International Journal of Technology 14(2) 399-410 (2023)
Received May 2022 / Revised August 2022 / Accepted January 2023

International Journal of Technology

Performance Evaluation of Triangular Number Sequence Backoff Algorithm for Constrained Application Protocol

Chanon Jangkajit[1], Chanwit Suwannapong[2*]

[1]*Department of Computer Network Engineering, Faculty of Industrial Technology, Ubon Ratchathani Rajabhat University, 2 Nai Mueang, Mueang Ubon Ratchathani, Ubon Ratchathani 34000, Thailand*
[2]*Department of Computer Engineering, Faculty of Engineering, Nakhon Phanom University, 214 Nongyard, Mueang Nakhon Phanom, Nakhon Phanom 48000, Thailand*

Abstract. In the of the Internet of Things (IoT) realm, congestion is considered a serious issue affecting network throughput due to the requirement of multiple nodes for message exchange. With free-space optical communications, which can help send a message wirelessly, congestion control mechanisms nowadays depend upon the carrier sense of multiple access with collision avoidance (CSMA/CA), using the so-called Backoff Algorithm (BA). These algorithms, including Binary Exponential Backoff (BEB), Enhanced Fibonacci Backoff (EFB), Estimation-Based Backoff (EBA), and Backoff Interval Isolation (BII), have been introduced to facilitate congestion control mechanisms. Implementing such algorithms, however, might not deliver the greatest performance for Constrained Application Protocol (CoAP) which typically operates under limited resources. Therefore, the present study aims to introduce a new backoff algorithm, Triangular Number Sequence Backoff (TNSB), and to compare its performance with that of the aforementioned algorithms under the Transmission Control Protocol (TCP) and the Cooja network simulator. Statistical analysis involves ANOVA (F-test) and post hoc multiple comparison tests. The study shows that its performance is not significantly different from the others at the low congestion level. At the middle and the high congestion levels, it yields the highest throughput with the shortest settling time, while the packet loss rate and the response time are satisfactory.

Keywords: Backoff algorithm; Constrained application protocol; Performance evaluation; Triangular number sequence

1. Introduction

CoAP, a well-known application layer protocol, operates in devices that have limited resources, thus allowing a client to request action on network resources on a server. It has long been developed to be paired with numerous Internet of Things (IoT) applications. (Chin *et al.*, 2022; Ong, Connie, and Goh, 2022; Jonny and Toshio, 2021). The protocol implements four method codes, including HTTP GET, PUT, POST, and DELETE, through reliable and unreliable message transmissions (IETF, 2014). Figure 1 below shows the reliable message transmission whose performance resembles the Transmission Control Protocol (TCP). A request in a confirmable (CON) message is transferred to a target server which, later, sends back an Acknowledgment (ACK) message. This is known as a piggy-

*Corresponding author's email: schanwit@npu.ac.th, Tel.: +66-82-8488802; Fax.: +66-42-503558
doi: 10.14716/ijtech.v14i2.5686

backed response designed to prevent packet loss from network congestion.

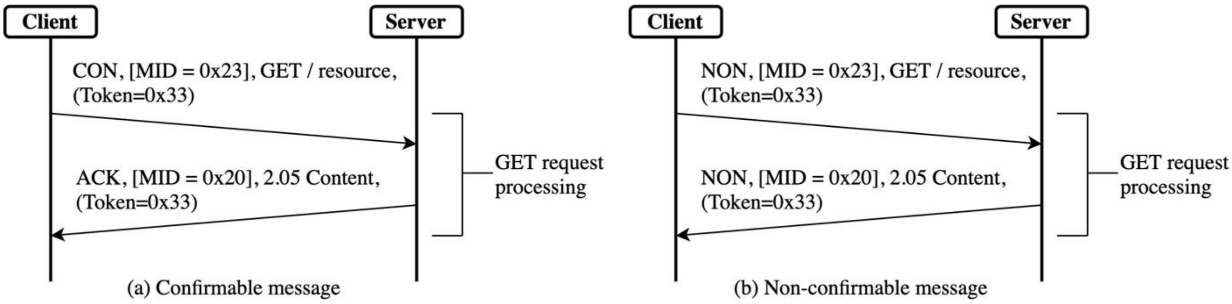

Figure 1 Message transmission patterns in the CoAP

Additionally, the CoAP relies upon a retransmission timeout (RTO) with a Binary Exponential Backoff (BEB) algorithm (Lee *et al.*, 2016). As regards unreliable message transmission, there are two possibilities. First, when the client sends a request with a non-confirmable (NON) message, the server receives the message and then sends back the NON-message instead. The other case is sending the NON-message for a repeated task without responses, e.g., using CoAP in the application that measures temperatures, and the data will be sent to the weather station. It can also be implemented for setting time to measure pressure in the compressed air system before the data are sent to the base station (Tariq *et al.*, 2020).

