

# Back-off Algorithm for CSMA/CD

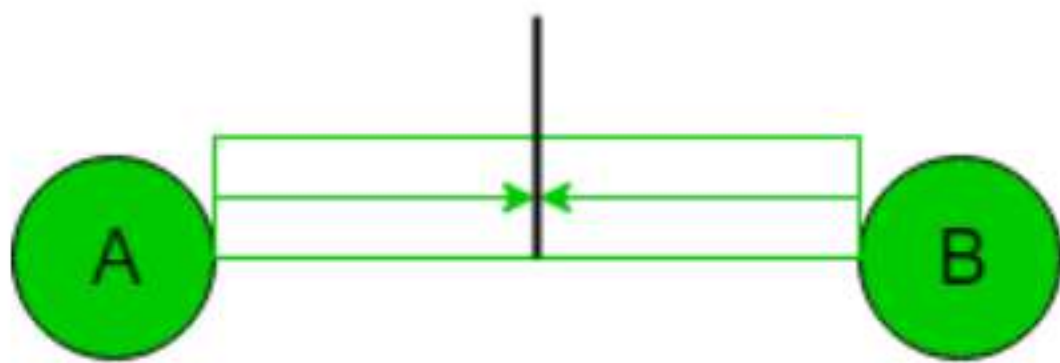
Back-off algorithm is a **collision resolution** mechanism which is used in random access *MAC* protocols (*CSMA/CD*). This algorithm is generally used in Ethernet to schedule re-transmissions after collisions.

If a collision takes place between 2 stations, they may restart transmission as soon as they can after the collision. This will always lead to another collision and form an infinite loop of collisions leading to a deadlock. To prevent such scenario back-off algorithm is used.

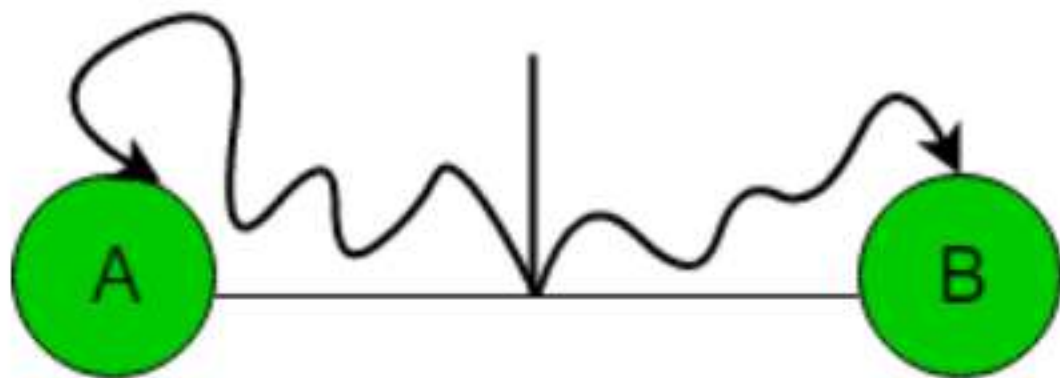
Let us consider an scenario of 2 stations  
A and B transmitting some data:



At  $t = 0$  , both A and B start transmission



Packets of both A and B collide



Both stations A and B detect collision

After a collision, time is divided into discrete slots ( $T_{slot}$ ) whose length is equal to  $2t$ , where  $t$  is the maximum propagation delay in the network.

The stations involved in the collision randomly pick an integer from the set  $K$  i.e  $\{0, 1\}$ . This set is called the contention window. If the sources collide again because they picked the same integer, the contention window size is doubled and it becomes  $\{0, 1, 2, 3\}$ . Now the sources involved in the second collision randomly pick an integer from the set  $\{0, 1, 2, 3\}$  and wait that number of time slots before trying again. Before they try to transmit, they listen to the channel and transmit only if the channel is idle. This causes the source which picked the smallest integer in the contention window to succeed in transmitting its frame.

So, Back-off algorithm defines a *waiting time for the stations involved in collision*, i.e. for how much time the station should wait to re-transmit.

