Chapter 12

Multiple Access

Multiple Access

- **Broadcast link** used in LAN consists of multiple sending and receiving nodes connected to or use a single shared link

Broadcast links Examples

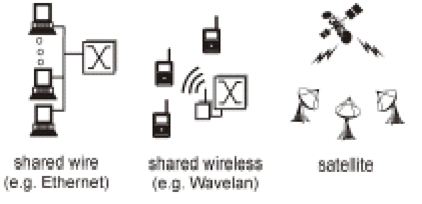
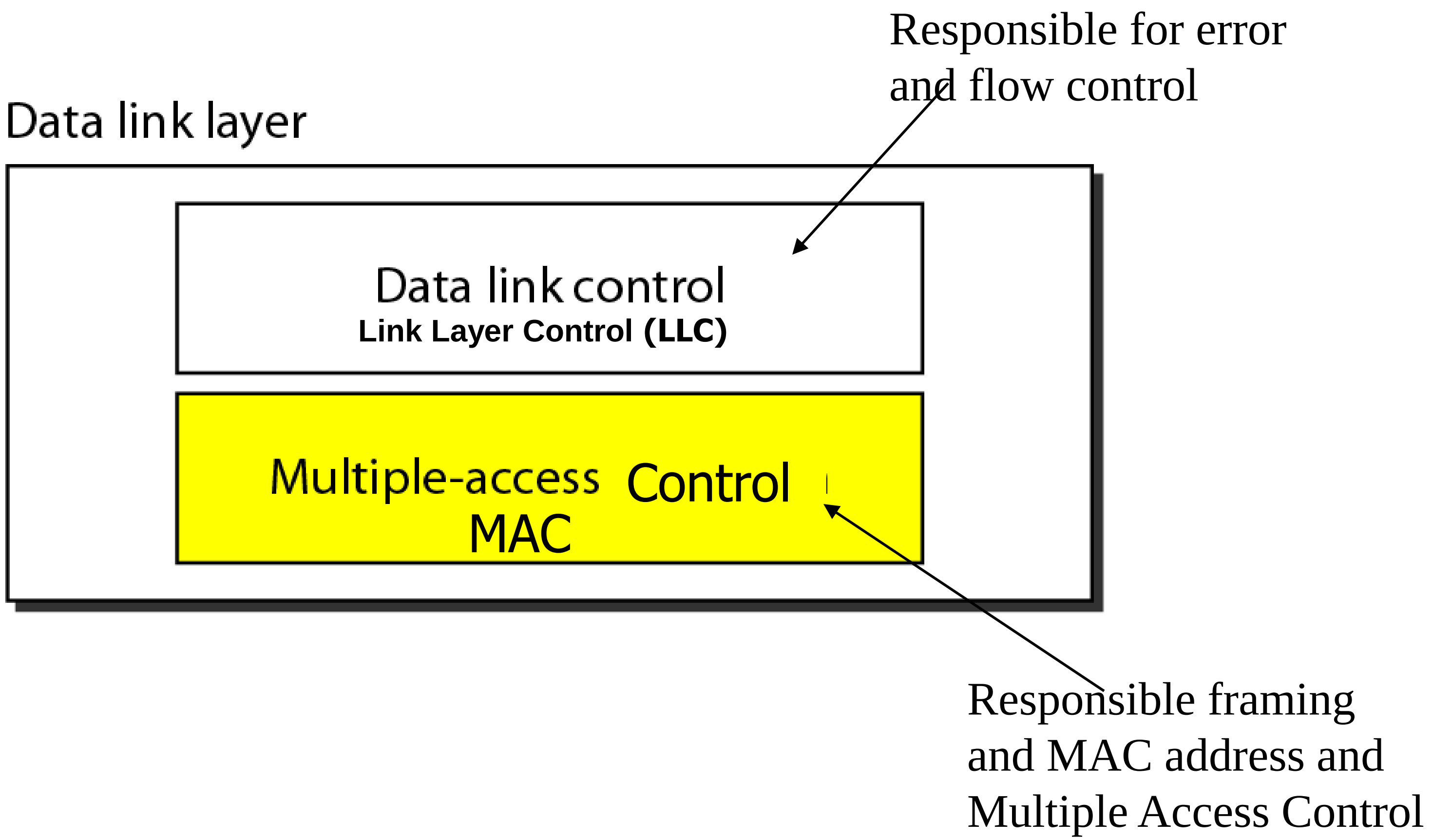


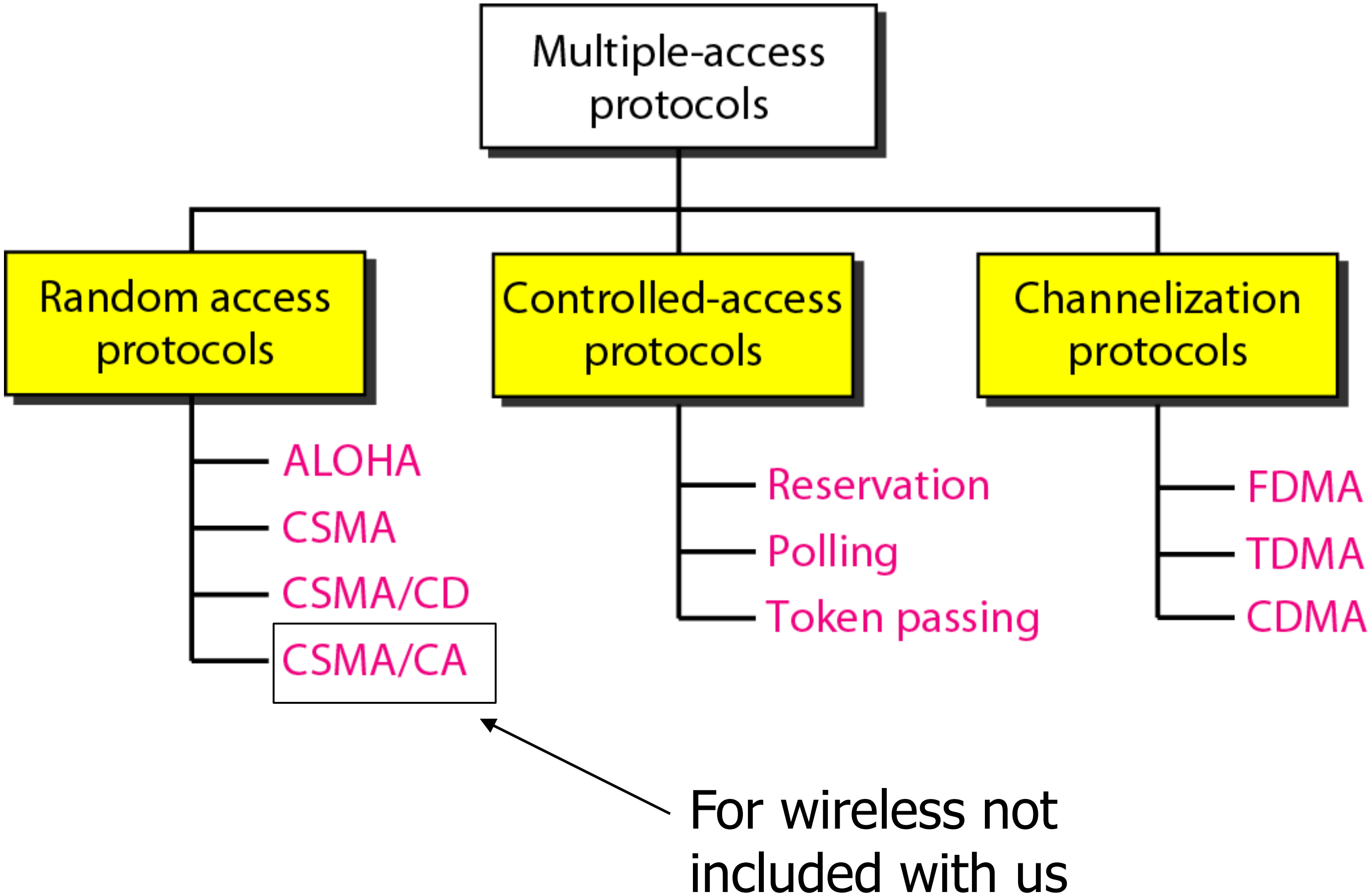
Figure 12.1 *Data link layer divided into two functionality-oriented sublayers*



Multiple Access

- **Problem**: When two or more nodes transmit at the same time, their frames <u>will collide</u> and the link bandwidth is **wasted** during collision
  - How to coordinate the access of multiple sending/receiving nodes to the shared link**???**
- **Solution**: We need a **protocol** to coordinate the transmission of the active nodes
- These protocols are called **Medium or Multiple Access Control (MAC) Protocols** belong to a **sublayer** of the data link layer called **MAC** (Medium Access Control)
- What is expected from Multiple Access Protocols:
  - Main task is to **minimize collisions** in order to **utilize the bandwidth** by:
    - Determining **when** a station can use the link (medium)
    - **what** a station should do when the link is **busy**
    - **what** the station should do when it is involved in **collision**

Taxonomy of multiple-access protocols discussed in this chapter



Random Access

- **Random Access (or contention) Protocols:**

  - No station is superior to another station and none is assigned the control over another.

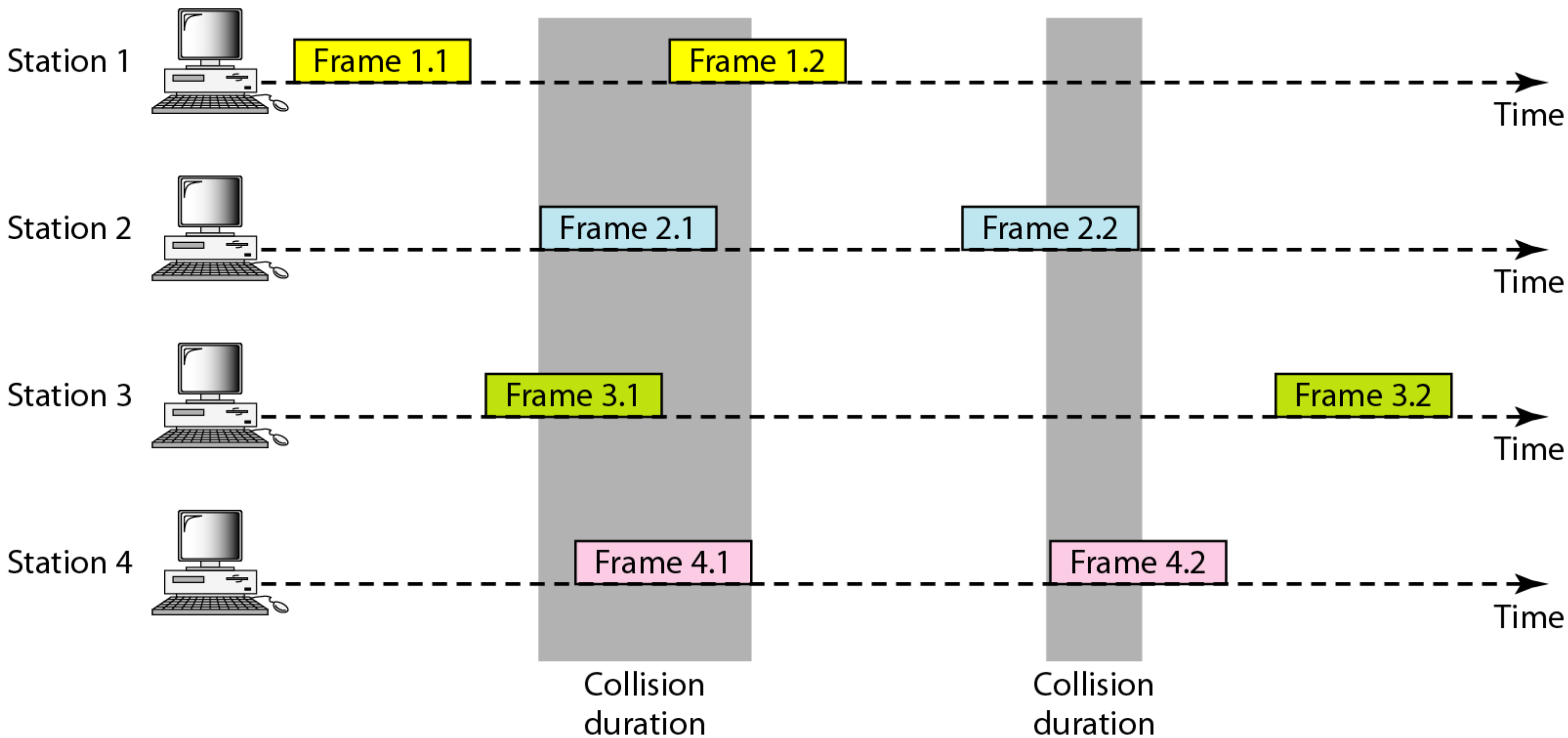
  - A station with a frame to be transmitted **can use the link directly based** on a procedure defined by the protocol to make a decision on whether or not to send.

- .

