Unit-2

Data Link Control

- The two main functions of the data link layer are data link control and media access control. The first, data link control, deals with the design of procedures for communication between two adjacent nodes: node-to-node communication. The second function of the data link layer is media access control, or how to share the link.

- Data link control functions include framing, flow and error control protocols that provide smooth and reliable transmission of frames between nodes.

Data Link Layer

Error Detection and Correction

Error Detection and Correction

Single bit error

In single bit error, only 1 bit in the data unit has changed.

Burst error

A burst error means that 2 or more bits in the data unit have changed.

Redundancy

To detect or correct errors, we need to send extra bits with data.

Block Coding

In block coding, we 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.

Error Detection and Correction

Hamming Distance

- The Hamming distance between two words (of the same size) is the number of differences between the corresponding bits.
- The Hamming distance can easily be found if we apply the XOR operation on the two words and count the number of 1's in the result.

Example

Find the Hamming distance between the following words:-

a = 10101110 and b = 01010100

Minimum Hamming Distance

The minimum Hamming distance is the smallest Hamming distance between all possible pairs. We use d_{min} to define the minimum Hamming distance in a coding scheme.

Example

Find the minimum Hamming distance for the following set of words:-

{ 00000, 10101, 01011, 11110 }.

Error Detection and Correction

Minimum Hamming Distance for Error Detection

To guarantee the detection of up to **s** errors in all cases, the minimum Hamming distance in a block code must be

$$d_{min} = s + 1.$$

Minimum Hamming Distance for Error Correction

To guarantee the correction of up to t errors in all cases, the minimum Hamming distance in a block code must be

$$d_{min} = 2t + 1.$$

Error Detection and Correction

Simple Parity-Check Code

- In this code, a k-bit dataword is changed to an k+1-bit codeword.
- The extra bit, called the parity bit.
- It is selected to make the total number of 1's in the codeword even.

Note:

A simple parity-check code is a single-bit error-detecting code in which n = k + 1 with $d_{min} = 2$.

Error Detection and Correction

Hamming Code

Hamming code is a set of error-correction codes that can be used to **detect and correct the errors** that can occur when the data is moved or stored from the sender to the receiver.

It is **technique developed by R.W. Hamming for error correction**.

Redundant bits –

The number of redundant bits can be calculated using the following formula:

$$2^r >= m+r+1$$

Where, r = number of redundant bits, and

m = number of data bits

Error Detection and Correction

Algorithm of Hamming code

1. Write the bit positions starting from 1.
2. All the bit positions that are a power of 2 are marked as parity bits (1, 2, 4, 8, etc).
3. All the other bit positions are marked as data bits.
4. Each data bit is included in a unique set of parity bits, as determined its bit position in binary form.

   - Parity bit 1 covers all the bits positions whose binary representation includes a 1 in the least significant position (1, 3, 5, 7, 9, 11, etc).
   - Parity bit 2 covers all the bits positions whose binary representation includes a 1 in the second position from the least significant bit (2, 3, 6, 7, 10, 11, etc).
   - Parity bit 4 covers all the bits positions whose binary representation includes a 1 in the third position from the least significant bit (4–7, 12–15, 20–23, etc).
   - Parity bit 8 covers all the bits positions whose binary representation includes a 1 in the fourth position from the least significant bit bits (8–15, 24–31, 40–47, etc).

Error Detection and Correction

- In general, each parity bit covers all bits where the bitwise AND of the parity position and the bit position is non-zero.

5. Set a parity bit to 1 if the total number of ones in the positions it checks is odd.

6. Set a parity bit to 0 if the total number of ones in the positions it checks is even.

Error Detection and Correction

Ex. Construct an even parity Hamming code word for a data word 1011001.

Solution:

Step-1: First we compute the number of redundant bits r.

Here, number of bits in the given data word (1011001), m = 7

Therefore, we compute r as following:-

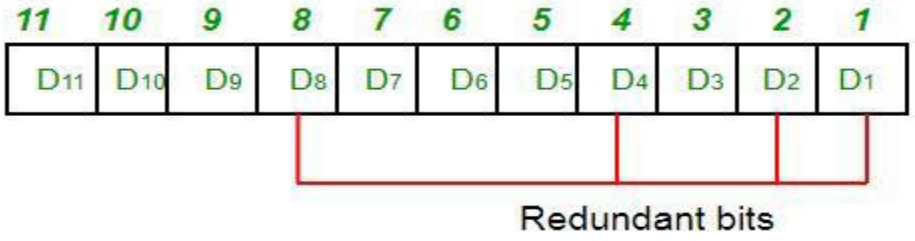
$2^r >= m+r+1$ ➔ $2^r >= 7+r+1$

Minimum value of r which satisfies above inequality = 4.

Therefore, r=4.

Step-2: Now, we compute the position of redundant bits in the codeword.

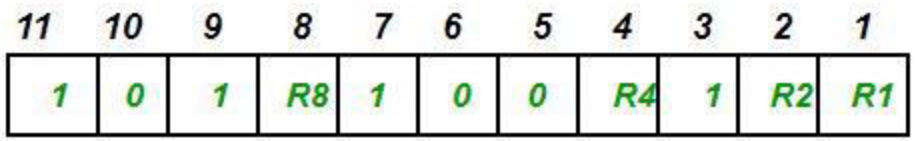
These redundancy bits are placed at positions that correspond to the power of 2. Therefore, the position these redundant bits will be 1, 2, 4 and 8.



Error Detection and Correction

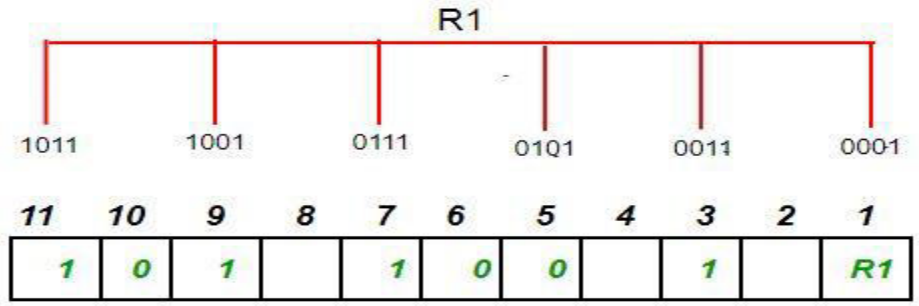
Step-3: Now, we compute the codeword.

Since the data to be transmitted is 1011001, therefore the bits will be placed as follows:



Determining the Parity bits:

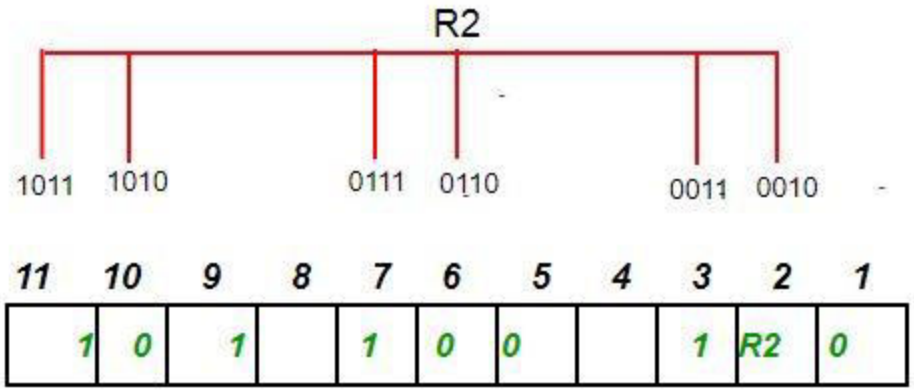
R1 bit is calculated using parity check at all the bits positions whose binary representation includes a 1 in the least significant position i.e. positions 1, 3, 5, 7, 9, 11.



To find the redundant bit R1, we check for even parity. Since the total number of 1's in all the bit positions corresponding to R1 is an even number the value of R1 (parity bit's value**) = 0**

Error Detection and Correction

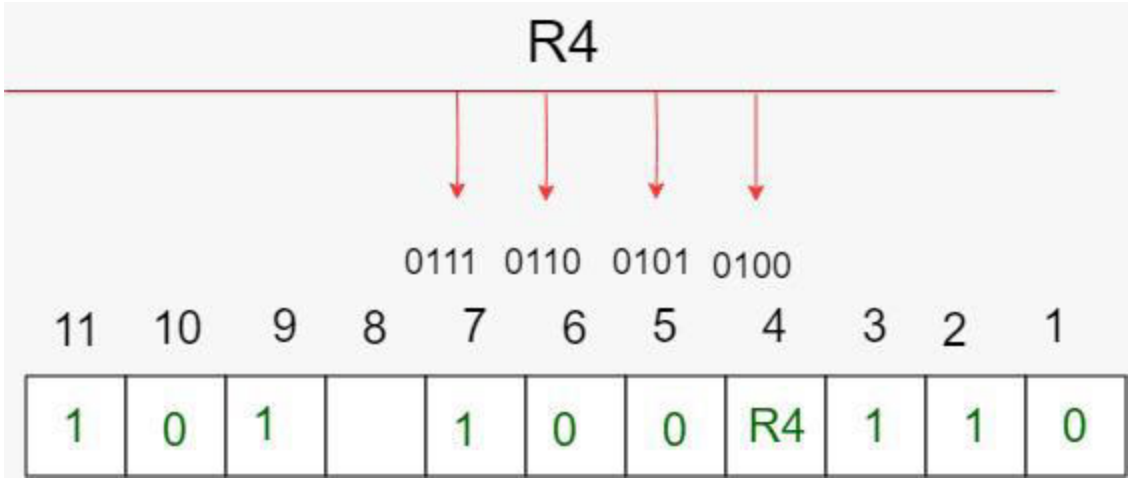
R2 bit is calculated using parity check at all the bits positions whose binary representation includes a 1 in the second position from the least significant bit. R2: bits 2,3,6,7,10,11.



To find the redundant bit R2, we check for even parity. Since the total number of 1's in all the bit positions corresponding to R2 is odd the value of R2(parity bit's value)**=1**

Error Detection and Correction

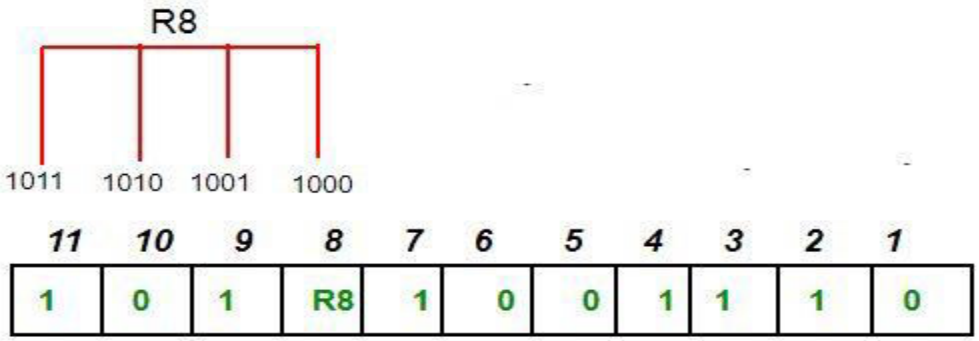
R4 bit is calculated using parity check at all the bits positions whose binary representation includes a 1 in the third position from the least significant bit. R4: bits 4, 5, 6, 7.



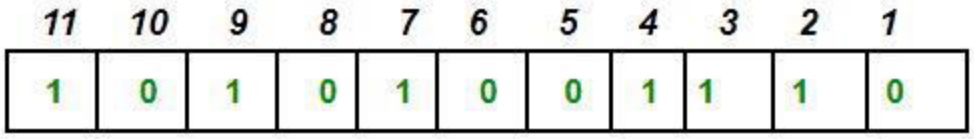
To find the redundant bit R4, we check for even parity. Since the total number of 1's in all the bit positions corresponding to R4 is odd the value of R4(parity bit's value) **= 1.**

Error Detection and Correction

R8 bit is calculated using parity check at all the bits positions whose binary representation includes a 1 in the fourth position from the least significant bit. R8: bit 8, 9, 10, 11.

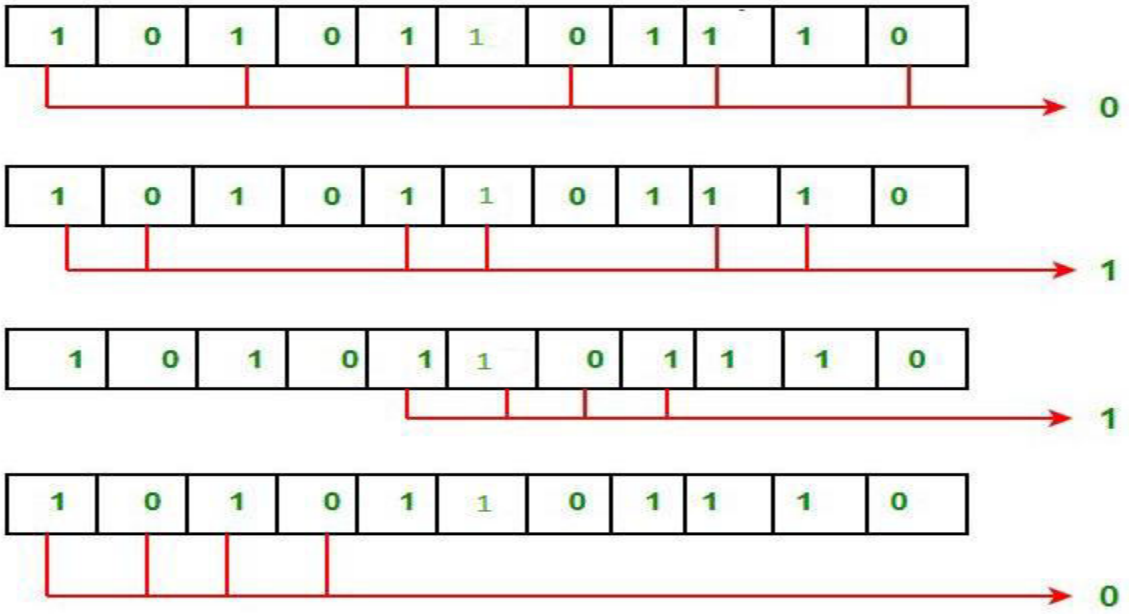


To find the redundant bit R8, we check for even parity. Since the total number of 1's in all the bit positions corresponding to R8 is an even number the value of R8(parity bit's value**)= 0**. **Thus, the data transferred is:**



Error Detection and Correction

Error detection and correction: Suppose in the above example the 6th bit is changed from 0 to 1 during data transmission, then it gives new parity values in the binary number:



The bits give the binary number **0110** whose decimal representation is **6.** Thus, bit **6** contains an error. **To correct the error** the 6th bit is changed from **1 to 0.**

Error Detection and Correction

Cyclic Redundancy Check (CRC)

- Suppose size of dataword is k-bits.
- This technique uses a divisor to find a codeword.
- Suppose size of divisor is m-bits.

At sender end:

The codeword corresponding to dataword is found in the following way:-

1. First we find a word by augmenting m-1 0's to the right end of the dataword.
2. Now, we divide this new word by the divisor and find a remainder.
3. Codeword is obtained by augmenting the remainder to the right end of dataword.

Error Detection and Correction

At receiver end:

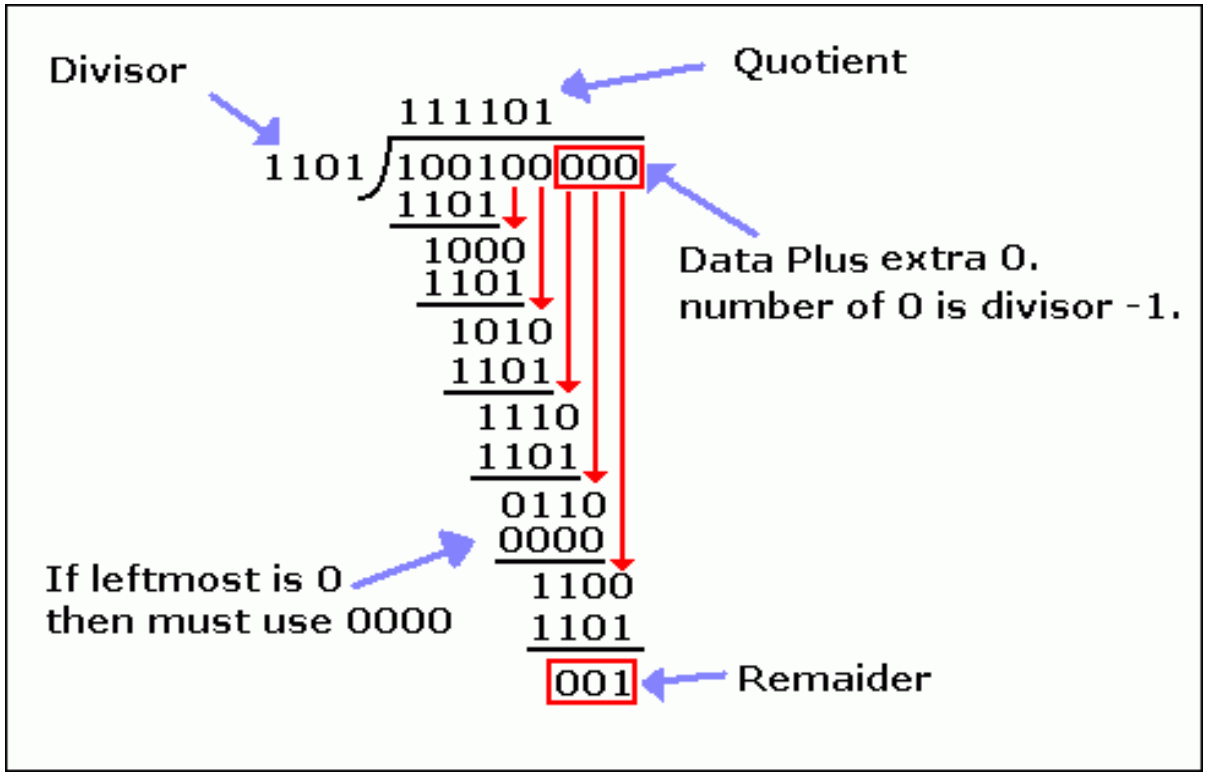
The dataword at the receiver end is found in the following way:-

1. First we divide the received codeword by divisor and find the remainder.
2. Remainder is called syndrome. If remainder is zero, then dataword will be accepted otherwise dataword will be rejected or discorded.
3. If remainder is zero, then dataword will be found by removing m-1 least significant bits of received codeword.

Error Detection and Correction

Example: If divisor is 1101, then find codeword corresponding to the dataword 100100.

Solution:



Therefore codeword = 100100001

Error Detection and Correction

Example:

(1) If Codeword 100100001 is received at receiver end, then find syndrome.

(2) If Codeword 100100101 is received at receiver end, then find syndrome.

CRC in polynomial

- The divisor in CRC is normally called generator.
- We define the following terms:-

Dataword = d(x) Codeword = c(x) Generator = g(x)

Syndrome = s(x) Error = e(x)

Error Detection and Correction

In a cyclic code,

1. If s(x) ≠ 0, one or more bits is corrupted.
2. If s(x) = 0, then either
   a. No bit is corrupted. or
   b. Some bits are corrupted, but the decoder failed to detect them.

Received codeword = c(x) + e(x)

The receiver divides the received codeword by g(x) to get the syndrome.

$$\frac{Received\ codeword}{g(x)} = \frac{c(x)}{g(x)} + \frac{e(x)}{g(x)}$$

$\frac{c(x)}{g(x)}$ $does\ not\ have\ a\ remainder.\ So\ the\ syndrome\ is\ the$ $remainder\ of\ \frac{e(x)}{g(x)}$.

In a cyclic code, those e(x) errors that are divisible by g(x) are not caught.

Error Detection and Correction

Example:

Let dataword $d(x) = x^3+1$, generator $g(x) = x^3+x+1$.

Find codeword.

Solution:

Augmented dataword $= x^6+x^3$

$x^3+x+1)\ x^6+x^3\ (x^3+x$

$x^6+ x^4 + x^3$

x^4

$x^4+ x^2 + x$

$x^2 + x$ ← Remainder

Therefore, $c(x) = x^6+ x^3 + x^2 + x$

Error Detection and Correction

Single-Bit Error

If the generator has more than one term and the coefficient of x^0 is 1, then all single bit errors can be caught.

Example:

Which of the following g(x) values guarantees that a single-bit error is caught? For each case, what is the error that cannot be caught?

(a) $x + 1$

(b) x^3

(c) 1

Solution:

(a) No x^i can be divisible by $x + 1$. In other words, $x^i/(x + 1)$ always has a remainder. So the syndrome is nonzero. Any single-bit error can be caught.

Error Detection and Correction

(b) If i is equal to or greater than 3, then x^i is divisible by g(x). The remainder of x^i/x^3 is zero, and the receiver is fooled into believing that there is no error, although there might be one.

Note that in this case, the corrupted bit must be in position 4 or above. All single-bit errors in positions 1 to 3 are caught.

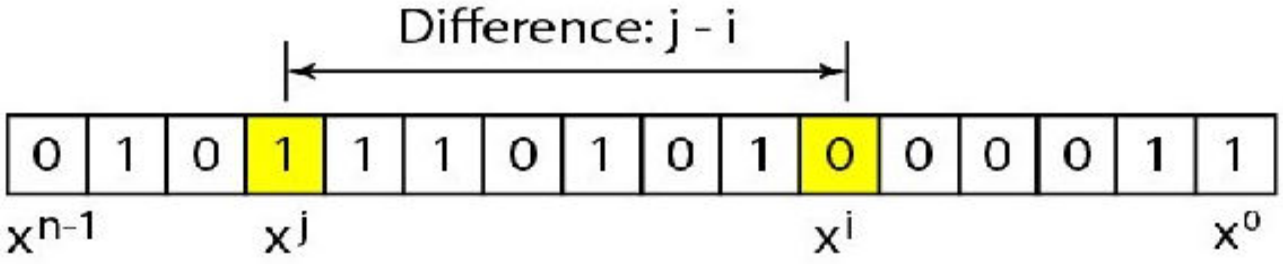
(c) For all values of i, x^i is divisible by g(x). No single-bit error can be caught. In addition, this g(x) is useless because it means the codeword is just the dataword augmented with n-k zeros.

Error Detection and Correction

Two Isolated Single-Bit Errors

$$e(x) = x^j + x^i$$

The values of i and j define the positions of the errors, and the difference j - i defines the distance between the two errors.



- If a generator cannot divide $x^t + 1$ (t between 0 and n - 1), then all isolated double errors can be detected.

Error Detection and Correction

Example:

Find the status of the following generators related to two isolated, single-bit errors.

(a) $x + 1$

(b) $x^4 + 1$

(c) $x^7 + x^6 + 1$

(d) $x^{15} + x^{14} + 1$

Solution:

(a) This is a very poor choice for a generator. Any two errors next to each other cannot be detected.

(b) This generator cannot detect two errors that are four positions apart. The two errors can be anywhere, but if their distance is 4, they remain undetected.

Error Detection and Correction

(c) This is a good choice for this purpose.

(d) This polynomial cannot divide any error of type $x^t + 1$ if t is less than 32,768. This means that a codeword with two isolated errors that are next to each other or up to 32,768 bits apart can be detected by this generator.

Odd Numbers of Errors

A generator that contains a factor of x + 1 can detect all odd-numbered errors.

