An Anti-collision Neighbor Discovery Protocol for Multi-node Discovery

Mingjin Gao[†§], Rujing Shen[†§], Linhong Mu[†*], Xinying Liao[◇], Jun Li[‡], Yiqing Zhou[†§]
[†]Institute of Computing Technology, Chinese Academy of Sciences, Beijing, CHINA
[§]University of Chinese Academy of Sciences, Beijing, CHINA
[*]Chongqing University of Posts and Telecommunications, Chongqing, CHINA
[◇]Beijing Information Science and Technology University, Beijing, CHINA
[‡]School of Electronic and Optical Engineering, Nanjing University of Science and Technology, Nanjing, CHINA
Email: gaomingjin@ict.ac.cn, shenrujing@ict.ac.cn, jun.li@njust.edu.cn

***Abstract*—The surging mobile devices provide both opportunities and challenges for building autonomous or adhoc networks. To build these networks, the top priority is to make devices discover each other quickly. However, existing neighbor discovery protocols (NDPs) may introduce large discovery latency due to beacon collisions in multi-node networks. In this paper, we propose an anti-collision NDP to avoid beacon collisions by carrier sensing. Moreover, we reduce the discovery latency by prioritising nodes which wait for a long time and controlling active slot duration dynamically. Simulation results show that our approach avoids most beacon collisions and reduces the discovery latency by more than 40%.**

***Index Terms*—Neighbor discovery protocol, beacon collisions, carrier sense**

I. Introduction

Due to the increasing popularity of mobile devices, people tend to be connected tightly with each other through autonomous or adhoc networks. These networks are composed of multiple mobile devices with communication and collaboration capacities [1–8]. It is a prerequisite for establishing an autonomous or adhoc network to enable devices discover one another with a short latency, which has attracted significant interest.

Many neighbor discovery protocols (NDPs) have been proposed to conduct discovery among neighbors by allowing devices to alternate between the active and sleeping status periodically [9–13]. The Birthday protocol was inspired by the Birthday Paradox, that each node listens, transmits, or sleeps under probabilities for achieving fast neighbor discovery in the average case [9]. The Searchlight protocol tried to increase the chance that active status durations of neighbors meet by adopting over-half occupation [10]. The Quorum protocol followed the notion of quorum [11], that each node arbitrarily picks one column and one row of the group as active status durations [12]. The Disco protocol was derived from the Chinese Remainder Theorem, that each node wakes up at moments of the primes selected in advance [13]. The works mentioned above trade off between the power consumption and the discovery latency of mobile devices. However, nowadays many smart devices relax the requirement on the energy consumption [14]. Thus, compared with low energy consumption, the importance of faster discovery has greatly risen [15]. However, the aforementioned works did not appropriate for demands of smart devices due to relatively high discovery latency. Actually, the high discovery latency is due to collisions of the neighbor discovery beacons, which is seldom considered before. Moreover, with the increasing popularity of mobile device, NDPs designed for two or three nodes do not fit the multi-node networks any more. Motivated by this, our work investigates avoidance of beacon collisions and reduction of the discovery latency among multiple nodes.

Only a few works in the literature have studied the beacon collision problem. The Carrier-Sensing Multiple Access (C-SMA) [16] was proposed to avoid beacon collisions during message transmission by carrier sensing. But CSMA is not applied to neighbor discovery, and it does not mention the carrier sensing process in detail. For example, when to carrier sense or what to do if there is a collision. Authors in [17] designed a mechanism that allowed each node to autonomously control its active status duration based on the detected amount of beacon collisions. However, this mechanism cannot avoid collisions of beacons in advance as it is conducted after collision occurs.

In this paper, we design a scheme which can be applied to existing NDPs to avoid beacon collisions as well as to reduce the discovery latency. In this scheme, each node uses the carrier sensing [18] to avoid the collision of beacons. Moreover, when a node abandon its beacon opportunity to avoid collision, it can dynamically extend its active status duration as compensation to increase its opportunity for neighbor discovery. Theoretical analysis and simulation results show the benefit of adopting our scheme compared to the existing NDPs in terms of the discovery latency and collision avoidance performance.

