Chapter 3
Data Link Layer (DLL)

- ***Outline:***

  3.1 Functions of Data link layer

  3.2 Framing

  3.3 Error Detection and Corrections,

  3.4 Flow Control

  3.5 Examples of Data Link Protocol, HDLC, PPP

  3.6 The Medium Access Sub-layer

  3.7 The channel allocation problem

  3.8 Multiple Access Protocols

  3.9 Ethernet,

  3.10 Networks: FDDI, ALOHA, VLAN, CSMA/CD, IEEE 802.3(Ethernet), 802.4(Token Bus), 802.5(Token Ring), and 802.11(Wireless LAN).

Introduction to DLL

- Data link layer is layer 2 in OSI Model
- The Data Link Layer sits between the Network Layer and the Physical Layer.
- Concerned with local delivery of frames between devices on the same LAN/WAN
- It has a number of **functions :**

  1) Providing a well-defined service interface to the network layer to send information from one machine to another.
  2) Dealing with transmission errors.
  3) Regulating the flow of data so that slow receivers are not swamped by fast senders.

- The principal service is transferring data from the network layer on the source machine to the network layer on the destination machine

Packets and Frames

- The data link layer takes the packets it gets from the network layer and encapsulates them into **frames** for transmission.
- Each frame contains a **frame header**, a **payload field** for holding the packet, and a **frame trailer.**

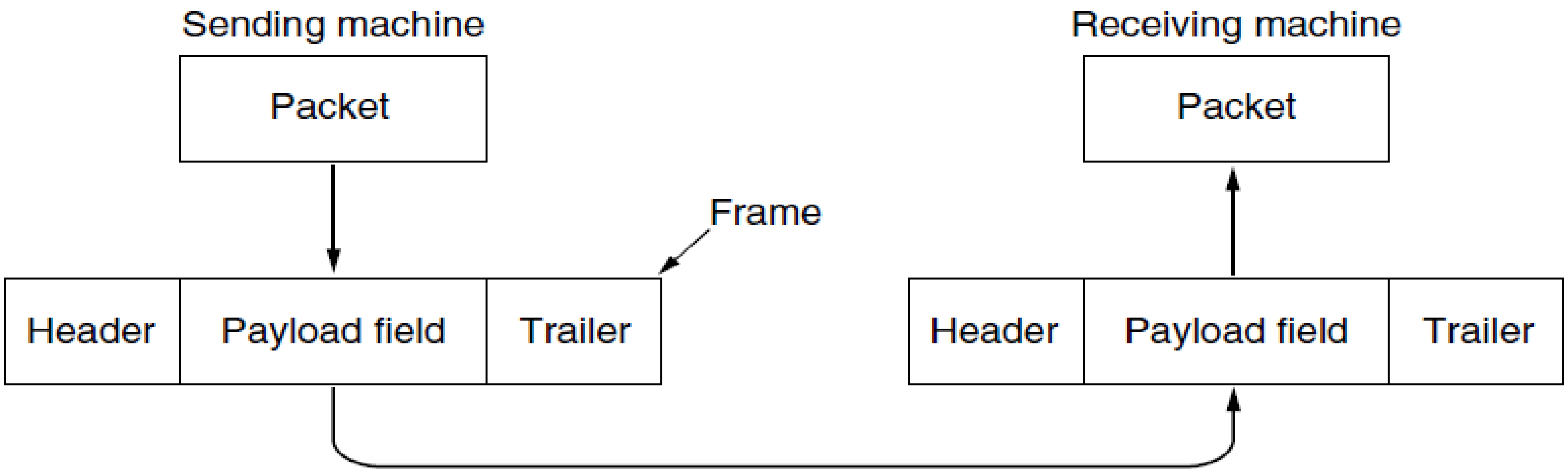


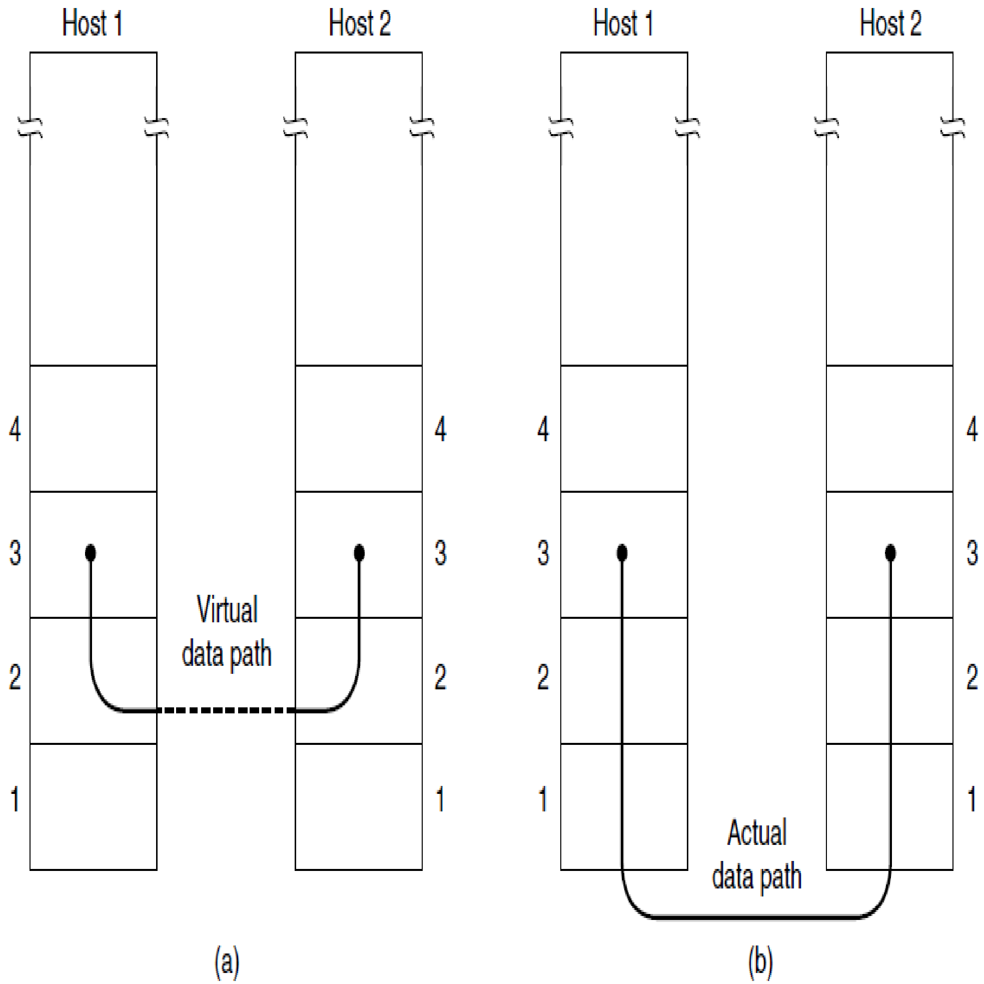
Figure 3-1. Relationship between packets and frames.

Data Link Layer Design issues

- Data Link Layer **Design issues:**
  1) Services provided to the Network Layer
  2) Framing
  3) Error control
  4) Flow control

Services provided to the Network Layer

- The principal service is **transferring data** from the **network** layer on the **source machine** to the **network layer on the destination** machine.
- The job of the data link layer is to transmit the **bits to the destination machine** so they can be handed over to the network layer there, as shown in Figure (a).
- The **actual transmission follows the path** of Figure (b), but it is easier to think in terms of two data link layer processes communicating using a data link protocol.



Services provided to the Network Layer

- The data link layer can be designed to offer **various services**.
- The actual services that are offered vary **from protocol to protocol**.

1) Unacknowledged connectionless service.
2) Acknowledged connectionless service.
3) Acknowledged connection-oriented service.

- **Unacknowledged connectionless service:**
  - No acknowledgement from the receiving machine.
  - No logical connection is set up between the two machines.
  - The DLL will make no attempt to detect the loss of or recover a lost frame.
  - This service is useful for low error rate networks and also appropriate for real-time traffic, such as voice, in which late data are worse than bad data.

Services provided to the Network Layer

- **Acknowledged connectionless service:**
  - There are still no logical connections used, but each frame sent is individually acknowledged.
  - If frame has not arrived within a specified time interval, it can be sent again.
  - This service is useful over unreliable channels, such as wireless systems.
- **Acknowledged connection-oriented service:**
  - A connection is established between the two machines.
  - The frames are then transmitted and each frame is acknowledged.
  - The frames are guaranteed to arrive only once and in order.
  - The connection is released once the communication is complete.

Framing

- The Process of **creating Frames** by the Data Link Layer is known as **Framing**.
- The usual approach is for the DLL to **break up the bit stream into discrete frames**, compute **a checksum** for each frame, and include the checksum in the frame when it **is transmitted**.
- When a frame arrives **at the destination**, the **checksum is recomputed**.
- If the newly computed **checksum is different** from the one contained in the frame, the data link layer knows that an **error has occurred** and **takes steps** to deal with it.
- **Breaking up** the bit stream into frames is **more difficult** than it at first appears.
- **Framing Method:**
  - ***Byte count***, ***Flag bytes with byte stuffing***, ***Flag bits with bit stuffing***

Framing Method: Byte/Character Count

- Uses a field **in the header** to specify the **number of bytes** in the frame.
- The **destination sees** the byte count, it knows **how many bytes follow** and hence where the end of the frame is.
- The **trouble with this algorithm** is that the count can **be garbled by a transmission** error. For example, if the byte count of 5 in the second frame of Figure-(b) becomes a 7 due to a single bit flip**, the destination will get out of synchronization.**
- Sending a **frame back to the source** asking for a retransmission **does not help either**, since the **destination does not know** how many bytes **to skip over to get to the start** of the retransmission.
- For this reason, the byte count method is **rarely used** by itself.

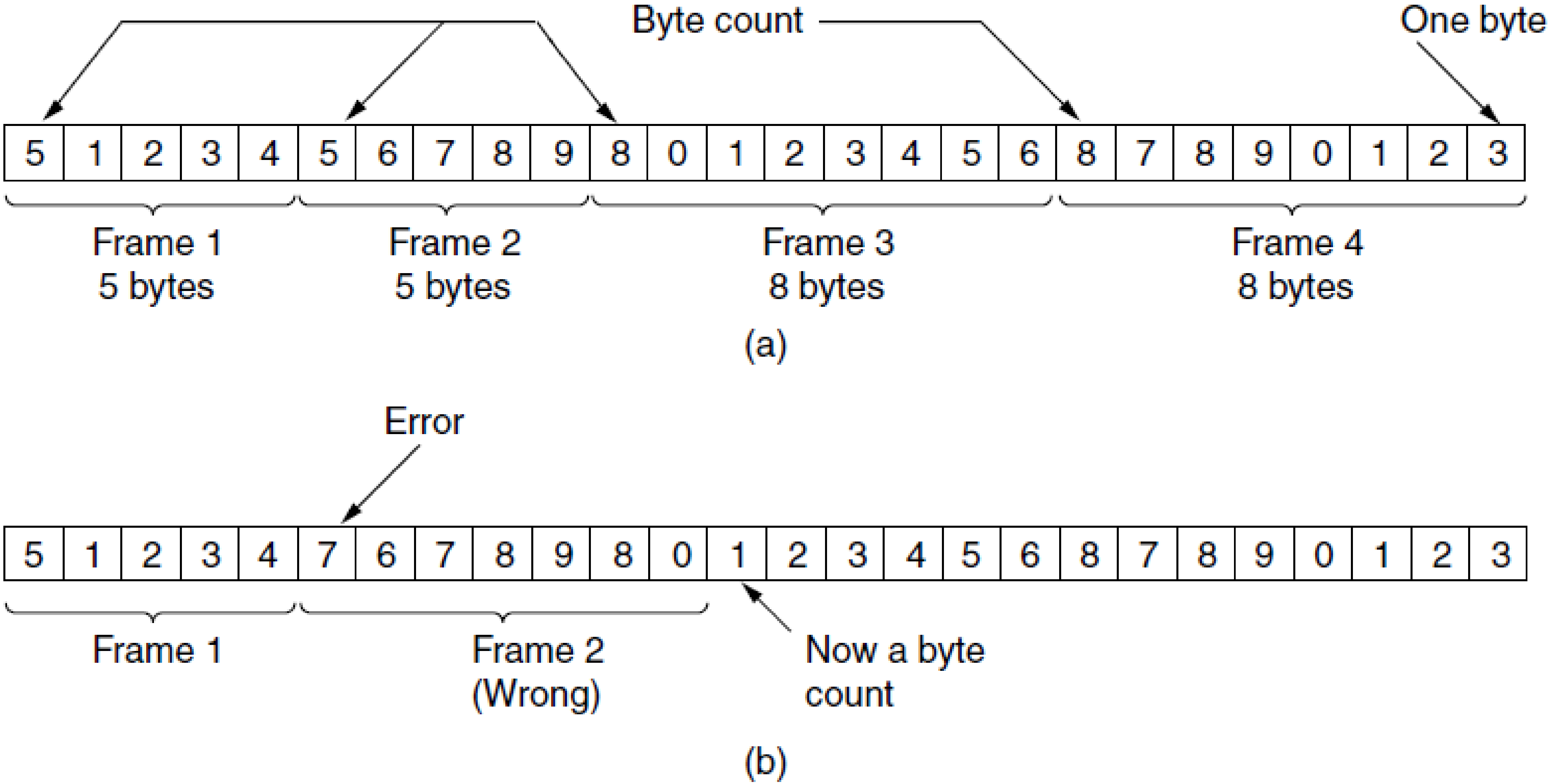


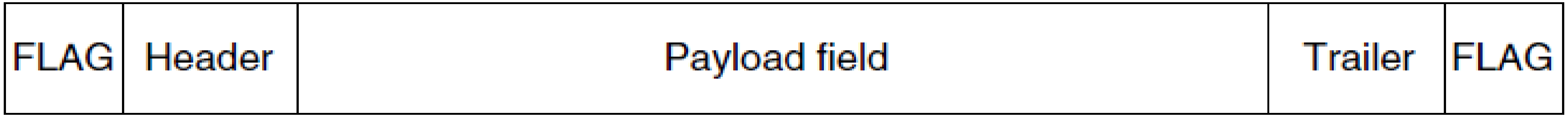
Figure: A byte stream. (a) Without errors. (b) With one error.

Framing Method: Flag bytes with byte stuffing

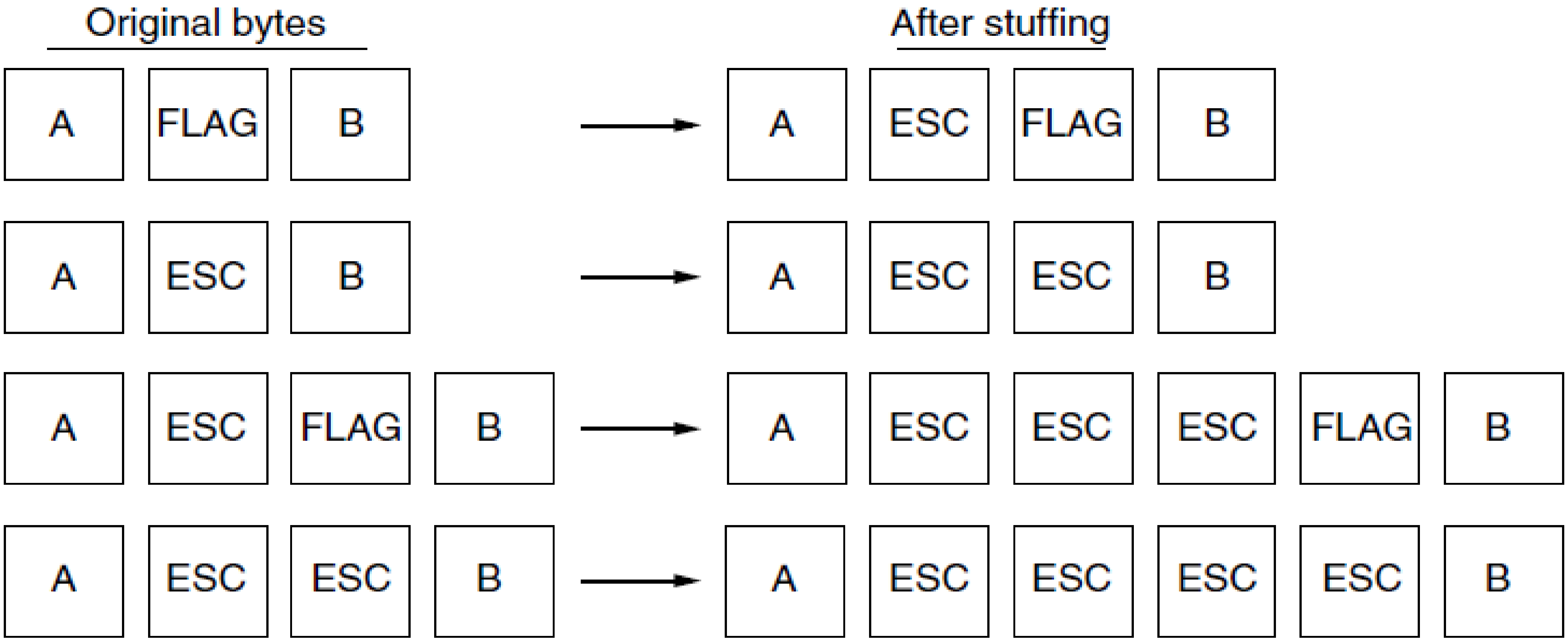
- A **flag byte**, is used as both the starting and ending delimiter. This byte is shown in Figure (a) as FLAG.
- Two consecutive flag bytes indicate the end of one frame and the start of the next.
- Thus, if the receiver ever loses synchronization it can just search for two flag bytes to find the end of the current frame and the start of the next frame.
- It may happen that the **flag byte occurs in the data**. This situation would interfere with the framing.
- One way to solve this problem is to have the sender's data link layer **insert a special escape byte (ESC)** just before each "accidental" flag byte in the data.

Framing Method: Flag bytes with byte stuffing

- Thus, a framing flag byte can be distinguished from one in the data by the absence or presence of an escape byte before it.
- The data link layer on the receiving end removes the escape bytes before giving the data to the network layer.
- This technique is called **byte stuffing**.
- ***What happens if the text contains one or more escape characters followed by a flag?***
- If the **escape character is part of the text**, an extra one is added to show that the second one is part of the text.
- The universal coding systems in use today, such as Unicode, have 16-bit and 32-bit characters that **conflict with 8-bit** characters. ***Solution: stuffs bit instead of byte.***


FLAG
Header
Payload field
Trailer
FLAG

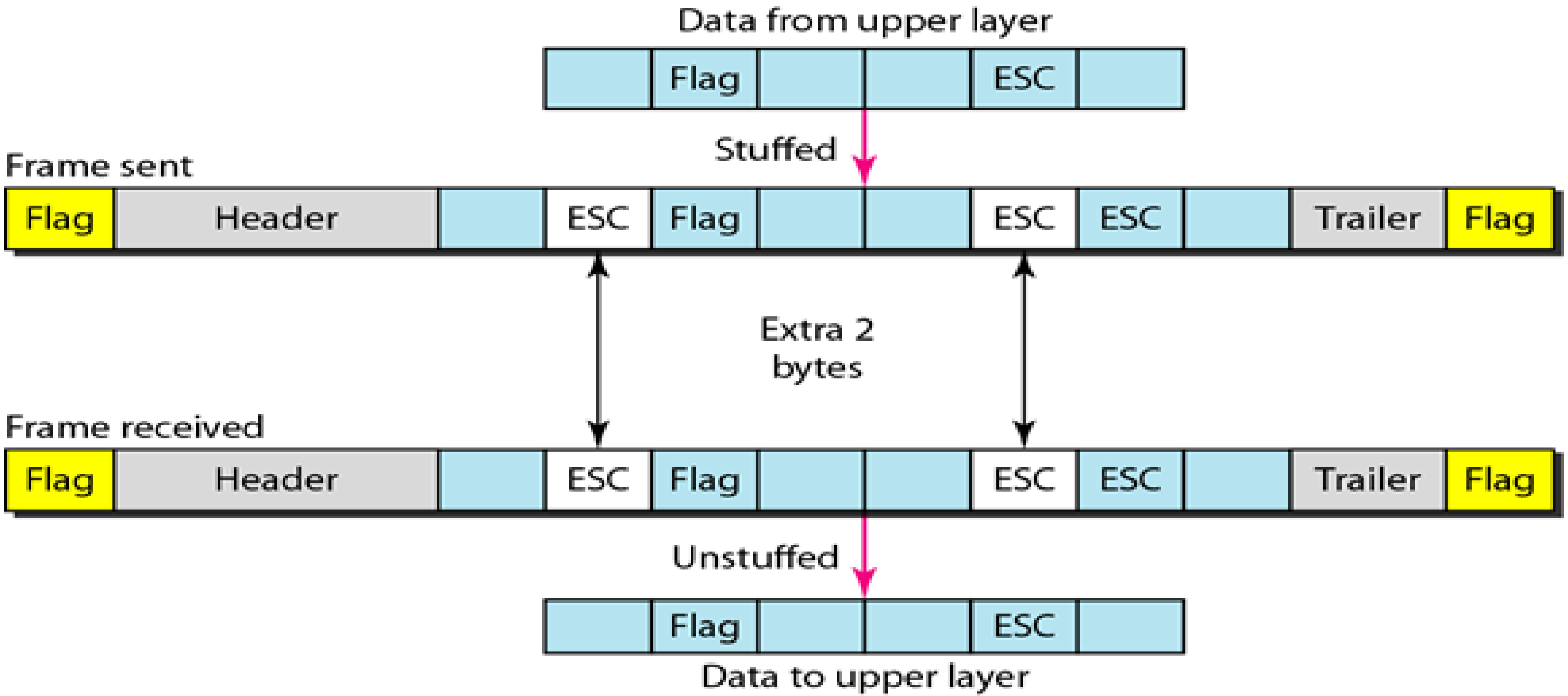
(a)


Original bytes
After stuffing
A FLAG B → A ESC FLAG B
A ESC B → A ESC ESC B
A ESC FLAG B → A ESC ESC ESC FLAG B
A ESC ESC B → A ESC ESC ESC ESC B

(b)

Byte stuffing and unstuffing

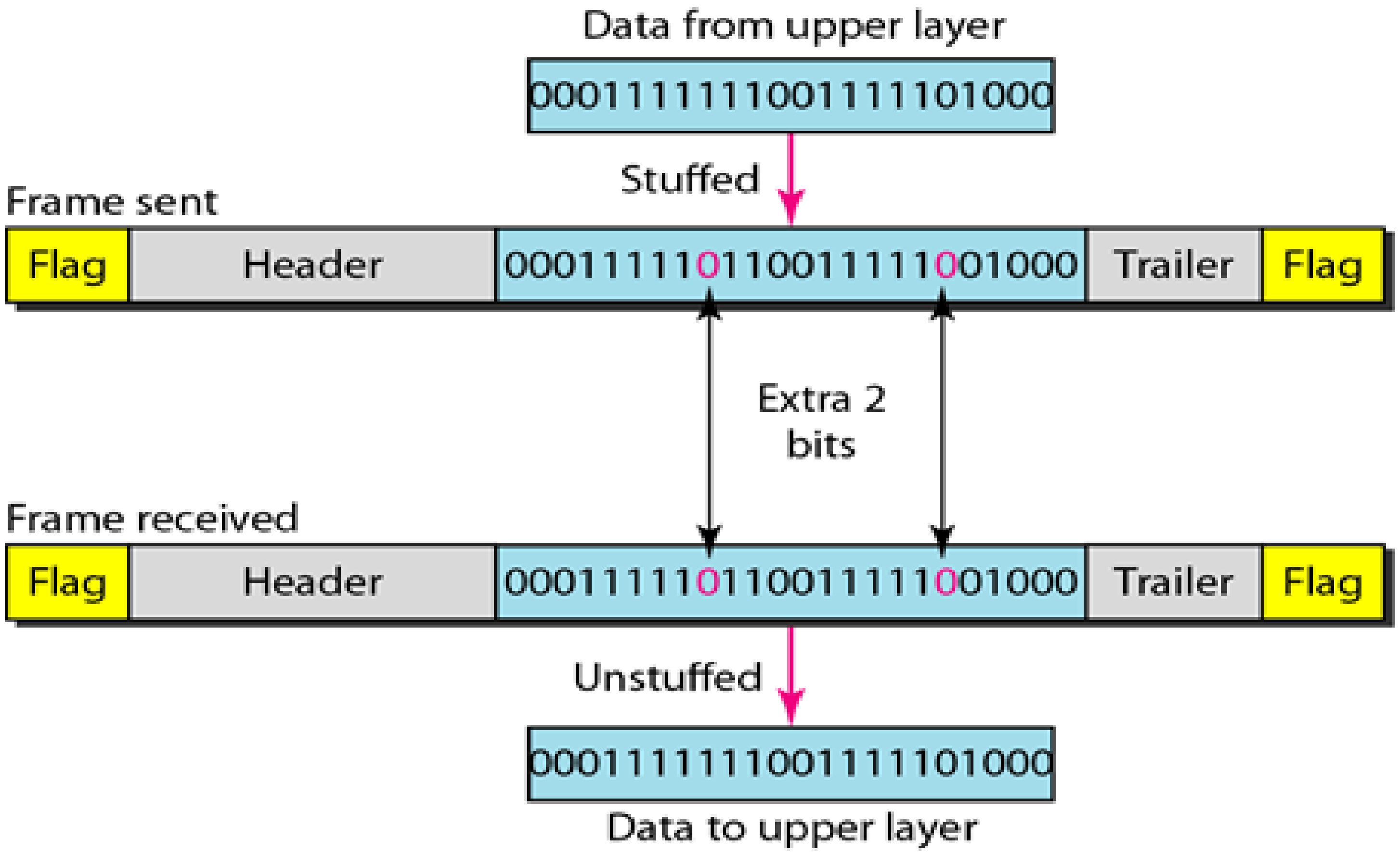
Byte stuffing is the process of adding 1 extra byte whenever there is a flag or escape character in the text.



Framing Method: Flag bytes with bit stuffing

- Framing can be done at the bit level, so frames can contain an arbitrary number of bits made up of units of any size.
- At the **start and end of each frame** is a flag byte consisting of the **special bit pattern 01111110 .**
- whenever the sender's data link layer **encounters five consecutive 1s** in the data, it automatically **stuffs a zero bit into the outgoing bit** stream. ***This technique is called bit stuffing.***
- When the receiver **sees five consecutive 1s** in the incoming data stream, **followed by a zero** bit, it automatically **de-stuffs the 0 bit.**
- The **boundary** between **two frames** can be determined by **locating the flag pattern.**

Bit stuffing and unstuffing



Data from upper layer

0001111111001111101000

Stuffed

Frame sent

Flag | Header | 000111110110011111001000 | Trailer | Flag

Extra 2 bits

Frame received

Flag | Header | 000111110110011111001000 | Trailer | Flag

Unstuffed

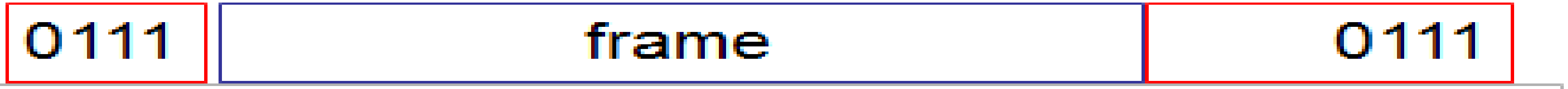
0001111111001111101000

Data to upper layer

Framing → Bit Stuffing

Use reserved bit patterns to indicate the start and end of a frame.

For instance, use the 4-bit sequence of 0111 to delimit consecutive frames. A frame consists of everything between two delimiters.


0111

frame

0111

Problem: What happens if the reserved delimiter happens to appear in the frame itself? If we don't remove it from the data, the receiver will think that the incoming frame is actually two smaller frames!

Solution: Use *bit stuffing*. Within the frame, after every occurrence of two consecutive 1's insert 0. E.g., append a zero bit after each pair of 1's in the data. This prevents 3 consecutive 1's from ever appearing in the frame.

Likewise, the receiver converts two consecutive 1's followed by a 0 into two 1's, but recognizes the 0111 sequence as the end of the frame.

Error Control

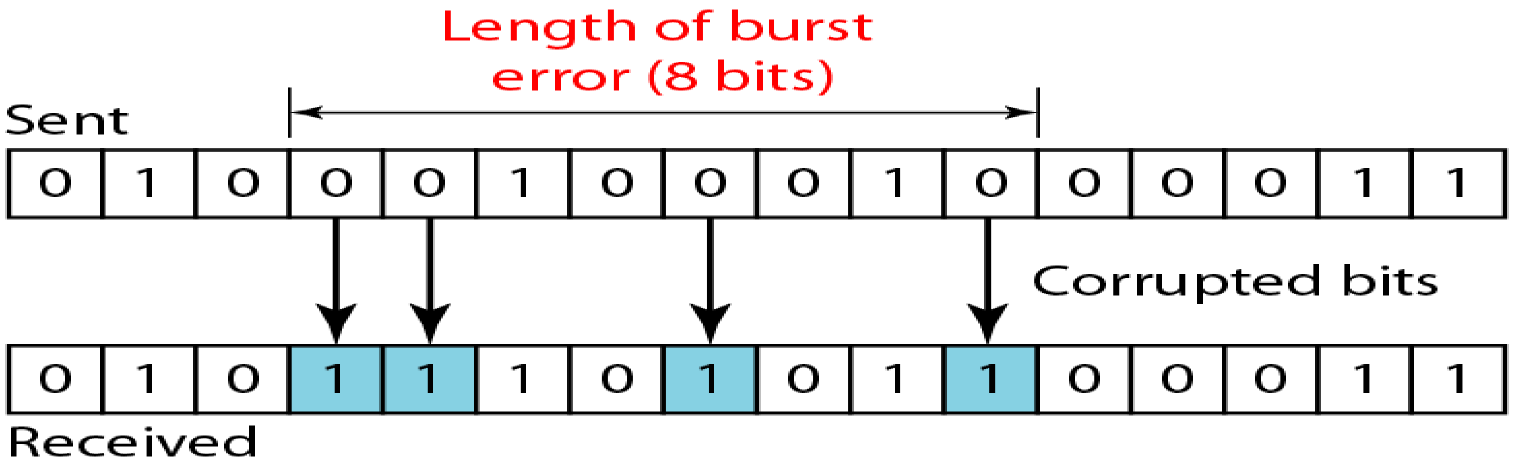
- We use bit and byte stuffing as a method for detecting and determining errors in the data that we send.
- We also have to deal with making sure that the frames make it to their destination and in the proper order.
- The receiver sends back a control frame acknowledging the received frame and the condition of the frame. A timeout can occur if the acknowledgement doesn’t arrive, resulting in the frame being resent.
- Resending the frame can also cause problems – what happens when the same frame is received twice?
- We can also sequentially number the frames to prevent this problem.
- Error control in the data link layer is based on automatic repeat request, which is the retransmission of data.
- Error control **includes both error detection and error correction.**

Error Detection and Corrections

- **Three different error-detecting codes**
  - *CRC (Cyclic Redundancy Check) ,Parity Check, CheckSum*
- **Four different error-correcting codes**
  - *Hamming codes, Binary convolutional codes, Reed-Solomon codes, Low-Density Parity Check codes*
- Error-correcting codes are **widely used on wireless links**, which are **notoriously noisy and error prone** when compared to optical fibers. Without error-correcting codes, it would be hard to get anything through.
- Error **detection and retransmission** is usually more efficient for dealing with the **occasional error or the error rate is much lower**.

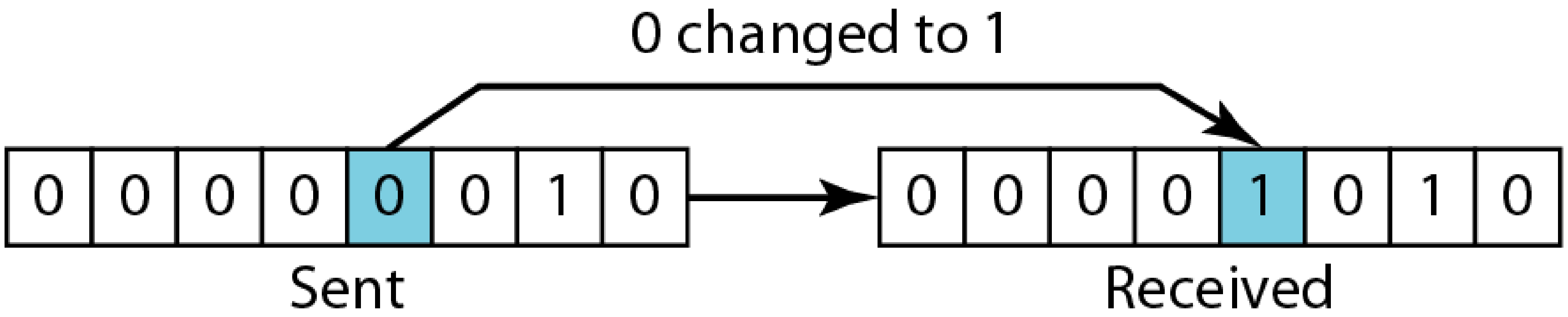
Types of Errors

- ***Single-Bit Error:*** It means that only 1 bit of a given data unit is changed from 1 to 0 or from 0 to 1.
- ***Burst Error:*** A burst error means that 2 or more bits in the data unit have changed.