Nowadays, several backoff algorithms supporting congestion control mechanisms have been developed continuously. An RTO-computing algorithm, this BEB helps to lessen the probability of entities that request access together when the network cannot control the congestion. That is, its operation is based on delay time to avoid collisions. It is extensively used as a basis for analyzing and designing other backoff algorithms, which can also yield maximum throughput (Cheng *et al.*, 2014; Kang, Cha, and Kim, 2010; Yassein *et al.*, 2010). However, there are some limitations to this algorithm. First, collision probability normally increases when the number of active nodes rises, thus lowering the throughput. Second, when there is no response in the ACK message, retransmission takes longer if the RTO is high. This adversely impacts the throughput since the response time is greater than expected. Third, the RTO is greater when the congestion increases, resulting in multiple retransmissions. These limitations of the BEB indicate that when network congestion occurs continuously, the quality of communication suffers. Even worse, the congestion may continue rising and finally result in a network collapse.

This research article aims to introduce the new backoff algorithm, Triangular Number Sequence Backoff (TNSB), which uses an arithmetic sequence whereby numbers are represented in an equilateral triangle arranged in a sequence. It is expected to outperform the BEB, the EFB, the EBA, and the BII. The performance analysis of such algorithms is conducted in continuous, periodic, and bursty traffic scenarios and by considering four performance metrics, i.e. throughput, packet loss, response time, and settling time.

The rest of this paper is organized as follows. Section 2 provides the literature review with an emphasis on different backoff mechanisms once proved to have the potential to reduce collisions in the TCP. Section 3 introduces our proposed TNSB algorithm. Section 4 shows simulation parameters, network topologies, traffic scenarios, and performance metrics employed to evaluate the performance of all backoff algorithms. The results are presented in Section 5, and the conclusion is given in Section 6.

2. Literature Review

This section gives information regarding the CoAP and backoff algorithms, including the BEB, the EFB, the EBA, and the BII, which depend upon the delay time for preventing collisions in the Wireless Local Area Network (WLAN) in the TCP. These algorithms are used to compare with the TNSB in terms of performance in different scenarios.

The technique implemented for controlling the congestion of message transmissions, like the aforementioned CoAP, relies upon the RTO setting for retransmissions. An increase or decrease in the RTO mainly affects the traffic congestion in the network and communication efficiency. Therefore, in this section, different backoff algorithms corresponding to different RTO values are presented as follows:

2.1. CoAP

This protocol, widely known as default CoAP, involves an estimation of the RTO for message retransmission when the client does not receive any response from the server to avoid congestion. In a normal situation, the client sending the CON message waits for and obtains the ACK message from the server. However, should there be network congestion, retransmission occurs. In this protocol, the RTO estimate is based on the BEB algorithm. The initial RTO (RTO_{init}) is randomly picked from the interval of 2-3 seconds before transmitting the CON message to the server. An RTO expiration without the ACK message response can cause the overall RTO ($RTO_{overall}$) to double.

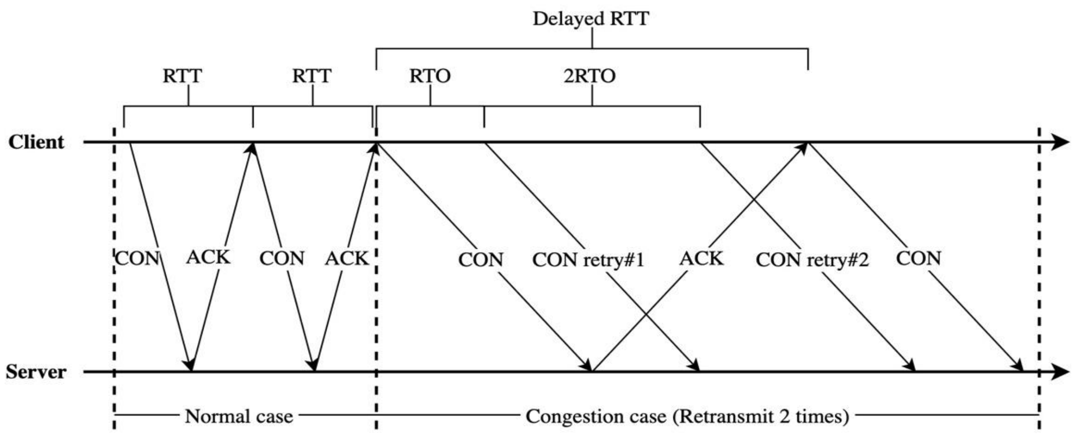

Figure 2 The default CoAP scheme

2.2. BEB

This backoff algorithm facilitates the default CoAP in yielding the suitable RTO. It was first proposed for computer networking, and its performance has been compared to other algorithms (Al-Fuqaha *et al.*, 2015). Due to its less complicated operation, the overall performance tends to be satisfactory. The algorithm starts with the source node randomly selecting the RTO_{init} before sending the CON message. As mentioned earlier, when the RTO expires without receiving the ACK message, there comes the retransmission, and the $RTO_{overall}$ is doubled. In this regard, the maximum number of retransmissions of four. Figure 3 illustrates the transitions in the backoff stage in the BEB algorithm, along with the calculation and increasing rates of the RTO in Table 1. Here, the i value which ranges from 1 to 4, refers to the number of collision events (i^{th}) the node detects. The maximum RTO in the 4^{th} retransmission is 48 seconds (if the RTO_{init} is 3 seconds) for the same message ID (MID). After the message exchange, whether successful or not, the RTO returns to its initial value randomly chosen from the interval of 2-3 seconds. However, the collision problem and packet loss are likely to occur when more than two source nodes communicate together and their random RTO_{init} is the same.

