A Bio-inspired Smart Access Algorithm for Large Scale Self-organizing Wireless Networks

Enfu Jia(✉), Jiaming Cao, Xiaorong Zhu, and Jinfeng Li

Nanjing University of Posts and Telecommunications, Nanjing 210003, China
1018010223@njupt.edu.cn

Abstract. Efficient wireless access of a large number of nodes is one important issue for large scale self-organizing wireless networks. In order to achieve this goal, collision problem between nodes must be resolved. Bio-inspired algorithms provide some significant characteristics such as stability, adaptability, and scalability, and hence many researchers have attempted to apply bio-inspired algorithms to solve some problems in networks. In this paper, we propose a smart access algorithm for large scale self-organizing wireless networks, which is inspired by Stigmergy, which is able to make group members implement information interaction symmetry in some ways, and members are able to influence and interact with each other to avoid collision. Then we build an analysis model based on Markov chain for the proposed algorithm. Simulation results show that the proposed algorithm can maintain a low collision probability with the increasing number of competing nodes even in a dynamically changing network topology. In addition, the results show that compared with traditional algorithms, the proposed algorithm has better performances on channel throughput and access delay.

Keywords: Bio-inspired · Stigmergy · Self-organizing · Large scale wireless networks

1 Introduction

With the rapid development of the Internet of Things, it brings unprecedented opportunities and challenges. For example, with the explosive growth of the number of nodes, in order to implement efficient access without or with little collision between nodes, the nodes need to have a high degree of self-organization in the current network. As is known that self-organizing phenomenon is a very common phenomenon in nature. It is caused by the cooperative interaction between individual organisms. The full openness of self-organizing phenomena, autonomous interaction, decentralized control, complexity, and emergence have promoted the development of various disciplines. For example, Grasse studied nesting and foraging behavior of ant colonies in 1959 [1, 2]. He called this phenomenon "Stigmergy", which has been widely studied in aspect of social insects [3] and applied in different scenes [4–6].

Q. Wu et al. (Eds.): WiSATS 2020, LNICST 358, pp. 611–623, 2021.
https://doi.org/10.1007/978-3-030-69072-4_50

Stigmergy is an information coordination mechanism for biological individual autonomy. It is an indirect coordination mechanism between agents or individual behaviors [7]. In the absence of central control and direct contact with communication, the group implements information interaction symmetry through some methods, such as co-oscillation. Individuals act independently and influence and interact with each other on the basis of perceiving changes in the external environment, which can generate more complex and highly intelligent group structures through the adjustment and modification of members within the group. Therefore, it can be considered to resolve the collision problem that is easy to happen among the distributed multi nodes competing to access the same channel in the future complex and changing network environment. To the best of our knowledge, using Stigmergy for collision in wireless network has not been considered. The contributions of this work are as follows:

- We propose a smart access algorithm for large scale self-organizing wireless networks, which is inspired by Stigmergy, which is able to make group members implement information interaction symmetry in some ways, and members are able to influence and interact with each other to avoid collision.
- We build an analysis model based on Markov chain for the proposed algorithm, which mainly analyzes collision probability, channel throughput and access delay.
- Simulations results including collision probability, channel throughput and access delay are compared with the CSMA/CA mechanisms.

The rest of this article is organized as follows. In Sect. 2, a suitable system model is built by analyzing the current network topology. Moreover, we propose a new access algorithm. In Sect. 3, the proposed algorithm is divided into three phases, and each phase is analyzed and described in detail. In Sect. 4, according to the proposed algorithm, the system performance is analyzed. In Sect. 5, the proposed algorithm is compared with the CSMA/CA mechanisms. In the simulations results, the efficiency and advantages of the algorithm are verified. Finally, we summarize and prospect this paper.

2 System Model

As shown in Fig. 1, it is a self-organizing wireless network composed of multiple distributed nodes. Learning from the Stigmergy mechanism, a coordination mechanism between a large-scale number of nodes in the wireless network is implemented, which is based on the pheromone communication interaction. It provides an effective solution to reduce collision that is prone to occur in the network.

