Module 4

Data Link Layer

Medium Access Control

BECE401L

Outline

Random access Protocols – Ethernet (IEEE 802.3) – Wireless LAN (IEEE 802.11);

Scheduling approaches to MAC – Controlled Access – Token Bus/Ring (IEEE 802.4/5)

Why Medium Access Control ?

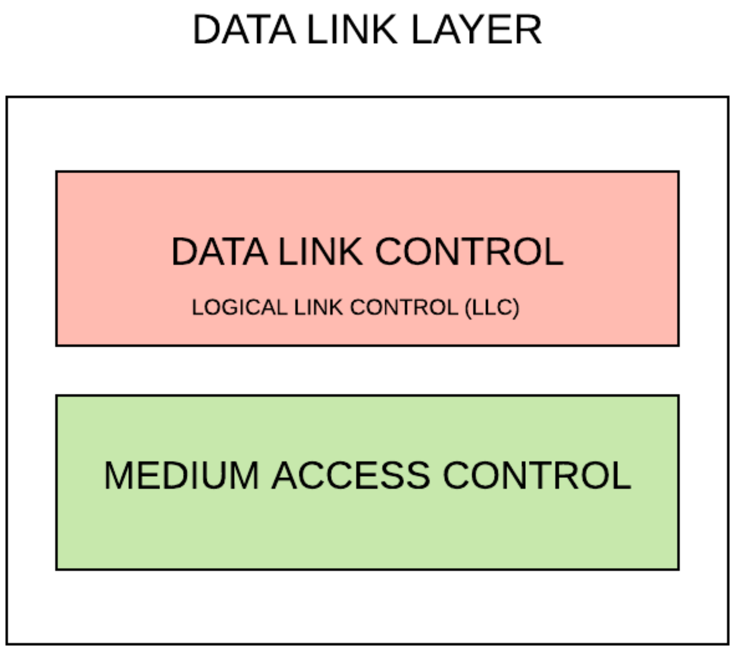
- **Shared Medium (wired / wireless)**
- **Who accesses the medium when ?**
  - Example: Wired LAN, Wireless LAN, Satellite etc.,
  - Real world example: ***Group discussions***
    - The procedures guarantee that the right to speak is upheld and ensure that two people do not speak at the same time, do not interrupt each other, do not monopolize the discussion, and so on.
- **When two stations transmit simultaneously,**
  - The data transmission is lost (collision)
  - Bandwidth wasted
- **Transmissions among different stations needs to be coordinated via Protocols**
- These protocols are called as ***medium access control*** protocols or ***multiple access control protocols***
  - Defined as a sub layer of the Data Link Layer (DLL)

Medium Access Control Sub-layer

Medium Access Control is a sub-layer of Data Link Layer (Layer 2)

DLC: Responsible for frame and error control

MAC: Responsible for framing, MAC address, multiple access control

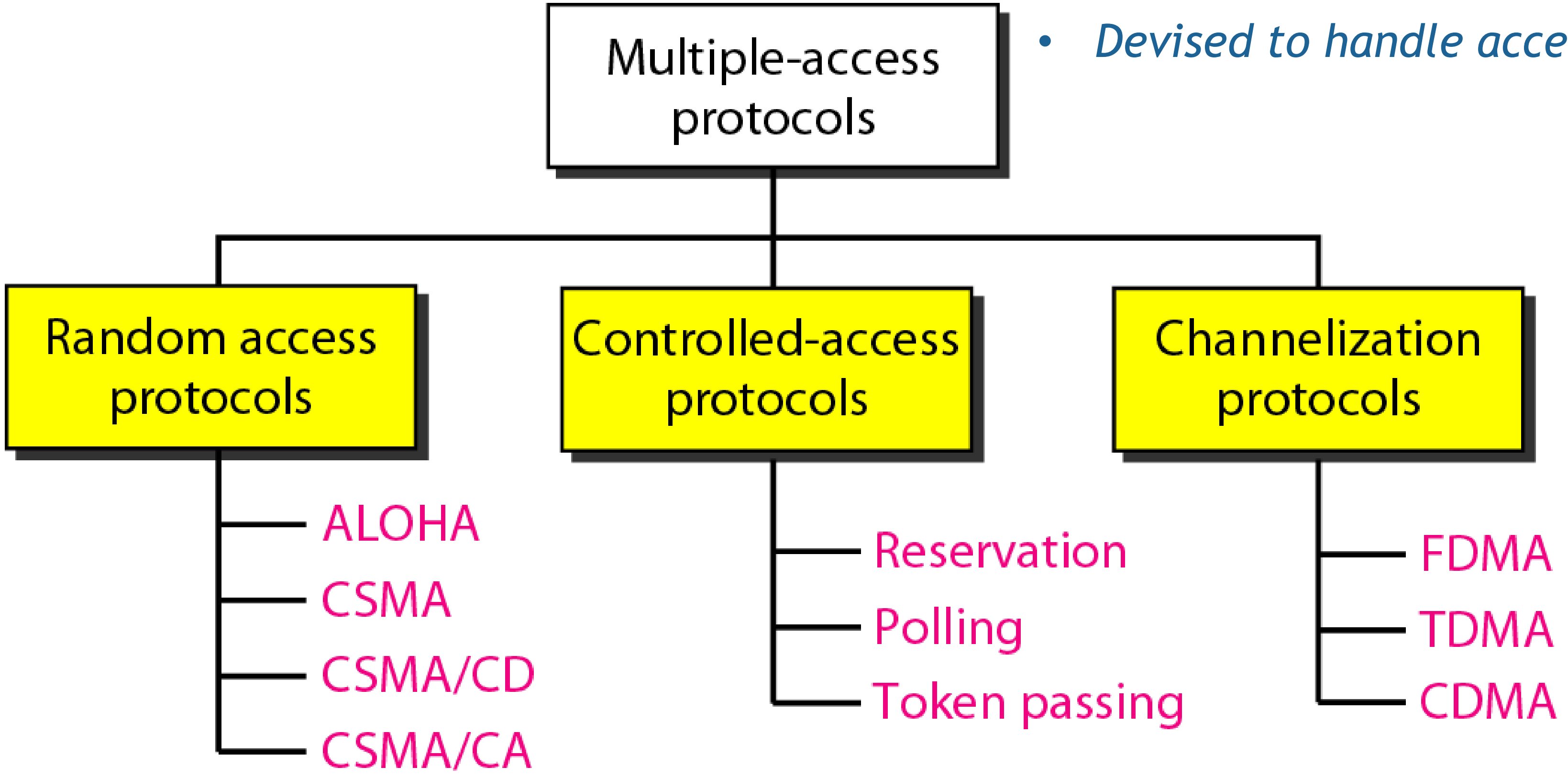


Main Task of MAC Layer

- To **maximize the utilization** of the resource (**bandwidth**) by **reducing the collisions**
- How?
  - Determine when to transmit
  - What to be done if the link is busy
  - What to be done when collision occurs

MAC Protocols

When nodes or stations are connected and use a common link, called a *multipoint* or *broadcast link*, multiple-access protocol to coordinate access to the link.



- *Devised to handle access to a shared link*

Random Access Protocols

- In random access or contention methods,
  - No station is superior to another station and none is assigned the control over another.
  - No station permits, or does not permit, another station to send.
  - 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*.

- **Features:**
  - ***Random access:***
    - There is no scheduled time for a station to transmit.
  - ***Contention Methods:***
    - Stations compete with one another to access the medium.

Random Access Protocols

- 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*.

- To ***avoid access conflict*** or to resolve it when it happens, each station follows a procedure that answers the following questions:
  - When can the station access the medium?
  - What can the station do if the medium is busy?
  - How can the station determine the success or failure of the transmission?
  - What can the station do if there is an access conflict?

Random Access Protocols

Protocols:

- ALOHA:
  - Which used a very simple procedure called multiple access (MA).
- Carrier Sense Multiple Access (CSMA)
  - ALOHA was improved with the addition of a procedure that forces the station to sense the medium before transmitting.
- Carrier Sense Multiple Access with Collision Detection (CSMA/CD)
  - Which tells the station what to do when a collision is detected,
- Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA)
  - Which tries to avoid the collision

Random Access Protocols

Pure ALOHA

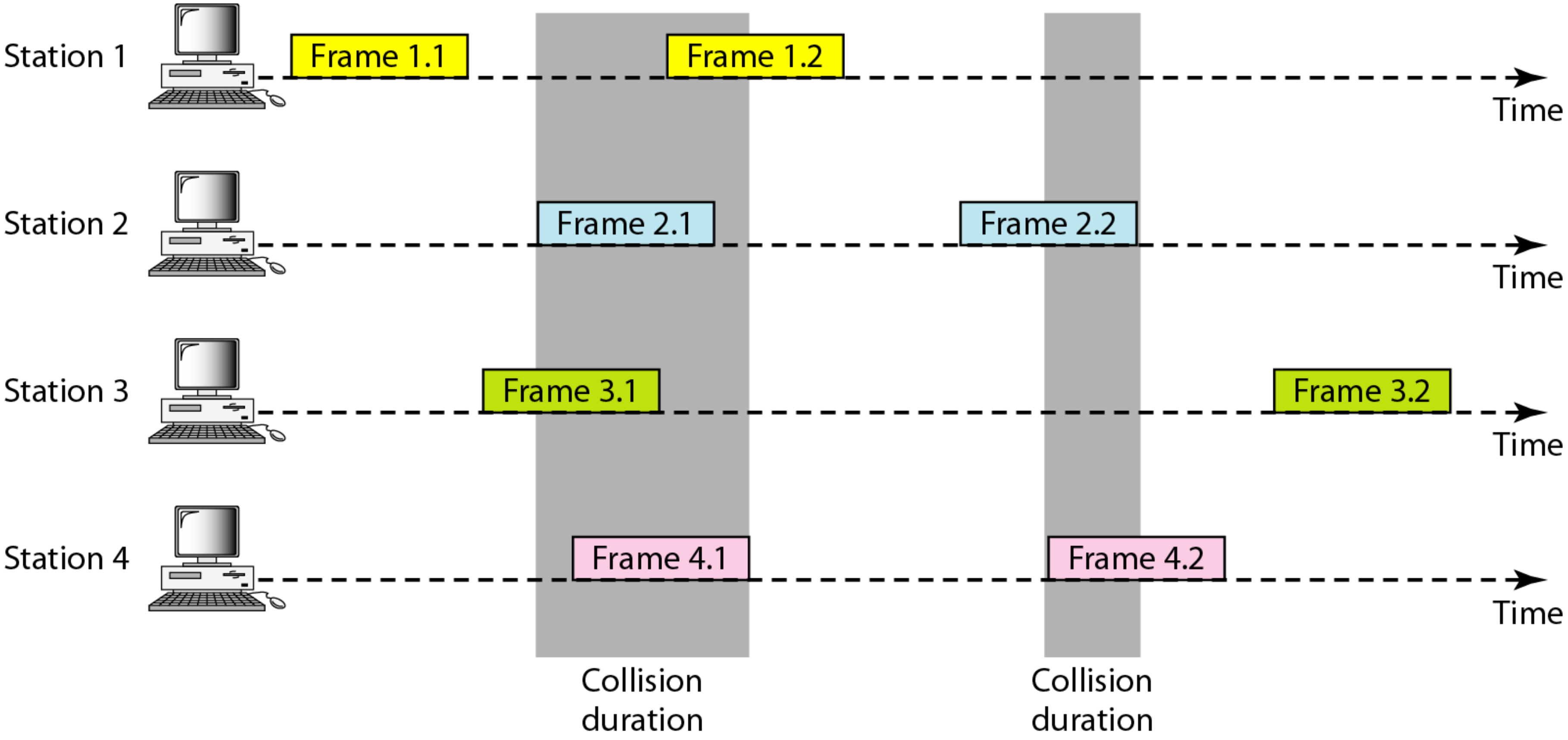
ALOHA, the earliest random access method, developed at the University of Hawaii in early 1970. Designed for a radio (wireless) LAN, but it can be used on any shared medium.

Each station sends a frame whenever it has a frame to send (multiple access).

Collisions can happen.

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.

Relies on acknowledgments from the receiver.



Pure ALOHA

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.

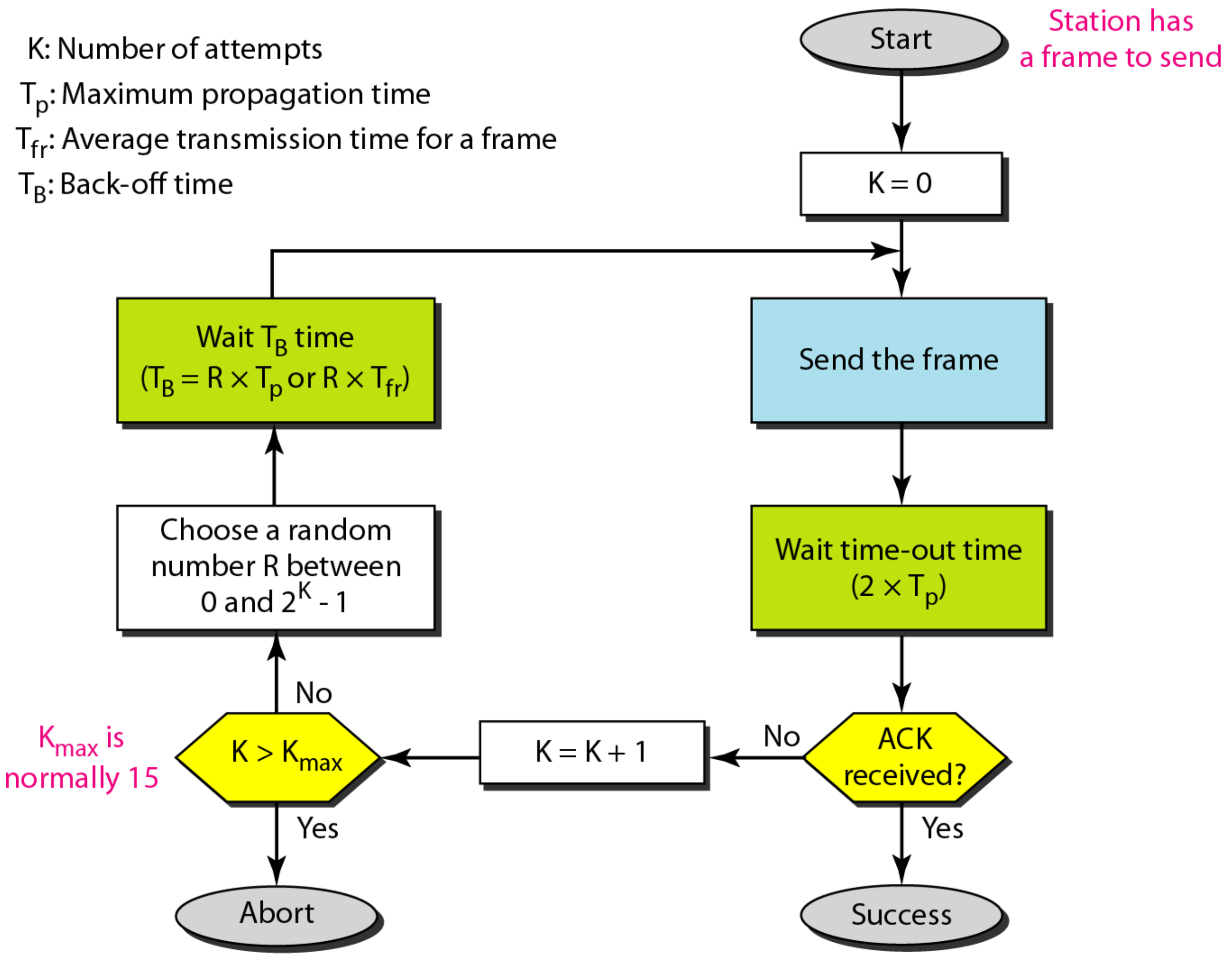
Called as: backoff time TB.

Random Access Protocols

Pure ALOHA: Procedure

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 time-out period is equal to the maximum possible round-trip propagation delay, 2 × *Tp*.
- The backoff time *TB* is a random value that normally depends on K.
  - One common formula is the ***binary exponential backoff***.
  - *F*or each retransmission, a multiplier $R = 0$ to $2^K - 1$ is randomly chosen and multiplied by *Tp* or *Tfr*


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 $R \times T_{fr}$)

Pure ALOHA: Example

The stations on a wireless ALOHA network are a maximum of 600 km apart. If we assume that signals propagate at 3 × 10^8 m/s, we find

Tp = (600 × 10^3) / (3 × 10^8) = 2 ms.

Now we can find the value of T_B for different values of K .

