

An Improved Algorithm for Neighbor Node Discovery in Wireless Ad Hoc Networks

J. Jenifer Kani

M.E Scholar

Department of Computer Science and Engineering
Manonmaniam Sundaranar University
Tirunelveli-627012, India

G. Muthulakshmi

Assistant Professor

Department of Computer Science and Engineering
Manonmaniam Sundaranar University
Tirunelveli-627012, India

Abstract- Neighbor discovery is one of the first steps in configuring and managing a wireless network. In earlier approach of ALOHA-like neighbor discovery algorithm, whose running time is $\theta(n \ln n)$, nodes cannot differentiate between the collision and an idle slot. The proposed system uses an Order-optimal neighbor discovery algorithm that employs feedback from receiving nodes and allows each node to discover all its neighbors in $\theta(n)$ time with high probability. Interestingly, it is found that receiver feedback can be used even when nodes cannot detect collisions, and proposed a novel algorithm that achieves a $\theta(n)$ running time. When nodes do not know n , the system provides a provably correct termination condition that allows each node to terminate neighbor discovery after discovering all its neighbors. Experimental results shows that the ALOHA-like algorithm is at most a factor $\min(\Delta, \ln n)$ worse than the optimal.

Keywords— Ad hoc networks; initialization; neighbor discovery; wireless networks;

I. INTRODUCTION

Wireless Ad hoc networks are networks in which communication links are based on wireless technologies and the network structure or topology is not pre-defined rather build through dynamic network connectivity. Wireless ad hoc networks are infrastructure-less type of wireless networks that are required to configure themselves upon deployment, in order to establish an efficient communication infrastructure. For instance, immediately after deployment, a node has no knowledge of other nodes in its transmission range and needs to discover its neighboring nodes in order to efficiently communicate with other nodes in the network. Knowledge of one-hop neighbors is required by medium access control protocols[9], routing algorithms, and topology control algorithms to work efficiently and correctly. Thus, neighbor discovery is one of the first steps in the configuration of a large wireless network.

Neighbor discovery is nontrivial for several reasons.

- 1) Neighbor discovery needs to cope with collisions. Ideally, a neighbor discovery algorithm needs to minimize the probability of collisions and, therefore, the time to discover neighbors.
- 2) In many practical settings, nodes have no knowledge of the number of neighbors, which makes coping with collisions even harder.

3) When nodes do not have access to a global clock, they need to operate asynchronously and still be able to discover their neighbors efficiently.

4) In asynchronous systems, nodes can potentially start neighbor discovery at different times and, consequently, may miss each other's transmissions.

5) Furthermore, when the number of neighbors is unknown, nodes do not know when or how to terminate the neighbor discovery process.

Several neighbor discovery algorithms have been proposed earlier that has various practical limitations, since, nodes requires priori estimates of number of neighbors. This paper analyzes two neighbor discovery algorithms, $\theta(n \ln n)$ ALOHA-like neighbor discovery algorithm and $\theta(n)$ Order-optimal receiver feedback based algorithm. The neighbor discovery algorithms discussed in this paper do not require nodes to have a priori knowledge of the number of neighbors and allows asynchronous operation. They also allow nodes to begin execution at different time instants and enable each node to detect when to terminate the neighbor discovery process.

II. NEIGHBOR DISCOVERY

Neighbor discovery is a process by which a node in a network determines the total number and identity of other nodes in its vicinity. It is a fundamental building block of many protocols including localization, routing, leader election and group management. Time based communications and many media access control mechanisms rely on accurate neighbor information. Neighbor discovery is especially important to the proper functioning of wireless networks. Nodes found within the neighborhood are neighbors and, depending on network configuration and topology, may cooperate in the performance of various tasks including communication, sensing and localization.

Neighbor discovery algorithms can be classified into two categories, viz. randomized or deterministic. In randomized neighbor discovery, each node transmits at randomly chosen times and discovers all its neighbors by a given time with high probability (w.h.p.). In deterministic neighbor discovery, on the other hand, each node transmits according to a predetermined transmission schedule that allows it to discover all its neighbors by a given time with probability one.