In this paper, a similar cooperation mechanism model is established between nodes in a wireless network through the Stigmergy cooperation mechanism model, as shown in Fig. 2. Each node is regarded as a different agent. As the agent's behavior state changes and if the state reaches a certain condition, it will send broadcast information in the network. This process is regarded as the agent leaving pheromone in the environment. The other agents adjust their current behavior state through receiving the information.

According to the established cooperative mechanism model, we propose a bio-inspired smart access algorithm (BSAA). This algorithm is a kind of de-synchronization

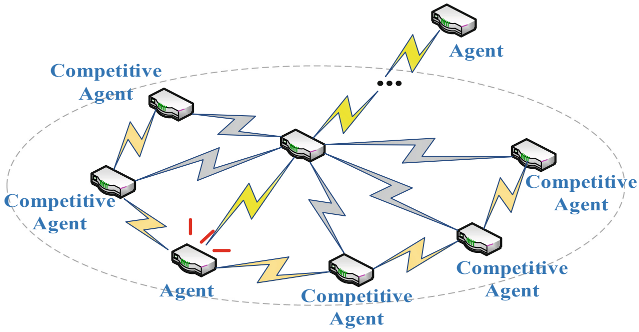

Fig. 1. A self-organizing wireless network composed of multiple distributed nodes.

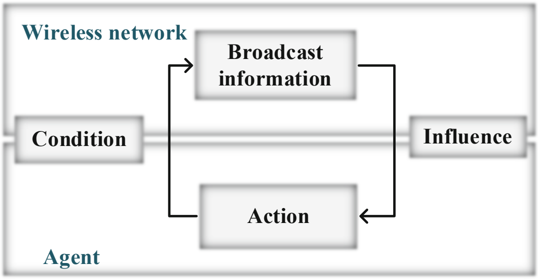

Fig. 2. Cooperative mechanism model between nodes in wireless network.

[8, 9] that is implemented in the paper by means of combining the cooperative mechanism model and the phase oscillation model in synchronization theory. We make the phase represent the state of node. Then node modifies its own behavior to achieve desynchronization by means of interacting with other nodes.

3 Algorithm Introduction

The state of each node at time t is regarded as a phase value $\theta(t)$. The behavior of the node includes phase change and phase update. Each node has an initial phase value, which is determined by the service type of the data and the remaining energy of the current node. The phase change of a node is similar to the oscillation of a phase oscillator. When the phase value of the node reaches 2π, the node can access the channel to transmit data packets and inform the adjacent nodes of the current status by the way of sending broadcast information. If a node receives the broadcast information from adjacent node during the phase change, it will extract the information and immediately send broadcast information to the adjacent node as well. At the same time, the node updates the phase to avoid collision with the node that is transmitting data. According to the process, this paper divides the algorithm into three stages, as shown in Fig. 3.

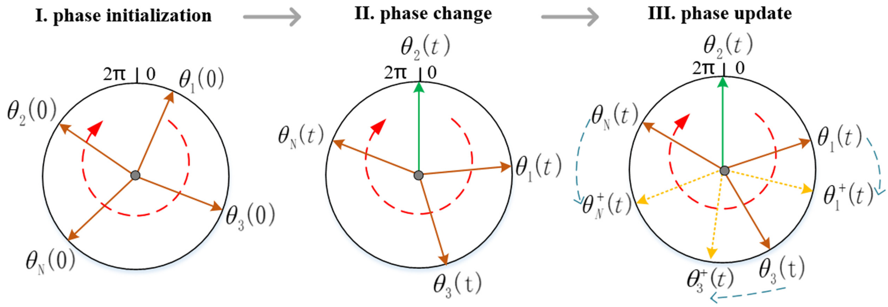

Fig. 3. Phase diagram of nodes in three stages.

The three stages are described in detail.

