2.1 Error Control

In data communication, errors are inevitable due to noise, attenuation, distortion, or interference in the transmission medium. Error control is therefore an essential function of data link and transport layers, ensuring that information sent by a sender is received accurately by the receiver. Error control techniques include mechanisms for **error detection** and **error correction**.

Types of Errors

When data is transmitted in binary form (as 0s and 1s), errors occur when bits are altered during transmission.

1. **Single-bit error**
   A single bit in the data unit changes from 0 to 1 or 1 to 0. For example, the original bit sequence 1011011 might be received as 1011111. This is more common in parallel transmission channels.
2. **Burst error**
   Two or more consecutive bits in the data unit are altered. For example, 1011011 might be received as 1000111. Burst errors are more likely in serial transmission channels and noisy lines.
3. **Packet-level error (loss or duplication)**
   Sometimes entire frames or packets may be lost, duplicated, or delivered out of order. While bit-level techniques like parity can't fix this, higher-layer protocols (like TCP) detect missing segments and request retransmission.

Thus, error control must handle not only bit errors but also frame-level issues in practical systems.

Redundancy

The principle of **redundancy** is at the heart of error detection. Instead of sending only the raw data bits, extra bits (redundant information) are added to help the receiver detect or correct errors.

- The sender encodes data with redundancy (extra check bits).
- The receiver uses those bits to verify correctness.
- If an error is detected, the receiver may request retransmission (ARQ) or correct the error if enough information is available (FEC).

For example, a simple **parity bit** adds 1 extra bit to make the total number of 1s either even (even parity) or odd (odd parity). If one bit flips, the parity check fails.

Checksum

A **checksum** is a method used in higher-layer protocols such as IP, UDP, and TCP. Instead of just counting 1s, it uses numerical addition.

- The sender divides the data into fixed-size segments (say 16 bits each).
- These segments are added together using binary arithmetic.
- The sum is complemented (1's complement) and sent as the checksum.
- The receiver repeats the addition with the received data and checksum; if the result is all 1s, no error is detected.

For example, suppose two 16-bit words are 1010101010101010 and 1111000011110000. Their binary sum is added, and the complement is sent as checksum. If the receiver's recomputation does not match, an error is flagged.

Hamming Code

The **Hamming code** is an **error correction** method, invented by Richard Hamming. Unlike parity or checksum (which only detect errors), Hamming codes can **detect and correct single-bit errors** and detect double-bit errors.

The principle is to insert parity bits at positions that are powers of 2 (1, 2, 4, 8, etc.) in the data stream. Each parity bit checks a particular set of bits.
For example, for a 7-bit data word, 4 parity bits are added at positions 1, 2, 4, and 8, creating an 11-bit Hamming code. At the receiver, all parity checks are recomputed. If no parity error is detected, the word is correct. If a parity mismatch occurs, the binary combination of parity-check results points to the exact bit in error, which can then be flipped back to correct the word.

Cyclic Redundancy Check (CRC)
The **Cyclic Redundancy Check (CRC)** is one of the most powerful error detection techniques. It treats the data as a long binary number (polynomial) and divides it by a predetermined generator polynomial.

- The sender performs polynomial division (modulo-2 arithmetic) of the message appended with zeros. The remainder of this division is the CRC, which is appended to the message before transmission.
- The receiver divides the received message (data + CRC) by the same generator polynomial. If the remainder is zero, no error is detected; otherwise, the message is considered corrupted.

For example, suppose the dataword is 101100 and the generator polynomial is 1101 (which corresponds to $x^3 + x^2 + 1$). The sender appends 3 zeros (since degree = 3), performs division, and obtains a remainder (say 011). This remainder is sent as CRC. At the receiver, if the division result is not zero, the frame is rejected.

2.2 Framing and Flow Control
1. Framing
In data communication, the data link layer must transmit raw streams of bits in a way that the receiver can recognize **where a frame begins and where it ends**. This process is called **framing**. A frame is a well-defined block of data that contains not only the payload (actual data) but also control information such as addresses and error-checking bits.
There are different methods of framing:

1. **Character count method**: The first field of the frame specifies the number of characters (bytes) in the frame. However, if this field itself is corrupted, synchronization is lost. **Character stuffing**: A special character (such as DLE in ASCII) is used to mark the start and end of a frame. If this character appears in the data, an escape character is stuffed before it so the receiver does not confuse it with a boundary.
2. **Bit stuffing**: A specific bit pattern, such as 01111110 in HDLC, is used as the frame delimiter. To prevent confusion, the sender automatically inserts a 0 after every five consecutive 1s in the data. The receiver removes these stuffed bits to restore the original data.
3. **Physical layer coding violations**: Sometimes, unused symbols in the physical encoding scheme (like extra voltage levels) are reserved to signal frame boundaries.

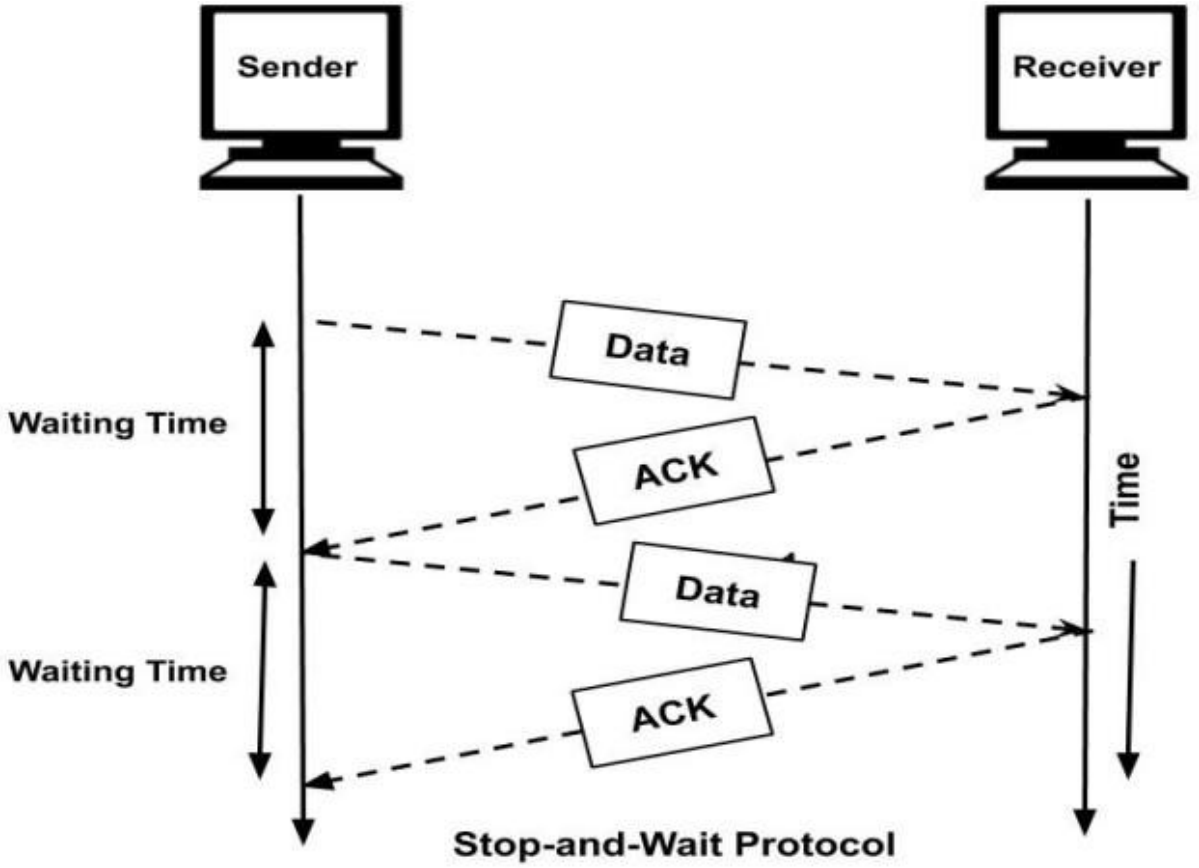
Framing ensures that the receiver knows exactly which bits belong together, enabling higher layers to process data correctly.

2. Flow Control
Flow control refers to techniques that ensure the **sender does not overwhelm the receiver** by transmitting data faster than it can be processed. Imagine a fast sender transmitting at 1 Gbps to a slower device that can only handle 100 Mbps; without flow control, the receiver's buffer would overflow, causing data loss.
Flow control is implemented at the data link and transport layers using special protocols that regulate the rate of transmission.

3. Flow Control Protocols

a) Stop-and-Wait Protocol

In the **stop-and-wait** protocol, the sender transmits one frame and then waits for an acknowledgment (ACK) from the receiver before sending the next frame. This ensures the receiver is never overloaded. While simple and reliable, stop-and-wait is inefficient over long distances, because the sender spends much time waiting. The efficiency is defined by the ratio of transmission time to the total cycle time (transmission + propagation + acknowledgment).



Example: If Host X sends Frame F1 to Host Y, it must wait until Y replies with an ACK before sending F2.

b) Go-Back-N (GBN) Protocol

Go-back-n was introduced because the 'stop-and-wait protocol' was not that efficient as the sender can send only one packet at a time and has to wait to transmit the next frame until the previous frame is acknowledged. In this way, the stop-and-wait protocol wastes the channel bandwidth and even increase the round trip delay.

This is because the stop-and-wait protocol does not use the concept of pipelining. **Pipelining** is a general concept where the processing of the next task starts before the ending of the previous task. In networking, the concept of pipelining specify that the source can send multiple frames before receiving the acknowledgement for the first transmitted frame.

Go-Back-N ARQ Definition

Go-back-n ARQ protocol is a sliding window protocol that uses the concept of pipelining. In go-back-n, the 'N" determines the size of the sender window. The value of N defines the number of frames that can be sent while the sender is waiting for the acknowledgement. The value of N is always greater than 1 else it would behave in a stop-and-wait manner.

The frames at the sender side are numbered sequentially. In go-back-n, the sender can buffer N frames whereas the size of the receiver window is 1 which means the receiver can buffer only one frame at a

time. However, there can be multiple data frames and acknowledgement in the channel at the same time and the sliding sender window can also slide by one or more frames at once.

If the sender does not receive an acknowledgement before the corresponding timer expires the sender retransmit all the frames in the current sender window. While if the receiver receives a corrupted frame it silently discards that frame without taking any action this results in the expiration of the corresponding timer at the sender side which lead to retransmission of frames in the current sender window. That's why it is referred to as an automatic repeat request.

Working and Example of Go-Back-N

Let us understand the working of go-back-n with the help of an example. Consider that the sender has 16 frames to transmit. The first thing is to provide a sequence number to the frame so that each frame has an independent identity.

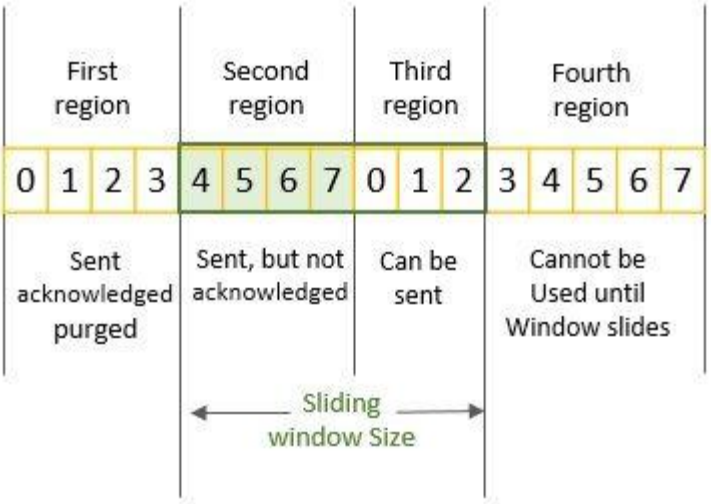
Sequencing

The frames to be transmitted are numbered using modulo 2^m. Where **m** is the number of bits allowed by the frame header to represent the sequence number of a frame. Now if the sender has 16 frames to be transmitted which if number serially would lie from 0, 1, 2 …, 15.

Consider that the header allows only 3 bits to represent the sequence number of the frame i.e. m= 3. The sequence number would range from 0 to 2^3-1 i.e. from 0 to 7. So the 16 frames at the sender side are numbered as 0, 1, 2, 3, 4, 5, 6, 7, 0, 1, 2, 3, 4, 5, 6, 7 and so on.

