CENG210
Chapter 05
Data Link Layer

UAEU College of Information Technology

Objectives

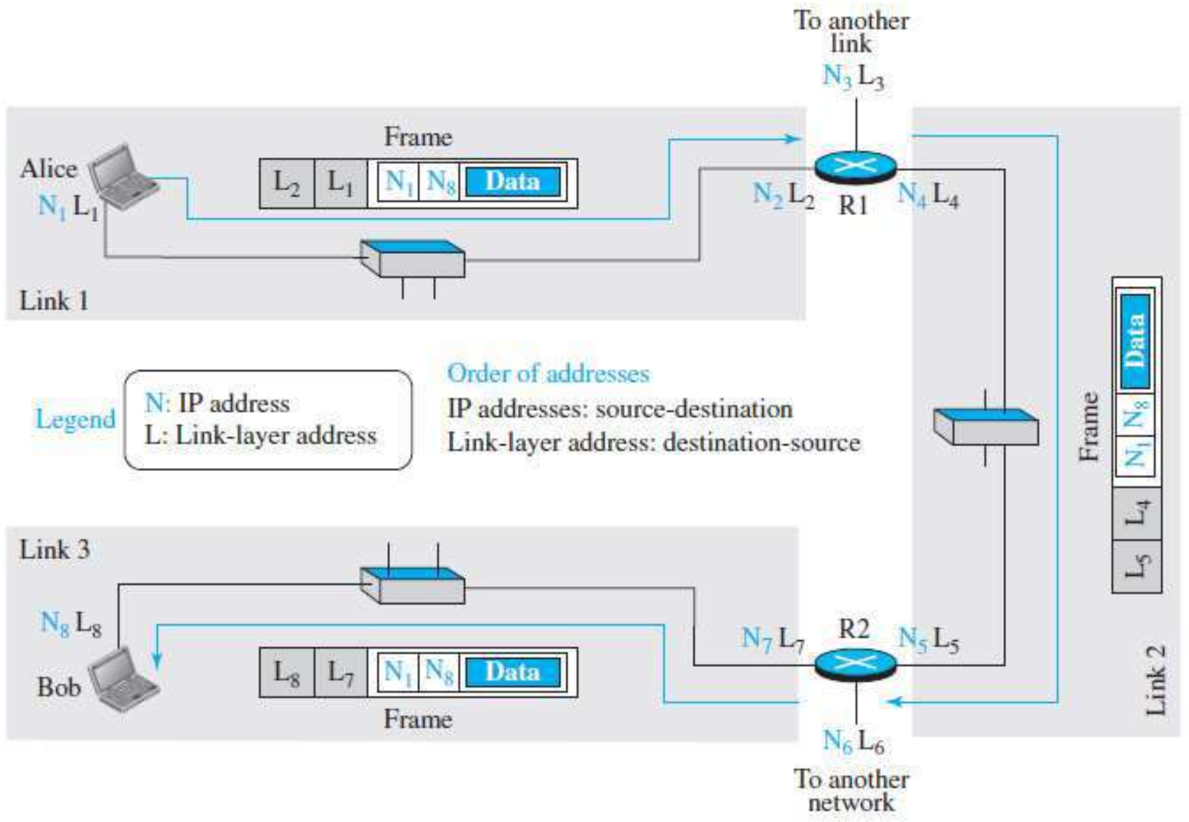
- Provide an overview of the Data Link Layer and its functions including:
  - Error Detection and Correction
  - Flow Control and
  - Medium Access Control

- Material feeds into CLO-2
  - Apply different mechanisms of error control, flow control and medium access control at the data link layer

Data Link Layer Services

- The data-link layer is located between the physical and the network layers
- The data link layer provides services to the network layer; it receives services from the physical layer
- Data link layer is responsible for the following:
  - Framing
  - Physical addressing
  - Flow control
  - Error control
  - Access control

Data Link Layer Service



If frames are to be distributed to different systems on the network, the data link layer adds a header to the frame to define the sender and/or receiver of the frame.

If the frame is intended for a system outside the sender's network, the receiver address is the address of the device that connects the network to the next one.

Error Detection and Correction

Objectives

- Introduce types of errors & the concept of redundancy
- Distinguish between error detection and correction
- Discuss block coding and show how errors can be detected using block coding
- Introduce the concept of Hamming distance
- Discuss cyclic codes and show how CRC is calculated

- Material feeds into CLO-2
  - Apply different mechanisms of **error control**, flow control and medium access control at the data link layer

Error Detection and Correction

- Any time data are transmitted from one node to the next, they can become corrupted in passage

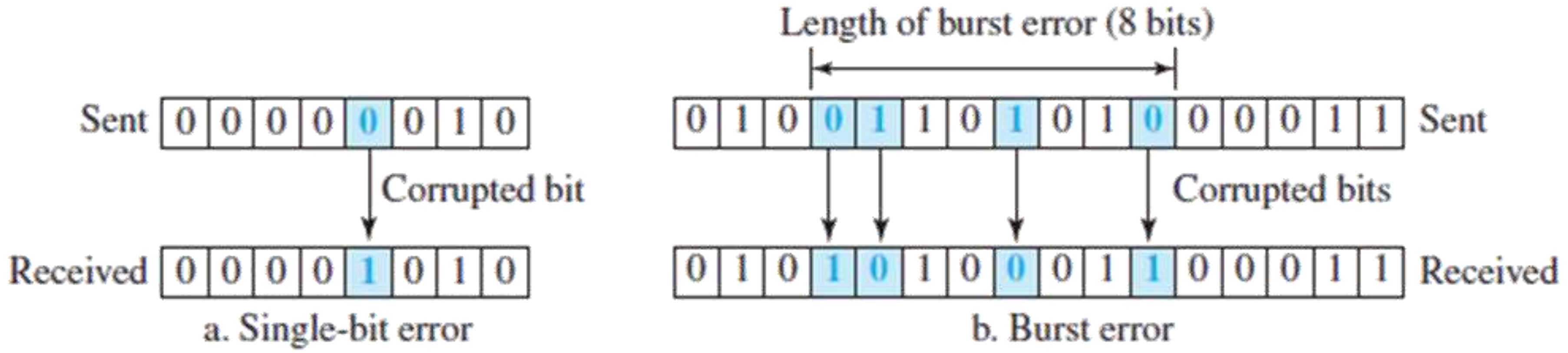
- At the data-link layer, if a frame is corrupted between the two nodes, it needs to be corrected before it continues its journey to other nodes
  - Most link-layer protocols simply <u>discard the frame</u> and let the <u>upper-layer protocols handle the retransmission</u> of the frame

- Some multimedia applications, however, try to correct the corrupted frame

Error Detection and Correction

- Whenever bits flow from one point to another, they are subject to unpredictable changes because of interference

- Type of Errors are:
  - **Single-bit error** which means that only 1 bit of a given data unit (such as a byte, character, or packet) is changed from 1 to 0 or from 0 to 1.
  - **Burst error** which means that 2 or more bits in the data unit have changed from 1 to 0 or from 0 to 1



Sent 0 0 0 0 0 0 1 0
Corrupted bit
Received 0 0 0 0 1 0 1 0
a. Single-bit error
Length of burst error (8 bits)
0 1 0 0 1 1 0 1 0 1 0 0 0 0 1 1 Sent
Corrupted bits
0 1 0 1 0 1 0 0 0 1 1 0 0 0 1 1 Received
b. Burst error

Error Detection and Correction

- **Redundancy Concept**
  - To be able to detect or correct errors, we need to send some extra bits with our data
  - These redundant bits are added by the sender and removed by the receiver
  - Their presence allows the receiver to detect or correct corrupted bits

- **Detection versus Correction**
  - In *error detection*, we are only looking to see if any error has occurred. The answer is a simple yes or no
    - A single-bit error is the same for us as a burst error
  - In *error correction*, we need to know the exact number of bits that are corrupted and, more importantly, their location in the message

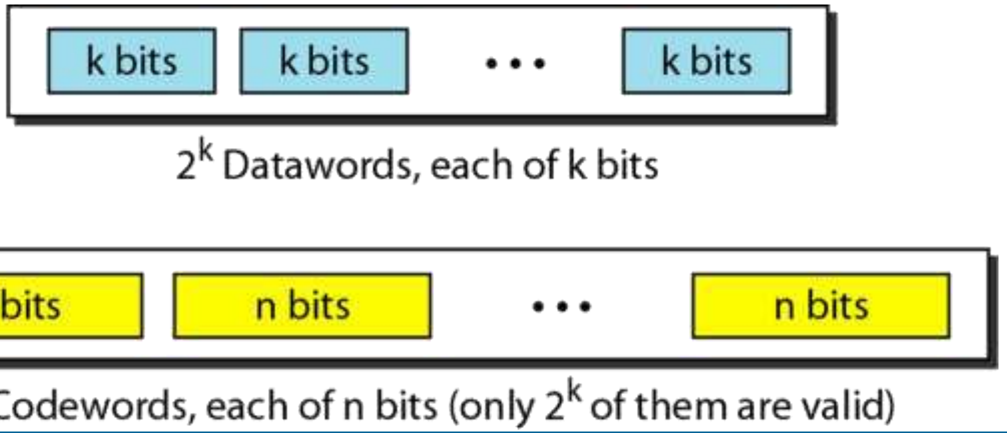
Error Detection and Correction

- **Forward Error Correction Versus Retransmission**
  - ***Forward error correction*** is the process in which the receiver tries to guess the message by using redundant bits
  - ***Correction by retransmission*** is a technique in which the receiver detects the occurrence of an error and asks the sender to resend the message. Resending is repeated until a message arrives that the receiver believes is error-free

- **Coding**
  - Redundancy is achieved through various coding schemes
  - The sender adds redundant bits through a process that creates a relationship between the redundant bits and the actual data bits
  - The receiver checks the relationships between the two sets of bits to detect errors

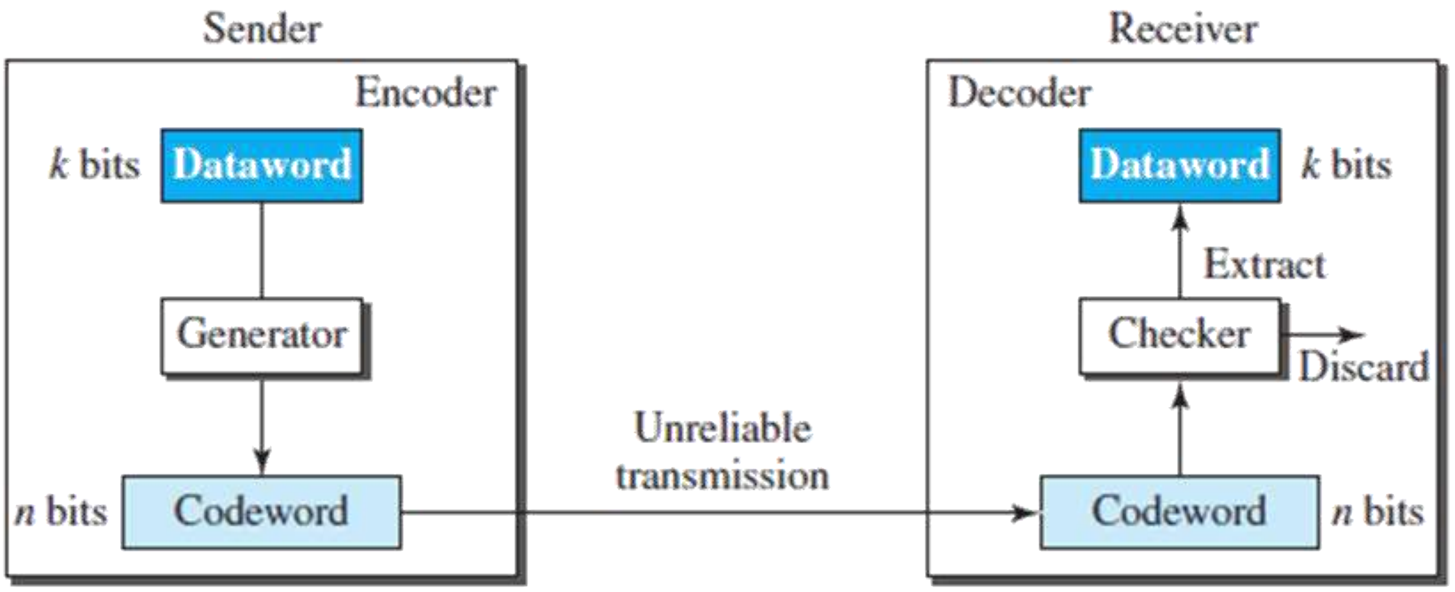
Block Coding

- 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***
- The number of possible codewords is larger than the number of possible datawords
- The block coding process is one-to-one; the same dataword is always encoded as the same codeword.


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

Block Coding: Error Detection

- If the following two conditions are met, the receiver can detect a change in the original codeword
    1. The receiver has (or can find) a list of valid codewords.
    2. The original codeword has changed to an invalid one



Exercise

- In a block code, a dataword is 20 bits and the corresponding codeword is 25 bits. What are the values of k, r, and n according to the definitions in the text?
  - How many redundant bits are added to each dataword?
  - How many codewords exist in this block code?
  - How many of these codewords are valid codewords?
- In a codeword, we add two redundant bits to each 8-bit data word. Find the following:
  - number of valid codewords
  - number of invalid codewords
- There are 64 valid code words in a block of 256 codewords. Calculate the value of k, r and n.

Error Detection: An Example

Let us assume that k = 2 and n = 3

The table below shows the list of datawords and codewords

Later, we will see how to derive a codeword from a dataword

Dataword	Codeword	Dataword	Codeword
00	000	10	101
01	011	11	110

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

Case 1: The receiver receives 011. It is a valid codeword

<u>The receiver extracts the dataword 01 from it</u>

Error Detection: An Example

Dataword	Codeword	Dataword	Codeword
00	000	10	101
01	011	11	110

Assume the sender encodes the dataword 01 as 011 and sends it to the receiver.

Case 2: The codeword is corrupted during transmission, and 111 is received.

This is not a valid codeword and is discarded.

Error Detection: An Example

Dataword	Codeword	Dataword	Codeword
00	000	10	101
01	011	11	110

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

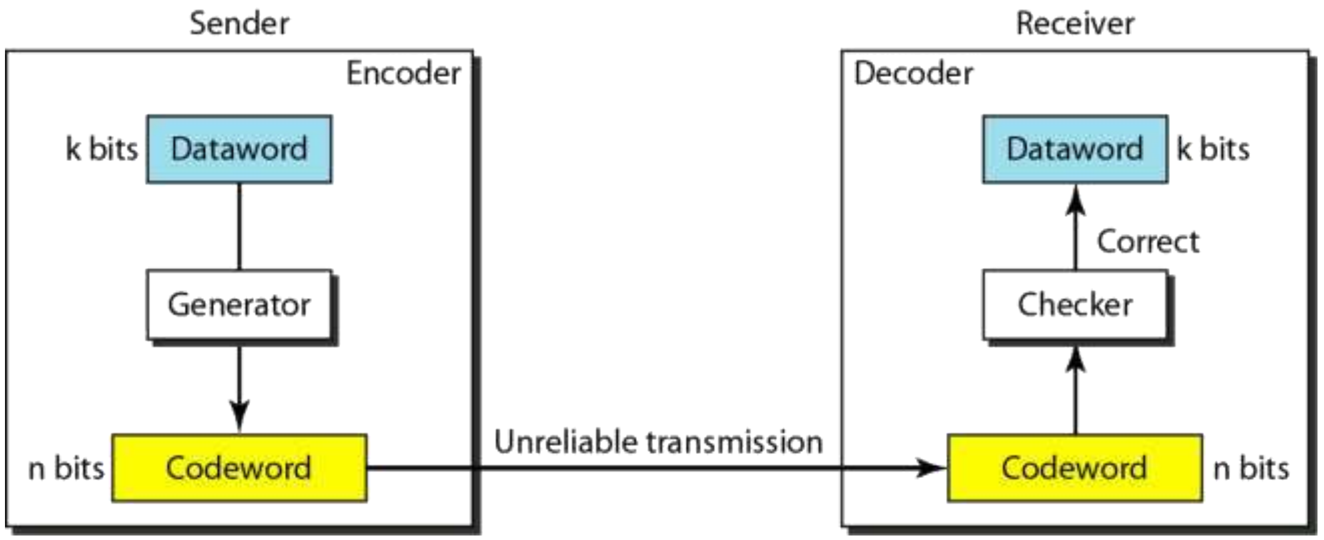
Case 3: The codeword is corrupted during transmission, and 000 is received. This is a valid codeword!

<u>The receiver incorrectly extracts the dataword 00</u>
<u>Two corrupted bits have made the error undetectable</u>

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

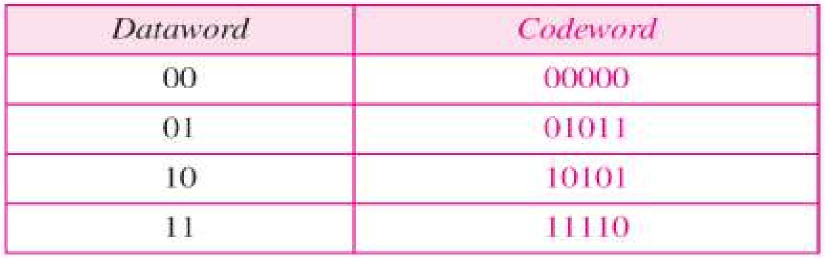
Block Coding: Error Correction

- Error correction is much more difficult than error detection
- In error correction, the receiver needs to find (or guess) the original codeword sent
- More redundant bits are needed for error correction than for error detection



Error Correction: An Example

- We add 3 redundant bits to the 2-bit dataword to make 5-bit codewords
- Assume the dataword is 01. The sender creates the codeword 01011
- The codeword is corrupted during transmission, and 01001 is received
- Receiver finds that the received codeword is not in the table (means an error has occurred). Receiver, assuming that there is only 1 bit corrupted, uses the following strategy to guess the correct dataword



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

Answer

Comparing the received codeword 01001 with the list of available codewords

01001 and
00000 there are 2 different bits

01001 and
01011 there is 1 different bits ✓

01001 and
10101 there are 3 different bits

01001 and
11110 there are 4 different bits

Hamming Distance

- The Hamming distance between two words (of the same size) is the number of differences between the corresponding bits
  - Hamming distance between two words x and y denoted as d(x, y)

- The Hamming distance between the received codeword and the sent codeword is the number of bits that are corrupted during transmission.
  - Example: If sent = 00000 and received = 01101, 3 bits are in error
    - Hamming distance between the two is = d(00000, 01101) = 3

- The Hamming distance can easily be found if we apply the XOR operation ($\oplus$) on the two words and count the number of 1s in the result.

Hamming Distance: Examples

- Find the Hamming distance between two pairs of words
  - The Hamming distance d(000, 011)

```
          000
XOR       011
          011      number of 1's  is 2
d(000, 011)  = 2
```

  - The Hamming distance d(10101, 11110)

```
          10101
XOR       11110
          01011   number of 1's  is 3
d(10101, 11110) = 3
```

UAEU College of Information Technology

Minimum Hamming Distance for Error Detection

- The **minimum Hamming distance (d$_{min}$)** is the smallest Hamming distance between all possible pairs of codewords

- 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 Detection: An Example

Find the minimum Hamming distance of the coding scheme in Table below

Dataword	*Codeword*	*Dataword*	*Codeword*
00	000	10	101
01	011	11	110

Solution: We first find all Hamming distances.

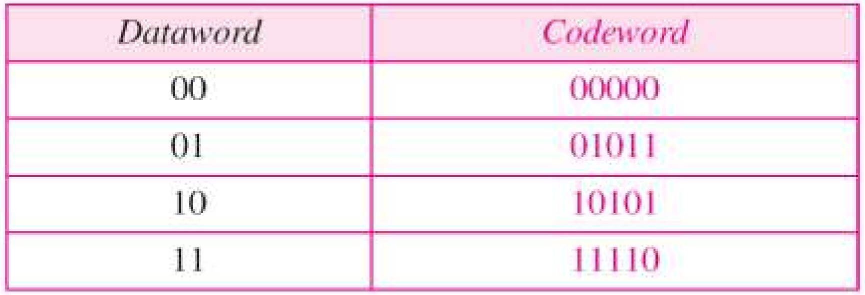
d(000,011) = 2 d(000,101) = 2 d(000,110) = 2
d(011,101) = 2 d(011, 110) = 2
d(101,110) = 2

The d_{min} in this case is 2

This code guarantees detection of only a **single error**. For example, if the third codeword (101) is sent and one error occurs, the received codeword does not match any valid codeword. If two errors occur, however, the received codeword may match a valid codeword and the errors are not detected.

Minimum Hamming Distance for Error Detection: An Example

Find the minimum Hamming distance of the coding scheme in Table below



Dataword	Codeword
00	00000
01	01011
10	10101
11	11110

Solution: We first find all Hamming distances.

d(00000,01011) = 3   d(00000,10101) = 3   d(00000,11110) = 4
d(01011,10101) = 4   d(01011, 11110) = 3
d(10101,11110) = 3

The d_{min} in this case is 3

This code can detect up to **two errors**. Again, we see that when any of the valid codewords is sent, two errors create a codeword which is not in the table of valid codewords. The receiver cannot be fooled.

