CSMA/CA Backoff Algorithms: A GUI-driven Simulation and Analysis

by George Mensah, Tilak Murapilla, Robert Quainoo, and Soniya Kadam
Wireless and Network Engineering Program, Northeastern University, Boston, MA - USA

***Abstract*— The role of the backoff mechanism in Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA) protocols is critical in managing network contention and optimizing channel access. This project explores the various backoff mechanisms and their impact on contention window size within CSMA/CA networks. Through the development of a graphical user interface (GUI) on the Google Colab platform using Python, simulations of CSMA/CA are conducted, incorporating various backoff algorithms to determine specific use cases. The study focuses on three distinct backoff algorithms: Binary Exponential Backoff (BEB), Exponential Increase Exponential Decrease (EIED), and Logarithmic Increment Backoff. By simulating these algorithms within the context of CSMA/CA networks, this project provides insights into their effectiveness, adaptability, and overall influence on network performance. Through comprehensive analysis, the project aims to enhance understanding of backoff mechanisms in CSMA/CA protocols and their implications for network efficiency and reliability.**

Keywords*— CSMA/CA, Backoff Mechanism, Contention Window, Binary Exponential Backoff (BEB), Exponential Increase Exponential Decrease (EIED), Logarithmic Increment Backoff, Simulation, Graphical User Interface (GUI), Performance Evaluation

I. Introduction

Carrier-sense multiple-access with collision avoidance (CSMA/CA) protocols rely on the random deferment of packet transmissions for the efficient use of a shared wireless channel among many nodes in a network [**?**]. CSMA/CA protocol depends extensively on the backoff mechanism to manage network contention and optimize channel access; this class of medium access control (MAC) protocols is one of the most popular for wireless networks [2].

Distributed Coordination Function (DCF) is a MAC technique used in IEEE 802.11-based WLAN standards. It is a mandatory technique used to prevent collisions in wireless networks. DCF is used in areas where CSMA/CA is used. The technique involves the following steps: When a station has a frame to transmit, it waits for a random backoff time. If the channel is busy during the contention period, the station pauses its timer until the channel is clear. At the end of the backoff period, if the channel is idle, the station waits for an amount of time equal to Distributed Inter-Frame Space (DIFS) and then senses the channel again. If the channel is still clear, the station transmits an RTS (request to send) frame – at this point, the transmitting station sets its network allocation vector (NAV) to indicate the duration it expects to be occupied with the upcoming transmission. The destination station responds using a clear-to-send (CTS) frame if it is available, however before sending the CTS frame, the destination station adjusts its NAV to reflect the expected duration of the data transmission. Then the transmitting station sends the data frames. After sending the frames, the transmitting station waits for a time equal to SIFS (Short Inter-Frame Space) for acknowledgment. Finally, the station waits for the backoff time before the next transmission.

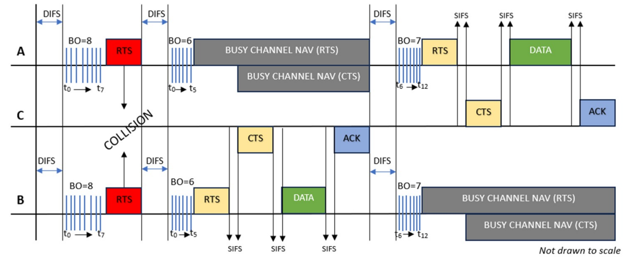

Fig. 1. Overview of CSMA/CA [1]

DCF includes Exponential Increase Exponential Decrease (EIED), Binary Exponential Backoff (BEB), and Logarithmic Increment (Log) algorithms used in backoff mechanisms. These backoff algorithms play a crucial role in managing contention and mitigating collisions in wireless networks, contributing to overall network efficiency and reliability.

