Sliced Binary Exponential Backoff Algorithm For CSMA

Sidharth T R, Ankur Jat, Sandeep R, B. R. Chandavarkar
Department of Computer Science and Engineering
National Institute of Technology Karnataka
Surathkal, Mangalore, India
8590383683, 8000950925, 8105090281,9880270961
sidharthtr.221cs249@nitk.edu.in, ankurjat.221cs208@nitk.edu.in,sandeepr.221cs244@nitk.edu.in,brc@nitk.edu.in

***Abstract*—Carrier sense multiple access protocols in networking will help to detect and avoid collision and will wait for some back-off time if some collisions chances are there,which is calculated using a binary exponential back-off algorithm (BEB). Some random values will be selected from a range of 0 to $2^n - 1$ where n is the collision number. The algorithm helps in reducing collision but still has vulnerabilities like capture effect and increased overhead. In this paper, we are proposing a sliced binary exponential back-off algorithm that will reduce collision probability and capture the effect better than the traditional one. Representing results as tables using various examples of datasets and showcase the improvement through graphs.**

***Index Terms*—CSMA, Back-off algorithm, contention window, capture effect.**

I. Introduction

In the realm of computer networks, efficient utilization of shared communication channels is paramount. The Carrier Sense Multiple Access (CSMA) [1] protocol plays a pivotal role in managing access to these shared channels. Let's delve into the key components that constitute this fascinating field.CSMA/CD serves as the bedrock of Ethernet networks and CSMA/CA [2] enhances wireless networks like IEEE 802.11.With its accompanying binary backoff algorithm, CSMA ensures fair and efficient access to network resources.

The CSMA/CD protocol operates on the principle of nodes sensing the carrier to detect ongoing transmissions before attempting to transmit data. However, collisions may still occur when multiple nodes contend for channel access simultaneously. This inherent contention problem is mitigated by the binary backoff algorithm, which dynamically adjusts retransmission attempts to alleviate congestion and improve network efficiency.

One crucial aspect of CSMA [3]is the concept of contention windows. Contention windows dictate the duration during which nodes defer their transmissions after detecting a collision. Through the manipulation of contention windows, network administrators can fine-tune network performance and optimize throughput.

Moreover, the capture effect, a phenomenon inherent in CSMA/CD networks, influences the probability of successful transmissions [4]. Nodes with stronger signals or closer proximity to the receiver have a higher probability of capturing the channel, leading to asymmetrical access patterns and potential unfairness in resource allocation.

Understanding collision probability within CSMA/CD networks is fundamental to assessing their performance and scalability. Collision probability quantifies the likelihood of multiple nodes transmitting simultaneously, leading to data corruption and retransmissions.

In this paper, we are slicing the contention window values by collision number. i,e dividing contention window by n. We analyze the reason for the capture effect and collision probability through various examples and reduce it to better by using sliced BEB. If two stations are competing for transmitting data, our algorithm reduces collision probability even to half for some cases and gives better performance if two station's collision numbers are of prime numbers and not equal to each other. As of all we are mainly concentrating on the 2-node scenario and CSMA/CD.

The paper is organized as follows: Section II explains already existing works about BEB, Section III gives a proper explanation of BEB like how it works,advantages and disadvantages. Section IV gives a performance analysis of SBEB on collision probability and capture effect. Section V and Section VI explains result analysis using various examples and simulating respectively. Section VII refers about the conclusion of our works.

II. Related Works

Back-off algorithms are widely used in wireless mediums like IEEE 802.11 and wired mediums like ethernet also. So we can see a lot of related work about the backoff algorithm, especially in 802.11 networks.

A. Adaptive State Backoff Algorithm

[5] proposes an Adaptive State Backoff (ASBA)contention window control mechanism, which strengthens the performance of IEEE802.11 DCF.ASBA operates in three stages, adapting to network traffic flow by varying the size of the Contention Window using exponential and linear adjustments. In high-traffic conditions, CW is adjusted linearly to prevent transmission delays and maintain network efficiency. Retransmission Counter: ASBA employs a retransmission counter

to monitor packet success and failures, adjusting the CW size accordingly to optimize network throughput and reduce collision rates.

B. Modified binary exponential backoff algorithm

[6] Put forward an alpha binary exponential backoff algorithm(αBEB). The αBEB algorithm, an extension of the Binary Exponential Backoff (BEB) algorithm used in CSMA/CD networks, introduces a scalar factor α to dynamically adjust the contention window size after each successful transmission. Unlike the traditional BEB algorithm where the contention window size resets to a minimum value after successful transmissions, αBEB offers a more adaptive approach by scaling the window size with α, a scalar parameter chosen from the set 1, 0.8, 0.6, 0.4.

