Unit 2

COMPUTER NETWORKS

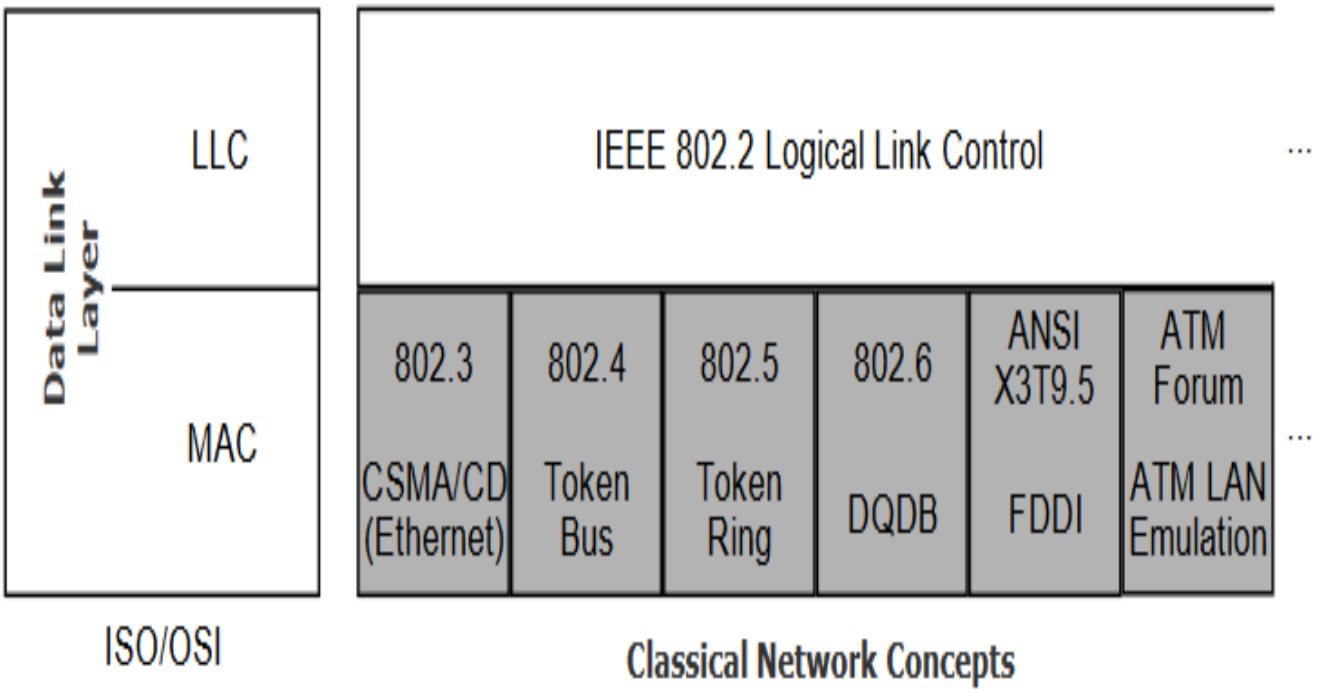
RCS-601

Unit-2

- **Medium Access sub layer:** Medium Access sub layer - Channel Allocations, LAN protocols - ALOHA protocols - Overview of IEEE standards - FDDI. Data Link Layer - Elementary Data Link Protocols, Sliding Window protocols, Error Handling.

Network Types for the Local Range

- **LLC layer**: uniform interface and same frame format to upper layers
- **MAC layer**: defines medium access



Both concepts are implemented together in existing networks (as a device driver):

1. **Packing of data into frames:** error detection during frame transmission and receipt
2. **Media Access Control:** this contains the frame transmission and the reaction to transmission errors

Standardization: IEEE

- **Institute of Electrical and Electronic Engineers (IEEE)**
  - Standardization of the IEEE 802.X-Standards for Local Area Networks (www.ieee802.org) – many historical!

www.ieee.org

- 802.1 Overview and Architecture of LANs
- 802.2 Logical Link Control (LLC)
- 802.3 CSMA/CD (Ethernet)
- 802.4 Token Bus
- 802.5 Token Ring
- 802.6 DQDB (Distributed Queue Dual Bus)
- 802.7 Broadband Technical Advisory Group (BBTAG)
- 802.8 Fiber Optic Technical Advisory Group (FOTAG)
- 802.9 Integrated Services LAN (ISLAN) Interface
- 802.10 Standard for Interoperable LAN Security (SILS)
- 802.11 Wireless LAN (WLAN)
- 802.12 Demand Priority (HP's AnyLAN)
- 802.14 Cable modems
- 802.15 Personal Area Networks (PAN, Bluetooth)
- 802.16 Wireless MAN
- 802.17 Resilient Packet Ring
- 802.18 Radio Regulatory Technical Advisory Group (RRTAG)
- 802.19 Coexistence Technical Advisory Group
- 802.20 Mobile Broadband Wireless Access (MBWA)
- 802.21 Media Independent Handover

Medium Access sub layer (Introduction)

A network of computers based on *multi-access medium* requires a protocol for effective *sharing* of the media. As only one node can send or transmit signal at a time using the broadcast mode, the main problem here is how different nodes get control of the medium to send data, that is *"who goes next?". The protocols used for this purpose are known as Medium Access Control (MAC) techniques. The key issues involved here are - Where and How the control is exercised.*

A centralized scheme has a number of advantages as mentioned below:

- Greater control to provide features like priority, overrides, and guaranteed bandwidth.
- Simpler logic at each node.
- Easy co-ordination.

Goals of MACs

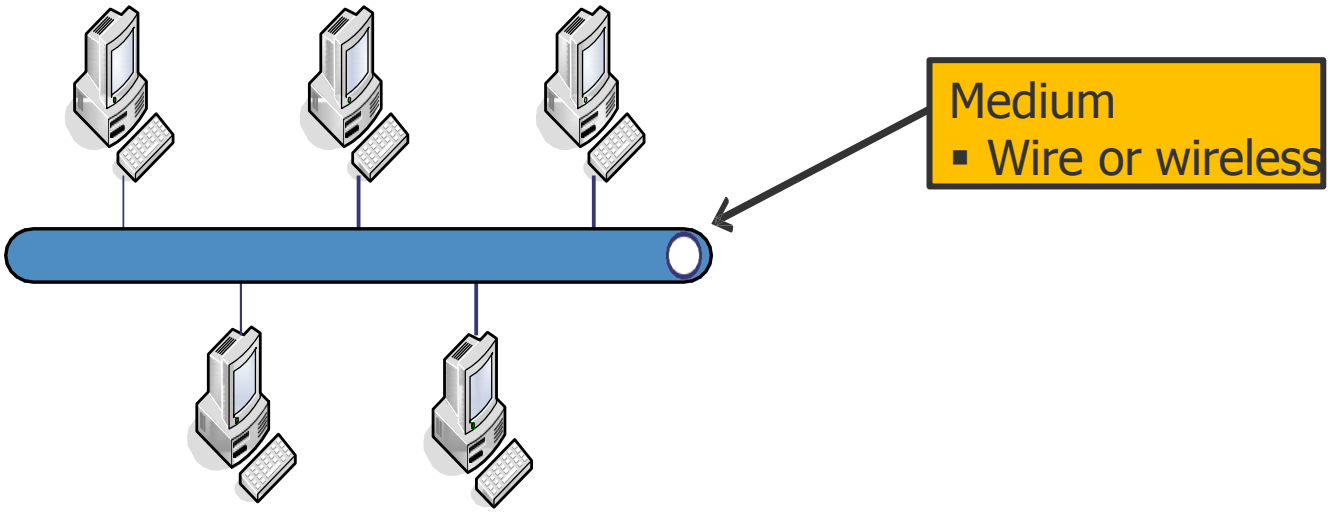
Medium Access Control techniques are designed with the following goals in mind

1. Initialisation
2. Fairness
3. Priority
4. Limitations to one station
5. Receipt
6. Error Limitation
7. Recovery
8. Reconfigurability
9. Compatibility
10. Reliability

The Channel Allocation Problem

The Channel Allocation Problem

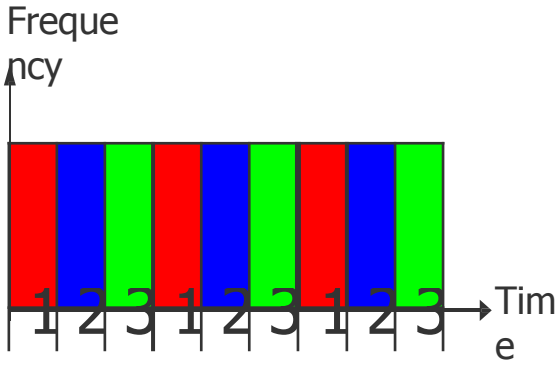
- The channel allocation problem
  - Given N independent stations which want to communicate over a single channel
  - Organize the sending order of the stations

