A Cross-layer Neighbour Discovery Algorithm in Ad hoc Networks based on Hexagonal Clustering and GPS

Qingcai Wang¹*, Xiandeng He¹ and Nan Chen¹

¹State Key Laboratory of Integrated Services Networks, Xidian University, Xi’an, Shaanxi, China
*qcwang_xd@163.com

Abstract. Neighbour discovery is an important part of ad hoc networks. In a typical ad hoc routing protocol, each node broadcasts hello messages at a fixed time interval to update the latest information of neighbour nodes and tracks neighbour relationship between the nodes. However, the higher density of network nodes, the more collisions happen between the hello messages, the more bandwidths are wasted and the worse network congestion produces. So we design a cross-layer neighbour discovery algorithm, ND_HC, to actualize the neighbour discovery in MAC layer. In this algorithm, the hello messages are produced in MAC layer, and are sent in TDMA method with the help of hexagonal clustering and GPS, which reduce packets collision probability and improve throughput in the network. The NS-2’s simulation results show that compared to the traditional neighbour discovery algorithm in IEEE 802.11, the proposed algorithm improves the efficiency of neighbour discovery obviously.

1 Introduction

Ad hoc networks are multi-hop wireless networks, which have the following characteristics. Firstly, nodes are free to move randomly and communicate with each other without any fixed centralized control or base stations [1]. Secondly, nodes that are within each other's radio range can communicate directly, while distant nodes which are beyond one-hop need the help of the routing protocol to relay the packets efficiently through intermediate nodes [2]. Thirdly, each node has equal status which means that there is no problem of network breakdown caused by failure of fixed central node in centralized networks.

Due to the mobility of nodes in ad hoc, local connections change frequently so that the network topology changes constantly. In a typical ad hoc routing protocol [3-4], such as proactive protocol (e.g. Optimized Linked State Routing, OLSR) and reactive protocol (e.g. Ad hoc on-demand Distance Vector, AODV), each node broadcasts short beacon messages (also called hello messages) at a fixed time intervals to collect one-hop neighbour information, and uses them to maintain the up-to-date local connections and track neighbour relationship between the nodes.

The mobility of nodes makes it challenging to design an efficient MAC protocol with low delay and high reliability. The typical MAC protocols based on TDMA and IEEE 802.11 are studied in [5-6]. Although TDMA-based MAC protocol provides stable delay bounds by resource reservation, it is sensitive to topology changes, which may cause waste of channel resources. On the other hand, IEEE 802.11 protocol based on CSMA performs well in low density network, but may cause network congestion in a high density network due to the increased probability of message collision. Therefore, the IEEE 802.11 protocol can not guarantee a reliable broadcast with bounded communication delay. If the hello message doesn't arrive in time, the neighbour table and routing table of nodes will not
update accordingly, and the node may broadcast routing packages to other nodes repeatedly, which causes the broadcast storm problem [7].

The traditional ad hoc networks design the network layer and the MAC layer separately, which brings convenience for debugging but limits the improvement of network performance at the same time [8]. Nowadays, there are many studies on the cross-layer design aiming at overcoming the performance limitations caused by lack of coordination between layers. For example, the authors in [9] propose a cross-layer based location-aware forwarding protocol using a distributed TDMA MAC for ad hoc networks, which significantly increase the packet delivery ratio while maintaining a confined forwarding overhead. On one hand, the channel utilization can be improved and the end-to-end delay in MAC layer can be decreased through the information of the routing in network layer. On the other hand, the network layer can use the information of MAC layer to improve the efficiency of routing, robustness, and reduce the routing overheads.

Clustering algorithm is proposed to improve performance in dense networks which can reduce the number of nodes involved in routing calculation, thereby lessen the routing information exchanges [10]. So it is more suitable for communication between nodes in large scale networks. The proposed algorithm in [11] uses regular hexagon grid as the clustering model and combine TDMA and FDMA together in a cluster to reduce the interference during the inter-cluster and intra-cluster communication. The authors in [12] propose a clustering algorithm by tessellating regular hexagon among the network area.

In this paper, to achieve a better performance in ad hoc, we take the advantages of both TDMA and CSMA and design a novel cross-layer neighbour discovery algorithm named ND_HC which combines TDMA and network clustering with GPS to determine the neighbour nodes of each node in the network. The proposed ND_HC algorithm is realized in MAC layer instead of routing layer. ND_HC can avoid collisions between different time frames, obtain a stable delay bounds, and achieve high channel utilization for the contention-based access.

The remainder of the paper is organized as follows. The system model is described in section 2, the ND_HC algorithm is designed in section 3, and the simulation results and analyses are given in section 4. Finally, the conclusions are drawn in section 5.

2 System Model

In this section, we give the description of hexagonal clustering, TDMA frame allocation and hello messages transmission of our proposed ND_HC algorithm.

The larger scale of the ad hoc network is, the more challenging it is for making an efficient management of network. Aiming at this problem, many cluster-based protocols and papers have proved that regular hexagonal clustering is suitable for large-scale wireless networks [13]. So we take the advantage of hexagonal clustering in our model to improve the performance of ad hoc.