Following a successful transmission, the current contention window size is multiplied by α, and the resulting value is rounded to the nearest integer while ensuring it remains at least as large as the minimum contention window size (CWmin[AC]). This adjustment allows for a more nuanced control over channel access, aiming to decrease collision occurrences and subsequently reduce global retransmission attempts while enhancing overall network throughput.

C. Capture Avoidance Binary Exponential Backoff

[7]The Enhanced CABEB (Collision Avoidance Binary Exponential Backoff) algorithm represents a refinement of traditional BEB algorithms, offering enhanced collision resolution strategies and deterministic behavior in two-node network scenarios.

In CABEB, stations employ the Binary Exponential Backoff (BEB) algorithm for collision resolution in most cases. However, it introduces a novel approach for handling collisions that occur during the transmission of a second packet after successful transmission, provided the channel remains idle before the station's second transmission attempt.

Specifically, if a station encounters a collision while transmitting a second packet after successful transmission and the channel was idle before the second transmission, CABEB employs an enhanced backoff algorithm for collision resolution. Under this scenario:

- For the first collision of the second packet, CABEB imposes a fixed 2-slot backoff time.
- For the second collision of the second packet, CABEB imposes a fixed 0-slot backoff time.
- For subsequent collisions of the second packet, CABEB reverts to using the standard BEB algorithm for backoff.

Remarkably, CABEB exhibits deterministic behavior in two-node network environments, wherein the two nodes alternate their transmissions systematically. This deterministic behavior ensures efficient channel utilization and minimizes contention, thereby enhancing overall network performance and reliability.

By incorporating targeted collision resolution strategies and deterministic transmission patterns, CABEB offers a robust solution for managing collisions and optimizing channel access in small-scale network environments. Its adaptive approach to collision resolution contributes to improved network efficiency and throughput, making it a valuable enhancement to traditional BEB algorithms in modern networking contexts.

All of these works were motivating for us to come up with our idea. As of all we are morely concentrating on the 2-node scenario as mentioned in the CABEB method and on CSMA/CD.

III. Overview of Binary Exponential Back-off Algorithm

Based on the time domain, if two senders send some packets/frames at the same time there is a chance of collision at the receiver. If another node is heard, we wait for a while (usually random) for the node to stop transmitting before listening again for a free communications channel random wait time is scheduled based on the back-off algorithm [8] which helps to reduce the probability of collision.

$$Waitingtime = K * T_{(slot)}$$

K will be a random number range from 0 to $2^n - 1$.
Where n is the attempt number.

After a collision, time is segmented into discrete slots known as Tslot, each lasting 2t, where t represents the maximum propagation delay in the network. Nodes involved in the collision randomly select a value from the set 0, 1, referred to as the contention window [8] [6]. In the event of a subsequent collision due to the selection of the same integer, the contention window size is doubled, expanding to 0, 1, 2, 3. Nodes engaged in the second collision randomly choose another value from the set 0, 1, 2, 3 and await that duration before attempting retransmission. Before initiating transmission attempts, nodes listen to the channel and only transmit if the channel is idle. This ensures that the source with the smallest integer in the contention window successfully transmits its frame. Consequently, the Back-off algorithm establishes a waiting time for nodes involved in a collision, specifying how long a node should wait before retransmitting.

The random delay time is determined through an exponential back-off algorithm, commencing at a minimum value and doubling after each collision, with an upper limit. The hardware implementation of the back-off algorithm resides in the network interface card (NIC) of each device. Upon collision detection, the NIC generates a value lying between 0 and 2n where 2n is excluded, where n denotes the number of collisions. Subsequently, the NIC multiplies this value by the slot time to ascertain the delay time before retransmission. In the event of reaching the maximum number of retransmissions without successful transmission, the device reports a transmission error.

But as mentioned in Table I, table II, and Table III, still there is a chance of collision and we can observe the capture effect where the fewer nodes having less number of collisions have a higher probability of successful transmission.

