Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 111 (2017) 268–276

Procedia Computer Science

www.elsevier.com/locate/procedia

8th International Conference on Advances in Information Technology, IAIT2016, 19-22 December 2016, Macau, China

A novel broadcasting MAC algorithm for ad hoc networks

Chen Liang[a,b,*], Yu Zhang[a,c], Jian Song[a,c]

[a]Tsinghua National Laboratory for Information Science and Technology (TNList), Tsinghua University Department of Electronics Engineering, Tsinghua University, Beijing 100084, China
[b]Science and Technology on Information Transmission and Dissemination in Communication Networks Laboratory
[c]Shenzhen City Key Laboratory of Digital TV System, Shenzhen 518057, China

Abstract

Wireless ad hoc networks with broadcasting are widely applied in IoT and emergency situations as the network can be set up without traditional infrastructures. In this paper, we propose a TDMA-based MAC algorithm which supports data transmission by broadcasting. The algorithm is similar with the mechanism in NbIA algorithm [1,2] in neighbour discovery, but differs a lot in collision detection and slot allocation. Our algorithm can dynamically allocate slots according to the demand of nodes and ensures every node get at least one slot a cycle. Based on spatial reuse, idle slots might be found and be used as extra slots of nodes which can promote throughput of our algorithm. After the network converges, only a few nodes need to adjust their slots to adapt to topology changes, while most of the nodes maintain their slots to transmit data. The performance of our algorithm and NbIA algorithm are compared with OPNET simulation. The results show that our algorithm outperforms NbIA especially in convergence time and average throughput of nodes.

Peer-review under responsibility of the organizing committee of the 8th International Conference on Advances in Information Technology

Keywords: Ad hoc network; MAC algorithm; Broadcast; Slot assignment

1. Introduction

Wireless ad hoc networks are under extensive research because of its flexibility that nodes can transmit data without infrastructures and wide applications in the Internet of Things (IoT). An ad hoc network consists of a

* Corresponding author. Tel.: +1-500-129-6065.
E-mail address: liangc14@mails.tsinghua.edu.cn

1877-0509 © 2017 The Authors. Published by Elsevier B.V.
Peer-review under responsibility of the organizing committee of the 8th International Conference on Advances in Information Technology
10.1016/j.procs.2017.06.063

collection of nodes, each of which communicates over wireless channels. With the movement of the nodes, the network has to be self-organized and adjust to the unpredictable topology changes. Due to the advantages such as flexibility, low cost and ease of deployment, ad hoc networks are very suitable to be applied for massive moving sensor nodes which need to be internet-connected to connect "things" [3]. Networks can be easily set up in a fully distributed way and provide efficient connectivity. Moreover, broadcast is highly needed in the transmission of IoT to broadcast related information to all the nearby sensors [4]. For instance, in Vehicle Ad Hoc Networks, status messages are periodically broadcasted to enhance public safety [5]. Therefore, in this paper, we propose a novel MAC broadcasting algorithm in ad hoc networks to pursue higher throughput and faster network convergence.

In general, distributed MAC algorithms are usually applied in ad hoc networks to guarantee that the nodes could access to the network in a fast and self-organized manner. Such algorithms are especially efficient in emergency situations. Under different application scenarios, different MAC algorithms have been proposed to provide solutions of neighbour coordination and slot assignment. In most distributed MAC algorithms, TDMA is widely used to provide a more reliable and simple communication resource allocation scheme. In a point-to-point algorithm, communication slots might be allocated to a pair of nodes by time slot reservation mechanism, so that data can be transmitted between these nodes. In such algorithms, each pair of nodes would negotiate and finally get a reasonable slot assignment solution. However, there might be a problem with two or more destination nodes at the same time, which could happen in practical when higher authorities give orders to the lower ones, or in other IoT scenarios mentioned above. Thus, an algorithm that supports broadcast transmission should be adopted in these scenarios.

However, present broadcasting algorithms have some shortcomings such as low channel utilization efficiency. If one node occupies a slot broadcasting at the time, all of the neighbouring nodes in its two-hop range need to be silent, in order not to interfere with the transmission. Therefore, it is more difficult to assign slots when the negotiation takes place among all the nodes in its two-hop range. Low success ratio of the slot assignment would lead to low channel utilization efficiency. In addition, for the same reason, the network cannot converge fast and may be insensitive to topology changes.

