

ALOHA Improvement Algorithm for Dynamic Frame Time Slots with Transformer

Weijie Zhan

School of Physics and Electronic Science, Hunan University of Science and Technology, Xiangtan, 411201, China

Keywords: RFID; Dynamic frame slot; ALOHA algorithm; Transformer

Abstract: In recent years, with the widespread application of RFID technology in production and daily life, the demand for tag reading systems has been increasing. When faced with a large number of tags, RFID systems often experience severe collisions within the same reading frame due to tag responses, leading to low reading efficiency. The key to solving this problem lies in the speed and accuracy of the tag number estimation algorithm. Based on the analysis of traditional algorithms, this paper proposes a new tag number estimation algorithm. This algorithm generates tag datasets with specific word lengths based on the principle of the dynamic framed slotted ALOHA (DFSA) algorithm and establishes a model using a Transformer neural network to predict the number of tags. The network establishes a mapping relationship between the reader and the remaining number of tags to estimate the tag count. Compared with traditional algorithms, the innovation of this paper lies in the introduction of the self-attention mechanism, which significantly improves the accuracy of tag number prediction while reducing the time consumption of the reading system. Simulation results show that the proposed algorithm improves system efficiency while maintaining accuracy, offering a new solution for large-scale RFID applications.

1. Introduction

Radio Frequency Identification (RFID) is a non-contact automatic identification technology that automatically identifies targets and retrieves related data through radio frequency signals[1]. Compared to traditional identification technologies such as barcodes and magnetic strips, RFID offers several significant advantages, including short response time, large data storage capacity, high accuracy, and good real-time performance. Due to these advantages, RFID technology has been widely applied in various fields such as healthcare services[2-3], warehouse management[4-5], transportation[6], and security control[7-8]. An RFID system typically consists of three main components: electronic tags, readers, and application systems. When an RFID system operates, the reader communicates with the electronic tags via radio waves to read or write data in the tags. However, when a single reader attempts to read multiple tags simultaneously, the signals received by the reader may interfere with each other, leading to the tag reading collision problem. This collision can prevent the reader from correctly identifying and reading the information in the tags, thereby affecting the overall performance and reliability of the system. Therefore, achieving effective and

reliable identification of tags and resolving the tag collision problem have become key points in RFID technology research[9].

Currently, the main algorithms for solving the tag collision problem in RFID systems can be categorized into ALOHA anti-collision algorithms and binary tree anti-collision algorithms. Due to the higher complexity of binary tree anti-collision algorithms, they are seldom used in practical systems. In contrast, the ALOHA algorithm, known for its simplicity and ease of implementation, has become one of the most widely used anti-collision algorithms. Among the ALOHA algorithms, the most commonly applied is the Dynamic Framed Slotted ALOHA (DFSA) algorithm. The key to the DFSA algorithm lies in estimating the number of remaining tags based on the feedback from the previous frame, thereby determining the frame length for the next frame. Literature proposes an anti-collision algorithm based on equal-area division, which can effectively improve system throughput in large-scale tag application scenarios [10]. Literature introduces a scheme that uses dynamic predictive weights to estimate the number of tags, ensuring that the number of tags is approximately equal to the data frame length, thereby maintaining an average collision ratio of only 1.1% [11]. Additionally, Literature combines the DFSA algorithm with LSTM and BP neural networks to predict the number of tags in the next frame, thereby effectively improving the recognition accuracy of RFID systems and reducing slot waste [12-13].

In RFID systems, predicting the number of tags is key to addressing the tag collision problem. Traditionally, LSTM and BP neural networks have been widely used. LSTM (Long Short-Term Memory) neural networks can effectively handle time-series data, making them suitable for scenarios that require consideration of historical information; however, they are computationally complex, have long training times, and are less efficient in large-scale data processing [14-15]. BP (Back Propagation) neural networks are simple in structure and easy to implement, but they perform poorly in handling long-term dependencies and complex data, tend to fall into local optima, and are not ideal for large-scale data processing [16-17]. In contrast, the Transformer model, which employs the Self-Attention Mechanism, can capture global information without relying on sequence order, making it more efficient and accurate in handling long-term dependencies and large-scale data [18]. Moreover, the Transformer model has strong parallel computing capabilities, faster training speed, and meets the needs of large-scale data processing [19-20].