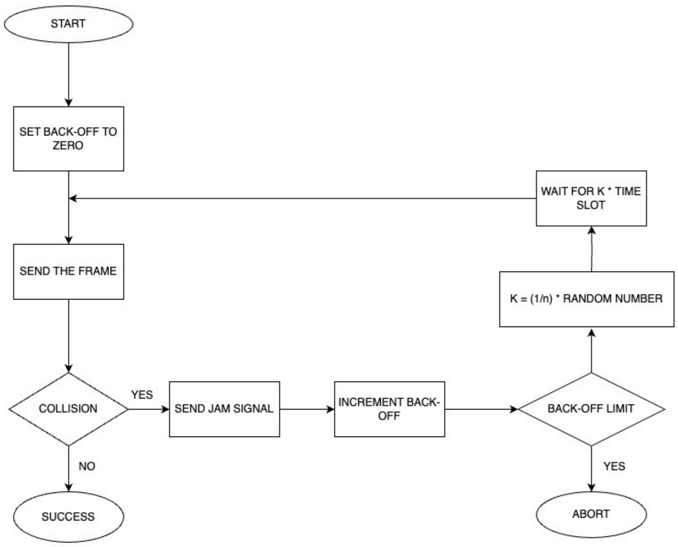

Figure 1: SBEB algorithm in CSMA

So through our paper, we are proposing SBEB algorithm that can minimize the collision probability and capture the effect up to more extent. And also reduces the overhead faced by the stations during transmission.

IV. Performance Investigation Of SBEB Algorithm

In our paper, we are mainly targeting to investigate on collision probability and capture effect.

A. Collision Probability

If two stations S1 and S2 are participating in the transmission of data within the range to each other. There is a chance of collision of data packets and data will get corrupted. So by using the back-off algorithm, stations may select random numbers K from the contention window set to avoid collision. So if both have collision number n equals to one. So CW = $\{0,1\}$.

So we can see that the probability of having the same number for both stations is half (½) which is also the collision probability of transmission. If we analyze Table I, Table II and Table III ,We can notice that the probability of having the same number is equal to the probability of having a collision.

So to reduce collision, we can solve this problem by reducing the probability of the station having the same number. For that, we are proposing the idea of dividing CW with collision number n.

$$K' = K/n$$

where K is the number selected from the contention window set.

So in the case of stations with the same value for n, we can't notice any change due to our algorithm. But in the case of stations with different collision numbers, we can clearly see the difference in values of both the stations as they are being divided by different values of n. So thus we can reduce collision probability more as compared to the traditional back-off algorithm.

B. Capture Effect

The phenomenon of successful transmission or winning of stations having less collision number n is called the capture effect. - The winning will keep on winning.

It is the main vulnerability of the back-off algorithm and CSMA.

Let's analyze Table III (BEB column),
we can see n1 = 2 and n2 = 3
CW of S1 = $\{0, 1, 2, 3\}$
CW of S2 = $\{0, 1, 2, 3, 4, 5, 6, 7\}$
From this, we can deduce that $\{4,5,6,7\}$ are not in S1 or we can finalize that 4 numbers in S2 are greater than S1.

So more than half the probability is there for S1 to have values that are less than S2 which results in more than half the probability of S1 to successfully transfer data before S1 So we can finalize that S2 needs to reduce the probability of selecting higher values than S1 to successfully transmit first But it has more high value in its CW set.

As we know n2>n1, so dividing the n value of S1 and S2 will result in a reduction of CW in S2 more than in S1 which is marked in Table III (SBEB column).

C. Why Dividing With n Is Optimum?

- If different stations have different collision numbers by dividing with n results in different values in the contention window set which helps in reducing collision probability.
- Stations with high collision have higher n values and a set of CW. So dividing by higher values (n) results in smaller values of CW which decreases waiting time together with decreasing capture effect.
- Reduces retransmission time as CW is being divided by n which results in small k and reduces waiting time.

V. Results And Analysis

In this paper, we are analyzing the working nature of exponential back-off algorithms using appropriate examples and proposing a better algorithm that helps in reducing the capture effect and decreasing collision probability.

A. BEB Algorithm With Example

Considering a data transmission of stations S1 and S2 are in the range of common router r. And we are applying a binary exponential back-off algorithm.

$$Waitingtime = K * T_{(slot)}$$

Where K is in the range of 0 to $2^n - 1$ (contention window) and n is the number of collisions that happen. in our case let's consider time slot = 1s; And collision number of S1 = n1 and S2 = n2.

If one collision had happened, then K=0,1

So waiting time of S1 and S2 = 0 or 1

Table 1 explains the results for S1 and S2 transmissions while both of the stations have a collision number of one (n1 = 1 and n2 = 1).

Based on Table 1, a total of four combinations are there and we can see two collisions are happening which results

Collision probability = 1/2
Data transmission success rate for S1 = 1/4
Data transmission success rate for S2 = 1/4

But if two collisions happen in the case of S1 (n1 = 2) and one collision happens in the case of S2 (n2 = 1).
Then , K = 0,1,2,3 for S1 and K=0,1 for S2 which mentioned in Table 2.