The rest of this paper is organized as follows: we introduce the terms and assumptions in Section II. In Section III, we propose our protocol. In Section IV, we analyse the feasibility and superiority of the proposed protocol. In Section V, we apply our approach to the existing NDPs, and analyse the discovery latency and collision avoidance performance. Finally, we conclude this paper in Section VI.

II. System Model

In this section, we detail the system model. We first introduce several basic terminologies that will be adopted throughout the rest of this paper.

- **Active slots and idle slots:** When nodes are in active slots, they can listen beacons, sense carrier and send beacons. On the contrary, nodes sleep in idle slots.
- **Carrier sense [16]:** Nodes detect the channel status (idle/occupied) before deciding to send a beacon.
- **Duty cycle:** It is the fraction of time a node spends in active slots.
- **Discovery latency:** It measures time nodes spend to wait until they discovers all their neighbors.
- **Beacon collision:** When multiple nodes simultaneously send beacons, conflicts arise. It means that all the conflicted beacons fail to transmit.

Based on the terminologies introduced above, we present the following assumptions.

- **Bidirectional links:** We assume that neighbor nodes have the same transmission range.
- **Radio model:** In an active slot, listening and carrier sensing can be proceed simultaneously, while beaconing is independent.

III. The Anti-collision NDP

In this section, we discuss the anti-collision neighbor discovery protocol or simply 'Ac NDP' for short. The protocol can avoid beacon collisions and significantly reduce discovery latency among multiple nodes.

A. Overall Process

Most of NDPs adopt a time-slotted model, where continuous time is separated into discrete time slots. These slots have the same duration and is long enough for neighbor discovery. In this paper, we adopt the time-slotted model proposed by Dutta and Culler, where a beacon is transmitted at both the start and the end of an active slot [13]. It is demonstrated that, this time-slotted model can ensure an adequate overlap of active slots even when slots of nodes are unaligned.

In our design, each active slot is further divided into n units. Each unit is equal to the time needed to send a beacon. Specifically, as shown in Fig. 1, within an active slot, it takes one unit for one carrier sensing or one beaconing, while random waiting and listening may cost more units (e.g., k_1 or k_2 units for random waiting and remaining units for listening).

Unlike traditional NDPs, in our design, in order to avoid collisions, each node first utilizes carrier sense to detect whether the channel is idle before sending its beacon. If the channel is idle as shown in Fig. 1, the node will wait for a random time k_1, and then conduct a second carrier sense. The reason for this is to guarantee the idleness of the channel. Otherwise, if the channel is occupied as shown in Fig. 2, the node will give up beaconing so as to avoid

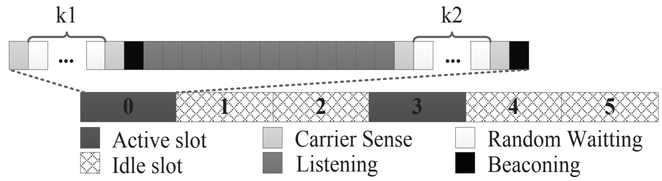

Fig. 1. Time-slotted model without beacon collisions.

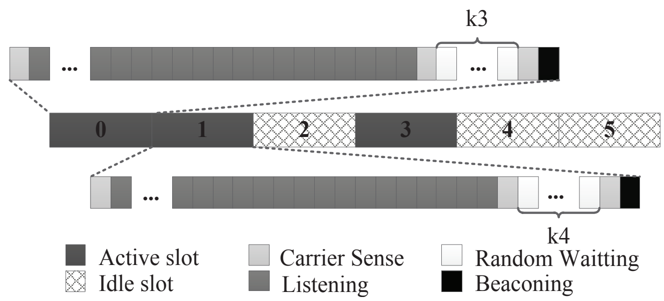

Fig. 2. Time-slotted model with two beacon collisions.

