

Data Link Layer

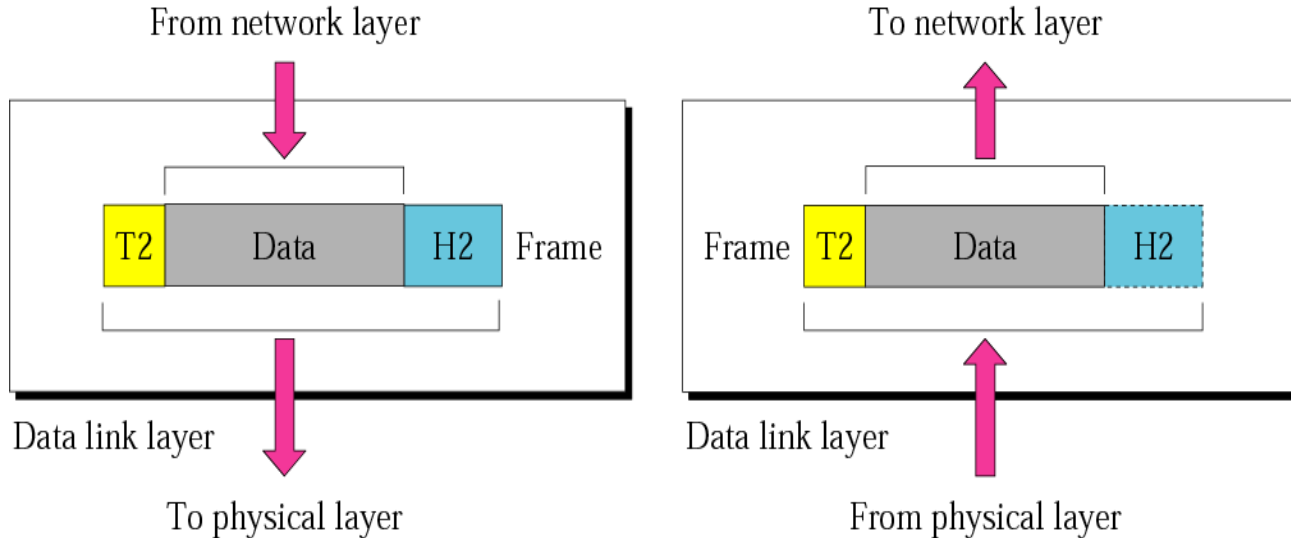
- Data Link Layer Responsibilities
- Error Detection and Correction
- Media Access Control

Data Link Layer

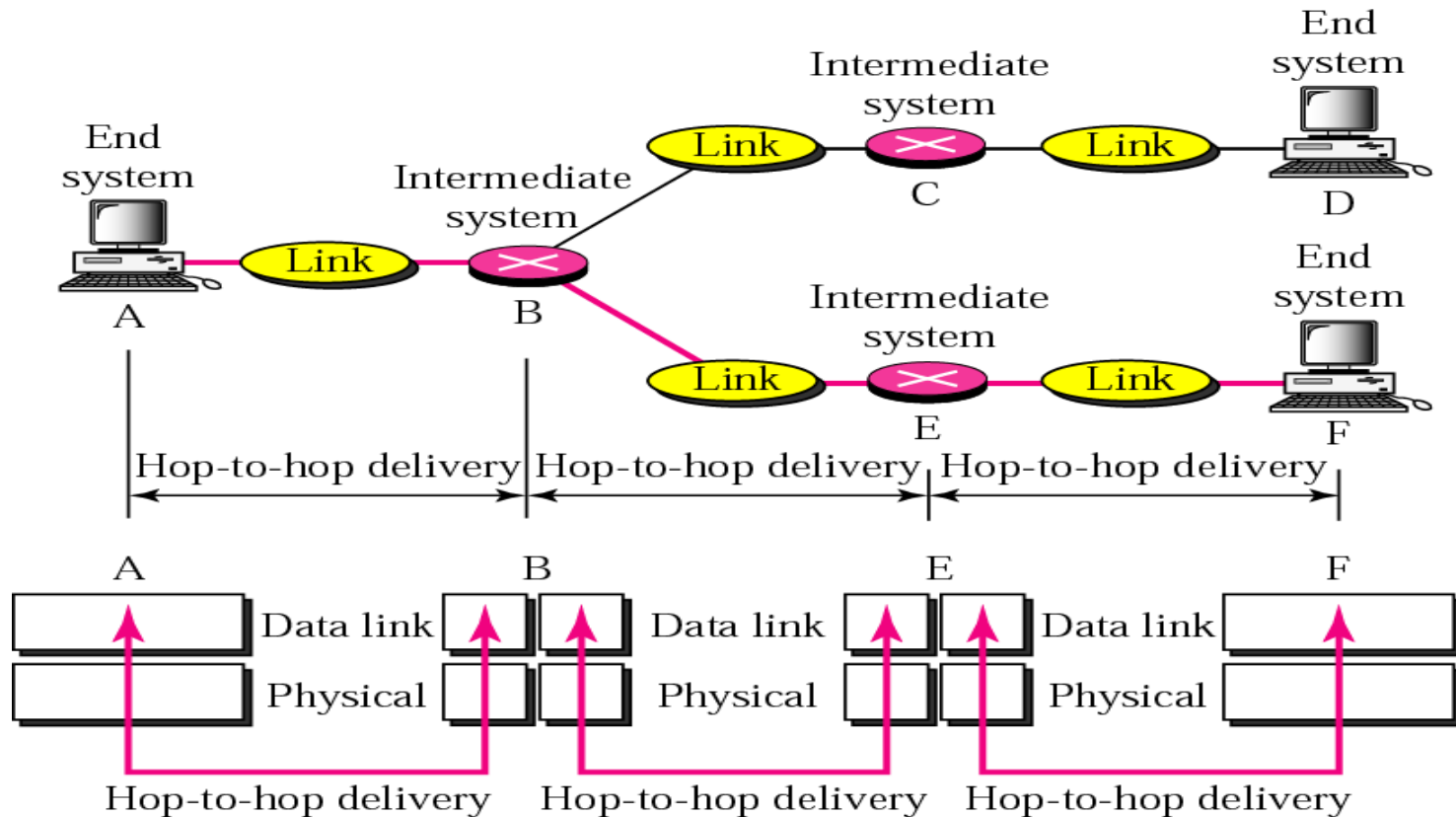
- Data Link Layer is second layer of OSI Layered Model.
- The data link layer transforms the physical layer, a raw transmission facility, to a reliable link.
- **It makes the physical layer appear error-free to the upper layer (network layer).**
- The Data Link Layer protocols are Ethernet, token ring, FDDI and PPP

Data Link Layer

The data link layer is responsible for moving frames from one hop (node) to the next.



Hop-to-hop (node-to-node) delivery by data link layer



Data Link Layer Responsibilities

■ Physical addressing.

- 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.*

Data Link Layer Responsibilities

■ Framing

- The data link layer divides the **stream of bits** received from the network layer into **manageable data units called frames.**

Data Link Layer Responsibilities

■ Synchronization

- When data frames are sent on the link, both machines must be synchronized in order to transfer to take place.

