History-Aware Adaptive Backoff for Neighbor Discovery in Wireless Networks

Zimu Yuan[†], Lizhao You[⋆], Wei Li[†], Biao Chen[§], Zhiwei Xu[†]
[§]University of Macau, Macao
[†]Institute of Computing Technology, CAS, China
[⋆]State Key Laboratory for Novel Software Technology, Nanjing University, China
yuanzimu@ict.ac.cn, lzyou@smail.nju.edu.cn, liwei@ict.ac.cn, bchen@umac.mo, zxu@ict.ac.cn

***Abstract*—The ability of discovering neighboring nodes, namely neighbor discovery, is essential for the self-organization of wireless ad hoc networks. In this paper, we propose a history-aware adaptive backoff algorithm for neighbor discovery assuming collision detection and feedback mechanisms. Given successful discovery feedback, undiscovered nodes can adjust their contention window. With collision feedback and historical information, only transmission nodes enter the re-contention process, and decrease their contention window to accelerate neighbor discovery process after collision. Then, we give theoretical analysis of our algorithm on the discovery time and energy consumption, and derive the optimal size of contention windows by two rounds of optimization. Finally, we validate our theoretical analysis by simulations, and show the performance improvement over existing algorithms.**

***Index Terms*—wireless ad hoc networks; neighbor discovery; history-aware; adaptive backoff;**

I. Introduction

Neighbor discovery is a fundamental step for the initialization of wireless ad-hoc networks, and the knowledge of neighbors is essential for further operations, e.g. routing protocols [1], media access control (MAC) protocols [2]. For example in [1] the authors assume that nodes know the information about their one-hop neighbors to perform routing in multi-hop networks. In [2] MAC protocols need the two-hop neighbor list to do time slot assignment.

For a given node, the problem of neighbor discovery is to find all nodes in its transmission range by receiving packets from its neighbors. We want to find all neighbors using minimal costs, e.g. discovery time and energy consumption. However, it is not easy, because all nodes will share a channel to transmit packets. If two or more nodes transmit at the same time, a collision happens at the reception nodes. Therefore, the key to successful neighbor discovery is to solve the collision caused by simultaneous transmissions. Many algorithms [3]–[8] have been proposed for neighbor discovery to handle collisions, especially probabilistic discovery algorithms [4]–[8]. For example, [5] proposes several ALOHA-like algorithms, and transforms the neighbor discovery time analysis to the *Coupon Collectors' Problem* while coping with collisions.

Most of algorithms [4], [6], [7] consider the neighbor discovery problem without collision detection and feedback mechanisms. [5] is the first paper to show the huge improvement of discovery time with collision detection and feedback mechanisms. In [8], the authors propose a reliable energy detection mechanism in physical layer that allows receivers to detect collisions, and feedback the reception status. They also show that the discovery time is significantly smaller compared to the case where the reception status is not available. Therefore, in this paper we focus on the feedback model based on energy detection. On the other hand, most of works [4]–[6], [8] use ALOHA model for simplicity, i.e. the behavior of nodes in each time slot is independent. Instead, we consider a more realistic backoff model.

In this paper, we propose a history-aware adaptive backoff neighbor discovery algorithm. The algorithm performs in rounds. In each round, a node randomly backoffs some time slots, and then transmits determinedly. After discovering a new node, the contention window is different. In other words, our algorithm uses *adaptive backoff* mechanism. Furthermore, in each round, if a collision happens due to the same selected slot, we utilize *historical information* (transmission or reception in the collision slot) to divide nodes into different states, and re-contention only happens among those transmission nodes. We also decrease the contention window after the first collision to accelerate the process of neighbor discovery. To the best of our knowledge, the concepts of *adaptive backoff* and *history-aware* for probabilistic neighbor discovery have not been studied before under feedback scenarios. The main contribution of this paper can be summarized as follows:

- We propose a history-aware adaptive backoff neighbor discovery algorithm, and conduct two rounds of optimization to get the optimal parameters by theoretical analysis.
- We show through simulations that our algorithm allows nodes to discover their neighbors much faster than ALOHA-like algorithms, consuming less energy.

The rest of the paper is organized as follows. Section II introduces the model and assumptions. Section III describes our algorithm. Section IV gives a detailed analysis. Then we show our simulation results in Section V. The related work is given in Section VI. Finally, we conclude and give the future work in Section VII.

978-0-7695-4610-0/11 $26.00 © 2011 IEEE
DOI 10.1109/MSN.2011.38

II. Network Model and Assumptions

In wireless networks, nodes are distinguished by unique identifiers, like MAC address, location of the node, etc. Nodes exchange the unique identifiers for neighbor discovery by sending control packets. The problem of neighbor discover for a node is to find its neighbors' identifiers with minimal costs, e.g. discovery time or energy consumption. Each node is equipped with a transceiver that allow a node either transmit or receive packets, but not simultaneously, i.e. not full-duplex. Two nodes are neighbors if they are in the communication range of each other. For the sake of simplicity, we assume errors caused by fading are negligible. In other words, if packets are transmitted without collision, they must be received correctly. We also assume time is slotted with equal-length, and all nodes are synchronized on slot boundary. Both assumptions are adopted by [4], [6], [8].