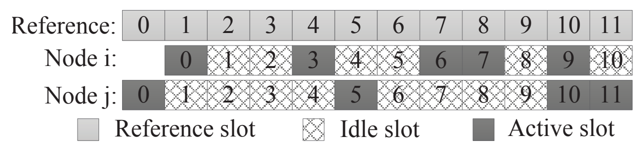

Fig. 3. Neighbor discovery process under two-node environment.

collisions. In addition, as shown in Fig. 2, if the node can not send two beacons in one active slot, it will automatically extend its active duration to next slot until two beacons are sending out cumulatively in one active duration.

This anti-collision NDP can be applied to existing NDPs. In this paper, we take Disco protocol as an example. For easy understanding, we consider there are two nodes with aligned slots need to discover each other, as shown in Fig. 3. They first select a pair of prime numbers (p, q) respectively. For example, assume that node i chooses $(3, 7)$ while node j chooses $(5, 11)$, and the prime number pairs decide their active and idle patterns. Then, we apply our Ac NDP to the two nodes. Thus, each node will send beacons at both the start and the end of an active slot. They can also use carrier sense to avoid beacon collisions as we introduced in Fig. 4. However, considering that there may be nodes always giving up sending beacons, we design beaconing priorities for fair.

B. The Priority of Waiting Time

In order to compensate those nodes who have already given up sending a beacon previously, we prioritize their waiting time. Specifically, we define k as the amount of units taken for each random waiting, and define r_l and r_u as lower and upper bounds in advance, where $0 < r_l < r_u$. For the node who has already given up sending a beacon before, k is randomly generated from $(0, r_l)$ to reduce its next waiting time for compensation. For the node who sends

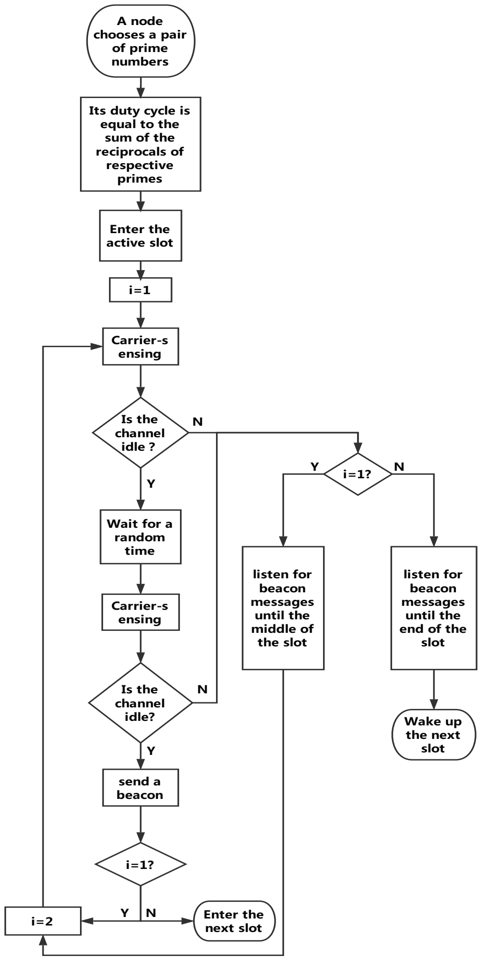

Fig. 4. Flow chat of the Ac NDP.

beacons successfully, k is randomly generated from (r_l, r_u) to avoid collision with nodes who need to be compensated.

As shown in Fig. 1, the node sends the first beacon successfully. Thus, its waiting times k_2 before sending the second beacon is randomly generated from (r_l, r_u). However, in Fig. 2, the node gives up beaconing to avoid collisions. For compensation, we prioritize the node by randomly generating its waiting time k_3 from $(0, r_l)$.

In addition, considering the length of each slot, we should note that the upper bound should satisfy $r_u < \frac{n-6}{2}$. The reason is that if a node can send two beacons in an active slot as shown in Fig. 1, it will spent 2 units on beaconing and 4 units on carrier sensing, and thus these 6 units can not be used for random waiting. Thus, its duration for each random waiting is less than $\frac{n-6}{2}$ units since the node waits twice in an active slot.

