Research on Neighbor Discovery Latency of Low Duty Cycle Ad Hoc Network

Zhipeng Jia, Zhiqing Huang, Xiongye Su, Mengjia Li, and Huan Zhang

***Abstract*—Neighbor discovery is the key to achieve network connectivity in ad hoc networks. Wireless nodes usually work in low duty cycle state to prolong the node life cycle, but it will increase the latency of the neighbor discovery. The discovery latency of nodes with low duty cycle is the focus of this study. This paper analyzes four typical asynchronous neighbor discovery algorithms: Birthday, Disco, U-Connect, and SearchLight, and compare the discovery latency of four algorithms in simulation platform.**

***Index Terms*—Ad hoc network, neighbor discovery, discovery latency, duty cycle.**

I. Introduction

With the rapid development of portable computing platforms and wireless nodes, ad hoc network [1], [2] have received more and more attention. The nodes of ad hoc network usually rely on batteries to provide power, but due to the slow advancement of battery technology, power consumption in wireless networks remains a bottleneck to limit wide applications of ad hoc networks. The nodes work in low duty cycle is a method to reduce power consumption in ad hoc networks. This requires the nodes can still achieve neighbor discovery in low duty cycle. The neighbor discovery is divided into synchronous neighbor discovery and asynchronous neighbor discovery that synchronous neighbor discovery requires complex clock synchronization, so asynchronous neighbor discovery algorithm has become the focus of research.

Asynchronous neighbor discovery algorithm refers to awaken and sleep period to each node in the network is not unified. This kind of algorithm in practical application is bigger, the value of application of lower cost, need not to consider the problem of clock synchronization. At present, the classical asynchronous neighbor discovery algorithm is probabilistic algorithm Birthday, deterministic algorithm Disco, U-Connect and SearchLight. All of them support both symmetric and asymmetric networks. Discovery latency as a performance indicator for the neighbor discovery, we simulate four algorithms and analyze the results of CDF of discovery latency, duty cycle and the average discovery latency.

Manuscript received October 16, 2015; revised May 21, 2016. This work was supported in part by the Beijing University of Technology Fundamental Research Funding Project (No.025000514314004).

Zhipeng Jia, Zhiqing Huang, Xiongye Su, and Mengjia Li are with the School of Software Engineering at Beijing University of Technology, Chaoyang District, Beijing China, 100124 (e-mail: s201425029@emails.bjut.edu.cn, zqhuang@bjut.edu.cn, 417984021@qq.com, 15011009295@163.com).

Huan Zhang is with the School of Software Technology at Zhejiang University, Hangzhou Zhejiang China, 310058 (e-mail: lumiahuan@163.com).

II. Asynchronous Neighbor Discovery Algorithm

A. Birthday

The mathematical principle of Birthday algorithm [3] is Birthday Paradox [4]. It is a probabilistic algorithm that supports both symmetric and asymmetric networks. Over a period of n slots, the two wireless nodes are independent and randomly select k slots to transmit messages to another node, and the another node uses only the *k* slots to receive messages. At the rest of the *n-k* slots, these two nodes are in a sleep state. The probability of one node monitor another node is

$$P(n,k) = 1 - \frac{\binom{n-k}{k}}{\binom{n}{k}} \quad (1)$$

When the ratio of *k*/*n* is very small, the value of *P* is close to 1, such as *P* (1000, 70) = 0.995, According to these two parameters, we can see that each node has a very high probability of discovery when the sleep time is 93% of the total time.

When the probability of nodes in the active state is *p*, the probability of the node first discovery another node is$(1-p^2)^{n-1}p^2$, from this we can get the average discovery latency for the nodes is

$$E(l) = p^2 \sum_{n=1}^{\infty} n(1-p^2)^{n-1} \quad (2)$$

Because of

$$\sum_{n=1}^{\infty} n(x)^{n-1} = \frac{1}{(1-x)^2} \quad (3)$$

So the average discovery latency of Birthday algorithm is $\frac{1}{p^2}$.