Data Link Layer Responsibilities

- **Flow control.**

- If the rate at which the data are absorbed by **the receiver is less than** the rate at which data are produced in **the sender**, the data link layer imposes a **flow control** mechanism **to avoid overwhelming the receiver.**

Contd.

■ Error control

- The data link layer adds reliability to the physical layer by adding mechanisms to detect and retransmit damaged or lost frames.
 - It also uses a mechanism to recognize duplicate frames.
 - Error control is normally achieved through a trailer added to the end of the frame.

Contd.

- **Access control**

- When two or more devices are connected to the same link, data link layer protocols are necessary to determine which device has control over the link at any given time.

Framing

- Definition and Function
- frame structure

Framing

- The data link layer, needs to pack bits into frames, so that each frame is distinguishable from another.
- Framing separates a message from one source to a destination, 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.

 NB: Addressing here is about the next node in the LAN

Contd.

- a very large Frame, making flow and error control very inefficient.
- even a single-bit error would require the retransmission of the whole message.
- When a message is divided into smaller frames, a single-bit error affects only that small frame.
- Frames can be of fixed or variable size.

Layer 2 frame structure

Start Frame (Flag)	Header		Data	Trailer	Stop Frame (Flag)
	Address	Type/Length		FCS	
		h			

1. Flag field:

- an 8-bit sequence
- identifies both the beginning and the end of a frame
- serves as a synchronization pattern for the receiver.

2. Data field:

- contains the user's data from the network layer.
- Its length can vary from one network to another.

Layer 2 frame structure

Start Frame (Flag)	Header		Data	Trailer	Stop Frame (Flag)
	Address	Type/Length		FCS	
		h			

3. FCS field:

- The frame check sequence (FCS) is the error detection field.
- It can contain either a 2- or 4-byte **ITU-T CRC**.

3. Address field:

- contains the address of the secondary station.
- If a primary station created the frame, it contains a to address.
- If a secondary creates the frame, it contains a from address.
- An address field can be 1 byte or several bytes long, depending on the needs of the network

Error Detection and Correction

- Error Detection
- Error Correction

Error Causes

- Errors in transmitted data can occur for a variety of reasons.
 1. Some errors are due to **equipment failure**.
 2. Some errors are due **dispersion in optical fibers** (i.e. light pulses spread out).
 3. Some errors are due to **attenuation** (loss of signal power over a line).
 4. Most errors are due to thermal **noise** that occurs naturally on the line.

Errors in Data

- Data is sent in the form of binary numbers.
- The binary numbers consist bites-which are either 0 or 1.
- There are four possible ways that noise can affect a bit:
 - If a bit is 0, the noise can affect it so it stays 0
no error
 - If a bit is 0, the noise can change it to 1
error
 - If a bit is 1, the noise can affect it so it stays 1
no error
 - if a bit is 1, the noise can change it to 0
error

Types of Errors

- There may be three types of errors:
- **Single bit error**
 - In a frame, there is only one bit, anywhere though, which is corrupt.
- **Multiple bits error**
 - Frame is received with more than one bits in corrupted state.
- **Burst error**
 - Frame contains more than 1 consecutive bits corrupted.

Types of Errors

- Single bit error

1 0 1 1 0 0 1 1 => 1 0 1 1 0 1 1 1

- Multiple bits error

1 0 1 1 0 0 1 1 => 1 0 1 0 0 1 1 1

- Burst error

1 0 1 1 0 0 1 1 => 1 1 0 0 0 1 1 1

Dealing With Errors

- We need to build systems that are resilient to errors in data.
- There is no way to guarantee that all bits will be sent uncorrupted.
- One way to cope with this is to detect errors and request that corrupted data should be retransmitted.

Detecting Errors

- **Problem 1:** how can the receiver know when an error has occurred?

Solution : detect most errors.

- We could try sending the data twice and comparing the two transmissions to see where the errors are.(This is inefficient)
- **Problem 2:** Even when we detect an error, what to do about it?

Error Control Mechanisms

- may involve two possible ways:
 - Error detection
 - Error correction

Error Detecting Techniques

- The most popular Error Detecting Techniques are:
 - Single parity check
 - Two-dimensional parity check
 - Checksum
 - Cyclic redundancy check

Parity Checking (Vertical Redundancy Check)

- **count the bits in a character** to see if there is an even or odd number.
- Before transmission, an **extra bit (parity bit)** is appended to the character to force the number of bits to be even (or odd).
- If the received character does not have an even (or odd) number of bits then an error must have occurred.
- Both the sender and receiver must know which form of parity to use.

VRC

- A character such as 0110001 would be transmitted as:

Odd Parity: 01100010 (There are an odd number of 1s)

Even Parity: 01100011 (There are an even number of 1s)

- Parity checking will detect a single error in a character but not double errors.

7 bits of data (count of 1 bits)	8 bits including parity	
	Even	odd
0000000 (0)	00000000 (0)	10000000 (1)
1010001 (3)	11010001 (4)	01010001 (3)
1101001 (4)	01101001 (4)	11101001 (5)
1111111 (7)	11111111 (8)	01111111 (7)

LRC

(Longitudinal Redundancy Check)

- A block of bits is organized in rows and columns
- Also known as Two dimensional Parity
- The parity bit is calculated for each column and sent along with the data
- The block of parity acts as the redundant bits.

LRC :Example

- **Data Blocks:** 11100111 11011101 00111001 10101001
- Find the LRC and determine the data that is transmitted
- **Step 1:**the 1st data block in the 1st row ,2nd data block 2nd row

D1	1	1	1	0	0	1	1	1
D2	1	1	0	1	1	1	0	1
D3	0	0	1	1	1	0	0	1
D4	1	0	1	0	1	0	0	1

LRC :Example

- **Step 2:** Find LRC

- **Rule:** odd number of 1's -> 1 , Even number of 1's-> 0

D1	1	1	1	0	0	1	1	1
D2	1	1	0	1	1	1	0	1
D3	0	0	1	1	1	0	0	1
D4	1	0	1	0	1	0	0	1
LRC	1	0	1	0	1	0	1	0

- **Step 3:** message= LRC and Data Blocks

10101010 11100111 11011101 00111001 10101001

- In the receiver side, calculate LRC and compare it with senders LRC

Hamming Distance

- The Hamming distance between two bit patterns is **the number of dissimilar bits**.
- It measures *the minimum number of substitutions* required to change one string into the other, or *the number of errors* that transformed one string into the other.
- **Tip: XOR bits and count number of 1s**

Hamming Distance

- The Hamming distance between 01000001 ('A') and 01000010 ('B') is 2 because there are two dissimilar bits.
- It would take two errors in the wrong place to turn an 'A' into a 'B'.
- Adding a parity bit ensures that there is at least a Hamming distance of 2 between any two code words.