Then, nodes have collision detection capability, i.e. the receiver can tell collision from successful reception. Also, there is a feedback mechanism to transmitters, which is proposed by [5], [8]: each time slot is further divided into two sub-slots (namely *transmission* sub-slot and *feedback* sub-slot). If receivers cannot decode the packets transmitted in the first sub-slot, they reply a small packet in the feedback slot to notice transmitters. If the transmission node in the first sub-slot detects energy in the second sub-slot, it assumes that there is a collision. Otherwise, it assumes that the transmission is successful. Note that [8] gives a detailed physical layer design about the collision detection and feedback mechanism based on energy detection.

III. History-aware Adaptive Backoff Algorithm

In this section, we first give the basic idea of our algorithm, and then describe the algorithm. To show the key idea, we first consider the algorithms in a clique with number of nodes n known beforehand.

A. Basic Idea

Before describing our algorithm, we give two definitions: *round* and *phase*. A *round* starts when all active nodes participate in the backoff, which is the basic unit of neighbor discovery for nodes. We call a round finished if there is a collision or a successful transmission. Note that the length of a round must be smaller or equal to contention window W, since the channel can not be idle within a contention window. The time to discover a new node is called *phase*, which may be composed of many rounds. If there is a successful transmission, we call a round finished, and a phase also ends.

Our algorithm includes two key components: *adaptive back-off* and *history-aware* collision resolution. Let j-th phase denote the process of discovering j-th nodes, W_j be the initial contention window of j-th phase, and W'_j be the re-contention window of j-th phase, where $j \in [1, n]$. Suppose we are in the j-th phase, all nodes first randomly choose a time slot S from $[1, W_j]$ to transmit. Before their transmission, they just listen to the channel. At each time slot, based on the feedback information, the algorithm behaves as follows:

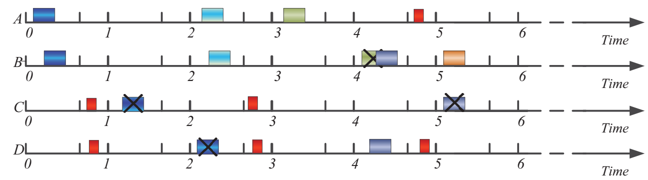

Fig. 1. Illustration of Our Algorithm

- If there is a successful transmission feedback, then the successful node is discovered by all other nodes. It becomes silent, and quits the neighbor discovery transmission, just listening. All remaining nodes start a new round with a new contention window W_{j+1}. This is called *adaptive backoff*.
- If there is a collision feedback, then all transmission nodes in the collision slot enter a re-contention round with a new contention window W'_j while other reception nodes quit the re-contention. The process goes on until a node is discovered. This is called *history-aware* collision resolution because the node partition is according to the historical information.

The optimal value of W_j and W'_j will be obtained by theoretical analysis to make the best trade-off.

We illustrate our key idea by Figure 1 to make our algorithm easy to understand. Take nodes (A, B, C, D) for example, all of them are in a clique, which means that they are in transmission range of each other. For the sake of simplicity, we assume the contention window $W_1 = W_2 = W'_1 = 3$. At the beginning, all nodes choose $S \in [1, W_1]$ to backoff. Here both node A and B choose slot 0 to transmit, C chooses slot 1, and D chooses slot 2. At the first sub-slot of the first slot, obviously a collision occurs; node (C, D) cannot decode the error packets and feedback an error message at the second sub-slot. Then, node (A, B) know that there is a collision, and (C, D) cancel the scheduled transmission. (A, B) start a new round in slot 1. In the new round, only A and B participate in the contention, choosing a backoff in $[1, W']$. Both choose slot 2 to transmit, and a collision happens again. A new round starts in slot 3, with A chooses slot 3, and B chooses slot 4. Then A transmits successfully. B cancels the scheduled transmission, and begins a new round with node (C, D) with window $[1, W_2]$. The process goes on until all nodes are discovered.

B. Formal Description

We use state transform diagram to describe our algorithm, including states and events. Given the feedback model with two sub-slots in one time slot, we classify three types of events according to the time events happened: before a time slot, at the first sub-slot and at the second sub-slot. Suppose there is j nodes undiscovered now. The meaning of these events are explained as follows.

Segment 1: Before a time slot

At this time, following events may happen to a node:

E_{0r}: a node needs to re-decide the random back-off slots W from $[0, W_j - 1]$;

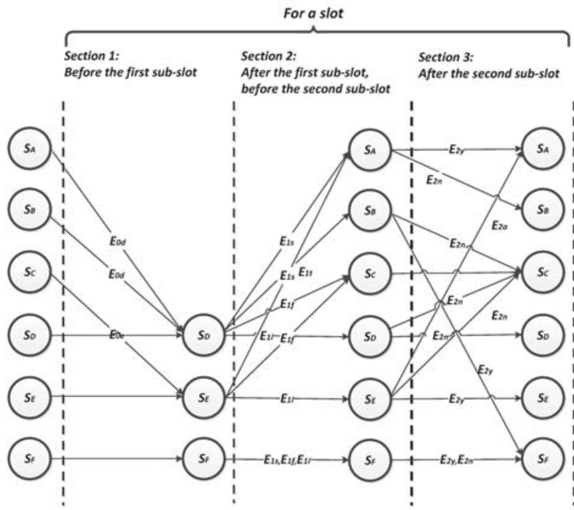

Fig. 2. State Transition Diagram of Our Algorithm

E_{0s}: a node needs to re-decide the random back-off slots W from $[0, W_j^{'} - 1]$;

E_{0d}: count down the value of W for transmission;

E_{0e}: increase the value of W to delay transmission;

E_{idle}: do nothing.

