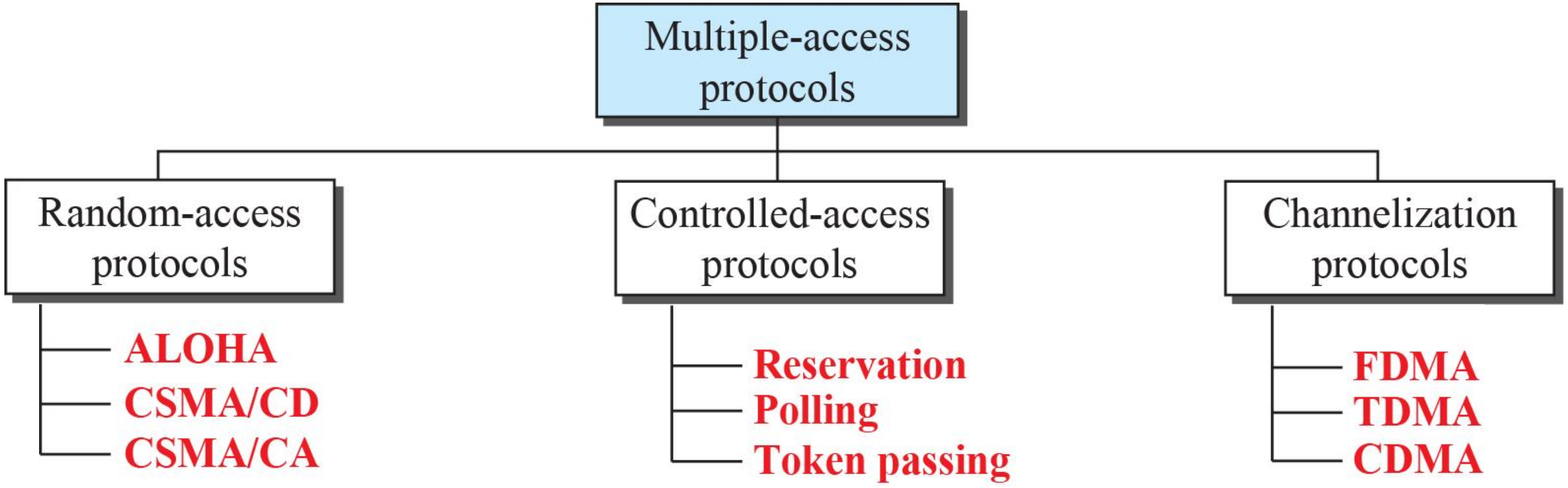
MULTIPLE ACCESS PROTOCOLS

We said that the data-link layer is divided into two sublayers: data link control (DLC) and media access control (MAC). We discussed DLC in the previous section; we talk about MAC in this section.



Random Access

In random-access or contention methods, no station is superior to another station and none is assigned the control over another. At each instance, a station that has data to send uses a procedure defined by the protocol to make a decision on whether or not to send. This decision depends on the state of the medium (idle or busy). In other words, each station can transmit when it desires on the condition that it follows the predefined procedure, including the testing of the state of the medium.

Controlled Access

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

- ***Reservation***
- ***Polling***
  - ***Select***
  - ***Poll***
- ***Token Passing***
  - ***Logical Ring***

Reservation

In the reservation method, a station needs to make a reservation before sending data.

The time line has two kinds of periods:

- Reservation interval of fixed time length
- Data transmission period of variable frames.

If there are M stations, the reservation interval is divided into M slots, and each station has one slot.

Suppose if station 1 has a frame to send, it transmits 1 bit during the slot 1. No other station is allowed to transmit during this slot.

In general, i th station may announce that it has a frame to send by inserting a 1 bit into i th slot. After all N slots have been checked, each station knows which stations wish to transmit.

The stations which have reserved their slots transfer their frames in that order.

After data transmission period, next reservation interval begins.

Since everyone agrees on who goes next, there will never be any collisions.

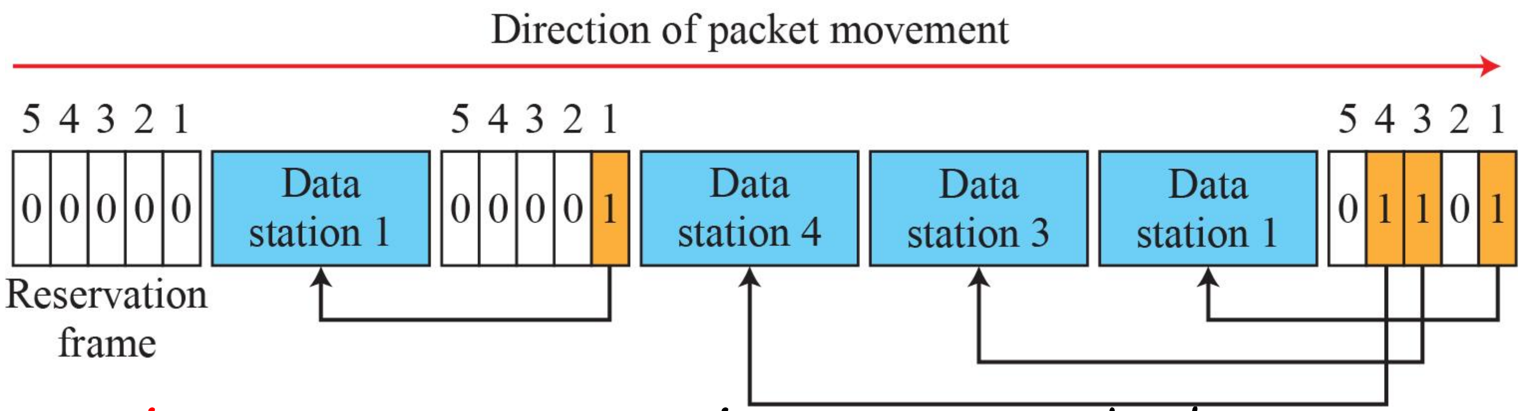


Figure 5.42: Reservation access method

Polling:

Polling process is similar to the roll-call performed in class. Just like the teacher, a controller sends a message to each node in turn.

In this, one acts as a primary station(controller) and the others are secondary stations. All data exchanges must be made through the controller.

The message sent by the controller contains the address of the node being selected for granting access.

Although all nodes receive the message but the addressed one responds to it and sends data, if any. If there is no data, usually a “poll reject”(NAK) message is sent back.