ALOHA Protocols

- Was designed for **wireless LAN** and can be used for **any shared medium**
- **Pure ALOHA** Protocol Description
  - All frames from any station are of fixed length (**L bits**)
  - Stations transmit at equal **transmission time (*all stations produce frames with equal frame lengths*)**.
  - A station that has data **can transmit at any time**
  - **After transmitting a frame**, the sender **waits** for an **acknowledgment** for an amount of time (time out) equal to the **maximum round-trip propagation delay** = 2* t_{prop}
  - If **no ACK** was received, sender assumes that the **frame or ACK** has been destroyed and **resends** that frame after it **waits for a *random* amount of time**
  - If station fails to receive an ACK after repeated transmissions, **it gives up**
  - **Channel utilization or efficiency or Throughput** is the **percentage** of the transmitted frames that arrive **successfully** (without collisions) **or** the **percentage** of the **channel bandwidth** that will be used for transmitting frames without collisions

Frames in a pure ALOHA network

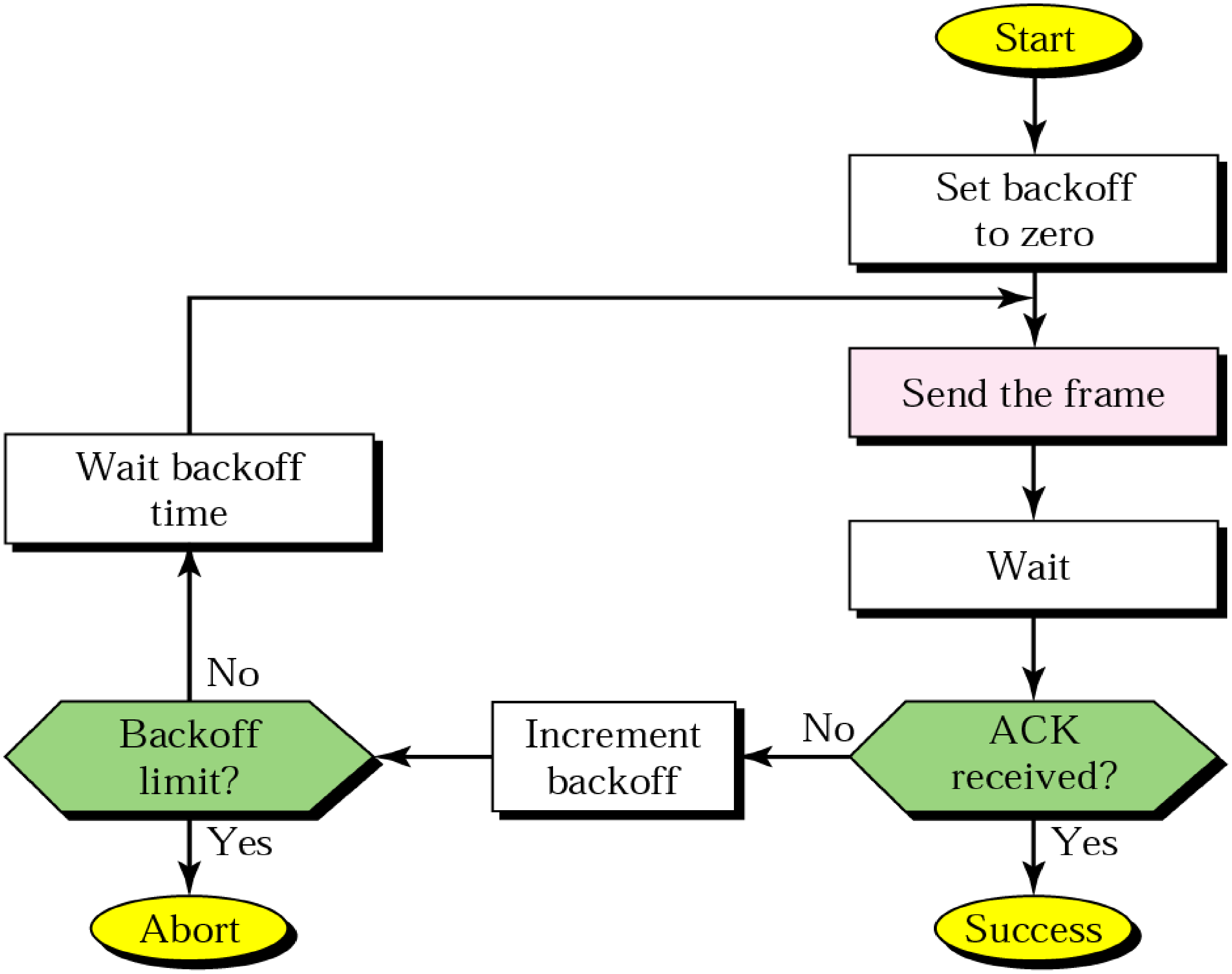


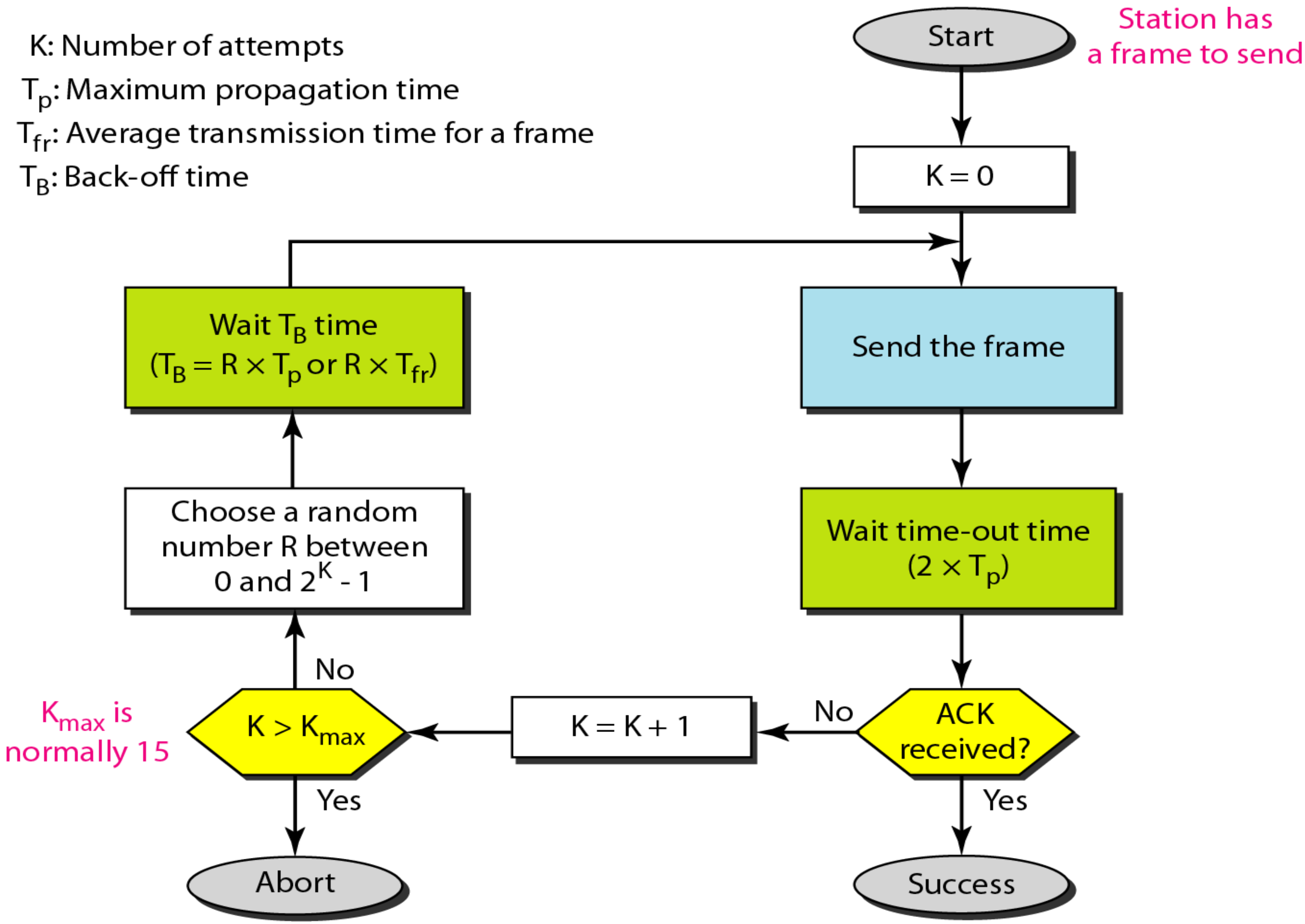
Pure ALOHA:

1. Each station sends a frame whenever is has a frame to send

2. One channel to share, possibility of collision between frames from different stations

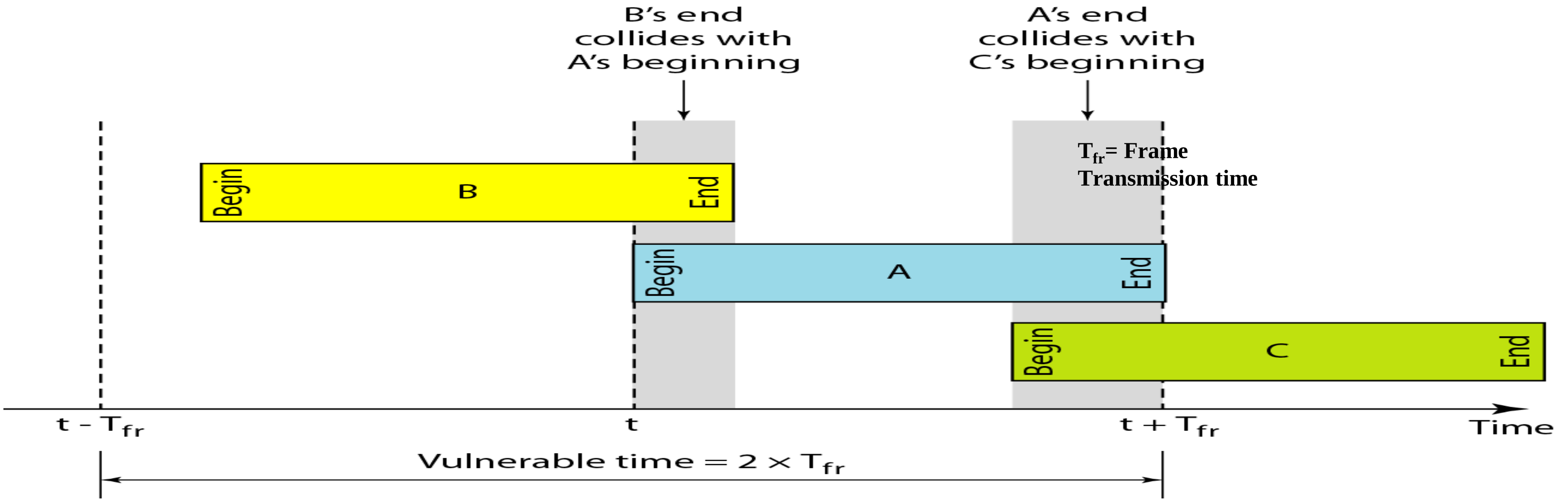
Procedure for ALOHA protocol





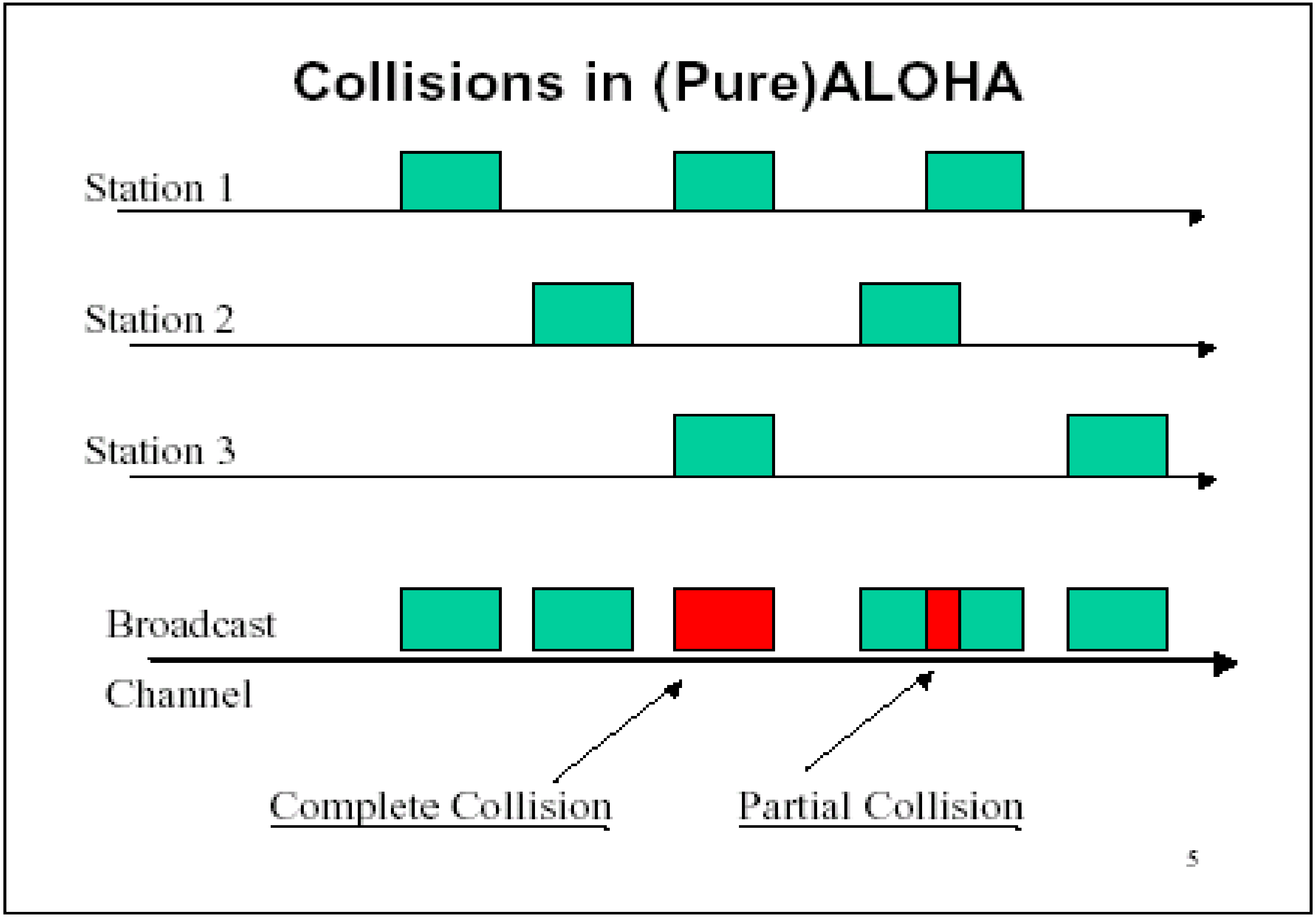
K: Number of attempts
T_p: Maximum propagation time
T_{fr}: Average transmission time for a frame
T_B: Back-off time
Start
Station has a frame to send
$K = 0$
Send the frame
Wait time-out time ($2 \times T_p$)
ACK received?
No
Yes
Success
$K = K + 1$
$K > K_{max}$
K_{max} is normally 15
No
Yes
Abort
Choose a random number R between 0 and $2^K - 1$
Wait T_B time ($T_B = R \times T_p$ or $R \times T_{fr}$)

Critical time for pure ALOHA protocol



If the frame <u>transmission time</u> is T sec, then the vulnerable time is = 2 T sec.
This means no station should send during the T-sec before this station starts transmission and no station should start sending during the T-sec period that the current station is sending.