Segment 2: At the first sub-slot

Here, reception nodes will get packets from transmission nodes. Packets may be correctly decoded or not. Thus, three types of events may happen for a reception node:

E_{1s}: receive packets successfully;

E_{1f}: fail to receive packets;

E_{1i}: the channel is idle.

For a transmission node, following events may happen:

E_{1t}: transmitting at the first sub-slot.

Segment 3: At the second sub-slot

At the second sub-slot, following events may happen to a node who senses the channel:

E_{2y}: not receive collision feedback;

E_{2n}: receive collision feedback;

E_{2t}: transmit a collision feedback;

E_{2a}: choose a new backoff slot.

Next we will present the states of nodes at different time, which is critical for understanding our algorithms. There are five states in each segment. Note that each slot can be divided into three segments. At the beginning of neighbor discovery, all nodes are in

- S_A: node i needs to decide (or re-decide) the random backoff slots W from $[0, W_j - 1]$.

If a node transmits successfully in current slot, it changes state to

- S_F: node i has been discovered by all its neighbors. In this state, node i keeps in silent mode for the rest of time, which means node i just senses the channel to discover new neighbors.

Other nodes change to S_A state to begin the new time slot. If there is a collision, based on the feedback information, nodes can classify themselves into transmission nodes and reception nodes. For those transmission nodes, who transmitted a packet at the last first sub-slot, their state changes to

- S_B: node i needs to re-decide the random back-off slots W from $[0, W_j^{'} - 1]$, the same behaviour as the random back-off algorithm depicted in above part.

For those reception nodes, who received a collision packet at the last first sub-slot, their states change to

- S_C: node i chooses to delay $W_j^{'}$ slots to avoid potential collision.

Note that in a network, there exists some nodes who are two-hop away from transmitters. These nodes can detect feedback signal at the last second sub-slot. In order to prevent the potential transmission, we also make these nodes in state S_C.

Then if the current slot is idle, for those backoff nodes, they change to state

- S_D: it is a special state for those backoff nodes in a new round. If $W = 1$, that means node i intends to transmit at this slot. Otherwise, $W = W - 1$.

Compared with those backoff nodes, silent nodes have different actions. In state S_C, they change the backoff period W to $-W_j^{'}$. Therefore, they need to increase W to realize the delay action. So the new state is

- S_E: it is a special state for those silent nodes in a new round. If $W = -1$, it means the silent period of $W_j^{'}$ will end in the next slot. Otherwise, $W = W + 1$.

For nodes with different states, different events will bring them to different states. However, some events only happen on some states, e.g. S_A with E_{0r}. The valid combinations and transition are shown in Figure 2. Due to the space limitation, we omit the detailed explanation, which can be found in [12]. Based on this state transition diagram, we present our history-based random backoff algorithm in Algorithm 1.

Distributed Implementation: We formally present the History-Aware Adaptive Backoff Algorithm in clique above. This algorithm can be easily applied to network environment and unknown number of neighbors. We only need to remove the **while** loop of line 5 and line 28, and add a time waiting process. For example, when a node is set to state S_F, it keeps on reception status for a period. If the node finds channel just remaining idle in this period, e.g. a contention window W, it terminates the neighbor discovery process. When all nodes finish their discovery process, the neighbor discovery for the whole network also ends.

IV. Theoretical Analysis

This section gives a theoretical analysis of our algorithm. The metric is discovery time and energy consumption, two important concern of neighbor discovery. In the following parts, we only take discovery time for example. The discussion on energy consumption can be found in [12]. In order to

Algorithm 1: History-Aware Adaptive Back-off Algorithm

Input: a clique of n nodes; for phase j, contention window W_j, re-contention window $W_j^{'}$
Output: time slots for node i to discover all its neighbors
define $k = n$ as the number of neighbors undiscovered
define $T_i = 0$ as the current time slot
define $S_i = S_A$ as the current state
define $W = 0$ as the transmission slots for each round
while $k > 0$ **do**
    $T_i = T_i + 1$
    if $S_i = S_A$ **then** $W \in [1, W_{n-k}]$; $S_i = S_D$
    elif $S_i = S_B$ **then** $W \in [1, W_{n-k}^{'}]$; $S_i = S_D$
    elif $S_i = S_C$ **then** $W = -W_{n-k}^{'}$; $S_i = S_E$
    elif $S_i = S_D$ **then** $W = W - 1$
    elif $S_i = S_E$ **then** $W = W + 1$
    end if
    Node i transmits or receives at the first sub-slot
    if $W \neq 1$ and packets fail received **then** $S_i = S_C$
    elif $W \neq 1$ and packets successfully received **then**
        $k = k - 1$
        if $S_i \neq S_F$ **then** $S_i = S_A$ **end if**
    end if
    if node i receives error packets at the first sub-slot, node i feedbacks an error status at the second sub-slot
    if $W \neq 1$ and detect energy at the second sub-slot **then**
        $S_i = S_C$
    elif $W = 1$ and detect energy at the second sub-slot **then**
        $S_i = S_B$
    elif $W = 1$ and not detect energy at the second sub-slot **then**
        $S_i = S_F$
    elif $W = 0$ and $S_i = S_E$ **then** $S_i = S_A$
    end if
end while
return T_i

show the key idea, we give the analysis under a single clique of n nodes, and verify the theoretical results under network scenario in simulations. Moreover, we will make "first-round" optimization on contention window W to achieve adaptive backoff. We further conduct a "second-round" optimization on further contention window $W^{'}$ to utilize historical information to reduce the collisions. Note that the optimal value depends on phase, i.e. for phase j we have W_j and $W_j^{'}$. The important notations used in this section are summarized in Table I.