1) Phase Initialization

Each node $n(n = 1, 2, 3, \ldots, N)$ sets up an initial phase value, which is represented as follows:

$$\theta_n(1) = w_n \tag{1}$$

We consider the service type and remaining energy as factors to calculate the initial phase. The initial phase value of the node with the higher priority data and large remaining energy is larger. Therefore, the Eq. (1) is further developed as follows:

$$w_n = \begin{cases} \left(\alpha \frac{[T]}{k_1+[T]} + (1-\alpha)\frac{E_{rest}/E_{max}}{k_2+E_{rest}/E_{max}}\right) * \frac{2\pi}{N}, & N > 1 \\ 2\pi & , N = 1 \end{cases} \tag{2}$$

where, $[T]$ represents the proportion of the current data type of the node to all data types, which satisfies $0 < [T] \le 1$. k_1 and k_2 are constants. E_{rest} represents the remaining energy of the node. E_{max} represents the total energy of the node. E_{rest}/E_{max} represents the remaining energy rate of the node, which satisfies $0 < E_{rest}/E_{max} \le 1$. N represents the number of nodes in the range of the node n. This formula indicates that when there is only one node competing for the channel, the data is sent directly and when there is more than one node, different initial phases values must be calculated.

2) Phase Change

The phase value of node n is a discrete and time-varying function, then the phase change should satisfy as follows:

$$\theta_n(t) = w_n t \tag{3}$$

When $\theta_n(t)$ reaches 2π at time t, the node n will send adjacent nodes broadcast information including the phase value of the current node and the estimated time of the occupied channel. Then it starts transmitting data packets, and resets its phase value to 0. If the node n receives broadcast information from an adjacent node during the phase change, it will immediately carry out the phase update and send a broadcast information including the phase value of the current node.

In order to improve system performance, the phase value of a node is updated up to x times after reaching 2π. If the number of updates exceeds x, the node no longer carries out the phase updates, and if a collision occurs, the data packet is discarded.

3). Phase Update

After receiving the broadcast information from adjacent node m, the node n extracts the information to carry out the phase update. The phase of the updated node is expressed as follows:

$$\theta_n\left(t^+\right) = \theta_n(t) + F_n(t) = w_n t + F_n(t) \tag{4}$$

where, $\theta_n\left(t^+\right)$ represents the phase value for the node n after updating and the t^+ satisfies $\lim_t (t^+ - t) = 0$. $F_n(t)$ represents the value of phase update.

The node n makes different update rules based on the broadcast information of different adjacent nodes. We divide them into two cases.

- When $\theta_n(t) \geq \theta_m(t)$, it indicates that when the phase value of node n is larger than the phase values of other nodes except the node that is transmitting the data packet. Due to the phase update is only related to the node that is transmitting the data packet, the updated phase value of node n is calculated based on the estimated time T_m(the channel occupied by the broadcast information sent by the node m). Hence, the value of phase update satisfies as follows:

$$\theta_m(t) - \theta_n\left(t^+\right) = \frac{d\theta_n(t)}{dt} * T_m \tag{5}$$

- When $\theta_n(t) < \theta_m(t)$, it indicates that there are $M\,(M = 1, 2, 3, ..., N - 1)$ nodes whose phase value is larger than node n. In this paper, we only consider the influence of two nodes with the smallest phase difference from node n. Therefore, the value of phase update function is expressed as follows:

$$F_n(t) = K \sum_{m=1}^{2} G_{mn}(t) \sin(\theta_m(t) - \theta_n(t) - \theta_\alpha) \tag{6}$$

where, K is a coupling coefficient. In order to make the value of the phase difference between the two nodes in $[0, 2\pi)$, the Mod function is applied to the phase difference of the nodes, that is,

$$\emptyset_{m,n}(t) = mod(\theta_m(t) - \theta_n(t), 2\pi) \tag{7}$$

When the node's phase difference $\emptyset_{m,n}(t) = 0$, in order to avoid the value of the phase update function being 0, a phase delay θ_α is introduced into the phase difference function, which is used to solve the problem between adjacent nodes.