A. Outline of the Proposed work

1. Initially, ALOHA-like neighbor discovery algorithm is proposed in which each node discovers all its n neighbors in an expected amount of time equal to $n \epsilon (\ln n + c)$, for some constant c .

2. The proposed neighbor discovery algorithm assumes nodes can detect collisions i.e., nodes can distinguish between collisions and idle slots. We show that collision detection results in a $\ln n$ improvement over the ALOHA-like algorithm.

3. Next it shows that the absence of an estimate of the number of neighbor's results in a slowdown of no more than a factor of two, compared to when nodes know n .

4. Further it shows that lack of synchronization among nodes results in at most a factor of two slow down in algorithm performance from the case when nodes are synchronized.

5. Finally, it shows that despite starting execution at different time instants, each node can discover all its neighbors. Furthermore, when nodes do not know n , it shows that with high probability (w.h.p), each node can determine that it has discovered all its neighbors.

6. Thus the proposed system have $O(n \ln n)$ ALOHA-like discovery algorithm when nodes do not have a collision detection mechanism and $O(n)$ Collision Detection-based algorithm for the case when nodes can detect collisions.

III. RELATED WORKS

R.Khalili et.al [3] proposed a physical layer mechanism for how nodes in receive mode detect the channel status, describing algorithms at higher layers that exploit such a knowledge, and characterizing the significant gain obtained. In particular, it describes a possible physical layer architecture that allows receivers to detect collisions, and then introduce a feedback mechanism that makes the collision information available to the transmitters. This allows nodes to stop transmitting packets as soon as they learn about the successful reception of their discovery messages by the other nodes in the network. Hence, the number of nodes that need to transmit packets decreases over time. These nodes transmit with a probability that is inversely proportional to the number of active nodes in their neighborhood, which is estimated using the collision information available at the nodes.

C.Law et.al [11] proposed an efficient collision resolution protocol and its variations for the tag identification problem, where an electromagnetic reader attempts to obtain within its read range the unique ID number of each tag. The novelty of this protocol is that each tag is memory less, i.e., the current response of each tag only depends on the current query of the reader but not on the past history of the reader's queries. Moreover, the only computation required for each tag is to match its ID against the binary string in the query. Here a designated tag reader needs to determine the IDs of tags in its range. *Prima facie*, the tag identification problem bears resemblance to the neighbor discovery problem.

W.Zeng et.al [2] proposed a neighbor discovery in MPR networks that allow multiple packets to be received successfully at a receiver. It is motivated by the increasing prevalence of multipacket reception (MPR) technologies such as CDMA and MIMO. Starting with a simple ALOHA-like algorithm, it shows that the time for all the nodes to discover their respective neighbors is $\Theta(\ln n)$ in an idealized MPR network that allows an arbitrary number of nodes to transmit simultaneously.

M.J. McGlynn [10] proposed a synchronous ALOHA-like neighbor discovery algorithm. More recently, an asynchronous, randomized neighbor discovery algorithm has been proposed in [6]. Keshavarzian *et al.*[8] propose a novel, deterministic neighbor discovery algorithm. However, nodes need to be synchronized with each other and need to know the maximum number of neighbor's n a priori. Furthermore, the neighbor discovery needs time-slots to discover all the neighbors.

D.Angelosante et.al [5] proposed a neighbor discovery algorithms using a multiuser detection approach have been proposed. However, these algorithms require synchronization between nodes and also require each node to know the signatures of every other node in the network. J.Luo et.al [4] proposed an interesting approach to neighbor discovery based on group testing, but it also suffers from the same practical limitations as in Multiuser Detection approach.

S.Vasudevan et.al [7] proposes several probabilistic algorithms in which nodes perform random, independent transmissions to discover their one-hop neighbors. In general, these solutions propose antenna scanning strategies for efficient neighbor discovery. However, none of these proposals addresses the practical challenges of neighbor discovery.