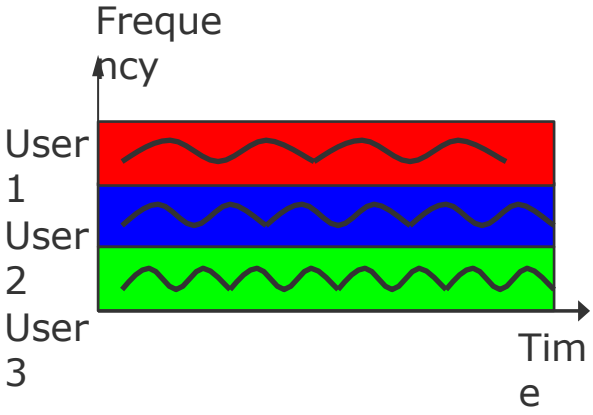


- Approaches
  - Static channel allocation
    - Simple procedures
  - Dynamic channel allocation
    - Complex procedure, that adapt to changes

Static Channel Allocation

- Time Division Multiple Access (**TDMA**)
  - Each user gets the **entire** transmission capacity for a fixed time interval
  - Baseband transmission
- Frequency Division Multiple Access (**FDMA**)
  - Each user gets a **portion** of the transmission capacity for the whole time
    - Frequency range
  - Broadband transmission





Static Channel Allocation

- Problems with static channel allocation
  - Works only for a fixed number of users
    - When number of users change, the allocation scheme does not work
  - Data traffic is very often bursty, i.e., long time no data and for a short time high data (ok for classical voice communication!)
  - Thus, users do not use their allocated channel capacity

    Most of the channels will be idle most of the time

- **Dynamic Channel Allocation**

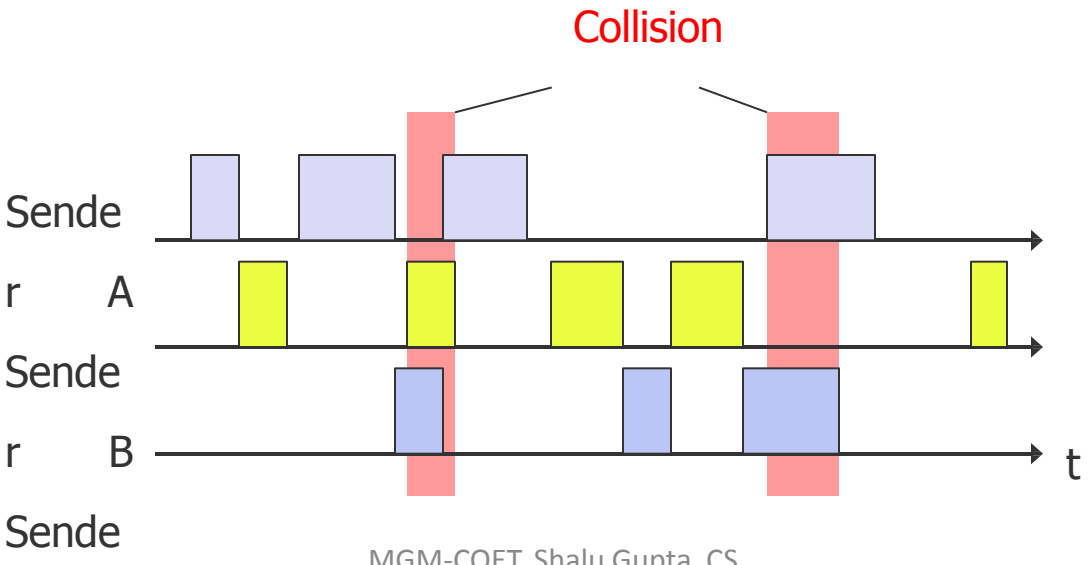
Dynamic Channel Allocation

- Assumptions on **dynamic** channel allocation
  - Station Model
    - There are N independent stations (computers) that generate frames for transmission.
  - Single channel
    - A single channel is available for communication and all stations can transmit and receive on it.
  - Collisions
    - If two frames are transmitted simultaneously, they overlap and the signals are garbled.
  - Time
    - Continuous time: No master clock, transmission of frames can begin at anytime.
    - Slotted time: Time is divided into discrete intervals called slots. Frame transmissions begin always at the start of a slot.
  - Sensing of the medium
    - Carrier sense: Stations can sense channel and tell whether it is busy. If so, stations do not start with transmissions.
    - No carrier sense: Stations can not sense the channel.

Multiple Access Protocols

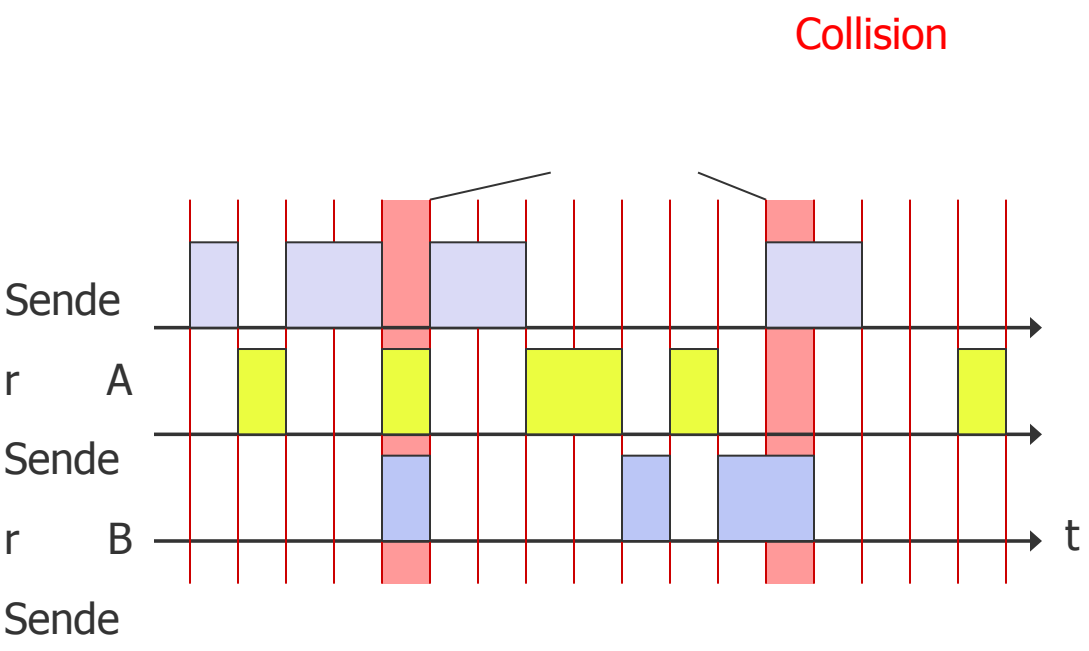
Multiple Access Protocols: ALOHA

- Best known protocol: ALOHA
  - Developed on the Hawaiian islands in 1970s: stations are connected by a satellite
  - Very simple principle, no coordination:
    - Stations are sending completely uncoordinated (random), all using the same frequency
    - When two (or more) stations are sending at the same time, a collision occurs: both messages are destroyed.
      - Collisions occur even with very **small overlaps**!
      - **Vulnerability period:** 2 times the length of a frame
    - When a collision occurs, frames are repeated after a random time
    - Problem: since traffic runs over a satellite a sender only hears after a very long time whether the transmission was successful or not.



Multiple Access Protocols: ALOHA

- Problem with ALOHA: even small overlaps result in transmission conflicts. Therefore, often collisions result in many repetitions:
  - No guaranteed response times
  - Low throughput
- Improvement: Slotted ALOHA
  - The time axis is divided into **time slots** (similar to TDMA, but time slots are not firmly assigned to stations)
  - The transmission of a block has to start at the beginning of a time slot
    - Fewer collisions, **vulnerability period of one frame length**
  - But: the stations must be synchronized!



A slotted ALOHA network transmits 200-bit frames using a shared channel with a 200 Kbps bandwidth. Find the throughput of the system (all stations put together) produces 250 frames per second:

A. 49
B. 368
C. 149
D. 151

Ans: 49 Frames

Explanation:

The frame transmission time=200/200 kbps=1ms

G=1/4

So $S=Ge^{-G} = 0.195 = 19.5\%$

This means that the throughput is 250 * 0.195=49 frames.

Only 49 out of 250 frames will survive.

Aktu 2018-19

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
Explain ARQ Error Control technique, in brief.

