Study on Improved Truncated Binary Exponential Back-off Collision Resolution Algorithm

Yongfa Ling and Deyu Meng

Faculty of Science, Xi'an Jiaotong University, Xi'an 710049, China

Abstract: The truncated binary exponential back-off algorithm has been widely applied in the collision resolution process of random multi-access channel. Based on the analysis of its basic mechanism, one improved algorithm to set initial window dynamically, and the other to set initial and end window dynamically, were proposed. The experimental results indicated that these improved algorithms were stable and effective, and had higher resolution efficiency and better throughput curve than the basic algorithm.

Key words: random multi-access channel; binary exponential back-off algorithm; collision resolution; time slot

Multi-access is an access control protocol of common channel that is shared by multi-user. Its logic topology shows as figure

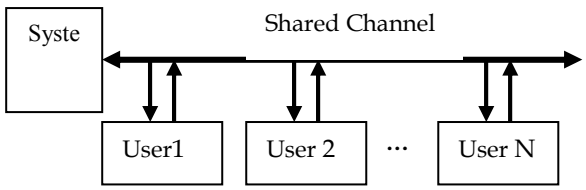


Fig. 1 Multi-access System

1. It has been widely applied in various co-

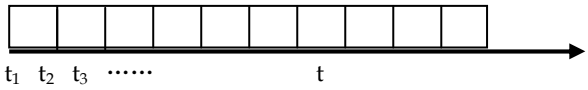


Fig. 2 Discrete Time Channels

-mmunication systems[1], such as satellite communication system , mobile communication system , local area network(LAN) and metropolitan area network(MAN). Multi-access chiefly follows the three communication protocol models: fixed allocation, allocation according to needs, random contention. Specially, the random contention model has been widely applied due to its effective channel resource utilization and reducing switch delay under certain condition[2-3]. ALOHA[4] and slotted ALOHA[5] which were proposed by N. Abramsona and L. G. Roberts in 1970 and 1972 respectively are a kind of typical random contention system model. In the slotted ALOHA system, common channel is divided into discrete fixed-length time slices called slot, see figure 2. Every slot has three possible states: idle (no packet occupies this slot), success (just one packet occupies this slot), collision (two or more packets apply for this slot). A user terminal can transmit packets only at the beginning of a slot (t_i) and only one packet is permitted in one slot. The users in the system randomly occupy the channel resource, and collision occurs when two or more user terminals apply for a slot to send packets simultaneously. The users involved in collision need to retransmit the packets in the subsequent slots according to a specific collision resolution rule until all the conflict users successfully send their packets. The slotted ALOHA system is outstanding due to its no need of centralized control, easily adding or reducing user terminal, simply operation and small transmission delay, etc. When lightly loaded, the conflict probability of data packets is small，hence every user terminal can effectively utilize the channel resource according to needs. With the system load grows, collision will increase which results in access delay growing, throughput decreases, evenly packets losing or system breakdown. So it is critical to properly choose access model and collision resolution algorithm in order to improve the performance of random contention multi-access system.

1 Truncated Binary Exponential Back-off Algorithm (TBEB)

The binary exponential back-off algorithm is an algorithm model that the retransmission delay and retransmission time of a conflict terminal consist a binary exponential relationship. With the retransmission times grow, the span of the back-off delay increases according to 2-exponential. It has been widely used in LAN and HFC network [6]. The algorithm can be described as follows:

When collision occurs, the conflict terminal randomly chooses a value from the slot window span provided by the algorithm each time, and this random value is the slot number that the terminal must "give up" before retransmit the packets. Assume that the two terminals T_1 and T_2 conflict, the algorithm sets the scope of initial slot window 1~16 , and the collision resolution progress randomly allocates 5 slots and 12 slots for terminal T_1 and terminal T_2 respectively, this means that the terminal T_1 and terminal T_2 can only retransmit information after 5 slots and 12 slots respectively.