IV. IMPLEMENTATION METHODOLOGY

The overall architecture of implementation methodology is shown in Fig 1. Initially, an ad hoc network is created consisting of 50 nodes. Few nodes are clustered together to form mobile groups. The coverage for each node in mobile groups is identified. To identify the coverage, each node sends a HELLO message to all the other nodes in a mobile group. The nodes which acknowledge this particular node are called coverage nodes. These nodes become neighboring nodes. Once the neighboring nodes are identified, the node can start its transmission. This paper focuses on neighbor discovery using order-optimal receiver feedback based algorithm which exploits feedback from receiving nodes when nodes can detect collisions. Further, the results are analyzed with $\theta(n \ln n)$ ALOHA-like neighbor discovery algorithm when nodes cannot detect collisions.

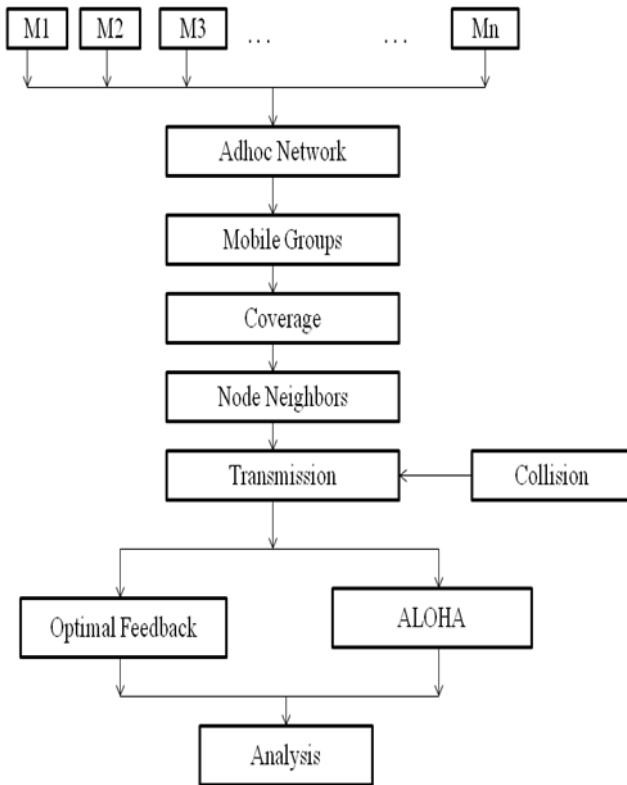


Fig 1. Overall architecture of implementation methodology

A) ALOHA-Like Neighbor Discovery

The ALOHA-like algorithm is a randomized algorithm that operates as follows. In each slot, a node independently transmits a DISCOVERY message announcing its ID, with probability, and listens with probability $1-P_x$. A discovery is made in a given slot only if exactly one node transmits in that slot.

Slotted ALOHA

1) Time is "slotted" and all users are synchronized to these slot intervals.

2) When a packet arrives within a slot, user delays the transmission until next time slot. Therefore, the vulnerable period is P only. Using similar assumption that the total traffic is a Poisson process:

$$S = Ge^{-G} \quad - (1)$$

3) To obtain the maximum value of S , use similar technique, the maximum occurs at $G = 1$, we have

$$S^* = e^{-1} = 0.368 \quad - (2)$$

where, e denotes collision.

4) It is twice that of pure ALOHA.

Asynchronous Operation

The asynchronous ALOHA-like algorithm operates as follows. Between successive transmissions, each of fixed duration, each node remains in receive mode for an exponentially distributed time interval with mean $1/\lambda$.