In this paper, we propose a TDMA-based fully distributed MAC algorithm for mobile ad hoc networks which supports data transmission by broadcasting. In our algorithm, the frame includes two phases, i.e., the beacon phase and the data phase. Both phases are divided into 16 slots. During the beacon phase, each node contends with others for one slot, which is called the main-slot, and broadcasts its beacon packet in this slot. Its neighbouring nodes will receive and reserve the beacon packet. The node gets to know the situation of conflicts by collecting feedback information through the beacon packet from its neighbours. Nodes with serious conflicts will switch the slots prior to the nodes with slighter conflicts. After solving conflicts, all the nodes get their main-slot each and use the corresponding slot in the data phase to transmit data.

Concretely, we make the following contributions:

- We propose a new MAC algorithm that supports packet transmission with broadcasting. Novel collision detection and slot allocation schemes are designed to ensure every node get at least one slot (so called main-slot) in a cycle.
- Fast convergence is guaranteed in our proposed algorithm. Each node contends for its one main-slot in only one cycle, so that a lot of idle slots are left to promote success rate of slot allocation which ensures a fast convergence.
- High throughput is guaranteed. All the nodes contend for extra slots, which is called standby-slot, based on topology information collected in their main-slots, and the channel utilization efficiency is improved. Nodes take these extra slots as standbys of main-slots to recover transmission from conflict.

The rest of this paper is organized as follows. In Section II, we summarize related work about MAC algorithms in ad hoc networks. We describe our proposed protocol in Section III. In Section IV, simulation results of the protocol are demonstrated in comparison with NbIA algorithm. Finally, we give our conclusions in Section V.

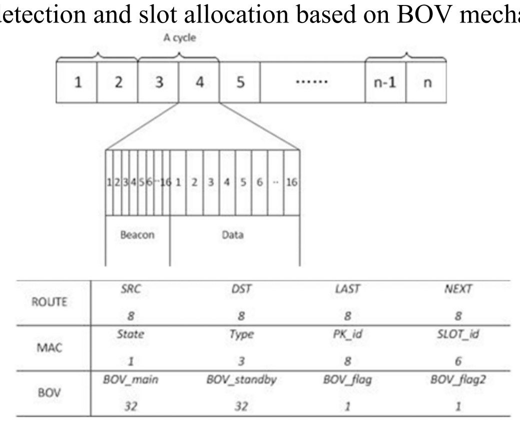
2. Related Work

The MAC protocol is an important part of ad hoc networks. To guarantee both efficiency and robustness, slot reservation protocols are proposed, which means that each node needs to negotiate its slot before using it to transmit. Existing slot reservation protocols can be grouped into two categories: centralized-liked and fully distributed.

Typical systems with centralized protocols [6] have central or server node to control slot assignment after it obtains the topology of the whole network. The network would finally converge to an allocation solution closer to the global optimization solution in a centralized protocol. But the overhead to collect topology information is enormous, especially when the structure of the network is complex and changing with the mobility of the nodes. SVC [7] is also a centralized-liked protocol which implements advanced rules to guide the network behaviour with no server node. As the degree of the network grows greater, the efficiency of the rules decreases and the performance of the protocol degrades sharply. Thus the network robustness and degree of the network are barriers to centralized-liked protocols.

There are some distributed broadcasting algorithms executed in MAC layer based on contention. The IEEE 802.11 broadcast protocol [8] uses carrier sense multiple access with collision avoidance(CSMA/CA), and some optimized algorithms based on the mechanism have been proposed, such as BSMA [9]. But there is some problem with collision. FPRP [10] is essentially a fully distributed broadcasting algorithm based on contention, avoiding collecting topology information. The frame of FPRP is divided into reservation frame (RF) and information frame (IF). Nodes contend for slots in RF and the corresponding slots are used for data transmission in IF. This algorithm updates the slot assignment solution every cycle to support dynamic topology. But contention-based algorithm cannot guarantee the access of all the nodes, which makes it not suitable in practical applications because of the lack of QoS guarantee (e.g., the end-to-end delay).