Solution

a) **For K = 1,** the range is {0, 1}. The station needs to generate a random number with a value of 0 or 1. This means that **T_B is either 0 ms** (0 × 2) **or 2 ms** (1 × 2), based on the outcome of the random variable.

b) **For K = 2,** the range is {0, 1, 2, 3}. This means that **T_B can be 0, 2, 4, or 6 ms**, based on the outcome of the random variable.

c) **For K = 3,** the range is {0, 1, 2, 3, 4, 5, 6, 7}. This means that **T_B can be 0, 2, 4, ... , 14 ms**, based on the outcome of the random variable.

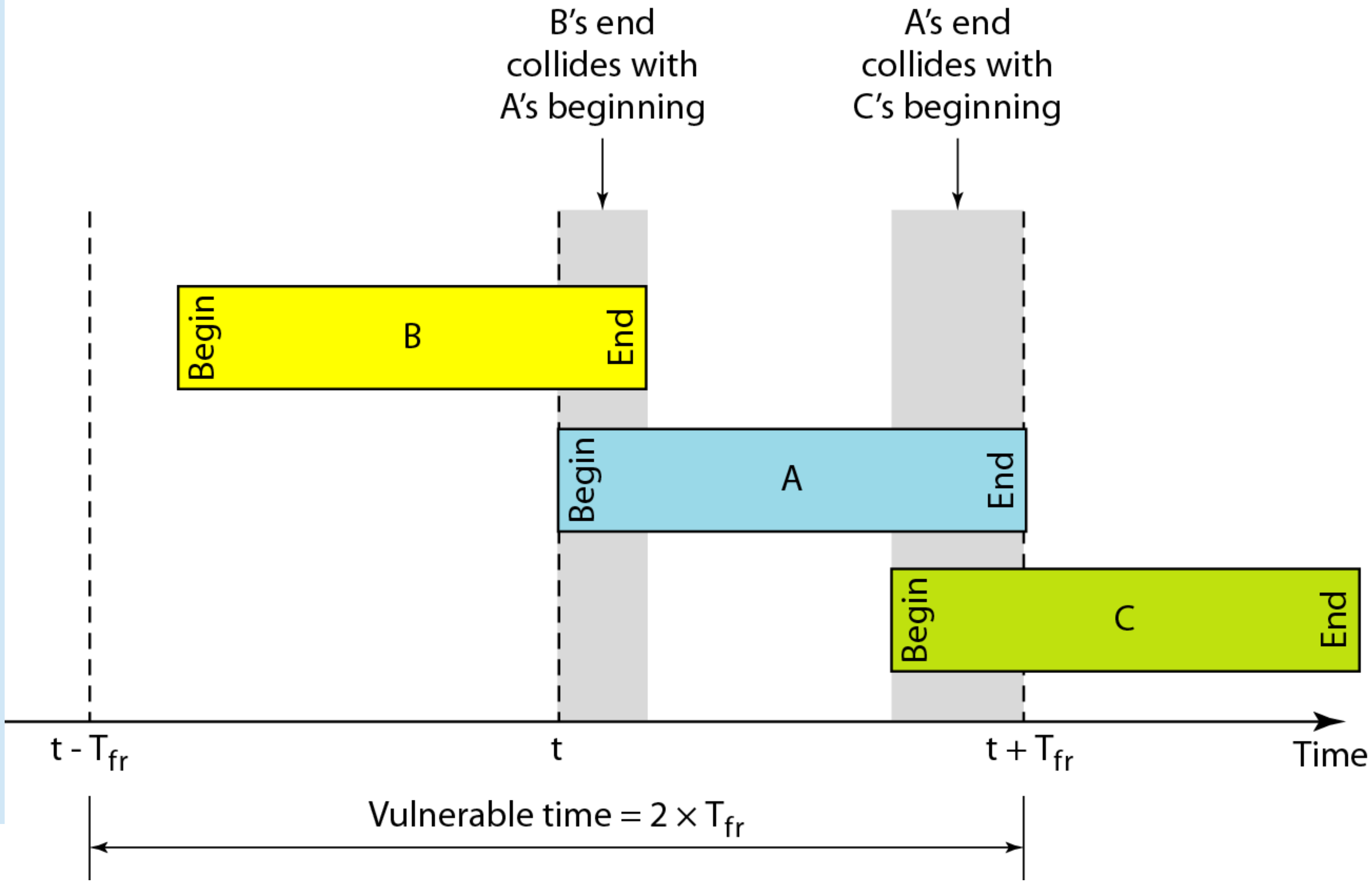
Random Access Protocols

Pure ALOHA: Vulnerable Time

The length of time in which there is a possibility of collision.

Figure shows the **vulnerable time for station B.**

1. **Station B** starts to send a frame at time *t*.
2. Now imagine **station A** has started to **send** its **frame after *t – Tfr***.
   - ***This leads to a collision between the frames from station B and station A.***
3. On the other hand, suppose that **station C starts to send a frame before time *t + Tfr***.
   - Here, there is also a collision between frames from station B and station C.
4. Vulnerable time during **which a collision may occur in pure ALOHA is 2 times the frame transmission time.**



B's end collides with A's beginning
A's end collides with C's beginning
Begin
B
End
Begin
A
End
Begin
C
End
$t - T_{fr}$
t
$t + T_{fr}$
Time
Vulnerable time = $2 \times T_{fr}$

Pure ALOHA: Vulnerable Time

Example:
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?

Solution

- **Average frame transmission time T_{fr} is 200 bits/200 kbps or 1 ms.**
- The **vulnerable time is 2 × 1 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: Throughput

Let, *G* the average number of frames generated by the system during one frame transmission time.

The **throughput for pure ALOHA** is

$S = G \times e^{-2G}$.

The maximum throughput $S_{max} = 0.184$ when G= (1/2).

Which means,

- If one-half a frame is generated during one frame transmission, then 18.4 percent of these frames reach their destination successfully.
- It is expected *G* = 1/2 to produce the maximum throughput because the vulnerable time is 2 times the frame transmission time.
- Therefore, if a station generates only one frame in this vulnerable time (and no other stations generate a frame during this time), **the frame will reach its destination successfully.**

Random Access Protocols

Pure ALOHA: Throughput

Example:
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) The system creates 1000 frames per second, this is 1 frame per ms.

- **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 × 0.135 = 135 frames.**

- **Only 135 frames out of 1000 will probably survive**.

Random Access Protocols

Pure ALOHA: Throughput

Example:
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

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.

Pure ALOHA: Throughput

Example:
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

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 × 0.152 = 38.
- **Only 38 frames out of 250 will probably survive**.

Random Access Protocols

Slotted ALOHA

Slotted ALOHA

- Pure ALOHA has a vulnerable time of 2 × T_{fr}.
  - As there is no rule that defines when the station can send.
- Slotted ALOHA was invented to improve the efficiency of pure ALOHA.

In slotted ALOHA

- Divide the time into slots of T_{fr} seconds and
- Force the station to send only at the beginning of the time slot.
- As 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.

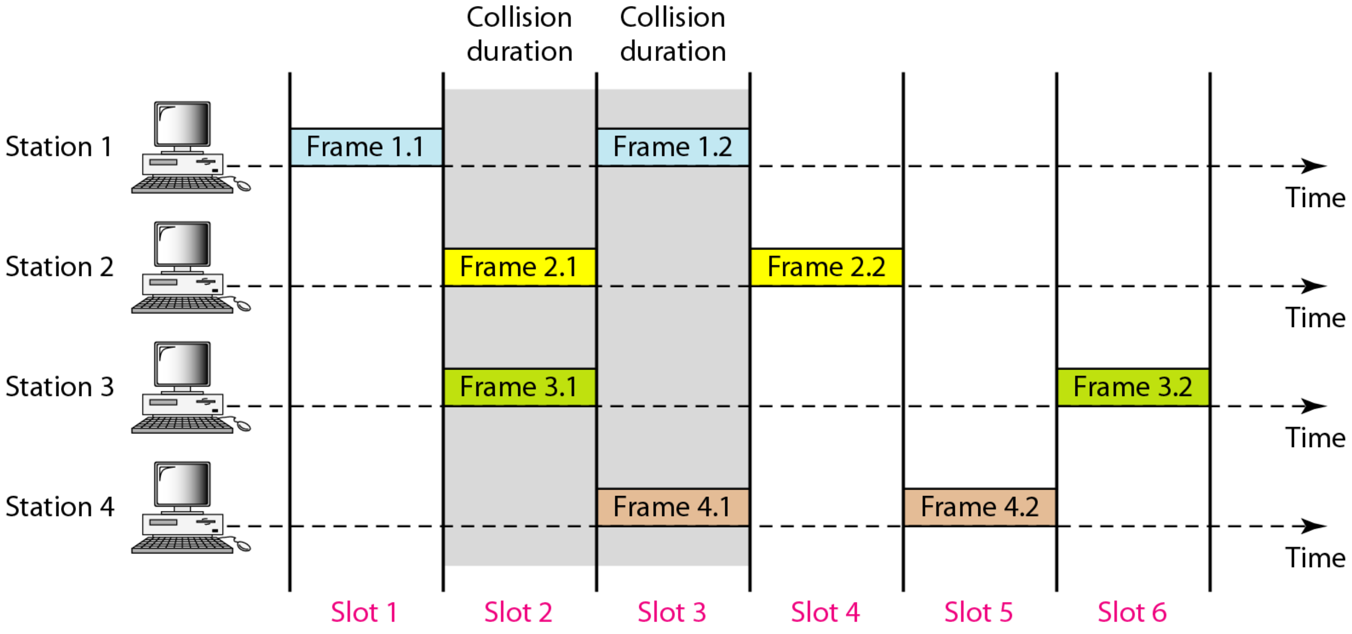
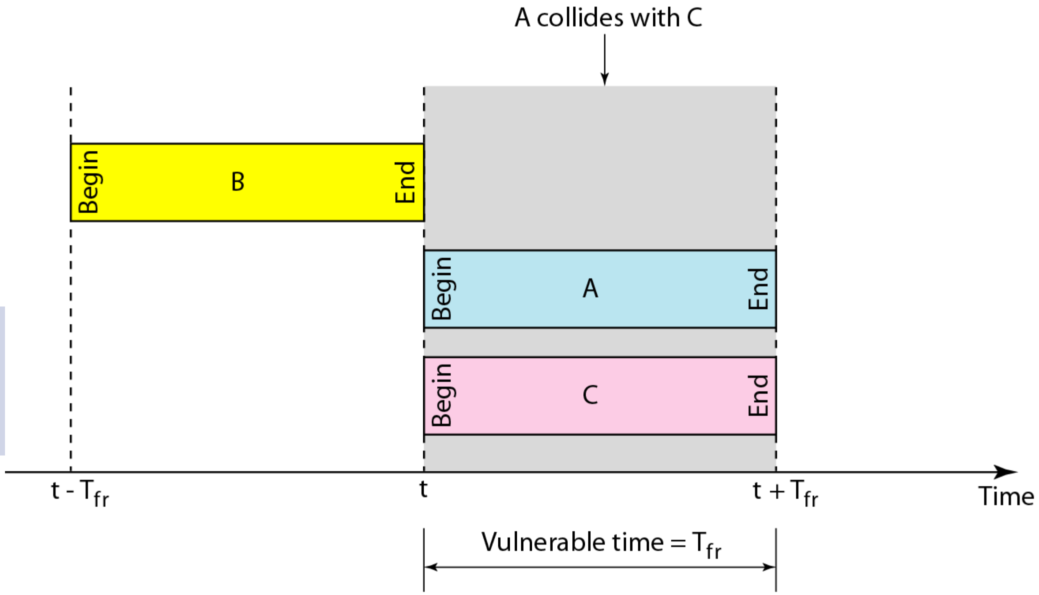


Figure shows an example of frame collisions in slotted ALOHA.

Random Access Protocols

Slotted ALOHA

- However, the vulnerable time is now reduced to one-half, equal to *Tfr*.
- Figure shows the situation.



Throughput

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

- In other words, if ***one frame is generated during one frame transmission time***, then *36.8 percent of these frames reach their destination successfully*.

- Therefore, if a station generates only one frame in this vulnerable time (and no other station generates a frame during this time), **the frame will reach its destination successfully**.

Slotted ALOHA: Example

Example:

A slotted 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× e^{-G} or S = 0.368 (36.8 percent). This means that the throughput is 1000 × 0.0368 = 368 frames. Only 386 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 × e^{-G} or S = 0.303 (30.3 percent). This means that the throughput is 500 × 0.0303 = 151. Only 151 frames out of 500 will probably survive.

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 × e^{-G} or S = 0.195 (19.5 percent). This means that the throughput is 250 × 0.195 = 49. Only 49 frames out of 250 will probably survive.

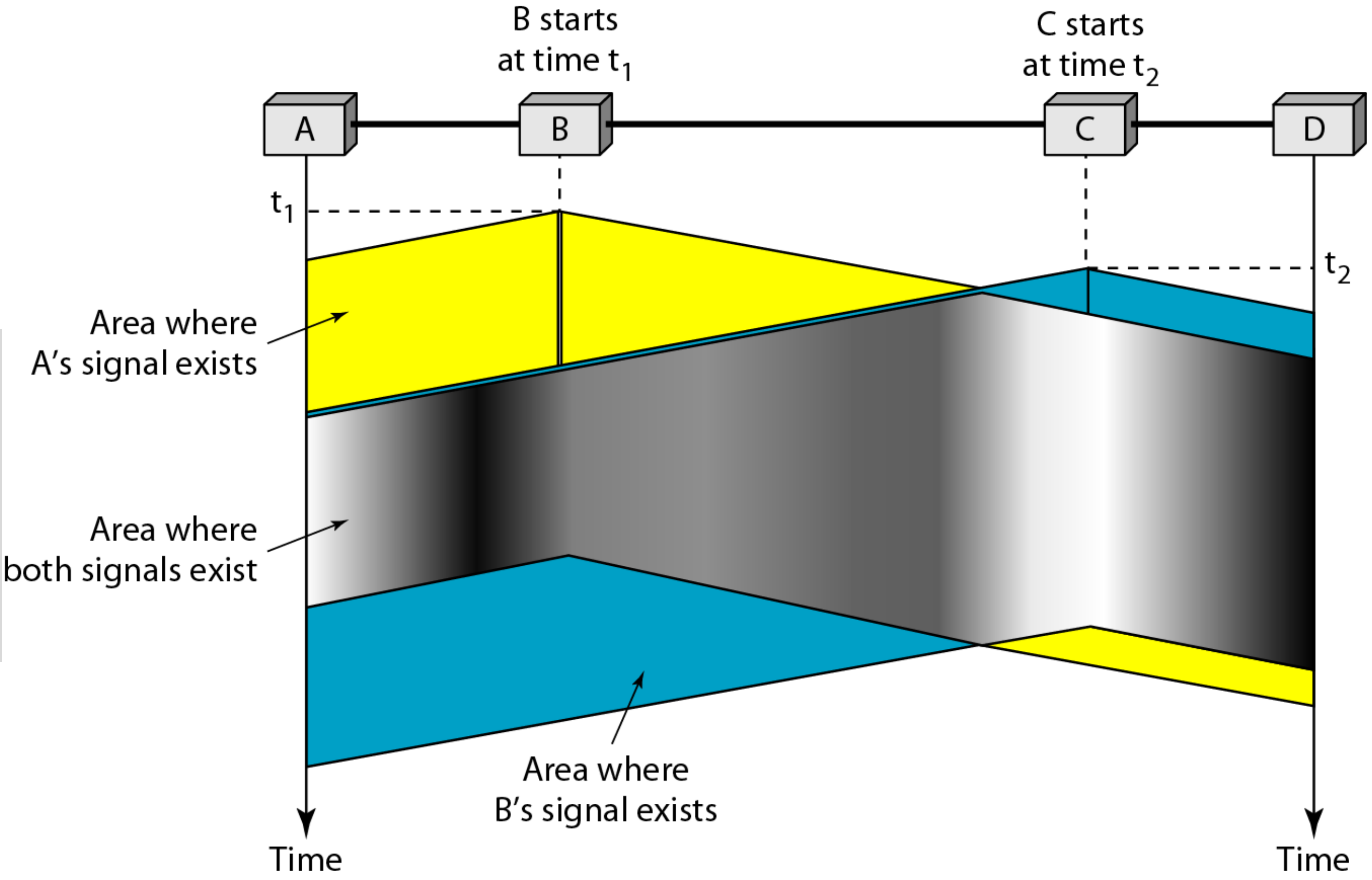
Carrier Sense Multiple Access (CMSA)

CSMA was developed:

- To minimize the chance of collision and, therefore, increase the performance.
- Chance of collision can be reduced if a station senses medium before trying to use it.