A. Analysis on Adaptive Backoff Approach

Let S_j be the state that there are j nodes undiscovered. In each round, there are two states: a node transmits successfully, or there is a collision. Then, the former probability can be denoted by $P_{j,j-1}$, and the latter $P_{j,j}$. Let X_j be the time slots needed to discover all j nodes. According to the conditional expectation formula, we have

$$E[X_j] = E[E[X_j|Y]] = P_{sc}E[X_j|Y_{sc}] + P_{fc}E[X_j|Y_{fc}] \quad (1)$$

where P_{sc} denotes the probability that current round is successful, and P_{fc} denotes the collision probability. Actually, $P_{sc} = P_{j,j-1}$ and $P_{fc} = P_{j,j}$.

For successful round, we can divide X_j into X_{j-1} and T_j', where T_j' denotes the time slots spent in current round. For collision round, X_j can be divided into X_j' and T_j', where X_j' denotes the time slots spent on discovering j nodes after

TABLE I
TERMINOLOGY

Symbol	Definition
n	Number of nodes in a clique.
X_j	Time slots needed to discover j nodes.
S_j	State that there are j nodes undiscovered in current slot.
W_j	Initial contention window for S_j.
$W_j^{'}$	The contention window after first collision for S_j.
Y_{sc}	Event that some node transmits successfully in current round.
Y_{fc}	Event that there is a collision in current round.
$P_{j,j}$	The probability that there is a collision in current round for S_j, $Pr[Y_{fc}] = P_{j,j}$.
$P_{j,j-1}$	The probability that some node transmits successfully in current round for S_j, $Pr[Y_{sc}] = P_{j,j-1}$.
P_j	The probability that nodes choose a slot to transmit when there are j nodes undiscovered.
T_j	The expected time slots from S_j to S_{j-1}.
$T_j^{'}$	Time slots spent on current round.
Tx	Transmission energy consumption in the first sub-slot.
Rx	Reception energy consumption in the first sub-slot.
Ax	Feedback transmission energy consumption in the second sub-slot.
Bx	Feedback reception energy consumption in the second sub-slot.
Sh_m	In previous round, m nodes transmit, esp. $Sh_{j+1} = S_j$.
$Ph_{m,k}$	The transfer probability that from state Sh_m to state Sh_k. In other words, m nodes participate in the round, while only k nodes transmit.
Th_m	The expected time cost from Sh_m to the other states.
Eh_m	The expected time cost needed for a node to finish the current phase from state Sh_m, esp. $Eh_{j+1} = T_j$.

current collision round. Therefore, we have

$$E[X_j|Y_{sc}] = E[(X_{j-1} + T_j')|Y_{sc}] = E[X_{j-1}|Y_{sc}] + E[T_j^{'}|Y_{sc}]$$

$$E[X_j|Y_{fc}] = E[(X_j^{'} + T_j')|Y_{fc}] = E[X_j^{'}|Y_{fc}] + E[T_j^{'}|Y_{fc}]$$

Notice that the current round situation is independent of previous rounds, so $E[X_{j-1}|Y_{sc}] = E[X_{j-1}]$ and $E[X_j^{'}|Y_{fc}] = E[X_j]$. Then

$$E[X_j|Y_{sc}] = E[X_{j-1}] + E[T_j^{'}|Y_{sc}] \quad (2)$$

$$E[X_j|Y_{fc}] = E[X_j] + E[T_j^{'}|Y_{fc}] \quad (3)$$

With equation(2)(3), we can rewrite equation(1) as

$$E[X_j] = E[X_{j-1}] + \frac{P_{sc}E[T_j^{'}|Y_{sc}] + P_{fc}E[T_j^{'}|Y_{fc}]}{P_{sc}}$$

For S_j, $P_{sc} = P_{j,j-1}$, $P_{fc} = P_{j,j}$, and

$$P_{j,j-1} + P_{j,j} = 1 \quad (4)$$

We can define

$$T_j = \frac{P_{j,j-1}E[T_j^{'}|Y_{j,j-1}] + P_{j,j}E[T_j^{'}|Y_{j,j}]}{P_{j,j-1}} \quad (5)$$

Obviously T_j is the expectation time cost from state S_j to S_{j-1}, i.e. the expectation time of phase j. Note that $E[X_0] = 0$, so the expectation of overall discovery time is

$$E[X_n] = \sum_{j=1}^{n} T_j \quad (6)$$

If we can minimize T_j for each j, we can minimize $E[X_n]$. So we try to derive a formula about T_j.

Note that the current contention window is W_j for phase S_j. In each time slot the clique has only three states: idle; collision; successful transmission. Moreover, if there is a collision or a successful transmission in slot i, it means that the channel is idle in the past $i-1$ slots, since the collision or successful transmission leads to the end of current phase.