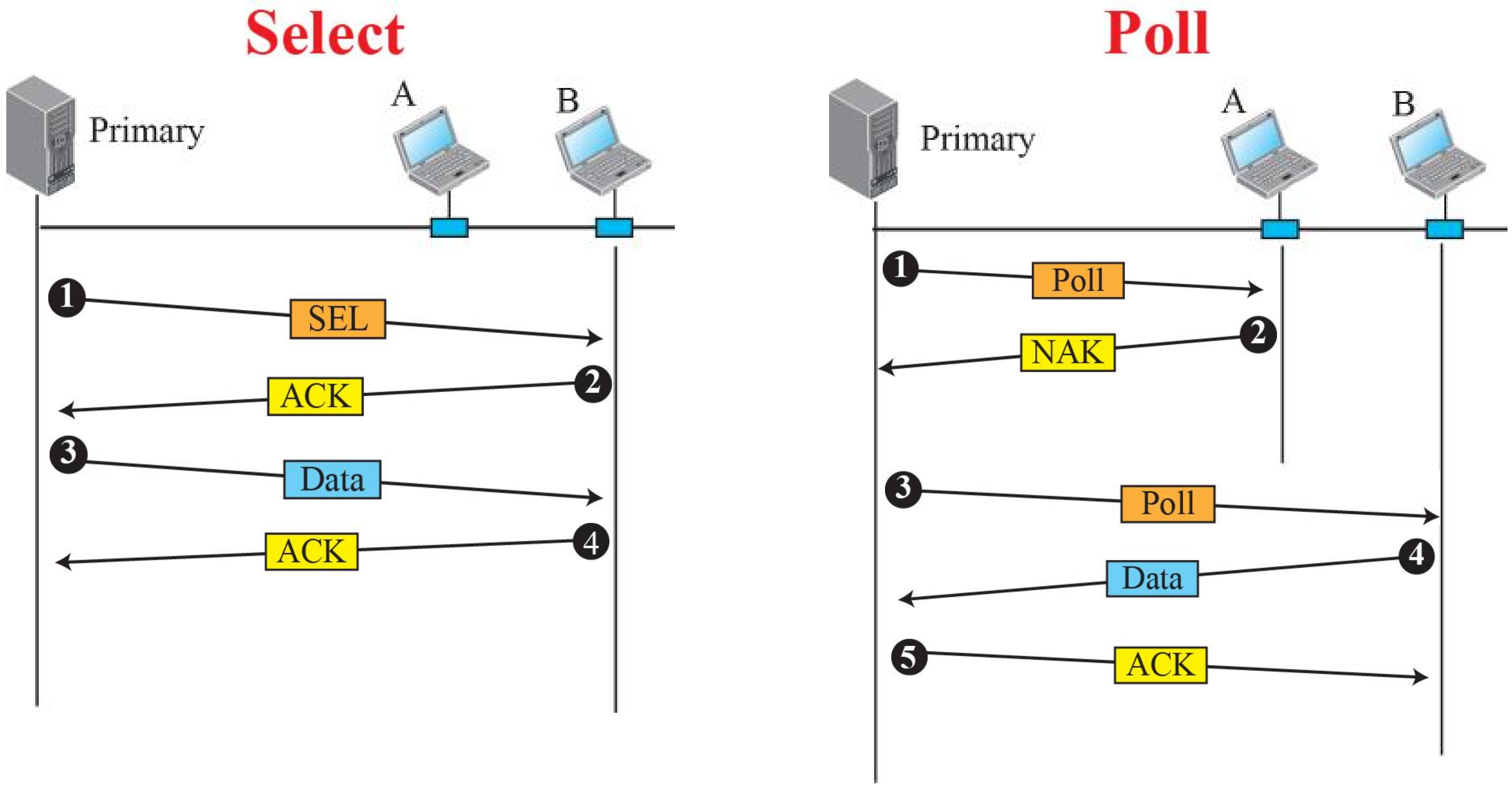
Problems include high overhead of the polling messages and high dependence on the reliability of the controller.

Efficiency:

Let Tpoll be the time for polling and Tt be the time required for transmission of data. Then,

Efficiency = Tt/(Tt + Tpoll)

Figure 5.43: Select and poll functions in polling-access method



Token Passing:

In token passing scheme, the stations are connected logically to each other in form of ring and access of stations is governed by tokens.
A token is a special bit pattern or a small message, which circulate from one station to the next in the some predefined order.
In Token ring, token is passed from one station to another adjacent station in the ring whereas incase of Token bus, each station uses the bus to send the token to the next station in some predefined order.
In both cases, token represents permission to send. If a station has a frame queued for transmission when it receives the token, it can send that frame before it passes the token to the next station. If it has no queued frame, it passes the token simply.
After sending a frame, each station must wait for all N stations (including itself) to send the token to their neighbors and the other N – 1 stations to send a frame, if they have one.
There exists problems like duplication of token or token is lost or insertion of new station, removal of a station, which need be tackled for correct and reliable operation of this scheme.

Performance:

Performance of token ring can be concluded by 2 parameters:-

- Delay, which is a measure of time between when a packet is ready and when it is delivered.So, the average time (delay) required to send a token to the next station = a/N.
- Throughput, which is a measure of the successful traffic.

Throughput, $S = 1/(1 + a/N)$ for $a<1$

and

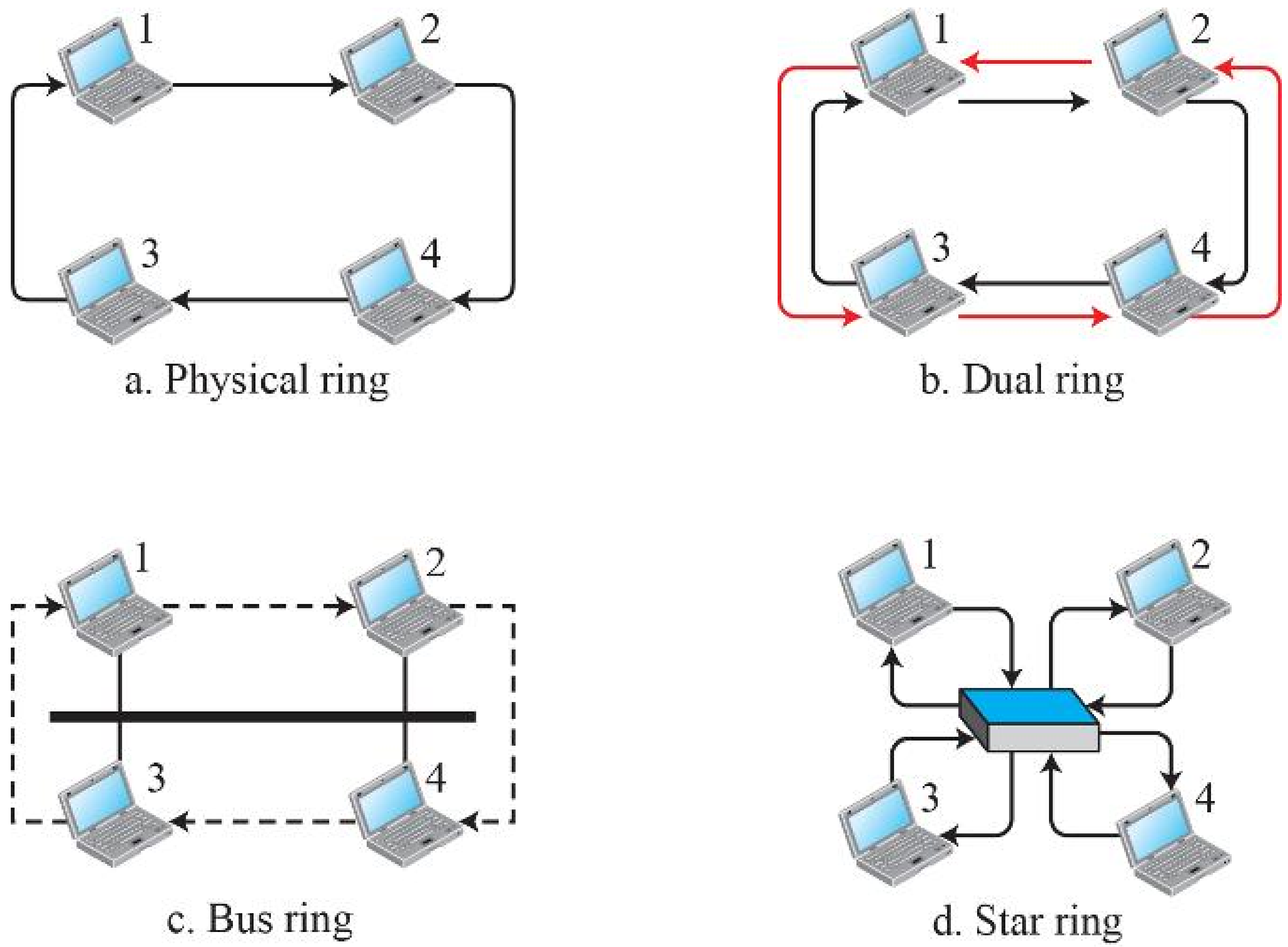
$S = 1/\{a(1 + 1/N)\}$ for $a>1$.

where N = number of stations

$a = Tp/Tt$

(Tp = propagation delay and Tt = transmission delay)

Figure 5.44: Logical ring and physical topology in token-passing access method



Channelization

Channelization (or channel partition, as it is sometimes called) is a multiple-access method in which the available bandwidth of a link is shared in time, frequency, or through code, among different stations.

