IOP Conference Series: Materials Science and Engineering

PAPER • OPEN ACCESS

A Fast and Collision Avoidance Distributed TDMA Schedule Based on the Multi-Arms Bandit

To cite this article: ChaoYi Zheng *et al* 2019 *IOP Conf. Ser.: Mater. Sci. Eng.* **608** 012007

A Fast and Collision Avoidance Distributed TDMA Schedule Based on the Multi-Arms Bandit

ChaoYi Zheng[1,*], **ShengChun Huang**[1] **and TaiLi Li**[1]

[1]College of Electronic Science, National University of Defense Technology, Changsha 410073, China

[*]Corresponding author : zcy754420194@163.com

Abstract. In this paper, we propose a novel distributed TDMA protocol based on the Multi-Arms Bandit model for the dynamic UAVs environment. Because of the frequent topology changes of UAVs, we consider a distributed communication protocol, which do not require the maintenance of accurate network topology information. Through the adaptive interaction between nodes, perceiving the surrounding topology environment and learning from historical experience, each node in the network can transmit information in a fast collision-free way. Also, the MAB model is utilized in our protocol, so that the time slot can obtain sufficient multiplexing rate through multiple rounds of node selection. Experiments show that the protocol can achieve better throughput and fast enough convergence speed, even in the case of high network density, and scales well with the size of the network.

1. Introduction

In recent years, Unmanned Air Vehicles(UAVs) have developed rapidly, and their applications have been utilized from civilian to military. Because of the complex environment, sending and receiving data requirements, and the long fly distance, the topology of the Unmanned Air Vehicles(UAVs) swarm changes so quickly that we have a high demand for resource scheduling, which needs fast convergence and stabilized. To coordinate the access of the shared wireless resources, a MAC protocol is employed.

MAC design aims at optimizing the performance of communication by formulating the strategies sensor nodes use to access the common channel [1]. At present, the distributed TDMA protocol applies universally for UAVs swarm, which can be broadly divided into two categories : topology-transparent scheduling schemes and topology-based scheduling schemes. The nodes with topology-based scheduling schemes need network topology information to schedule schemes [2][3], and it will cause plenty of computation and communication overhead, which likely consume UAVs excessively. Therefore, topology-transparent scheduling schemes attracted considerable research interest.

Rhee et al. [4] proposed the DRAND scheme which is a randomized distributed time slot scheduling algorithm. It allocated the time slot sequence by coordinating requests among network node. However, the completely randomness decreased the efficiency, and also resulted in the high collision rate of the message in the distribution process. Based on the DRAND, LI et al. [5] made improvement and proposed the E-T DRAND which allocates time slot through the remainder energy for each node, the results show that there is a certain improvement in the scheduling rate and an appropriately decrease in the overhead.

And other researchers discuss mathematics models to schedule scheme and attempt to get a collision-free allocation through probability analysis. In [6][7], Farago et al proposed a topology-transparent algorithm based on Polynomial function. The algorithm's performance depends on the number of nodes in the network and the maximum degree, i.e. the number of neighbors that each a node can have. In [8], Ju and Li improved the algorithm in order to maximize the minimum throughput. They used the Hamming weight and the Hamming distance in order to describe the relationship between the transmission slot assignments of two nodes. Liu et al. [9] proposed a topology-transparent algorithm, i.e. m-MPR-l-code algorithm. The results showed that the algorithm had a certain improvement on the performance of resource scheduling. The polynomial-based approach can achieve considerable improvements in efficiency and robustness in mobile environments [10] by taking advantage of the multi-packet reception capability. However, these approaches introduce complicated polynomial-based algorithms and an important problem is that it exits redundant slots not to be utilized efficiently.

As machine learning developing rapidly, plenty of researchers combined the allocation with Machine Learning and Reinforcement learning models, which is very popular recently. In [11], Liu et al proposed a reinforcement learning model named the Multi-Armed Bandit with Multiple players. And the RL-Mac protocol [10] used the Markov decision to infer the state of other nodes, the results show that the protocol can achieve a high throughput, but the computational complexity and power consumption are some high. Qiao et al. [12] combined the polynomial function with reinforcement learning and gradient descent model, which chooses the slot and reduce redundant time slots. The theoretical performance of the algorithm is better than [13], and there is a certain improvement in throughput and convergence speed.

