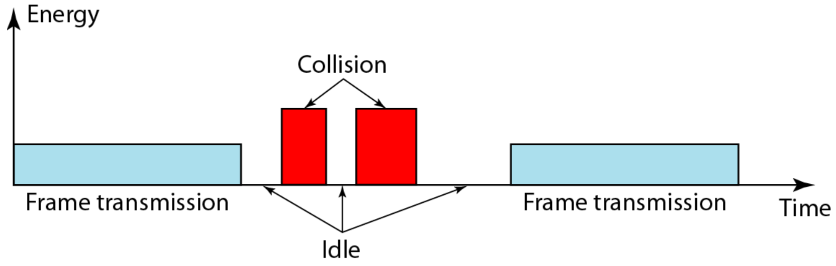
How the station detects a collision?

- ❑ Detecting voltage level on the line
- ❑ Detecting energy level .

- ✓ Energy in channel can have three values: zero, normal and abnormal.
- ✓ at zero level, the channel is idle
- ✓ At the normal level, a station has successfully captured the channel and is sending its frame.

✓ At the abnormal level, there is a collision and the level of the energy twice the normal level.

Figure : Energy level during transmission, idleness, or collision



Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA)

The basic idea behind *CSMA/CD is that a station needs to be able to receive while* transmitting to detect a collision.

When there is no collision, the station receives one signal: its own signal. When there is a collision, the station receives two signals: its own signal and the signal transmitted by a second station.

In a wired network, the received signal has almost the same energy as the sent signal. This means that in a collision, the detected energy almost doubles. So they can easily detect collision.

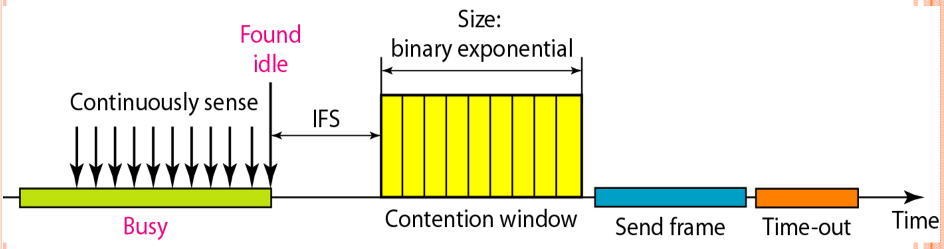
In a wireless network, much of the sent energy is lost in transmission. The received signal has very little energy. This is not useful for effective collision detection.

We need to avoid collisions on wireless networks because they cannot be detected. Carrier sense multiple access with collision avoidance (CSMA/CA) was invented for this network.

Collisions are avoided through the use of CSMA/CA's three strategies:

1. The interframe space
2. The contention window
3. Acknowledgments

Figure : Timing in CSMA/CA



Interframe Space (IFS)

When an idle channel is found, the station does not send immediately. It waits for a period of time called the interframe space or IFS. Even though the channel may appear idle when it is sensed, a distant station may have already started transmitting. The distant station's signal has not yet reached this station.

The IFS time allows the front of the transmitted signal by the distant station to reach this station. If after the IFS time the channel is still idle, the station can send, but it still needs to wait a time equal to the contention time.

In CSMA/CA, the IFS can also be used to define the priority of a station or a frame.

Contention Window

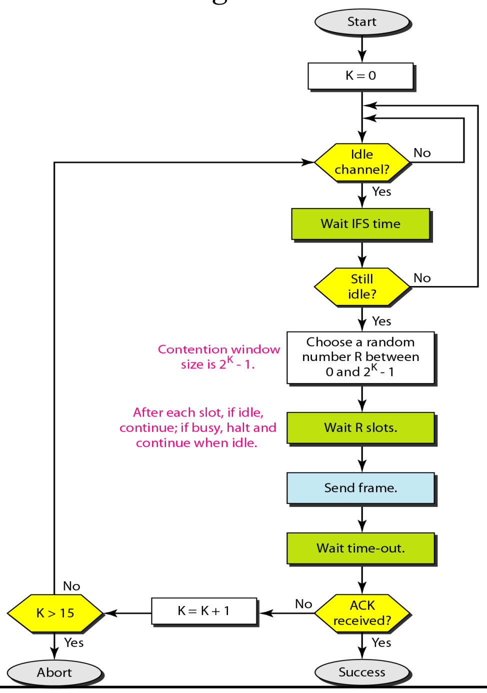
1. The contention window is an amount of time divided into slots. A station that is ready to send chooses a random number of slots as its wait time. The number of slots in the window changes.
2. This means that it is set to one slot the first time and then doubles each time the station cannot detect an idle channel after the IFS time.
3. One point about the contention window is that the station needs to sense the channel after each time slot.

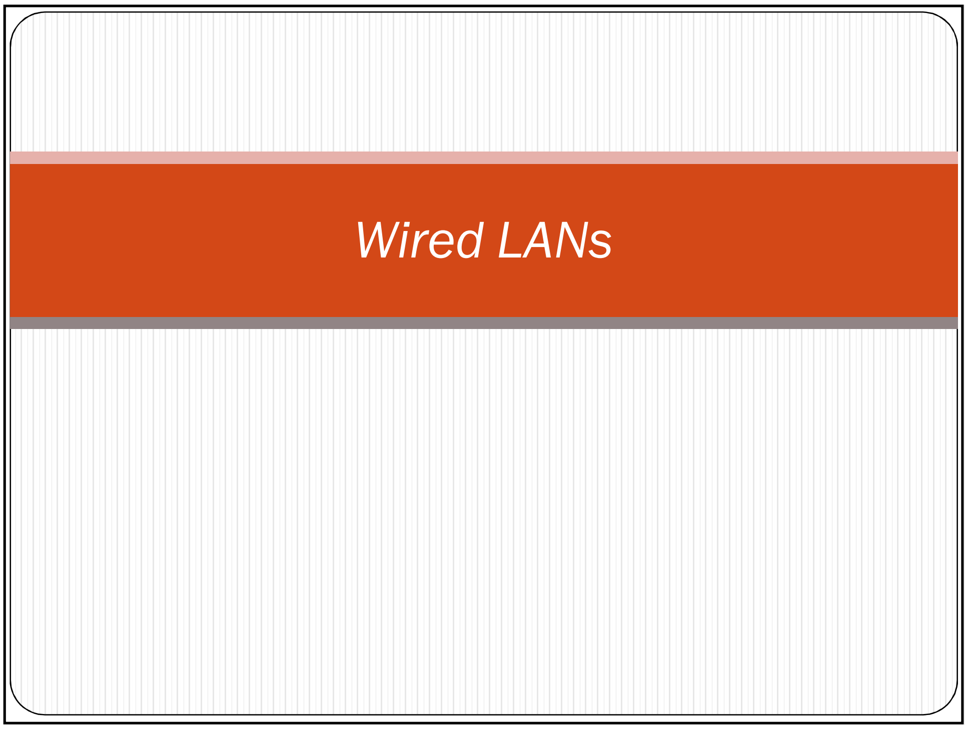
In CSMA/CA, if the station finds the channel busy, it does not restart the timer of the contention window; it stops the timer and restarts it when the channel becomes idle.

Acknowledgment

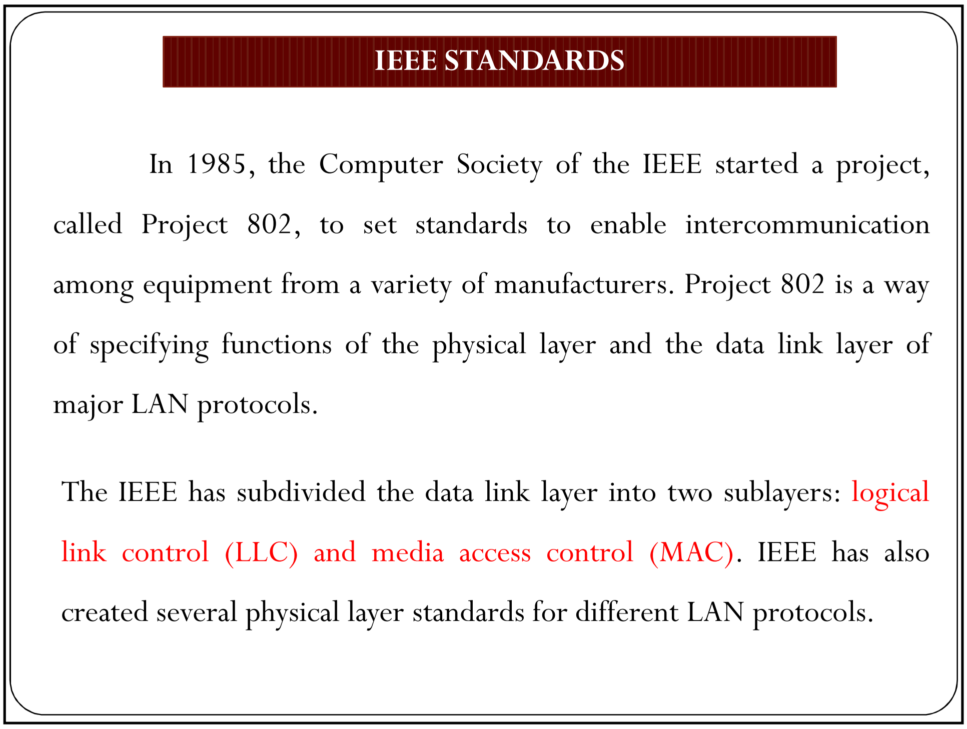
The positive acknowledgment and the time-out timer can help guarantee that the receiver has received the frame.

Figure : Flow diagram for CSMA/CA


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



Wired LANs

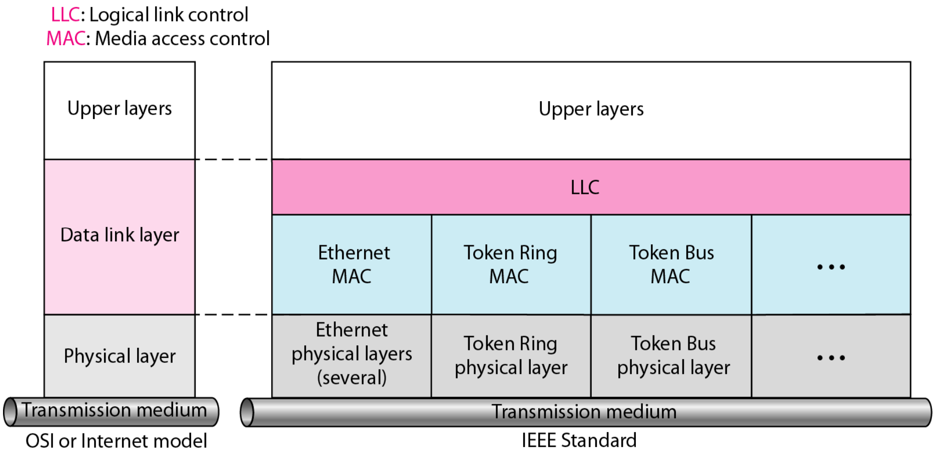


IEEE STANDARDS

In 1985, the Computer Society of the IEEE started a project, called Project 802, to set standards to enable intercommunication among equipment from a variety of manufacturers. Project 802 is a way of specifying functions of the physical layer and the data link layer of major LAN protocols.