C. Dynamic Active Slot Length Control

Considering that not all nodes can send two beacons in one active slot successfully, we try to increase beaconing opportunities for nodes that meet collisions. Inspired by [17], where slots are doubled when beacon collision occurs and are halved when no collisions detected, we propose a dynamic active slot length control mechanism. In this mechanism, if a node can not send two beacons in one active slot, it will automatically extend its active duration to next slot until two beacons are sending out cumulatively as shown in Fig. 2.

IV. Analysis

In this section, we analyse the feasibility and superiority of our proposed Ac NDP scheme. We take the simplest co-primality based NDP for example. Two nodes i and j, pick two numbers m_i and m_j which are co-primes. Node i starts counting at slot s_i and its duty cycle is $1/m_i$. Thus, we know that its active slot number is $x_i = s_i + m_i v, \forall v \in Z^+$. Similarly, for node j, its slot number is $x_j = s_j + m_j v, \forall v \in Z^+$.

A. Feasibility for Neighbor Discovery

To help with the illustration of the feasibility of Ac NDP scheme, we introduce the Chinese Remainder Theorem.

Lemma 1: Let m and n be positive integers. For any integers a and b, there exists an integer $x \in \{0\} \cup Z^+$ such that

$$x \equiv a \ (mod \ m), \quad (1)$$

$$x \equiv b \ (mod \ n) \quad (2)$$

$$\text{if and only if } a \equiv b \ (mod \ gcd(m, n)). \quad (3)$$

Based on the Lemma 1, the co-primality based NDPs (e.g. Disco) have been proved to guarantee discovery for any two nodes [13]. Thus, the Ac NDP is feasible since it is based on feasible NDPs.

B. Superiority in Discovery Latency

The Chinese Remainder Theorem also provides a common solution for x. Specifically, for the simplest co-primality based NDP, modes can discover each other at the slot $x = x_0 + m_i m_j v$ for some integer v. One x_0 is

$$x_0 = s_i w_i m_j + s_j w_j m_i, \quad (4)$$

where w_i and w_j must satisfy

$$w_i m_j = 1 \ (mod \ m_i), \quad (5)$$

$$w_j m_i = 1 \ (mod \ m_j). \quad (6)$$

However, if there is an beacon collision at slot $x_0 + m_i m_j v$, node i and j have to wait for at least $m_i m_j$ slots to discover each other, which means $x = x_0 + m_i m_j (v + u), u \in Z^+$. But if we apply the Ac NDP, node i and j will wake up in the next slot to send beacons again. Thus we have $x = x_0 + m_i m_j v + u, u \in Z^+$. It is obvious that the Ac NDP can significantly reduce the discovert latency compared with the existing NDPs by avoiding beacon collisions and controlling the active slot length dynamically.

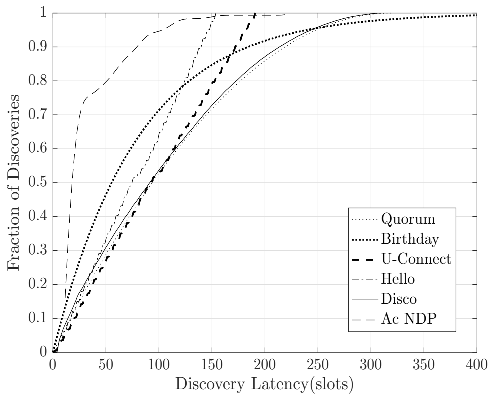

Fig. 5. CDF of discovery latency for different NDPs when N=100.

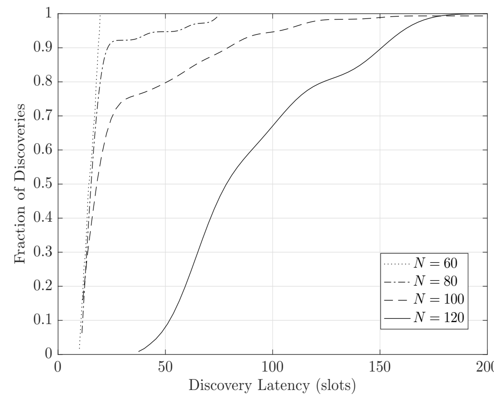

Fig. 6. CDF of discovery latency for different nodes amounts.