B. Disco

Disco algorithm [5] derived from the Chinese remainder theorem [6]. The algorithm requires nodes to choose a pair of prime numbers *p*, *q*, and makes the *p***q* as a node of the working cycle, it can guarantee two nodes find each other in *p***q* slots, the sum of the reciprocal of the prime equal to the duty cycle of the node.

Let node c_i select $m_i = 3$, then the duty cycle of the node c_i is $DC \approx 33\%$ and start counting at the slot *x*=1, another node c_j choice $m_j = 5$, duty cycle is $DC \approx 20\%$ and start counting at the slot *x*=2. Through the observation in the Fig. 1 that the nodes in the *x*=7 and *x*=22 to achieve neighbor discovery.

Choosing only one prime number is not perfect of one node, because when the prime number of two nodes is the

doi: 10.18178/jacn.2016.4.2.217

same, it is possible to cause the situation that two nodes can't find each other, we select two different prime numbers p_{i_1} and p_{i_2} for a node, the duty cycle is $DC_D \approx \frac{1}{p_{i_1}} + \frac{1}{p_{i_2}}$. When the duty cycle of the Disco algorithm is *d*,

$$DC_D \approx \frac{1}{p_{i_1}} + \frac{1}{p_{i_2}} \approx \frac{2}{p} = d \tag{4}$$

Its max discovery latency is

$$DL_D = p^2 = \frac{4}{d^2} \tag{5}$$

x	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
c_i	-	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22
c_j	-	-	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21

Fig. 1. Disco node working process.

C. U-Connect

U-Connect algorithm [7] uses the Chinese Remainder Theorem and optimizes it. U-Connect selects a $p \times p$ square matrix as its working cycle, where p is prime. It make sure that each node wake up is no more than (p+1) /2 slots. U-Connect algorithm neighbor discovery process can be expressed as

$$\Psi_u(m,t) = \begin{cases} 1, & \text{if}[t]_p = 0 \; or \; 0 \le [t]_{p^2} \le \frac{p+1}{2} \\ 0, & \text{otherwise} \end{cases} \tag{6}$$

When we consider the period $T = p^2$, T is the max discovery latency, so U-Connect is a deterministic algorithm. For Fig. 2, the yellow block represents the wake up.

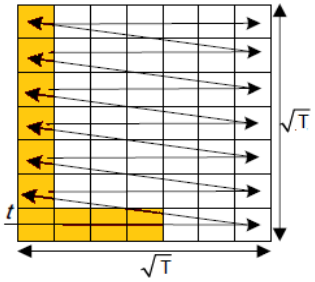

Fig. 2. U-Connect discovery process.

From the Fig. 2, when the duty cycle of the U-Connect algorithm is *d*.

$$DC_U = \frac{\frac{3\sqrt{T}}{2}}{T} = \frac{3\sqrt{T}}{2T} \approx \frac{3}{2\sqrt{T}} = d \tag{7}$$

The max discovery latency of U-Connect is

$$DL_U = T = \frac{9}{4d^2} \tag{8}$$

D. SearchLight

Although the probability algorithm (such as Birthday algorithm) in the average performance is good, but they have a long tail problem cause the larger the max discovery latency. Deterministic algorithms (such as Disco, U-Connect), while improving the worst boundary of the discovery delay, have lost the average performance. In order to solve these two problems, SearchLight algorithm [8] is proposed.

SearchLight is divided into S (sequential) and R (randomized), which combines the characteristics of deterministic and probability algorithms, and applies to symmetric and asymmetric networks.

1) Sequential probing

This model is referred to as SearchLight-S, changes in the position of the Probe slot is regular, starts with 1 and each of the next cycle will add 1 until added to $\left\lfloor \frac{t}{2} \right\rfloor$. As shown in Fig. 3, *A* represents Anchor slots, *P* represents Probe slots, *P* will be probe in every *t* cycle, and the position of each cycle is different from the previous position.