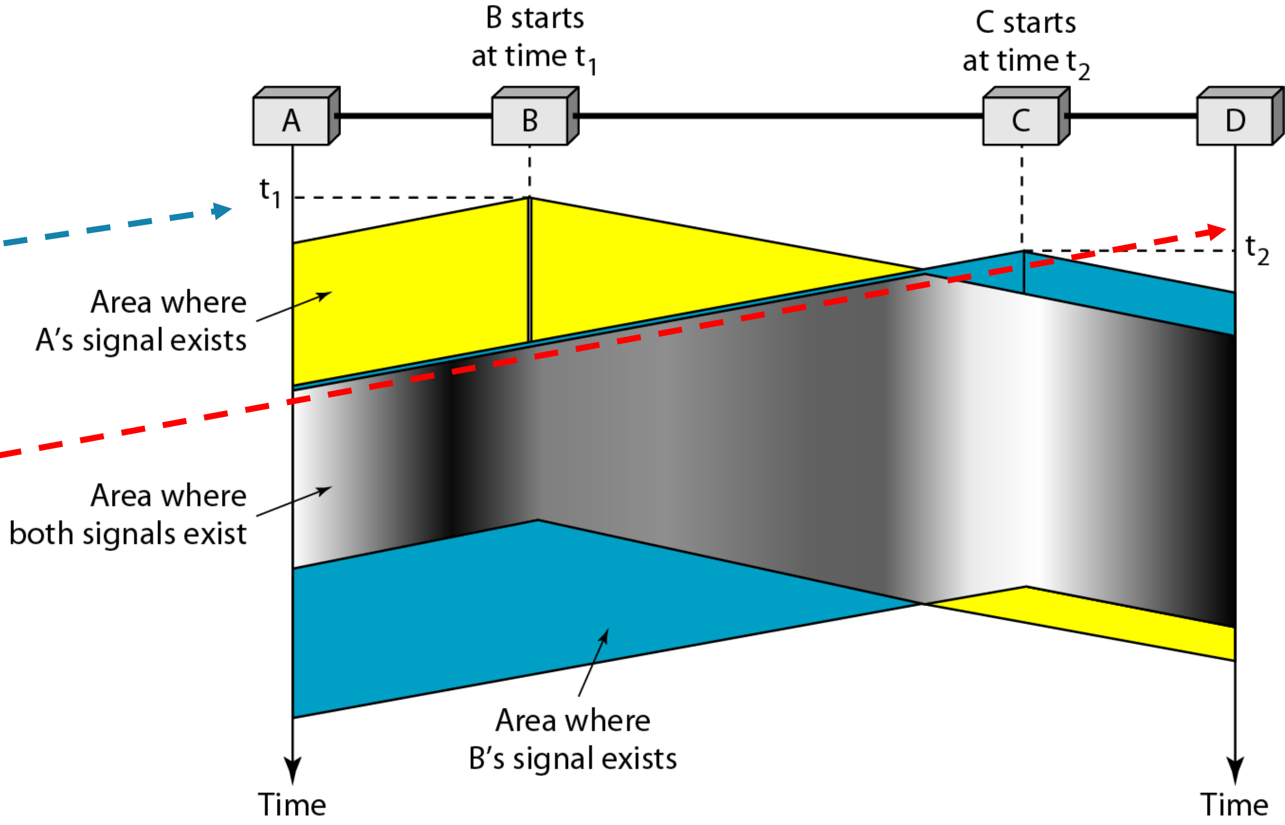
CSMA

- Requires that each station **first listen to the medium** before sending.
- CSMA is based on the principle "sense before transmit" or "listen before talk."



CMSA

- Stations are connected to a shared channel.
- Possibility of collision: Because of **propagation delay**;
  - *When a station sends a frame*
    - It still takes time for the first bit to reach every station and for every station to sense it.
  - *Another 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
  - As, at this time:
    - The first bits from station B have not reached station C.
  - Station C also sends a frame.
    - Two signals collide and both frames are destroyed.



B starts at time t_1
C starts at time t_2
A
B
C
D
t_1
t_2
Area where A's signal exists
Area where both signals exist
Area where B's signal exists
Time
Time

Figure: a space and time model of a CSMA network.

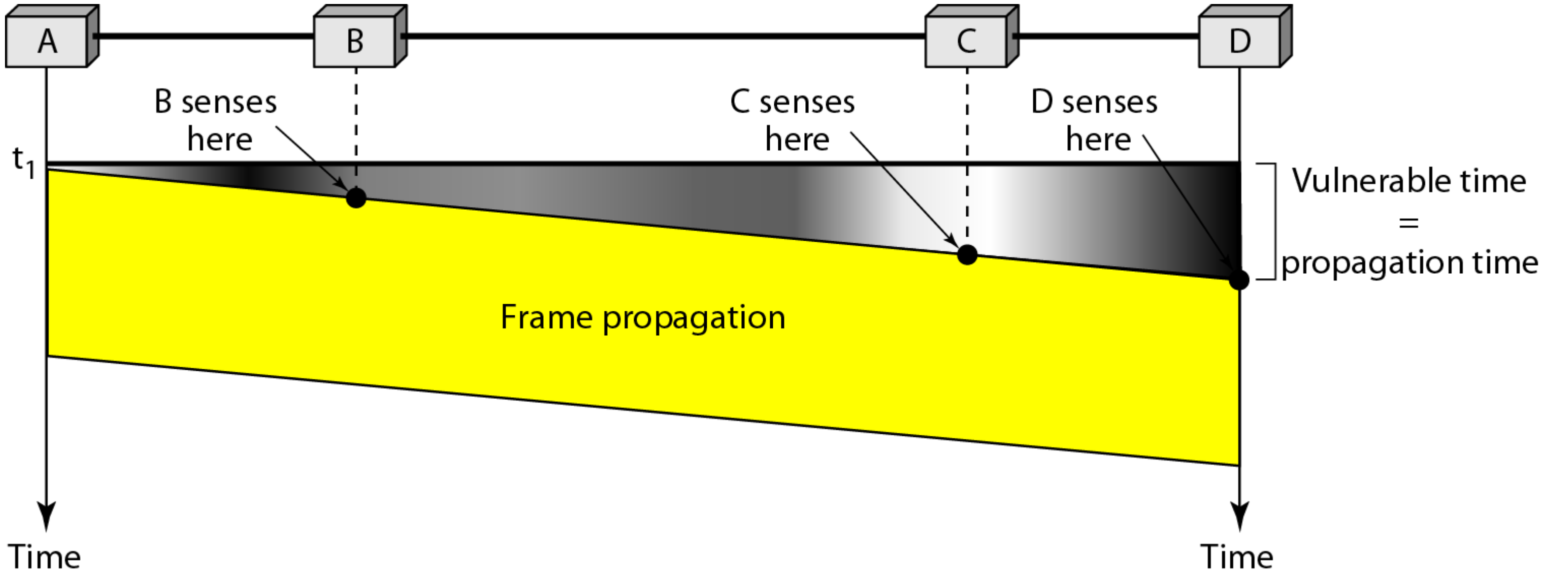
Vulnerable Time in CMSA

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 frame during this time, 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**

Figure shows the worst case.

- The leftmost station, A,
  - Sends a frame at time *t*1,
  - Which reaches the rightmost station, D, at time *t*1 + *Tp*.
  - Gray area shows the vulnerable area in time and space.



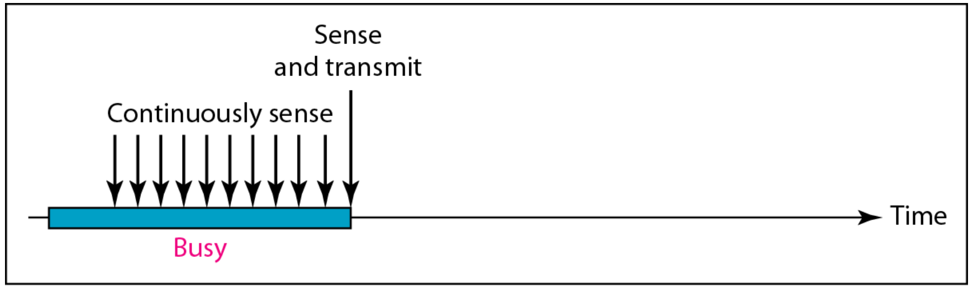
Persistence Time in CMSA

- What should a station do if the channel is busy?
- What should a station do if the channel is idle?

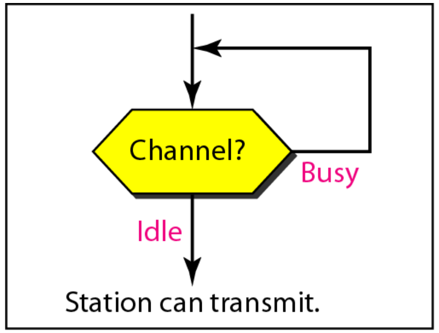
Three methods: 1-persistent method; Nonpersistent method & *p*-persistent method.

1-Persistent (Simple & straightforward)

- Here, *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.
- Used in **ethernet**



a. 1-persistent

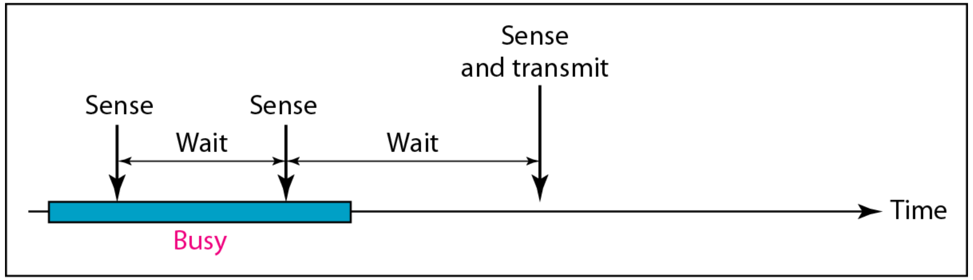


a. 1-persistent

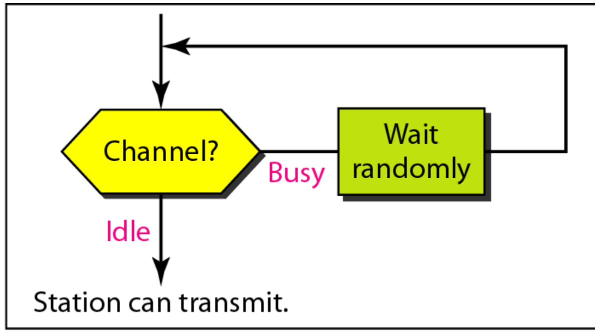
Persistence Time in CMSA

Non-Persistent

- Here 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.
- Reduces the chance of collision
  - As it is unlikely that two or more stations will wait the same amount of time and retry to send simultaneously.
- Reduces the efficiency of the network
  - As the medium remains idle when there may be stations with frames to send.



b. Nonpersistent



b. Nonpersistent

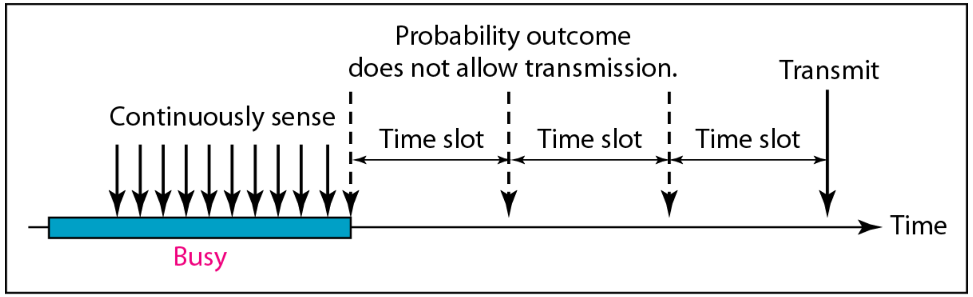
Persistence Time in CMSA

p-Persistent

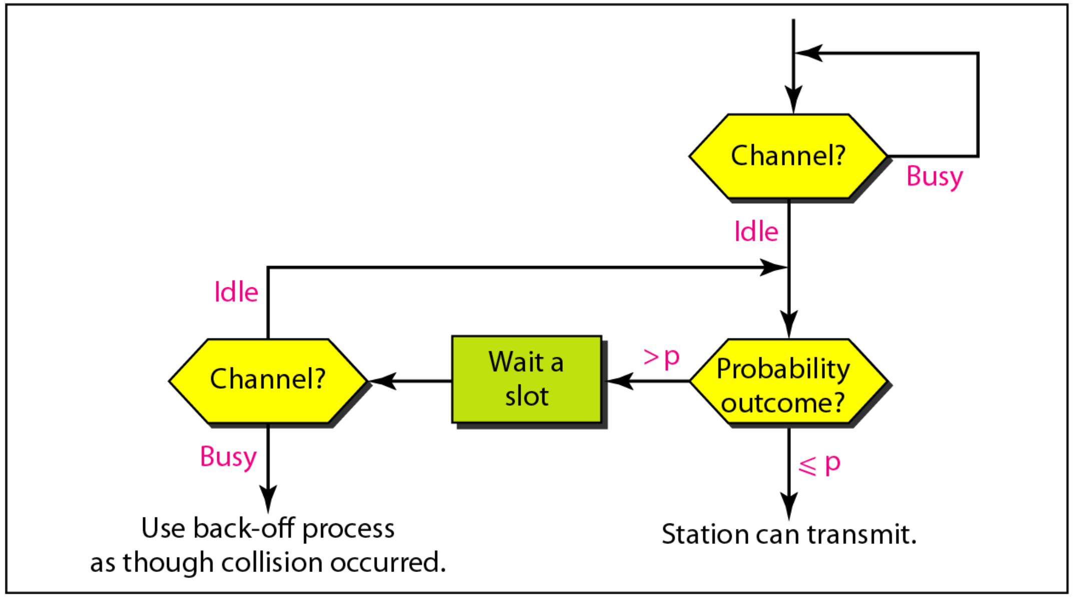
- Method is used if the channel has time slots with a **slot duration equal to or greater than the** maximum propagation time.
- Combines the advantages of the other two strategies.
- Reduces the chance of collision and improves efficiency.

Here, after the station finds line idle it follows:

1. With probability *p*, the station sends its frame.
2. With probability *q* = 1 − *p*,
   - Station waits for the beginning of the next time slot and checks the line again.
     - *If the line is idle,*
       - It goes to step 1.
     - *If the line is busy,*
       - It acts as though a collision has occurred and uses the backoff procedure.