Hamming Distance

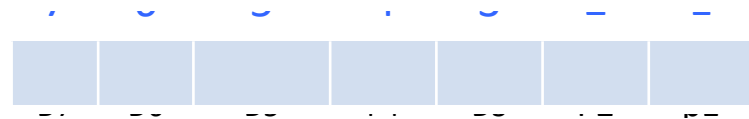
A=01000001
B=01000010

A XOR B= 00000011

- Hamming Distance is 2 because there are two dissimilar bits(count 1s).
- It would take two errors in the wrong place to turn an 'A' into a 'B'.
- Adding a parity bit ensures that there is at least a Hamming distance of 2 between any two code words.

Hamming Code

- 7 bit hamming code is used commonly
- 4 bit data and 3 bit parity
- **Message** =data bit + parity bit
- **Position :**
 - Parity bit position is 2^n where $n=\{0,1,2....n\}$
 - For 7 bit $p1=2^0, p2=2^1, p3=2^2$
- **Association**
 - P1 associated with D3,D5,D7
 - P2 associated with D3,D6,D7
 - P4 associated with D5,D6,D7



Hamming Code :Example

- Data bit=1101
- Position :
 - Parity bit position is 2^n where $n=\{0,1,2,\dots,n\}$
 - For 7 bit $p1=2^0, p2=2^1, p3=2^2$

- Let :Even Parity

- Parity bit

- $P1 \rightarrow D3, D5, D7 = (1, 0, 1)$ $p1=0$
- $P2 \rightarrow D3, D6, D7 = (1, 1, 1)$ $p2=1$
- $P4 \rightarrow D5, D6, D7 = (0, 1, 1)$ $p4=0$

7	6	5	4	3	2	1
1	1	0	0	1	1	0
D7	D6	D5	P4	D3	P2	p1

Checksum

- **The Sender follows the given steps:**
 - The block unit is divided into k sections, and each of n bits.
 - All the k sections are added together by using one's complement to get the sum.
 - The sum is complemented and it becomes the checksum field.
 - The original data and checksum field are sent across the network.

Checksum

- **The Receiver follows the given steps:**
 - The block unit is divided into k sections and each of n bits.
 - All the k sections are added together by using one's complement algorithm to get the sum.
 - The sum is complemented.
 - If the result of the sum is zero, then the data is accepted otherwise the data is discarded.

Checksum: Example

Sender

Data: 1001100111100010 0010010010000100

- Step 1: Data unit :

10011001 11100010 00100100 10000100

- Step 2: Sum

10000100

00100100

11100010

10011001

1000100011

Checksum: Example

Sender

- Step 3: sum the carry

```
10000100
00100100
11100010
10011001
-----
00100011
      10
-----
00100101
```

- Step 4 : 1's Complement

```
00100101
-----
11011010
```

- Step 5 : append the checksum and send

```
11011010
10011001
11100010
00100100
10000100
```

Checksum: Example

Receiver

- Step 1: collect all data blocks

11011010
10011001
11100010
00100100
10000100

- Step 2: sum all the data blocks and checksum

11011010
10011001
11100010
00100100
10000100

101111101

sum the carry
=11111101
10

- Step 3 : accept if the result is all 1's else reject

11111111

Cyclic Redundancy Code (CRC)

- A far more effective way of detecting errors in a block of data is to use a *Cyclic Redundancy Code*.
- In CRC, a number is mathematically calculated for a packet by its source computer, and then recalculated by the destination computer.
- If the original and recalculated versions at the destination computer differ, the packet is corrupt and needs to be resent or ignored.

CRC

Sender

- Find the length of the divisor 'L'
- Append 'L-1' bits to the original message
- Perform binary division operation
- Remainder of the division=CRC

Note:

- The CRC must be of L-1 bits

A	B	A XOR B
1	1	0
1	0	1
0	1	1
0	0	0

CRC

original message
1 0 1 0 0 0 0

@ means X-OR

Sender

```

1001 | 10100000000
@ 1001
-----
00110000000
@ 1001
-----
01010000
@ 1001
-----
0011000
@ 1001
-----
01010
@ 1001
-----
0011
  
```

Message to be transmitted

```

10100000000
+ 011
-----
1010000011
  
```

Generator polynomial
 x^3+1
 $1 \cdot x^3 + 0 \cdot x^2 + 0 \cdot x^1 + 1 \cdot x^0$
CRC generator
1001 4-bit

If CRC generator is of n bit then append $(n-1)$ zeros in the end of original message

```

1001 | 1010000011
@ 1001
-----
0011000011
@ 1001
-----
01010011
@ 1001
-----
0011011
@ 1001
-----
01001
@ 1001
-----
0000
  
```

Receiver

Zero means data is accepted

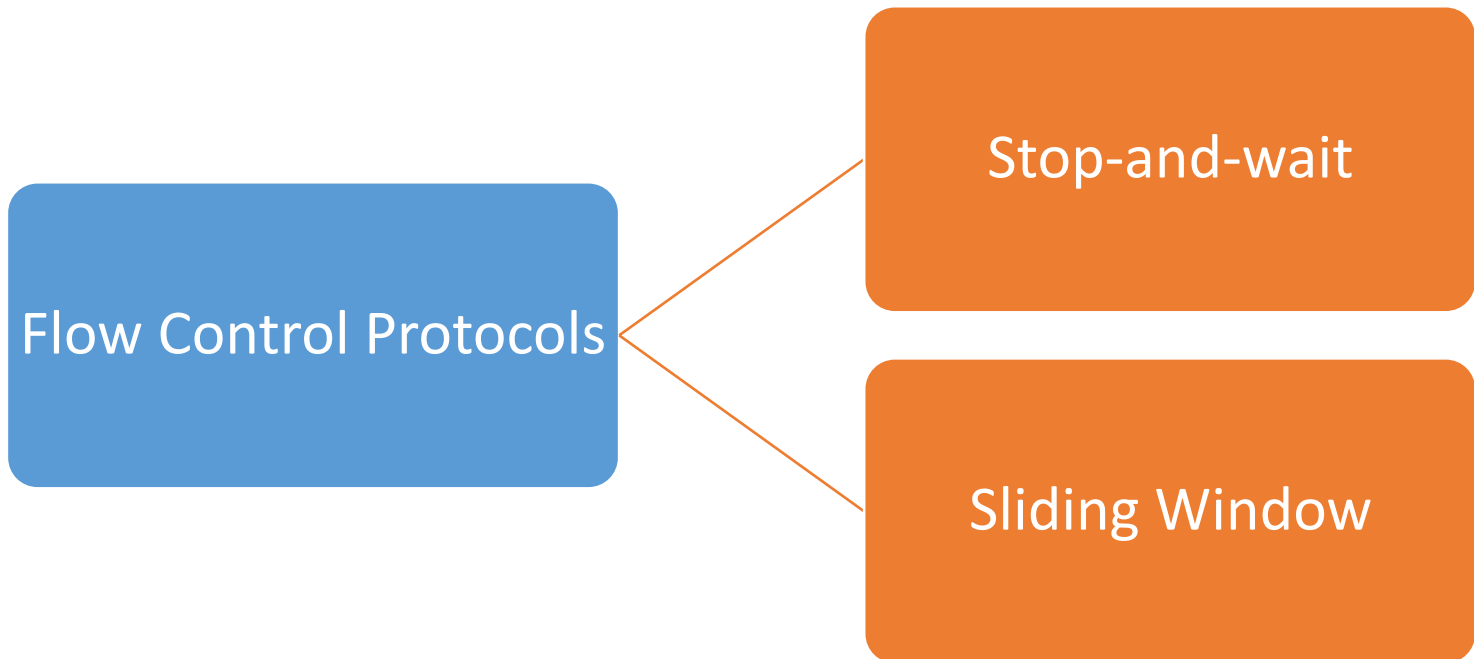
Flow Control

- tells the sender how much data it can transmit before it must wait for an acknowledgment from 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.
- request that the transmitting device send fewer frames or stop temporarily.