Initiating Neighbor Discovery

Suppose the wireless network is deployed during time interval $[t, t + \eta]$, where η is an upper bound on the deployment period and assumed to be known in advance. When nodes have access to a global clock, initiating neighbor discovery is trivial, as each node can begin execution at a globally agreed upon time instant. When nodes do not have access to a global clock, initiating neighbor discovery is nontrivial since clocks at different nodes may proceed at different rates resulting in clock offsets between nodes. Hence, this algorithm assumes that, the maximum clock offset between any two nodes in the network is bounded by δ . Each node starts neighbor discovery when its local clock reaches, which is determined prior to deployment. To account for clock offsets, the algorithm extend each phase by δ time units to each phase, i.e., the r th phase lasts duration of $2^{r+2} e \ln 2^r + \delta$ time units. Thus, all nodes are simultaneously in phase for at least $2^{r+2} e \ln 2^r$ time units, guaranteeing that each node discovers all its neighbors w.h.p. when $r = \lceil \log n \rceil$,

Terminating Neighbor Discovery

When the value of n is not known to a node, it cannot know when it has discovered all its neighbors and terminate neighbor discovery. The conclusion of ALOHA-like algorithm in the clique setting is made by describing a provably correct termination condition that allows each node to determine if it has discovered all its neighbors.

Let $D_{i,r}$ be the number of nodes discovered by node i in the r th phase. Then, the termination condition used by node i is as follows:

TC: Halt at the end of r th phase if $D_{i,r-1} \geq 2^{r-2}$ and $D_{i,r} < 2^{r-1}$, where $r \geq 2$.

B) Order-Optimal neighbor discovery using feedback

The proposed system describes a $\theta(n)$ neighbor discovery algorithm that exploits feedback from receiving nodes. A feedback to a transmitting node allows it to determine if it has been discovered by other nodes, which then allows it to stop transmitting.

It first describes a $\theta(n)$ neighbor discovery algorithm employing feedback when nodes can detect collisions, i.e., each node can distinguish between a collision and an idle slot. Subsequently, it describes how feedback can be exploited to achieve a $\theta(n)$ algorithm even when nodes cannot detect collisions. The system assumes that all the n nodes are arranged in a clique.

The proposed system employs Order-optimal neighbor discovery using feedback algorithm. They are of two types.

- Collision Detection-Based Neighbor Discovery
- Neighbor Discovery without Collision Detection

1) Collision Detection-Based Neighbor Discovery

The key idea behind the algorithm is as follows: Divide each slot into two sub slots. Upon successful reception of a

DISCOVERY message in the first sub slot, each receiving node transmits bit "1" to the source of the message (say node) in the second sub slot, potentially causing a collision at node. Collision detection allows node to detect that the second sub slot is not idle and that its transmission in the first sub slot was received by all other nodes. Upon doing so, it "drops out" of neighbor discovery, i.e., stops transmitting. The remaining nodes (henceforth, called surviving nodes) then increase their transmission probabilities in the next slot. As allowing nodes that have been discovered to drop out and requiring the surviving nodes to increase their transmission probabilities yields a significant improvement over the ALOHA-like algorithm.

Asynchronous Collision Detection Based Neighbor Discovery

Unlike the synchronous version, receiving nodes transmit feedback in response to an unsuccessful transmission. The reason for this choice is as follows.

During an unsuccessful busy period, a message transmission may overlap with the feedback period of another message. Hence, sensing energy during the feedback period has to be taken as an indication of unsuccessful transmission. This algorithm allows a node to begin transmission, despite detecting a busy period. The algorithm performance can be improved by suppressing such transmissions. However, despite allowing such an event to occur, our analysis shows that the algorithm takes only twice as long to complete neighbor discovery than its synchronous counterpart. Thus, transmission suppression only improves this constant and not the asymptotic order.

2) Neighbor Discovery without Collision Detection

This section describes, how Order-optimal neighbor discovery can be employed, even when nodes cannot detect collisions. The algorithm is divided into *rounds* (as indexed by variable). Round r lasts for a duration of T_r slots where

$$T_r = \frac{8n}{2^r} + 8 \log 2n \quad - (3)$$

Instead of providing a single bit of feedback, each node now includes in its DISCOVERY messages the ID of the most recently discovered neighbor in addition to its own ID. If a receiving node hears its ID during a round, it drops out of neighbor discovery at the end of the round.