Linear Block Codes

- A linear block code is a code in which the exclusive OR (addition modulo-2) of two valid codewords creates another valid codeword

Dataword	Codeword	Dataword	Codeword
00	000	10	101
01	011	11	110

$$\begin{array}{rr} & 011 \\ \text{XOR} & 101 \\ \hline & 110 \end{array}$$

College of Information Technology

Minimum Distance for Linear Block Codes

- The minimum Hamming distance for a linear block code is the number of 1s in the nonzero valid codeword with the smallest number of 1s

Example:

Dataword	Codeword	Dataword	Codeword
00	000	10	101
01	011	11	110

The numbers of 1s in the nonzero codewords are 2, 2, and 2. So the minimum Hamming distance is $d_{min} = 2$.

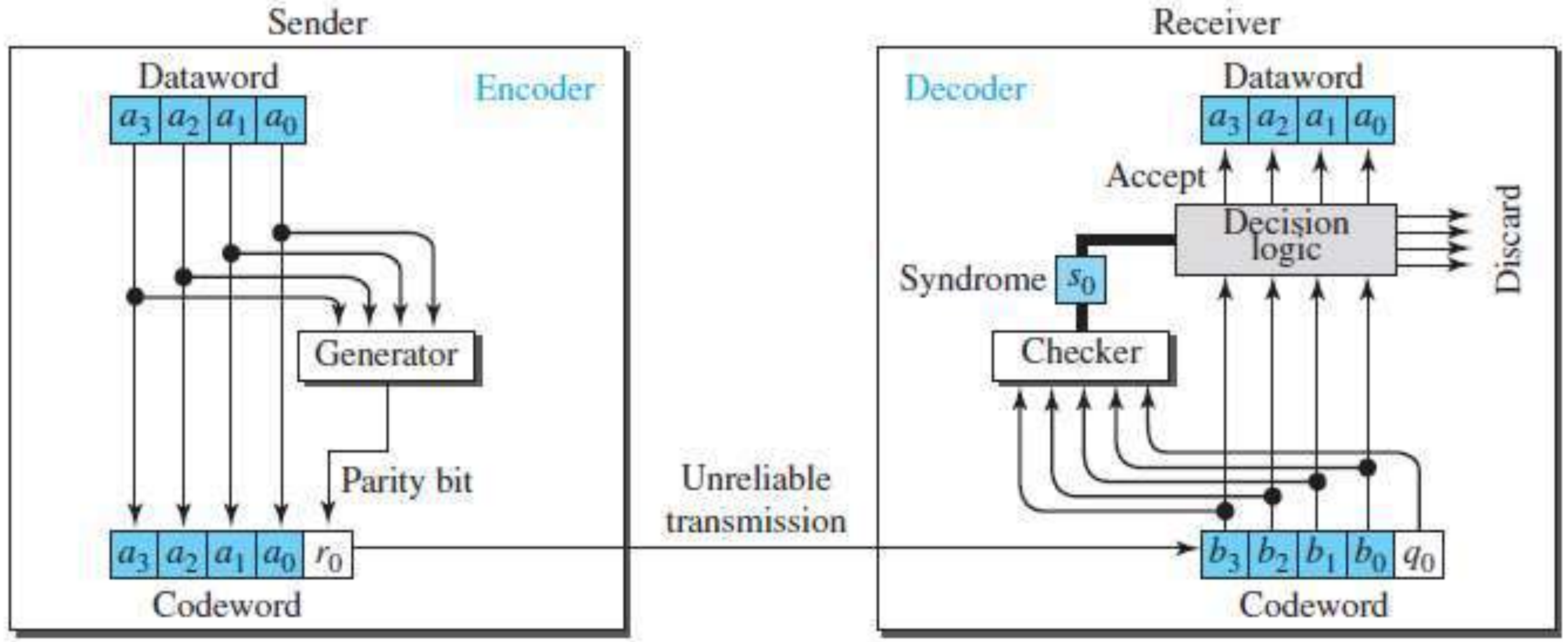
Parity-Check Code

- Perhaps the most familiar error-detecting code is the **parity-check code**. This code is a linear block code

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

- The minimum Hamming distance for this category is $d_{min} = 2$, which means that the code is a single-bit error-detecting code

Encoder and decoder for simple parity-check code

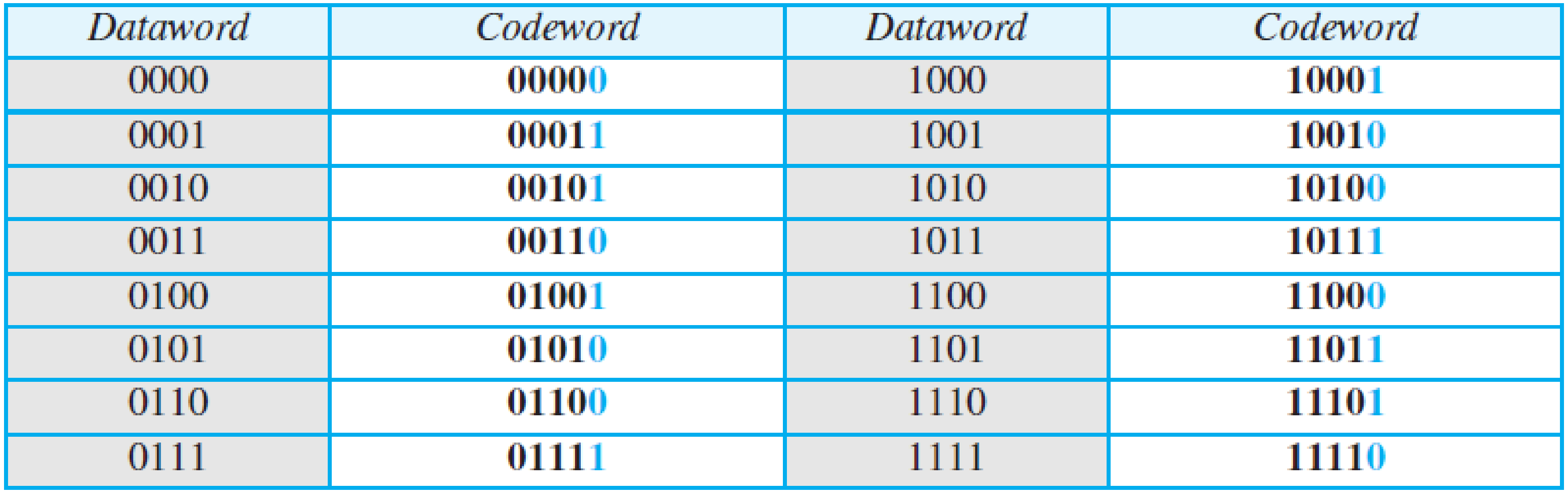


The calculation is done in modular arithmetic

$$r_0 = a_3 + a_2 + a_1 + a_0 \qquad \text{(modulo-2)}$$

$$s_0 = b_3 + b_2 + b_1 + b_0 + q_0 \qquad \text{(modulo-2)}$$

Parity-Check Code: An Example



Dataword	Codeword	Dataword	Codeword
0000	00000	1000	10001
0001	00011	1001	10010
0010	00101	1010	10100
0011	00110	1011	10111
0100	01001	1100	11000
0101	01010	1101	11011
0110	01100	1110	11101
0111	01111	1111	11110

Cyclic Codes

- **Cyclic codes** are special linear block codes with one extra property. In a cyclic code, if a codeword is cyclically shifted (rotated), the result is another codeword.

Dataword	*Codeword*	*Dataword*	*Codeword*
0000	0000000	1000	1000101
0001	0001011	1001	1001110
0010	0010110	1010	1010011
0011	0011101	1011	1011000
0100	0100111	1100	1100010
0101	0101100	1101	1101001
0110	0110001	1110	1110100
0111	0111010	1111	1111111

```
        0001011
XOR     1010011
        1011000
```

Rotate 0001011 one bit to right
1000101

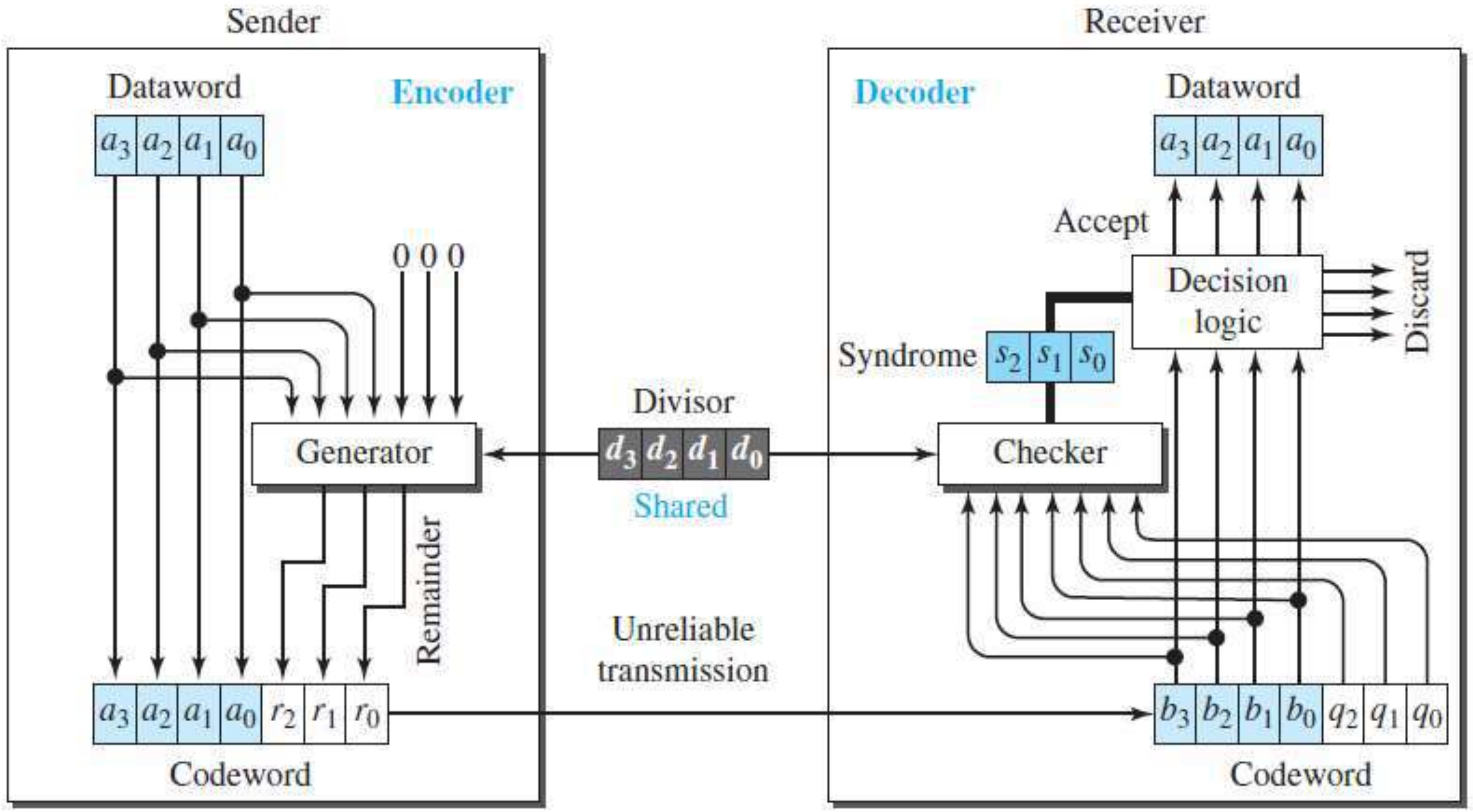
Cyclic Redundancy Check (CRC)

- A **cyclic redundancy check (CRC)** is an error-detecting code commonly used in digital networks and storage devices to detect accidental changes to raw data

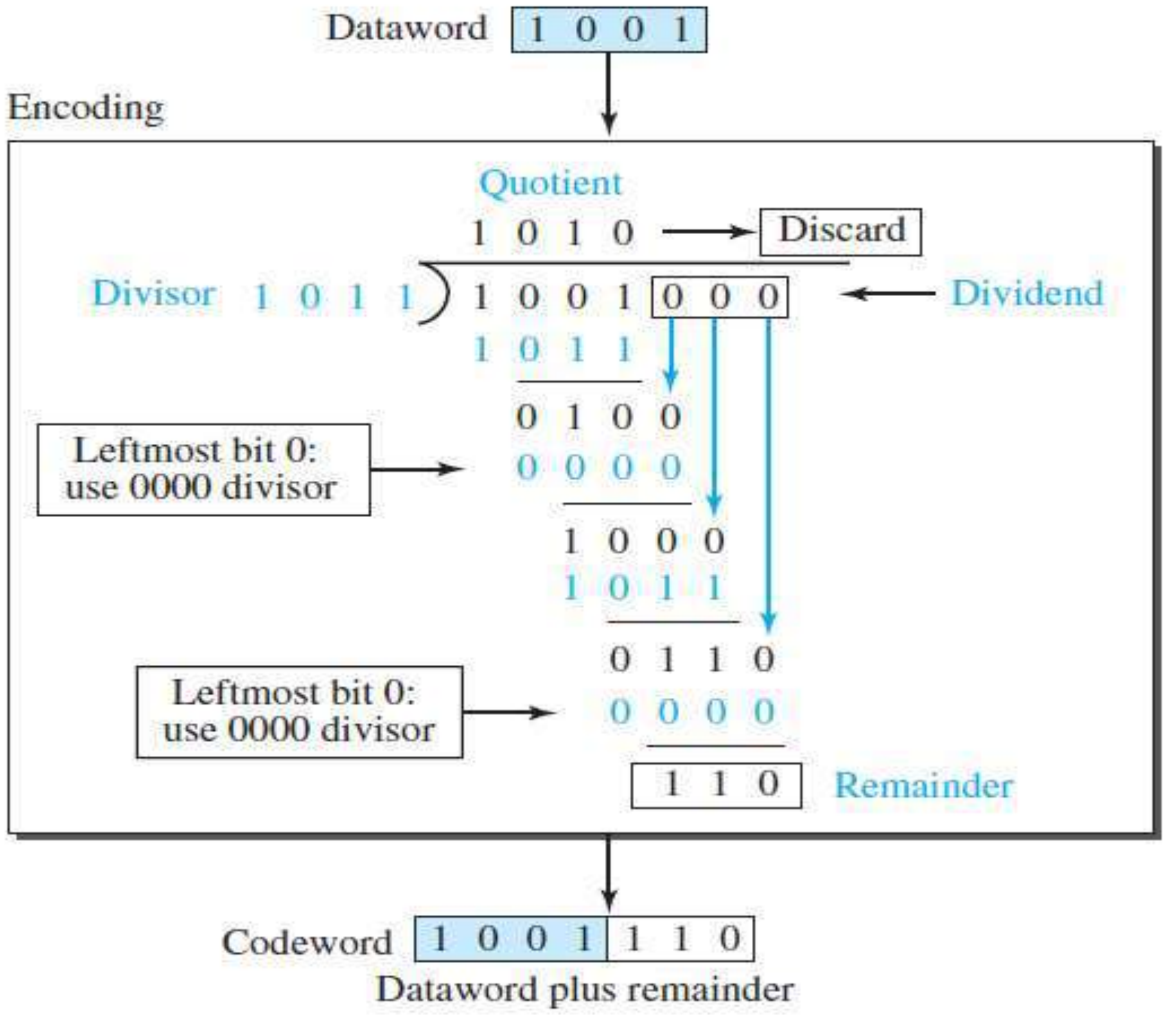
- Blocks of data entering these systems get a short check value attached, based on the remainder of a polynomial division of their contents

- On retrieval the calculation is repeated, and corrective action can be taken against presumed data corruption if the check values do not match

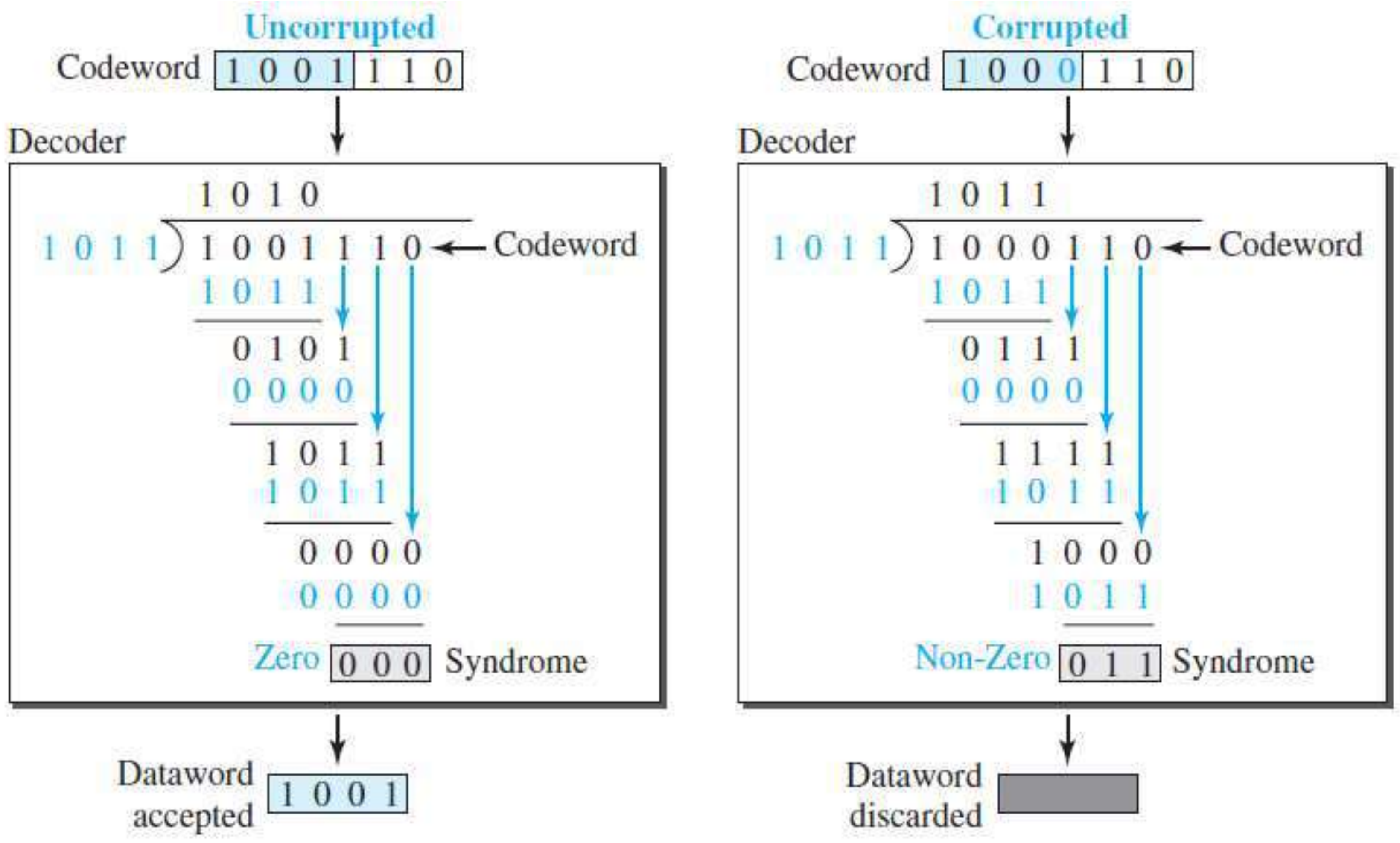
CRC Encoder and Decoder



Division in CRC Encoder



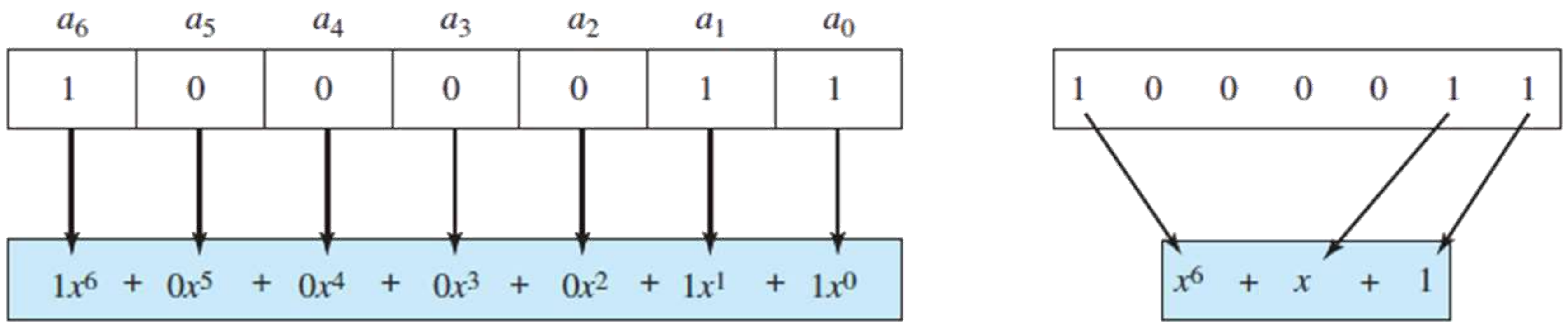
Division in the CRC Decoder for Two Cases



UAEU College of Information Technology

Polynomials

- A pattern of 0s and 1s can be represented as a polynomial with coefficients of 0 and 1.
- The power of each term shows the position of the bit; the coefficient shows the value of the bit.


a_6 a_5 a_4 a_3 a_2 a_1 a_0
1 0 0 0 0 1 1
$1x^6 + 0x^5 + 0x^4 + 0x^3 + 0x^2 + 1x^1 + 1x^0$
1 0 0 0 0 1 1
$x^6 + x + 1$