Contd.

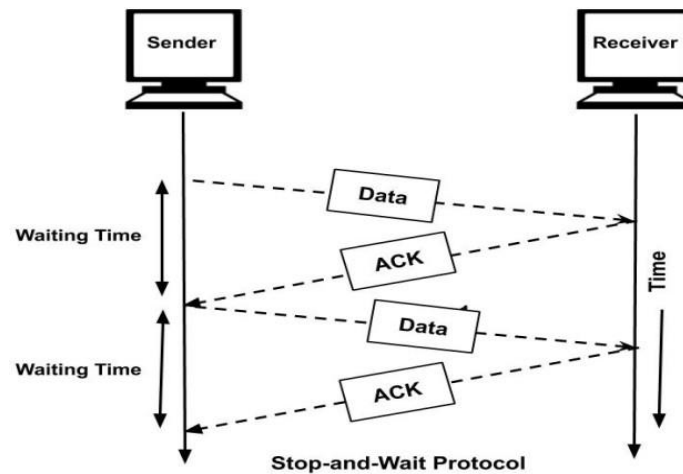
- each receiving device has a block of memory, called a *buffer*, reserved for storing incoming data until they are processed.
- If the buffer begins to fill up, the receiver must be able to tell the sender to *halt transmission* until it is once again able to receive.

Flow Control Protocols



Stop-and-Wait

- This flow control mechanism forces the sender after transmitting a data frame to stop and wait until the acknowledgement of the data-frame sent is received.



Stop-and-Wait

- **Advantage:**

- its simplicity

- **Disadvantage:**

- One frame at a time
- Poor utilization of bandwidth
- Poor performance

Stop-and-Wait : Problems

1. Problems due to lost data

- Sender waits ack for infinite amount of time
- Receiver waits data for infinite amount of time

2. Problems due to lost ack

- Sender waits ack for infinite amount of time

3. Problems due to delayed ACK/data

- After timeout on sender side, a delayed ACK might be wrongly considered as ack of some other data packet.

Sliding Window

- In this flow control mechanism, both sender and receiver agree on the number of data-frames after which the acknowledgement should be sent.
- As we learnt, stop and wait flow control mechanism wastes resources, this protocol tries to make use of underlying resources as much as possible.

Sliding Window

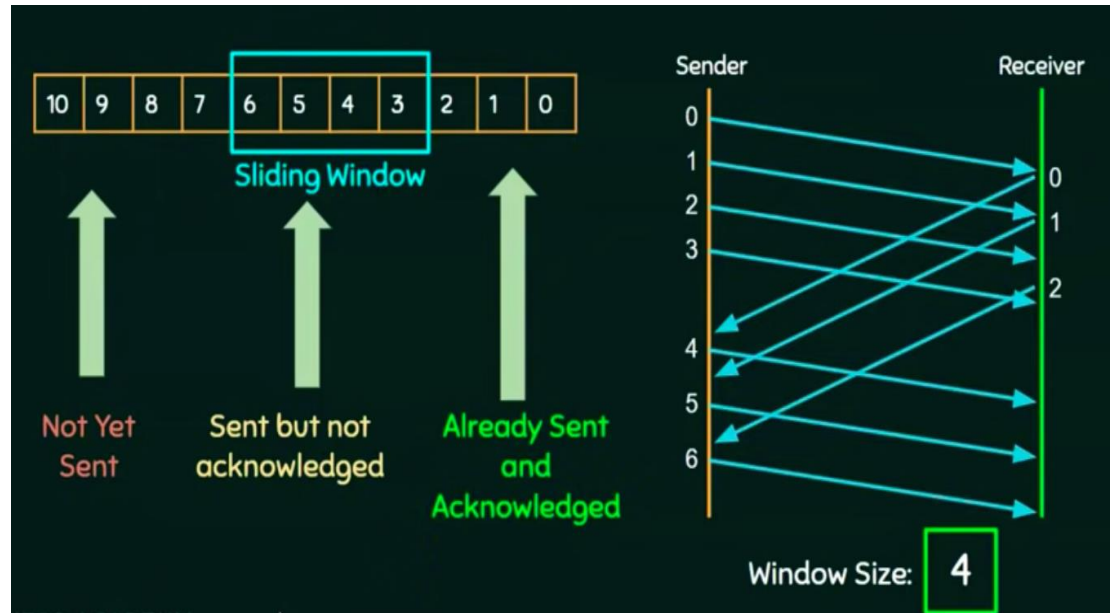
Send multiple frames at time

Number of frames to be sent is based on

window size

Each frame is numbered

-> **Sequence Number**



Error Control

■ Two methods of error correction are-

1. Reverse error correction (REC)

- Once the error is discovered, the receiver requests the sender to retransmit the entire data unit.

2. Forward error correction(FEC)

- the code set is so designed that it is possible for the receiver to detect and correct error as well by itself.

Requirements for Error Control Mechanism

- **Error detection**

- The sender and receiver, either both or any, must ascertain that there is some error in the transit.

- **Positive ACK**

- When the receiver receives a correct frame, it should acknowledge it.

- **Negative ACK**

- When the receiver receives a damaged frame or a duplicate frame, it sends a NACK back to the sender and the sender must retransmit the correct frame.