Pure ALOHA



In pure ALOHA, frames are transmitted at completely arbitrary times.

Note

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

G = Average number of frames generated by the system (all stations) during one frame transmission time

ALOHA Maximum channel utilization is 18%
(i.e, if the system produces F frames/s, then 0.18 * F frames will arrive successfully on average without the need of retransmission)

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

Solution

The frame transmission time is 200/200 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 $1000 \times 0.135 = 135$ frames. Only 135 frames out of 1000 will probably survive.

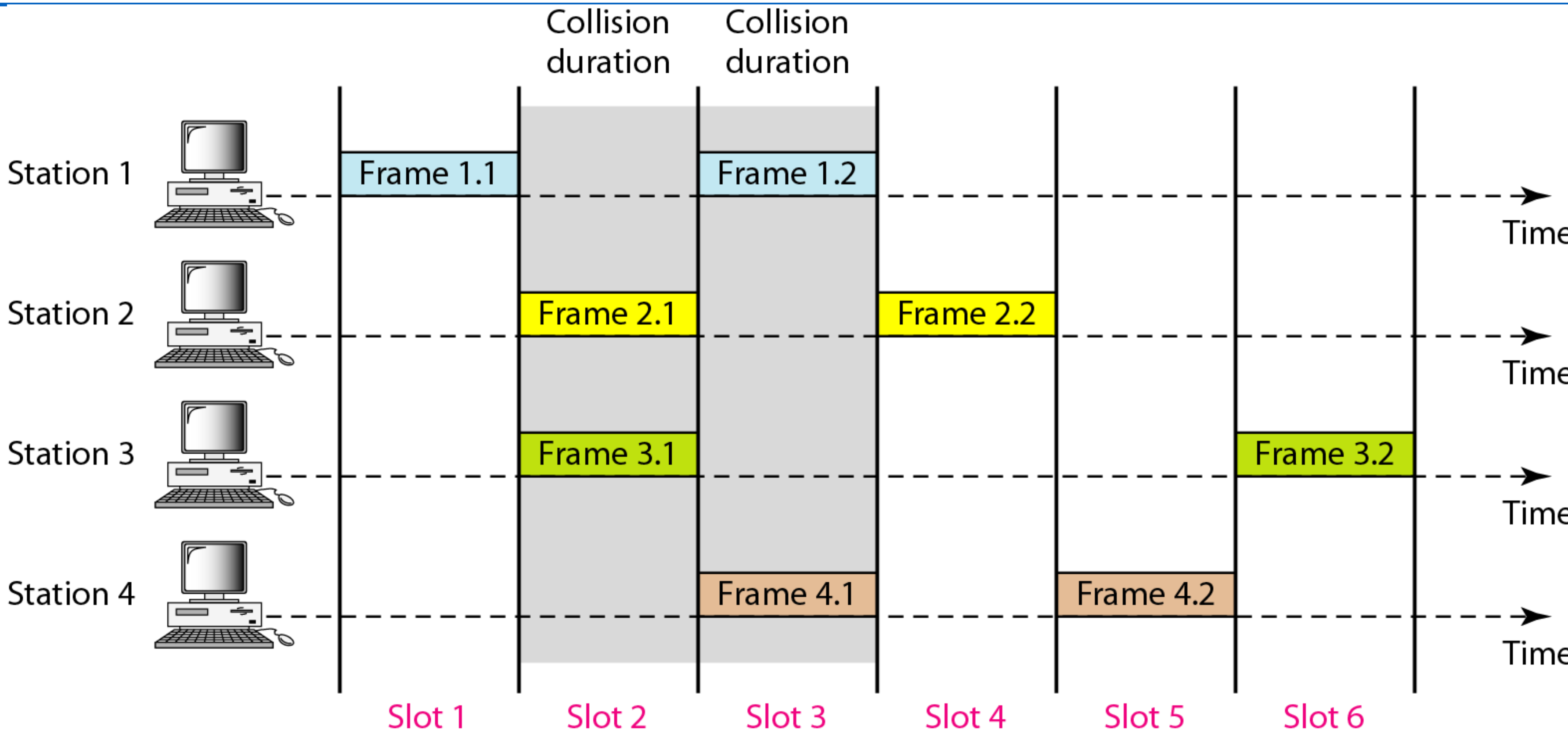
b. If the system creates 500 frames per second, this is (1/2) frame per millisecond. The load is (1/2). In this case $S = G \times e^{-2G}$ or $S = 0.184$ (18.4 percent). This means that the throughput is $500 \times 0.184 = 92$ and that only 92 frames out of 500 will probably survive. Note that this 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 $250 \times 0.152 = 38$. Only 38 frames out of 250 will probably survive.

Random Access – Slotted ALOHA

- Time is divided into slots equal to a **frame transmission time (T_{fr})**
- A station can transmit at the beginning of a slot only
- If a station misses the beginning of a slot, it has to wait until the beginning of the next time slot.
- **A central clock** or station informs all stations about the start of a each slot
- Maximum channel utilization is **37%**

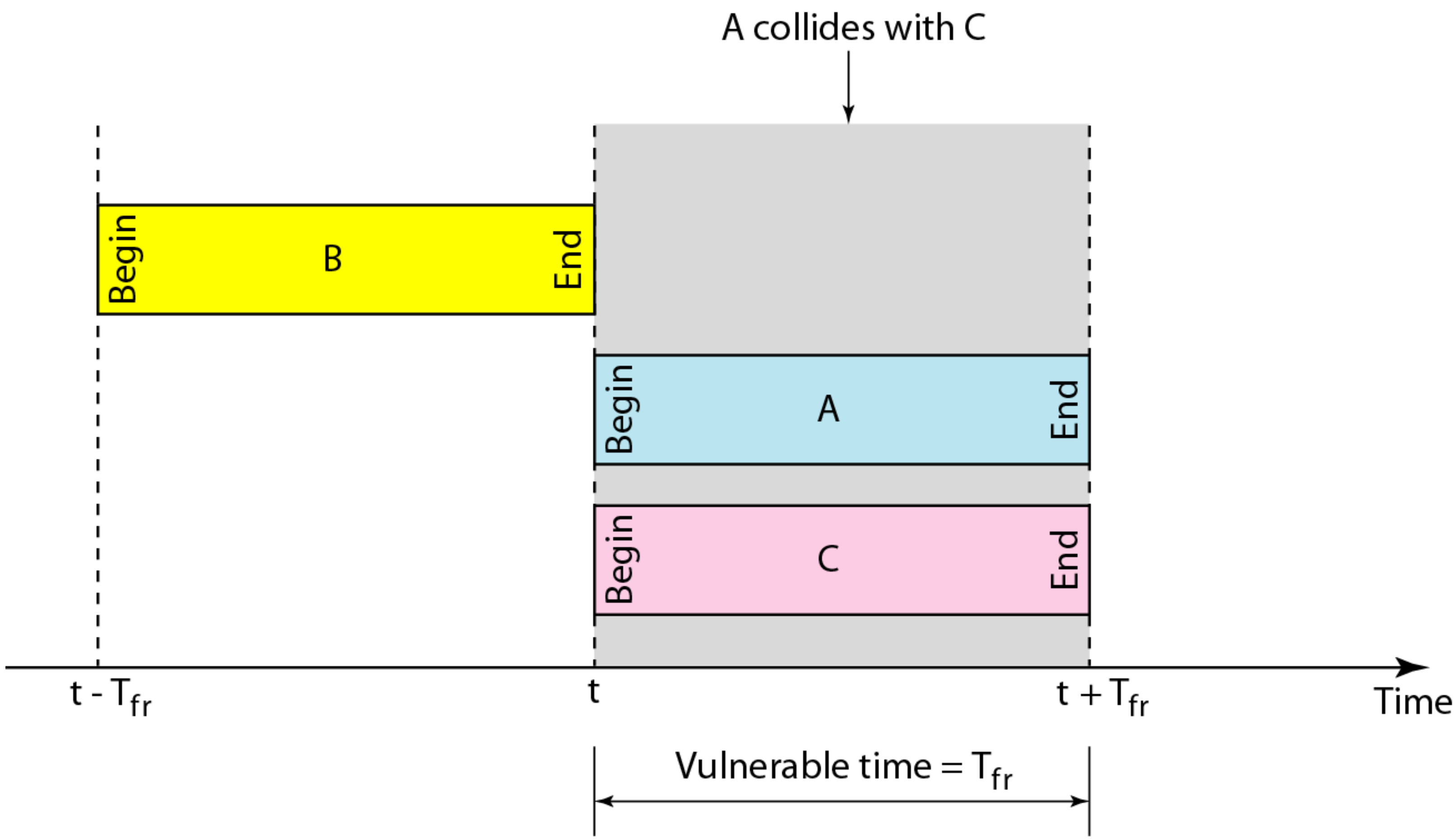
Frames in a slotted ALOHA network



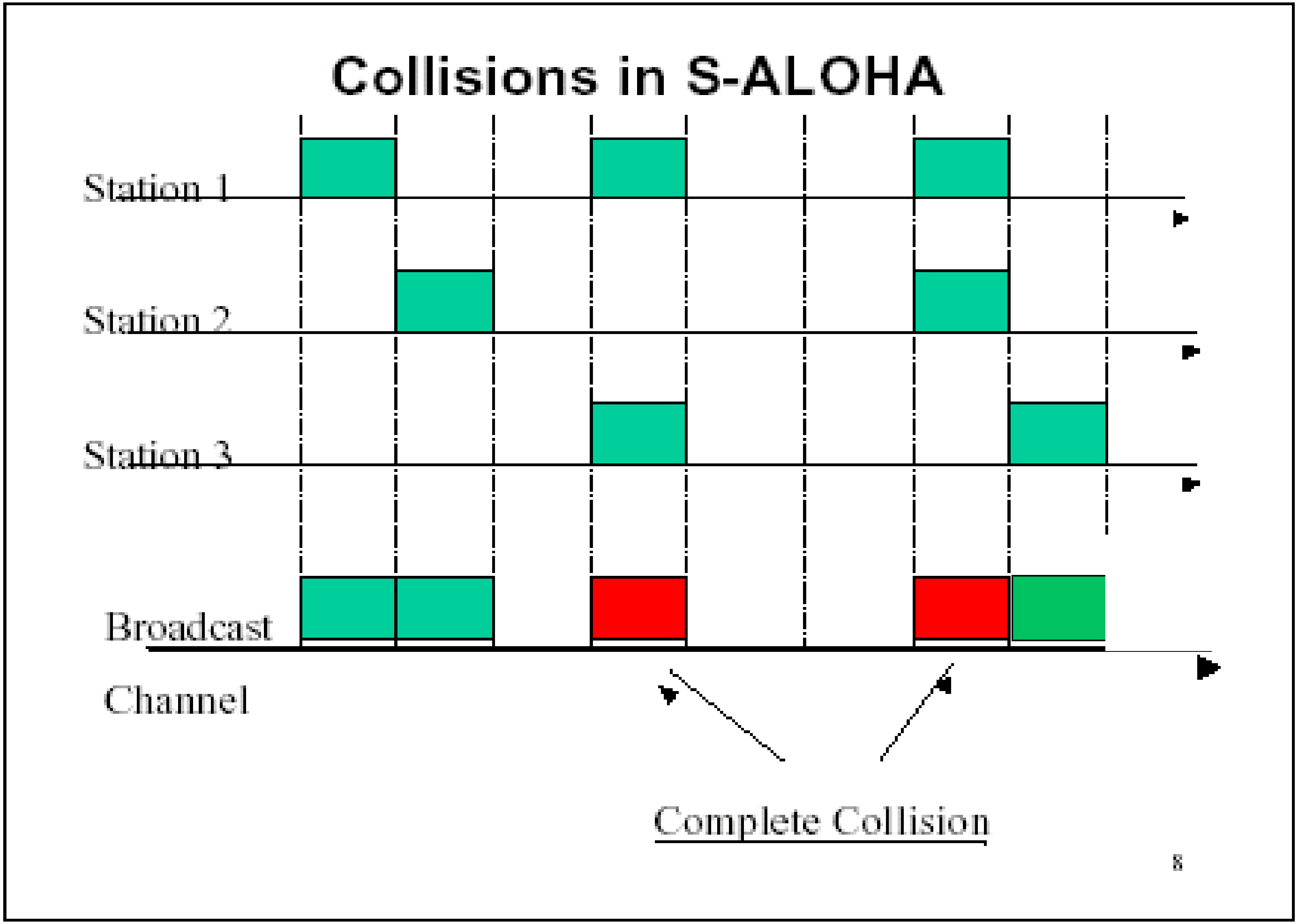
Slotted ALOHA:

1. We divide the time into slots and force the station to send only at the beginning of the time slot

In danger time for slotted ALOHA protocol



Random Access – Slotted ALOHA



Note

The throughput for slotted ALOHA is

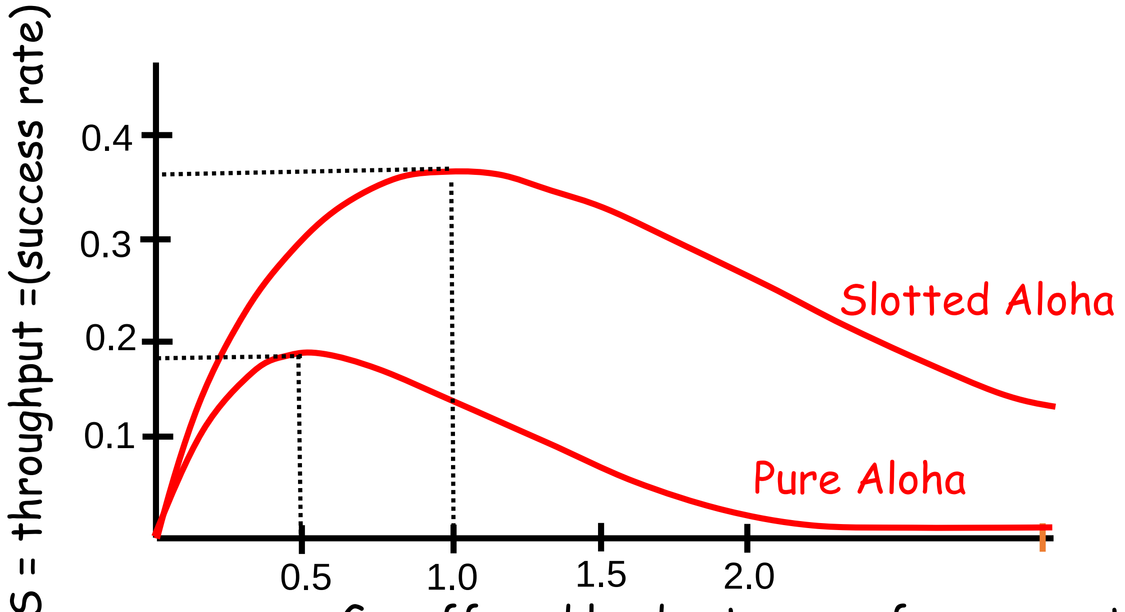
$$S = G \times e^{-G}.$$

The maximum throughput

$$S_{max} = 0.368 \text{ when } G = 1.$$

G = <u>Average number</u> of frames generated by the system (all stations) during one <u>frame transmission time</u>

Efficiency of Aloha



G = offered load rate= new frames+ retransmitted
= Total frames presented to the link per the transmission time of a single frame

- Advantage of ALOHA protocols
  - A node that has frames to be transmitted can **transmit continuously** at the **full rate of channel (R bps)** if it is the <u>only node</u> with frames
  - Simple to be implemented
  - No master station is needed to control the medium
- Disadvantage
  - If (M) nodes want to transmit, many collisions can occur and the rate allocated for each node will **not be on average R/M bps**
  - This causes low channel utilization

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 $1000 \times 0.0368 = 368$ frames. Only 368 out of 1000 frames will probably survive. Note that this is the maximum throughput case, percentagewise.

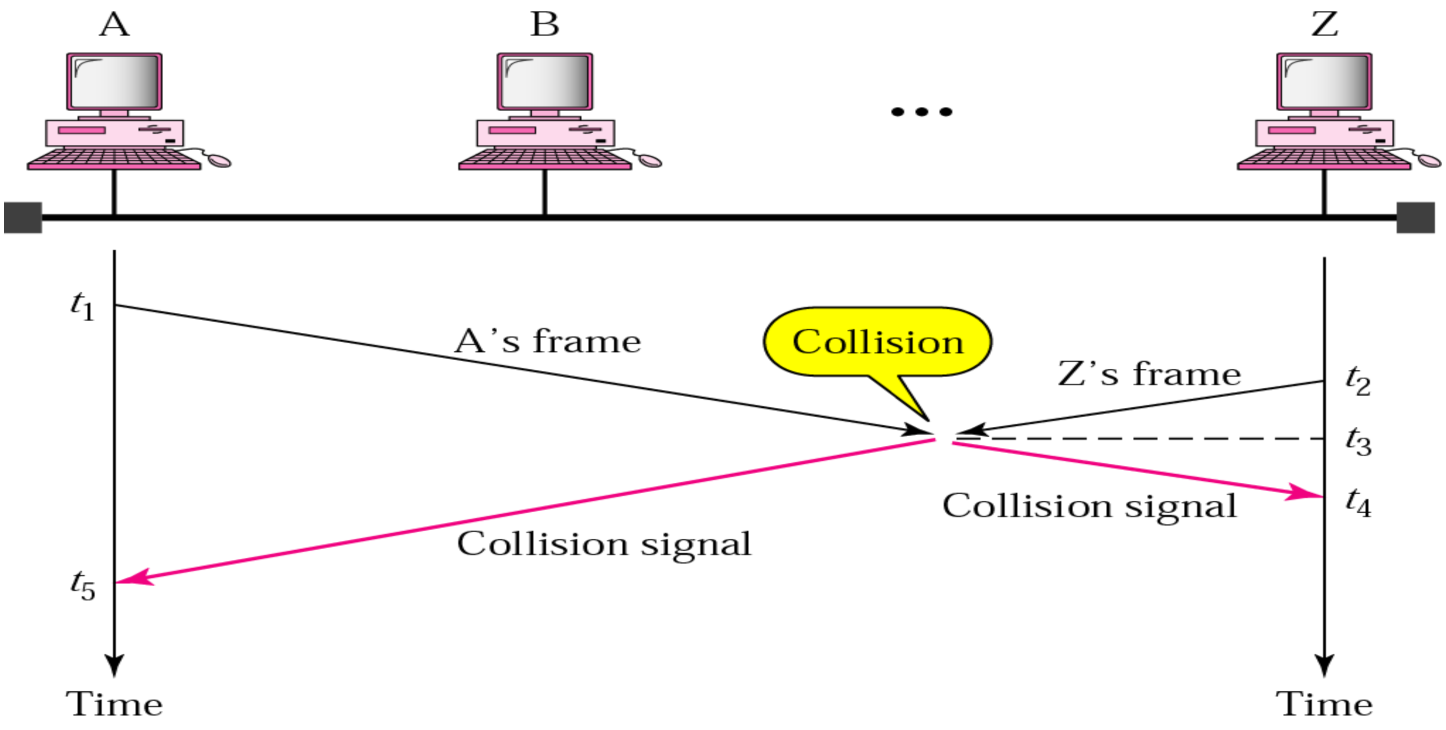
b. Here G is $\frac{1}{2}$. In this case $S = G \times e^{-G}$ or $S = 0.303$ (30.3 percent). This means that the throughput is $500 \times 0.0303 = 151$. Only 151 frames out of 500 will probably survive.

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 $250 \times 0.195 = 49$. Only 49 frames out of 250 will probably survive.

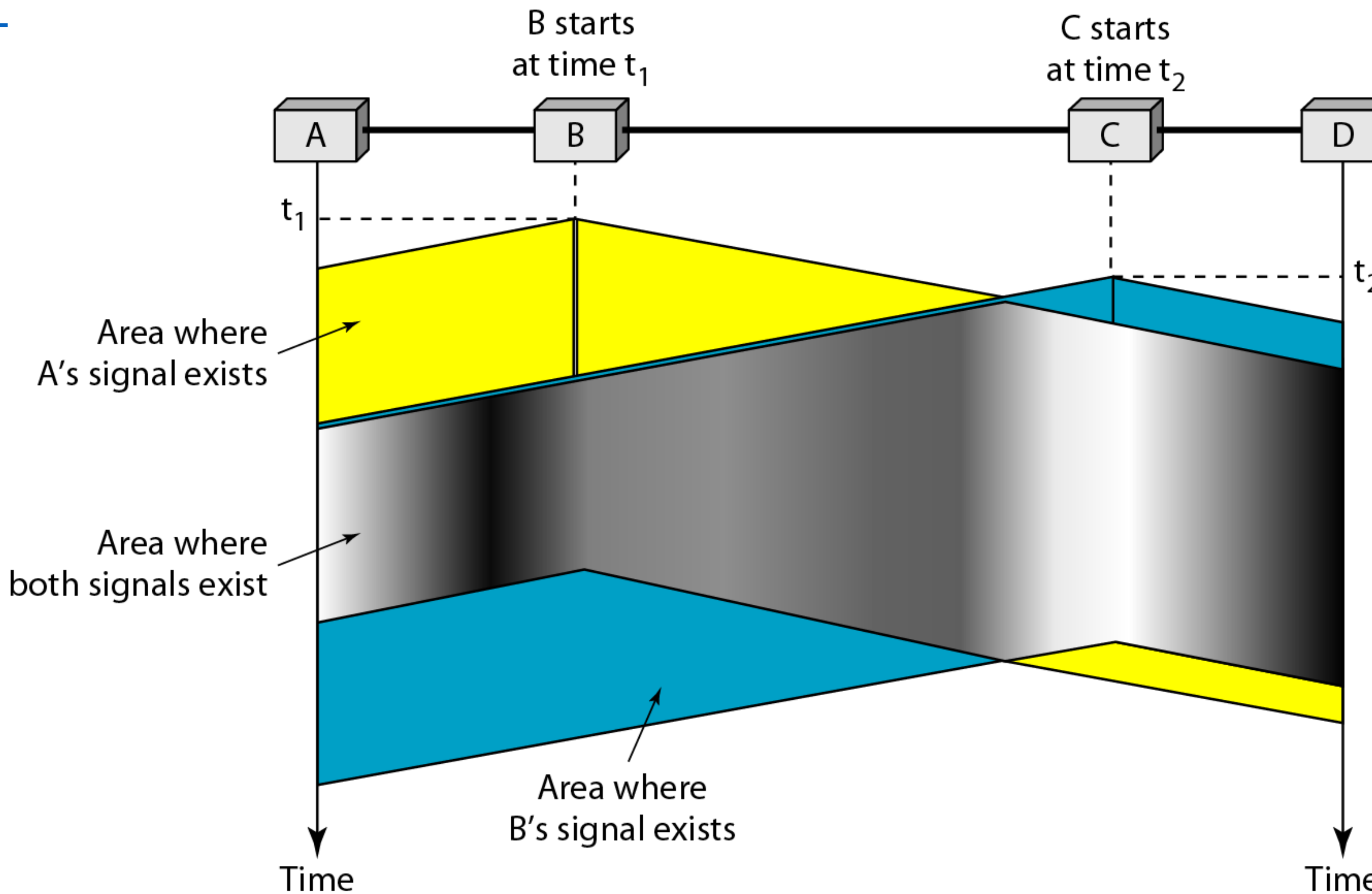
Random Access – Carrier Sense Multiple Access (CSMA)

- To improve performance, avoid transmissions that are certain to cause collisions
- Based on the fact that in LAN propagation time is **very small**
- ➔ If a frame was sent by a station, All stations knows immediately so they can **wait before start sending**
  - ➔ A station with frames to be sent, should **sense the medium** for the presence of another transmission (carrier) before it starts its own transmission

- This can **reduce** the possibility of collision but it <u>cannot eliminate</u> it.
    - Collision can only happen when more than one station begin transmitting within a short time (the **propagation time** period)



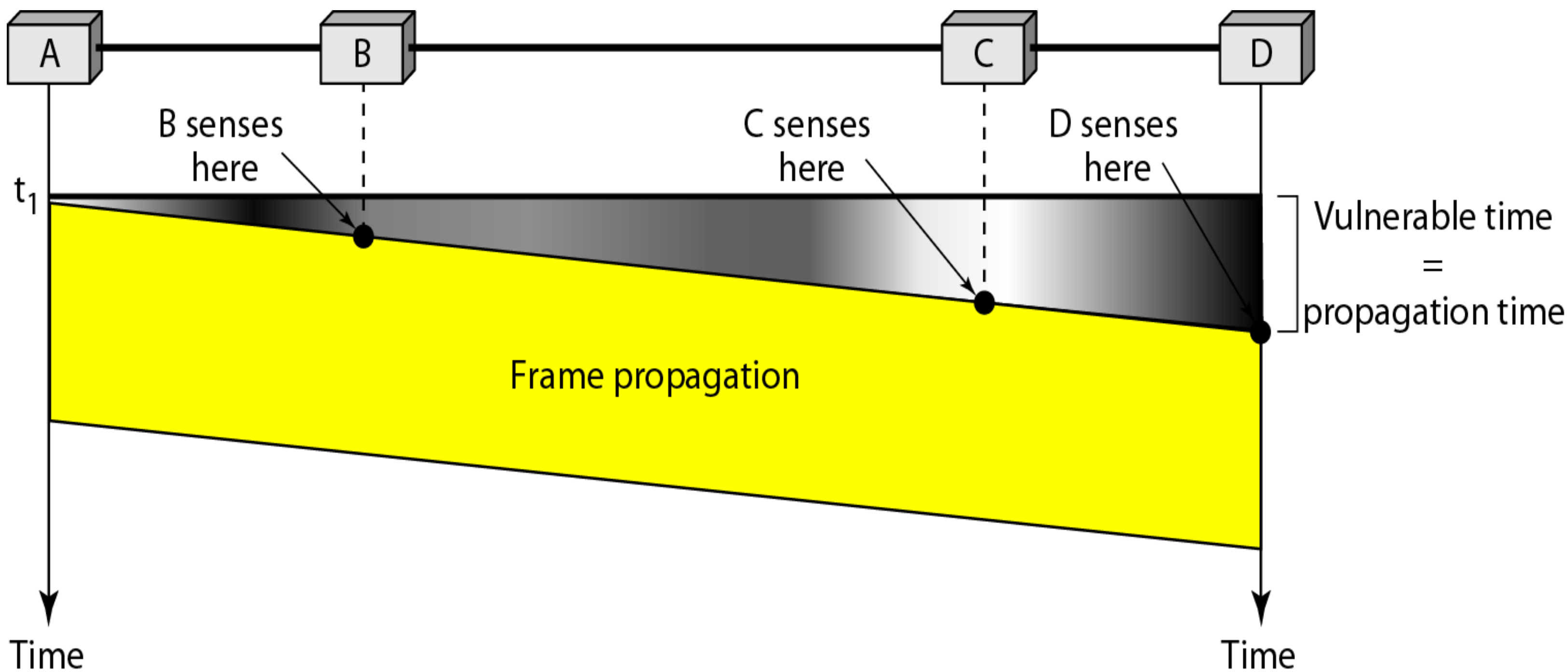
Space/time model of the collision in CSMA



CSMA – each station first listen to the medium before sending
Principle : "sense before transmit" or "listen before talk"

Random Access – Carrier Sense Multiple Access (CSMA)

- Vulnerable time for CSMA is the <u>**maximum propagation time**</u>
- The longer the propagation delay, the <u>worse the performance</u> of the protocol because of the above case.