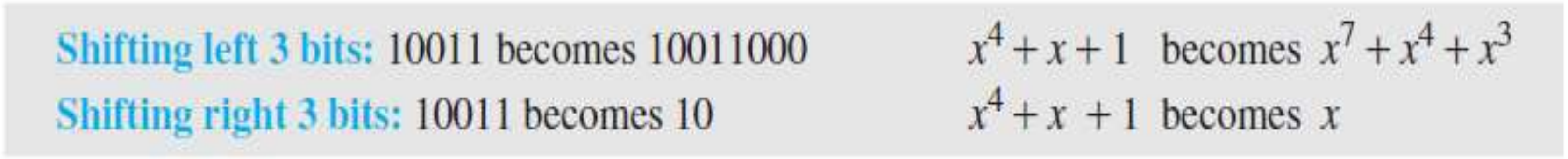
a. Binary pattern and polynomial

b. Short form

UAEU College of Information Technology

Operations on Polynomials

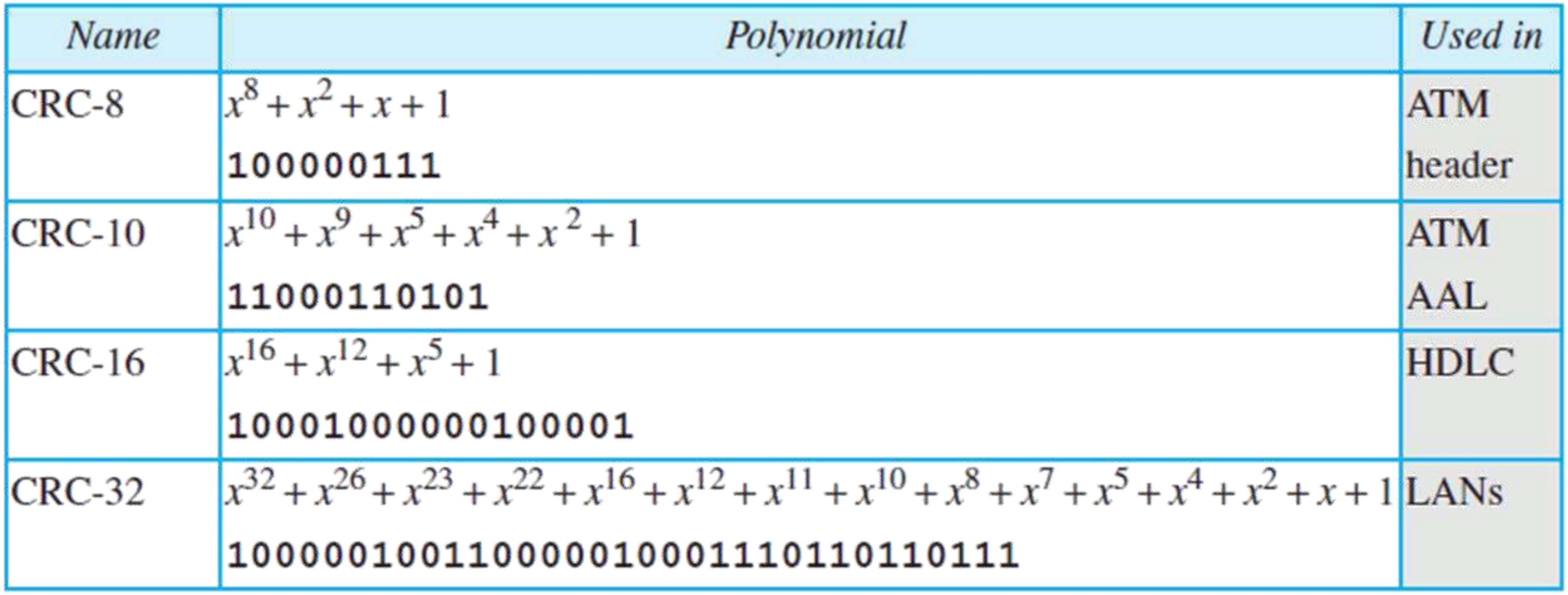
- **Shifting**
  - A binary pattern is often shifted a number of bits to the right or left
  - Shifting to the left means adding extra 0s as rightmost bits; shifting to the right means deleting some rightmost bits
  - Shifting to the left is accomplished by multiplying each term of the polynomial by x^m, where m is the number of shifted bits
  - shifting to the right is accomplished by dividing each term of the polynomial by x^m, where m is the number of shifted bits



Shifting left 3 bits: 10011 becomes 10011000	$x^4 + x + 1$ becomes $x^7 + x^4 + x^3$
Shifting right 3 bits: 10011 becomes 10	$x^4 + x + 1$ becomes x

Standard Polynomials

- Some standard polynomials used by popular protocols for CRC generation are shown in the following Table along with the corresponding bit pattern.



Name	*Polynomial*	*Used in*
CRC-8	$x^8 + x^2 + x + 1$ 100000111	ATM header
CRC-10	$x^{10} + x^9 + x^5 + x^4 + x^2 + 1$ 11000110101	ATM AAL
CRC-16	$x^{16} + x^{12} + x^5 + 1$ 10001000000100001	HDLC
CRC-32	$x^{32} + x^{26} + x^{23} + x^{22} + x^{16} + x^{12} + x^{11} + x^{10} + x^8 + x^7 + x^5 + x^4 + x^2 + x + 1$ 100000100110000010001110110110111	LANs

UAEU College of Information Technology

Summary

- We have discussed:
  - Data Link Layer and its services
  - Error Detection and Correction
  - Block Coding
  - Cyclic Coding

FLOW AND ERROR CONTROL

What will we learn?

- What is Framing and why is it important?
- What is Flow and Error Control?
- Why is it important?
- What are the different approaches?
- In a nutshell, appreciate how certain aspects of the MAC layer are realised!
- Feeds into CLO-2
  - Apply different mechanisms of error control, **<u>flow control</u>** and medium access control at the data link layer

Outline

- Data Link Control Services
  - Framing
  - Flow and Error Control
- Flow and Error Control Protocols
  - Simple
  - Stop-and-Wait ARQ
  - Go-Back-N ARQ
  - Selective Repeat ARQ

Data Link Control Services

- Deals with procedures for communication between two adjacent nodes whether there is a:
  - Dedicated link
  - Broadcast link
- Data link control functions include **framing** , **flow and error control**.

Framing

- Pack bits into frames, distinguish one frame from another
  - Postal System: Letter is the information & Envelope is the frame!
- Separate a message from one source to a destination by adding a sender address and a destination address
  - To address: tells where the packet is going to
  - From address: helps the recipient acknowledge receipt
- Divide larger messages into smaller frames
  - Larger frames occupy the medium for longer
  - Even a single bit error might require retransmitting entire frame
  - Retransmission expensive for larger messages
  - Spread the risk across multiple smaller frames

Frame Size

- Frames can be of fixed or variable size

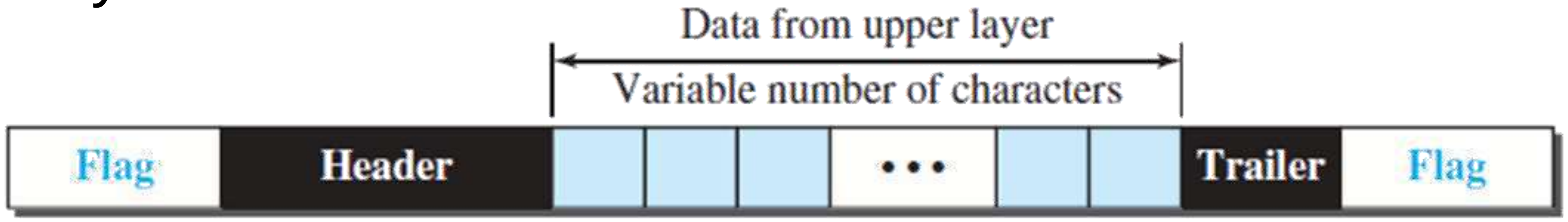
- In fixed-size framing, there is no need for defining the boundaries of the frames; the size itself can be used as a delimiter

- In variable-size framing, we need a way to define the end of one frame and the beginning of the next
  - Two approaches: character-oriented and bit-oriented

- Our focus is on variable sized framing
  - Most commonly used in networks we encounter in day-to-day life

Character-Oriented Framing

- Also known as byte-oriented framing
- Data to be carried are 8-bit characters from a coding system such as ASCII.



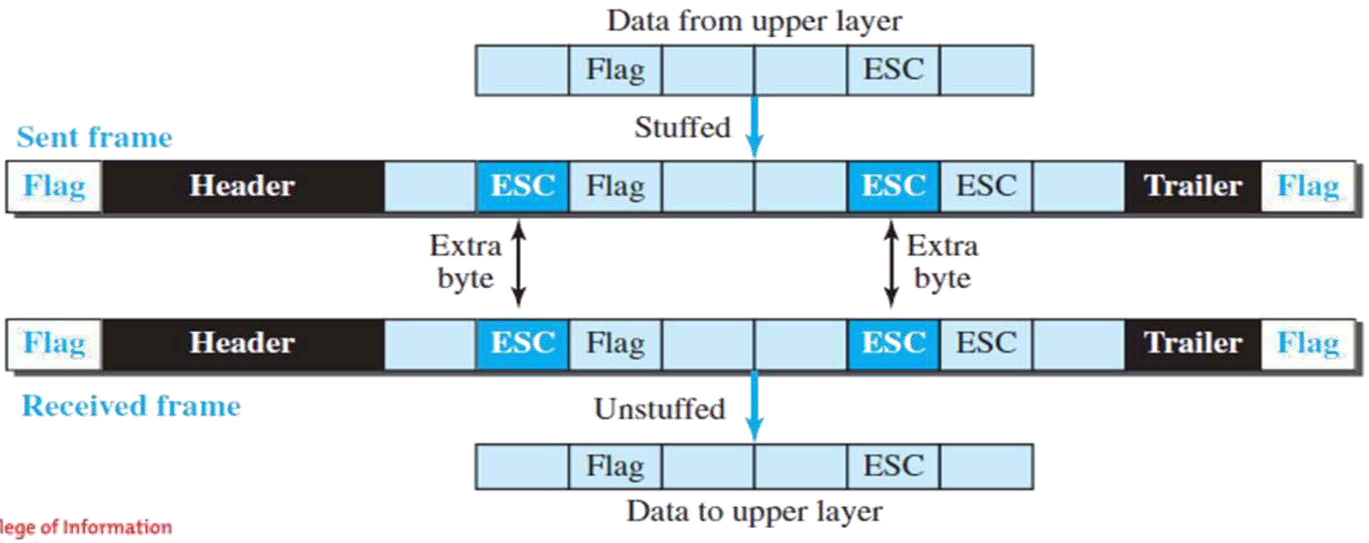
- Header carries the source and destination addresses and other control information
- The trailer carries error detection redundant bits
- A 1-byte **flag** is added at the beginning and the end of a frame to separate one frame from the next
- Flag signals the start or end of a frame

An Issue to think about…

- Let F denote the Flag
  - Sender wants to send this data DEDDFDDEEDFD
- You are the receiver. How will you interpret this?
- The Flag appears in the Data!
  - What should be done?

Byte Stuffing

- A special byte added to the data section of the frame when there is a character with the same pattern as flag
- Special byte is called the escape character (ESC)
  - ESC 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 as a delimiting flag

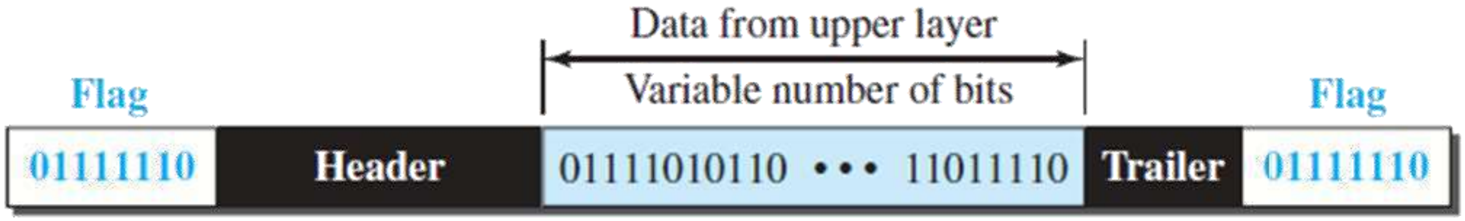


UAEU College of Information Technology

Bit-Oriented Framing

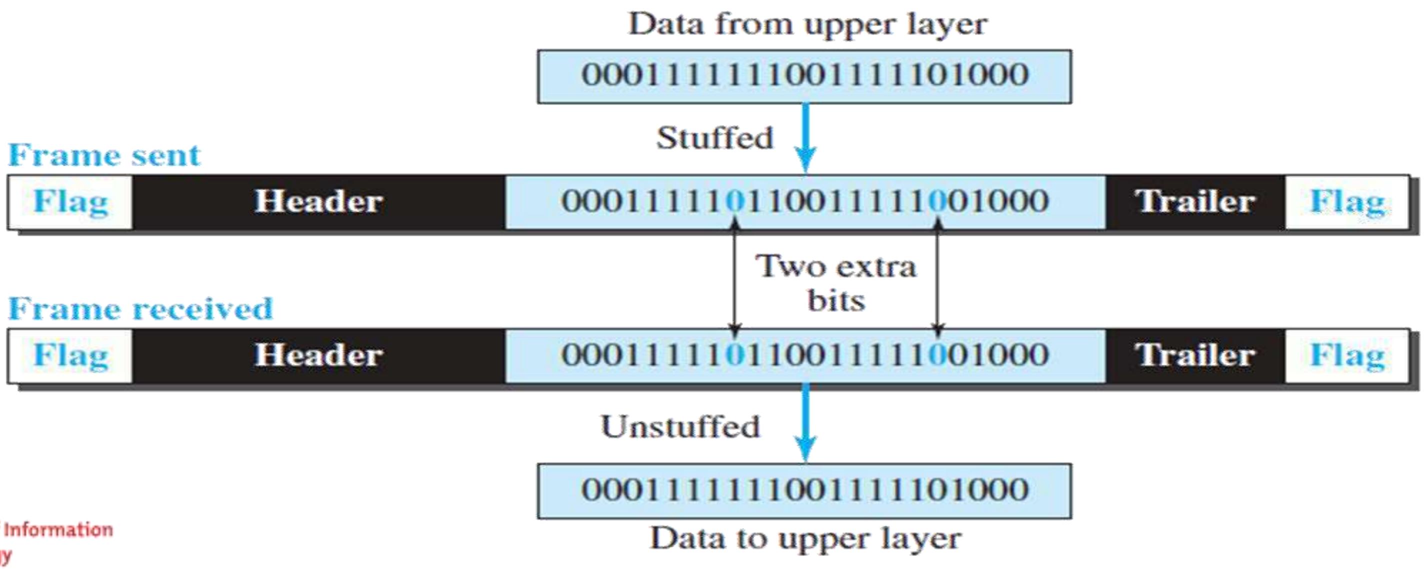
- In **bit-oriented framing**, the data section of a frame is a sequence of bits to be interpreted by the upper layer as text, graphic, audio, video for example

- Most protocols use a special 8-bit pattern flag, 01111110, as the delimiter to define the beginning and the end of the frame.



Bit Stuffing

- A Flag pattern (01111110) can appear in the data too!
  - Need to explicitly tell the receiver that this isn't a Flag
- Achieved by stuffing one single bit to prevent the pattern from looking like a flag
- In bit stuffing, if a 0 and five consecutive 1 bits are encountered, an extra 0 is added
  - This stuffed bit is removed from the data by the receiver



UAEU College of Information Technology

Solve these Problems

- Byte stuff the following frame payload DEDDFDDEEDFD
  - E (Escape), F (Flag), D (Data) characters

- Bit Stuff the following frame payload 0001111100000111110001111110111

Solution

- Byte stuff the following frame payload DEDDFDDEEDFD
  - E (Escape), F (Flag), D (Data) characters
- Solution: D**E**EDD**E**FDD**E**E**E**ED**E**FD

- Bit Stuff the following frame payload 000111110000011111000111111101111
- Solution: 00011111**0**000011111**0**00011111**0**101111

UAEU College of Information Technology

Flow and Error Control

- Data communication: at least one sender, one receiver
  - Sender and receiver have to coordinate

- Most important responsibilities of the data link layer are:
  - **flow control**
  - **error control**

- Collectively, these functions in addition to framing are known as **Data Link Control**

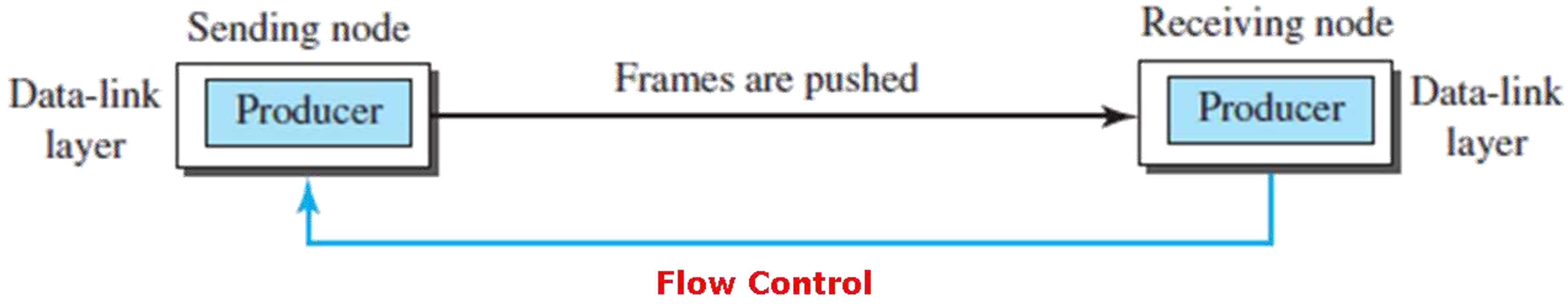
Flow Control

- Sender produces data, Receiver consumes it
  - Need a balance between production and consumption rates
- Faster production can overwhelm consumer
  - Buffer Overflows at the receiver
- Slower production makes a consumer wait
  - Poorer resource utilization (not using available capacity)
- **Flow control:** set of procedures to address the overwhelm issue

Flow Control Concept

- Link layer at the sender sends frames to link layer at the receiver

- Flow control process at the receiver side provides feedback to the sender to stop/slow down



Error Control

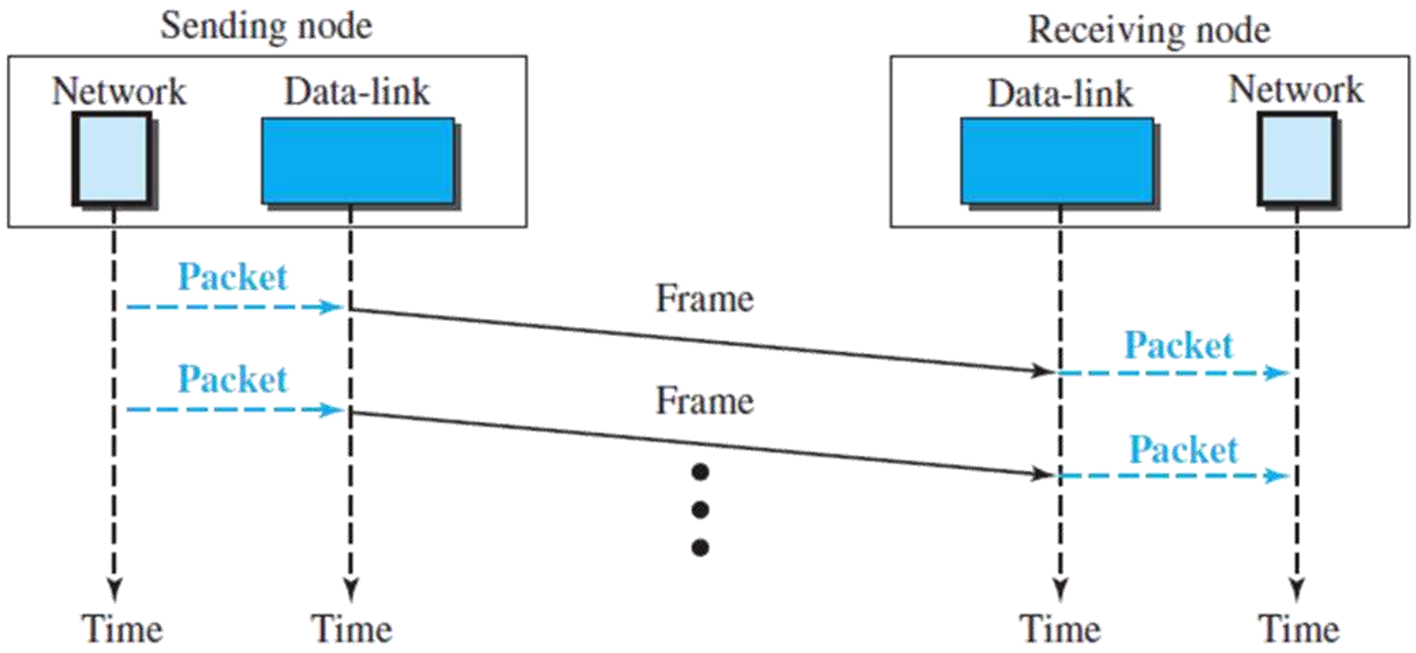
- Need to know if what you sent is what was received
- How to detect error? How do you react?
- Error control refers primarily to methods of error detection and retransmission
- Sender adds CRC in the frame, receiver checks
- Different strategies
  - Discard if corrupted frame received
  - Send an ACK / NAK to the sender
- **Automatic Repeat reQuest** (ARQ): Receiver coordinates with the sender to retransmit lost frames