- ***Frequency Division Multiple Access (FDMA) –*** The available bandwidth is divided into equal bands so that each station can be allocated its own band. Guard bands are also added so that no to bands overlap to avoid crosstalk and noise.
- ***Time Division Multiple Access (TDMA) –*** In this, the bandwidth is shared between multiple stations. To avoid collision time is divided into slots and stations are allotted these slots to transmit data. However there is a overhead of synchronization as each station needs to know its time slot. This is resolved by adding synchronization bits to each slot. Another issue with TDMA is propagation delay which is resolved by addition of guard bands.
- ***Code Division Multiple Access (CDMA) –*** One channel carries all transmissions simultaneously. There is neither division of bandwidth nor division of time. For example, if there are many people in a room all speaking at the same time, then also perfect reception of data is possible if only two person speak the same language. Similarly data from different stations can be transmitted simultaneously in different code languages.

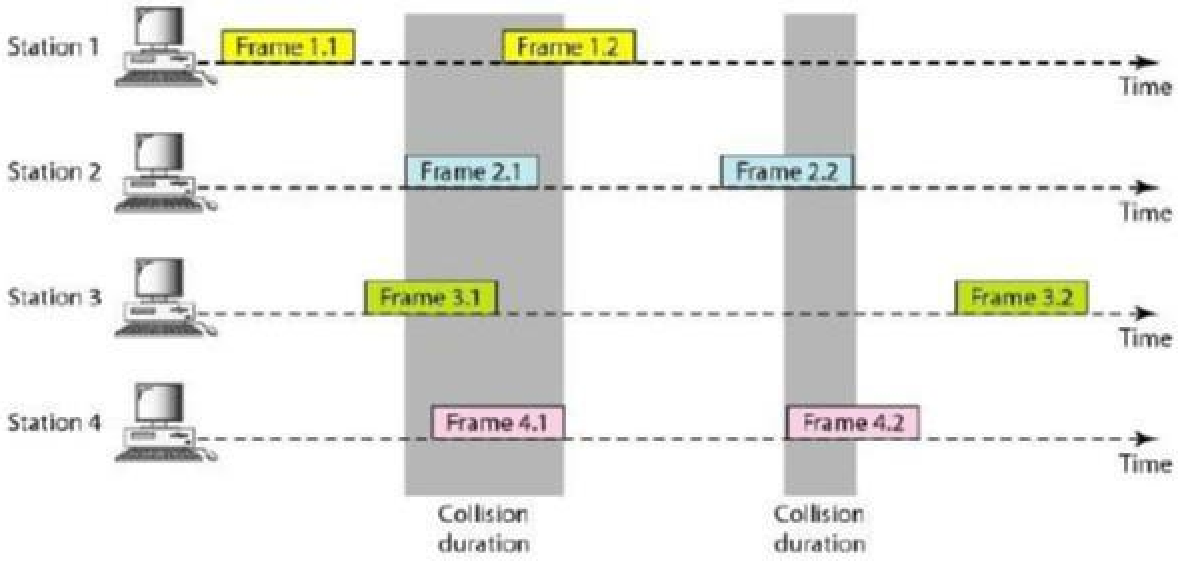
Random Access Protocol

- Why the name *random access*?
  - There is no scheduled time for a station to transmit.
  - Transmission is random among the stations.
  - No station is superior to another station and none is assigned the controlled over another.
- In a random access method, each station has the right to the medium without being controlled by any other station. However, if more than one station tries to send, there is an access conflict-collision-and the frames will be either destroyed or modified.

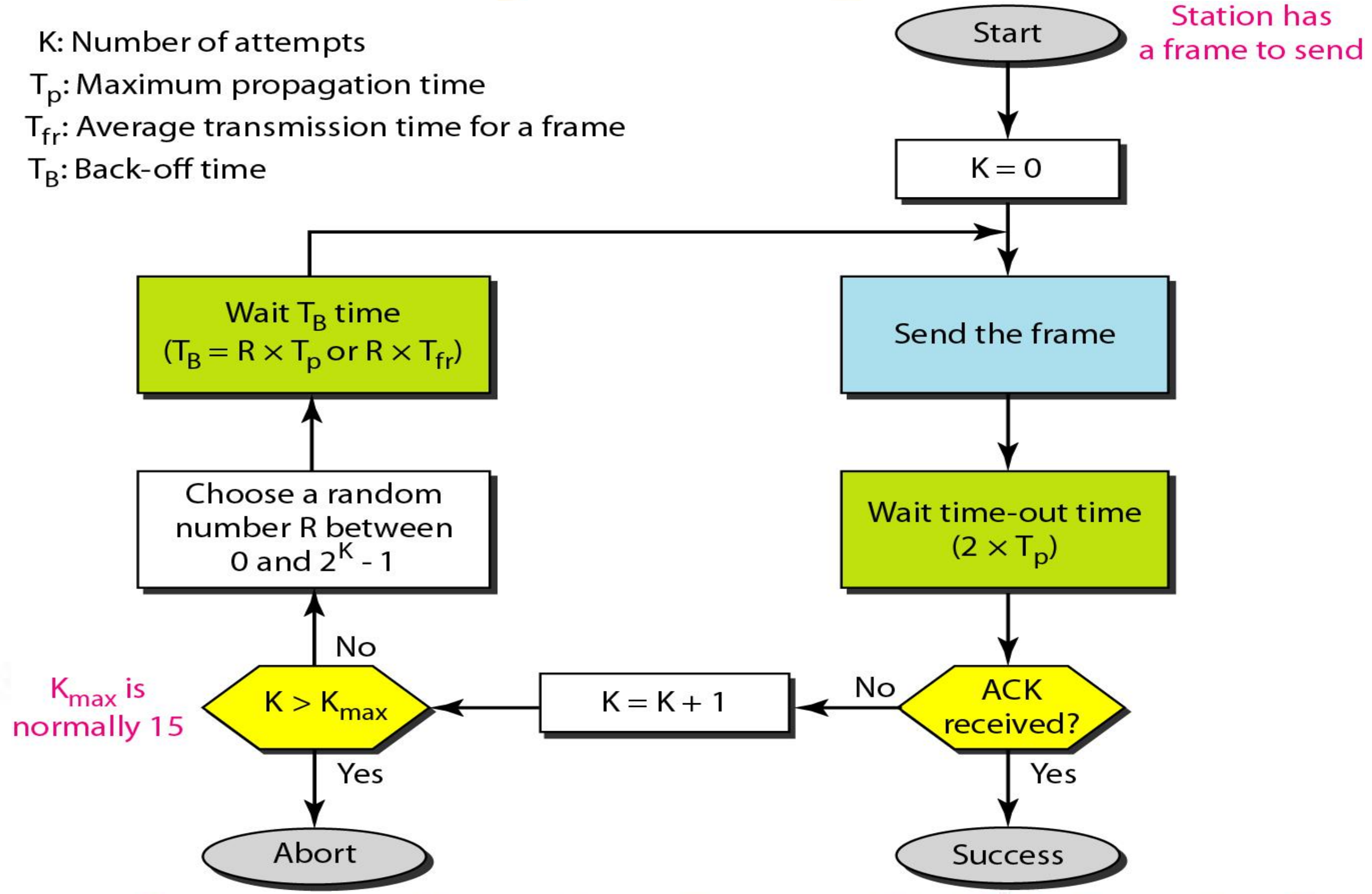
ALOHA: Pure ALOHA

- Based upon the simplest solution:
  - A station transmits whenever it has data to transmit.
  - If more than one frames are transmitted, they interfere with each other (collide) and are lost.
  - If ACK not received within timeout, then a station picks **random back-off time** (to avoid repeated collision).
  - Station retransmits frame after back-off time denoted as T_B
- **Note:** 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.
- After a maximum number of retransmission attempts K_{max} a station must give up and try later.