Sender Sliding Window

For the proper working of the protocol, the maximum size of the sliding window in go-back-n must be $2^m - 1$. As in the explanation above we have m=3 the maximum size of the window would be $2^2 - 1 = 7$.



Sender Window in Go-Back-N

Now, the entire sender's window can be classified into four regions. The very *first* region of the sender window has the sequence numbers of frames that are acknowledged and the sender can purge these frames. The sequence number of frames in the second region are those frames that are sent but not yet acknowledged by the receiver. These are also called outstanding packets. The third region is the sequence of frame numbers that can be sent once the frames with the sequence number in the second region are acknowledged. The fourth region has sequence numbers that can be used when the window slides.

Depending on the number of acknowledgements arrived the sliding window can slide one or more slots toward the right. Consider that the sender has received acknowledgement for frame with sequence number 4 then it will slide only one slot towards the right.

If the sender receives two acknowledgements for the frame with sequence numbers 4 and 5 then it will slide two slots towards the right.

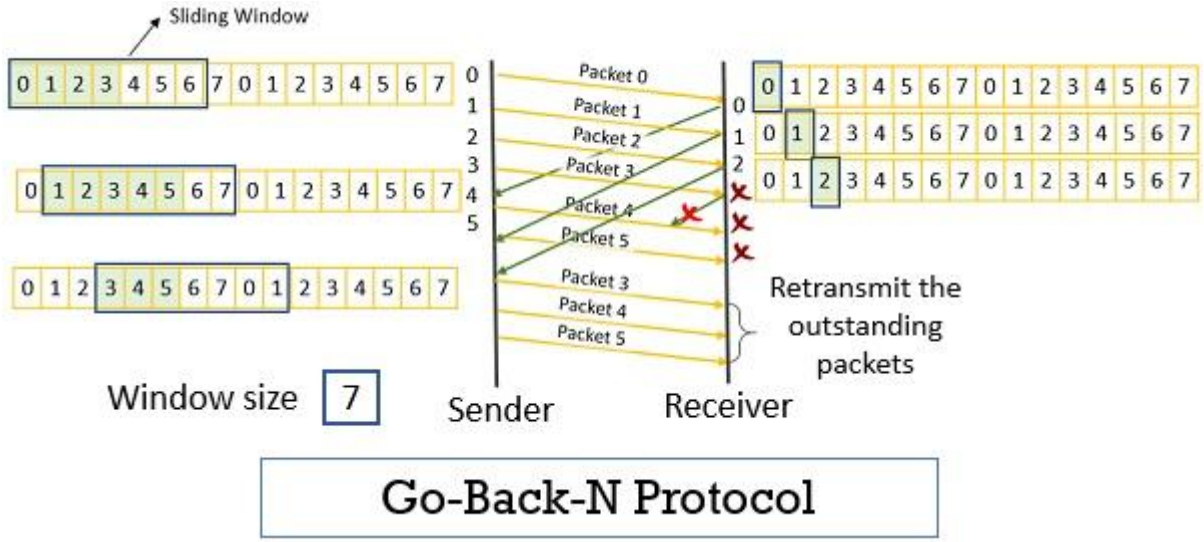
Receiver Sliding Window

In go-back-n, the size of the receiver side sliding window is 1 which limits the receiver to look for the arrival of a specific packet. The receiver in go-back-n accepts the packets sequentially. If it receives any packet out of order then it discards those packets which are later retransmitted by the sender.

The receiver window can be categorized into three regions. The first region to the left has the sequence number of the frames that are received and even acknowledged. The second region has a sequence number of only one frame that the receiver is expecting to arrive. The third region has a sequence of frames that cannot be accepted.

Now let us understand the working of go-back-n step by step with the help of the figure below.

The sender sends frames 0, 1, 2, 3 while waiting for the acknowledgement of the first outstanding packet i.e. 0. However, the sender sliding window covers the sequence number of the frames from 0 to 6 as the sliding window size here is 7.



The frames with sequence numbers 0, 1, 2, and 3 are the outstanding packet means these frames are sent and the sender is waiting for the acknowledgement whereas the data for frames 4, 5, and 6 is yet to come from the sender's application layer.

Meanwhile, the sender receives an acknowledgement for frame 0, the senders sliding window slides by 1 slot to right covering sequence number 7. Even the data for frames with sequences 4 and 5 is received from the sender's application layer. And the sender sends a frame with sequence numbers 4 and 5.

After sending packet 5 the sender receives the acknowledgement for packets 1 and 2 which makes the sender sliding window shift by two slots covering sequence numbers 0 and 1.

However, in the meantime, the timer for acknowledgement of packet 3 expires. The reason may be that packet 3 may have arrived at the receiver but with an error or the packet may have lost or may its acknowledgement may have lost.

When the timer for acknowledgement of packet 3 expires the sender has to go back and resend all the outstanding packets i.e. packets 3, 4, and 5.

So this is how the go-back-n protocol controls error and flow of the data packets from sender to receiver. Though the go-back-n protocol is still inefficient as for one loosed packet the sender has to retransmit all the outstanding packets wasting the bandwidth of the channel.

c) Selective Repeat (SR) Protocol

Selective repeat protocol is a sliding window protocol that uses the concept of pipelining where multiple packets can be sent while the sender is waiting for the acknowledgement for the first sent packet. The selective repeat protocol manages error and flows control between the sender and receiver.

In this section, we will discuss the need for selective repeat ARQ (Automatic Repeat Request) and how it overcomes the shortcomings of the go-back-n protocol. We will also discuss the working of selective repeat with the help of an example.

Why Selective Repeat ARQ?

The go-back-n protocol that we have studied in our previous content is efficient if the error rate and loss of packets is less. If the connection is poor, there will be frequent loss of packets and the sender would have to retransmit all the outstanding packets. This wastes the bandwidth of the channel.

In go-back-n, the size of the receiver window is 1 so it buffers only one packet in order that it has to acknowledge next and if this expected packet is lost or corrupted all the received out of order packets are discarded. And the sender has to retransmit all the outstanding packets though some of these may have arrived at the receiver safely but out of order.

This retransmission of packets increases the traffic on the network creating a cumulative increase in congestion. The alternative approach for this is selective repeat protocol. In the section ahead we will discuss the working of this selective repeat protocol.

Working of Selective Repeat Protocol

Like go-back-n, selective repeat protocol is also a sliding window protocol. Unlike the go-back-n protocol, the selective repeat protocol resends only a selective packet that is either lost or corrupted. Let us first discuss the windows in selective repeat.

Window

In selective repeat, both sender and receiver have a sliding window. On the sender side, the window covers the sequence of packets that are either sent or can be sent. At the receiver, the sliding window covers the sequence number of the packets that are either received or are expected to be received. In selective repeat, the size of the sender and receiver window is the same. Let us study the sender window and receiver window in brief.

1. Sender Window

The sender window in selective repeat is much smaller as compared to the go-back-n protocol. The size of the sender window here is 2^{m-1}. Here m is the number of bits used by the packet header to express the sequence number of the corresponding packet.

The sender window covers the packets that are sent but not yet acknowledged, one that is acknowledged out of order and the one that can be sent once the data for the corresponding are received by the sender's application layer.

2. Receiver Window

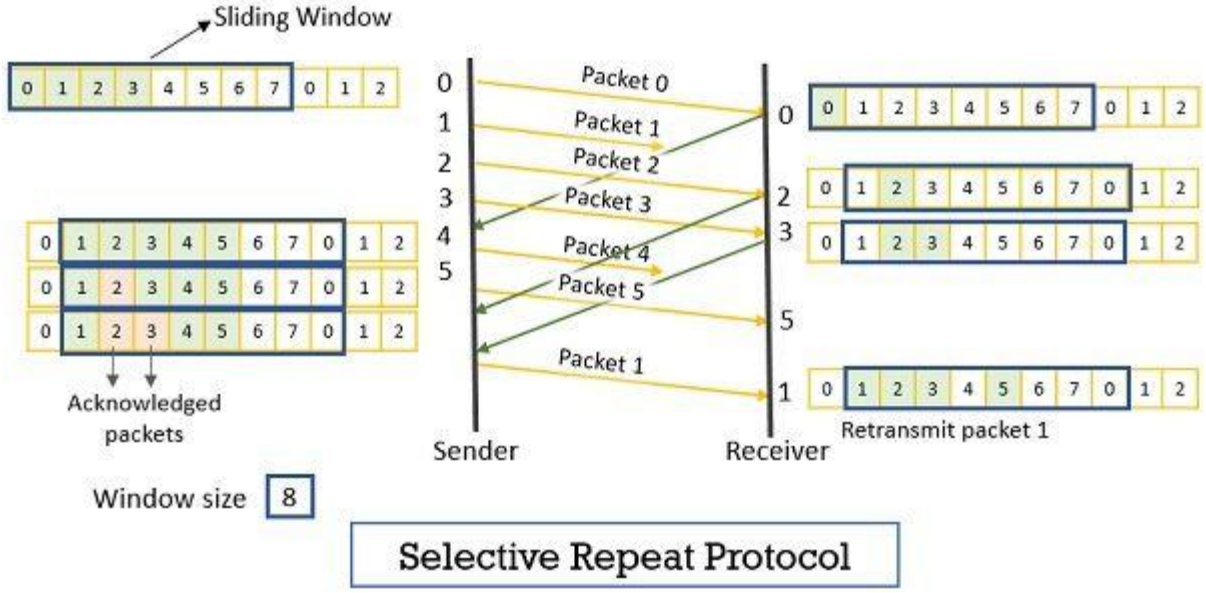
The maximum size of the receiver window is 2^{m-1} which is the same as the sender window. The receiver window covers the sequence number of the packets that are received out of order and are waiting for the packets that were sent earlier but are not yet received.

The receiver transport layer does not deliver packets out of order to the application layer. It waits until a set of consecutive packets are received so that they can be delivered to the application layer.

In selective repeat, at the sender side, a timer is attached to each sent packet and if the acknowledgement is not received before the timer expire the corresponding packet is resent.

Let understand selective repeat protocol with the help of an example. Consider that the header of a packet has allotted 3 bits to resemble the sequence number of the corresponding packet so m=3 here. Now as m is 3 the packets are sequenced using modulo 2^m. So the sequence number of packets lie from 0, 1, 2, 3, …, 7.

The size of the receiver and sender window will be 2^{m-1} i.e. $2^{3-1} = 4$. So the size of the receiver and sender window is 4.



Operation at Sender Side

As you can see in the figure above the sender window covers the sequence number of the packets from 0 to 7 as the window size is 8. The sender has sent packets 0, 1, 2, and 3 and is waiting to receive the data of packets 4, 5, 6, and 7 for its application layer.

Before the application layer provides data for packets 4 and 5, the sender receives an acknowledgement for frame 0 and the sender window slides by one slot. After receiving data for packets 4 and 5 the sender sends packets 4 and 5.

Later the sender receives out of order acknowledgement for packets 2 and 3. The sender window does not slide as it has not yet received acknowledgement for packet 1. Meanwhile, the timer for packet 1 get expired and the sender has to resend packet 1.

Operation at Receiver Side

The size of the receiver window is 8 so the receiver window covers the sequence number from 0 to 7. If first receives packet 0 and immediately send an acknowledgement for it. Later it receives out of order packets 2 and 3 and sends an acknowledgement for these received packets. Late it again receives an out of order packet 5 and 1.