Sent: 0 1 0 0 0 1 0 0 0 1 0 0 0 0 1 1

Received: 0 1 0 1 1 1 0 1 0 1 1 0 0 0 1 1

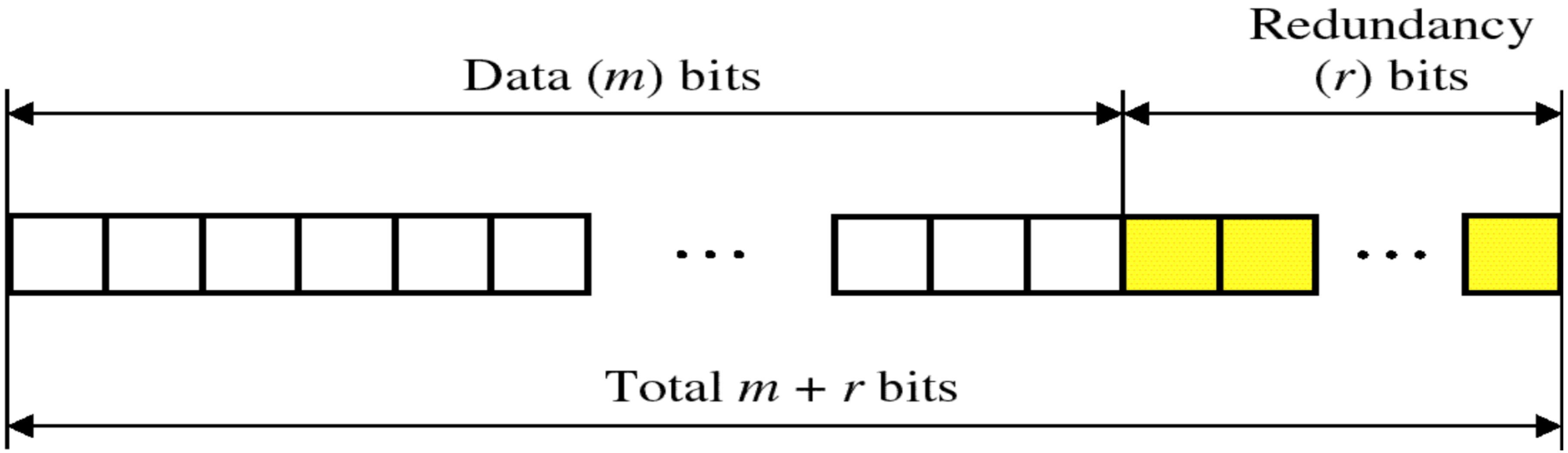


Sent: 0 0 0 0 0 0 1 0

Received: 0 0 0 0 1 0 1 0

Error Correction Codes

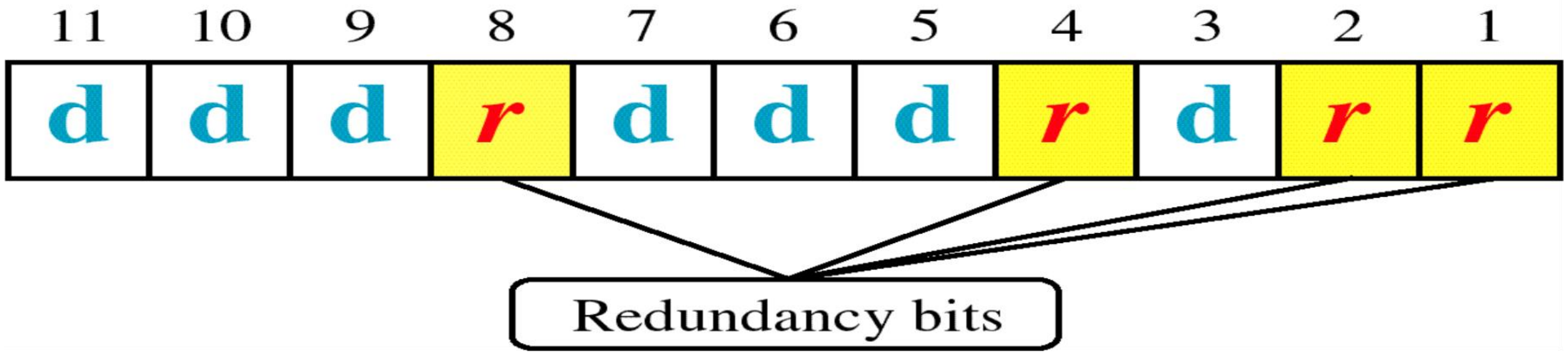
- All of error correction codes add redundancy to the information that is sent. A frame consists of m data (i.e., message) bits and r redundant (i.e. check) bits.



- The value of r must satisfy the following relation: $2^r \geq m+r+1$
- In a **block code**, the *r* check bits are computed solely as a function of the *m* data bits with which they are associated.
- An *n*-bit unit containing data and check bits is referred to as an *n*-bit **codeword**.

Error Corrections Codes: Hamming codes

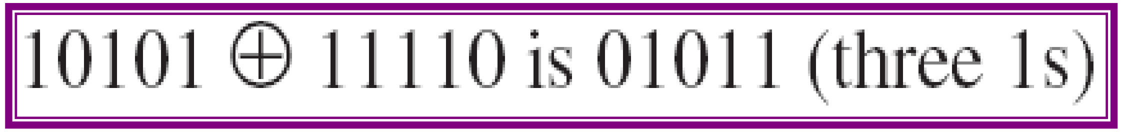
- Richard Hamming invented Hamming codes in 1950.
- Hamming Code: The bits that are powers of 2 (1, 2, 4, 8, 16, etc.) are check bits. The rest (3, 5, 6, 7, 9, etc.) are filled up with the *m* data bits.



- The number of bit positions in which two codewords differ is called the **Hamming distance** (Hamming, 1950).
- Its significance is that if two codewords are a Hamming distance *d* apart, it will require *d* single-bit errors to convert one into the other.

Error Corrections Codes: Hamming codes

- The Hamming distance is the XOR operation on the two words and count the number of 1s in the result.
- Given any two codewords that may be transmitted or received. The Hamming distance d(10101, 11110) is 3 because

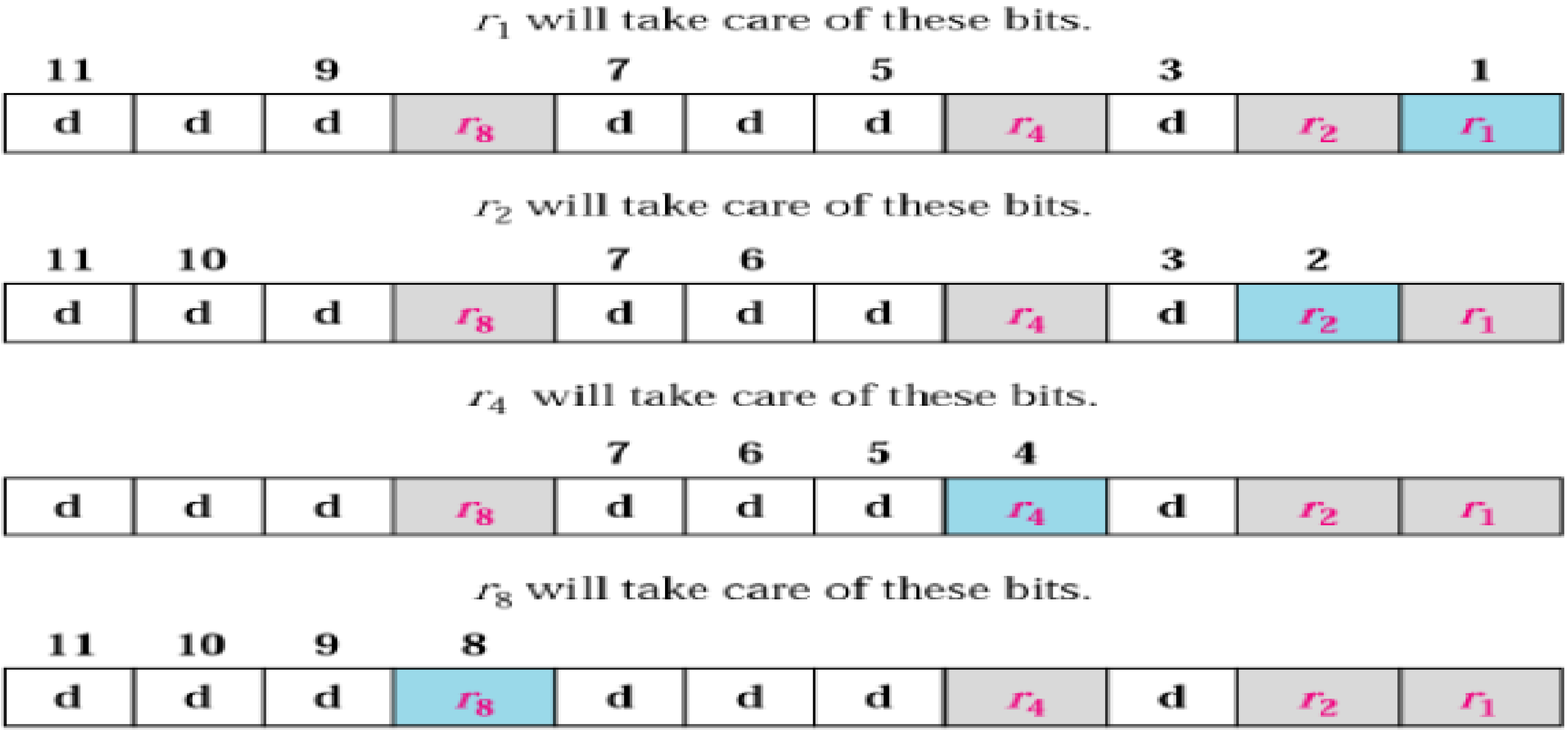


10101 ⊕ 11110 is 01011 (three 1s)

- For **detecting d errors** we need distance **d+1 code**.
- For **correcting d errors** we need distance **2d+1 code**.
- Valid codewords: 0000000000, 0000011111, 1111100000, and 1111111111
- The Hamming distance of the code=5 => we can correct 2d+1=5 => d=2 bit errors.

1. 0000000000---→0000000011 => the closest code is still 0000000000 !
2. 0000000000---→0000000111 => **the closest code is not correctly determined anymore !!**

Error Corrections Codes: Hamming codes



r_1 will take care of these bits.

11		9		7		5		3		1
d	d	d	r_8	d	d	d	r_4	d	r_2	r_1

r_2 will take care of these bits.

11	10			7	6			3	2	
d	d	d	r_8	d	d	d	r_4	d	r_2	r_1

r_4 will take care of these bits.

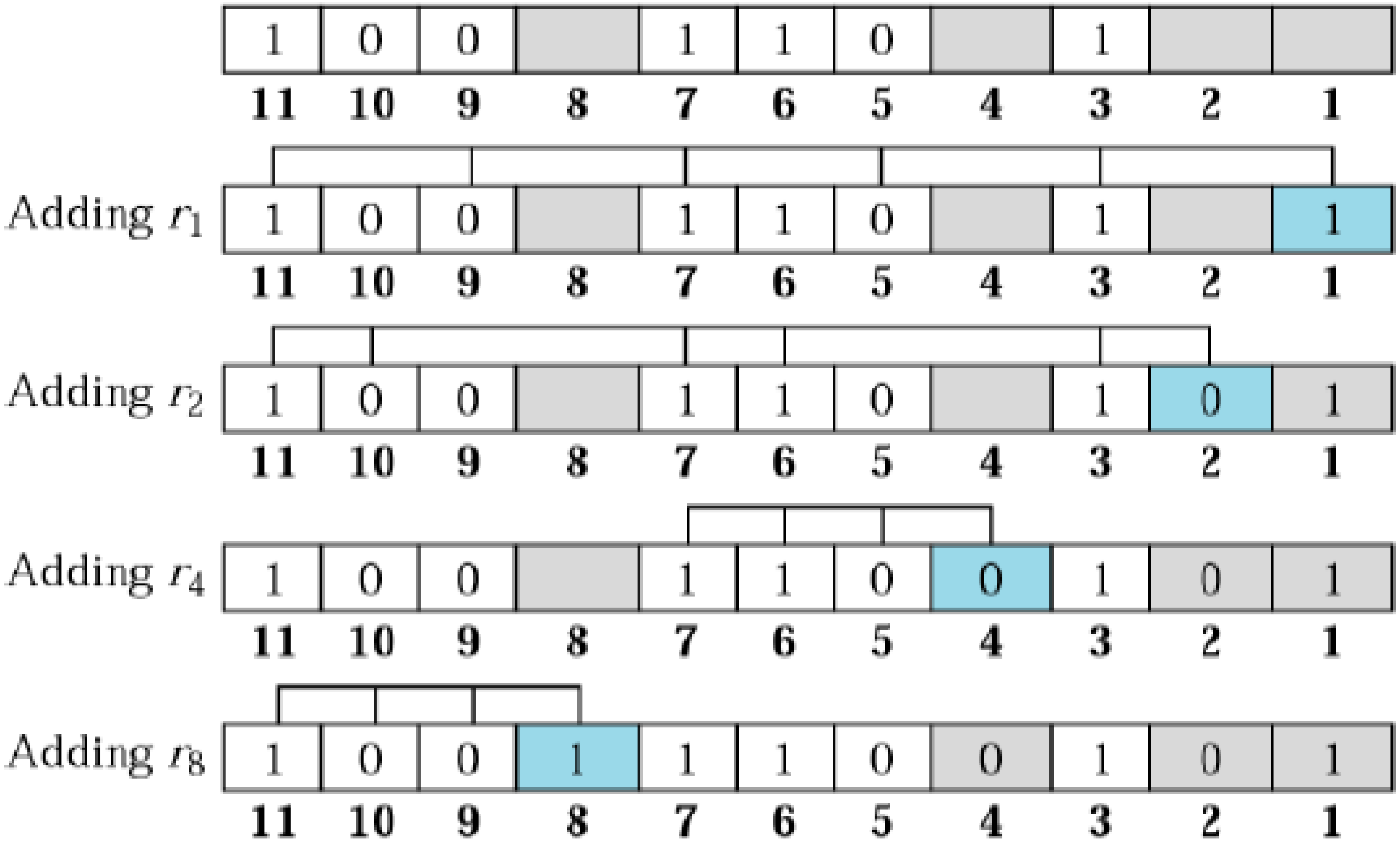
				7	6	5	4			
d	d	d	r_8	d	d	d	r_4	d	r_2	r_1

r_8 will take care of these bits.

11	10	9	8							
d	d	d	r_8	d	d	d	r_4	d	r_2	r_1

Calculate even parity for r1,r2 r4 and r8 using their respective bit position

Error Corrections Codes: Hamming codes

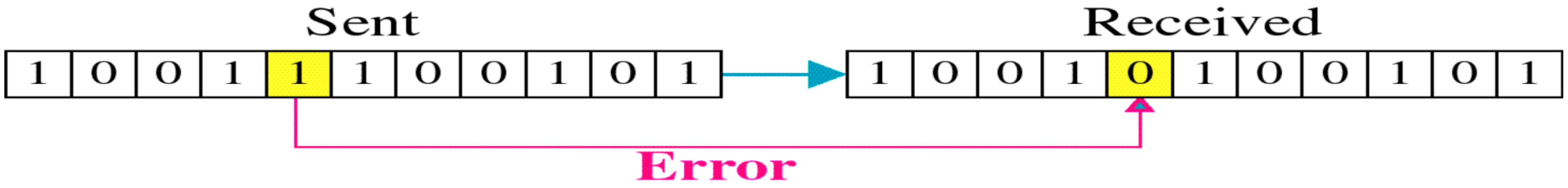


	11	10	9	8	7	6	5	4	3	2	1
	1	0	0		1	1	0		1		
Adding r_1	1	0	0		1	1	0		1		1
Adding r_2	1	0	0		1	1	0		1	0	1
Adding r_4	1	0	0		1	1	0	0	1	0	1
Adding r_8	1	0	0	1	1	1	0	0	1	0	1

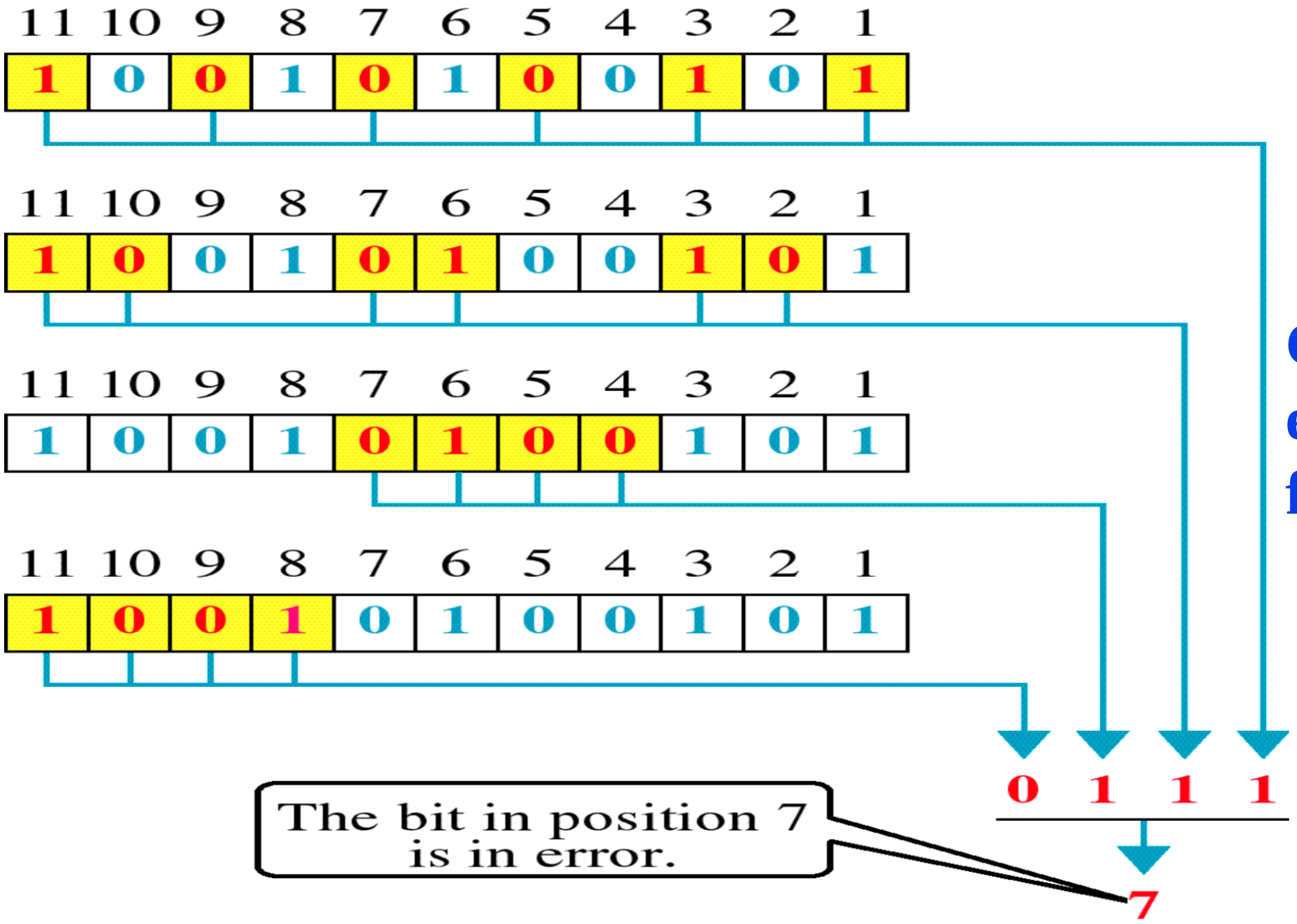
Data:
1 0 0 1 1 0 1

Code:
1 0 0 1 1 1 0 0 1 0 1

Single-bit error



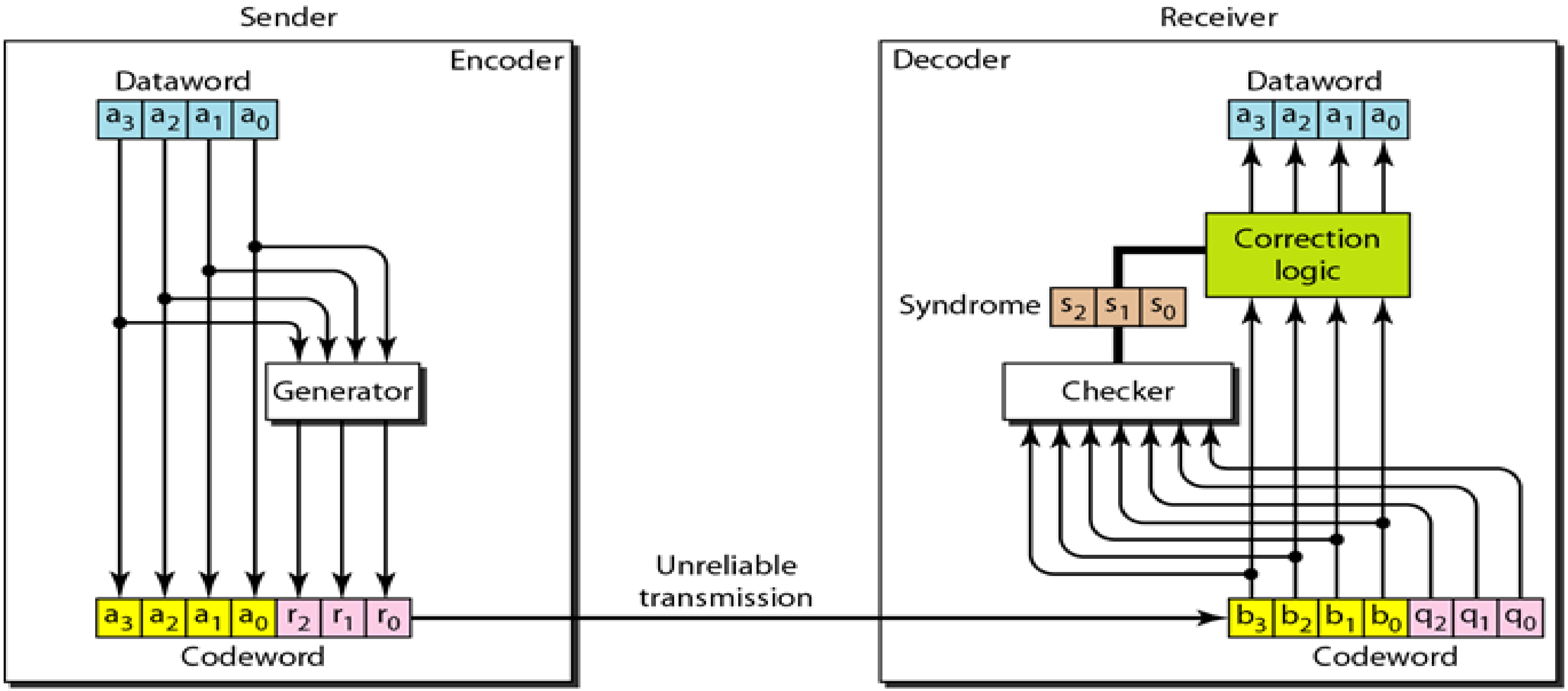
Error Detection



The bit in position 7 is in error.

Calculate even parity for $r_{1,}$ $r_{2,}r_{4,}$ r_8

The structure of the encoder and decoder for a Hamming code



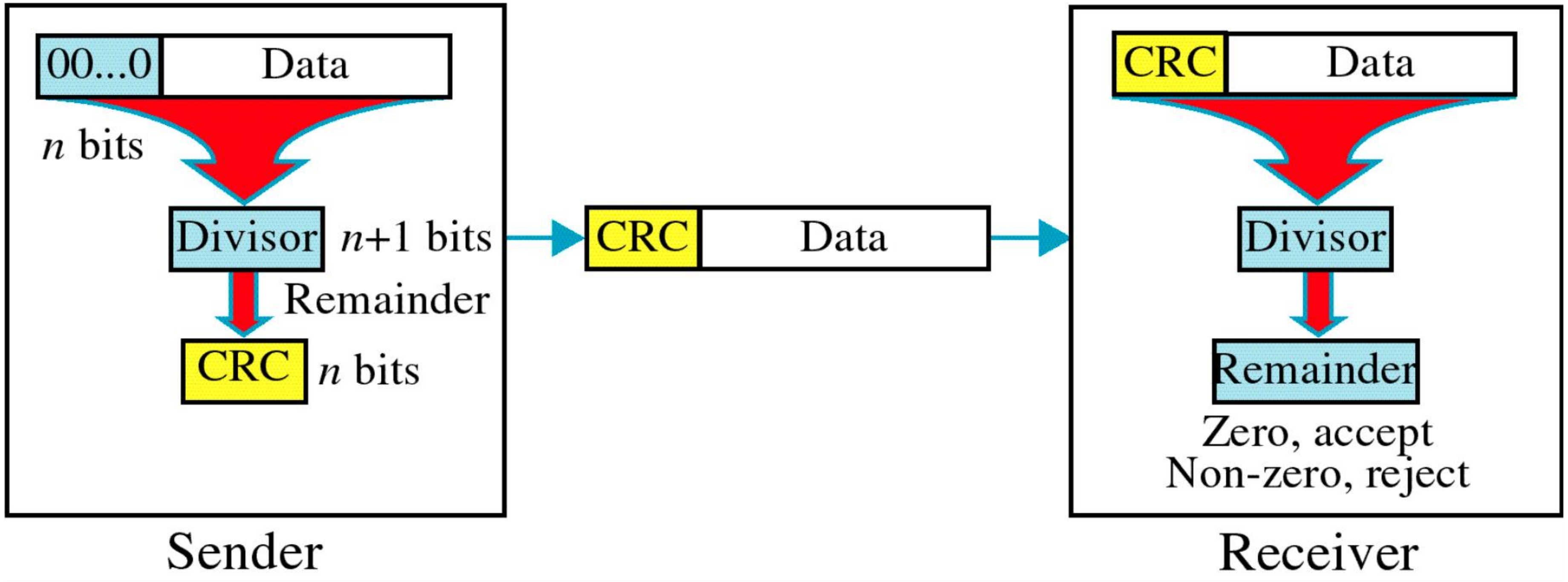
Error Detection Codes

Error Detection Process

- **Transmitter**
  - For a given frame, an error-detecting code (check bits) is calculated from data bits
  - Check bits are appended to data bits
- **Receiver**
  - Separates incoming frame into data bits and check bits
  - Calculates check bits from received data bits
  - Compares calculated check bits against received check bits
  - Detected error occurs if mismatch

Error Detection Codes: CRC (Cyclic Redundancy Check)

- Given a k-bit frame or message, the transmitter generates an n-bit sequence, known as a *frame check sequence (FCS)*, so that the resulting frame, consisting of (k+n) bits, is exactly divisible by some predetermined number.
- The receiver then divides the incoming frame by the same number and, if there is no remainder, assumes that there was no error.



Error Detection Codes: CRC (Cyclic Redundancy Check)

- CRC (Cyclic Redundancy Check), also known as a polynomial code.
- Polynomial codes are based upon treating bit strings as representations of polynomials with coefficients of 0 and 1 only.
- A *k*-bit frame is regarded as the coefficient list for a polynomial with *k* terms, ranging from x^{k-1} to x^{0}.
- Polynomial arithmetic is done modulo 2. It does not have carries for addition or borrows for subtraction. Both are identical to exclusive OR.
- A six-term polynomial with coefficients 1, 1, 0, 0, 0, and 1:

$$1x^5 + 1x^4 + 0x^3 + 0x^2 + 0x^1 + 1x^0.$$

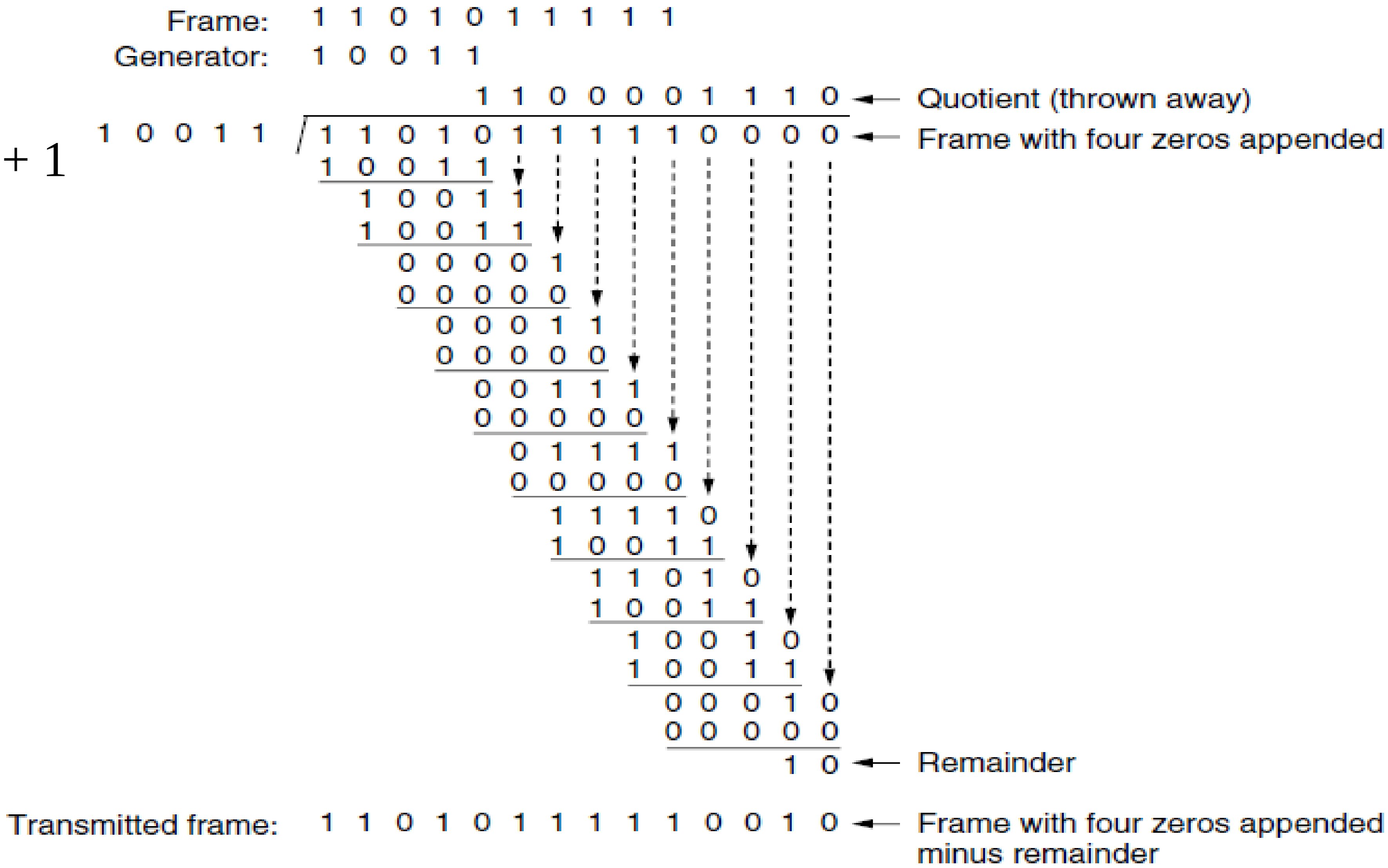
Error Detection Codes: CRC (Cyclic Redundancy Check)

- The data or polynomial **is appended** with **number of zeros equal to the degree of generator** (divisor).
- The polynomial **formed is divided** by the divisor.
- The **remainder of the division** will be the **value of CRC** that will replace the data plus extra zeros i.e., **remainder is added to the appended polynomial** .
- This value is **now transmitted** to the receiver as the transmitted frame.
- At the **receiver side**, the data string and the CRC value is **divided by** the same **value of divisor in the** sender part.
- Then the remainder determines either to accept or reject the received data bit string. If the **remainder is zero**, then the data will be **accepted** or **else it will be rejected.**

Error Detection Codes: CRC (Cyclic Redundancy Check)