Some standard generators

(1) CRC-8 = $x^8+x^2+x + 1$

(2) (2) CRC-10 = $x^{10}+x^9+x^5+x^4+x^2+ 1$

(3) CRC-16 = $x^{16} + x^{12} + x^5 + 1$

(4) CRC-32 = $x^{32} + x^{26} +x^{23} + x^{22} + x^{16} + x^{12} + x^{11} + x^{10} + x^8 + x^7 + x^5 + x^4 + x^2 + x + 1$

Error Detection and Correction

CHECKSUM

- Checksum is an error detection method.
- The checksum is used in the Internet by several protocols.
- The checksum is based on the concept of redundancy.

Internet Checksum

Internet uses 16-bit checksum. The sender calculates the checksum by following these steps.

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.

Error Detection and Correction

The receiver uses the following steps for error detection.

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 and Correction

Example: Calculate the checksum for a text of 8 characters ("Forouzan").

Solution:

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

b. Checksum at the receiver site

Error Detection and Correction

Some questions:

1. What is the Hamming distance for each of the following codewords:

   a. d (10000, 00000)

   b. d (10101, 10000)

   c. d (11111,11111)

   d. d (000, 000)

2. Find the minimum Hamming distance for the following cases:

   a. Detection of two errors.

   b. Correction of two errors.

   c. Detection of 3 errors or correction of 2 errors.

   d. Detection of 6 errors or correction of 2 errors.

Error Detection and Correction

3. Which of the following CRC generators guarantee the detection of a single bit error?

   a. $x^3 + x + 1$

   b. $x^4 + x$

   c. 1

   d. $x^2 + 1$

4. Sender needs to send the four data items 0x3456, 0xABCC, 0x02BC, and 0xEEEE. Answer the following:

   a. Find the checksum at the sender site.

   b. Find the checksum at the receiver site if there is no error.

   c. Find the checksum at the receiver site if the second data item is changed to 0xABCE.

   d. Find the checksum at the receiver site if the second data item is changed to 0xABCE and the third data item is changed to 0x02BA.

Data Link Control

Data Link Control

Framing:

- 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.
- Frames can be of fixed size or of variable size.

Fixed-Size Framing

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

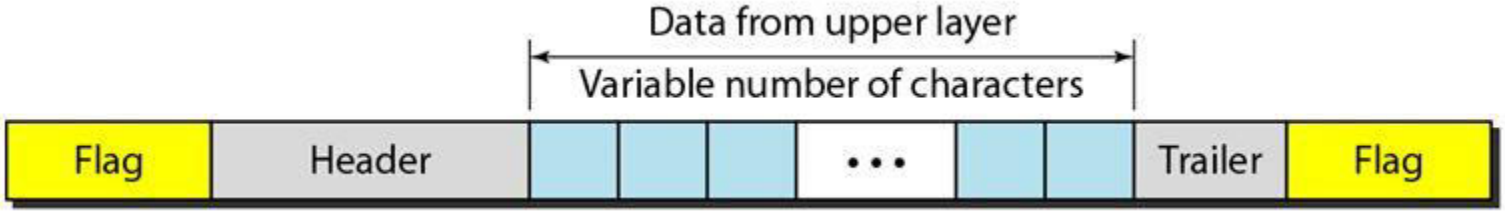
Data Link Control

Variable-Size Framing

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

Character-Oriented Protocols (byte oriented)

In 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, and the trailer, which carries error detection or error correction redundant bits, are also multiples of 8 bits. To separate one frame from the next, an 8-bit (1-byte) flag is added at the beginning and the end of a frame. The flag, composed of protocol-dependent special characters, signals the start or end of a frame. Following figure shows the format of a frame in a character-oriented protocol.



Character-Oriented Protocols (byte oriented)

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 fix this problem, a byte-stuffing strategy was added to character-oriented framing. In byte stuffing (or character stuffing), a special byte is added to the data section of the frame when there is a character with the same pattern as the flag. The data section is stuffed with an extra byte. This 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.

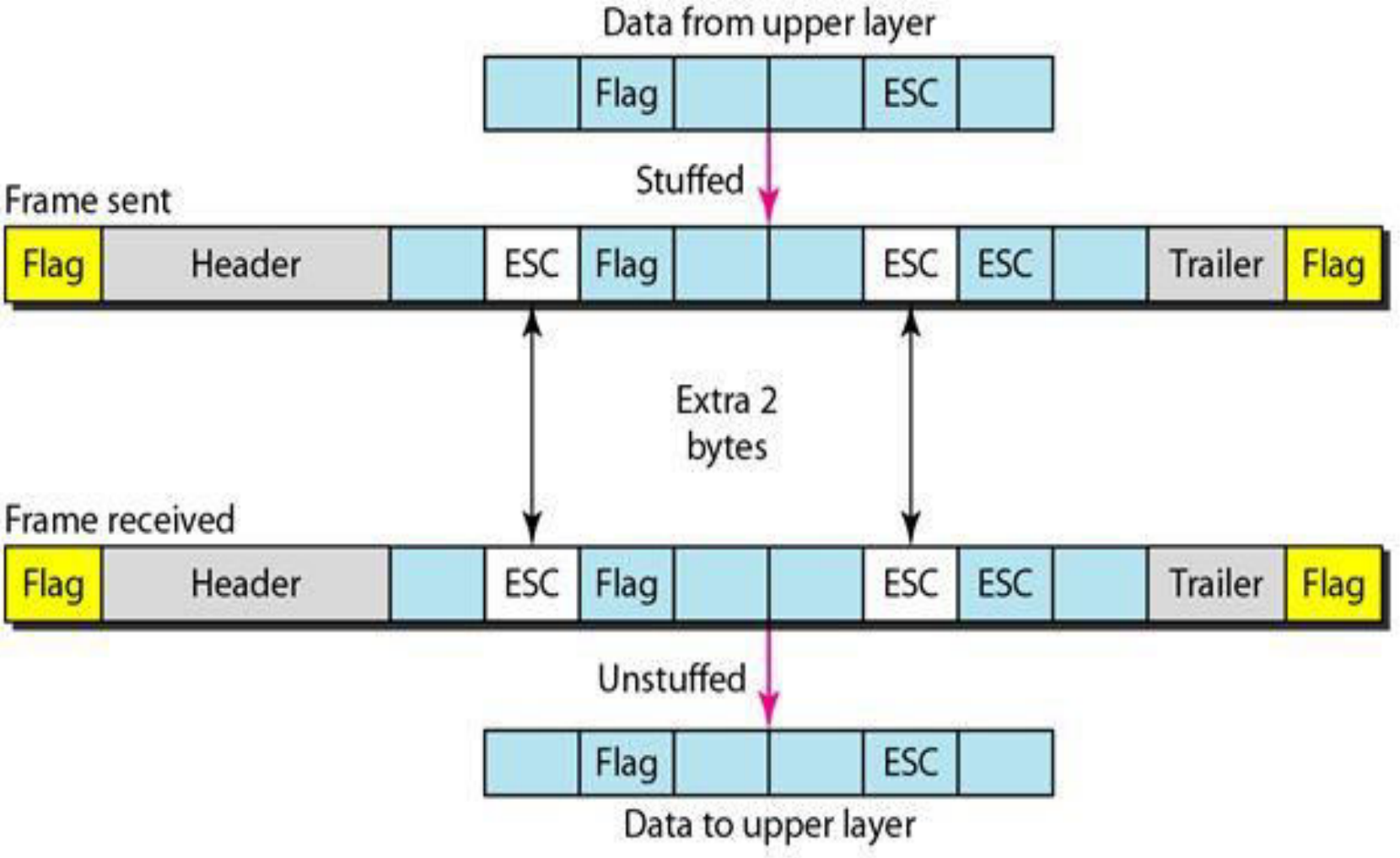
Character-Oriented Protocols (byte oriented)

Byte stuffing by the escape character allows the presence of the flag in the data section of the frame, but it creates another problem. What happens if the text contains one or more escape characters followed by a flag? The receiver removes the escape character, but keeps the flag, which is incorrectly interpreted as the end of the frame. To solve this problem, the escape characters that are part of the text must also be marked by another escape character. In other words, 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.

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

Character-Oriented Protocols (byte oriented)

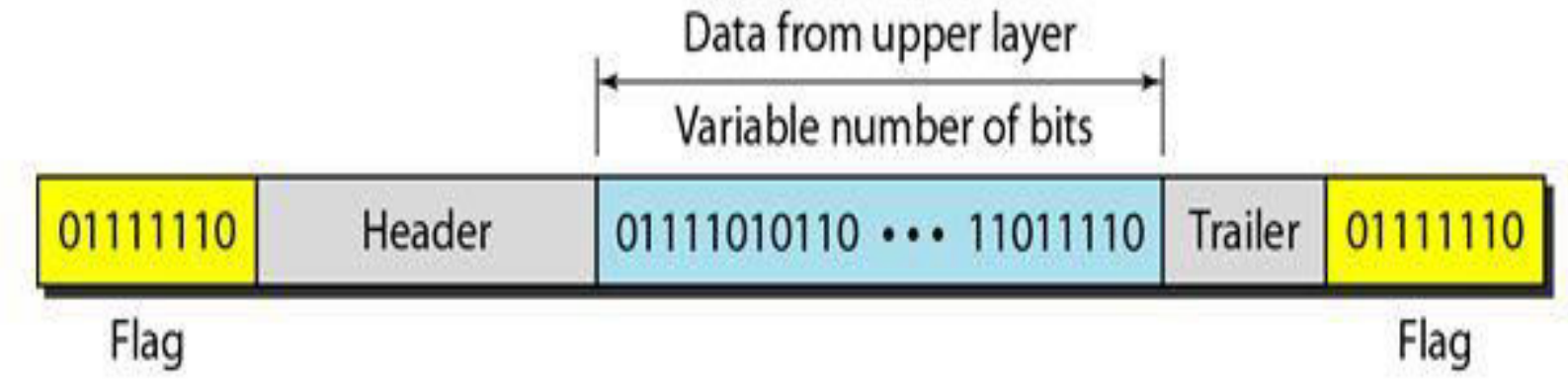
Byte stuffing and unstuffing



Bit-Oriented Protocols

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. However, in addition to headers (and possible trailers), we still need a delimiter to separate one frame from the other. Most protocols use a special 8-bit pattern flag 01111110 as the delimiter to define the beginning and the end of the frame, as shown in following figure .

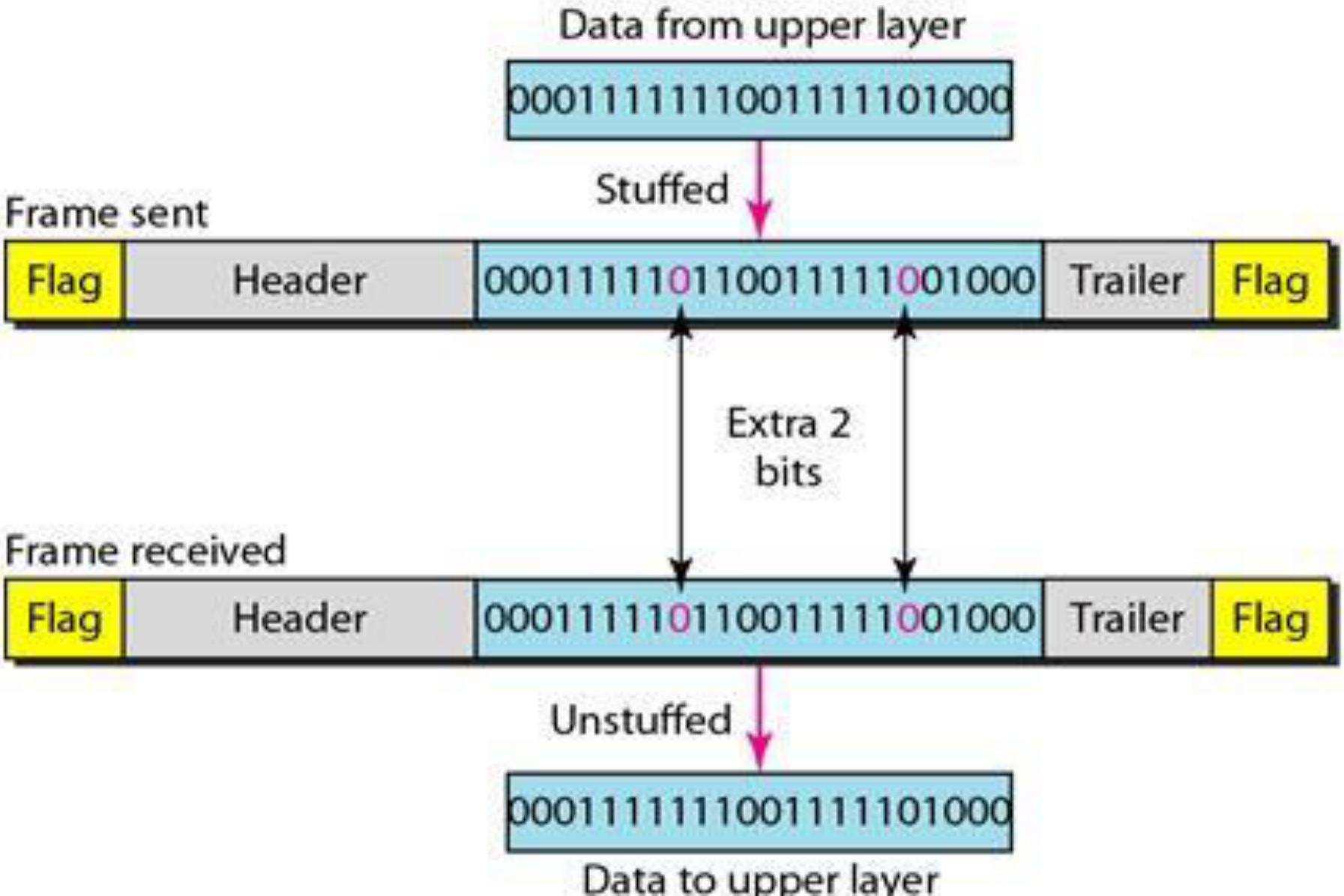


Bit-Oriented Protocols

This flag can create the same type of problem we saw in the byte-oriented protocols. That is, if the flag pattern appears in the data, we need to somehow inform the receiver that this is not the end of the frame. We do this by stuffing 1 single bit (instead of 1 byte) to prevent the pattern from looking like a flag. The strategy is called bit stuffing. In bit stuffing, if a 0 and five consecutive 1 bits are encountered, an extra 0 is added. This extra stuffed bit is eventually removed from the data by the receiver. Note that the extra bit is added after one 0 followed by five 1's regardless of the value of the next bit. This guarantees that the flag field sequence does not inadvertently appear in the frame.

Note: Bit stuffing is the process of adding one extra 0 whenever five consecutive 1's follow a 0 in the data, so that the receiver does not mistake the pattern 0111110 for a flag.

Bit-Oriented Protocols



Bit-Oriented Protocols

Figure shows bit stuffing at the sender and bit removal at the receiver. Note that even if we have a 0 after five 1's, we still stuff a 0. The 0 will be removed by the receiver. If the flag like pattern 01111110 appears in the data, it will change to 011111010 (stuffed) and is not mistaken as a flag by the receiver. The real flag 01111110 is not stuffed by the sender and is recognized by the receiver.

Flow and Error Control

The most important responsibilities of the data link layer are flow control and error control.

Flow control

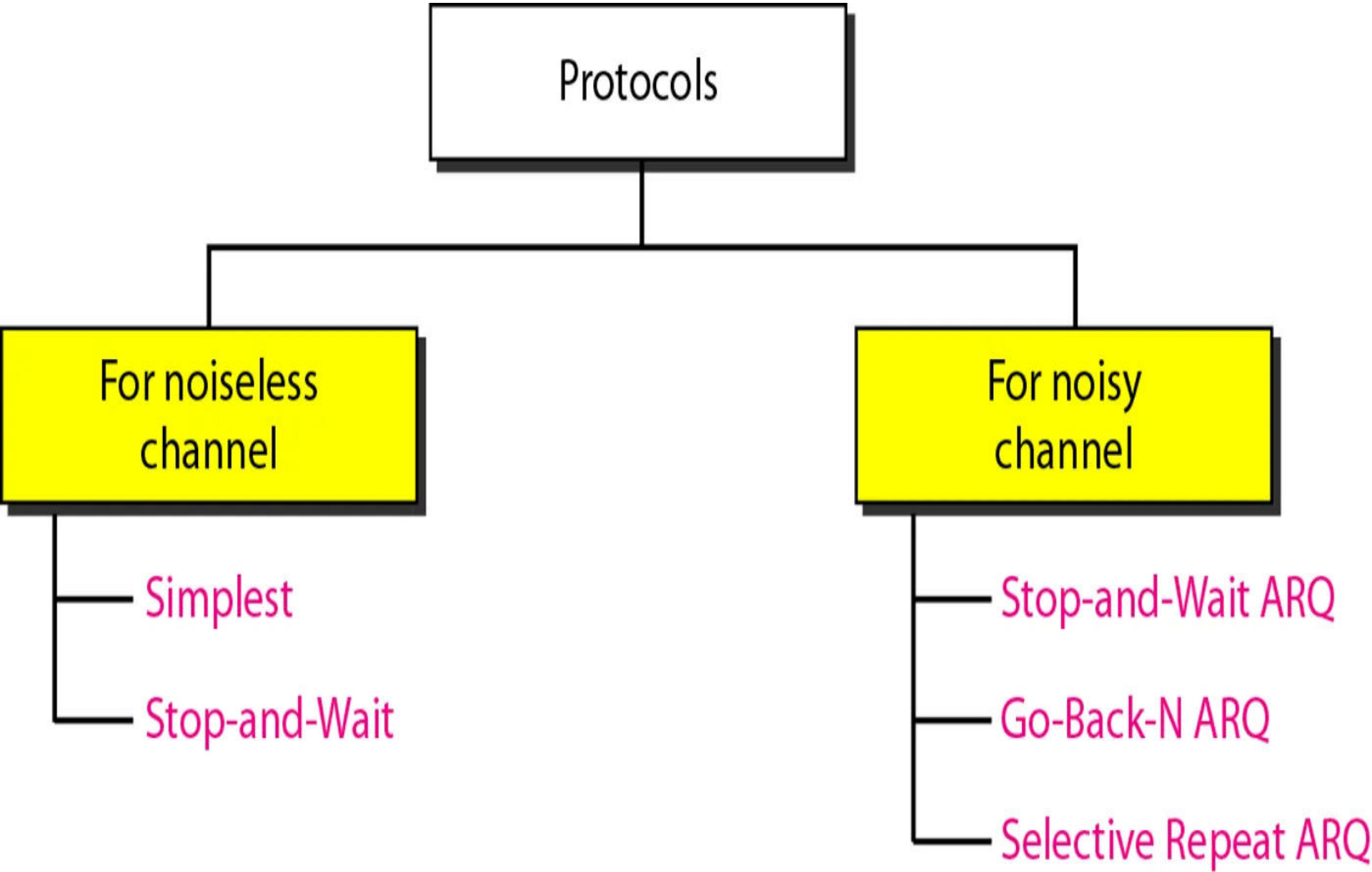
- Flow control coordinates the amount of data that can be sent before receiving an acknowledgment.
- In most protocols, flow control is a set of procedures that tells the sender how much data it can transmit before it must wait for an acknowledgment from the receiver.
- The flow of data must not be allowed to overwhelm the receiver. Any receiving device has a limited speed at which it can process incoming data and a limited amount of memory in which to store incoming data. The receiving device must be able to inform the sending device before those limits are reached and to request that the transmitting device send fewer frames or stop temporarily.

Flow and Error Control

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.
- Error control in the data link layer is often implemented simply: Any time an error is detected in an exchange, specified frames are retransmitted. This process is called automatic repeat request (ARQ).

Data link control protocol



Data link control protocol

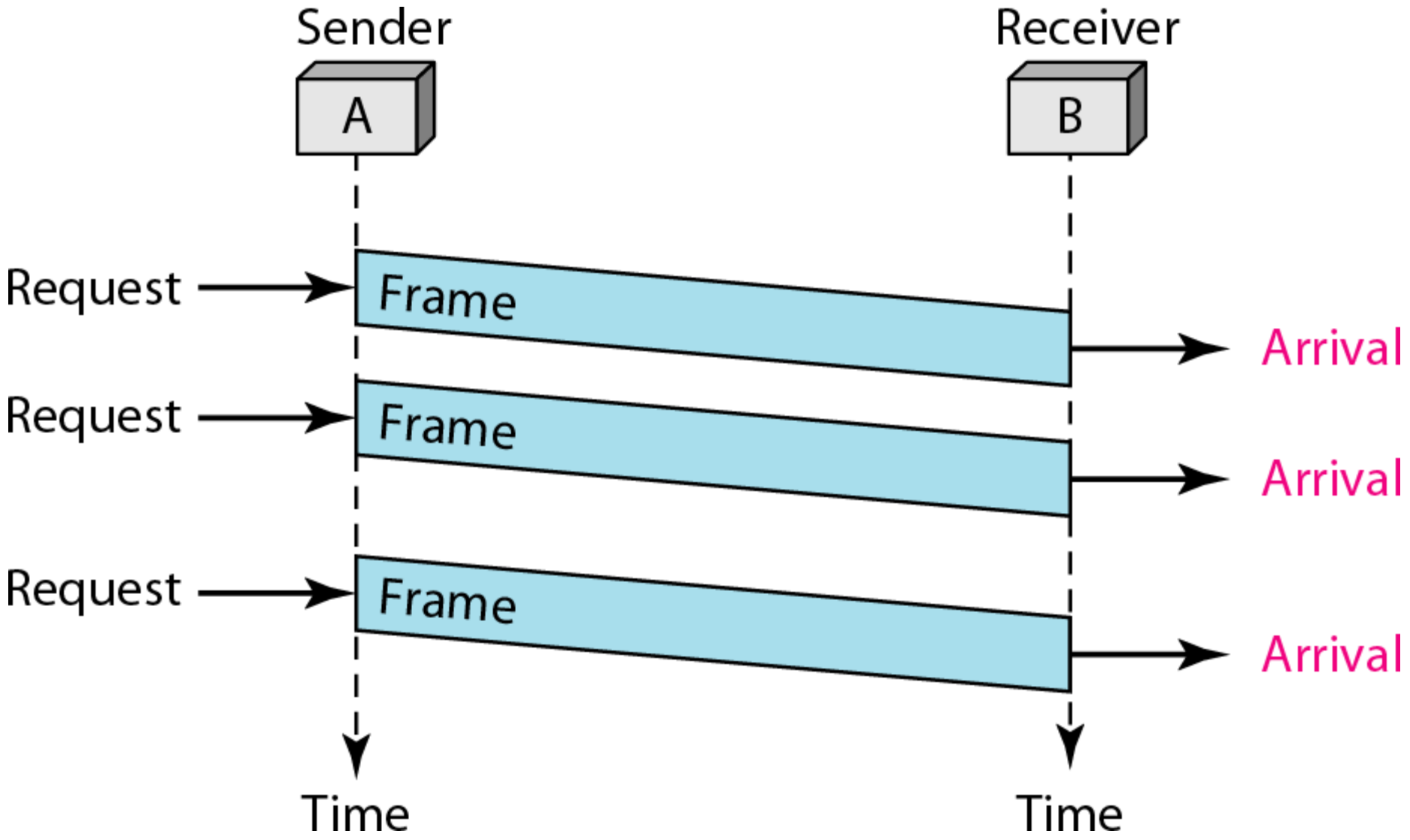
Noiseless Channels

- Assume we have an ideal channel in which no frames are lost, duplicated, or corrupted.
- We have two protocols for this type of channel. The first is a protocol that does not use flow control; the second is the one that does.
- Neither has error control because we have assumed that the channel is a perfect noiseless channel.

