Slot Assign Algorithm with Threshold Based on Irregular Repetition Slotted ALOHA (IRSA)

Huaicui Zheng[1], Changle Li[1](✉), Ni Tian[1], and Jun Cheng[2]

[1] State Key Laboratory of Integrated Services Networks, Xidian University, Xi'an 710071, China
clli@mail.xidian.edu.cn

[2] Department of Intelligent Information Engineering and Science, Doshisha University, Kyoto 6008586, Japan
jcheng@mail.doshisha.ac.jp

Abstract. As successive interference cancellation (SIC) technology has become a research hotspot recently, random multiple access protocols based on SIC such as contention resolution diversity slotted ALOHA (CRDSA) and irregular repetition slotted ALOHA (IRSA) are put forward one after the other. Although throughput performance for these protocols has been improved significantly. Yet the throughput will drop dramatically when the load increases gradually. In this paper, in order to get a higher throughput in the case of high load, we propose a slot assign algorithm with threshold based on IRSA to obtain a throughput T which is more than 1.

Keywords: Random multiple access protocol · SIC · CRDSA · IRSA · Slot assign algorithm with threshold

1 Introduction

Traditional random multiple access protocol ALOHA [1] is proposed in 1971, followed by slotted ALOHA [2]. However, there exists many drawbacks such as high probability collisions and low throughput, which may result in great communication delay and a waste of resource. In [3], diversity slotted ALOHA (DSA) is proposed. By allowing a user to transmit multiple copies of the same packet, DSA has a slight enhancement in throughput and decrease in delay when the load is moderate. As a multi-user detection technology, successive interference cancellation (SIC) is adopted as early as in the third generation of wireless mobile telecommunications technology (3G) and used widely in code division multiple access (CDMA). It could play disruptive effect in guaranteeing high-reliability and low-latency and satisfy the requirement of Ultra-Reliable Low Latency Communication (URLLC) in 5G [4–6].

[7] applies SIC to multiple access protocol based on ALOHA and then proposes the contention resolution diversity slotted ALOHA (CRDSA) protocol. In

B. Li et al. (Eds.): IoTaaS 2018, LNICST 271, pp. 225–231, 2019.
https://doi.org/10.1007/978-3-030-14657-3_22

CRDSA, a user is allowed to transmit two replicas on a MAC frame. And each header of the replica contains the position information of the another replica. As soon as one of the replicas is recovered correctly, then interference caused by the other replica can be eliminated. And this procedure can be continued to recover the remain packets on a frame. As a result, the throughput T (defined as the normalized throughput) in CRDSA can reach about 0.55. In [8], the author changes the number of a user to transmit replicas of a packet and then the irregular repetition slotted ALOHA (IRSA) is put forward. In IRSA, a variable repetition rate is employed and each user can choose the repeat times according to the probability distribution function (PDF). Compared to the CRDSA, there is a significant increase in the throughput performance for IRSA and throughout T can reach about 0.88. In order to achieve a higher throughput when the load increases, we propose a slot assign algorithm with threshold based on IRSA. An upper bound for the number of packets which can be decoded successfully in each slot will be set. In other words, as long as the number of packets in one slot is less than the given threshold, the packet can be recovered correctly.

The remainder of this paper is organized as follows. Section 2 will give a description of bipartite graph representation of the process of SIC. System model follows in Sect. 3. The simulation is conducted in Sect. 4 and then analysis is provided. At last, conclusions are shown in Sect. 5.

2 Bipartite Graph Representation of the Process of SIC

Considering the condition that a frame is composed of n slots, within which m users will transmit packets. There is no coordination among users when transmitting packets. Every user has one packet to transmit and l replicas of a generic packet can be transmitted within one frame. The slots occupation information is in the header of packets. Assuming the threshold of packets decoded successfully in each slot is s_{up}, which can be realized by power control [9]. When decoding in some slot, if the number of unrecovered packets in the slot is no more than the threshold s_{up}, all packets in the slot can be recovered successfully. At the same time, due to slots occupation information is included in the header of the recovered packets, these recovered packets in the slot can be removed from the other slots which they have been transmitted into.