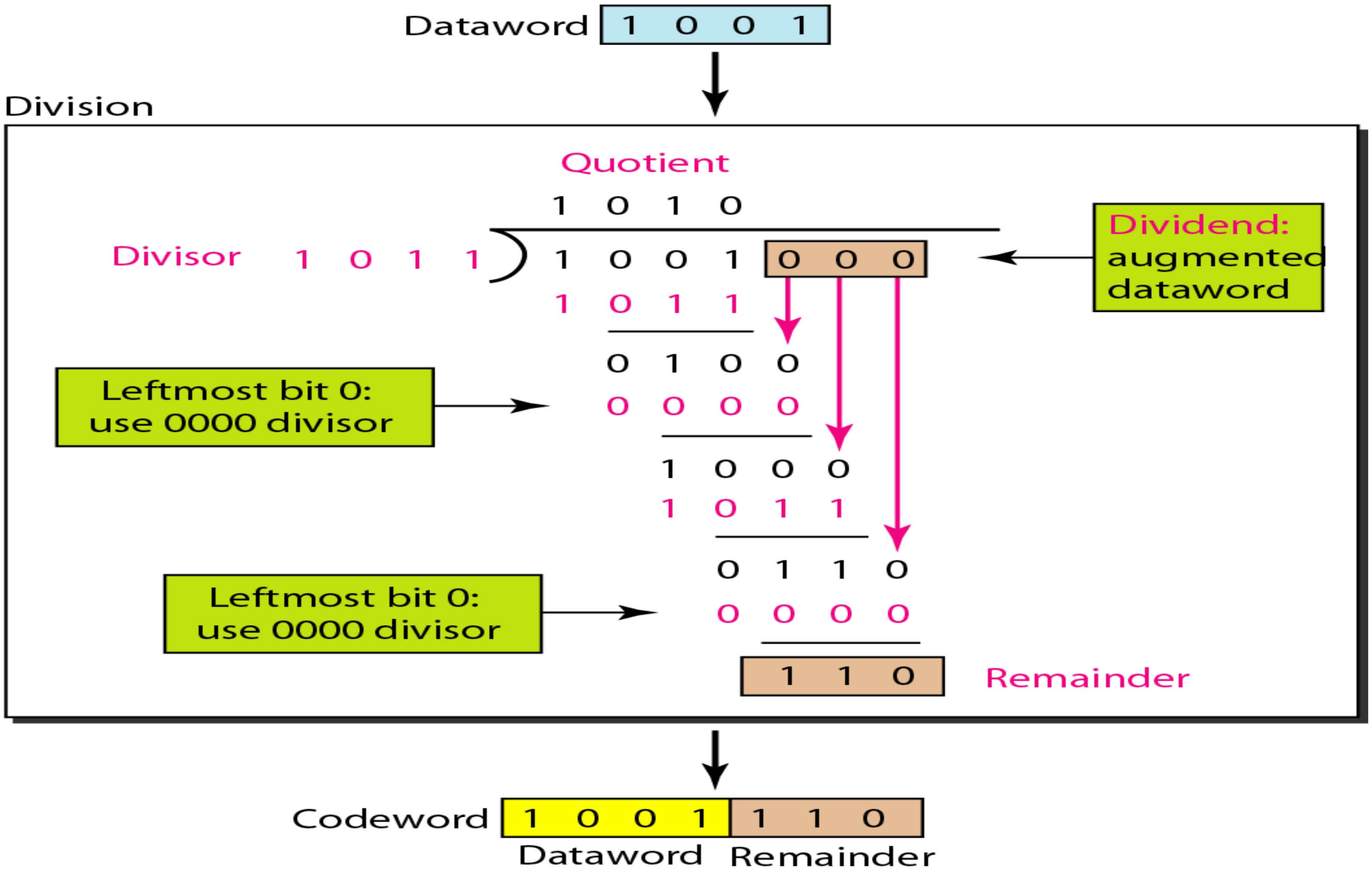
Generator = $x^4 + x^1 + 1$

=10011

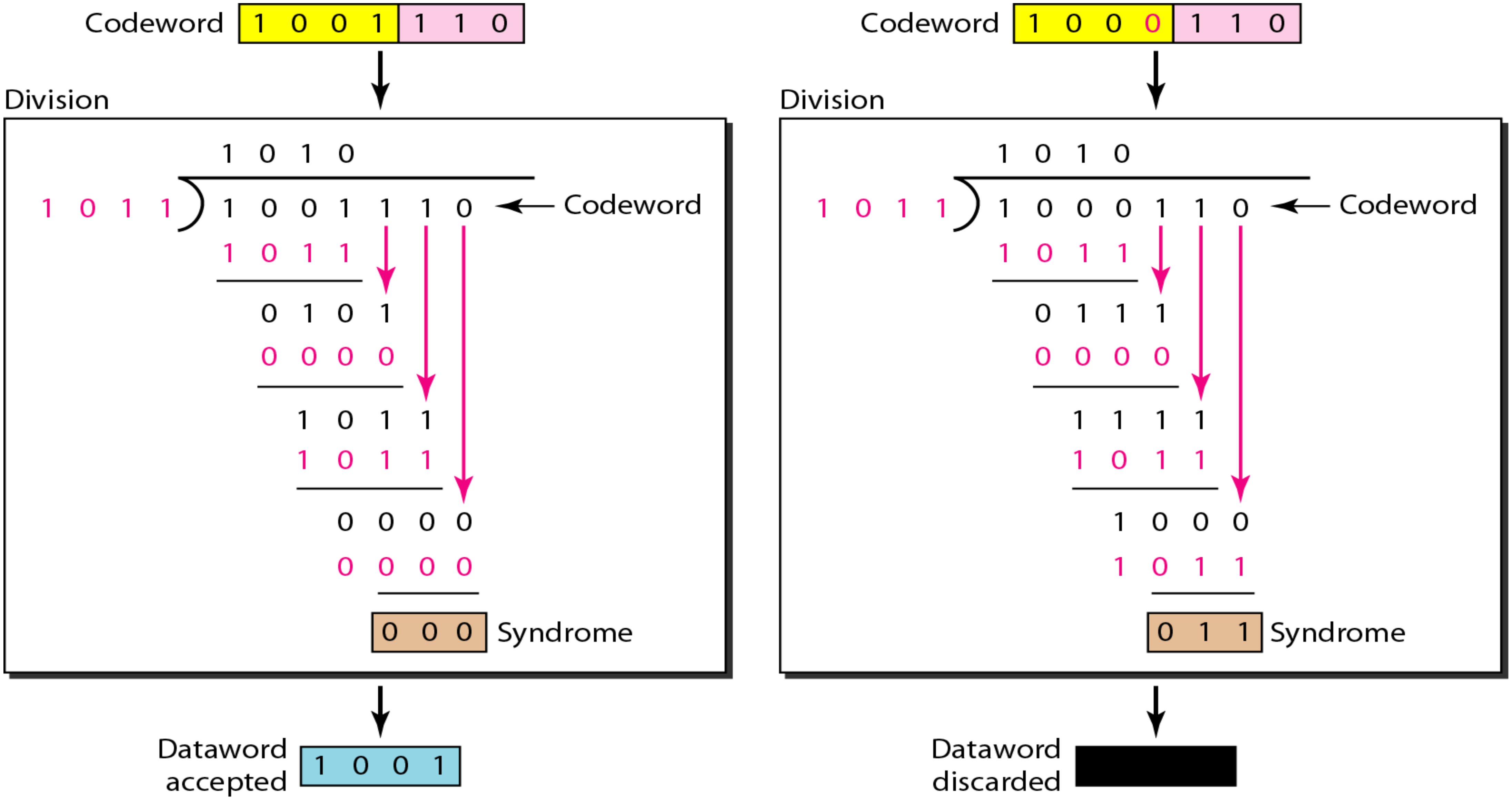


```
Frame:      1 1 0 1 0 1 1 1 1 1
Generator:  1 0 0 1 1

                      1 1 0 0 0 0 1 1 1 0   ← Quotient (thrown away)
  1 0 0 1 1 / 1 1 0 1 0 1 1 1 1 1 0 0 0 0   ← Frame with four zeros appended
              1 0 0 1 1
                1 0 0 1 1
                1 0 0 1 1
                  0 0 0 0 1
                  0 0 0 0 0
                    0 0 0 1 1
                    0 0 0 0 0
                      0 0 1 1 1
                      0 0 0 0 0
                        0 1 1 1 1
                        0 0 0 0 0
                          1 1 1 1 0
                          1 0 0 1 1
                            1 1 0 1 0
                            1 0 0 1 1
                              1 0 0 1 0
                              1 0 0 1 1
                                0 0 0 1 0
                                0 0 0 0 0
                                      1 0   ← Remainder

Transmitted frame:  1 1 0 1 0 1 1 1 1 1 0 0 1 0   ← Frame with four zeros appended
                                                    minus remainder
```

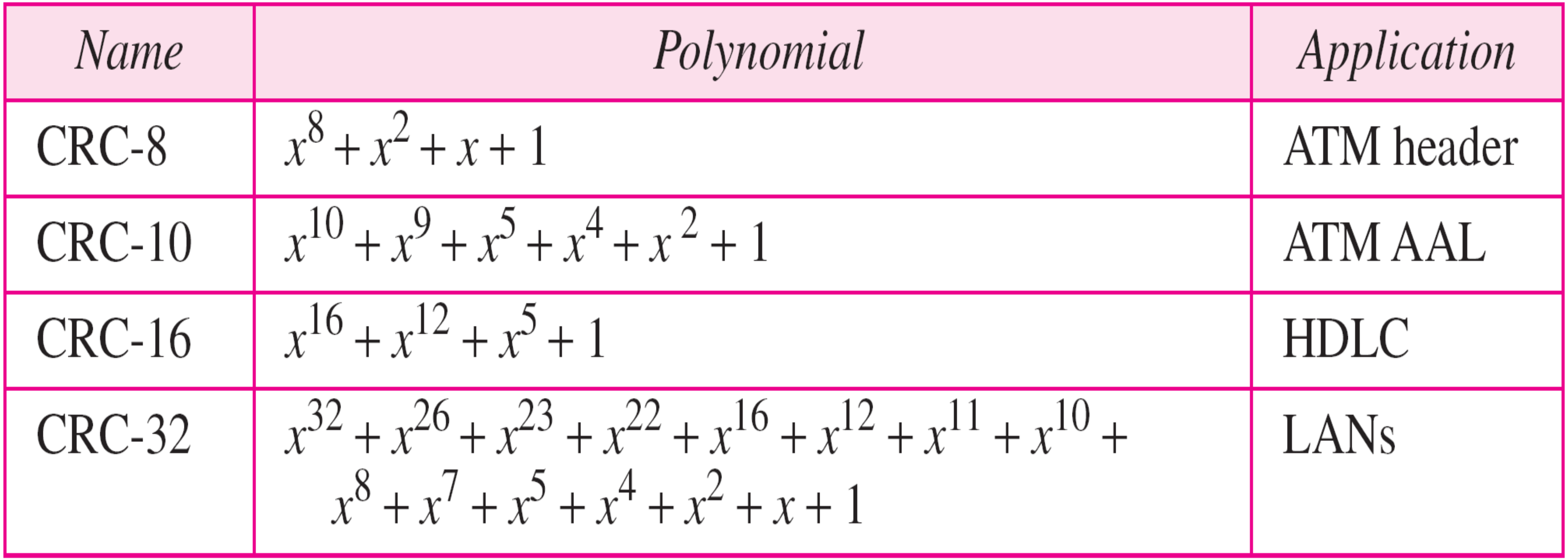
Error Detection Codes: CRC Encoding



Error Detection Codes: CRC Decoding

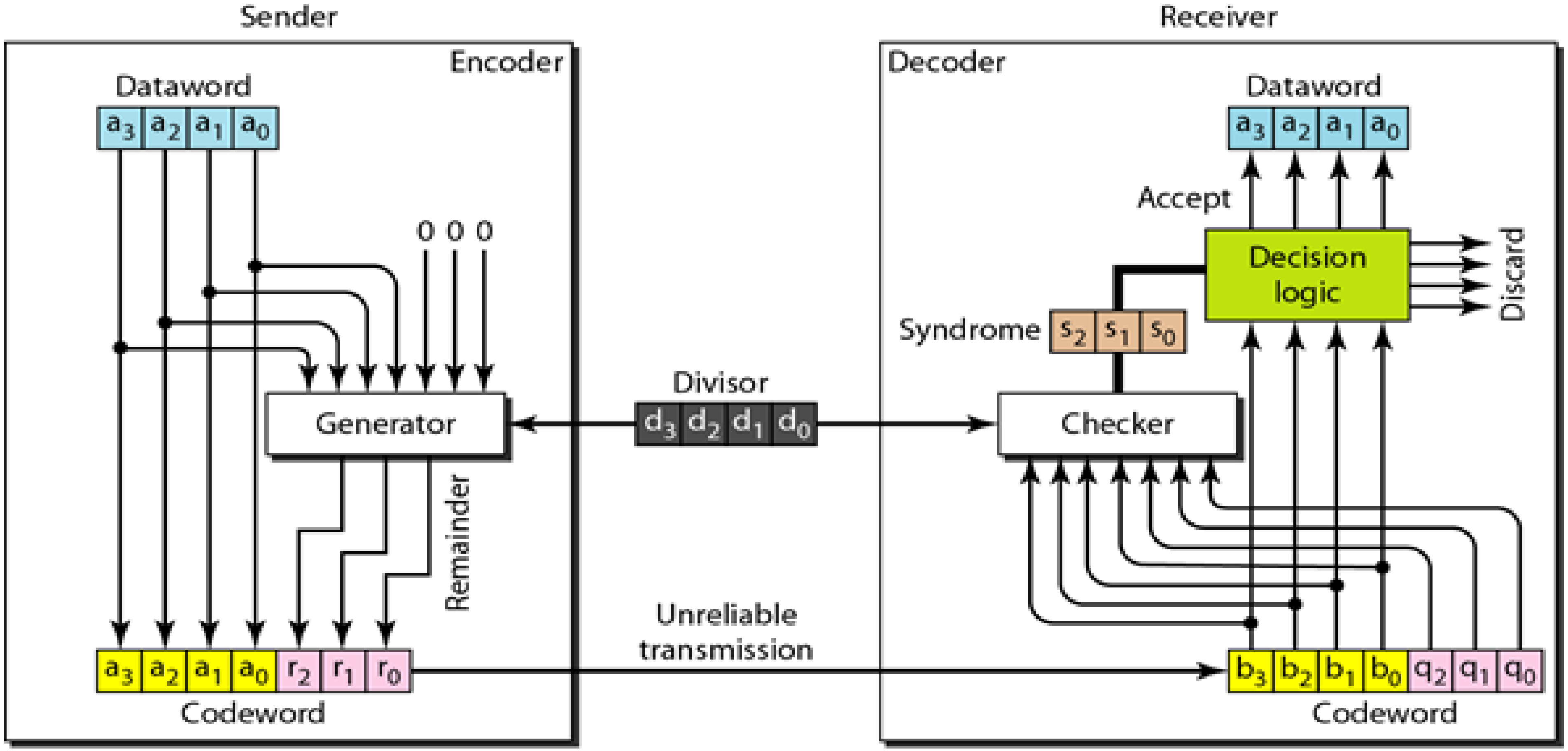


CRC Standards polynomials



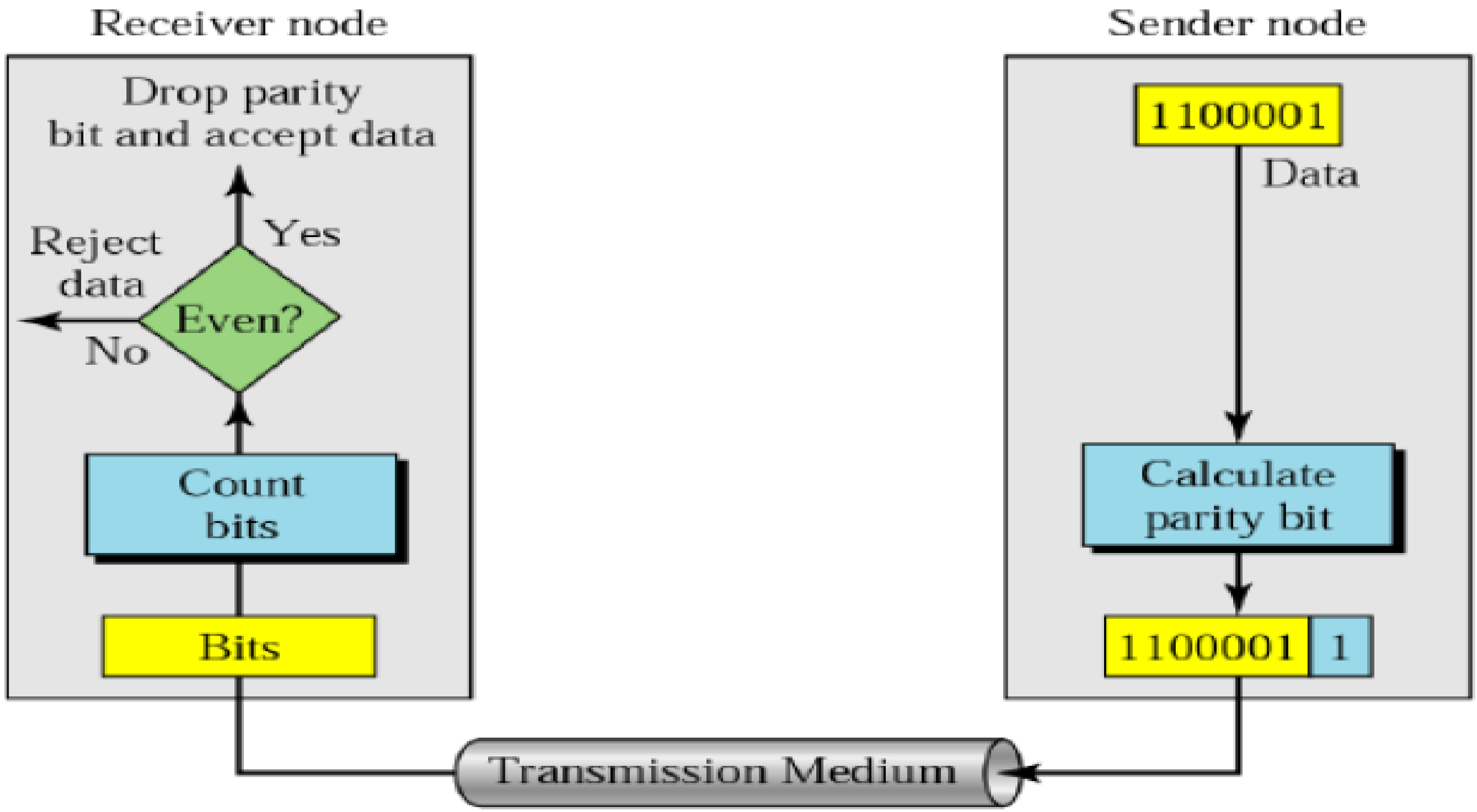
Name	*Polynomial*	*Application*
CRC-8	$x^8 + x^2 + x + 1$	ATM header
CRC-10	$x^{10} + x^9 + x^5 + x^4 + x^2 + 1$	ATM AAL
CRC-16	$x^{16} + x^{12} + x^5 + 1$	HDLC
CRC-32	$x^{32} + x^{26} + x^{23} + x^{22} + x^{16} + x^{12} + x^{11} + x^{10} +$ $x^8 + x^7 + x^5 + x^4 + x^2 + x + 1$	LANs

CRC encoder and decoder



Error Detection Codes: Parity Check

- In parity check, a parity bit is added to every data unit so that the total number of 1s is even (or odd for odd-parity).
- **Even Parity:**



Error Detection Codes: Parity Check

Examples:

- Now suppose the word is received by the receiver without being corrupted in transmission. ***11101110 11011110 11100100 11011000 11001001***
- The receiver counts the 1s in each character and comes up with even numbers (6, 6, 4, 4, 4). The data are accepted.
- Now suppose the word is corrupted during transmission.

11111110 11011110 11101100 11011000 11001001

- The receiver counts the 1s in each character and comes up with even and odd numbers (7, 6, 5, 4, 4). The receiver knows that the data are corrupted, discards them, and asks for retransmission

Error Detection Codes: Parity Check

- Interleaving is a general technique to convert a code that detects (or corrects) isolated errors into a code that detects (or corrects) burst errors
- Interleaving, we will compute a parity bit for each of the *n* columns and send all the data bits as *k* rows, sending the rows from top to bottom and the bits in each row from left to right in the usual manner.
- At the last row, we send the *n* parity bits.

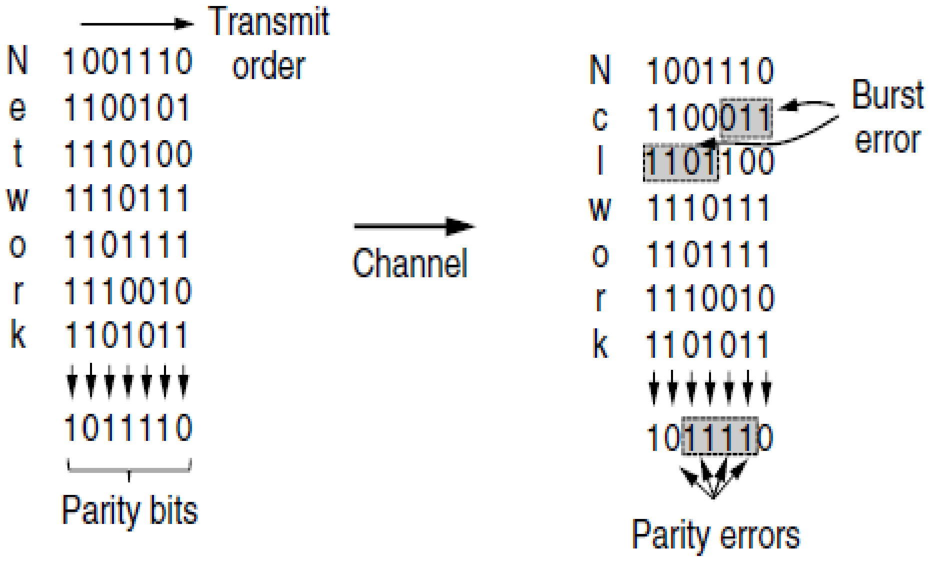
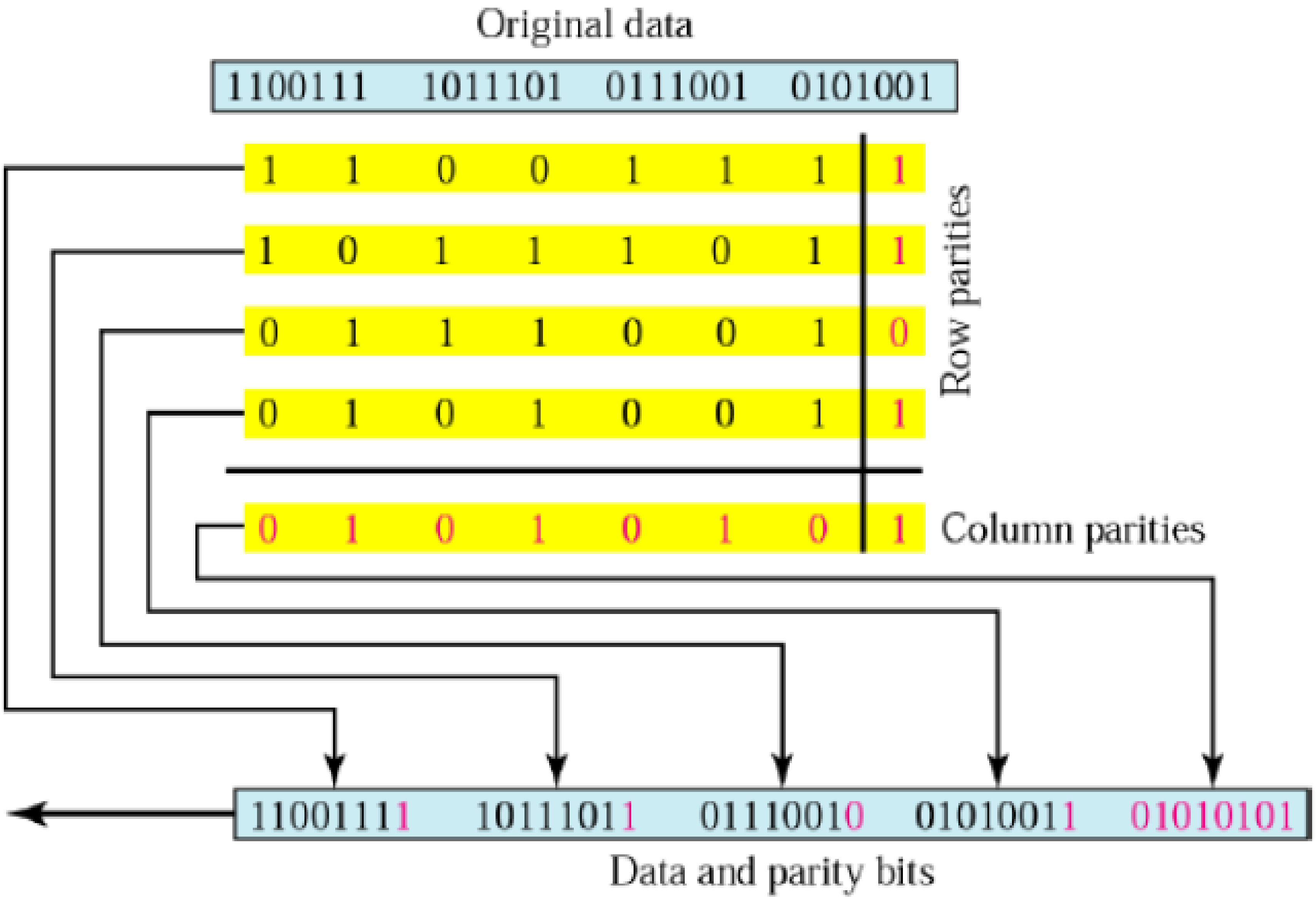


Figure 3-8. Interleaving of parity bits to detect a burst error.

Two Dimensional Parity Check



Original data

1100111 1011101 0111001 0101001

1	1	0	0	1	1	1	1
1	0	1	1	1	0	1	1
0	1	1	1	0	0	1	0
0	1	0	1	0	0	1	1
0	1	0	1	0	1	0	1

Row parities

Column parities

11001111 10111011 01110010 01010011 01010101

Data and parity bits

Error Detection Codes: Checksum

- “Checksum” is often used to mean a group of check bits associated with a message, regardless of how are calculated.
- A group of parity bits is one example of a checksum.
- The checksum is usually placed at the end of the message.
- Errors may be detected by summing the entire received codeword, both data bits and checksum.
- If the result comes out to be zero, no error has been detected.
- One example of a checksum is the 16-bit Internet checksum used on all Internet packets as part of the IP protocol.

Error Detection Codes: Internet Checksum

Sender site:

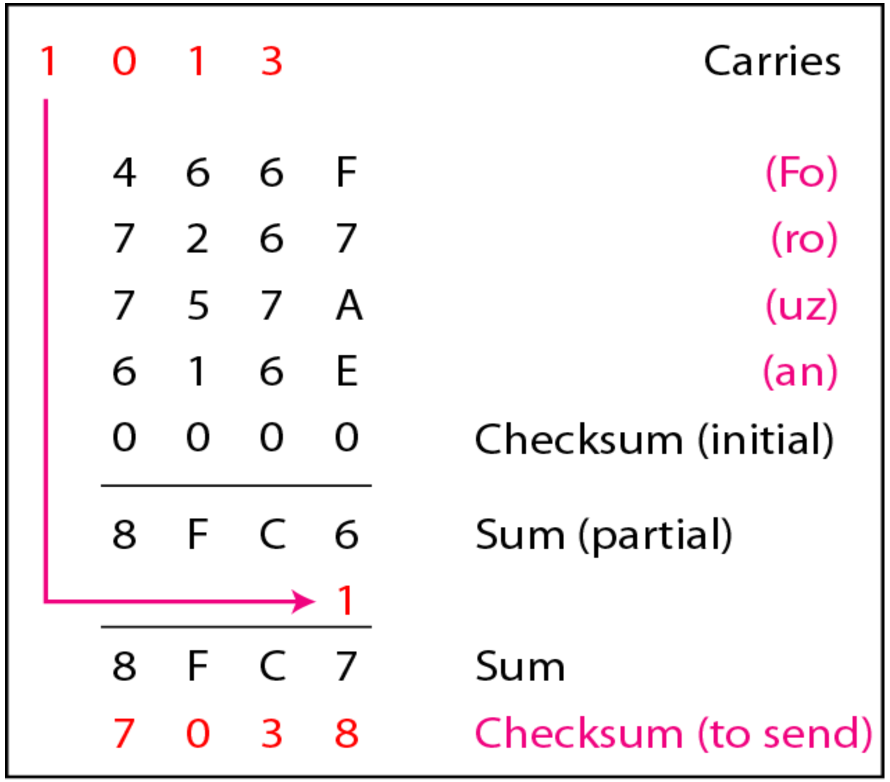
1. The message is divided into 16 bit words.
2. The value of the checksum word is set to 0.
3. All words including the checksum are added using one’s complement addition.
4. The sum is complemented and becomes the checksum.
5. The checksum is sent with the data.

Receiver site:

1. The message (including checksum) is divided into 16-bit words.
2. All words are added using one’s complement addition.
3. The sum is complemented and becomes the new checksum.
4. If the value of checksum is 0, the message is accepted; otherwise, it is rejected.

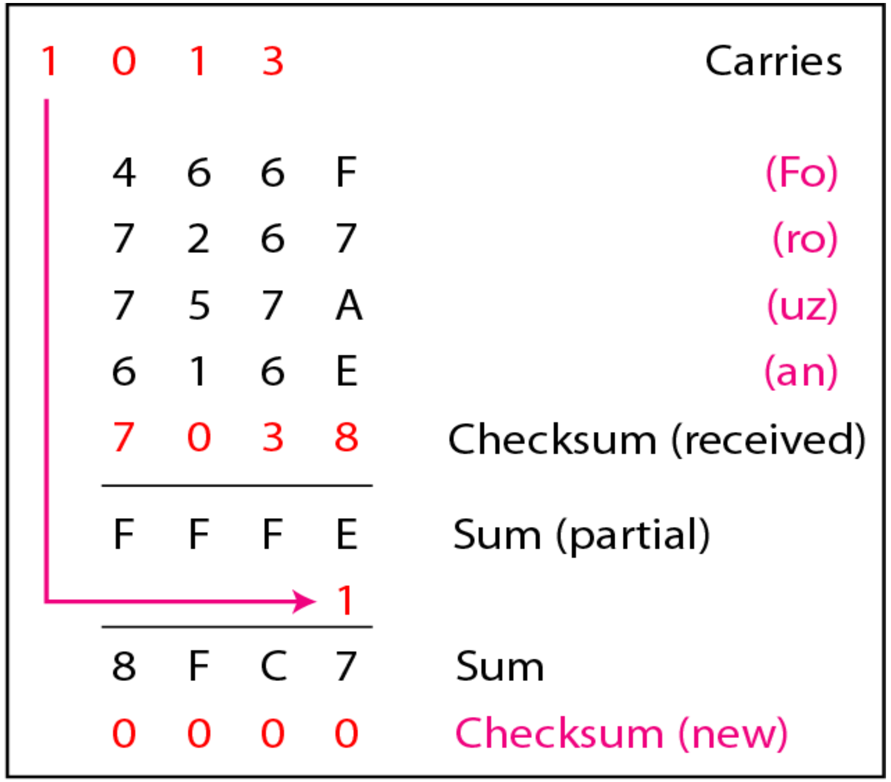
Error Detection Codes: Checksum Example

Let us calculate the checksum for a text of 8 characters (“Forouzan”). The text needs to be divided into 2-byte (16-bit) words. We use ASCII to change each byte to a 2-digit hexadecimal number. For example, F is represented as Ox46.



1	0	1	3	Carries
4	6	6	F	(Fo)
7	2	6	7	(ro)
7	5	7	A	(uz)
6	1	6	E	(an)
0	0	0	0	Checksum (initial)
8	F	C	6	Sum (partial)
			1	
8	F	C	7	Sum
7	0	3	8	Checksum (to send)

a. Checksum at the sender site

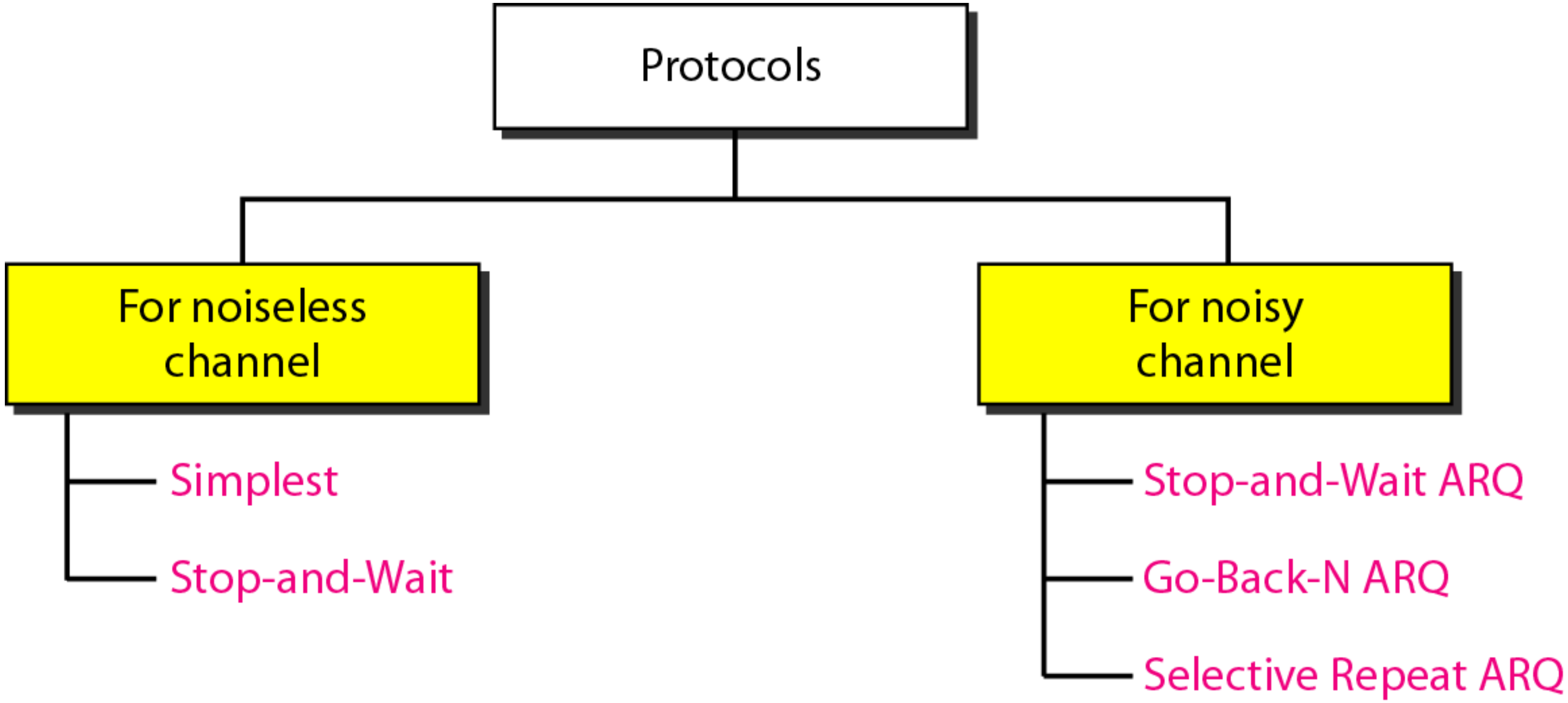


1	0	1	3	Carries
4	6	6	F	(Fo)
7	2	6	7	(ro)
7	5	7	A	(uz)
6	1	6	E	(an)
7	0	3	8	Checksum (received)
F	F	F	E	Sum (partial)
			1	
8	F	C	7	Sum
0	0	0	0	Checksum (new)

a. Checksum at the receiver site

Flow Control

- We must deal with the issue where the sender is sending data at a higher rate than the receiver can receive the data.
- Flow Control is a technique for speed-matching of transmitter and receiver.
- Flow control ensures that a transmitting station does not overflow a receiving station with data.