The IEEE has subdivided the data link layer into two sublayers: logical link control (LLC) and media access control (MAC). IEEE has also created several physical layer standards for different LAN protocols.

Figure : IEEE standard for LANs



LLC: Logical link control
MAC: Media access control

Upper layers

Data link layer

Physical layer

Transmission medium

OSI or Internet model

Upper layers

LLC

Ethernet MAC

Token Ring MAC

Token Bus MAC

• • •

Ethernet physical layers (several)

Token Ring physical layer

Token Bus physical layer

• • •

Transmission medium

IEEE Standard

Data Link Layer

The data link layer in the IEEE standard is divided into two sublayers: LLC and MAC.

Logical Link Control (LLC)

- In IEEE project 802, flow control , error control, and part of the framing duties are collected into one sublayer called the logical link control (LLC).
- LLC provides one single data link control for all IEEE LANs.

Framing :-

- LLC defines a protocol data unit (PDU) that is somewhat similar to that of HDLC.

  HDLC is High Level Data Link Control protocol. It falls under the ISO standards.

- The header contains a control field; this field is used for flow and error control.
- The two other header fields define the upper-layer protocol at the source and destination. These fields are called the destination service access point (DSAP) and the source service access point (SSAP).

- A frame defined in HDLC is divided into a PDU at the LLC sublayer and a frame at the MAC sublayer

Need for LLC :-

The purpose of the LLC is to provide flow and error control for the upper-layer protocols that actually demand these services.

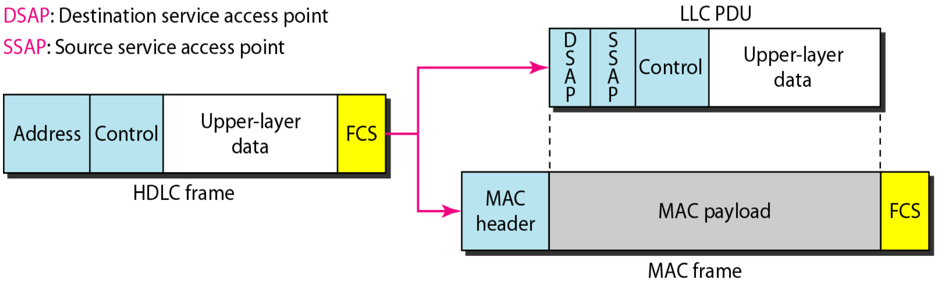
Media Access Control (MAC)

- IEEE Project 802 has created a sublayer called media access control that defines the specific access method for each LAN.

Figure : *HDLC frame compared with LLC and MAC frames*

DSAP: Destination service access point

SSAP: Source service access point



For example:

- CSMA/CD as media access method for Ethernet LANs.
- Token passing method for Token Ring and Token Bus LANs.

✓ In contrast to the LLC, MAC contains a number of distinct modules: each defines the access method and the framing format specific to the corresponding LAN protocol

✓ Framing is handled in both the LLC and MAC sublayer.

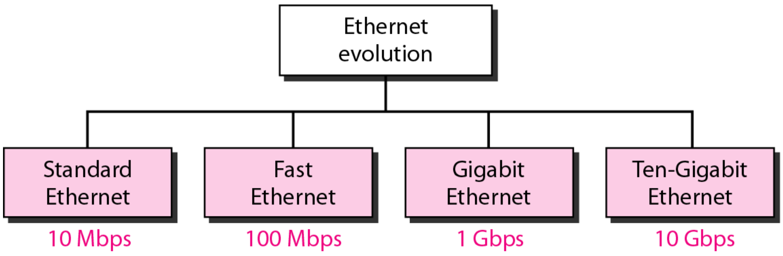
Physical Layer

- Physical layer is dependent on the implementation and type of the physical media used.
- IEEE define detailed specifications for each LAN implementation.
- For example, although there is only one MAC sublayer for Standard Ethernet(CSMA/CD), there is a different physical layer specifications for each Ethernet implementations.

STANDARD ETHERNET

The original Ethernet was created in 1976 at Xerox's Palo Alto Research Center (PARC). Since then, it has gone through four generations

Figure : *Ethernet evolution through four generations*



STANDARD ETHERNET

MAC Sublayer

In Standard Ethernet, the MAC sublayer governs the operation of the access method. It also frames data received from the upper layer and passes them to the physical layer.

Frame Format

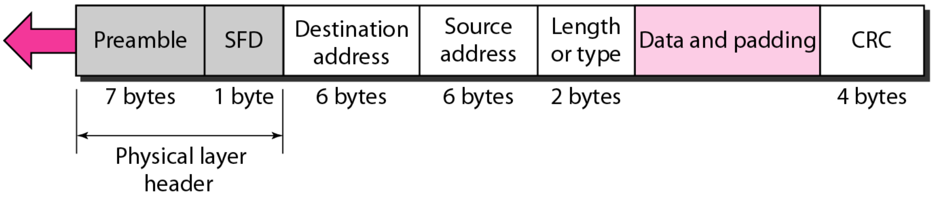
The Ethernet frame contains seven fields:

1. Preamble
2. SFD
3. DA
4. SA
5. Length or type of protocol data unit (PDU),
6. Upper-layer data
7. The CRC

Figure : *802.3 MAC frame*

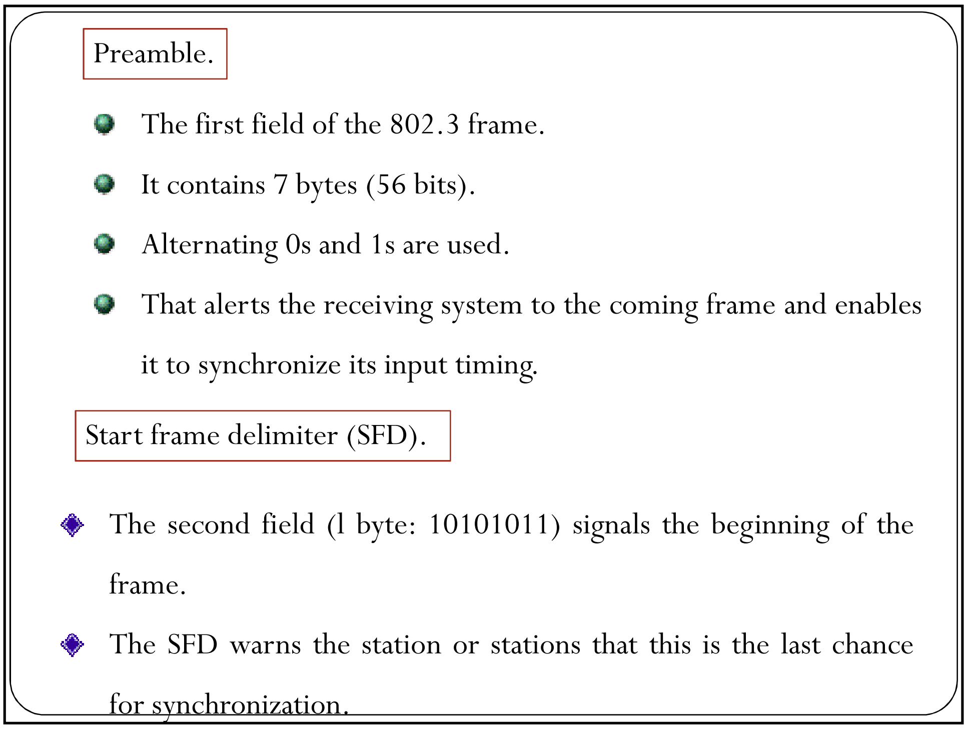
Preamble: 56 bits of alternating 1s and 0s.

SFD: Start frame delimiter, flag (10101011)



Preamble	SFD	Destination address	Source address	Length or type	Data and padding	CRC
7 bytes	1 byte	6 bytes	6 bytes	2 bytes		4 bytes

Physical layer header

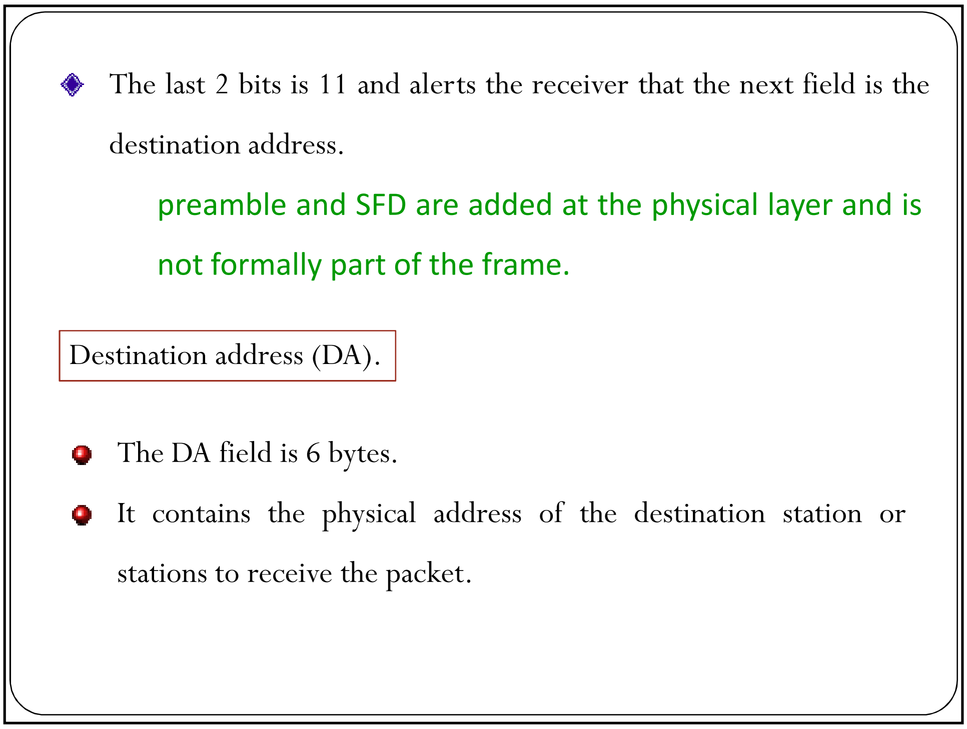


Preamble.

- The first field of the 802.3 frame.
- It contains 7 bytes (56 bits).
- Alternating 0s and 1s are used.
- That alerts the receiving system to the coming frame and enables it to synchronize its input timing.