$G_{mn}(t)$ represents the weight function of the phase difference, which is defined as follows:

$$G_{mn}(t) = \begin{cases} e^{-\emptyset_{m,n}(t)}, 0 \leq \emptyset_{m,n}(t) < \pi \\ e^{-2\pi + \emptyset_{m,n}(t)}, \ \pi \leq \emptyset_{m,n}(t) < 2\pi \end{cases} \tag{8}$$

4 System Performance Analysis

4.1 Research Model

This paper analyzes packet transmission probability, channel throughput and access delay. Assuming that there are a fixed number of nodes and an ideal single channel. According to the BSAA, the scenario where multiple nodes access the same channel is modeled as a two-dimensional Markov model with the number of updates and time slice as state variables, as shown in Fig. 4. Let $s(t)$ represent the number of node updates at time t, which satisfies $s(t) \in [0, m]$. m is the maximum number of updates. Let $b(t)$ represent process of reducing the node's time slice (TS). The TS is a value of size L. t and $t+1$ represent the starting time of two adjacent slot time. And at the beginning of each slot time, L is reduced by one unit of time. Therefore, $\{s(t), b(t)\}$ can be regarded as a two-dimensional Markov model in discrete time. Let p represent the probability of the channel busy, and $(1-p)$ be the probability that a node can transmit a data packet. Moreover, each node corresponds to an L value on each update, that is, $L_i = \frac{\emptyset_{m,n}(t) - F_n(t)}{W_n}$, $i \in (0, N-1]$, and the value of L_i will change accordingly after each update.

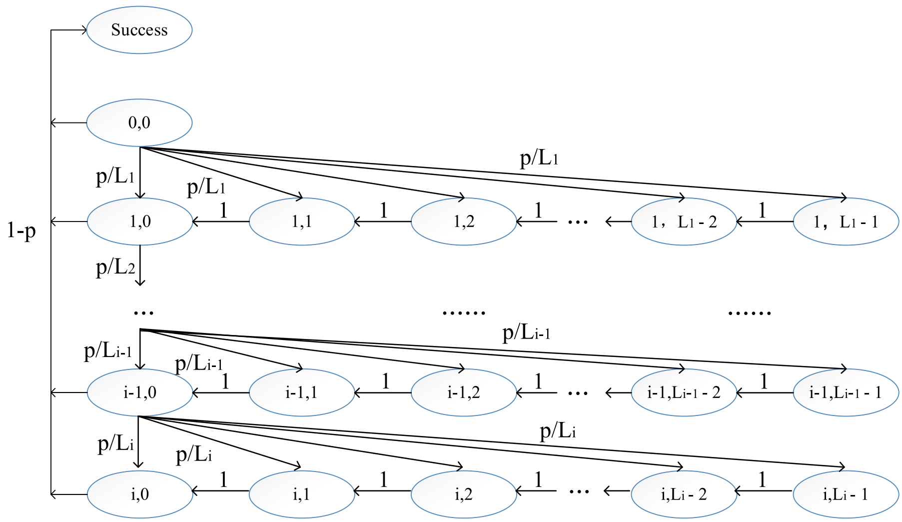

Fig. 4. Two-dimensional Markov model.

From the Markov model shown in Fig. 4, one step state transition probability can be obtained as follows:

$$\begin{cases} P\{i, k|i, k+1\} = 1, i \in (0, N-1+x), k \in [0, L_i - 2] \\ P\{i, k|i-1, 0\} = \frac{p}{L_i}, i \in (0, N-1+x), k \in [0, L_i - 1] \end{cases} \tag{9}$$

where, the first formula indicates that at the beginning of each slot time, the value of L_i will decrease by one unit, and it is inevitable that $k+1$ will become k. The second formula indicates that after the $i-1$th update, if a node receives broadcast information from an adjacent node at a certain time k, it will update its own phase again and recalculate the phase value.