It's convenient to use bipartite graph $G = \{P, S, E\}$ to represent the process of SIC, in which $P = \{p_1, p_2, p_3, \ldots, p_m\}$ and $S = \{s_1, s_2, \ldots, s_n\}$ denotes the set of slots and packets respectively. The set of E is $E = \{e_{ij} | i = 1, 2, \ldots, m; j = 1, 2, \ldots, n\}$, representing edges between slots and packets. If and only if a replica of *i-th* which is denoted by p_i is transmitted to *j-th* slot which is denoted by s_j, the p_i and s_j are connected with the edge e_{ij} in bipartite graph as shown in Fig. 1. It is obvious that l replicas of a packets are connected with l edges. The number of connection edges for slots and packets is referred to as their degree distribution in the bipartite graph. The definition of the slot and packet degree distribution is given by Eq. (1), in which $\Lambda(x)$ represents the slots degree distribution and $\Psi(x)$ represents packets degree distribution respectively. According to the bipartite graph in Fig. 1, the degree distribution of slots means the number of received packets and for packets the degree distribution means the number of replicas, i.e. the number of transmission for a packets [10]. Thus, the average transmission packets $average_{trans}$ for users can be derived by Eq. (2) and the average received packets in each slot $average_{rece}$ can be obtained by Eq. (3) within one frame. It is easy to prove that average number of transmission packets for one slot, which is also referred as to normalize offered load G, can be derived using the degree distribution of slots and packets by the Eq. (4).

$$\begin{cases} \Lambda(x) = \sum_l \Lambda_l x^l \\ \Psi(x) = \sum_l \Psi_l x^l \end{cases} \tag{1}$$

$$average_{trans} = \sum_l l\Lambda_l = \Lambda'(1) \tag{2}$$

$$average_{rece} = \sum_l l\Psi_l = \Psi'(1) \tag{3}$$

$$G = \frac{m}{n} = \frac{\Lambda'(1)}{\Psi'(1)} \tag{4}$$

If the packet is recovered in some slot, the connected edge between the packet and corresponding slot becomes dotted line and edges between the packet and the other slots also become dotted lines at the same time, which denotes the recovered packets are removed from the other slots. An example is given in Fig. 2 to better express the detailed process for SIC with an upper bound s_{up}. The example gives analysis for the condition that 6 packets will be transmitted within one frame divided into 4 same slots. Otherwise, the threshold s_{up} is set to 3. If one packet is recovered, we replace the edges with dotted lines, which are connected with the packet. The initial bipartite graph for the condition is shown in Fig. 1 and the SIC process starts by the Fig. 2(a), decoding in the first slot. In the first slot, packets p_2 and p_3 can be recovered on the account of the number of unrecovered packets being less than s_{up}. Once packets p_2 and p_3 are recovered in slot s_1, they will be removed from the other slots they occupy, i.e., they will be removed from the slots s_3 and s_4. That is to say the edges connected with the packets p_2 and p_3 become dotted lines in the bipartite graph as shown in Fig. 2(a). For the second slot s_2, though there is a collision, p_1 and p_4 still can be recovered due to the number of unrecovered packet is no more than 3. So the edges connected with p_1 and p_4 are replaced with dotted lines shown in Fig. 2(b). For the third slot, there is no collision because the packet p_2 is recovered in the first slot and has been removed from the slot s_3. Thus the packet p_5 can be recovered. The edges connected with packet p_5 become dotted lines in Fig. 2(c). For the last slot s_4, though the number of received packets is more than 3, the packets p_3, p_4 and p_5 is recovered in slot s_1, s_2 and s_3 respectively. Hence, the degree of the slot s_4 reduces to 1 and p_6 can be revealed. We can see all the packets can be recovered in the end shown in Fig. 2(d).