Start frame delimiter (SFD).

- The second field (l byte: 10101011) signals the beginning of the frame.
- The SFD warns the station or stations that this is the last chance for synchronization.

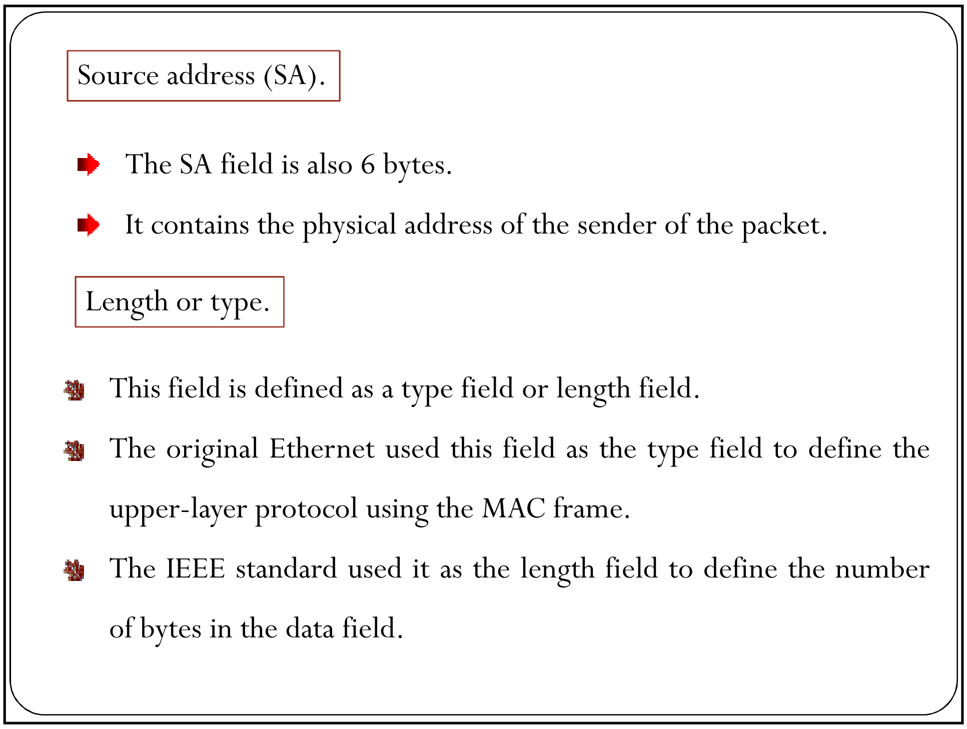


- The last 2 bits is 11 and alerts the receiver that the next field is the destination address.

preamble and SFD are added at the physical layer and is not formally part of the frame.

Destination address (DA).

- The DA field is 6 bytes.
- It contains the physical address of the destination station or stations to receive the packet.

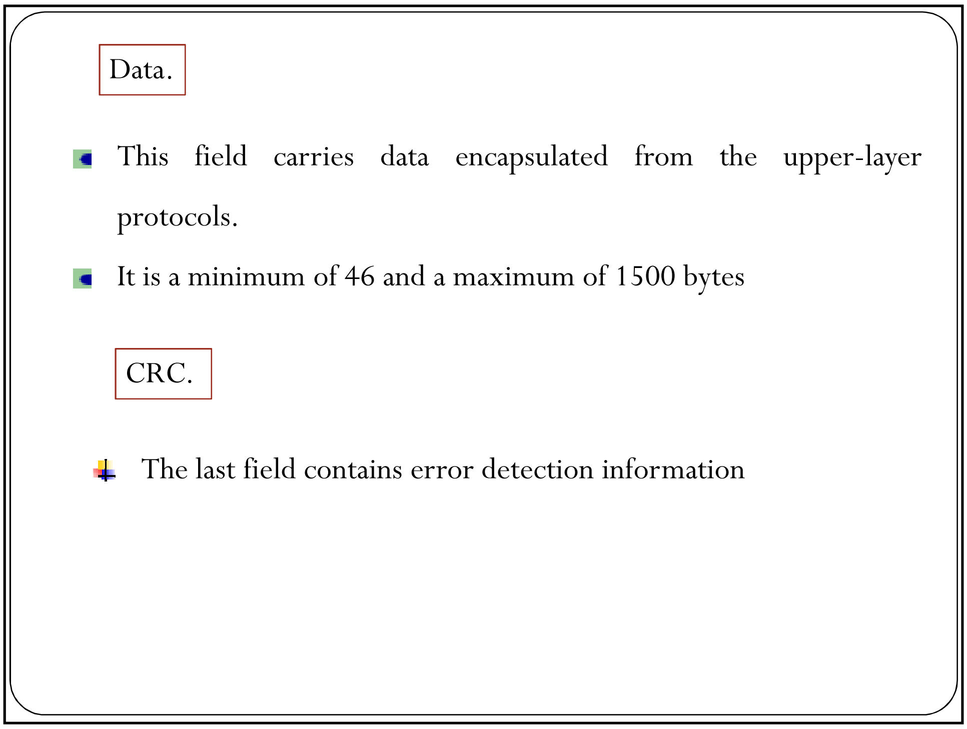


Source address (SA).

- The SA field is also 6 bytes.
- It contains the physical address of the sender of the packet.

Length or type.

- This field is defined as a type field or length field.
- The original Ethernet used this field as the type field to define the upper-layer protocol using the MAC frame.
- The IEEE standard used it as the length field to define the number of bytes in the data field.



Data.

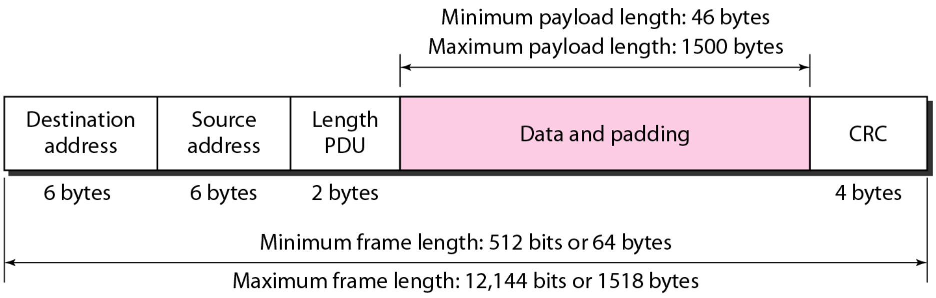
- This field carries data encapsulated from the upper-layer protocols.
- It is a minimum of 46 and a maximum of 1500 bytes

CRC.

- The last field contains error detection information

Frame Length

Figure : *Minimum and maximum lengths*



Minimum payload length: 46 bytes

Maximum payload length: 1500 bytes

Destination address	Source address	Length PDU	Data and padding	CRC
6 bytes	6 bytes	2 bytes		4 bytes

Minimum frame length: 512 bits or 64 bytes

Maximum frame length: 12,144 bits or 1518 bytes

- Minimum frame length restriction(64 bytes) is required for the correct operation of CSMA/CD.
    - Min data length =64 -18 (6+-6+2+4) = 46 bytes
    - If the upper-layer packet is less than 46 bytes, padding is added to make up the difference.
- Maximum length restriction; two historical reasons:
    - Memory was very expensive when Ethernet was designed.
    - Prevents one station from monopolizing the shared medium, blocking other stations that have data to sent.
- Max data length =1518-18= 1500 bytes.

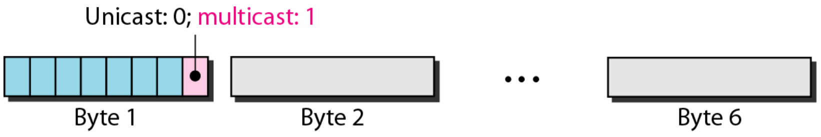
Addressing

- Each station (PC or printer) has a network interface card (NIC) which provides the station with a 6-byte [48 bits] physical address (MAC address)
- It is written in hexadecimal notation, with a colon between the bytes.

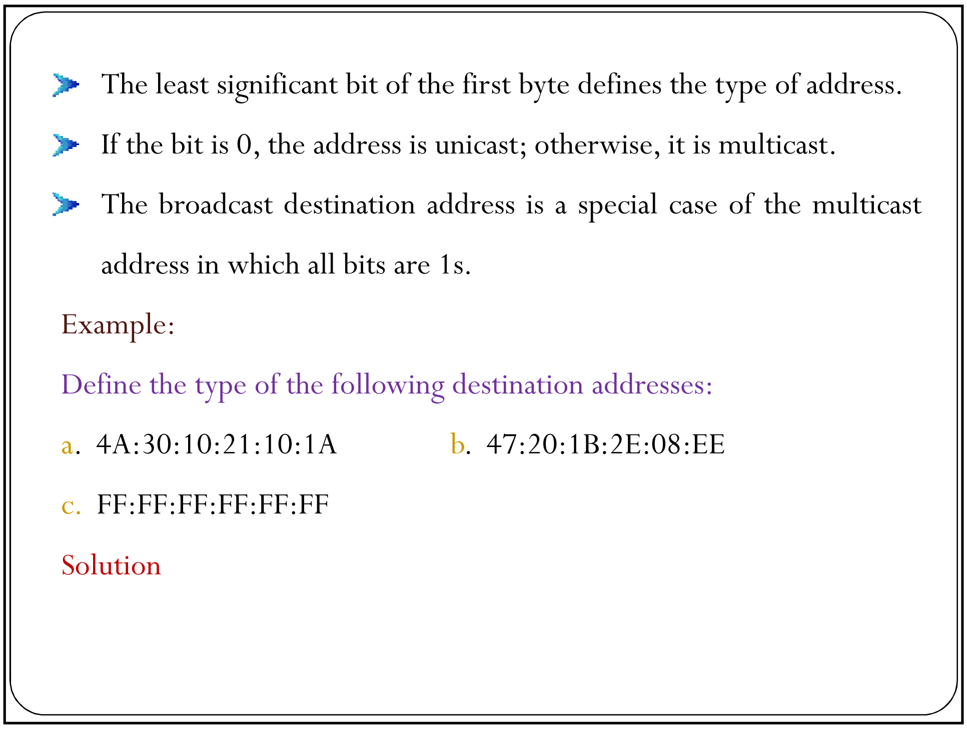
Figure : *Example of an Ethernet address in hexadecimal notation*

$$\underbrace{06:01:02:01:2C:4B}_{\text{6 bytes = 12 hex digits = 48 bits}}$$

Unicast, Multicast, and Broadcast Addresses



- Source address is always a unicast address –the frames comes from only one station.
- Destination address can be:
  - unicast: defines only one recipient; one to one
  - multicast: a group of addresses; one to many
  - Broadcast: the recipients are all the stations on the LAN



- The least significant bit of the first byte defines the type of address.
- If the bit is 0, the address is unicast; otherwise, it is multicast.
- The broadcast destination address is a special case of the multicast address in which all bits are 1s.