4.2 Packet Transmission Probability

In order to calculate the channel throughput, it is necessary to obtain the packet transmission probability. Let τ represent the probability that the phase value of the node reaches 2π in a slot time. According to the BSAA, the node is able to transmit data packets only when the phase value reaches 2π, and there should be only one node in the channel for data packet transmission. Let $b_{i,k} = \lim_{t\to\infty} P\{s(t) = i, b(t) = k\}, i \in [0, N-1+x], k \in [0, L_i - 1]$ represent the steady-state distribution of the two-dimensional Markov model. From Eq. (9), the steady state distribution for any state is obtained as follows:

$$b_{i,k} = \frac{L_i - k}{L_i} b_{i,0} = \frac{L_i - k}{L_i} p^i b_{0,0}, \quad i \in [0, N-1+x] k \in [0, L_i - 1] \tag{10}$$

From the nature of the Markov model, the probability sum of the states is 1, that is,

$$1 = \sum_{i=0}^{N-1+x} \sum_{k=0}^{L_i - 1} b_{i,k} \tag{11}$$

According to the definition of τ, the phase value of the node reaches 2π when L_i decreases to 0, and its probability is calculated as follows:

$$\tau = \sum_{i=0}^{N-1+x} b_{i,0} \tag{12}$$

The probability that a node transmits a packet but the channel is busy is equivalent to the probability that at least one node has a phase value of 2π in the same slot time. Therefore, p can be expressed as follows:

$$p = 1 - (1 - \tau)^{N-1} \tag{13}$$

4.3 Channel Throughput

According to the definition of throughput, the channel throughput is expressed as the ratio of the payload to the time of successful transmission in the channel. Let P_u represent the probability that the phase value of at least one node reaches 2π, which is equivalent to the probability of a node carrying out the phase update. P_u can be expressed as follows:

$$P_u = 1 - (1 - \tau)^N \tag{14}$$

Let P_s represent the probability of successfully transmitting a data packet when the phase value of a node reaches 2π, which can be expressed as follows:

$$P_s = \frac{N\tau(1-\tau)^{N-1}}{P_u} = \frac{N\tau(1-\tau)^{N-1}}{1-(1-\tau)^N} \tag{15}$$

Let S represent the channel throughput, which can be expressed as follows:

$$S = \frac{E[P_s]}{E[t_s]} \tag{16}$$

where, $E[P_s]$ represents the expectation of the size of a successfully transmitted data packet in a slot time, and $E[t_s]$ represents the expectation of the length of a slot time.

Assuming that the expectation of a data packet size is $E[P]$, the amount of data successfully transmitted in a slot time is $P_u P_s E[P]$. A slot time consists of a slot time with no nodes with a phase value of 2π, a slot time with no phase update when the phase value of a node reaches 2π, and a slot time with the phase update when the phase value of a node reaches 2π. Therefore, the throughput S can be further expressed as follows:

$$S = \frac{P_u P_s E[P]}{(1 - P_u)\sigma + P_u P_s T_s + P_u (1 - P_s) T_u} \tag{17}$$

where, σ represents a complete length of slot time. T_s represents the time required for a node to transmit a data packet without the phase update. T_u represents the time required for a node to carry out the phase update. The expressions for T_s and T_u are as follows:

$$\begin{cases} T_s = H + E[P] + T_{bro} \\ T_u = H + E\big[P^*\big] + \delta + T_{bro} \end{cases} \tag{18}$$

where, $H = PHY_{hdr} + MAC_{hdr}$ represents the sum of the physical layer header length and the MAC layer header length. δ represents the propagation delay. T_{bro} represents the time required to send broadcast information. $E\big[P^*\big]$ represents the expectation of the data packet size of the node that finally sends data packet in the channel when the phase update occurs. In order to facilitate calculation, the length of all transmitted packets is the same.

4.4 Access Delay

In the BSAA, the access delay mainly includes the phase change delay and the phase update delay of the node. It can be known that the phase update delay exists with a certain probability. Therefore, the access delay is analyzed based on whether there is the phase update during the phase change.

Assuming that T_s^* represents the delay of successfully transmitting a data packet when there is no phase update during the phase change. Then,

$$T_s^* = (1 - P_u)\left(\frac{\theta_n}{w_n} + T_s\right) \tag{19}$$

Assuming that T_u^* represents the delay of successfully transmitting a data packet when there is the phase update during the phase change. Then,

$$T_u^* = P_u\left(\sum_{i=1}^{N-1+x}\left(\frac{\theta_{n,i} - \theta_{n,i-1}}{w_n} + L_i\right) + T_u\right) \tag{20}$$

where, $\frac{\theta_{n,i}-\theta_{n,i-1}}{w_n}$ represents the time that it takes for a node to change the phase value before the next phase update, and $\theta_{n,0} = 0$. i represents the update times.