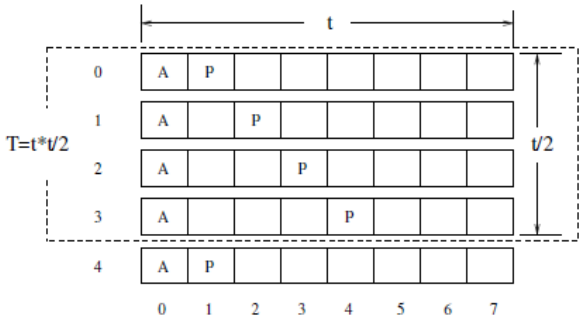

Fig. 3. SearchLight-S node working process.

2) Randomized probing

This model is called SearchLight-R, where the Probe slots position change randomly. At the random placement of Probe slots, SearchLight-R learned the idea of the birthday paradox, makes the Probe slots overlap probability greatly improved. Faster discovery rates are obtained using the Randomized Probing model instead of the Sequential Probing model. When the duty cycle of SearchLight is set to *d:*

$$DC_S = \frac{t}{\frac{t^2}{2}} = \frac{2}{t} = d \tag{9}$$

The max discovery latency for SearchLight is

$$DL_S = \frac{t^2}{2} = \frac{2}{d^2} \tag{10}$$

III. Discovery Latency Analysis of Symmetric Network

Firstly, we compare the performance of different algorithms with the same duty cycle in the nodes. As shown in Fig. 4, it shows the discovery latency CDF (cumulative distribution function) in the 5% duty cycle. Before Slots=500 Birthday algorithm is the fastest of these algorithms, but to the latter part, it encountered unpredictable large latency, appear the long tail phenomenon.

After Slots=500, SearchLight-R is the fastest in several algorithms, and the first to complete the task with SearchLight-S at the same time. The effect of the Disco algorithm is not very well, whether it is the average discovery latency or the max discovery latency. U-Connect overall performance is straight and narrow, the worst

performance in early, later performance is better than Birthday and Disco, its max discovery latency only behind of SearchLight. At the same time, we can see that the max latency of the four kinds of deterministic algorithm is basically consistent with the theoretical value.

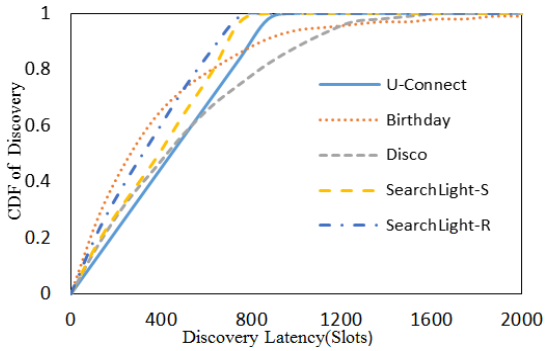

Fig. 4. CDF of discovery latency at 5% duty cycle.

Next, we analyze the average latency of each algorithm in different duty cycle conditions. The performance of Disco is not very well, so it is not included. Statistical results are shown in Fig. 5.

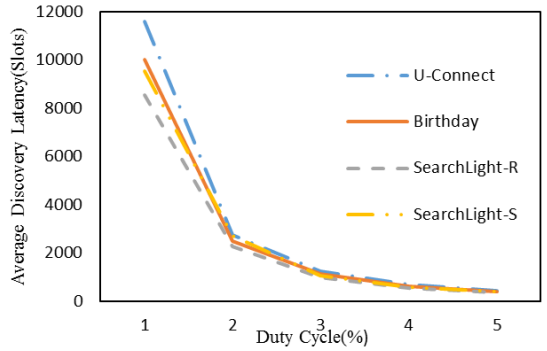

Fig. 5. Average latency in different duty cycle.

According to Fig. 5, The Searchlight-R algorithm can obtain the smallest average discovery latency in the same duty cycle. All the algorithms show that the latency decreases with the increase of duty cycle.