A slotted ALOHA network transmits 200-bit frames using a shared channel with a 200-kbps bandwidth. Find the throughput if the system (all stations together) produces

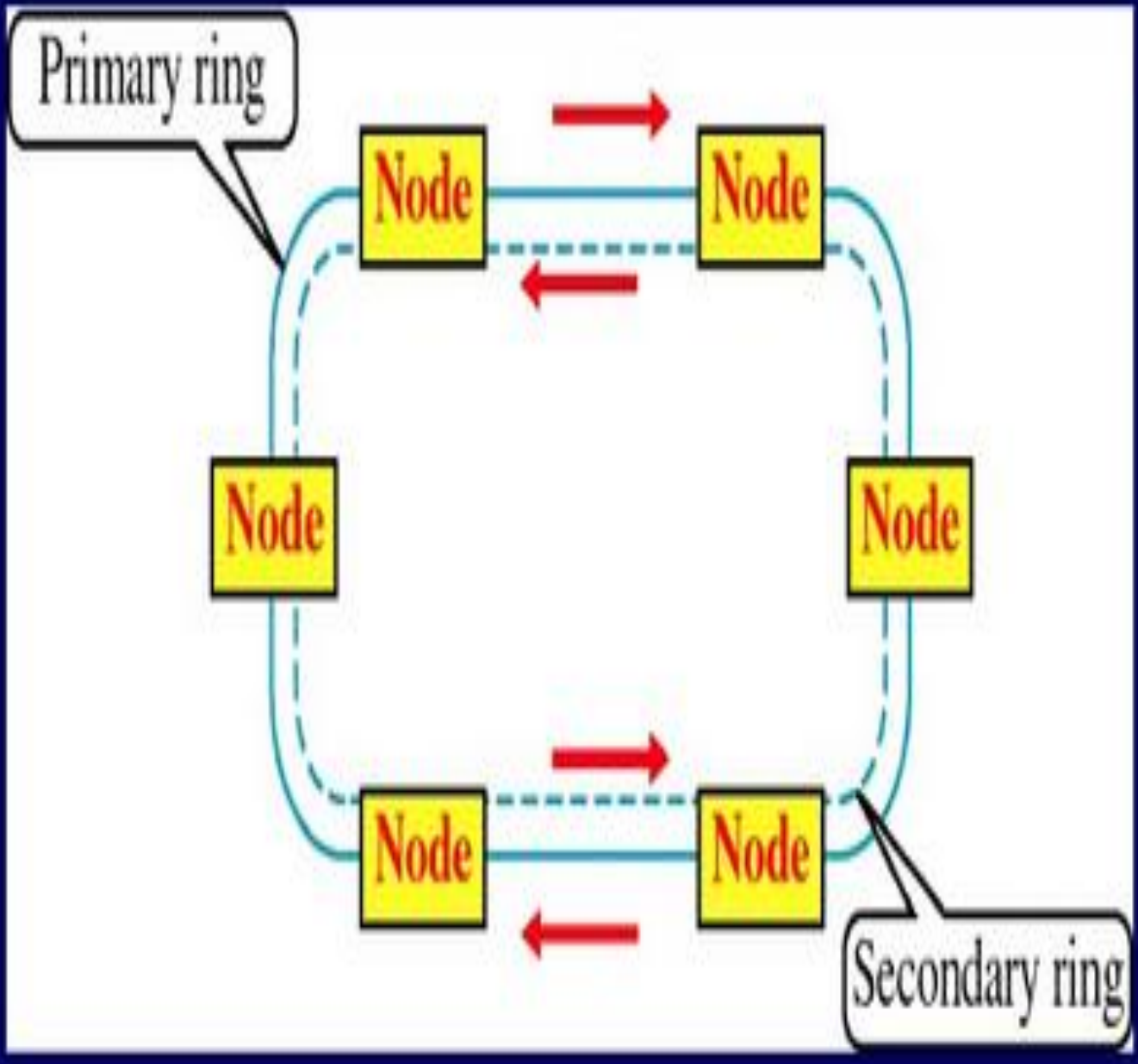
a. 1000 frames per second
b. 500 frames per second
c. 250 frames per second

Solution

This situation is similar to the previous exercise except that the network is using slotted ALOHA instead of pure ALOHA. The frame transmission time is *200/200* kbps or 1 ms.

a. In this case G is 1. So $S = G \times e^{-G}$ or $S = 0.368$ (36.8 percent). This means that the throughput is 1000 x 0.0368 = 368 frames. Only 368 out of 1000 frames will probably survive. Note that this is the maximum throughput case, percentagewise.
b. Here G is $\frac{1}{2}$. In this case $S = G \times e^{-G}$ or $S = 0.303$ (30.3 percent). This means that the throughput is 500 x 0.0303 = 151. Only 151 frames out of 500 will probably survive.
c. Now G is $\frac{1}{4}$. In this case $S = G \times e^{-G}$ or $S = 0.195$ (19.5 percent). This means that the throughput is 250 x 0.195 = 49. Only 49 frames out of 250 will probably survive.

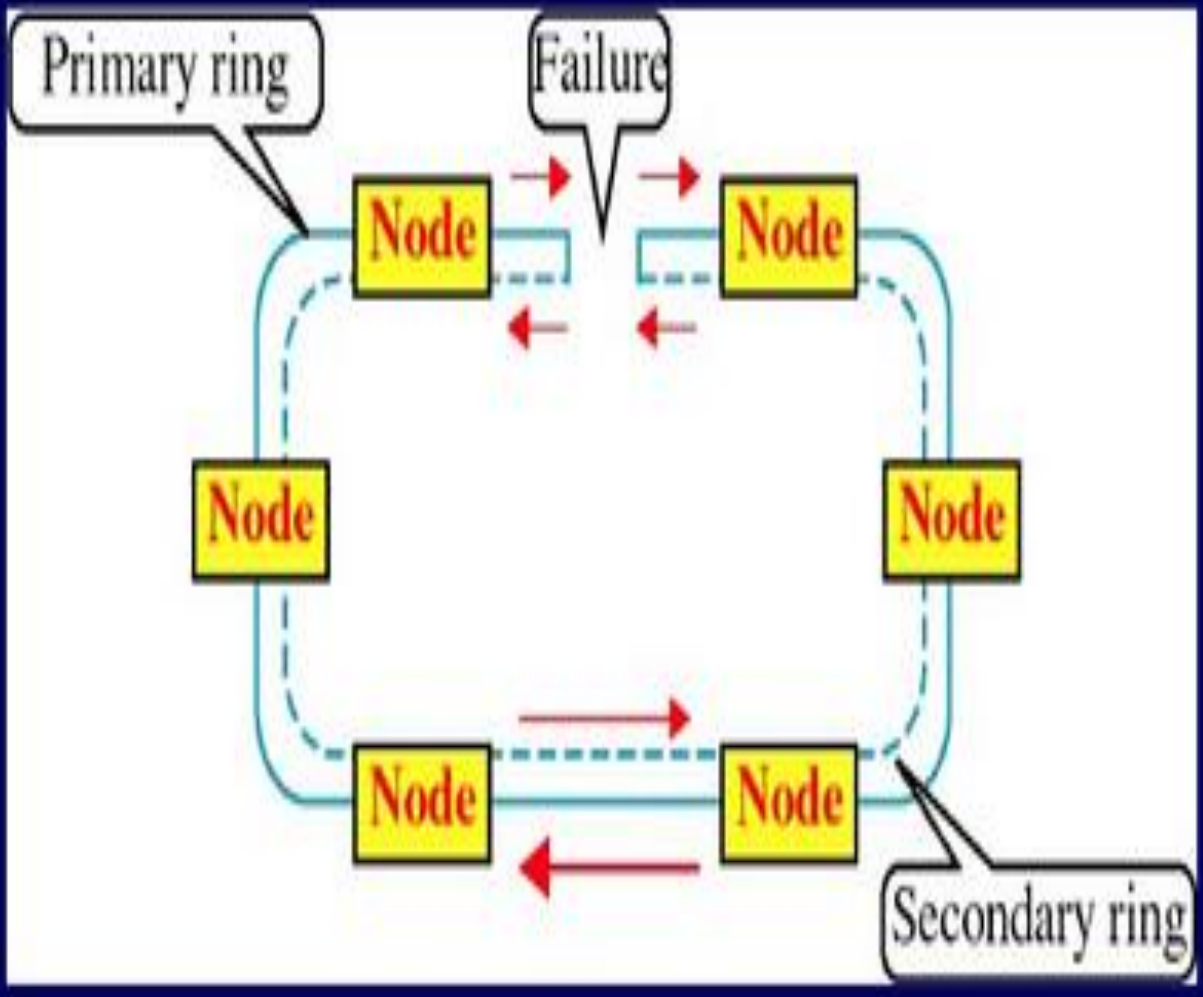
FDDI Architecture



Components of FDDI

- Fiber optic cable
- A concentrator (ring)
- Stations: 2 types
  - DAS (Dual Attachment Station) or Class A:
    - Connected to both the rings
  - SAS (Single Attachment Station) or Class B:
    - Connected to primary ring