c. p-persistent

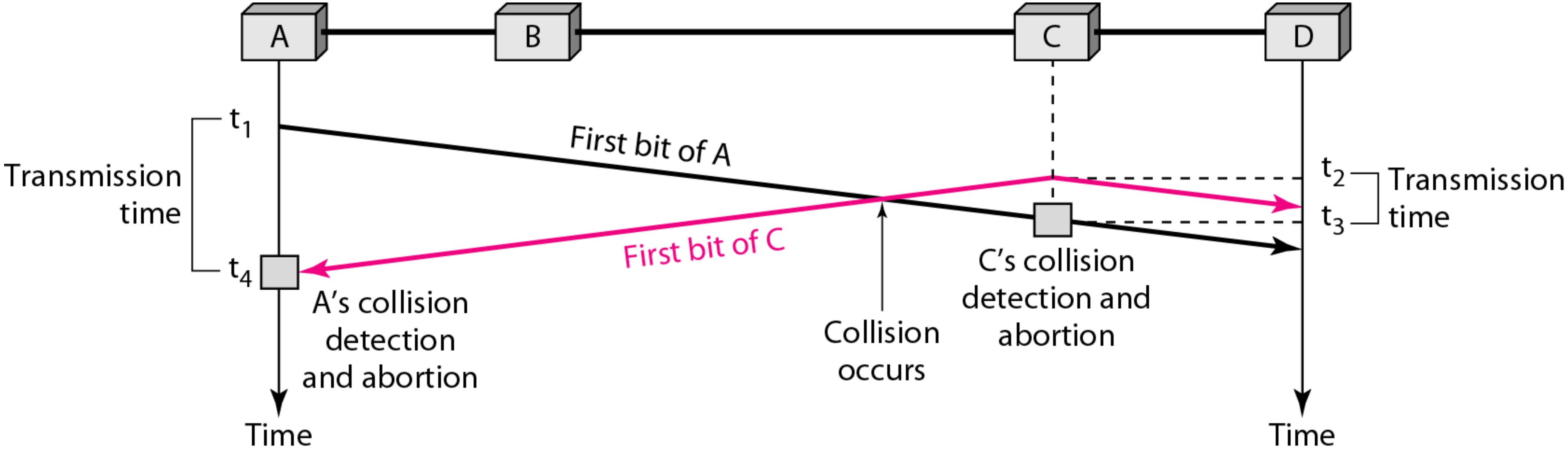


c. p-persistent

CMSA/CD

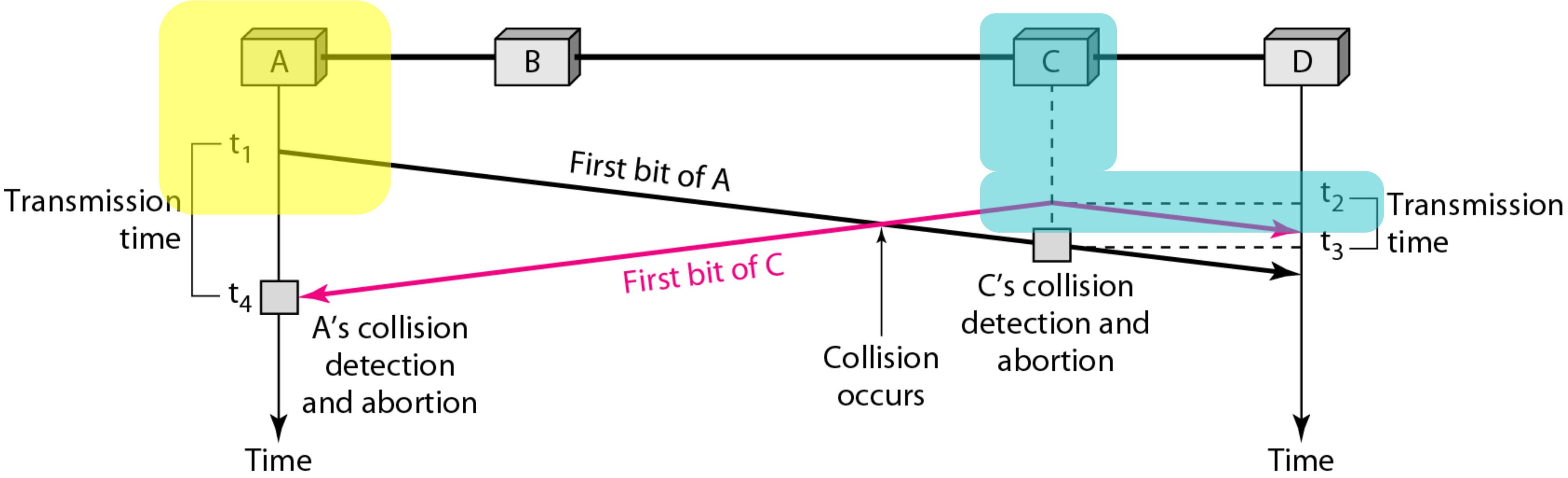
Carrier sense multiple access with collision detection (CSMA/CD).

- 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.



CMSA/CD

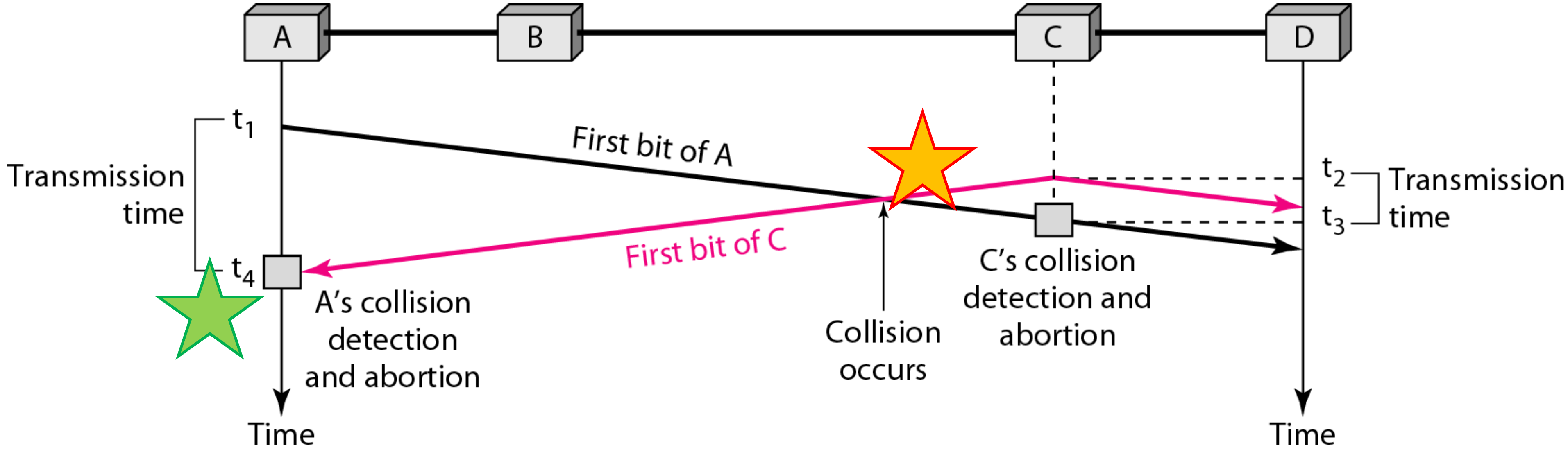
- At time t1:
  - Station A has executed its persistence procedure and starts sending the bits of its frame.
- At time t2:
  - Station C has not yet sensed the first bit sent by A.
  - Station C executes its persistence procedure and
  - Starts sending the bits in its frame,
    - Which propagate both to the left and to the right.



CMSA/CD

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 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.



CMSA/CD Frame Length

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.

- Therefore, the Frame transmission time T_{fr}
  - Must be at least two times the maximum propagation time T_p.

- **Worst case scenario:**
  - If the two stations involved in a collision are the maximum distance apart
  - The signal from the first takes time T_p to reach the second, and
  - The effect of the collision takes another time T_p to reach the first.
  - So the requirement is that the first station must still be transmitting after $2T_p$.

CMSA/CD 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) 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.*

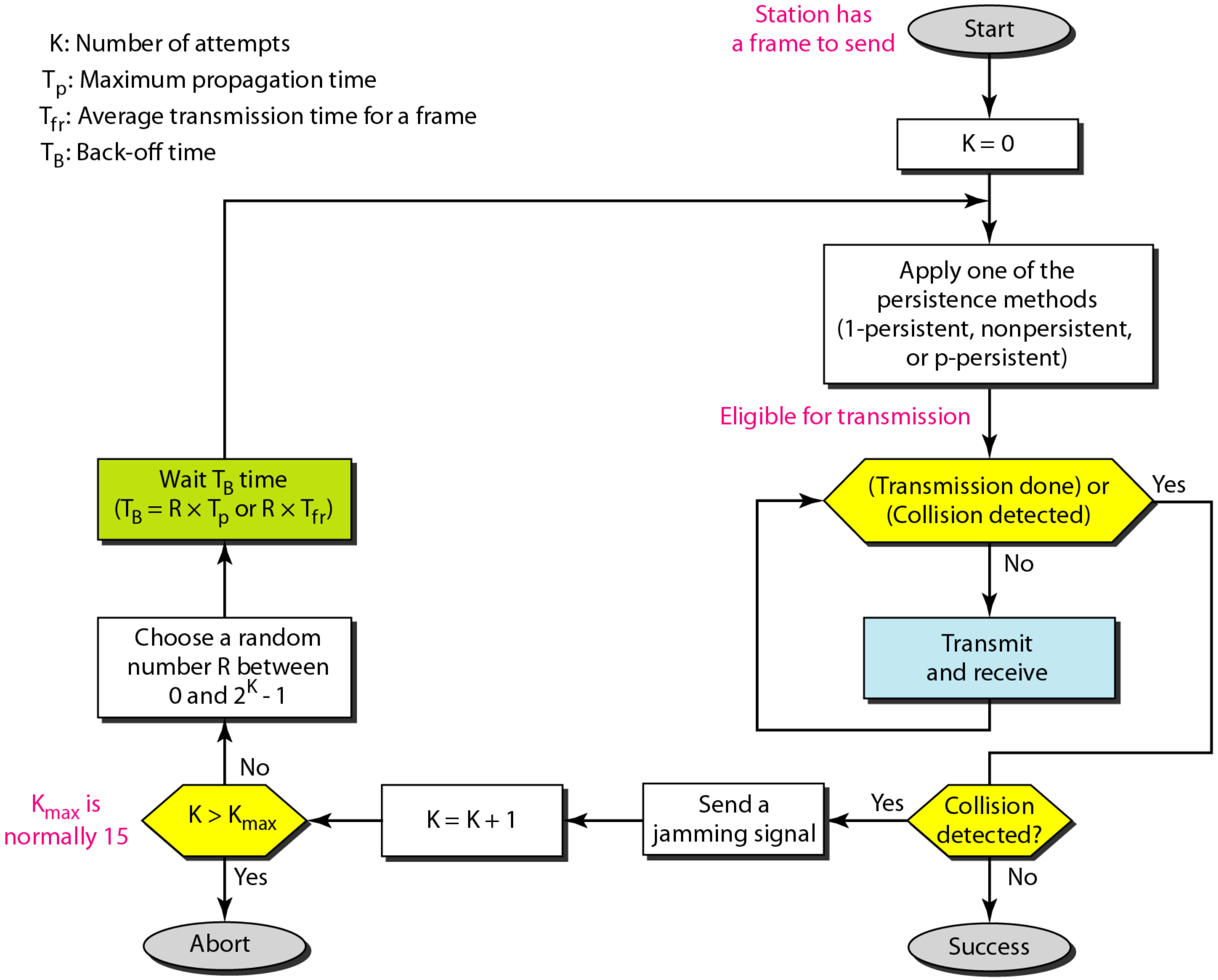
- The minimum size of the frame is 10 Mbps × 51.2 μs = 512 bits or 64 bytes.
- This is actually the *minimum size of the frame for Standard Ethernet*.

CMSA/CD Flow Chart

- Similar to ALOHA protocol with some differences.

- 1st : Addition of the persistence process.
  - Need to sense the channel before we start sending the frame
  - Use **persistent methods** (nonpersistent, 1-persistent, or *p*-persistent).

- 2nd: frame transmission.
  - In ALOHA, we first transmit the entire frame and then wait for an acknowledgment.
  - In CSMA/CD, *transmission and collision detection are continuous processes*.
    - We do not send the entire frame and then look for a collision.
  - The station transmits and receives continuously and simultaneously (using two different ports or a bidirectional port).


K: Number of attempts
T_p: Maximum propagation time
T_{fr}: Average transmission time for a frame
T_B: Back-off time
Station has a frame to send
Start
K = 0
Apply one of the persistence methods (1-persistent, nonpersistent, or p-persistent)
Eligible for transmission
(Transmission done) or (Collision detected)
Yes
No
Transmit and receive
Collision detected?
Yes
No
Success
Send a jamming signal
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 $R \times T_{fr}$)

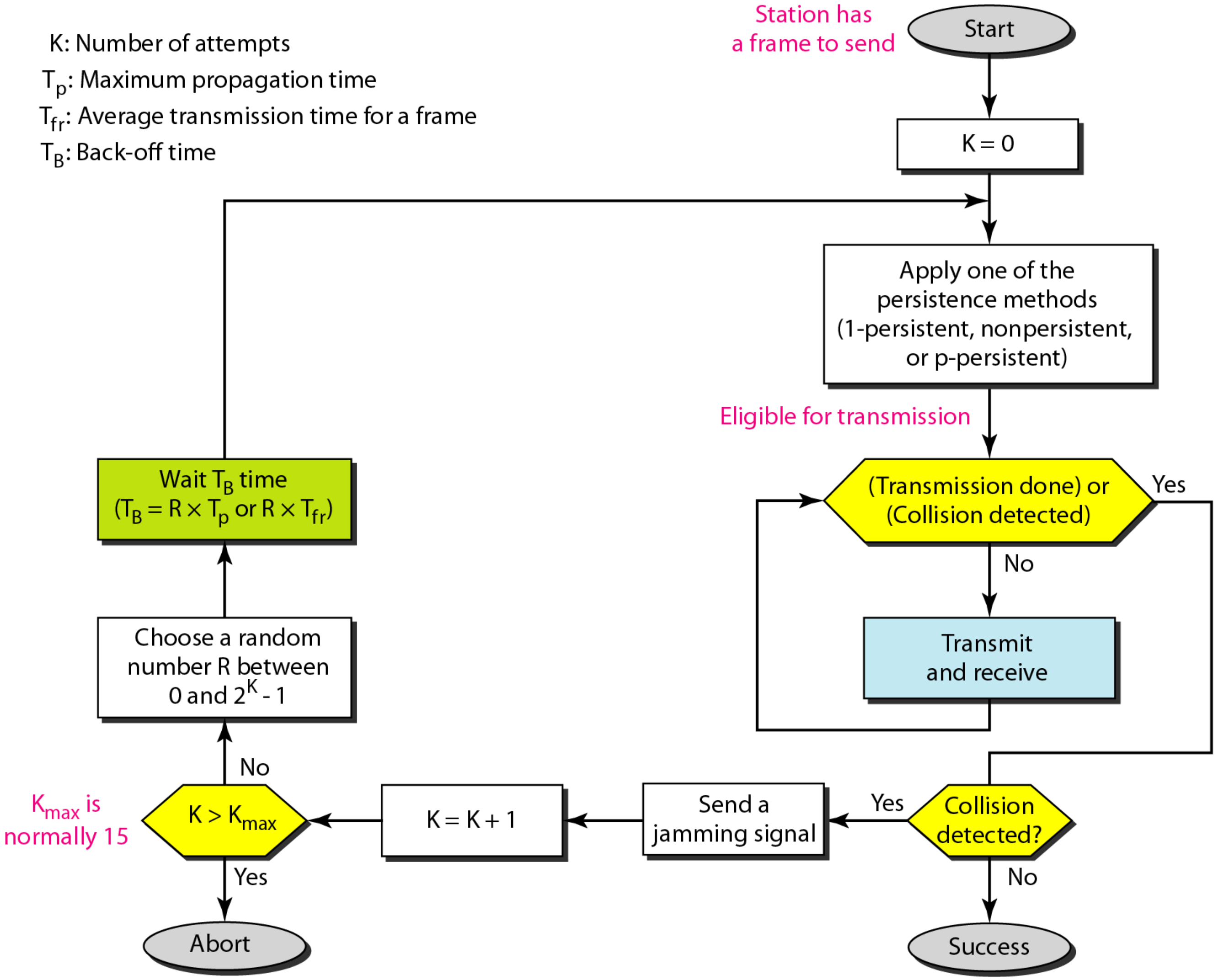
CMSA/CD Flow Chart

- Constantly monitor in order to detect one of two conditions: ***Either transmission is finished*** or ***a collision is detected***.

- Either event stops transmission.

- When we come out of the loop,
  - If a collision has not been detected,
    - It means that transmission is complete; the entire frame is transmitted.
  - Otherwise, a collision has occurred.

- 3rd: Sending of a short **jamming signal**
  - To make sure that all other stations become aware of the collision.

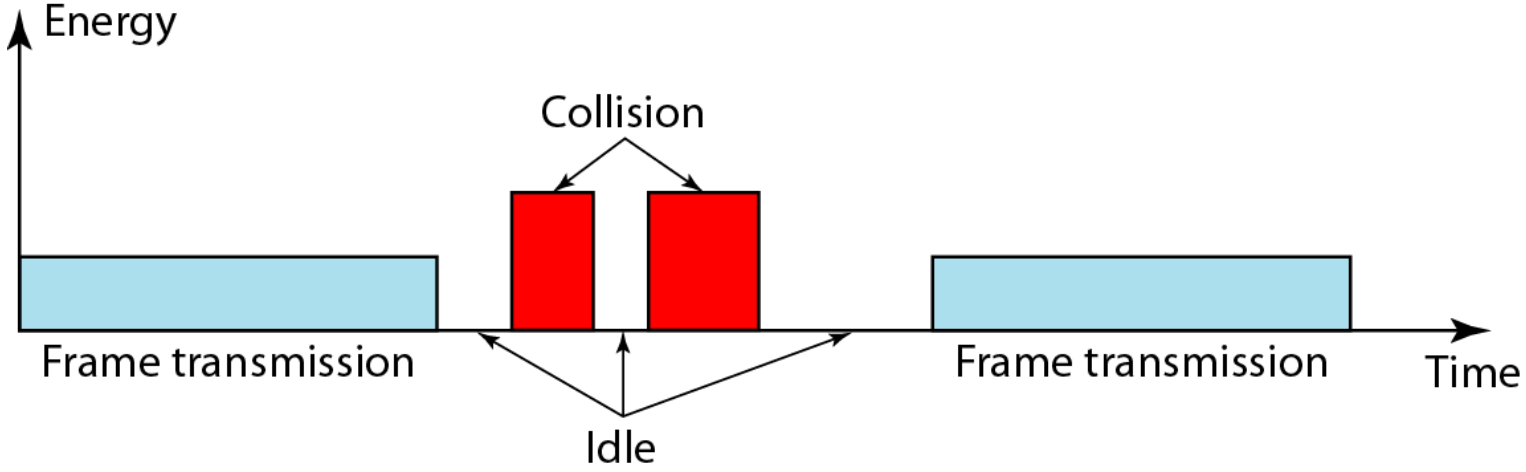

K: Number of attempts
T_p: Maximum propagation time
T_{fr}: Average transmission time for a frame
T_B: Back-off time
Station has a frame to send
Start
K = 0
Apply one of the persistence methods (1-persistent, nonpersistent, or p-persistent)
Eligible for transmission
(Transmission done) or (Collision detected)
Yes
No
Transmit and receive
Collision detected?
Yes
No
Success
Send a jamming signal
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 $R \times T_{fr}$)

CMSA/CD Energy Levels

Energy Levels 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



CMSA/CD Energy Levels

Throughput

- 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 the *1-persistent method*,
    - The Max throughput is around 50 percent when G = 1.