In summary, in the case of symmetric networks, SearchLight algorithm provides the lowest max discovery latency bounds. The results show that the average performance of the SearchLight-R algorithm is also the best. While the performance of Disco algorithm is the worst, the other two algorithms have their own advantages and disadvantages in different situations.

IV. Discovery Latency Analysis of Asymmetric Network

In practical application nodes often adjust their own duty cycle according to the different environment, the performance of this part is more in line with the daily application, has more reference value.

First, we adjust duty cycle of the two nodes to 1% and 10%, get the results of Fig. 6.

When the nodes are working at 1% and 10% duty cycle, the SearchLight-R is the faster in the initial discovery rate, but because of the long tail phenomenon, the max latency is poor. Birthday algorithm performance is more mediocre, and the emergence of a more serious phenomenon of the long tail. From this result, we can see that the probability of the algorithm will appear the long tail phenomenon Disco in the asymmetric environment performance is excellent, the early discovery rate is relatively fast. The performance of SearchLight-S is worse, early discovery performance is the worst. When we adjust the duty cycle to 1% and 5%, get the Fig. 7.

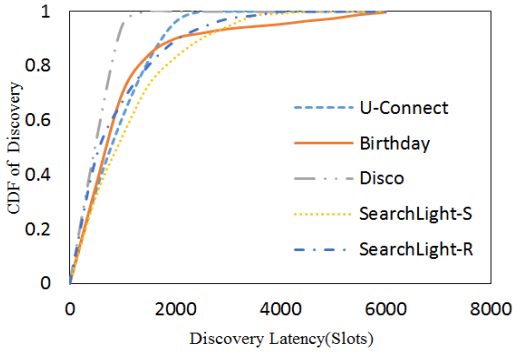

Fig. 6. CDF of discovery latency at 1% and 10% duty cycle.

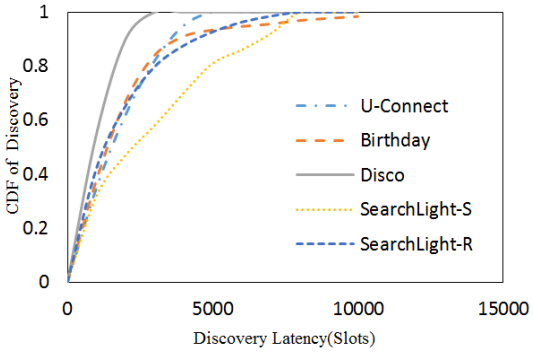

Fig. 7. CDF of discovery latency at 1% and 5% duty cycle.

For the asymmetric environment of 1% and 5% duty cycle, the Disco algorithm is still the best in many algorithms and first completed the task of discovery. U-Connect performance is also very good, although in the early discovery rate not compare with the SearchLight-R and Birthday algorithm, there is no long tail phenomenon. One must note that the SearchLight-S performance is the worst, and performance with other algorithms are not at the same level. Birthday performance doctrine of the mean, the long tail phenomenon is an important factor affecting its performance.

Next, we analyze the average discovery latency of each algorithm in the above two kinds of asymmetric environments, when duty cycle is in 1%~5% and 1%~10%, we can see the results in Fig. 8.

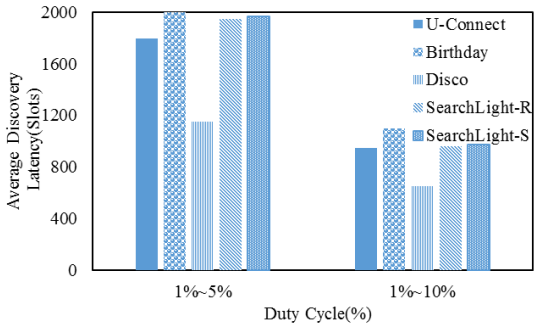

Fig. 8. Average latency for the 1%~5% and 1%~10% duty cycles.