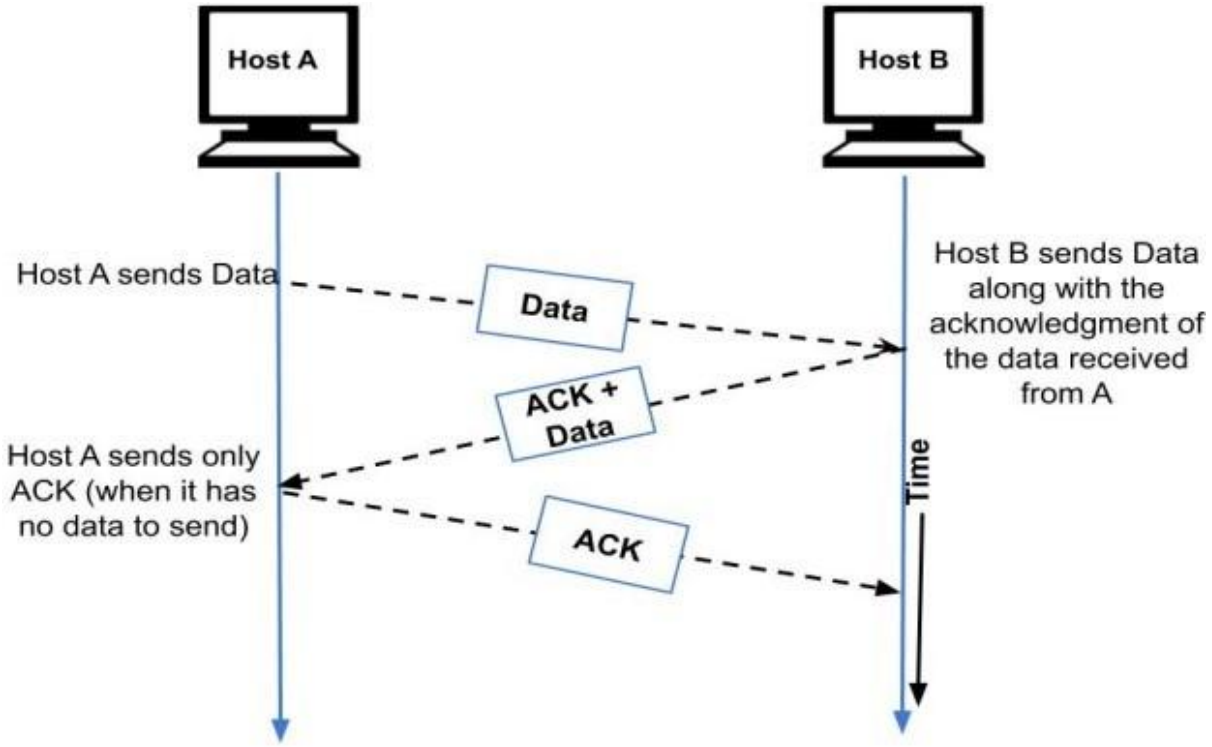
The receiver stores the packets in the receiver window until there is a set of consecutive packets to be delivered to the application layer. In selective repeat, there should be a complex logic to reinsert the retransmitted packet in proper order. However, the sender must also have complex logic to identify and resend the packet out of order.

In this way, the selective repeat avoids traffic on the network by retransmitting the corrupted the lost packet only. And this is possible as it allows the sending and receiving packets out of order.

d) Piggybacking

Piggybacking is a method of **attaching acknowledgment to the outgoing data packet** . The concept of piggybacking is explained as follows:

Consider a two-way transmission between host **A** and host **B** . When host A sends a data frame to B, then B does not send the acknowledgment of the frame sent immediately. The acknowledgment is delayed until the next data frame of host B is available for transmission. The delayed acknowledgment is then attached to the outgoing data frame of B. This process of delaying acknowledgment so that it can be attached to the outgoing frame is called **piggybacking** .



Now, as we are communicating between the host A and host B, three conditions can arise:

1. When the host has both data and the acknowledgment to send, then it will attach the data along with the acknowledgment. In the above diagram, the host B will attach the data frame along with the acknowledgment of the last frame received from host A.
2. When the host does not have any data to send then it will send only the acknowledgment. In the above diagram, when host A does not have any data frame to send. So, it will only send the acknowledgment of the last frame received.
3. When the host has only data to send then it will send the data along with the acknowledgment of the last frame received. The duplicate acknowledgment will be discarded by the receiver and the data would be accepted.

Advantages of Piggybacking

1. The available channel bandwidth is used efficiently.

Disadvantages of Piggybacking

1. As there is delayed transmission of acknowledgment so if the acknowledgment is not received within the fixed time then the sender has to retransmit the data.
2. There is additional complexity for implementing this method.

Framing ensures that bits are grouped correctly into frames for transmission, while flow control prevents sender overflow and keeps communication efficient. Stop-and-Wait is simple but inefficient for long delays. Go-Back-N improves throughput but retransmits unnecessarily on errors. Selective Repeat is more efficient but requires extra buffer and complexity. Piggybacking optimizes duplex channels by combining ACKs with data. Together, these mechanisms form the foundation of reliable and efficient data link and transport layer communication.

2.3 MAC Address and Random Access Protocols

1. MAC Address

The **MAC (Media Access Control) address** is a **hardware identifier** assigned to a network interface card (NIC) by the manufacturer. It uniquely identifies a device on a local network. A MAC address is 48 bits long, usually written in hexadecimal format as six groups of two characters (e.g., 08:00:27:5B:9C:EF).

- The **first 24 bits** (Organizationally Unique Identifier, OUI) specify the manufacturer.
- The **last 24 bits** (NIC-specific) uniquely identify the device.

MAC addresses operate at the **Data Link Layer** of the OSI model and are used for **local delivery** within a LAN segment. They are permanent to the hardware, unlike IP addresses, which are logical and can change.

Example: If Host A (MAC: AA-BB-CC-DD-EE-01) sends a frame to Host B (MAC: AA-BB-CC-DD-EE-02) in an Ethernet LAN, the NICs use MAC addresses to deliver the frame. The IP addresses don't matter within the LAN; MAC is the ultimate identifier.

2. Random Access Protocols

When multiple devices share a common communication channel (like in Ethernet or Wi-Fi), there must be rules for **who gets to transmit** and how **collisions** are handled if two devices transmit simultaneously. Random Access Protocols are contention-based techniques where each node competes for channel access.

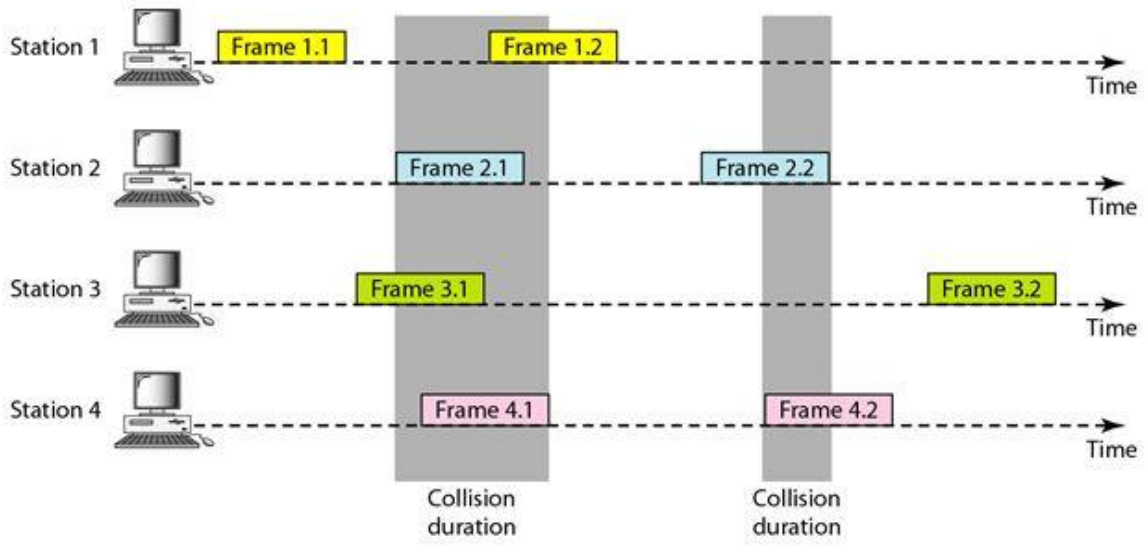
ALOHA Protocols

ALOHA, the earliest random access method was developed at the University of Hawaii in early 1970. It was designed for a radio (wireless) LAN, but it can be used on any shared medium.

It is obvious that there are potential collisions in this arrangement. The medium is shared between the stations. 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.

Pure ALOHA

The original ALOHA protocol is called pure ALOHA. This is a simple, but elegant protocol. The idea is that each station sends a frame whenever it has a frame to send. However, since there is only one channel to share, there is the possibility of collision between frames from different stations. The following figure shows an example of frame collisions in pure ALOHA.

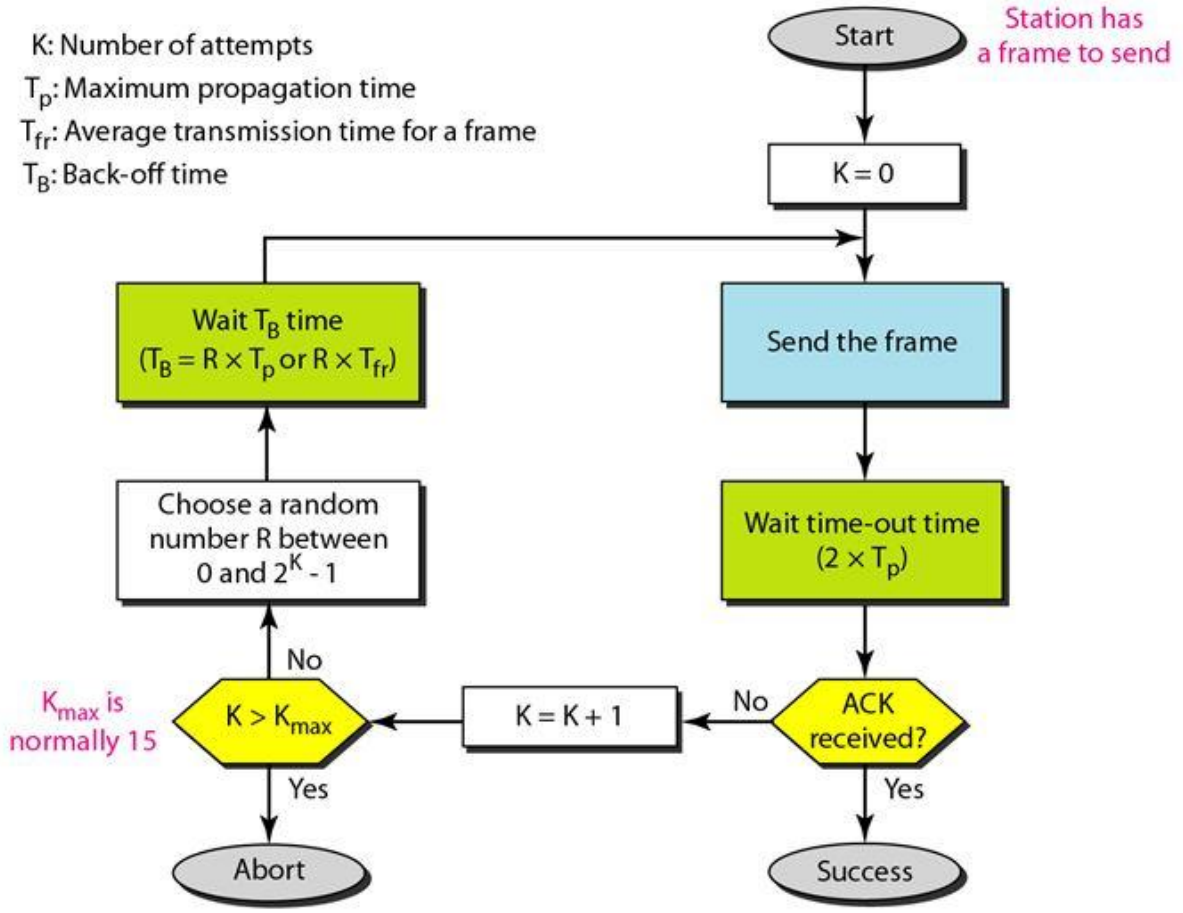


There are four stations (unrealistic assumption) that contend with one another for access to the shared channel. The figure shows that each station sends two frames; there are a total of eight frames on the shared medium. Some of these frames collide because multiple frames are in contention for the shared channel.

The above figure shows that only two frames survive: frame 1.1 from station 1 and frame 3.2 from station 3. We need to mention that even if one bit of a frame coexists on the channel with one bit from another frame, there is a collision and both will be destroyed.

It is obvious that we need to resend the frames that have been destroyed during transmission. The pure ALOHA protocol relies on acknowledgments from the receiver. If the acknowledgment does not arrive after a time-out period, the station assumes that the frame (or the acknowledgment) has been destroyed and resends the frame.A collision involves two or more stations. If all these stations try to resend their frames after the time-out, the frames will collide again. Pure ALOHA dictates that when the time-out period passes, each station waits a random amount of time before resending its frame. The randomness will help avoid more collisions. We call this time the back-off time TB.

Pure ALOHA has a second method to prevent congesting the channel with retransmitted frames. After a maximum number of retransmission attempts Kmax a station must give up and try later. The following figure shows the procedure for pure ALOHA based on the above strategy.