II. **COMPARISON OF BACKOFF ALGORITHMS**

We attempt to compare the three backoff algorithms: Binary Exponential Backoff (BEB), Logarithmic Backoff (LB) and Exponential Increase Exponential Decrease backoff (EIED). To do this, we set up a simulation with the following constants:

A]Distributed Inter-Frame Space (in seconds) (DIFS) = 0.05
B]Short Inter-Frame Space (in seconds) (SIFS) = 0.01
C]Slot time (in seconds) = 0.02
D]Minimum contention window size (CWmin) = 15
E]Maximum contention window size (CWmax) = 1023
F]Maximum number of retransmission attempts = 7

A. Binary Exponential Backoff (BEB):

In binary exponential backoff, the contention window size is doubled after every unsuccessful attempt. In the initial attempt, the CWmin is used and the BEB selects a random slot from the next contention window (which is equal to the CWmin). In successive attempts, the contention window size will keep doubling for every collision till it reaches its maximum CWmax or there is a successful transfer. This is emulated in the formula: CW = min (2*CW, CWmax) The actual backoff value is randomly chosen from the range [0, CW-1] where CW is the current contention window size. Thus, in our setup, assuming the successful transfer of the packet occurs in the 7th attempt we observe that:

Attempt	CWmin before (slots)	Slot Time (s)	CWmin After
1 (collision)	15	0.3	30
2 (collision)	30	0.6	60
3 (collision)	60	1.2	120
4 (collision)	120	2.4	240
5 (collision)	240	4.8	480
6 (collision)	480	9.6	960
7 (success)	960	19.2	15

The slot time keeps increasing as the number of slots increases. Beginning with an initial contention window size of 15 slots, subsequent attempts witness exponential growth, doubling after each collision, up to a maximum threshold defined by CW_MAX. For instance, after the second collision, the contention window increases by CW = min (2 * CWmin, CWmax) = min (2 * 15, 1023) = min (30, 1023) = 30 slots. The slot time also increases by 30 * 0.02 = 0.6 seconds. After the seventh unsuccessful transmission attempt, the contention window size reaches CW = min (2 * CWmin, CWmax) = min (2 * 480, 1023) = min (960, 1023) = 960 slots. However, upon a successful transmission, the contention window size resets to its minimum value of 15 slots, initiating a fresh cycle of contention. Each transmission attempt is associated with a specific time interval, calculated based on the number of slots and the given slot time of 0.02 seconds, ensuring a coordinated and adaptive approach to channel access within the network. This entire process is shown in figure 2.

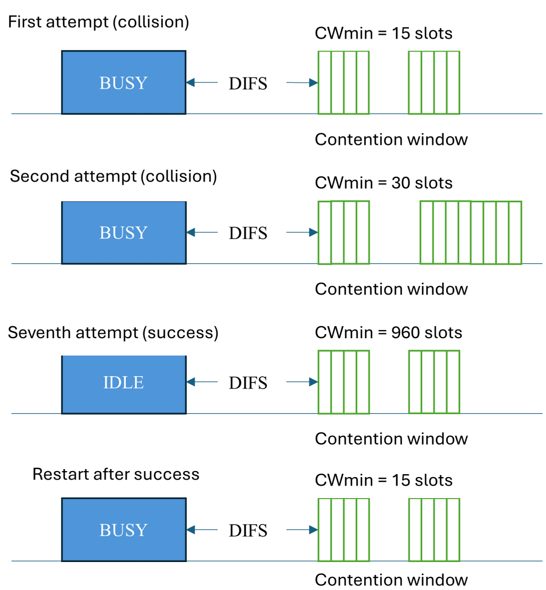

Fig. 2. BEB process

B. *Exponential Increase Exponential Decrease Backoff:*

In exponential increase exponential decrease backoff, the contention window is decreased and increased by factors. The initial attempt involves transmitting a packet using the CWmin. However, if the transmission is unsuccessful (a collision occurs), the contention window is increased by a backoff factor a. If the transmission after collision is successful, then the contention window is decreased by a backoff factor b. The EIED backoff algorithm can be represented by: CW = min (a*CW, CWmax) CW = max (CWmin/ b, CWmin) In EIED, the contention window size (CW) undergoes dynamic adjustments based on the outcome of transmission attempts. It increases the contention window size upon failed transmissions till it reaches the maximum contention window and reduces the contention window upon a successful transmission till it reaches the minimum contention window. In our setup, we assigned a=1.25 and b=0.8.