![Network divided with hexagon](image_url)
As we can see in Fig. 1, the network nodes (represented by black dots) are uniformly distributed in rectangular areas. The total area of the network is divided into several hexagonal cells. Hexagons are an ideal shape for clustering networks since it can seamlessly divide the clustered areas without overlapping and it is also the regular polygon with the largest number of sides. With the aid of the hexagonal clustering, we can divide the rectangular area into hexagonal clusters and furthermore divide a cluster into six equal-sided triangles indexed from 1 to 6. Besides, all the nodes in the network are equipped with GPS devices to obtain accurate geographical location information and keep the clock synchronized so that each node can be divided into corresponding clusters according to the distance from its current location and the central of each regular hexagon. The purpose of clustering is to allocate time frames for nodes in different hexagons, and there is no fixed central node in any cluster.

In order to reduce the number of nodes sending messages at the same time, we divide one time frame cycle into six time frames in a TDMA style and allocate them to nodes with different index obtained by hexagonal clustering. In other words, nodes locate in the triangle with the same index can send messages simultaneously. As we can see in Fig. 2, each cycle consists of six time frames and each time frame comprised of multiple time slots for nodes to compete. If the remaining time of a time frame is not sufficient for a message transmission, the node will not transmit the hello message and directly drop it.

The TDMA time frames provide contention free transmission opportunities for nodes that in triangles with different index, which reduces the number of nodes ready to send hello messages in each time frame and the message collisions which greatly alleviates the pressure of nodes accessing. Furthermore, the nodes in the same indexed triangle in different hexagons will compete for accessing the channel resource in the corresponding time frame in order to improve the channel utilization efficiency. So it is inevitable to suffer from the interference between the same triangles in different hexagons.

In our system, we design a cross-layer protocol that produces hello messages in MAC layer and set a specific period of time for sending hello messages. It should be noted that hello messages are sent periodically to update the latest information of neighbour nodes.

Fig. 3 shows the procedure before each successful message transmission. The authors in [14] conclude that a randomized jitter in hello message periodicity is an important element in avoiding this unwanted interference. Message collisions or idle slots may occur before a successful transmission. The message collision only happens when at least two nodes transmit at the same time slot in the same time frame and within the communication range of each other’s. The idle time slots are time interval in which the transmission medium remains free of any transmission.

3 Algorithm Design

In this section, taking TDMA, hexagonal clustering and GPS into account, we propose the neighbour discovery algorithm, ND_HC.

While the ad hoc network is initializing, split the rectangular area with hexagonal cells seamlessly and divide each cell into 6 triangle regions indexed by area index. At the beginning of each cycle, all nodes determine the index of triangle it belongs to (according to its geographical location) and produce one hello message in their MAC layer.
Then the cycle is divided into six time frames indexed from 1 to 6. If the channel is idle, the nodes with pktTx and in the triangle area with the same index can start sending the hello message after a random backoff time within the backoff window size. The hello messages are sent with a random timing jitter in order to avoid collisions. Instead, if the channel is busy, this node will repeat the above process again with another random backoff time until this hello message is able to send, or this time frame comes to the end and drop this hello message. If the current time frame is not the allocated time frame, the node will not try to send hello message and keep silent for reception.

The major parameters of ND_HC are listed in Table 1.

| Parameters  | Explanation                              | Parameters  | Explanation               |
|-------------|-----------------------------------------|-------------|---------------------------|
| frame index | the time frame index from 1 to 6         | pktTx       | cache of packets sending   |
| area index  | triangle area index from 1 to 6          | window size | max number of backoff slots|

4 Simulation Results and Analysis

In this section, with the help of NS-2 simulator, we present extensive simulation results to evaluate the effectiveness of our proposed algorithms through comparing the performance of ND_HC and ND_802.11 which stands for the traditional neighbour discovery algorithm in IEEE 802.11.

In the simulation, nodes are uniformly distributed within a rectangular of 1299m×1250m. The main simulation parameters are shown in Table 2. The unlisted parameters are set to be the default values in NS-2.

| Parameters               | Value        | Parameters              | Value |
|-------------------------|--------------|-------------------------|-------|
| Network area            | 1299m×1250m  | Window size W           | 20    |
| Transmission range R    | 250m         | Length of time slot     | 20us  |
| Carrier sensing range   | 500m         | Length of time frame    | 0.01s |
| Radius of regular hexagon | 250m       | Wireless transmission rate | 100kbps |

We first evaluate the basic performance metrics of ND_HC under different window_size W, and then compare the performance of ND_HC and ND_802.11. Note that the following figures are plotted based on the average values of 50 runs.

In the following, we give the definition of performance metrics and then calculate them to verify the correctness of our proposed algorithm.

Neighbour discovery is defined as a process that a node receives a hello messages successfully from its neighbour node during a specified time.

The ratio of neighbours found per cycle, R_N, is defined as the ratio of the number of hello messages node receives in one cycle and the number of actual neighbour nodes, which reflect the reliability of our proposed algorithm.