Flow Control

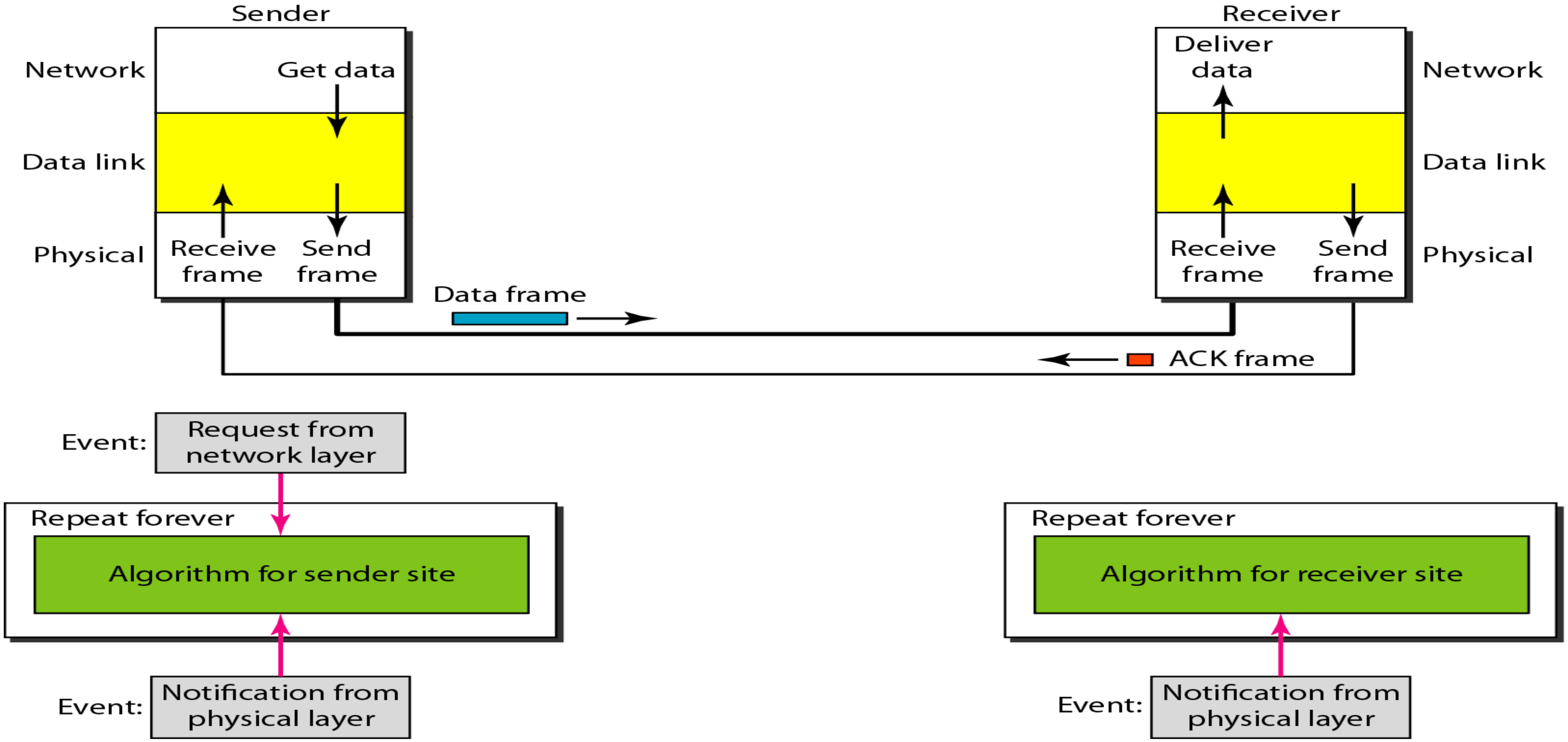
- ***Utopia Simplex Protocol***
  - Data are transmitted in one direction only. Both the transmitting and receiving network layers are always ready.
  - Processing time can be ignored. Infinite buffer space is available.
  - And best of all, the communication channel between the data link layers never damages or loses frames.
  - This thoroughly unrealistic protocol, which we will nickname “Utopia,” is simply to show the basic structure on which we will build.

Flow Control: Stop *and Wait*

- If data frames arrive at the receiver site faster than they can be processed, the frames **must be stored until their use.**
- Normally, the receiver **does not have enough storage space**, especially if it is receiving data from many sources.
- This may result in either the discarding of frames or denial of service.
- To prevent this, we somehow need to tell the sender to **slow down**.
- The protocol is called the **Stop-and-Wait Protocol** because the sender sends one frame, stops until it receives confirmation from the receiver and then sends the next frame.

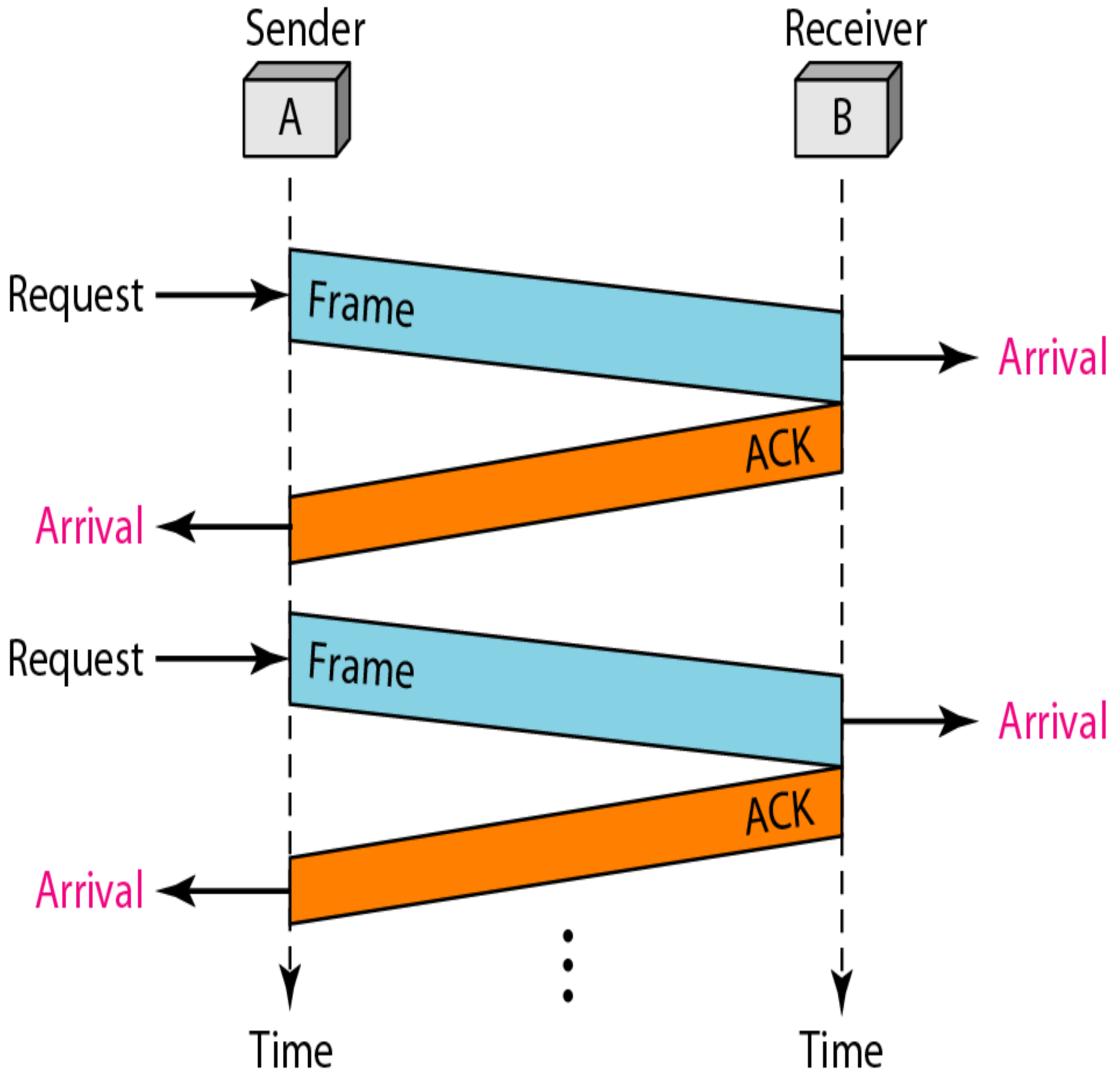
Flow Control: *Stop and Wait*

- The traffic is on the forward channel (from sender to receiver) and the reverse channel. At any time, there is either one data frame on the forward channel or one ACK frame on the reverse channel. We therefore need **a half-duplex link**.



Flow Control: *Stop and Wait*

- The sender sends one frame and waits for feedback from the receiver.
- When the ACK arrives, the sender sends the next frame.



Flow Control: Stop *and Wait* Automatic Repeat reQuest (ARQ)

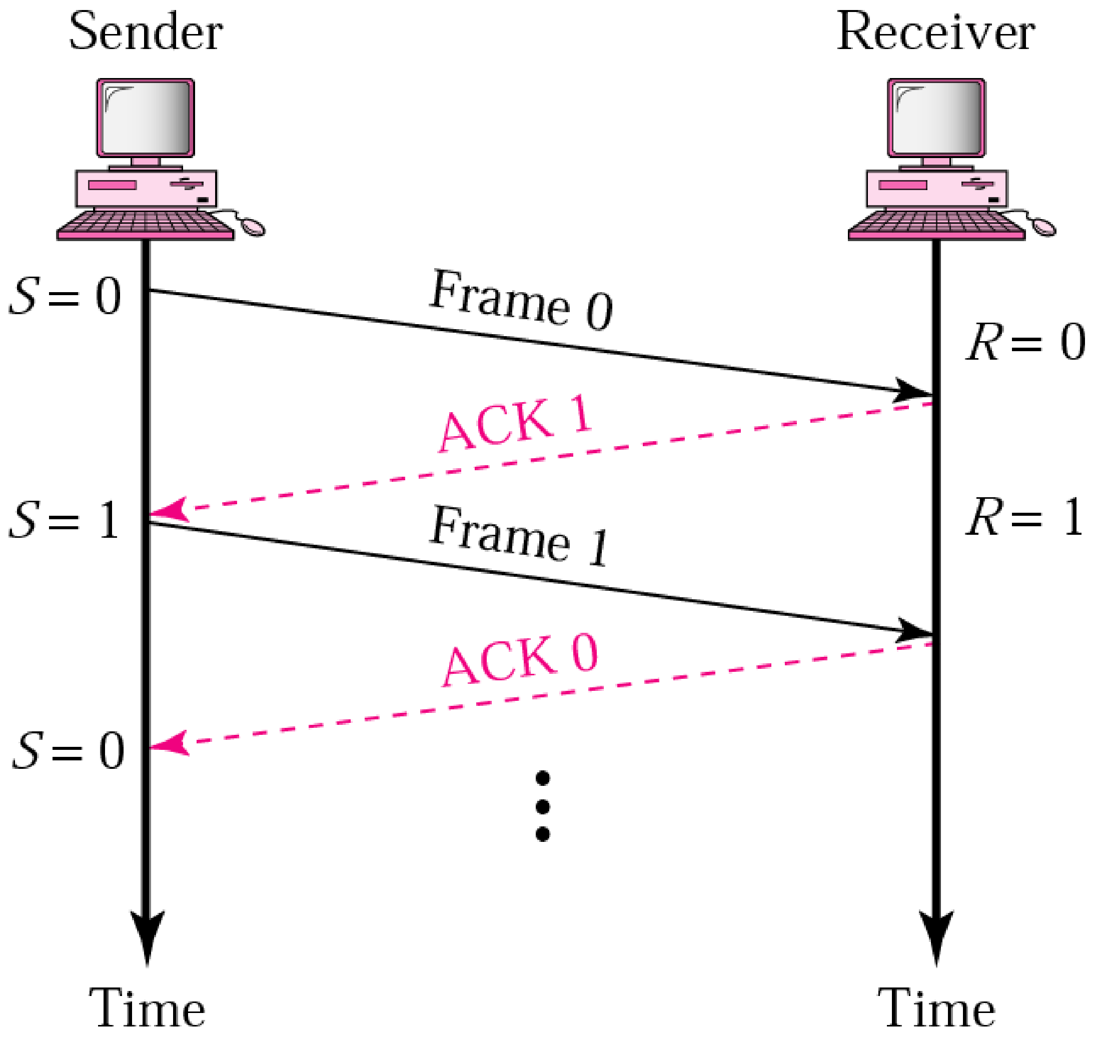
- It adds a simple error control mechanism to the Stop-and-Wait Protocol.
- Error control in the data link layer is based on automatic repeat request, which is the retransmission of data.
- **Stop-and-Wait ARQ has the following features***:*
  - The sending device keeps a copy of the sent frame transmitted until it receives an acknowledgment(ACK).
  - The sender starts a timer when it sends a frame. If an ACK is not received within an allocated time period, the sender resends it.
  - Both frames and acknowledgment (ACK) are numbered alternately 0 and 1 (two sequence number only).
  - This numbering allows for identification of frames in case of duplicate transmission.

Flow Control: Stop *and Wait* Automatic Repeat reQuest (ARQ)

- **Stop-and-Wait ARQ has the following features contd..**
  - The acknowledgment number defines the number of next expected frame. (frame 0 received ACK 1 is sent)
  - A damage or lost frame treated by the same manner by the receiver.
  - If the receiver detects an error in the received frame, or receives a frame out of order it simply discards the frame.
  - The receiver send only positive ACK for frames received safe; it is silent about the frames damage or lost.
  - The sender has a control variable that holds the number of most recently sent frame (0 or 1). The receiver has control variable R, that holds the number of the next frame expected (0,or 1)

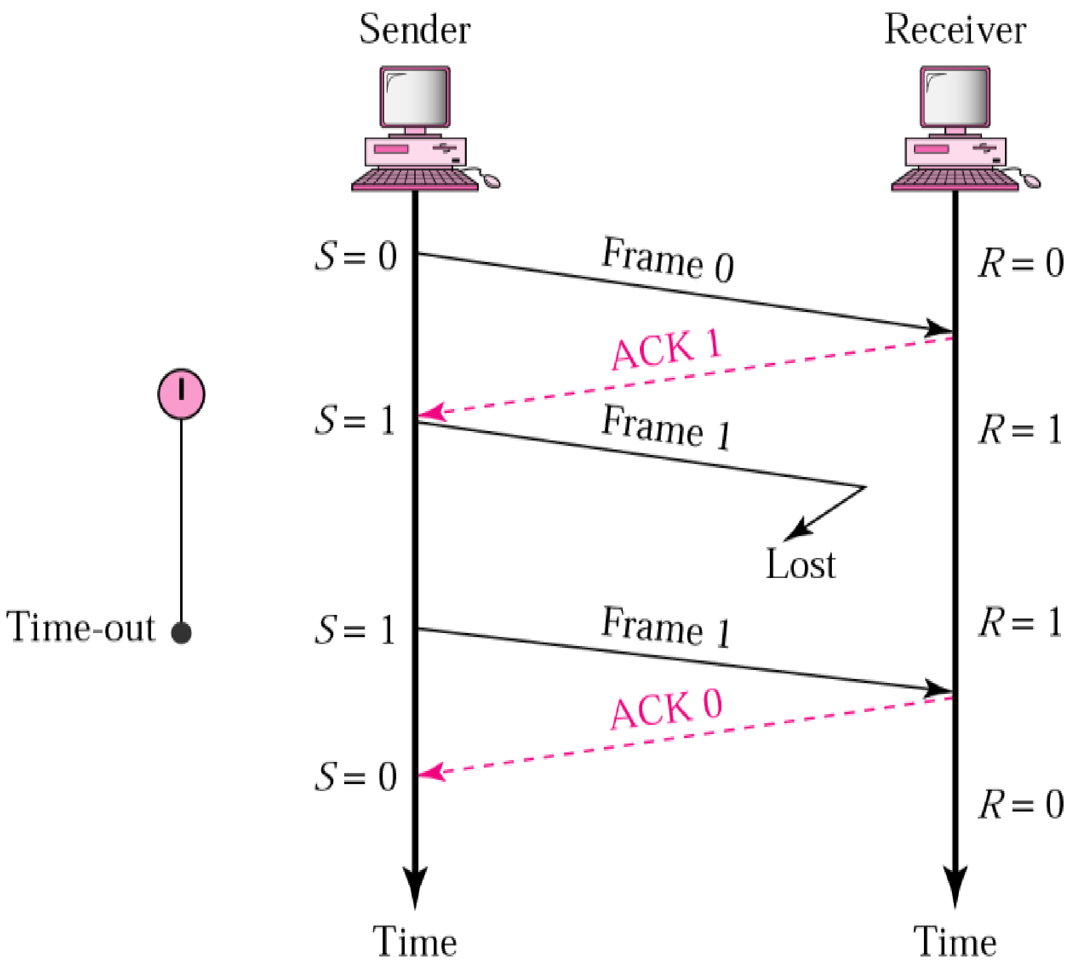
Stop and Wait (ARQ): Normal operation

- The sender will not send the next frame until it is sure that the current one is correctly receive
- sequence number is necessary to check for duplicated frames



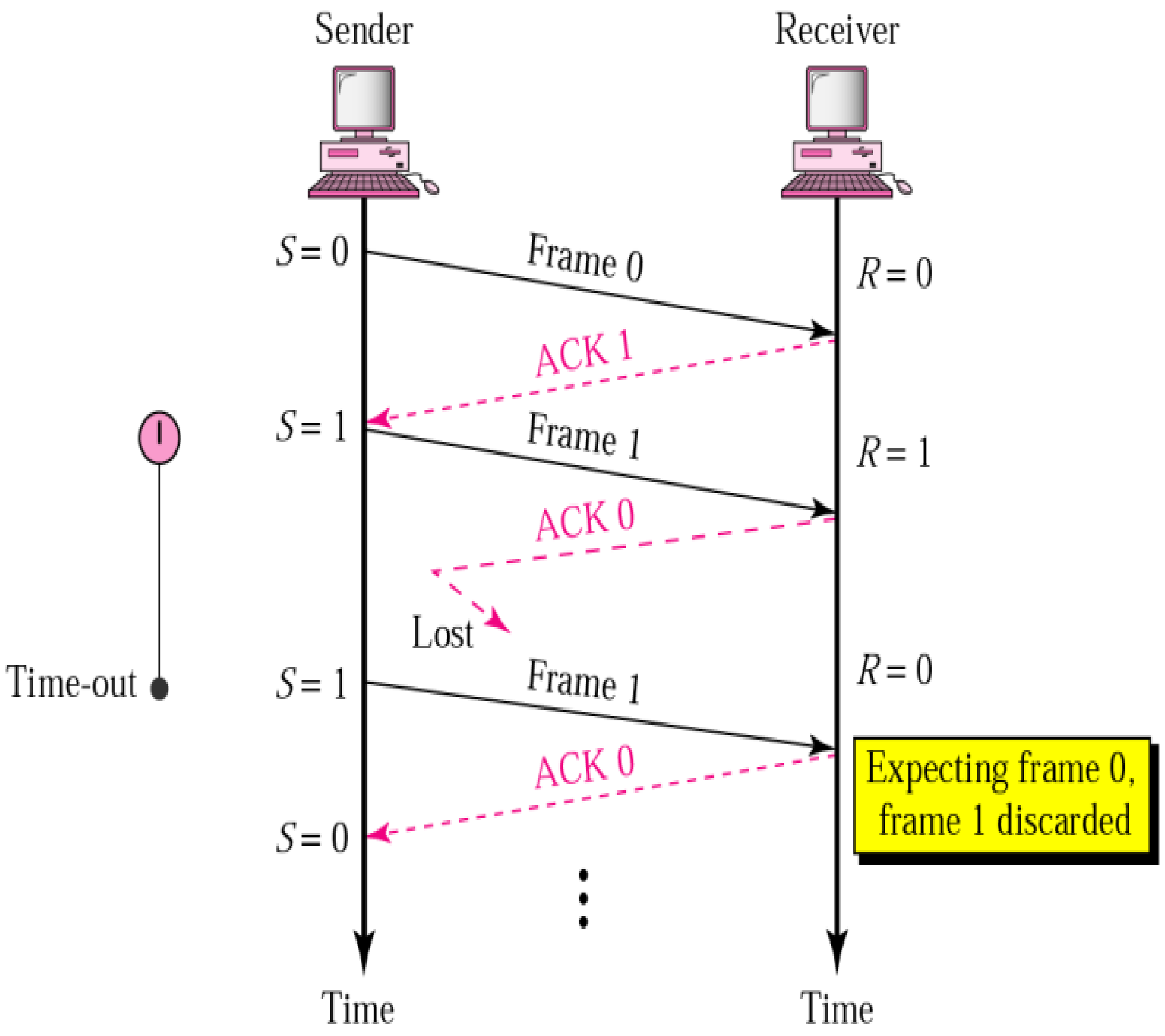
Stop *and Wait* (ARQ): The frame is lost

- A damage or lost frame treated by the same manner by the receiver.
- No NACK when frame is corrupted / duplicate.
- When a receiver receives a damaged frame, it discards it and keeps its value of R.
- After the timer at the sender expires, another copy of frame 1 is sent.



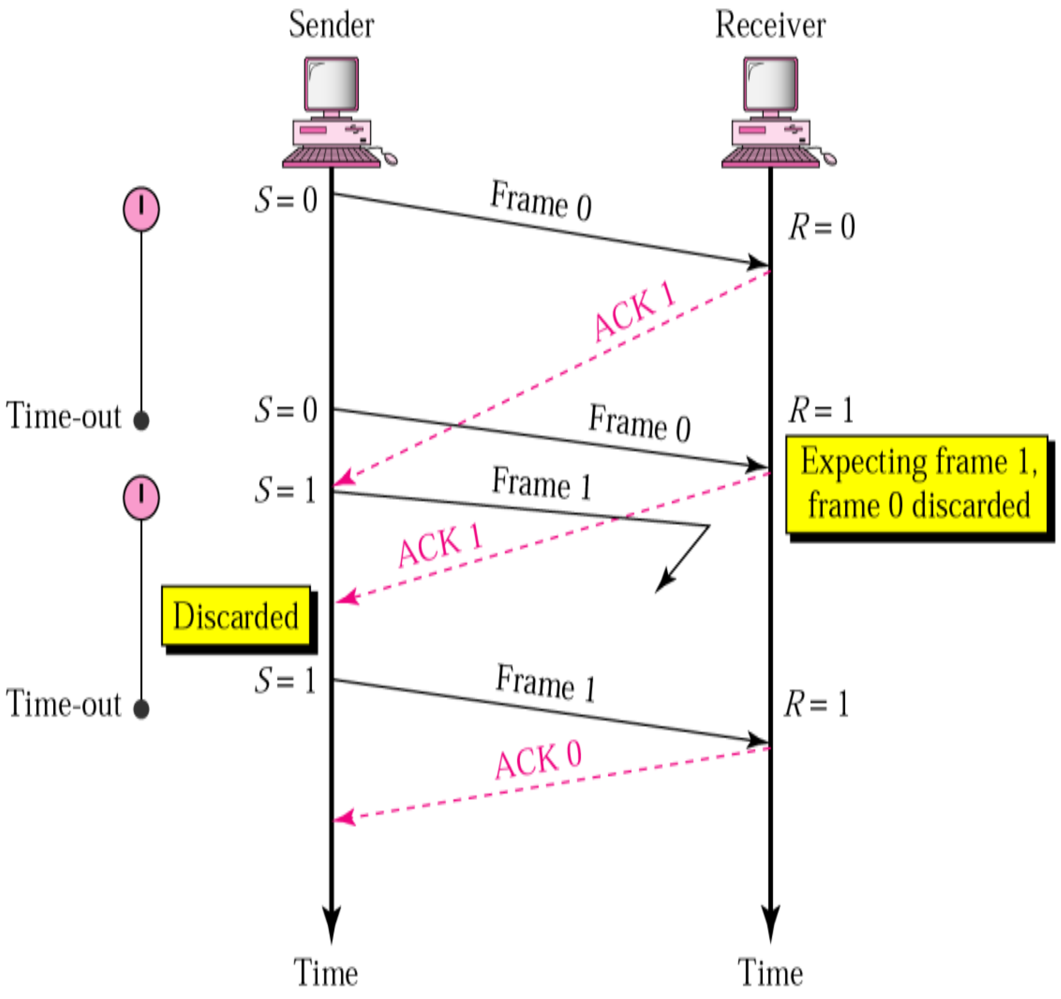
Stop *and Wait* (ARQ): The Acknowledgment (ACK) is lost

- If the sender receives a damaged ACK, it discards it.
- When the timer of the sender expires, the sender retransmits frame 1.
- Receiver has already received frame 1 and expecting to receive frame 0 (R=0). Therefore it discards the second copy of frame 1.
- ***In Stop and-Wait ARQ, numbering frames prevents the retaining of duplicate frames.***



Stop *and Wait* (ARQ): The ACK is delayed

- The ACK can be delayed at the receiver or due to some problem.
- It is received after the timer for frame 0 has expired.
- Sender retransmitted a copy of frame 0. However, R =1 means receiver expects to see frame 1. Receiver discards the duplicate frame 0.
- Sender receives 2 ACKs, it discards the second ACK.
- ***Numbered acknowledgments are needed if an acknowledgment is delayed and the next frame is lost.***

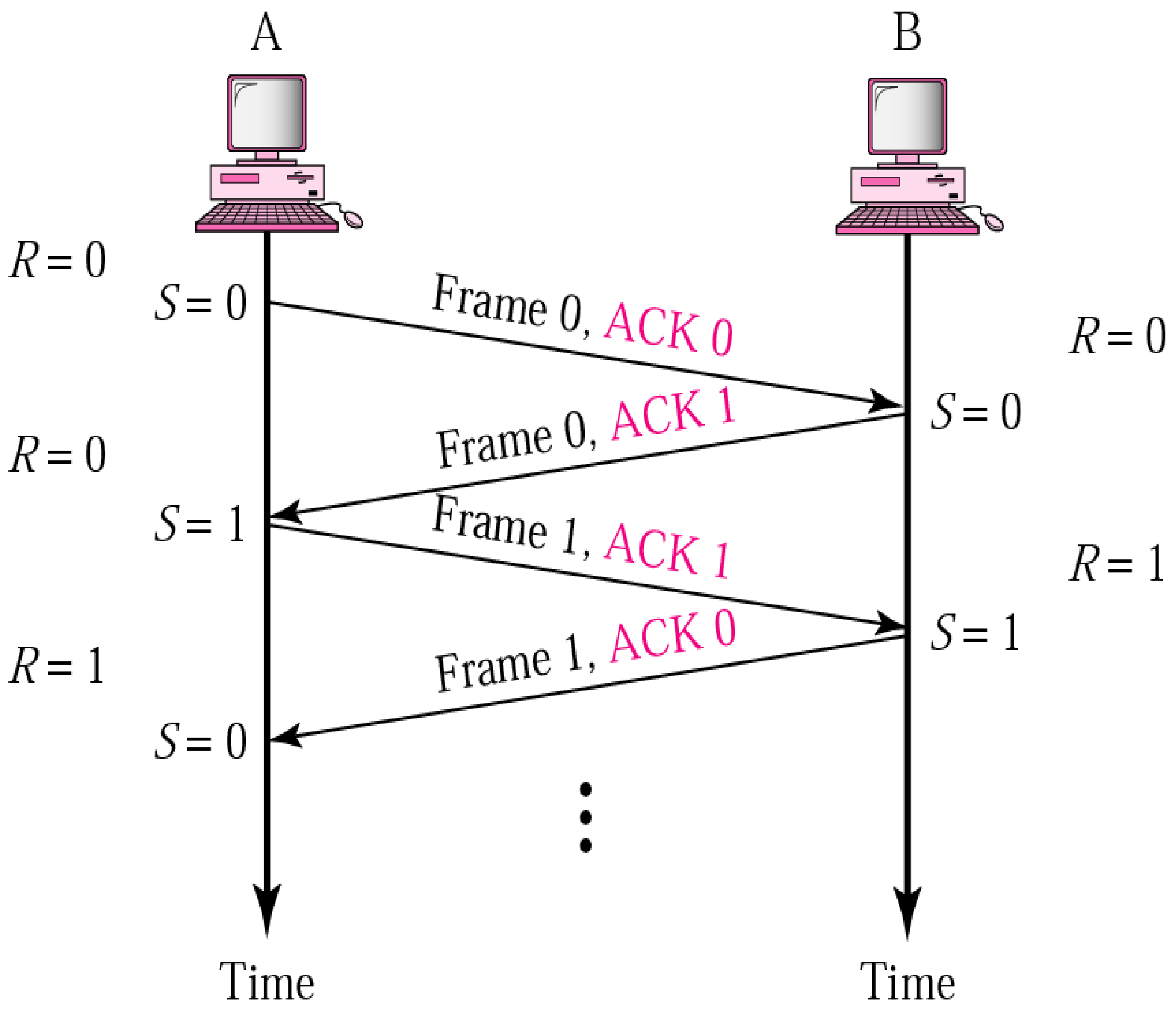


Disadvantage of Stop-and-Wait

- In stop-and-wait, at any point in time, there is only one frame that is sent and waiting to be acknowledged.
- This is not a good use of transmission medium.
- To improve efficiency, multiple frames should be in transition while waiting for ACK.
- Two protocol use the above concept: *Sliding Window protocol*
    - ***Go-Back-N ARQ***
    - ***Selective Repeat ARQ***

Piggybacking

- A method to combine a data frame with ACK.
- Station A and B both have data to send.
- Instead of sending separately, station A sends a data frame that includes an ACK.
- Station B does the same thing.
- Piggybacking saves bandwidth



Sliding Window protocol

- ***Sending***
  - The sender has a “window” of frames that it can be sending at any point in time.
  - The larger the window, the more frames that it can have “on the go” at once.
  - All of the frames in the window must be buffered in case one must be resent.
  - The size of the sending window does not have to match the receiver, nor must it remain a constant size.
- ***Receiving***
  - This is the window of frames that the receiver is allowed to receive at any point in time.
  - A receiving window of 1 means that the frames must be received one-at-a-time and in order.

Sliding Window protocol

- ***Receiving Contd…***
  - A receiving window larger than 1 results in buffering of frames at the receiver end.
  - Anything outside of the receiving window is automatically discarded.
  - Anything inside the window can be accepted.
  - If the sequence number is equal to the lower edge of the window, then that frame is acknowledged and the packet passed to the network layer.
  - Other buffered data, if it now has the sequence number of the lower edge of the window, will also be passed to the network layer.
  - As the lower edge is received, the window moves.

Go-Back-N ARQ

- **Setup –Receiver sliding window**
  - The size of the receiver window is always **1** and points to the **next expected frame** number to arrive
  - This means that frames should arrive **in order**
  - If the expected frame is received without errors, the receiver window slides over the **next sequence number**.
- **Operation**
  - The receiver sends a positive ACK if a frame has arrived **without error** and **in order** (with the expected sequence number)
  - Receiver does not have to acknowledge each individual frame received correctly and in order.
  - Receiver can send **cumulative ACK** for several frames (**ACK 5 acknowledges frames (0,1,2,3,4) and expecting frame 5)**

Go-Back-N ARQ

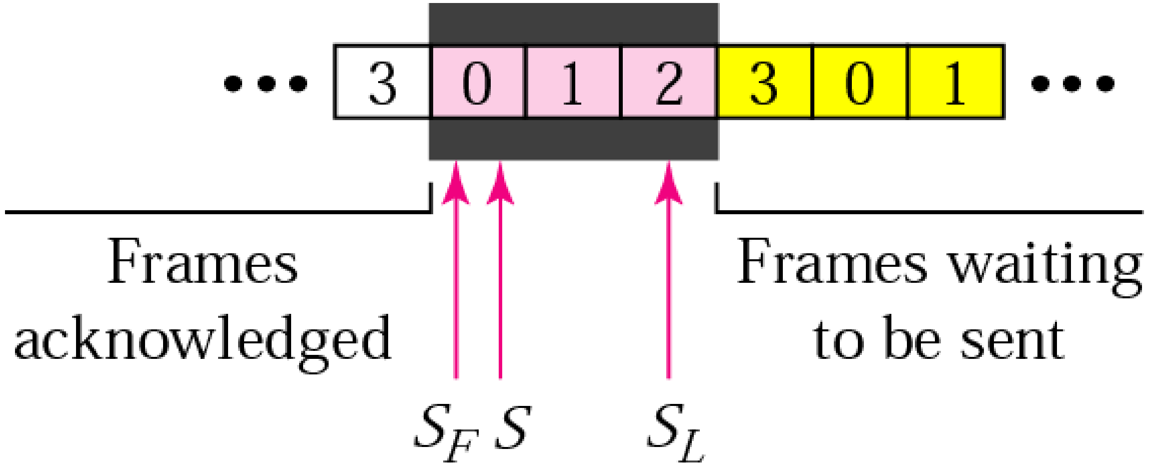
- **Operation Contd…**
  - If the frame is **damaged or out-of-order**, the receiver **discards it** (and **stay *silent***) and also **discards all subsequent frames** until it receives the one expected.
    - ✓In this case, **no** ACK will be transmitted
  - If the sender timer expires before receiving an ACK, it will **resend *ALL* frames beginning with the one expired until the last one sent.**
- ***Sequence Number***
  - Frames from a sender are numbered sequentially.
  - We need to set a limit since we need to include the sequence number of each frame in the header .
  - If the header of the frame allows m bits for sequence number, the sequence numbers range from 0 to $2^m – 1$.
  - for m = 3, sequence numbers are: 0,1, 2, 3, 4, 5, 6, 7.
  - We can repeat the sequence number. Sequence numbers are: 0, 1, 2, 3, 4, 5, 6, 7, 0, 1, 2, 3, 4, 5, 6, 7, 0, 1, …

Go-Back-N ARQ: Acknowledgement

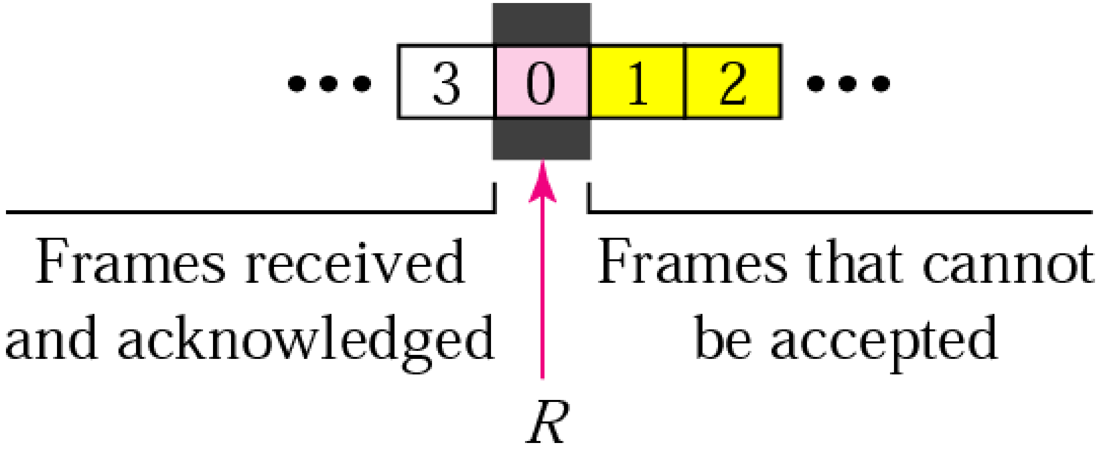
- Receiver sends positive ACK if a frame arrived safe and in order.
- If the frames are damaged/out of order, receiver is silent and discard all subsequent frames until it receives the one it is expecting.
- The silence of the receiver causes the timer of the unacknowledged frame to expire.
- Then the sender resends all frames, beginning with the one with the expired timer.
- For example, suppose the sender has sent frame 6, but the timer for frame 3 expires (i.e. frame 3 has not been acknowledged), then the sender goes back and sends frames 3, 4, 5, 6 again. Thus it is called Go-Back-N-ARQ
- The receiver does not have to acknowledge each frame received, it can send one cumulative ACK for several frames.