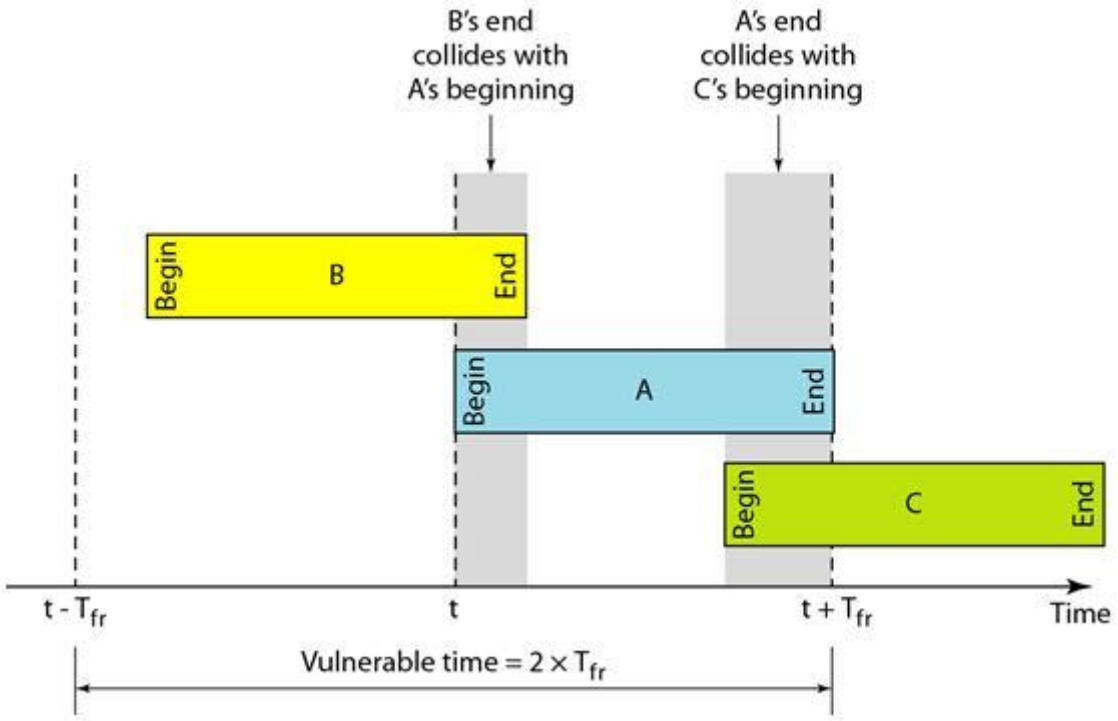
The time-out period is equal to the maximum possible round-trip propagation delay, which is twice the amount of time required to send a frame between the two most widely separated stations (2 x Tp)

The back-off time TB is a random value that normally depends on K (the number of attempted unsuccessful transmissions). The formula for TB depends on the implementation. One common formula is the binary exponential back-off.

In this method, for each retransmission, a multiplier in the range 0 to 2K - 1 is randomly chosen and multiplied by Tp (maximum propagation time) or Tfr (the average time required to send out a frame) to find TB' Note that in this procedure, the range of the random numbers increases after each collision. The value of Kmax is usually chosen as 15.

Vulnerable time:

The vulnerable time is in which there is a possibility of collision. We assume that the stations send fixed-length frames with each frame taking Tfr S to send. The following figure shows the vulnerable time for station A.



Station A sends a frame at time t. Now imagine station B has already sent a frame between t - Tfr and t. This leads to a collision between the frames from station A and station B. The end of B's frame collides with the beginning of A's frame. On the other hand, suppose that station C sends a frame between t and t + Tfr . Here, there is a collision between frames from station A and station C. The beginning of C's frame collides with the end of A's frame.

Throughput:

Let us call G the average number of frames generated by the system during one frame transmission time. Then it can be proved that the average number of successful transmissions for pure ALOHA is S = G x e-2G. The maximum throughput Smax is 0.184, for G = 1. In other words, if one-half a frame is generated during one frame transmission time (in other words, one frame during two frame transmission times), then 18.4 percent of these frames reach their destination successfully.

2.2 Slotted ALOHA

To improve efficiency, **Slotted ALOHA** divides time into equal slots, each equal to one frame transmission time. Stations can only start transmitting at the beginning of a slot.

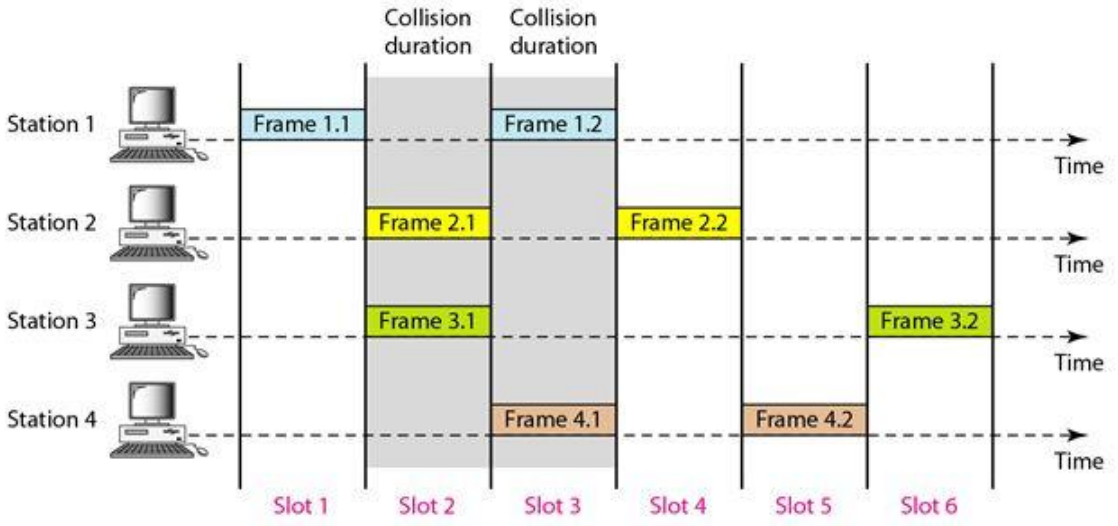
- If two or more stations start in the same slot → collision.
- If exactly one station transmits in a slot → success.

The efficiency of Slotted ALOHA doubles compared to Pure ALOHA, reaching about **36.8%**. This is because the vulnerable period is reduced to **1 × frame time**.

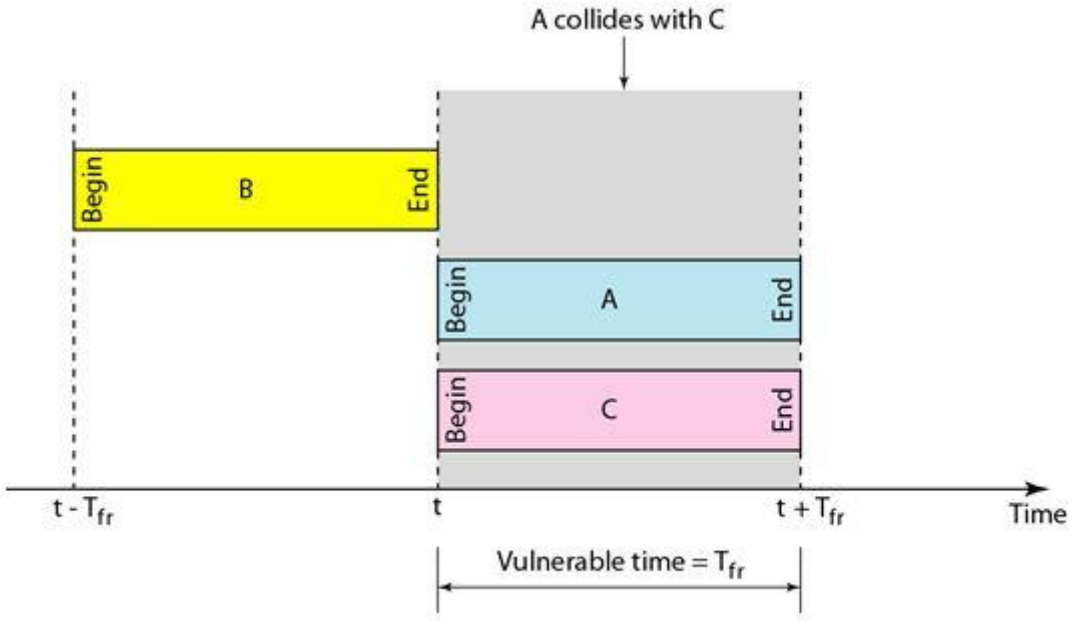
Example: If slots are 10 ms long and the channel is 100 kbps, the maximum throughput is ~36 kbps.

Pure ALOHA has a vulnerable time of 2 x Tfr. 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 soon before another station has finished. Slotted ALOHA was invented to improve the efficiency of pure ALOHA.

In slotted ALOHA we divide the time into slots of Tfr s and force the station to send only at the beginning of the time slot. The following figure shows an example of frame collisions in slotted ALOHA.



Because a station is allowed to send only at the beginning of the synchronized time slot, if a station misses this moment, it must wait until the beginning of the next time slot. This means that the station which started at the beginning of this slot has already finished sending its frame. But, still there is the possibility of collision if two stations try to send at the beginning of the same time slot. However, the vulnerable time is now reduced to one-half, equal to Tfr. The following figure shows the situation.



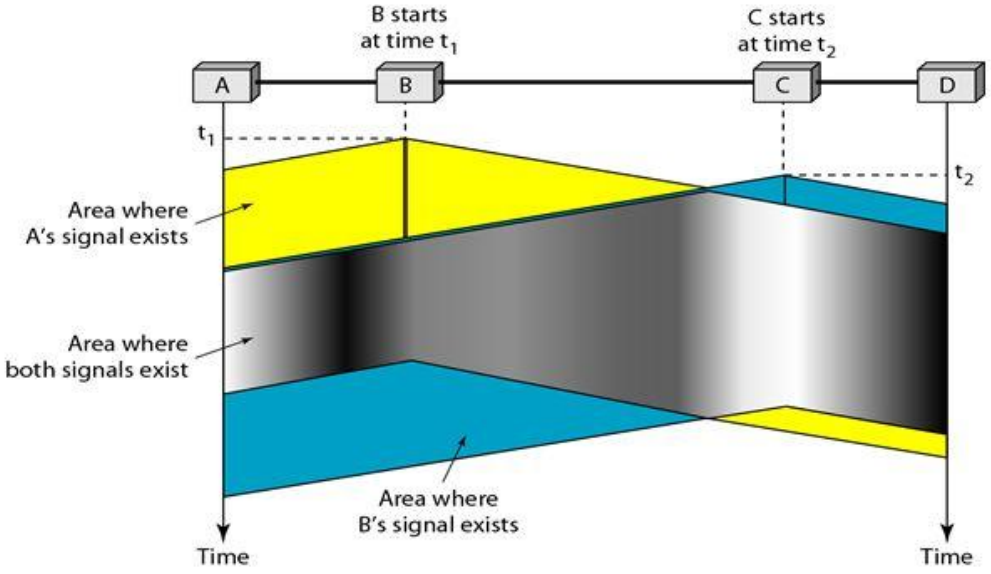
Throughput:

It can be proved that the average number of successful transmissions for slotted ALOHA is S = G x e-G. The maximum throughput Smax is 0.368, when G = 1. In other words, if a frame is generated during one frame transmission time, then 36.8 percent of these frames reach their destination successfully.

2.3 CSMA (Carrier Sense Multiple Access)

Carrier Sense Multiple Access (CSMA) Protocol

To minimize the chance of collision and, therefore, increase the performance, the CSMA method was developed. 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. The following figure shows a space and time model of a CSMA network. Stations are connected to a shared channel (usually a dedicated medium).

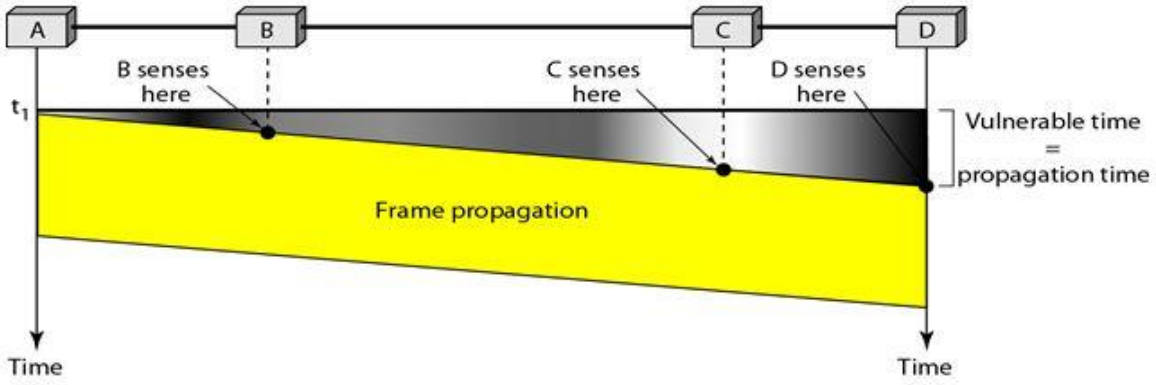


The possibility of collision still exists because of propagation delay, when a station sends a frame, it still takes time (although very short) for the first bit to reach every station and for every station to sense it. In other words, a station may sense the medium and find it idle, only because the first bit sent by another station has not yet been received.