In the paper, we propose a novel time slot algorithm based on Multi-Armed Bandit model so that each node can be allocated with a collision-free time slot. Considering the surrounding environment and interaction of each node, we enable each node to learn from history knowledge to better performance on convergence and scalability. Our main contributions consist in :

1. We propose a mac protocol based on the MAB model, which is a distributed, modelless, online learning algorithm. In this algorithm, each node only feeds back its own local information, which can reduce the convergence time of the algorithm and acheive a better throughput.
2. A large scale dynamic simulation network is implemented on the matlab, and we compare the algorithm with other algorithms from the aspects of convergence speed and throughput to evaluate the overall scheme performance.

2. System model

2.1 Network Model

We consider a wireless communication network that contains M nodes, and the initial density of the network is medium. each node does not need to support multi-packet reception and just knows its own address and the sum of time slots. The network can be represented by an directed graph G(V,E), where V and E are the sets of nodes and edges respectively to indicate some nodes are connected. The degree of a node u, D(u), is defined as the number of its interference neighbors. The maximum node degree D_{max} is defined as D_{max} = max D(u). We assume that D_{max} is smaller than the number of nodes N and remains constant while the network topology changes. In our paper, time is divided into frames of fixed duration, each one consisting of N time slots. At the beginning of each frame each node randomly chooses one of the N available slots to transmit its packet and start the algorithm we propose. Also, We propose a definition referred to as Priority Factor(Pr): $Pr = F(\alpha Dv - \beta Nv)$, where Nv is the number of time slots occupied by nodes v, α and β are coefficients($\alpha, \beta \in [0,1]$). According to the topology, each node will have their own Pr to have positive influence on network 's business and throughput, it will be mentioned in the follow section. We assume that the transmission channel is error-free and the only reason for unsuccessful transmission between nodes is due to collisions, for a transmission from node v to its neighbor in slot t, a collision occurs when another node which is within two-hop neighbors range of node k simultaneously intends to transmit in slot t.

2.2 The MULTI-ARMs BANDIT Model

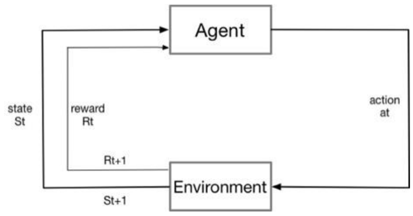

Figure 1. the elements of reinforcement learn

Multi-Arms Bandit(MAB) model is essentially a kind of reinforcement learning model(Fig.1). In the classic multi-armed bandit (MAB) problem, there are N independent arms and a single player. Playing arm i (i = 1, · · ·, N) yields i.i.d random rewards with a distribution parameterized by an unknown θ_i. At each time, the player chooses one arm to play, aiming to maximize the total expected reward in the long run. Under an unknown reward model, the essence of the problem lies in the well-known tradeoff between exploitation and exploration.

Besides the single player MAB, a multi-armed bandit with multiple players is put forward. Support that there are N multi-arms bandits(MAB) and players will get reward with an unknown probability while they pull one arm to take N kinds of action. At each time, a player chooses one arm to play based on its local observation and decision history. Players do not exchange information on their decisions and observations. Collisions occur when multiple players choose the same arm, which cause that all players can get positive or negative rewards with different probabilities.

In order to find the optimal way, exploration and exploitation are considered inevitably in the MAB model. Exploration means that the player tries all the arms as much as possible, which makes the stability of the reward poor, and exploitation means that once the player finds the right arm, he will not continue to try, which makes the overall income poor. In the MAB model, for each player, initial probability of arms is set by softmax function :

$$x(a) = \frac{e\hat{}H(a)}{\sum_{b=1}^{k} e\hat{}H(a)} \tag{1}$$

where H(a) is the priority factor of every arm for player a and k is the maximum number of time slots that nodes can choose. We also set a variable e=1/t where t is running round. Hence, player can take some action, namely choose an arm with the probability p_{max} and p_i

$$p_{max} = \frac{1-e}{n} \tag{2}$$

$$p_i = \frac{e}{(N-n)} \frac{xi}{1 - xmax * n} \tag{3}$$

where n is the number of maximum probability arm.