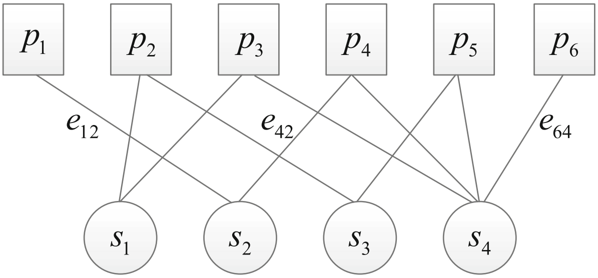

Fig. 1. The initial bipartite graph with 6 packets and 4 same packets

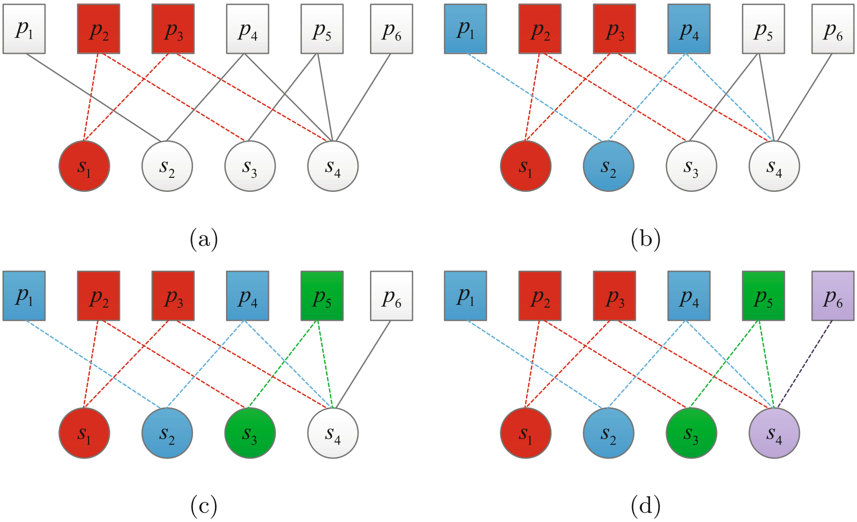

Fig. 2. The successive interference cancellation process

3 System Model

In the process of multi-user random access, each frame T_F for communication system is divided into n slots shown in Fig. 3, which is denoted by $s_j (j = 1, 2, \ldots, n)$. The length of each slot is defined as $T_n = T_F/n$ and m users transmit packets into the frame. Also there is no coordination among these users when they transmit their packets. That's to say when users transmit their packets, they are independent with each other. Each user starts to transmit their own packets at the beginning of each slot and the slot length is equal to the time of a packets transmission. Then we define system load $G = m/n$ (defined as the averaged number of packets in each slot) and throughout T (defined as probability of successful transmission packets).

In our proposed scheme, each user repeats their packets l times like IRSA shown in Fig. 3. And each header of replica carry the position information of

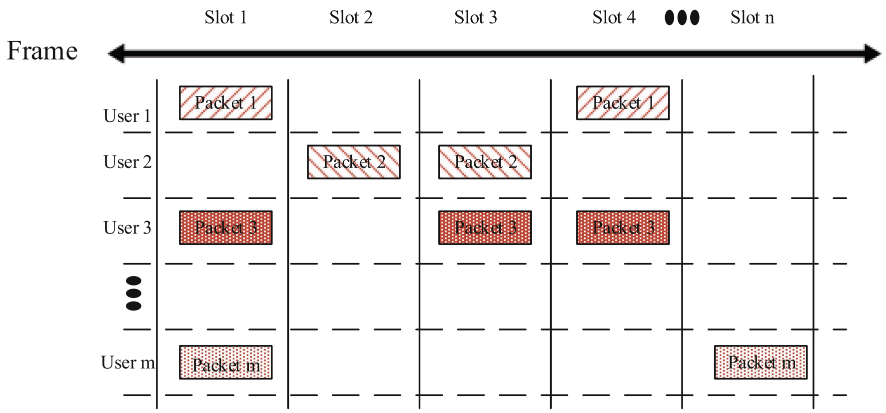

Fig. 3. The random access for IRSA and our proposed scheme

other replicas. If a replica is recovered, then corresponding replicas in other slots can be recovered. Then the received packets and upper bound of packets which can be recovered in each slot are denoted as $s_{j_{num}}(j = 1, 2, .., n)$ and s_{up} respectively. When decoding, the number of the packets in each slot will be counted. If $s_{j_{num}} \leq s_{up}$, the packets in this slot can be recovered correctly according to the SIC. Otherwise, if the number is greater than the threshold and then these packets in current slot cannot be recovered directly. The packets which have been recovered in other slots should be found first and then those packets will be subtracted correspondingly. The number of packets in one slot may be counted again and if the number is still larger than the threshold, those packets in the slot will be thrown way. Then if the number is below the threshold, the process of SIC will be continued.

4 Simulation and Discussion

In this paper, simulations are performed in MATLAB. So as to observe the performance of different protocols, we choose $\Lambda(x) = 0.5x^2 + 0.28x^3 + 0.22x^8$ as the degree distribution in the simulation of IRSA and slot assign algorithm with threshold proposed in this paper. Then we assume that each frame shares the same length and the frame is divided into 2000 slots. More detailed simulation parameters are present in Table 1.

4.1 Results and Analysis of Throughout

Figure 4 are the simulation results for CRDSA and IRSA. From Fig. 4, we can see that when the load is about 0.6, the throughput can obtain a maximum of 0.55. Correspondingly, the throughput for IRSA can reach about 0.88 with the load is equal to 0.90. Compared to the throughput for CRDSA, the throughput for IRSA drops more dramatically than that for CRDSA after the throughput

Table 1. Simulation parameters

Simulation parameters	Configuration
The length of access frame	2000
The max iterations of sic	100
Rate of max repetition	8
Probability of ready packet	1%
Time of simulation time	100 ms
Bound of slot	3

reaches a peak. From relevant literature [8,11], when we adopt the distribution $\Lambda(x) = 0.5x^2 + 0.28x^3 + 0.22x^8$, if the load G is below the 0.938 (defined as the extreme value of G), the throughput T can almost reach the peak. However, this theoretical result can be obtained when the length of frame is infinite. Meanwhile, when the load G is larger than the extreme value, the reason why the throughput drops sharply is that packets in the slot can be recovered correctly and the SIC technology cannot work any more.