Waiting time = back-off time

Let  $n$  = collision number or re-transmission

Then,

Waiting time =  $K * T_{slot}$

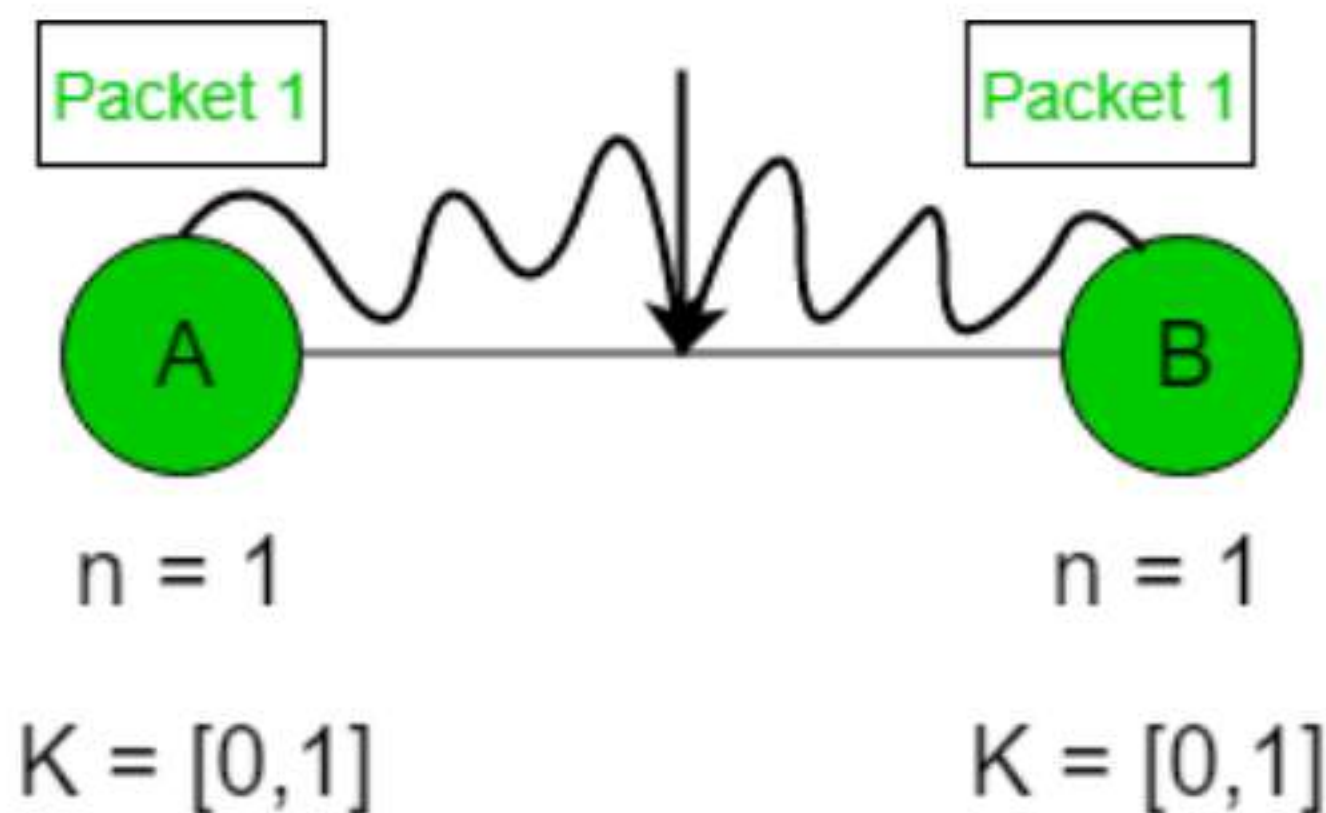
where  $K = [0, 2^n - 1]$



# Example –

## Case-1 :

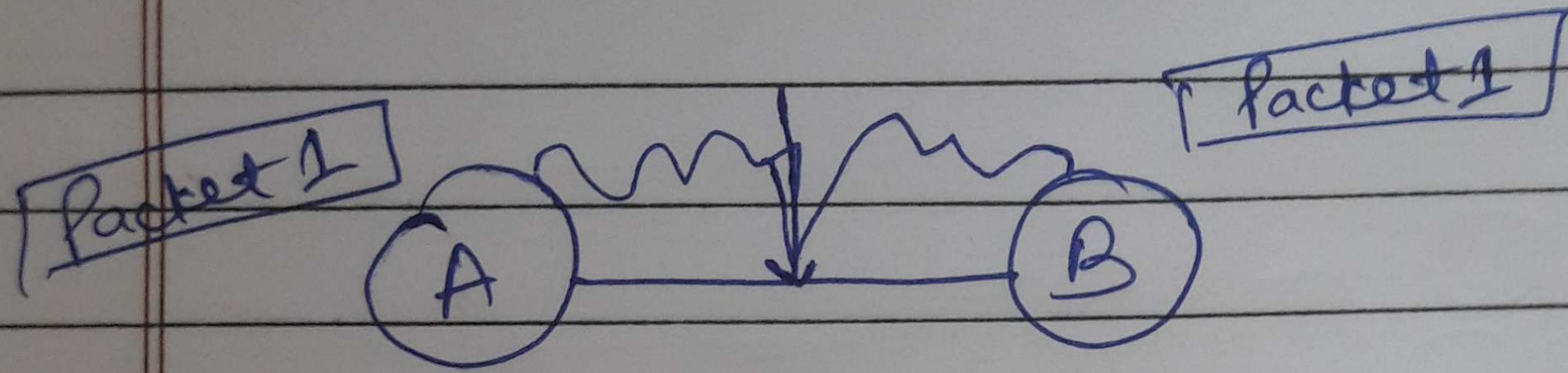
Suppose 2 stations A and B start transmitting data (Packet 1) at the same time then, collision occurs. So, the collision number  $n$  for both their data (Packet 1) = 1. Now, both the station randomly pick an integer from the set  $K$  i.e.  $\{0, 1\}$ .



Value of K

A	B
0	0
0	1
1	0
1	1





$$n=1$$

$$n=1$$

→ no. of collisions for packet 1.

$$K = [0, 2^n - 1]$$

$$K = [0, 1]$$

∴ Put  $n=1$  in above range

$$K = [0, 2^1 - 1]$$

$$K = [0, 1]$$



- When both A and B choose  $K = 0$**

→ Waiting time for A =  $0 * T_{\text{slot}} = 0$

Waiting time for B =  $0 * T_{\text{slot}} = 0$

Therefore, both stations will transmit at the same time and hence collision occurs.
- When A chooses  $K = 0$  and B chooses  $K = 1$**

→ Waiting time for A =  $0 * T_{\text{slot}} = 0$

Waiting time for B =  $1 * T_{\text{slot}} = T_{\text{slot}}$

Therefore, A transmits the packet and B waits for time  $T_{\text{slot}}$  for transmitting and hence A wins.
- When A chooses  $K = 1$  and B chooses  $K = 0$**

→ Waiting time for A =  $1 * T_{\text{slot}} = T_{\text{slot}}$

Waiting time for B =  $0 * T_{\text{slot}} = 0$



Therefore, B transmits the packet and A waits for time  $T_{\text{slot}}$  for transmitting and hence B wins.

- **When both A and B choose  $K = 1$**

→ Waiting time for A =  $1 * T_{\text{slot}} = T_{\text{slot}}$

Waiting time for B =  $1 * T_{\text{slot}} = T_{\text{slot}}$

Therefore, both will wait for the same time  $T_{\text{slot}}$  and then transmit. Hence, collision occurs.