Ring Wrapping

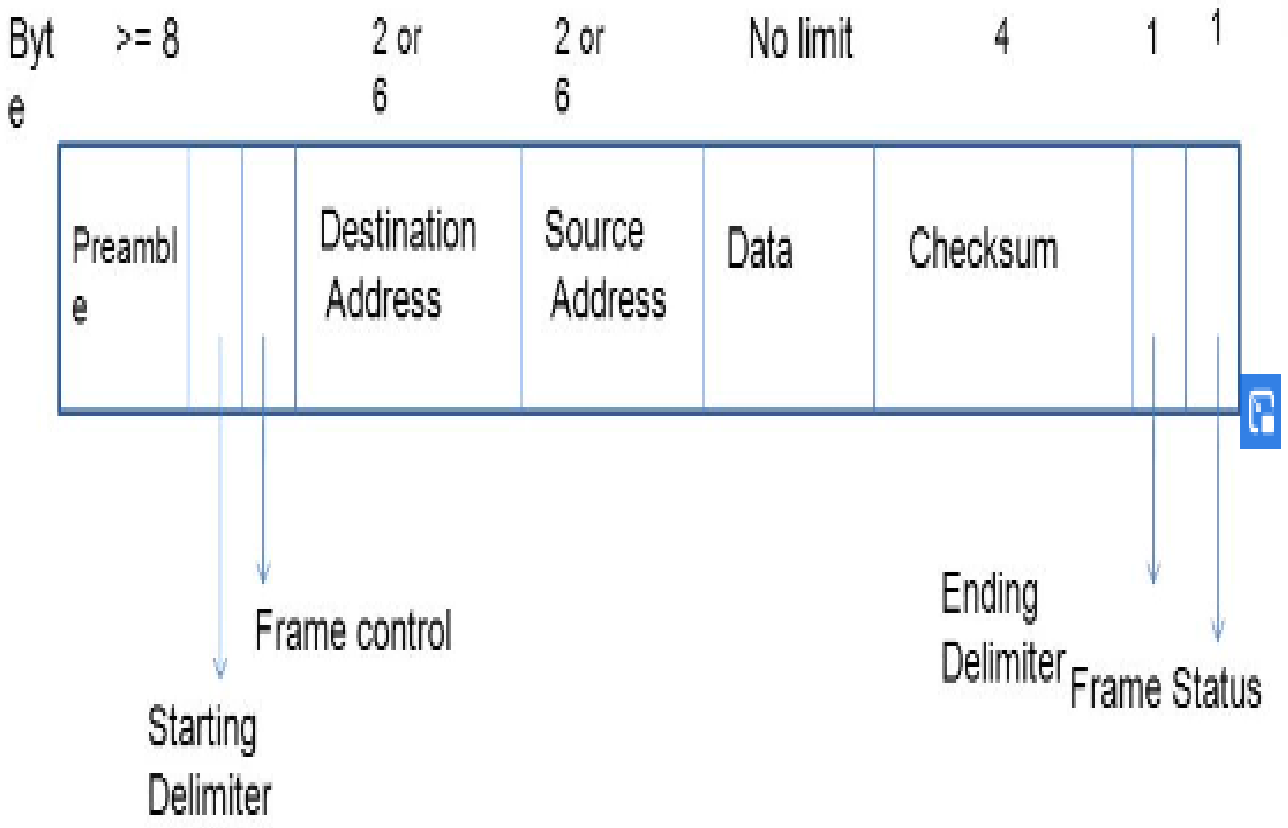


When a single station fails, devices on either side of the failed (or powered-down) station wrap, forming a single ring. Network operation continues for the remaining stations on the ring.

FDDI Access Method

- FDDI uses token passing as an access method similar to the IEEE 802.5 token ring.
- Any station wants to transmit information holds the token and then transmits the information and when it finish it releases the token in the ring.
- The time a station holds the token is called Synchronous Allocation Time (SAT) and this time is variable for each station. The allocation of this time to each station is achieved by Station Management (SMT).
- The functions of SMT are ring control, ring initialization, station insertion and station removal.

FDDI Frame Format



FDDI Frame Fields

- **Preamble**: Gives a unique sequence that prepares each station for an upcoming frame
- **Start delimiter:** Indicates the beginning of a frame
- **Frame control:** Indicates the size of the address fields and whether the frame contains asynchronous or synchronous data, among other control information
- **Destination address:** Contains a unicast (singular), multicast (group), or broadcast (every station) address
- **Source address:** Identifies the single station that sent the frame
- **Data:** Contains either information destined for an upper-layer protocol or control information.

FDDI Frame Fields (continued)

- **Frame check sequence (FCS):** Is filed by the source station with a calculated cyclic redundancy check value dependent on frame contents. The destination address recalculates the value to determine whether the frame was damaged in transit. If so, the frame is discarded.
- **End delimiter:** Contains unique symbols; cannot be data symbols that indicate the end of the frame.
- **Frame status**: Allows the source station to determine whether an error occurred; identifies whether the frame was recognized and copied by a receiving station.

FDDI Characteristics

- 100 Mbps of data throughput
- Two interfaces
- Connects equipment to the ring over long distances
- Allows all stations to have equal amount of time to transmit data
- FDDI is a LAN with Station Management

AKTU 2018-19

- Q:

A large FDDI ring has 100 stations and a token rotation time of 40 msec. The token holding time is 10 msec. What is the maximum achievable efficiency of the ring?

- A:
  - With a rotation time of 40 msec and 100 stations, the time for the token to move between stations is 40/100=0.4 msec.
  - A station may transmit for 10 msec, followed by a 0.4 msec gap while the token moves to the next station.
  - The best case efficiency is then 10/10.4=96%.

Token Rotation Time

An FDDI ring may contain 500 stations and 100km of fiber optic cable. Suppose that each station grabs the token and transmits a 1000 byte frame.

The time for the token to come back to node #1 is:

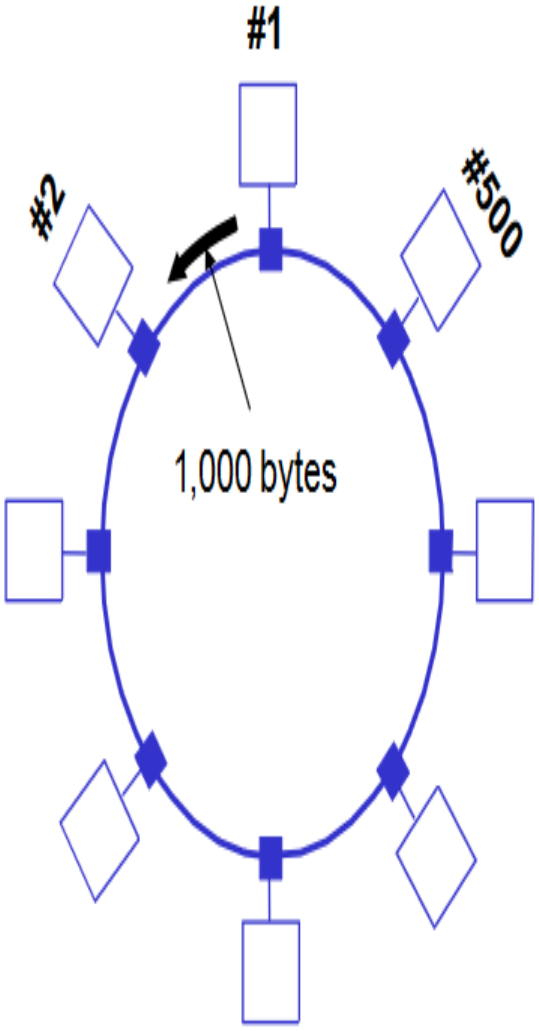
TokenRotationTime =

500 x TransmissionTime + RingLatency

= 500 x (1000x8 / 100 Mb/s) +
(100,000m / 0.65c)