Go-Back-N ARQ: Control Variables

- Sender has 3 variables: S, S_F, and S_L
- S holds the sequence number of recently sent frame
- S_F holds the sequence number of the first frame
- S_L holds the sequence number of the last frame
- Receiver only has the one variable, R, that holds the sequence number of the frame it expects to receive. If the seq. no. is the same as the value of R, the frame is accepted, otherwise rejected.



a. Sender window

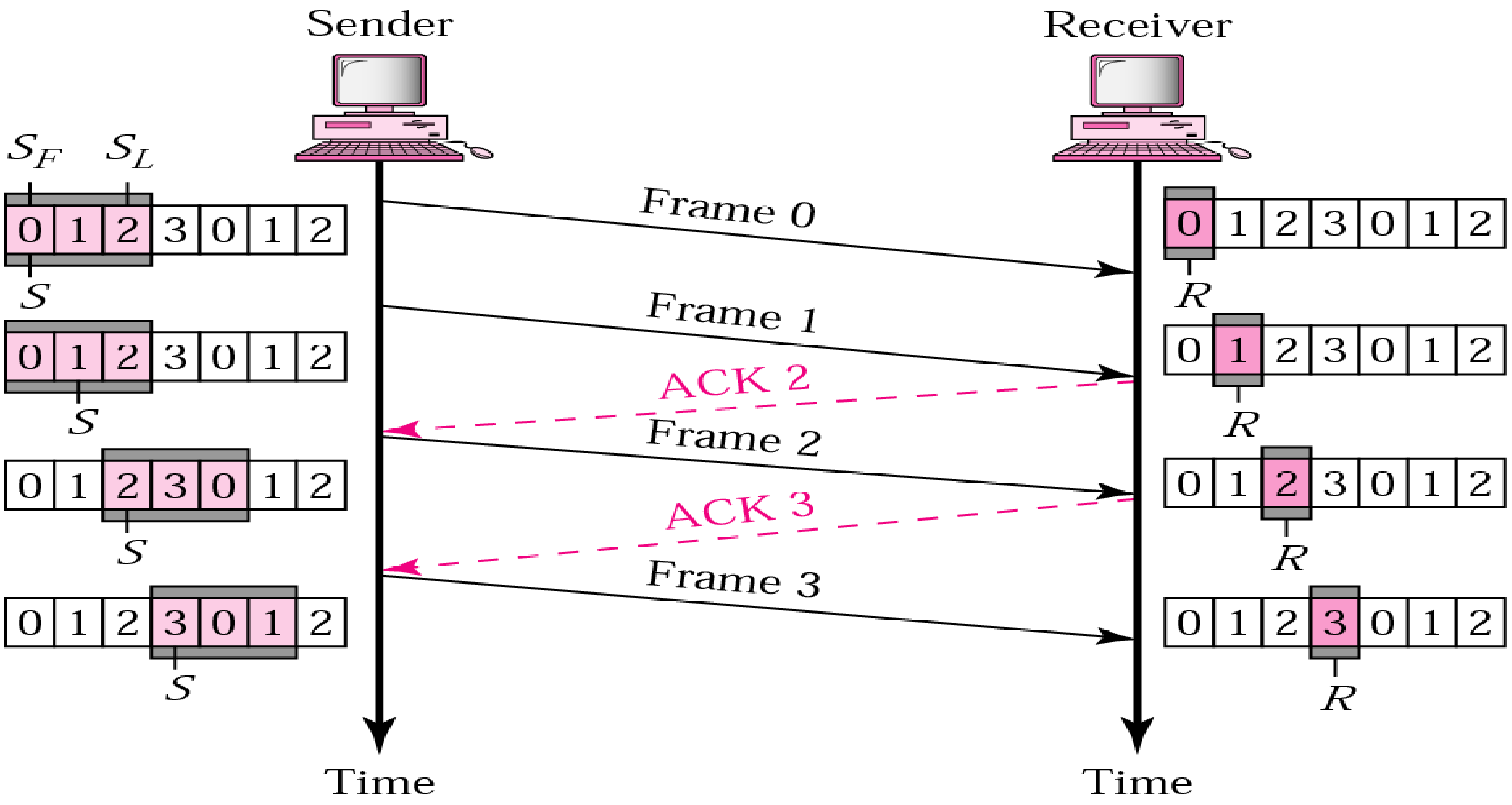


• • •
3
0
1
2
• • •
Frames received and acknowledged
Frames that cannot be accepted
R

b. Receiver window

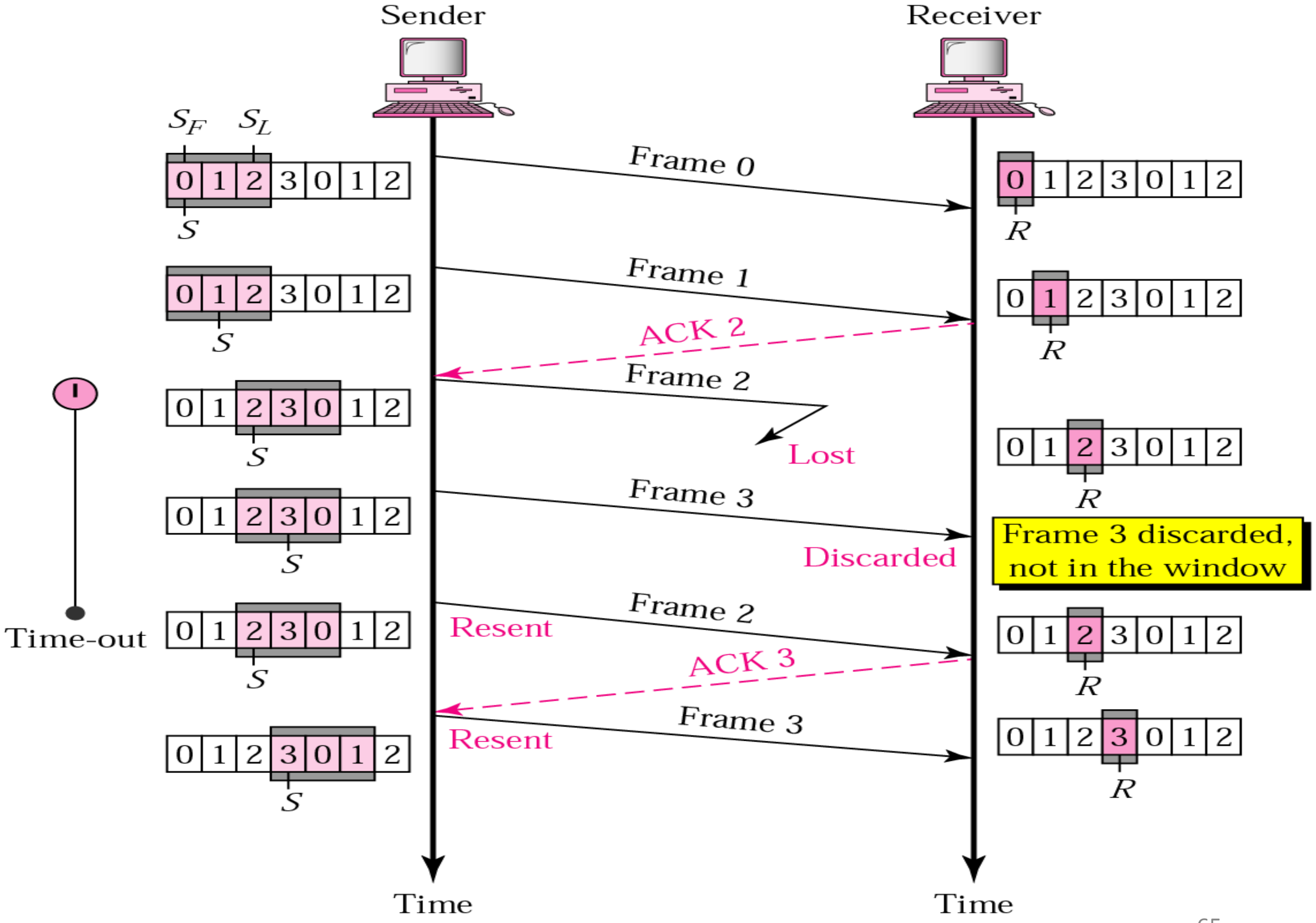
Go-Back-N ARQ: Normal Operation

- The sender keeps track of the outstanding frames and updates the variables and windows as the ACKs arrive.



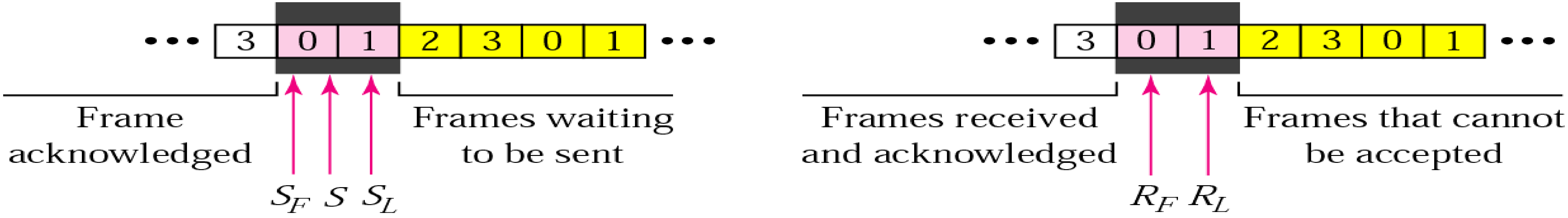
Go-Back-N ARQ: Lost Frame

- Frame 2 is lost
- When the receiver receives frame 3, it discards frame 3 as it is expecting frame 2 (according to window).
- After the timer for frame 2 expires at the sender site, the sender sends frame 2 and 3. (go back to 2)



Selective Repeat ARQ: Sender And Receiver Windows

- Go-Back-N ARQ simplifies the process at the receiver site. Receiver only keeps track of only one variable, and there is no need to buffer out-of-order frames, they are simply discarded.
- However, Go-Back-N ARQ protocol is inefficient for noisy link. It bandwidth inefficient and slows down the transmission.
- In Selective Repeat ARQ, only the damaged frame is resent. More bandwidth efficient but more complex processing at receiver.
- It defines a negative ACK (NAK) to report the sequence number of a damaged frame before the timer expires.

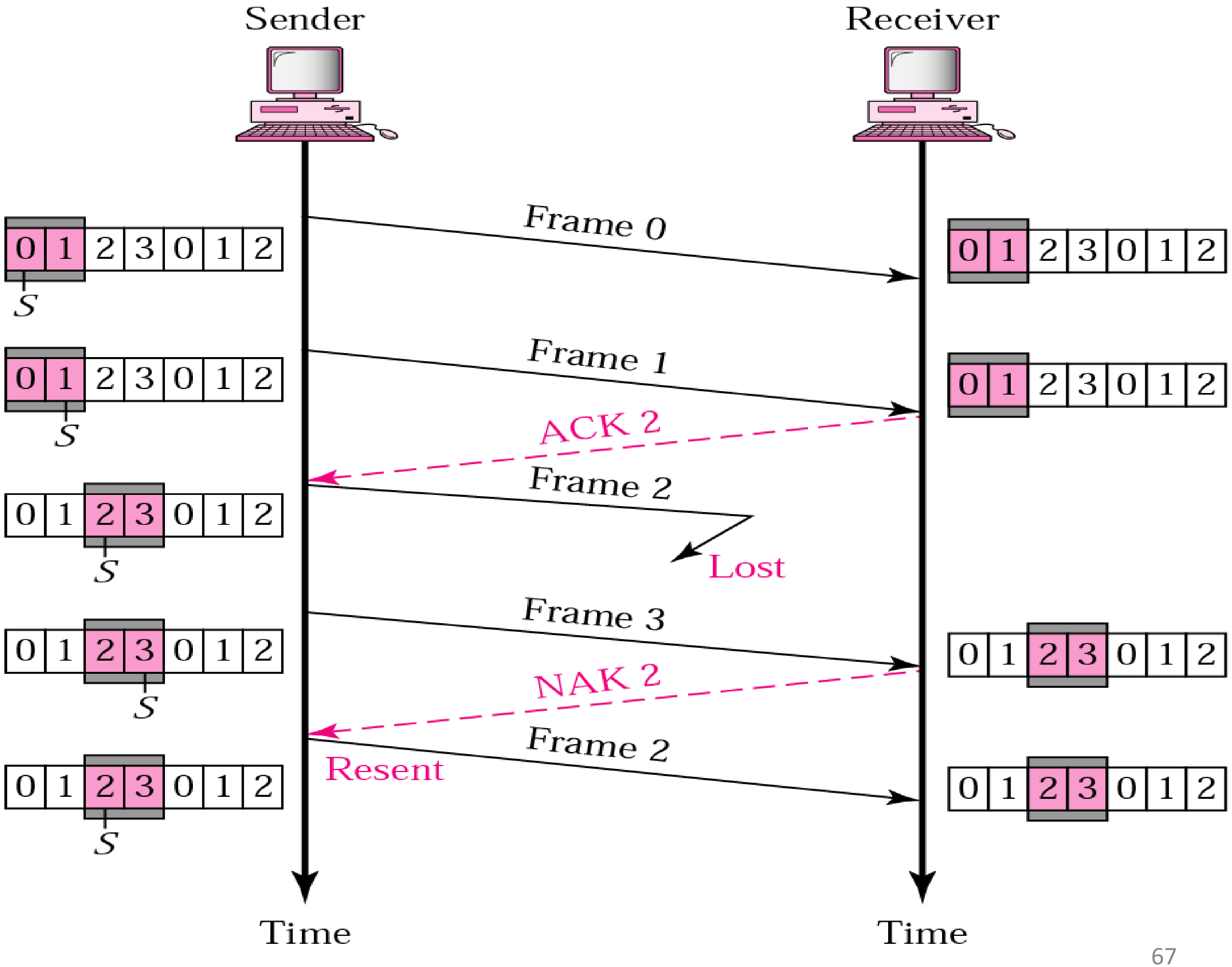


a. Sender window

b. Receiver window

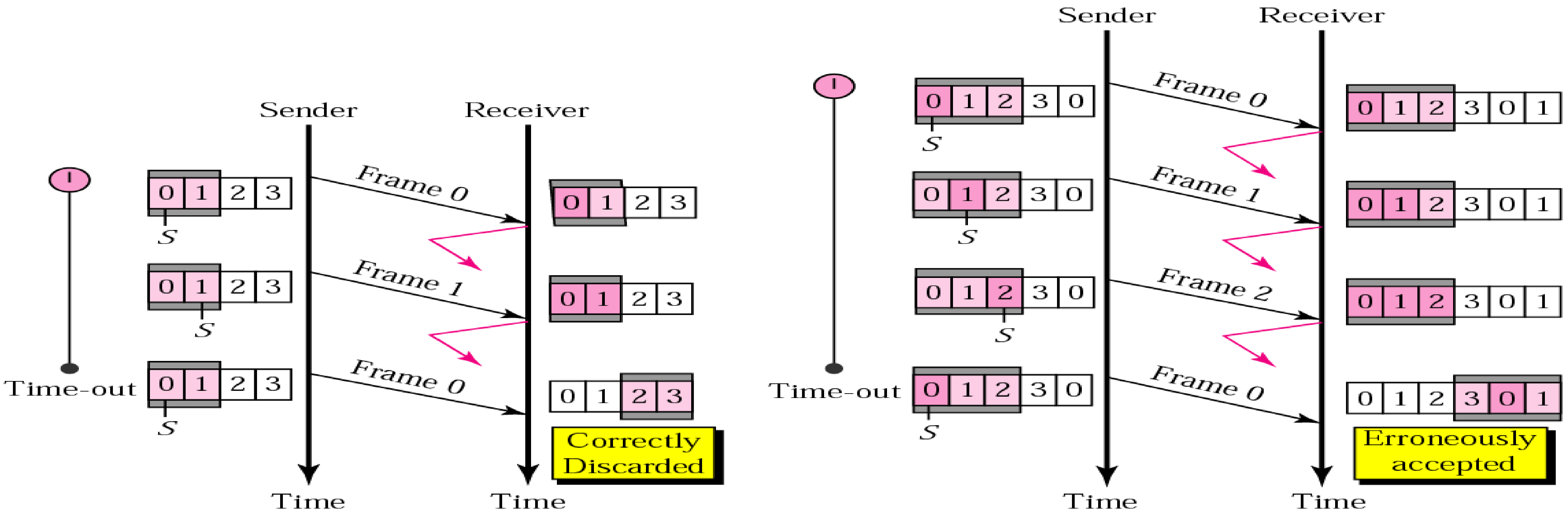
Selective Repeat ARQ: Lost Frame

- Frames 0 and 1 are accepted when received because they are in the range specified by the receiver window. Same for frame 3.
- Receiver sends a NAK2 to show that frame 2 has not been received and then sender resends only frame 2 and it is accepted as it is in the range of the window.



Selective Repeat ARQ: Sender Window Size

- Size of the sender and receiver windows must be at most one-half of 2^m. If m = 2, window size should be $2^m/2 = 2$.
- Fig compares a window size of 2 with a window size of 3. Window size is 3 and all ACKs are lost, sender sends duplicate of frame 0, window of the receiver expect to receive frame 0 (part of the window), so accepts frame 0, as the 1st frame of the next cycle – an **error**.



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

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

Example Data Link Protocol: HDLC

- High-level Data Link Control (HDLC)
- It is a **bit-oriented** protocol using **bit stuffing**.
- It supports **half and full duplex** communication over **point-to-poin**t and **multipoint** links.
- It implements the **ARQ** mechanisms.
- HDLC provides **two common transfer modes** that can be used in different configurations: **normal response mode (NRM)** and **asynchronous balanced mode (ABM)**.

HDLC: Station Types

- **Primary station**
  - Responsible to control the operation of link
  - Frames issued by primary are called commands
- **Secondary station**
  - Operates under control of primary
  - Frames issued by secondary are called responses
- **Combined station**
  - A combination of above
  - Issues commands and responses

HDLC: Link Configurations

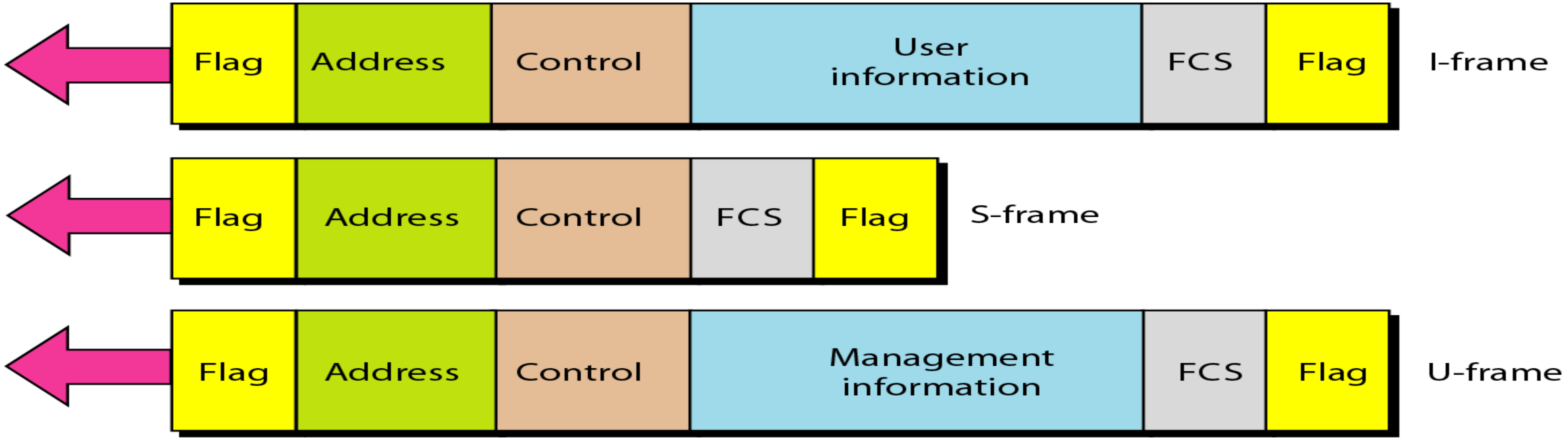
- **Unbalanced**
  - Consists of one primary and one or many secondary stations
  - Primary is responsible for controlling secondary
  - Primary maintains and establishes link and responsible for error recovery
- **Balanced**
  - Consists of two combined stations
  - Can be used on point to point lines only
  - Stations are peer and share equal responsibility for error recovery and line management

HDLC: Data Transfer Modes

- **Normal Response Mode (NRM)**
  - Used with unbalanced configuration
  - Primary may initiate data transfer
  - Secondary can transmit data only as a response
  - Used on point-to-point and multi-point links
- **Asynchronous Balanced Mode**
  - The configuration is balanced.
  - The link is point-to-point, and each station can function as a primary and a secondary (acting as peers).
  - Any combined station may initiate data transfer without permission from the other.
  - Most widely used because more efficient on full duplex point to-point link.

HDLC: Frames

- HDLC defines three types of frames: information frames **(I-frames),** supervisory frames **(S-frames), and** unnumbered frames **(V-frames).**
- **Each type of frame serves** as an **envelope** for the transmission of a different type of message.
- **I-frames** are used **to transport user data and control information** relating to user data (piggybacking).
- **S-frames** are used only to **transport control information**.
- **V-frames** are reserved for **system management**. Information carried by V-frames is intended for managing the link itself.

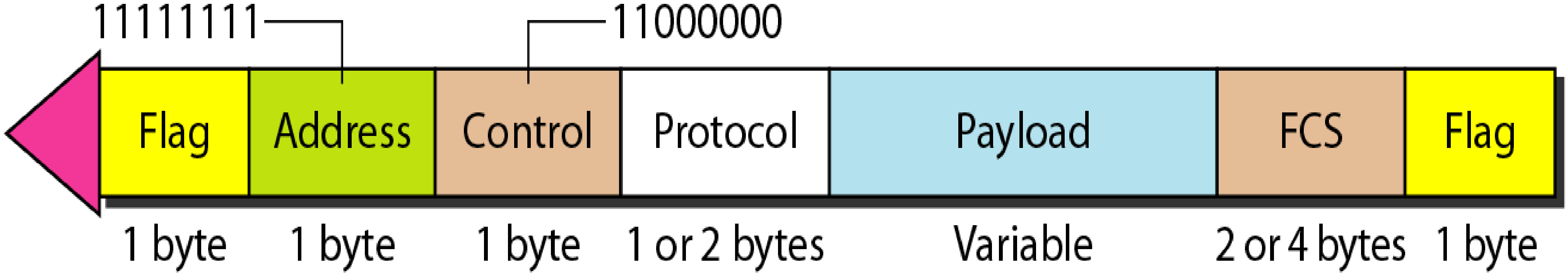


Example Data Link Protocol: PPP

- Point-to-Point Protocol (PPP).
- one of the most common protocols for **point-to-point access is the PPP**.
- PPP uses to connect **home computers to** the server of an **ISP**.
- PPP is a **byte-oriented protocol** using byte stuffing with the escape byte 01111101.
- It provides error detection.
- It defines **Link Control Protocol (LCP)** for:
  - Establishing the link between two devices.
  - Maintaining this established link.
  - Configuring this link.
  - Terminating this link after the transfer.

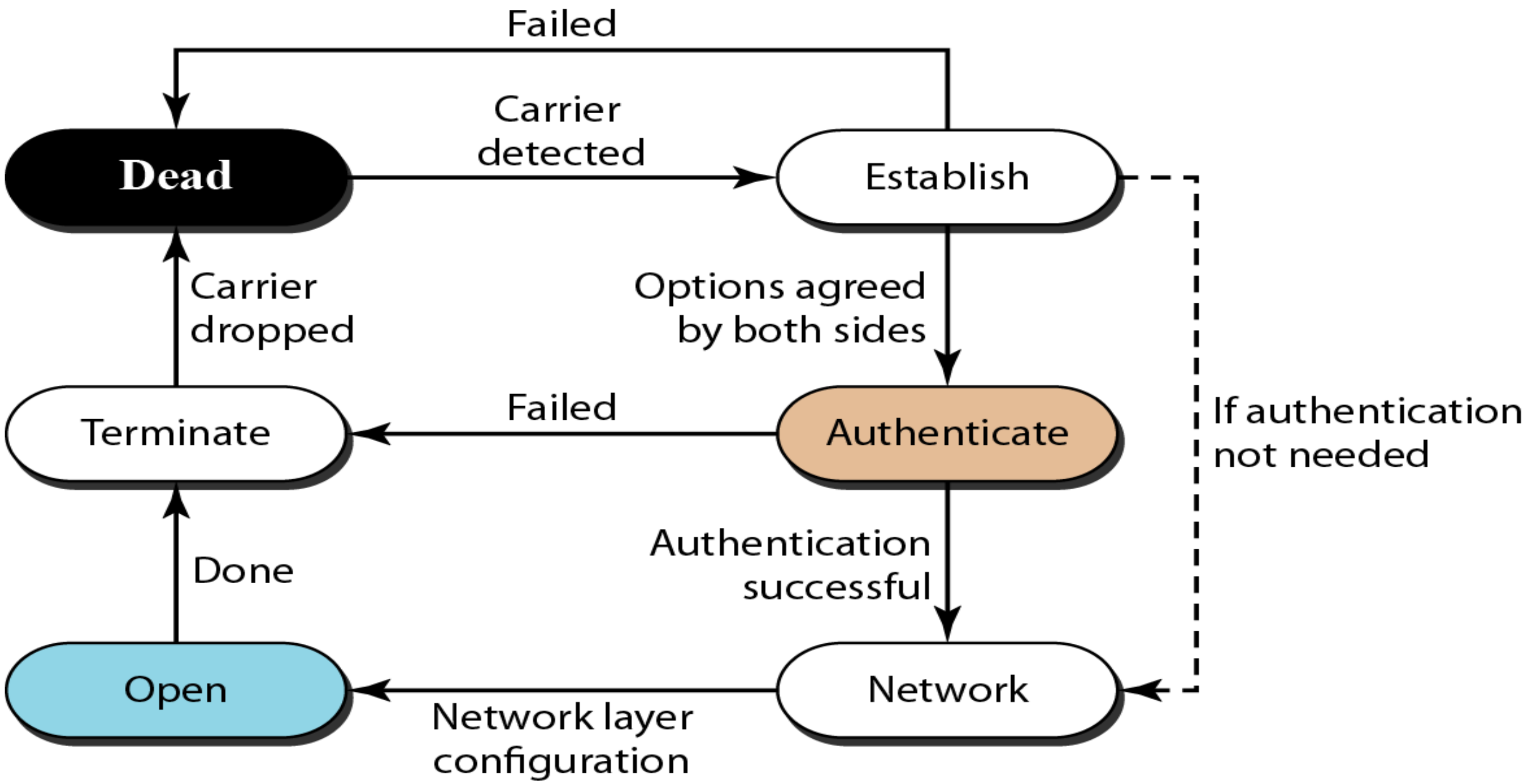
PPP frame format

- **Flag Field:** It marks the beginning and end of the PPP frame. Flag byte is 01111110.
- **Address:** 11111111, which means all stations can accept the frame.
- **Control Field:** It is also of 1 byte. The value is always 00000011 to show that the frame does not contain any sequence number and there is no flow control or error control.
- **Protocol field:** tells what kind of packet is in payload. ***Payload carries user data***.
- **Information Field:** Its length is variable. It carries user data or other information.
- **FCS Field:** It stands for Frame Check Sequence. It contains checksum. It is either 2 bytes or 4 bytes.



PPP Transition phases/Operations

- **Open.** In the open phase, data transfer takes place.
- **Terminate.** In the termination phase the connection is terminated.
- **Authenticate.** The authentication phase is optional; the two nodes may decide, during the establishment phase.

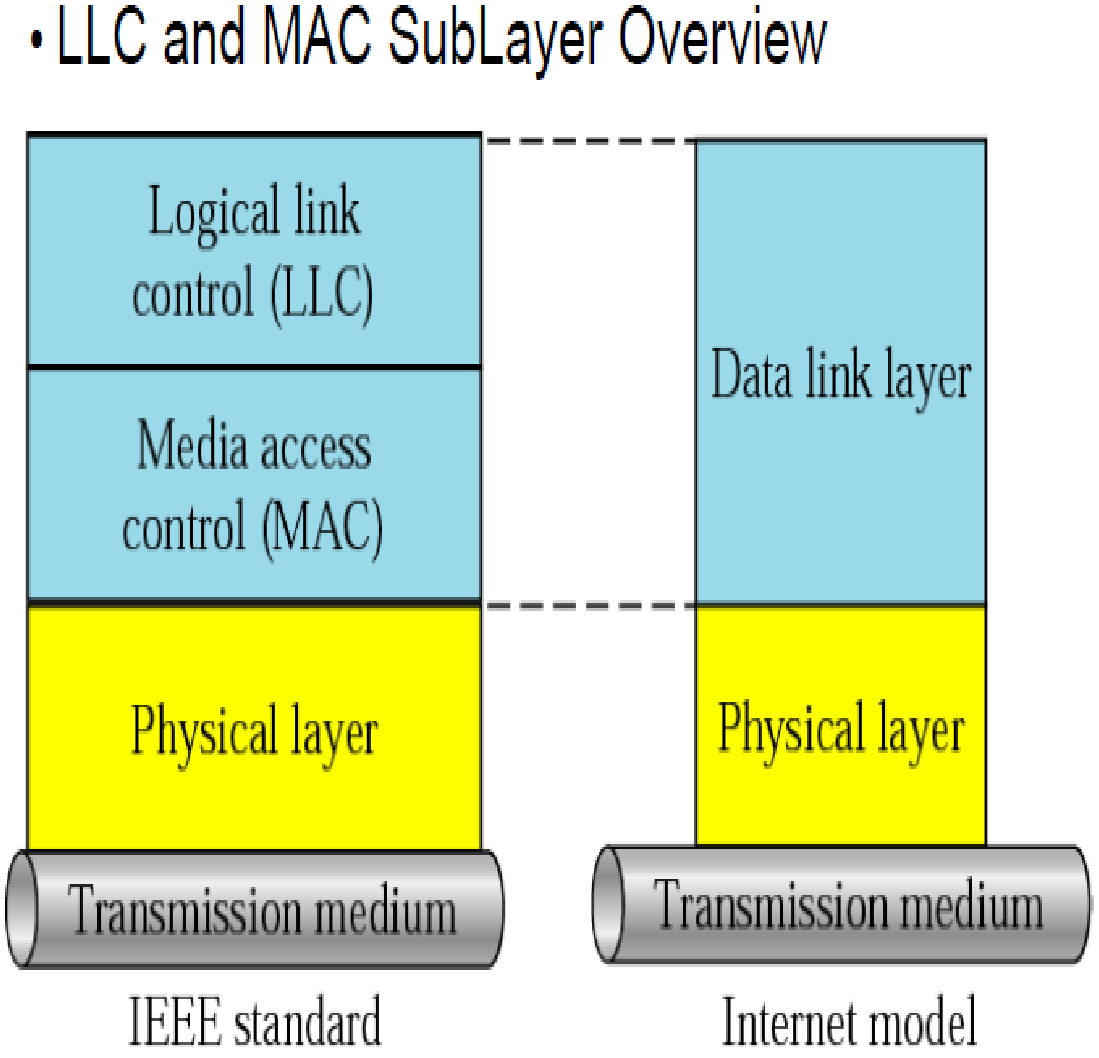

Failed
Carrier detected
Dead
Establish
Carrier dropped
Options agreed by both sides
Terminate
Failed
Authenticate
If authentication not needed
Done
Authentication successful
Open
Network
Network layer configuration

The Medium Access Control Sub-layer

- Networks can be divided into two categories:
  - *those using point-to-point connections and*
  - *those using broadcast channels.*
- This Medium Access Sub-layer **deals with broadcast networks** and their protocols
- In any broadcast network, the key issue is how **to determine who gets** to use the channel when there **is competition** for it. To make this **point clearer**, consider a **conference call in which six people**, on six different telephones, are all connected so that **each one can hear and talk to all the others.**
- It is very likely that when one of them stops speaking, two or more will start talking at once, leading to chaos.
- In a **face-to-face meeting**, chaos is avoided by external means, for example, at a meeting, people **raise their hands to request permission to speak**. When only a single channel is available, determining who should go next is much harder.

The Medium Access Control Sub-layer

- In the literature, broadcast channels are sometimes referred to as **multi-access channels** or **random access channels**.
- The protocols used to determine who goes next on a multi-access channel belong to a sub-layer of the data link layer called the **MAC** (**Medium Access Control**) sub-layer.
- The MAC sub-layer is especially important in LANs, many of which use a multi-access channel as the basis for communication.