NbIA algorithm has been proposed for several years. This broadcasting algorithm ensures all the nodes could get one slot per cycle and the idea of neighbour occupancy vector was originally put forward in NbIA algorithm. There are three slot allocation mechanisms called indirect acknowledgement(ACK), Beacon Occupancy Vector (BOV) and Exchange Mechanism (EM) to detect collision. And each cycle has N beacon slots for node to contend for slot and N data slots to transmit according to beacon slot. Firstly in ACK phase, every node is in state I, they send beacon packet to contend for slot and acknowledge its neighbours' beacon slot indirectly. Node who get enough acknowledgement enters state II and starts to execute BOV mechanism. Node in state II means a few neighbours hear it but not all neighbours. Then it use a vector called BOV to collect the proportion of success neighbour and a changing BOV-threshold to judge if it needs to change his slot. There will be no conflict until the BOV-threshold is 1. Exchange Mechanism (EM) is a mechanism infrequently and in parallel with BOV. Where a node in state II will temporarily swap its slot with a neighbour to reveal indiscoverable collision which waste a lot of convergence time at the same time. NbIA is a fully distributed broadcasting algorithm which does not depend on degree of the network, so that it is robustly applicable to practical ad hoc networks. However, it could cause collision constantly by using EM method or, in a word, the network cannot converge fast. What's more, channel utilization efficiency of the algorithm is low as all the nodes occupy only one slot in a cycle. Therefore, we give new schemes on collision detection and slot allocation based on BOV mechanism and improve the performance a lot.



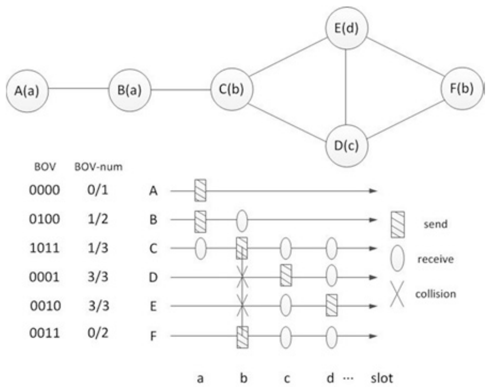


Fig. 1. (a) Frame structure; (b) The structure and corresponding numbers of bits in a beacon packet; (c) An example topology

3. Protocol

In this section, we first introduce the assumptions for the networks and then give an introduction on the frame structure. The rationality of the algorithm is analysed and the proposed protocol is explained in detail, including the process of BOV and standby-slot selection.

3.1. Assumptions and Conditions

In order to give a detailed description of our protocol, several assumptions are listed as follows.

- The network operates without any central node or infrastructure and each node is completely equal to others, not only in function, but also in hardware configuration.
- A node with a steady main-slot could access network successfully.
- The error correcting capability of the system is poor. The system does not support data transmission and receiving simultaneously.

3.2. Frame Structure

Every frame consists of two components where a beacon phase is followed by a data phase. Both the beacon and data phase are divided into 16 time slots, which means one frame supports 16 neighbouring nodes in a single-hop range. If there are 30 nodes in one single-hop range, serious conflict will exist for 16-slot frame and then we could use flag to combine every pair of frames as one which is shown in Fig. 1(a).

In the proposed MAC algorithm, we allocate the time slots by transmitting beacon packet in the beacon phase. Each node would choose only one main-slot in beacon phase and use the corresponding slot in data phase to broadcast data, or update routing information infrequently. In our algorithm, a node updates its neighbours' information through their beacon packets every cycle. Beacon packet is shown in Fig. 1(b). ROUTE domain contains the source and destination of the packet. MAC domain declares ID information. BOV domain contains information of its main-slot and standby-slots, and BOV-flag indicates the convergence condition in two hop range.

3.3. Algorithm Description and Rationality Analysis

To describe our algorithm clearly, we consider a network with the topology shown in Fig. 1(c). Where A-F are ID tags of the nodes and *a*-*d* are slot ID tags. The slots will arrive in the order *a*-*d*. Note that $\mathrm{N_A}$ represents the set of nodes in the one-hop range of node A. Therefore, $\mathrm{N_A} = \{B\}$ means that node B is one-hop neighbours of node A. For the 6 nodes we assume that both A and B select slot *a*, both C and F select slot *b*, and D and E select *c* and *d* as their randomly-selected initial main-slots, respectively. The algorithm is summarized as the following steps and more detailed description will be given in the following sections.