- **Retransmission:**

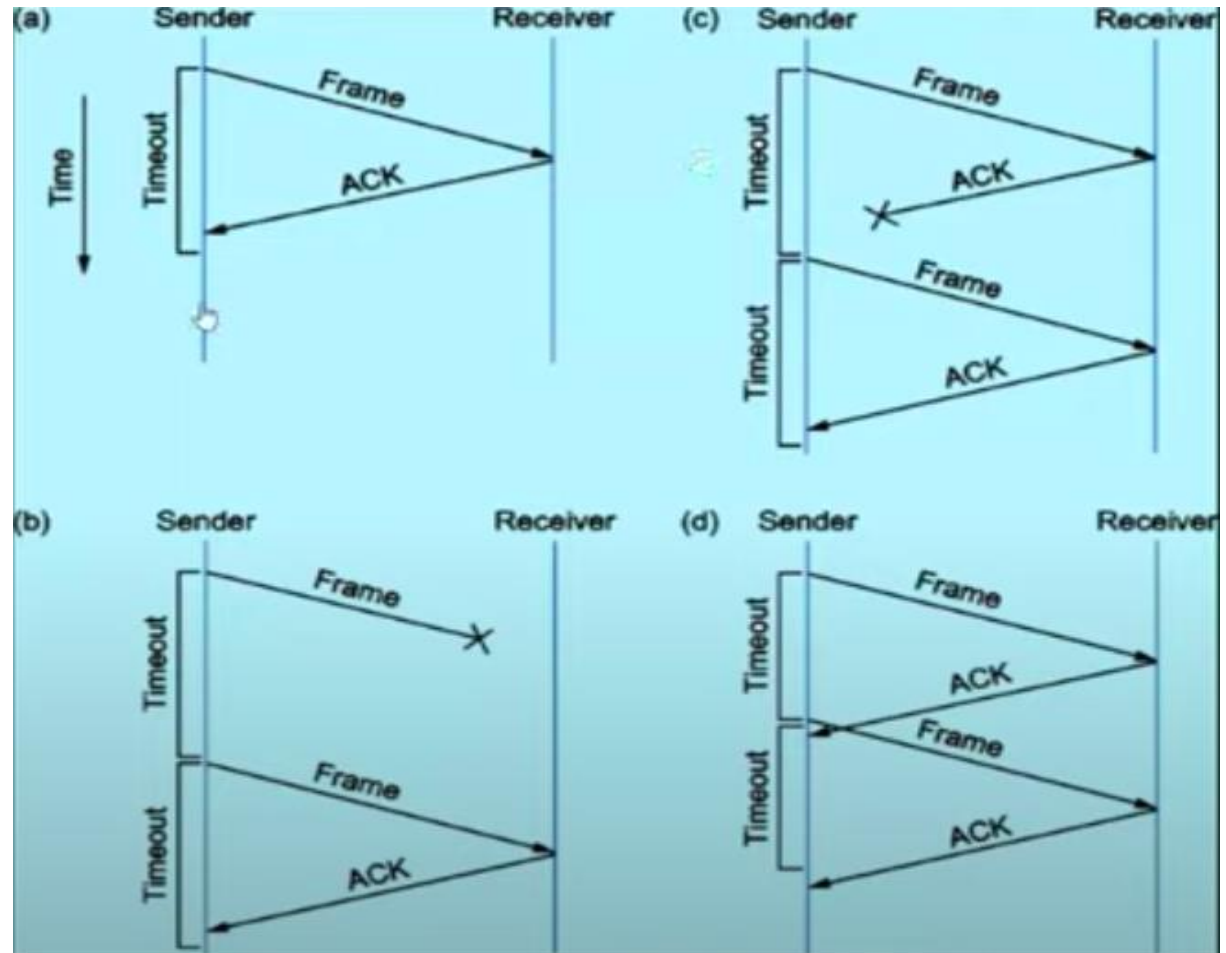
- The sender maintains a clock and sets a timeout period.
- If an acknowledgement of a data-frame previously transmitted does not arrive before the timeout the sender retransmits the frame, thinking that the frame or it's acknowledgement is lost in transit.

Error Control

- There are three types of techniques available which Data-link layer may deploy to control the errors by Automatic Repeat Requests (ARQ)
 - Stop-and-wait ARQ
 - Go-Back-N ARQ
 - Selective Repeat ARQ

Stop-and-wait ARQ

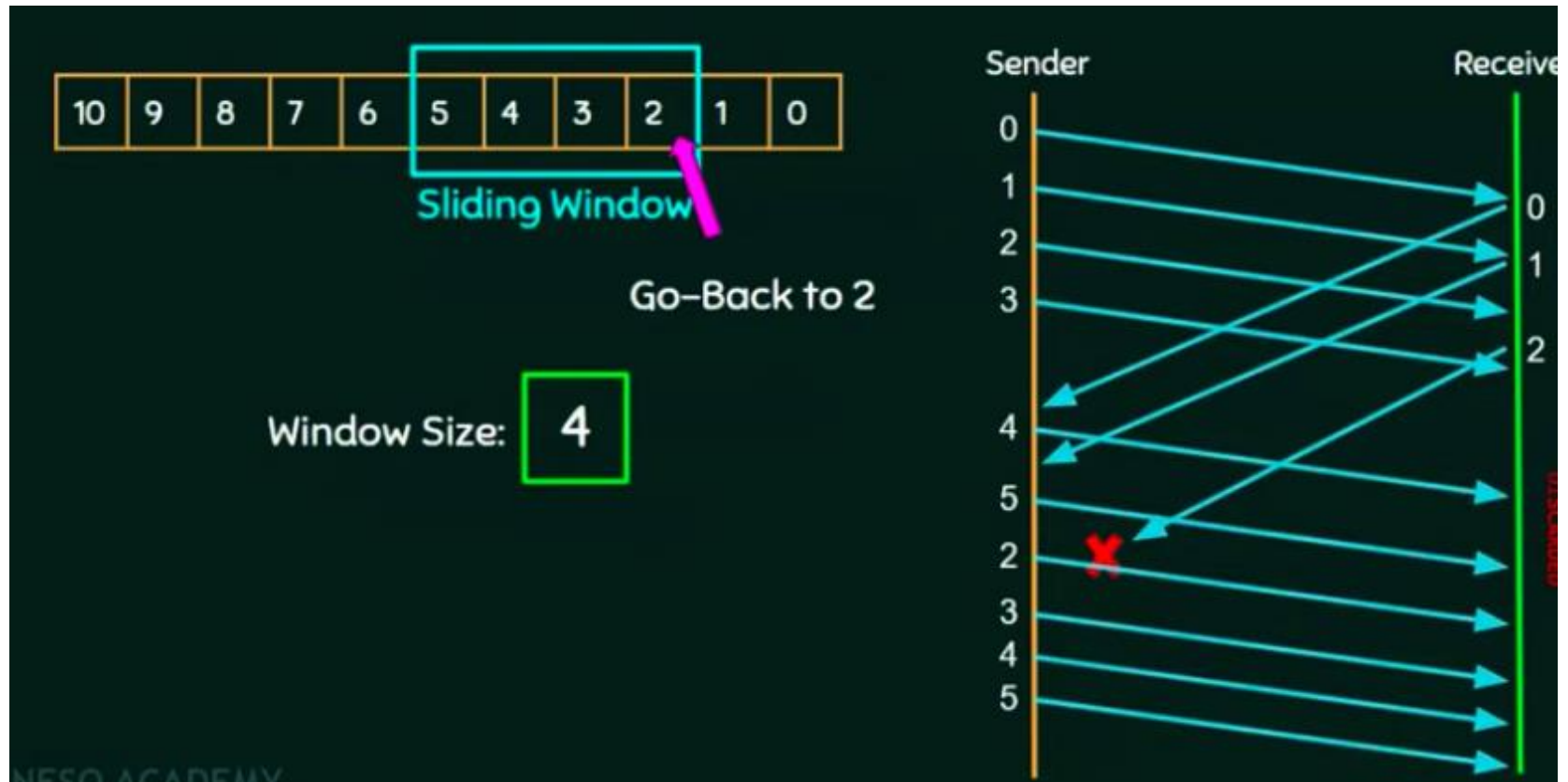
- If the ack does not arrive after a certain period of time ,the sender times out and retransmits the original frame
- **Stop-and-wait ARQ**=Stop-and-wait + Timeout timer + Sequence number



