CS-224

(Computer Networks)

Assignment-2

Name: Sravan K Suresh

Roll No: 22B3936

Instructor: Prof. Vinay Ribeiro

IIT BOMBAY

Solution 1:

An efficient approach:

The goal of the malicious node should be –
Get as close as possible to N% of all the successfully transmitted msgs and yet remain above N%, thereby nullifying the chances of suspicion by the other nodes ($\because$ their frames are being blocked)
(Also, the approach of always transmitting frames or always collision increases backoff time to a limit, thus, affecting communication.)

Our approach : Let the Malicious node be M.
Now, M keeps count of no. of successful transmissions by itself and by others.
n_m (itself) → n_o (others)

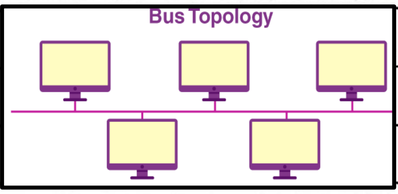
$$\therefore \boxed{n_m + n_o = n}$$
n → total no. of successful transmissions

* After every successful transmission (by M or by others) the ratio $\boxed{\frac{n_m}{n_m+n_o} \times 100}$ is monitored by M!

For M to identify successful transmissions:

★ If M is sending → The message length is kept greater than a threshold which allows M to detect any collision before it stops sending that frame.

★ If other node is sending → M will know that a message is sent ($\because$ ETHERNET LAN is a wired communication).
If there has been no jamming signal within the completion of message, then it was a 'successful' transmission.



(NOTE: For Initialization → Set the tunable parameter N of the malicious node $(0 \leq N \leq 100)$).

* Monitoring the network: Continuously monitor the network for the presence of idle or quiet periods where no other nodes are transmitting.
* Frame injection: During an idle period, the malicious node can inject its frame into the network. It must ensure that it does NOT collide with other frames!

* Collision detection: After transmitting a frame, the malicious node listens for a collision. If a collision is detected, it will back off and wait for a random amt. of time (using exponential backoff algorithm) before re-attempting transmission.

Chasing the desired success rate (N%):