- BOV: In beacon phase, each node sends a beacon packet at its main-slot and the nodes who receive the packet will update their BOV information. As the result, at slot a, node A and B send beacon packet but only node C can hear node B and update the first bit of node C's BOV to '1'. Then node C and F start to send at slot b. Since one node cannot receive two packet simultaneously, node D and E cannot hear both of them. Node F will be aware of the situation when it receives node D's packet in slot c and finds a '0' in the third bit of D's BOV. Nodes like C may finally realize that, among all the three neighbours, only one neighbour B has found it while node D and E can not hear node C. Therefore, we define 1/3 as BOV-number of node C. Node C compares 1/3 with a changing threshold to decide if it needs to switch slots immediately when BOV-number goes lower than the threshold.
- Standby-slot selection: When all the neighbours report their BOV-number as 1, each node will try to get extra standby-slots based on BOV information. E.g., if node A get to know node B, its unique neighbour, is busy in slot b, node A would skip slot b and try slot c first when get extra standby-slots is needed.

Each node in the network follows the algorithm above to contend for slots. We define convergence when all the nodes get main-slot to join the network. We can use probability theory to analyse the rationality of the algorithm.

In our algorithm, conflicts occur only between neighbours in the two-hop range. As a result, the worst case happens when there are many node clusters in the two-hop range. We assume our multi-hop network contains *K* slots in total, and there are *M* nodes in the most crowded two-hop range with node A in the center. Then the only factor that would influence getting a high BOV-number is whether there are more than one nodes in the two-hop range within the same slot. Let *m* represent the number of nodes select slot *a*. Therefore, the probability that no nodes other than node A select slot *a* as the initial main-slot is

$$P(m=0)=\frac{(K-1)^{M-1}}{K^{M-1}} \tag{1}$$

Similarly, the probability that only one neighbouring node, called X, selects the same slot is

$$P(m=1)=\frac{(M-1)*(K-1)^{M-2}}{K^{M-1}}\approx 0.34 \tag{2}$$

when *K*=32 and we assume *M*=20 first. In summary, we have

$$P(m=0)\approx 0.55,\ \ P(m=1)\approx 0.34 \tag{3}$$

When *m*=0, node A's BOV-number is 1. When *m*=1, the BOV-number depends on the size of N_A and N_X . Let $|N_A|$ denote the number of nodes in this set and $\complement N_A$ denote the set of all the other nodes out of N_A. The nodes in $N_A \cap \complement N_X$ may hear of node A, while the nodes in $N_A \cap N_X$ cannot. In this case, BOV-number of node A can be determined by

$$BOV\_number_A=\frac{|N_A \cap \complement N_X|}{|N_A|} \tag{4}$$

The BOV-number of node A would be compared with a random-selected threshold to judge if it needs to switch slots. According to the randomness of the threshold, we consider that 50% * 0.34 = 17% of situations are treated as serious collisions and node A would switch slots at once, while in the other 17% of these situations, node A keeps the slot. For the rest P(m > 1)=0.11, node A has at least two conflicting neighbours which may cause serious conflicts. Thus, the probability of node A to switch slots immediately is 0.11+0.17=0.28, which means 28% of nodes need to switch their initial slots. Part of these nodes may create new collision and switch slots together with other 17% of nodes. In summary, at any time, only about 30% of nodes would adjust slots which reduces the chance of new collisions and provides our algorithm a fast convergence.

Then when *M*=25 and *M*=30 , the node that need to adjust slots will be 36% and 41% by calculation. This is the worst case as when *M*>32 we may combine two cycle as one to use. According to simulation result, it costs less than 20 cycles to converge in the worst case.

3.4. BOV Threshold Adjustment

Most of the collisions should be detected in BOV threshold Adjustment phase. The concept of BOV in NbIA algorithm is used as reference, but a new detection scheme and BOV-threshold is proposed in this paper. Let us assume node A receives a packet from node B successfully in the third slot, and node C sends a packet in the fourth slot, but node A cannot hear from node C because of collision. In this situation, node A maintains a BOV like xx10, each bit corresponding to one slot. The vector is used for BOV-main domain in the beacon packet and then reset to all zeros. After node A sends out its packet, node B finds that the third bit is '1' and consequently node B realizes A is a success neighbour. However, node C comes to know that A is a failure neighbour because the fourth bit is '0'.