For successful transmission state, it can happen in slot i with probability $\frac{j}{W_j}(1-\frac{i}{W_j})^{j-1}$, where $(1-\frac{i}{W_j})^{j-1}$ means $j-1$ nodes never transmit in all i slots. So we have

$$P_{j,j-1} = \sum_{i=1}^{W_j} \frac{j}{W_j}(1-\frac{i}{W_j})^{j-1} \tag{7}$$

Then we have

$$\begin{aligned} P_{j,j-1}E[T_j|Y_{j,j-1}] &= P_{j,j-1}\sum iP(T_j=i|Y_{j,j-1}) \\ = \sum iP(T_j=i,Y_{j,j-1}) &= \sum_{i=1}^{W_j}\frac{ij}{W_j}(1-\frac{i}{W_j})^{j-1} \end{aligned} \tag{8}$$

After a similar analysis, for collision slot, we have

$$\begin{aligned} &P_{j,j}E[T_j|Y_{j,j}] \\ &= \sum_{i=1}^{W_j} i\left((1-\frac{(i-1)}{W_j})^j - (1-\frac{i}{W_j})^j - \frac{j}{W_j}(1-\frac{i}{W_j})^{j-1}\right) \end{aligned} \tag{9}$$

where $(1-(i-1)/W_j)^j$ denotes the probability that the channel is idle in the past $i-1$ slots, $(1-i/W_j)^j$ means that the channel is idle for al i slots, and the last one means only one node transmits at slot i.

Combining equation (5)(6)(8)(9), we have

$$E[X_n] = \sum_{j=1}^{n} T_j = \sum_{j=1}^{n} \frac{\sum_{i=1}^{W_j} i^j}{j\sum_{i=1}^{W_j-1} i^{j-1}} \tag{10}$$

If we can minimize T_j for each j, we can minimize $E[X_n]$. Note that it is not easy to derive a close-form formula for $E[X_n]$ if we do not make some approximation. It is obviously that approximation can lose some accuracy, so instead we keep this form and use numerical calculation to derive the optimal W_j for each T_j in the simulations.

B. Analysis on History-Aware Approach

In order to accelerate the process in a phase, we use the history-aware approach to handle collisions. Thus, we need a "second-round" optimization to get the new contention window $W_j^{'}$ for S_j. Our algorithm improves the process of backoff in a single phase. Therefore, we only need to consider the situation from S_j to S_{j-1}. In order to present the procedure clearly, a state transition diagram is given in Figure 3.

Let us consider S_j. Let Sh_j be the state that j nodes transmit in previous round, and Sh_{j+1} is the starting state of S_j. Note that Sh_1 means only one node transmits, and the phase terminates definitely. Let Eh_m be the expected

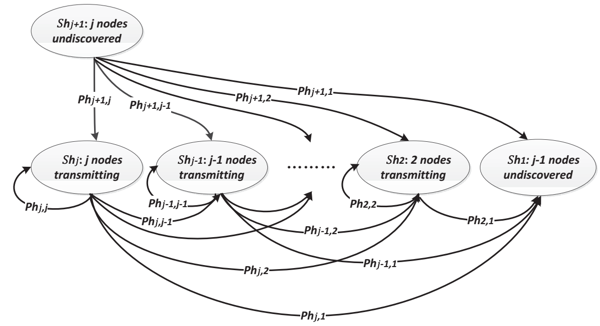

Fig. 3. State Transition Diagram from S_j to S_{j-1}

time slots needed for a node to finish the current phase from state Sh_m to state Sh_1. Note that Eh_{j+1} is T_j. We need to minimize Eh_{j+1} to get the optimal W_j'.

At the beginning of state Sh_{j+1}, we use the contention window W_j with transmission probability in single slot $P_j = 1/W_j$ obtained from the solution of equation (10). When a collision happen, Sh_{j+1} exactly transfers to the state Sh_2 - Sh_j. After the first collision, we need to decrease the number of potential transmission nodes and reduce the contention window from W_j to W_j'. Now the transmission probability in single slot is $P_j' = 1/W_j'$. When successful transmission, states transfer to the state Sh_1, there are remaining $j-1$ nodes undiscovered in clique. The current phase ends. According to the conditional expectation formula, we have

$$Eh_m = \begin{cases} \sum_{k=1}^{j+1} Ph_{j+1,k}Eh_k + Th_{j+1} & \text{if } m = j+1 \\ \sum_{k=1}^{m} Ph_{m,k}Eh_k + Th_m & \text{else } 2 \le m \le j \end{cases} \tag{11}$$

where $Ph_{m,k}$ here denotes the transfer probability from Sh_m to Sh_k, Th_{j+1}, and Th_m denote the expected time cost from Sh_m to any other states.