S1	S2	result(winner)
0	0	Collision
1	0	S2
0	1	S1
1	1	Collision

Table I: BEB Data Transmission(n1=1 and n2=1)

Based on Table 2, we can see again two collisions and a total of eight combinations are happening which is leading to

Collision Probability = 2/8
Data transmission success rate for S1 = 1/8
Data transmission success rate for S2 = 5/8

So we can see collision probability keeps on decreasing as n increases. But more interestingly we can see that the S2 success rate is more than the S1 success rate, where S2 has less number of collisions than S1. This is called the capture effect.

Winning node will keep on winning or will have a high probability of winning.

Lets consider one more situation where S1 has n = 2 and S2 has n = 3;

S1	S2	result(winner)
0	0	Collision
1	0	S2
2	0	S2
3	0	S2
0	1	S1
1	1	Collision
2	1	S2
3	1	S2

Table II: BEB vs SBEB Data Transmission (n1=2 and n2=1)

BEB			SBEB		
S1	S2	Result	S1	S2	Result
0	0	Collision	0	0	collision
1	0	S2	0.5	0	S2
2	0	S2	1	0	S2
3	0	S2	1.5	0	S2
0	1	S1	0	0.3	S1
1	1	Collision	0.5	0.3	S2
2	1	S2	1	0.3	S2
3	1	S2	1.5	0.3	S2
0	2	S1	0	0.6	S1
1	2	S1	0.5	0.6	S1
2	2	Collision	1	0.6	S2
3	2	S2	1.5	0.6	S2
0	3	S1	0	1	S1
1	3	S1	0.5	1	S1
2	3	S1	1	1	collision
3	3	Collision	1.5	1	S2
0	4	S1	0	1.3	S1
1	4	S1	0.5	1.3	S1
2	4	S1	1	1.3	S1
3	4	S1	1.5	1.3	S2
0	5	S1	0	1.6	S1
1	5	S1	0.5	1.6	S1
2	5	S1	1	1.6	S1
3	5	S1	1.5	1.6	S1
0	6	S1	0	2	S1
1	6	S1	0.5	2	S1
2	6	S1	1	2	S1
3	6	S1	1.5	2	S1
0	7	S1	0	2.3	S1
1	7	S1	0.5	2.3	S1
2	7	S1	1	2.3	S1
3	7	S1	1.5	2.3	S1

Table III: BEB vs SBEB Data Transmission (n1=1 and n2=1)

In Table III, we are comparing BEB and SBEB with n1 = 2 and n3 =3.
After plotting values in Table 3, we can see total of 32 combinations are there. For BEB we are getting four collisions results,

Probability of collision = 4/32
Data transmission success rate for S1 = 22/32
Data transmission success rate for S2 = 6/32

From these analyses also, we can see how much the capture effect is affecting the successful transmission of S2 due to its high value of collision number.

B. SBEB algorithm with example

In this paper, we put forward an approach of dividing each contention window or K value with n
So,

$$K' = K/n \tag{1}$$

By applying (1), we have created the SBEB column of Table 3, which helps to compare how our algorithm is better than the traditional one. By analyzing Table 3 SBEB column, We can notice that only two collisions happens where four collisions was there in BEB.So for SBEB algorithm,

Probability of collision = 2/32

Data transmission success rate for S1 = 20/32

Data transmission success rate for S2 = 10/32

From Table III and its analysis, we can see that dividing with collision numbers is helping to reduce collision probability even to half of its old value and decreasing the capture effect by increasing the data transmission success rate of stations (S2 in the above situation) having more collisions.

VI. Simulation Results

In order to prove that our proposed SBEB algorithm performs well in all collision numbers except for some peculiar cases. We plotted graphs using matplotlib in Python.Source code Github Link:https://github.com/Sidharthtr/Research-Paper-SBEB/blob/main/src/graph_plot.ipynb

We plotted for two nodes A and B where A have varying collision number from 1 to 10 and B has a constant collision number of 3.