The Channel Allocation Problem

- The central theme is how to allocate a single broadcast channel among competing users.
- The channel might be a portion of the wireless spectrum in a geographic region, or a single wire or optical fiber to which multiple node are connected.
- In broadcast networks, single channel is shared by several stations.
- This channel can be allocated to only one transmitting user at a time.
- There are two different methods of channel Allocations:
  - Static Channel Allocation
  - Dynamic Channel Allocation

Static Channel Allocation

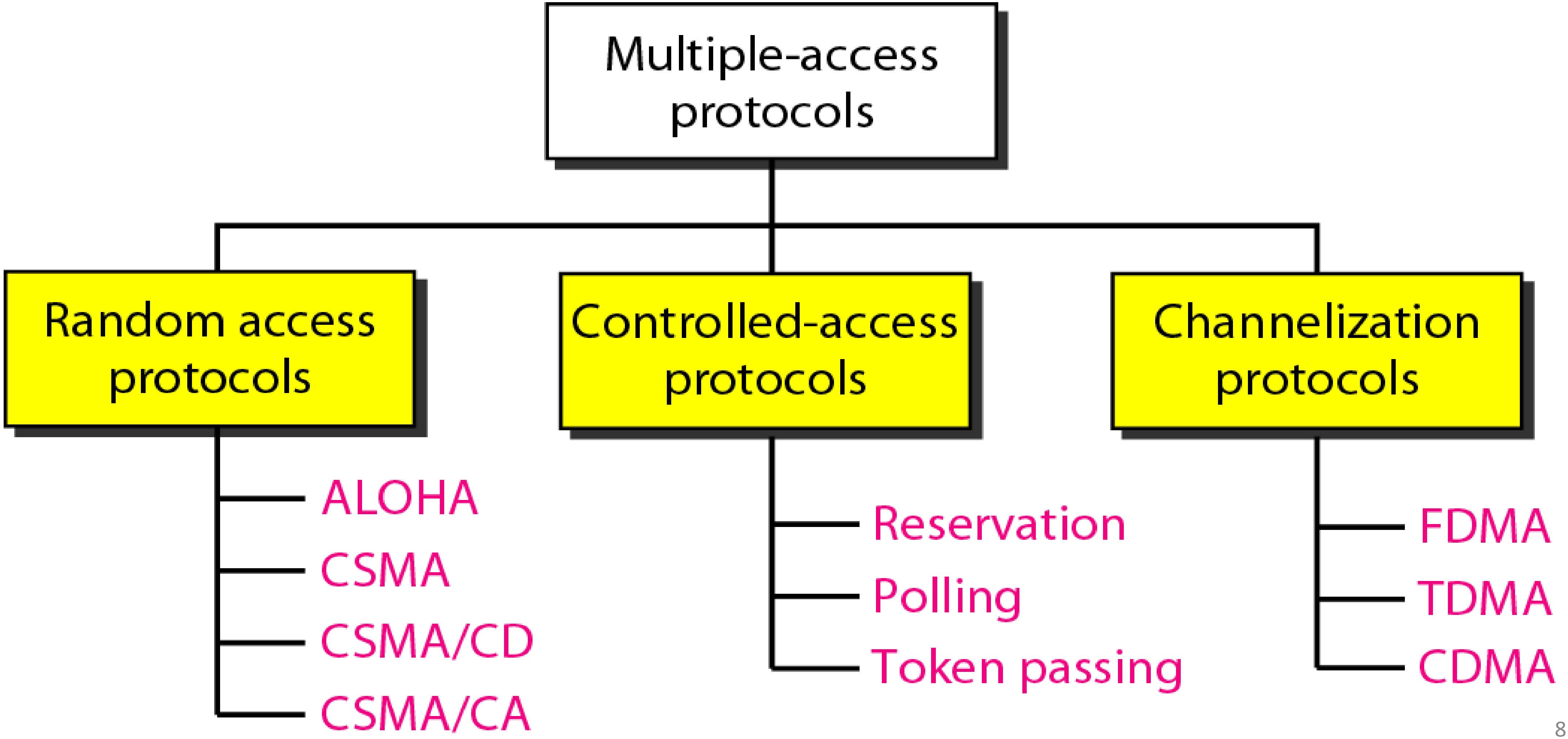
- The **traditional way of allocating a single channel**, such as a telephone trunk, among multiple competing users is FDM.
- If there are **N users**, the bandwidth is **divided** into **N equal-sized** portions, each user being assigned one portion. **Since each user has a private frequency** band, there is no interference between users.
- When there is only **a small and constant number** of users, each of which has a **heavy (buffered) load of traffic** (e.g., carriers' switching offices), **FDM** is a **simple** and **efficient** allocation **mechanism**.
- **However**, when the number of senders is large and **continuously varying** or the traffic is bursty, **FDM presents some problems**.
- If the spectrum is cut up into **N regions and fewer than N users** are **currently interested** in communicating, a large piece of valuable spectrum **will be wasted**.
- If **more than N users want to communicate**, some of them will be denied permission for lack of bandwidth, **even if some of the users** who have been assigned a frequency band **hardly ever transmit or receive anything**.

Dynamic Channel Allocation

- In this method, no user is assigned fixed frequency or fixed time slot.
- All users are dynamically assigned frequency or time slot, depending upon the requirements of the user.
- ***Key Assumptions for Dynamic Channel Allocation:***
  - **Independent Traffic:** The model consists of *N* independent stations.
  - **Single Channel:** A single channel is available for all communication. All stations can transmit on it and all can receive from it.
  - **Observable Collisions**: All stations can detect that a collision has occurred. A collided frame must be transmitted again later.
  - **Continuous or Slotted Time**: Time may be assumed continuous. Alternatively, time may be slotted or divided into discrete intervals.
  - **Carrier Sense or No Carrier Sense**: With the carrier sense assumption, stations can tell if the channel is in use before trying to use it.

Multiple Access Protocols

- Distributed algorithm that determines how nodes share channel, i.e., determine when node can transmit.



Random Access Protocols

- ***Two features give this method its name.***
  - First, 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*.
  - Second, **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.
- The various random access methods are:
  - ALOHA
  - CSMA (Carrier Sense Multiple Access)
  - CSMA/CD (Carrier Sense Multiple Access with Collision Detection)

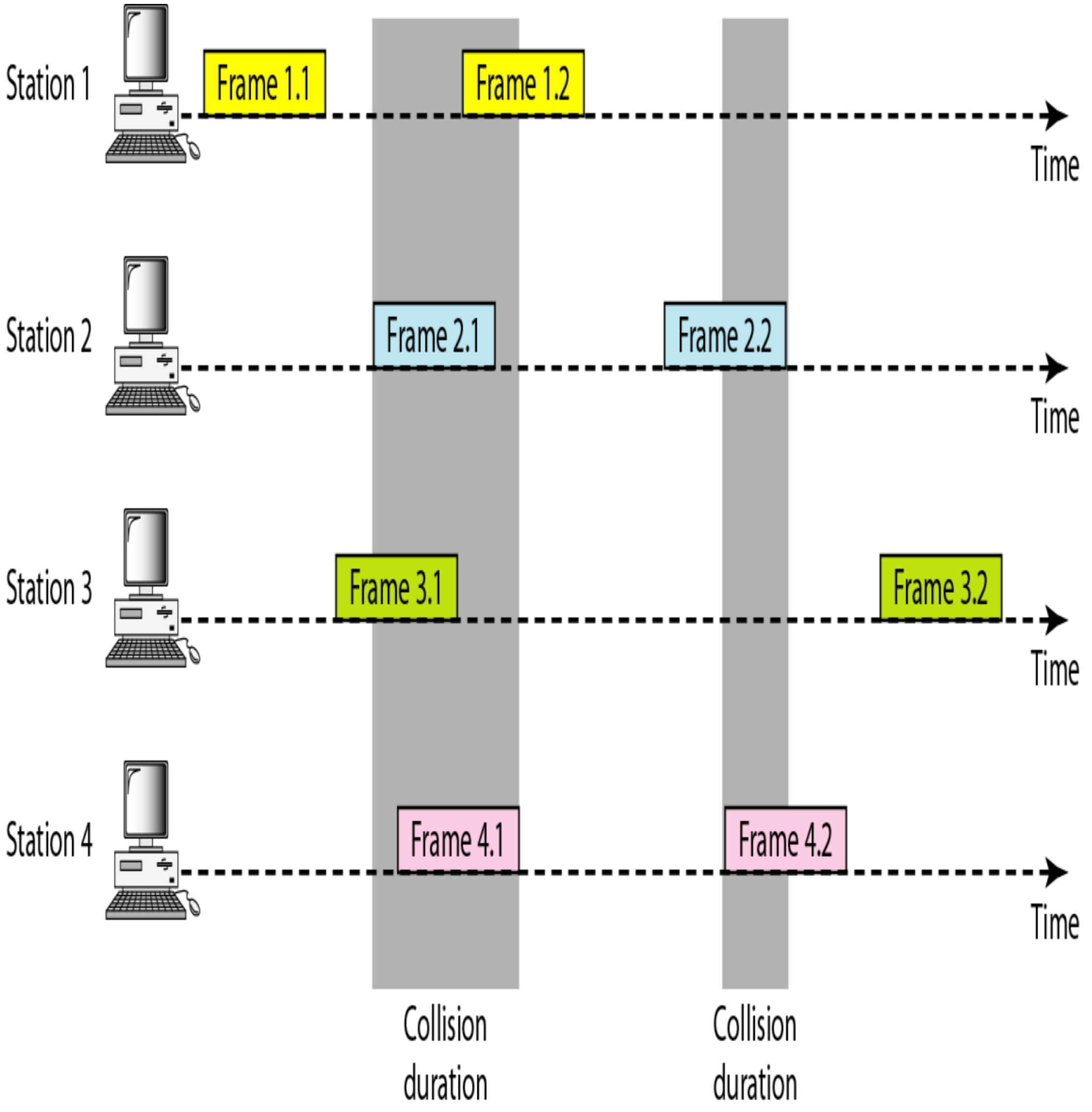
ALOHA

- In the 1970s, Norman Abramson and his colleagues at the University of Hawaii devised a new and elegant method to solve the channel allocation problem.
- Their work has been extended by many researchers since then (Abramson, 1985).
- Although Abramson's work, called the ALOHA system, used ground-based radio broadcasting, the basic idea is applicable to any system in which uncoordinated users are competing for the use of a single shared channel.
- There are two different versions of ALOHA:
    - **Pure ALOHA**
    - **Slotted ALOHA**

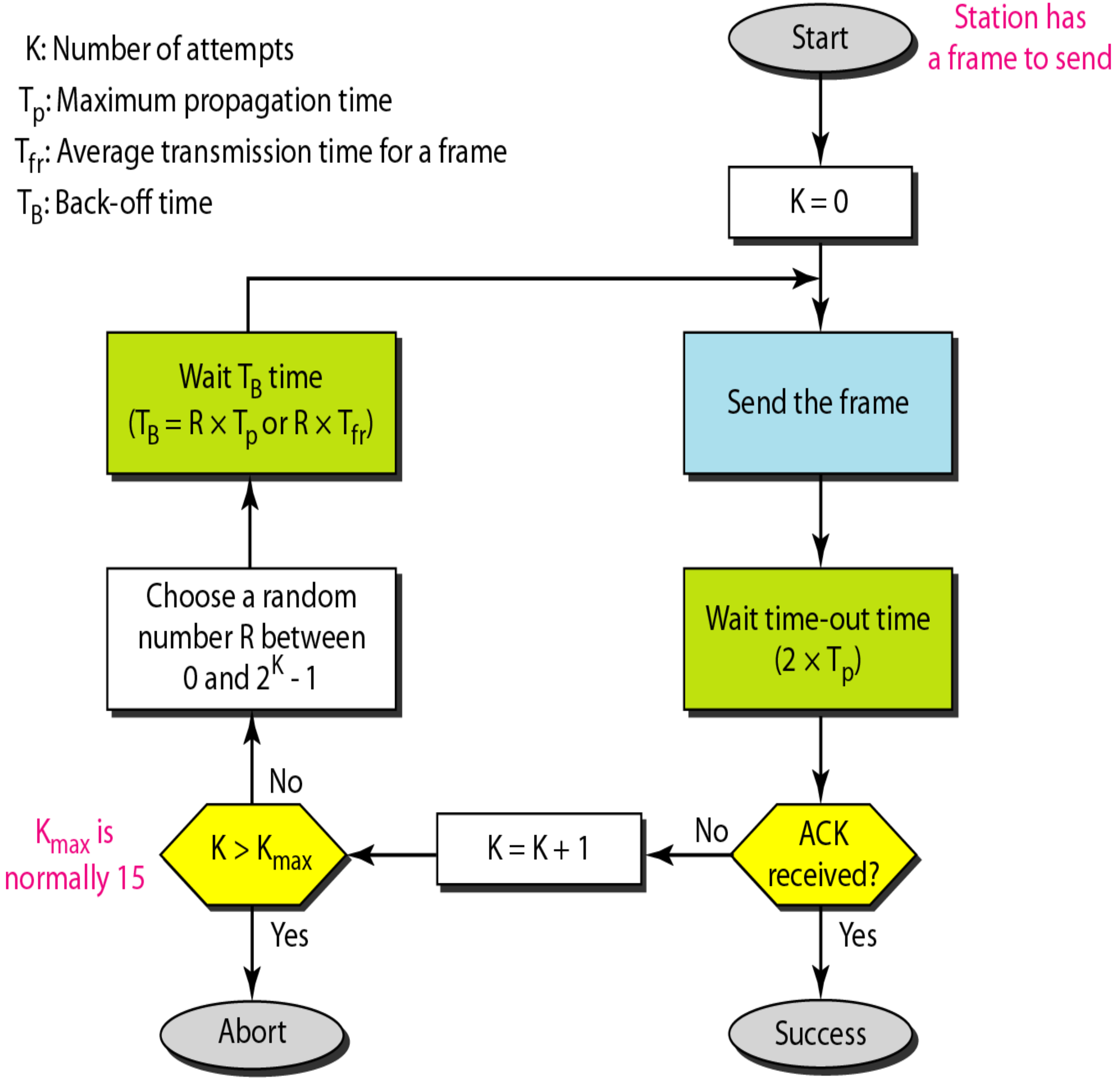
Pure ALOHA

- In pure ALOHA, **stations transmit frames whenever they have** data to send.
- When two stations **transmit simultaneously**, there is **collision** and frames are **lost**.
- In pure ALOHA, whenever any station **transmits a frame**, it **expects** an **acknowledgement** from the receiver. If **acknowledgement is not received** within specified time, the station **assumes that the frame has been lost**.
- If the frame is **lost**, station **waits** for a **random amount** of time and **sends it again.** This waiting time must be random, otherwise, same frames will collide again and again.
- If first bit of a new frame **overlaps with the last bit** of a frame almost finished, **both frames will be los**t and both will have to be **retransmitted**.

- **Back-off time:** *random amount of waiting time before resending frame.*



Frames in a pure ALOHA

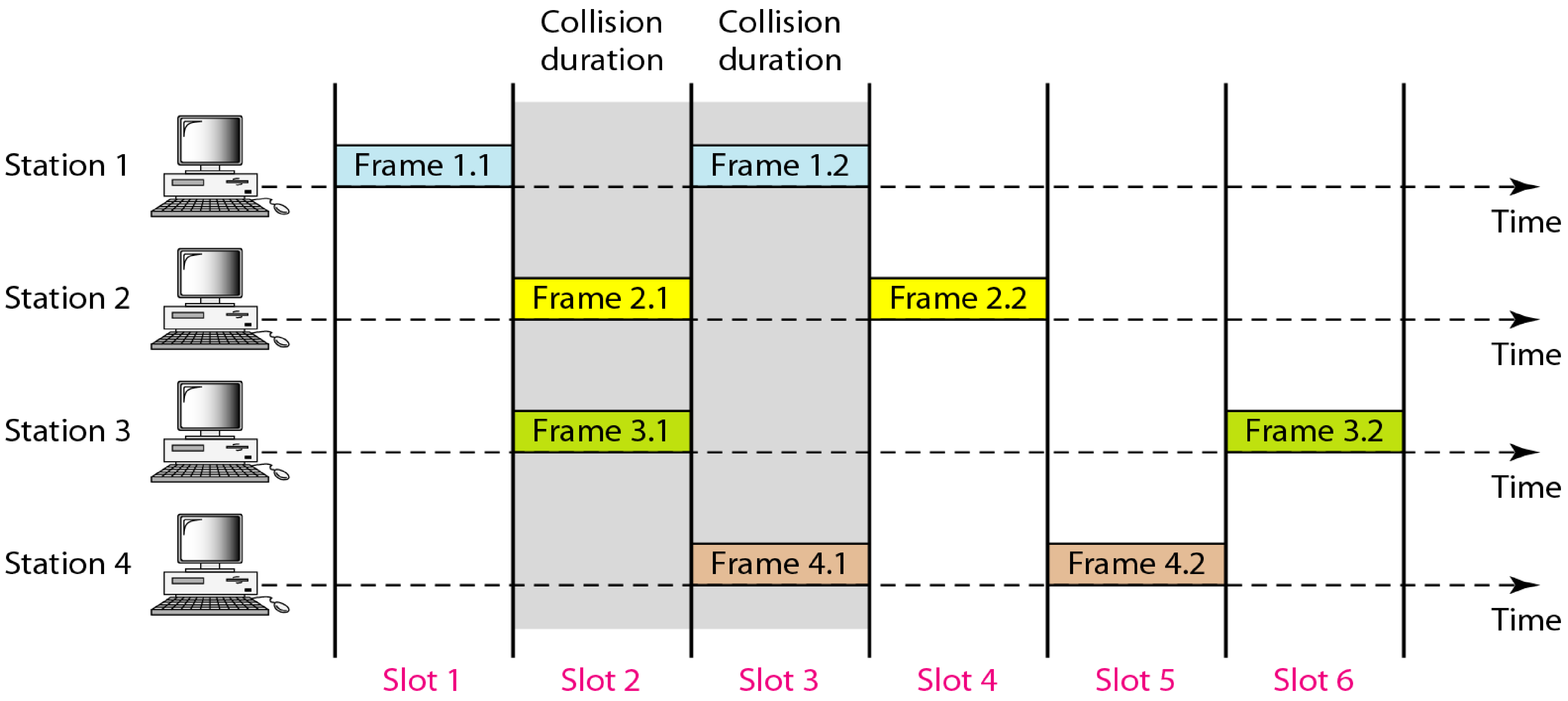

K: Number of attempts
T_p: Maximum propagation time
T_{fr}: Average transmission time for a frame
T_B: Back-off time
Start
Station has a frame to send
K = 0
Send the frame
Wait time-out time ($2 \times T_p$)
ACK received?
No
Yes
Success
K = K + 1
$K > K_{max}$
K_{max} is normally 15
No
Yes
Abort
Choose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

Procedure for pure ALOHA

Slotted ALOHA

- Slotted ALOHA was invented to improve the efficiency of pure ALOHA.
- In slotted ALOHA, time of the channel is **divided into intervals** called slots.
- The station can send a frame **only at the beginning** of the slot and only one frame is sent in each slot.
- If any station is **not able to place** the frame onto the channel **at the beginning** of the slot, it **has to wait** until the next time slot.
- There is still a **possibility of collision** if two stations try to **send at the beginning** of the same time slot.
- The throughput for **slotted ALOHA** is $S = G \times e^{-G}$. The maximum throughput $S_{max} = 0.368$ when G = 1.
- G the average number of frames generated by the system during one frame transmission time.
- The throughput for **pure ALOHA** is $S = G \times e^{-2G}$. The maximum throughput $S_{max} = 0.184$ when G= (1/2).

Slotted ALOHA



Frames in a slotted ALOHA network

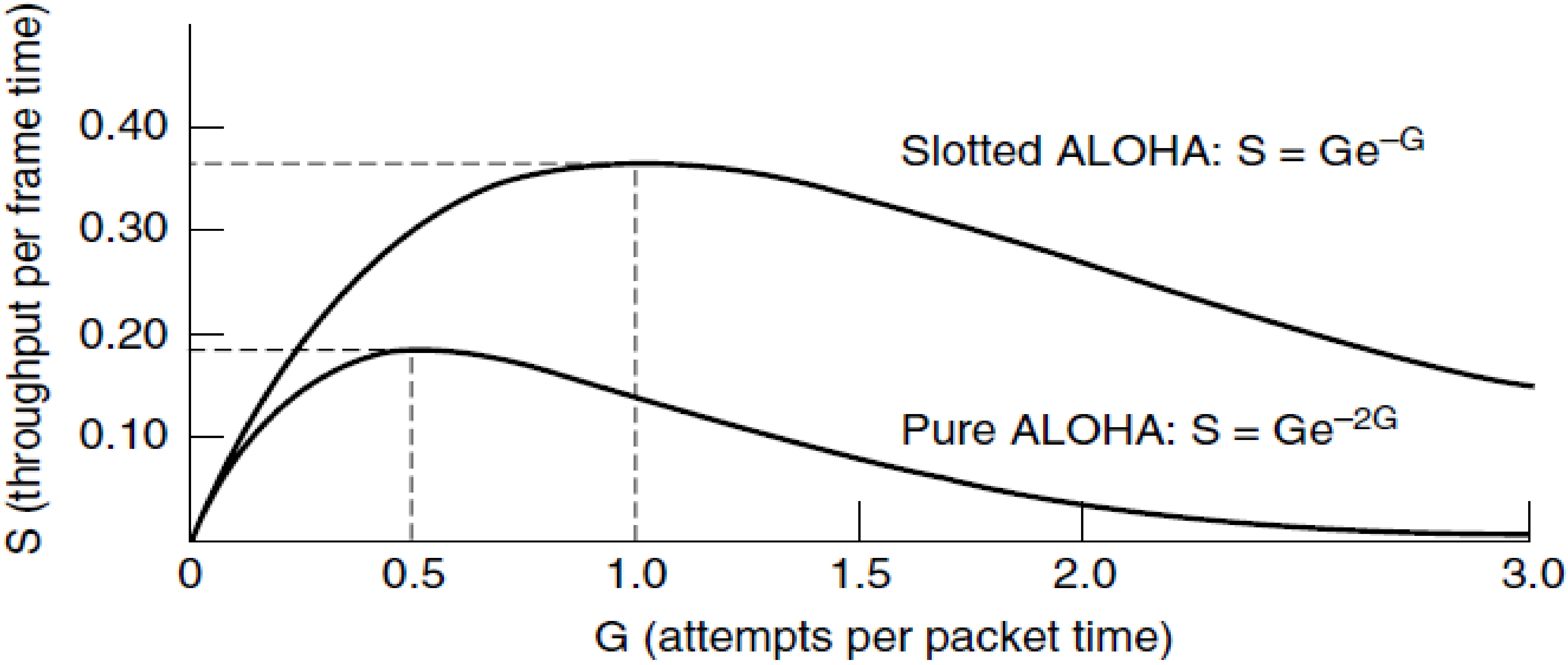


Figure 4-3. Throughput versus offered traffic for ALOHA systems.

Carrier Sense Multiple Access (CSMA) Protocols

- CSMA was developed to overcome the problems of ALOHA i.e. to minimize the chances of collision.
- Protocols in which stations listen for a carrier (i.e., a transmission) and act accordingly are called **carrier sense protocols**.
- Based on the principle "sense before transmit" or "listen before talk."
- Node verifies the absence of other traffic before transmitting on a shared transmission medium.
- Multiple access means that multiple stations send and receive on the medium
- The chances of collision still exists because of propagation delay.
- There are three different types of CSMA protocols:
  - 1-Persistent CSMA
  - Non-Persistent CSMA
  - P-Persistent CSMA

1-Persistent CSMA

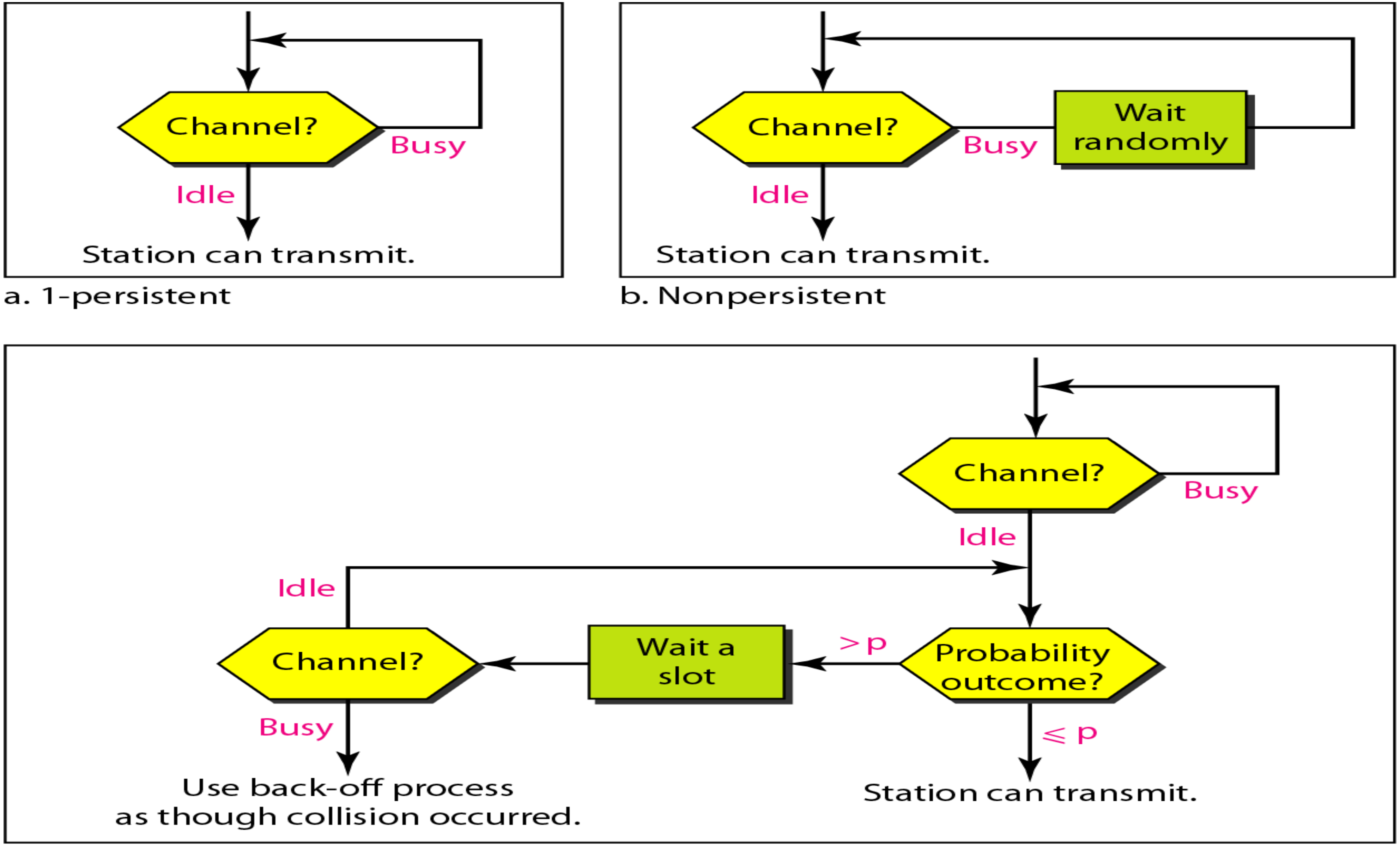
- When a station has **data to send**, it **first listens** to the channel to see if anyone else is transmitting at that moment.
- If the **channel is idle**, the stations **sends** its data. **Otherwise**, if the channel is busy, the **station just waits** until it becomes idle. Then the station transmits a frame.
- If a collision occurs, the station **waits a random amount of time** and starts all over again.
- The protocol is **called 1-persistent** because the station transmits with a **probability of 1** when it finds the channel idle.
- This method has the **highest chance of collision** because **two or more stations** may **find channel to be idle** at the **same time** and **transmit** their frames.
- The propagation delay has an important effect on the performance of the protocol

Non-Persistent CSMA

- In this protocol, a conscious attempt is made to be less greedy than in the previous one.
- **Before sending**, a station **senses the channel**. If **no one** else is **sending**, the station **begins doing so itself**.
- However, if the channel is **already in use**, the station **does not continually sense** it. **Instead**, it **waits a random period** of time and **then repeats the algorithm**.
- Consequently, this **algorithm leads to better channel utilization** but longer delays than 1-persistent CSMA.
- It **reduces the chance of collision** because the stations **wait for a random amount** of time .
- It is **unlikely that two or more stations** will **wait for the same amount** of time and **will retransmit at the same time.**

P-Persistent CSMA

- It applies to slotted channels and **works as follows**:
  - When a station becomes **ready to send**, it **senses the channel**.
  - If it is **idle**, it **transmits with a probability $\boldsymbol{p}$**. With a **probability $\boldsymbol{q = 1 - p}$**, it **defers** until the **next slot**.
  - If that slot is **also idle**, it either **transmits or defers again**, with **probabilities $\boldsymbol{p}$ and $\boldsymbol{q}$.**
  - This process is **repeated until either the frame has been transmitted** or another station **has begun transmitting**.



a. 1-persistent

b. Nonpersistent

c. p-persistent

Flow diagram for three persistence methods

CSMA/CD

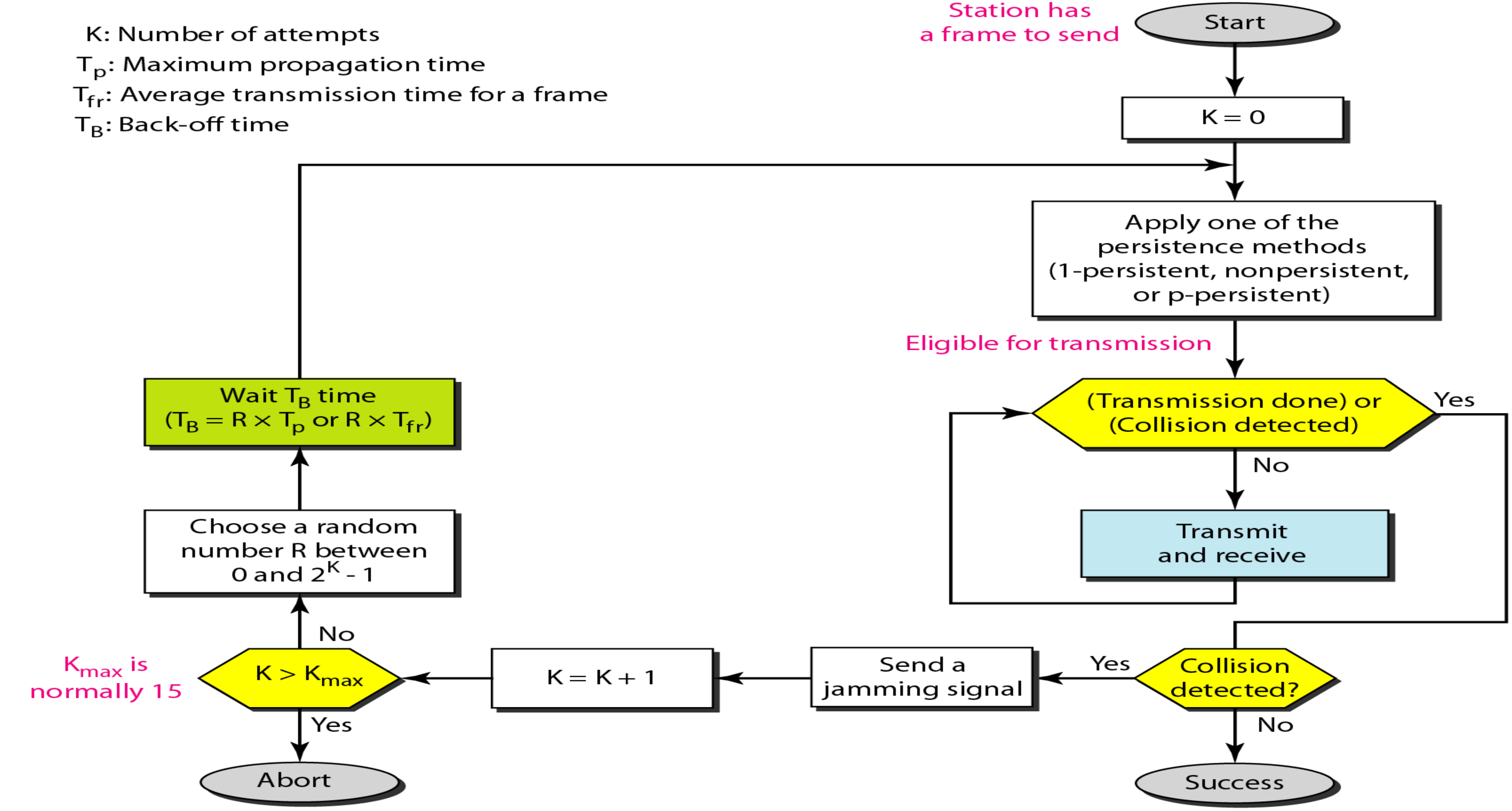
- CSMA with **Collision Detection**.
- In this protocol, the **station senses** the channel **before transmitting** the frame. If the channel is **busy**, the station **waits**.
- **Additional feature** in CSMA/CD is that the stations can **detect collisions**. The stations **abort** their transmission **as soon as** they **detect collision**.
- In CSMA/CD, the station that **sends its data on the channel**, **continues to sense** the channel **even after data transmission**.
- If collision **is detected**, the station **aborts its transmission** and **waits** for a **random amount of time** & sends its data again.
- **As soon as** a collision **is detected**, the transmitting station **release a jam signal**.
- **Jamming signal** is a signal that carries a **32-bit binary pattern** sent by a data station **to inform the other stations** of the **collision** and that they **must not transmit.**

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 > Kmax
Kmax is normally 15
No
Yes
Abort
Choose a random number R between 0 and 2^K - 1
Wait TB time (TB = R × Tp or R × Tfr)

Flow diagram for the CSMA/CD

IEEE LAN Standards

- **IEEE 802.3 Ethernet (CSMA/CD)**
- **IEEE 802.4 Token Bus**
- **IEEE 802.5 Token Ring**
- IEEE 802.6 Metropolitan Area Networks
- IEEE 802.7 Broadband LANs
- IEEE 802.8 Fiber Optic LANs
- IEEE 802.9 Integrated Data and Voice Networks
- **IEEE 802.11 Wireless Networks**
- IEEE 802.14 Cable TV

IEEE 802.3 Ethernet (CSMA/CD)

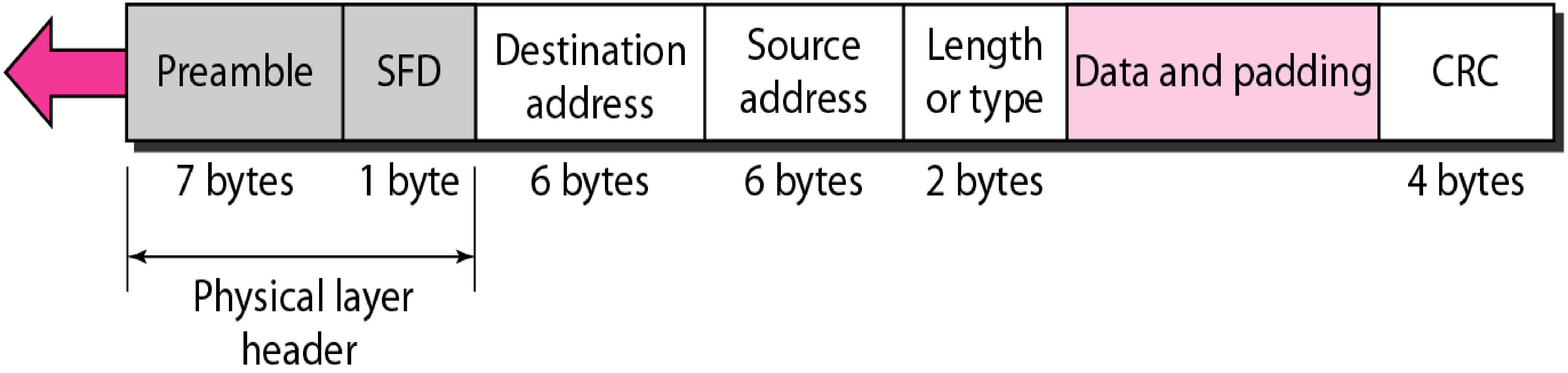
- The IEEE standard for Ethernet is 802.3.
- Ethernet operates in two areas of the OSI model
  - the lower half of the data link layer, which is known as the MAC sub layer,
  - and the physical layer.
- The CSMA/CD is the access method used in Ethernet to detect and avoid collision in network.
- Ethernet has been a relatively inexpensive, reasonably fast, and very popular LAN technology for several decades.
- The most commonly installed Ethernet systems are called 10BASE-T and provide transmission speeds up to 10 Mbps.
- Fast Ethernet or 100BASE-T provides transmission speeds up to 100 megabits per second.