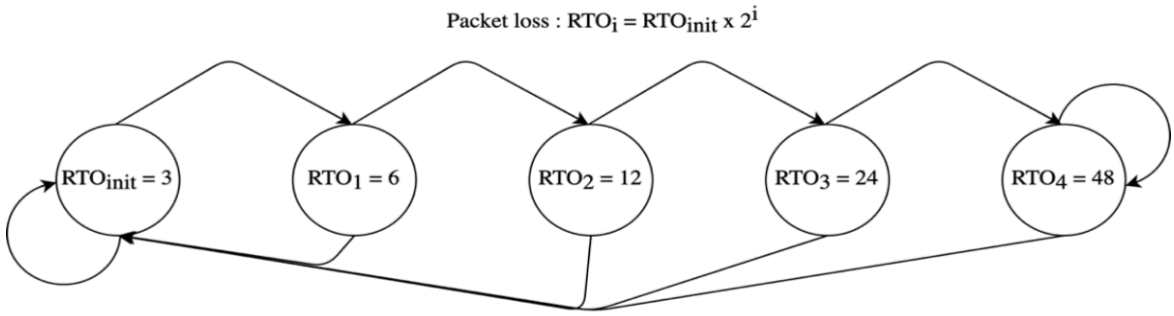

Figure 3 The BEB scheme

Table 1 Binary Exponential Backoff (BEB) algorithm

Algorithm 1. BEB
1. Initialize random value from [2 s, 3 s] to RTO_{init}
2. when transmitting CON
3. RTO = RTO_{init}
4. for i = 1 to 4
5. if RTO expires without having received an ACK
6. RTO = $RTO_{init} * 2^i$
7. i = i+1
8. else
9. return transmission success
10. end for
11. return transmission fail

2.3. EFB

The EFB applies a Fibonacci sequence to a linear algorithm to enhance the efficiency of the Distributed Coordination Function (DCF). Due to its simplicity, this algorithm can operate under different conditions and yield maximum system throughput as well as minimum packet delay (Yassein *et al.*, 2010). Its operation at the initial stage does not differ from that of the BEB; that is, if the client does not receive the ACK message within the specified time, the retransmission occurs. However, the new RTO is calculated using the Fibonacci sequence (fib(i)), as seen in Figure 4 and Table 2.

In Table 2, i is randomly picked as 1, 2, 3, and 4. During the message exchange, if the 4th retransmission happens and the random RTO_{init} is 3 seconds, the RTO of the last transmission is equivalent to fib(4), which is 10 seconds (fib(4) = 3+1+1+2+3). This is considered very small if compared to that of the BEB.

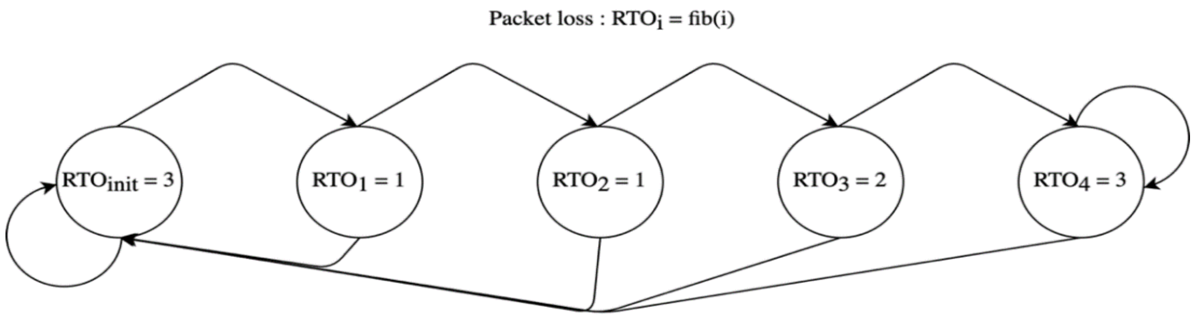

Figure 4 The EFB scheme

Table 2 Enhanced Fibonacci Backoff (EFB) algorithm

Algorithm 2. EFB
1. Initialize random value from [2 s, 3 s] to RTO_{init}
2. Initialize Fibonacci Series fib to [0, 1, 1, 2, 3]
3. when transmitting CON
4. RTO = RTO_{init}
5. for i = 1 to 4
6. if RTO expires without having received an ACK
7. RTO = fib(i)
8. i = i+1
9. else
10. return transmission success
11. end for
12. return transmission fail

2.4. EBA

This newly developed algorithm evaluates system status and selects a backoff period corresponding to the current network condition. According to (Kang, Cha, and Kim, 2010), it determines the number of nodes based on idle slots during the backoff. To clarify this point, its operation is initiated by the source node randomly selecting the RTO_{init}. When the RTO is equal to the number of nodes in the case of retransmissions, the throughput is at its peak. This algorithm, therefore, works statically. The diagram of the transitions in the backoff stage for the EBA is shown in Figure 5, and the RTO set for the EBA can be drawn in Table 3. For the latter, the RTO of each retransmission equals the number of nodes in the network. Like other backoff algorithms, the maximum number of retransmissions is four. When the number of nodes varies dramatically, the RTO turns higher, affecting the idle time.