At time t1 station B senses the medium and finds it idle, so it sends a frame. At time t2 (t2> t1) station C senses the medium and finds it idle because, at this time, the first bits from station B have not reached station C. Station C also sends a frame. The two signals collide and both frames are destroyed.

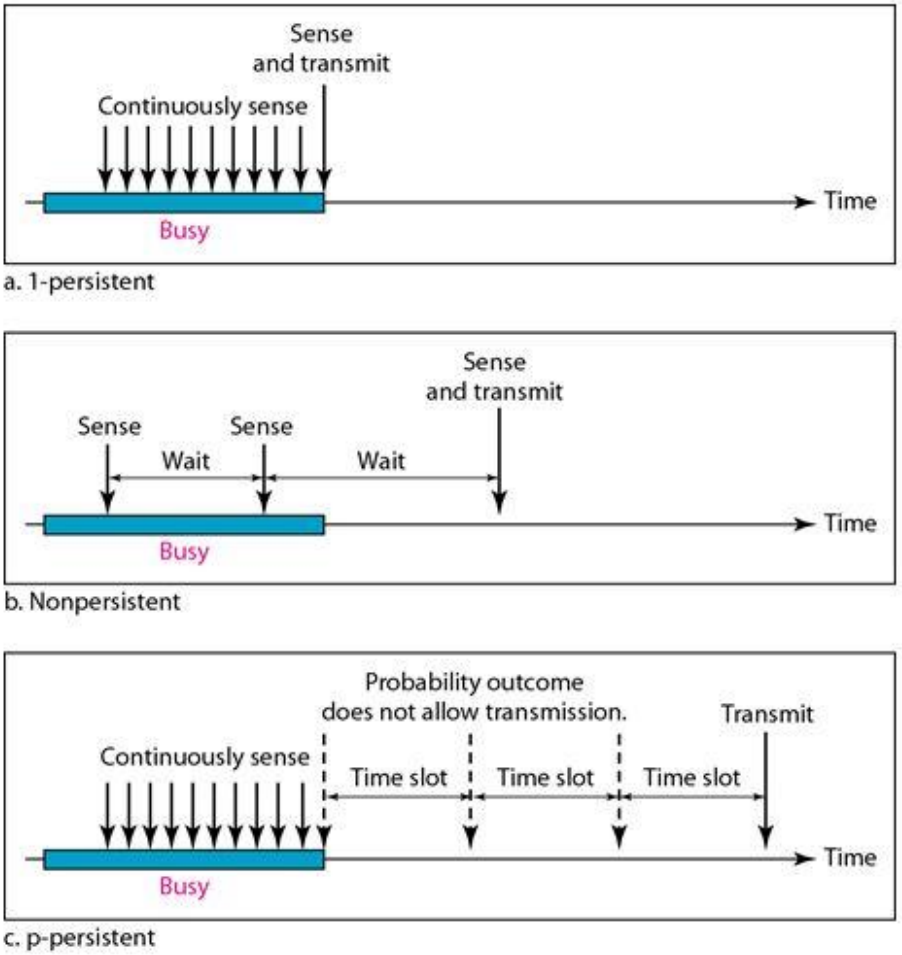
Vulnerable Time:

The vulnerable time for CSMA is the propagation time Tp. This is the time needed for a signal to propagate 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. The following figure shows the worst case. The leftmost station A sends a frame at time t1 which reaches the rightmost station D at time t1 + Tp. The gray area shows the vulnerable area in time and space.



Persistence Methods:

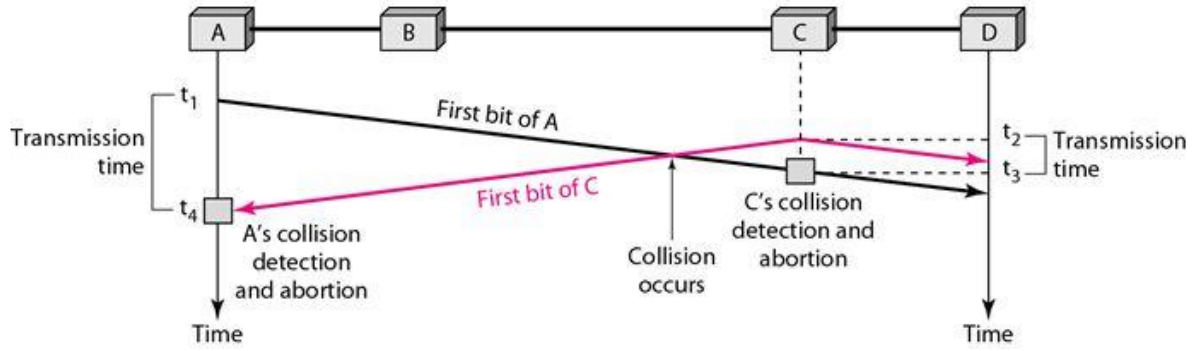
What should a station do if the channel is busy? What should a station do if the channel is idle? Three methods have been devised to answer these questions: the 1-persistent method, the nonpersistent method, and the p-persistent method. The following figure shows the behavior of three persistence methods when a station finds a channel busy.



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

• **Nonpersistent:** In the nonpersistent method, a station that has a frame to send senses the line. If the line is idle, it sends immediately. If the line is not idle, it waits a random amount of time and then senses the line again. The nonpersistent approach reduces the chance of collision because it is unlikely that two or more stations will wait the same amount of time and retry to send simultaneously. However, this method reduces the efficiency of the network because the medium remains idle when there may be stations with frames to send.
• **P-Persistent:** The p-persistent method is used if the channel has time slots with a slot duration equal to or greater than the maximum propagation time. The p-persistent approach combines the advantages of the other two strategies. It reduces the chance of collision and improves efficiency. In this method, after the station finds the line idle it follows these steps:
1. With probability p, the station sends its frame.
2. With probability q = 1 - p, the station waits for the beginning of the next time slot and checks the line again.
1. If the line is idle, it goes to step 1.
2. If the line is busy, it acts as though a collision has occurred and uses the back off procedure.

2.4 CSMA/CD (Collision Detection)
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.
In this method, a station monitors the medium after it sends a frame to see if the transmission was successful. If so, the station is finished. If, however, there is a collision, the frame is sent again.
To better understand CSMA/CD, let us look at the first bits transmitted by the two stations involved in the collision. Although each station continues to send bits in the frame until it detects the collision, we show what happens as the first bits collide. In the following Figure stations A and C are involved in the collision.



• At time t 1, station A has executed its persistence procedure and starts sending the bits of its frame.
• At time t2, station C has not yet sensed the first bit sent by A. Station C executes its persistence procedure and starts sending the bits in its frame, which propagate both to the left and to the right.
• The collision occurs sometime after time t2' Station C detects a collision at time t3 when it receives the first bit of A's frame. Station C immediately (or after a short time, but we assume immediately) aborts transmission.
• Station A detects collision at time t4 when it receives the first bit of C's frame; it also immediately aborts transmission.
• Looking at the figure, we see that A transmits for the duration t4 – t1. C transmits for the duration t3 - t2. The protocol to work, the length of any frame divided by the bit rate in this protocol must be more than either of these durations. At time t4, the transmission of A's frame, though incomplete, is aborted. At time t3, the transmission of C's frame, though incomplete, is aborted.

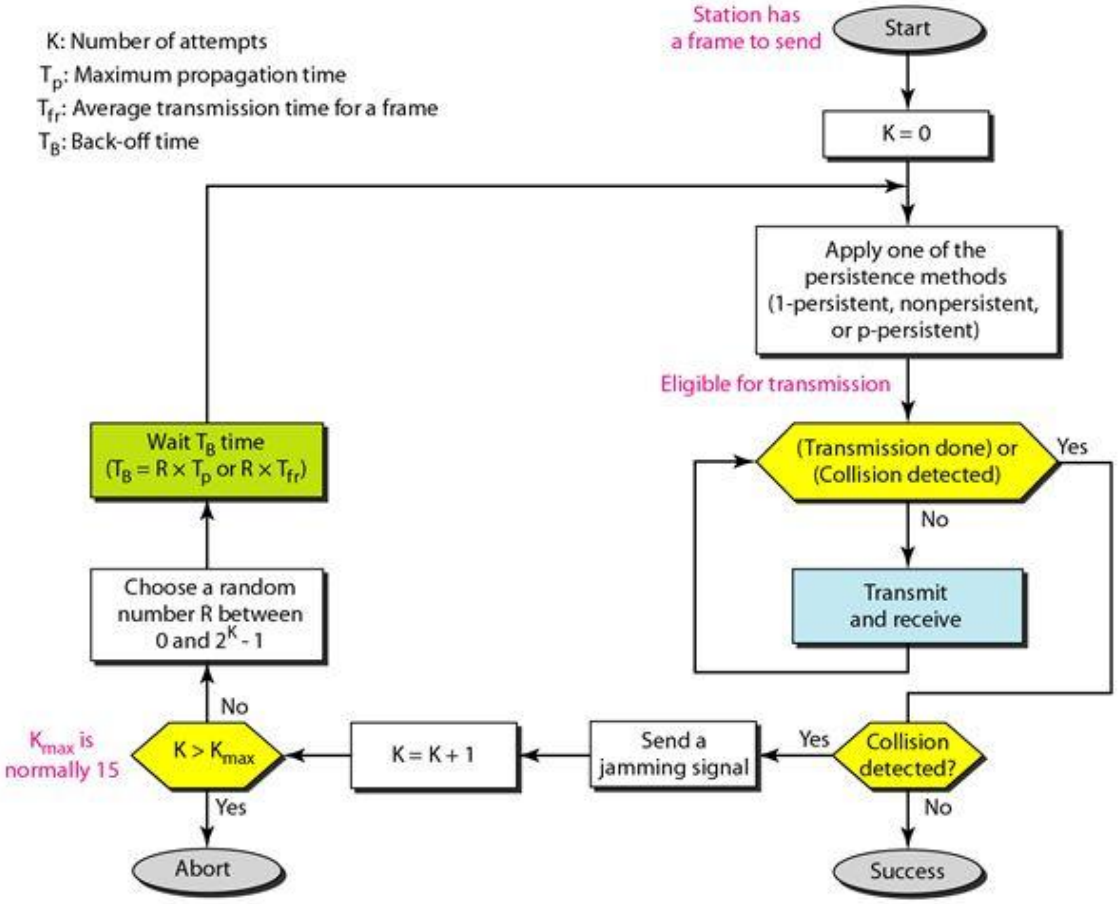
Minimum Frame Size:

For CSMA/CD to work, we need a restriction on the frame size. Before sending the last bit of the frame, the sending station must detect a collision, if any, and abort the transmission. This is so because the station, once the entire frame is sent, does not keep a copy of the frame and does not monitor the line for collision detection. Therefore, the frame transmission time Tfr must be at least two times the maximum propagation time Tp.

To understand the reason, let us think about the worst-case scenario. If the two stations involved in a collision are the maximum distance apart, the signal from the first takes time Tp to reach the second and the effect of the collision takes another time Tp to reach the first. So the requirement is that the first station must still be transmitting after 2Tp .

Procedure

Now let us look at the flow diagram for CSMA/CD in the following figure. It is similar to the one for the ALOHA protocol, but there are differences.



• The first difference is the addition of the persistence process. We need to sense the channel before we start sending the frame by using one of the persistence processes we discussed previously (nonpersistent, I-persistent, or p-persistent).

• The second difference is the frame transmission. In ALOHA, we first transmit the entire frame and then wait for an acknowledgment. In CSMA/CD, transmission and collision detection is a continuous process. We constantly monitor in order to detect one of two conditions: either transmission is finished, or a collision is detected. Either event stops transmission.