Let t_ζ denote the back-off delay of the terminal, the algorithm can be described as follows:

$$\begin{cases} \zeta = random[1,2^n] \\ t_\zeta = \zeta \cdot \tau \end{cases}$$

Where, τ is a system-related time constant. After a conflict resolution is finished, judge whether collision resolution over each terminal succeeds. If the collision resolution fails, the conflict terminal must enter the next resolution. Then the algorithm adds the back-off slot window size according to 2-exponential (but still smaller than the biggest back-off slot window), each conflict ter-

* The authors wish to thank anonymous reviews for useful comments. This work was supported by the National Science Foundation (No. 10371097) of China.

* Ling Yongfa (1973-), male, native of Shangyou of Jiangxi Province, postdoctor oriented in Network Algorithm and Network Control

minal stochastically selects a value again in the new slot window, and repeats the above delay process until the collision resolution succeeds. In order to guarantee the time delay performance of the system, the repetition process cannot be unlimited, and the algorithm specifies the maximum value of the binary index 10 and the max number of repetition 16. When the resolution time exceeds 16, the resolution fails and packets of the conflict terminals lose.

According to the rule above, suppose there are N terminals conflicts in the system. The collision resolution process based on the fundamental truncated binary exponential back-off algorithm can be described as follows:

Firstly, the system enters the first 10 collision resolution periods, suppose the window size of the p th collision resolution is $[m_p, n_p]$:

$$\begin{cases} m_p = m_{p-1} + 2^{p-1} \\ n_p = n_{p-1} + 2^p \end{cases} \qquad p \in [1,10]$$

Let m_p, n_p denote the minimum and maximum of the window of the p th collision resolution respectively. When p =1, m_0 and n_0 represent the initial values equal to 0. For example, in the 1st conflict resolution(p =1) , the algorithm provides 2^1 slot values in [$0+2^0$, $0+2^1$](here are 1 and 2) , then each terminal randomly choose a slot value in the above scope and judges whether the resolution succeeds, namely, compare the N slot values allocated to the N conflict terminals to find whether there are values identical. All the terminals that have different slot values remain their values separately which means that the collision resolutions over these terminals succeeds; the other terminals (N_1, $N_1 \le N$) that have the same slot values still conflict and collision resolution continues, thus the 2nd resolution(p =2) occurs. In the 2nd resolution , the algorithm provides 2^2 slot values in [2^0+2^1, 2^1+2^2] (here are 3 , 4 , 5 and 6) and the above process will repeat until successfully resolved .

If the resolution is yet unsuccessful until p =10, then enter the latter 6 collision resolution periods. Unlike the first 10 collision resolutions, in the latter 6 collision resolution, the slot window size remains 2^{10} (1,024). The slot window size of the t^{th} resolution is $[m_t, n_t]$

$$\begin{cases} m_t = m_{t-1} + 2^{10} \\ n_t = n_{t-1} + 2^{10} \end{cases} \qquad t \in [1,6]$$

Where m_p and n_p represent the minimum and the maximum of the window of the p th collision resolution respectively. When t =1, m_0 and n_0 represent the initial value, respectively equal to 1023 and 2046. In the latter 6 conflict resolution periods, the method of randomly allocating slots to each terminal and judging whether the resolution is successful is same as the method in the first 10 conflict resolutions. The size scopes of slot window in the 16 collision resolutions are given in table 1.

If collision still exists after 16 times resolution, then the entire collision resolution fails and all the information to be sent of the conflict terminals is discarded.

Assume the number of conflict terminals (N) is 6, the collision resolution process shows in Figure 3. In the figure, a, b, c, d, e and f with arrow represent six conflict terminals respectively. The results indicate that the collision of the six conflict terminals is successfully resolved at p=3, thus don't enter the latter 6 collision resolutions periods. The request information of a, b, c, d, e and f will be sent successfully after waiting 8, 14, 12, 3, 6 and 9 slots respectively. 14 slots are used in the entire resolution process.