Simplest Protocol

- It is one that has no flow or error control.
- It is a unidirectional protocol in which data frames are traveling in only one direction-from the sender to receiver.
- We assume that the receiver can immediately handle any frame it receives with a processing time that is small enough to be negligible. 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. In other words, the receiver can never be overwhelmed with incoming frames.

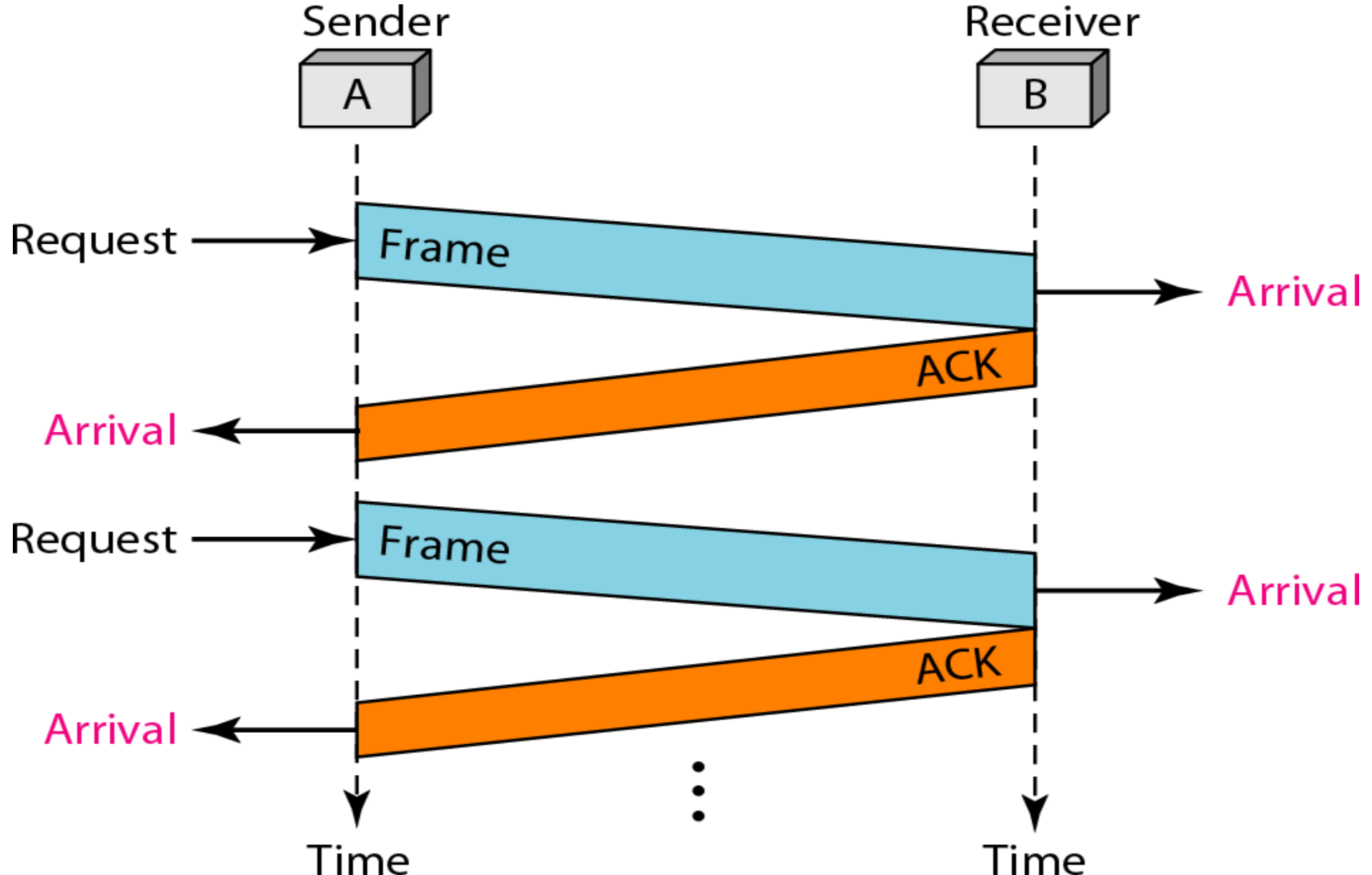
Simplest Protocol



Stop-and-Wait Protocol

- The sender sends one frame, stops until it receives confirmation from the receiver (okay to go ahead), and then 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.
- We add flow control to our previous protocol.

Stop-and-Wait Protocol



NOISY CHANNELS

Stop-and-Wait Automatic Repeat Request (Stop-and-Wait ARQ)

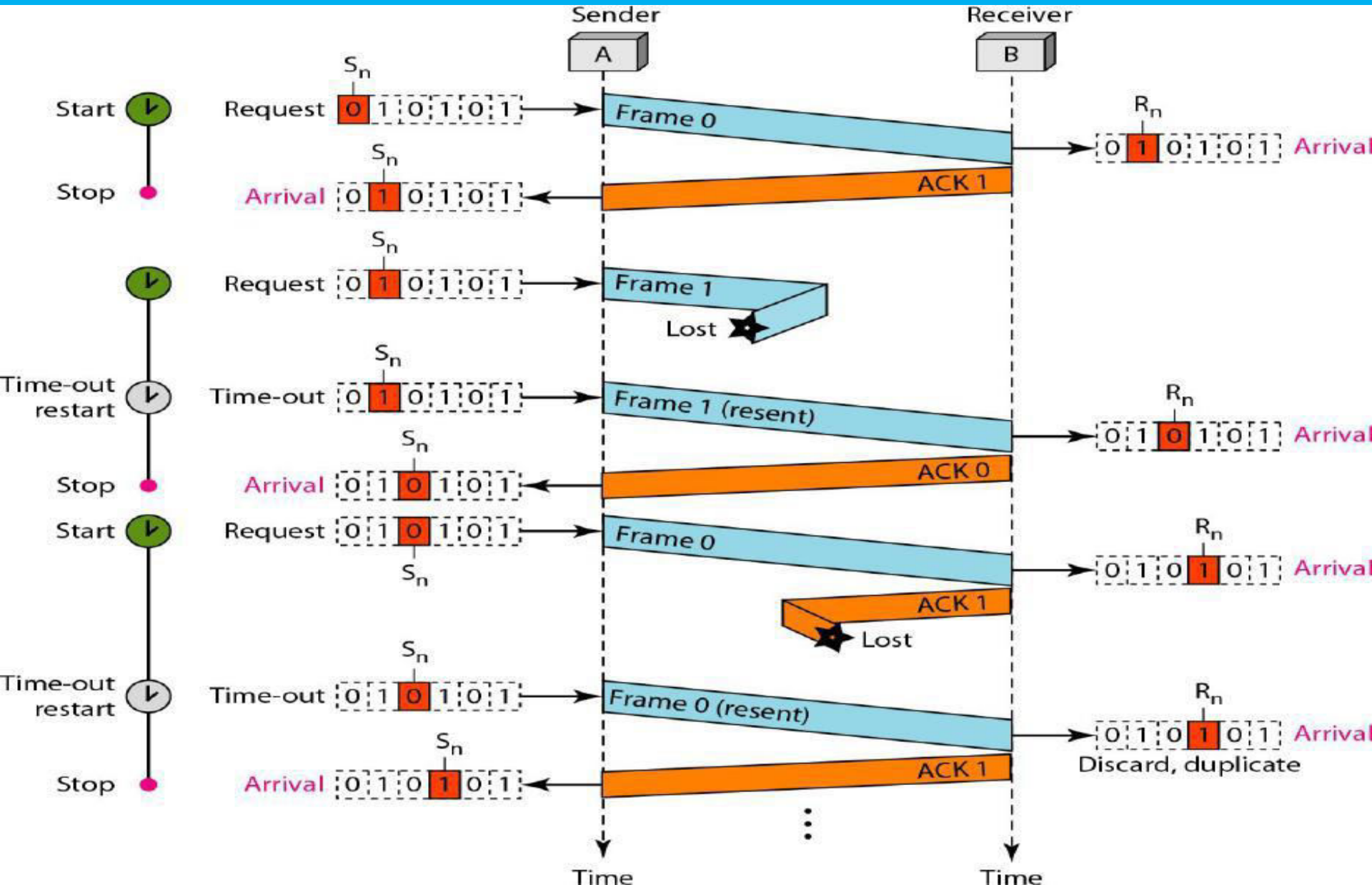
- This protocol adds a simple error control mechanism to the Stop-and-Wait Protocol.
- To detect and correct corrupted frames, we need to add redundancy bits to our data frame. When the frame arrives at the receiver site, it is checked and if it is corrupted, it is silently discarded.
- Lost frames are more difficult to handle than corrupted ones.
- To handle lost frames, this protocol uses sequence number.

Stop-and-Wait ARQ

Stop-and-Wait Automatic Repeat Request (Stop-and-Wait ARQ)

- 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.
- When the sender sends a frame, it keeps a copy of the sent frame. At the same time, it starts a timer. If the timer expires and there is no ACK for the sent frame, the frame is resent, the copy is held, and the timer is restarted.

Stop-and-Wait ARQ



Stop-and-Wait ARQ

- Since an ACK frame can also be corrupted and lost, it too needs redundancy bits and a sequence number.
- The ACK frame for this protocol has a sequence number field.
- In this protocol, the sender simply discards a corrupted ACK frame or ignores an out-of-order one.
- A field is added to the data frame to hold the sequence number of that frame.

Stop-and-Wait ARQ

- If the number of bits used for sequence number is m, then range of sequence numbers is 0 to 2^m-1.
- This protocol uses 1 bit for sequence number i.e. m=1.
- This protocol uses the following sequence numbers 0, 1, 0, 1, 0, 1, 0, 1, 0, 1……
- This protocol uses the sliding window concept.
- The size of both sender and receiver window in this protocol is 1.

Stop-and-Wait ARQ

Efficiency

- The Stop-and-Wait ARQ discussed in the previous section is very inefficient if our channel is thick and long. By thick, we mean that our channel has a large bandwidth; by long, we mean the round-trip delay is long. The product of these two is called the bandwidth delay Product.
- The channel is always there. If we do not use it, we are inefficient. The bandwidth-delay product is a measure of the number of bits we can send out of our system while waiting for news from the receiver.

Stop-and-Wait ARQ

Example:

Assume that, in a Stop-and-Wait ARQ system, the bandwidth of the line is 1 Mbps, and 1 bit takes 20 ms to make a round trip. What is the bandwidth-delay product? If the system data frames are 1000 bits in length, what is the utilization percentage of the link?

Stop-and-Wait ARQ

Solution:

The bandwidth-delay product is

$$= 1*10^{6} * 20*10^{-3} = 20*10^{3}$$

$$= 20000 \text{ bits}$$

- The system can send 20,000 bits during the time it takes for the data to go from the sender to the receiver and then back again. However, the system sends only 1000 bits. We can say that the link utilization is only 1000/20,000, or 5 percent.
- For this reason, for a link with a high bandwidth or long delay, the use of Stop-and-Wait ARQ wastes the capacity of the link.

Stop-and-Wait ARQ

Example:

What is the utilization percentage of the link in the previous example if we have a protocol that can send up to 15 frames before stopping and worrying about the acknowledgments?

Solution:

The bandwidth-delay product is still 20,000 bits. The system can send up to 15 frames or 15,000 bits during a round trip. This means the utilization is 15,000/20,000, or 75 percent.

Note: Of course, if there are damaged frames, the utilization percentage is much less because frames have to be resent.

Go-Back-N Automatic Repeat Request

- To improve the efficiency of transmission (filling the pipe), multiple frames must be in transition while waiting for acknowledgment.
- In this protocol we can send several frames before receiving acknowledgments; we keep a copy of these frames until the acknowledgments arrive.
- In the Go-Back-N Protocol, the sequence numbers are modulo 2^m, where m is the size of the sequence number field in bits.
- In this protocol, the sliding window is an abstract concept that defines the range of sequence numbers that is the concern of the sender and receiver.

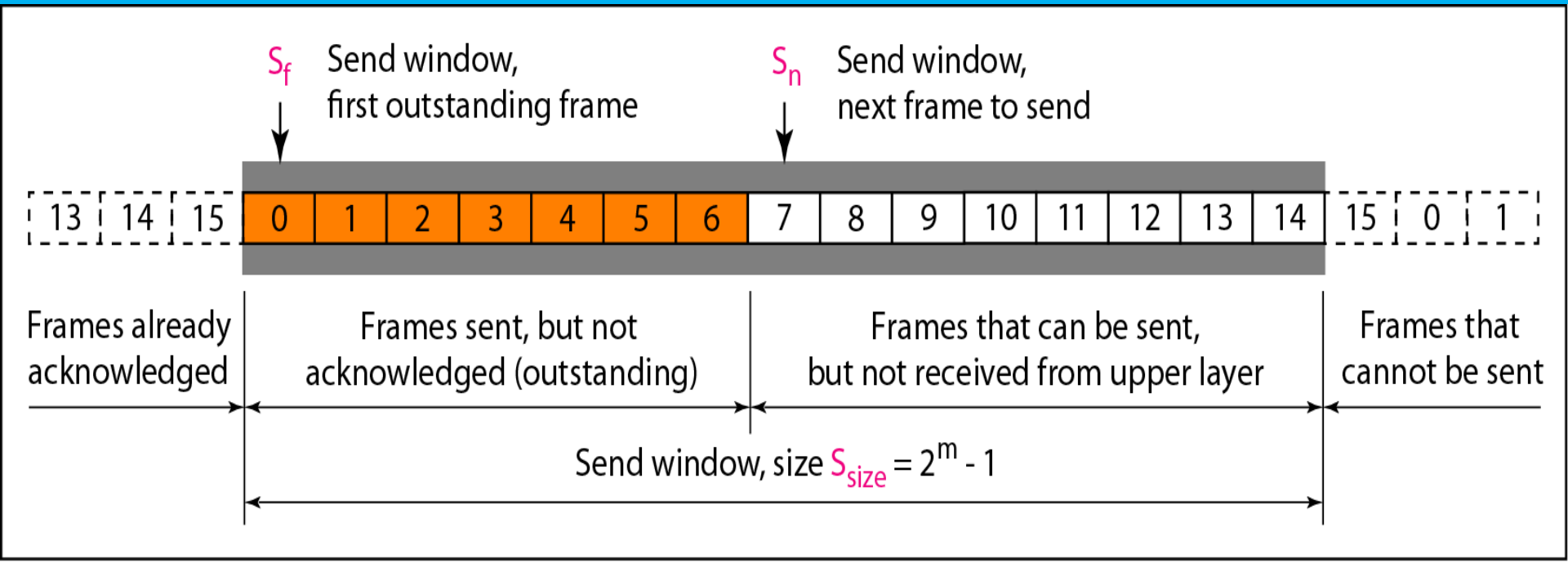
Go-Back-N Automatic Repeat Request

- If m=4 bits are used for sequence number, then the only sequence numbers are 0 through 15 inclusive. So the sequence numbers are

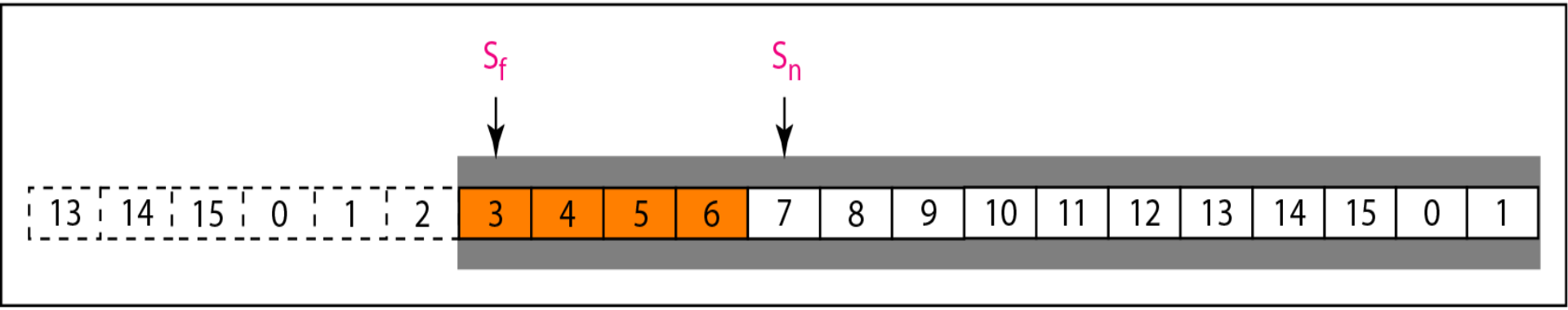
0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, ...

- The maximum size of the sender window is 2^m-1.
- The size of the receiver window is 1.
- The sender window at any time divides the possible sequence numbers into four regions.