Types of CSMA Protocols

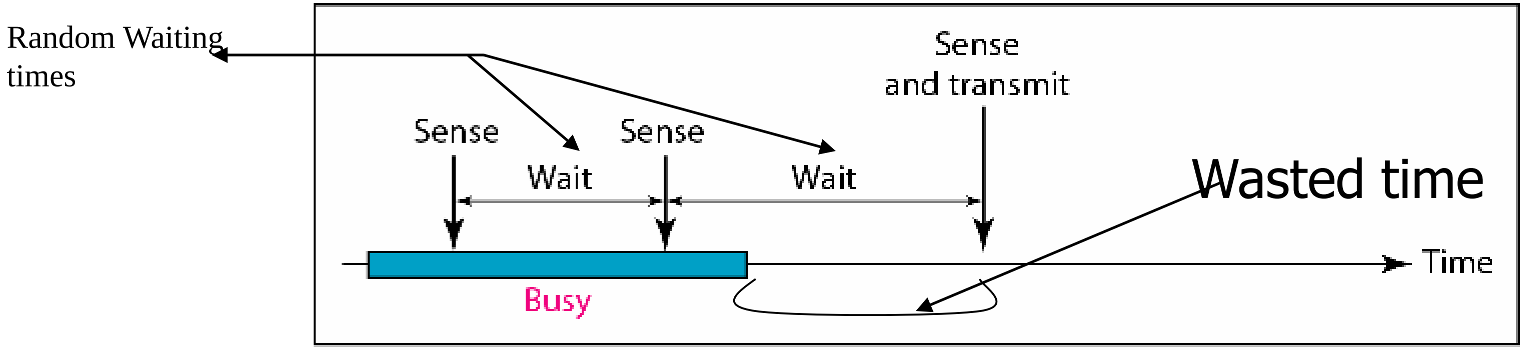
Different CSMA protocols that determine:

- What a station should do when the medium is **idle**?
- What a station should do when the medium is **busy**?

1. Non-Persistent CSMA
2. 1-Persistent CSMA
3. p-Persistent CSMA

Nonpersistent CSMA

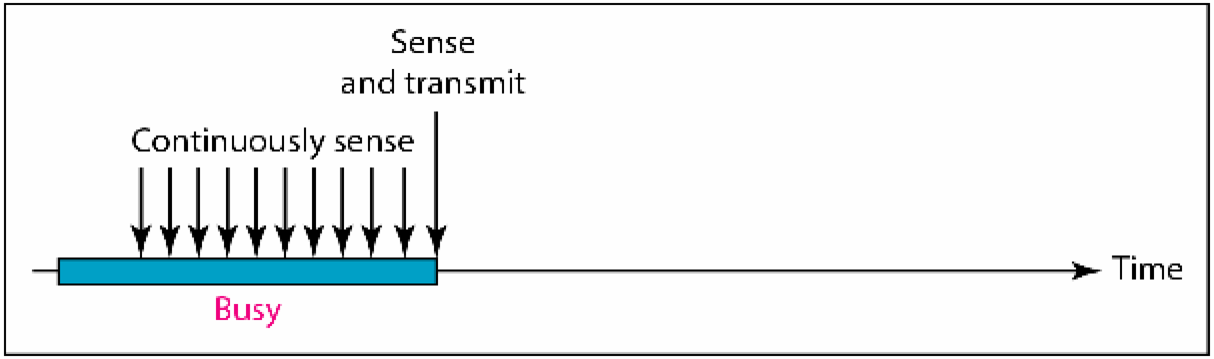
- A station with frames to be sent, should sense the medium
  1. If medium is idle, **transmit**; otherwise, go to 2
  2. If medium is busy, (**backoff**) wait a ***random*** **amount of time** and repeat **1**
- Non-persistent Stations are **deferential (respect others)**
- Performance:
  - Random delays reduces probability of collisions because two stations with data to be transmitted will wait for different amount of times.
  - Bandwidth is **wasted** if waiting time (backoff) is large because medium will remain idle following end of transmission even if one or more stations have frames to send



b. Nonpersistent

1-persistent CSMA

- To avoid idle channel time, 1-persistent protocol used
- Station wishing to transmit listens to the medium:
  1. If medium idle, **transmit** immediately;
  2. If medium busy, **continuously listen** until medium becomes idle; then transmit immediately with probability 1
- Performance
  - 1-persistent stations are **selfish**
  - If two or more stations becomes ready at the same time, **collision guaranteed**

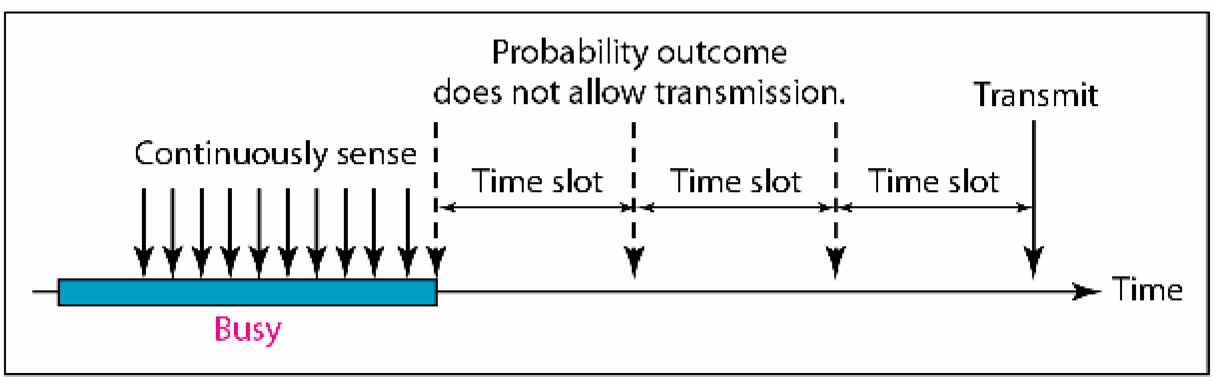


a. 1-persistent

P-persistent CSMA

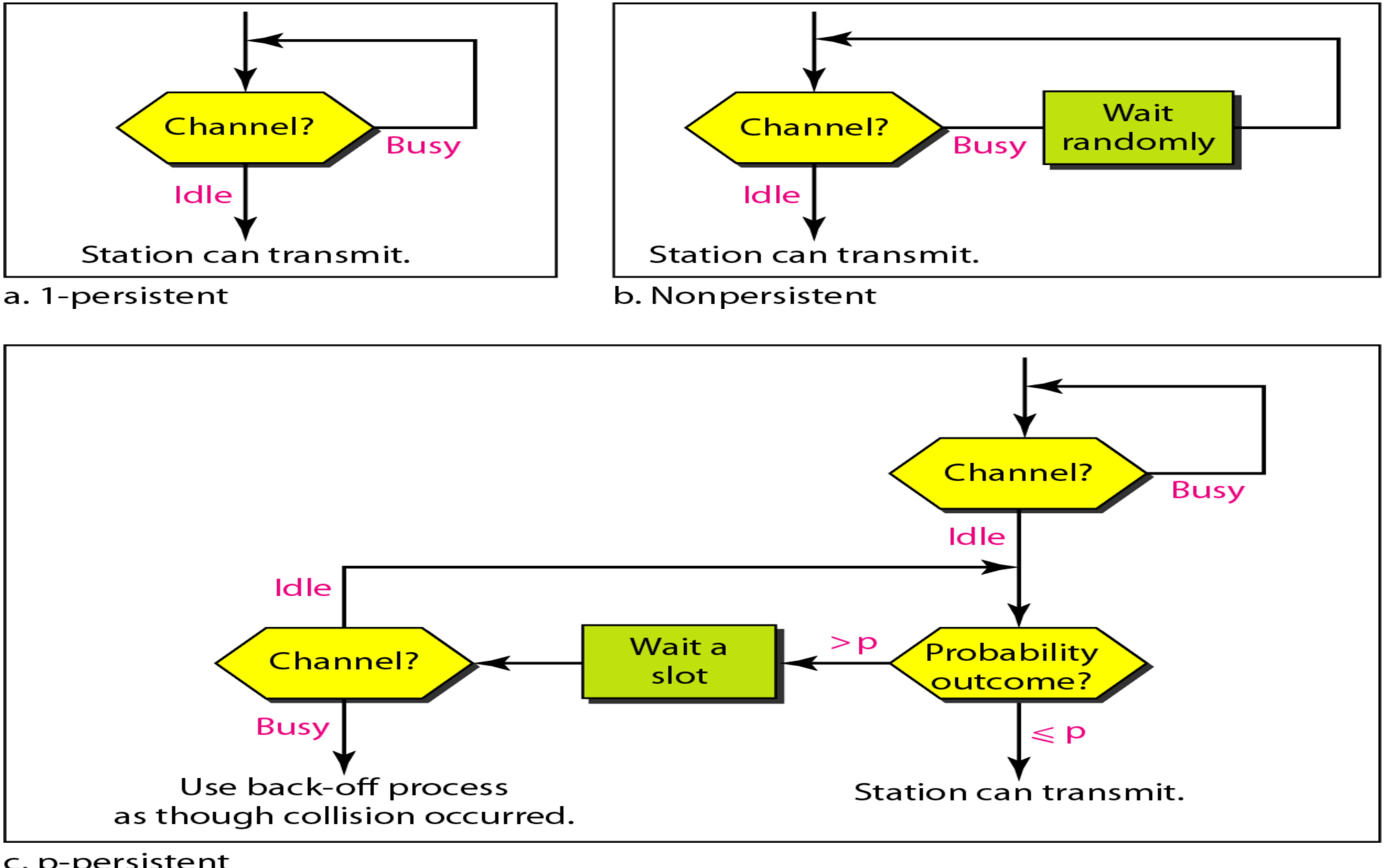
- Time is divided to slots where each Time unit (slot) typically equals **maximum propagation delay**
- Station wishing to transmit listens to the medium:

1. If medium idle,
   - transmit with probability (**p**), OR
   - wait **one time unit (slot)** with probability (**1 – p**), then repeat 1.
2. If medium busy, **continuously listen until idle** and repeat step **1**
3. Performance
   - Reduces the possibility of collisions like **nonpersistent**
   - Reduces channel idle time like **1-persistent**

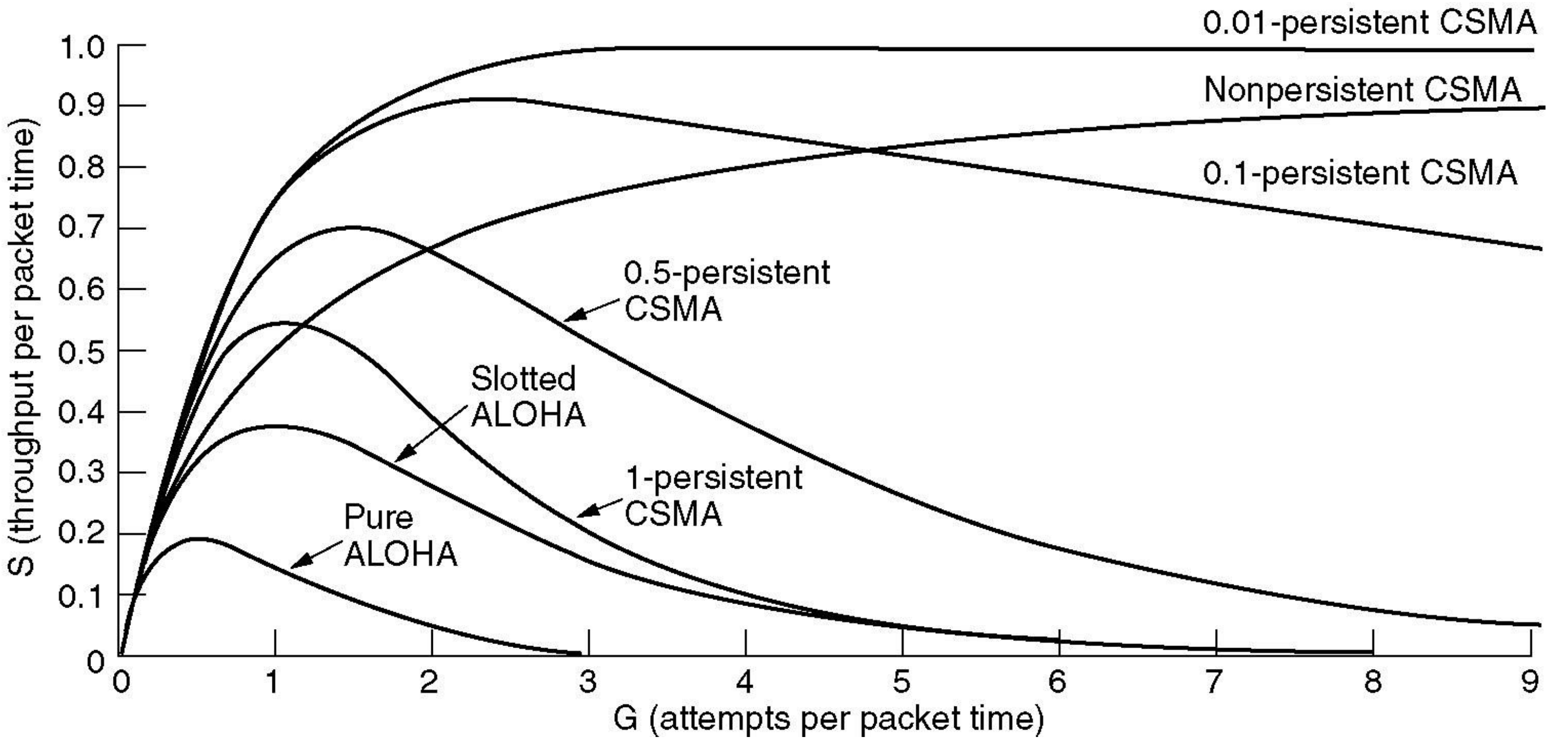


c. p-persistent

Flow diagram for three persistence methods



Persistent and Nonpersistent CSMA



Comparison of the channel utilization versus load for various random access protocols.