IEEE 802.3 Ethernet Frame Format

- **The Preamble:** This consists of seven bytes, all of the form "10101010". This allows the receiver's clock to be synchronized with the sender's.
- **The Start Frame Delimiter (SFD):** This is a single byte ("10101011") which is used to indicate the start of a 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

IEEE 802.3 Ethernet Frame Format

- **The Destination Address (DA):** The DA field is 6 bytes and contains the physical address of the destination station or stations to receive the packet. The addresses in 802.3 use globally unique hardwired 48 bit addresses.
- **The Source Address (SA):** The SA field is also 6 bytes and contains the physical address of the sender of the packet.
- **The Length:** This is the length of the data in the Ethernet frame, which can be anything from 0 to 1500 bytes.
- **Data :** This field carries data encapsulated from the upper-layer protocols. It is a minimum of 46 and a maximum of 1500 bytes
- **Checksum:** This is used for error detection and recovery.

IEEE 802.3 Ethernet (CSMA/CD)

- The 802.3 standard describes the operation of the MAC sub-layer in a bus LAN that uses carrier sense multiple access with collision detection (CSMA/CD).
    - Beside carrier sensing, collision detection and the binary exponential back-off algorithm, the standard also describes the format of the frames and the type of encoding used for transmitting frames.
    - The minimum length of frames can be varied from network to network.
    - The standard also makes some suggestions about the type of cabling that should be used for CSMA/CD bus LANs.
- The 802.3 CSMA/CD bus LAN is said to be a non-deterministic network. This means that no host is guaranteed to be able to send its frame within a reasonable time.
    - When the network is busy, the number of collisions rises dramatically and it may become very difficult for any hosts to transmit their frames.

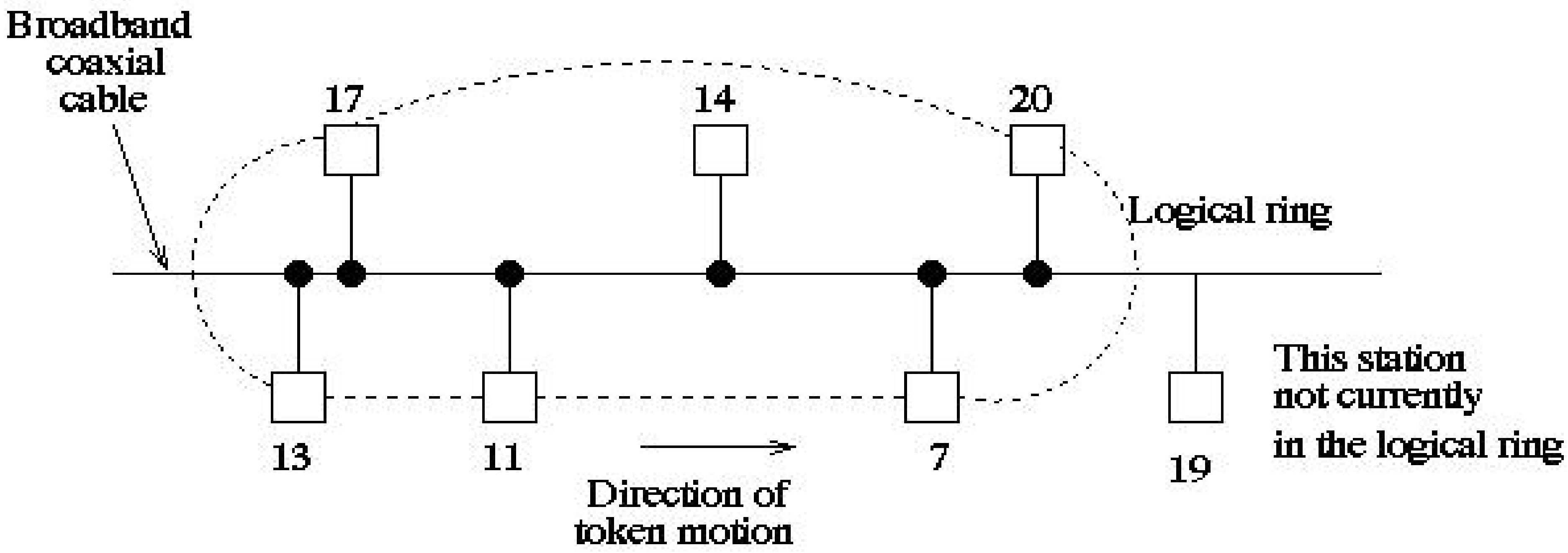
IEEE 802.4 Token Bus

- **Evolution of 802.4**
  - **802.3 suffer** from the **difficulty of large delay** in getting the access and at the same time
  - **Poor performance** under heavy load.
  - There are also **no priorities in 802.3**, making them unsuited for real time systems.
- Token passing protocols were proposed and were found to be very attractive for situations with heavy load.
- The basic **idea is to generate a token** in the network. Only the **holder** of the **token can transmit**. Thus with one token, only one station can transmit at a time, **eliminating collisions** totally.
- **Logically**, all stations are organized **into a ring**. Stations are **inserted into ring** in order of station address, **from highest to lowest. Token passing** is also done **from high to low** addresses.

IEEE 802.4 Token Bus

- **Basic Operation:**
  - **Token** is a **special Frame** which gives the holder station the **“Right to Transmit”.**
  - All stations are **connected to a linear cable** but organized **in a Logical ring**.
  - Frames are **passed** from the **Predecessor to the successor** after a specified time interval.
  - When there is **no data to be sent** the token **circulates** around the logical ring.
  - Whenever a station **has data to send**, it **waits for a token to arrive**.
  - **Station then captures** the token and keeps **transmitting data** until allocated time for keeping the token expires.
  - **After** the **specified time** the token must be **passed on to the successor**.
- **Four priority classes:** (0, 2, 4, 6) for traffic, with **0 the lowest and 6 the highest**.
- When the token comes into the station, **it passes to priority 6 substation**, which may begin transmitting frames, if it has any. When it is done, (or when its timer expires), the **token is passed to the priority 4 substations**, *etc.*

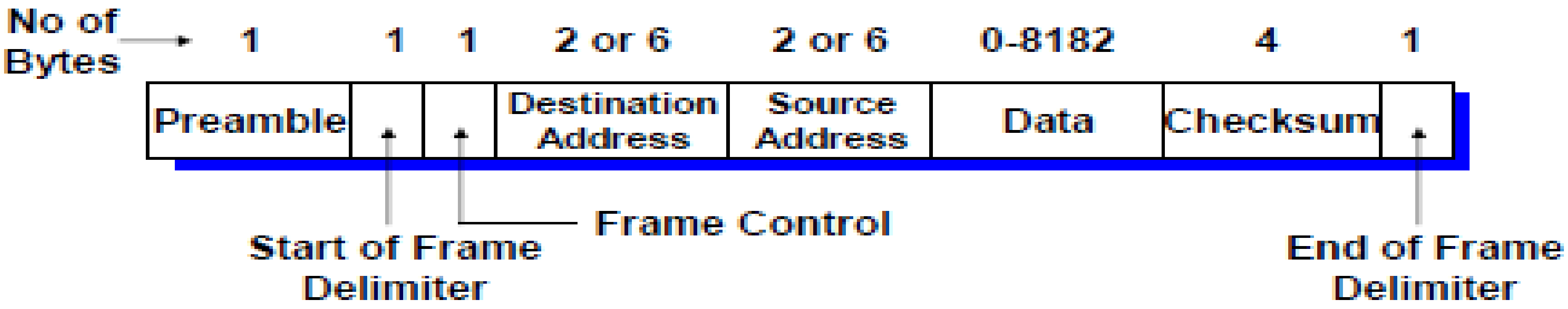
IEEE 802.4 Token Bus


Broadband coaxial cable
17
14
20
Logical ring
13
11
7
19
This station not currently in the logical ring
Direction of token motion

A token bus

IEEE 802.4 Token Bus Frame Format

- **Preamble:** The preamble is used to synchronize the receiver’s clock.
- The **start and end delimiter** fields are used to mark the frame boundaries.
- The **frame control** field is used to distinguish **data frames from control frames**.
  - For data **frames**, it carries the frame’s priority. The token bus defines four priority classes-0, 2, 4 and 6 for traffic.
  - For **the control frame**, the frame control field is used to specify the frame type. The allowed types **include token passing and various ring maintenance** frames such as mechanism for letting new stations enter the ring or the mechanism for allowing stations to leave the ring.



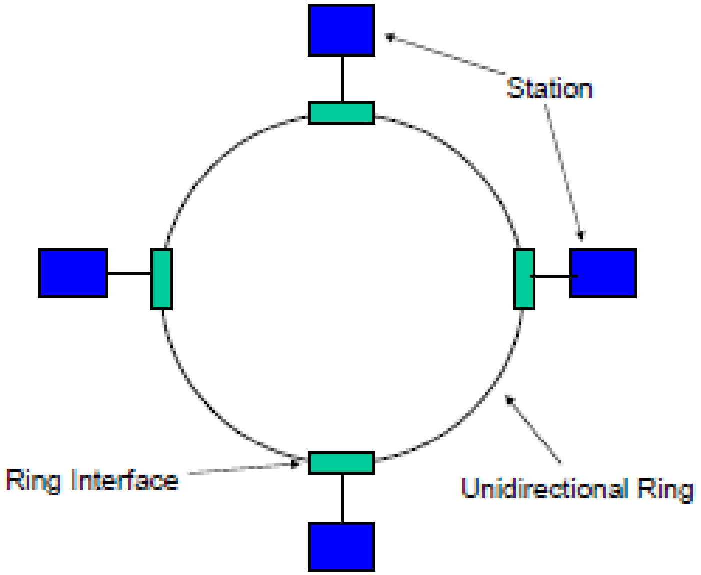
No of Bytes →	1	1	1	2 or 6	2 or 6	0-8182	4	1
	Preamble			Destination Address	Source Address	Data	Checksum	

IEEE 802.5 Token Ring

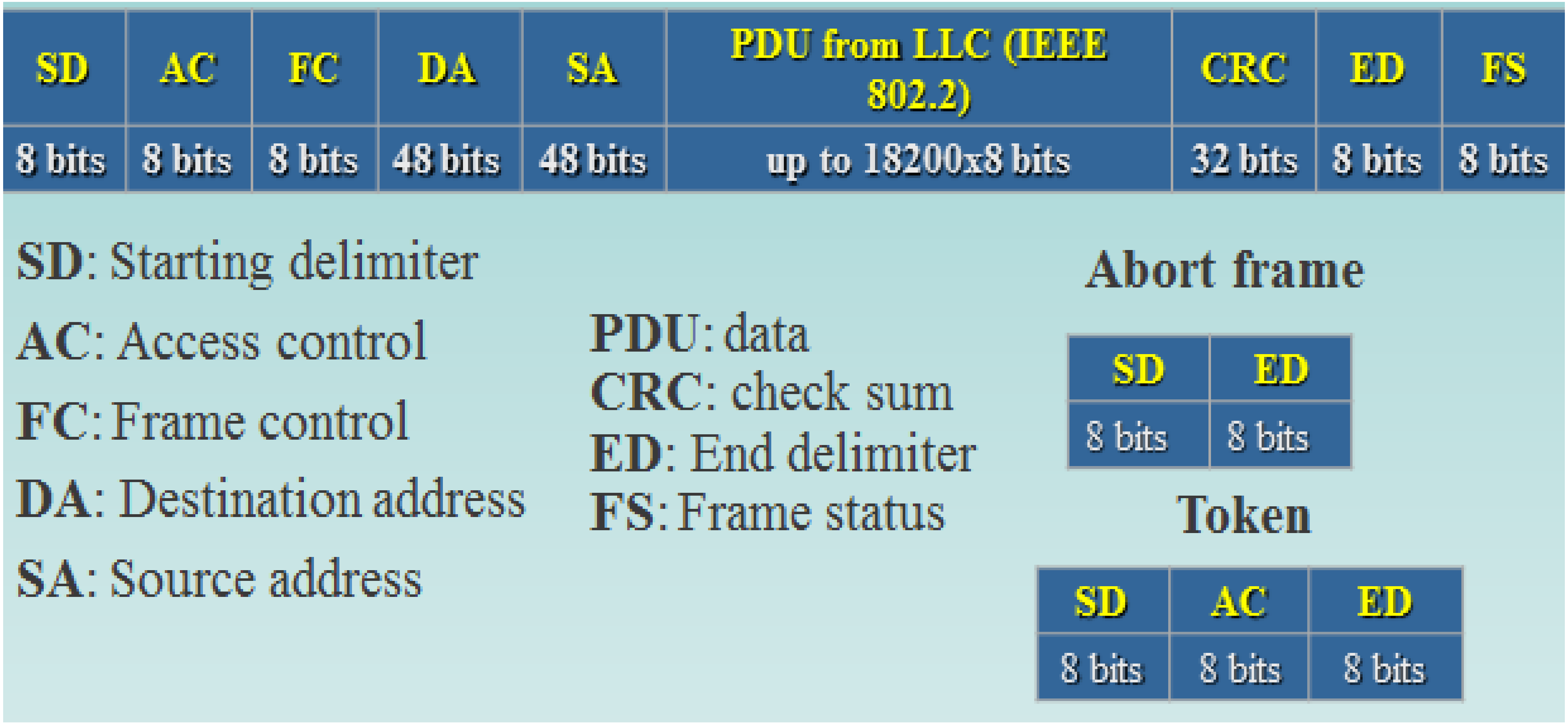
- Ring is **not a broadcast medium** but a collection of **point-to-point links** forming a circle.
- Rings can be based **on twisted pair, coaxial or a fiber optics** cable.
- **Channel access problem** is solved with the help of a special frame called a **"Token".**
- **Possession** of the token grants the **right to transmit**. If a node receiving the token has **no information to send**, it **passes** the token to the **next end station**.
- A **free token circulates** the ring when all stations are idle.
- Each station **can hold** the token for a **maximum period of time**.
- A station **wishing to transmit** must wait **until it detects a free token** passing by.
- **Bits of the frame** that have traversed the ring **must be removed from the ring by the sender.**

IEEE 802.5 Token Ring

- Since the **entire frame does not appear** on the ring at one time, there is **no limit on frame size**. It only needs to **be pre-decided**. The **only limit** is the **token holding time**.
- **Acknowledgements** are sent **by the receiver** in the **same received frame**, by **setting an Acknowledgement bit** in the received frame.



IEEE 802.5 Token Ring Frame Format



SD	AC	FC	DA	SA	PDU from LLC (IEEE 802.2)	CRC	ED	FS
8 bits	8 bits	8 bits	48 bits	48 bits	up to 18200x8 bits	32 bits	8 bits	8 bits

SD: Starting delimiter

AC: Access control

FC: Frame control

DA: Destination address

SA: Source address

PDU: data
CRC: check sum
ED: End delimiter
FS: Frame status

Abort frame

SD	ED
8 bits	8 bits

Token

SD	AC	ED
8 bits	8 bits	8 bits

IEEE 802.5 Token Ring Frame Format

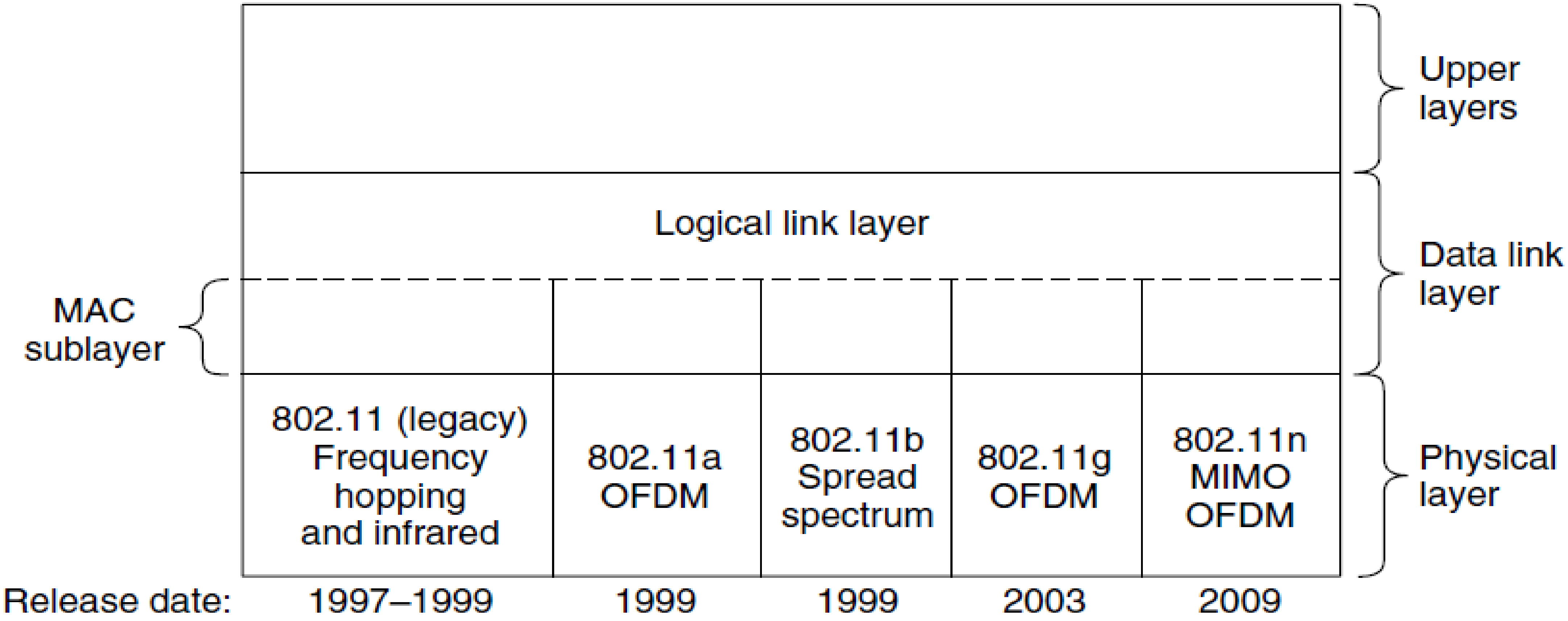
- The **starting and ending delimiter** fields mark the beginning and ending of the frame.
- **Access-control byte**—Contains the Priority field (the most significant 3 bits) and the Reservation field (the least significant 3 bits), as well as a token bit and a monitor bit.
- **Frame-control bytes**—Indicates whether the frame contains data or control information. In control frames, this byte specifies the type of control information.
- The **frame status** byte contains A and C bits.

A	**C**	**Significance**
0	0	Destination not present or not powered up
1	0	Destination present but frame not accepted
1	1	Destination present and frame copied.

IEEE 802.11 Physical Layer Options

	Frequency Band	Bit Rate	Modulation Scheme
802.11	2.4 GHz	1-2 Mbps	Frequency-Hopping Spread Spectrum, Direct Sequence Spread Spectrum
802.11b	2.4 GHz	11 Mbps	Complementary Code Keying & QPSK
802.11g	2.4 GHz	54 Mbps	Orthogonal Frequency Division Multiplexing (OFDM) & CCK for backward compatibility with 802.11b
802.11a	5-6 GHz	54 Mbps	Orthogonal Frequency Division Multiplexing
802.11n	2.4GHz 5-6GHz	600+Mbps	OFDM, Multiple input multiple output (MIMO)

IEEE 802.11 Wireless LAN



Upper layers

Logical link layer

Data link layer

MAC sublayer

802.11 (legacy) Frequency hopping and infrared

802.11a OFDM

802.11b Spread spectrum

802.11g OFDM

802.11n MIMO OFDM

Physical layer

Release date: 1997–1999 1999 1999 2003 2009

Figure 4-24. Part of the 802.11 protocol stack.

IEEE 802.11 Wireless LAN

- Physical Layer
  - ***Infrared***
    - Uses Diffused infrared (ie. Not line of sight)
    - 1 Mbps and 2 Mbps Speeds
    - Not popular option
  - ***FHSS(Frequency Hopping Spread Spectrum)***
    - 79 channels of 1 MHz each starting at low end of 2.4 GHz
    - Pseudo random number generator produce sequence of frequencies
    - Each station stays in one freq. for fixed time (called dwell time)
    - As long as all stations use same seed, they stay tuned to each other.
    - Secured due to freq. hopping and dwell time
    - But bandwidth is low

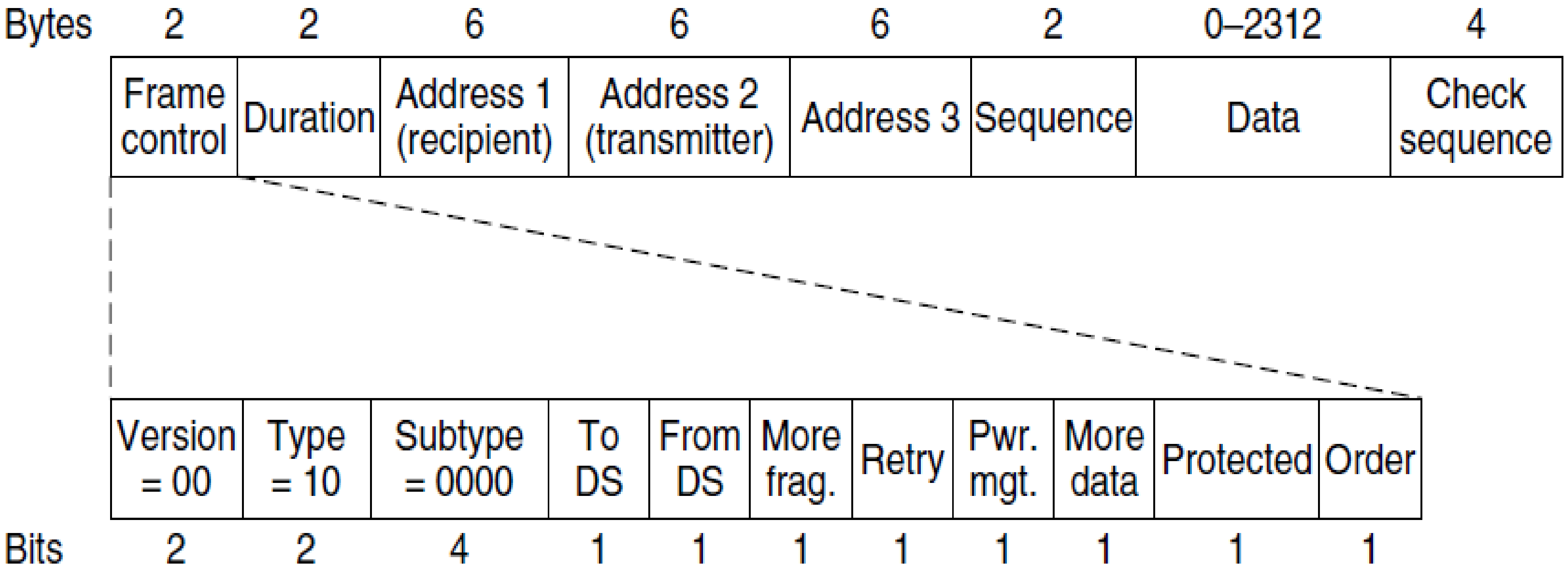
IEEE 802.11 Wireless LAN

- ***DSSS(Direct Sequence Spread Spectrum)***
    - Restricted to 1 or 2 Mbps as FHSS
    - Some similarity with CDMA system
    - Each bit transmitted as 11 chips (called **Baker Sequence**)
    - Uses phase shift modulation at 1Mbaud
        - 1 bit per baud => 1 Mbps
        - 2 bit per baud => 2 Mbps
- ***OFDM (Orthogonal FDM) used in IEEE 802.11a***
- ***HR-DSSS(High Rate DSSS) used in IEEE 802.11b***

IEEE 802.11 Wireless LAN

- MAC Sub-Layer
  - 802.11 supports two modes of operation
    - ***DCF(Distributed Coordination Function)***
      - No central control (as in Ethernet)
      - Ad-hoc wireless LAN
      - Normal CSMA/CA or CSMA/CA based on MACAW is used
    - ***PCF(Point Coordination Function)***
      - Base station to control all activities
      - Communicate via Central Station e.g. Using Access Points
      - Central station periodically broadcast Beacon frame containing Hopping sequence and dwell times(for FHSS), Clock Sync. Also invites new station to sign up polling service

IEEE 802.11 Wireless LAN Frame Format



Bytes	2	2	6	6	6	2	0–2312	4
	Frame control	Duration	Address 1 (recipient)	Address 2 (transmitter)	Address 3	Sequence	Data	Check sequence

	Version = 00	Type = 10	Subtype = 0000	To DS	From DS	More frag.	Retry	Pwr. mgt.	More data	Protected	Order
Bits	2	2	4	1	1	1	1	1	1	1	1

Figure 4-29. Format of the 802.11 data frame.

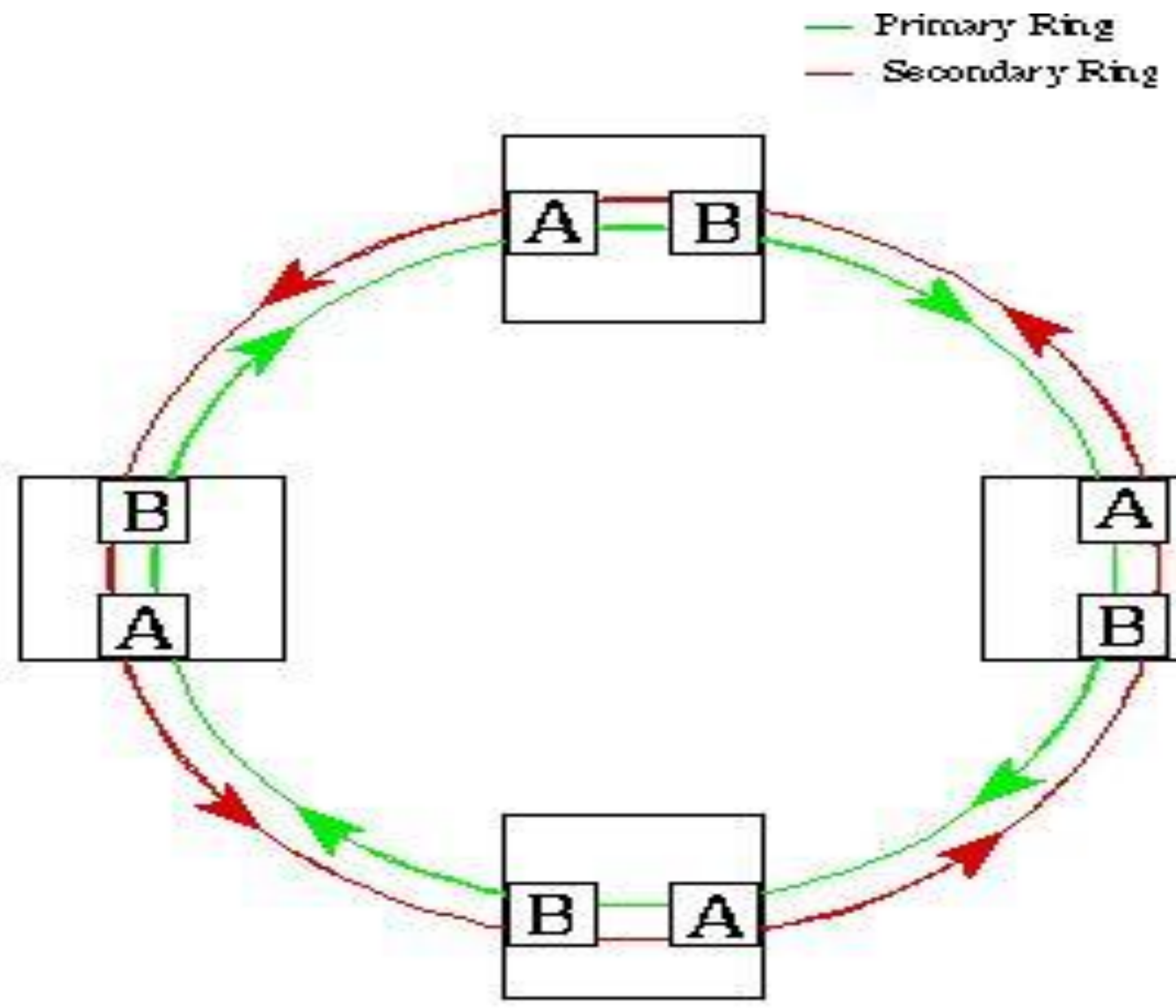
IEEE 802.11 Wireless LAN Frame Format

- Frame Control: Contains following
  - **Version:** Protocol version
  - **Type :** data, control or mgmt.
  - **Subtype** : Request To Send (RTS) or Clear To Send (CTS)
  - **To/From DS:** Going to or Coming from inter-cell distribution (e.g.. Ethernet)
  - **MF:** More fragments to follow
  - **Retry:** Retransmission of earlier frame
  - **Power Mgmt.:** used by base station to sleep or wake receiver
  - **More Data:** sender has more frames for receiver
  - **W:** WEP Encryption
  - **O:** sequence of frames must be processed in order
  - **Duration:** time to occupy channel, used by other stations to manage NAV
  - **Addresses:** Two are source and destination Address of sender and receiver, other two are that of base stations for inter-cell traffic.

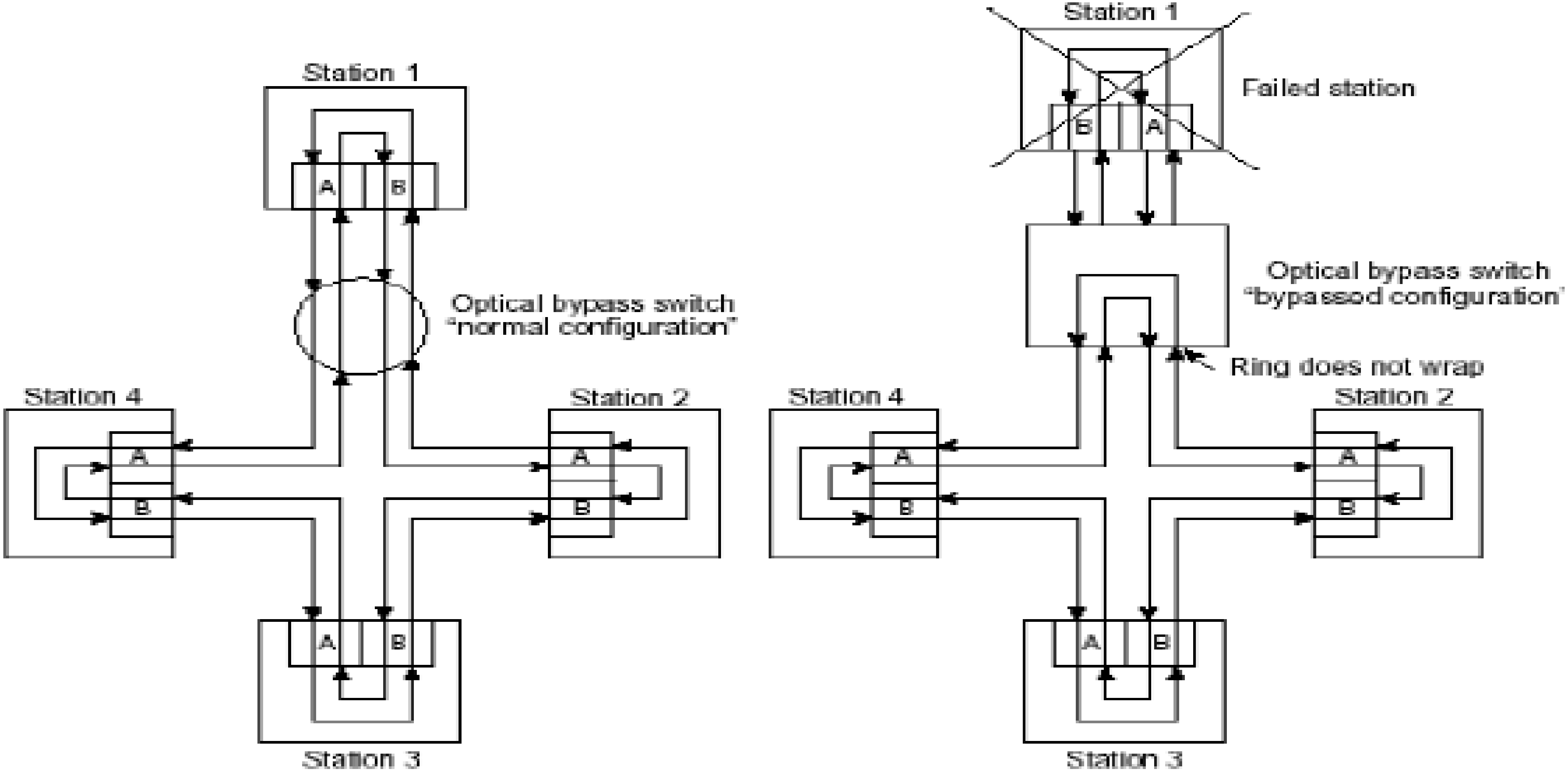
Fiber Distributed Data Interface(FDDI)

- *FDDI* specifies a **100-Mbps token-passing**, **dual-ring** LAN using **fiber-optic cable**.
- FDDI is frequently used as high-speed backbone technology because of its support for **high bandwidth and greater distances** than copper.
- FDDI supports extensions **up to 100 Km**.
- A total of **1000 stations** can be connected with a **maximum separation of 2 km**.
- **Node-to-node distance** : **2 km** using multi mode & **40 km** using single mode fiber.
- FDDI uses **dual counter-rotating rings** (called the primary and secondary). Data normally travels on the primary ring. Stations can be attached to the primary ring as **single attachment stations** (SAS) or both rings as **dual attachment stations** (DAS).
- An **important feature** of FDDI is its **ability to handle a breaks** in the network by forming a single temporary ring out of the pieces of the primary and secondary rings.
- Once the stations **detect the break**, **traffic is rerouted through a new ring** formed out of the parts of the primary and secondary rings not affected by the break.