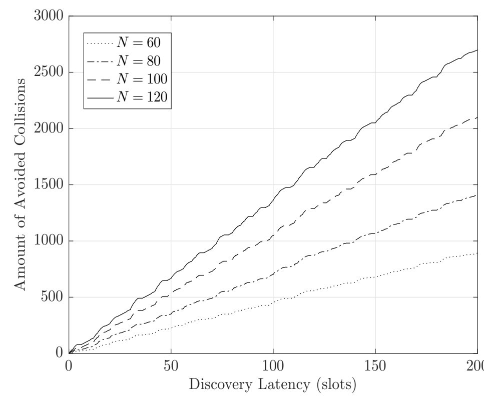

Fig. 7. Amount of collisions avoided by the Ac NDP for different nodes amounts.

V. Performance Evaluation

In this section, we present the simulation results for the evaluation of our proposed design on discovery latency as well as collision avoidance. The results are derived by repeating neighbor discovery process by 5000 times for each NDP with multiple nodes. In the simulation, we set $n = 100, r_l = 10, r_u = 15$.

In Fig. 5, we present the discovery latency of several popular NDPs by showing the cumulative distribution function (CDF) of the latency when the amount of nodes N is 100. The figure shows that, although Birthday protocol can ensure a fastest discovery in most case, it may lead to huge latency periodically. Disco and Quorum have similar CDFs, which are inferior to Birthday for over 95% of the time. However, they both have shorter predictable maximum latencies compared with Birthday. Hello and U-Connect's performance are mediocre, but they have shorter predictable maximum latencies. In general, we can find the average discover latency of these NDPs is around 250 slots.

In order to illustrate the benefits of our design, we test the discovery latency of Ac NDP under different amounts of nodes N as shown in Fig. 6. We can find that, when $N = 100$, Ac NDP can find almost 70% neighbors within only 25 slots and can find almost 95% neighbors within 100 slots. However, in Fig. 5, Birthday, as the fastest NDP can only find 70% neighbors within 100 slots. Moreover, the predictable maximum latency of Ac NDP is around 150 slot, which is close to the shortest predictable maximum latency. Therefore, we can infer that the Ac NDP can reduce the discovery latency by more than 40%. Fig. 6 also shows that although the discover latency increases significantly with the increasing of N, the Ac NDP can find most neighbors within a short time.

In Fig. 7, we record the amount of collisions avoided by the Ac NDP under different N. As we known, collisions happen much more frequently when N is larger, but the amount of avoided collisions increases linearly with the growth of N. It means that, the Ac NDP rearrange the beacon transmission process, which avoid the potential collisions.

VI. Conclusion

In this paper, we proposed a mechanism, namely Ac NDP, to avoid beacon collision in neighbor discovery, which can further reduce the discovery latency. In this scheme, each node uses carrier sensing and random waiting to ensure that the channel is idle before sending its beacon to avoid collision. We also presented a priority mechanism to prevent node from waiting too long. In addition, considering the loose requirement on energy consumption for current smart devices, nodes can dynamically wake up in the next slot to increase the probability of neighbor discovery. We illustrate the feasibility and superiority of the proposed protocol theoretically. Simulations results show that, the Ac NDPs can significantly reduce beacon collisions, and reduce discovery latency by over 40%. In addition, although the discover latency increases significantly with the increasing of N, the Ac NDP can find most neighbors within a short time. In conclusion, our mechanism is suitable for multi nodes environment and it can effectively avoid beacon collisions and reduce discovery latencies.

Acknowledgments

This work is supported in part by the key project of the National Natural Science Foundation of China under Grant

61431001, in part by Beijing Natural Science Foundation under Grant L172049.

References

[1] M. Collotta, G. Pau, T. Talty, and O. K. Tonguz, "Bluetooth 5: a concrete step forward towards the iot," *IEEE Commun. Mag.*, 2017.

[2] X. Sun and N. Ansari, "Edgeiot: Mobile edge computing for the internet of things," *IEEE Commun. Mag.*, vol. 54, no. 12, pp. 22–29, 2016.