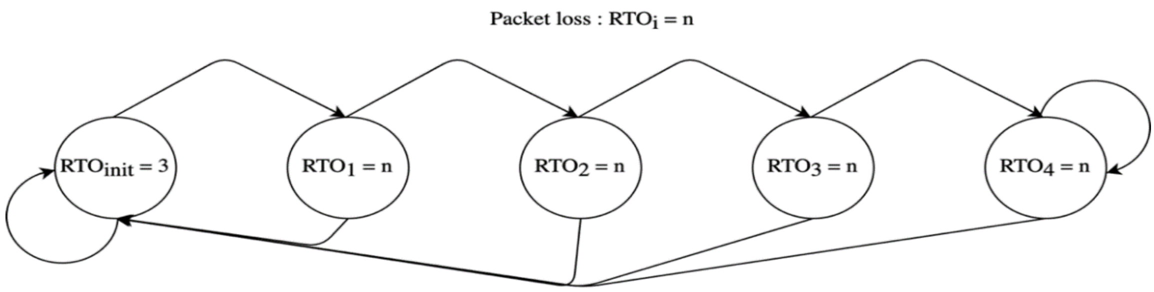

Figure 5 The EBA scheme

Table 3 Estimation-Based Backoff (EBA) algorithm

Algorithm 3. EBA
1. Initialize random value from [2 s, 3 s] to RTO_{init}
2. Initialize the number of nodes to n
3. when transmitting CON
4. $RTO_{overall}$ = RTO_{init}
5. for i = 1 to 4
6. if RTO expires without having received an ACK
7. $RTO_{overall}$ = n
8. i = i+1
9. else
10. return transmission success
11. end for
12. return transmission fail

2.5. BII

This backoff algorithm is proposed due to the limitations of the BEB; that is, the BEB suffers from high collision probability in the node with overlapped backoff intervals, and too many nodes also cause repeated collisions and the increasing number of retransmissions until the RTO is sufficient. This wastes time and bandwidth accordingly (Cheng *et al.*, 2014). The RTO of this algorithm comes from the Random integer (R) multiplied by the length of a time slot (T). In the case of retransmissions, our study adjusts the RTO by multiplying the R with the RTO_{init}. The details of this BII are presented in Table 4. When the RTO expires without receiving the ACK message, the BII randomly picks the RTO in each round of retransmission with different values for retransmission, and the RTO is at its peak when the R is equal to the RTO_{init}.

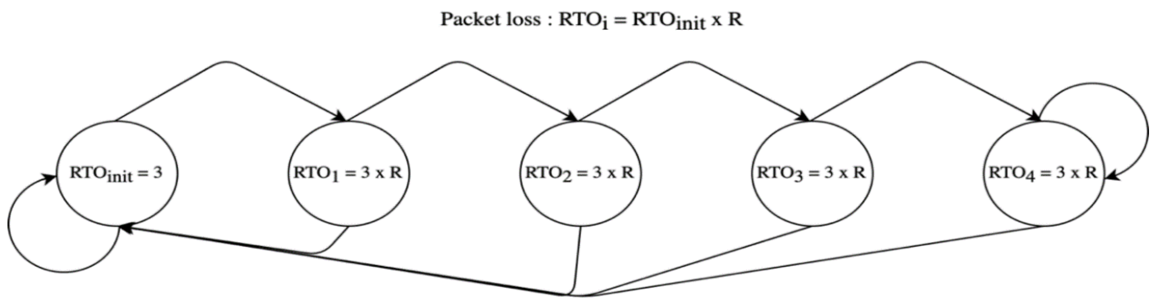

Figure 6 The BII scheme

Table 4 Backoff Interval Isolation (BII) algorithm

Algorithm 4. BII
1. Initialize random value from [2 s, 3 s] to RTO_{init}
2. Initialize random integer from [0, RTO_{init}]
3. when transmitting CON
4. RTO = RTO_{init}
5. for i = 1 to 4
6. if RTO expires without having received an ACK
7. RTO = RTO_{init} * R
8. i = i+1
9. else
10. return transmission success
11. end for
12. return transmission fail

3. TNSB

Our proposed backoff algorithm is motivated by the notion of a Figurate Number, i.e., a specific representation of dots formulating an image of an equilateral triangle. This algorithm relies upon the sum of the n^{th} triangular number (Aguayo-Alquicira *et al.*, 2020; Kane, 2009) to better retransmissions of the Default CoAP and overall system throughput. That is, it deals with the delay of message exchange and helps to lessen packet loss in a system. It can be selected as part of the default CoAP to substitute the former BEB algorithm, which demands reliable message transmission. This algorithm requires the delay from the RTO prior to the next retransmission, and the RTO corresponds to the sum of the Figurate Number, i.e., $Dn = n(n+1)/2$ where D_n refers to the number of dots in the equilateral triangle, as illustrated in Figure 7.