Go-Back-N Automatic Repeat Request

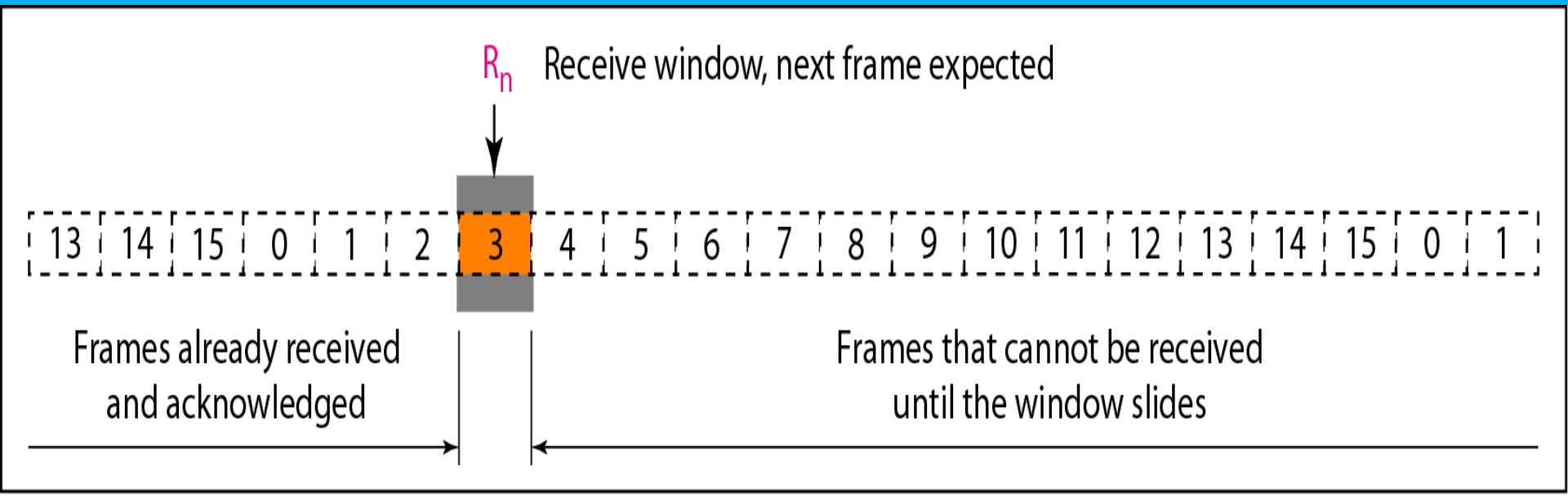


a. Send window before sliding

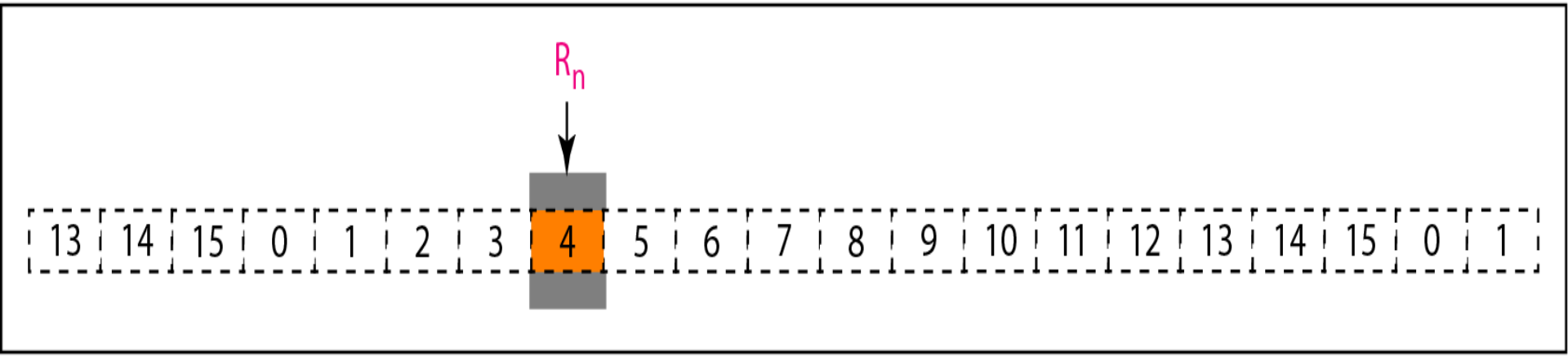


b. Send window after sliding

Go-Back-N Automatic Repeat Request



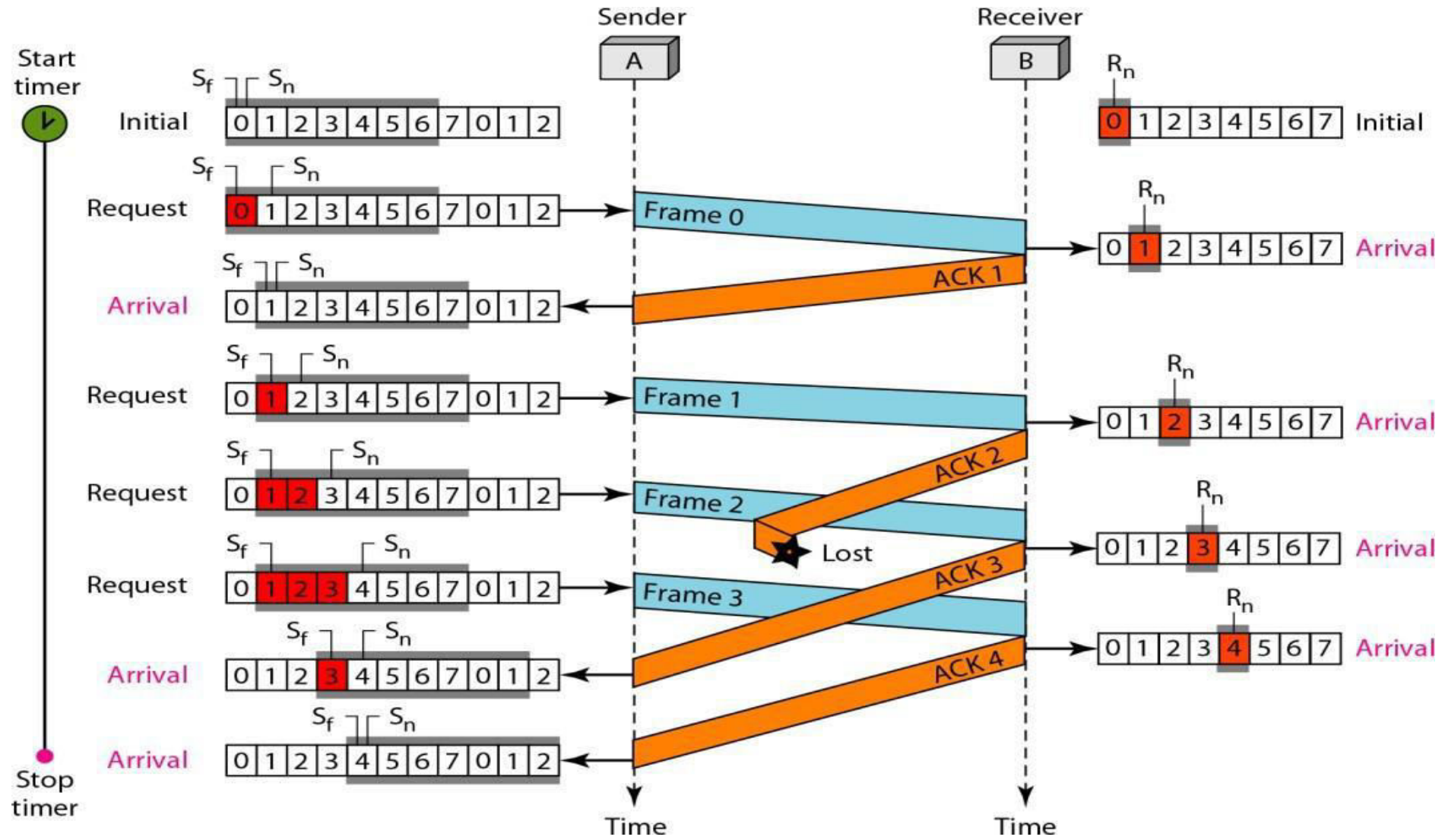
a. Receive window



b. Window after sliding

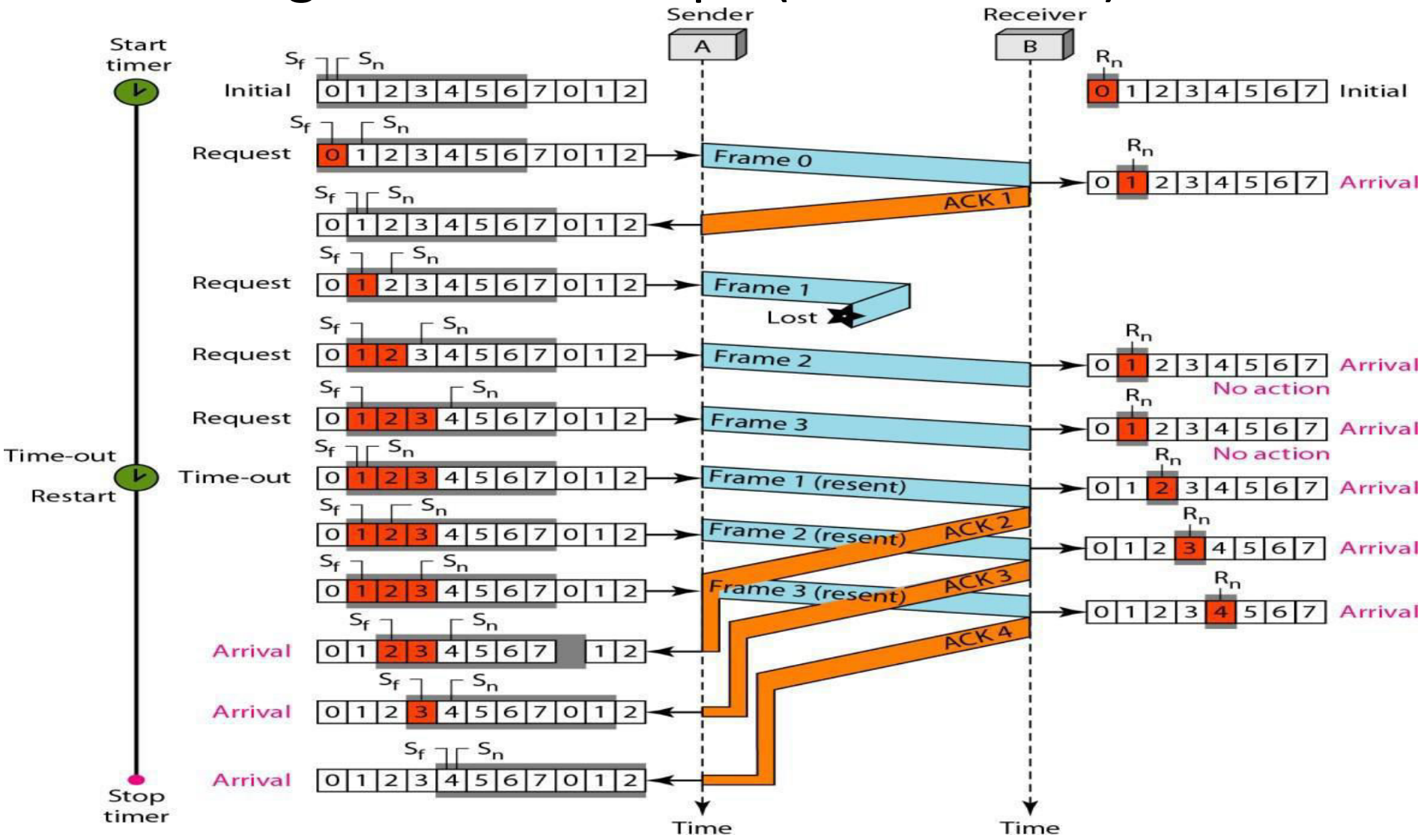
Go-Back-N Automatic Repeat Request

❖ **Below figure is an example(if ack lost)**



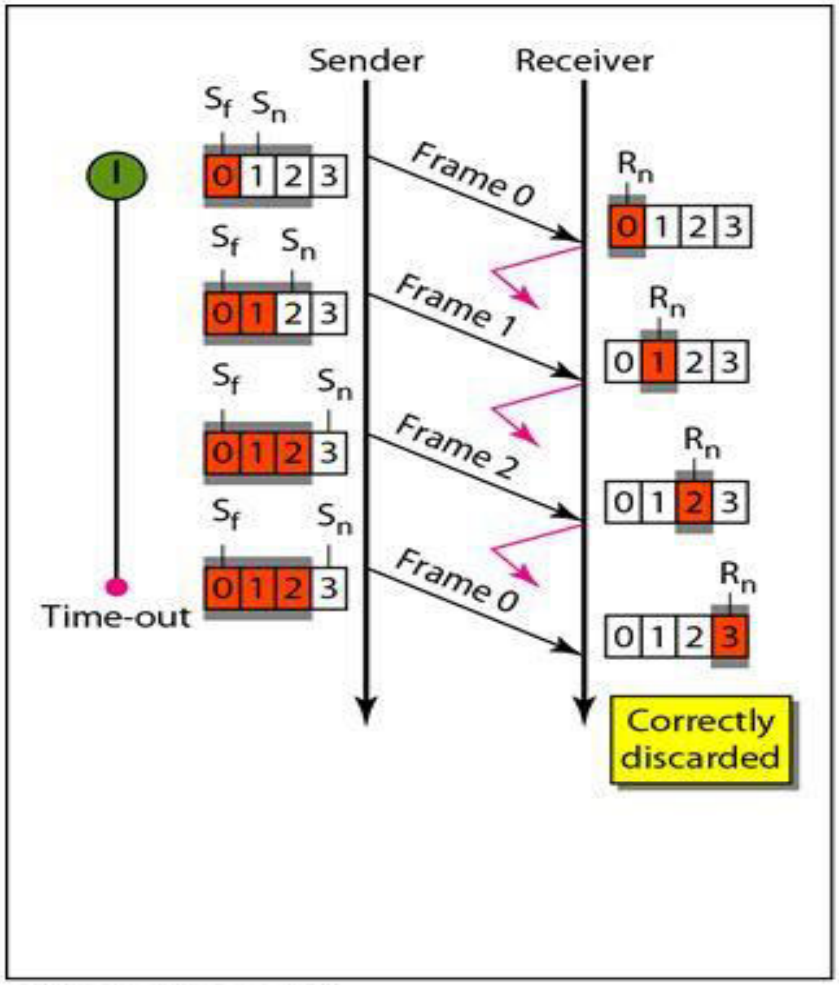
Go-Back-N Automatic Repeat Request

❖ Below figure is an example(if frame lost)

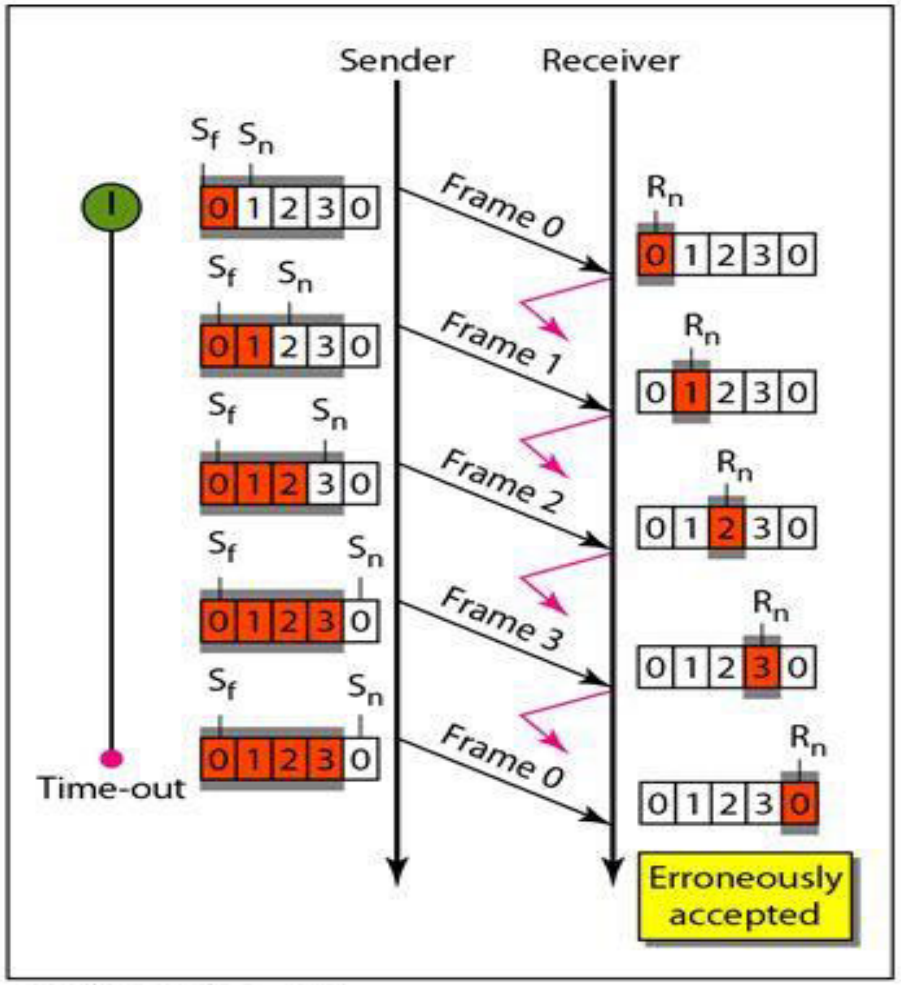


Go-Back-N Automatic Repeat Request

Note: In Go-Back-N ARQ, the size of the sender window must be less than 2^m; the size of the receiver window is always 1.



a. Window size $< 2^m$



b. Window size $= 2^m$

Go-Back-N Automatic Repeat Request

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

Efficiency:

The efficiency of Go-Back-N ARQ ,

Efficiency = N/(1+2a),

Where N is the size of sender window and $a = T_p/T_t$.

Where T_p is propagation delay and T_t is the transmission delay

Also, $T_t = D/B$;

and here D = data size and B = bandwidth

And $T_p = d/v$,

here d = distance and v = propagation speed.

Go-Back-N Automatic Repeat Request

Now to find the effective bandwidth (or throughput),

Effective bandwidth = efficiency * bandwidth,

which means,

Effective bandwidth = (N/(1+2a)) *B

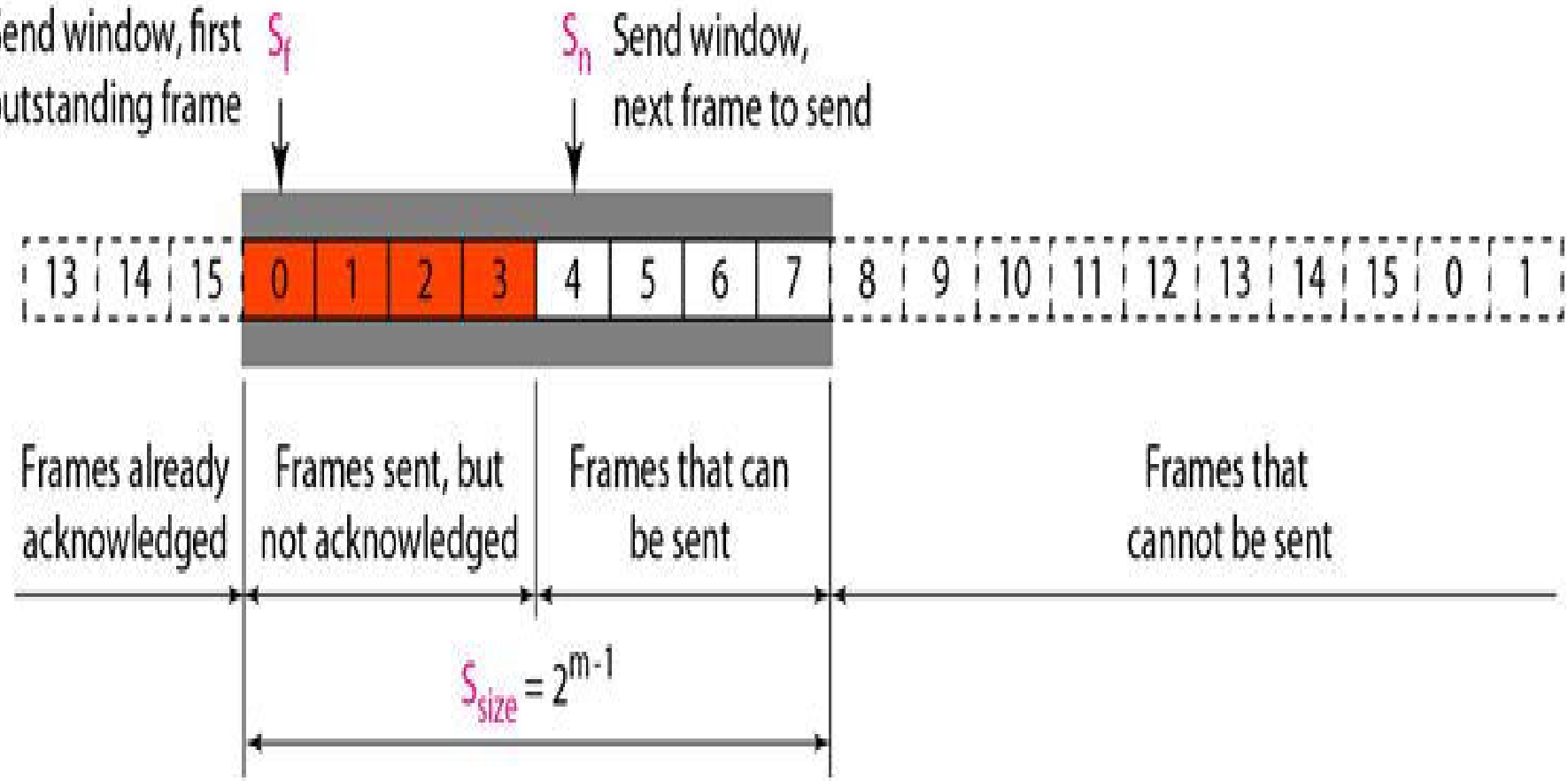
Selective Repeat Automatic Repeat Request

- Go-Back-N ARQ protocol is inefficient for noisy links because in a noisy link a frame has a higher probability of damage, which means the resending of multiple frames. This resending uses up the bandwidth and slows down the transmission.
- For noisy links, there is another mechanism that does not resend N frames when just one frame is damaged; only the damaged frame is resent. This mechanism is called Selective Repeat ARQ.
- It is more efficient for noisy links, but the processing at the receiver is more complex.

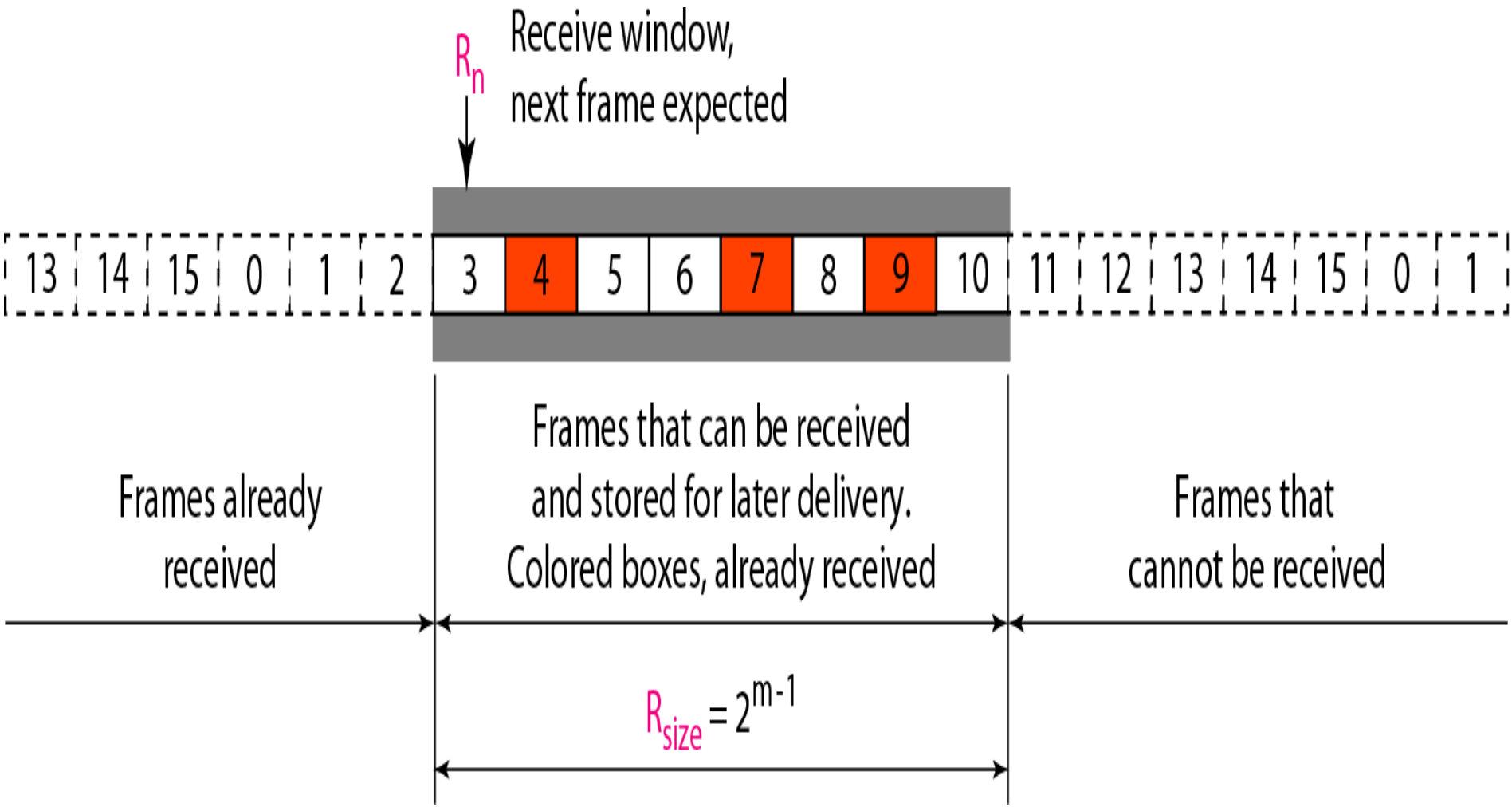
Selective Repeat Automatic Repeat Request

The size of the sender window is 2^{m-1}.

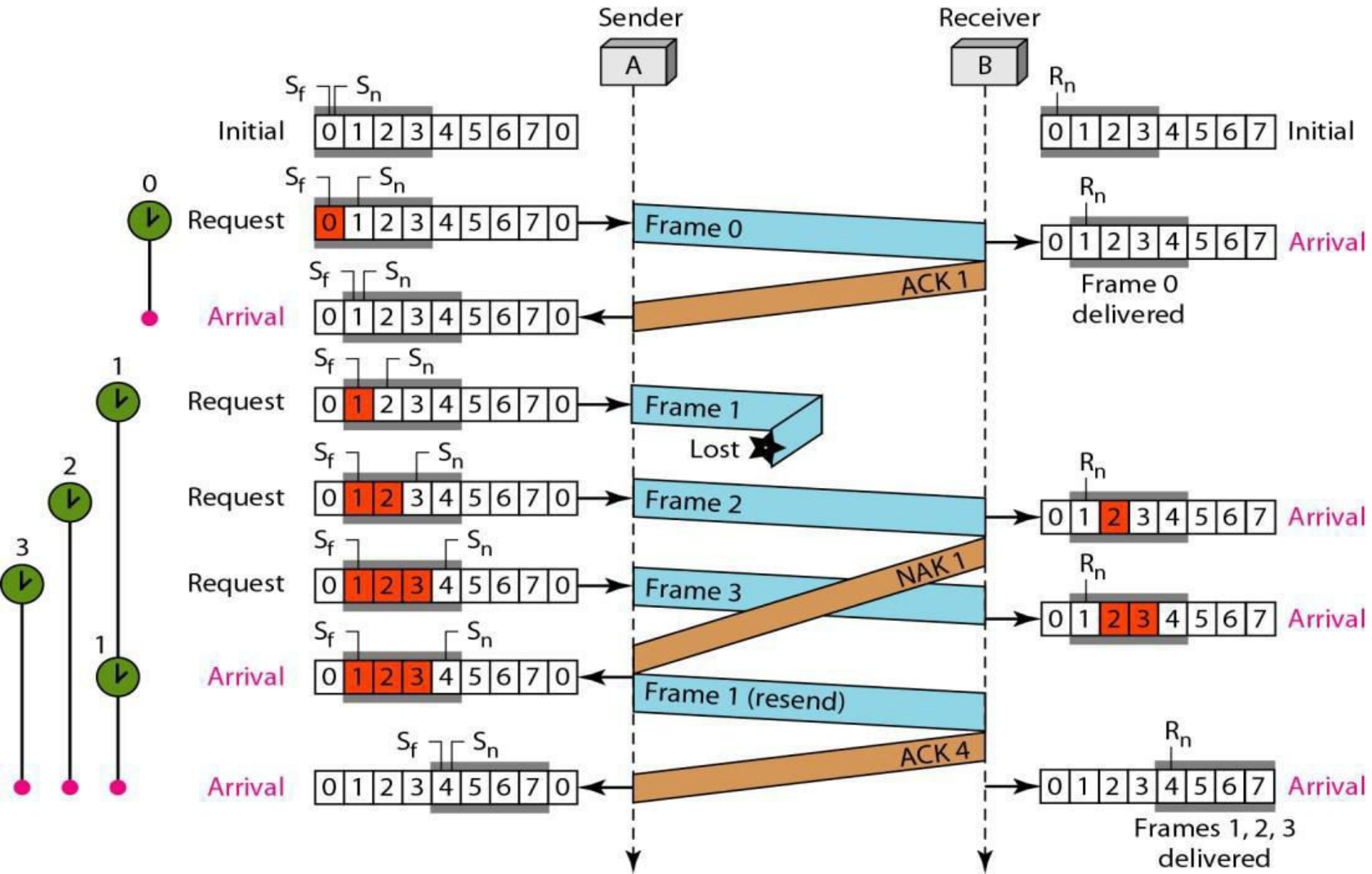
The size of the receiver window is 2^{m-1}.