Four stations transmitting 2 frames each.
Out of all the frames, only two frames survive: frame 1.1 and frame 3.2



Procedure for pure ALOHA protocol



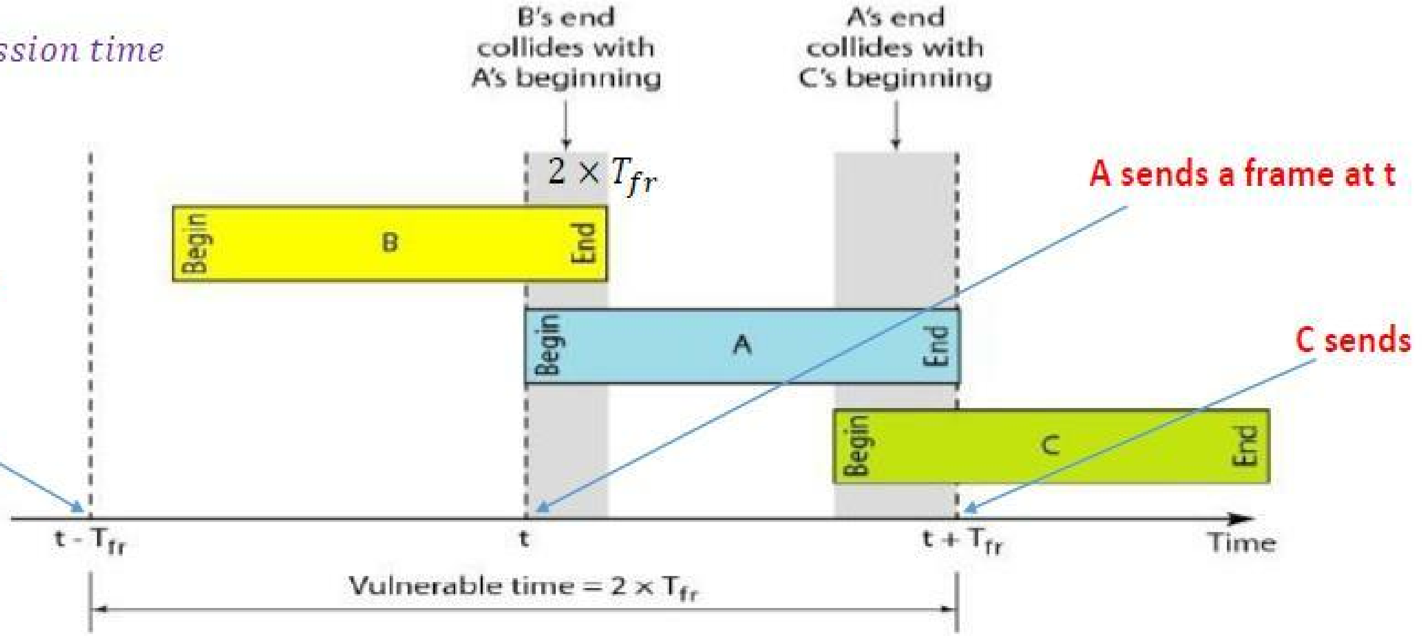
Note: R is a random number chosen from the range 0 to $2^k - 1$, and value of the random number increases after each collision.

Example 1: The stations on a wireless ALOHA network are a maximum of 600 km apart. If we assume that signals propagate at $3 \times 10^8\ m/s$, we find T_P = (600 x 10^3) / (3 x 10^8) = 2 ms. Now we can find the value of T_B for different values of *K*.

a) *For K=1, the range of R is {0, 1}. The station needs to generate a random number with value 0 or 1. So, T_B is either 0 or 2ms, based on outcome of the random variable.*
b) *For K=2, the range of R is {0, 1, 2, 3}. So, T_B can be 0, 2, 4 or 6ms, based on outcome of the random variable.*
c) *For K=3, the range of R is {0, 1, 2, 3,..., 7}. So, T_B is can be 0, 2, 4, 6, 8, 10, 12, or 14ms, based on outcome of the random variable.*
d) *So on………………*
e) *We need to mention that if k >10, it is normally set to 10.*

Vulnerable time: It is the time duration , in which there is a possibility of collision. $\boxed{\textbf{Vulnerable time in pure ALOHA} = 2 \times T_{fr}}$

$T_{fr} = frame\ transmission\ time$



Example 2: A pure ALOHA network transmits 200-bit frames on a shared channel of 200 kbps. What is the requirement to make this frame collision-free?

Sol:

Average frame transmission time T_{fr} *is 200 bits/200 kbps or 1 ms. The vulnerable time is 2 x1 ms =2 ms.*

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

Pure ALOHA Model

- Definitions and assumptions
  - T_{fr} frame transmission time (assume constant)
  - S: throughput (average # successful frame transmissions per T_{fr} seconds)
  - G: load (average # transmission attempts per T_{fr} sec.)
  - $P_{success}$: probability a frame transmission is successful

Note: Any transmission that begins during vulnerable period leads to collision. Success if and only if no arrivals during 2 T_{fr} seconds.

Throughput is given by,

$$\boxed{S = GP_{success}}$$

Abramson's assumption for calculation of $P_{Success}$

- *What is probability of no arrivals in vulnerable period?*

- **Abramson's assumption:** Effect of back-off algorithm is that frame arrivals are equally likely to occur at any time interval.
- G is avg. # arrivals per T_{fr} seconds
- Divide T_{fr} into n intervals of duration $\Delta = T_{fr}/n$
- p = probability of arrival in Δ interval, then

$G = n\,p$ since there are n intervals in T_{fr} seconds

$$P_{success} = P[0 \text{ arrivals in } 2T_{fr} \text{ seconds}] =$$

$$= P[0 \text{ arrivals in 2n intervals}] \qquad \ldots\ldots\ldots\ldots \textbf{Abramson's assumption:}$$

$$= (1-P)^{2n} = \left(1-\frac{G}{n}\right)^{2n} \quad \to e^{-2G} \quad \text{as } n \to \infty$$

Throughput of ALOHA

$$S = GP_{success} = Ge^{-2G}$$

Bimodal behavior:

Small G, S≈G

Large G, S↓0

Example 3: A pure ALOHA network transmits 200-bit frames on a shared channel of 200 kbps. What is the throughput if the system (all stations together) produces

a. 1000 frames per second

b. 500 frames per second

c. 250 frames per second

Sol: *Here,* T_{fr} *is 200 bits/200 kbps or 1 ms.*

a. If the system creates 1000 frames per second, this is 1 frame per millisecond. The load is 1. In this case S = $G \times e^{-2G}$ or S =0.135 (13.5 percent). This means that the throughput is 1000 X 0.135 =135 frames. Only 135 frames out of 1000 will probably survive.

b. If the system creates 500 frames per second, this is (1/2) frame per millisecond. The load is *(1/2)*. In this case S = $G \times e^{-2G}$ or S = 0.184 (18.4 percent). This means that the throughput is 500 x 0.184 = 92 and that only 92 frames out of 500 will probably survive. Note that this is the maximum throughput case, percentagewise.