Probability that A wins =  $1/4$

Probability that B wins =  $1/4$

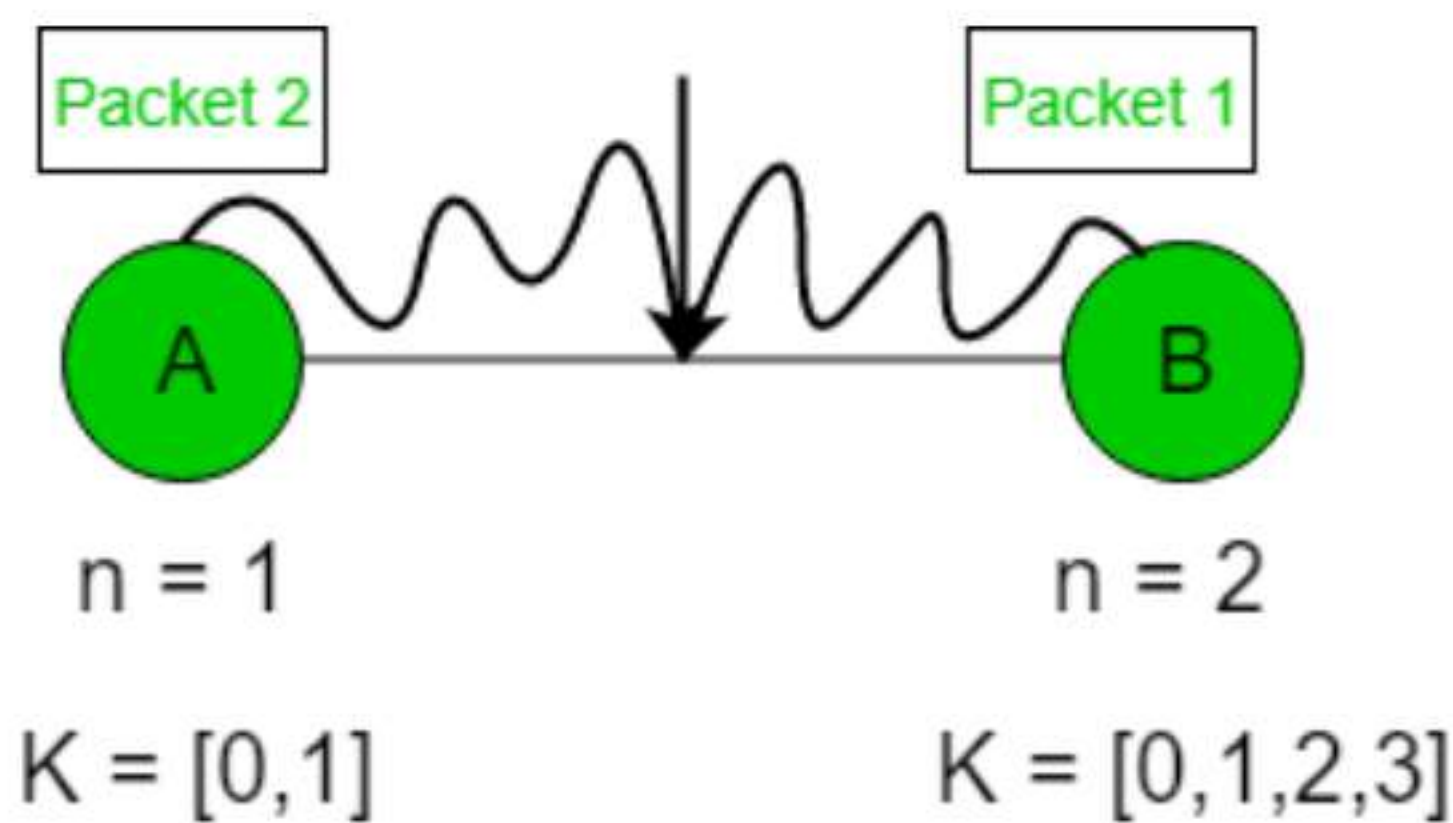
Probability of collision =  $2/4$

## Case-2 :

Assume that A wins in Case 1 and transmitted its data(Packet 1). Now, as soon as B transmits its packet 1, A transmits its packet 2. Hence, collision occurs. Now collision no. n becomes 1 for packet 2 of A and becomes 2 for packet 1 of B.