  - For the *nonpersistent method*,
    - The maximum throughput can go up to 90 percent when G is between 3 and 8.

Traditional Ethernet

- One of the LAN protocols that used CSMA/CD is the traditional Ethernet with the data rate of 10 Mbps.
- Ethernet was a broadcast LAN that used the 1-persistence method to control access to the common media.

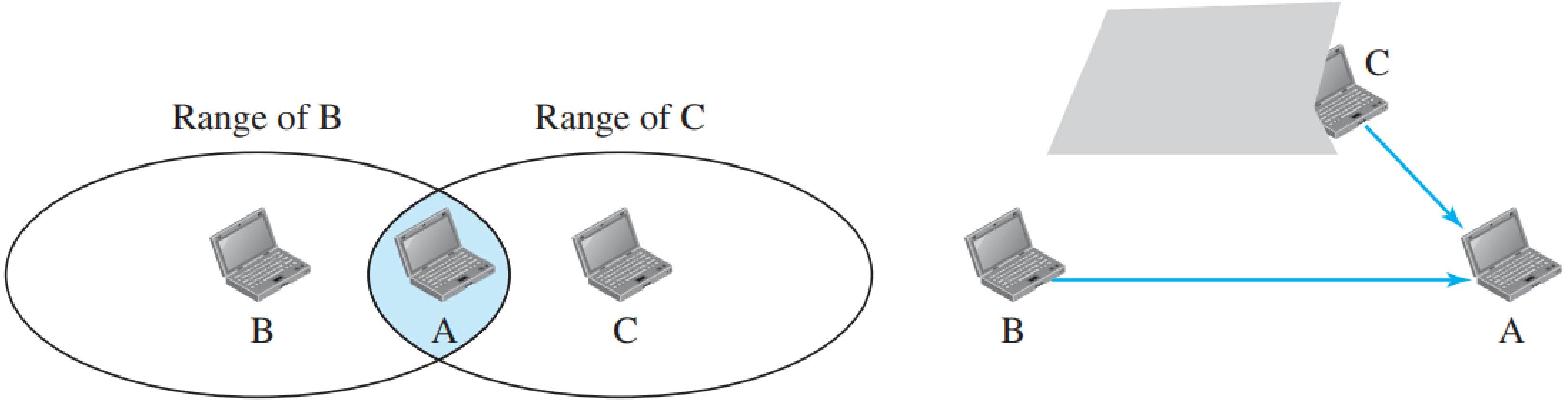
CSMA/CD Does Not Work in Wireless Networks

To detect a collision, CSMA has to work in a duplex mode.

- Wireless hosts do not have enough power to do so. They can only send or receive at one time.

Hidden Terminal Problem

- Due to this, in which **a station may not be aware of another station's transmission due to some obstacles or range problems**, collision may occur but not be detected.
- Figure shows an example of the hidden station problem.
  - Station B & C has a transmission range (sphere in space);
  - Station C is outside the transmission range of B; likewise, station B is outside the transmission range of C.
  - Station A, however, is in the area covered by both B and C; it can hear any signal transmitted by B or C.
  - The figure also shows that the hidden station problem may also occur due to an obstacle.



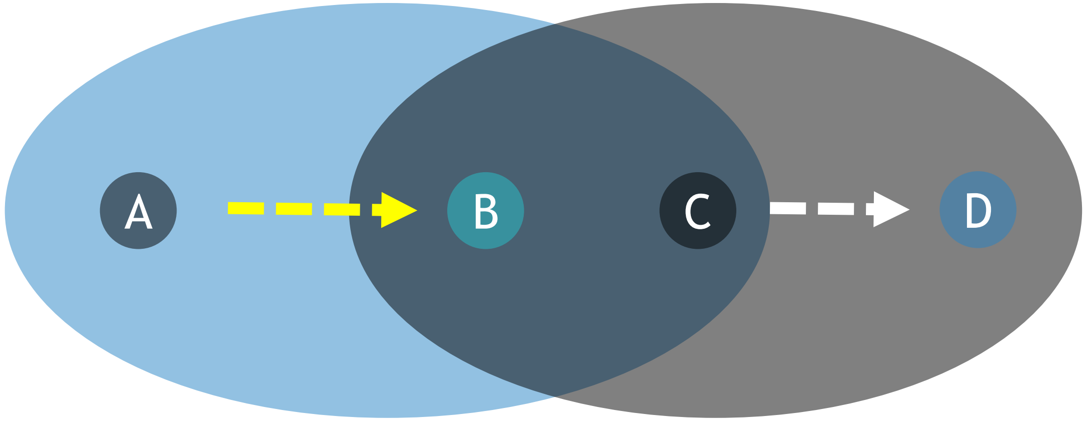
a. Stations B and C are not in each other's range.

b. Stations B and C are hidden from each other.

CSMA/CD Does Not Work in Wireless Networks

Exposed Terminal Problem

- In this problem a station refrains from using a channel when it is, in fact, available.
- In Figure, station A is transmitting to station B.
- Station C has some data to send to station D,
  - Which can be sent without interfering with the transmission from A to B.
- However, station C is exposed to transmission from A;
  - It hears what A is sending and thus refrains from sending.
- In other words, C is too conservative and wastes the capacity of the channel.



CMSA/CA

Carrier sense multiple access with collision avoidance was invented for wireless networks.

Collisions are avoided through the use of CSMA/CA's three strategies:

- The interframe space (IFS),
- The contention window, and
- Acknowledgments

In CSMA/CA,
IFS can also be used to **define the priority of a station or a frame**.

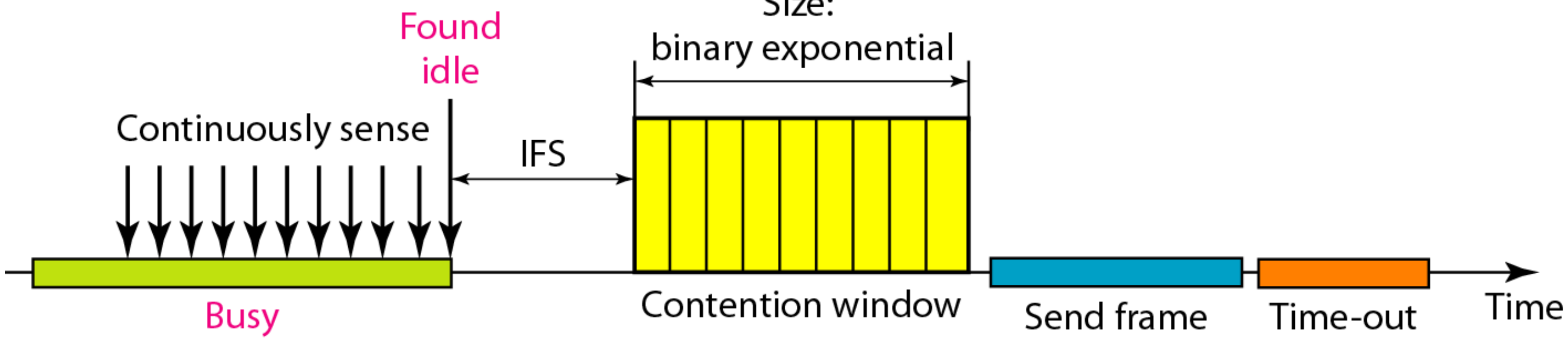
In CSMA/CA, if the station finds the channel busy,
It **does not restart the timer** of the **contention window**;
it stops the timer and restarts it when the channel becomes idle.

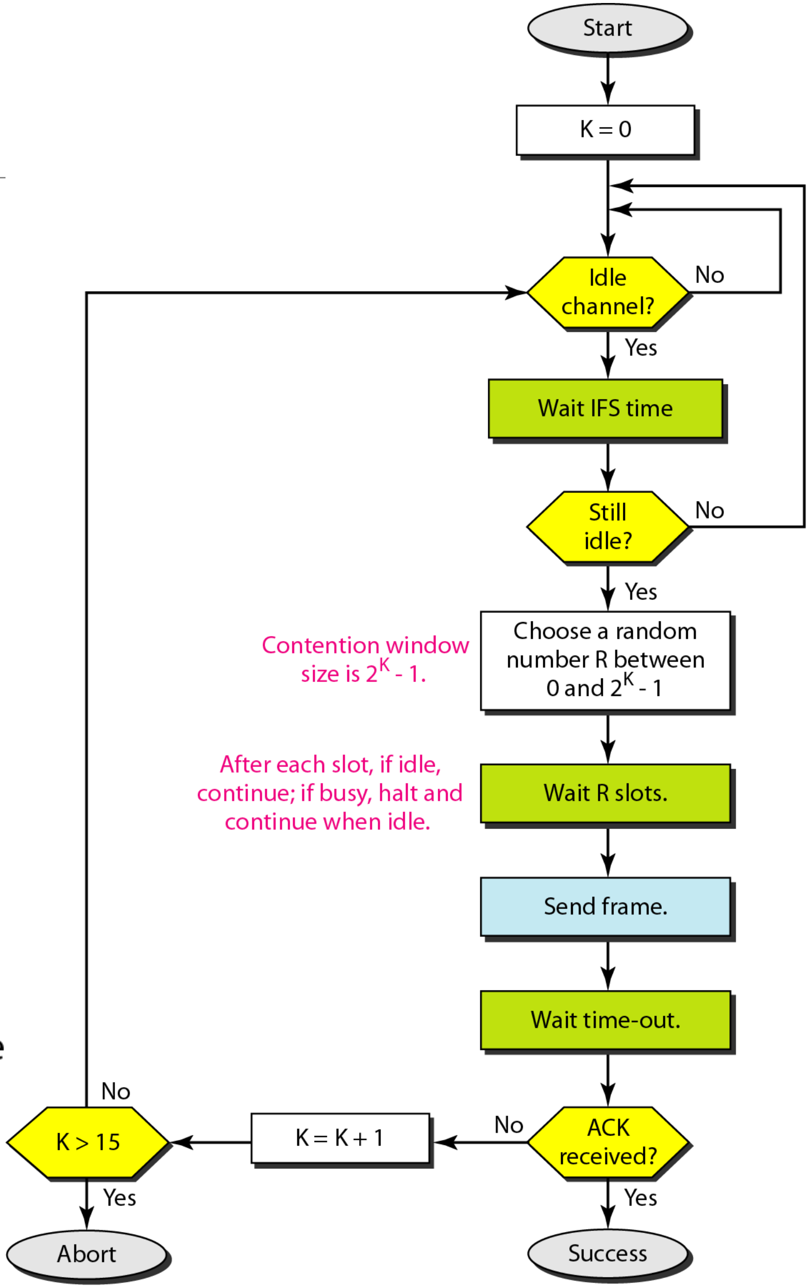
CMSA/CA

Interframe Space (IFS).

- 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***.

- After waiting an IFS time,
  - If the channel is still idle, the station can send,
  - But it still needs to wait a time equal to the contention window.





Start
K = 0
Idle channel?
No
Yes
Wait IFS time
Still idle?
No
Yes
Contention window size is $2^K - 1$.
Choose a random number R between 0 and $2^K - 1$
After each slot, if idle, continue; if busy, halt and continue when idle.
Wait R slots.
Send frame.
Wait time-out.
ACK received?
No
Yes
K = K + 1
K > 15
No
Yes
Abort
Success

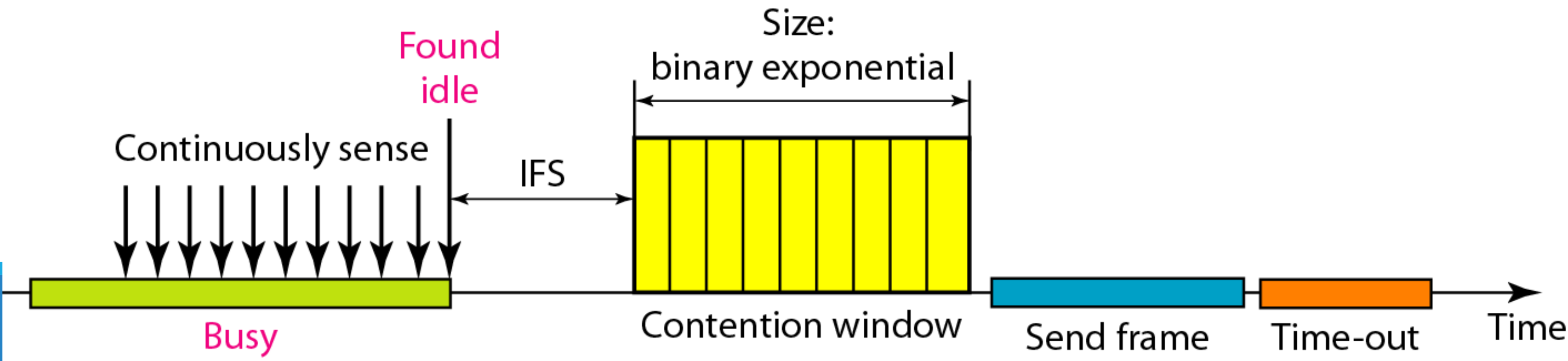
CMSA/CA

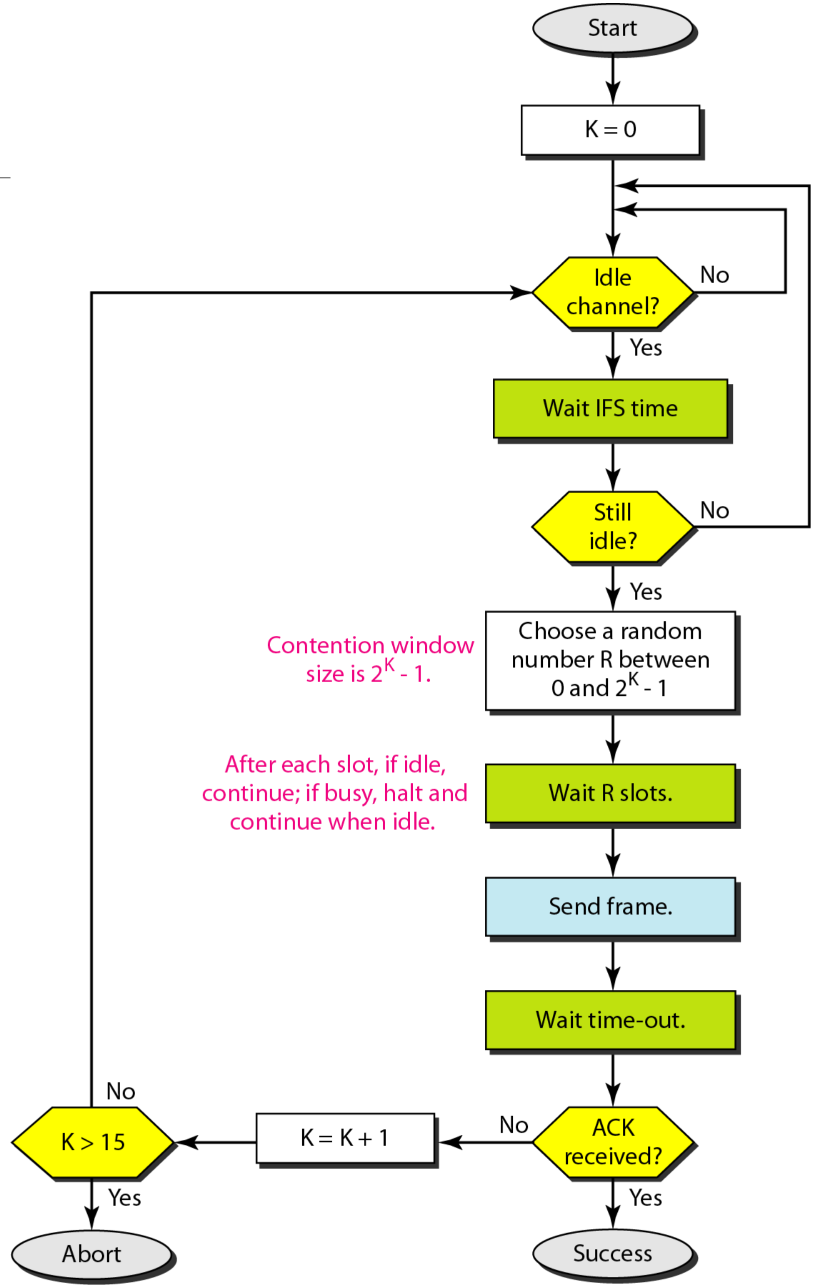
Contention Window.