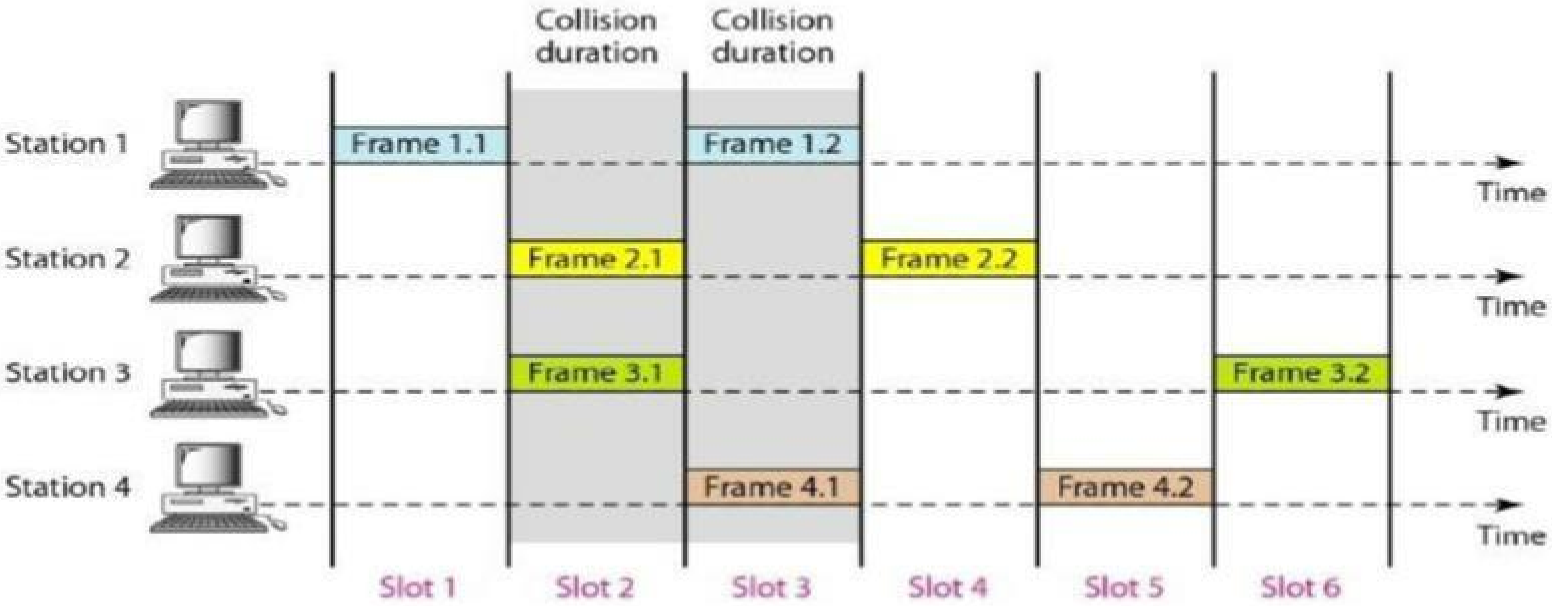
c. If the system creates 250 frames per second, this is (1/4) frame per millisecond. The load is (1/4). In this case S = $G \times e^{-2G}$ or S =0.152 (15.2 percent). This means that the throughput is 250 x 0.152 = 38. Only 38 frames out of 250 will probably survive.

Slotted ALOHA

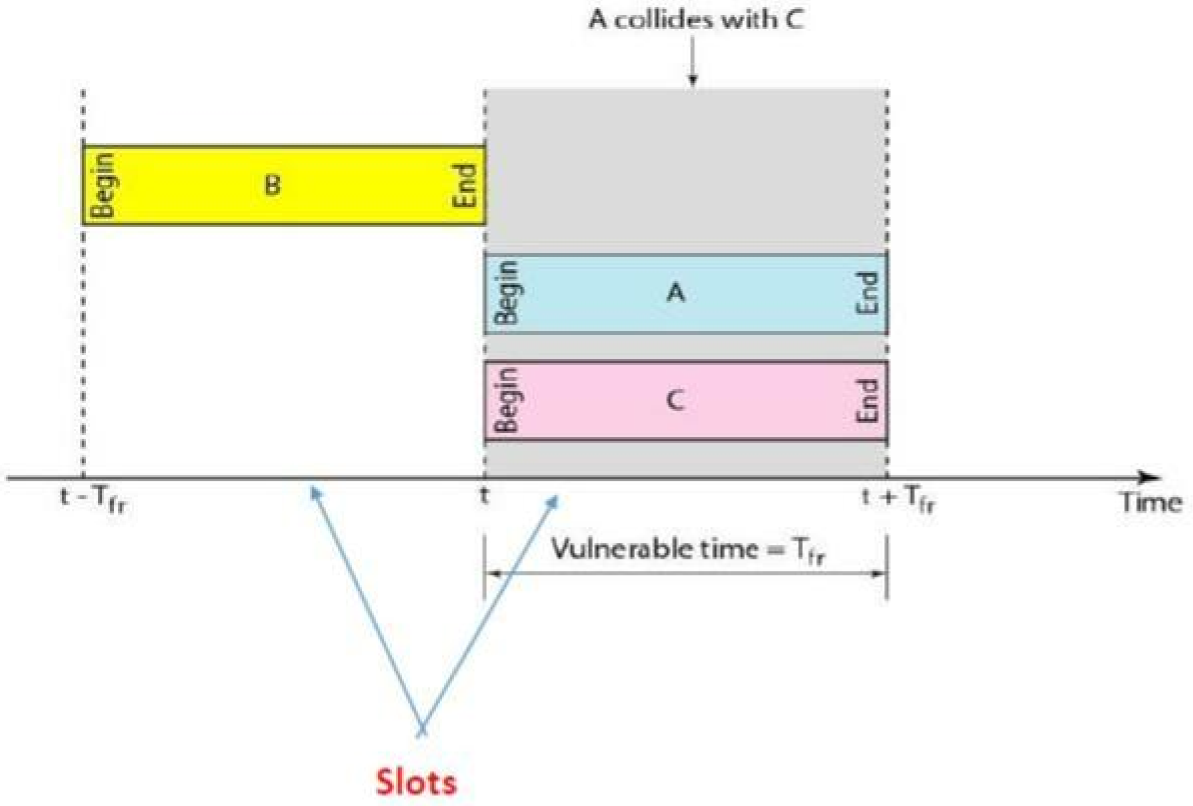
Time is slotted in T_{fr} seconds slots Stations synchronized to frame times

Stations transmit frames in first slot after frame arrival

Backoff intervals are in multiples of slots



Vulnerable time for slotted aloha



Throughput of Slotted ALOHA

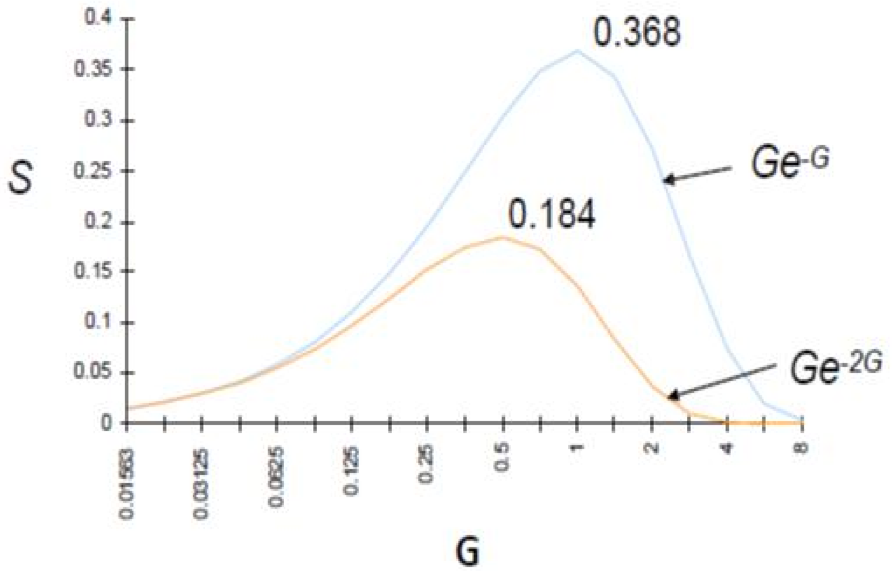
$$P_{success} = P[0 \text{ arrivals in } T_{fr} \text{ seconds}]$$

$$= P[0 \text{ arrivals in n intervals}]$$ …**Abramson's assumption**

$$= (1-P)^n = \left(1 - \frac{G}{n}\right)^n$$

$$\to e^{-G}$$ ………………….as $n \to \infty$

$$\boxed{\therefore S = GP_{Success} = Ge^{-G}}$$



Limitations of ALOHA

The throughput for pure ALOHA is S =G x *e-2G*.
The maximum throughput S_{max} = 0.184 when G =(1/2).