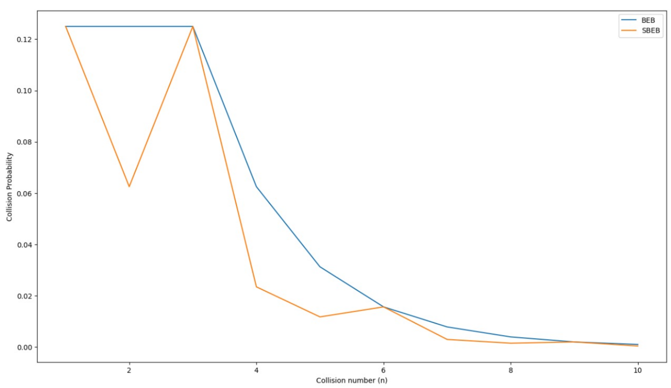

Figure 2: Collision Probability

In Figure 2, we plotted collision probability vs collision number where we can clearly see that the SBEB line(orange) comes below the BEB line(blue) which reduces collision probability for our proposed algorithm. Also, we can see no change in probability for n equals factors of 3 because both will have some of the same numbers to coincide due to mathematical logic.

In Figure 3, we plotted the success rate vs collision number to analyze the capture effect. As the value of n equals 3, we can see all four lines are intersecting which clearly states that in success rate of both stations are same and even no change with the new algorithm. Also, we can clearly see that from starting A(blue line) has a higher success rate than B(green line) as A has less value of n. At that time,SBEB algorithm is reducing the success rate of A(orange) and increasing B(red) which enhances the existing algorithm. The Vice-versa situation happens when the collision number of A starts exceeding 3, which promises the prettier working of our algorithm.

From these simulations and example analysis, we can clearly see how adaptive and better the SBEB algorithm performs. It performs well and decreases both capture effect and collision probability to increase throughput during transmission.

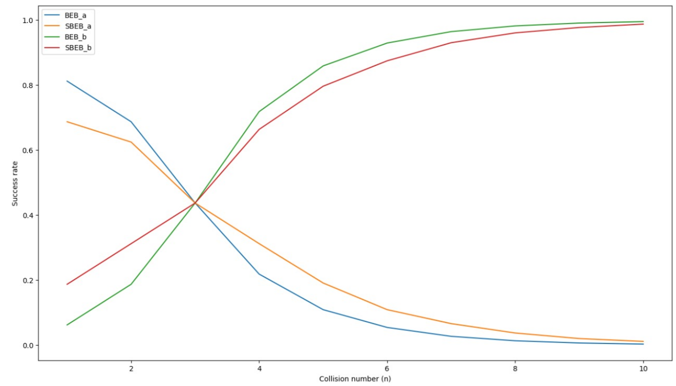

Figure 3: Capture Effect Analysis

VII. Conclusion And Future Work

From the above analysis, we conclude that the capture effect is increasing due to the high contention window of a higher collided station. So as n value of a higher collided station will also be high. So dividing the value of the contention window with the n value helps to minimize the contention window which decreases the capture effect and decreases collision frequency. By this method, we can optimize CSMA protocol in a better way in IEEE 802.3(ethernet). It will bring better results for higher differences in collision numbers concerning the normal algorithm.

For our future work, we would like to extend this approach to 802.11 devices with more than two stations. So it will enhance the algorithm to a more suitable level.

References

[1] M. Gamal, N. Sadek, Markov model of modified unslotted csma/ca for wireless sensor networks, https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9021836, [Accessed: 21-02-2024] (2019).

[2] J. H. A. B. V. S. P. Thiran, Csma/ca in time and frequency domains, https://ieeexplore.ieee.org/document/7437134, [Accessed: 21-02-2024] (2015).

[3] Wireless lan medium access control (mac) and physical layer (phy) specifications, https://ieeexplore.ieee.org/document/9390505, [Accessed: 21-02-2024] (2020).

[4] N. R. . C. Mala, Analysis and comparative study of different backoff algorithms with probability-based backoff algorithm, https://link.springer.com/chapter/10.1007/978-3-642-14478-3_34, [Accessed: 21-02-2024] (2010).

[5] M.-H. C. C.-I. C. W.-S. H. Y.-J. W. C.-H. Lin, An adaptive state backoff algorithm for wireless mesh networks, https://ieeexplore.ieee.org/document/8394428, [Accessed: 21-02-2024] (2018).

[6] A modified binary exponential backoff algorithm for improving the quality of service in highly populated ieee 802.11e networks, https://ieeexplore.ieee.org/document/6651192, [Accessed: 21-02-2024] (2013).

[7] K. K. Ramakrishnan, H. Yang, The ethernet capture effect: Analysis and solution, https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=386597, [Accessed: 21-02-2024] (1994).

[8] M. M. S. M. F. A.-M. Z. A. Alomari, Mac layer back-off algorithm for ad hoc networks, https://ieeexplore.ieee.org/document/6596299, [Accessed: 21-02-2024] (2013).

****** END ******

Note:

- References to be referred to within the content as [1]–[8]