Table 1 Window Size of Collision Resolution (Unit: time slot)

The prior 10 resolution window size			The latter 6 resolution window size		
Times (p)	Initial value (m_p)	End value (n_p)	Times (t)	Initial value (m_t)	End value (n_t)
1	1	2	1	2047	3070
2	3	6	2	3071	4094
3	7	14	3	4095	5118
4	15	30	4	5119	6142
5	31	62	5	6143	7166
6	63	126	6	7167	8190
7	127	254			
8	255	510			
9	511	1022			
10	1023	2046			

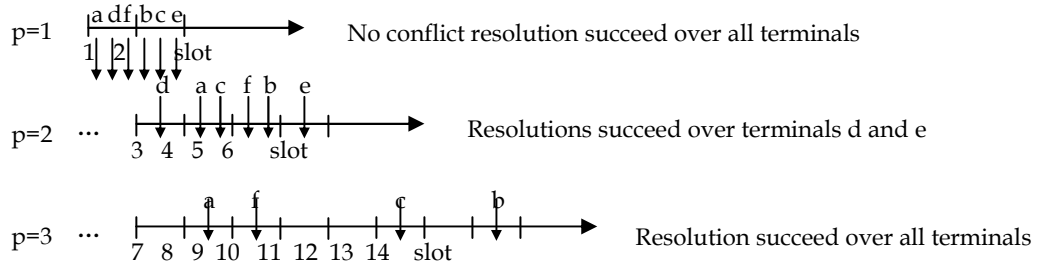


Fig. 3 Collision Resolution Process over 6 Conflict Terminals

2 TBEB Algorithm with Dynamically Setting Initial Window

In TBEB algorithm, when the number of conflict terminals N ≥ 4, the collision resolution starts still from exponential p=1 (provides slot window scope [1, 2]), and as a result, the probability of invalid resolution (namely in each time crack does not have successful state) appears to be very large. Therefore, if we can set the initial window according to the number of conflict terminals dynamically, the collision resolution efficiency will be improved greatly.

The important difference between the improved algorithm with setting initial window dynamically and the basic algorithm lies in: Unlike basic algorithm that sets the initial window size as [1,2] without consideration of the number of conflict terminals, when applying improved algorithm, we set the initial window size dynamically. According to the reference [7], a theory that dynamically sets the initial window size based on the initial conflict terminal number is proposed. Table 2 shows the results.

Table 2 Setting Initial Window Based on Conflict Terminals

Number of Conflict terminal	4~10	11~20	21~50	51~100	101~200
Initial window (W)	6	14	30	62	126

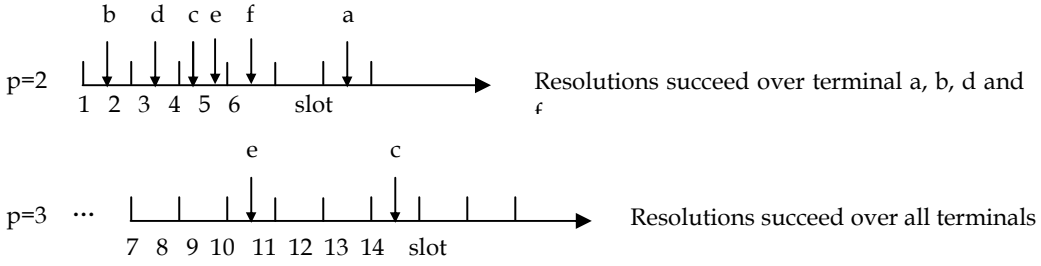


Fig. 4 Collision Resolution Process over 6 Conflict Terminals（Set Initial Window）

Note that the initial window size includes the value of the prior windows' time slots. For example, suppose the number of conflict terminals N=20, p=3 (in basic algorithm p=1), then the initial window size should be [1,14], not [1,8], as the size includes the time slots scope [1, 6] provided by the priori window p=1 and p=2 and then adds on time slots [7,14] provided by p=3, see Table 1. Other than setting the initial window size according to the system load, the collision resolution processes in the improved algorithm remains consistent with the basic algorithm.

Similarly, suppose the number of conflict terminals N=6, the collision resolution processes of the initial window (this time p=2) are showed in figure4.

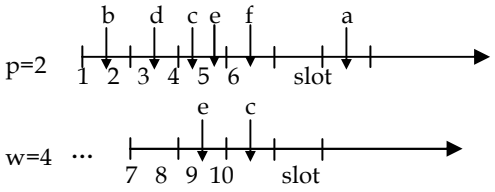
The results indicate that the 6 conflict terminals successfully resolve the collision when p=3; compared with the basic algorithm, the improved algorithm only carries on two collision resolution, and the collision resolution has not entered the latter 6 conflict resolution cycles. The request information of conflicts terminals a, b, c, d, e and f will be successfully transmitted after waiting 6, 1, 12, 2, 9 and 4 slots respectively. Altogether, the collision resolution process has used 14 time slots.