Flow and Error Control Protocols

- Simple
- Stop-and-Wait ARQ
- Sliding Window protocol
- Go-Back-N ARQ
- Selective Repeat ARQ

Simple Protocol

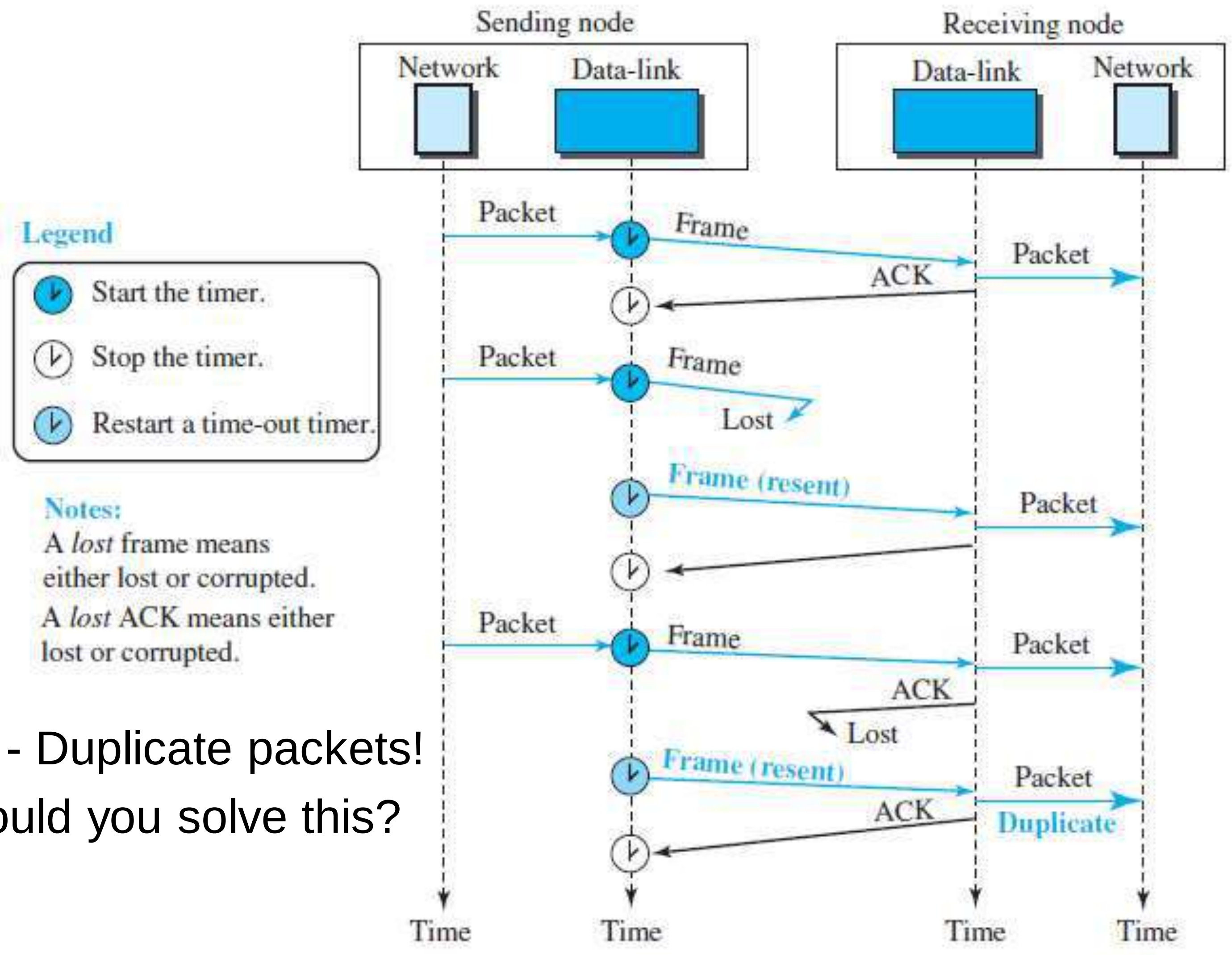
- <u>Uses neither flow nor error control</u>
- Receiver can immediately handle any frame it receives
- Receiver can never be overwhelmed with incoming frames



Stop-and-Wait Protocol

- <u>Uses both flow and error control</u>
- Sender sends one frame at a time and waits for an ACK
  - Does not send the next frame before an ACK is received
- What happens if the frame is never received?
- What happens if the frame is received but corrupted?
- What happens if the ACK is never received?
- What happens if the ACK is corrupted?
- Does the sender know what the problem is?
- Sender adds CRC, receiver checks
  - If CRC incorrect, the frame is corrupted and silently discarded
  - Every time the sender sends a frame, it starts a timer
    - If ACK arrives before the timer expires, timer stopped and next frame is sent
  - If the timer expires, the sender resends the previous frame, assuming that the frame was either lost or corrupted

Stop-and-Wait Protocol



Problem - Duplicate packets!

How would you solve this?

Sequence and Acknowledgment Numbers

- Duplicate packets need to be avoided

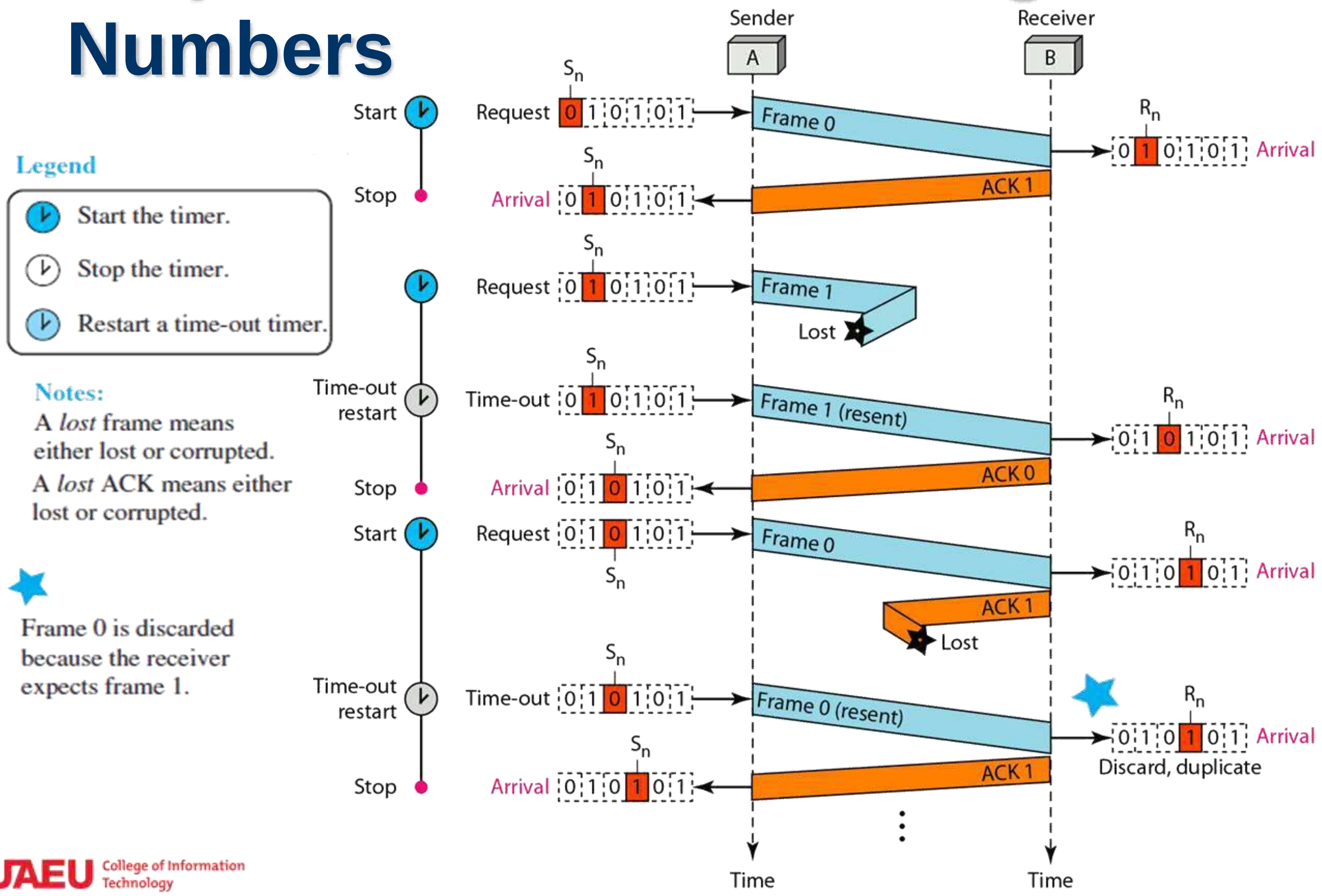
- E.g., assume we are ordering some item online. If each packet defines the specification of an item to be ordered, duplicate packets mean ordering an item more than once!

- How to fix this?
  - Add sequence numbers to the data frames and acknowledgment numbers to the ACK frames
  - Sequence numbers are 0, 1, 0, 1, 0, 1, . . .
  - The acknowledgment numbers can also be 1, 0, 1, 0, 1, 0, …
  - ACK no. defines the sequence no. of the next frame to receive!

Stop-and-Wait ARQ

- Stop and wait with ARQ
  - Error control method incorporated with stop and wait flow control protocol
- If receiver detects error, it discards the frame and sends a negative ACK (NAK), sender will re-send the frame
- What if the frame never got to the receiver?
  - Sender has a timer: each time a frame is sent, timer is set!
    - If no ACK or NAK is received during timeout period, frame resent
- What if the ACK/NAK was not received (although frame received at the receiver)?
- Sender retransmits on timeout (duplicate at the receiver)
- To avoid duplicates, frames and ACKs are alternatively labeled 0 or 1

Sequence and Acknowledgment Numbers



Stop and Wait ARQ - Efficiency

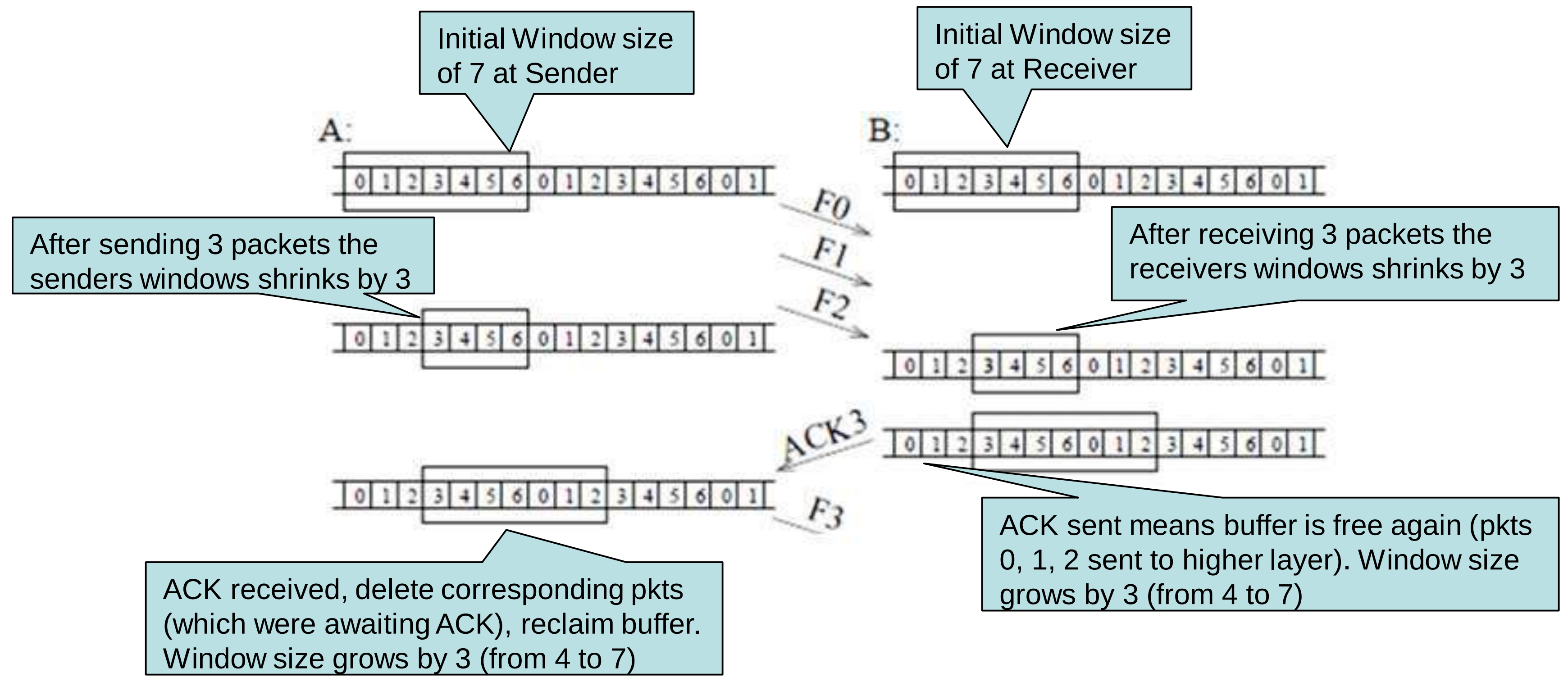
- Sender can’t send another frame until receiver acknowledges previous one
  - Could waste capacity in high Bandwidth environments
- Receiver may not ACK immediately
  - It could be busy, perhaps overwhelmed, so can’t ACK immediately
    - But this creates further problems
      - Unnecessary retransmissions (upon timeout at sender)
- Desirable to improve link utilisation

Sliding Window Protocol

- Flow control procedure where multiple packets can be in flight
- Frames and acknowledgements are numbered using sequence numbers.
- Sender maintains a list of sequence numbers (frames) it is allowed to transmit, called sending window.
  - A sending window of size N means that sender can send up to N frames un-acknowledged
- Receiver maintains a list of sequence numbers it is prepared to receive, called receiving window
  - A window size of N implies buffer space for N frames

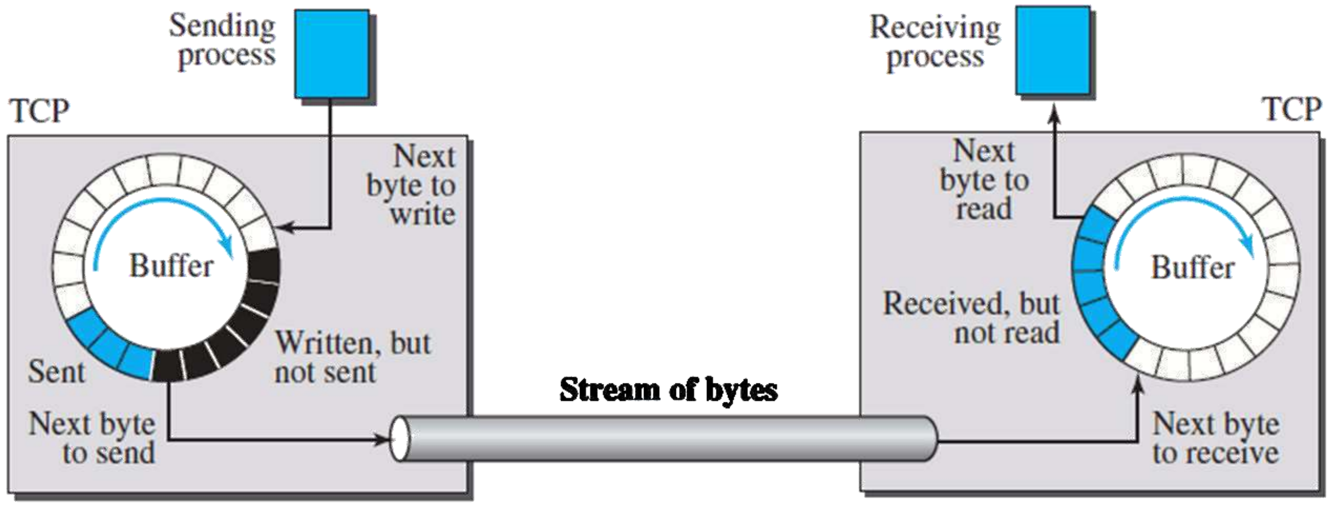
Sliding Window (continue)

Consider the case maximum window size N = 7. Window shrinks/grows


Initial Window size of 7 at Sender
Initial Window size of 7 at Receiver
A:
B:
0 1 2 3 4 5 6 0 1 2 3 4 5 6 0 1
F0
F1
F2
ACK3
F3
After sending 3 packets the senders windows shrinks by 3
After receiving 3 packets the receivers windows shrinks by 3
ACK sent means buffer is free again (pkts 0, 1, 2 sent to higher layer). Window size grows by 3 (from 4 to 7)
ACK received, delete corresponding pkts (which were awaiting ACK), reclaim buffer. Window size grows by 3 (from 4 to 7)

ACK3 means that receiver has received frame 0 to frame 2 correctly, ready to receive frame 3 (and rest of N frames within window)

Sliding Window Protocol

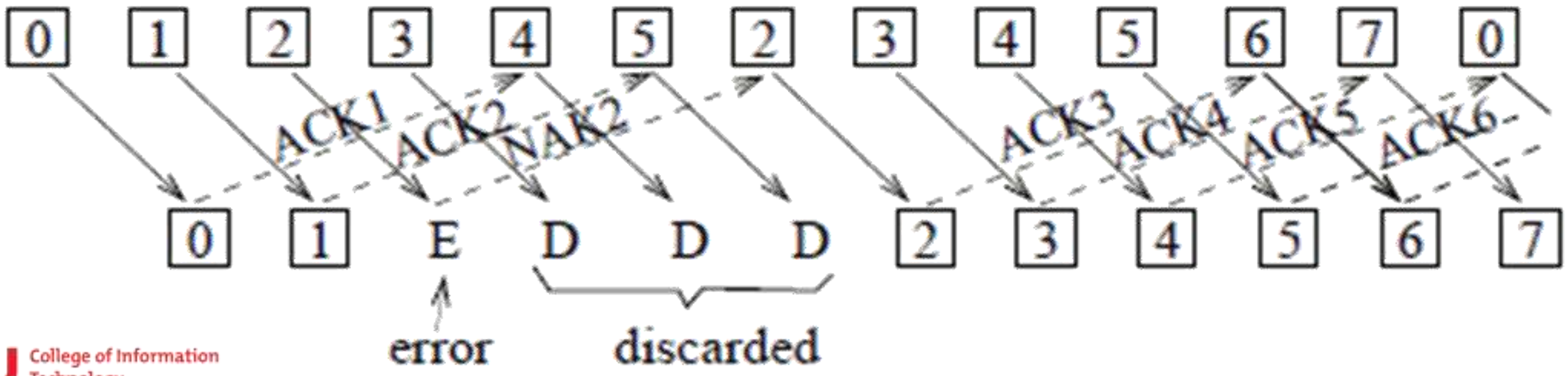


- Example from TCP but the principle is the same!
- There is a finite buffer at both the sender and the receiver
- At the sender,
  - Data already transmitted awaiting ACKs is shown in Blue cells
  - Data awaiting transmission is shown in Black cells
  - Data being received from layer above is stored in White cells
- At the receiver
  - Data received but yet to be passed to the higher layer is in blue cells
  - New data coming in can be inserted into the white cells

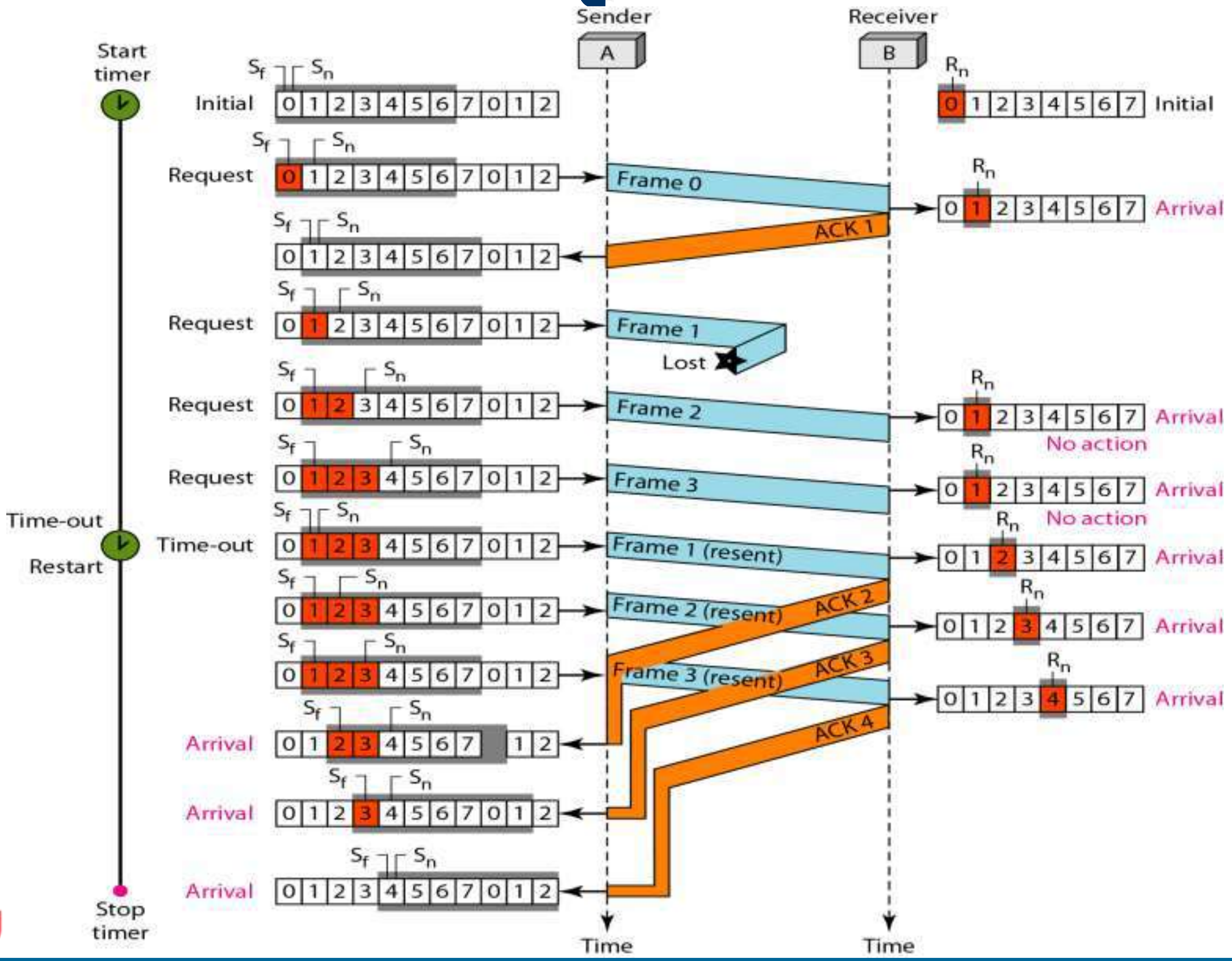
Go-Back-N ARQ

- The basic idea of Go-Back-N error control is: If frame *i* is damaged, receiver requests retransmission of all frames starting from frame *i*

- The receiver keeps track of only one variable, and there is no need to buffer out-of-order packets; they are simply discarded.