Energy Level:

We can say that the level of energy in a channel can have three values: zero, normal, and abnormal. At the zero level, the channel is idle. At the normal level, a station has successfully captured the channel and is sending its frame. At the abnormal level, there is a collision and the level of the energy is twice the normal level. A station that has a frame to send or is sending a frame needs to monitor the energy level to determine if the channel is idle, busy, or in collision mode.

Throughput

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

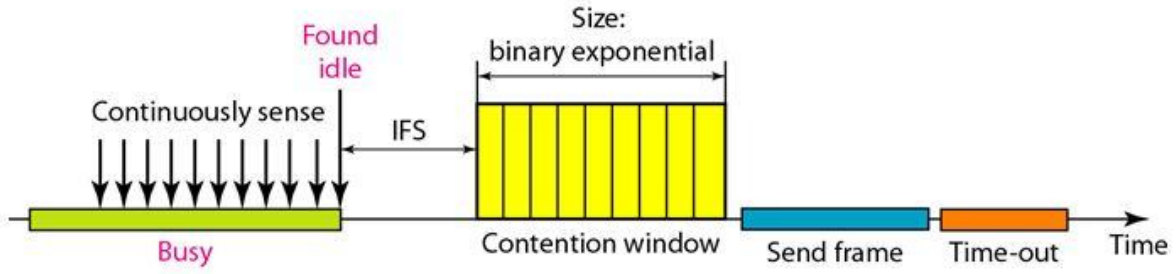
2.5 CSMA/CA (Collision Avoidance)

The basic idea behind CSMA/CD is that a station needs to be able to receive while transmitting to detect a collision. When there is no collision, the station receives one signal: its own signal. When there is a collision, the station receives two signals: its own signal and the signal transmitted by a second station. To distinguish between these two cases, the received signals in these two cases must be significantly different. In other words, the signal from the second station needs to add a significant amount of energy to the one created by the first station.

In a wired network, the received signal has almost the same energy as the sent signal because either the length of the cable is short or there are repeaters that amplify the energy between the sender and the receiver. This means that in a collision, the detected energy almost doubles.

However, in a wireless network, much of the sent energy is lost in transmission. The received signal has very little energy. Therefore, a collision may add only 5 to 10 percent additional energy. This is not useful for effective collision detection.

Carrier sense multiple access with collision avoidance (CSMA/CA) was invented to avoid collisions on wireless networks. Collisions are avoided through the use of CSMAICA's three strategies: the interframe, space, the contention window, and acknowledgments, as shown in the following figure.



Interframe Space (IFS):

First, 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. Even though the channel may appear idle when it is sensed, a distant station may have already started transmitting.

The distant station's signal has not yet reached this station. The IFS time allows the front of the transmitted signal by the distant station to reach this station. If after the IFS time the channel is still idle, the station can send, but it still needs to wait a time equal to the contention time. The IFS

variable can also be used to prioritize stations or frame types. For example, a station that is assigned a shorter IFS has a higher priority.

Contention Window:

The contention window is an amount of time divided into slots. A station that is ready to send chooses a random number of slots as its wait time. The number of slots in the window changes according to the binary exponential back-off strategy. This means that it is set to one slot the first time and then doubles each time the station cannot detect an idle channel after the IFS time. This is very similar to the p-persistent method except that a random outcome defines the number of slots taken by the waiting station.

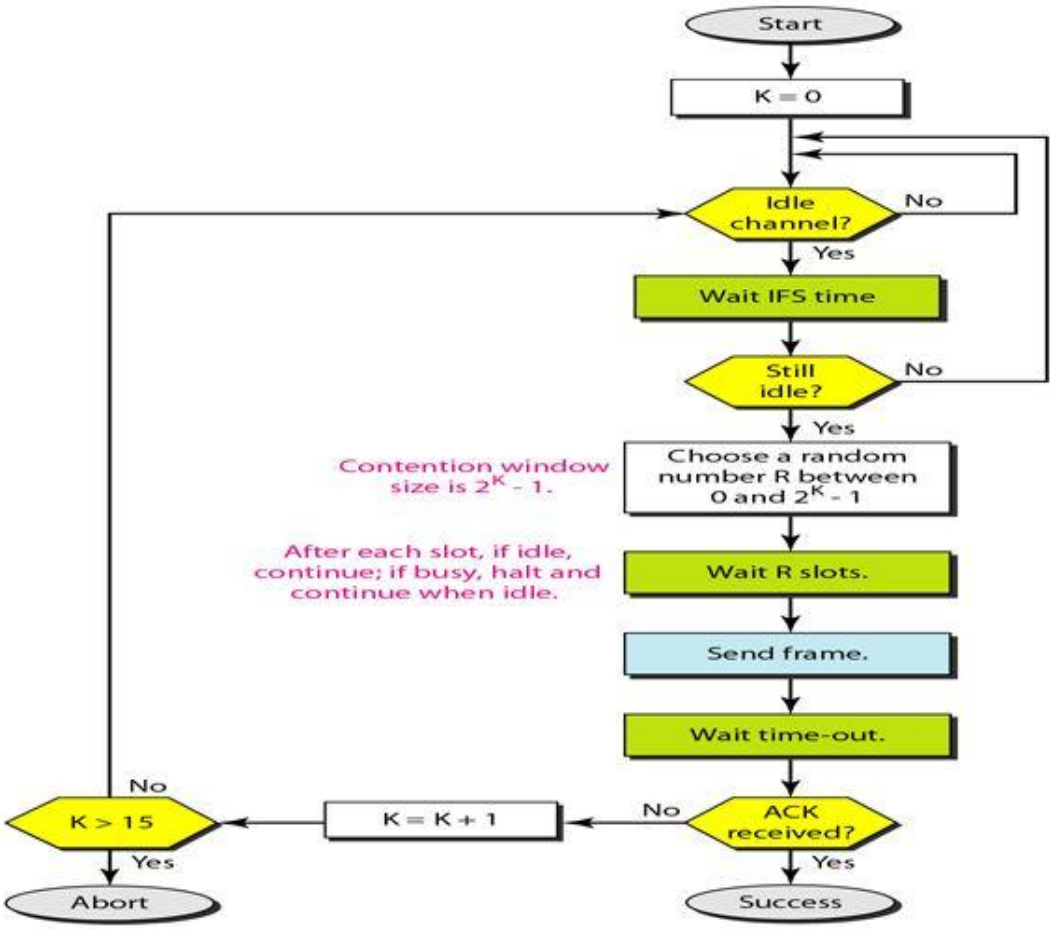
One interesting point about the contention window is that the station needs to sense the channel after each time slot. However, if the station finds the channel busy, it does not restart the process; it just stops the timer and restarts it when the channel is sensed as idle. This gives priority to the station with the longest waiting time.

Acknowledgment

With all these precautions, there still may be a collision resulting in destroyed data. In addition, the data may be corrupted during the transmission. The positive acknowledgment and the time-out timer can help guarantee that the receiver has received the frame.

Procedure

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



CSMA/CA and Wireless Networks

CSMA/CA was mostly intended for use in wireless networks. However, it is not sophisticated enough to handle some particular issues related to wireless networks, such as hidden terminals or exposed terminals.

3. Efficiency Comparison

- **Pure ALOHA**: ~18.4% maximum efficiency.
- **Slotted ALOHA**: ~36.8% maximum efficiency.
- **CSMA**: More efficient than ALOHA; depends on propagation delay.
- **CSMA/CD**: Very efficient for wired LANs; efficiency approaches 90% under low load.
- **CSMA/CA**: Less efficient than CSMA/CD, but necessary for wireless communication.

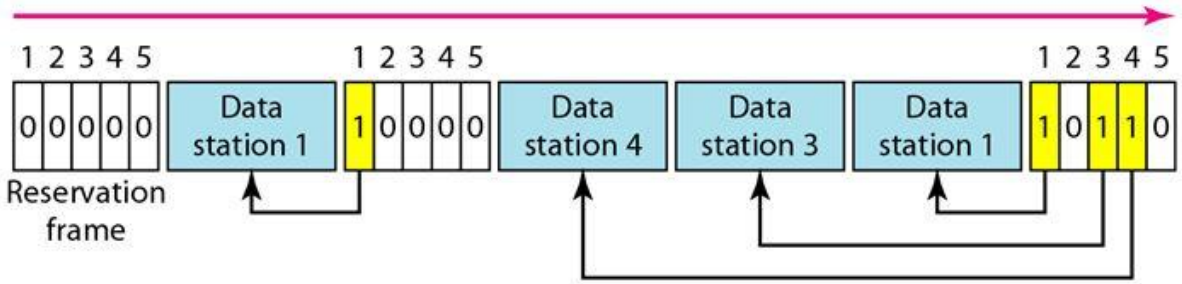
Controlled Access, Channelization, IEEE Standards, and Ethernets

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. The three popular controlled-access methods are as follows.

1. 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.

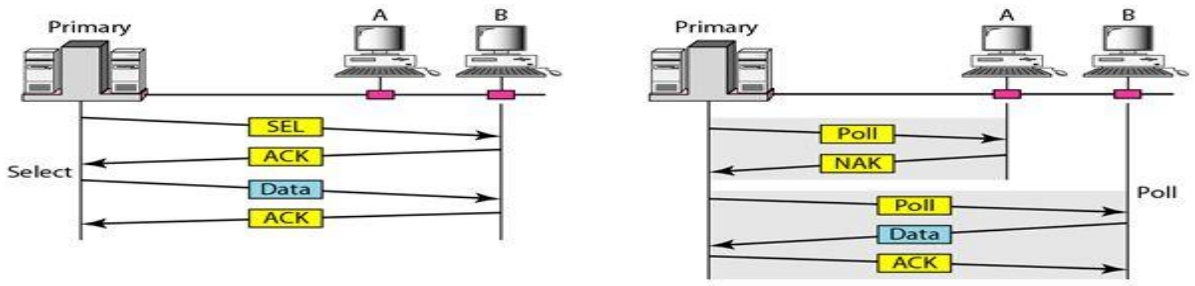
If there are N stations in the system, there are exactly N reservation minislots in the reservation frame. Each minislot belongs to a station. When a station needs to send a data frame, it makes a reservation in its own minislot. The stations that have made reservations can send their data frames after the reservation frame. 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.



2. 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 ultimate destination is a secondary device.

The primary device 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. Consider the following figure.



If the primary wants to receive data, it asks the secondaries 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.

Select:

The select function is used whenever the primary device has something to send. If it has something to send, the primary device sends it. It has to know whether the target device is prepared to receive or not. So the primary must alert the secondary to the upcoming transmission and wait for an acknowledgment of the secondary's ready status. Before sending data, the primary creates and transmits a select (SEL) frame, one field of which includes the address of the intended secondary.

Poll:

The poll function is used by the primary device to solicit transmissions from the secondary devices. When the primary is ready to receive data, it must ask (poll) each device in turn if it has anything to send. When the first secondary is approached, it responds either with a NAK frame if it has nothing to send or with data (in the form of a data frame) if it does. If the response is negative (a NAK frame), then the primary polls the next secondary in the same manner until it finds one with data to send. When the response is positive (a data frame), the primary reads the frame and returns an acknowledgment (ACK frame), verifying its receipt.

3. Token Passing:

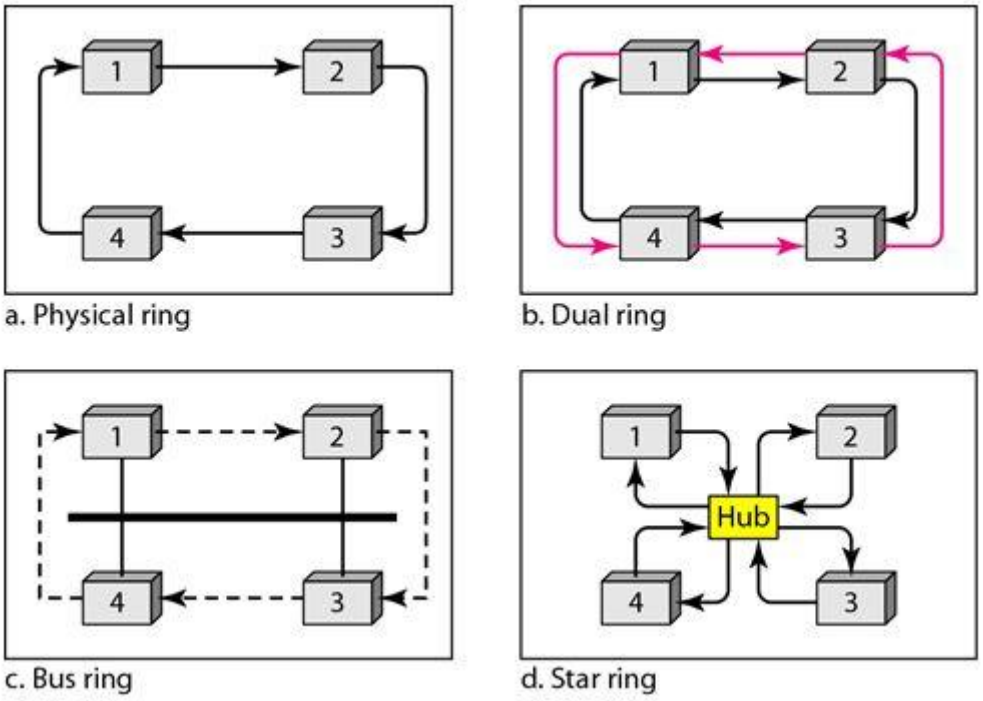
In the token-passing method, the stations in a network are organized in a logical ring. In other words, for each station, there is a predecessor and a successor. The predecessor is the station which is logically before the station in the ring; the successor is the station which is after the station in 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.