CSMA/CD (Collision Detection)

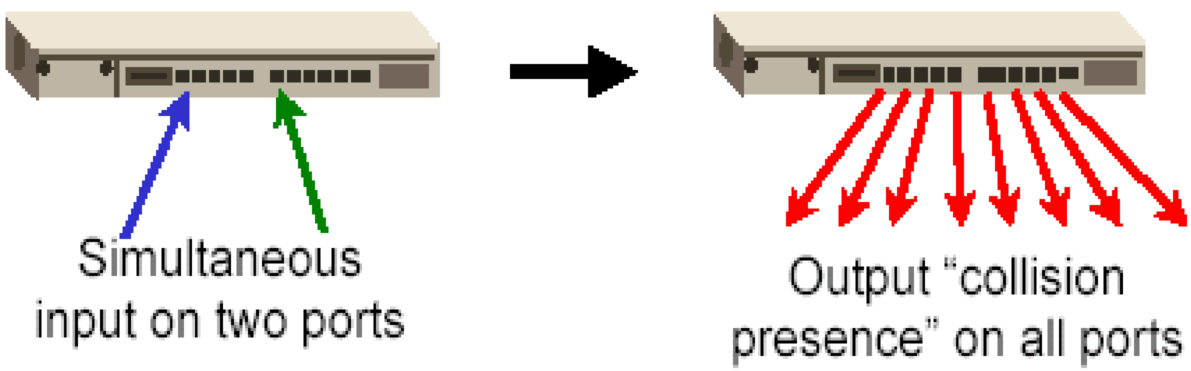
- ***CSMA (all previous methods) has an inefficiency:***
  - If a collision has occurred, the channel is **unstable** until colliding packets have **been fully transmitted**
- ***CSMA/CD*** *(Carrier Sense Multiple Access with Collision Detection)* ***overcomes this as follows:***
  - While transmitting, the sender is **listening to medium** for collisions.
  - Sender **stops transmission** if collision has occurred **reducing channel wastage** .

CSMA/CD is Widely used for **bus topology LANs** (IEEE 802.3, Ethernet).

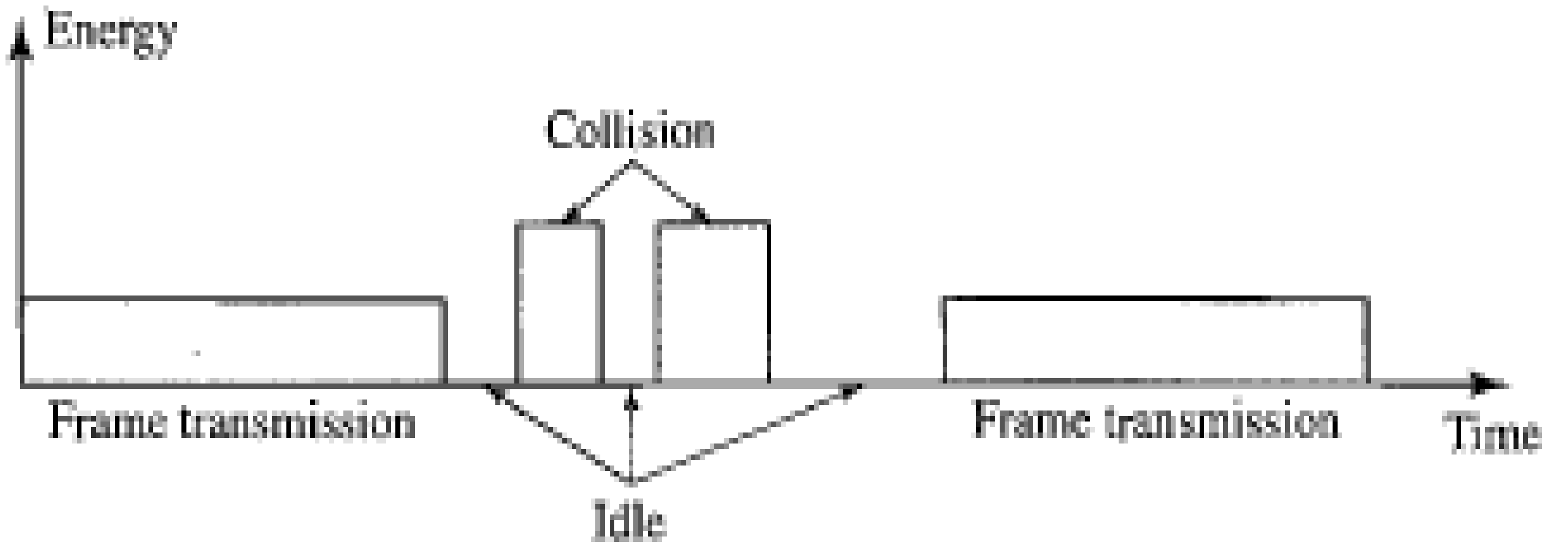
How does a node detect a collision?

Transceiver: **A node monitors the media while transmitting. If the observed power is more than transmitted power** of its own signal, it means collision occurred

Hub: **if input occurs simultaneously on two ports, it indicates a collision. Hub sends a collision presence signal on all ports.**



Energy level during transmission, idleness, or collision



CSMA/CD Protocol

- Use one of the CSMA persistence algorithm

(non-persistent, 1-persistent, p-persistent) for transmission.

- If a collision is detected by a station during its transmission then it should do the following:

i. Abort transmission and

ii. Transmit a ***jam signal*** (48 bit) to notify other stations of collision so that they will **discard the transmitted fra**me also to make sure that the collision signal will stay until detected by <u>the furthest station</u>

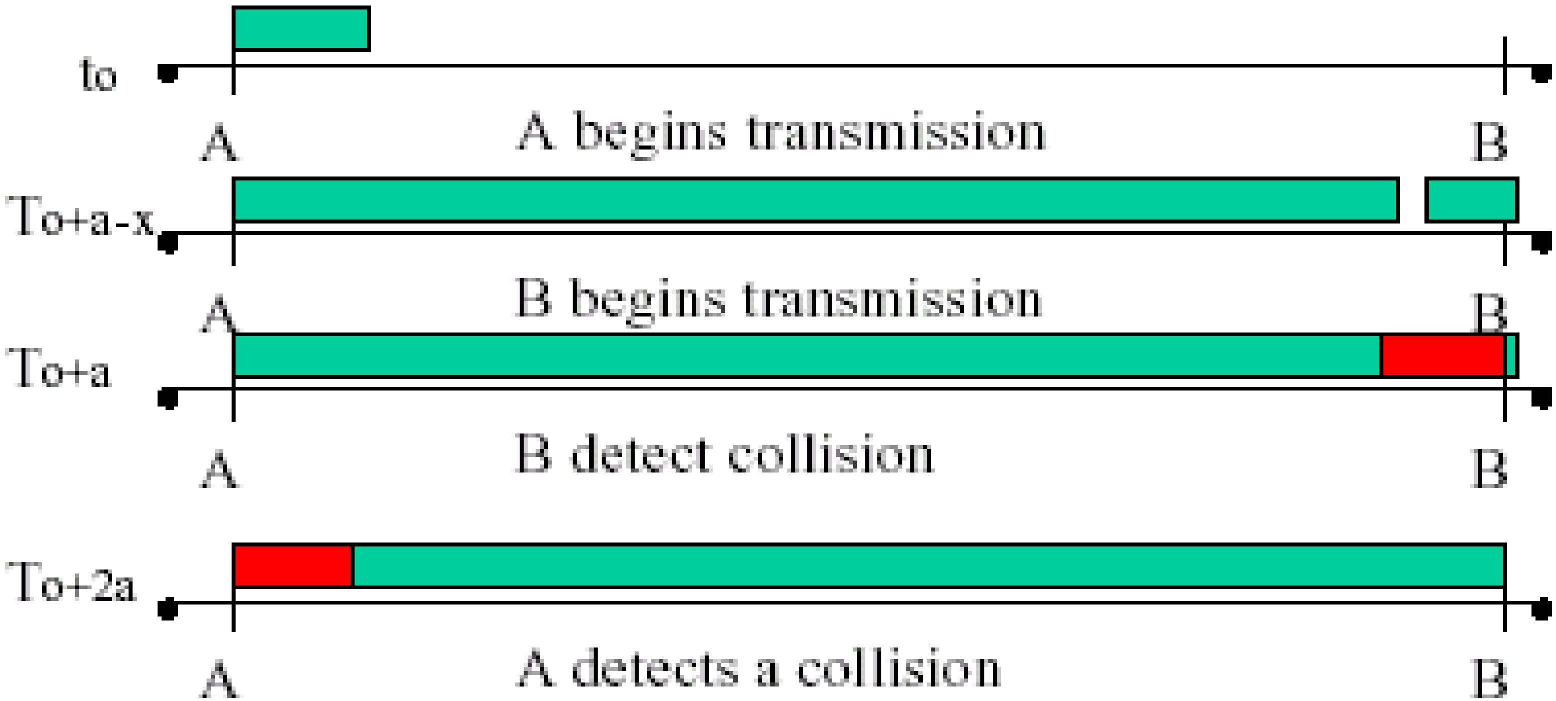
iii. After sending the ***jam signal***,

backoff (wait) for a *random* amount of time, then transmit the frame again

CSMA/CD

- ***Question:*** How long does it take to detect a collision?
- ***Answer:*** *In the* ***worst case***, **twice the maximum propagation delay of the medium**

Note: **a = maximum propagation delay**



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.
- In 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
- Therefore, the frame transmission time T_{fr} must be at least two times the maximum propagation time T_p

$$T_{fr} = 2 \text{ X } T_p$$

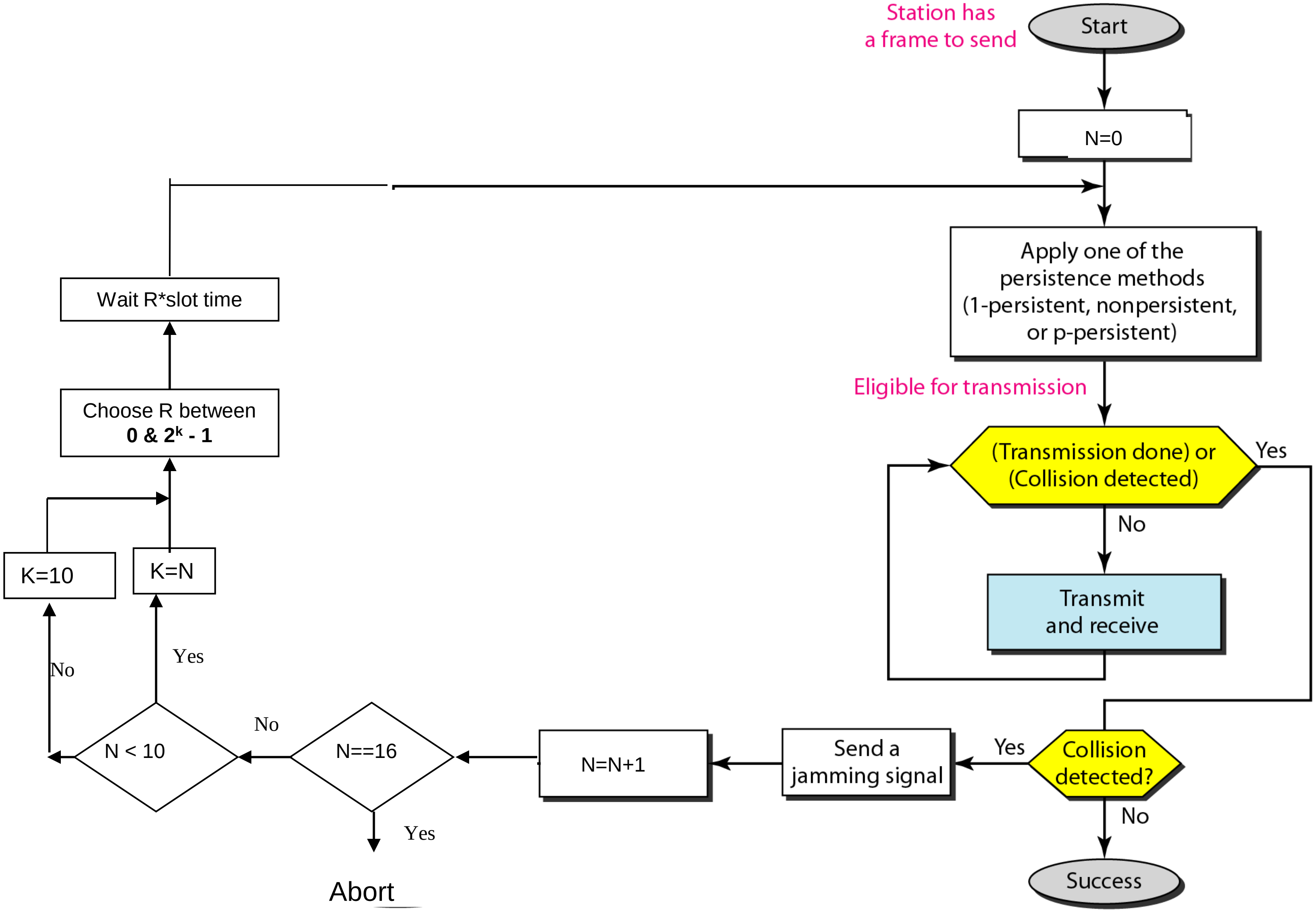
CSMA/CD

- Restrictions of CSMA / CD:
  - Packet **transmission time** should be **at least** as long as the time needed to detect a collision (2 * maximum propagation delay + *jam sequence* transmission time)
  - Otherwise, CSMA/CD does not have an advantage over CSMA

Exponential Backoff Algorithm

- Ethernet uses the exponential backoff algorithms to determine **the best duration of the random waiting period after the collision happens**
- Algorithm:
  - Set "**slot time**" equal to 2*maximum propagation delay + Jam sequence transmission time (= 51.2 usec for Ethernet **10-Mbps** LAN)
  - After K^{th} collision, select a random number (R) between 0 and $2^k - 1$ and **wait** for a period equal to (R***slot time**) then **retransmit** when the medium is **idle, for example:**
    - After first collision (K=1), select a number (R) between 0 and $2^1 - 1$ {0 ,1} and wait for a period equal to R*slot times (Wait for a period 0 usec or 1x51.2 usec) then retransmit when the medium is idle
  - Do not increase random number range, if K=10
    - ➔ Maximum interval {0 – 1023}
  - Give up after 16 unsuccessful attempts and report failure to higher layers

Figure 12.14 *Flow diagram for the CSMA/CD*



Exponential Backoff Algorithm

- Reduces the chance of two waiting stations picking the same random waiting time
- When network traffic is light, it results in **minimum** waiting time before transmission
- As congestion increases (traffic is high), collisions increase, stations backoff by **larger amounts** to reduce the probability of collision.
- Exponential Back off algorithm gives **last-in, first-out effect**
  - Stations with **no or few collisions** will have the chance to transmit before stations that have waited longer because of their previous unsuccessful transmission attempts.