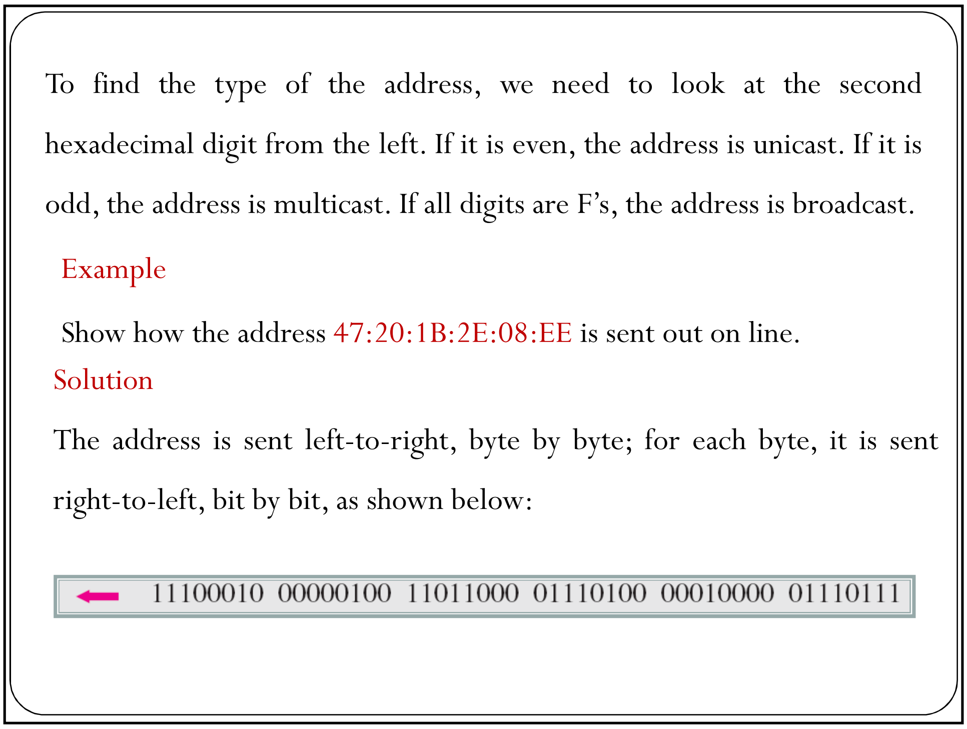
Example:

Define the type of the following destination addresses:

a. 4A:30:10:21:10:1A b. 47:20:1B:2E:08:EE

c. FF:FF:FF:FF:FF:FF

Solution



To find the type of the address, we need to look at the second hexadecimal digit from the left. If it is even, the address is unicast. If it is odd, the address is multicast. If all digits are F's, the address is broadcast.

Example

Show how the address 47:20:1B:2E:08:EE is sent out on line.

Solution

The address is sent left-to-right, byte by byte; for each byte, it is sent right-to-left, bit by bit, as shown below:

← 11100010 00000100 11011000 01110100 00010000 01110111

Slot Time

- Slot time defined in bits:

  It is the time required for a station to send 512 bits (min frame size).

Slot Time = round trip time + time required to send the jam sequence

- It depends on the data rate, for traditional 10-Mbps Ethernet it is 51.2 microseconds (512/10Mbps)
- The choice of 512-bit Slot Time to allow the proper function of CSMA/CD.

If the sender sends a frame larger than the minimum size (512 to 1518 bits)

- If the station has sent out the first 512 bits and has not heard a collision, it is guaranteed that the collision will never occur during the transmission of the frame.
- The reason is that all stations sensed the existence of the signal and refrained from sending .
- Collisions can only occur during the first half of the slot time (slot time/2), and if it does, it can be sensed by the sender during the slot time.

Slot Time and Maximum Network Length

Slot time $= 2 \times T_p$ (neglecting time required to send jam signal)

$= 2$ x Max Length/propagation speed

Max.Length = propagation speed x (SlotTime/2)

Let propagation speed = $2\text{x}10^8$m/s

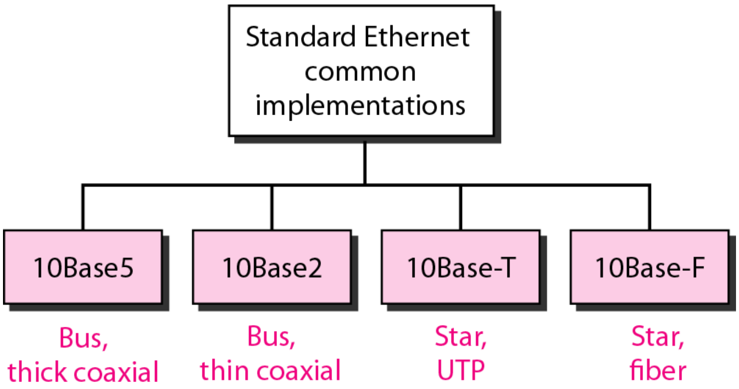
Max.Length = $(2 \times 10^8) \times (51.2 \times 10^{-6}/2) = 5120$ m

Consider the delay times in repeaters and interfaces, and the time required to send the jam sequence.

Max.Length = 2500 m (= 48 % of the theoretical)

Physical Layer

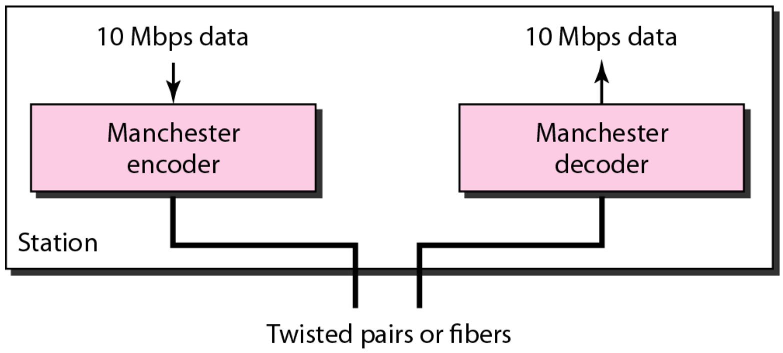
The Standard Ethernet defines several physical layer implementation, four of the most common:



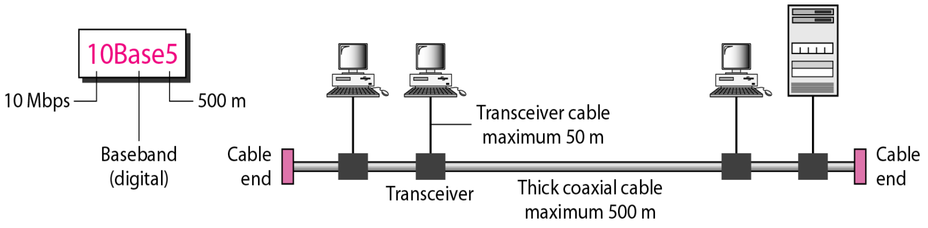
Encoding and Decoding

- All standard implementations use digital signaling(baseband) at 10 Mbps.
- At the sender, data are converted to a digital signal using the Manchester scheme.
- At the receiver, the received signal is interpreted as Manchester and decoded into data.
- Uses CSMA/CD with 1-persistent

Figure : *Encoding in a Standard Ethernet implementation*



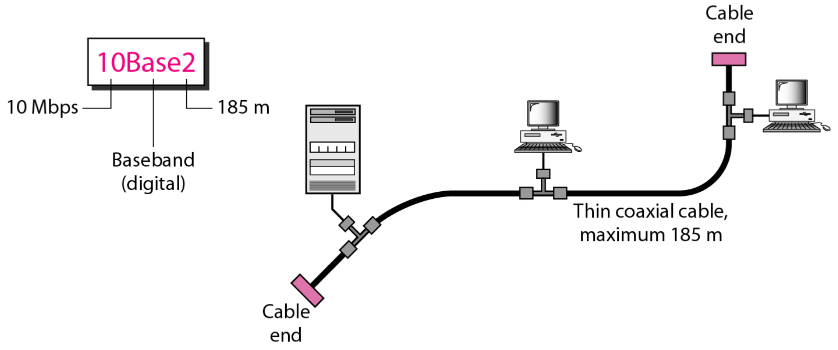
10Base5:Thick Ethernet (Thicknet)



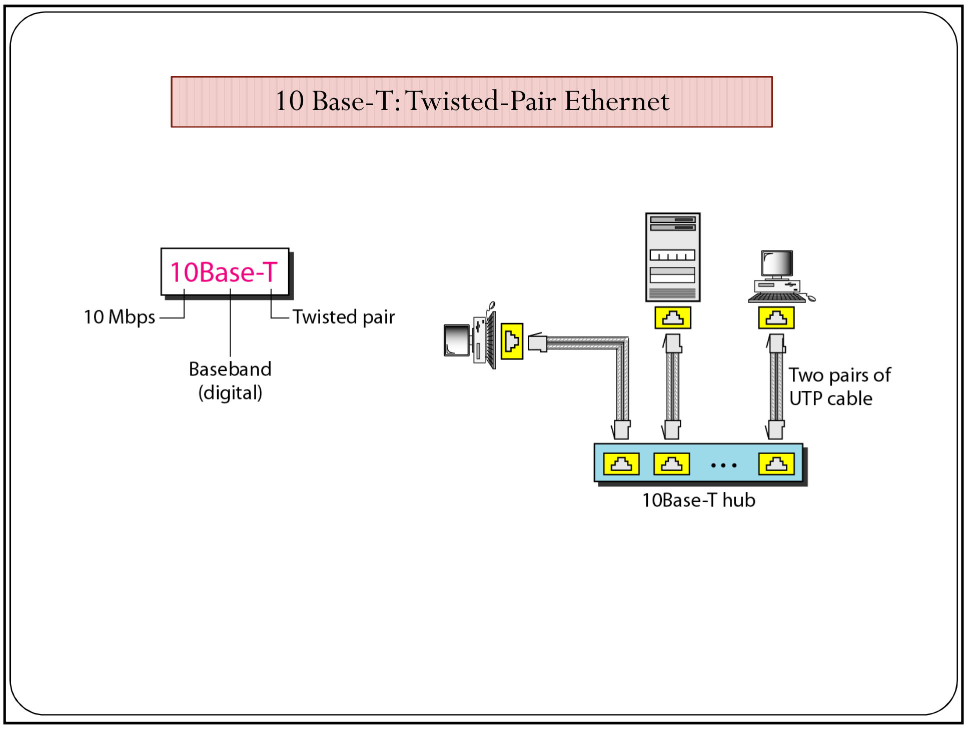
- Which is roughly the size of a garden hose and too stiff to bend with your hands.
- Uses coaxial cable and Bus topology
- With an external transceiver(transmitter/receiver) connected via a tap.

