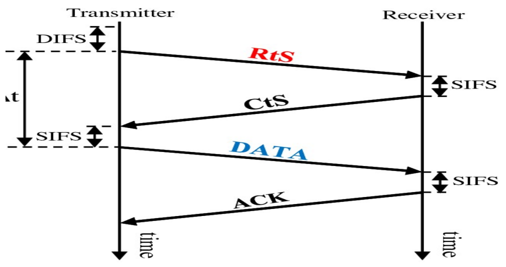
Communication - Module

1.1 Distributed coordination function (DCF)

DCF procedure as its MAC protocol to access the shared medium in first the station can be transmission any data cand be produce the clear channel assignment (CCA) to determine the channel condition to get channel state in idle state or busy state DCF minimizes the collision probability for data being transmitted from other stations by using the BEB algorithm and several control frame or sub frames such as

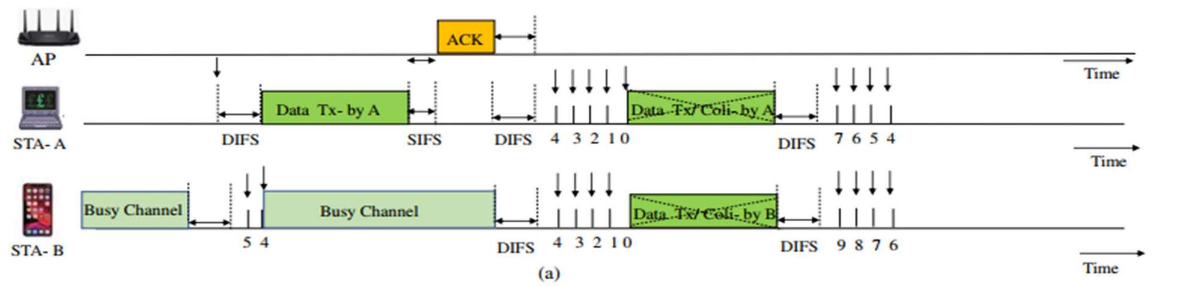
1- Request to Send Frame (RTS)
2- Clear to Send Frame (CTS)
3- Acknowledgment Frame (ACK)



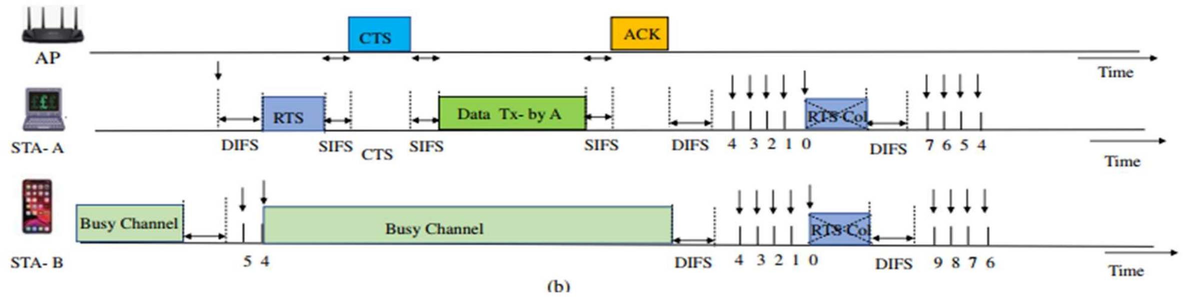
The RTS/CTS mechanism aims to reduce frame collisions introduced by the hidden terminal problem. CTS frame is sent by the receiver after it gets the RTS frame prior to receiving of the actual data frame.
And ACK the response of the data Frame is success or not

4- Distributed Inter-Frame Spacing (DIFS): The time interval that a station should wait before it sends its request frame is known as DCF Interframe Spacing (DIFS). DISF is calculated as the sum of SISF and twice the slot time.
5- Short Inter-frame Space (SIFS): the time interval required by device in between receiving a frame and responding to the frame

each contending station employs BEB and a deferral mechanism that helps to distinguish each transmission time to reduce the collision probabilities. In addition, the contending stations choose randomly discrete back-off slots in the range of $(0, CW - 1)$ ***before accessing the channel. The DCF enables the BEB algorithm if the channel is sensed idle for a period of DIFS, while the remaining stations set network allocation vector (NAV) to avoid the collisions.***
During the exponential back-off rules, each contending station decrements one from the back-off counter if the channel is sensed idle, freezes back-off counter when it detects a transmission on the channel (channel busy) and reactivates the back-off counter process when the channel is sensed idle again for a period of DIFS. Finally, the contending station proceeds the transmission when the back-off counter reaches zero, and the receiver sends back an ACK frame immediately (with a difference of short period SIFS) after the successful reception of the data