Go-Back-N ARQ

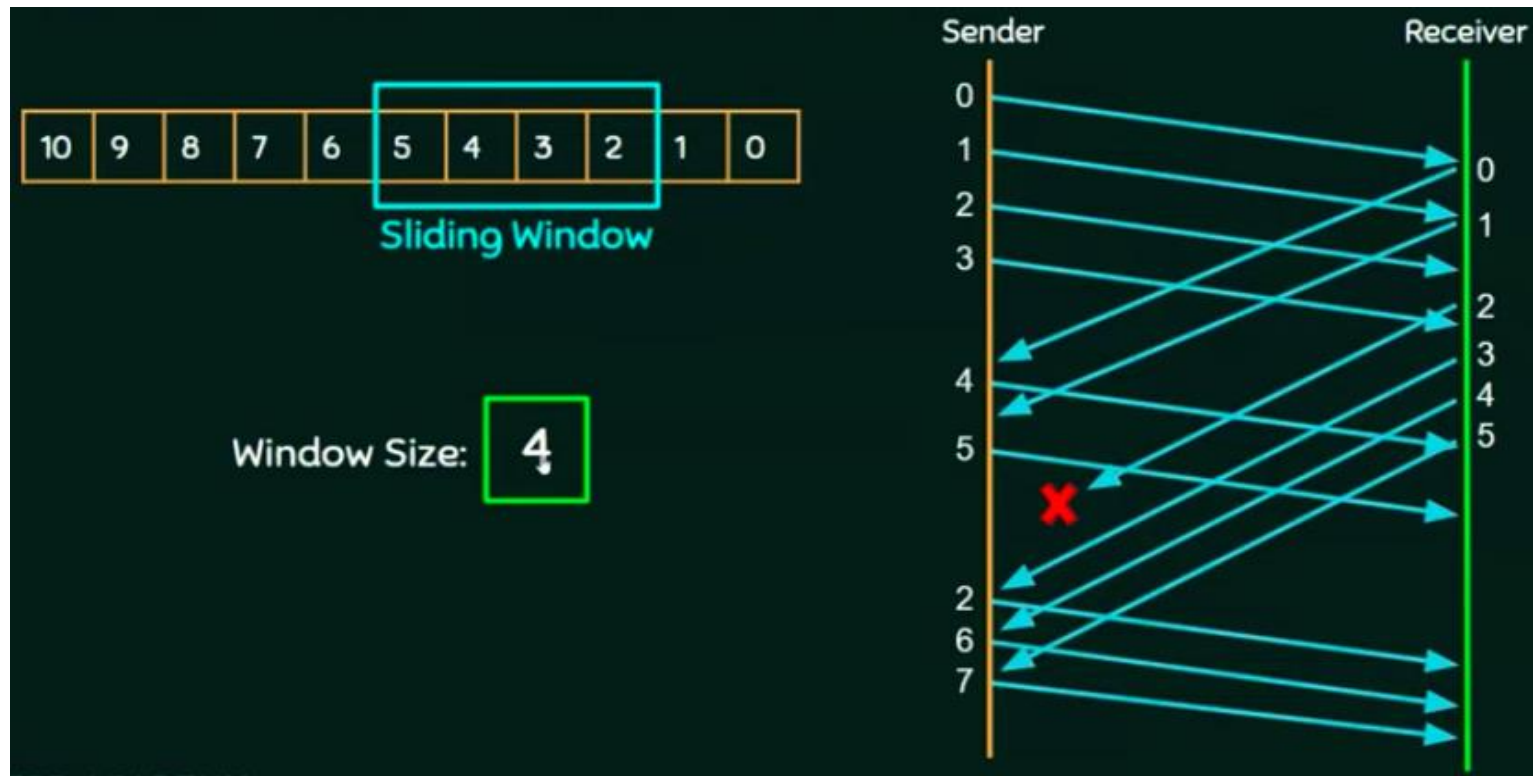
- N- the sender window size
- Uses the concept of protocol pipelining i.e. the sender can send multiple frames before receiving the ack for the first frame.
- Frames are numbered in sequential manner
- If the ack of a frame is not received within an agreed upon time period, **all frames in the current window are retransmitted**
- Sequence number is based on the window size
 - If $N=4$ (i.e. 2^2) the seq numbers will be **0,1,2,3,0,1,2,3,0,1,2,3** and so on /00,01,10,11....

Go-Back-N ARQ: Example



Selective Repeat ARQ

- Only the erroneous or lost frames are retransmitted, while correct frames are received and buffer.