For the case k nodes transmit simultaneously at slot i with m nodes intending to transmit during this round, the probability can be expressed as $\binom{m}{k}{P_j'}^k(1-iP_j')^{m-k}$. Therefore, the transfer probability $Ph_{m,k}$ from state Sh_m to state Sh_k can be represented as

$$Ph_{m,k} = \begin{cases} \sum_{i=1}^{W_j}\binom{j}{k}P_j^k(1-iP_j)^{j-k} & \text{if } m = j+1 \\ \sum_{i=1}^{W_j'}\binom{m}{k}{P'}_j^k(1-i{P'}_j)^{m-k} & \text{if } 2 \le m \le j \\ 0 & \text{otherwise} \end{cases} \tag{12}$$

The total time cost Th_m transferred out of node m can be

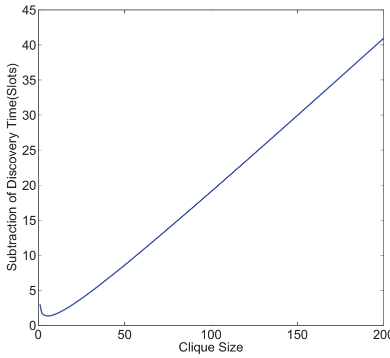

Fig. 4. Theoretical Discovery Time in Cliques

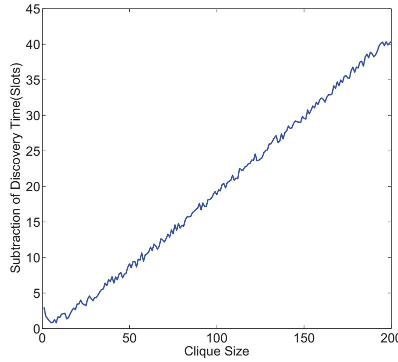

Fig. 5. Simulation Discovery Time in Cliques

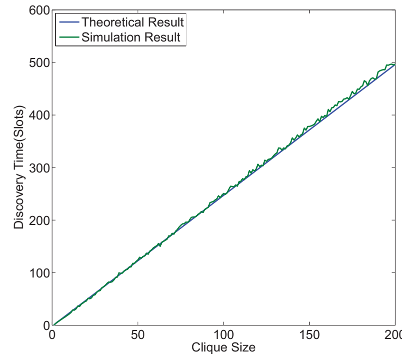

Fig. 6. Theoretical and Simulation Time

represented as

$$Th_m = \begin{cases} \sum_{k=1}^{j} \sum_{i=1}^{W_j} i \binom{j}{k} P_j^k (1 - iP_j)^{j-k} & \text{if } m = j+1 \\ \sum_{k=1}^{m} \sum_{i=1}^{W_j'} i \binom{m}{k} {P'}_j^k (1 - i{P'}_j)^{m-k} & \text{if } 2 \le m \le j \\ 0 & \text{otherwise} \end{cases} \quad (13)$$

We can rewrite equation (11) as $PX = 0$, where

$$P = \begin{bmatrix} Ph_{1,1} - 1 & 0 & ... & 0 & 0 & Th_1 \\ Ph_{2,1} & Ph_{2,2} - 1 & ... & 0 & 0 & Th_2 \\ Ph_{3,1} & Ph_{3,2} & ... & 0 & 0 & Th_3 \\ ... & ... & ... & ... & ... & ... \\ ... & ... & ... & ... & ... & ... \\ Ph_{j,1} & Ph_{j,2} & ... & Ph_{j,j} - 1 & 0 & Th_j \\ Ph_{j+1,1} & Ph_{j+1,2} & ... & Ph_{j+1,j} & -1 & Th_{j+1} \end{bmatrix} \quad (14)$$

$$X = \begin{bmatrix} Eh_1 & Eh_2 & ... & ... & Eh_{j+1} & 1 \end{bmatrix} \quad (15)$$

Note that negative 1 is got from the right side of equation of (11) in the matrix. Because $Eh_1 = 0$, by solving the linear equations $PX = 0$, we can get the expression of Eh_{j+1} (exactly T_j defined in Table I), and derive the optimal backoff window W_j' after collision happened. We minimize Eh_{j+1} (or T_j) for each j nodes undiscovered case to get optimal W_j'. Then, we calculate the minimum $E[X_n]$ as.

$$E[X_n] = \sum_{j=1}^{n} T_j = \sum_{j=1}^{n} Eh_{j+1} \quad (16)$$

Note that Sh_m, $Ph_{m,k}$, Eh_m are defined in each phase, which are separated across different S_j. Due to the same reason, we use numerical calculation and find that W_j' usually has very small values, e.g. 3, 4, 5. by simulations.

V. Performance Evaluation

A. Simulation Setup

Our simulation environment includes two types of setting: clique and network. In a clique of size n, nodes are neighbors mutually. In network setting, we generate a 3km×3km region, and nodes are randomly distributed with the transmission range 150m. Average number of neighbors per node is set from 1 to 30 so as to view different performance of algorithms. The number of nodes in a network is determined by the average neighbors, and we run 30 times for each average neighbors in $[1, 30]$ to validate the results for different node placement.

We compare our algorithm with ALOHA-with-feedback [5] [8] algorithm. In [5] [8], the advantage of ALOHA-with-feedback algorithm is shown when it is compared with other algorithms without feedback, so we will not compare our algorithm with those algorithms here, e.g. ALOHA-like algorithm.