FDDI's dual-ring environment



Implementation of optical bypass switch



Fiber Distributed Data Interface(FDDI)

Media Access Control

- FDDI **uses a token passing** system. Computers wanting to send packets **wait to receive a token** before transmitting.
- **Multiple packets** can be **attached** to the token as it **moves around the network**.
- When a station **receives the token**, it **looks for attached packets addressed** to it and removes them from the incoming packet.
- If the station **wants to send a packet** it attaches **it to the token** and sends the token with its attached packets to the **next station**.
- This controlled access technique **provides a higher performance** level at high traffic levels compared to a contention-based technique like Ethernet.

FDDI: Frame Structure

Token Frame Format

PRE	SD	FC	ED

Data Frame Format

8	1	1	2 or 6	2 or 6		4	1	1
PRE	SD	FC	Destination Address	Source Address	Information	FCS	ED	FS

Preamble

Frame Control CLFFZZZZ

C = Synch/Asynch
L = Address length (16 or 48 bits)
FF = LLC/MAC control/reserved frame type

FDDI: Frame Layout

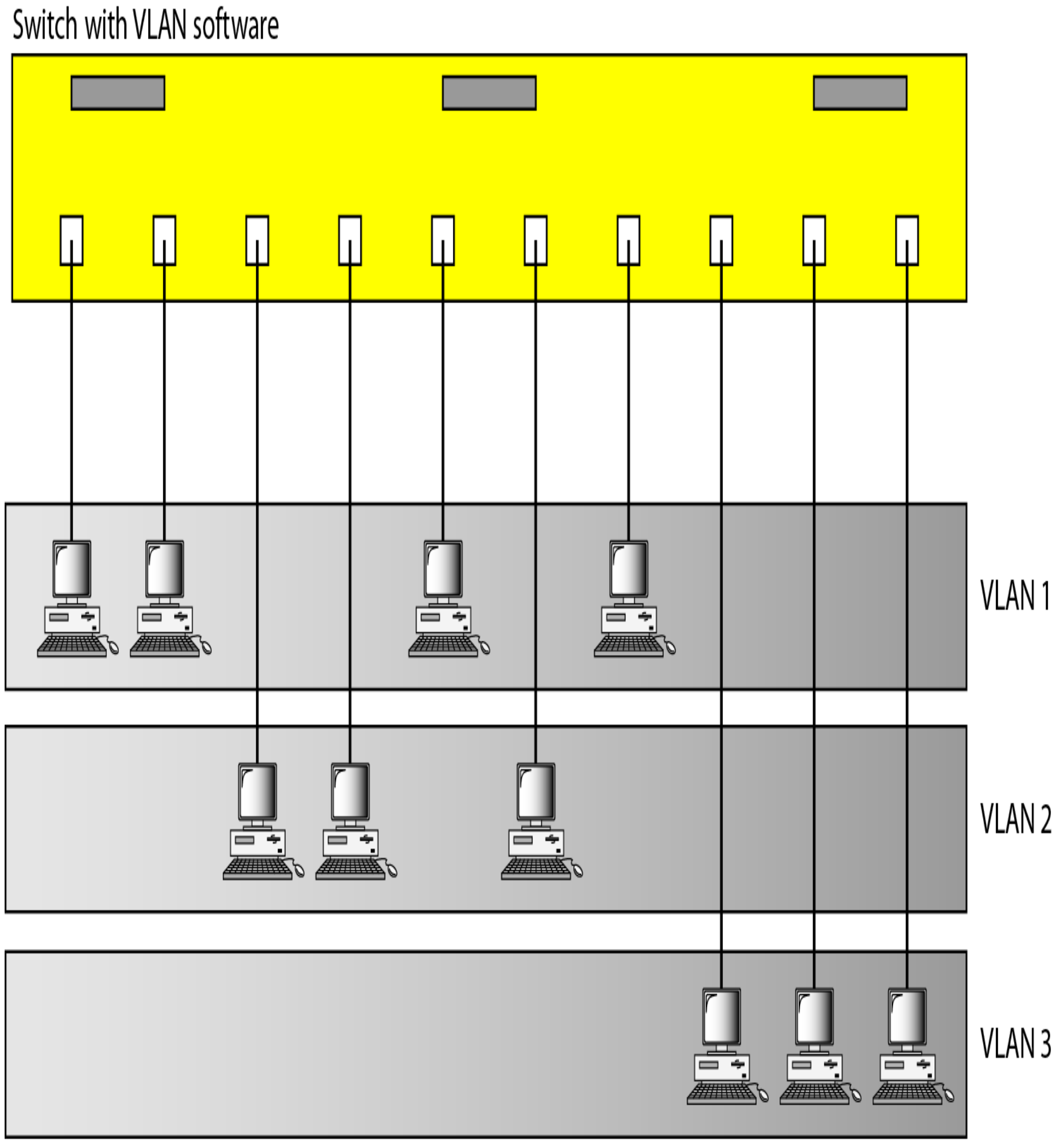
- FDDI Frame can be as long as 4500bytes.
  - **Preamble:** Unique sequence that prepares each station for an upcoming frame.
  - **Frame Control:** Indicates size of address field and whether the frame contains synchronous or asynchronous data, among other control information
  - **Destination Address:** Contains a unicast, multicast or broadcast address.
  - **Source Address:** 6 byte address source Address.
  - **Data:** Contains either information destined for upper layers or control information.
  - **Frame Check Sequence:** For Error detection.
  - **End Delimiter:** End of Frame.
  - **Frame status:** Allows the source station to determine whether an error occurred; identifies whether the frame was recognized and copied by a receiving station.

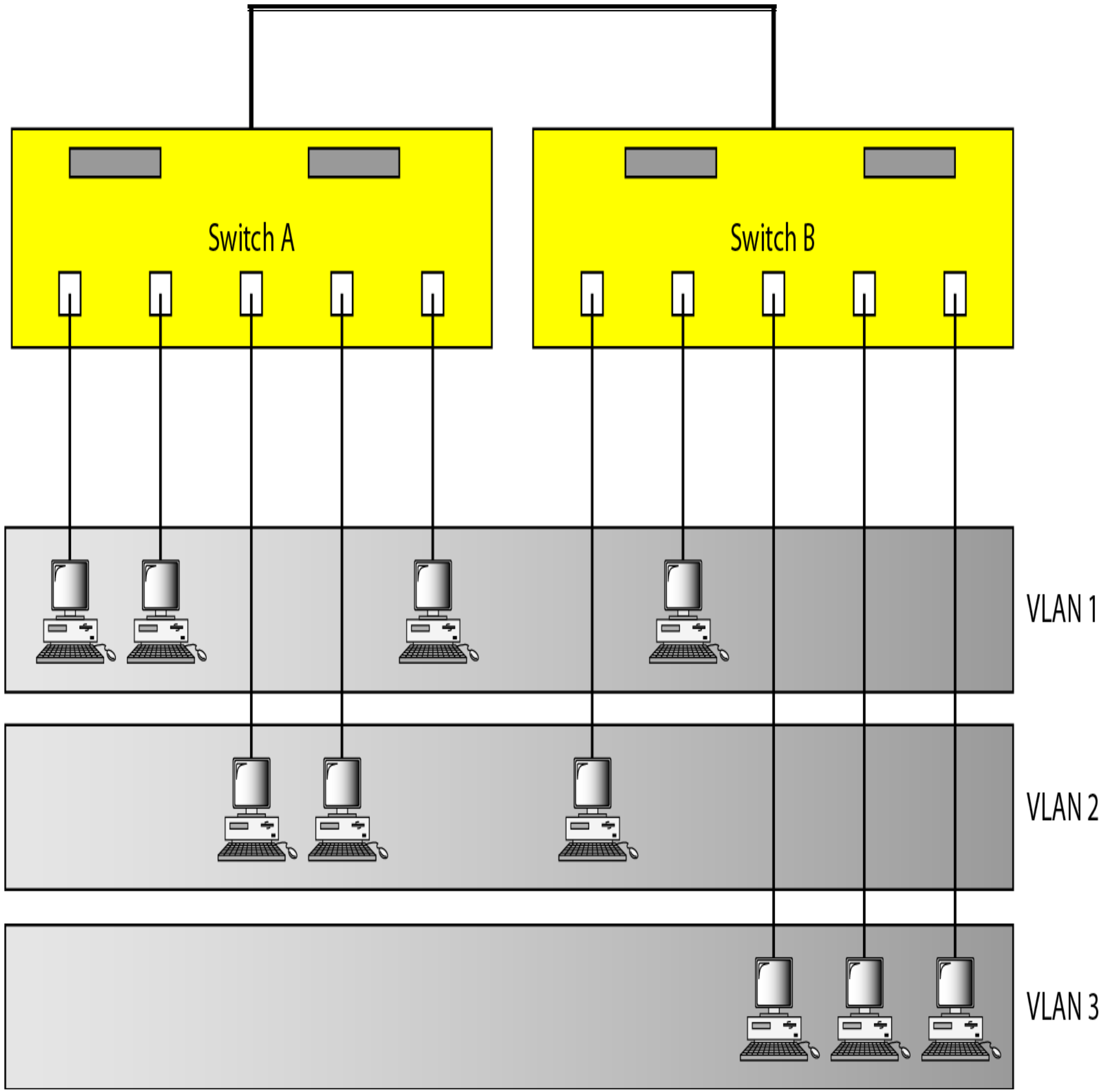
VLAN

- VLANs are a new type of LAN architecture using intelligent, high-speed switches. VLANs assign computers to LAN segments by software.
- VLANs have been standardized as IEEE 802.1q and IEEE 802.1p.
- **To set up a VLAN-based network**, *the network administrator decides* ***how many VLANs*** *there will be*, ***which computers will be on which VLAN***, *and what the* ***VLANs will be called***.
- VLAN's offer a number of advantages over traditional LAN's:
  - **Performance:** For example, in a broadcast domain consisting of 10 users, if the broadcast traffic is intended only for 5 of the users, then placing those 5 users on a separate VLAN can reduce traffic.
  - **Formation of Virtual Workgroups**
  - **Simplified Administration**
  - **Security:** VLAN's can be used to control broadcast domains, set up firewalls, restrict access, and inform the network manager of an intrusion.

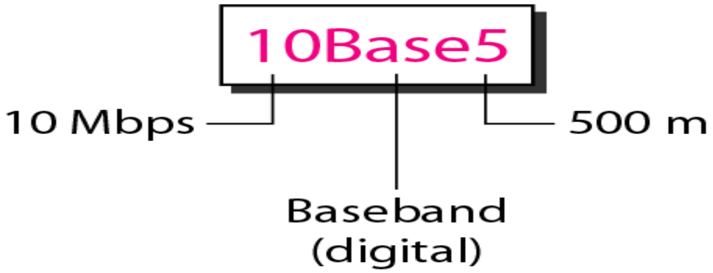
VLAN

- Computers can be assigned to **VLANs in four ways**:
  - ***Port-based VLANs*: It** can be defined based on the ports that belong to the VLAN. For example, in a bridge with four ports, ports 1, 2, and 4 belong to VLAN 1 and port 3 belongs to VLAN 2.
  - ***MAC-based VLANs:*** assign computers according to each computer's data link layer address.
  - ***IP-based VLANs:*** assign computers using their IP-address
  - ***Application-based VLANs:*** assign computers depending on the application that the computer typically uses. For example, file transfer protocol (FTP) applications can be executed on one VLAN and telnet applications on another VLAN.


Switch with VLAN software
VLAN 1
VLAN 2
VLAN 3


Switch A
Switch B
VLAN 1
VLAN 2
VLAN 3

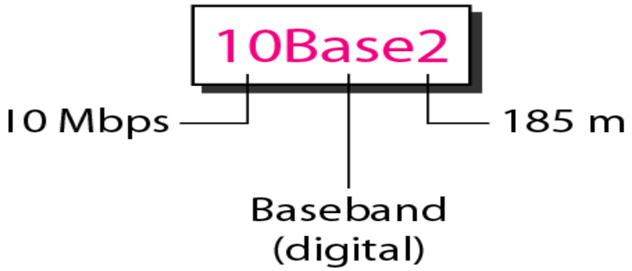
Ethernet Cabling


10Base5

10 Mbps

500 m

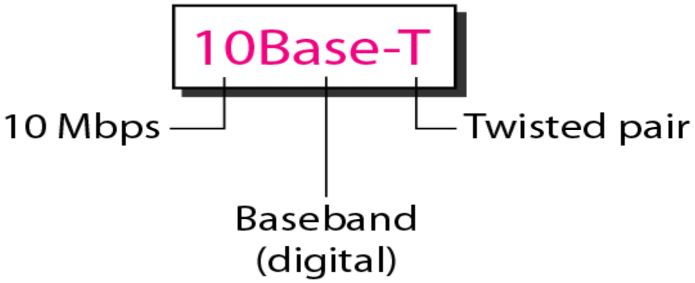
Baseband (digital)


10Base2

10 Mbps

185 m

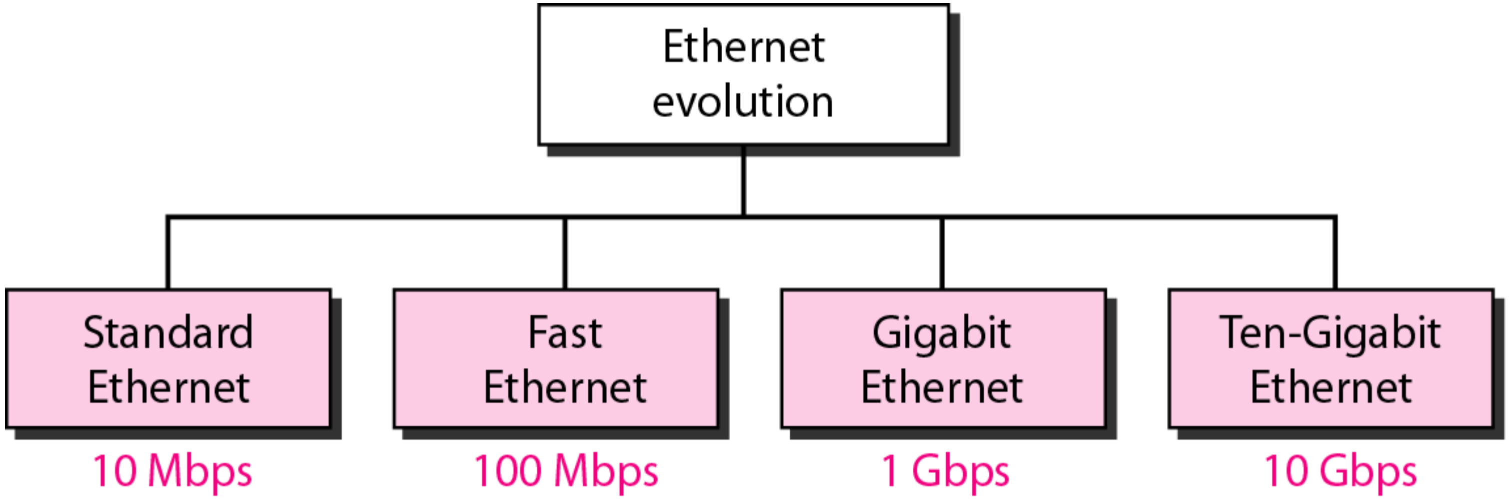
Baseband (digital)


10Base-T

10 Mbps

Twisted pair

Baseband (digital)

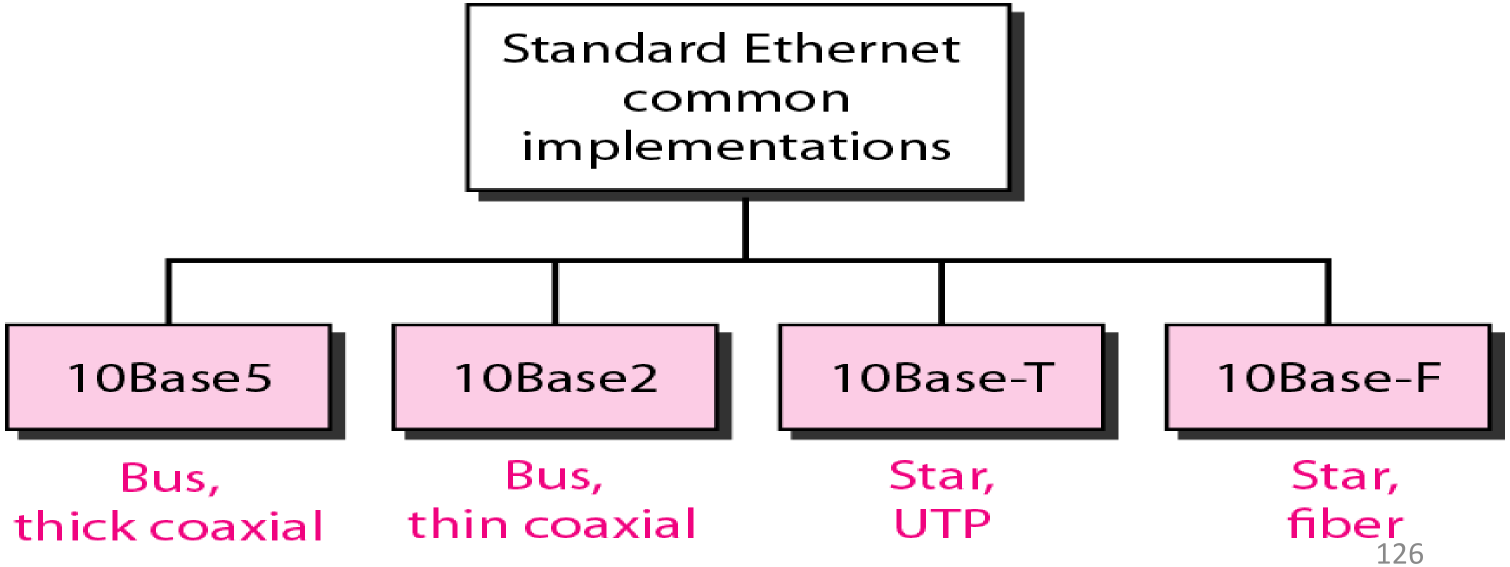

Ethernet evolution

Standard Ethernet
10 Mbps

Fast Ethernet
100 Mbps

Gigabit Ethernet
1 Gbps

Ten-Gigabit Ethernet
10 Gbps


Standard Ethernet common implementations

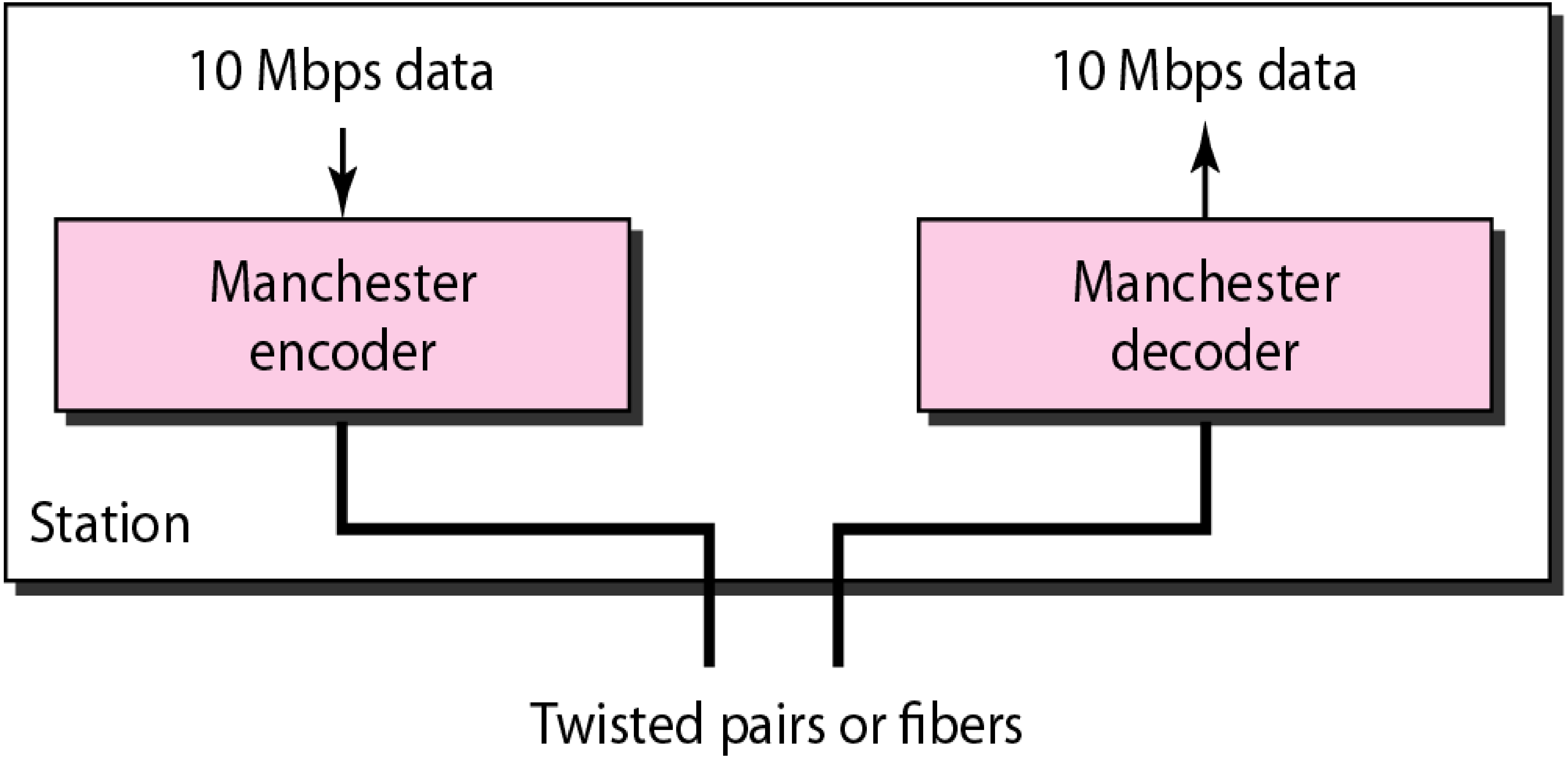
10Base5
Bus, thick coaxial

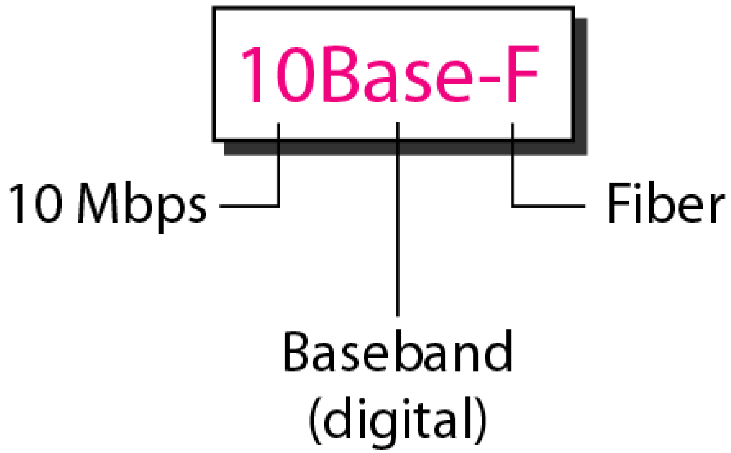
10Base2
Bus, thin coaxial

10Base-T
Star, UTP

10Base-F
Star, fiber

Encoding in a Standard Ethernet implementation



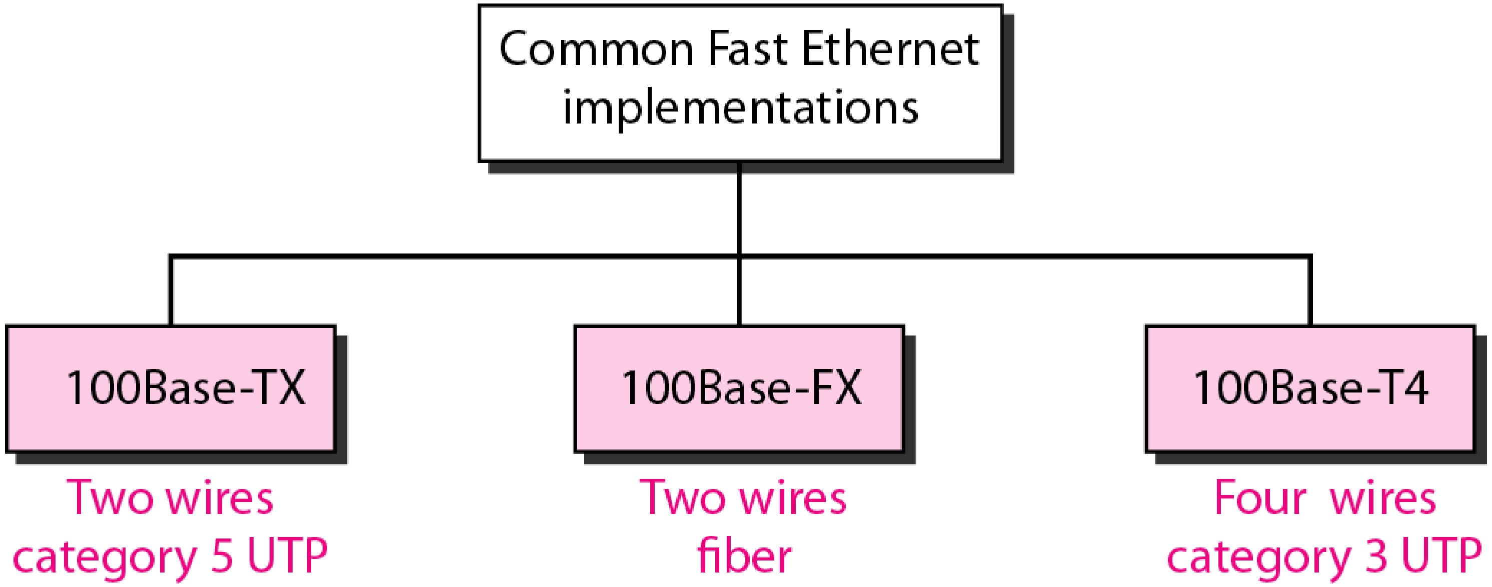


10Base-F

10 Mbps

Baseband (digital)

Fiber



Common Fast Ethernet implementations

100Base-TX

Two wires category 5 UTP

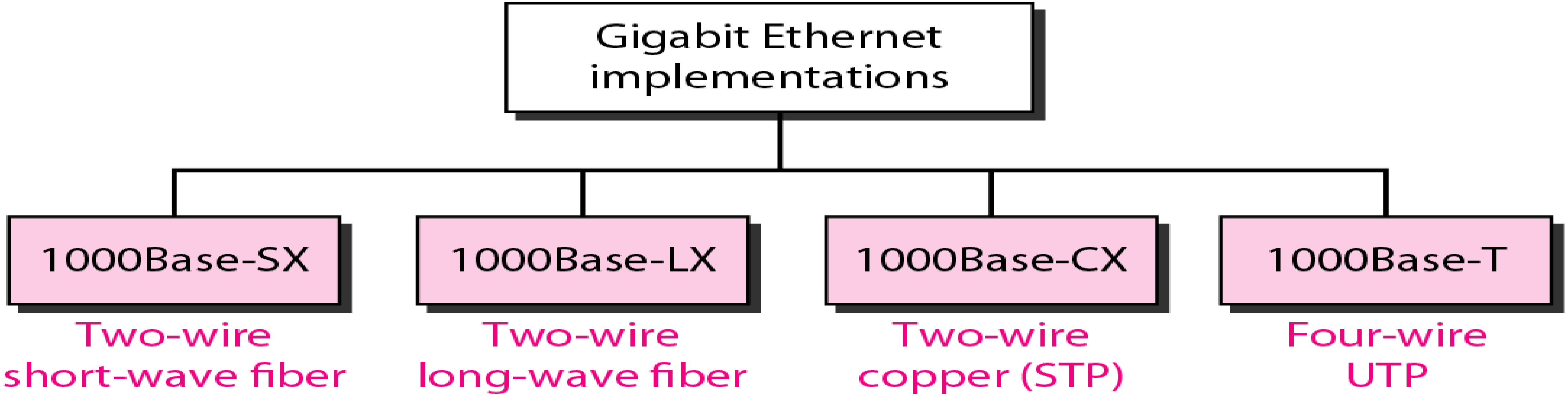
100Base-FX

Two wires fiber

100Base-T4

Four wires category 3 UTP

Characteristics	*100Base-TX*	*100Base-FX*	*100Base-T4*
Media	Cat 5 UTP or STP	Fiber	Cat 4 UTP
Number of wires	2	2	4
Maximum length	100 m	100 m	100 m
Block encoding	4B/5B	4B/5B	
Line encoding	MLT-3	NRZ-I	8B/6T



Gigabit Ethernet implementations

1000Base-SX

Two-wire short-wave fiber

1000Base-LX

Two-wire long-wave fiber

1000Base-CX

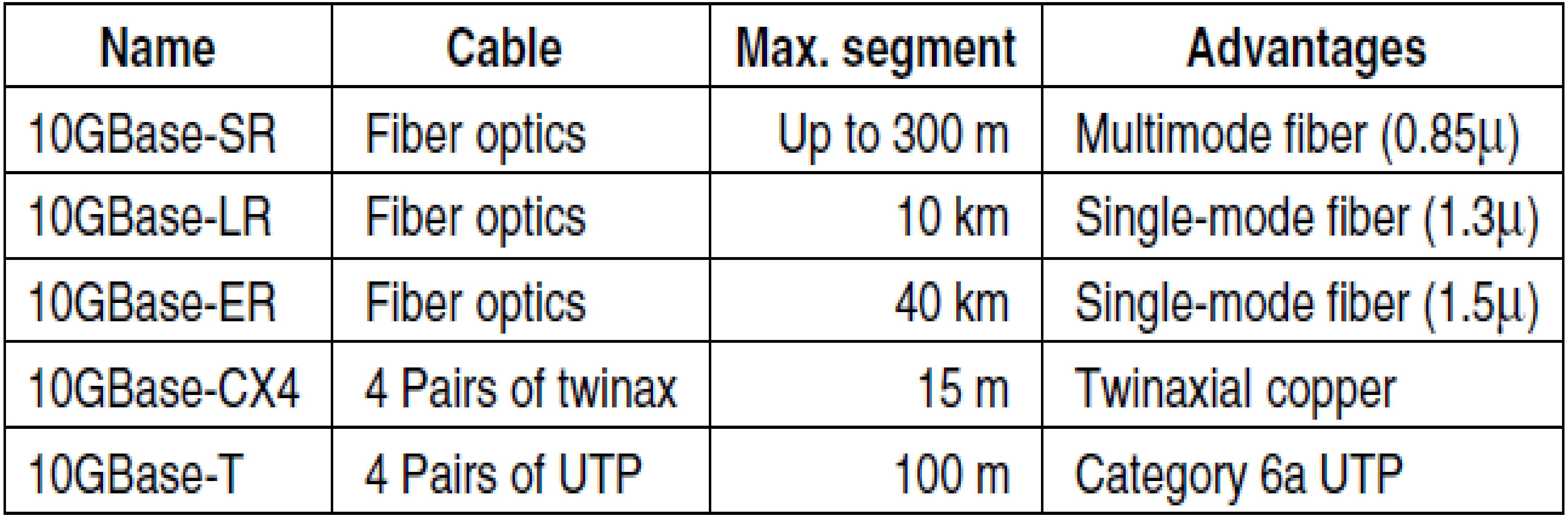
Two-wire copper (STP)

1000Base-T

Four-wire UTP

Characteristics	*1000Base-SX*	*1000Base-LX*	*1000Base-CX*	*1000Base-T*
Media	Fiber short-wave	Fiber long-wave	STP	Cat 5 UTP
Number of wires	2	2	2	4
Maximum length	550 m	5000 m	25 m	100 m
Block encoding	8B/10B	8B/10B	8B/10B	
Line encoding	NRZ	NRZ	NRZ	4D-PAM5

Summary of Ten-Gigabit Ethernet implementations



Name	Cable	Max. segment	Advantages
10GBase-SR	Fiber optics	Up to 300 m	Multimode fiber (0.85µ)
10GBase-LR	Fiber optics	10 km	Single-mode fiber (1.3µ)
10GBase-ER	Fiber optics	40 km	Single-mode fiber (1.5µ)
10GBase-CX4	4 Pairs of twinax	15 m	Twinaxial copper
10GBase-T	4 Pairs of UTP	100 m	Category 6a UTP

Thank You
???

References:

- Data Communications and Networking “ Behrouz A. Forouzan” Fourth Edition.
- Computer Networks “A. S. Tanenbaum” Fifth Edition
- Data and Computer Communications “William Stallings” Tenth Edition.