= 40 msec

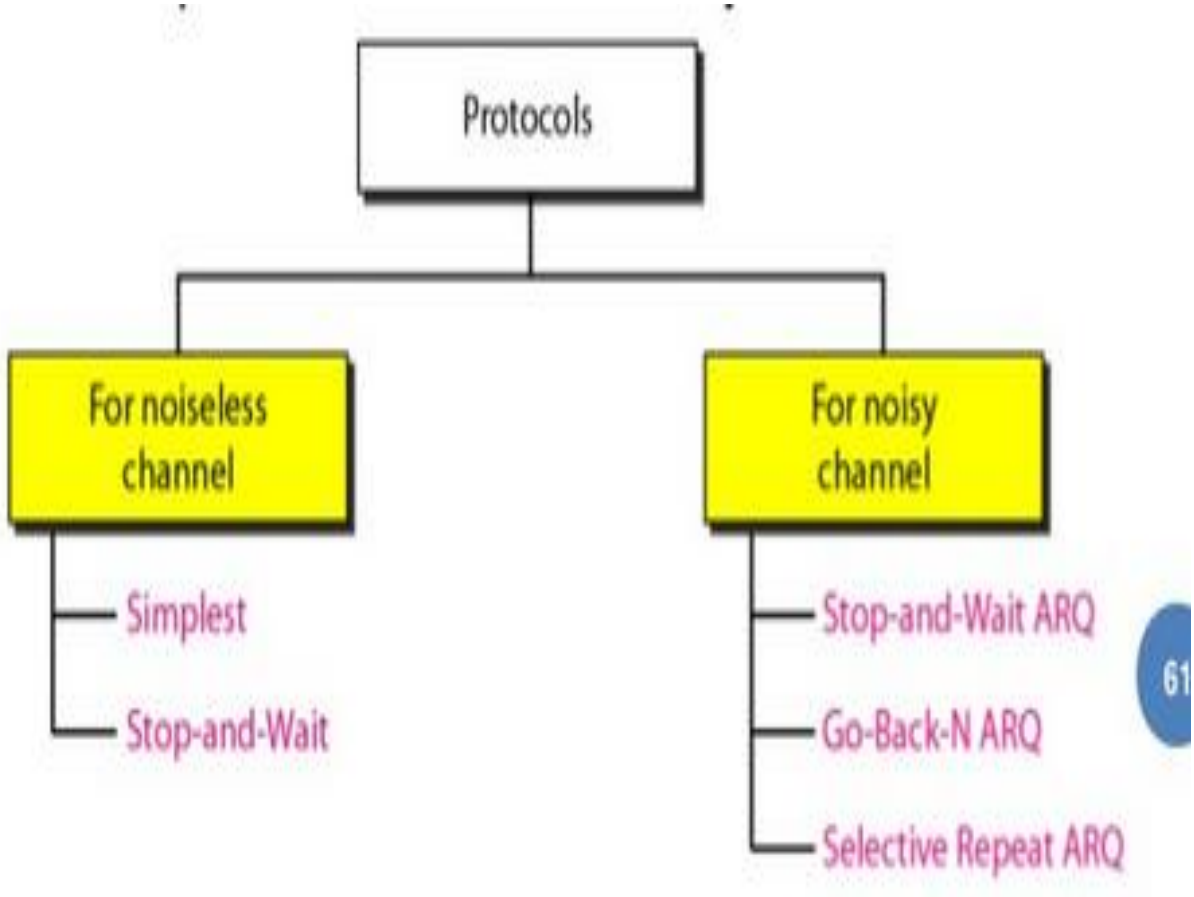
Too slow!!



Data Link Layer

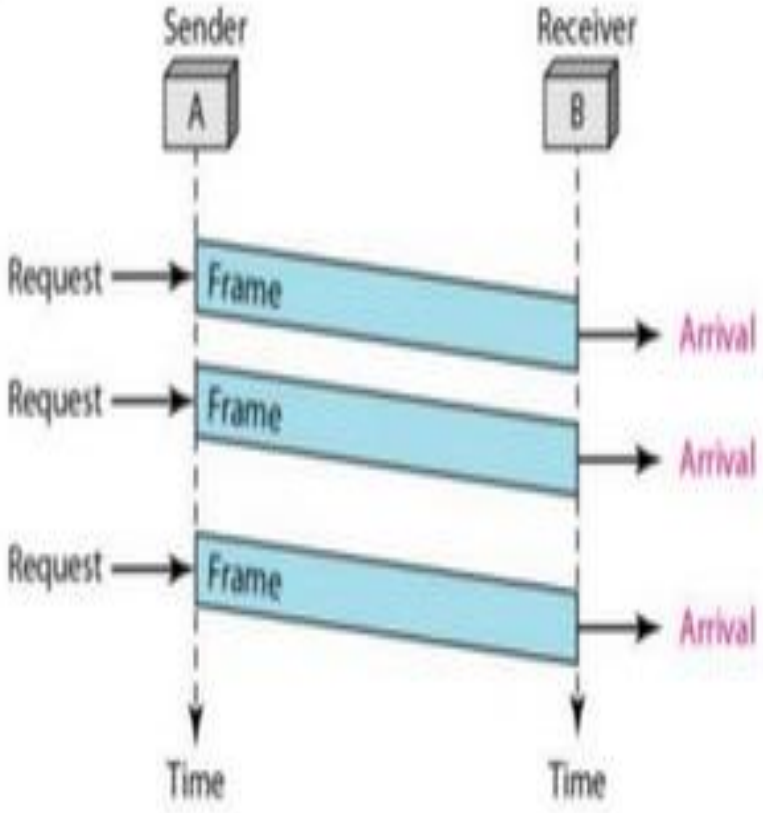
Already Discussed in Unit -1..

Protocols in the data link layer are designed so that this layer can perform its basic functions:. *framing, error control and flow control* **Framing** is the process of dividing bit - streams from physical layer into data frames whose size ranges from a few hundred to a few thousand bytes. **Error control** mechanisms deals with transmission errors and retransmission of corrupted and lost frames. **Flow control** regulates speed of delivery and so that a fast sender does not drown a slow receiver**.**


Protocols
For noiseless channel
For noisy channel
Simplest
Stop-and-Wait
Stop-and-Wait ARQ
Go-Back-N ARQ
Selective Repeat ARQ

AN UNRESTRICTED SIMPLEX PROTOCOL

- In order to appreciate the step by step development of efficient and complex protocols we will begin with a simple but unrealistic protocol. In this protocol: Data are transmitted in one direction only
- The transmitting (Tx) and receiving (Rx) hosts are always ready
- Processing time can be ignored
- Infinite buffer space is available
- No errors occur; i.e. no damaged frames and no lost frames (perfect channel)



62

A SIMPLEX STOP-AND-WAIT PROTOCOL

- In this protocol we assume that Data are transmitted in one direction only
- No errors occur (perfect channel)
- The receiver can only process the received information at a finite rate
- These assumptions imply that the transmitter cannot send frames at a rate faster than the receiver can process them.

The problem here is how to prevent the sender from flooding the receiver.

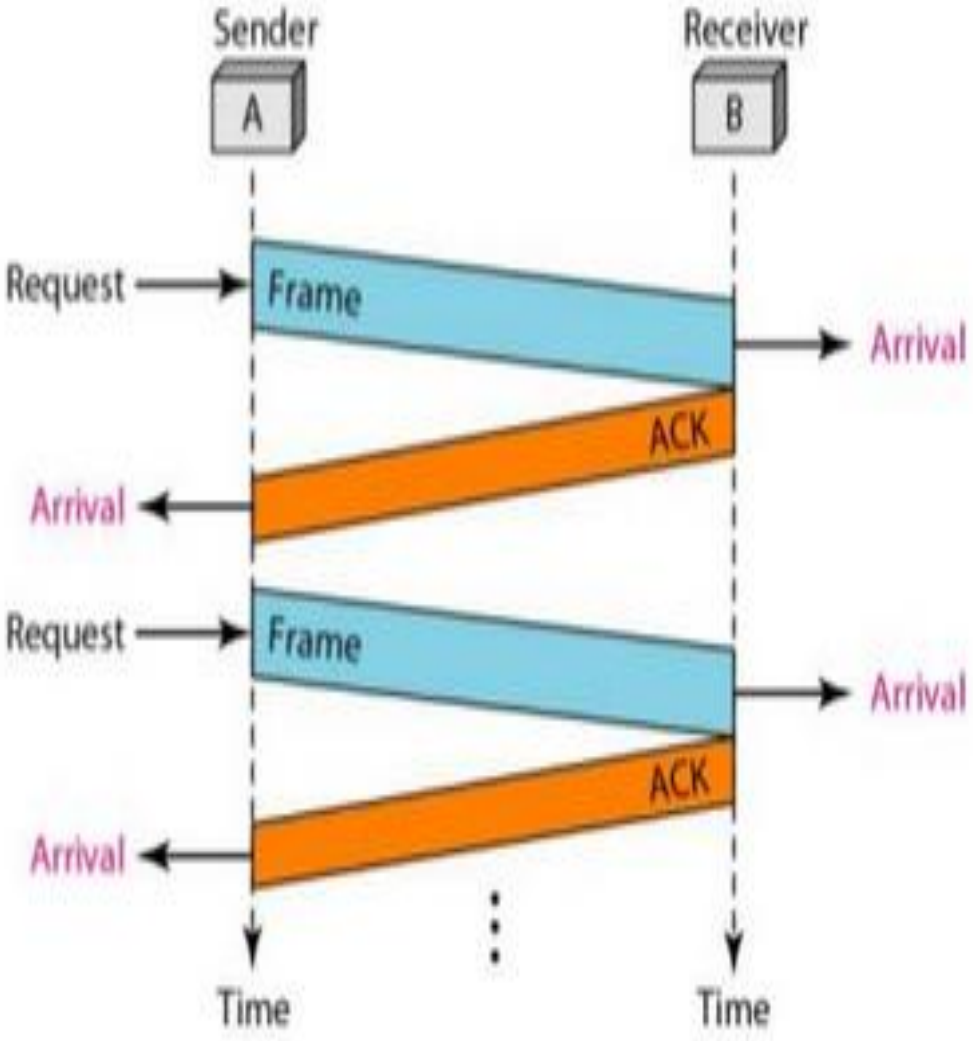
- A general solution to this problem is to have the receiver provide some sort of feedback to the sender. The process could be as follows: The receiver send an acknowledge frame back to the sender telling the sender that the last received frame has been processed and passed to the host; permission to send the next frame is granted. The sender, after having sent a frame, must wait for the acknowledge frame from the receiver before sending another frame.