In this method, 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.

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. For example, if a station that is holding the token fails, the token will disappear from the network. Another function of token management is to assign priorities to the stations and to the types of data being transmitted. And finally, token management is needed to make low- priority stations release the token to high priority stations.

Logical Ring:

In a token-passing network, stations do not have to be physically connected in a ring; the ring can be a logical one. The following figure show four different physical topologies that can create a logical ring.



a. Physical ring
b. Dual ring
c. Bus ring
d. Star ring

• In the physical ring topology, when a station sends the token to its successor, the token cannot be seen by other stations; the successor is the next one in line. This means that the token does not have to have the address of the next successor. The problem with this topology is that if one of the links-the medium between two adjacent stations fails, the whole system fails.
• The dual ring topology uses a second (auxiliary) ring which operates in the reverse direction compared with the main ring. The second ring is for emergencies only. If one of the links in the main ring fails, the system automatically combines the two rings to form a temporary ring. After the failed link is restored, the auxiliary ring becomes idle again.
• In the bus ring topology, also called a token bus, the stations are connected to a single cable called a bus. They, however, make a logical ring, because each station knows the address of its successor (and also predecessor for token management purposes). When a station has finished sending its data, it releases the token and inserts the address of its successor in the token. Only the station with the address matching the destination address of the token gets the token to access the shared media. The Token Bus LAN, standardized by IEEE, uses this topology.
• In a star ring topology, the physical topology is a star. There is a hub, however, that acts as the connector. The wiring inside the hub makes the ring; the stations are connected to this ring through the two wire connections. This topology makes the network less prone to failure because if a link goes down, it will be bypassed by the hub and the rest of the stations can operate. Also adding and removing stations from the ring is easier. This topology is still used in the Token Ring LAN designed by IBM.
Controlled access is efficient under high load since it avoids the chaos of collisions, but it adds management overhead and can delay low-load transmissions.

2. Channelization
Channelization is a multiple-access method where the available bandwidth of a channel is divided among multiple stations so they can all transmit simultaneously without interference. Unlike random or controlled access, channelization avoids collisions by allocating distinct resources.
a) Frequency-Division Multiple Access (FDMA)

Frequency-Division Multiplexing

Frequency-division multiplexing (FDM) is an analog technique that can be applied when the bandwidth of a link (in hertz) is greater than the combined bandwidths of the signals to be transmitted. In FDM, signals generated by each sending device modulate different carrier frequencies. These modulated signals are then combined into a single composite signal that can be transported by the link. Carrier frequencies are separated by sufficient bandwidth to accommodate the modulated signal. These bandwidth ranges are the channels through which the various signals travel.
Channels can be separated by strips of unused bandwidth-guard bands-to prevent signals from overlapping. The following figure gives a conceptual view of FDM. In this illustration, the transmission path is divided into three parts, each representing a channel that carries one transmission.

Multiplexing Process:
The following figure is a conceptual illustration of the multiplexing process. Each source generates a signal of a similar frequency range. Inside the multiplexer, these similar signals modulates different carrier frequencies (f1, f2 and f3). The resulting modulated signals are then combined into a single composite signal that is sent out over a media link that has enough bandwidth to accommodate it.

Demultiplexing Process:
The demultiplexer uses a series of filters to decompose the multiplexed signal into its constituent component signals. The individual signals are then passed to a demodulator that separates them from their carriers and passes them to the output lines.

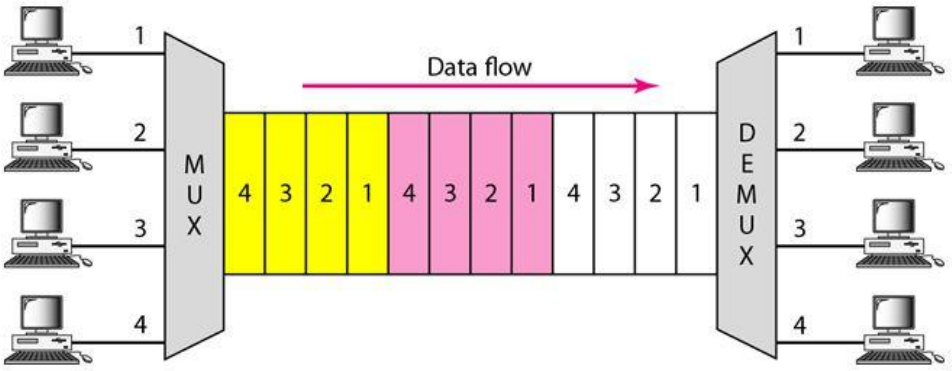
Applications of FDM:

- To maximize the efficiency of their infrastructure, telephone companies have traditionally multiplexed signals from lower-bandwidth lines onto higher-bandwidth lines.
- A very common application of FDM is AM and FM radio broadcasting.
- The first generation of cellular telephones (still in operation) also uses FDM.

Implementation:
FDM can be implemented very easily. In many cases, such as radio and television broadcasting, there is no need for a physical multiplexer or demultiplexer. As long as the stations agree to send their broadcasts to the air using different carrier frequencies, multiplexing is achieved. In other cases, such as the cellular telephone system, a base station needs to assign a carrier frequency to the telephone user. There is not enough bandwidth in a cell to permanently assign a bandwidth range to every telephone user. When a user hangs up, her or his bandwidth is assigned to another caller.

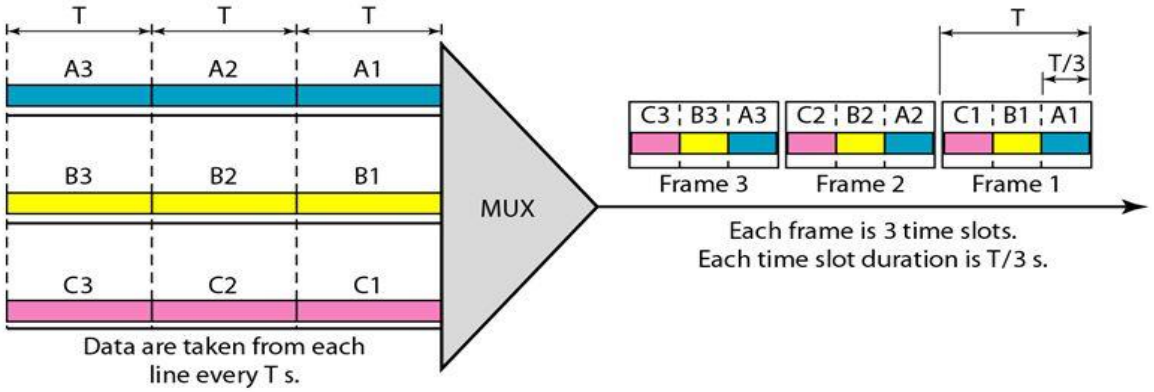
b) Time-Division Multiple Access (TDMA)

Time-division multiplexing (TDM) is a digital process that allows several connections to share the high bandwidth of a linle Instead of sharing a portion of the bandwidth as in FDM, time is shared. Each connection occupies a portion of time in the link.



We can divide TDM into two different schemes: synchronous and statistical.
In synchronous TDM, each input connection has an allotment in the output even if it is not sending data. In synchronous TDM, the data flow of each input connection is divided into units, where each input occupies one input time slot.

A unit can be 1 bit, one character, or one block of data. Each input unit becomes one output unit and occupies one output time slot. However, the duration of an output time slot is n times shorter than the duration of an input time slot. If an input time slot is T s, the output time slot is T/n s, where n is the number of connections. In other words, a unit in the output connection has a shorter duration; it travels faster. The following figure shows an example of synchronous TDM where n is 3.



In synchronous TDM, a round of data units from each input connection is collected into a frame. If we have n connections, a frame is divided into n time slots and one slot is allocated for each unit, one for each input line. If the duration of the input unit is T, the duration of each slot is Tin and the duration of each frame is T.

Time slots are grouped into frames. A frame consists of one complete cycle of time slots, with one slot dedicated to each sending device. In a system with n input lines, each frame has n slots, with each slot allocated to carrying data from a specific input line.

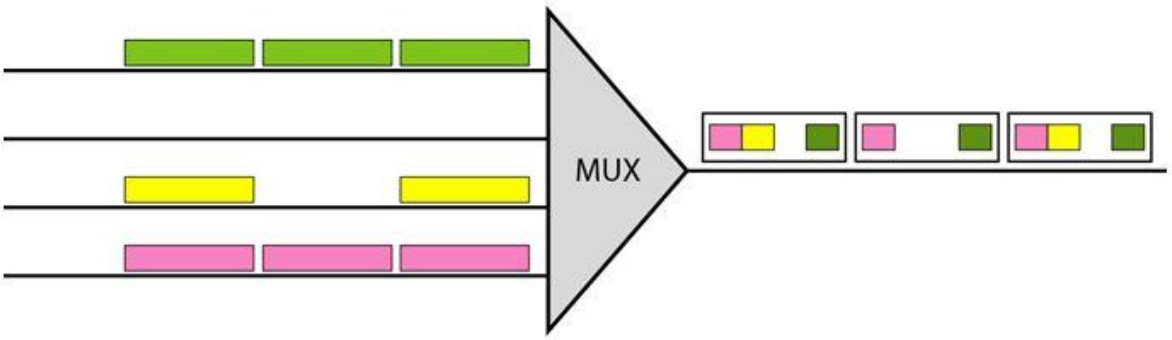
Interleaving

TDM can be visualized as two fast-rotating switches, one on the multiplexing side and the other on the demultiplexing side. The switches are synchronized and rotate at the same speed, but in opposite directions. On the multiplexing side, as the switch opens in front of a connection, that connection has the opportunity to send a unit onto the path. This process is called interleaving. On the demultiplexing

side, as the switch opens in front of a connection, that connection has the opportunity to receive a unit from the path.

Empty Slots

Synchronous TDM is not as efficient as it could be. If a source does not have data to send, the corresponding slot in the output frame is empty. The following figure shows a case in which one of the input lines has no data to send and one slot in another input line has discontinuous data.



The first output frame has three slots filled, the second frame has two slots filled, and the third frame has three slots filled. No frame is full. We learn in the next section that statistical TDM can improve the efficiency by removing the empty slots from the frame.

Data Rate Management

One problem with TDM is how to handle a disparity in the input data rates. If data rates are not the same, three strategies, or a combination of them, can be used. The three different strategies are multilevel multiplexing, multiple-slot allocation, and pulse stuffing.

Multilevel Multiplexing:

Multilevel multiplexing is a technique used when the data rate of an input line is a multiple of others. For example, if we have two inputs of 20 kbps and three inputs of 40 kbps. The first two input lines can be multiplexed together to provide a data rate equal to the last three. A second level of multiplexing can create an output of 160 kbps.

Multiple-Slot Allocation:

Sometimes it is more efficient to allot more than one slot in a frame to a single input line. For example, we might have an input line that has a data rate that is a multiple of another input. The input line with a 50-kbps data rate can be given two slots in the output. We insert a serial-to-parallel converter in the line to make two inputs out of one.

Pulse Stuffing:

Sometimes the bit rates of sources are not multiple integers of each other. Therefore, neither of the above two techniques can be applied. One solution is to make the highest input data rate the dominant data rate and then add dummy bits to the input lines with lower rates. This will increase their rates. This technique is called pulse stuffing, bit padding, or bit stuffing. The input with a data rate of 46 is pulse-stuffed to increase the rate to 50 kbps. Now multiplexing can take place.