- **Contention window** is an amount of time divided into slots.
- No. of slots in the window changes according to the binary exponential backoff strategy.
- **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.**
- **Gives priority to the station with the longest waiting time.**

Acknowledgment.

- The positive acknowledgment and the time-out timer can help guarantee that the receiver has received the frame.




Start
K = 0
Idle channel?
No
Yes
Wait IFS time
Still idle?
No
Yes
Contention window size is 2^K - 1.
Choose a random number R between 0 and 2^K - 1
After each slot, if idle, continue; if busy, halt and continue when idle.
Wait R slots.
Send frame.
Wait time-out.
No
K > 15
Yes
Abort
K = K + 1
No
ACK received?
Yes
Success

Controlled Access Protocols

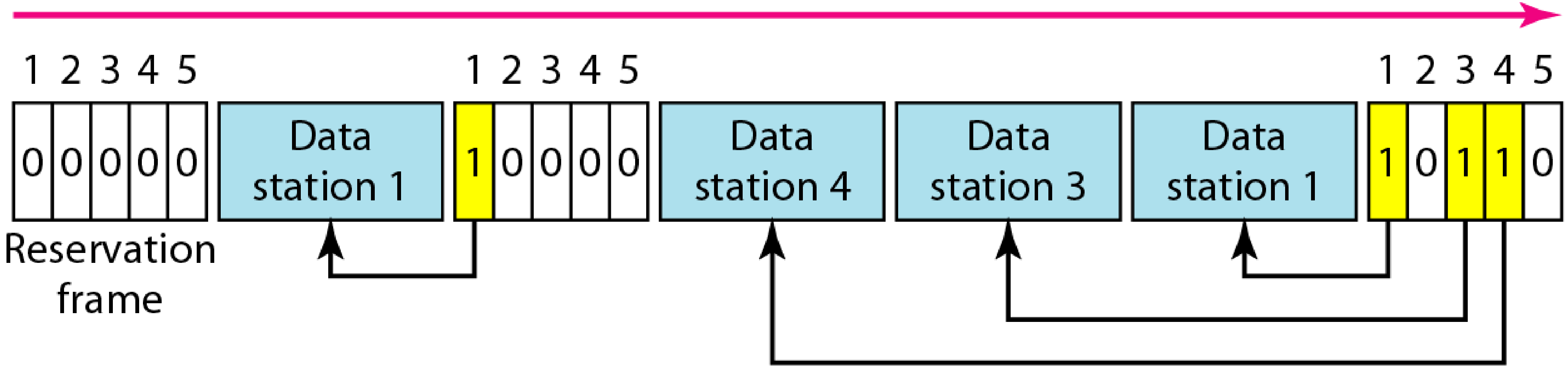
Controlled Access Protocols

- Stations consult one another to find which station has the right to send.
- A station cannot send unless it has been authorized by other stations.

- Reservation
- Polling
- Token Passing

Reservation Access 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.
  - *N* stations, there are exactly *N* reservation minislots in the reservation frame.
  - Each minislot belongs to a station.



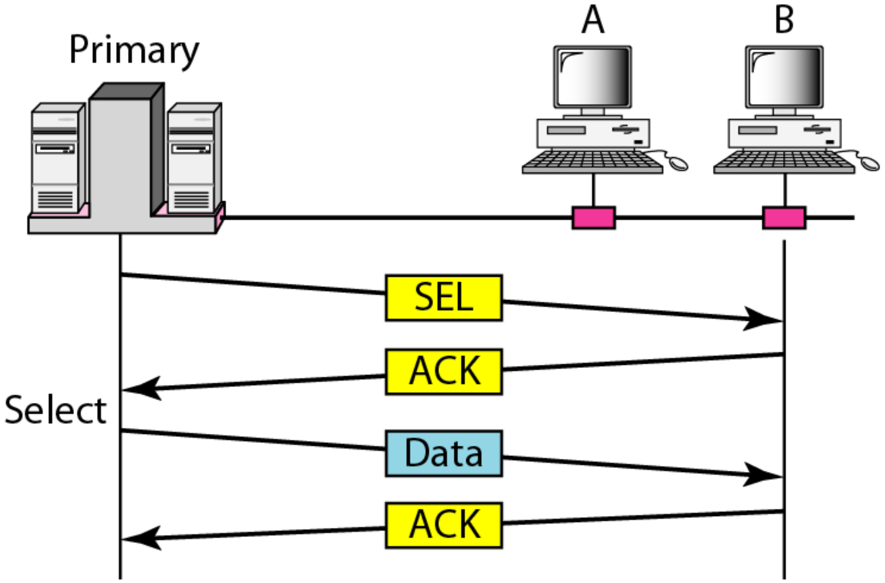
Poll Access Method

- **For topologies:** Where 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.
  - The **primary device controls the link**;
  - The **secondary devices follow its instructions**.

- **Primary Device:**
  - Decides **which device** is **allowed** to **use the channel** at a **given time**.
  - Is **always** the **initiator of a session**.
  - Uses **poll** and **select functions** to **prevent collisions**.
  - Drawback is if the primary station fails, the system goes down.

Poll Access Method

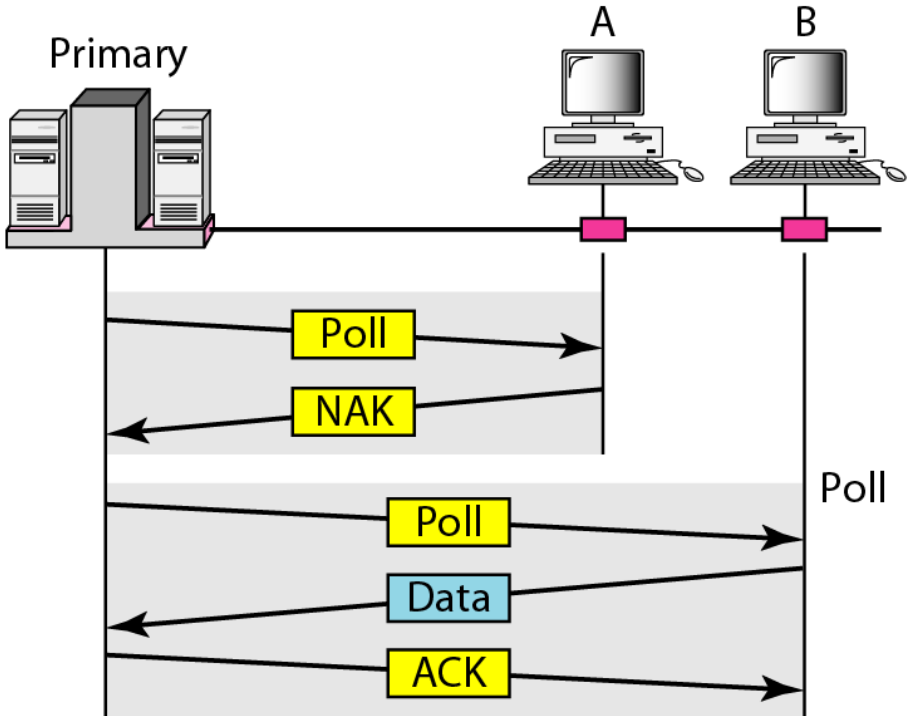
- Select
  - Used whenever the primary device has something to send.
  - Primary controls the link.
  - If the primary is neither sending nor receiving data,
    - Stations knows the **link is available**.
  - If station has something to send,
    - The primary device sends it.
  - What station does not know,
    - is whether the target device is prepared to receive.
  - 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 Access Method

- **Poll**
  - 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.



Token Passing Access Method

- The stations in a network **are organized in a logical ring**.
- For each station, there is a *predecessor* and a *successor*.
- A special packet called a ***token*** circulates through the ring.
- Token management is needed.

Logical Ring

Here, stations do not have to be physically connected in a ring;
The ring can be a logical one.

Current Station: The one that is **accessing the channel now.**

- **Right to this access** has been **passed from the predecessor** to the current station.
- **Right will be passed** to the **successor** when the **current station** has **no more data to send.**

Token Passing Access Method

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

- In this method, a special packet called a ***token*** circulates through the ring.
- **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.

- **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 data to the next station.

Token Passing Access Method

Token management is needed for this access method.

- Stations must be limited in the time they can have possession of the token.
- Token must be monitored to ensure it has not been lost or destroyed.

Token Management,

- 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**.
- Needed to make low-priority stations release the **token** to **high-priority** **stations**.

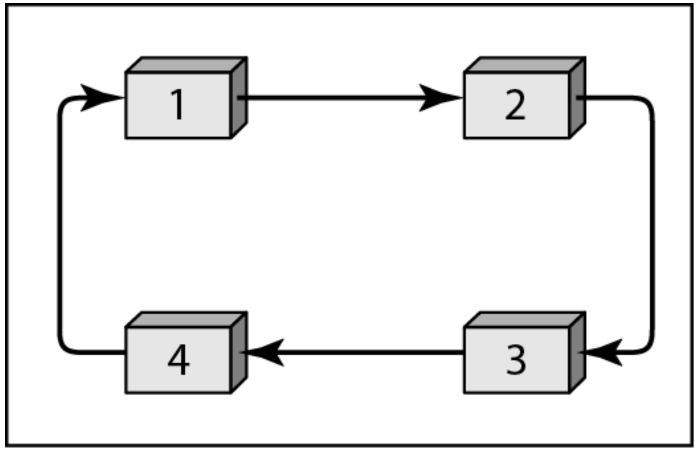
Token Passing Access Method

Logical Ring

Here, stations do not have to be physically connected in a ring; the ring can be a logical one.

Physical Ring Topology

- When a station sends the token to its successor.
  - Token cannot be seen by other stations;
  - The successor is the next one in line.
  - 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 between two adjacent stations fails, the whole system fails.

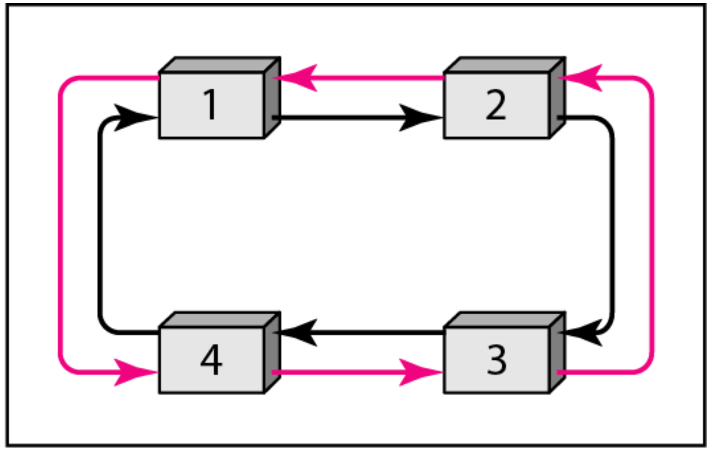


a. Physical ring

Token Passing Access Method

Dual Ring Topology

- Uses a second (auxiliary) ring which **operates in the reverse direction compared with the main ring**.
  - Second ring is for emergencies only.
- If one of the links in the main ring fails,
  - 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**.
- High-speed Token Ring networks called *FDDI (Fiber Distributed Data Interface)* and *CDDI (Copper Distributed Data Interface)* use this topology.

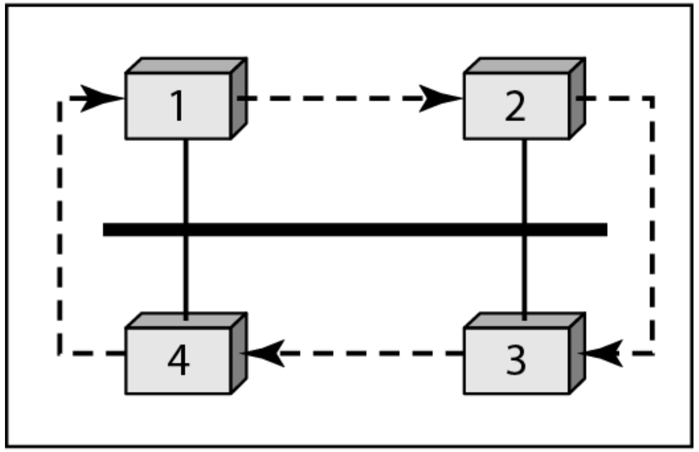


b. Dual ring

Token Passing Access Method

Bus Ring Topology (Token Bus)

- Stations are connected to a single cable called a *bus*.
- Form a logical ring, as **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 token** and inserts the address of its successor in 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.

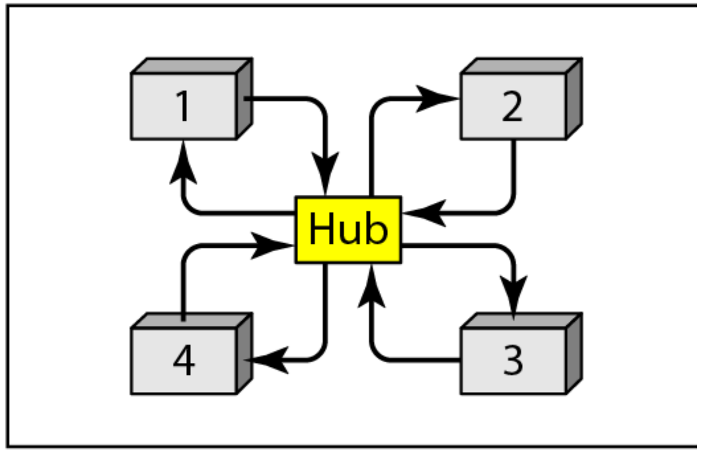


c. Bus ring

Token Passing Access Method

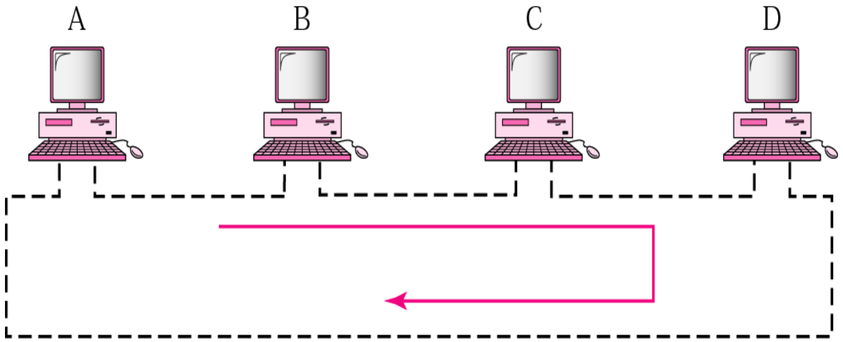
Star Ring Topology

- 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 two wire connections.
- 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