The expectation that the player choose the arm b at t time can be shown as:

$$E[Rt] = \sum_{b} \pi t(b) q_*(b) \tag{4}$$

where $\pi t(b)$ indicates the probability of choosing arm b at t time. Hence the expectation of the reward is :

$$\frac{\partial E[Rt]}{\partial Ht(a)} = \frac{\partial}{\partial Ht(a)} \left[\sum_{b} \pi t(b)\, q_*(b) \right]$$

$$= \sum_b q_*(b) \frac{\partial \pi t(b)}{\partial Ht(a)} = \sum_b \left(q_*(b) - Xt\right) \frac{\partial \pi t(b)}{\partial Ht(a)} \tag{5}$$

Let the random variable be At, and change Xt and $q_*(b)$ to $\overline{Rt}$, which is the average reward, and Rt. Therefore, the expectation becomes :

$$= E[(Rt - \overline{Rt}) \frac{\partial \pi t(At)}{\partial Ht(a)} / \pi t(At)] \tag{6}$$

according to the softmax function, $\frac{\partial \pi t(At)}{\partial Ht(a)} = \pi t(At)[1_{a=At} - \pi t(a)]$, and we get:

$$= E\{\frac{\left(Rt - \overline{Rt}\right)\pi t(At)[1_{a=At} - \pi t(a)]}{\pi t(At)}\} \tag{7}$$

Hence, the reward function can be shown as:

$$H_{t+1}(a) = H_t(a) + \alpha E\{\left(Rt - \overline{Rt}\right)[1_{a=At} - \pi t(a)]\} \tag{8}$$

where $\overline{Rt}$ is the average of the sum of previous R, and α is a variable that can influence the convergence. In this paper, we set α=0.5 and it seems to be great in the experience. After repeated cycles, player keeps learning the environment information and update his reward, eventually an optimal way will be found.

3.Alogrithm

3.1 Alogrithm Description

In the distributed TDMA system, the node select different time slots with its two-hop neighbors to make the transmission get rid of collision. Before utilizing the algorithm, each node is in the unknown environment and only knows the sum of time slots, i.e. arms, and stores their own Pr, current occupied time slot, and the number of neighbors, where the number of neighbors is collected and updated by receiving ack from other nodes each round. According to the softmax function, every node will get a H factor to calculate the probability. Initially, the probabilities of all arms for every slot are the same.

Each node which can be seen as an independent player follows the MAB model rule every round, and the time slot can be seen as the arm. But what we should notice is that the reward received by taking action is related to other node's action. For example, node u selects slot k in the s state and its neighbors may also select this slot, so collision will be caused which influences the selection of node u in the s+1 state. Hence, the process is like a game that finding the balance among all nodes without a negotiate.

3.1.1 the First Round. The reward calculation algorithm of the first round is somewhat different from the other rounds, so we introduce it firstly.

All nodes store the maximum number of ack packets receiving from other nodes which we define as History, and the History factor is set to 1 initially. In order to avoid falling into a local optimum trap, we definite an exploration factor e as 1/t (t is current cycle). The factor is beneficial to the exploration of the entire arms, which can make the node explore more node in the first few cycles. We stipulate that the reward equals 1 while receiving ack packets of node u is more than the number in the History, and updating the History value ; the reward equals -3 while receiving ack packets is less than the number in the History, and there must be some collisions for node u in this situation. If receiving ack packets is the same as the number in the History, reward equals 0. When node u receives some hello packets from its neighbors, it collects these neighbors Pr and compares with its own Pr. If node u’s Pr is larger than some of these neighbors, the reward still equals 1 while node u does not receive hello packets from the neighbors whose Pr is less than u.