63

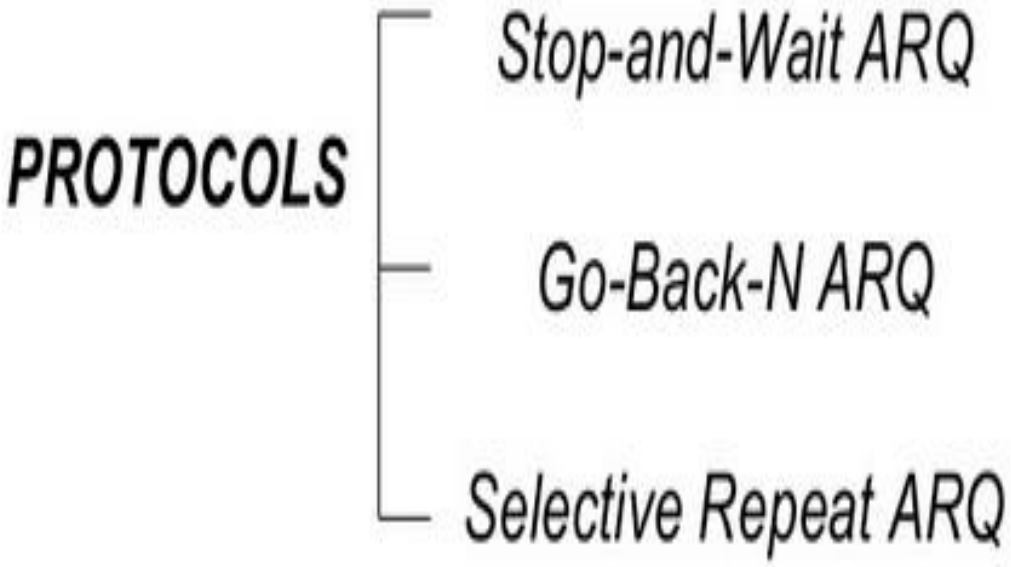
This protocol is known as *stop-and-wait.*

STOP & WAIT PROTOCOL

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



64


PROTOCOLS
Stop-and-Wait ARQ
Go-Back-N ARQ
Selective Repeat ARQ

Stop-and-Wait ARQ

- A transmitter sends a frame then stops and waits for an acknowledgment.
- Stop-and-Wait ARQ has the following features:

✓ The sending device keeps a copy of the sent frame transmitted until it receives an acknowledgment(ACK)

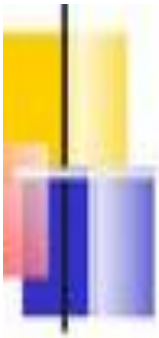
✓ The sender starts a timer when it sends a frame. If an ACK is not received within an allocated time period, the sender resends it

✓ Both frames and acknowledgment (ACK) are numbered alternately 0 and 1(two sequence number only)

✓ This numbering allows for identification of frames in case of duplicate transmission

Stop-and-Wait ARQ

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



Stop-and-Wait ARQ

Cases of Operations:

1. Normal operation
2. The frame is lost
3. The Acknowledgment (ACK) is lost
4. The Ack is delayed

Go-Back-N ARQ

a) Based on sliding window

b) If no error, ACK as usual with next frame expected
   - ACK means "I am ready to receive frame i" and "I received all frames between i and my previous ack"

a) Sender uses window to control the number of unacknowledged frames

b) If error, reply with rejection (negative ack)
   - Discard that frame and all future frames until the frame in error received correctly
   - Transmitter must go back and retransmit that frame and all subsequent frames

Selective Repeat ARQ

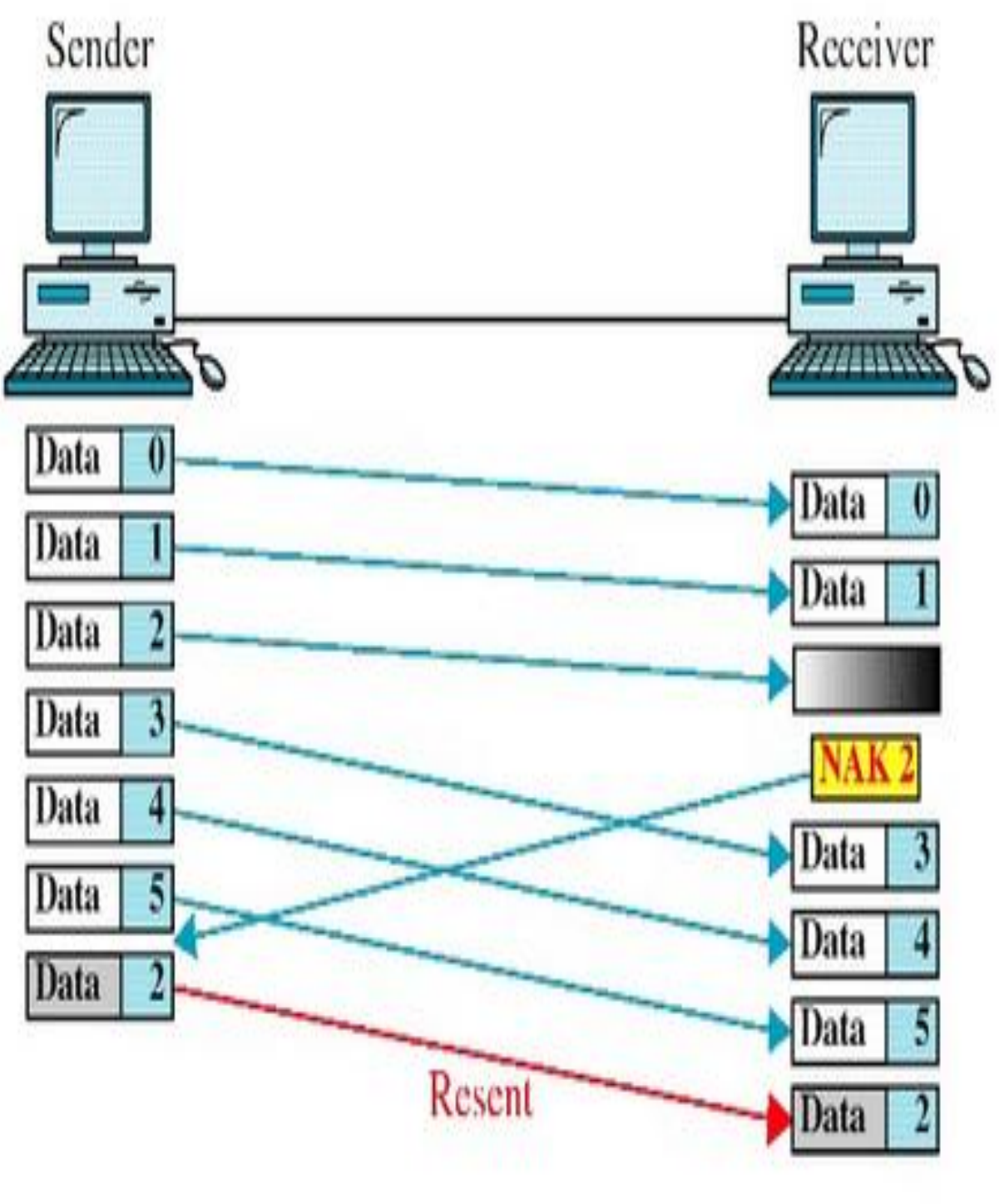
Go-Back-N ARQ is inefficient of a **noisy** link.