All the nodes update the number of their success neighbours every cycle. If node A has three success neighbours in all five neighbours, the BOV-number is 3/5. The BOV-threshold, selected randomly every two cycles, is used to determine whether one node needs switch the main-slot immediately. If the BOV-threshold is larger than the BOV-number, it implies there is a severe collision and nodes have to switch their main-slots. Moreover, to ensure fast convergence, only one node of two collision nodes would switch its slot.BOV-threshold is selected randomly within the range of 0.19-0.84 in NbIA algorithm, which leaves the following problems to solve:

- Collision nodes may continuously select small BOV-threshold which influences the speed of convergence.
- There are still collisions when BOV-threshold is larger than 0.84 and smaller than 1.
- Just as shown in Fig. 1(c), a situation called Non Isolated Deadlock happens between node C, D and F. For node D, it may not get enough ACK as it can only be heard by node E. But switching slot has no benefit for node D to access the network because the collision is in fact between C and F.

Instead of Exchange Mechanism in NbIA algorithm, we adjust the BOV-threshold according to network situation to improve the speed of convergence. For each node, BOV-threshold is randomly updated within 0.22 to 0.79 every two cycles which is selected according to simulation results. It is called a BOV-stop when the BOV-number remains unchanged for consecutive six to ten cycles. When a BOV-stop occurs, the next BOV-threshold will be 0.1 larger compared to the current one, after that, we promote the lower limit of BOV-threshold from 0.22 to the current BOV-threshold and new BOV-threshold will be randomly generated from new lower limit to 0.79. If BOV-threshold is bigger than 0.79, BOV-stop will still happen and BOV-threshold will finally reach to 1. As a result, the first two problems listed above can now be solved. As for problem 3, in our algorithm, node D does not need to be acknowledged. And then node C and F will switch their slots according to their BOV-num so that node D can access the network successfully.

3.5. Standby Slot Selection

After all the collisions are fixed, each node tends to get several standby-slots to improve resource utilization efficiency in this phase. When a node converges, its BOV-number should be 1. The parameter to declare the convergence of the node itself is BOV-flag as shown in Fig. 1(b). If all the BOV-flags of its one-hop neighbours are 1, BOV-flag2 would be set to 1. Similarly, node may use BOV-flag2 of its neighbours to judge if all the nodes in its two-hop neighbours have converged. And then, standby-slots have a chance to be generated.

Based upon simulation results, adjacent nodes always have similar two-hop convergence time, which may cause conflicts when nodes are contending for standby-slots. Therefore, priorities of standby-slot contention could be set according to traffic demands. When a node promotes its priority because of bursty demands, the priority level in its beacon will remind its neighbours of releasing standby-slots to ensure the quality of service for higher priority nodes.

To generate standby-slots, a node waits for a random period of time, selects an empty slot in its two-hop slot table and sends a probe packet in beacon phase. All its neighbours would receive the probe packet and record it in the BOV-standby domain, unless another probe request comes at the same time. If all the neighbours allow the probe packet, then the slot becomes the node's standby-slot. Otherwise, the node realizes there is a conflict. After several cycles waiting, it would contend for an empty slot again.

According to our algorithm, a node will get standby-slots successfully unless other nodes apply for the same slot in the exactly same cycle. After that, the node occupies the slot until it releases the slot on its own initiative or is told by other higher priority neighbours. The standby-slot is recorded in BOV-standby domain, just as the main-slot recorded in BOV-main domain. Whenever a node needs to switch slots, it collects BOV of all its neighbours so that it grasps the situation in the two-hop range, which is the max range that influences the node. It then selects an empty slot in its two-hop slot table. When there is no empty slot due to temporary congestion, the node should choose an empty slot in its one-hop slot table. This leads to a potential solution although the node robs the slot of an one-hop neighbour. After that, the neighbour may switch the slot with an empty slot in its slot table, so that the whole network converges gradually. Non Isolated Deadlock will be exposed by selecting standby-slots. Several cycles later, all slots are occupied as standby-slots which ensures high utilization efficiency based on spatial reuse.