- Transceiver is responsible for:
  - transmitting, receiving and
  - detecting collisions.
- The length of each segment cannot exceed 500 m
  - If cable > 500 m ,degradation in the signal, using repeaters to connect multiple “segments” of cable.
- No two stations can be separated by more than 2500m(max length of the bus) and 4 repeaters.

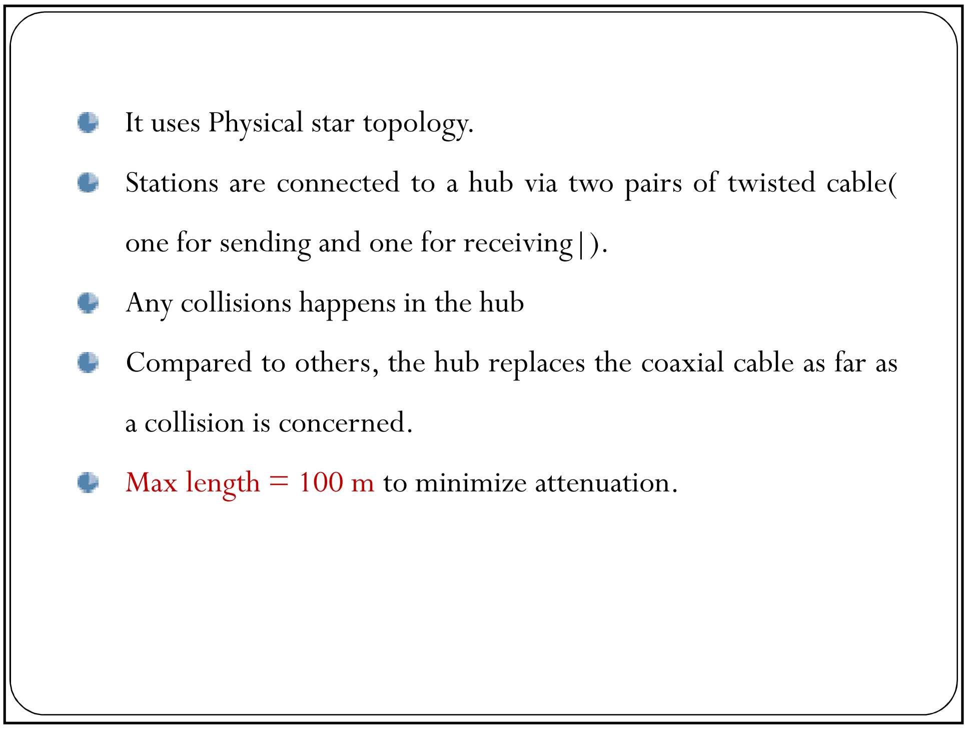
10Base2:Thin Ethernet (Cheapernet)



- Uses Bus topology with thinner and more flexible cables.
- Transceiver is part of a NIC (Network Interface card) card.
- The implementation is more cost effective than 10Base5 because:
  - Thin coaxial cable is less expensive than the thick
  - tee connections are cheaper than taps
- Installation is simpler because thin coaxial cable is very flexible.
- The length of each segment under 200 (cannot exceed 185 m) due to the high level of attenuation
- Repeaters are used to connect multiple segments

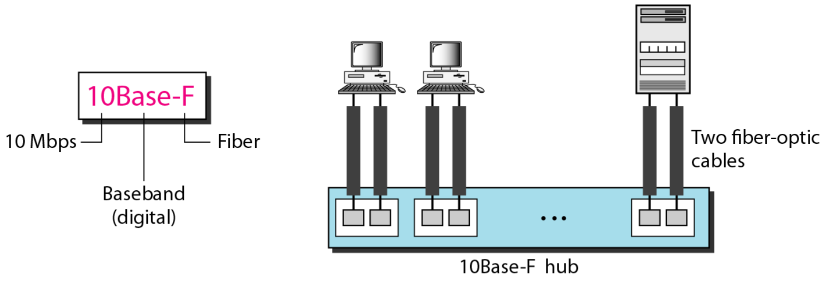


10 Base-T: Twisted-Pair Ethernet



- It uses Physical star topology.
- Stations are connected to a hub via two pairs of twisted cable(one for sending and one for receiving|).
- Any collisions happens in the hub
- Compared to others, the hub replaces the coaxial cable as far as a collision is concerned.
- Max length = 100 m to minimize attenuation.

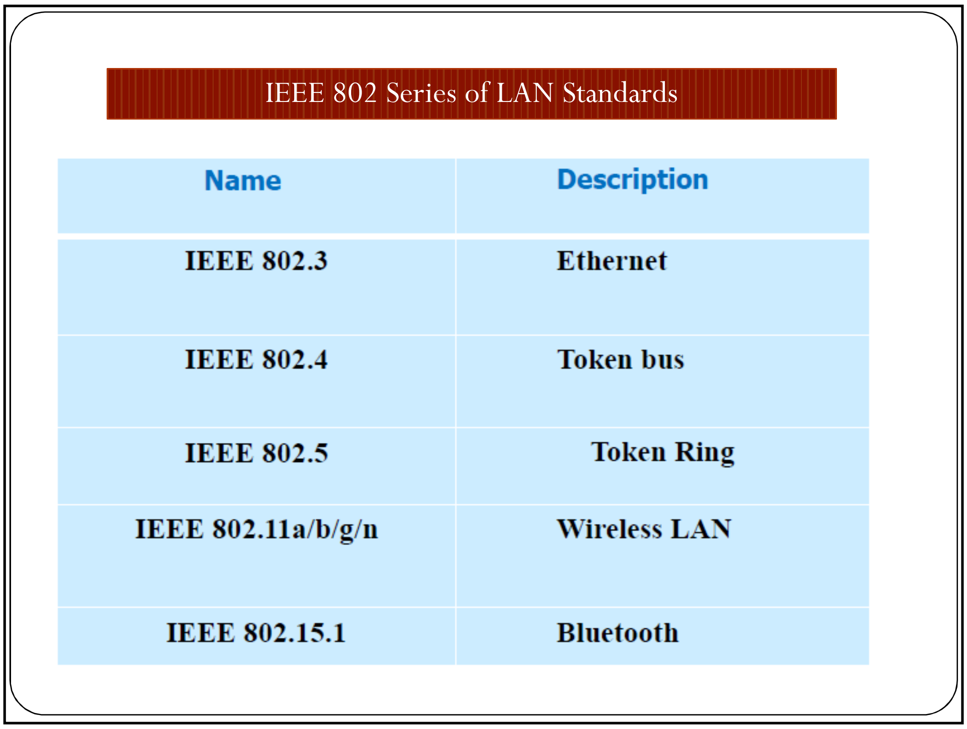
10Base-F: Fiber Ethernet



- Uses star topology to connect stations to a hub
- Stations is connected to the hub by using two pairs of fiber-optic cables.

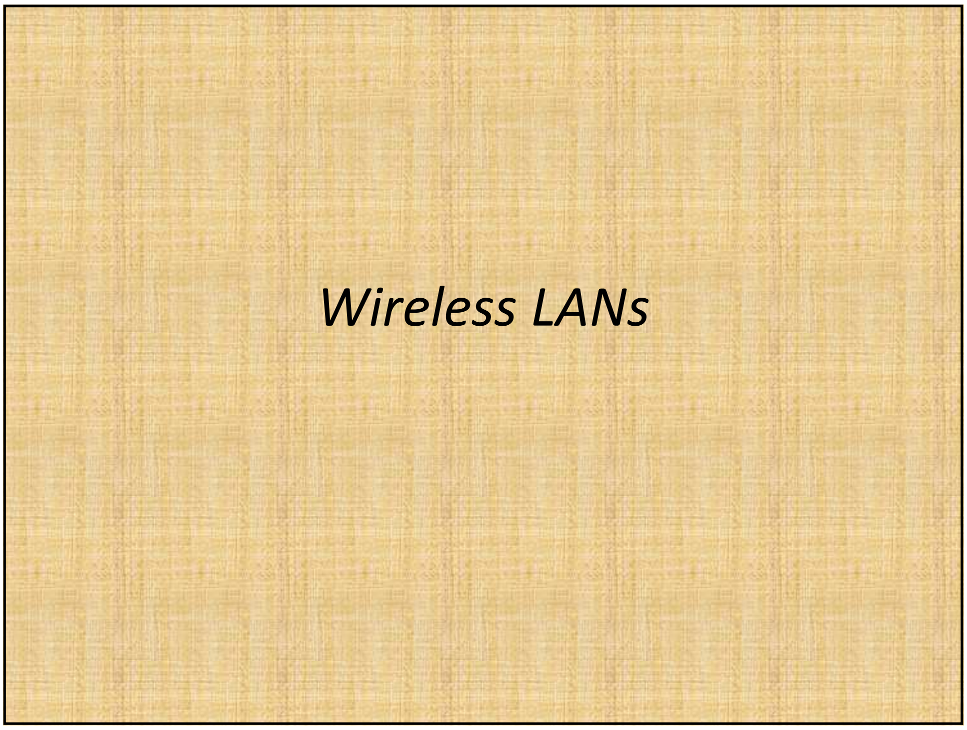
Summary of Standard Ethernet implementations

	10Base 5	10Base2	10Base-T	10Base-F
Media	Thick coaxial cable	Thin coaxial cable	Two UTP	2 Fiber
Maximum length	500 m	185 m	100 m	2000 m
Topology	Bus	Bus	Star	star
Data rate	10Mbps	10Mbps	10Mbps	10Mbps
Line coding	Manchester	Manchester	Manchester	Manchester