The throughput for slotted ALOHA is S = G x e-G.
The maximum throughput S_{max} = 0.368 when G =1.

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

- In this method, a station **listens while transmitting (LWT)** to see if the transmission was successful.
- Possible cases when channel is idle:
    1. Packet is transmitted in case of non-persistent or 1-persistent
    2. For p-persistent, the packet is sent with probability p or delayed by end-to-end propagation delay (T_P) with probability (1-p).
- If channel is busy, possible cases:
    1. For non-persistent, the packet is **backed off** and the algorithm is repeated.
    2. For 1-persistent, the station **differs** the transmission until the channel is sensed idle, and then immediately transmits when channel becomes idle.
    3. For p-persistent, the station **differs** until the channel is idle, then follow the channel idle procedure (as used in p-persistent CSMA case).

Persistence methods

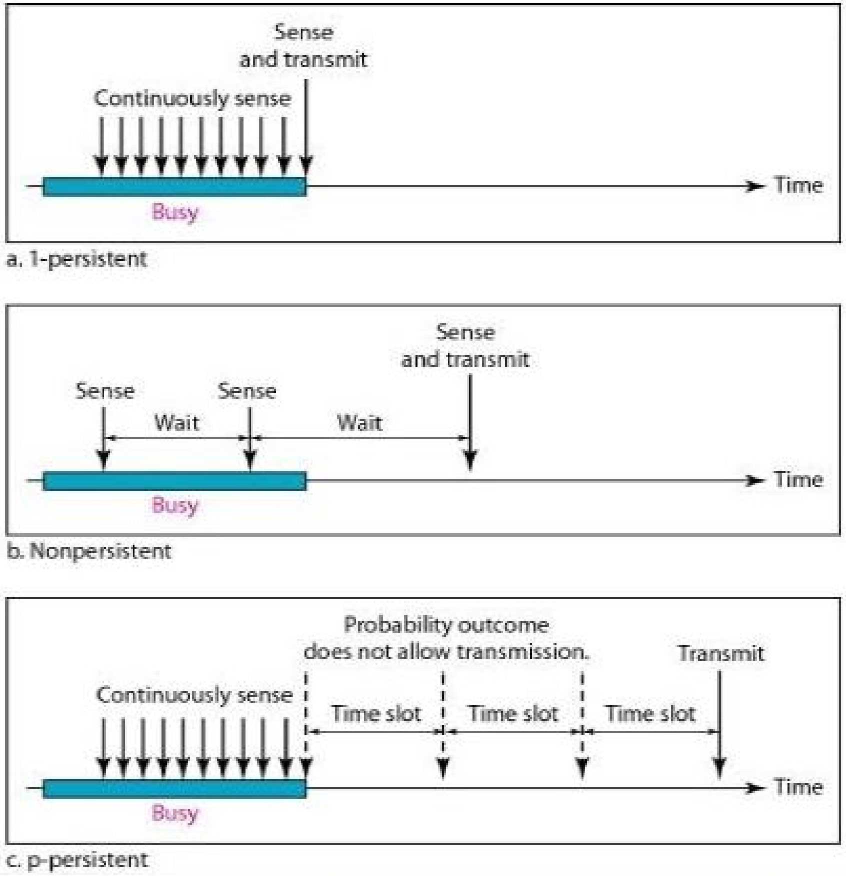


Fig: Behaviour of three persistence methods

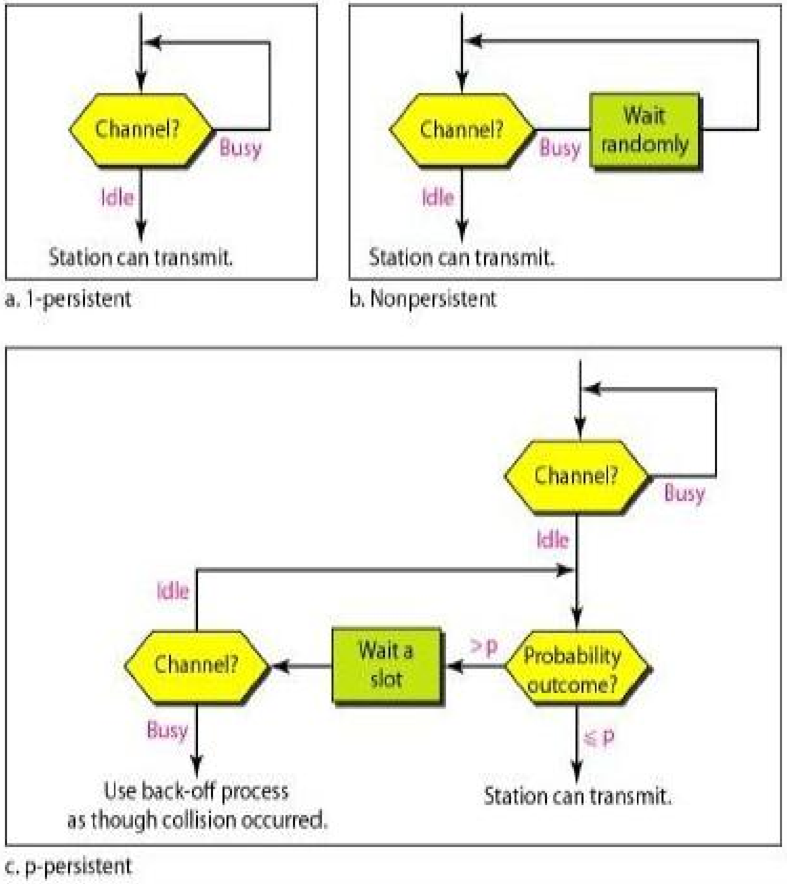
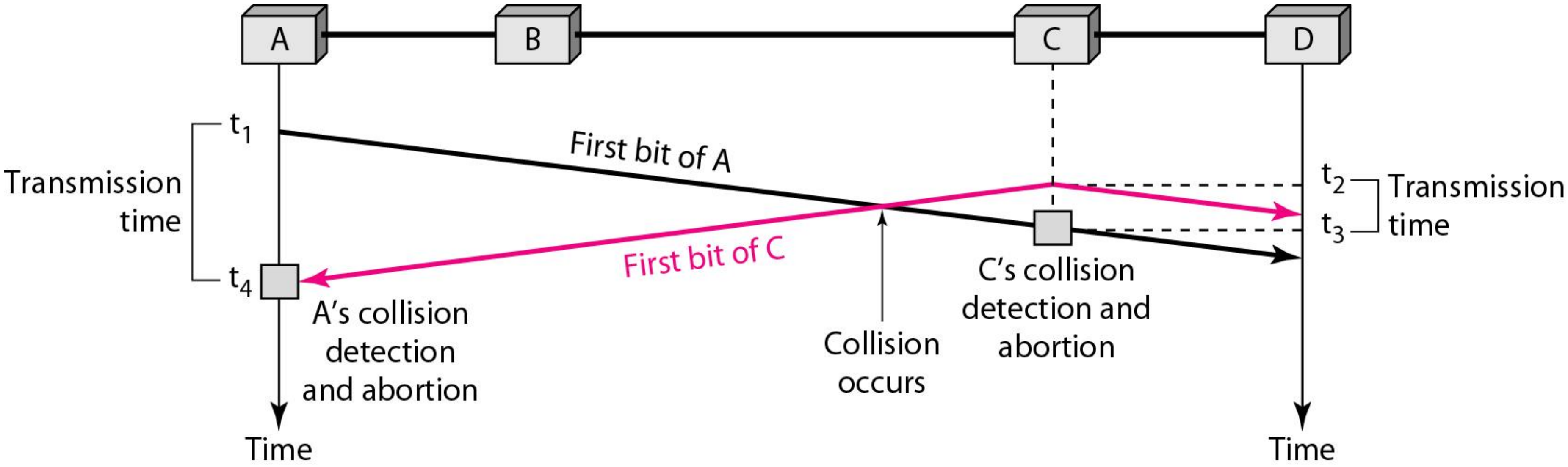