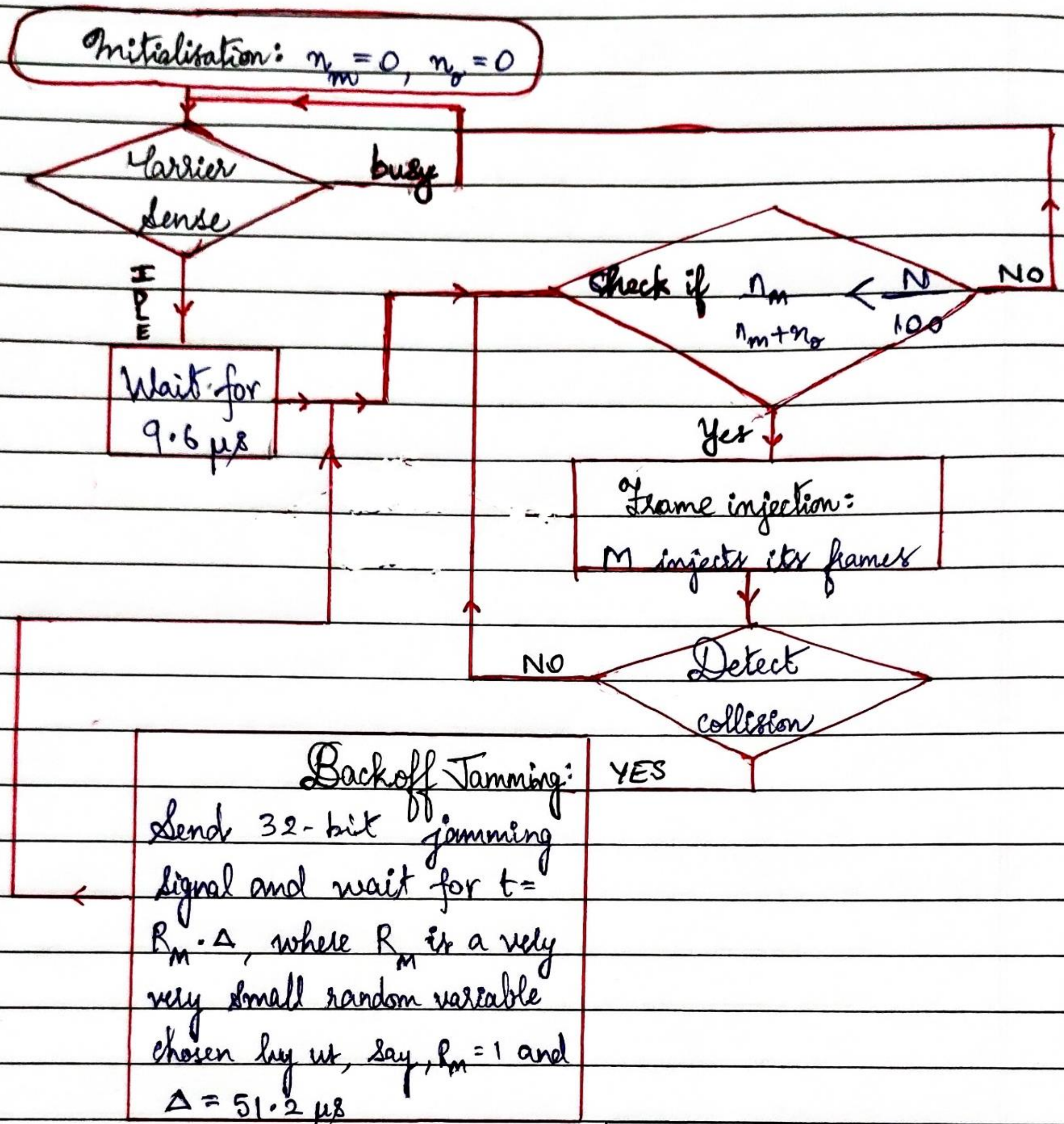
$\frac{n_m}{n_m + n_o} < \frac{N}{100}$ → Continue injecting frames during idle periods, and adjusting backoff period.

$n \geq \frac{N}{100}$ → Stop injecting frames to maintain the desired success rate.

NOTE :

∵ Given M has infinite data to transmit, ∴ the cycle of monitoring and adjusting must continue in an infinite loop.

FLOWCHART



CONCLUSION:

★ The malicious node achieves its goal much more efficiently by using this backoff jamming protocol as it jams the channel for a relatively longer duration keeping its random var as the least, thus causing other nodes to back off and give the malicious node more time and chance to transmit successfully.

NOTE: The flowchart is looped without termination/exit ∵ M has infinite data to transmit.

Solution 2:

Representing $d(X,Y)$ as distance b/w X and Y,

$$d(A_1A_2) = d(B_1B_2) = d(C_1C_2) = 2 < 3$$
$$d(A_1B_1) = d(B_1C_1) = 2 < 3$$
$$d(A_1,B_2) = d(B_1,A_2) = d(B_1C_2) = d(C_1,B_2) = \sqrt{8} < 3$$
$$d(A_1,C_1) = d(A_2C_2) = 4 > 3$$
$$d(A_1,C_2) = d(A_2C_1) = 2\sqrt{5} > 3$$

By symmetry, it gets obvious that
$T_A = T_C$ - (1)
Also, $T_A > T_B$ - (2) (NOTE that A, B, C stay sending data continuously as they're ∞ data)

REASON ↓

When B_1 sends to B_2, congestion window of A_1, C_1 will freeze and they will NOT be able frames earlier than other in steady state.
Along the similar notion,
$T_C > T_B$ - (3)
From (1), (2) and (3),
$\Rightarrow \boxed{T_A = T_C > T_B}$

At any point of time; if

- A_1 sending frame to A_2:

✓ B_1 will hear A_1's RTS and A_2's RTS ∴ it will remain silent.