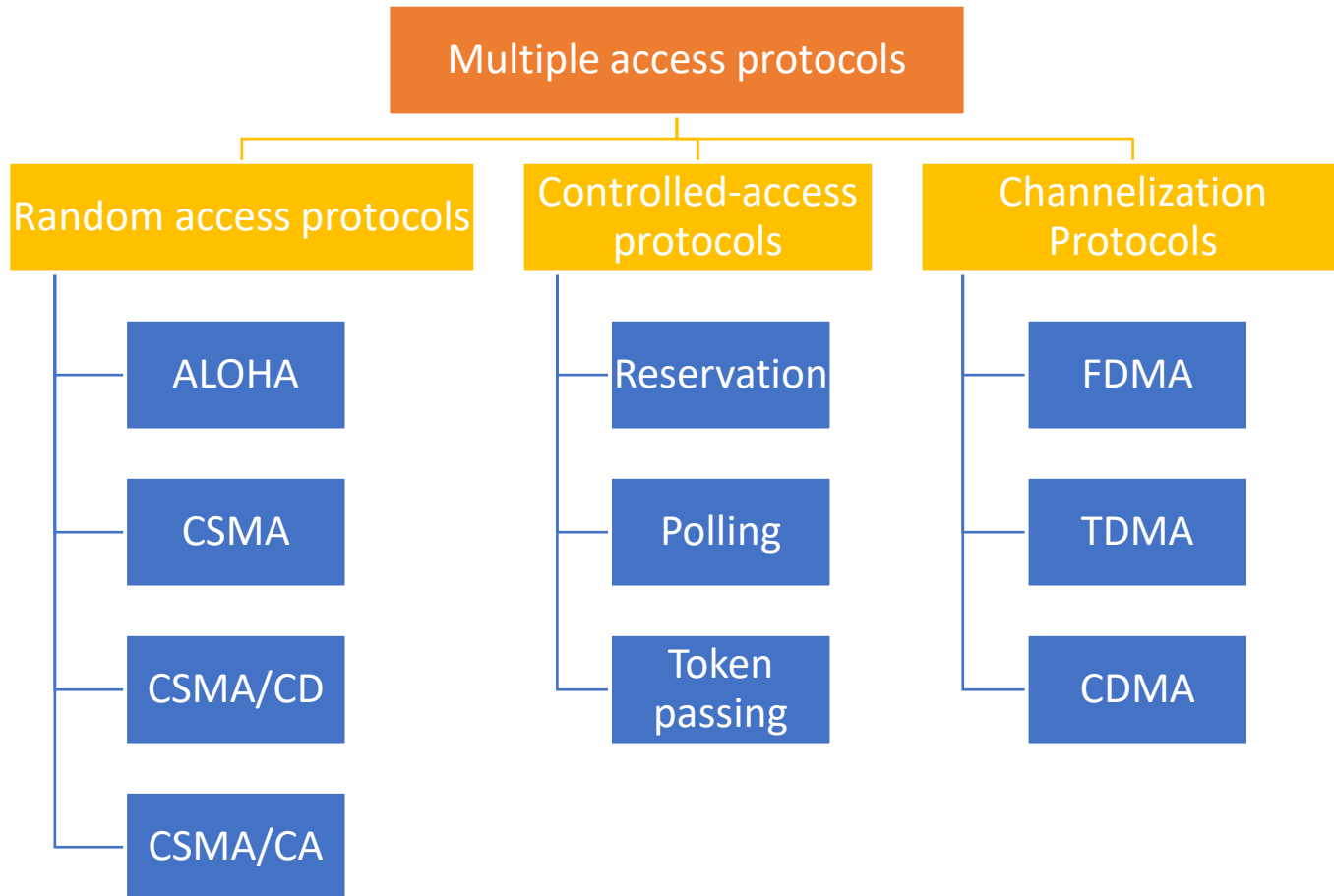
Media Access control

- The data link layer can further be divided in to two layers:
 - the **upper sub-layer** that is responsible for *flow and error control* is called the **logical link control (LLC) layer**
 - the **lower sub-layer** that is mostly responsible for *multiple access resolution* is called the **media access control (MAC) layer**

Media Access control

- The set of rules that defines how the computer puts data onto the network cable and takes data from the cable is called an **access method**.
- Once data is moving on the network, access methods help to regulate the flow of network traffic.

Multiple Access Protocols



Random Access

- no station is superior to another station and none is assigned the control over another.
- At each instance, a station that has data to send uses a procedure defined by the protocol to make a decision on whether or not to send.
- decision depends on the state of the medium (idle or busy).

Contd.

- if more than one station tries to send, there is an access conflict-collision-and the frames will be either destroyed or modified.
- To avoid access conflict, 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?

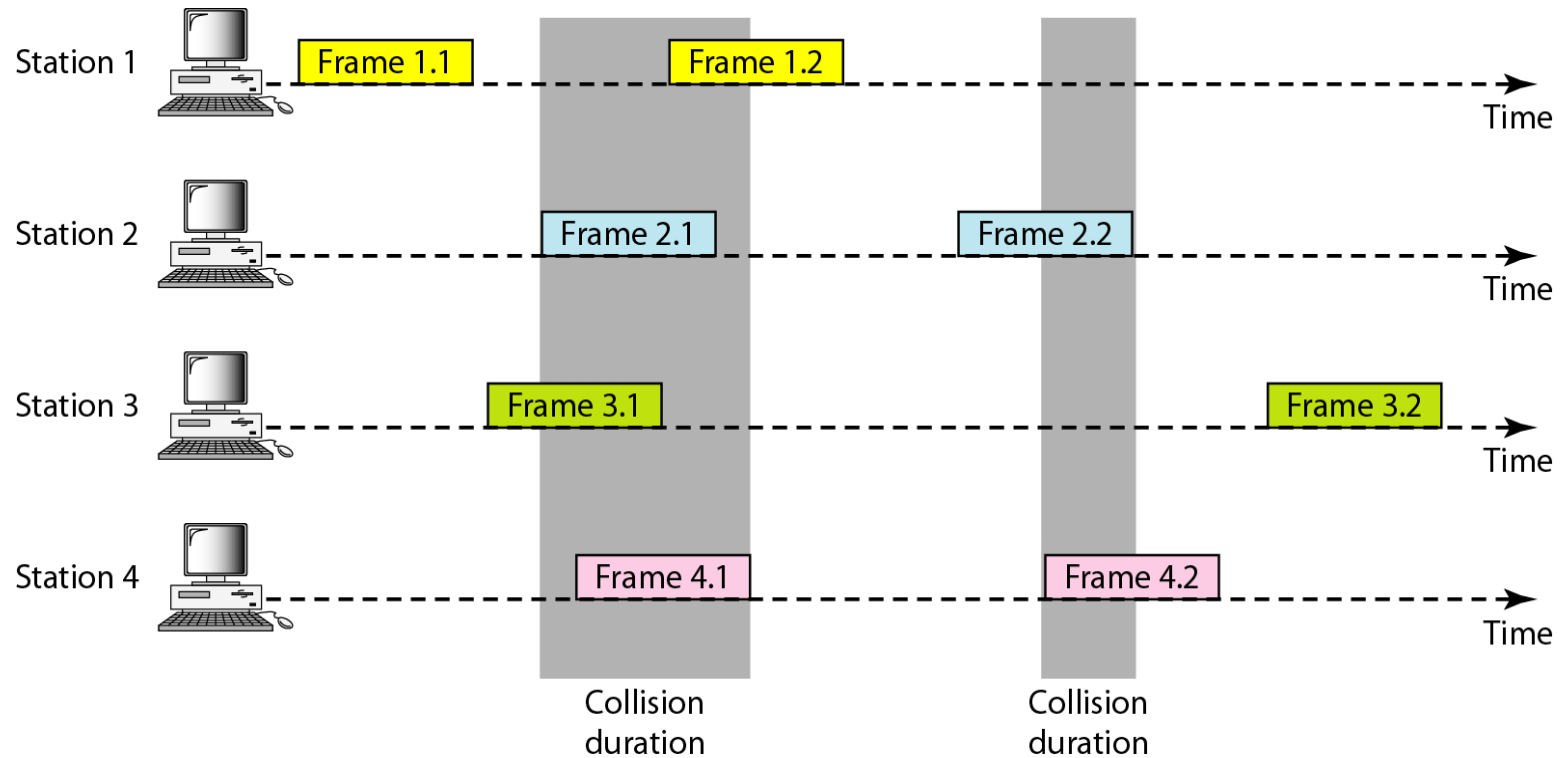
Contd.

- The random access methods have evolved from a very interesting protocol known as **ALOHA**, which used a very simple procedure called **multiple access (MA)**.
- The method was improved with the addition of a procedure that forces the station to sense the medium before transmitting.
 - This was called **carrier sense multiple access**.
 - This method later evolved into two parallel methods:
 - *CSMA/CD* tells the station what to do when a collision is detected.
 - *CSMA/CA* tries to avoid the collision.

Pure ALOHA

- ALOHA is the simplest technique in multiple accesses.
- Basic idea of this mechanism is a user can transmit the data whenever they want.
- When a station sends data it waits for an ack. If the ack doesn't come within the allotted time then the station waits for a random amount of time called back-off-time(T_b) and resends the data
- Since different stations wait for different amount of time ,the probability of further collision decreases
- The throughput of pure aloha is maximized when frames are of uniform length

Procedure for pure ALOHA protocol




Reading Assignment: Slotted ALOHA

Carrier Sense Multiple Access (CSMA)

- Continually listens to the cable for the presence of a signal prior to transmitting./sense before transmit /
- There are two variants of CSMA.
 - CSMA/CD and CSMA/CA
- The possibility of collision still exist because of propagation delay/the station may sense the medium and find it idle, only because the first bit sent by another station has not yet been received.
- Schema : If the station waits for the medium to become idle it is called **persistent** otherwise it is called **non persistent**.