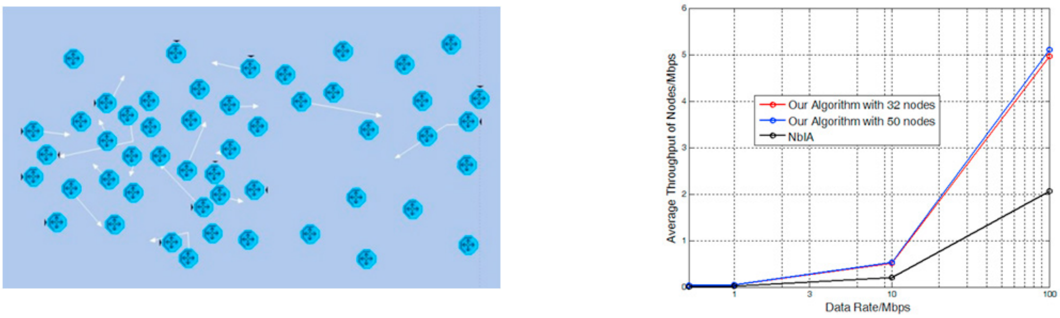


Fig. 2. (a) An example 50 mobile nodes topology;(b) The average throughput of nodes in different data rate and node number of two algorithms.

4. Simulation results

We use OPNET software to simulate both NbIA and our algorithm in some randomly generated scenarios.

The frame structure is set the same as Fig. 1(a) and one second consist of 4 cycles. The length of our beacon packet is increased to 158 bits to match NbIA algorithm, so that the typical convergence time and average throughput of both two algorithms can be compared. In addition, we set the transceivers' rates to be 100Mbps.

We assume our network contains 50 nodes with the same priority randomly deployed in a 10km*10km range which is shown in Fig.2(a). Each node has enough number of packets in their buffers and uses the main-slot and two possible standby-slots to send data through multi-hop routing in a broadcasting way. To reveal the performance of our algorithm, all the nodes will stay put in the first 10 seconds of network organizing. After 10s, half of the nodes will start moving with a random path with a speed lower than 100km/h. Then at 30s, ten nodes will leave the network and others will keep moving until the end of our simulation.

The node whose main-slot is in conflict with others will be treated as a collision node. The time-varying number of collision nodes is shown in Fig. 3(a). Based on the results, the convergence time of our algorithm is about 2s which means there are no conflicts in the whole network at the time, while NbIA need 8s to access network, of which 4s are used by ACK and BOV phase, and the other 4s are needed because of the EM mechanism, with which a node swaps its slot with neighbours for cycles, which is inefficient in most scenarios.

After 10s, moving nodes joining in other local networks might cause conflicts. Two conflicting nodes may be aware of the conflict within one cycle according to our algorithm, then the invaded node may simply exchange its main-slot and standby-slot to fix the problem or select another free main-slot based on its two-hop slot table. In summary, our algorithm can fix the impact of mobile nodes in about two cycles, while NbIA does not have standby-slots and causes less conflicts.

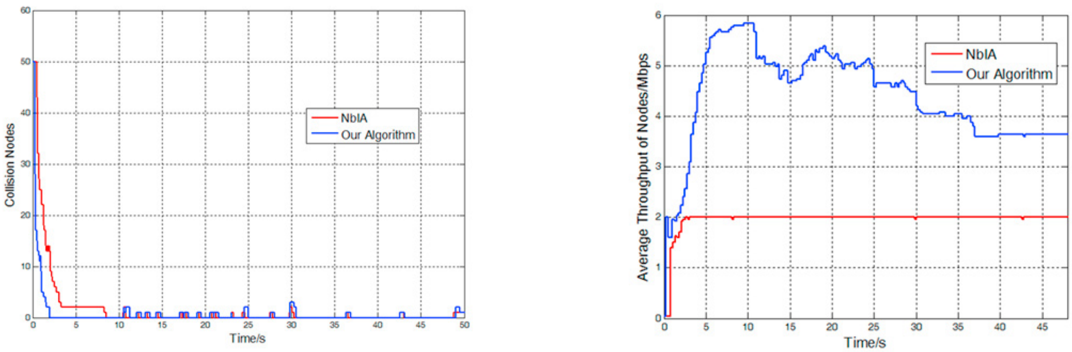


Fig. 3. The time-varying number of (a) collision nodes (b) average throughput of nodes of two algorithms.

The average throughput of every single node is shown in Fig. 3(b). After setting up the network, nodes of NbIA will only occupy one slot each in every cycle, so the throughput is steady after 4s. For the nodes in our proposed algorithm, they will contend for 2 possible standby-slots after network organizing shown in the first 5s in Fig. 3(b) and tend to be steady from 5 to 10 seconds. After that, the changing topology forces some nodes to release and reapply standby-slots, which has impact on the throughput of the network.