In summary, the access delay T can be expressed as follows:

$$T = T_s^* + T_u^* = (1 - P_u)\left(\frac{\theta_n}{w_n} + T_s\right) + P_u\left(\sum_{i=1}^{N-1+x}\left(\frac{\theta_{n,i} - \theta_{n,i-1}}{w_n} + L_i\right) + T_u\right) \tag{21}$$

5 Experimental Simulation

According to the analysis in the previous section, this paper uses MATLAB to verify them and compare with the CSMA/CA mechanisms meanwhile. The number of nodes in the simulation experiment starts from 5. There are at most 50 competing nodes at the same time, and each time it increases by 5 nodes. The service types of data are divided into three categories: emergency services, streaming media services, and ordinary data services. Nodes are randomly selected among the three service types. Other simulation parameters are shown in Table 1.

Table 1. Simulation parameters.

Parameter name	Parameter value
Packet payload	3808 bits
MAC header	240 bits
PHY header	192 bits
ACK	112 bits + PHY header
RTS	160 bits + PHY header
CTS	112 bits + PHY header
Propagation delay	1 µs
Slot time (σ)	20 µs
SIFS	10 µs
DIFS	50 µs
Channel bit rate	2 Mbit/s
E_{max}	50000 J
x	6

In the experiment, we determine whether a collision has occurred by calculating the phase difference between each node and the remaining nodes. As shown in Fig. 5 and Fig. 6, they show the time series data of the phase difference between this node and the other nodes when the number of competing nodes is $N = 20$ and $N = 40$. It can be seen from Fig. 5 that the phase difference is 0 at about $t = 650$, but at this time the number of updates x is 1. Therefore, no collision will occur, and the node will continue to carry out the phase update. While in Fig. 6, there are two moments that the phase difference is 0, where the number of node updates x is 3 when $t = 750$, and x is 5 when $t = 1350$. Compared to Fig. 5, the number of node updates x is bigger. The reason is the number of competing nodes increases, which causes the probability that the phase value of the node reaches 2π at the same time to increase. However, the phase difference between the two nodes is randomly distributed between $(0, 2\pi)$ at most of time, which indicates that the probability of packet collision between nodes is very small.

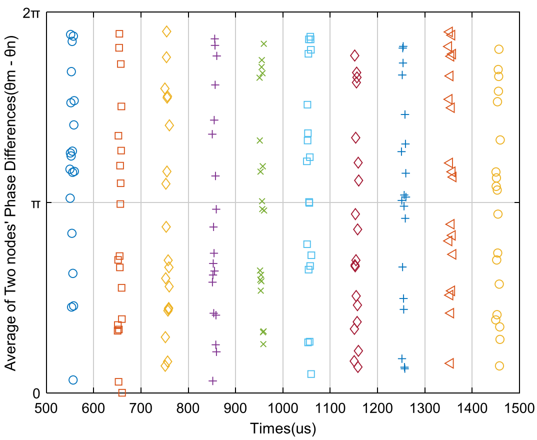

Fig. 5. Time series data of phase difference with adjacent nodes when $N = 20$.

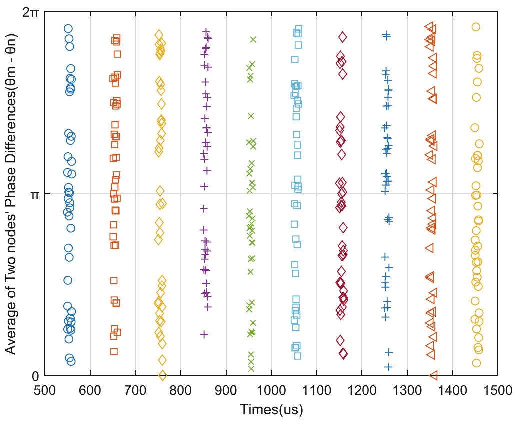

Fig. 6. Time series data of phase difference with adjacent nodes when $N = 40$.