3 TBEB Algorithm with Dynamically Setting Initial Window and End Window

The improved algorithm with setting the initial window and the end window must dynamically set the size of the window on which the last collision resolution is carried. According to the regulation of the truncated binary exponential back-off algorithm collision resolution algorithm, the back-off of the window size changes by 2-exponential, but it differs from the first two algorithms, in the collision resolution process, when only a certain terminals conflict in a time slot in a specific collision resolution, the algorithm enters the last the collision resolution process. The window size of the last the collision resolution will be dynamically set according to the number of conflict terminals (not back off by 2-expenential, but the last conflict number). Provided that there still exits conflict after the last collision resolution, then the entire collision resolution fails.

For instance, suppose the conflict terminal number N=20, when p=4 (assume there are only 3 terminals conflict in the 18th time slot), then enters the last the collision resolution process, provides a time slot window such as [31,38] for the 3 conflict terminals (Notation: when p=4, the prior window provides the time slot size [1,30]) and enter the last the collision but not provides time slot window [31, 62] according to 2-expenential back-off manner.

Suppose the conflict terminals number N=6,and the collision resolution process of the initial window (p=2) and the end window (p=4, namely [7,10]) show in figure 5. The result indicated that the 6 conflict terminals enter the last collision resolution when p=2, and the algorithm also only carried on two collision resolution, more, only 10 time slots (14 time slots were used in prior improved algorithm) are used, the collision resolution has not entered the last 6 conflict resolution periods. The request information of conflict terminals a, b, c, d, e and f will be successfully transmitted after waiting 6, 1, 9, 2, 8 and 4 slots respectively.



Resolutions succeed over a, b, d and f and enter the last resolution.

Resolutions succeed over all terminals; w represents window size

Fig. 5 Collision Resolution Process over 6 Conflict Terminals（Set Initial and End Windows）

Table 3 Simulation Results of Basic and Improved Algorithms (Unit：time slot)

N	Basic algorithm		Initial window algorithms		Initial and end window algorithm		N	Basic algorithm		Initial window algorithm		Initial and end window algorithm	
	Count	Maxslot	Count	Maxslot	Count	Maxslot		Count	Maxslot	Count	Maxslot	Count	Maxslot
2	1.64	5.13	1.64	5.13	1.64	5.13	12	5.06	69.24	2.78	57.57	2.63	55.62
3	2.39	9.33	2.39	9.33	2.39	9.33	13	5.20	76.69	2.92	63.36	2.75	59.65
4	2.94	14.99	1.88	14.42	1.88	14.42	14	5.33	84.35	3.05	69.03	2.91	60.57
5	3.39	21.01	2.22	18.62	2.20	18.02	15	5.46	92.57	3.17	75.53	2.99	68.15
6	3.73	27.16	2.52	23.31	2.32	20.21	16	5.58	100.85	3.29	82.21	3.14	76.41
7	4.04	33.59	2.79	28.49	2.55	24.27	17	5.70	109.36	3.40	89.06	3.25	84.02
8	4.29	40.42	3.06	34.43	2.70	28.63	18	5.81	117.62	3.51	96.31	3.26	91.53
9	4.50	47.13	3.29	40.58	2.92	36.51	19	5.91	125.50	3.61	103.49	3.32	99.29
10	4.70	54.53	3.50	47.16	3.19	46.12	20	6.00	133.30	3.72	111.16	3.47	104.65
11	4.90	62.04	2.64	51.94	2.52	48.24	21	6.09	141.76	3.85	120.78	3.57	110.89

5 Results Analysis

10000 simulations are carried out on the computer. Let N denote the number of conflict terminals and Maxslot represent the average maximum slots value needed in the 10000 simulations of the N conflict terminals. The formula is given by:

$$\text{Maxslots} = \left(\sum_{i=1}^{10000}\sum_{j=1}^{p} 2^j\right)/10000$$

Where p is the max back-off exponential in a certain successful collision resolution, and the simulation results of the three algorithms are shown in Table 3, and count is the resolution times of a successful conflict resolution.