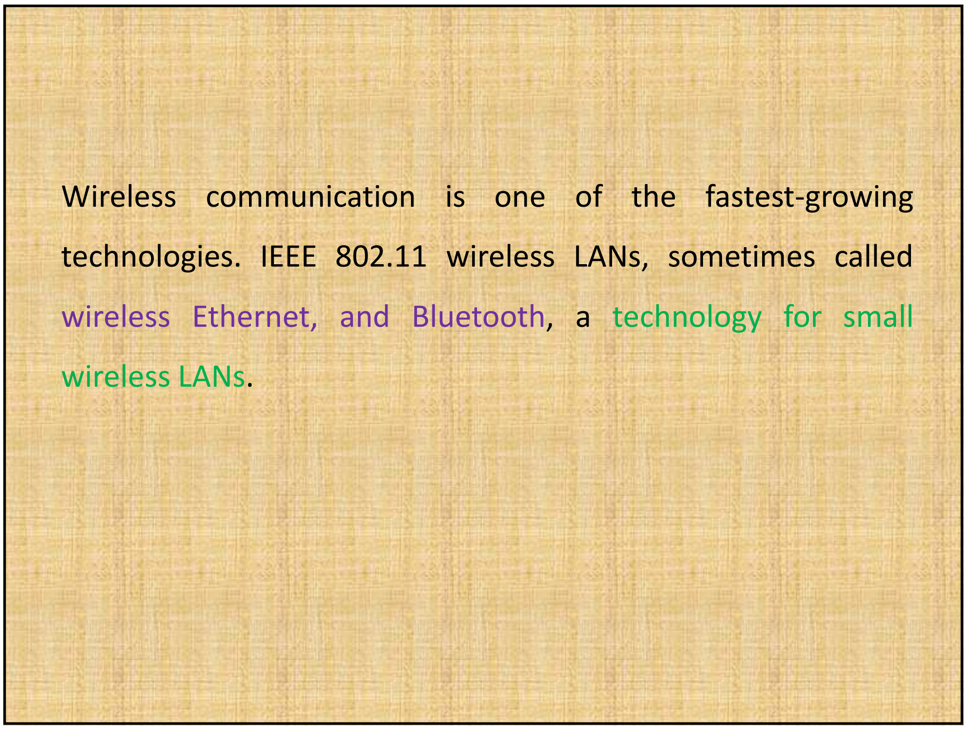


IEEE 802 Series of LAN Standards

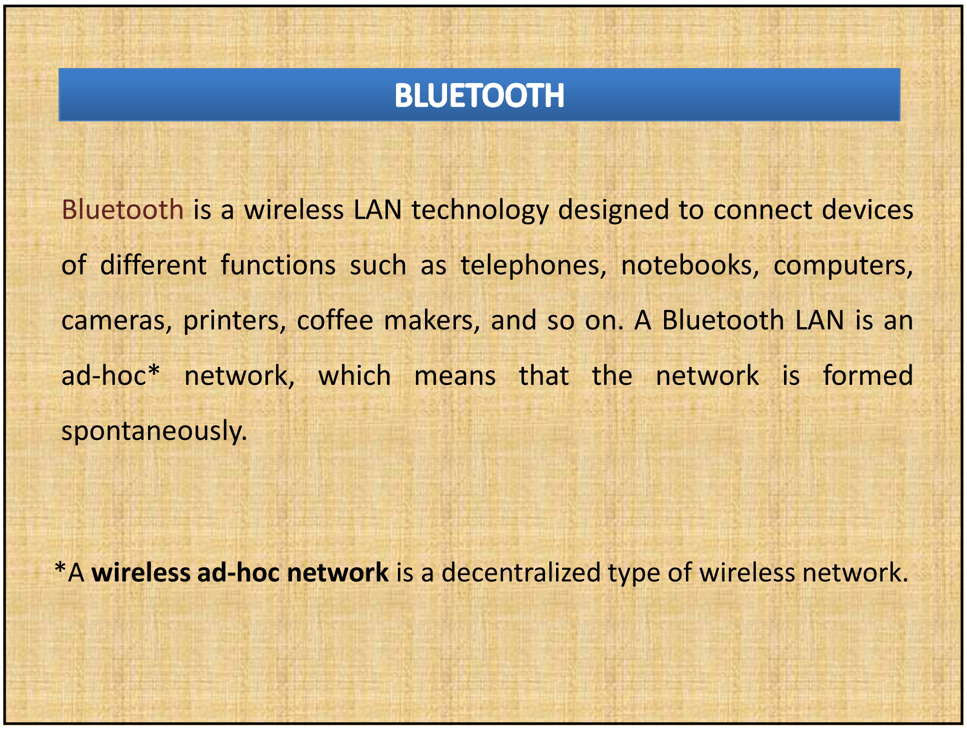
Name	Description
IEEE 802.3	Ethernet
IEEE 802.4	Token bus
IEEE 802.5	Token Ring
IEEE 802.11a/b/g/n	Wireless LAN
IEEE 802.15.1	Bluetooth



Wireless LANs



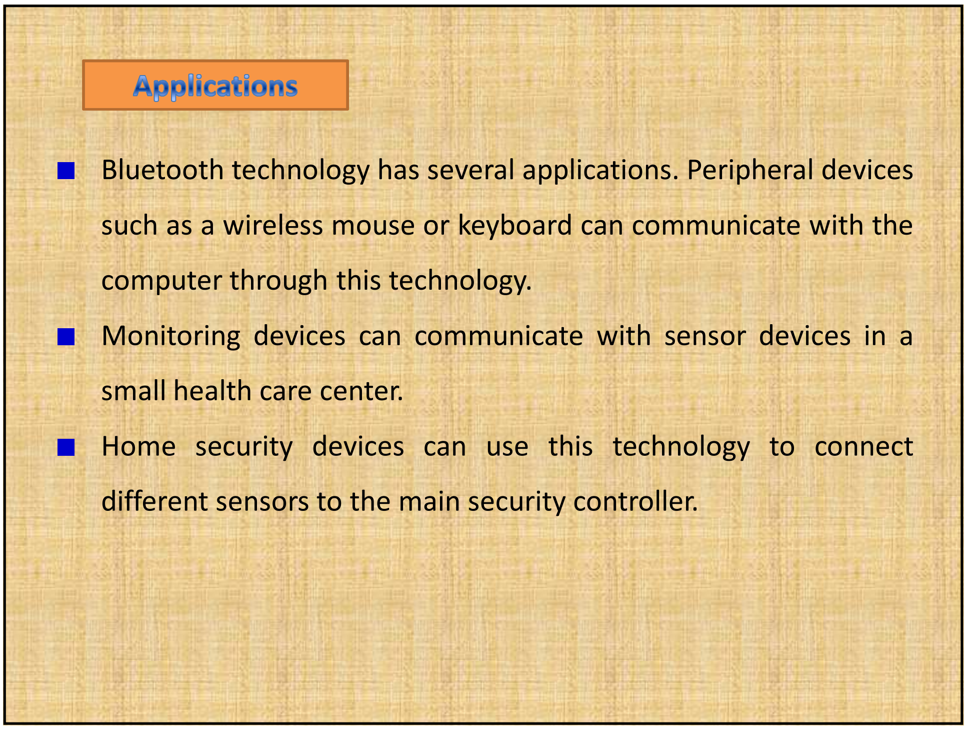
Wireless communication is one of the fastest-growing technologies. IEEE 802.11 wireless LANs, sometimes called wireless Ethernet, and Bluetooth, a technology for small wireless LANs.



BLUETOOTH

Bluetooth is a wireless LAN technology designed to connect devices of different functions such as telephones, notebooks, computers, cameras, printers, coffee makers, and so on. A Bluetooth LAN is an ad-hoc* network, which means that the network is formed spontaneously.

*A **wireless ad-hoc network** is a decentralized type of wireless network.



Applications

- Bluetooth technology has several applications. Peripheral devices such as a wireless mouse or keyboard can communicate with the computer through this technology.
- Monitoring devices can communicate with sensor devices in a small health care center.
- Home security devices can use this technology to connect different sensors to the main security controller.

Architecture

Bluetooth defines two types of networks:

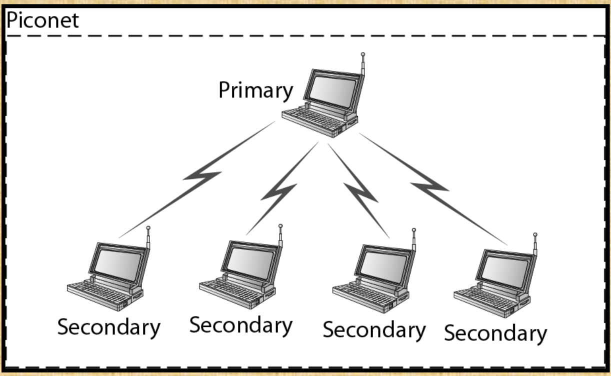
piconet and scatternet.

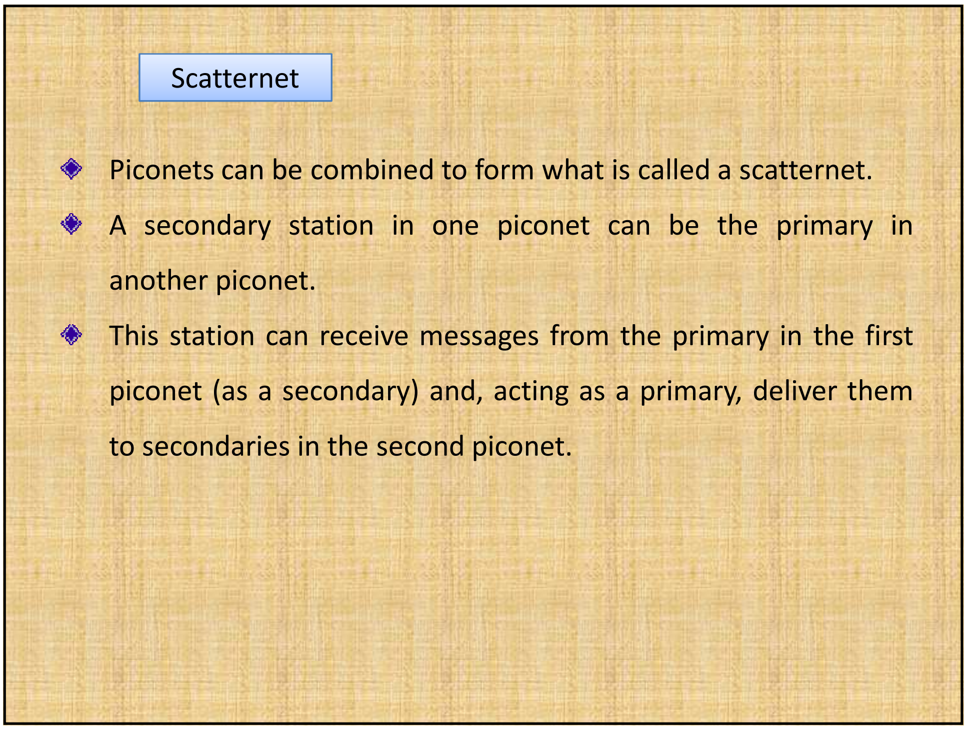
Piconet

- A Bluetooth network is called a piconet, or a small net.
- It can have up to eight stations, one of which is called the primary; the rest are called secondaries.
- Maximum of seven secondaries. Only one primary.
- Secondaries synchronize their clocks and hopping sequence with the primary.

- But an additional eight secondaries can stay in parked state, which means they can be synchronized with the primary but cannot take part in communication until it is moved from the parked state.