TABLE I
ATTEMPT AND CWMIN CHANGES

Attempt	CWmin before (slots)	Slot Time (s)	CWmin After
1 (collision)	15	0.3	19
2 (collision)	19	0.38	24
3 (collision)	24	0.48	30
4 (collision)	30	0.6	38
5 (collision)	38	0.76	48
6 (collision)	48	0.96	60
7 (success)	60	0.96	48

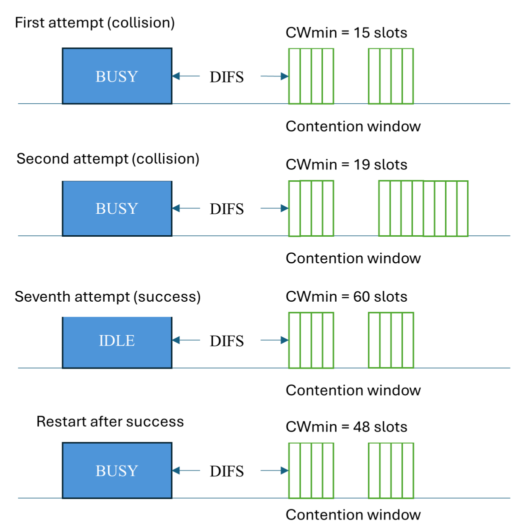

Fig. 3. BEB process

Upon a failed transmission attempt, the contention window size is increased by a factor of 1.25, gradually expanding to accommodate potential congestion and mitigate collisions. Conversely, when a transmission succeeds, the contention window size undergoes a reduction, decreased by 0.8, to promote efficient channel utilization and mitigate unnecessary delays. For instance, starting with an initial contention window size of 15 slots, it reaches CW = CWmin * 1.25 = 15 * 1.25 = 18.75 (rounded to 19) slots and reaches 60 slots after the sixth unsuccessful transmission. However, upon a successful transmission, the contention window size is revised back to 48 slots. These adjustments influence the duration of each transmission attempt, with the calculated number of slots translating to specific time intervals, ensuring a balance between contention window size and transmission efficiency. This entire process is shown in Figure 3.

C. Logarithmic Increment Backoff:

In logarithmic increment backoff, the waiting time before retransmission increases logarithmically with the number of successive collisions or failed transmission attempts. The purpose of this is to gradually increase the minimum contention window size in response to congestion or contention. A basic example of a formula for calculating the contention window size CW in a logarithmic backoff algorithm is: CW = CWmin + klog2(n) Where k is a scaling factor that determines the rate of increase of the contention window size and n is the number of consecutive unsuccessful transmission attempts (collisions). Log2(n) is the base-2 logarithm of n. In this formula, as n increases (indicating more collisions), the contention window size CW grows logarithmically with n. The scaling factor k determines the rate of increase, allowing for adjustments to the growth rate based on specific requirements and network conditions. Logarithmic backoff is highly dependent on the protocol used to generate it. This is because the protocol determines what happens after a successful transmission of a packet. There are two main protocols used to define what happens to CWmin: the first being CWmin gets reset to its base after a successful transmission (much like BEB) or CWmin gets assigned a random value. In our setup we considered both showing that CWmin can either be 15 (original value) or 28 (random value).

TABLE II
ATTEMPT AND CWMIN CHANGES

Attempt	CWmin before (slots)	Slot Time (s)	CWmin After
1 (collision)	15	0.3	15
2 (collision)	15	0.3	25
3 (collision)	25	0.5	30
4 (collision)	30	0.6	35
5 (collision)	35	0.7	38
6 (collision)	38	0.76	41
7 (success)	41	0.82	15 / 28

In a logarithmic backoff scenario, the contention window size evolves logarithmically with each unsuccessful transmission attempt, reflecting a strategic adaptation to network conditions. Commencing with a modest contention window size of 15 slots, subsequent attempts witness an incremental increase, following the logarithmic progression. For instance, after the seventh unsuccessful transmission attempt, the contention window size escalates to 41 slots, embodying the logarithmic growth pattern. Each transmission attempt is associated with a specific time interval, calculated based on the number of slots and the provided slot time of 0.02 seconds, ensuring a calibrated approach to channel access within the network. This process is shown in the diagram below:

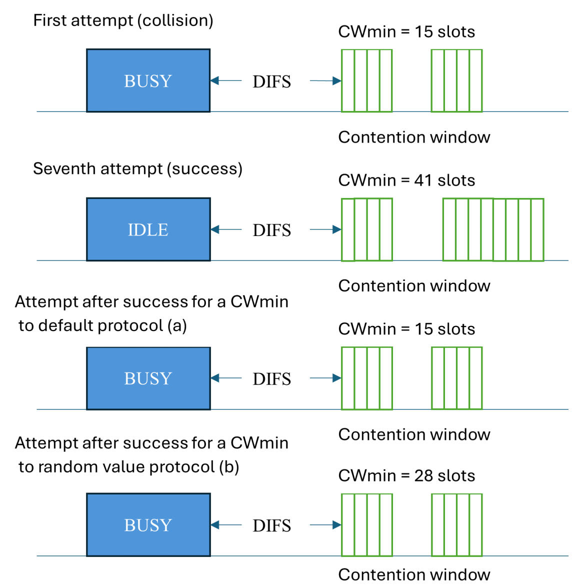

Fig. 4. BEB process

III. References

Abbott, S. (2019). Building a Cybersecurity Home Lab: Practical Guide for Beginners. Cybersecurity Journal, 8(2), 45-56.

Jones, M. P., and Smith, A. R. (2020). Virtualization Technologies in Cybersecurity Education: A Review of Current Practices. Journal of Information Security Education, 12(3), 112-125.

Johnson, T. W. (2018). Implementing pfSense Firewall in Home Networks: A Step-by-Step Guide. Home Networking Magazine, 15(4), 78-89.

Garcia, E., and Martinez, R. (2017). Setting Up a Domain Controller for Home Labs: Best Practices and Considerations. Home IT Solutions Journal, 6(1), 22-35.

Brown, K. L., and Wilson, B. S. (2019). Metasploitable: A Vulnerable Virtual Machine for Cybersecurity Training. International Journal of Cybersecurity Education, 4(2), 67-78.

White, L. E., and Davis, P. R. (2020). Practical Applications of Kali Linux in Cybersecurity Education. Journal of Ethical Hacking, 7(1), 34-47. Thompson, G. F. (2016). Designing and Building a Cybersecurity Home Lab: Challenges and Solutions. Cybersecurity Research Review, 3(2), 112-125.

National Institute of Standards and Technology. (2018). Guide to Industrial Control Systems (ICS) Security. NIST Special Publication, 800-82.

VirtualBox Documentation. (n.d.). Retrieved from https://www.virtualbox.org/manual/

Virtual Hacking Labs. (n.d.). Retrieved from https://www.virtualhackinglabs.com/

Cyber Range. (n.d.). Retrieved from https://cyberrange.com/

Offensive Security Proving Grounds. (n.d.). Retrieved from https://www.offensive-security.com/labs/

Hack the Box. (n.d.). Retrieved from https://www.hackthebox.eu/

Kim, P. (2018). The Hacker Playbook 3: Practical Guide to Penetration Testing. Publisher.

Murdoch, D. (Year). Blue Team Handbook: Incident Response Edition. Publisher.

REFERENCES

[1] F. Hong-yu, X. Lei, and L. Xiao-hui, "Logarithmic backoff algorithm of mac protocol in ad hoc networks," in *2010 International Conference on E-Business and E-Government*, 2010, pp. 1695–1698.

[2] G. D. Greenwade, "The Comprehensive Tex Archive Network (CTAN)," *TUGBoat*, vol. 14, no. 3, pp. 342–351, 1993.

[3] N.-O. Song, B.-J. Kwak, J. Song, and M. Miller, "Enhancement of ieee 802.11 distributed coordination function with exponential increase exponential decrease backoff algorithm," in *The 57th IEEE Semiannual Vehicular Technology Conference, 2003. VTC 2003-Spring.*, vol. 4, 2003, pp. 2775–2778 vol.4.

[4] S. Manaseer, "On backoff mechanisms for wireless mobile ad hoc networks," 01 2010.