✓ C_1 will NOT hear A_1's RTS and A_2's RTS ∴ it can send a frame.

- B_1 sending frame to B_2:

✓ A_1 and C_1 both will hear the RTS and B_2's CTS and both will remain silent.

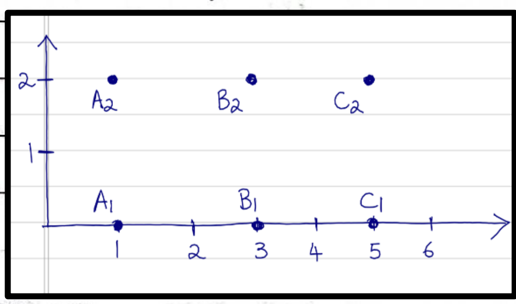
- C_1 sending frame to C_2:

✓ B_1 will hear C_1's RTS and C_2's CTS ∴ it will remain silent.

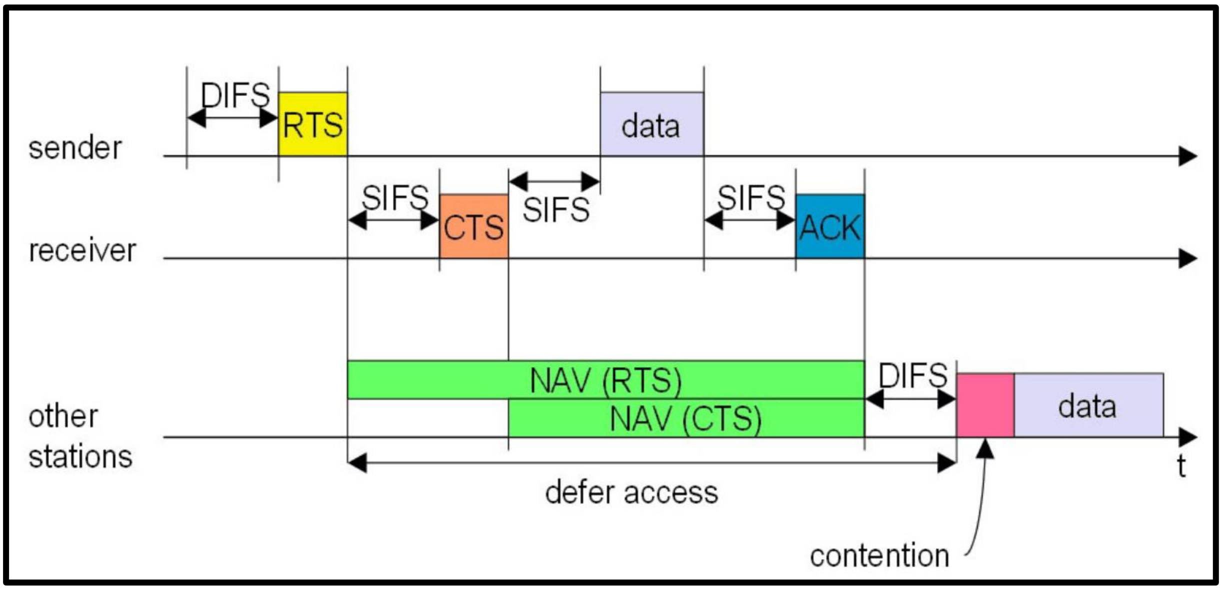
✓ A_1 won't hear any CTS/RTS ∴ it sends frame.

RULE: Any WiFi node upon hearing RTS or CTS should remain silent for the duration of NAV specified.

∵ in an idle carrier situation, all 3 (A_1, B_1, C_1) have frames to send, ∴ who sends the frame is decided randomly (as wait time ↔ contention window is RANDOM).
Effectively, B_1's throughput < that of A_1 and C_1 as it has to wait when either A_1 or C_1 or both are sending frame whereas A_1 and C_1 wait only if B_1 is sending a frame.



Wi-Fi Three-Pairs



IEEE 802.11 channel access using RTS/CTS method