The main contribution of this paper lies in the proposal of a novel tag number estimation method that combines the DFSA algorithm with the Transformer model. Compared to traditional LSTM and BP neural networks, the Transformer model offers stronger parallel computing capabilities and better handling of long-term dependencies. By introducing the self-attention mechanism, the proposed algorithm not only enhances the accuracy of tag number prediction but also significantly reduces slot consumption in the system, thereby improving the overall efficiency of the RFID system.

2. DFSA Algorithm

2.1 Algorithm Principle

The pure ALOHA algorithm is one of the most basic multiple access control methods. Within the coverage area of the reader, each tag randomly selects a moment within a fixed time period to transmit data frames. If multiple tags transmit data frames at the same time, a collision occurs, causing the data from these tags to be unreadable. As the number of tags in the system increases, collisions become more frequent, ultimately leading to a significant decrease in the system's reading efficiency. To address this issue, the Dynamic Framed Slotted ALOHA (DFSA) algorithm was developed[21]. This algorithm divides the communication channel into several time slots of equal length, as shown in Figure 1.

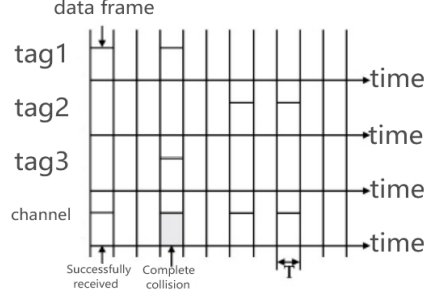


Figure 1: Schematic diagram of dynamic frame time slot algorithm

In this algorithm, tags can only transmit data within the predetermined time slots of the system. As observed in Figure 1, three scenarios can occur within each time slot: a single tag successfully transmits data, no tags transmit data, or multiple tags transmit data simultaneously, resulting in a collision. By counting the number of successful transmissions, idle slots, and collision slots during the previous round of reading, the number of remaining unread tags can be estimated. Based on this estimate, the number of time slots can be dynamically adjusted to prevent severe collisions caused by an excessive number of tags, thereby improving the system's throughput.

2.2 Mathematical Analysis

In the improved Dynamic Framed Slotted ALOHA algorithm, the key to system performance lies in accurately estimating the recognition status within the current frame to dynamically adjust the parameters for the next frame. Assume the frame length is M , and there are k tags within the reader's coverage area awaiting recognition. The number of tags x responding within a single time

slot follows a binomial distribution, i.e., $x \sim B(k, \frac{1}{M})$. The probability P_x that exactly x tags respond simultaneously is given by:

$$P_x = \binom{k}{x} \left(\frac{1}{M}\right)^x \left(1 - \frac{1}{M}\right)^{k-x} \quad (1)$$

Within the current frame, there are three possible outcomes for each time slot: no tags respond (P_{idle}), exactly one tag responds successfully (P_{succ}), or multiple tags respond, causing a collision (P_{coll}). These probabilities can be expressed as:

$$P_{\text{idle}} = \left(1 - \frac{1}{M}\right)^k \quad (2)$$

$$P_{\text{succ}} = k \times \frac{1}{M} \times \left(1 - \frac{1}{M}\right)^{k-1} \quad (3)$$

$$P_{\text{coll}} = 1 - P_{\text{idle}} - P_{\text{succ}} \quad (4)$$

Using the above equations, we can calculate the expected number of idle slots (y_{idle}), successful recognitions (y_{succ}), and collision slots (y_{coll}) for different numbers of tags k and frame lengths M :

$$y_{\text{idle}} = k \times P_{\text{idle}} \quad (5)$$

$$y_{\text{succ}} = k \times P_{\text{succ}} \quad (6)$$

$$y_{\text{coll}} = k \times P_{\text{coll}} \quad (7)$$

Clearly, the estimated number of remaining unrecognized tags k_{estimate} is:

$$k_{\text{estimate}} = k - y_{\text{succ}} \quad (8)$$

2.3 Vogt Estimation Algorithm

In tag number estimation, the Vogt estimation algorithm [22] is a method based on comparing actual observed data with theoretical expectations. It estimates the number of remaining unrecognized tags by analyzing the number of idle slots y_{idle} , successful slots y_{succ} , and collision slots y_{coll} recorded by the reader during a cycle, and calculating the difference between these values and their corresponding theoretical values s_{idle} , s_{succ} , and s_{coll} . Specifically, the Vogt estimation algorithm first constructs an error vector ϵ :

$$\epsilon = \begin{pmatrix} y_{\text{idle}} - s_{\text{idle}} \\ y_{\text{succ}} - s_{\text{succ}} \\ y_{\text{coll}} - s_{\text{coll}} \end{pmatrix} \quad (9)$$