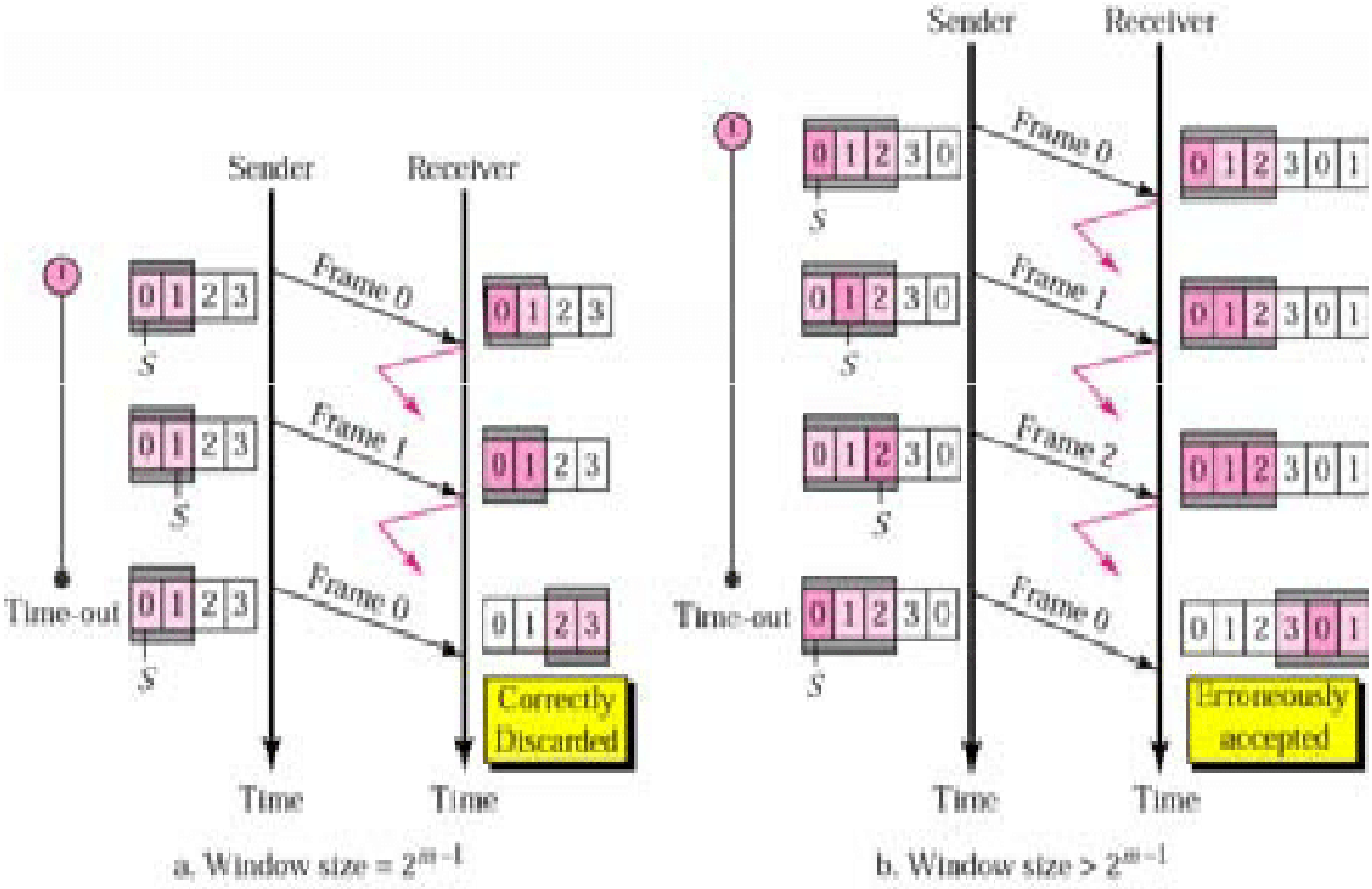
Selective Repeat Automatic Repeat Request



Selective Repeat Automatic Repeat Request



Selective Repeat Automatic Repeat Request



Selective Repeat Automatic Repeat Request

Efficiency:

The efficiency of selective repeat protocol is the same as of Go-Back-N ARQ protocol's efficiency.

$$\text{Efficiency} = N/(1+2a),$$

Where N is the size of sender window and $a = T_p/T_t$.

Piggybacking

A technique called piggybacking is used to improve the efficiency of the bidirectional protocols. When a frame is carrying data from A to B, it can also carry control information about arrived (or lost) frames from B; when a frame is carrying data from B to A, it can also carry control information about the arrived (or lost) frames from A.

Data Link Layer

Exercise

1. A sender sends a series of packets to the same destination using 5-bit sequence numbers. If the sequence number starts with 0, what is the sequence number after sending 100 packets?

2. Using 5-bit sequence numbers, what is the maximum size of the sender and receiver windows for each of the following protocols?

a. Stop-and-Wait ARQ

b. Go-Back-N ARQ

c. Selective-Repeat ARQ

Data Link Layer

3. A system uses the Stop-and-Wait ARQ Protocol. If each packet carries 1000 bits of data, how long does it take to send 1 million bits of data if the distance between the sender and receiver is 5000 Km and the propagation speed is 2×10^8 m? Ignore transmission, waiting, and processing delays. We assume no data or control frame is lost or damaged.

4. Repeat Exercise 3 using the Go-back-N ARQ Protocol with a window size of 7. Ignore the overhead due to the header and trailer.

5. Repeat Exercise 3 using the Selective-Repeat ARQ Protocol with a window size of 4. Ignore the overhead due to the header and the trailer.

6. Consider a selective repeat sliding window protocol uses a frame size of 1KB to send data on a 15Mbps link with a one-way latency of 50 ms. To achieve a link utilization of 60%, find the minimum number of bits required to represent the sequence number field.

7. Consider the sliding window flow-control protocol operating between a sender and a receiver over a full-duplex free link. Assume the following:

(i) The time take for processing the data frame by the receiver is negligible.

(ii) The time taken for processing the acknowledgement frame by the sender is negligible.

(iii) The sender has infinite number of frames available for transmission

(iv) The size of the data frame is 2,000 bits and the size of the acknowledgement frame is 10 bits.

(v) The link data rate in each direction is 1 Mbps (10^6 bits per second)

(vi) One way propagation delay of the link is 100 milliseconds

The minimum value of the sender's window size in terms of the number of frames, (rounded to the nearest integer) needed to achieve a link utilization of 50% is _____.

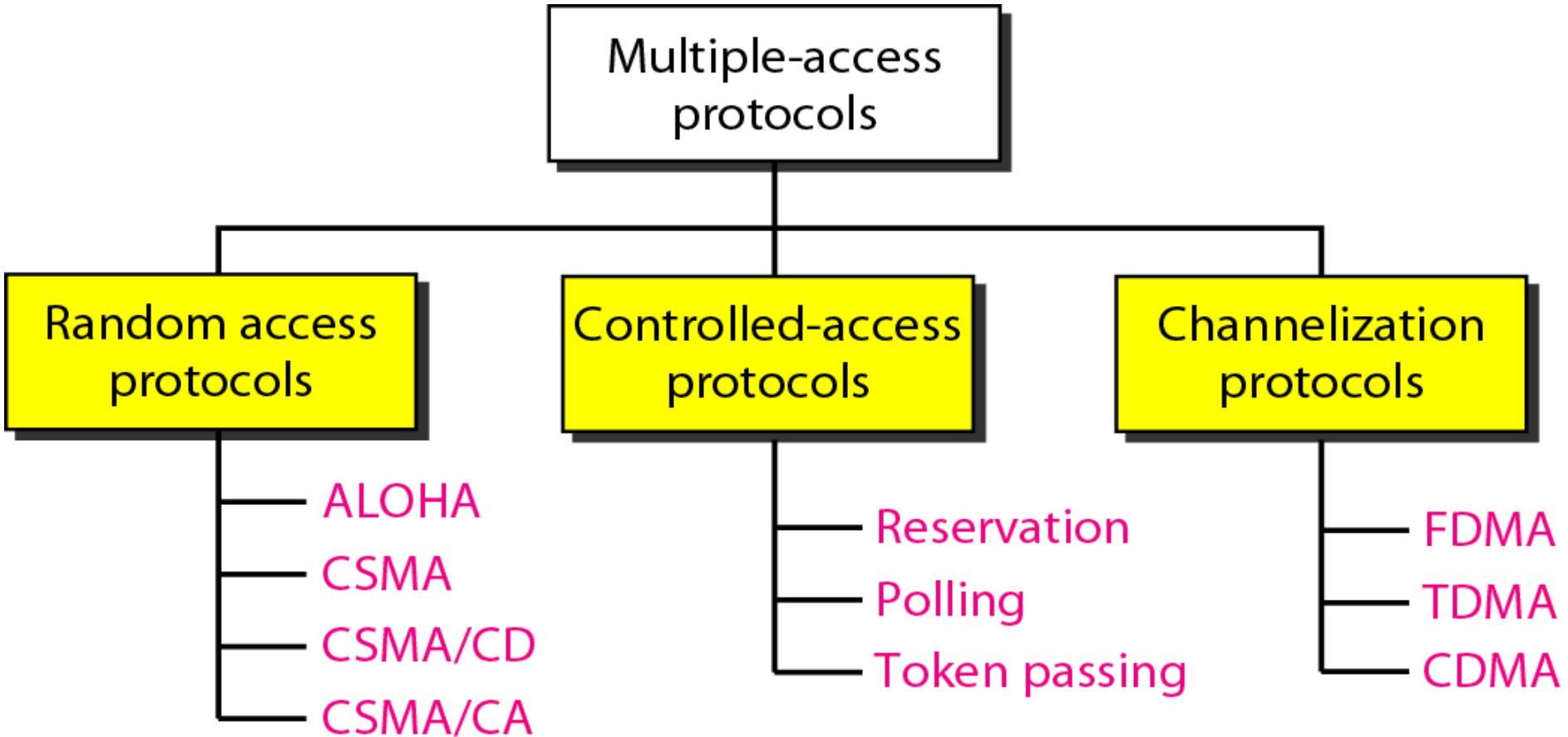
Media Access Control

Media Access Control

- Data link layer is considered as two sub-layers.
- The upper sub-layer is responsible for data link control. The upper sub-layer that is responsible for flow and error control is called the logical link control (LLC) layer.
- The lower sub-layer is responsible for resolving access to the shared media. The lower sub-layer that is mostly responsible for multiple access resolution is called the media access control (MAC) layer.

Media Access Control

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.



Random Access Protocol

- In random access or contention methods, no station is superior to another station and none is assigned the control over another.
- No station permits, or does not permit, another station to send.
- At each instance, a station that has data to send uses a procedure defined by the protocol to make a decision on whether or not to send. This decision depends on the state of the medium (idle or busy).
- If more than one station tries to send, there is an access conflict (collision) and the frames will be either destroyed or modified.

Random Access Protocol

To avoid access conflict or to resolve it when it happens, each station follows a procedure that answers the following questions:

- When can the station access the medium?
- What can the station do if the medium is busy?
- How can the station determine the success or failure of the transmission?
- What can the station do if there is an access conflict?

Random Access Protocol

ALOHA Protocol

ALOHA was the earliest random access method. There are two types of ALOHA.

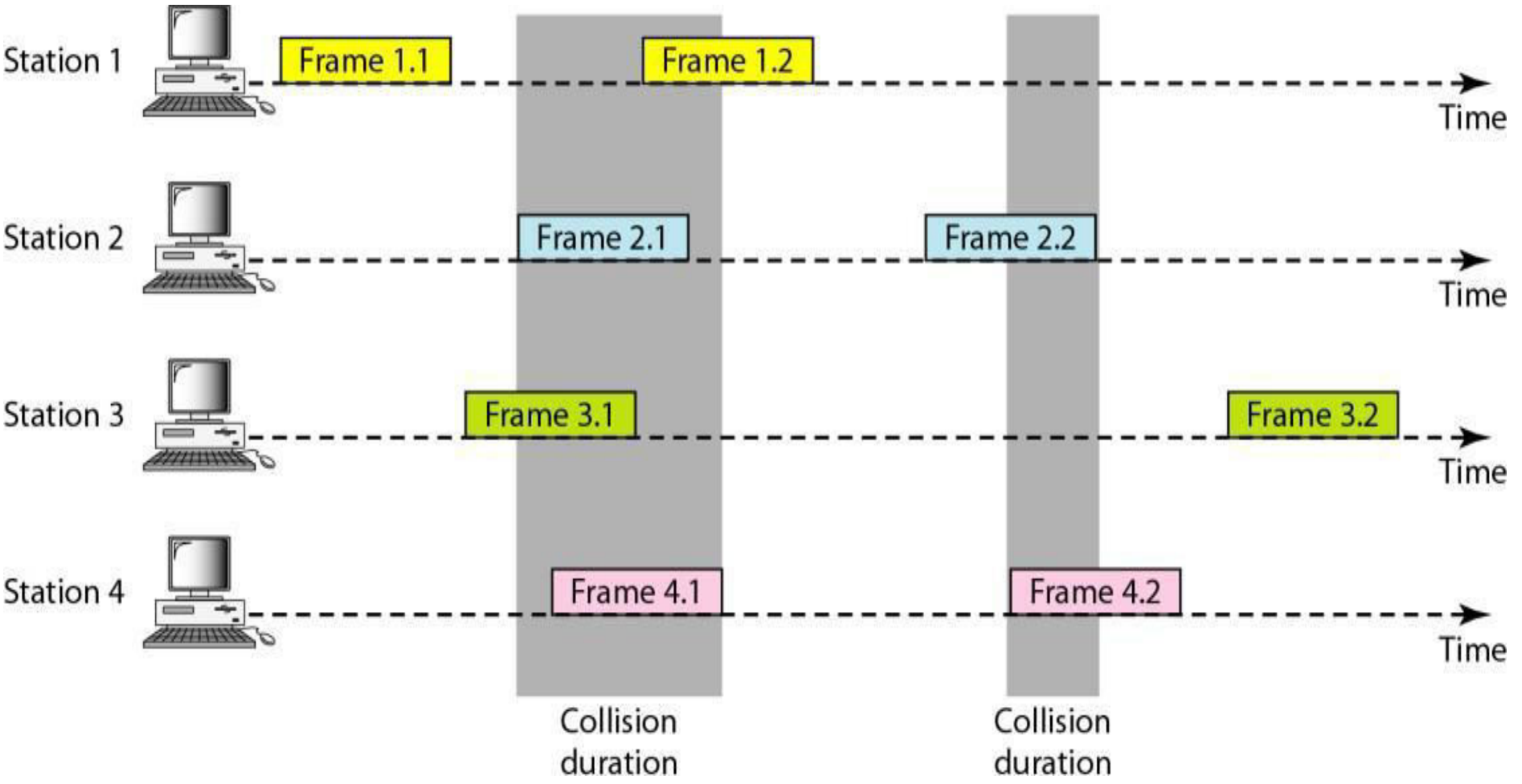
1. Pure ALOHA
2. Slotted ALOHA

Pure ALOHA

- The original ALOHA protocol is called pure ALOHA. This is a simple protocol.
- Full form of ALOHA is **Additive Links On-line Hawaii Area.**
- In this protocol each station sends a frame whenever it has a frame to send. Since there is only one channel to share, there is the possibility of collision between frames from different stations.

Pure ALOHA

Following figure shows an example of frame collisions in pure ALOHA.



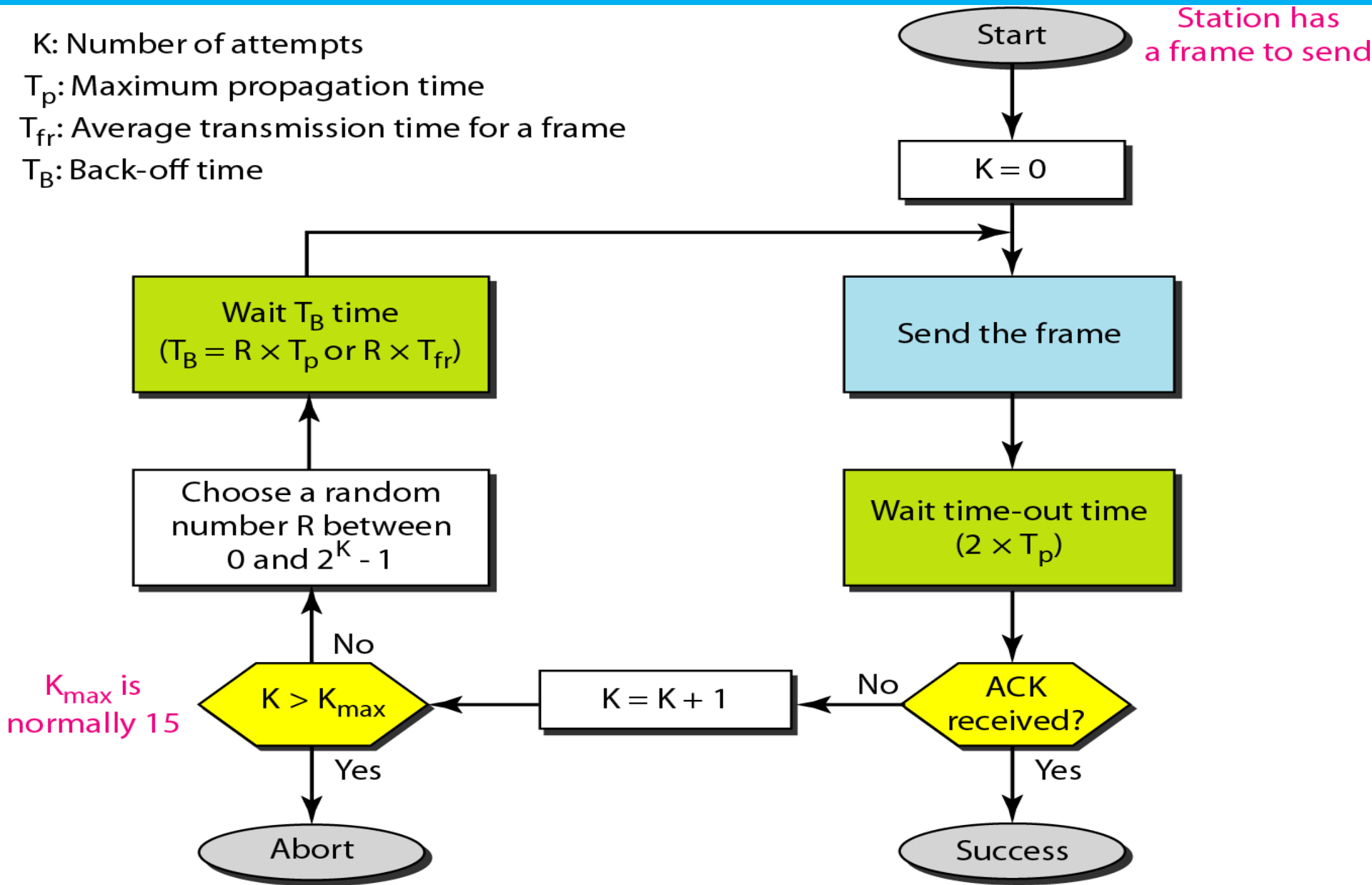
Pure ALOHA

- There are four stations that contend with one another for access to the shared channel.
- 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.
- Figure shows only two frames survive: frame 1.1 from station 1 and frame 3.2 from station 3.

Pure ALOHA

- The pure ALOHA protocol relies on acknowledgments from the receiver. When a station sends a frame, it expects the receiver to send an acknowledgment. If the acknowledgment does not arrive after a time-out period, the station assumes that the frame (or the acknowledgment) has been destroyed and resends the frame.
- A collision involves two or more stations. If all these stations try to resend their frames after the time-out, the frames will collide again. Pure ALOHA dictates that when the time-out period passes, each station waits a random amount of time before resending its frame. The randomness will help avoid more collisions. We call this time the back-off time T_B.
- After a maximum number of retransmission attempts Kmax' a station must give up and try later.

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
$K > K_{max}$
No
Yes
Abort
K_{max} is normally 15
Choose a random number R between 0 and $2^K - 1$
Wait T_B time ($T_B = R \times T_p$ or $R \times T_{fr}$)

Pure ALOHA

- The time-out period is equal to the maximum possible round-trip propagation delay i.e.

  $t_{out} = 2T_p$

  Where T_p is the propagation time between two most widely separated stations.

- The back-off time T_B is random value that normally depends on K, where K is the number of attempted unsuccessful transmissions.
- T_B is calculated by **binary exponential back-off** algorithm.

According to this algorithm, for each retransmission, a multiplier in the range 0 to $2^K - 1$ is randomly chosen and multiplied by T_p (maximum propagation time) or T_{fr} (the average time required to send out a frame) to find T_B.

- In this procedure, the range of the random numbers increases after each collision.
- The value of K_{max} is usually chosen as 15.

Pure ALOHA

Example: The stations on a wireless ALOHA network are a maximum of 600 km apart. Signals propagate with speed at 3×10^8 m/s. Find the value of T_B for different value of K.

Solution:

$$T_p = (600 \times 10^3) / (3 \times 10^8) = 2 \text{ ms}.$$

Now we can find the value of T_B for different values of K.

(a) For K = 1, the range is {0, 1}. The station needs to generate a random number with a value of 0 or 1. This means that T_B is either 0x2 = 0ms or 1x2 = 2ms, based on the outcome of the random variable.

(b) For K =2, the range is {0, 1, 2, 3}. This means that T_B can be 0, 2, 4, or 6 ms, based on the outcome of the random variable.

(c) For K =3, the range is to {0,1,2,3,4,5,6,7}. This means that T_B can be 0, 2, 4, ... , 14 ms, based on the outcome of the random variable.