The key idea behind this algorithm is as follows. At the end of r rounds, where $1 \leq r < \log \log n$, there are at most $n/2^r$ surviving nodes w.h.p., and the remaining nodes, which hear their IDs back, drops out of neighbor discovery.

Therefore, in the $r+1$ st round, each surviving node increase its transmission probability to $2^r/n$. After $\log \log n - 1$ rounds, there are at most $n/\log n$ surviving nodes w.h.p. At this point, each surviving node simply runs the ALOHA-like neighbor discovery.

V. PERFORMANCE ANALYSIS

To analyze the performance of the proposed order-optimal receiver feedback based algorithm, with the one which is implemented earlier, two performance metrics have been discussed.

- Throughput
- Delay

Let us analyze the performance of a network with 50 nodes using the proposed order-optimal neighbor discovery algorithm. The results are compared with that of the ALOHA-like neighbor discovery algorithm and Beacon based neighbor discovery.

Table I shows the analysis of throughput of the proposed neighbor discovery algorithm with that of those which are proposed earlier. Results show that, proposed system yields a maximum throughput of about 60%, which is more than twice that of the other.

TABLE I. THROUGHPUT ANALYSIS

	10	20	30	40	50
ALOHA	0.14	0.18	0.24	0.32	0.38
Beacon based ND	0.12	0.16	0.18	0.24	0.28
ND with collision detection	0.33	0.42	0.54	0.55	0.62

The graphical representation of the throughput analysis is shown in Fig 2.

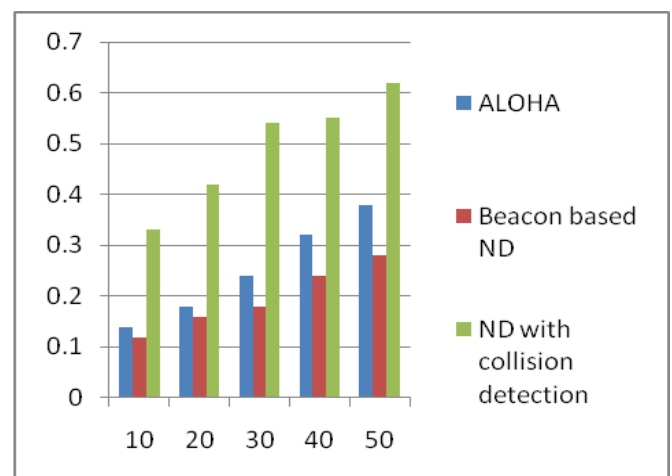


Fig 2. Throughput analysis

Table II shows the analysis of delay between the packets from source to destination. Since the earlier neighbor discovery algorithms have no collision detection mechanism, there are greater chances for collision. This results in more and more collision between the packets. Therefore, the delay between the packets is larger. The proposed system employs collision

detection mechanism and hence, the chance for collision is reduced thereby reducing the delay between the packets.

TABLE II. DELAY ANALYSIS

	10	20	30	40	50
ALOHA	6	10	16	22	30
Beacon based ND	10	14	22	32	38
ND with collision Detection	2	6	10	18	24

The graphical representation of the throughput analysis is shown in Fig 3.