Fig: Flow diagram of three persistence methods

Collision of the first bit in CSMA/CD:



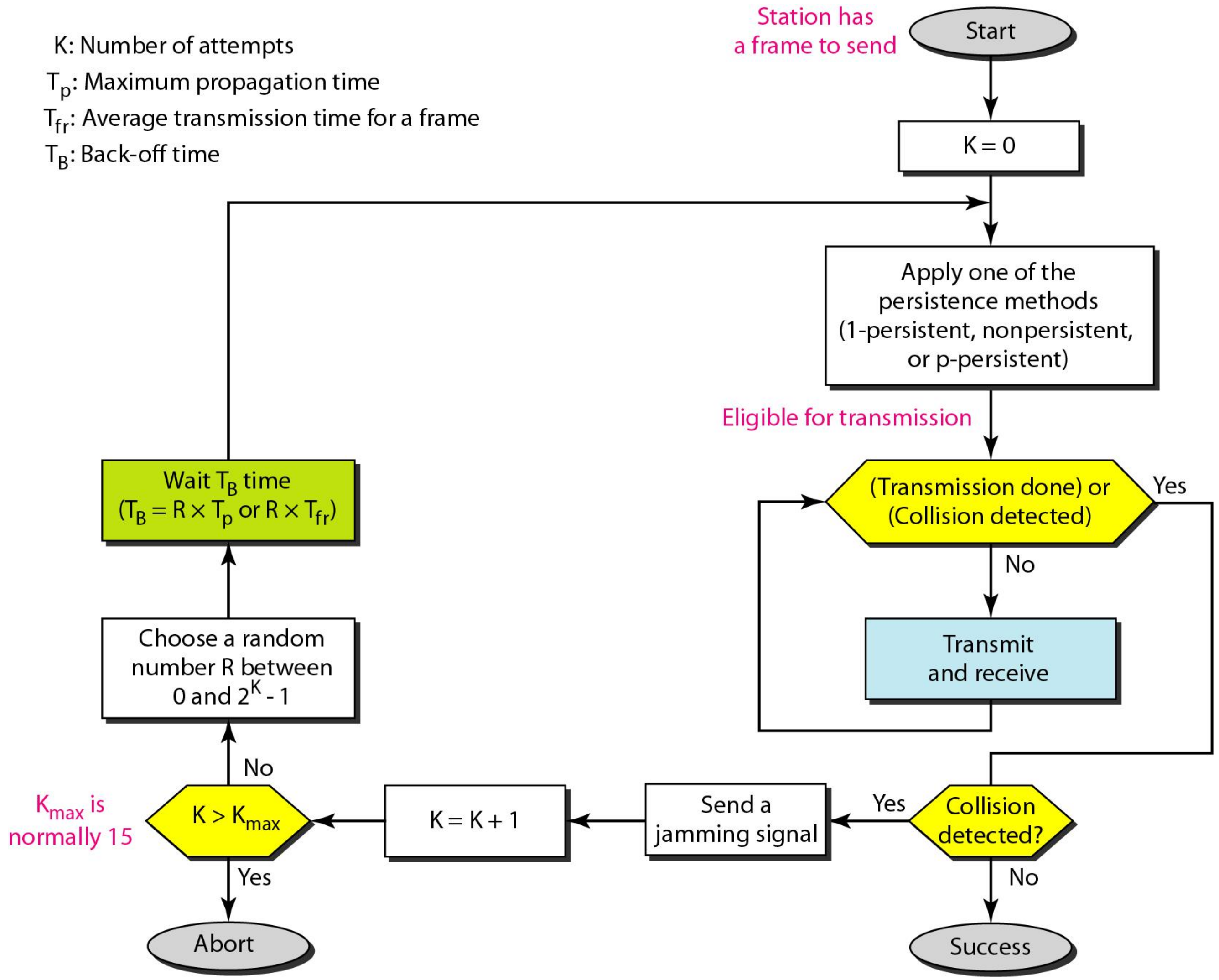
Restrictions in CSMA/CD

- Frame/Packet **transmission time** should be **at least** as long as the time needed to detect a collision + time needed to send a jamming signal (i.e., 2 * maximum propagation delay + *jam sequence* transmission time).
- Otherwise, CSMA/CD does not have an advantage over CSMA

$$\boxed{\therefore T_{fr} \geq 2T_P + jam\ sequence \text{ transmission time}}$$

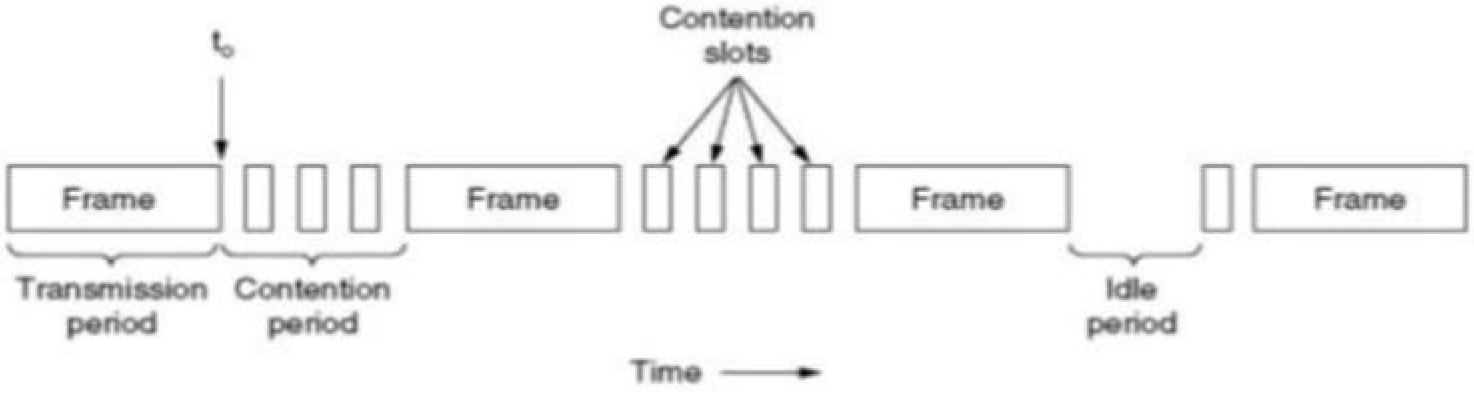
Note: The station which detects collision, sends a jamming signal to all other stations to inform that a collision has detected. So, please stop transmission.