Pure ALOHA

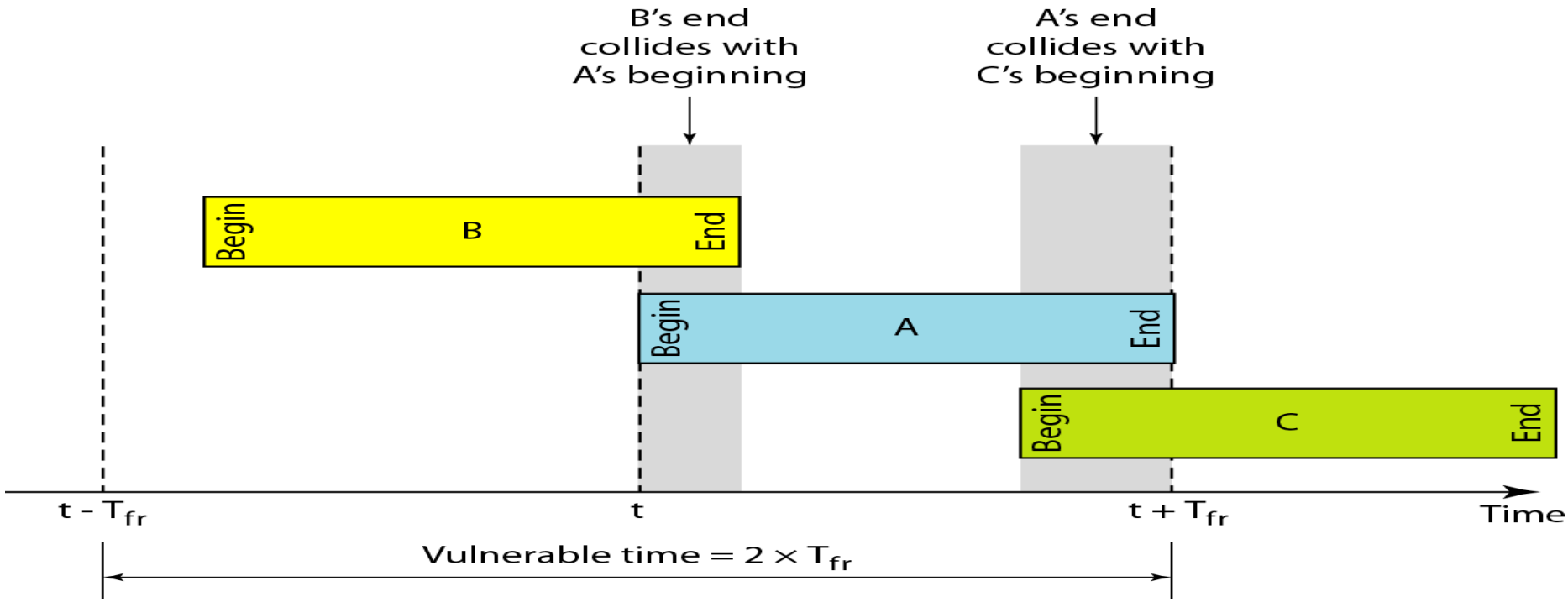
Vulnerable time

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 T_{fr} second to send.

This figure shows the vulnerable time for station A.

Pure ALOHA vulnerable time = 2 x T_{fr}



Pure ALOHA

Throughput

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

Average number of successful transmissions for pure ALOHA i.e. throughput, $\mathbf{S = G \times e^{-2G}}$.

The maximum throughput

$\mathbf{S_{max} = 0.184,}$ **for G = 1/2**.

Pure ALOHA

Example:

A pure ALOHA network transmits 200-bit frames on a shared channel of 200 kbps. What is the requirement to make this frame collision-free?

Example:

A pure ALOHA network transmits 200-bit frames on a shared channel of 200 kbps. What is the throughput if the system (all stations together) produces

a. 1000 frames per second

b. 500 frames per second

c. 250 frames per second

Pure ALOHA

Solution:

Average frame transmission time T_{fr} = 200 bits/200 kbps

= 1 ms.

Therefore, the vulnerable time = 2 x T_{fr} = 2 ms

This means no station should send later than 1 ms before this station starts transmission and no station should start sending during the 1ms period that this station is sending.

Pure ALOHA

Solution:

Here, $T_{fr} = 200/(200*10^{-3}) = 1$ ms

(a) Throughput $S = G * e^{-2G}$

Here, G is the average number of frames generated in frame transmission time.

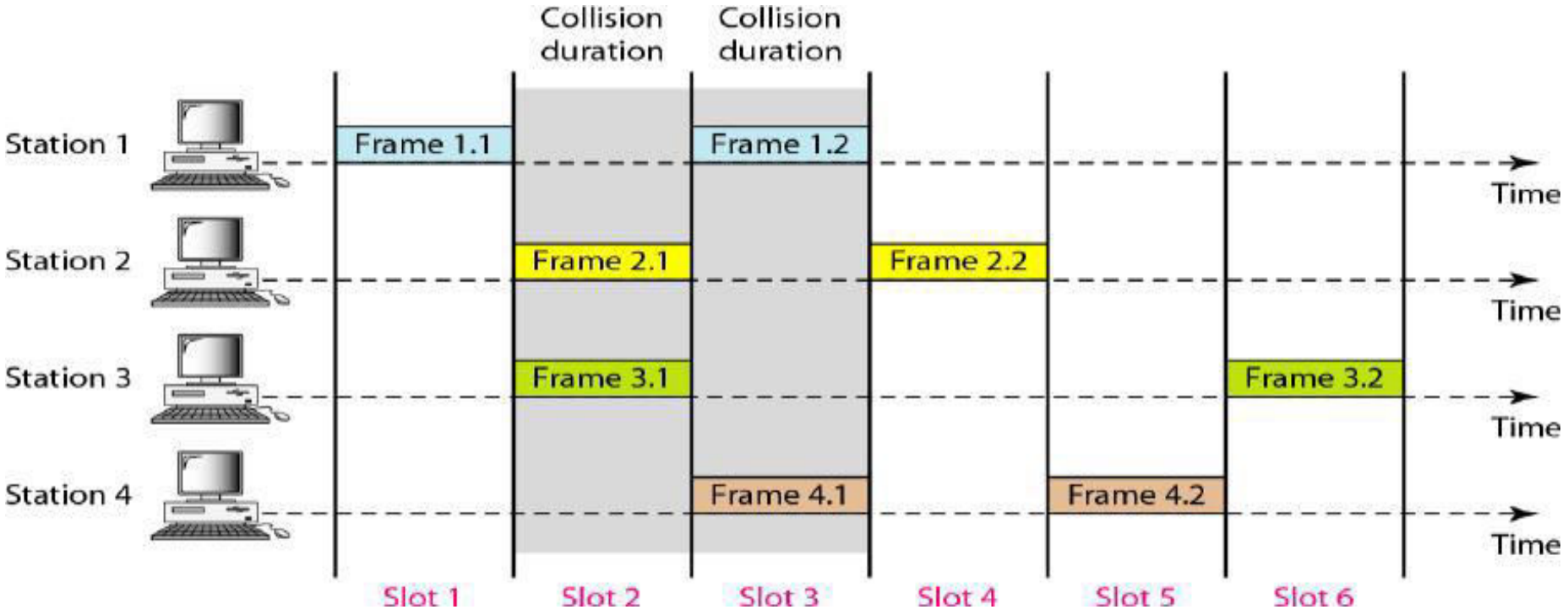
G = 1000/1000 = 1 frame

$S = 1*e^{-2*1} = e^{-2} = 0.135$ (13.5 percent)

This means that the throughput is 1000 X 0.135 =135 frames. Only 135 frames out of 1000 will probably survive.

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} and force the station to send only at the beginning of the time slot.
- Figure shows an example of frame collisions in slotted ALOHA.



Slotted ALOHA

- Since 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.
- There is still the possibility of collision if two stations try to send at the beginning of the same time slot.
- The vulnerable time is now reduced to one-half, equal to T_{fr}.
- Throughput, $S = G*e^{-G}$
- Maximum throughput $S_{max} = 0.368$, when $G = 1$.

Slotted ALOHA

Example:

A slotted ALOHA network transmits 200-bit frames on a shared channel of 200 kbps. What is the throughput if the system (all stations together) produces

a. 1000 frames per second

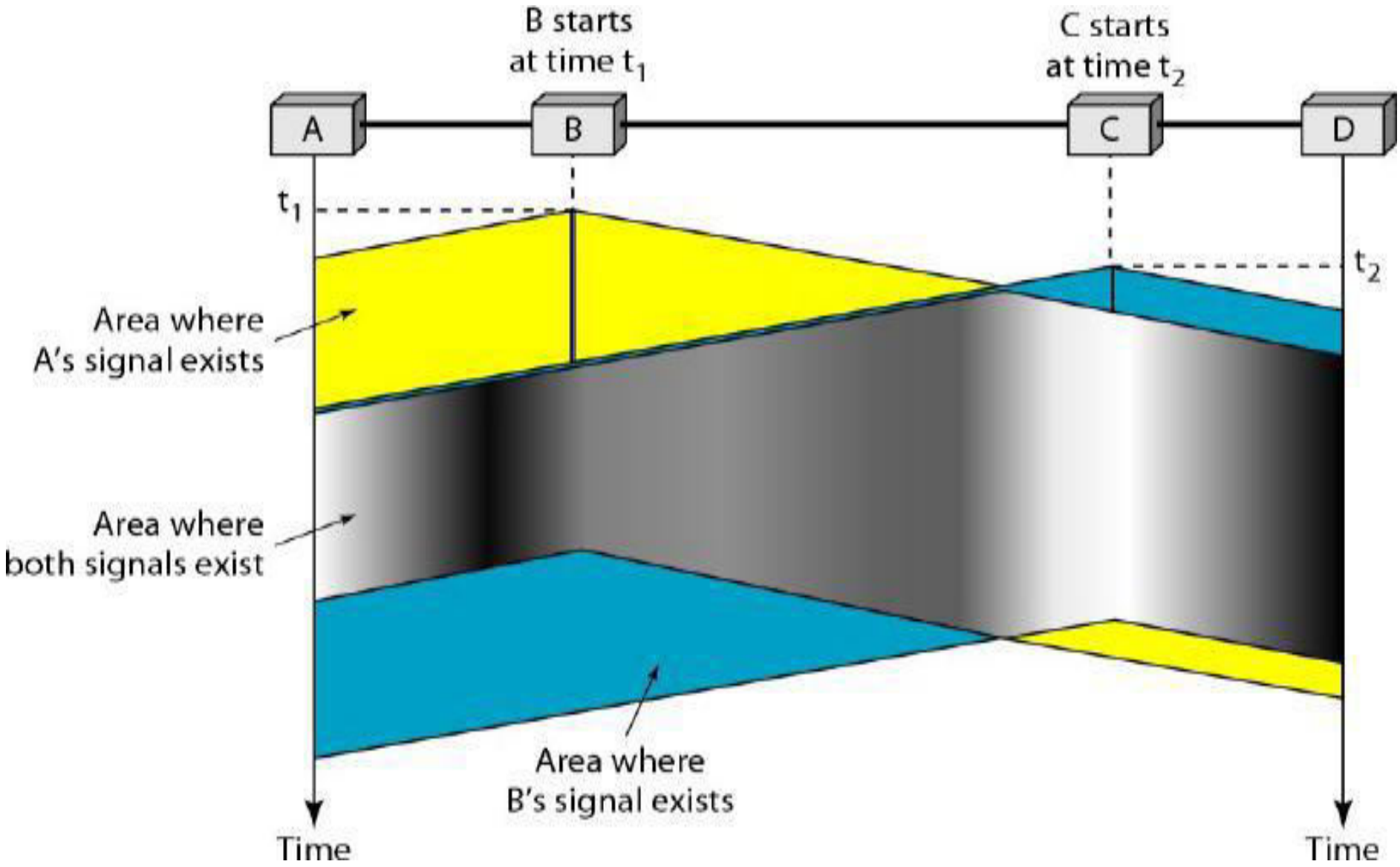
b. 500 frames per second

c. 250 frames per second

Carrier Sense Multiple Access (CSMA)

- To minimize the chance of collision and, therefore, increase the performance, the CSMA method was developed.
- Carrier sense multiple access (CSMA) requires that each station first listen to the medium (or check the state of the medium) before sending. In other words, CSMA is based on the principle "sense before transmit" or "listen before talk.“
- CSMA can reduce the possibility of collision, but it cannot eliminate it.
- The possibility of collision still exists because of propagation delay; when a station sends a frame, it still takes time for the first bit to reach every station and for every station to sense it.
- In other words, a station may sense the medium and find it idle, only because the first bit sent by another station has not yet been received.

Carrier Sense Multiple Access (CSMA)



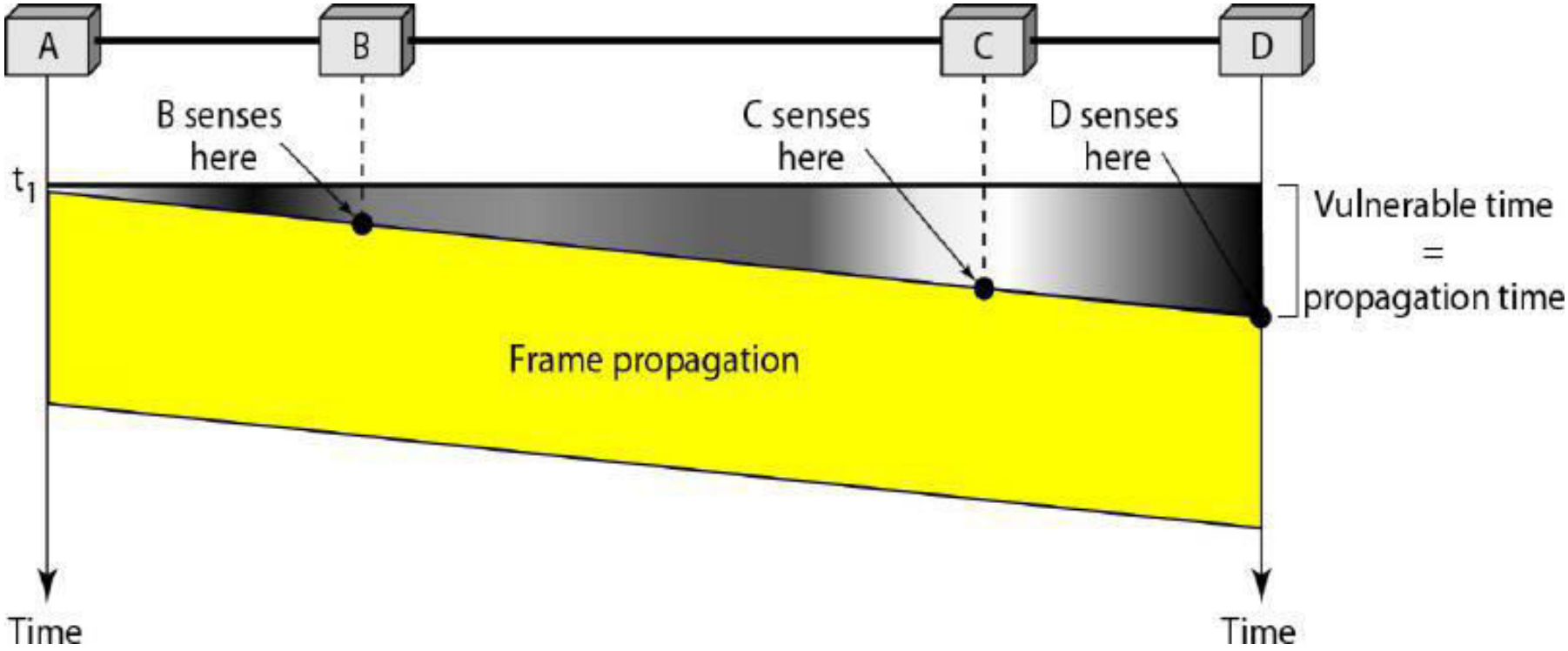
Carrier Sense Multiple Access (CSMA)

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.

Carrier Sense Multiple Access (CSMA)

Vulnerable Time

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



Carrier Sense Multiple Access (CSMA)

Persistence Methods

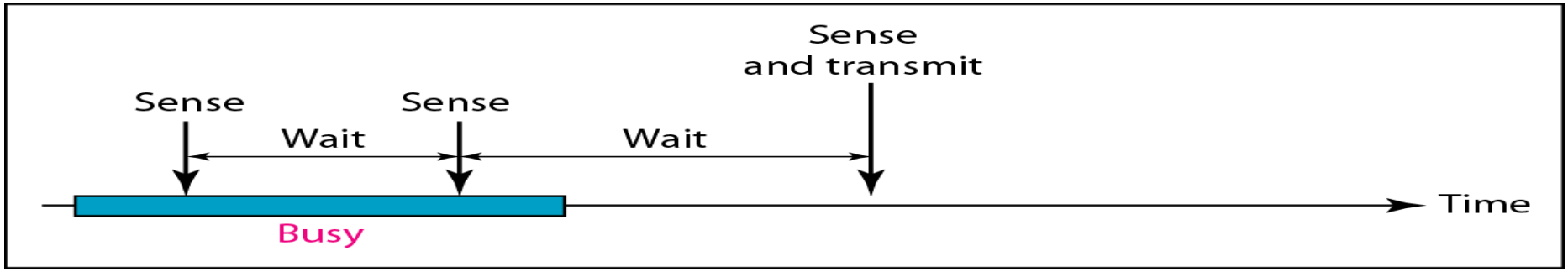
- What should a station do if the channel is busy?
- What should a station do if the channel is idle?

There are three methods to answer these questions:

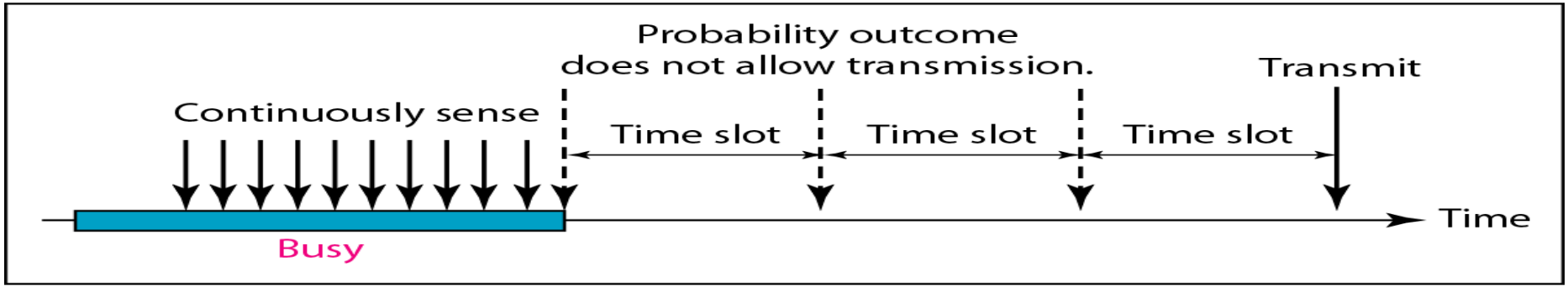
1. 1-persistent method
2. Non-persistent method
3. p-persistent method

Carrier Sense Multiple Access (CSMA)

a. 1-persistent



b. Nonpersistent



c. p-persistent

Carrier Sense Multiple Access (CSMA)

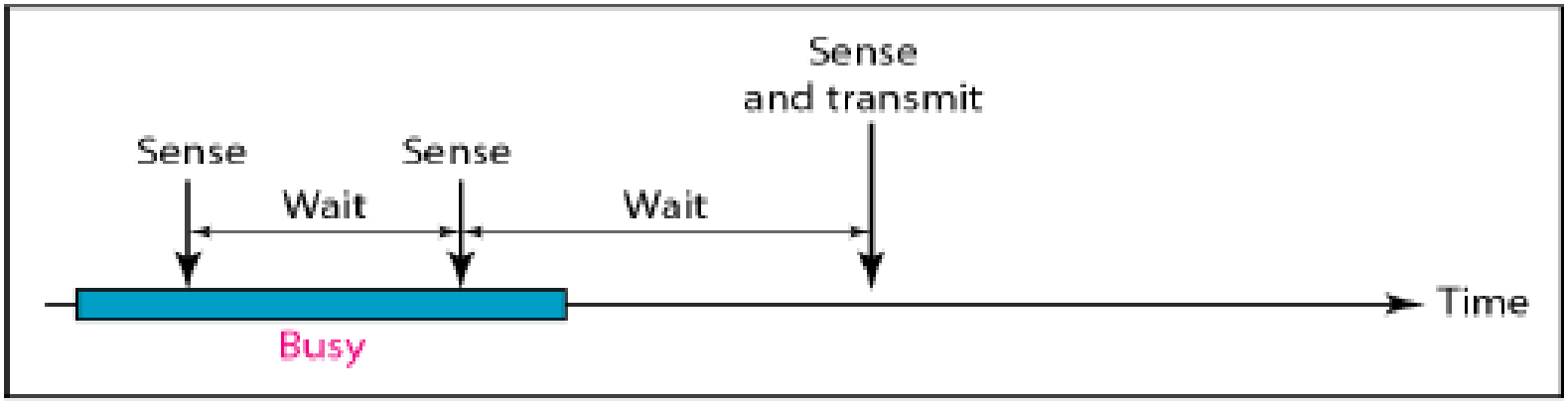
1-Persistent

a. 1-persistent

- In this method, after the station finds the line idle, it sends its frame immediately (with probability 1).
- This method has the highest chance of collision because two or more stations may find the line idle and send their frames immediately.

Carrier Sense Multiple Access (CSMA)

Non-persistent

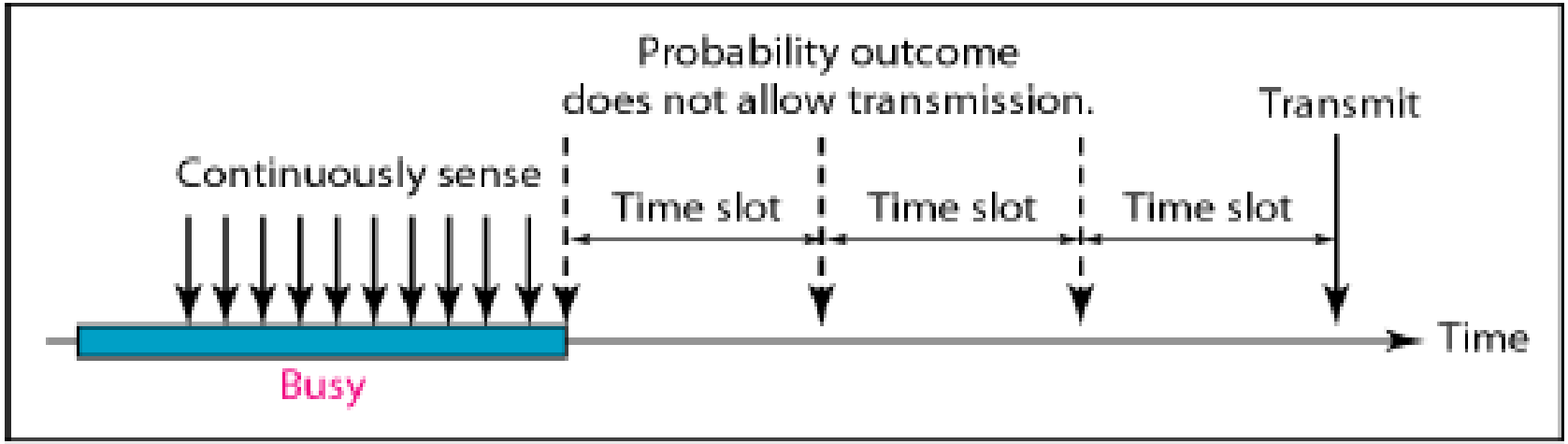


b. Nonpersistent

- In the non-persistent method, a station that has a frame to send senses the line. If the line is idle, it sends immediately. If the line is not idle, it waits a random amount of time and then senses the line again.
- The non-persistent approach reduces the chance of collision because it is unlikely that two or more stations will wait the same amount of time and retry to send simultaneously. However, this method reduces the efficiency of the network because the medium remains idle when there may be stations with frames to send.