We simulate our algorithm with different data rates and different node numbers in several random scenarios and the result is shown in Fig.2(b). For our algorithm, node number has little influence on the network, because each node are only sensitive to its two-hop range instead of all the other nodes in the system. Based on the simulation results, our algorithm has shorter access time and higher throughput. A similar recovery time is guaranteed which ensures the basic performance of an ad hoc network. Standby-slots in our algorithm allow node to send more data without causing serious conflicts even with mobility. Moreover, based on the results shown in Fig. 2(b), our algorithm outperforms NbIA algorithm in average network throughput by approximately 110%. According to the collision node numbers of the two algorithms, there will be no collision nodes at all after 2 seconds for our proposed algorithm, while about 2 more seconds are needed to execute corresponding mechanism in NbIA algorithm.

5. Conclusion

We propose a fully distributed TDMA-based MAC protocol for mobile wireless ad hoc networks in this paper. The protocol allows nodes to access network efficiently and transmit with other nodes by broadcasting. With our protocol, nodes can get one main-slot in a cycle to ensure stable transmission and get extra standby-slots depending on their traffic demands. After setting up the network, time slots are guaranteed to be made full use of and the throughput of network is improved. Conflicts could also be fixed immediately when nodes join in. The algorithm is robustly applicable to the practical ad hoc networks. Our simulation results reveal that our algorithm achieves a better performance than NbIA algorithm in terms of throughput and access time. The slot utilization efficiency of our proposed algorithm is around 110% higher than the NbIA algorithm. Our future work includes optimizing the slot allocation of high priority nodes.

Acknowledgements

This work was supported by Science Fund for Creative Research Groups of NSFC(61321061), National Basic Research Program of China (973 Program)2012CB316000 and a grant (ITD-U15005/KX152600014) from Science and Technology on Information Transmission and Dissemination in Communication Networks Laboratory.

References

1. N. C. Pun, "TDMA channel access scheduling with neighbor indirect acknowledgment algorithm (NbIA) for ad-hoc networks," U.S. Patent 7 768 992, August 3, 2010.
2.W. M. Rudnick, N. C. Pun, D. Clark D and C. J. Datz, "Mobile ad hoc network with dynamic TDMA slot assignments and related methods," U.S. Patent 8 942 197, Jan. 27, 2015.
3. B. Oladayo, Z. Sherali and B. Mohamad, "Network layer inter-operation of Device-to-Device communication technologies in Internet of Things (IoT)," *Ad Hoc Networks*, available online 23 June 2016, doi:10.1016/j.adhoc.2016.06.010.
4. T. A. Naqishbandi and S. C. Imthyaz, "A Resilient Strategy against Energy Attacks in Ad-Hoc WSN and Future IoT", *International Journal of Advanced Research in Computer Science and Software Engineering*, vol. 4, no. 2, pp. 766-773, Feb. 2014.
5. L. Zhang, Y. Liu, Z. Wang and Y. Du, "Cross-Layer Safety-Critical Broadcast Service Architecture Integrating VANETs with 3G Networks in IoT Environments", *China Communications*, vol. 8, no. 8, pp. 13-24, 2011.
6. S. T. Cheng, M. H. Tao, and C. Y. Wang, "Adaptive channel switching for centralized MAC protocols in multihop wireless networks," *IEEE Trans. On Commun.*, vol. 58, no. 1, pp. 228-234, Jan. 2010.
7. J. Yeo, H. Lee, S. Kim, "An efficient broadcast scheduling algorithm for TDMA ad-hoc networks," *Computers and Operations Research*, vol. 29, no.13, pp. 1793-1806, Feb. 2002.
8. IEEE 802.11 Working group,Wireless LAN Medium Access Control(MAC) and Physical layer(PHY)Specifications,Sept. 1999 IEEE Specification 802.11

9. K. Tang and M. Gerla, "Random Access MAC for Efficient Broadcast Support in Ad Hoc Networks," in *Proc. IEEE WCNC*, 2000, pp. 454 - 459.
10. C. Zhu, M. S. Corson, "A five-phase reservation protocol (FPRP) for mobile ad hoc networks," in *Proc. IEEE INFOCOM*, 1998, pp. 322-331.