For packet 2 of A,  $K = \{0, 1\}$

For packet 1 of B,  $K = \{0, 1, 2, 3\}$



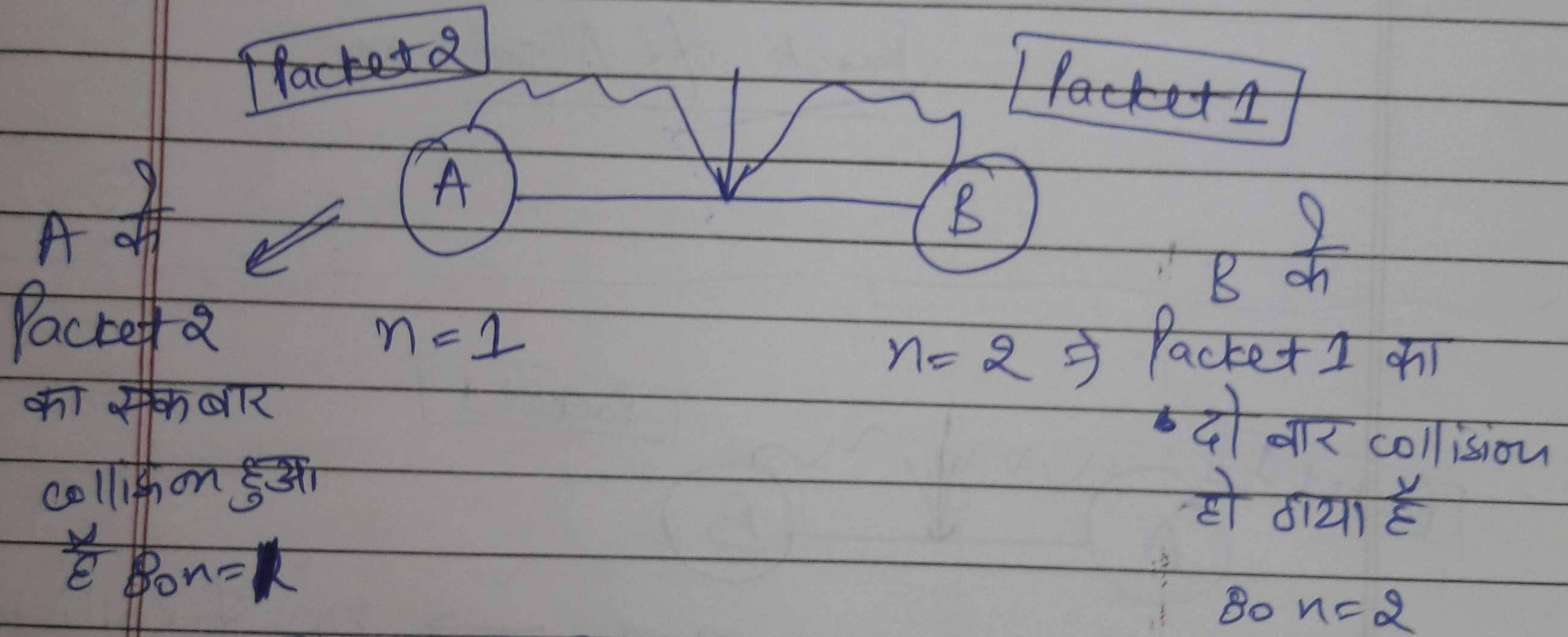
Value of K

A	B
0	0
0	1
0	2
0	3
1	0
1	1
1	2
1	3



Now, 1st packet of A has been transmitted & B is waiting for its 1st packet to transmit.