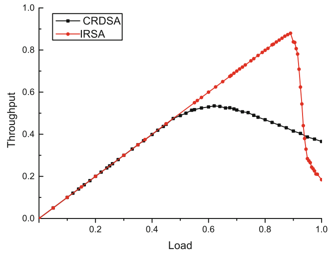

Fig. 4. Throughput for CRDSA and IRSA

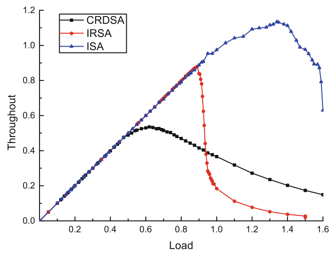

Fig. 5. Throughput for CRDSA, IRSA and ISA

Also, in the simulation above, the default of the upper bound (defined as packets which can be recovered correctly in each slot) is 1. Different from above, in Fig. 5, the upper threshold for the number of packets in one slot is set to 3. The process of recovering packets is slightly from that above and more detailed description has been shown in Sect. 3. From the result, we can also see the throughput for our proposed scheme known as ISA increases to 1 and even greater than 1. The peak of throughput occurs when the load is equal to 1.34. In this case, the throughput can reach about 1.13. Simulation results show that the improvement of throughput is obvious when the load is more than 1. In other words, by adding a decoding threshold to existing IRSA, performance of SIC can be improved significantly. What's more, our proposed can get a comparatively higher throughout compared to the CRDSA and IRSA.

5 Conclusion

In this paper, by analyzing throughput for existing random access protocol CRDSA and IRSA based on SIC, then we propose slot assign algorithm with threshold based on IRSA. Compared to known IRSA and CRDSA, we get a better throughput performance especially in the case of higher load and the value of throughput can be over 1. However, it also brings larger packet loss rate, which should be solved next. Also, we will carry out further simulation with actual communication scenario in network simulator OMNET++ to get a better performance evaluation such as delay and packet loss rate.

Funding Information. This work was supported in part by the National Natural Science Foundation of China under Grants U1801266, and 61571350, and in part by the Key Research and Development Program of Shaanxi under Contract 2017KW-004, Contract 2017ZDXM-GY-022, Contract 2018ZDXMGY-038, and Contract 2018ZDCXL-GY-04-02.

References

1. Abramson, N.: THE ALOHA SYSTEM - another alternative for computer communications. In: Proceedings of Fall Joint Computer Conference, pp. 281–285. ACM, New York (1970)
2. Davis, D.-H., Gronemeyer, S.-A.: Performance of slotted ALOHA random access with delay capture and randomized time of arrival. IEEE Trans. Commun. **28**(5), 703–710 (1980)
3. Gagan, L.-C., Stephen, S.-R.: Diversity ALOHA - a random access scheme for satellite communications. IEEE Trans. Commun. **31**(3), 450–457 (1983)
4. Liang, Y., Li, X., Zhang, J., Liu, Y.: A novel random access scheme based on successive interference cancellation for 5G networks. In: Wireless Communications and Networking Conference, pp. 1–6. IEEE, San Francisco (2017)
5. Stefanovic, C., Lazaro, F., Popovski, P.: Frameless ALOHA with reliability-latency guarantees. In: 2017 IEEE Global Communications Conference, pp. 1–6. IEEE, Singapore (2017)
6. Lzaro, F., Stefanović, Ć.: Finite-length analysis of frameless ALOHA with multi-user detection. IEEE Commun. Lett. **214**, 769–772 (2017)
7. Casini, E., De Gaudenzi, R., Del Rio Herrero, O.: Contention resolution diversity slotted ALOHA (CRDSA): an enhanced random access schemefor satellite access packet networks. IEEE Trans. Wirel. Commun. **6**(4), 1408–1419 (2007)
8. Liva, G.: Graph-based analysis and optimization of contention resolution diversity slotted ALOHA. IEEE Trans. Commun. **59**(2), 477–487 (2011)
9. Vem, A., Narayanan, K.-R., Cheng, J., Chamberland J.-F.: A user-independent serial interference cancellation based coding scheme for the unsourced random access Gaussian channel. In: 2017 IEEE Information Theory Workshop, pp. 6–10. IEEE, Kaohsiung (2017)
10. Narayanan, K.-R., Pfister, H.-D.: Iterative collision resolution for slotted ALOHA: an optimal uncoordinated transmission policy. In: 2012 7th International Symposium on Turbo Codes and Iterative Information Processing, pp. 136–139. IEEE, Gothenburg (2012)
11. Storn, R., Price, K.: Differential evolution - a simple and efficient heuristic for global optimization over continuous spaces. J. Glob. Optim. **11**(4), 341–359 (1997)