Then, the algorithm iteratively adjusts the estimated tag number k within a range defined by $[s_{\text{succ}} + \delta s_{\text{coll}}, \alpha \cdot (s_{\text{succ}} + s_{\text{coll}})]$, where α and δ are preset constants, to find the value of k that minimizes each component of the error vector ϵ . This adjustment is used to fine-tune the calculation's precision and efficiency. The final determined tag number k_{estimate} is taken as the algorithm's estimate of the number of unrecognized tags. While this algorithm offers high estimation accuracy, its computational complexity is relatively high, which may impact the system's real-time performance, especially in scenarios with a very large number of tags.

3. DFSA Algorithm

The Transformer model is a deep learning architecture based on the self-attention mechanism, composed of an encoder and a decoder, with each part consisting of multiple layers. The core idea is to capture the relationships between different positions in the input sequence using the self-attention mechanism and to enhance the model's ability to focus on different parts of the information using the multi-head attention mechanism.

The calculation for dot-product attention is as follows:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \quad (10)$$

where Q is the query matrix, K is the key matrix, V is the value matrix, and d_k is the dimension of the keys. Dot-product attention works by dividing the dot product of the query and key by the square root of the key dimension, then applying the softmax function to obtain weights, which are subsequently multiplied by the value matrix to produce the final output.

The multi-head attention mechanism splits the input into multiple heads, each head independently calculates attention, and then the results are concatenated together and passed through a linear transformation:

$$\text{MultiHead}(Q, K, V) = \text{Concat}(\text{head}_1, \dots, \text{head}_h)W^O \quad (11)$$

The computation for each head is given by:

$$\text{head}_i = \text{Attention}(QW_i^Q, KW_i^K, VW_i^V) \quad (12)$$

This allows the model to focus on different subspaces and capture more features and information.

These mechanisms enable the Transformer to efficiently handle long sequence tasks, offering significant advantages in both computational efficiency and performance.

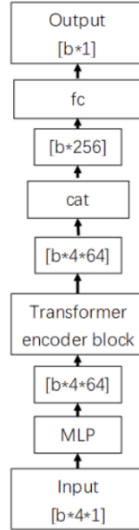


Figure 2: Transformer - model architecture

As shown in Figure 2, this model architecture combines a Multi-Layer Perceptron (MLP) and Transformer encoder blocks to process input data with 4 features. First, the input layer receives the 4-dimensional features of each sample, and the MLP processes these features, mapping the input from 4 dimensions to 16 dimensions. Then, these features are further processed by the Transformer encoder block, capturing the contextual relationships between features, with the output dimensions being [batch, 4, 64]. Next, these features are combined through a feature concatenation layer (cat), forming a new input format. In the fully connected layer (fc), the features are linearly transformed to [batch, 256], and finally, a prediction is generated through the output layer.

The specific steps of the proposed algorithm are as follows:

(1) Using equations (5) to (8), calculate the expected number of idle slots y_{idle} , successful slots y_{succ} , collision slots y_{coll} , and the estimated number of remaining tags k_{estimate} for different numbers of tags k to be read and different frame lengths M (assuming $k \in [0, \dots, 5M]$). These values form the dataset $T_k = \{(k, y_{\text{idle}}, y_{\text{succ}}, y_{\text{coll}}) | k_{\text{estimate}}\}$;

(2) Input the datasets T_k corresponding to each frame length M into the network for training. Once the network metrics converge, the corresponding network model G_M for each frame length M is obtained;

(3) Input the actual number of detected tags, idle slots, successful slots, and collision slots $(k, y_{\text{idle}}, y_{\text{succ}}, y_{\text{coll}})$ from the current reading cycle of the reader into the corresponding network model G_M to obtain the estimated number of remaining tags k_{estimate} . The reader then adjusts the frame length for the next reading cycle according to Table 1;

Table 1: Relationship between frame size and label number of dynamic frame time slot algorithm

Frame length	4	8	16	32	64	128	256
Number of tags	1~4	5~10	11~22	23~44	45~88	89~177	Above 177

4. Transformer-Optimized DFSA Algorithm

Using the formula, calculate the number of tags read under different frame lengths M and generate the corresponding dataset T_k . Use the generated datasets to train the network models G_M for different frame lengths. During the actual reading process, use the trained model G_M to predict the number of remaining tags k_{estimate} and adjust the frame length to optimize reading efficiency. This algorithm significantly improves recognition speed and substantially reduces the number of training epochs.