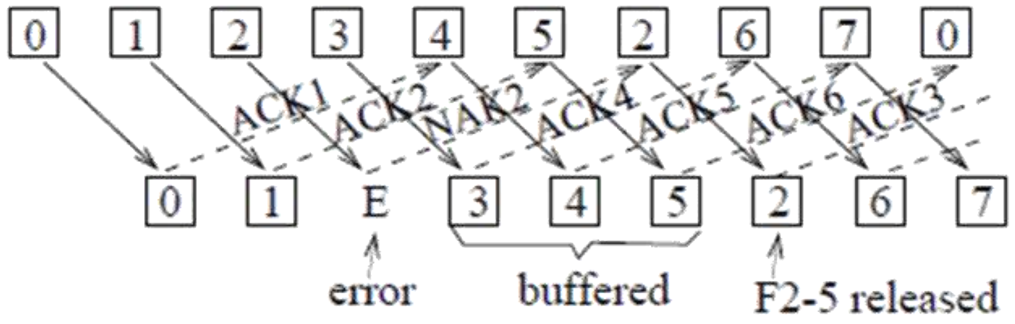
Go-Back-N ARQ



Sender
Receiver
A
B
Start timer
Initial
Request
Request
Request
Request
Time-out
Time-out
Restart
Arrival
Arrival
Arrival
Stop timer
Frame 0
ACK 1
Frame 1
Lost
Frame 2
Frame 3
Frame 1 (resent)
Frame 2 (resent)
Frame 3 (resent)
ACK 2
ACK 3
ACK 4
Initial
Arrival
Arrival
No action
Arrival
No action
Arrival
Arrival
Arrival
S_f
S_n
R_n
Time
Time

Selective Repeat ARQ

- Selective-Repeat protocol, resends only selective packets, those that are actually lost.



- Selective-Repeat would appear to be more efficient than Go-Back-N, but it is harder to implement and less used

UAEU College of Information Technology

Food for Thought

- How do you choose appropriate Window size?
- How do you choose an appropriate timeout value?
- Should these be fixed / adaptive?
- What dictates these choices?

Summary

- We discussed
  - Data Link Control Services
    - Framing
    - Flow and Error Control
  - Flow and Error Control Protocols
    - Simple
    - Stop-and-Wait ARQ
    - Sliding Window Protocol
    - Go-Back-N ARQ
    - Selective Repeat ARQ

Further Resources

- Illustrations & Examples of Framing
  - http://www.mathcs.emory.edu/~cheung/Courses/455/Syllabus/3-datalink/framing.html
- Sliding Window Protocols:
  - http://www.ece.northwestern.edu/~rberry/ECE333/Lectures/lec9.pdf
- MAC for Lunar Surface Communications
  - https://ntrs.nasa.gov/archive/nasa/casi.ntrs.nasa.gov/20090019005.pdf

Medium Access Control

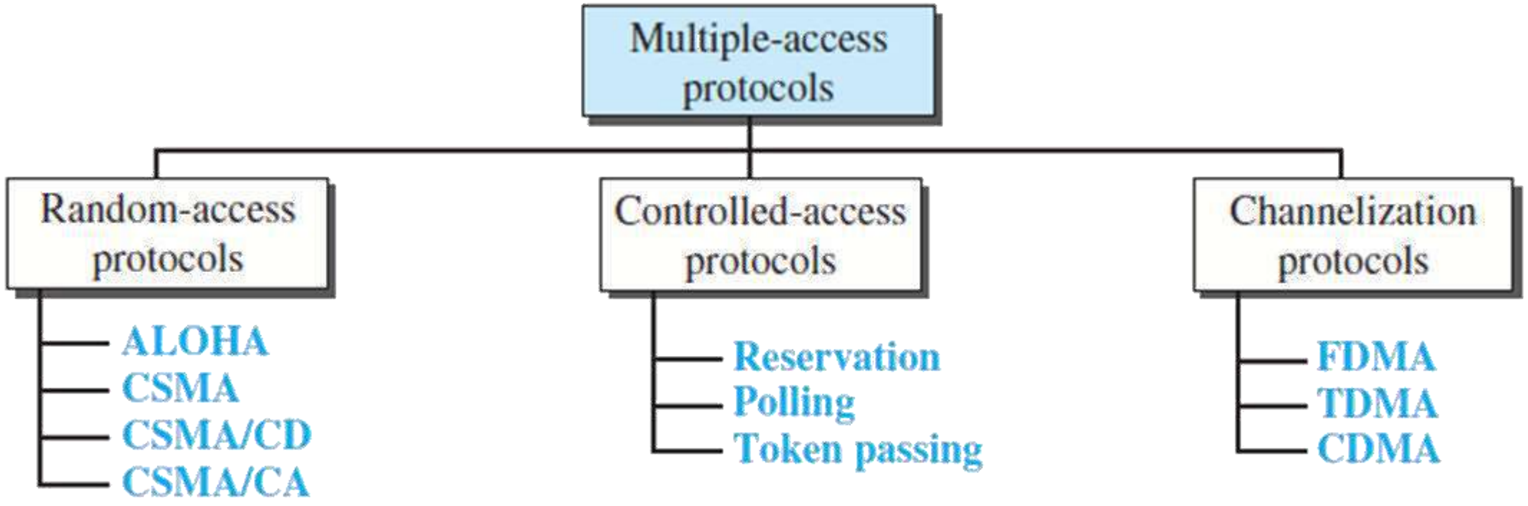
Objectives:

- Understand the need for Media-Access Protocols
- Discuss and explain:
  - Random access protocols such as: Pure ALOHA, Slotted ALOHA, CSMA/CD, CSMA/CA
  - Controlled access protocols such as : Reservation, Polling, Token Passing
  - Channelisation protocols such as: FDMA, TDMA, CDMA
- Feeds into CLO-2
  - Apply different mechanisms of error control, flow control and **medium access control** at the data link layer

Medium Access an Introduction

- 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

- Protocols that handle this belong to a sublayer in the data-link layer called media access control (MAC)



Access control

- Examples of common packet mode multiple access protocols for wired multi-drop networks are:
  - CSMA/CD (used in Ethernet and IEEE 802.3)
  - Token bus (IEEE 802.4)
  - Token ring (IEEE 802.5)
  - Token passing (used in FDDI)

- Examples of common multiple access protocols that may be used in packet radio wireless networks are:
  - CSMA/CA (used in IEEE 802.11/WiFi WLANs)
  - Dynamic TDMA
  - Mobile Slotted Aloha (MS-ALOHA)
  - OFDMA

Random Access

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

- At each instance, a station that has data to send uses a procedure defined by the protocol to make a decision on whether or not to send
  - No scheduled time for a station to transmit, transmission is random among the stations (**Random access**)
  - No rules specify which station should send next. Stations compete with one another to access the medium (**Contention**)

- If more than one station tries to send at once, there is an access conflict—collision—and the frames may be corrupted

Random Access

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

ALOHA

- ALOHA, the earliest random access method, was developed at the University of Hawaii in early 1970[1]. It was designed for a radio (wireless) LAN, but it can be used on any shared medium.
  - ALOHA in Hawaiian means “HELLO”

- When a station sends data, another station may attempt to do so at the same time. The data from the two stations collide and become garbled.

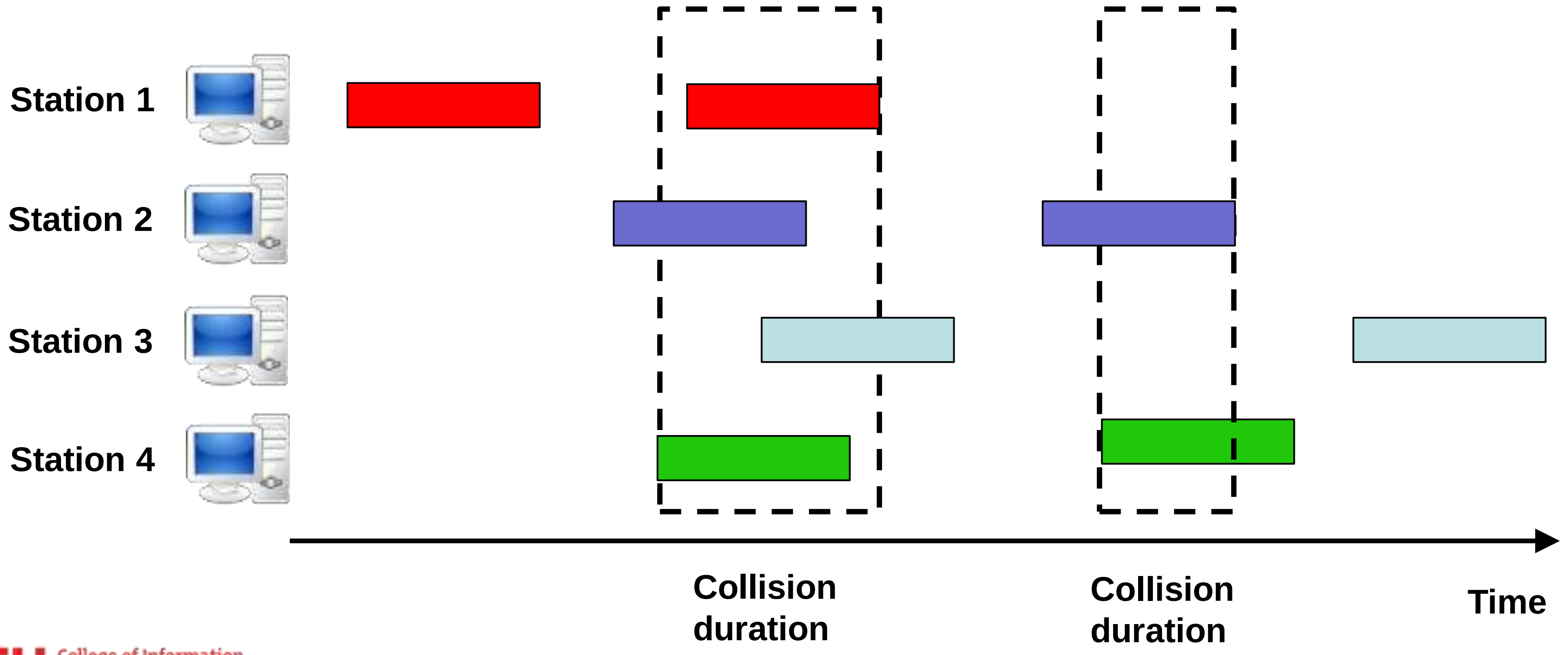
- There are mainly two types of ALOHA protocols: pure ALOHA and slotted ALOHA.

1. N. Abramson (1970). "The ALOHA System - Another Alternative for Computer Communications" (PDF). *Proc. 1970 Fall Joint Computer Conference*. AFIPS Press.

UAEU College of Information Technology

Pure ALOHA

- In pure ALOHA, each station sends a frame whenever it has a frame to send (multiple access)
- There is only one channel to share
  - Possibility of collision between frames from different stations.



Pure ALOHA Operation Concept

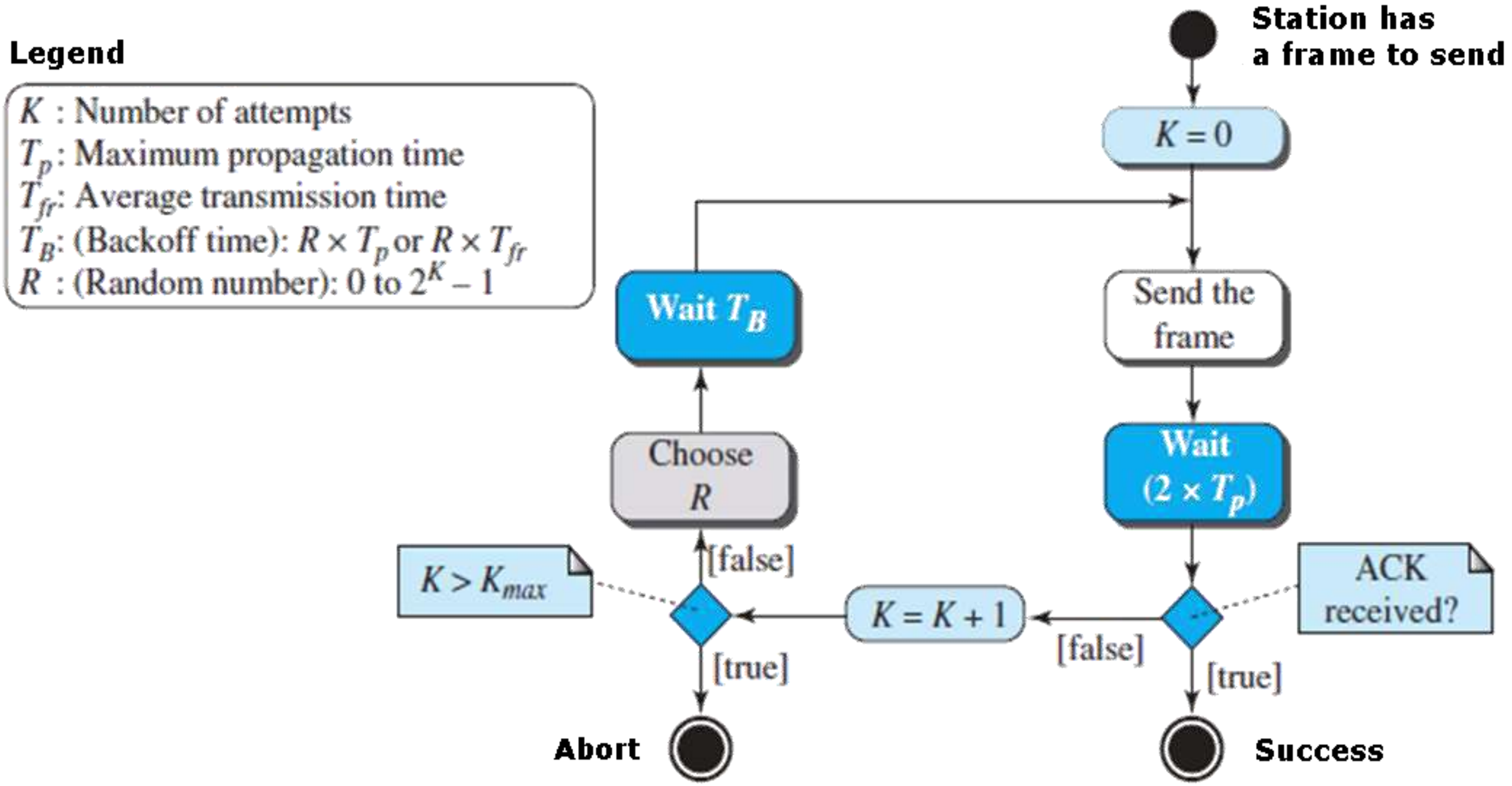
- If station has frame to send, it will send the frame

- When a station sends a frame, it expects the receiver to send an acknowledgment (ACK)
  - If ACK doesn't arrive after a time-out period, the station assumes that the frame (or the ACK) is lost 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 (**backoff time**) **before resending** its frame.

Procedure for Pure ALOHA Protocol

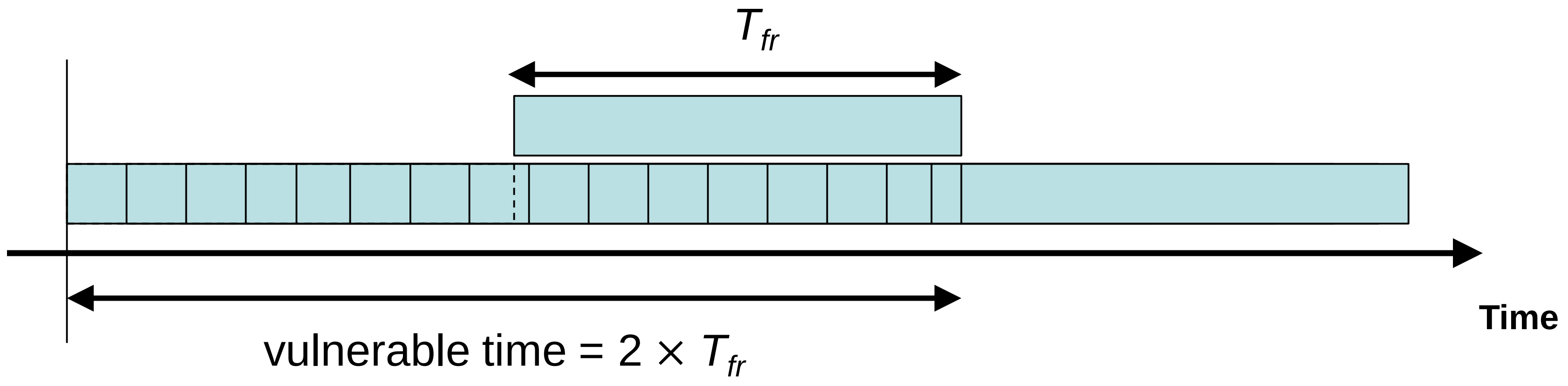


Legend
K : Number of attempts
T_p : Maximum propagation time
T_{fr}: Average transmission time
T_B: (Backoff time): $R \times T_p$ or $R \times T_{fr}$
R : (Random number): 0 to $2^K - 1$
Station has a frame to send
$K = 0$
Send the frame
Wait $(2 \times T_p)$
ACK received?
[true]
Success
[false]
$K = K + 1$
$K > K_{max}$
[true]
Abort
[false]
Choose R
Wait T_B

UAEU College of Information Technology

Vulnerable Time for Pure ALOHA Protocol

- **Vulnerable time**: the length of time in which there is a possibility of collision.



$$\text{Pure ALOHA}_{\text{vulnerable time}} = 2 \times T_{fr}$$

Vulnerable Time: An 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?

Solution

Average frame transmission time T_{fr} is

$$T_{fr} = 200 \text{ bits}/200 \text{ kbps} = 1 \text{ ms}.$$

The vulnerable time is 2 × 1 ms = 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 period (1 ms) that this station is sending.

Throughput of Pure ALOHA

- G: avg. no. of frames generated by the system during one frame transmission time

- Avg. no. of successfully transmitted frames is

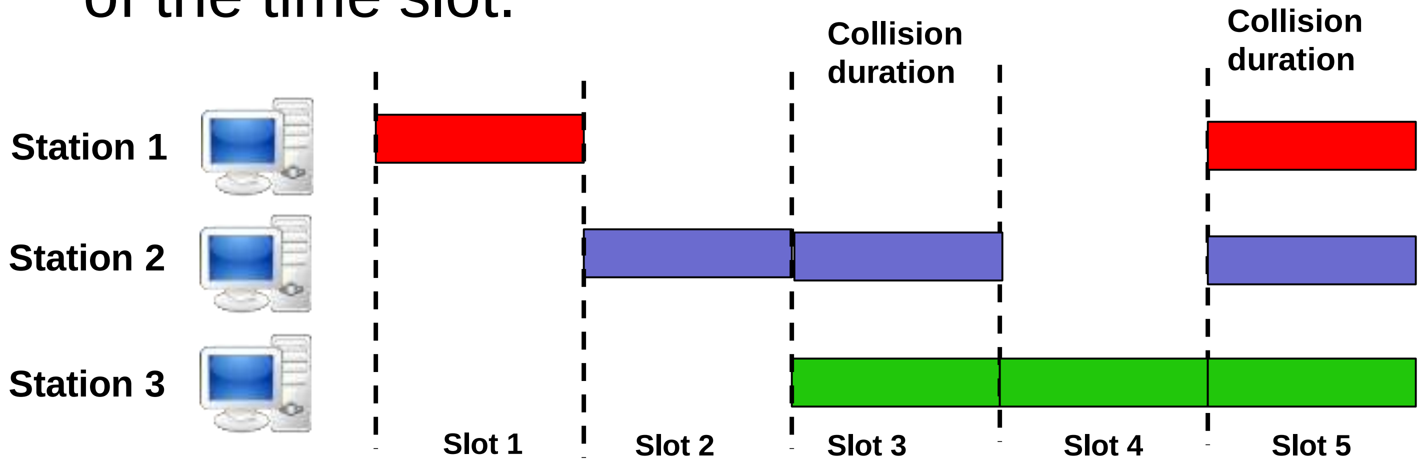
$$S = G \times e^{-2G}$$

- The maximum throughput S_{max} is 0.184, for G = 1/2.