It is not difficult to see that the average latency in the case of 1%~5% is more than 1%~10%, because the duty cycle is lower, the shorter the working time, the average latency is longer. In the case of the two duty cycle SearchLight-S, the average discovery latency is the highest, the worst performance. In contrast, the performance of the Disco algorithm in these two duty cycle is better than the performance of the algorithm. SearchLight-R because of its poor long tail phenomenon in asymmetric networks, the average latency is large.

V. Conclusion

In this article, we simply introduce four kinds of asynchronous neighbor discovery algorithms, then we analyze four algorithms from relationship between duty cycle and discovery latency in simulation platform. The results show that the probability of the algorithm in the initial stage of the discovery generally have a faster rate of discovery. And when the duty cycle is increased, the neighbor discovery delay will be significantly smaller, but there is no one algorithm is applicable in both symmetric and asymmetric network.

References

[1] M. Conti and S. Giordano, "Multihop Ad Hoc networking: The reality," *IEEE Communications Magazine*, April 2007.

[2] M. Conti and S. Giordano, "Multihop Ad Hoc networking: The theory," *IEEE Communications Magazine*, April 2007.

[3] M. J. McGlynn and S. A. Borbash, "Birthday protocols for low energy deployment and flexible neighbor discovery in ad hoc wireless networks," in *Proc. the 2nd AtuCM International Symposium on Mobile Ad Hoc Networking & Computing*, New York, NY, USA, 2001, pp. 137–145.

[4] Wikipedia. Birthday paradox. [Online]. Available: http://en.wikipedia.org/wiki/Birthdayproblem

[5] P. Dutta and D. Culler, "Practical asynchronous neighbor discovery and rendezvous for mobile sensing applications," *Proceedings of SenSys*, vol. 8, pp. 71-84, 2008.

[6] I. Niven, H. S. Zuckerman, and H. L. Mongomery, *An Introduction to the Theory of Numbers*, John Wiley & Sons, 1991.

[7] A. Kandhalu, K. Lakshmanan, and R. Rajkumar, "U-connect: A low-latency energy-efficient asynchronous neighbor discovery protocol," *Proceedings of IPSN*, vol. 10, pp. 350-361, 2010.

[8] M. Bakht and R. Kravets, "SearchLight: Asynchronous neighbor discovery using systematic probing," *Mobile Computing and Communications Review*, vol. 14, no. 4, pp. 31-33, 2010.

Zhipeng Jia was born in Beijing, China in 1992. He received the B.E. degree in electronic and information engineering from Beijing University of Technology, Beijing, China, in 2014. He is now a M.E student in the School of Software Engineering, Beijing University of Technology. His research interests in wireless ad hoc networks.

Zhiqing Huang was born in Sichuan, China in 1970. He received the B.E. degree in mechanical engineering and the M.S. degree in computer engineering from Guizhou University, Guizhou, China in 1992 and 1995 respectively, and the Ph.D. degree in computer integrated manufacturing system from Chongqing University, Chongqing, China in 2001. He has been with Bell Labs Lucent Technologies China from April 2001 to February 2011 as a member of technical staff with focus on the telecommunication network and IP network protocols. From March 2011 until now, he is an associate professor at Beijing University of Technology. His research interests include internet of things (IoT), wireless ad hoc networks and indoor localization and navigation.

Xiongye Su was born in Guangdong, China in 1992. He received the B.E. degree in software engineering from Beijing University of Technology, Beijing, China, in 2015. He is now a M.E student in the School of Software Engineering, Beijing University of Technology. His research interests in wireless ad hoc networks.

Mengjia Li was born in Beijing, China in 1993. She received the B.E. degree in software engineering from Beijing University of Technology, Beijing, China, in 2015. She is now a M.E student in the School of Software Engineering, Beijing University of Technology. Her research interests in wireless ad hoc networks.

Huan Zhang was born in Beijing, China in 1992. He received the B.E. degree in software engineering from Beijing University of Technology, Beijing, China, in 2015. He is now a M.E student in the School of Software Technology, Zhejiang University. His research interests in wireless ad hoc networks.