Figure 7 shows the comparison graph of the collision probability between the BSAA and the CSMA/CA mechanisms. It can be seen from the Fig. 7 that as the number of competing nodes increases, the curves of the collision probability of all three show an upward trend.

In the BSAA, nodes are able to adjust their behaviors to avoid sending data packets at the same time, therefore, the probability of collision is relatively smaller than the CSMA/CA mechanism. In addition, the larger the number of competing nodes, the greater the gap between the two, which indicates that our proposed algorithm is able to avoid collisions better in the case of a large number of nodes.

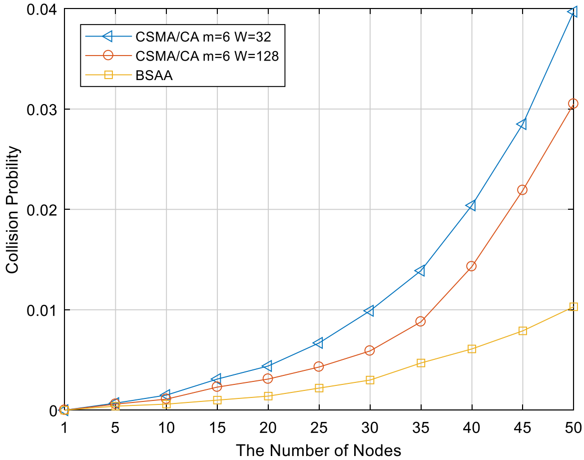

Fig. 7. Comparison of the collision probability between BSAA and CSMA/CA.

Figure 8 is a comparison graph of the channel throughput performance. It can be seen from the Fig. 8 that the channel throughput performance of the BSAA is overall improved by about 12%–15% compared to the CSMA/CA mechanisms. The throughput here refers to the average time of the payload successfully transmitted in a time slot to the length of the entire time slot. For the BSAA, the node is more likely to get access to the channel, and the length of the entire time slot is smaller than the CSMA/CA mechanisms, therefore, the BSAA is able to get better performance.

Figure 9 is a comparison graph of the access delay performance. It can be seen from the Fig. 9 that when the number of competing nodes is small, the access delay of the BSAA is slightly longer than the CSMA/CA ($m = 6$, $W = 32$) mechanisms. This is because when there are fewer nodes, the backoff waiting time of the nodes will not be long in the CSMA/CA mechanisms. While each node needs to go through the phase change and the phase update in the BSAA, and its time may be slightly longer than the backoff waiting time. Therefore, when the number of nodes is small, the delay performance of the BSAA is not as good as the CSMA/CA mechanisms. However, with the increase in the number of competing nodes, the time for backoff retransmission and backoff waiting is increasing, and the delay is also increasing in the CSMA/CA mechanisms. While in the BSAA, the increase in the time spent by the nodes on the phase change and the phase update is not very large. Therefore, in the case of a large number of competing nodes, the delay performance of the BSAA is better than the CSMA/CA mechanisms.

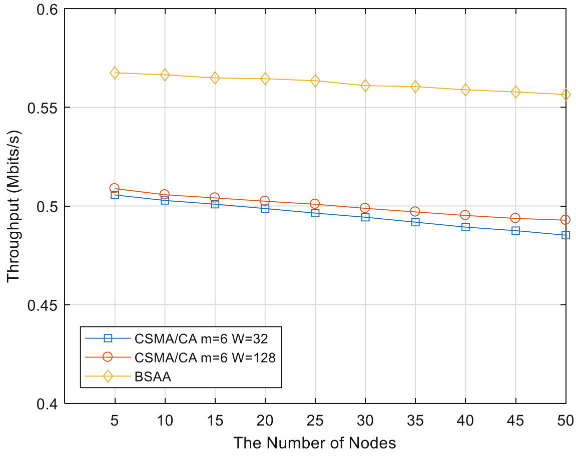

Fig. 8. Comparison of channel throughput between BSAA and CSMA/CA.