Persistent CSMA

- If it senses the channel **idle**, station starts transmitting the data.
- If it senses the channel **busy** it waits until the channel is idle, by **continuously sensing** the channel.

 *wait if busy and transmit only when the media becomes idle again (not transmission after a triggered timer expire)*

Non-Persistent CSMA

- less aggressive compared to persistent protocol.
- before sending the data, the station senses the channel and if the channel is idle it starts transmitting the data.
- if the channel is busy, the station does not continuously sense it but instead of that it waits for random amount of time and repeats the algorithm.
- better channel utilization but also results in longer delay compared to persistent.

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

- If two stations sense the channel to be idle and begin transmitting **simultaneously**, this causes a **collision**.
- the two computers involved **stop transmitting for a random period of time**
- After a random time interval, the stations that collided attempt to **transmit again**.
- If another collision occurs, the **time intervals from** which the random waiting time is selected are **increased step by step**.
This is known as **exponential back off**.

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

- In CSMA/CA, the computer actually broadcasts a warning packet before it begins transmitting the real data on the wire.
- Each computer on the network does not attempt to broadcast when another computer sends the warning packet.
- All other computers wait until the data is sent.
- The major drawback of trying to avoid network collisions is that the network traffic is high due to the broadcasting of the intent to send a message.
- Used in wire less networks ,where CSMA/CD is not possible due to wireless transmitters desensing their receivers during packet transmission.

Controlled access

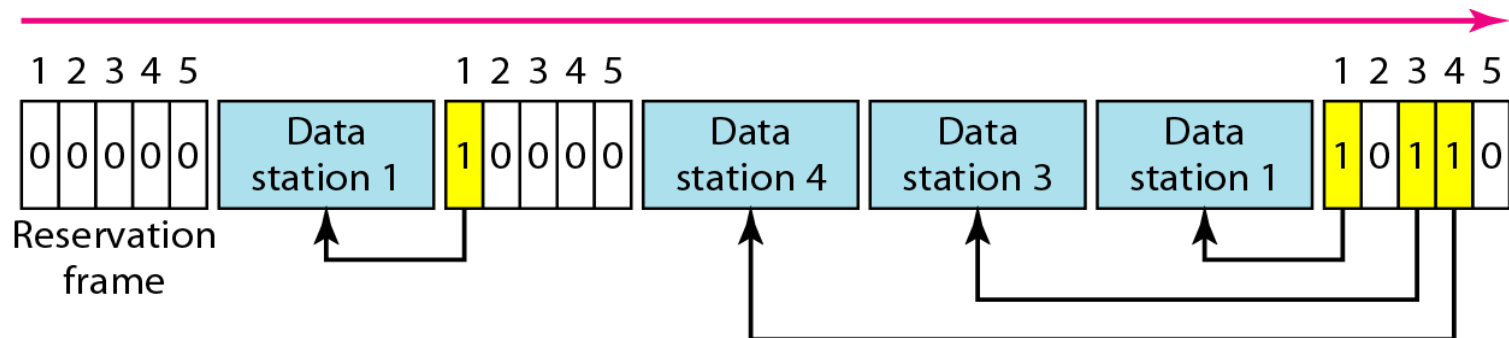
- In controlled access, the stations **consult one another** to find which station has the right to send.
- A station cannot send unless it has been authorized by other stations.

Reservation

- a station needs to make a **reservation** before sending data.
- 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.
- Each mini slot belongs to a station. 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.

Contd.

- The following figure shows a situation with **five stations** and a **five-minislot** reservation frame.
- In the first interval, **only stations 1, 3, and 4** have made reservations. In the second interval, **only station 1** has made a reservation.

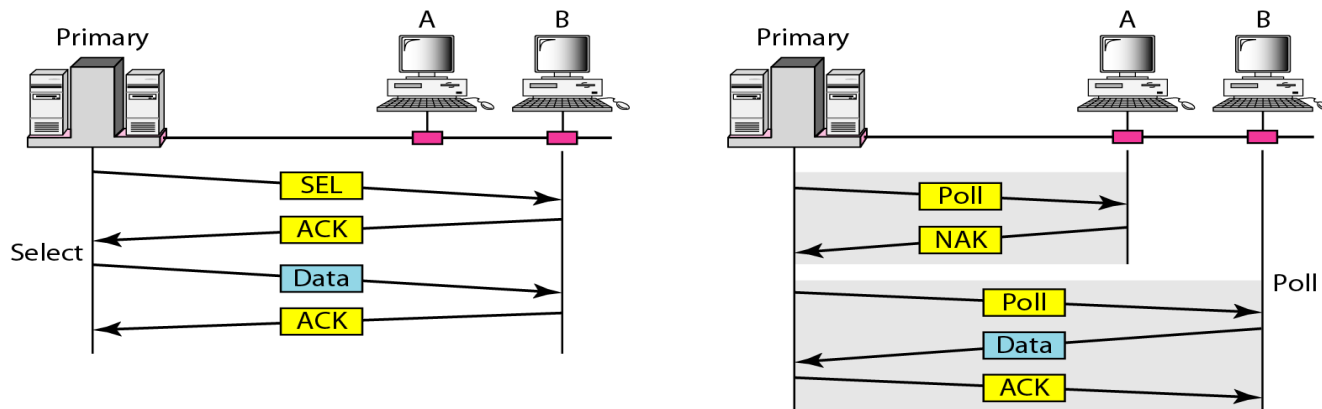


Polling (Demand-Priority)

- Polling works with topologies in which one device is designated as a **primary station** and the other devices are **secondary stations**.
- The **primary** device(**master node**) **controls the link**; the **secondary devices** follow its **instructions**.
- It is up to the primary device to determine which device is allowed to use the channel at a given time.
- The primary device, therefore, is always the **initiator of a session**.

Contd.

- If the primary wants to **receive data**, it asks the secondary devices if they have anything to send; this is called **poll function**.
- If the primary wants to send data, it tells the secondary to get **ready to receive**; this is called **select function**.



Token Passing

- the stations in a network are organized in a **logical ring**. i.e. for each station, there is a **predecessor** and a **successor**.
- 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.

Contd.

But how is the right to access the channel passed from one station to another?

- 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.
- When a station has some data to send, it waits until it receives the token from its predecessor. It then holds the token and sends its data.
- When the station has no more data to send, it releases the token, passing it to the next logical station in the ring.
- The station cannot send data until it receives the token again in the next round.

Access methods summary

- The following table summarizes the major features of each access method:

Feature/function	CSMA/CD	CSMA/CA	Token passing	Demand priority
Type of communication	Broadcast based	Broadcast based	Token based	Hub based
Type of access method	Contention (collisions may occur)	Contention	Non-contention	Contention
Type of Network	Ethernet	LocalTalk	Token Ring ARCnet	100VG-AnyLAN

Thank You