Performance of Random Access Protocols

- Simple and easy to implement
- Decentralized (no central device that can fail and bring down the entire system)
- In low-traffic, packet transfer has low-delay
- However, limited throughput and in heavier traffic, packet delay has no limit.
- In some cases, a station <u>may never</u> have a chance to transfer its packet. (**unfair protocol**)
- A node that has frames to be transmitted can **transmit continuously** at the **full rate of channel (R)** if it is the **only node with frames**
- If (M) nodes want to transmit, many collisions can occur and the rate for each node will <u>**not be on average R/M**</u>

A network using CSMA/CD has a bandwidth of 10 Mbps. If the maximum propagation time (including the delays in the devices and ignoring the time needed to send a jamming signal, as we see later) is 25.6 μs, what is the minimum size of the frame?

Solution

The frame transmission time is $T_{fr} = 2 \times T_p = 51.2\ \mu s$. This means, in the worst case, a station needs to transmit for a period of 51.2 μs to detect the collision. The minimum size of the frame is 10 Mbps × 51.2 μs = 512 bits or 64 bytes. This is actually the minimum size of the frame for Standard Ethernet,

Example

A network using CSMA/CD has a bandwidth of 10 Mbps. If the maximum propagation time (including the delays in the devices and ignoring the time needed to send a jamming signal, as we see later) is 25.6 μs, what is the minimum size of the frame?

Solution

The frame transmission time is $T_{fr} = 2 \times T_p = 51.2$ μs. This means, in the worst case, a station needs to transmit for a period of 51.2 μs to detect the collision. The minimum size of the frame is 10 Mbps $\times$ 51.2 μs = 512 bits or 64 bytes. This is actually the minimum size of the frame for Standard Ethernet.

Problem

- In a *CSMA/CD* network with a data rate of 10 Mbps, the minimum frame size is found to be 512 bits for the correct operation of the collision detection process. What should be the minimum frame size if we increase the data rate to 100 Mbps? To 1 Gbps? To 10 Gbps?

Solution

- Let us find the relationship between the minimum frame size and the data rate. We
- know that

$$T_{fr} = \text{(frame size)} / \text{(data rate)} = 2 \times T_p = 2 \times \text{distance} / \text{(propagation speed)}$$

or

$$\text{(frame size)} = [2 \times \text{(distance)} / \text{(propagation speed)}] \times \text{(data rate)}]$$

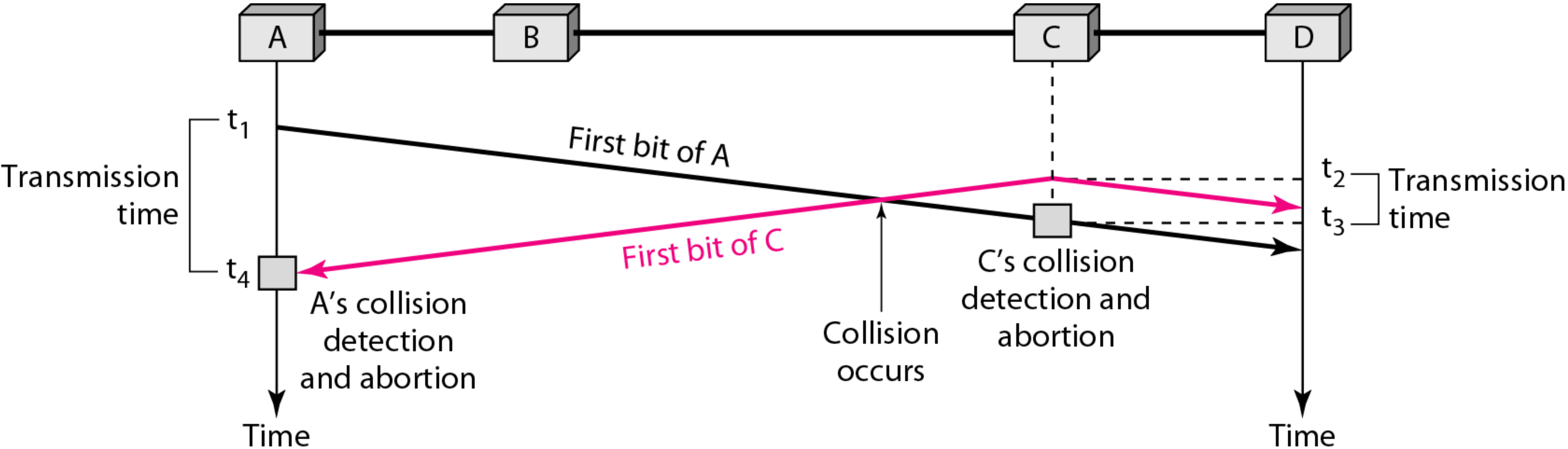
or

$$\mathbf{(frame\ size) = K \times (data\ rate)}$$

This means that minimum frame size is proportional to the data rate (K is a constant).

- When the data rate is increased, the frame size must be increased in a network with a fixed length to continue the proper operation of the CSMA/CD.
- We calculate the minimum frame size based on the above proportionality relationship
- Data rate = 10 Mbps → minimum frame size = **512 bits**
- Data rate = 100 Mbps → minimum frame size = **5120 bits**
- Data rate = 1 Gbps → minimum frame size = **51,200 bits**
- Data rate = 10 Gbps → minimum frame size = **512,000 bits**

Collision of the first bit in CSMA/CD



In the above figure, the data rate is 10 Mbps, the distance between station A and C is 2000 m, and the propagation speed is 2 x 10^8 m/s. Station A starts sending a long frame at time t1 =0; station C starts sending a long frame at time t2 =3μs. The size of the frame is long enough to guarantee the detection of collision by both stations. Find:

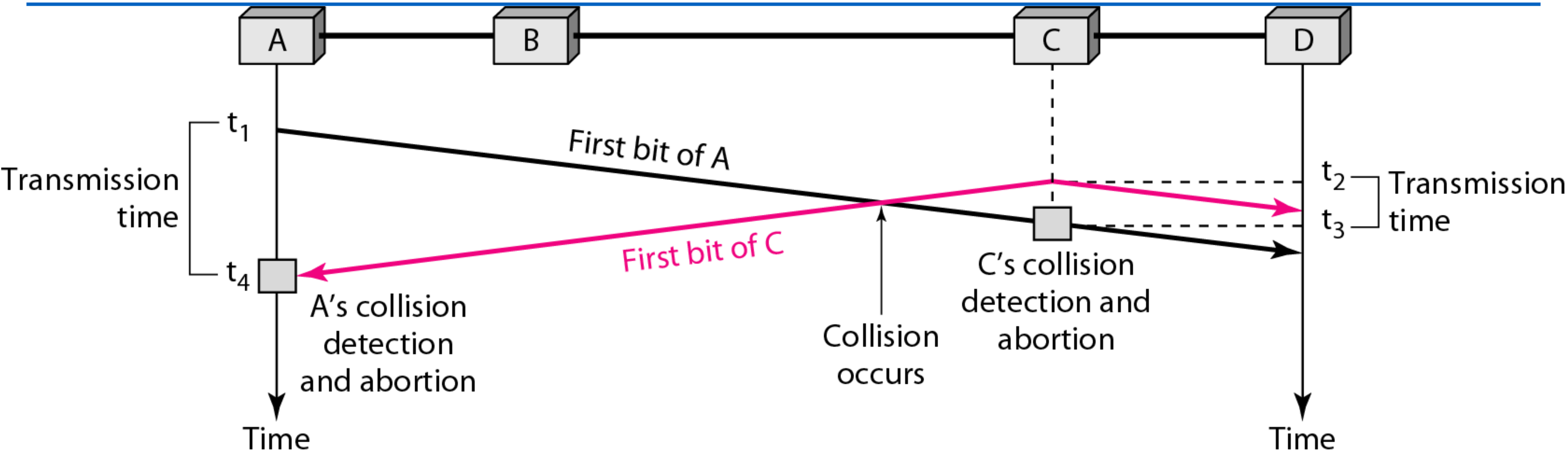
a. The time when station C hears the collision (t3)'

b. The time when station A hears the collision (t4)'

c. The number of bits station A has sent before detecting the collision.

d. The number of bits station C has sent before detecting the collision.

Collision of the first bit in CSMA/CD



We have t1 = 0 and t2 = 3 μs

a. $t3 - t1 = (2000\ m) / (2 \times 10^8\ m/s) = 10\ \mu s \rightarrow t3 = 10\ \mu s + t1 = 10\ \mu s$

b. $t4 - t2 = (2000\ m) / (2 \times 10^8\ m/s) = 10\ \mu s \rightarrow t4 = 10\ \mu s + t2 = 13\ \mu s$

c. $T_{fr(}A) = t4 - t1 = 13 - 0 = 13\ \mu s \rightarrow BitsA = 10\ Mbps \times 13\ \mu s = 130$ bits

d. $T_{fr}(C) = t3 - t2 = 10 - 3 = 07 \mu s \rightarrow BitsC = 10\ Mbps \times 07\ \mu s = 70$ bits.

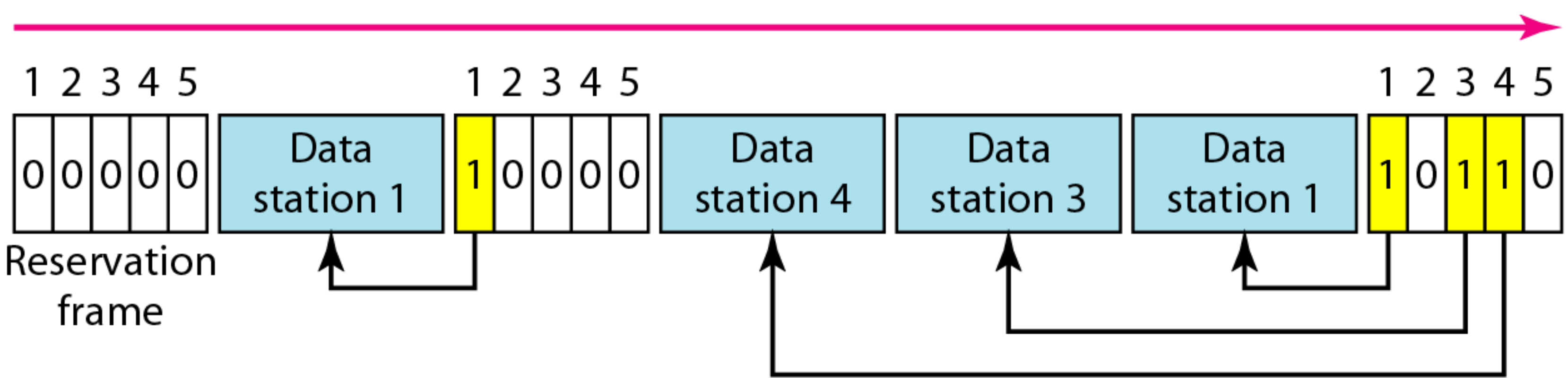
13.2 Controlled Access or Scheduling (reservation and Polling only for reference)

- Provides **in order access** to shared medium so that every station has chance to transfer (**fair protocol**)

- ***Eliminates*** <u>collision completely</u>
- **Three methods** for controlled access:
  - Reservation
  - Polling
  - Token Passing

1-Reservation access method

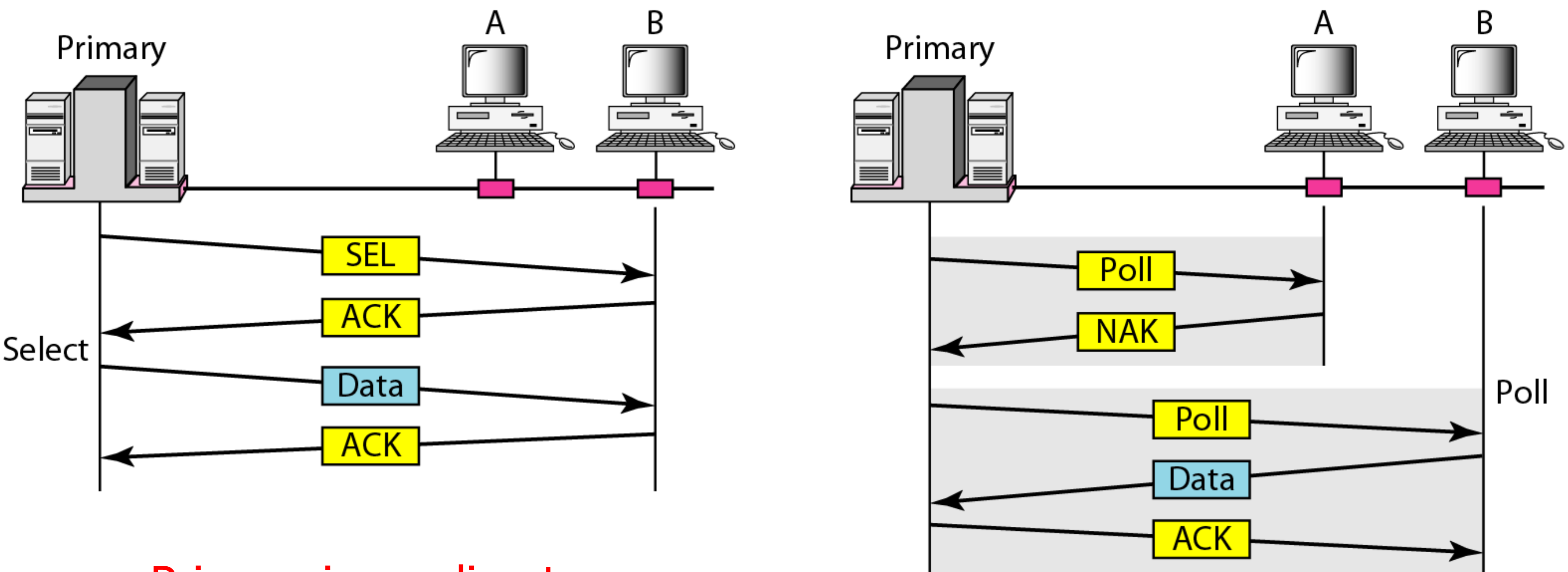
- Stations **take turns transmitting a single frame** at a **full rate (R) bps**
- Transmissions are organized into variable length cycles
- Each cycle begins with a reservation interval that consists of (N) minislots. One minislot for each of the N stations
- When a station needs to send a data frame, it makes a **reservation** in its own minislot.
- By listening to the reservation interval, every station knows which stations will transfer frames, and in which order.
- The stations that made reservations can send their data frames after the reservation frame.