[3] J. R. Jiang, Y. C. Tseng, C. S. Hsu, and T. H. Lai, "Quorum-based asynchronous power-saving protocols for ieee 802.11 ad hoc networks," in *Proceedings of the International Conference on Parallel Processing*, 2003.

[4] Y. Zhou, L. Tian, L. Liu, and Y. Qi, "Fog computing enabled future mobile communication networks: A convergence of communication and computing," *IEEE Commun. Mag.*, vol. 57, no. 5, pp. 20 – 27, May 2019.

[5] L. Liu, Y. Zhou, J. Yuan, W. Zhuang, and Y. Wang, "Economically optimal ms association for multimedia content delivery in cache-enabled heterogeneous cloud radio access networks," *IEEE J. Sel. Areas Commun.*, vol. 37, no. 7, pp. 1584 – 1593, Jul. 2019.

[6] L. Liu, Y. Zhou, W. Zhuang, J. Yuan, and L. Tian, "Tractable coverage analysis for hexagonal macrocell-based heterogeneous udns with adaptive interference aware comp," *IEEE Trans. Wireless Commun.*, vol. 18, no. 1, pp. 503 – 517, Jan. 2019.

[7] L. Liu, Y. Zhou, V. Garcia, L. Tian, , and J. Shi, "Load aware joint comp clustering and inter-cell resource scheduling in heterogeneous ultra dense cellular networks," *IEEE Trans. Veh. Technol.*, vol. 67, no. 3, pp. 2741 – 2755, Mar. 2018.

[8] L. Liu, Y. Zhou, L. Tian, and J. Shi, "Cpc-based backward-compatible network access for lte cognitive radio cellular networks," *IEEE Commun. Mag.*, vol. 53, no. 7, pp. 93 – 99, Jul. 2015.

[9] M. J. McGlynn and S. a. Borbash, "Birthday protocols for low energy deployment and flexible neighbor discovery in ad hoc wireless networks," in *Proc. 2nd ACM Int. Symp. Mob. ad hoc Netw. Comput. - MobiHoc 01, 2001.*, 2001, pp. 137–145.

[10] M. Bakht, M. Trower, and R. H. Kravets, "Searchlight:won't you be my neighbor?" in *Proceedings of the 18th annual international conference on Mobile computing and networking*, 2012, pp. 185–196.

[11] S. J. Mullender and P. M. B. Vitnyi, "Distributed match-making," *Algorithmica*, vol. 3, no. 1-4, pp. 367–391, 1988.

[12] Y. C. Tseng, C. S. Hsu, and T. Y. Hsieh, "Power-saving protocols for ieee 802.11-based multi-hop ad hoc networks," in *Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, 2003.

[13] P. Dutta and D. Culler, "Practical asynchronous neighbor discovery and rendezvous for mobile sensing applications," in *Proceedings of the 6th ACM conference on Embedded network sensor systems - SenSys 08*, 2008.

[14] W. Sun, Z. Yang, X. Zhang, and Y. Liu, "Energy-efficient neighbor discovery in mobile ad hoc and wireless sensor networks: A survey," *IEEE Commun. Surv. Tutorials*, vol. 16, no. 3, pp. 1448–1459, 2014.

[15] B. Han and A. Srinivasan, "ediscovery: Energy efficient device discovery for mobile opportunistic communications," in *Proc. IEEE ICNP*, 2012, pp. 1–10.

[16] J. L. Sobrinho and A. S. Krishnakumar, "Quality-of-service in ad hoc carrier sense multiple access wireless networks," *IEEE J. Sel. Areas Commun.*, vol. 17, no. 8, pp. 1353–1368, 1999.

[17] Y. Zhang, K. Bian, L. Chen, P. Zhou, and X. Li, "Dynamic slot-length control for reducing neighbor discovery latency in wireless sensor networks," in *2017 IEEE Global Communications Conference*, 2018.

[18] J. Peng and L. Cheng, "Revisiting carrier sense multiple access with collision avoidance (csma/ca)," in *2006 40th Annual Conference on Information Sciences and Systems*, 2006.