d. Star ring

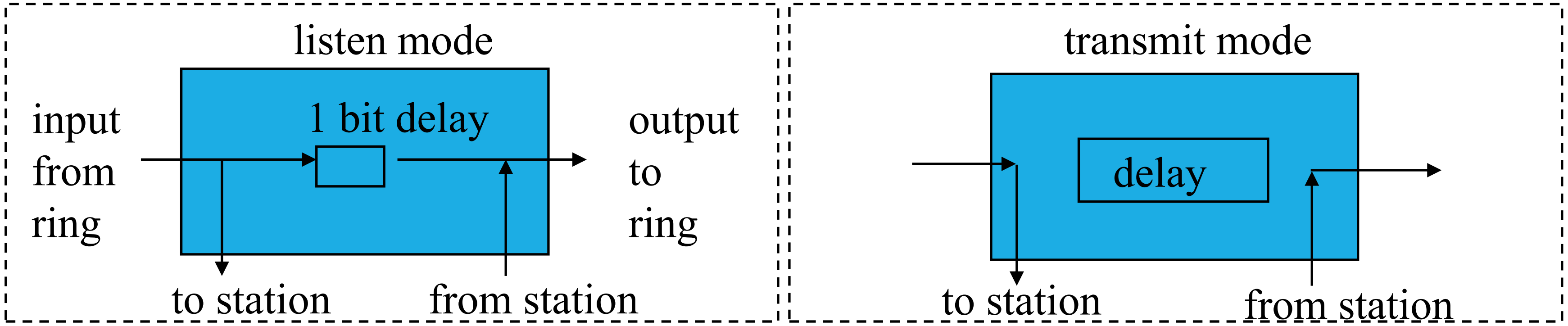
Token-Passing network



Implements Distributed Polling System

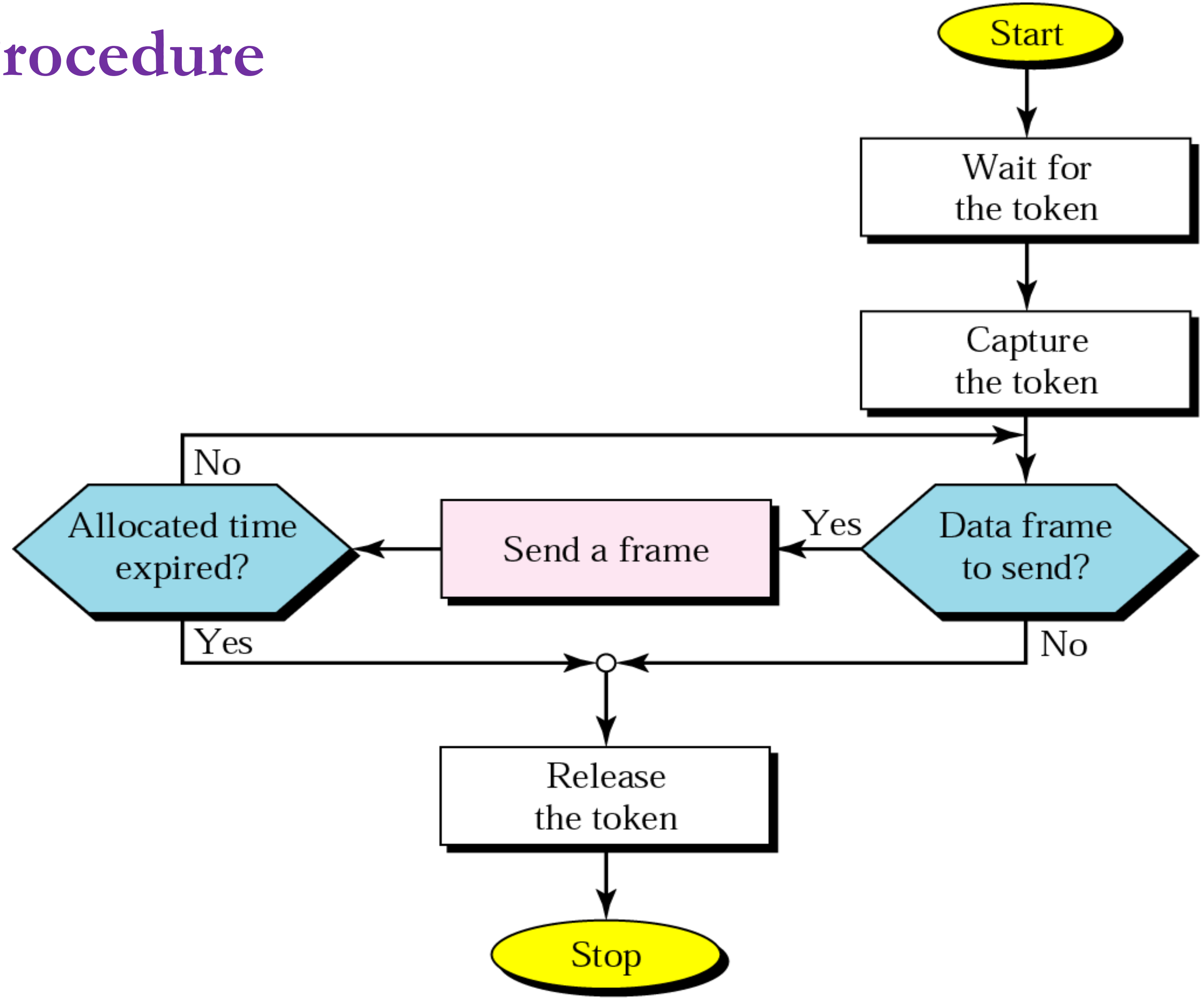
bits are copied to the output bits with a one bit delay

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.

Token Passing Procedure



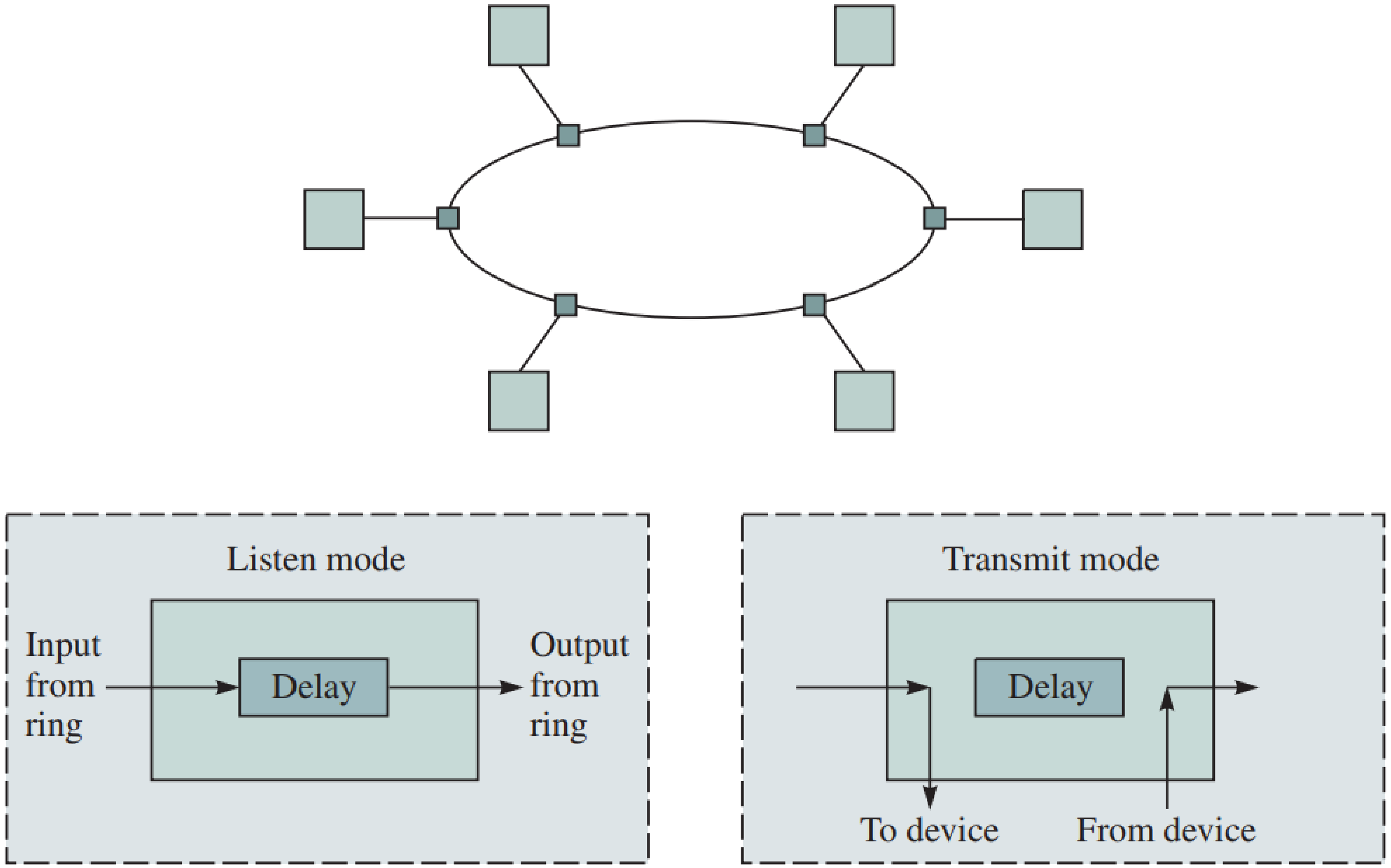
Token Ring Operation

In Fig., Such ring networks consist of **station interfaces**

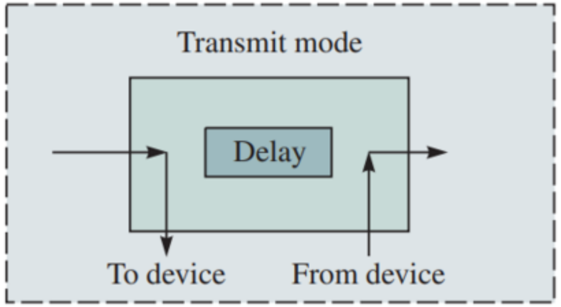
- That are connected by point-to-point digital transmission lines.
- Each interface acts like a **repeater**.

Listening Mode: An **interface** in the **listen mode**

- ***Reproduces each bit*** that is received after ***some constant delay***, ideally in the order of one bit time.
- This delay **allows the interface** to **monitor the passing bit stream for certain patterns**.



Token Ring Operation



Delay: For example,

- The interface will be looking for the **address of the attached station**.
  - When such an address is observed, the associated packet of information is copied bit by bit to the attached station.

Interface monitors the passing bit stream for the pattern corresponding to a "**free token**."

- When a **free token is received** and the **attached station has information** to send,
- The ***interface changes the passing token*** to ***busy*** by ***changing*** a bit in the passing stream.

In effect, receiving **a free token corresponds** to **receiving a polling message**.

The station interface then **changes to the transmit mode** where it proceeds to transmit packets of information from the attached station.

- These **packets circulate around** the **ring** and are **copied** at the **destination station interfaces**.

Token Ring Operation

While the station is transmitting its information:

- It is also receiving information at the input of the interface.

Ring Circulation Time (time to circulate around the ring)

- **Less than the time to transmit a packet,**
  - Then this arriving information corresponds to bits of the **same packet that the station is transmitting**.
- **Greater than a packet transmission time,**
  - **More than one packet** may be present in the ring at any given time.
  - In such cases the arriving information could correspond to bits of a packet from a different station,
    - So the station must buffer these bits for later transmission.

A **packet** that is **inserted** into the **ring** must be **removed**.

- One approach: To have the **destination station remove the packet from the ring**.
- Another approach: **Allow** the **packet** to **travel back** to the **transmitting station**.
  - This approach is usually preferred because the transmitting station interface can then forward the arriving packet to its attached station, thus providing a **form of acknowledgment**.

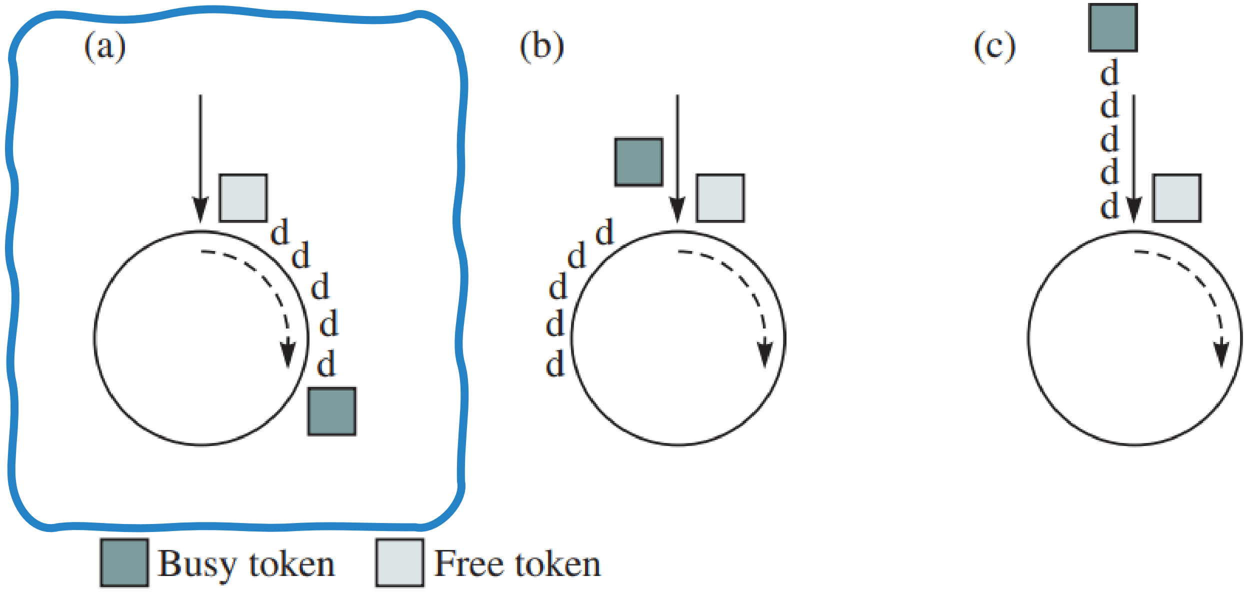
Token Ring Operation

Token rings differ according to the **method used to reinsert the token after transmission** has been completed.

- **Three approaches for token reinsertion**, as shown in Figure.
- Main **differences** between the methods arise when the **ring latency is larger than the packet length.**
  - **Ring latency:** The number of bits that can be simultaneously in transit around the ring.

1st Approach: Multitoken operation,

- **Free token is transmitted immediately after the last bit of the data packet.**
- **Minimizes the time required** to pass a free token to the next station.
- It also allows **several packets to be in transit in different parts of the ring.**

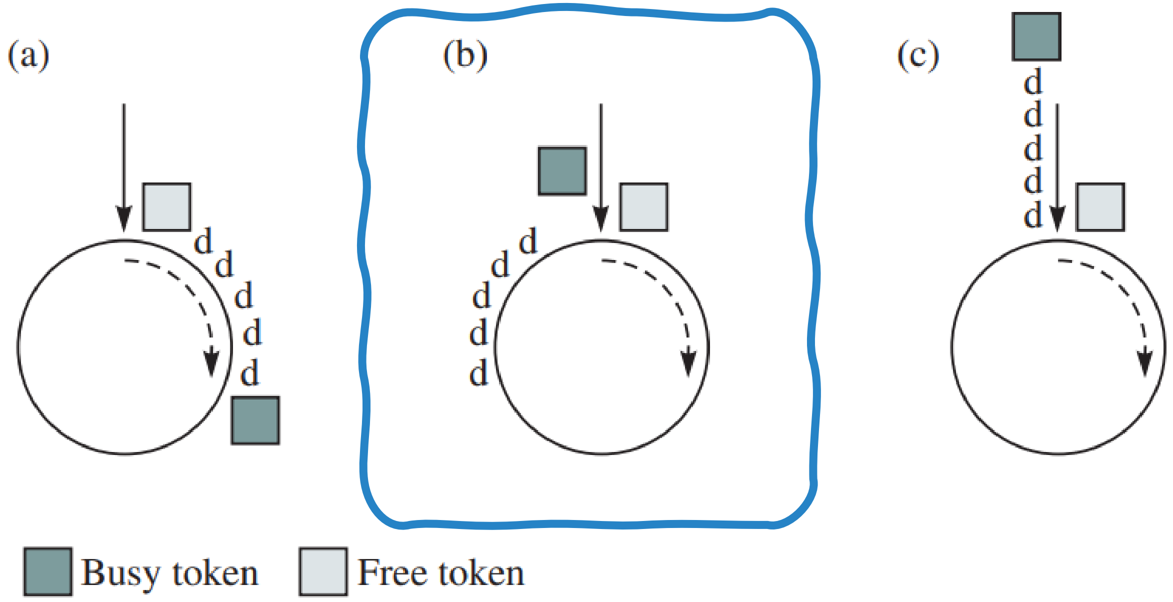


Approaches to token reinsertion: (a) multitoken, (b) single token, and (c) single packet

Token Ring Operation

2nd Approach: Single-token operation

- Involves **inserting the token** **after the last bit of the busy token is received back**.
- **If the packet is longer than the ring latency,**
  - Then the **free token will be inserted immediately after the last bit of the packet is transmitted**,
  - So the operation is equivalent to **multitoken operation**.
- **If the ring latency is greater than the packet length,**
  - Then a **gap will occur between the time of the last bit transmission and the reinsertion of the free token** as shown in Figure.
- **Recovery from errors in the token** is **simplified** by allowing **only one token to be present in the ring at any given time.**