Should there be congestion, the algorithm determines the RTO in accordance with the triangular number sequence, i.e., 0, 1, 3, 6, and 10. During message exchange, the RTO is slowly increased at the beginning, but it rises quickly at the 3rd and the 4th retransmission to keep the overall congestion at the optimal level. At the low congestion level, this algorithm yields the normal response time for message exchange. However, when the congestion level is high, the response time increases. The diagram of the transitions in the backoff stage for the TNSB is shown in Figure 8.

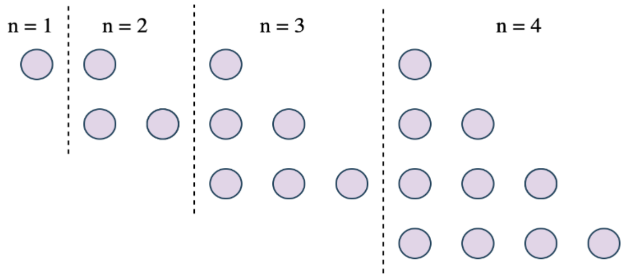

Figure 7 The number of dots that forms an image of an equilateral triangle

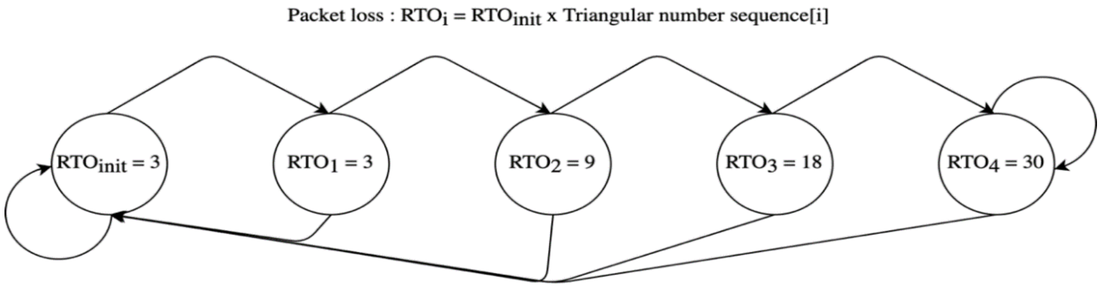

Figure 8 Our proposed TNSB scheme

Table 5 Triangular Number Sequence Backoff (TNSB) algorithm

Algorithm 5. TNSB
1. Initialize random value from [2 s, 3 s] to RTO_{init}
2. Initialize the Triangular number sequence to [0, 1, 3, 6, 10]
3. when transmitting CON
4. RTO = RTO_{init}
5. for i = 1 to (size of Triangular number sequence) - 1
6. if RTO expires without having received an ACK
7. RTO = RTO_{init} * Triangular number sequence[i]
8. i = i+1
9. else
10. return transmission success
11. end for
12. return transmission fail

Table 5 demonstrates how the RTO of the TNSB is set. It starts with the RTO_{init}, but when the retransmission occurs, the previous RTO is calculated with the RTO_{init} multiplied by the Triangular number sequence array. This algorithm also allows a maximum of four times for retransmission before the failure of message exchange is detected. The RTO, which is increased slowly in each round, helps to determine the waiting time (delay) for the retransmission and therefore lessens the probability of network congestion. For the increase in the RTO in each round, if the retransmission occurs, the traffic congestion or the number of nodes may increase. However, if the traffic congestion is at a high level continuously, some other techniques should also be implemented for more efficiency of the

TNSB, such as Buffer management, Explicit congestion notification (ECN), and Load balancing.

4. Network Simulation

4.1. An Experiment on the RTO in each Backoff Algorithm

This experiment is purposively designed to compare the RTO growth, ranging from the CON message transmission to the response in the ACK message in the 4th retransmission in all five backoff algorithms. The growth is observed when the RTO_{init} is fixedly set as 1.5 and 3 seconds for each backoff algorithm. This wider range of the RTO_{init} (1.5) can help to see the differences in the RTO growth clearly.

4.2. Performance of Five Backoff Algorithms under Different Traffic Scenarios

The performance evaluation is investigated in a continuous traffic scenario, a periodic traffic scenario, and a bursty traffic scenario with the Cooja simulation, which is operated by Contiki (Dunkels, Gronvall, and Voigt, 2004). The Z1 mote serves as the main module in the network (Zolertia, 2010). This includes a border router, a client, and a server, along with grid network typologies of 4 (2x2) (the dashed line), 9 (3x3) (the dotted line), and 16 (4x4) (the small, dotted line) in a two-dimensional scheme as seen in Figure 9. Each typology stands 10 meters away from the other, the transmission range is 15 meters, and the interference range is 30 meters, two times higher than the transmission range. The details of each parameter are also presented in Table 6.