Carrier Sense Multiple Access (CSMA)

p-Persistent



c. p-persistent

- The p-persistent method is used if the channel has time slots with a slot duration equal to or greater than the maximum propagation time.
- The p-persistent approach combines the advantages of the other two strategies.
- It reduces the chance of collision and improves efficiency.

Carrier Sense Multiple Access (CSMA)

In this method, after the station finds the line idle it follows these steps:

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.

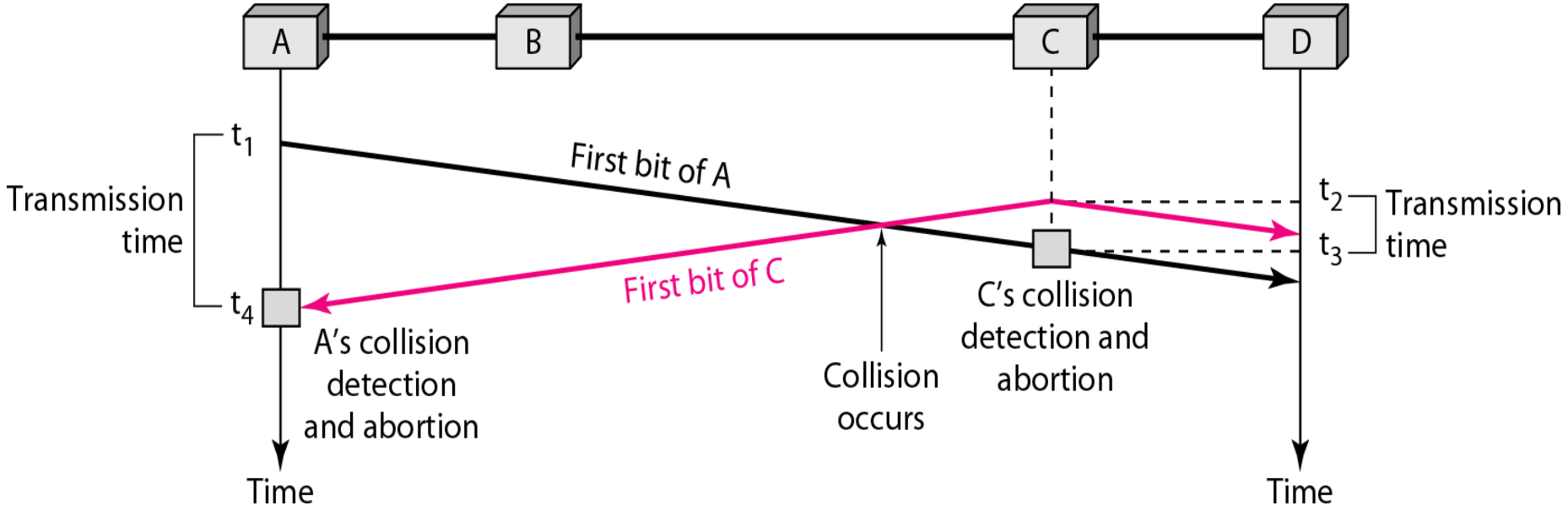
   a. If the line is idle, it goes to step 1.

   b. If the line is busy, it acts as though a collision has occurred and uses the backoff procedure.

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

- The CSMA method does not specify the procedure following a collision.
- CSMA/CD augments the algorithm to handle the collision.
- In this method, a station monitors the medium after it sends a frame to see if the transmission was successful. If so, the station is finished. If, however, there is a collision, the frame is sent again.

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



- In this figure, stations A and C are involved in the collision.
- 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.

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

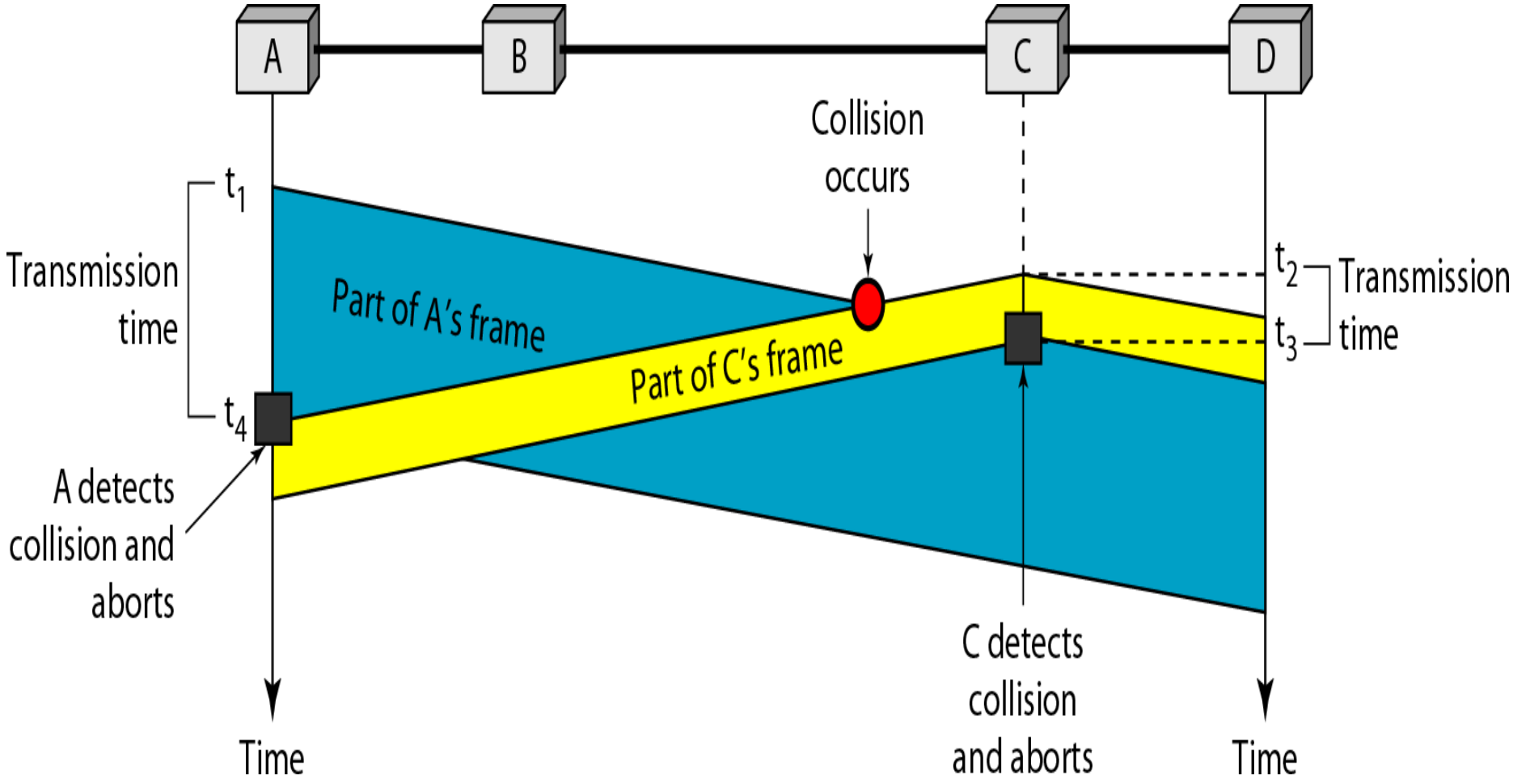
Minimum Frame Size

- For CSMA/CD to work, we need a restriction on the frame size.
- Before sending the last bit of the frame, the sending station must detect a collision, if any, 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 for collision detection.
- Therefore, the frame transmission time T_{fr} must be at least two times the maximum propagation time T_p i.e.

$$T_{fr} \geq 2T_p$$

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

Minimum Frame Size



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

Example:

A network using CSMA/CD has a bandwidth of 10 Mbps. If the maximum propagation time is 25.6 μs, what is the minimum size of the frame?

Solution:

Let the frame size is x.

In CSMA/CD, we know that Tfr ≥ 2Tp,

Therefore, $x/(10*10^6) \geq 2*25.6*10^{-6}$

$x \geq 2*10*25.6*10^6 * 10^{-6} = 512$

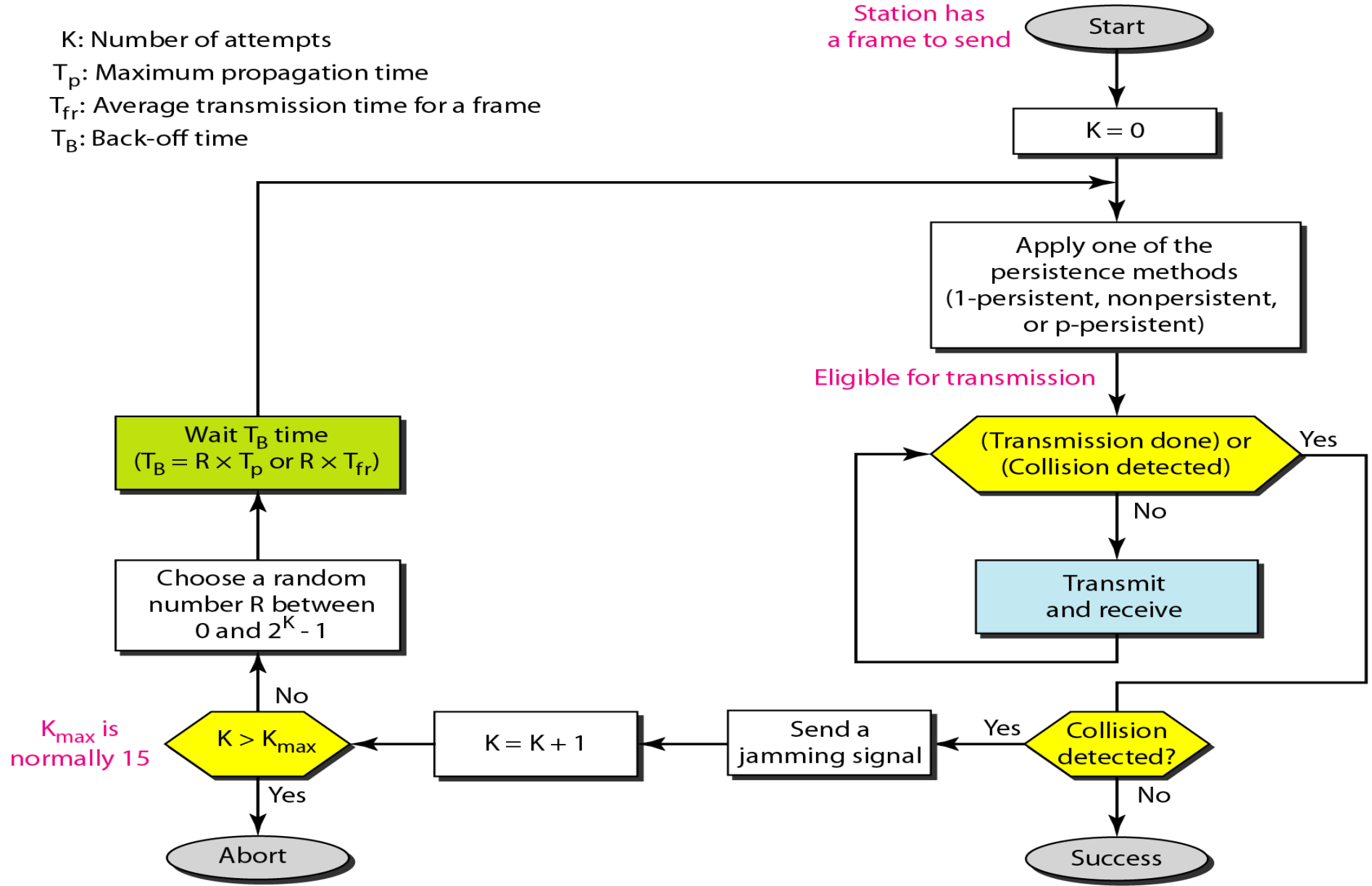
Therefore, minimum size of the frame = 512 bits

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

Throughput

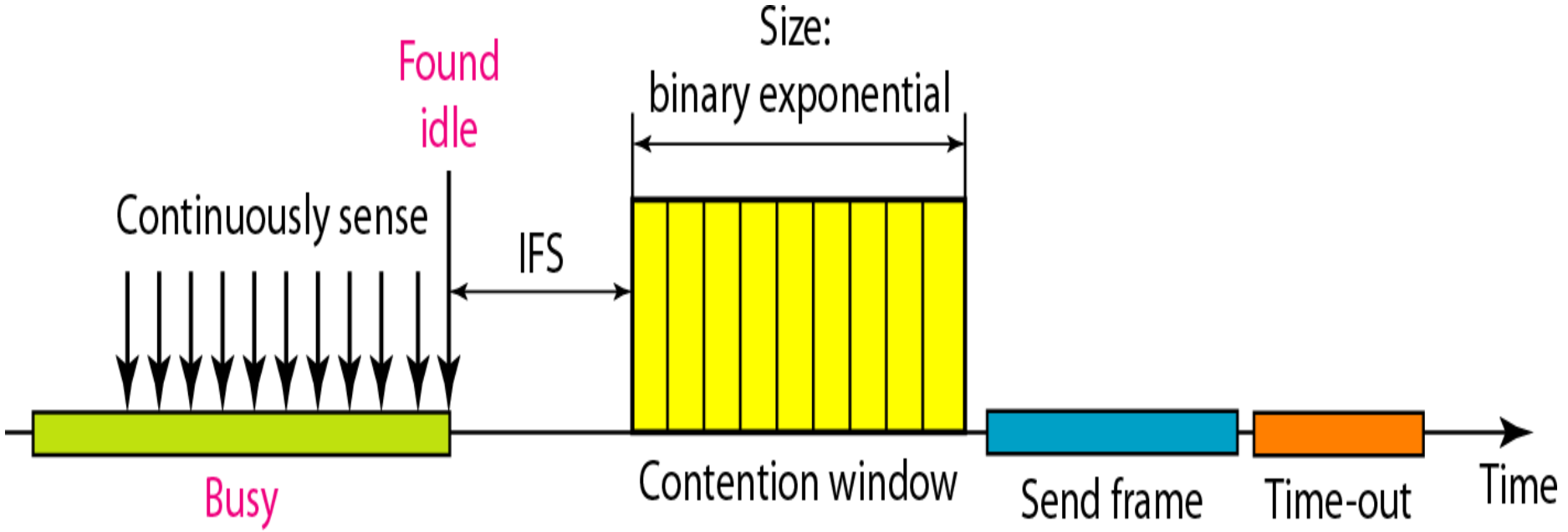
- The throughput of CSMA/CD is greater than that of pure or slotted ALOHA.
- The maximum throughput occurs at a different value of G and is based on the persistence method and the value of p in the p-persistent approach.
- For 1-persistent method the maximum throughput is around 50 percent when G=1.
- For non-persistent method, the maximum throughput can go up to 90 percent when G is between 3 and 8.

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



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

- CSMA/CA was invented for wireless network.
- Collisions are avoided through the use of CSMA/CA's three strategies: the interframe space, the contention window, and acknowledgments.



Carrier Sense Multiple Access with Collision Avoidance (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.

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 variable can also be used to define the priority of stations or frames.

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

Contention Window

- 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 according to the binary exponential back-off strategy.

- The station senses the channel after each time slot. However, if the station finds the channel busy, it does not restart the process; it just stops the timer and restarts it when the channel is sensed as idle.

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

Acknowledgment

With all these precautions, there still may be a collision resulting in destroyed data.

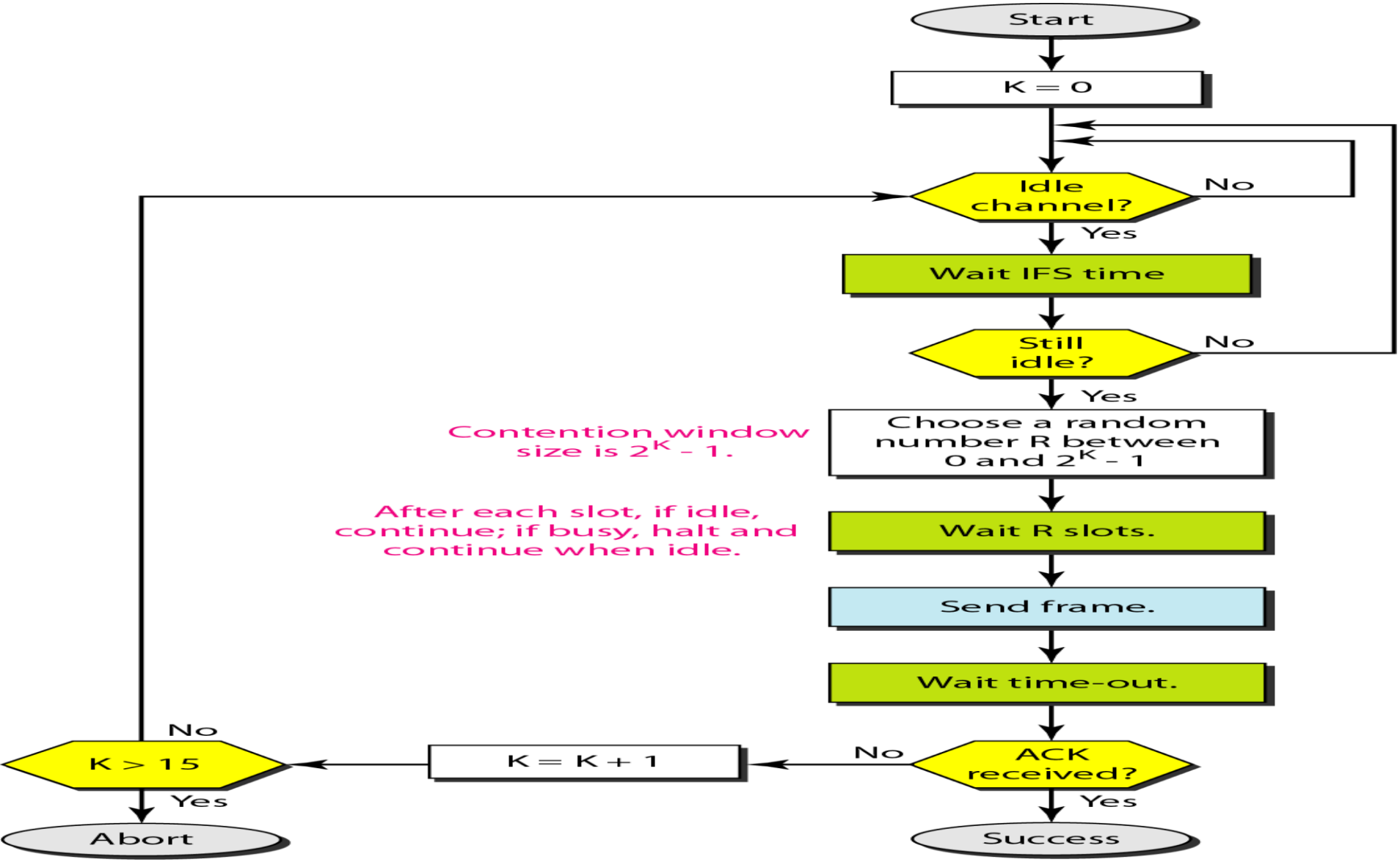
In addition, the data may be corrupted during the transmission. The positive acknowledgment and the time-out timer can help guarantee that the receiver has received the frame.

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

Procedure

- The channel needs to be sensed before and after the IFS.
- The channel also needs to be sensed during the contention time.
- For each time slot of the contention window, the channel is sensed. If it is found idle, the timer continues; if the channel is found busy, the timer is stopped and continues after the timer becomes idle again.

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

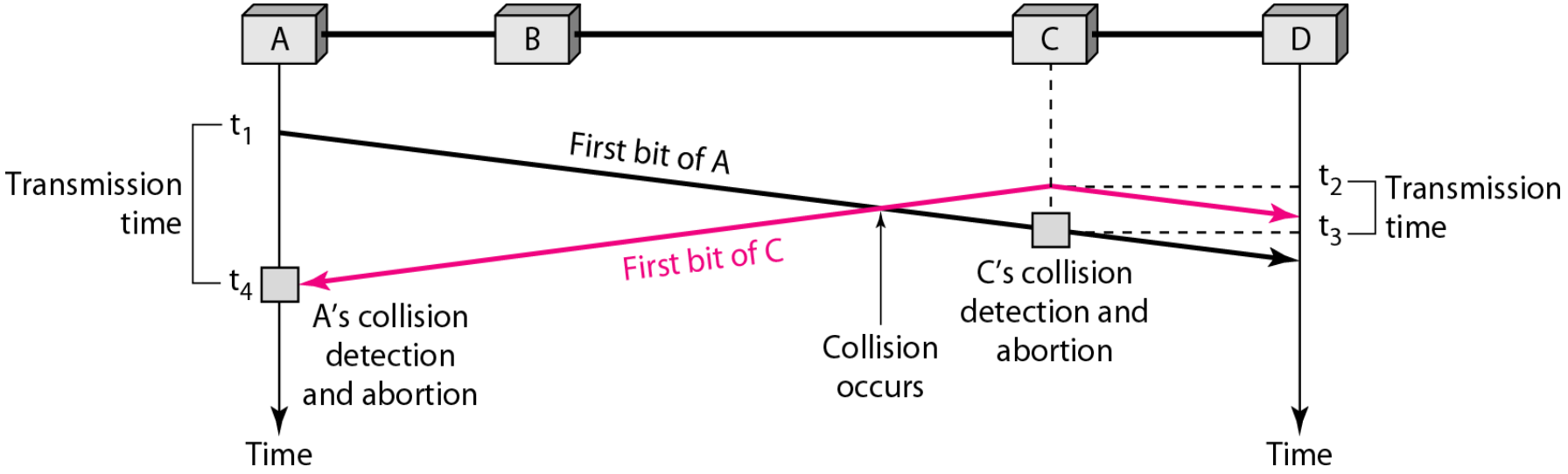


Exercise

1. In a CSMA/CD network with a data rate of 10 Mbps, the minimum frame size is found to be 512 bits for the correct operation of the collision detection process. What should be the minimum frame size if we increase the data rate to 100 Mbps? To 1 Gbps? To 10 Gbps?
2. One hundred stations on a pure ALOHA network share a 1-Mbps channel. If frames are 1000 bits long, find the throughput if each station is sending 10 frames per second.

Exercise

3.



The data rate is 10 Mbps, the distance between station A and C is 2000 m, and the propagation speed is 2 x 10^8 m/s. Station A starts sending a long frame at time t_1 = 0; station C starts sending a long frame at time t_2 =3µs. The size of the frame is long enough to guarantee the detection of collision by both stations. Find:

a. The time when station C hears the collision (t_3).

b. The time when station A hears the collision (t_4).

c. The number of bits station A has sent before detecting the collision.

d. The number of bits station C has sent before detecting the collision.

IEEE 802 STANDARDS

- IEEE 802 is a collection of networking standards that cover the physical and data-link layer specifications for technologies such as Ethernet and wireless. These specifications apply to local area networks (LAN) and metropolitan area networks (MAN).
- IEEE stands for Institute of Electrical and Electronics Engineers.