We stipulate that node u will store the probabilities while the number of receiving ack packets equals to the History value during k cycles (k is related to the density of the network), and improve reward to 5. If the node receives ack packets which is more than History at some later moment, changing the probabilities to the previous one and update the History. Follow the above steps, every node iterates in the order of state, action and reward. We set thresholds T1 and T2.

1. If a probability is more than T1, the node converges to the time slot
2. If a probability is less than T2, the probability turns to 0 and averages its probability to other time slots.

Figure 2 shows the relationship between convergence speed and average throughput with different R.

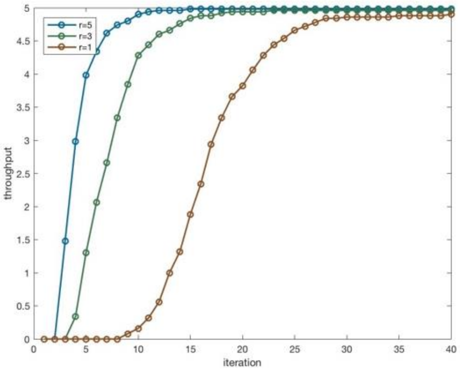

Figure 2. the comparison of M-TDMA with different reward

We can see from Fig. 2 that when the positive reward is set to 5, convergence is fastest, and as the setting reward falling, the convergence speed become dramatically slow. But we should notice that if the reward is set too high, there may more than two nodes' probabilities reach the threshold at the same cycle and make mistakes.

3.1.2 The Following Round. For each node, when they converge, this means that each node occupies one-time slot without collision, and knows the time slots occupied by its one-hop and two-hop neighbors and the Pr of its one-hop neighbors. According to these information, each node knows its unoccupied time slots. We can think that each node has built a table for the next round distribution.

When the node u has occupied a time slot after the first round, it will send hello packet or data packet in the time slot and listen to other nodes' hello packets so that it can ensure if its neighbors converge. If after k cycle neighbors' occupied slots do not change, node u can cognize its unoccupied time slots and starts the next round. In the next round, for each node we consider to make some changes about the reward algorithm. Since node u has known the number and Pr of its neighbors, taking action as mentioned above : sending hello packet to its neighbors, and comparing the number of ack packets with History.

1. If ack=History, which means there is no collision during the time slot, R=3 ;
2. If ack<History, which means there happened some collisions, R=-3 ;

Furthermore, node u will compare its own Pru with Prn receiving from neighbors in the same slot.

1. If Pru<Prn, the node will not choose this time slot in the next cycle.
2. If Pru>Prn, there will be no effect for the node.

Based on the rule, when there is no enough time slot for all nodes, the nodes with less Pr will be not allocated. The less Pr node has, the less slot node occupies. We also set a value to Pr so that if a node occupied enough slot, it will quit playing, which is conducive to the overall throughput and convergence speed. The thresholds are the same and if there are still some time slots remaining unoccupied, the algorithm will continue until the throughput meet our demand.

3.2 Throughput Analysis

If we consider the time slot allocation as a probabilistic problem, it is necessary to estimate the the normalized average node throughput by the proposed time slot allocation algorithm.

We reckon that node u only transmits information packet to its neighbor node v in a time slot in the distributed TDMA. Suppose that there are s time slots in a frame. Therefore, the minimum guaranteed node throughput can be expressed as :

$$T_{min} = \frac{s - kD}{s} \tag{9}$$

where k is the maximum number of collisions between two nodes in the communication rage. s-kD is the minimum number of successful transmissions for each node during one frame. kD is the number of packets which two nodes in the communication rage fail to transmit.

In a round, the probability that node u collides with at least one neighbor is :

$$P_{uc}(l) = \prod_{k=1}^{l} \frac{s^k - k}{s^{k+1} - k}, l \leq D \tag{10}$$

where l is the number of neighbors selecting the same slot with node u.