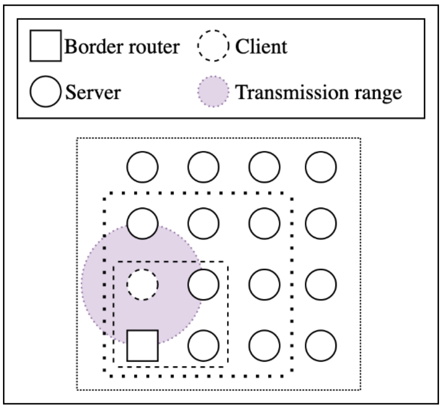

Figure 9 Grid network typologies for the simulation

Table 6 Parameters for the simulation

Settings	Value
Congestion mechanism	Default CoAP
Backoff algorithms	TNSB, BEB, EFB, EBA, BII
Wireless channel model	Unit disk graph model
Transport and network	UDP + uIPv6 + 6LoWPAN
Media access control	CSMA/CA
Radio duty cycling (RDC)	Null-RDC
Physical	IEEE 802.15.4 PHY
Radio band	2.4 GHz
RTO_{init}	1.5-3 s
Simulation time	360 s

4.2.1. Continuous traffic scenario

In this scenario, the server sends the CON message to the client. When the server receives the ACK message from the client, it immediately sends a new notification to the client. As the server with several active nodes sends the messages to the client at the same time, this can create a low congestion level. The evaluation concerns such performance metrics as throughput, packet loss, and response time.

4.2.2. Periodic traffic scenario

This periodic traffic scenario is used to explore the congestion control mechanism resulting from a continuous increase of congestion until it reaches the middle level. That is, the simulation initially creates a low congestion level where the server sends the CON message to the client with the speed of 1 message/s for 90 seconds, followed by sending the message with the speed of 2 messages/s for 90 seconds and with the speed of 3 messages/s until the end of the experiment. The throughput, the response time, and the settling time are not considered in this scenario since each round's message transmission speed is different. Packet loss is the only performance metric in this scenario.

4.2.3. Bursty traffic scenario

The congestion control mechanism under the bursty traffic scenario is initiated by the low congestion level from the server since the number of nodes in the server is reduced by half. For instance, if there are eight nodes in the server, the congestion is created from only four nodes, and the CON message is sent to the client for 180 seconds. After that, the bursty traffic is created by another server for 180 seconds, resulting in a high congestion level, and the parameter under investigation is the settling time.

5. Simulation Results and Analysis of Efficiency

5.1. Experimental Results of Comparing the RTO Growth in all Backoff Algorithms

The comparison of the RTO growth is illustrated in Figure 10. It is noticeable that the growth of the RTO in all backoff algorithms is in different patterns, even at the starting point. A clear distinction is found when the RTO_{init} is 3 s. For the TNSB, the RTO rises gradually to avoid the wait for the retransmission being too long, and the increase is obviously seen in the 4th and the 5th retransmissions. When the RTO_{init} is 1.5 s, the RTO increases at the middle level to avoid spurious retransmissions, compared to that of the others.

As regards the BEB, when the RTO_{init} is 1.5 s and 3 s, the RTO is higher than that of the TNSB, the BII, the EBA, and the EFB, respectively. For the CoAP in this study, the EFB is found with the shortest message retransmission, followed by the EBA, the BII, the TNSB, and the BEB, respectively. However, to account for the performance of each backoff algorithm, other aspects apart from the RTO should also be considered.

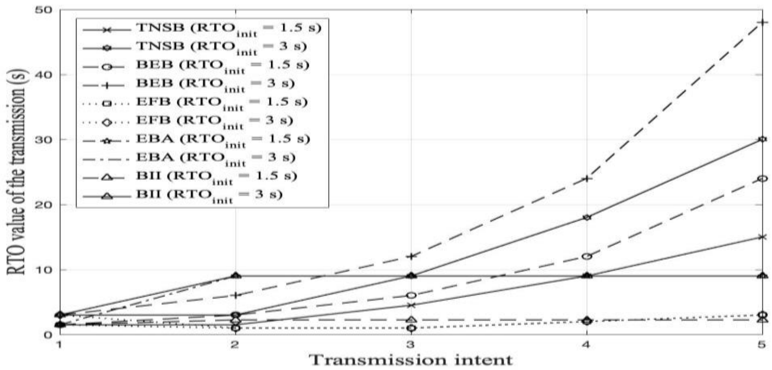

Figure 10 The growth of the RTO in the TNSB, the BEB, the EFB, the EBA, and the BII algorithms when the RTO_{init} is 1.5 s and 3 s

5.2. Experimental Results of Backoff Algorithm Performance in Different Traffic Scenarios

This section evaluates and discusses the performance of each backoff algorithm under different traffic scenarios. The number of connections in which the simulation is performed is 10, with different Seed numbers to obtain accurate mean scores of each performance metric. Based on the normal distribution, the statistical analysis involves hypothesis testing by means of ANOVA (F-test) and post hoc multiple comparisons from statistical software, namely Minitab.

(1) Hypotheses

H_0: $\mu_1 = \mu_2 = \mu_3 = \mu_4 = \mu_5$

H_1: Mean scores of the performance metrics are not equal in at least one pair of the backoff algorithms

When H_0 = There is no difference in mean scores of the performance metrics in backoff algorithms

H_1 = At least one pair of the backoff algorithms shows the difference in mean scores of the performance metrics

(2) The ANOVA test accounts for the behavior of performance metrics of all backoff algorithms in all scenarios. It involves the mean scores of the performance metrics at the significant level of 0.05 ($p \leq 0.05$).