- G = 1/2 produces the maximum throughput because the vulnerable time is 2 times the frame transmission time
  - If a station generates only one frame in this vulnerable time (and no other stations generate a frame during this time), the frame will reach its destination successfully

Slotted ALOHA

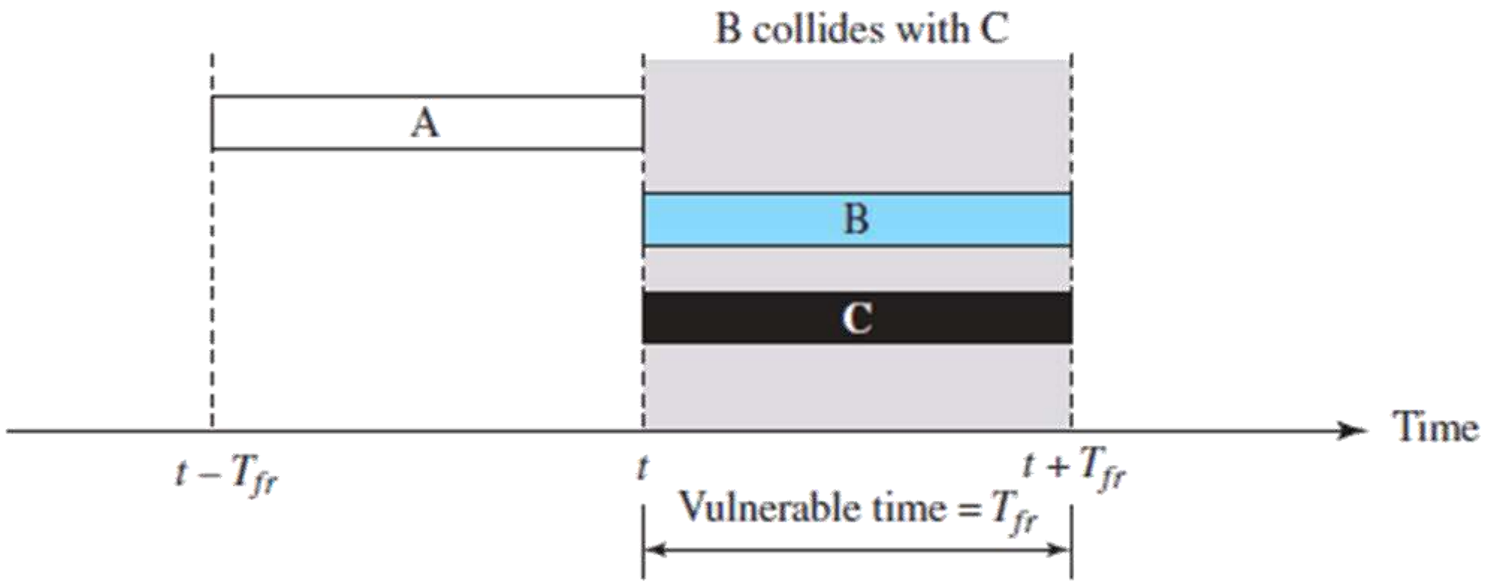
- Pure ALOHA has a vulnerable time of $2 \times T_{fr}$. This is so because there is no rule that defines when the station can send.
- A station may send soon after another station has started or just before another station has finished.
- In slotted ALOHA we divide the time into slots of T_{fr} seconds and force the station to send only at the beginning of the time slot.



Slotted ALOHA

- A station is allowed to send only at the beginning of the synchronized time slot
- If the station misses this moment, it must wait until the beginning of the next time slot
- Collision still possible 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}.

Vulnerable Time for Slotted ALOHA Protocol



Slotted ALOHA $_{\text{vulnerable time}}$ = T_{fr}

Throughput of Slotted ALOHA

- G: avg. no. of frames generated by the system during one frame transmission time
- Avg. no. of successful transmissions for slotted ALOHA is $S = G \times e^{-G}$
- The maximum throughput S_{max} is 0.368, when G = 1
- We expect G = 1 to produce maximum throughput because the vulnerable time is equal to the frame transmission time
  - If a station generates only one frame in this vulnerable time (and no other station generates a frame during this time), the frame will reach its destination successfully

Exercise

- Stations in an Aloha network send frames of size 1000 bits at the rate of 1 Mbps. What is the vulnerable time for this network if
  - Using Pure ALOHA
  - Using SLOTTED ALOHA

Probability Analysis

- To formulate the performance of a multiple-access network, we need a mathematical model.
- Poisson distribution, $\mathbf{p[x] = \lambda^{x} e^{-\lambda}/(x!)}$
  - p[x] is the probability of generating x number of frames in a period of time,
  - λ is the average number of generated frames during the same period of time.

Probability Analysis

- A multiple-access network with a large number of stations can be analyzed using the Poisson distribution. In a network with **N** stations, we assume each station has a frame to send during the frame transmission time (T_{fr}) with probability **p**. In such a network, a station is successful in sending its frame if the station has a frame to send during the vulnerable time and no other station has a frame to send during this period of time.
    - Find the probability that a station in a pure Aloha network can successfully send a frame during the vulnerable time.
    - Find the probability that a station in a slotted Aloha network can successfully send a frame during the vulnerable time.

Probability Analysis - Solution

- A multiple-access network with a large number of stations can be analyzed using the Poisson distribution. In a network with **N** stations, we assume each station has a frame to send during the frame transmission time (T_{fr}) with probability **p**. In such a network, a station is successful in sending its frame if the station has a frame to send during the vulnerable time and no other station has a frame to send during this period of time.
  - Find the probability that a station in a pure Aloha network can successfully send a frame during the vulnerable time.
  - Find the probability that a station in a slotted Aloha network can successfully send a frame during the vulnerable time.

The probability of success for a station is the probability that the rest of the network generates no frame during the vulnerable time. In other words, we are looking for ***p*[0]** in the Poisson distribution.

a. For a pure Aloha network, the vulnerable time is (2 x T_{fr}), which means that $\lambda = 2G$.

$$\textbf{P [success for a frame]} = p[0] = ((2G)^{0} \times e^{-2G}) / (0!) = e^{-2G}$$

b. For a slotted Aloha network, the vulnerable time is (T_{fr}), which means that $\lambda = G$.

$$\textbf{P [success for a frame]} = p[0] = ((G)^{0} \times e^{-G}) / (0!) = e^{-G}$$

UAEU College of Information Technology

Problem

There are only three active stations in a slotted Aloha network: A, B, and C. Each station generates a frame in a time slot with the corresponding probabilities $p_A = 0.2$, $p_B = 0.3$, and $p_C = 0.4$, respectively.

a. What is the throughput of each station?

b. What is the throughput of the network?

Solution

We can first find the throughput for each station. The throughput of the network is the sum of the throughputs.

a. The throughput of each station is the probability that the station has a frame to send and other stations have no frame to send.

$$S_A = p_A (1-p_B) (1-p_C) = 0.2 \times 0.7 \times 0.6 \approx 0.084$$

$$S_B = p_B (1 -p_A) (1-p_C) = 0.3 \times 0.8 \times 0.6 \approx 0.144$$

$$S_C = p_C (1-p_A) (1 -p_B) = 0.4 \times 0.8 \times 0.7 \approx 0.224$$

b. The throughput of the network is the sum of the throughputs.

$$S = S_A + S_B + S_C \approx 0.452$$

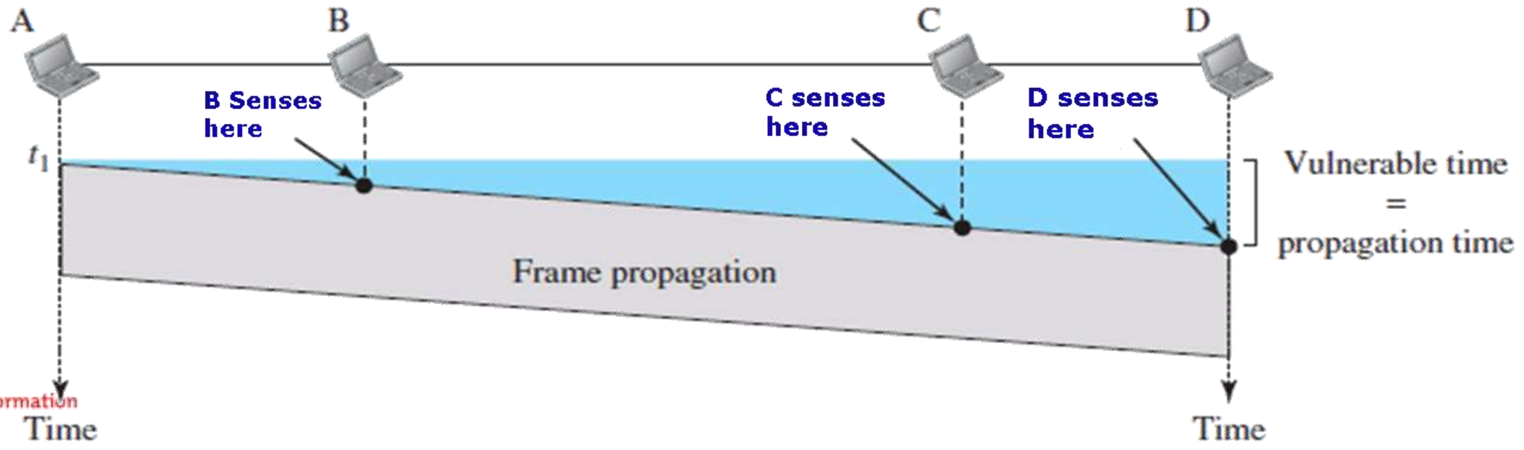
CSMA

- The chance of collision can be reduced if a station senses the medium before trying to use it
- Carrier sense multiple access (CSMA) requires that each station first listen to the medium (or check the state of the medium) before sending
- CSMA can reduce the possibility of collision, but it cannot eliminate it. WHY ?

The possibility of collision still exists because of propagation delay; when a station sends a frame, it still takes time (although very short) for the first bit to reach every station and for every station to sense it.

CSMA: Vulnerable Time

- The vulnerable time for CSMA is the propagation time T_p
  - Time needed for a signal to travel from one end of the medium to the other
- When a station sends a frame and any other station tries to send a frame during this time, a collision will result
- But if the first bit of the frame reaches the end of the medium, every station will already have heard the bit and will refrain from sending



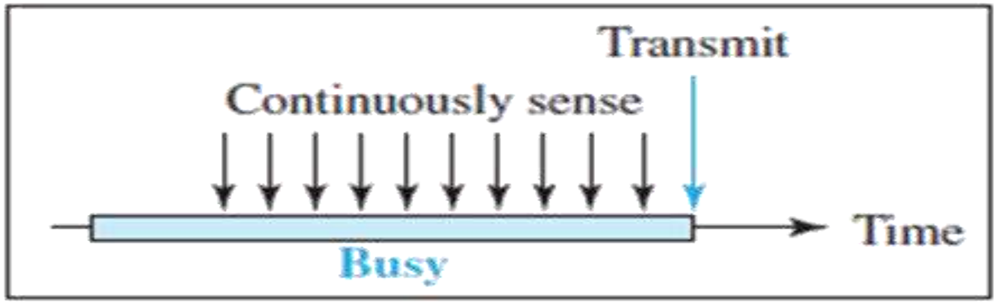
Exercise

- Assume the propagation delay in a broadcast network is 5 μs and the frame transmission time is 10 μs.
  - How long does it take for the first bit to reach the destination?
  - How long does it take for the last bit to reach the destination after the first bit has arrived?
  - How long is the network involved with this frame?

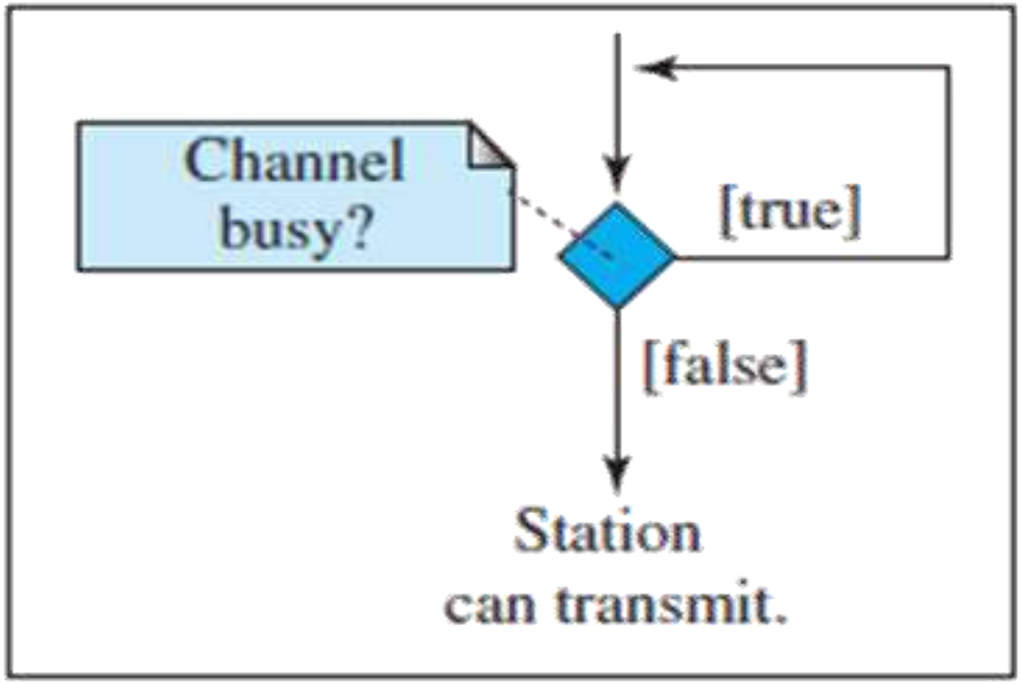
CSMA Persistence Methods : *1-Persistent*

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.



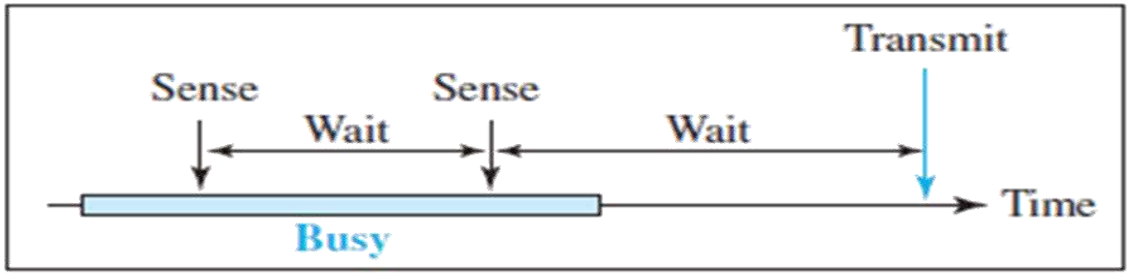
1-Persistent

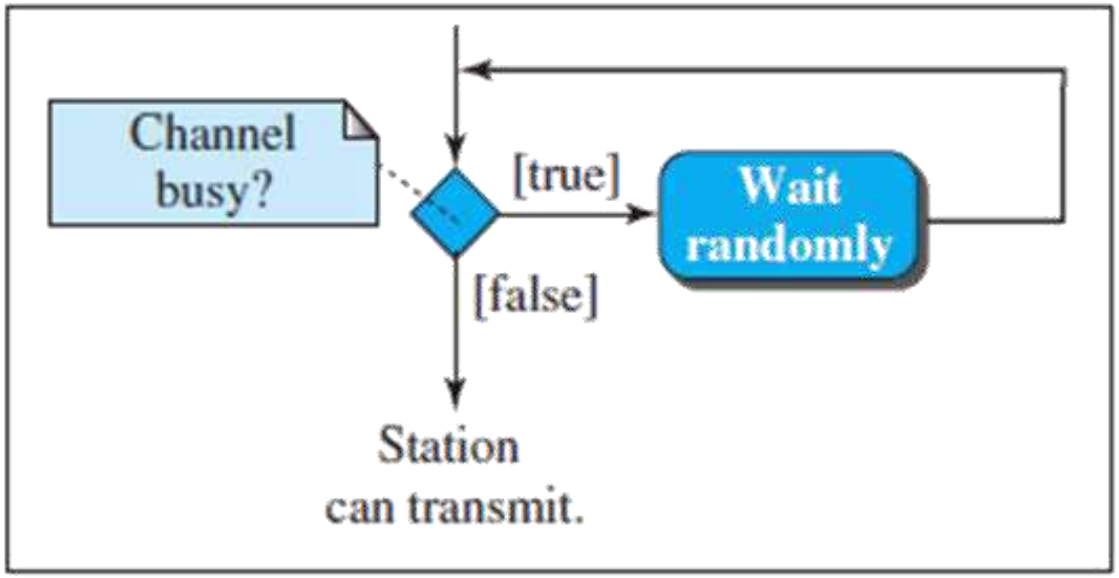


CSMA Persistence Methods: Nonpersistent

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.

This method reduces the efficiency of the network because the medium remains idle when there may be stations with frames to send.





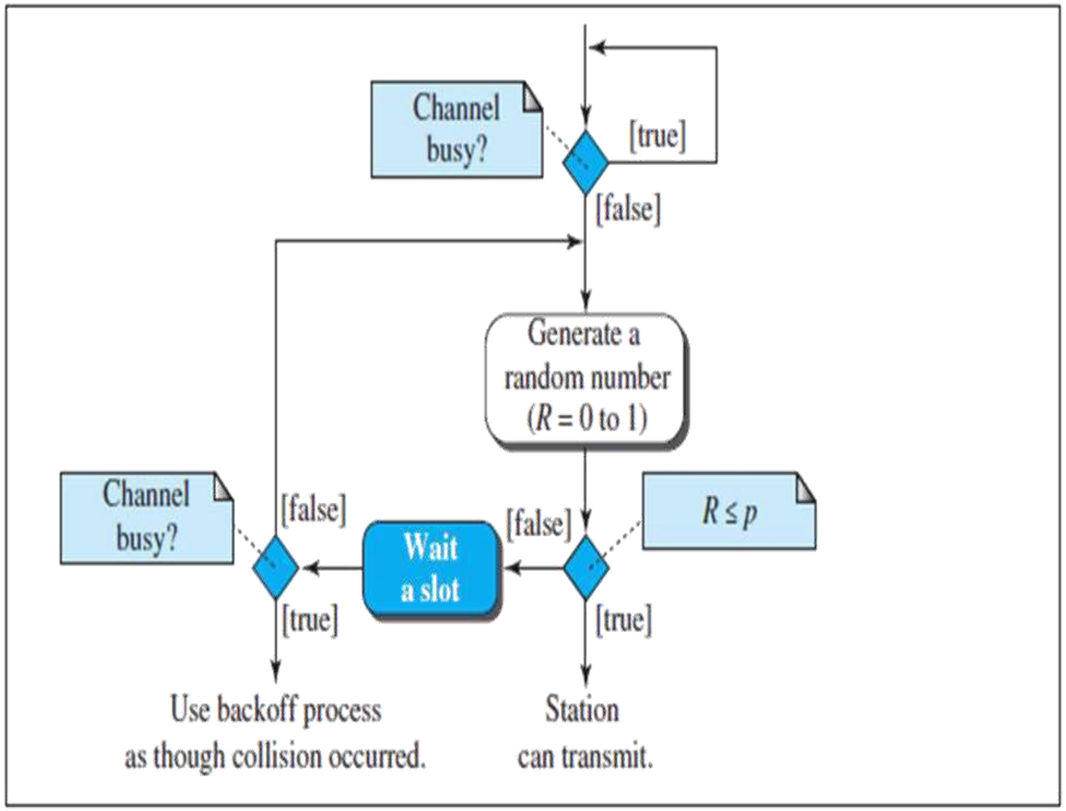
CSMA Persistence Methods : *P-Persistent*

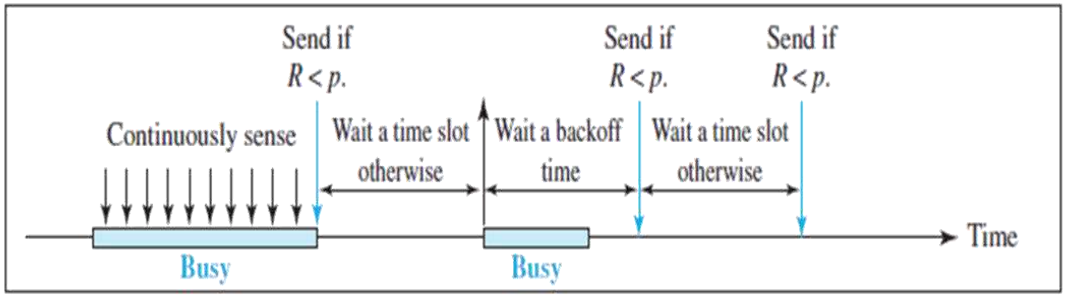
After the station finds the line idle it follows these steps:

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.