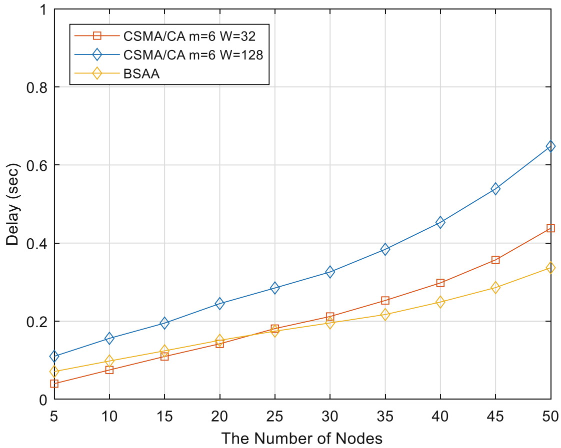

Fig. 9. Comparison of access delay between BSAA and CSMA/CA.

6 Conclusion

Inspired by Stigmergy, this paper proposes an algorithm to resolve access collisions between massive nodes in wireless networks. The algorithm makes each node completely self-organize. In addition, by means of analyzing the initialization status and update status of the nodes in different situations, adjacent nodes are able to self-adjust the current behavior to avoid collisions when one node is transmitting data packets. Simulation results show that the proposed algorithm is able to maintain a low collision probability with the increasing number of competing nodes even in a dynamically changing network topology. Moreover, comparing the proposed algorithm with CSMA/CA mechanisms in terms of collision probability, channel throughput and access delay. The results show that the collision probability and access delay of our proposed algorithm performs better in the case of the large number of competing nodes, and the performance of channel throughput is overall improved by about 12%–15%.

References

1. Grassé, P.P.: La reconstruction du nid et les coordinations inter—Individuelles chez Bellicositer- mes natalis et Cubitermes sp. La théorie dela stigmergie. Insectes Sociaux **6**(1), 41–80 (1959)
2. Göllner, U.: Grassé, Pierre-P.: Fondation des société—Construction. termitologia. 2. 624 S., 452 Figure., 28 Tab., Masson, Paris, New York, Barcelona, Milan, Mexico, Sao Paulo. Deutsche Entomologische Zeitschrift **32**(4–5), 379–379 (1984)
3. Dorigo, M., Bonabeau, E., Theraulaz, G.: Ant algorithms and stigmergy. Future Gener. Comput. Syst. FGCS **16**(8), 851–871 (2000)
4. Kassabalidis, I., El-Sharkawi, M.A., Marks, R.J., et al.: Swarm intelligence for routing in communication networks. In: Proceedings of the IEEE Global Telecommunications Conference on, San Antonio, TX, USA, 25–29 November 2001, pp. 3613–3617 (2001)
5. Werfel, J., Nagpal, R.: Extended stigmergy in collective construction. IEEE Intell. Syst. **21**(2), 20–28 (2006)
6. Alfeo, A.L., Cimino, M.G.C.A., Egidi, S., et al.: A stigmergy-based analysis of city hotspots to discover trends and anomalies in urban transportation usage. IEEE Trans. Intell. Transp. Syst. **19**(7), 2258–2267 (2018)
7. Tummolini, L., Castelfranchi, C.: Trace signals: the meanings of stigmergy. In: Weyns, Danny, Parunak, H Van Dyke., Michel, Fabien (eds.) E4MAS 2006. LNCS (LNAI), vol. 4389, pp. 141–156. Springer, Heidelberg (2007). https://doi.org/10.1007/978-3-540-71103-2_8
8. Degesys, J., Rose, I., Patel, A., et al.: DESYNC: self-organizing desynchronization and TDMA on wireless sensor networks. In: 6th International Symposium on Information Processing in Sensor Networks 2007, Cambridge, MA, USA, 25–27 April 2007, pp. 11–20 (2007)
9. Degesys, J., Nagpal, R.: Towards desynchronization of multi-hop topologies. In: 2008 Second IEEE International Conference on Self-adaptive and Self-organizing Systems, Venezia, Italy, 20–24 October 2008, pp. 129–138 (2008)