(3) The post hoc multiple comparisons account for differences in the mean scores in each pair.

5.2.1. Experimental Results from Continuous Traffic Scenario

The comparison of the mean scores in the performance metrics, i.e., the throughput, the packet loss, and the response time in all backoff algorithms, is depicted in Table 7. The findings reveal that, from the ANOVA test in the 2x2 grid typology, there is a non-significant difference in the mean scores. In the 3x3 grid typology, however, the mean scores in all three parameters are significantly different. As such, the post hoc multiple comparisons indicate that, of all backoff algorithms, the TNSB yields the highest throughput (1.46 ± 0.24 message/s), the EBA exhibits the lowest percentage of packet loss (18.71 ± 2.01%), and the EFB is found with the shortest response time (1.28 ± 0.08 s). As regards the 4x4 grid typology, the mean scores of all three parameters also exhibit some significant differences. The post hoc multiple comparisons suggest that the TNSB, again, offers the best throughput (0.36 ± 0.04 message/s). Regarding packet loss, the EBA is found with the lowest percentage of packet loss (19.85 ± 2.93%). The EFB accounts for the shortest response time (2.63 ± 0.39 s).

Table 7 The performance metrics with 95% confidence intervals in different grid topologies

Traffic scenarios	Grid topologies	Performance parameters	TNSB	BEB	EFB	EBA	BII
Continuous	2x2	Throughput (message/s)	5.84 ± 0.62	5.81 ± 0.66	5.85 ± 0.51	5.92 ± 0.53	5.60 ± 0.55
		Packet loss (%)	0	0	0	0	0
		Response time (s)	0.17 ± 0.02	0.17 ± 0.01	0.17 ± 0.01	0.17 ± 0.01	0.18 ± 0.01
Continuous	3x3	Throughput (message/s)	1.46 ± 0.24	0.27 ± 0.12	0.78 ± 0.15	0.31 ± 0.14	0.62 ± 0.11
		Packet loss (%)	22.54 ± 2.20	29.90 ± 1.95	37.64 ± 3.34	18.71 ± 2.01	31.50 ± 2.27
		Response time (s)	1.89 ± 0.34	3.65 ± 0.38	1.28 ± 0.08	3.20 ± 0.34	1.64 ± 0.27
Continuous	4x4	Throughput (message/s)	0.36 ± 0.04	0.11 ± 0.05	0.27 ± 0.03	0.12 ± 0.02	0.21 ± 0.02
		Packet loss (%)	24.30 ± 2.47	38.82 ± 3.22	39.73 ± 3.61	19.85 ± 2.93	36.12 ± 2.84
		Response time (s)	3.10 ± 0.41	10.83 ± 0.46	2.63 ± 0.39	8.64 ± 0.69	3.42 ± 0.33

5.2.2. Experimental Results from Periodic Traffic Scenario

The comparison of the packet loss percentages in all backoff algorithms can be drawn in Table 8. The study finds that, in the 2x2 grid typology, there is no significant difference in packet loss. Based on the post hoc multiple comparisons findings, the EBA shows the lowest packet loss percentage in the 3x3 (19.31 ± 2.76%) and the 4x4 (22.05 ± 2.86%) grid typologies.

Table 8 Packet loss and 95% confidence intervals of the packet loss for different grid topologies

Traffic scenarios	Grid topologies	Performance metrics	TNSB	BEB	EFB	EBA	BII
Periodic	2x2	Packet loss (%)	0	0	0	0	0
	3x3	Packet loss (%)	26.58 ± 2.05	32.30 ± 2.53	38.24 ± 2.08	19.31 ± 2.76	32.10 ± 2.56
	4x4	Packet loss (%)	24.31 ± 2.39	38.82 ± 2.92	39.73 ± 2.52	22.05 ± 2.86	36.52 ± 2.79

5.2.3. Experimental Results from Bursty Traffic Scenario

The comparison of the settling time in all backoff algorithms reveals that, in the 2x2 grid typology, there is a non-significant difference in the settling time. However, in the 3x3 grid typology of all backoff algorithms, a significant difference in the settling time is found. From post hoc multiple comparisons, the TNSB requires the shortest settling time; nevertheless, there is no significant difference compared to that of the EFB. Regarding the 4x4 grid typology, there is a significant difference since the TNSB requires the shortest settling time (195.10 ± 2.32 s).