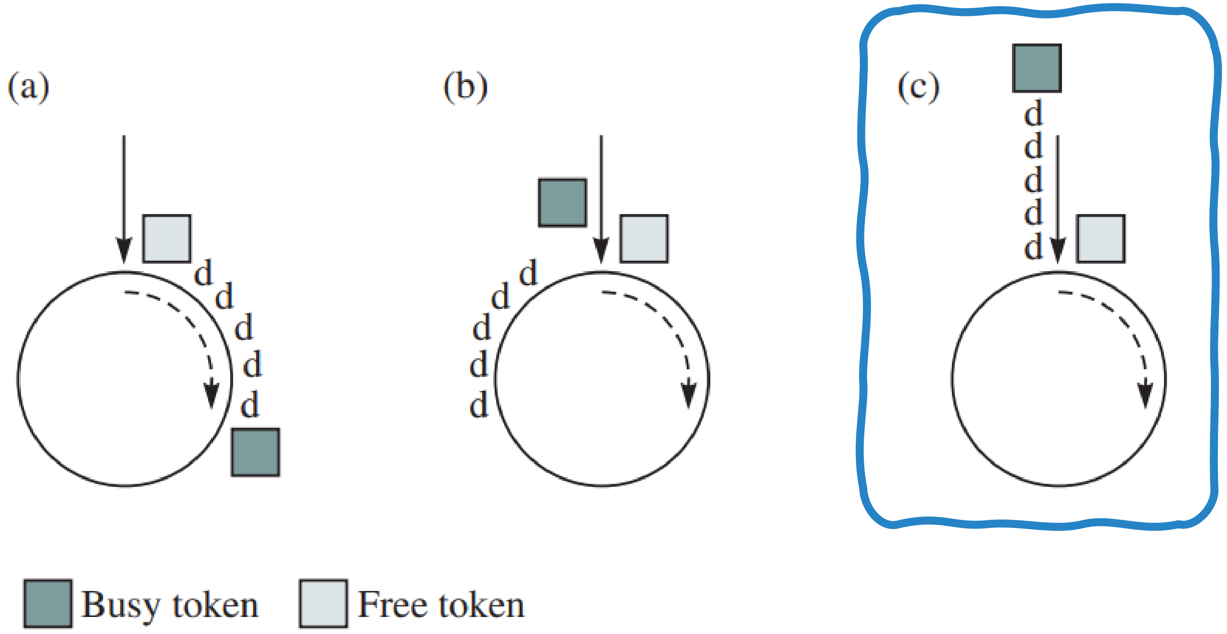


Approaches to token reinsertion: (a) multitoken, (b) single token, and (c) single packet

Token Ring Operation

3rd Approach,

- A single packet operation, the **free token is inserted after the transmitting station has received the last bit of its packet**.
- This approach allows the transmitting station to **check the return packet for errors** before **surrendering control of the token**.



Approaches to token reinsertion: (a) multitoken, (b) single token, and (c) single packet

Token Ring Operation

Token-ring operation usually also specifies a limit on the time that a station can transmit.

One approach: Allow a station to transmit an unlimited number of packets each time a token is received.

- **Minimizes the delay** experienced by packets

But allows the time that can elapse between consecutive arrivals of a free token to a station to be unbounded.

For this reason, a limit is usually placed either on the number of packets that can be transmitted each time a token is received or on the total time that a station may transmit information into the ring.

Token Ring Operation

The introduction of limits on the number of packets that can be transmitted per token affects the maximum achievable throughput.

Suppose that a maximum of one packet can be transmitted per token.

Let ***τ*' be the ring latency** (in seconds) and *a*' be the ring latency normalized to the packet transmission time.

We then have

$$\tau' = \tau + \frac{Mb}{R} \qquad a' = \frac{\tau'}{E[X]}$$

- where τ is the total propagation delay around the ring,
- b is the number of bit delays in an interface,
- Mb is the total delay introduced by the M station interfaces, and
- R is the speed of the transmission lines.

Token Ring Operation

Maximum throughput occurs when all stations transmit a packet.

- If the system uses **multitoken** operation,
- Total time taken to transmit the packets from the M stations is $ME[X] + \tau'$
- Because ME[X] of this time is spent transmitting information, the maximum throughput is then

$$\rho_{\max} = \frac{ME[X]}{ME[X] + \tau'} = \frac{1}{1 + \tau'/ME[X]} = \frac{1}{1 + a'/M} \text{ for multitoken.}$$

Token Ring Operation

Now suppose that the ring uses single-token operation.

- Assume that packets are of constant length L and that their transmission time is X = L/R.
- From Figure we can see that the effective packet duration is the maximum of X and r
- Therefore, the maximum throughput is then

$$\rho_{\max} = \frac{MX}{M\max\{X,\tau'\}+\tau'} = \frac{1}{\max\{1,a'\}+\tau'/MX}$$
$$= \frac{1}{\max\{1,a'\}+a'/M} \text{ for single token.}$$

Token Ring Operation

When the **packet transmission time is greater** than the **ring latency**,

- **The single-token operation has the same maximum throughput as multitoken operation.**

However, when the **ring latency is larger than the packet transmission time**,

- That is, a > 1, then the maximum throughput is less than that of multi-token operation.
- Finally, in the case of **single-packet operation** the effective packet transmission time is always $E[X] + \tau'$.
- Therefore, the maximum throughput is given by

$$\rho_{\max} = \frac{ME[X]}{M(E[X] + \tau') + \tau'} = \frac{1}{1 + a'\left(1 + \frac{1}{M}\right)} \text{ for single-packet.}$$

Module - 03 and 04

Problems

Problem 01

If the bandwidth of the line is 1.5 Mbps, RTT is 45 msec and packet size is 1 KB, then find the link utilization in stop and wait.

Solution-

- Given-

  Bandwidth = 1.5 Mbps

  RTT = 45 msec

  Packet size = 1 KB

Find:

- Transmission Delay (T_t) = Packet size / Bandwidth
- Propagation Delay (T_p) = RTT/2
- Calculating Value Of 'a' = T_p / T_t
- **Link Utilization or Efficiency** $\eta = \frac{1}{1+2a}$

- **Calculating Transmission Delay-**
- Transmission delay (T_t)

  = Packet size / Bandwidth

  = 1 KB / 1.5 Mbps

  = (2^{10} x 8 bits) / (1.5 x 10^6 bits per sec)

  = 5.461 msec

- **Calculating Propagation Delay-**
- Propagation delay (T_p)

  = Round Trip Time / 2

  = 45 msec / 2

  = 22.5 msec

- **Calculating Value Of 'a'-**

  a = T_p / T_t

  a = 22.5 msec / 5.461 msec a = 4.12

- **Calculating Link Utilization-**
- Link Utilization or Efficiency (η)

  = 1 / 1+2a

  = 1 / (1 + 2 x 4.12)

  = 1 / 9.24

  = 0.108

  = 10.8 %

Problem 02

Using stop and wait protocol, sender wants to transmit 10 data packets to the receiver. Out of these 10 data packets, every 4th data packet is lost. How many packets sender will have to send in total?

Solution

1, 2, 3, **4**, 4, 5, 6, **7**, 7, 8, 9, **10**, 10

- The lost packets are: 4, 7 and 10.
- Thus, sender will have to send 13 data packets in total.

Problem 03

A sender uses the stop and wait ARQ protocol for reliable transmission of frames.
Frames are of size 1000 bytes and the transmission rate at the sender is 80 Kbps.
Size of an acknowledgement is 100 bytes and transmission rate at the receiver is 8 Kbps.
The one way propagation delay is 100 msec.
Assuming no frame is lost, the sender throughput is ________ bytes/sec.

Solution-

- Given-
- Frame size = 1000 bytes
- Sender bandwidth = 80 Kbps
- Acknowledgement size = 100 bytes
- Receiver bandwidth = 8 Kbps
- Propagation delay (T_p) = 100 msec

Calculating

- Transmission delay (T_t)

- **Calculating Transmission Delay Of Data Frame-**
- Transmission delay (T_t)

= Frame size / Sender bandwidth

= 1000 bytes / 80 Kbps

= (1000 x 8 bits) / (80 x 10^3 bits per sec)

= 0.1 sec

= 100 msec

- **Calculating Transmission Delay Of Acknowledgement-**
- Transmission delay (T_t)

= Acknowledgement size / Receiver bandwidth

= 100 bytes / 8 Kbps

= (100 x 8 bits) / (8 x 10^3 bits per sec)

= 100 msec

- **Calculating Useful Time-**
- Useful Time

= Transmission delay of data frame

= 100 msec

- **Calculating Total Time-**
- Total Time

= Transmission delay of data frame + Propagation delay of data frame + Transmission delay of acknowledgement + Propagation delay of acknowledgement

= 100 msec + 100 msec + 100 msec + 100 msec

= 400 msec

Calculating Efficiency-

- Efficiency (η)

= Useful time / Total time

= 100 msec / 400 msec

= 1 / 4

= 25%

Calculating Sender Throughput-

- Sender throughput

  = Efficiency (η) x Sender bandwidth

  = 0.25 x 80 Kbps

  = 20 Kbps

  = (20 x 1000 / 8) bytes per sec

  = 2500 bytes/sec

Problem 04

The values of parameters for the stop and wait ARQ protocol are as given below-

- Bit rate of the transmission channel = 1 Mbps
- Propagation delay from sender to receiver = 0.75 ms
- Time to process a frame = 0.25 ms
- Number of bytes in the information frame = 1980
- Number of bytes in the acknowledge frame = 20
- Number of overhead bytes in the information frame = 20
- Assume that there are no transmission errors. Then the transmission efficiency (in %) of the stop and wait ARQ protocol for the above parameters is__________________________________. (correct to 2 decimal places)

- **Solution-**
- Given:

  Bandwidth = 1 Mbps

  Propagation delay (T_p) = 0.75 ms

  Processing time ($T_{process}$) = 0.25 ms

  Data frame size = 1980 bytes

  Acknowledgement frame size = 20 bytes

  Overhead in data frame = 20 bytes

- **Calculating Useful Time-**
- Useful data sent

  = Transmission delay of useful data bytes sent

  = Useful data bytes sent / Bandwidth

  = (1980 bytes – 20 bytes) / 1 Mbps

  = 1960 bytes / 1 Mbps

  = (1960 x 8 bits) / (10^6 bits per sec)

  = 15680 µsec

  = 15.680 msec

- **Calculating Total Time-**

- ***Total time*** = Transmission delay of data frame + Propagation delay of data frame + Processing delay of data frame + Transmission delay of acknowledgement + Propagation delay of acknowledgement

  = (1980 bytes / 1 Mbps) + 0.75 msec + 0.25 msec + (20 bytes / 1 Mbps) + 0.75 msec

  = 15.840 msec + 0.75 msec + 0.25 msec + 0.160 msec + 0.75 msec

  = 17.75 msec

- **Calculating Efficiency-**

- Efficiency (η)

  = Useful time / Total time

  = 15.680 msec / 17.75 msec

  = 0.8833

  = 88.33%

Problem 05

- Consider two hosts X and Y connected by a single direct link of rate 10^6 bits/sec. The distance between the two hosts is 10,000 km and the propagation speed along the link is 2×10^8 m/sec. Host X sends a file of 50,000 bytes as one large message to host Y continuously. Let the transmission and propagation delays be p milliseconds and q milliseconds respectively.
- Then the value of p and q are-
- p = 50 and q = 100
- p = 50 and q = 400
- p = 100 and q = 50
- p = 400 and q = 50

Solution-

- Given-

Bandwidth = 10^6 bits/sec

Distance = 10,000 km

Propagation speed = 2 x 10^8 m/sec

Packet size = 50,000 bytes

- **Calculating Transmission Delay-**
- Transmission delay (T_t)
  = Packet size / Bandwidth
  = 50000 bytes / 10^6 bits per sec
  = ($5 \times 10^4 \times 8$ bits) / 10^6 bits per sec
  = (4×10^5 bits) / 10^6 bits per sec
  = 0.4 sec
  = 400 msec

- **Calculating Propagation Delay-**
- Propagation delay (T_p)

  = Distance / Propagation speed

  = 10000 km / (2×10^8 m/sec)

  = 10^7 m / (2×10^8 m/sec)

  = 50 msec

Problem 06

On a wireless link, the probability of packet error is 0.2. A stop and wait protocol is used to transfer data across the link. The channel condition is assumed to be independent from transmission to transmission. What is the average number of transmission attempts required to transfer 100 packets?

- Given-

  Probability of packet error = 0.2 We

  have to transfer 100 packets

- When we transfer 100 packets, number of packets in which error will occur = 0.2 x 100 = 20
- Then, these 20 packets will have to be retransmitted.
- When we retransmit 20 packets, number of packets in which error will occur = 0.2 x 20 = 4.
- Then, these 4 packets will have to be retransmitted.
- When we retransmit 4 packets, number of packets in which error will occur = 0.2 x 4 = 0.8 ≅ 1.
- Then, this 1 packet will have to be retransmitted.

- From here, average number of transmission attempts required = 100 + 20 + 4 + 1 = 125.

Problem 07

Host A is sending data to host B over a full duplex link. A and B are using the sliding window protocol for flow control. The send and receive window sizes are 5 packets each. Data packets (sent only from A to B) are all 1000 bytes long and the transmission time for such a packet is 50 µs. Acknowledgement packets (sent only from B to A) are very small and require negligible transmission time. The propagation delay over the link is 200 µs. What is the maximum achievable throughput in this communication?

Solution-

- Given:
- Sender window size = Receiver window size = 5
- Packet size = 1000 bytes
- Transmission delay (T_t) = 50 μs
- Propagation delay (T_p) = 200 μs

Calculating Bandwidth-

- We know,
  Transmission delay = Packet size / Bandwidth
- So, Bandwidth
  = Packet Size / Transmission delay (T_t)
  = 1000 bytes / 50 μs
  = (1000 x 8 bits) / (50 x 10^{-6} sec)
  = 160 Mbps

Calculating Value of 'a'-

$a = T_p / T_t$

a = 200 μsec / 50 μsec

a = 4

Calculating Optimal Window Size-

- Optimal window size
  - = 1 + 2a
  - = 1 + 2 x 4
  - = 9

- **Calculating Efficiency-**
- Efficiency (η) = Sender window size / Optimal window size
  = 5 / 9
  = 0.5555
  = 55.55%

Calculating Maximum Achievable Throughput-

- Maximum achievable throughput

  = Efficiency (η) x Bandwidth

  = 0.5555 x 160 Mbps

  = 88.88 Mbps

  = 88.88 x 10^6 bps or 11.11 x 10^6 Bps

Problem 08

Consider 100 frames are being sent. Compute the fraction of the bandwidth that is wasted on overhead (headers and retransmissions) for a protocol on a heavily loaded 50 Kbps satellite channel with data frames consisting of 40 bits header and 3960 data bits. Assume that the signal propagation time from the earth to the satellite is 270 msec. ACK frames never occur. NAK frames are 40 bits. The error rate for data frames is 1% and the error rate for NAK frames is negligible.

Useful Data Sent-

Since each frame contains 3960 data bits, so while sending 100 frames,
Useful data sent
= 100 x 3960 bits
= 396000 bits

Useless Data Sent / Overhead-

In general, overhead is due to headers, retransmissions and negative acknowledgements.

Now,

The error rate for data frames is 1%, therefore out of 100 sent frames,
error occurs in one frame.
This causes the negative acknowledgement to follow which causes the retransmission.

- So, we have-
  Overhead due to headers = 100 x 40 bits = 4000 bits.
  Overhead due to negative acknowledgement = 40 bits.
  Overhead due to retransmission = 40 bits header + 3960 data bits = 4000 bits.

- From here,
  Total overhead
  = 4000 bits + 40 bits + 4000 bits
  = 8040 bits

- **Calculating Efficiency-**

- Efficiency (η) = Useful data sent / Total data sent

- Here,
  Useful data sent = 396000 bits

  Total data sent = Useful data sent + Overhead

  = 396000 bits + 8040 bits = 404040 bits

- Substituting the values, we get- Efficiency (η)
  = 396000 bits / 404040 bits
  = 0.9801

Calculating Bandwidth Utilization-

- Bandwidth Utilization

  = Efficiency x Bandwidth

  = 0.9801 x 50 Kbps

  = 49.005 Kbps

Calculating Bandwidth Wasted-

- Bandwidth wasted

  = Bandwidth – Bandwidth Utilization

  = 50 Kbps – 49.005 Kbps

  = 0.995 Kbps

Calculating Fraction of Bandwidth Wasted

- Fraction of bandwidth wasted

  = Wasted Bandwidth / Total Available Bandwidth

  = 0.995 Kbps / 50 Kbps

  = 0.0199

  = 1.99 %

Problem 09

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?

Solution

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