4.1 Model Training

During the training process, the learning rate is set to 0.01, with a maximum of 800 iterations. The neuron activation function used is the Rectified Linear Unit (ReLU). The specific settings are shown in Table 2.

Table 2: Neural Network Training Parameters Table

Parameter Name	Value
Maximum Epochs	800
Initial Learning Rate	0.01
Batch Size	128
Optimizer	adam
Loss Function	mse

Finally, as shown in Figure 3, the training results indicate that the model's training loss and test loss rapidly decrease and approach zero in the early stages, demonstrating that the model converges quickly and has good generalization ability. The loss curves are smooth and show no significant differences, indicating that the model performs consistently on both the training and test sets, with no

signs of overfitting or underfitting. Overall, the chosen neural network architecture and optimization parameters are appropriate, enabling the model to efficiently reach optimal performance within a limited number of epochs.

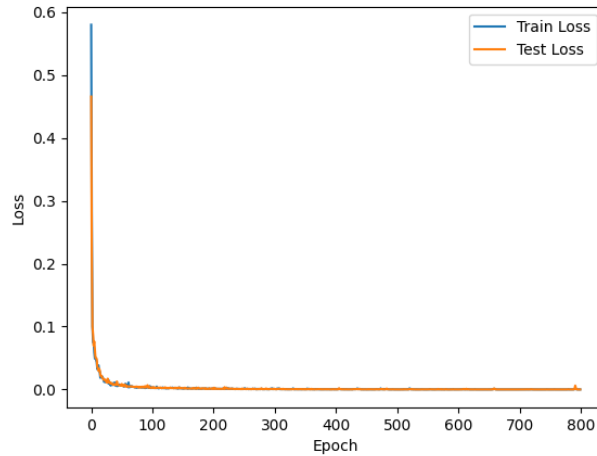


Figure 3: Transformer - model architecture

4.2 Algorithm Performance Analysis

There are two common metrics for measuring tag reading efficiency: system throughput and the number of consumed slots. This paper uses the total number of slots consumed during a single reading process as the measurement metric; the fewer slots consumed, the more accurate the estimation of the remaining tags. Figure 4 compares the number of slots consumed by the Vogt algorithm and the proposed algorithm when recognizing the same number of tags.

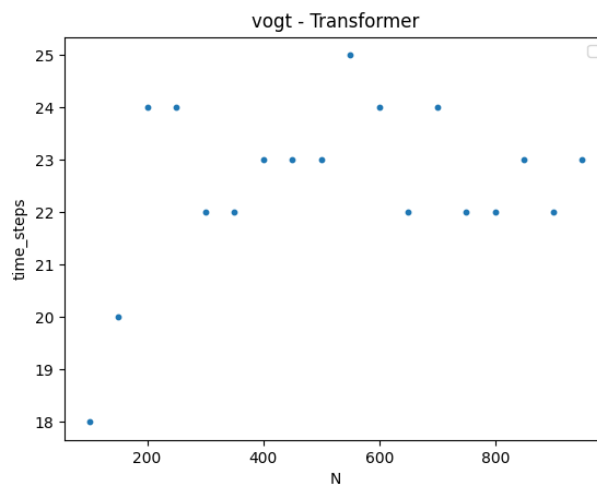


Figure 4: Comparison of the number of consumed slots

As shown in Figure 4, the number of slots consumed by the Transformer algorithm proposed in this paper is lower than that of the traditional Vogt algorithm.

5. Conclusions

This paper proposes an RFID tag recognition optimization method that combines the Transformer

and DFSA algorithms. By introducing the Transformer's self-attention mechanism, the method not only improves the accuracy of tag number prediction but also optimizes the frame length adjustment strategy. Experimental results show that the algorithm significantly enhances tag recognition efficiency and system performance compared to traditional methods, especially in large-scale tag application scenarios. Future research could further improve the algorithm's adaptability and real-time performance, explore hardware acceleration solutions, and integrate RFID with multimodal technologies to enhance computational efficiency and expand application scenarios, while also strengthening privacy and security protections.