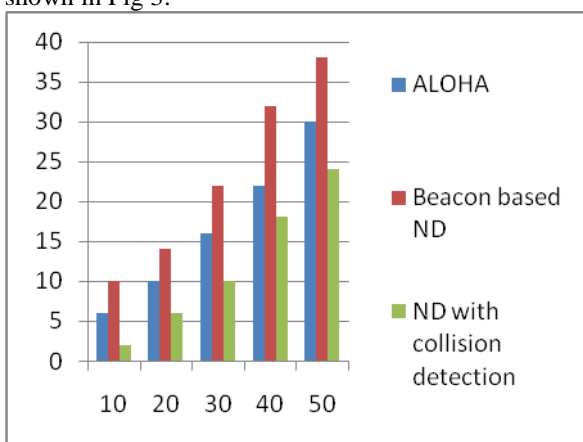


Fig 3. Delay analysis

VI. CONCLUSION

The proposed system presents an efficient neighbor discovery algorithm that exploits feedback from receiving nodes. This allows each node to determine if it has been discovered by all other neighbors in time w.h.p., which then allows it to stop transmitting. It also allows nodes to distinguish between collision and an idle slot. This algorithm can also be used when nodes cannot detect collision. The proposed system shows that, the absence of estimates of the number of neighbor's results in no more than a factor of two slowdown, compared to when nodes know n . Further it shows that lack of synchronization among nodes results in at most a factor of two slowdown in algorithm performance from the case when nodes are synchronized. Further, the proposed system analyzes the performance of these algorithms with the one which is implemented in Beacon based neighbor discovery. In dense setting where $n \sim 100$, beacon-based neighbor discovery is 65 times slower than ALOHA and 300 times slower than the collision-detection based algorithm. Thus, the proposed system works more efficient in terms of throughput and delay compared to those that are implemented earlier and comprehensively address various practical limitations of the earlier approaches.

FUTURE WORK

Analysis shows a gap between the lower and upper bounds on the running time for neighbor discovery in the network case. Clearly, the quest for an order-optimal neighbor discovery algorithm remains an intriguing prospect. Of particular interest is the question of whether the feedback-based algorithms, which are order-optimal in the single-hop case, can be extended to the multi hop network setting while outperforming the ALOHA-like algorithm. Another direction of interest is the extension of the various algorithms and the analysis presented in this paper to wireless channel models that incorporate phenomena such as fading and shadowing.

REFERENCES

- [1] S.Vasudevan, M.Adler, D. Towsley and D.Goeckel "Efficient algorithms for neighbor discovery in wireless networks," *IEEE ACM Transactions on Networking*, 2013.
- [2] W. Zeng, X. Chen, A. Russell, S. Vasudevan, B. Wang, and W. Wei, "Neighbor discovery in wireless networks with multipacket reception," in *Mobihoc'11*, 2011.
- [3] R. Khalili, D. Goeckel, D. Towsley, and A. Swami, "Neighbor discovery with reception status feedback to transmitters," in *Proc. IEEE INFOCOM*, 2010.
- [4] J. Luo and D. Guo, "Neighbor discovery in wireless ad hoc networks based on group testing," in *Proc. Annu. Allerton Conf.*, 2008.
- [5] D. Angelosante, E. Biglieri, and M. Lops, "Neighbor discovery wireless networks: A multiuser-detection approach," in *Proc. Inf. Theory Appl. Workshop*, 2007.
- [6] S. A. Borbash, A. Ephremides, and M. J. McGlynn, "An asynchronous neighbor discovery algorithm for wireless sensor networks," *Ad Hoc Netw.*, 2007.
- [7] S. Vasudevan, J. F. Kurose, and D. F. Towsley, "On neighbor discovery wireless networks with directional antennas," in *Proc. IEEE INFOCOM*, 2005.
- [8] A. Keshavarzian and E. Uysal-Biyikoglu, "Energy-efficient link assessment wireless sensor networks," in *Proc. IEEE INFOCOM*, 2004.
- [9] L. Bao and J. J. Garcia-Luna-Aceves, "A new approach to channel access scheduling for ad hoc networks," in *Proc. ACM Mobihoc*, 2001.
- [10] M. J. McGlynn and S. A. Borbash, "Birthday protocols for low energy deployment and flexible neighbor discovery ad hoc wireless networks," in *Proc. ACM Mobihoc*, 2001.
- [11] C. Law, K. Lee, and K. Y. Siu, "Efficient memoryless protocol for tag identification," in *Proc. 4th Int. Workshop Discrete Algor. Methods Mobile Comput. Commun*, 2000.