The probability that other neighbors do not collide with node u is:

$$P_{uc}f(D - l) = \prod_{k=1}^{D-l} \frac{p^{k+1} - p^k - k + l}{p^{k+1} - k + l}, l \leq D \tag{11}$$

Therefore, by jointly considering (10) and (11), the probability that node u collide in a time slot is :

$$P_u = \sum_{l=1}^{D} \binom{D}{l} \left(P_{uc}(l)\right) \left(P_{uc}f(D - l)\right) \tag{12}$$

Based on the above functions, the normalized average node throughput is:

$$\mathrm{T} = \frac{1}{N} \sum_{i=1}^{N} (1 - P_{ui}) \tag{13}$$

3.3 Theoretical Analysis

Theorem 1: The M-TDMA algorithm allocates time slots without collision.

Proof: In the protocol, we rule that a node will occupies a time slot while its probability of the time slot is bigger than threshold T. Based on the algorithm, if its two-hop neighbors send hello in this time slot, they will be received negative reward and choose other time slots with larger probabilities. Therefore, it can ensure that any pair of two neighbor nodes does not select the same time slot.

Theorem 2 : The M-TDMA algorithm can converge.

Proof : Initially, each node is not sure which time slot is occupied, so selection is random. For a node i, once it has been in the broadcast process without collision in a certain period, it will confirm its maximum ack number. Therefore, node i can get accurate and large enough reward to take the next action. If there are k two-hop neighbors and n time slots in the network, the probability that node i broadcasts without collision is: $(\frac{n-1}{n})^{k}$, and the probability of collision-free broadcast at least once in t cycles is:

$$p_{no\_collion} = 1 - [\frac{n^k - (n-1)^k}{n^k}]^t \tag{14}$$

We can see that if n and k is large enough, $p_{no\_collion}$ approaches 1. And the probability of node i selecting the same time slot with a two-hop neighbor in a certain period is :

$$\frac{1}{n} \sum_{i=1}^{k} \genfrac{}{}{0pt}{}{i}{d} (pi)^i \, (p'i)^{k-i} < \frac{1}{n} \sum_{i=1}^{k} \genfrac{}{}{0pt}{}{i}{d} (\frac{1}{n})^i \, (\frac{1}{n-1})^{k-i} \tag{15}$$

which is similar to $\frac{1}{n^3}$. If without convergence, the probability is $\frac{1}{n^{3k}}$ in the k period.

Therefor, as the density of the network increasing, the unconverged probability of node i is similar to 1. In the meanwhile, the convergence is like to flood. It starts from the node with least neighbors generally, because the k is less and get its acknumber faster so that it is more likely to converge. And after this, other nodes can occupy time slots with less cycles. There is no interaction process, so the sooner as many nodes converge, the faster the overall convergence can be achieved.

Theorem 3 : Priority factor(Pr) can coordinate the network time slot allocation.

Proof : If there are too many edge nodes, the convergence speed will be slowed down. It is necessary to balance the throughput and convergence speed, and the Pr makes a compromise to coordinate the throughput and convergence speed. The Pr is related to the number of two-hop neighbors and occupied time slot by itself. When the Pr of edge nodes reaches a certain value, they quit to play. Also, in the case where the number of time slots is less enough for the number of nodes, the node with a larger Pr can occupies the slot preferentially, and the node with least Pr will back off. The outcome is obvious that more time slots will be occupied by the center nodes averagely, which has more neighbors, i.e. more Pr, this can benefit the whole network business.

4.Evaluation

In the paper, we evaluate the performance of the proposed distributed tdma time slot allocation algorithm based on MAB model, and get the simulation result and model performance. As is shown in Fig.3, the topology of the network consists of 200 nodes which are randomly distributed in a 300 × 300m plane, the line represents there is a link between the two nodes in the presented network. Firstly, we compare the M-TDMA algorithm with DRAND and E-T DRAND. In order to reach the standard, the number of time slots is equal, and is approximated to the minimum number of slots, so that comparing the convergence speed.