Frame Synchronizing

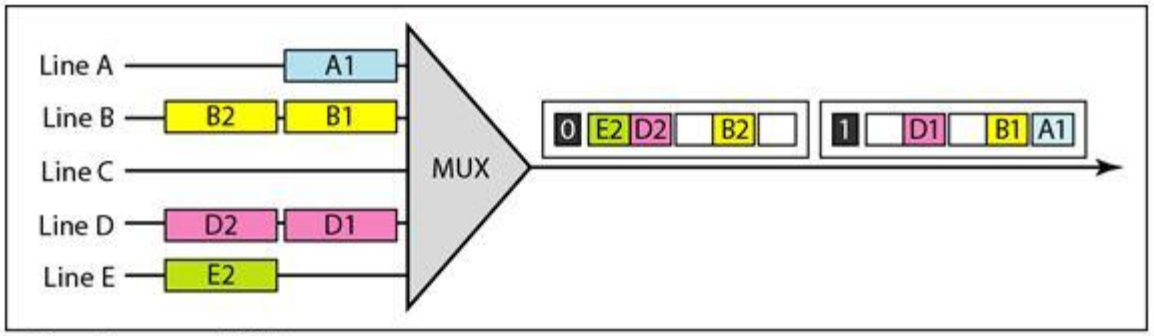
The implementation of TDM is not as simple as that of FDM. Synchronization between the multiplexer and demultiplexer is a major issue. If the, multiplexer and the demultiplexer are not synchronized, a bit belonging to one channel may be received by the wrong channel.

For this reason, one or more synchronization bits are usually added to the beginning of each frame. These bits, called framing bits, follow a pattern, frame to frame, that allows the demultiplexer to

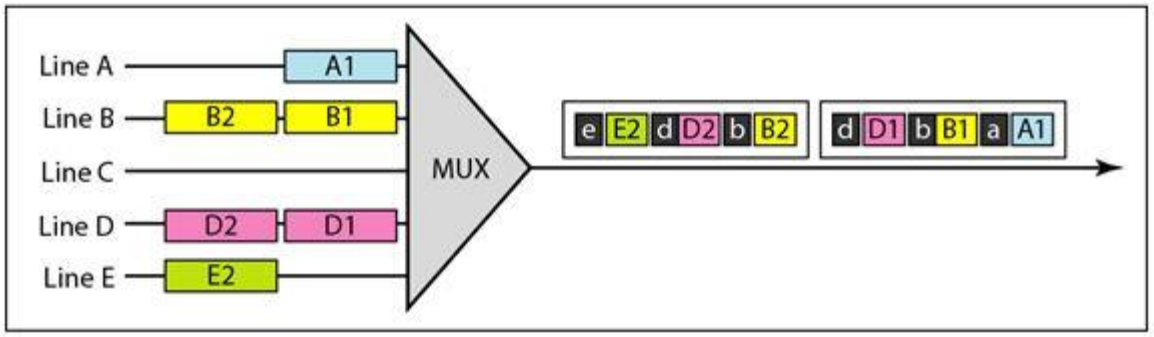
synchronize with the incoming stream so that it can separate the time slots accurately. In most cases, this synchronization information consists of 1 bit per frame, alternating between 0 and 1.

Statistical Time-Division Multiplexing:
In synchronous TDM, each input has a reserved slot in the output frame. This can be inefficient if some input lines have no data to send. In statistical time-division multiplexing, slots are dynamically allocated to improve bandwidth efficiency. Only when an input line has a slot's worth of data to send is it given a slot in the output frame.

In statistical multiplexing, the number of slots in each frame is less than the number of input lines. The multiplexer checks each input line in round robin fashion. It allocates a slot for an input line if the line has data to send otherwise it skips the line and checks the next line.
The following figure shows a synchronous and a statistical TDM example. In the former, some slots are empty because the corresponding line does not have data to send. In the latter, however, no slot is left empty as long as there are data to be sent by any input line.



a. Synchronous TDM



b. Statistical TDM

Addressing:
The above figure also shows a major difference between slots in synchronous TDM and statistical TDM. An output slot in synchronous TDM is totally occupied by data, in statistical TDM, a slot needs to carry data as well as the address of the destination.
In synchronous TDM, there is no need for addressing. Synchronization and preassigned relationships between the inputs and outputs serve as an address. We know, for example, that input 1 always goes to input 1. If the multiplexer and the demultiplexer are synchronized, this is guaranteed. In statistical multiplexing, there is no fixed relationship between the inputs and outputs because there are no preassigned or reserved slots. We need to include the address of the receiver inside each slot to show where it is to be delivered.The addressing in its simplest form can be n bits to define N different output lines with n =log 2 n. For example, for eight different output lines, we need a 3-bit address.

Slot Size

Since a slot carries both data and an address in statistical TDM, the ratio of the data size to address size must be reasonable to make transmission efficient. For example, it would be inefficient to send 1 bit per slot as data when the address is 3 bits. This would mean an overhead of 300 percent. In statistical TDM, a block of data is usually many bytes while the address is just a few bytes. No Synchronization Bit

There is another difference between synchronous and statistical TDM, but this time it is at the frame level. The frames in statistical TDM need not be synchronized, so we do not need synchronization bits.

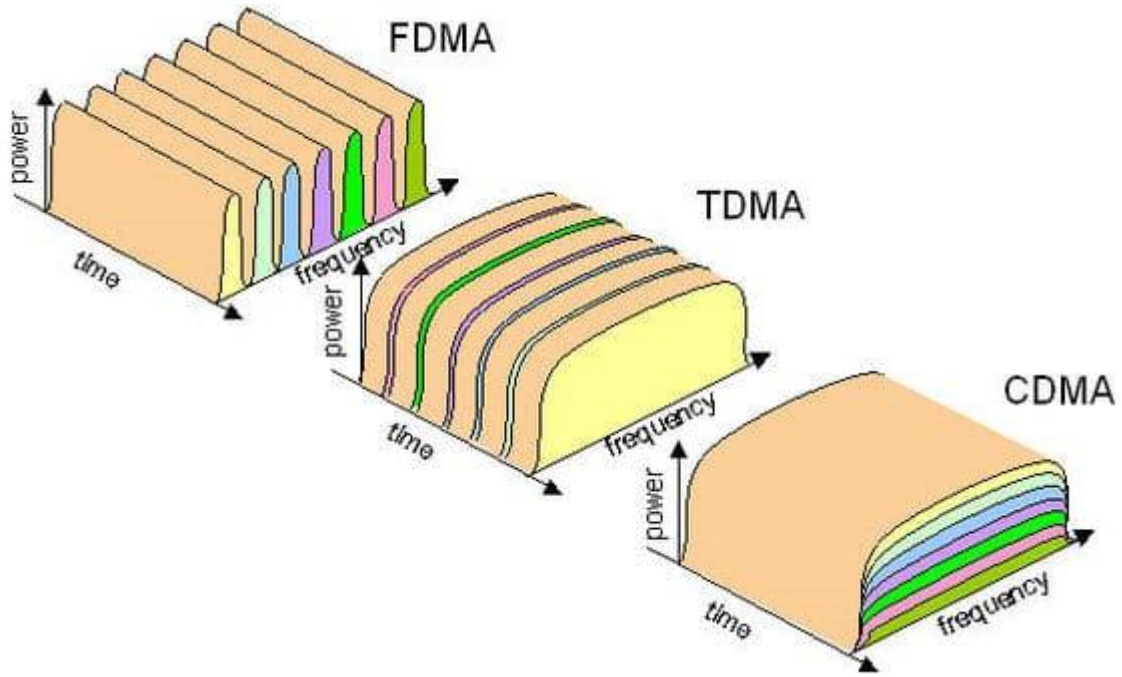
Bandwidth

In statistical TDM, the capacity of the link is normally less than the sum of the capacities of each channel. The designers of statistical TDM define the capacity of the link based on the statistics of the load for each channel. If on average only x percent of the input slots are filled, the capacity of the link reflects this. Of course, during peak times, some slots need to wait.

c) Code-Division Multiple Access (CDMA)

CDMA, or Code Division Multiple Access, is a digital cellular phone technology that uses spread spectrum wireless networking technologies.

CDMA is a digital cellular phone technology that uses spread-spectrum wireless networking technologies. Code Division Multiple Access (CDMA) can be used to refer both to a type of digital cellular phone system and to the specific media access method used by this kind of cellular system. CDMA was developed by Qualcomm in 1993, and it was adopted and ratified by the Telecommunications Industry Association (TIA) as part of their Interim Standard 95.



CDMA – Code Division Multiple Access

How CDMA works

CDMA uses the spread spectrum wireless networking technology—first developed for military communication systems in the 1940s because it spreads its transmission over a large bandwidth, making it difficult to jam. Instead of dividing the available radio spectrum into a series of discrete channels using the older Time Division Multiple Access (TDMA) media access method, a CDMA

channel occupies the entire available frequency band. The disadvantage is that CDMA is more complex to implement than TDMA digital cellular technologies.

The spread spectrum approach assigns a special digital code sequence to each user, and all users share the same broad portion of the radio frequency spectrum. Users thus share time and frequency resources on the available bandwidth, and their individual communications are channeled using these codes. The code tag then identifies the conversation to the transmission station.

All users in a cell that are transmitting at the same time are thus employing the same frequency band for their transmission.

CDMA combines voice and data into a single digital transmission at 9.6 Kbps, although speeds up to 19.2 Kbps per channel are possible by using error detection and correction techniques.

CDMA is a more secure cellular phone technology

Without knowledge of a conversation's code tag, eavesdropping on CDMA conversations is difficult, making CDMA a more secure cellular phone technology than the Advanced Mobile Phone Service (AMPS) still used widely in the United States. CDMA also has a much higher call capacity than AMPS and is comparable to the Global System for Mobile Communications (GSM) standard for cellular communication used in Europe.

3. IEEE Standards

The **IEEE 802 committee** defines networking standards for LANs and MANs. These standards specify physical and data link layer technologies.

- **IEEE 802.3 (Ethernet):** Defines wired LANs using CSMA/CD at the MAC sublayer. Speeds range from 10 Mbps to 400 Gbps in modern Ethernet.
- **IEEE 802.4 (Token Bus):** Defined token passing on a bus topology (rarely used today).
- **IEEE 802.5 (Token Ring):** Defines token passing in a ring topology (IBM networks in the 1980s–1990s).
- **IEEE 802.11 (Wi-Fi):** Defines wireless LANs using CSMA/CA. Popular standards include 802.11n (600 Mbps), 802.11ac (1 Gbps+), and 802.11ax (Wi-Fi 6, multi-gigabit).
- **IEEE 802.15:** Defines wireless personal area networks, including Bluetooth (802.15.1).
- **IEEE 802.16 (WiMAX):** Defines broadband wireless MANs, offering high-speed access.

4. Different Ethernets

Ethernet, defined under IEEE 802.3, has evolved over decades. Initially designed as a shared bus with CSMA/CD, it now uses full-duplex switches and supports very high data rates.

a) Standard Ethernet (10 Mbps)

The earliest Ethernet operated at 10 Mbps using coaxial cable (10Base5, 10Base2) or twisted pair (10Base-T). It used CSMA/CD for medium access.

b) Fast Ethernet (100 Mbps)

Fast Ethernet increased speed to 100 Mbps while maintaining backward compatibility. Common versions included 100Base-TX (twisted pair) and 100Base-FX (fiber). It was widely adopted in the 1990s.

c) Gigabit Ethernet (1 Gbps)

Gigabit Ethernet provided 1 Gbps speed, using twisted pair (1000Base-T) or fiber (1000Base-SX, 1000Base-LX). It quickly became the standard for backbone and enterprise LANs.

d) 10 Gigabit and Beyond

10 Gigabit Ethernet (10G) is widely used in data centers, with fiber-based standards like 10GBase-SR and 10GBase-LR. Modern Ethernet standards now include 40G, 100G, 200G, and even 400G for high-performance networking.

***100 Mbps → 1 Gbps → 10 Gbps → 40/100 Gbps → 400 Gbps**.)*

Controlled access methods (reservation, polling, token passing) avoid collisions by granting orderly transmission rights, while channelization methods (FDMA, TDMA, CDMA) allow simultaneous transmissions by dividing resources. IEEE 802 standards provide the backbone for LANs and MANs, with Ethernet (802.3) and Wi-Fi (802.11) being the most widely deployed. Ethernet itself has continuously evolved from 10 Mbps to 400 Gbps, making it the dominant wired LAN technology worldwide.