Now, A has to transmit its 2nd packet.



$$K = [0, 2^n - 1]$$

$$K = [0, 2^n - 1]$$

$$K = [0, 2^1 - 1]$$

$$K = [0, 2^2 - 1]$$

$$K = [0, 1]$$

$$K = [0, 1, 2, 3]$$



	A	B	
	0	0	→ Both send data packet at same time ∴ <u>collision</u> .
A retransmit immediately & B waits for 1 time slot.	0	1	→ <u>A wins (1)</u>
B waits for 2 time slots	0	2	→ <u>A wins (2)</u>
B waits for 3 time slots.	0	3	→ <u>A wins (3)</u>
	1	0	→ <u>B wins (A waits)</u>
	1	1	→ <u>collision</u>
(4) <u>A wins</u> ←	1	2	→ B waits for longer time
(5) <u>A wins</u> ←	1	3	→

Total chances = 8

$P(A) = 5/8$        $P(C) = 2/8$        $P(B) = 1/8$

---



Probability that A wins =  $5/8$

Probability that B wins =  $1/8$

Probability of collision =  $2/8$

So, probability of collision decreases as compared to Case 1.

### **Advantage –**

- Collision probability decreases exponentially.

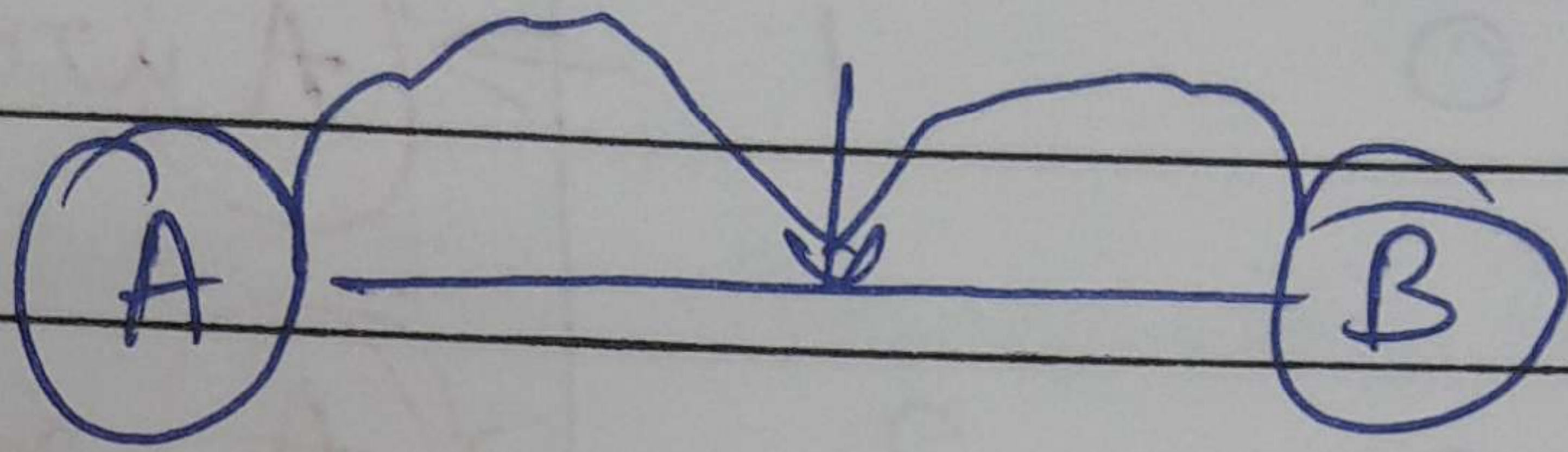
### **Disadvantages –**

- **Capture effect:** Station who wins ones keeps on winning.
- Works only for 2 stations or hosts.



Packet 3

Packet 1



$n=1$

$n=3$

(0, 1)

(0, 1, 2, 3, 4, 5, 6, 7)

No. of combinations will be

$$(2 \times 8 = 16)$$

~~W<sub>t</sub>~~

$$W_t = K * T_{spt}$$

$T_{spt} = \text{address} / \text{fixed time}$   
acc. to the network.

Note

(1)

If A has won in the first case after collision, then A's probability of winning keeps on increasing.

Disadvantage

(1) If one station wins once, then it will keep on winning many no. of times (which decreases the ~~chances~~ chances of winning for other station).

Capture effect



Probabilities increase or decrease exponentially

So Binary exponential back-off algorithm



Another name