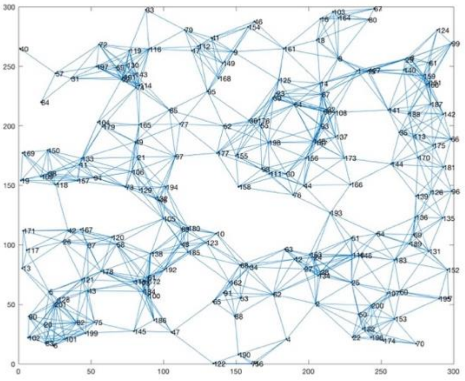

Figure 3. the topology structure of simulation

Fig. 4 shows the comparison of the average number of cycles where three algorithms success to allocate the slot based on different maximum number of neighbors. We can observe that when the number of neighbors increases, the collisions during allocation will increase, and their trends are similar. As D increases, the number of cycles required for successful time slot allocation increases. The number of cycles required by the DRAND algorithm is the most, and E-T DRAND is an improved algorithm, which the number of cycles required is less. The cycles they need is observably decrease when D is about 30, and then continue to increase. Observably, compared with the other two protocols, the M-TDMA needs less cycles to converge, and increase in a linearly similar way. Although when D is 30, the M-TDMA is a bit larger than E-T DRAND, its performance is better than E-T DRAND and DRAND on the whole.

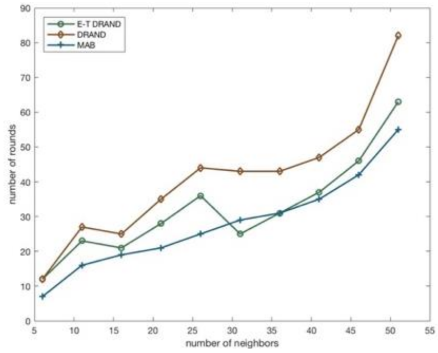

Figure 4. the average number of rounds for successful allocating time slot in different size of neighbor nodes

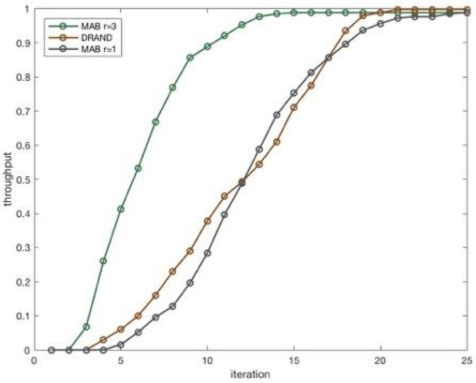

Figure 5. the achieved normalized average node throughput in 25 iterations

Fig. 5 performs a statistical analysis on the the achieved normalized average node throughput of protocols within 25 iterations, which the grey line represents the M-TDMA protocol with r=1 initially and the green line represents the MAB protocol with r=3 initially. Through the three curves in the picture, we can observe that during the 15 iterations, the M-TDMA protocol with r=3 raises fastest and its throughput can reach almost 1, i.e. the optimal throughput. The throughput of M-TDMA protocol with r=1 can only reach 0.75, and the DRAND protocol is the worst, which only reaches 0.7. After th 15th iteration, M-TDMA with r=1 and DRAND lead to rise alternately, and in the 20th iteration, the throughput of both of them approaches the maximum. We can observe that when setting a suitable reward, the M-TDMA with a great r can achieve the approximate optimal throughput at a fast rate., and even if the value r is not suitable initially, the throughput of M-TDMA is similar with DRAND. Hence, in the case of high-speed dynamic topology, M-TDMA is better than DRAND.

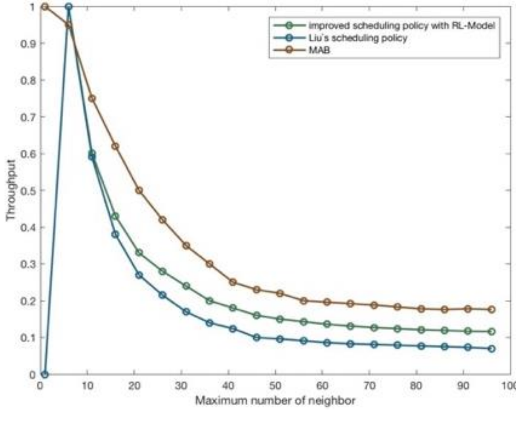

Figure 6. The achieved normalized average node throughput versus the maximum node degree.