(a)



(b)

1.2 Backoff Time Algorithm's

1. ***Binary Exponential Back-off (BEB)** algorithm for the medium contention The BEB follows a blind exponential increase and decrease of back-off window and that is inefficient for highly dense networks due to a limited size of contention window (CW) size. contention window CW based on the number of the frame which is sent successfully and unsuccessfully and each contending station selects a random back-off value from the initial CW (CW_{min}), and its value is doubled (exponentially increased) after each unsuccessful transmission until it reaches a given maximum value CW_{max} and decrease after each station is successfully Each contending station may experience a maximum number of consecutive collisions that shows the maximum number of transmission stages (m). Therefore, at the maximum value of m, the back-off window size no longer doubles, stays at its maximum value and is given by $CW_{max} = CW_{min} * 2^m$ The CW size reverts to its initial minimum value CW_{min}after every successful transmission or after the maximum number of re-transmissions is reached (data packet is dropped at this level)*
   ***Problems**: In the case of a dense network, the process of resetting CW size to its minimum value CW_{min}after each successful data-transmission may encounter more collisions due to the small CW size for large numbers of contending stations, which directly impacts the overall network performance. Similarly, in a small network, exponentially increasing the value of CW for avoiding collisions causes unnecessarily long delays. Hence, the performance of BEB mainly depends on the CW adjustment and back-off strategy, which is not designed for network requirements*
   And active station must collect the channel-state information using various methods and coordinate with neighboring stations to reduce the transmission collisions and enhance the network throughput
2. ***Channel Observation-based Scaled Back-off mechanism (COSB)***
   Control of CW size by observes the channel conditions to measure the collision probability
   The last stations collied forced to stay at a higher backoff time (large CW) without knowing number of contending stations. This results in low throughput and higher delay values.
3. ***Channel Window-Scaled Backoff (CWSB)***
   channel collision-based window-scaled backoff (CWSB) mechanism to optimize the CW considering the last transmission and each contending station for its i^{th} transmission perceive medium for the total backoff period calculate the channel condition collision probability P_{cc} where the P_{cc} value is based perceived slot times between two consecutive backoff stages for example the current backoff stage is b_i this value doesn't come to zero or one step reversed after successful transmission compared to BEB and COSB sequentially

1.3 channel collision-based window-scaled backoff (CWSB)

each contending station continuously senses the medium after the medium sensed as idle for a distributed inter-frame space (DIFS) all the contending stations start to the back-off procedure by selecting a random back-off value d For each contending station, immediately following an idle DIFS, the total back-off time is slotted into observed slot times μ During the back-off procedure, the slot time size is set equal to the time needed at any contending station n to detect the transmission of a data frame from any other $n - 1$ stations and the slot time μ is either the constant slot time when an idle period (σ) or variable slot time when busy period (successful or collided). During the back-off procedure as the channel is sensed idle during σ the $backoff\ time\ (d)$ decrement by 1 decrement operation each timer interrupts every (bit time duration) and the as the channel is sensed busy back-off process is frozen (frozen is operation not active)and the station continuously senses the channel to access the medium and the contending station transmits a data frame after d approaches zero

DIFS Time	*D reach to 0*	*Strat transmit*		*channel is sensed idle*
DIFS Time	*Wait for busy*	*D reach to 0*	*Strat transmit*	*channel is sensed busy*

Note successful or collided transmission depends upon reception of the ACK frame. The case of the ACK frame not receive the transmission is considered as collided or failed

1- *probability that a transmission by a contending station (p_{cc})*

CWSB quantizes the slot times into σ, where the value of p_{cc} is based on the perceived slot times between two consecutive backs off stages

Type of slot time

1- *channel idle slot time (N_i)*
2- *channel collision slot time (N_c)*
3- *channel success transmission slot time (N_s)*
4- *channel busy slot time ((N_b)*