Figure : *Piconet*

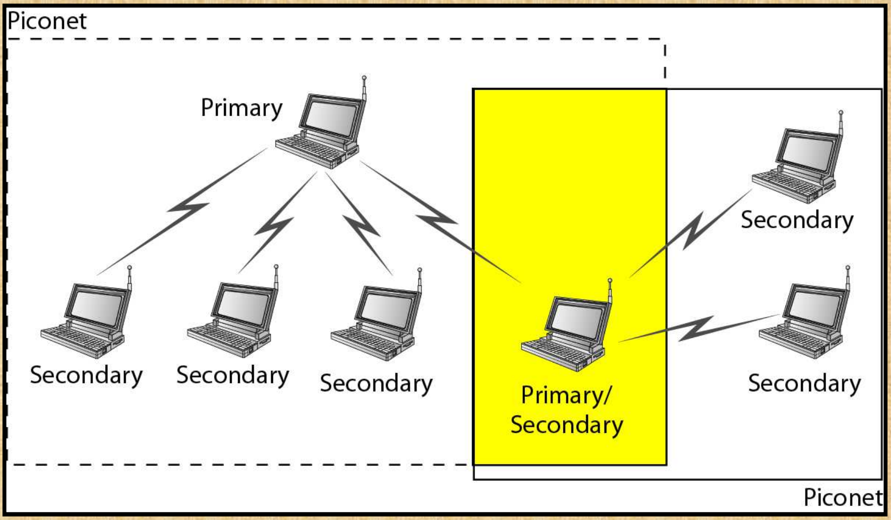




Scatternet

- Piconets can be combined to form what is called a scatternet.
- A secondary station in one piconet can be the primary in another piconet.
- This station can receive messages from the primary in the first piconet (as a secondary) and, acting as a primary, deliver them to secondaries in the second piconet.

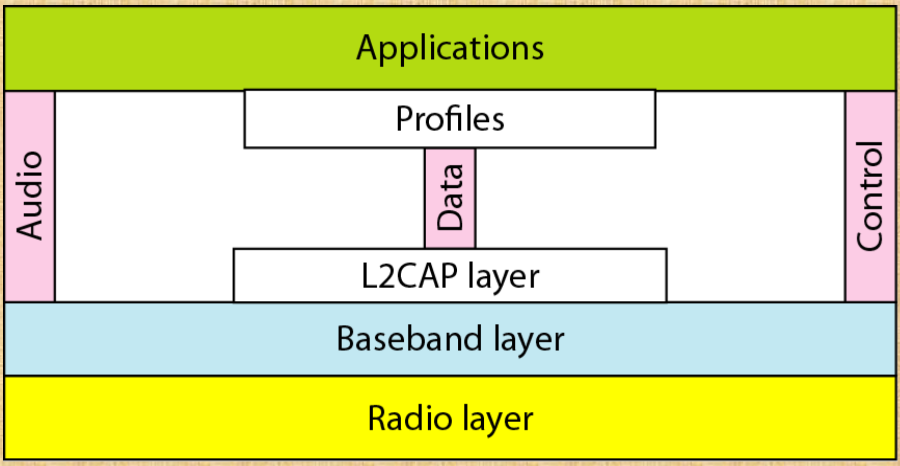
Figure : *Scatternet*



Bluetooth Devices

- A Bluetooth device has a built-in short-range radio transmitter.
- The current data rate is 1 Mbps with a 2.4-GHz bandwidth.

Bluetooth Layers



Applications

Profiles

Audio

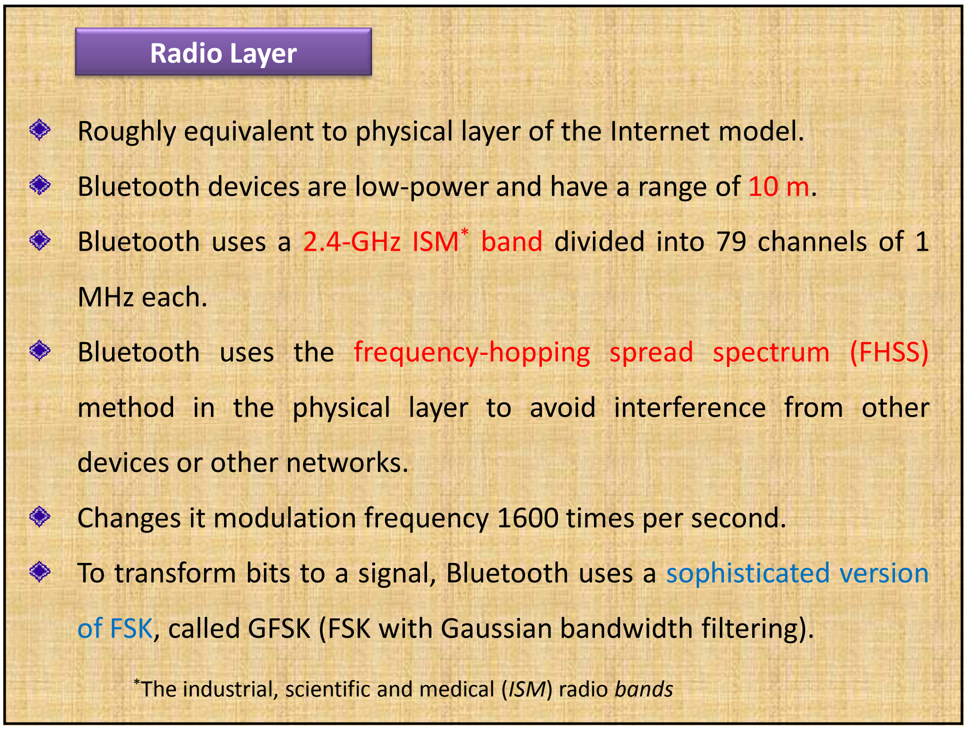
Data

Control

L2CAP layer

Baseband layer

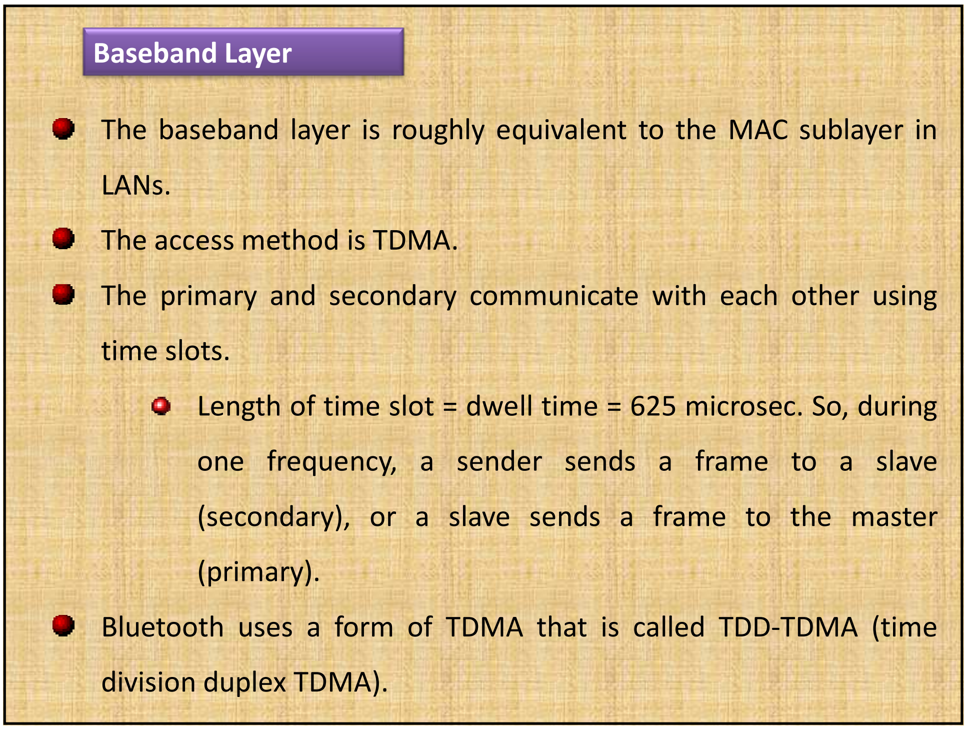
Radio layer



Radio Layer

- Roughly equivalent to physical layer of the Internet model.
- Bluetooth devices are low-power and have a range of 10 m.
- Bluetooth uses a 2.4-GHz ISM[*] band divided into 79 channels of 1 MHz each.
- Bluetooth uses the frequency-hopping spread spectrum (FHSS) method in the physical layer to avoid interference from other devices or other networks.
- Changes it modulation frequency 1600 times per second.
- To transform bits to a signal, Bluetooth uses a sophisticated version of FSK, called GFSK (FSK with Gaussian bandwidth filtering).

[*] The industrial, scientific and medical (*ISM*) radio *bands*



Baseband Layer

- The baseband layer is roughly equivalent to the MAC sublayer in LANs.
- The access method is TDMA.
- The primary and secondary communicate with each other using time slots.
  - Length of time slot = dwell time = 625 microsec. So, during one frequency, a sender sends a frame to a slave (secondary), or a slave sends a frame to the master (primary).
- Bluetooth uses a form of TDMA that is called TDD-TDMA (time division duplex TDMA).

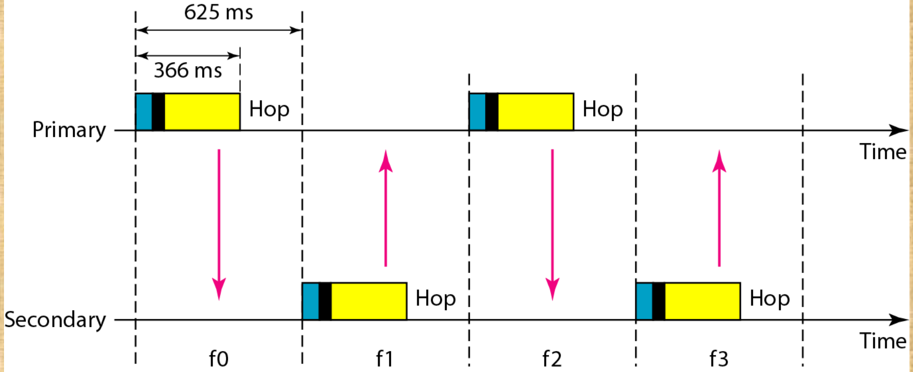
- Time division duplexing TDMA (TDD-TDMA) is a kind of half-duplex communication in which the slave and receiver send and receive data, but not at the same time (half-duplex). However, the communication for each direction uses different hops, like walkie-talkies.

Single-Secondary Communication

- If the piconet has only one secondary, the TDMA operation is very simple.
- The time is divided into slots of 625 μs.
- The primary uses even numbered slots; the secondary uses odd numbered slots.