- In a noisy link frames have higher probability of damage , which means the resending of multiple frames.
- this resending consumes the bandwidth and slow down the transmission .

Solution:

- Selective Repeat ARQ protocol : resent only the damage frame
- It defines a negative Acknolgment (NAK) that report the sequence number of a damaged frame before the timer expires
- It is more efficient for noisy link, but the processing at the receiver is more complex

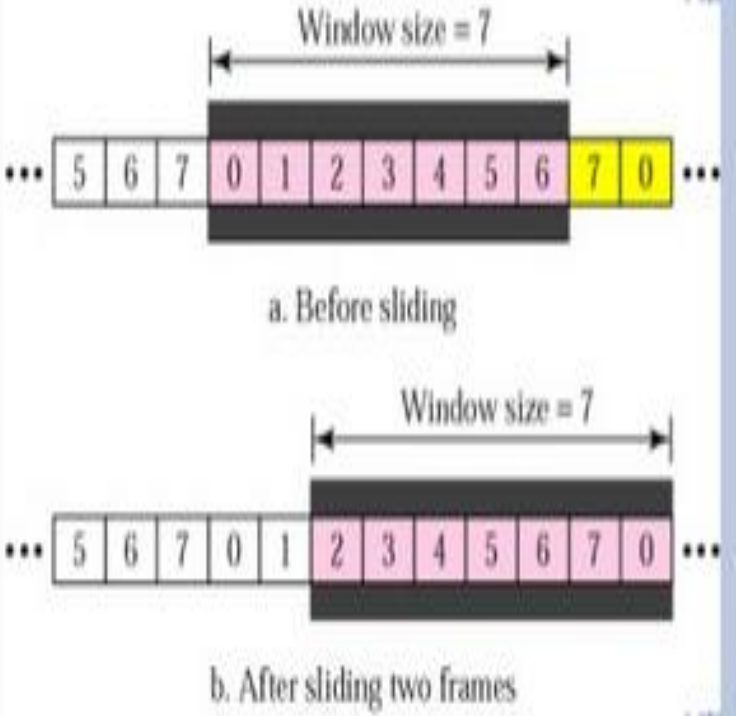
Selective Repeat ARQ



SLIDING WINDOW PROTOCOLS

SENDER SLIDING WINDOW

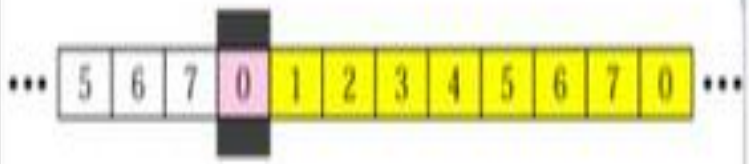
- At the sending site, to hold the outstanding frames until they are acknowledged, we use the concept of a window.
- The size of the window is at most $2^m - 1$ where m is the number of bits for the sequence number.
- Size of the window can be variable, e.g. TCP.
- The window slides to include new unsent frames when the correct ACKs are received



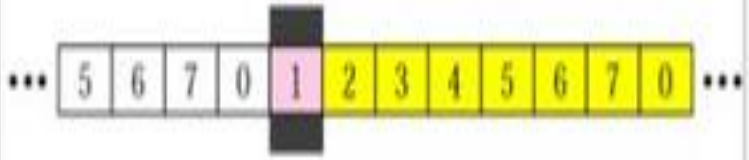
77

RECEIVER SLIDING WINDOW

- Size of the window at the receiving site is always 1 in this protocol.
- Receiver is always looking for a specific frame to arrive in a specific order.
- Any frame arriving out of order is discarded and needs to be resent.
- Receiver window slides as shown in fig.
  - Receiver is waiting for frame 0 in part a.



a. Before sliding

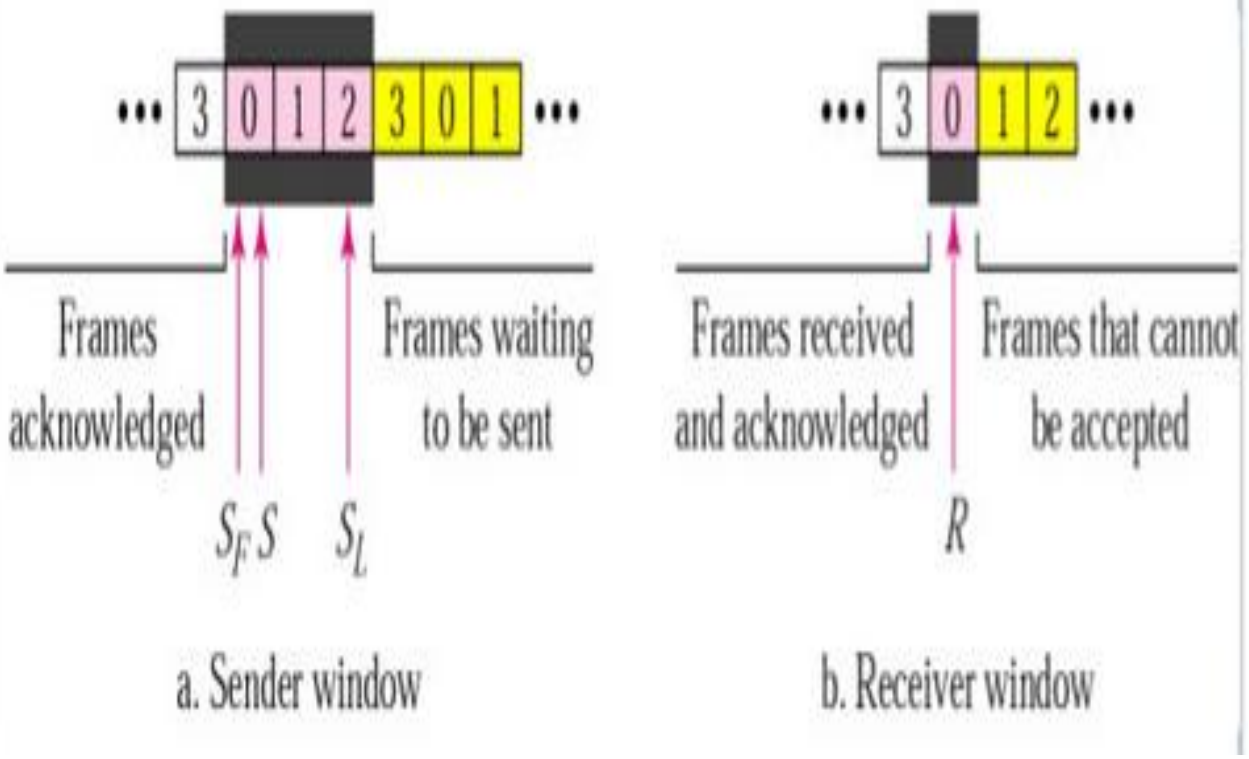


b. After sliding

78

CONTROL VARIABLES

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



PERFORMANCE OF STOP-AND-WAIT PROTOCOL

- Assumption of previous protocols:
  - Transmission time is negligible
  - False, when transmission time is long

- Example - satellite communication
  - channel capacity: 50 kbps, frame size: 1kb

    round-trip propagation delay: 500 msec
  - Time: t=0 start to send 1st bit in frame

    t=20 msec frame sent completely

    t=270 msec frame arrives

    t=520 msec best case of ack. Received

- Sender blocked 500/520 = 96% of time
  - Bandwidth utilization 20/520 = 4%

Conclusion:

Long transit time + high bandwidth + short frame length ⇒ **disaster**

Error Detection and Correction in Data link Layer

Data-link layer uses error control techniques to ensure that frames, i.e. bit streams of data, are transmitted from the source to the destination with a certain extent of accuracy.

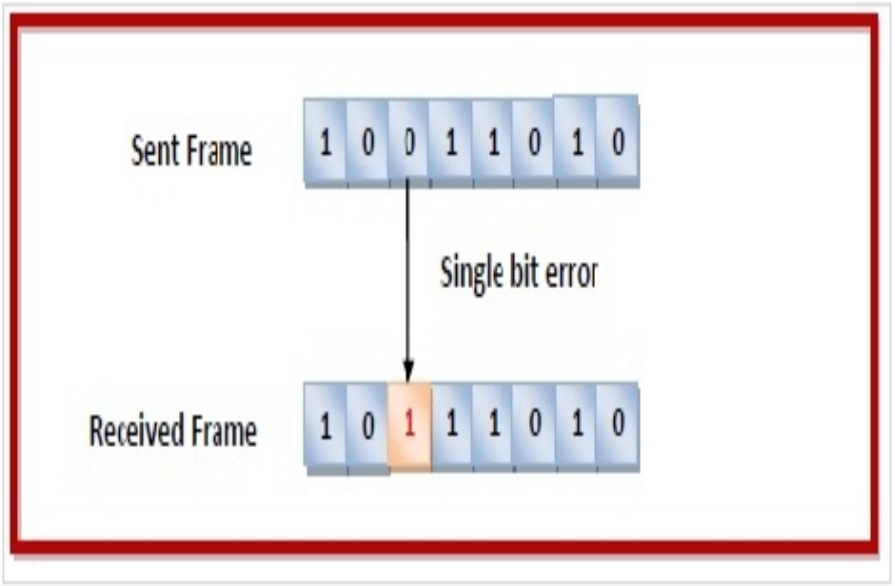
Errors

When bits are transmitted over the computer network, they are subject to get corrupted due to interference and network problems. The corrupted bits leads to spurious data being received by the destination and are called errors.