*to understand this timing can be assume backoff (**d**) is 8 and decrement the with each channel is idle*

*to send a Frame send the DIFS time and sense the channel idle to decrement (**d**) by one in fig 1 $N_i = 3$ to just three decrement value of backoff time and $(N_b) = 1$ the first channel is busy in produce the backoff after this operation the channel is idle to decrement the d by one in fig 2 $N_i = 2$ just 2 decrement of d and stop the decrement at sense channel busy and $(N_b) = 1$ and in fig 3 the $N_i = 3$ just 3 decrement of d and the d reach to zero (8-8) the backoff value is reach to zero can be start transmission of the Frame send Data Frame in fig 4 wait for SIFS (orange area) if the Ack time out the Increment the N_c by one*

or Ack is success received the Increment (N_s) by one

after this case update backoff stage from 1 to 2

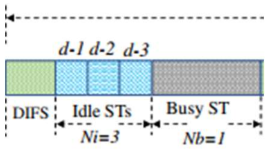


Fig 1

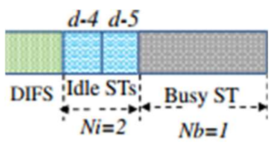
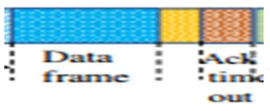


Fig 2



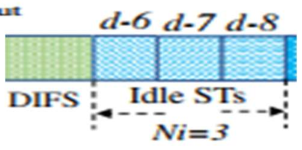
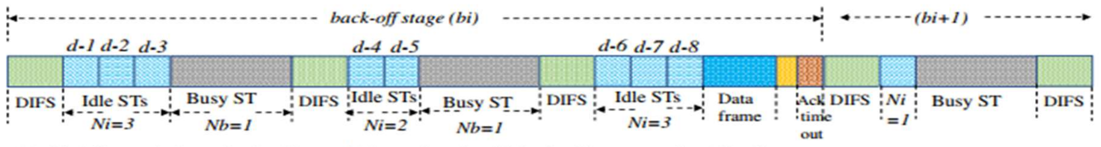
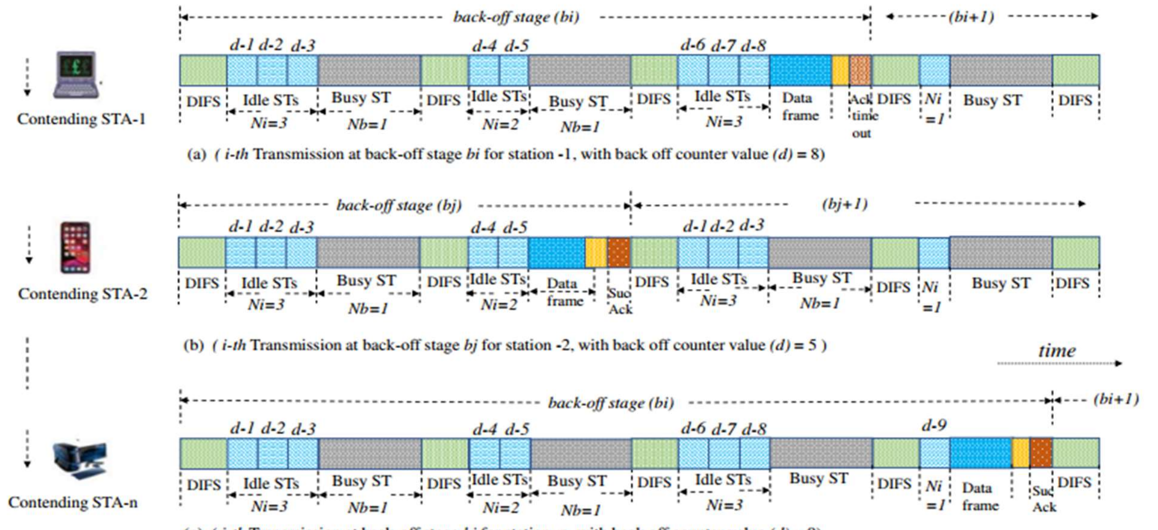


Fig 3

Fig 4



(a) (*i-th* Transmission at back-off stage *bi* for station -1, with back off counter value *(d)* = 8)