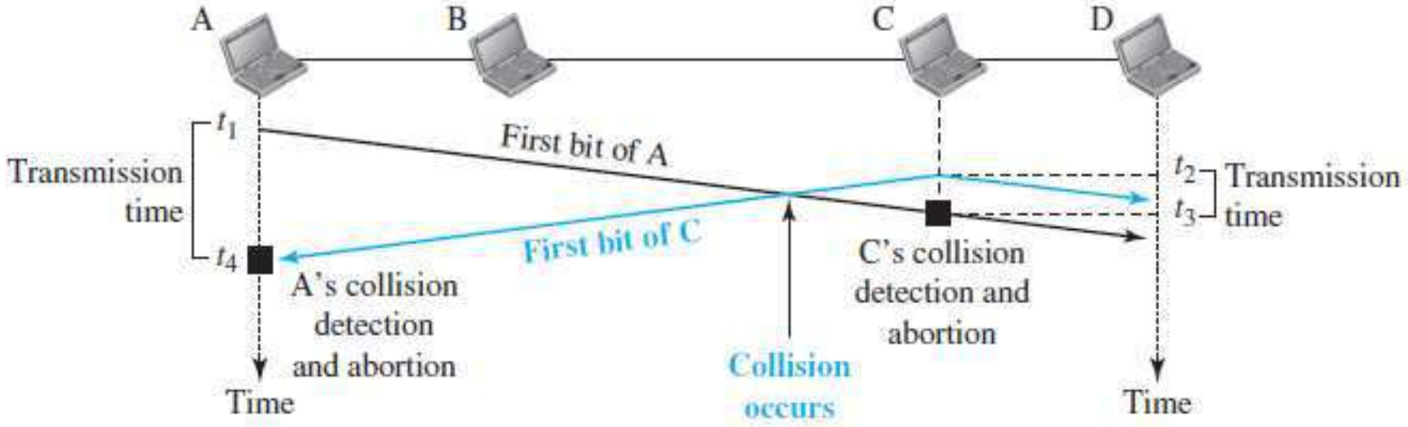
- **a.** If the line is idle, it goes to step 1.
- **b.** If the line is busy, it acts as though a collision has occurred and uses the backoff procedure.



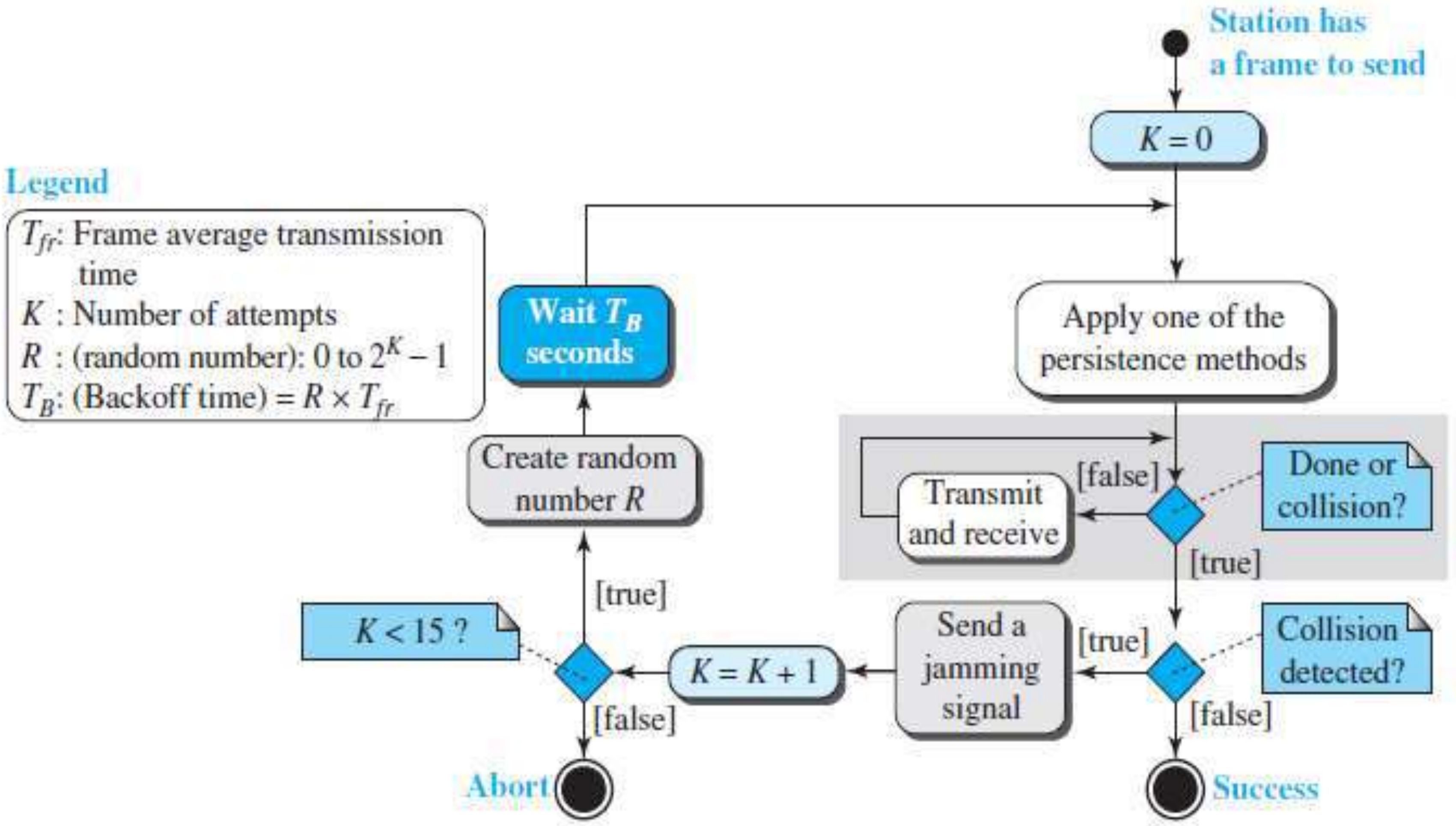


CSMA/CD

- The CSMA method does not specify the procedure following a collision
- Carrier sense multiple access with collision detection (CSMA/CD) augments the algorithm to handle the collision
- Station monitors the medium after it sends a frame
  - If there is a collision, the frame is sent again



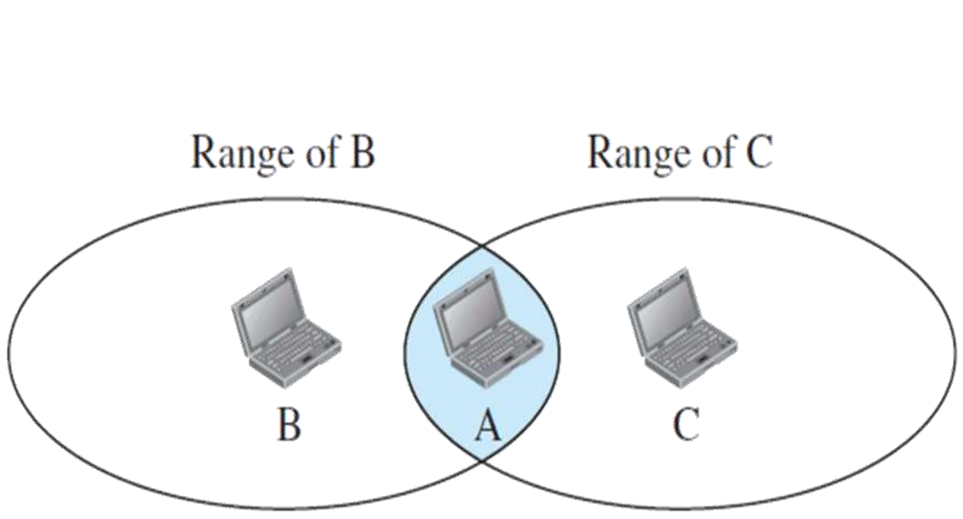
Flow diagram for the CSMA/CD



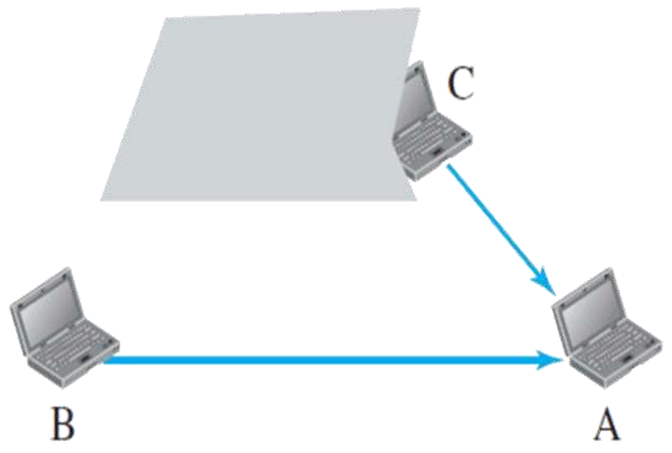
UAEU College of Information Technology

CSMA/CA

- Carrier sense multiple access with collision avoidance (CSMA/CA) was invented for wireless networks. WHY?



a. Stations B and C are not in each other's range.



b. Stations B and C are hidden from each other.

CSMA/CA

- Collisions are avoided through the use of three strategies:
  - the interframe space
  - the contention window and
  - ACKs
- **Interframe Space (IFS)**
  - Collisions are avoided by deferring transmission even if the channel is found idle
  - 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
- **The IFS variable can also be used to prioritize stations or frame types. A station that is assigned a shorter IFS has a higher priority.**

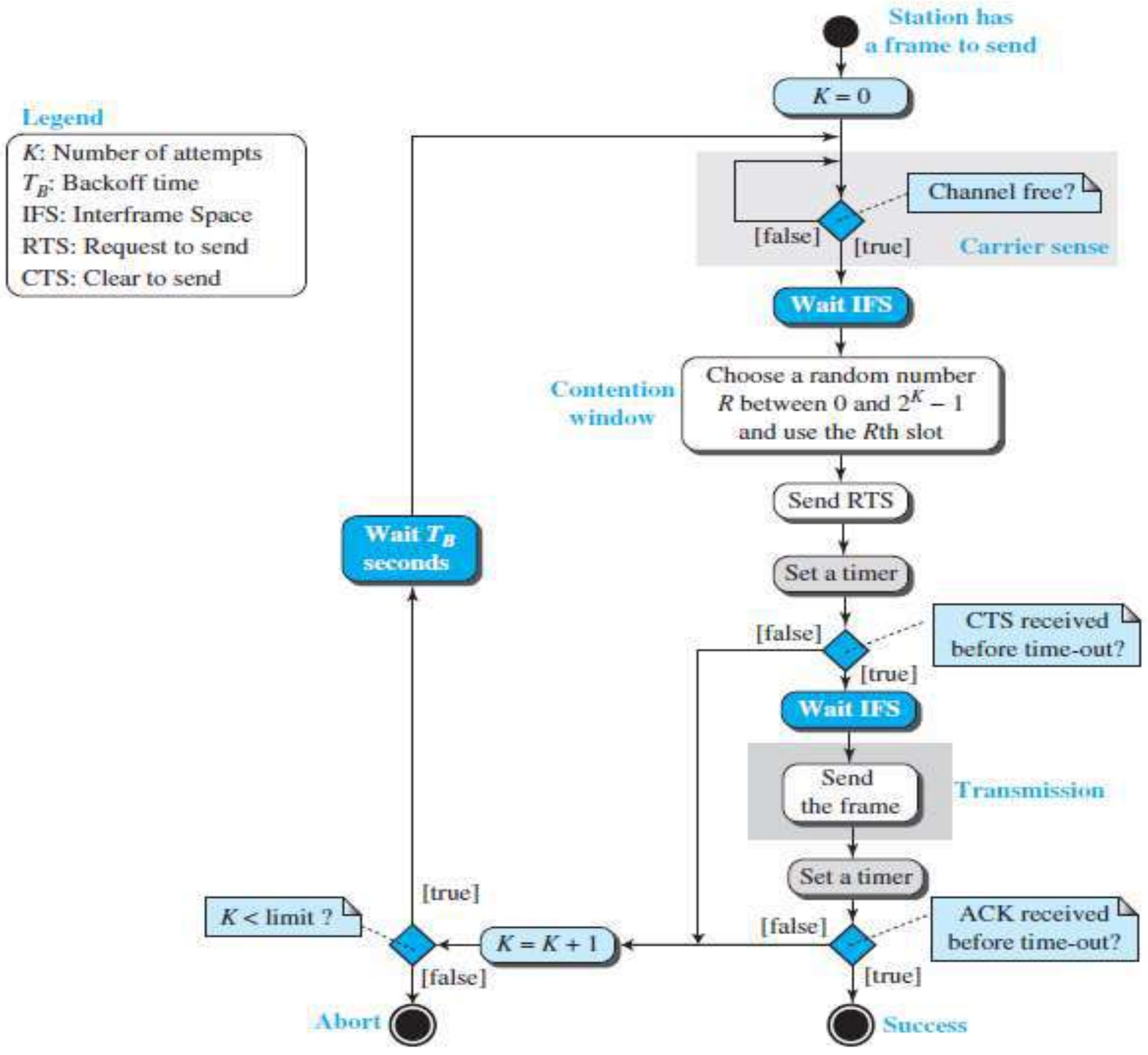
CSMA/CA

- **Contention window (CW)**
  - 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 backoff strategy
  - CW is set to one slot the first time and then doubles each time the station cannot detect an idle channel after the IFS time
- Video illustrating the idea: https://youtu.be/MgbLTgao3yw

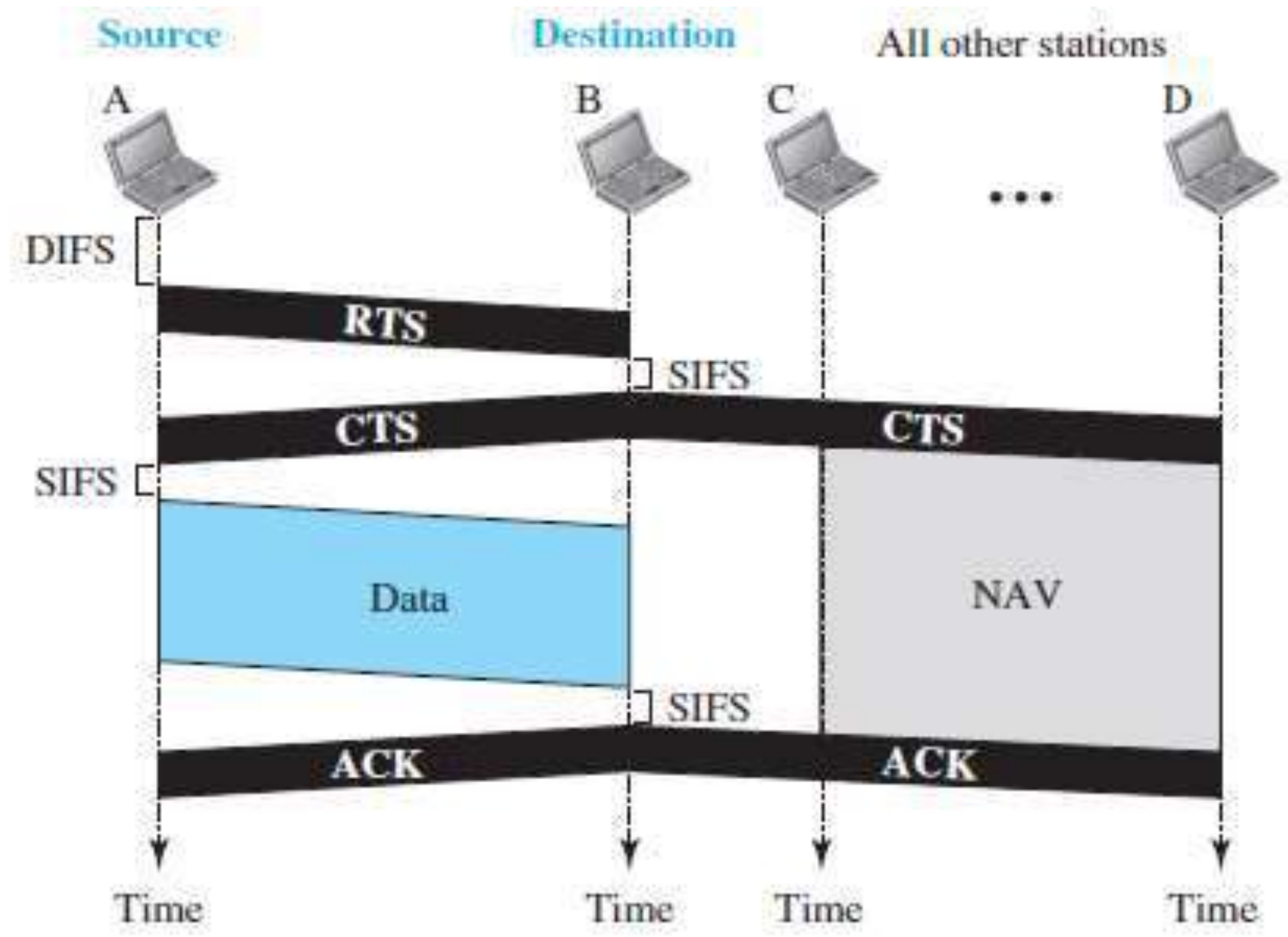
CSMA/CA

- **Acknowledgment (ACK)**
  - With all these precautions, there still may be a collision
  - The data may be corrupted during the transmission
  - A positive acknowledgment and the time-out timer can help guarantee that the receiver has received the frame
  - Shorter IFS for the ACK to increase its priority

CSMA/CA Flow Chart



CSMA/CA Frame Exchange Time Line



CSMA/CA Frame Exchange Time Line

- When a station sends an RTS frame, it includes the duration of time that it needs to occupy the channel
- The stations that are affected by this transmission create a timer called a **network allocation vector (NAV)** that shows how much time must pass before these stations are allowed to check the channel for idleness
- Each station, before sensing the physical medium to see if it is idle, first checks its NAV to see if it has expired

CSMA/CA Problem

- What is the Collision During Handshaking Problem?

- What is the Exposed Station Problem?

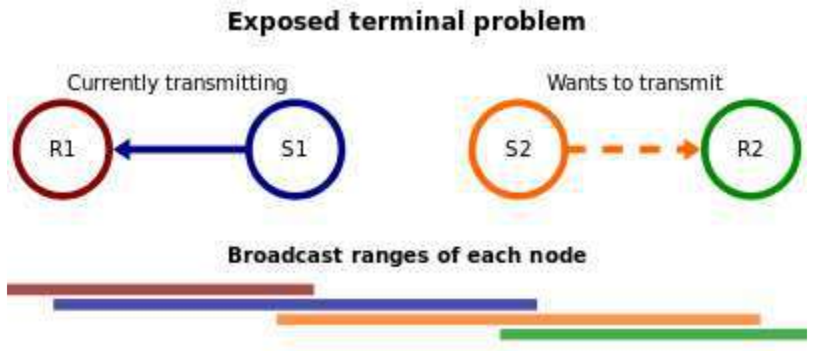


Image Source: https://en.wikipedia.org/wiki/Exposed_node_problem

Controlled Access

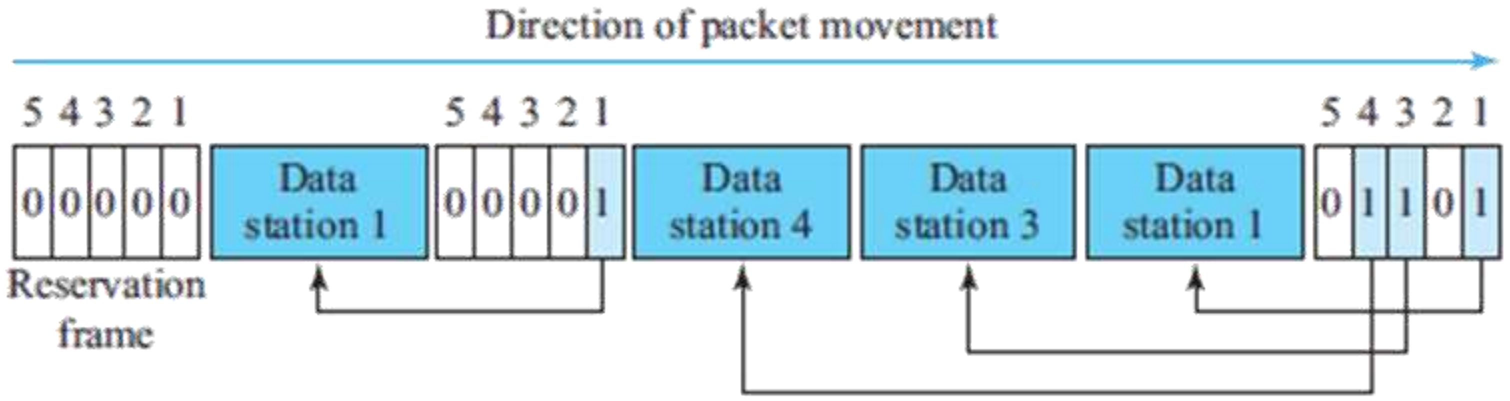
- The stations consult one another to find which station has the right to send

- A station cannot send unless it has been authorized by other stations

- We discuss three controlled-access methods:
  - Reservation
  - Polling
  - Token Passing

Reservation

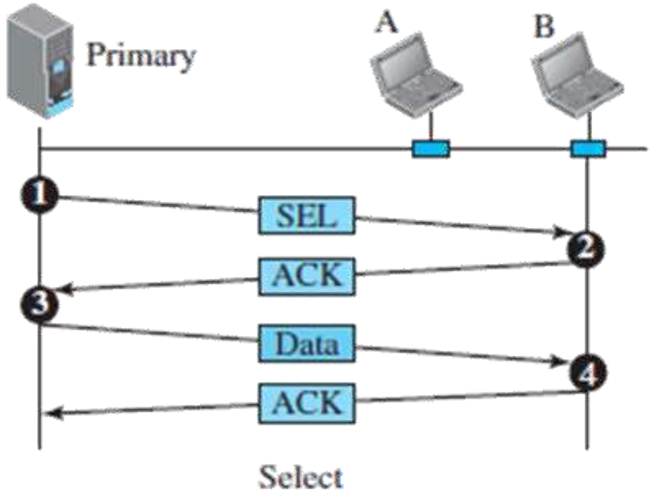
- In the **reservation method**, a station needs to make a reservation before sending data.
- Time is divided into intervals. In each interval, a reservation frame precedes the data frames sent in that interval.
- If there are N stations in the system, there are exactly N reservation minislots in the reservation frame.
- When a station needs to send a data frame, it makes a reservation in its own minislot. The stations that have made reservations can send their data frames after the reservation frame.