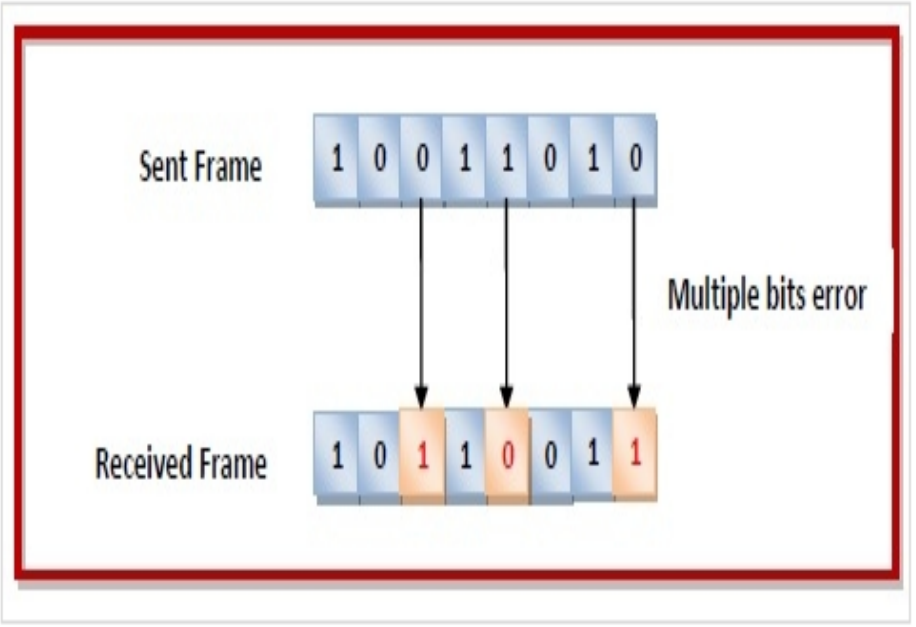
Types of Errors

Errors can be of three types, namely single bit errors, multiple bit errors, and burst errors.

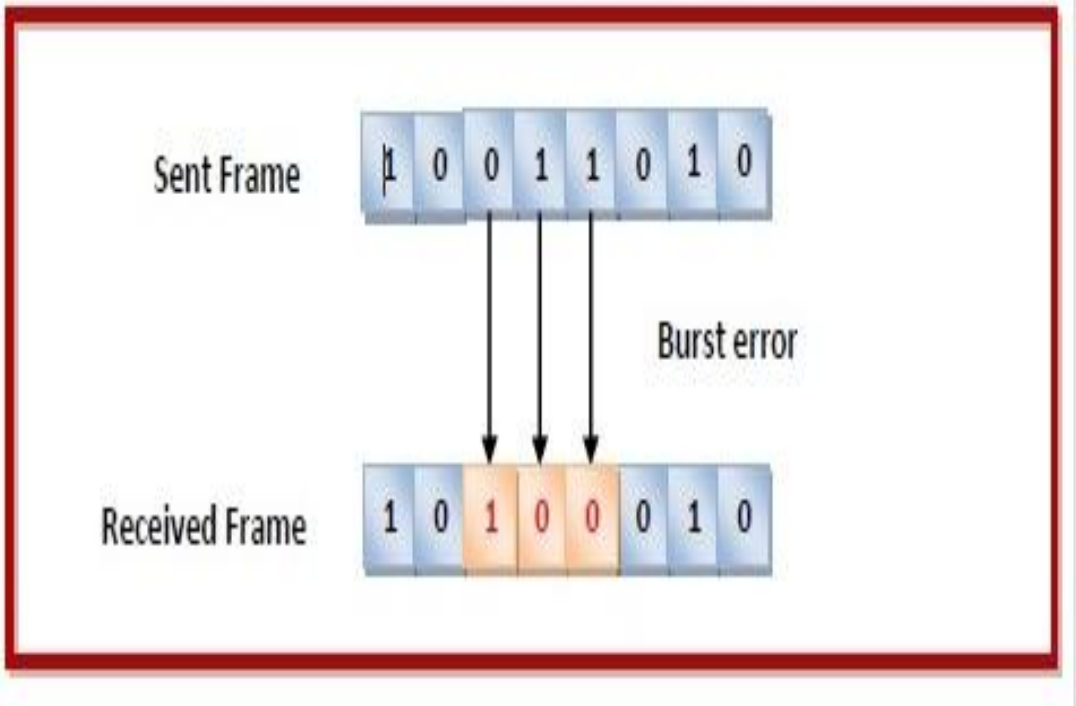
- **Single bit error** – In the received frame, only one bit has been corrupted, i.e. either changed from 0 to 1 or from 1 to 0.



- **Multiple bits error** – In the received frame, more than one bits are corrupted.



- **Burst error** – In the received frame, more than one consecutive bits are corrupted.



Error Control

Error control can be done in two ways –

Error detection − Error detection involves checking whether any error has occurred or not. The number of error bits and the type of error does not matter.

Error correction − Error correction involves ascertaining the exact number of bits that has been corrupted and the location of the corrupted bits.

For both error detection and error correction, the sender needs to send some additional bits along with the data bits. The receiver performs necessary checks based upon the additional redundant bits. If it finds that the data is free from errors, it removes the redundant bits before passing the message to the upper layers.

Error Detection Techniques

There are three main techniques for detecting errors in frames: Parity Check, Checksum and Cyclic Redundancy Check (CRC).

Parity Check

The parity check is done by adding an extra bit, called parity bit to the data to make a number of 1s either even in case of even parity or odd in case of odd parity.

While creating a frame, the sender counts the number of 1s in it and adds the parity bit in the following way

In case of even parity: If a number of 1s is even then parity bit value is 0. If the number of 1s is odd then parity bit value is 1.

In case of odd parity: If a number of 1s is odd then parity bit value is 0. If a number of 1s is even then parity bit value is 1.

On receiving a frame, the receiver counts the number of 1s in it. In case of even parity check, if the count of 1s is even, the frame is accepted, otherwise, it is rejected. A similar rule is adopted for odd parity check.

The parity check is suitable for single bit error detection only.

Checksum

In this error detection scheme, the following procedure is applied

- Data is divided into fixed sized frames or segments.
- The sender adds the segments using 1's complement arithmetic to get the sum. It then complements the sum to get the checksum and sends it along with the data frames.
- The receiver adds the incoming segments along with the checksum using 1's complement arithmetic to get the sum and then complements it.
- If the result is zero, the received frames are accepted; otherwise, they are discarded.

Cyclic Redundancy Check (CRC)

Cyclic Redundancy Check (CRC) involves binary division of the data bits being sent by a predetermined divisor agreed upon by the communicating system. The divisor is generated using polynomials.

- Here, the sender performs binary division of the data segment by the divisor. It then appends the remainder called CRC bits to the end of the data segment. This makes the resulting data unit exactly divisible by the divisor.
- The receiver divides the incoming data unit by the divisor. If there is no remainder, the data unit is assumed to be correct and is accepted. Otherwise, it is understood that the data is corrupted and is therefore rejected.

Error Correction Techniques

Error correction techniques find out the exact number of bits that have been corrupted and as well as their locations. There are two principle ways

Backward Error Correction (Retransmission) − If the receiver detects an error in the incoming frame, it requests the sender to retransmit the frame. It is a relatively simple technique. But it can be efficiently used only where retransmitting is not expensive as in fiber optics and the time for retransmission is low relative to the requirements of the application.
Forward Error Correction − If the receiver detects some error in the incoming frame, it executes error-correcting code that generates the actual frame. This saves bandwidth required for retransmission. It is inevitable in real-time systems. However, if there are too many errors, the frames need to be retransmitted.

The four main error correction codes are
Hamming Codes
Binary Convolution Code
Reed – Solomon Code
Low-Density Parity-Check Code