(a) (*i-th* Transmission at back-off stage *bi* for station -1, with back off counter value *(d)* = 8)

(b) (*i-th* Transmission at back-off stage *bj* for station -2, with back off counter value *(d)* = 5)

(c) (*i-th* Transmission at back-off stage *bi* for station -n, with back off counter value *(d)* = 9)

d : randomly generated back-off value, 🟨 :SIFS 🟫 : Successful Acknowledge, 🟦 : Data frame

i, j: discrete-time transmissions, b: current back-off stage, b ε (0, m), ▨ : Ack Timeout, ▨ : Busy due to other transmission

$$p_{cc} = \frac{N_b+N_c+N_s}{N_i+N_b+N_c+N_s} * N_c$$

$for\ example\ the\ d = 8\ f$or its ith transmission at b_i stage

in Fig 1 th$e$ $N_i = 3$ and Fig 2 $N_i = 2$, and Fig 3 $N_i = 3$

$$N_i = \sum N_i = 3 + 2 + 3 = 8\ idle\ solt\ time$$

in Fig 1 th $N_b = 1$ and Fig 2 $N_b = 1$, and Fig 3 $N_b = 0$

$$N_b = \sum N_b = 1 + 1 + 0 = 2\ busy\ solt\ time$$

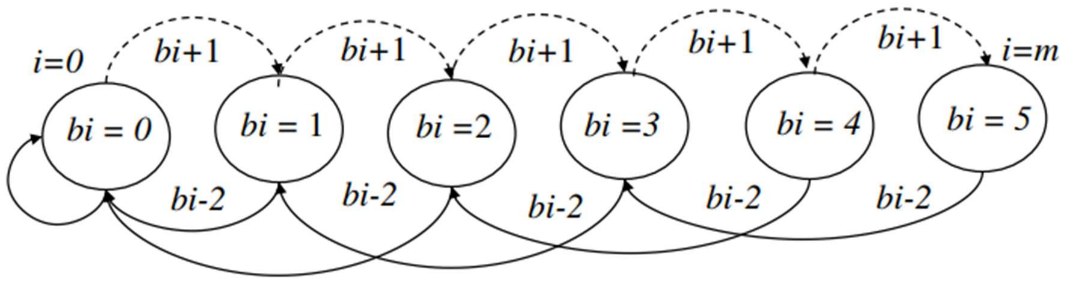
$$N_c = 1\ (ack\ not\ received), N_s = 0$$

$p_{cc} = \frac{2+1+0}{8+2+1+0} * 1 = 0.27$ *or the next stage bi+1, while, in the case of a successful transmission, Nc = 0 and Ns = 1.*

2- contention window CW_{b_i} when b_i is backoff stage

Furthermore, when a transmitted data frame experiences collision, the value of current contention window CW_{b_i} *of back-off stage* b_i *at the* ith *transmission is actively optimised (stepped up) according to the* p_{cc} *values and when a data frame is successfully transmitted, the current contention window*CW_{b_i} *of back-off stage* bi *at the* ith *transmission is also actively optimised (stepped down) the current back-off stage* b_i *does not come to zero or one step reversed after a successful transmission Moreover, the back-off stage* bi *in CWSB at the* ith *transmission adopts the below property towards increasing or decreasing for the next stage* b_{i+1}

$$b_{i+1} = \begin{cases} min\ (b_i + 1\ ,up\ to\ \ m) & i^{th}\ T_{xn}\ collided\ at\ b_i\ stage \\ min\ (b_i - 2\ ,up\ to\ 0) & i^{th}\ T_{xn}\ sucessfully\ at\ b_i\ stage \end{cases}$$



after, increment or decrement results in optimize (stepped up or stepped down) of the current CW_i *respectively. The optimization of* CW *operates as follows for a contending station.*

$$CW_{b_{i+1}} = \begin{cases} 2^{b_{i+1}} * CW_{min}{}^{(1+p_{cc})}\ ,CW_{max} & i^{th}\ T_{xn}\ collided\ at\ b_i\ stage \\ 2^{b_{i+1}} * CW_{min}{}^{(1+p_{cc})}\ ,CW_{min} & i^{th}\ T_{xn}\ sucessfully\ at\ b_i\ stage \end{cases}$$

3- Flow Chart of the backoff Algorisms