The time cost for finding all neighbours, TC_AN, is defined as the average time from the moment when MAC layer produces the packet to the time node receive all number of hello messages from its neighbours.
In Fig. 4, we compare the ratio of neighbours found per cycle $R_N$ under different number of total nodes. As can be observed, in general, the $R_N$ when $W=40$ is larger than that when $W=30$ and $W=20$ and $W=10$. And the overall trend is that the larger window size is, the more neighbour nodes can be discovered, which is because the probability of message collision becomes smaller when backoff window increases.

In Fig. 5, it is seen that the time cost for finding all neighbours $TC_AN$ increases with the number of total nodes since the network’s complexity increasing rapidly. On the other hand, $TC_AN$ increases when the length of $W$ is 10, 20, and 30, 40 respectively. To be specific, when the number of nodes is relatively small, $TC_AN$ under different values of the $W$ have little difference. This is because the probability of collision is very small so the backoff seldom happens. As the number of nodes increases, the length of backoff becomes more and more important to the network performance since backoff occurs frequently. Furthermore, the time takes for finding all neighbours when $W=10$ increases rapidly comparing to that when $W=20$ and $W=30$ and $W=40$, which is because of the message collision. When $W$ is bigger than 20, at the current node density, the improvement of network performance is relatively low which is because the collision probability is already small so the increase of time window is meaningless. Particularly, too big $W$ may degrade the network performance since it will increase the delay.

Then, we compare ND_HC and ND_802.11 in terms of $R_N$ and $TC_AN$ under different number of nodes.

It can be seen in Fig. 6 that the $R_N$ of both ND_HC and ND_802.11 decreases as the number of nodes increases, which is because more collision happens when more nodes compete for channel resource to send hello messages. With collision probability rising, ND_802.11 leads to more waste of network resources than ND_HC, which is because the nodes are partitioned into different parts in ND_HC to reduce the density of nodes in each time frame. In addition, the $R_N$ of our algorithm...
decreases much slower than ND_802.11 and it is worthwhile to emphasize that ND_HC is capable of finding 65% neighbours even when the number of nodes increases to 280, while that of ND_802.11 is 30%.

Fig. 7 shows how long it takes for ND_HC and ND_802.11 to find all the neighbour nodes when the total number of nodes increases. It can be seen that when the number of nodes is less than 56, ND_802.11 is slightly faster than ND_HC. It is because in a sparse distributed network, ND_HC divides a time period into six time frames so it is possible that the some time frames may have no nodes to send messages, which causes low utilization of time. When the number of nodes exceeds 56, ND_HC performs better than ND_802.11 and growth rate of ND_HC is much slower than that of ND_802.11. This is because ND_HC allocates time frames to send hello messages, which greatly reducing message collision.

5 Conclusion
In this paper, we put forward a cross-layer neighbour discovery algorithm, ND_HC, which combines TDMA, regular hexagonal clustering algorithm and GPS. We use the random backoff mechanism to improve the transmission efficiency of sending hello messages and study the influence of different backoff window size on the network performance. Compared with traditional neighbour discovery algorithm in IEEE 802.11, ND_HC is more effective for finding neighbour nodes in terms of the ratio of neighbours found per cycle and the time cost for finding all neighbours.

6 Acknowledgement
This work was supported by the National Natural Science Foundation of China (Grant Nos. 61372076, 61301171), Shaanxi Key Research and Development Program (Grant No. 2017GY-080), Foundation of Science and Technology on Communication Networks Laboratory (KX172600031) and the 111 Project (No. B08038)

7 References
[1] J. Lin, C. Wu, S. Ohzahata, T Kato, IEICE Trans. Commun. E 100, 42(2017).
[2] T. Terami, T. Ohta, Y. Kakuda, IEEE. COMPSAC 19(2017).
[3] B.P. Patil, D.M. Rahul, IJEECS 7, 795(2017).
[4] B. Hamid, E.N.E. Mokhtar, IEEE. ICTA 1(2016).
[5] J. Huang, Q. Li, S. Zhong, L. Liu, P. Zhong, Sensors. 17, 338(2017).
[6] Z. Shen, X. Zhang, M. Zhang, W. Li, D. Yang, IEEE. Access. 99, 1(2017).
[7] D. Lu, S. Dong, IJDSN 13, 11(2017).
[8] R. Mehta, D.K. Lobiyal, Telecommun. Syst. 66, 1(2017).
[9] H.K. Yarnagula, S.K. Deka, N. Sarma, Wireless. Pers. Commun. 95, 1(2017).
[10] S. Sharma, V. Pandit, D. Agrawal, IEEE. MASS 1(2012).
[11] S. Mukherjee, G.P. Biswas, Springer India. RAIT 266, 11(2014).
[12] J.F. Qiao, S.Y. Liu, H.E. Hui-Min, JNS. of HLJ. Univ. 31, 535(2014).
[13] D. Jia, M. Li, H. Zhu, B. Zhang, WL. Personal Comm. 94, 1(2017).
[14] M. Salmanian, J.D. Brown, M. Li, T. Willink, IEEE. MILCOM 563(2016)