References

- [1] F. Costa, S. Genovesi, M. Borgese, A. Michel, F. A. Dicandia, and G. Manara, "A Review of RFID Sensors, the New Frontier of Internet of Things," *Sensors*, vol. 21, no. 9, p. 3138, 2021.
- [2] Bao Shengwen, Fang Yongjun, Xiao Xiaobing, et al. *Research and Application of RFID Technology in Medical Equipment Management System* [J]. *Electronic Product World*, 2022, 29(03): 43-46.
- [3] Xiao Chang, Ding Huan, Fu Huiling, et al. *Design and Application of Medical Fabric Management Module Based on RFID and Barcode Technology* [J]. *China Medical Equipment*, 2022, 37(06): 119-122.
- [4] Fu Hao, Lin Kan, Qin Tong. *Research on Intelligent Warehouse Management System in Power Plants Based on RFID Technology* [J]. *Electric Power Survey and Design*, 2022, (05): 72-76.
- [5] Li Lingqiao. *Application of RFID Technology in the Field of Clothing Warehousing and Logistics* [J]. *Chemical Fiber & Textile Technology*, 2022, 51(09): 58-60.
- [6] Liu Feiran, Xu Dong, Tian Yunfeng, et al. *Application of RFID Technology in the Informationization of Urban Rail Transit Vehicle Maintenance* [J]. *Urban Rail Transit Research*, 2023, 26(10): 148-152.
- [7] He Tao. *Application of RFID Technology in Elevator Safety Management System* [J]. *IC Application*, 2022, 39(03): 176-177.
- [8] Yan Yijia, Yan Hongtao, Wang Zhibiao, et al. *Application of RFID in Safety Warning Management System in High-risk Places* [J]. *Science & Technology Innovation and Application*, 2019, (27): 167-168.
- [9] Wang Shuai. *Research on the Performance of RFID Anti-collision Algorithms* [D]. *Nanjing University of Posts and Telecommunications*, 2020.
- [10] Hu Huangshui, Zhang Guo, Wang Hongzhi, et al. *An RFID Anti-collision Algorithm Based on Equal-area Division* [J]. *Journal of Jilin University (Science Edition)*, 2020, 58(01): 120-126.
- [11] Wang Dongyun, Zhang Weiping, Wang Zhijia. *Research on RFID Tag Anti-collision Based on Improved ALOHA Algorithm* [J]. *Packaging Engineering*, 2021, 42(17): 244-248.
- [12] Yang Chao, Zhao Yifan, Li Bo, et al. *Research on RFID Tag Anti-collision Algorithm Based on Deep Learning* [J]. *Modern Electronics Technique*, 2021, 44(22): 21-25.
- [13] Ming Dongyue, Wang Shangpeng, Lei Ming, et al. *Improved Dynamic Framed Slotted ALOHA Algorithm Combined with BP Neural Network* [J]. *Mini-Micro Systems*, 2021, 42(09): 1920-1923.
- [14] Shayo E I, Abdalla A T, Mwambela A J. *Dynamic multi-frame multi-spreading factor scheduling algorithm for LoRaWAN* [J]. *Journal of Electrical Systems and Information Technology*, 2023, 10(1). DOI:10.1186/s43067-023-00077-2.
- [15] Fan Caiyun, Sun Ruyi, Tong Junyi. *Air Quality Index Prediction Based on Spatial Factors and LSTM Neural Network* [J]. *Practice and Understanding of Mathematics*, 2021, 51(15): 194-202.
- [16] Xu Lin, Xu Tingting. *Population Prediction Method Based on BP Neural Network Model* [J]. *Information & Computer (Theoretical Edition)*, 2023, 35(04): 181-183.
- [17] Guo Hui, Liu Yunting, Kong Zhenxing, et al. *Research on VO2max Prediction Method Based on BP Neural Network* [J]. *Laboratory Science*, 2020, 23(05): 44-48.
- [18] Ren Huan, Wang Xuguang. *A Survey of Attention Mechanism* [J]. *Application Research of Computers*, 2021, 41(S1): 1-6.
- [19] Wen Q S, Zhou T, Zhang C L, et al. *Transformers in Time Series: A Survey* [C]. *IJCAI 2023*: 6778-6786.
- [20] Liang Hongtao, Liu Shuo, Du Junwei, et al. *A Survey on the Application of Deep Learning in Time Series Prediction* [J]. *Journal of Computer Science and Exploration*, 2023, 17(06): 1285-1300.
- [21] Zhou Weihui, Jiang Niande, Wan Xinyue, et al. *A Hybrid RFID Anti-collision Algorithm Based on ALOHA and Multi-tree* [J]. *Journal of East China University of Technology (Natural Science Edition)*, 2021, 44(01): 96-100.
- [22] Lü Congying. *A Novel RFID Tag Anti-collision Algorithm* [J]. *Manufacturing Automation*, 2013, 35(20): 75-76.