B. Simulation Results

1) Evaluation In Cliques: It is hard to give the theoretical comparison of the neighbor discovery time between our history-aware adaptive backoff algorithm and ALOHA-with-feedback algorithm. Thus, for different clique size, we calculate the expectation of neighbor discovery time by our algorithm (denoted $E_0(x)$) based on the theoretical analysis of section IV, and compare it with ALOHA-with-feedback [5] [8] (denoted $E_1(x)$). Figure 4 depicts the result of $E_1(x) - E_0(x)$ as the clique size changes. The value of $E_1(x) - E_0(x)$ is always greater than 0, that is, history-aware adaptive backoff algorithm algorithm need shorter time to accomplish neighbor discovery than ALOHA-with-feedback. As we can see in Figure 4, the value of $E_1(x) - E_0(x)$ becomes larger when the clique size is increasing from 2 to 200. In addition, the clique size of 200 is large enough when considering the realistic network. The theoretical result shows our algorithm gets better performance than ALOHA-with-feedback.

We implement both our history-aware adaptive backoff algorithm and ALOHA-with-feedback algorithm, and run the two algorithms 30 times for each clique size in $[2, 200]$. The average neighbor discovery time by our algorithm (denoted $A_0(x)$) has almost the same value with the theoretical result $E_0(x)$. Also, The average discovery time using ALOHA-with-feedback (denoted $A_1(x)$) is almost the same with $E_1(x)$. As Figure 5 shows, the result of $A_1(x) - A_0(x)$ is nearly as $E_1(x) - E_0(x)$ shown in Figure 4 at clique size range $[2, 200]$.

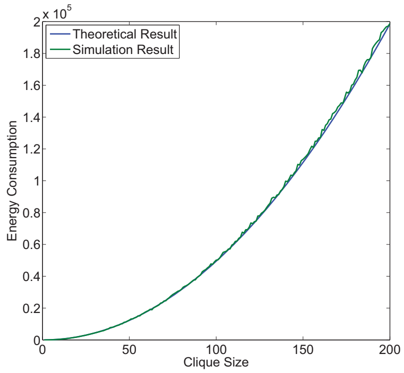

Fig. 7. Theoretical and Simulation Energy

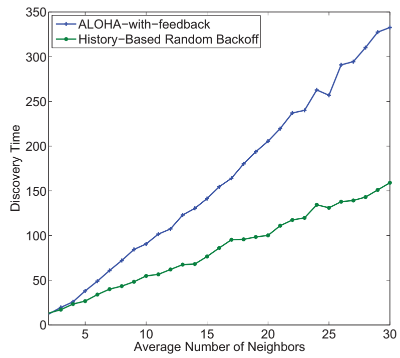

Fig. 8. Discovery Time in Networks

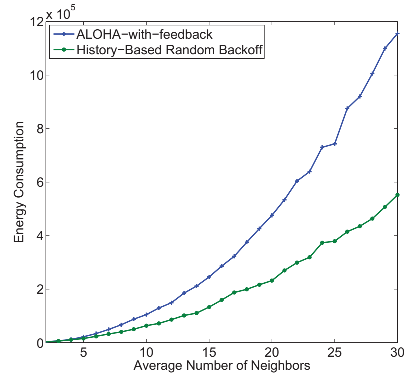

Fig. 9. Energy Consumption in Networks

The simulation result confirms our theoretical analysis. Furthermore, Figure 6 and Figure 7 show that our analysis is close to simulation results for our algorithm, both on discovery time and energy consumption.

The analysis results of the clique can also be used to give the average neighbor discovery time and average energy cost for nodes in a network. As we simulate networks with different densities, the results is close to our analysis. It means the clique analysis can be applied to a network, which has been verified by [5].

2) Evaluation In Networks: We also evaluate the performance of history-aware adaptive backoff and ALOHA-with-feedback under the network environment. In the network, nodes have no idea of their neighbors immediately after deployed, therefore nodes can't decide the optimal transmitting probability. In fact, we use average number of neighbors (denoted N) of the whole network for the initial transmitting probability setting. For ALOHA-with-feedback, we set the initial transmitting probability (denoted P_1) to $1/(N+1)$, the optimal probability in the clique analysis [5]. When collisions happen, decrease transmitting probability from P_1 to $P_1/(1+P_1)$; when the channel is idle, increase transmitting probability from P_1 to $P_1/(1-P_1)$. This setting helps ALOHA-with-feedback algorithm get better performance. For history-aware adaptive backoff, we set the initial contention window for a node to $W = N+1$, namely the backoff slot chosen probability $P_2 = 1/W = 1/(N+1)$; we set reduced contention window for a node to $W^{'} = 3$ when the node was interfered by collision, namely set $P_2^{'} = 1/3$. Note that $W = N+1$ is the optimal backoff window through our analysis. $W^{'}$ usually has very small values stated in Section IV, so we simply set $W^{'} = 3$ here. The value of P_2, $P_2^{'}$ is constant, no matter a collision happens, nor the channel is idle. This setting is adverse to the performance of history-aware adaptive backoff. However, our backoff algorithm gets much better performance than ALOHA-with-feedback despite this adverse settings.

Figure 8 depicts neighbor discovery time needed for network. Figure 9 shows energy consumption of neighbor discovery phase for networks. For the energy consumption, we simply set Ax:Bx:Tx:Rx=1:1:1:1 (Ax, Bx, Tx, Rx defined in Table I), which can be adjusted by more accurate power assumption model. We run 30 times for each N (namely density). As we can observe in Figure 8, neighbor discovery with ALOHA-with-feedback endures severe performance degradation when N increase, while neighbor discovery using history-aware adaptive backoff algorithm maintains a stable performance with less degradation. Different from clique, collisions do not just involve nodes are neighbors for each other, there are also existing interferences between nodes having common neighbors as we mentioned earlier. Therefore, more intensive collisions happen in network. ALOHA-with-feedback gets worse performance in this situation, while history-aware adaptive backoff utilizes historical collisions information to divide nodes into different state, thus avoiding the future collision. In Figure 9, we can get intuitive view. For example, when $N \geq 10$, neighbor discovery with history-aware adaptive backoff saves more than 40% energy compared to the ALOHA-with-feedback for each N. When $N \geq 20$, save more than 50% energy.