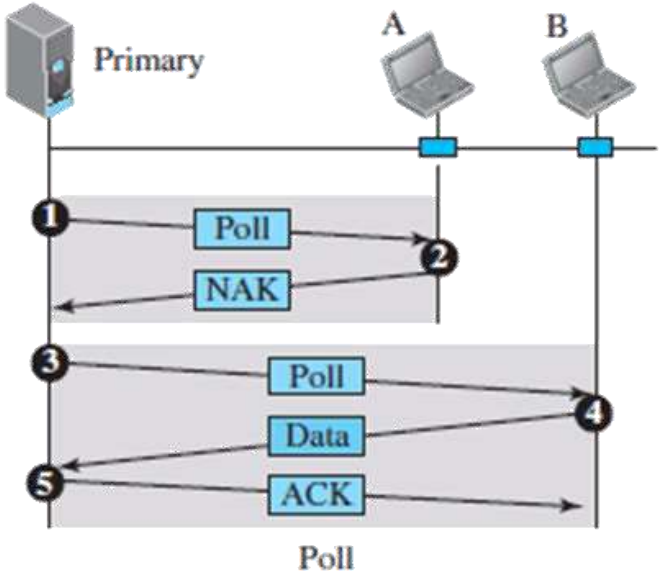


UAEU College of Information Technology

Polling

- Polling works with topologies in which one device is designated as a primary station and the other devices are secondary stations.
- All data exchanges must be made through the primary device even when the destination is a secondary device.
- The primary device controls the link; the secondary devices follow its instructions.





Token Passing 1/2

- In the token-passing method, the stations in a network are organized in a logical ring.
- For each station, there is a predecessor and a successor.
  - Predecessor: a station which is logically before the station in the ring
  - Successor: a station which is after the station in the ring
- The current station is the one that is accessing the channel now. The right to this access has been passed from the predecessor to the current station.
- The right will be passed to the successor when the current station has no more data to send.

Token Passing 2/2

- A special packet called a token circulates through the ring. The possession of the token gives the station the right to access the channel and send its data.
- The station cannot send data until it receives the token again in the next round.
- In this process, when a station receives the token and has no data to send, it just passes the token to the next station.
- Token management is needed for this access method:
  - Stations must be limited in the time they can have possession of the token.
  - The token must be monitored to ensure it has not been lost or destroyed.
  - Token management can make low-priority stations release the token to high-priority stations.

Random Access vs. Controlled Access

- What do you think about Delay, Throughput, Fairness, Sensitivity to node failure for these two access schemes?

	Random Access (ALOHA, CSMA)	**Controlled Access**
Delay	small under light loads	longer but generally less variable between stations
Throughput	sufficient under light load, drops significantly under heavy loads	increases under heavy load
Fairness	not guaranteed	Guaranteed
Sensitivity to node failure	Small	high, particularly in polling and token ring systems

UAEU College of Information Technology

Channelization

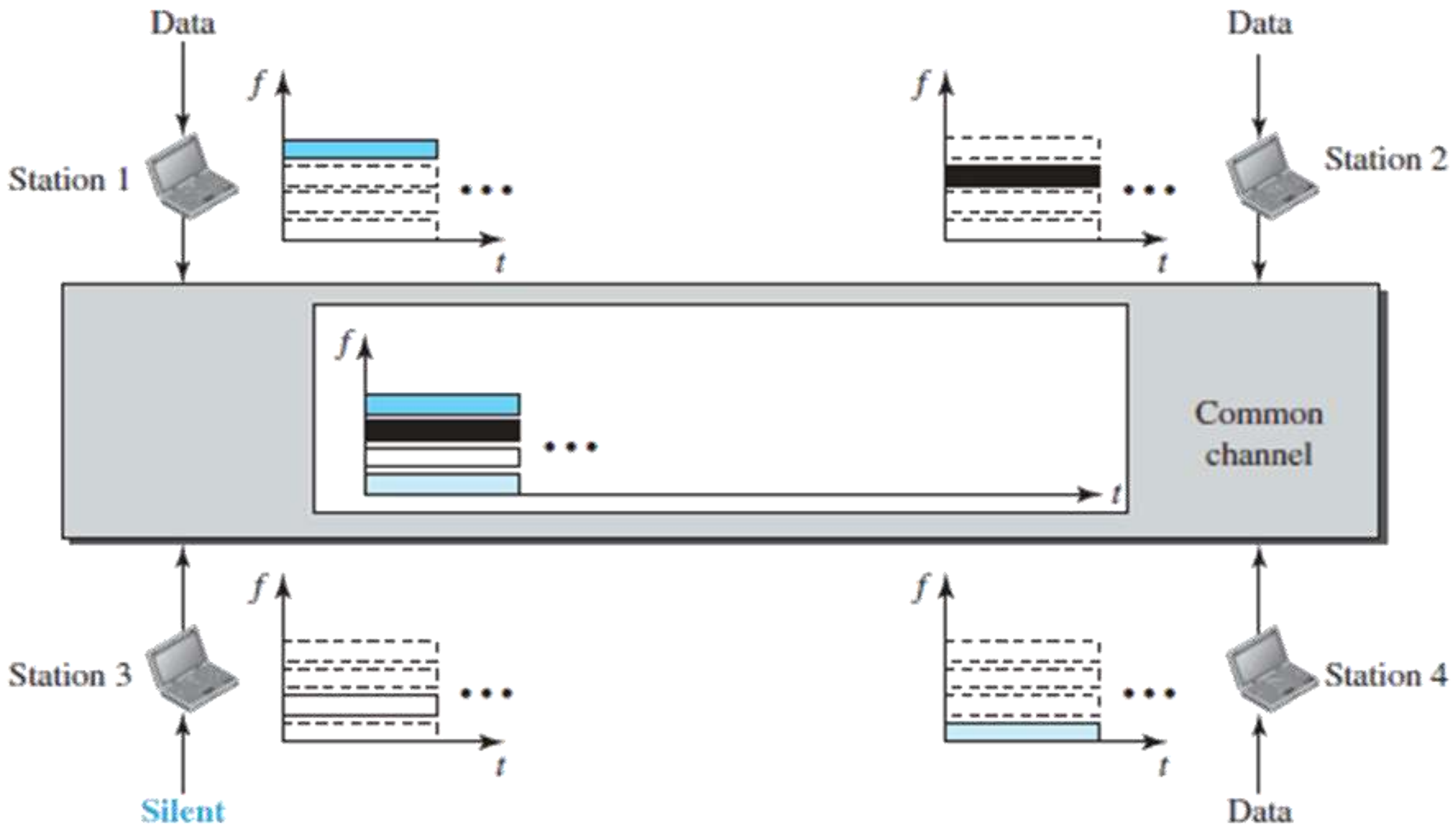
- Multiple-access method in which the available bandwidth of a link is shared in <u>time</u>, <u>frequency</u>, or through <u>code</u>, among different stations

- We will discuss three channelization protocols:
  - Frequency-Division Multiple Access (FDMA)
  - Time-Division Multiple Access (TDMA)
  - Code-division multiple access (CDMA)

FDMA

- The available bandwidth is divided into frequency bands
- Each station is allocated a band to send its data
  - Each band is reserved for a specific station and it belongs to the station all the time
- Each station also uses a band-pass filter to confine the transmitter frequencies
- To prevent station interferences, the allocated bands are separated from one another by small guard bands
- FDMA Advantage:
  - easy to implement – no need for node synchronization
- FDMA Disadvantage
  - guard bands ensure separation, but waste bandwidth
  - # of simultaneously served users ≤ # of channels

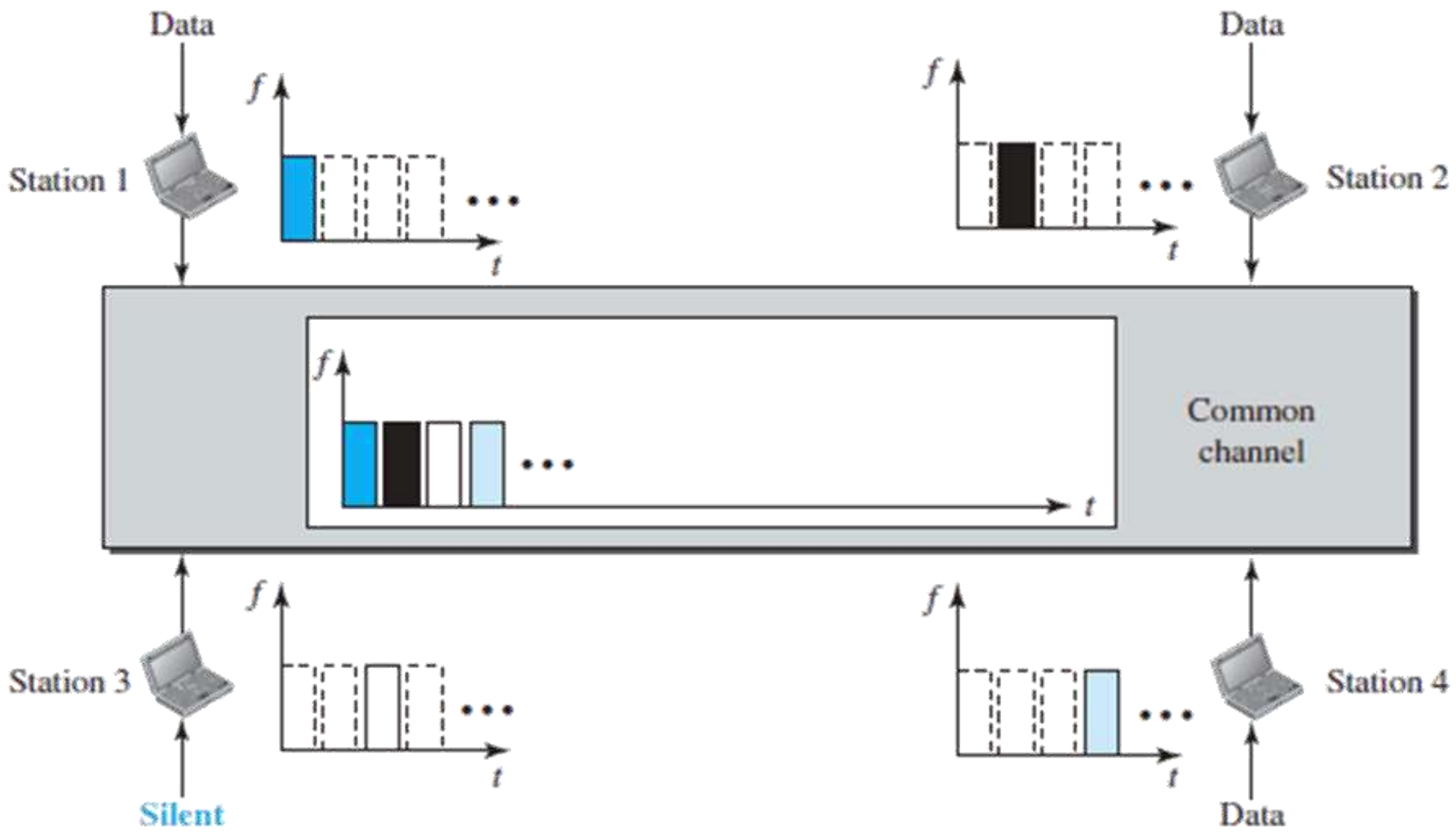
Idea of FDMA


Data
Data
Station 1
Station 2
f
t
Common channel
Station 3
Station 4
Silent
Data

TDMA

- Stations share the bandwidth of the channel in time
- Each station is allocated a time slot during which it can send data. Each station transmits its data in its assigned time slot.
- TDMA requires synchronization between the different stations
  - Station needs to know the beginning of its slot and the location of its slot
- TDMA Advantage:
  - Can accommodate a wider range of bit rates
    - allocate several slots to a station or
    - allow slots to be variable in duration
- TDMA Disadvantage:
  - Stations must be synchronized to a common clock

Idea of TDMA


Data
Station 1
f
t
Data
Station 2
f
t
Common channel
f
t
Station 3
f
t
Silent
Station 4
f
t
Data

CDMA

- One channel carries all transmissions simultaneously
- CDMA differs from FDMA in that only one channel occupies the entire bandwidth of the link
- CDMA differs from TDMA in that all stations can send data simultaneously; there is no timesharing
- CDMA is based on coding theory. Each station is assigned a code, which is a sequence of numbers called chips
- Chips are not chosen randomly; they are carefully selected

CDMA: Chips Properties

Chips are called orthogonal sequences and have the following properties:

- Each sequence is made of N elements, where N is the number of stations.
- If we multiply a sequence by a number, every element in the sequence is multiplied by that element. This is called multiplication of a sequence by a scalar. For example,

$$2 \bullet [+1\ +1\ -1\ -1] = [+2\ +2\ -2\ -2]$$

- If we multiply two equal sequences, element by element, and add the results, we get N, where N is the number of elements in each sequence. This is called the inner product of two equal sequences. For example,

$$[+1\ +1\ -1\ -1] \bullet [+1\ +1\ -1\ -1] = 1 + 1 + 1 + 1 = 4$$

CDMA: Chips Properties

- If we multiply two different sequences , element by element, and add the results, we get 0. This is called the inner product of two different sequences. For example,

$$[+1\ +1\ -1\ -1] \bullet [+1\ +1\ +1\ +1] = 1 + 1 - 1 - 1 = 0$$

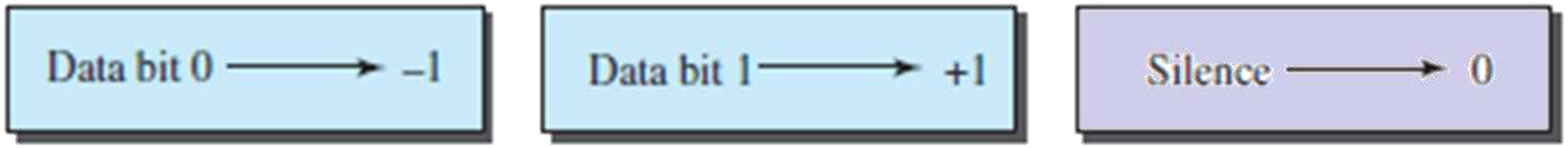
- Adding two sequences means adding the corresponding elements. The result is another sequence. For example,

$$[+1\ +1\ -1\ -1] + [+1\ +1\ +1\ +1] = [+2\ +2\ \ 0\ \ 0]$$

- **In plain English, the key chip properties to remember are:**
  - **If we multiply a code by another code, we get a 0**
  - **If we multiply a code by itself, we get the number of stations N**

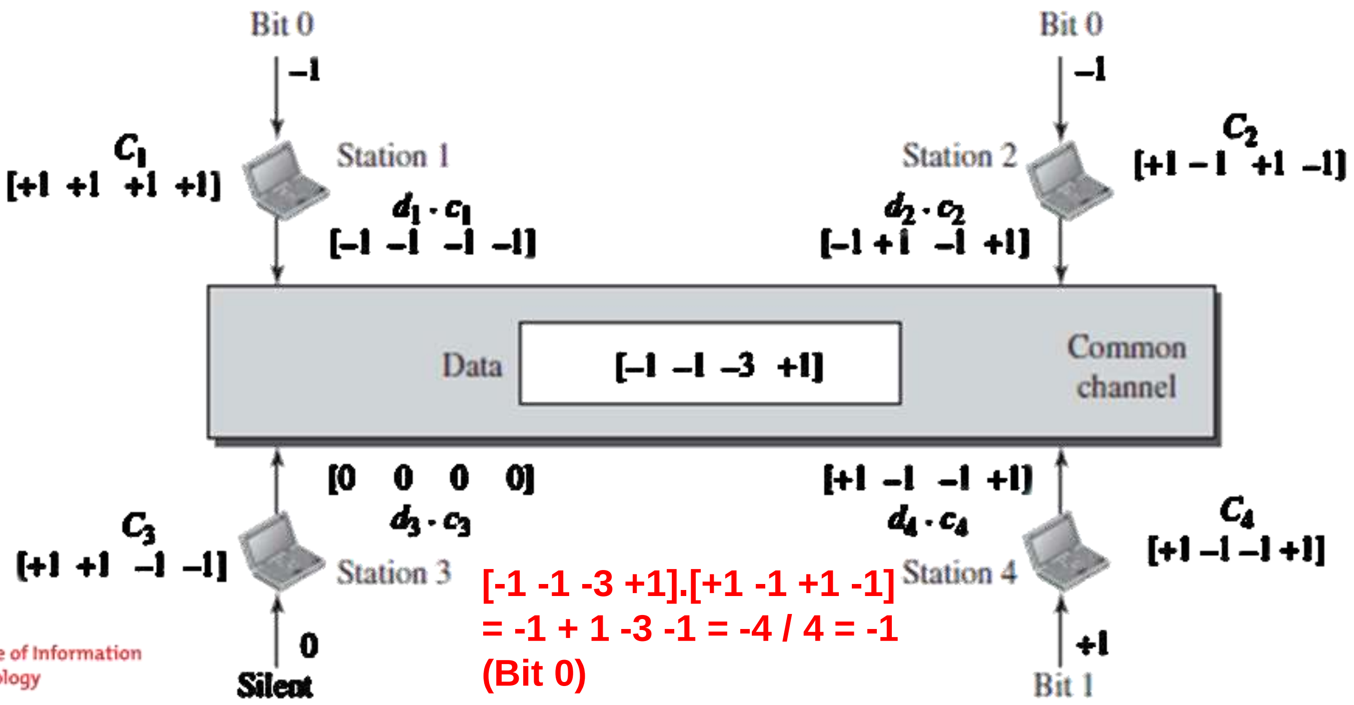
CDMA: Data Representation

- If a station needs to send a 0 bit, it encodes it as −1;
- If it needs to send a 1 bit, it encodes it as +1.
- When a station is idle, it sends no signal, which is interpreted as a 0.



CDMA: Encoding and Decoding

- Each station multiplies the corresponding number by its chip (its orthogonal sequence), which is unique for each station
- The result is a new sequence which is sent to the channel. For simplicity, we assume that all stations send the resulting sequences at the same time
- The sequence on the channel is the sum of all four sequences



[-1 -1 -3 +1].[+1 -1 +1 -1] = -1 + 1 -3 -1 = -4 / 4 = -1 (Bit 0)

To hear Station2, Station 3 multiplies data on channel with c2.

This yields c2.c1.d1 + c2.c2.d2 + c2.c3.d3 + c2.c4.d4 = 4d2

And divides data on the channel by 4 (-4/4 = -1)

UAEU College of Information Technology

CDMA: Sequence Generation

- To generate chip sequences, we use a Walsh table, which is a two-dimensional table with an equal number of rows and columns.
- In the Walsh table, each row is a sequence of chips.

$$W_1 = \begin{bmatrix} +1 \end{bmatrix} \quad W_{2N} = \begin{bmatrix} W_N & W_N \\ W_N & \overline{W_N} \end{bmatrix}$$

a. Two basic rules

$$W_2 = \begin{bmatrix} +1 & +1 \\ +1 & -1 \end{bmatrix} \quad W_4 = \begin{bmatrix} +1 & +1 & +1 & +1 \\ +1 & -1 & +1 & -1 \\ +1 & +1 & -1 & -1 \\ +1 & -1 & -1 & +1 \end{bmatrix}$$

b. Generation of W_2 and W_4

UAEU College of Information Technology

Summary

- We discussed
  - Random Access Protocols
    - Pure and slotted ALOHA, CSMA, CSMA/CD, CSMA/CA
  - Controlled Access Protocols
    - Reservation, Polling, Token Passing
  - Channelization
    - FDMA, TDMA, CDMA

- That concludes our discussion on the Data Link Layer!