Fig. 6 shows that the throughput comparison of three protocols which utilize reinforcement learning to allocate the time slot as maximum number of neighbor increasing. When there are more neighbors, the network topology will be more complicated, and the probability of collision raise concomitantly. Hence the throughput of three protocols all drops and stabilizes eventually. However, the performance of MAB protocol is better than the other two apparently. This finding can be explained as follows : the multiple rounds of iterative mechanism of MAB, redundant time slots can be utilized better than the other two protocols, so that the M-TDMA can avoid collision and achieves a larger throughput.

5.Conclusion

For swarm UAVs, dynamic topology represents a large amount of challenges to the TDMA schedule scheme, and the conventional algorithm can not solve perfectly. In this paper, we propose the M-TDMA protocol based on the MAB model, which learns from environment by local adaptive between nodes, avoids collision and makes full use of slots through multiple rounds of allocation. Also, we utilize the Pr to accelerate the converge. The experimental results show that the algorithm provides a better throughput and significantly improves the convergence speed.

References

[1] Nisioti E, Thomos N. Fast reinforcement learning for decentralized MAC optimization[J]. *arXiv preprint arXiv*:1805.06912, (2018).

[2] Omar H A, Zhuang W, Li L. VeMAC: A TDMA-based MAC protocol for reliable broadcast in VANETs[**J**]. *IEEE transactions on mobile computing*, (2013).

[3] Alinaghian M, Ghazanfari M, Norouzi N, et al. A novel model for the time dependent competitive vehicle routing problem: Modified random topology particle swarm optimization[J]. *Networks and Spatial Economics*, (2017).

[4] Rhee I, Warrier A, Min J, et al. DRAND: Distributed randomized TDMA scheduling for wireless ad hoc networks[J]. *IEEE Transactions on Mobile Computing*, (2009).

[5] Li Y, Zhang X, Zeng J, et al. A distributed TDMA scheduling algorithm based on energy-topology factor in Internet of Things[J]. *IEEE Access*, (2017).

[6] Chlamtac and A. Farago, ''Making transmission schedules immune to topology changes in multi-hop packet radio networks,'' *IEEE/ACM Trans. Netw.*, **vol. 2**, no. 1, pp. 23–29, Feb. (1994).

[7] J.N.TsitsiklisandB.VanRoy,''Ananalysisoftemporal-differencelearn- ing with function approximation,'' *IEEE Trans. Autom. Control*, **vol. 42**, no. 5, pp. 674–690, May (1997).

[8] Ju J, Kim D H, Bi L, et al. Four-color DNA sequencing by synthesis using cleavable fluorescent nucleotide reversible terminators[J]. *Proceedings of the National Academy of Sciences*, (2006).

[9] Liu Y, Li V O K, Leung K C, et al. Topology-transparent scheduling in mobile ad hoc networks with multiple packet reception capability[J]. *IEEE Transactions on Wireless Communications*, (2014).

[10] Y. Liu, V. O. K. Li, K.-C. Leung, and L. Zhang, ''Performance improve- ment of topology-transparent broadcast scheduling in mobile ad hoc networks,'' *IEEE Trans. Veh. Technol.*, **vol. 63**, no. 9, pp. 4594–4605, Nov. (2014).

[11] Liu K, Zhao Q. Distributed learning in multi-armed bandit with multiple players[J]. *IEEE Transactions on Signal Processing*, (2010).

[12] Liu Z, Elhanany I. RL-MAC: A QoS-aware reinforcement learning based MAC protocol for wireless sensor networks[C]//*Networking, Sensing and Control, 2006. ICNSC'06. Proceedings of the 2006 IEEE International Conference on. IEEE*, (2006).

[13] Y. Liu, V. O. K. Li, K.-C. Leung, and L. Zhang, ''Topology-transparent scheduling in mobile ad hoc networks with multiple packet recep- tion capability,'' *IEEE Trans. Wireless Commun.*, **vol. 13**, no. 11, pp. 5940–5953, Nov. (2014).