2- Polling

- Stations take turns accessing the medium
- Two models: **Centralized** and **distributed** polling
- **Centralized polling**
  - One device is assigned as **primary station** and the others as **secondary stations**
  - All data exchanges are done through the **primary**
  - When **the primary has a frame to send** it sends a **select** frame that includes the address of the intended secondary
  - When **the primary is ready to receive** data it send a **Poll** frame for each device to ask if it has data to send or not. If yes, **data** will be transmitted otherwise **NAK** is sent.
  - Polling can be done <u>in order</u> (Round-Robin) or based on <u>predetermined order</u>
- **Distributed polling**
  - **No primary and secondary**
  - Stations have a **known polling order** list which is made based on some protocol
  - **station with the highest priority** will have the access right first, then it passes the access right to the **next station (it will send a pulling message to the next station in the pulling list)**, which will passes the access right to the following next station, …

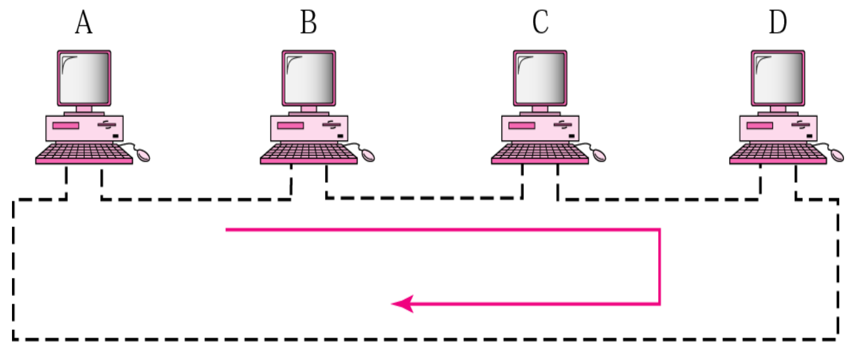
Figure 12.19 *Select and poll functions in polling access method*



Primary is sending to Secondary

Secondary is sending to Primary

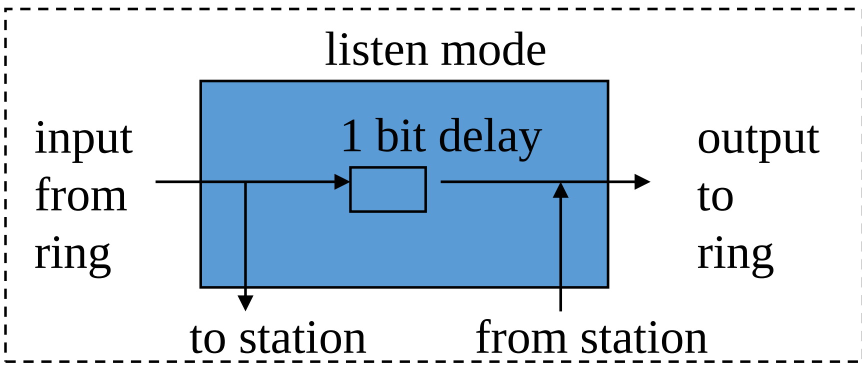
3- Token-Passing network

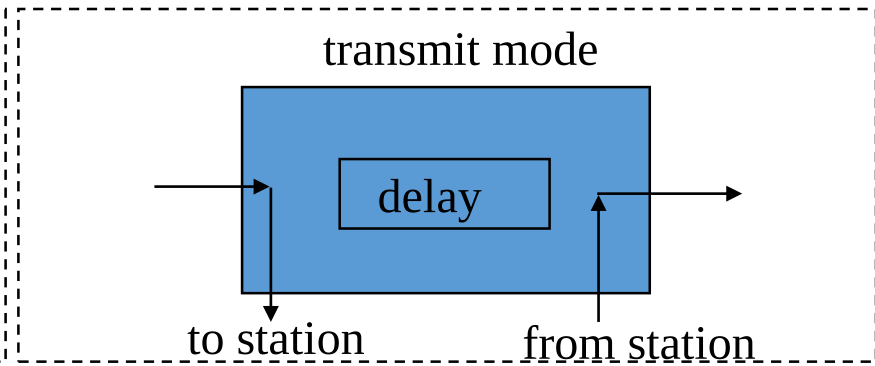


Implements Distributed Polling System

bits are copied to the output bits with a one bit delay

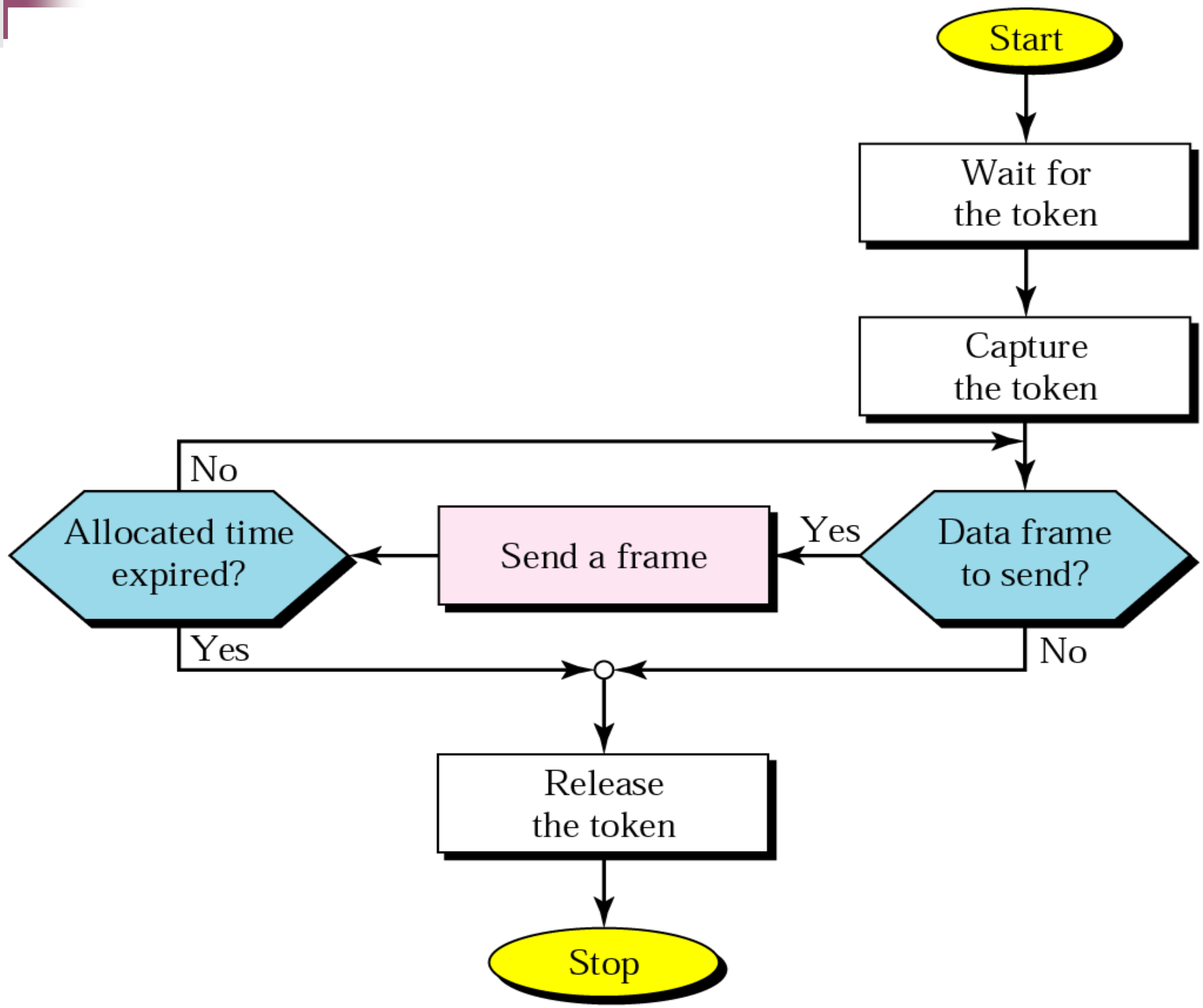
Bits are inserted by the station





- **Station Interface is in two states**:
  - **Listen state**: Listen to the arriving bits and check the destination address to see if it is its own address. If yes the frame is copied to the station otherwise it is passed through the output port to the next station.
  - **Transmit state**: station captures a special frame called **free token** and transmits its frames. **Sending** station is responsible for **reinserting** the free token into the ring medium and for **removing** the transmitted frame from the medium.

Figure 13.13 Token-passing procedure

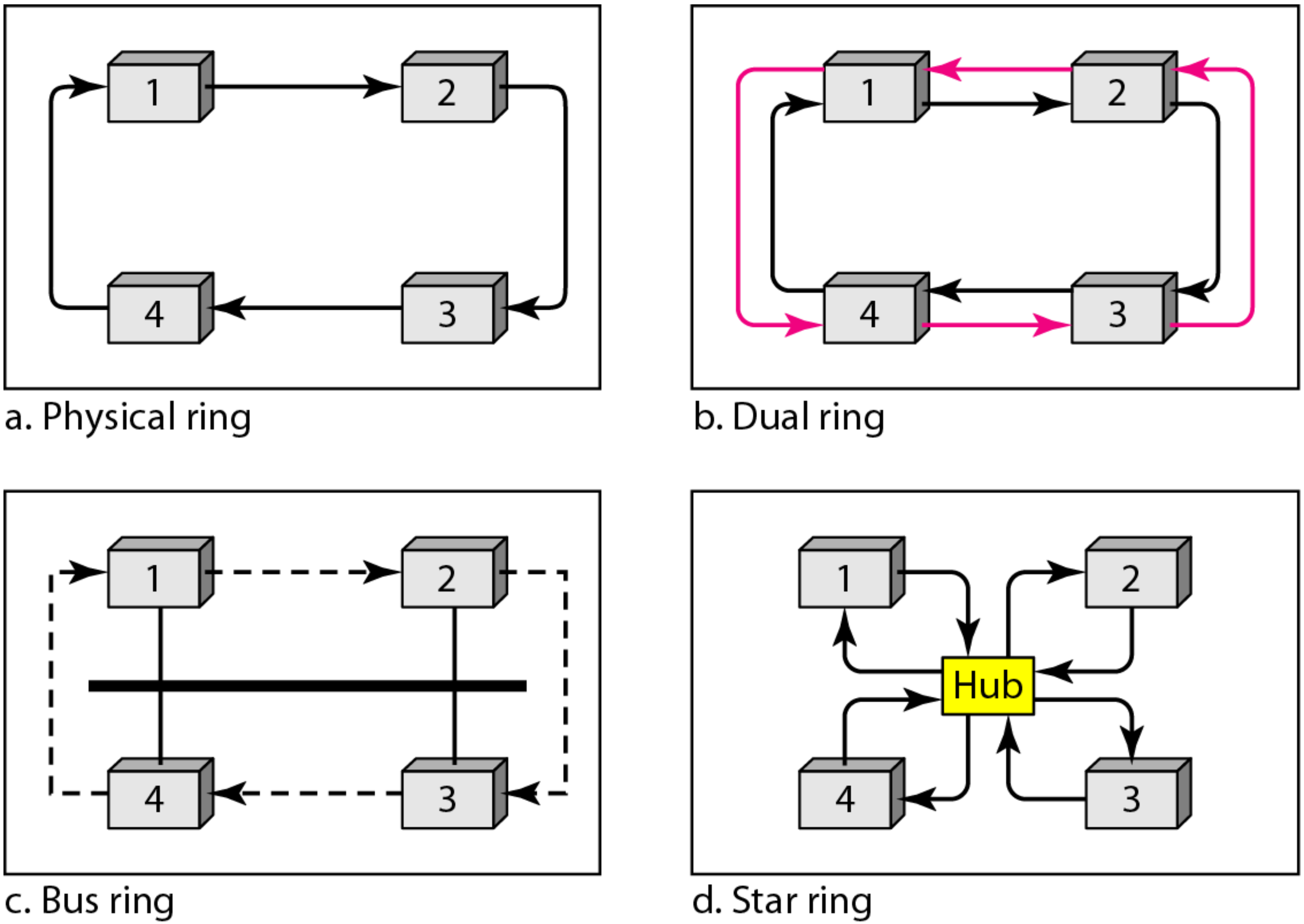


TOKEN PASSING

- 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.
- In this process, when a station receives the token and has no data to send, it just passes the token to the next station
- **Token management** is needed for this access method.
- Stations must be limited in the time they can have possession of the token. The token must be monitored to ensure it has not been lost or destroyed.
- And finally, token management is needed to make low-priority stations release the token to high priority stations.

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



- 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.
- Note that for this topology to work, each station needs to have two transmitter ports and two receiver ports.

- 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
- 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.

Syllabus

- Alberto Leon Garcia – “ Communuication Networks” – 6.1
- Chapter 12 of “Data Communications and Networking” by Forouzan.