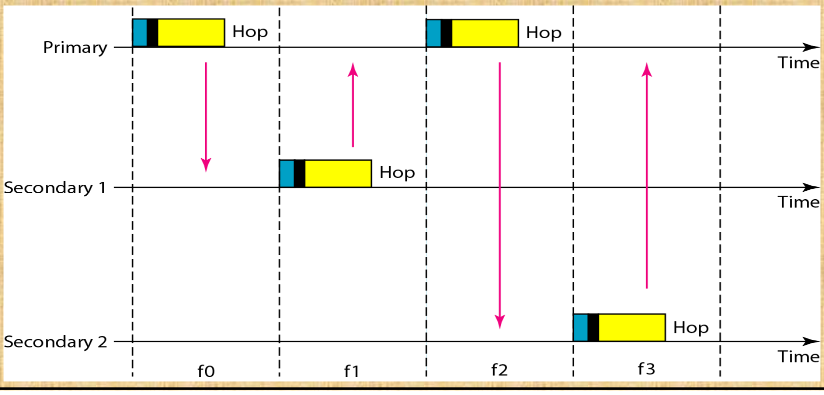
- TDD-TDMA allows the primary and the secondary to communicate in half-duplex mode.
- In slot 0, the primary sends, and the secondary receives; in slot 1, the secondary sends, and the primary receives. The cycle is repeated.

Figure : *Single-secondary communication*



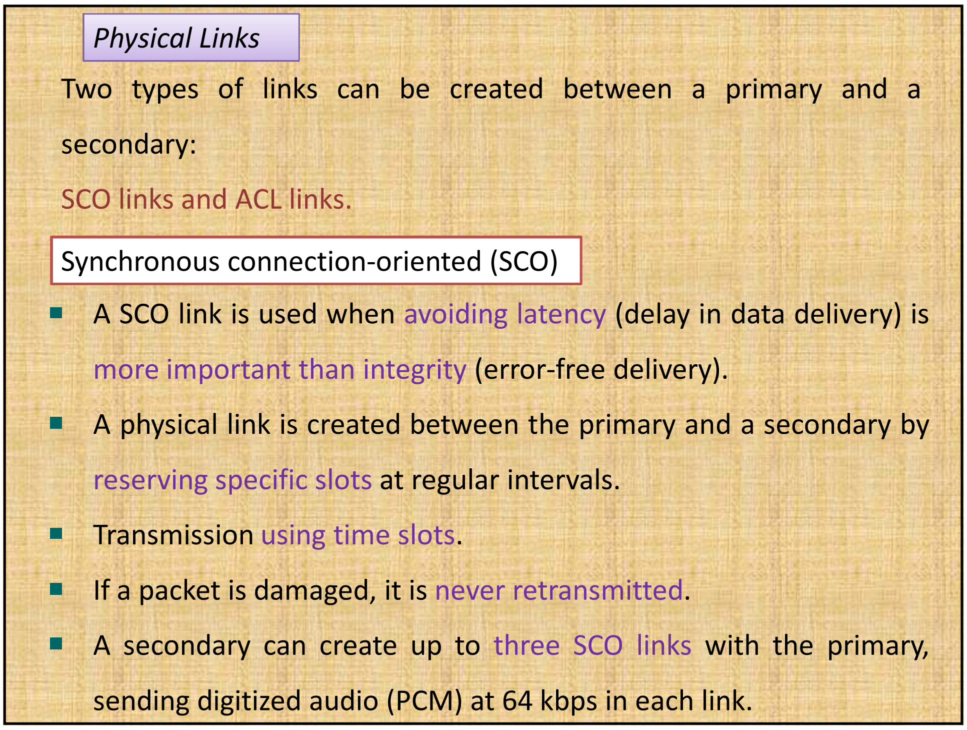
Multiple-Secondary Communication

- Primary uses the even-numbered slots.
- secondary sends in the next odd-numbered slot if the packet in the previous slot was addressed to it.
- All secondaries listen on even-numbered slots, but only one secondary sends in any odd-numbered slot.



Let us elaborate on the figure.

- In slot 0, the primary sends a frame to secondary 1.
- In slot 1, only secondary I sends a frame to the primary because the previous frame was addressed to secondary 1; other secondaries are silent.
- In slot 2, the primary sends a frame to secondary 2.
- In slot 3, only secondary 2 sends a frame to the primary because the previous frame was addressed to secondary 2; other secondaries are silent.
- The cycle continues.



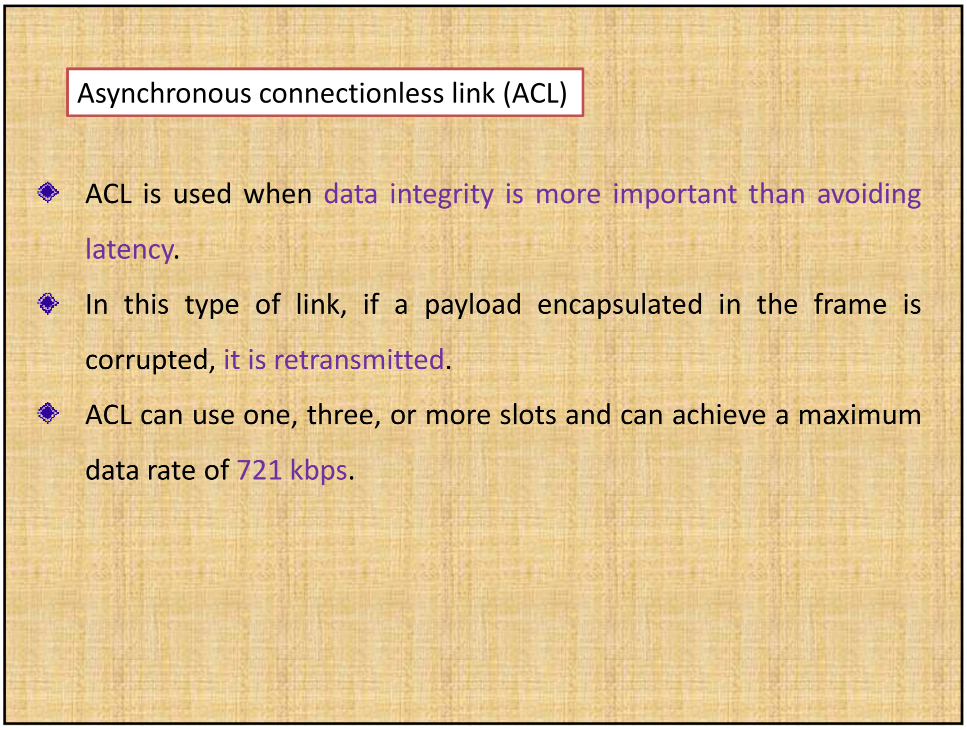
Physical Links

Two types of links can be created between a primary and a secondary:

SCO links and ACL links.

Synchronous connection-oriented (SCO)

- A SCO link is used when avoiding latency (delay in data delivery) is more important than integrity (error-free delivery).
- A physical link is created between the primary and a secondary by reserving specific slots at regular intervals.
- Transmission using time slots.
- If a packet is damaged, it is never retransmitted.
- A secondary can create up to three SCO links with the primary, sending digitized audio (PCM) at 64 kbps in each link.



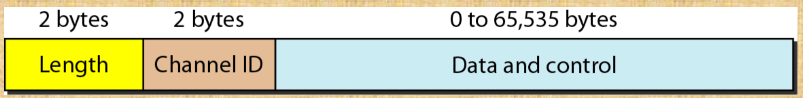
Asynchronous connectionless link (ACL)

- ACL is used when data integrity is more important than avoiding latency.
- In this type of link, if a payload encapsulated in the frame is corrupted, it is retransmitted.
- ACL can use one, three, or more slots and can achieve a maximum data rate of 721 kbps.

L2CAP

- The Logical Link Control and Adaptation Protocol, or L2CAP (L2 here means LL), is roughly equivalent to the LLC sublayer in LANs.
- It is used for data exchange on an ACL link; SCO channels do not use L2CAP.

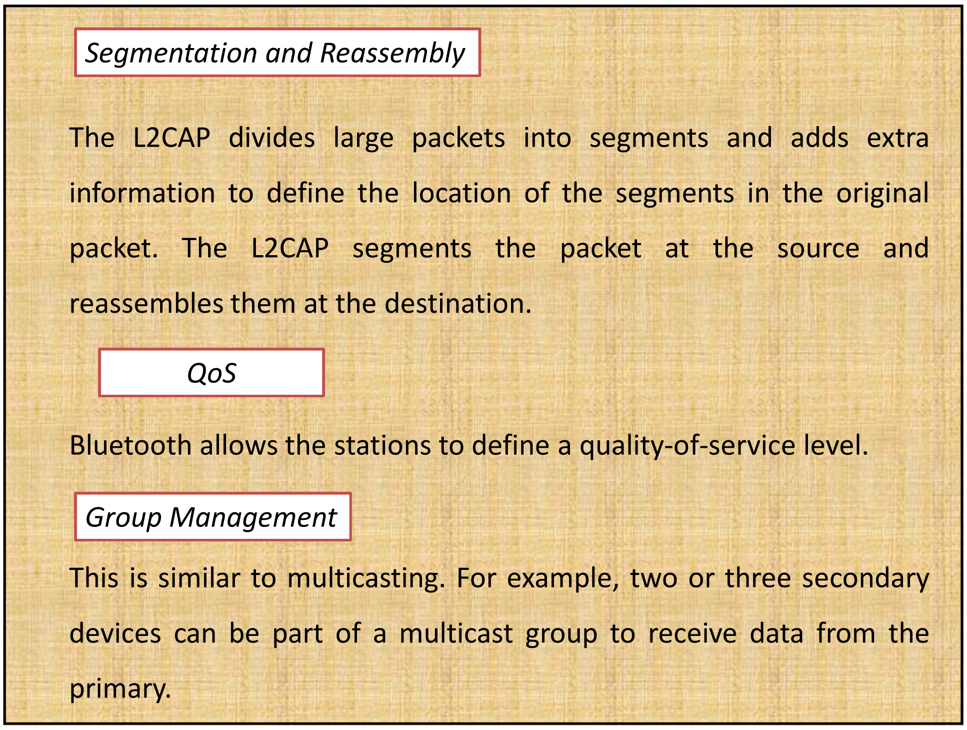
Figure : *L2CAP data packet format*



- The I6-bit length field defines the size of the data, in bytes, coming from the upper layers.
- The channel ID (CID) defines a unique identifier for the virtual channel created at this level.
- The L2CAP has specific duties: multiplexing, segmentation and reassembly, quality of service (QoS), and group management.

Multiplexing

The L2CAP can do multiplexing. At the sender site, it accepts data from one of the upper-layer protocols, frames them, and delivers them to the baseband layer. At the receiver site, it accepts a frame from the baseband layer, extracts the data, and delivers them to the appropriate protocol layer.



Segmentation and Reassembly

The L2CAP divides large packets into segments and adds extra information to define the location of the segments in the original packet. The L2CAP segments the packet at the source and reassembles them at the destination.

QoS

Bluetooth allows the stations to define a quality-of-service level.

Group Management

This is similar to multicasting. For example, two or three secondary devices can be part of a multicast group to receive data from the primary.