Flow diagram for the CSMA/CD:



The three periods in CSMA/CD

- The following three periods exists in CSMA/CD:
  1. **Contention period:** During this period, each station continuously senses (content) the channel to send a packet.
  2. **Transmission period:** During this, a station has found channel idle and sent a packet over it.
  3. **Idle:** No packet during this period; possible in case of light traffic channel.

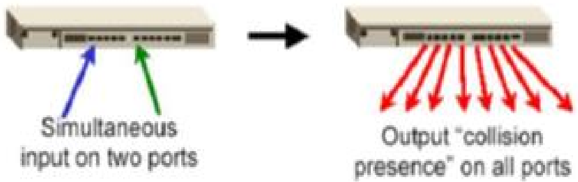


Q.) Collision detection is not possible/difficult in a wireless environment, why?

Ans: *When there is a collision, the station receives two signals: its own signal and a signal transmitted by the second station. To distinguish between, these two signals, they must differ by a significant amount of energy.*
In wired network, the received signal has almost same energy as sent signal. This is because, either the length of cable is short or repeaters are used. It means, in collision, the detected energy is almost double as shown below.

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

Not only this but also there exists the ***hidden station*** *and* ***exposed station*** *problem in wireless environment.*
Note: *In case of hub, if input occurs simultaneously on two ports, it indicates a collision. In this case Hub sends a collision presence signal on all ports.*

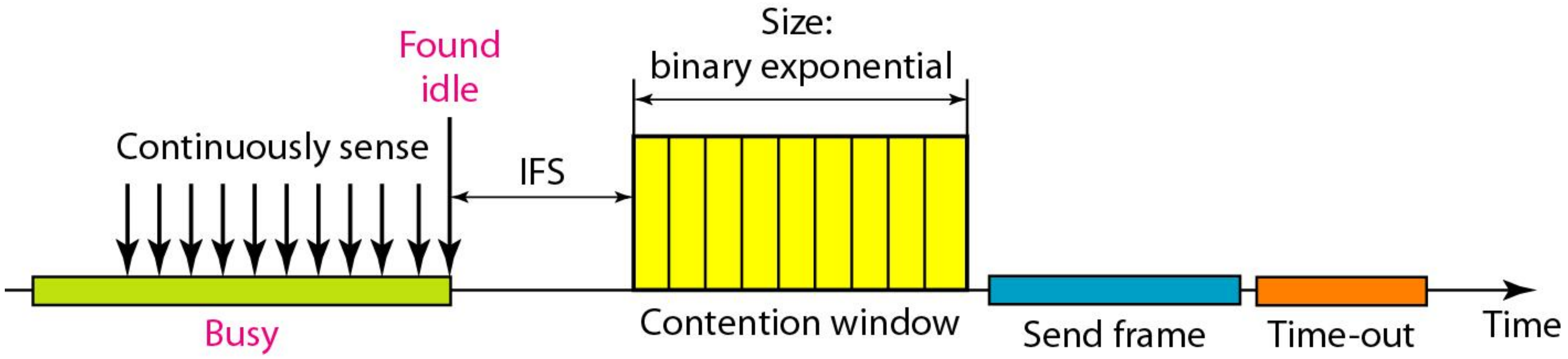


CSMA/CA

- Developed because collision detection is not possible in wireless system.
- In this case, collisions are avoided through the use of CSMA/CA's three strategies:
  1. The inter-frame space
  2. The contention window
  3. Acknowledgement

- **Interframe space:** When an idle channel is found, the station does not send immediately. It waits for a period of time called ***Interframe space*** or **IFS.**

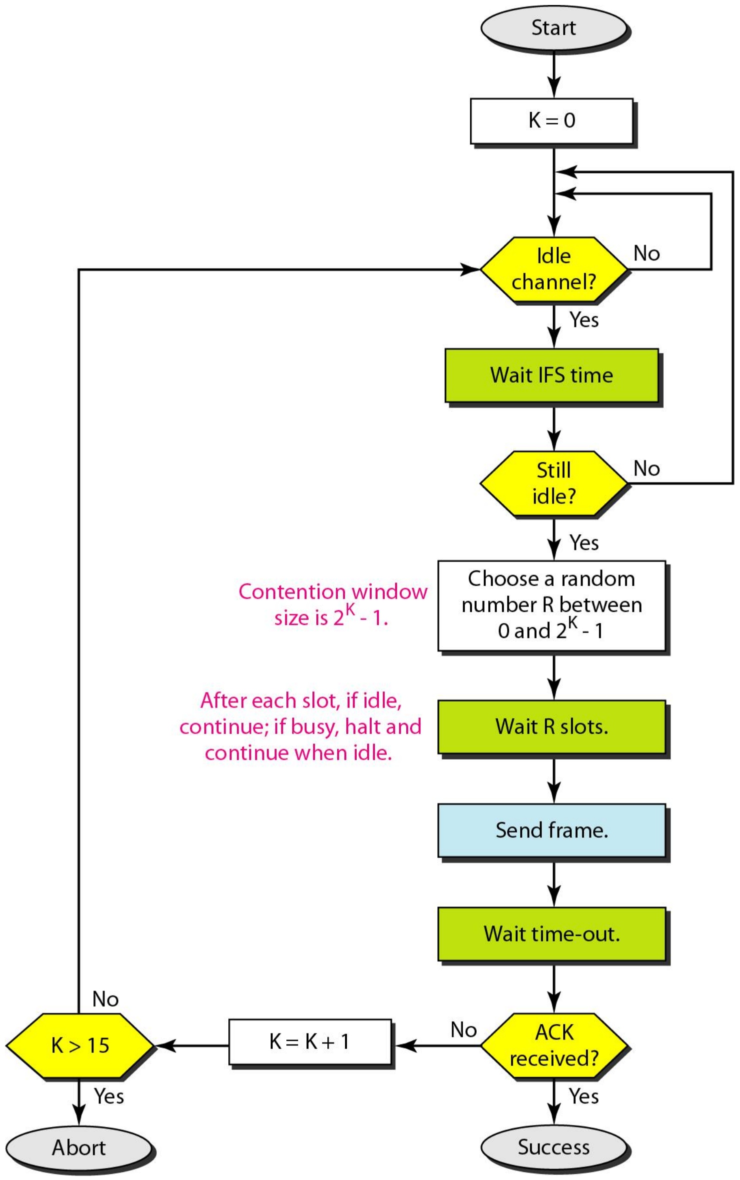
- **Contention window:** It is the amount of time divided into slots. A station which is ready to send chooses a random number of slots as its wait time.
- **Acknowledgement:** Even with all these precautions, there still may be a collision resulting in destroyed data, or corrupted data.



The algorithm of CSMA/CA :

- When a frame is ready, the transmitting station checks whether the channel is idle or busy.
- If the channel is busy, the station waits until the channel becomes idle.
- If the channel is idle, the station waits for an **Inter-frame space (IFS)** amount of time and then sends the frame.
- After sending the frame, it sets a timer.
- The station then waits for **acknowledgement** from the receiver. If it receives the acknowledgement before expiry of timer, it marks a successful transmission.
- Otherwise, it waits for a back-off time period and restarts the algorithm.

Flow diagram for CSMA/CA:



CSMA: Key-points

- In CSMA/CD Collisions are detected.
- In CSMA/CA collisions are avoided.
- CSMA/CD is used in wired LAN, Ethernet LAN (802.3).
- CSMA/CA is used in wireless LAN, 802.11 standard.
- Collisions are not completely eliminated in CSMA/CA. However, it is reduced as compared to CSMA/CD.
- Collision detection is not possible in wireless LAN because of:
  - ➢ Reduction in the strength of the received signal to a greater extent
  - ➢ Hidden station problem
  - ➢ Exposed station problem