VI. Related Work

Many algorithms [3]–[8] have been proposed for neighbor discovery. [3] presents a deterministic algorithm, which depends on additional orthogonal codes. Here we focus on probabilistic algorithms because they are simple and do not need additional information. [4] is the first paper to design an efficient probabilistic algorithm for neighbor discovery. [5] considers the neighbor discovery time given an ALOHA-like algorithm. The slotted algorithm can be extended to asynchronous scenario and unknown number of neighbors, which can be utilized to extend our work. [6] extends the protocol to the scenario of duty-cycle wireless sensor networks, and analyzes the discovery time. [7] analyzes the neighbor discovery time in wireless networks with multi-packet reception. All works [4]–[7] mainly consider algorithms without feedback information. [8] further discusses the feedback mechanism from the view of physical layer, and present an improved ALOHA-with-feedback algorithm. Our work can be viewed as a further exploration and an improvement based on the feedback model. We utilize the collision feedback information and historical behavior (transmission or reception in collision slot) to handle collisions, and thus improve the neighbor discovery efficiency, i.e. using less time and less energy.

Contention resolution algorithms [9]–[11] in MAC protocol design also use contention window adjustment (or adaptive backoff) technique. However, the goal is different. Contention resolution protocols often focus on finding algorithms to achieve throughput-optimal or delay-optimal given certain arrival rate and queue model, while neighbor discovery only wants to send one successful packet for each node and to optimize the overall discovery time or energy consumption. Hence, some nodes can quit the contention in our algorithms, and never transmit after their discovery. Moreover, we use a history-aware approach to resolve collision, which is not considered by contention resolution algorithms.

VII. Conclusion and Future Work

In this paper, we propose a history-aware adaptive backoff neighbor discovery algorithm under collision detection and feedback model. We also give a theoretically analysis of our algorithm on neighbor discovery time and energy consumption, and evaluate our algorithm through simulations compared with existing algorithm. In the future, we want to complete our work by considering unreliable channel models, node crash, asynchronous scenarios.

Acknowledgment

This work is supported in part by the National Basic Research (973) Program of China (Grant No.2011CB302502, 2011CB302803), and the major national science and technology special projects (Grant No. 2010ZX03004-003-03).

References

[1] D. B. Johnson and D. A. Maltz. Dynamic source routing in ad hoc wireless networks. In Mobile Computing, pages 153-181. Kluwer Academic Publishers, 1996.

[2] I. Rhee, A. Warrier, M. Aia, J. Min, and M.L. Sichitiu, "Z-MAC: a hybrid MAC for wireless sensor networks", presented at IEEE/ACM Trans. Netw., 2008, pp.511-524.

[3] A. Keshavarzian and E. Uysal-Biyikoglu, "Energy-efficient Link Assessment in Wireless Sensor Networks", in Proc. INFOCOM, 2004.

[4] M. McGlynn and S. Borbash. "Birthday protocols for low energy deployment and flexible neighbor discovery in ad hoc wireless networks", In Proc. MOBIHOC, 2001.

[5] S. Vasudevan, D.F. Towsley, D. Goeckel, and R. Khalili, "Neighbor discovery in wireless networks and the coupon collector's problem", in Proc. MOBICOM, 2009, pp.181-192.

[6] L. You, Z. Yuan, P. Yang, and G. Chen, "ALOHA-like Neighbor Discovery in Low-Duty-Cycle Wireless Sensor Networks", in Proc. WCNC, 2011, pp.749-754.

[7] W. Zeng, X. Chen, A. Russell, S. Vasudevan, B. Wang and W. Wei, "Neighbor Discovery in Wireless Networks with Multipacket Reception", in Proc. MOBIHOC, 2011.

[8] R. Khalili, D. Goeckel, D.F. Towsley, and A. Swami, "Neighbor Discovery with Reception Status Feedback to Transmitters", in Proc. INFOCOM, 2010, pp.2375-2383.

[9] S. Rajagopalan, D. Shah, and J. Shin, "Network adiabatic theorem: an efficient randomized protocol for contention resolution", in Proc. SIGMETRICS/Performance, 2009, pp.133-144.

[10] F. Cal, M. Conti, and E. Gregori, IEEE 802.11 protocol: Design and performance evaluation of an adaptive back-off mechanism, IEEE Journal on Selected Areas in Communications, Vol. 18, No. 9, pp. 1774-1786, 2000.

[11] Q. Xia, M. Hamdi, "Contention window adjustment for IEEE 802.11 WLANs: a control-theoretic approach", in Proc. ICC, 2006, pp.3923-3928.

[12] Z. Yuan, L. You, W. Li and Z. Xu, History-aware neighbor discovery algorithm for wireless networks, Technical Report, Institute of Computing Technology, CAS, 2010.