Under different arrival rate (G)， the throughput (S) can be described as[8]

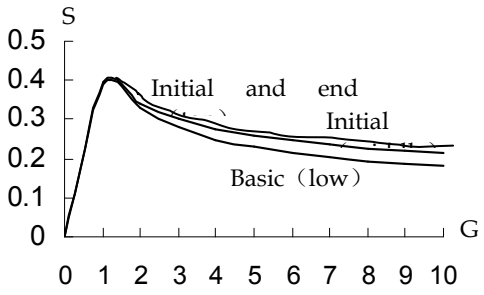


Fig. 6 Throughput Curves of Binary Exponential Back-off Algorithm

$$S = \frac{Ge^G}{1 + G + \sum_{k=1}^{N} \frac{G^k}{k!} E(L_k)}$$

Where N is the number of conflict terminals , and E (L_k) is the average number of slots. Figure 5 shows the throughput performance curves of the three algorithms.

(1) The simulation results indicate that under the permitted max repeat times, the max number of conflict terminals that the basic algorithm can resolve is about 250, the setting initial window algorithm is about 370 and the setting initial and end window algorithm is about 440. Thus greatly improves the conflict resolution efficiency.

(2) A conclusion can be made from table 3 that if the conflict terminal number is same, the two improved algorithms reduce the collision resolution times and save the slot space compared to the basic algorithm.

(3) A conclusion can be made from table 3 that the average needed slots increases linearly with the growing of the number of conflict terminals; From figure 5, we conclude that the system throughput is in the trend to be stable with the load (G) grows (the stable throughput value of the basic algorithm, initial window algorithm and initial and end window algorithm tends to be approximately 0.18, 0.21 and 0.23 respectively). This indicates that the binary exponential back-off conflict resolution algorithm is stable and valid.

(4) From the throughput formula, we can see that the max throughput of the initial and end window algorithm is Smax=0.3944 (G=1.06), the max throughput of the initial window algorithm is Smax=0.3941 (G=1.0), and the max throughput of the basic algorithm is Smax=0.3903 (G=0.9). A conclusion can be made from figure 5 that the performance of the two improved algorithms is better than the basic algorithm. The im-

proved algorithms have higher system throughput and can make the system work under higher throughput.
(5) The improved binary exponential back-off collision resolution algorithms proposed in the paper are helpful to enhance the performance of all kinds of binary exponential back-off algorithms.

References

[1] Shen Lansun, Tian Dong. The Development of Wireless Video Transmission Technology [J]. Electronic Technology Application, 2001, (1): 6-9.
[2] ROM R. SIDI M. Multiple access protocols [J]. New York Berlin Heidelberg, 1989, (6): 5-30.
[3] Vapnik V. Tree-Based multi-access protocols where collision multiplicities are known [J]. IEEE Trans Commun, 1985, (33): 999.
[4] N. Abramson. The ALOHA system-another alternative for computer communications [J]. Proc 1970 Fall Joint Computer conf, 1970, (37): 281-285.
[5] L. G. Roberts, ALOHA packet system with and with and without slots and capture [J], ACM SIGCOMM Compute. Common. Rev., 1975, (5): 28-42.
[6] Dolor, Sala. Contuntion resolution algorithm the Chapter 6 of the Ph.D[J]. Georgia Institute of Technology, 1998, (3): 126-133.
[7] Xu Tian, Ye Jiajun. Collision Resolution Algorithm Applicable to HFC Network [J]. Television Technology, 2000, 220 (10): 18-20.
[8] J. L. Massey. Collision-Resolution Algorithms in and Random-Access Communications [J]. *Multi-User Communications CISM Course and Lecture Series*, 1981, (265): 73-137.

Yongfa Ling, received the B.S. and M.S. degree from Yunnan Minzu University. Now he is the Postdoctoral students in Xi'an Jiaotong University. His scientific interests are in the fields of network algorithm and network control.

Deyu Meng, received the B.E. degree in Information Science in 2001 and M.E. degree in applied mathematics in 2004 from the Xi'an Jiaotong University. He is currently a doctoral student at School of Electronic and Information Engineering in Xi'an Jiaotong University. His scientific interests are in the fields of machine learning and data mining.