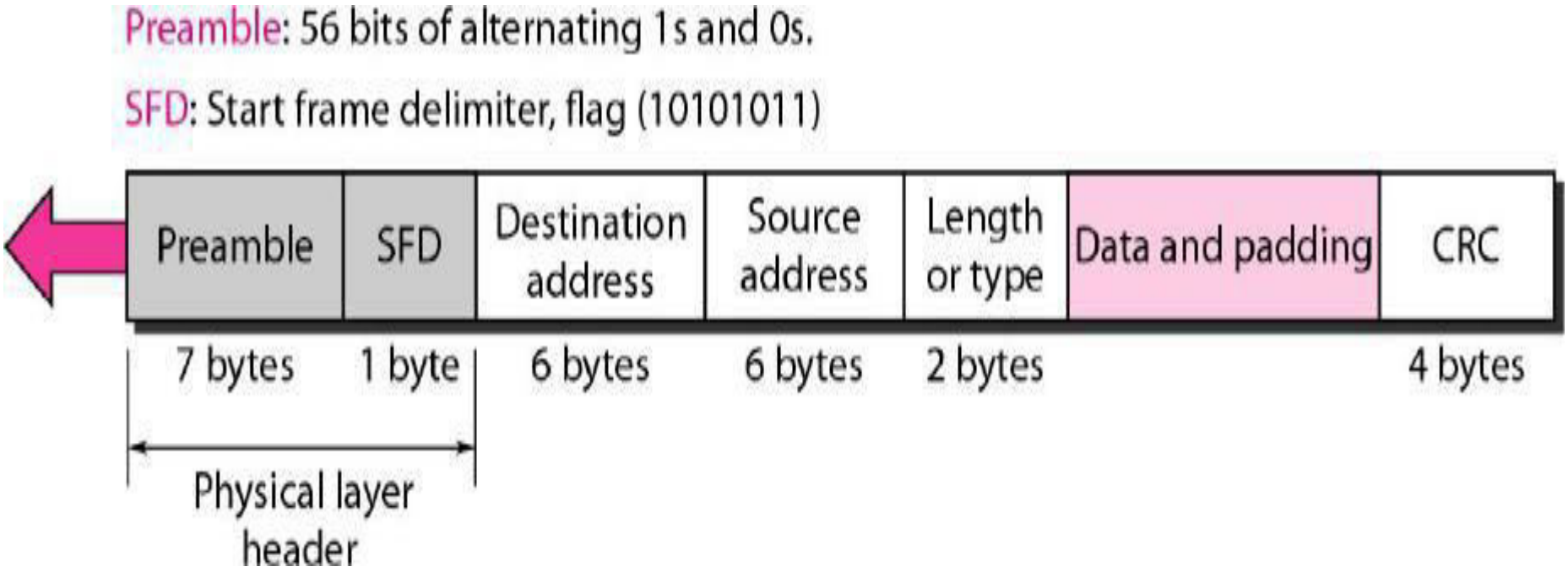
Ethernet(802.3 standard)

It uses CSMA/CD protocol to access the medium.

Frame Format

The Ethernet frame contains seven fields: preamble, SFD, DA, SA, length or type of protocol data unit (PDU), upper-layer data, and the CRC.

IEEE 802 STANDARDS



Preamble

This is the first field. It contains 7-bytes of alternating 0's and 1's that alerts the receiving system to the coming frame and enables it to synchronize its input timing. The preamble is actually added at the physical layer and is not part of the frame.

IEEE 802 STANDARDS

Start frame delimiter (SFD)

The second field (1 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.

Destination address (DA)

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

Source address (SA)

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

IEEE 802 STANDARDS

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.

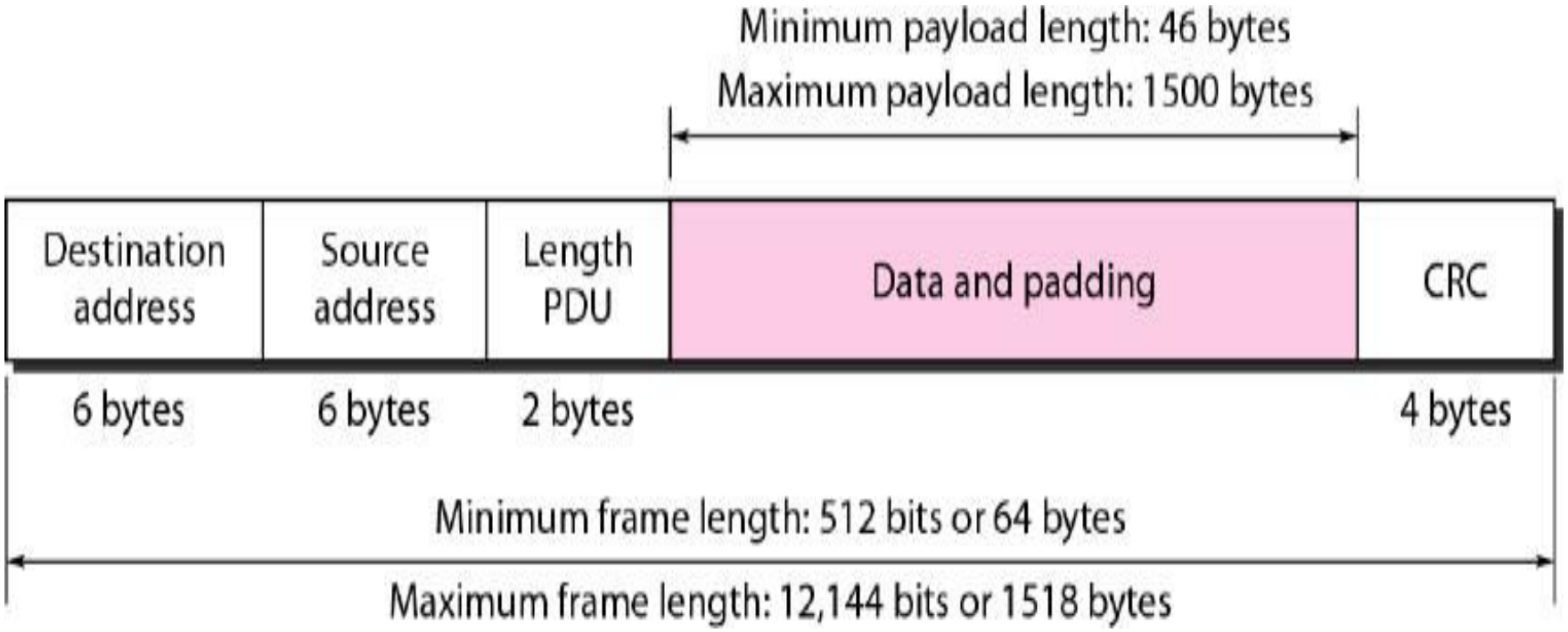
CRC

The last field contains error detection information. It is of 4 bytes.

IEEE 802 STANDARDS

Frame Length

Ethernet has imposed restrictions on both the minimum and maximum lengths of a frame, as shown in below Figure:-



IEEE 802 STANDARDS

Addressing

Each station on an Ethernet network has its own network interface card (NIC). The NIC fits inside the station and provides the station with a 6-byte physical address. It is written in hexadecimal notation, with a colon between the bytes.

06:01:02:01:2C:4B

Unicast, Multicast, and Broadcast Addresses

A source address is always a unicast address.

The destination address can be unicast, multicast, or broadcast.

If the least significant bit of the first byte in a destination address is 0, then the address is unicast; otherwise, it is multicast.

The broadcast address is a special case of the multicast address. In this case, the recipients are all the stations on the LAN. A broadcast destination address is forty eight 1's.

IEEE 802 STANDARDS

Example

Define the type of the following destination addresses:

a) 4A:30:10:21:10:1A

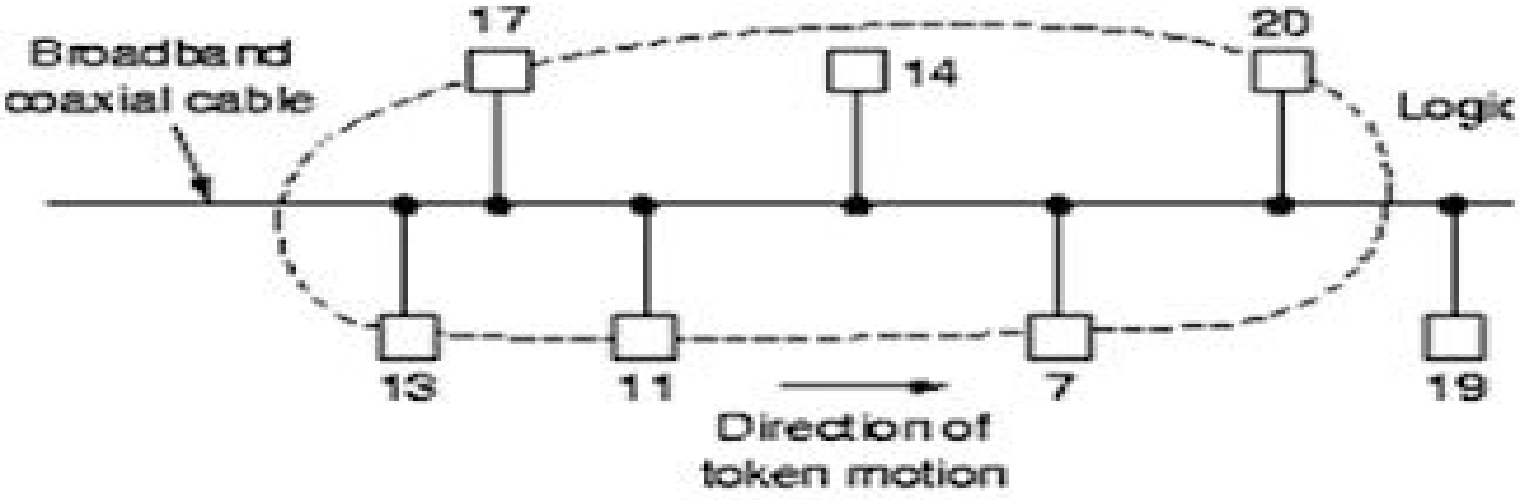
b) 47:20:1B:2E:08:EE

c) FF:FF:FF:FF:FF:FF

IEEE 802 STANDARDS

IEEE 802.4 (Token Bus)

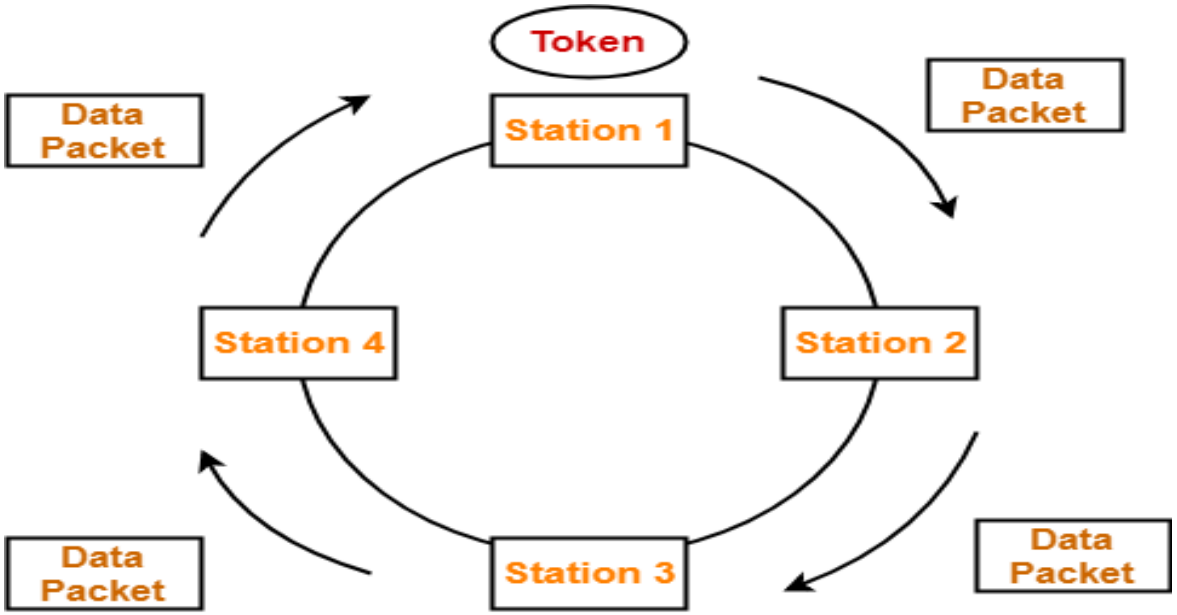
- IEEE specifications include physical layer and media access control sublayer for network that uses a bus topology and use token passing as the media access method.
- In this, all nodes are connected in a logical ring. It supports electrical(coaxial) and fiber optic cable.



IEEE 802 STANDARDS

IEEE 802.5 (Token Ring)

- A token ring network consists of a set of nodes connected in a ring. Data always flows in a particular direction around the ring.
- It uses token passing as the media access method.



Delayed Token Reinsertion Token Passing

IEEE 802 STANDARDS

Fiber Distributed Data Interface(FDDI)

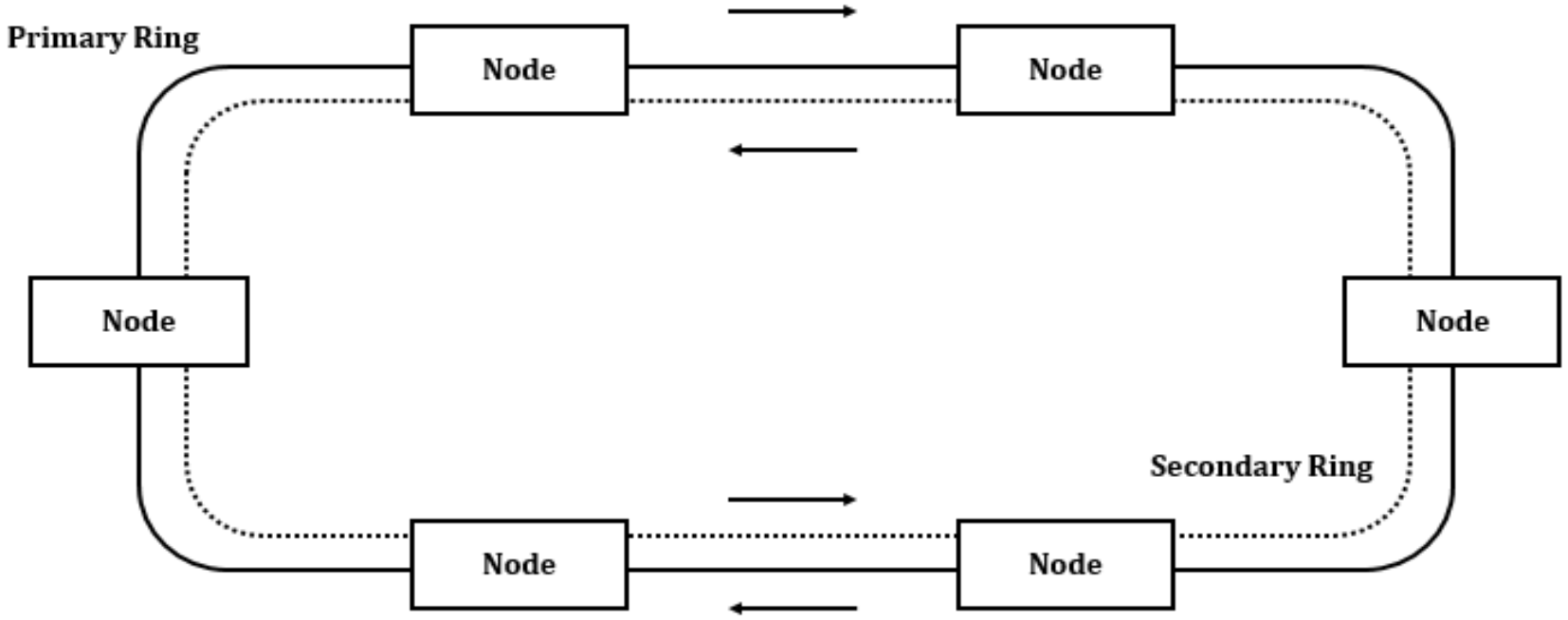
Fiber Distributed Data Interface (FDDI) is a set of ANSI and ISO standards for transmission of data in local area network (LAN) over fiber optic cables. It is applicable in large LANs that can extend up to 200 kilometers in diameter.

Features

- FDDI uses optical fiber as its physical medium.
- It operates in the physical and medium access control (MAC layer) of the Open Systems Interconnection (OSI) network model.
- It provides high data rate of 100 Mbps and can support thousands of users.
- It is used in LANs up to 200 kilometers for long distance voice and multimedia communication.

IEEE 802 STANDARDS

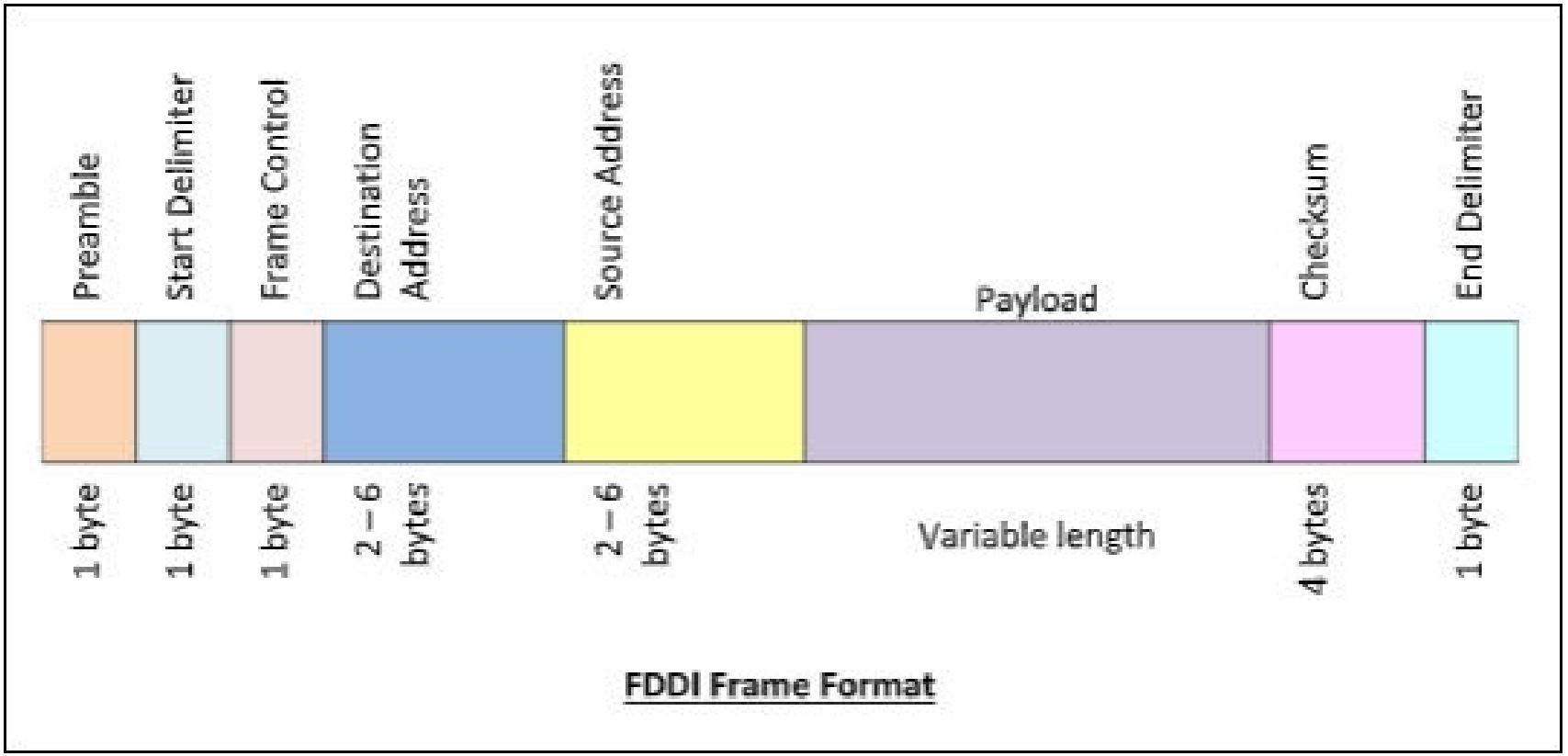
- It uses ring based token passing mechanism.
- It contains two token rings, a primary ring for data and token transmission and a secondary ring that provides backup if the primary ring fails.
- FDDI technology can also be used as a backbone for a wide area network (WAN).



IEEE 802 STANDARDS

Frame Format

The frame format of FDDI is similar to that of token bus as shown in the following diagram –



FDDI Frame Format

IEEE 802 STANDARDS

AKTU Examination Questions

1. A bit string 0001111111001111101000 needs to be transmitted at the data link layer. What is the string actually transmitted after bit stuffing?
2. Explain the working of pure ALOHA and slotted ALOHA protocols. How slotted ALOHA improve the performance of pure ALOHA?
3. List different carrier sense protocols. How CSMA/CD protocol is different from other CSMA/CA protocol?
4. What is piggybacking?
5. Measurement of slotted ALOHA channel with infinite number of users such that the 10 percent of slots are idle.

   (i) What is the channel load?

   (ii) What is the throughput?

AKTU Examination Questions

6. If a binary signal is sent over a 3KHZ channel. Whose signal to noise ratio is 20db. What is the maximum achievable data rate?
7. Discuss the issues in the data link layer and about its protocol on the basis of layering principle.
8. Discuss different carrier sense protocols. How are they different than collisions protocols?
9. Write short notes on following:

   i. Stop and Wait ARQ

   ii. Sliding Window Protocol

   iii. Go Back N ARQ

AKTU Examination Questions

10. An ALOHA network uses 9.2 kbps channel for sending message packets of 100 bits long size. Calculate the maximum throughput for pure ALOHA network.
11. What is the total delay (latency) for a frame size of 10 million bits that is being set up on link with 15 routers, each having queuing time of 2μs and a processing time of 1μs? The length of link is 3000km The speed of light inside the link is $2x10^8$ m/sec. The link has bandwidth of 6 Mbps.
12. What is hamming code? Explain its working with suitable example.
13. What are header and trailers and how do they get added and removed?

AKTU Examination Questions

14. A large FDDI ring has 100 stations & a token rotation time of 40 msec. The token holding time is 10 msec. What is the maximum achievable efficiency of the ring?
15. A channel has a bit rate of 20 kbps. The stop and wait protocol with frame size 4500 bits is used. The delay for error detection and sending ACK by the receiver is 0.25 seconds because of a fault. Find the maximum efficiency of the channel if the destination is 30000 km away and the speed of the propagation of the signal is $2.8X10^8$ m/s. Find the decrease in efficiency due to the fault.
16. A slotted ALOHA network transmits 400-bit frames on a shared channel of 400 kbps. What is the throughput if the system (all stations together) produces –

    (i) 1000 frames per second

    (ii) 500 frames per second

    (iii) 250 frames per second

AKTU Examination Questions

17. Explain ARQ Error Control technique, in brief.
18. Compare ALOHA with slotted ALOHA.
19. State the requirements of CRC.
20. Discuss the issues in the data link layer and about its protocol on the basis of layering principle.
21. Consider the use of 10 K-bit size frames on a 10 Mbps satellite channel with 270 ms delay. What is the link utilization for stop-and-wait ARQ technique assuming $P=10^{-3}$?
22. Brief about how line coding implemented in FDDI and describe its format.
23. Illustrate the performance issues for GO-BACK-N data link protocol.