Table 9 Settling time and 95% confidence intervals of the settling times for different grid topology sizes

Traffic scenarios	Grid topologies	Performance metrics	TNSB	BEB	EFB	EBA	BII
Bursty	2x2	Settling time (s)	186.94 ± 0.37	187.47 ± 0.31	186.89 ± 0.34	187.43 ± 0.44	187.18 ± 0.32
	3x3	Settling time (s)	192.59 ± 2.01	200.75 ± 3.82	193.08 ± 2.81	201.30 ± 3.39	199.04 ± 2.87
	4x4	Settling time (s)	195.10 ± 2.32	202.97 ± 3.06	203.22 ± 2.73	206.14 ± 2.22	203.02 ± 2.93

From the performance evaluation of all backoff algorithms in both continuous and periodic traffic scenarios, it is worth noting that the TNSB has the highest throughput in 3x3 and 4x4 grid typologies because the RTO increases gradually at the initial stage and then rises dramatically during the 3rd and the 4th retransmissions, but its percentage of packet loss during the message exchange is higher than that of the EBA whose packet loss rate is the lowest. In the EFB, the RTO is lower in each retransmission and results in the shortest response time, but this algorithm cannot tackle the packet loss problem in the network. This can later cause the CoAP nodes to increase the retransmissions and have more congestion. The packets are more likely to be dropped, and this cannot save energy due to more retransmissions than usual.

In the bursty traffic scenario, the TNSB requires the shortest settling time in both 3x3 and 4x4 grid typologies. This also indicates the stable behavior as seen in the transition from the low congestion level to the bursty traffic or high congestion level. It can exchange the message well until the end of the experiment. Regarding the RTO in the other algorithms, it tends to be too high or too low, resulting in a longer settling time due to such bursty traffic.

6. Conclusions

This research article introduces the new backoff algorithm, namely the TNSB, and suggests how to enhance its performance in different performance metrics and traffic scenarios. It relies upon the arithmetic sequence to increase the RTO, directly affecting its overall performance. The study uses different network simulation scenarios to compare the performance of the TNSB to that of the BEB, the EFB, the EBA, and the BII by considering their throughput, packet loss, response time, and settling time. It reveals that, in the middle and the high congestion levels, the TNSB yields the highest throughput with the shortest settling time. On the other hand, the RTO growth issue shows a longer backoff duration compared to that of the EFB, the EBA, and the BII. To implement a suitable backoff algorithm for the CoAP, communication characteristics, network typologies, and other mechanisms also need to be considered in yielding the most effective congestion control mechanism.

References

Aguayo-Alquicira, J., León-Aldaco, S.E.D., Calleja-Gjumlich, J.H., Claudio-Sánchez, A., 2020. Switching angles calculation in multilevel inverters using triangular number sequence– A THD minimization approach. *European Journal of Electrical Engineering*, Volume 22(1), pp. 49–55

Al-Fuqaha, A., Guizani, M., Mohammadi, M., Aledhari, M., Ayyash, M., 2015. Internet of things: a survey on enabling technologies, protocols, and applications. *IEEE Communications Surveys & Tutorials*, Volume 17(4), pp. 2347–2376

Cheng, H., Yan, X., Lian, H., Weng, L., Zhang, Q., Feng, Z., 2014. A novel collision avoidance algorithm for IEEE 802.11 Wireless LANs. *In:* Proceedings of the IEEE Military Communications Conference 2014, pp. 879–884

Chin, C.G., Jian, T.J., Ee, L.I., Leong, P.W., 2022. IoT-based indoor and outdoor self-quarantine system for COVID-19 patients. *International Journal of Technology,* Volume 13(6), pp. 1231–1240

Dunkels, A., Gronvall, B., Voigt, T., 2004. Contiki - a lightweight and flexible operating system for tiny networked sensors. *In:* Proceedings of the 29th Annual IEEE International Conference on Local Computer Networks 2004, pp. 455–462

Internet Engineering Task Force (IETF), 2014. The constrained application protocol (CoAP). Available Online at: https://tools.ietf.org/html/rfc7252, Accessed on November 23, 2020

Jonny, K., Toshio, M., 2021. Building implementation model of IoT and big data and its improvement. *International Journal of Technology*, Volume 12(5), pp. 1000–1008

Kane, B., 2009. Representing sets with sums of triangular numbers, *International Mathematics Research Notices*, Volume 2009(17), pp. 3264–3285

Kang, S., Cha, J., Kim, J., 2010. A novel estimation-based backoff algorithm in the IEEE 802.11 based wireless network. *In:* Proceedings of the 7th IEEE Consumer Communications and Networking Conference 2010, pp. 1–5

Lee, J.J., Chung S.M., Lee B., Kim K.T., Youn H.Y., 2016. Round trip time based adaptive congestion control with CoAP for sensor network. *In:* International Conference on Distributed Computing in Sensor Systems (DCOSS), pp. 113–115

Ong, Y.Q., Connie, T., Goh, M.K.O., 2022. A cow crossing detection alert system. *International Journal of Technology*, Volume 13(6), pp. 1202–1212

Tariq, M.A., Khan, M., Raza-Khan, M.T., Kim, D., 2020. Enhancements and challenges in CoAP- a survey. *Sensors*, Volume 20(21), p. 6391

Yassein, M.B., Oqaily, O.A., Min, G., Mardini, W., Khamayseh, Y., Manaseer, S.S., 2010. Enhanced fibonacci backoff algorithm for mobile Ad-Hoc network. *In:* Proceedings of the 10th IEEE International Conference on Computer and Information Technology 2010, pp. 749–754

Zolertia, 2010. Z1 Datasheet. Available Online at: http://zolertia.sourceforge.net/wiki/images/e/e8/Z1_RevC_Datasheet.pdf, Accessed on November 23, 2020