Collective Neighbour Discovery in Wireless Sensor Network

Shalima Binta Manir United International University House No. 80, Satmasjid Road, Dhaka 1209, Bangladesh and Centre for Quantum Technologies, National University of Singapore 3 Science Drive 2, 117543 Singapore

ABSTRACT

Neighbour discovery is one of the imperative elements of wireless sensor network which discerns close by nodes so that they can reciprocate information and collaborate. In this paper collective neighbour discovery is proposed to reduce latency period and accomplish the discovery more efficiently. To achieve this purpose each node will be active during recommended neighbours' active time to attain rapid neighbour discovery. Comparison shows that collective neighbour discovery performs better than the existing searchlight protocol. We evaluate the performance and characteristics of collective neighbour discovery by varying different parameters. One feature of this protocol is that it can be combined with searchlight protocol. Simulation and analysis shows that the combined protocol enhances the performence and abates the latency of searchlight effectively.

General Terms

Wireless sensor network, Protocol, Neighbour discovery

Keywords

Duty cycle, threshold, Window size, latency

1. INTRODUCTION

The rapid advancement of wireless sensor network integrates sensing, communication and computation between many small sensor nodes. These sensor nodes collect and assemble necessary data. Next, they analyse the data and finally transmit. Since in some applications the nodes are placed randomly, this kind of network is commonly deployed without any fixed routing infrastructure [1]. It has applications like environmental monitoring, military applications, agricultural monitoring, earthquake measurement etc. [2]. Quick discovery of neighbouring nodes is needed to make these applications efficient. Neighbour discovery is challenging because of unpredictable number of neighbours for each node and lack of synchronous global clock for every node [3]. One node can discover another only if they are active at the same moment and their transmission range covers one another. Mainly there are three types of neighbour discovery. These are probabilistic, quorum based and deterministic [4]. Typical discovery protocols only concentrate on more energy saving schemes. Usually energy saving protocols decreases the active period of the sensor nodes to conserve energy so that the discovery delay increases. Long discovery latency may not be the main concern of these protocols. However, longer delay may lose the goal of the applications. So, it is necessary to operate the nodes more intelligently such that the latency may be reduced. Previous quick discovery schemes mainly emphasised on indirect neighbour discovery through a transitive neighbourhood relation which may be practically unrealistic [5, 6]. There were some direct approaches with redundant active periods [7]. The motivation of collective neighbour discovery protocol is to allow the nodes to cooperate with each other instead of acting independently to make the discovery process more quick, efficient. This approach is also realistic.

2. RELATED WORK

There exists different kind of neighbour discovery algorithms based on different features where their discovery periods are relatively higher. Still there are some faster discovery protocols such as group-based discovery [7], on-demand acceleration discovery [5] and gossip based discovery [6]. On-demand acceleration discovery is an indirect discovery approach. Figure 1 illustrates the main idea of this discovery. Suppose node X's active time slots are 1,7,9 and node Y's active time slots are 0,2,4 and node Z's active time slots are 0,6,9. At global time 2, node X discovers node Y and node X includes the next active schedule of node Y as additional active slot. So node X would include slot 5 for possible indirect discovery. At global time 4, node Y discovers node Z so that when node X become active in its additional time slot node Y will forward node Z to node X. Instead of discovering node Z at global time 10, node X discovers node Z at global time 6. So this indirect discovery accelerates the procedure from time 10 to 6 [5]. There is a probability that node X and Z are also neighbour but this is not confirmed because node Z may be in out of range of node X. This is the main drawback of this discovery process. In group based discovery when two nodes discover one another during their active state they make a group and become aware of one anothers active schedule [7]. For example in figure 2, at time 2 node q and r are active and also they are in

Fig. 1. On-demand acceleration discovery .

time 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 p q r

Fig. 2. Group based discovery.

communication range. Then they make a group and become aware of one another next active scheduling. Node q discovers node p at time 4 and appends this node to its neighbour table. At time 6, node q wakes up because this is node p's active time and sends the next active time of node r to node p. Finally node p is active at time 10 and discovers node r as its neighbour. Each node has become active again in its existing neighbour's next active period to get recommended neighbour's next active time. So this process adds redundant time slots by waking up already discovered node's next active time. In gossip based discovery a node utilises location information and can know indirectly about its neighbours existence through other neighbours [6]. Other protocols of neighbour discovery have different types of scheduling algorithms. In asynchronous search light protocol there are two active slots in every cycle. One is static anchor slot and another one is moving probe slot [8, 9]. After probing of *n* periods it confirms two overlaps between two nodes which may not be aligned [4]. In probabilistic birthday protocol each node can be one of the three different states- transmit, listen and energy saving [10]. This protocol allocates different probabilities for sending, receiving and sleeping in individual slot [10]. It also proposes that the discovery of one node is accomplished when at least two nodes have the same birthday in same time. That means one of them is the sender, while the rest of them are receivers [11]. There are quorum based protocols where each node chooses one row and one column active slots entries in $m \times m$ matrix [4, 12]. The main limitation of this protocol is that in average case the performance may worsen due to extra intersection [9]. Extended quorum system is based on indirect neighbour discovery [13]. Several protocols like disco, u-connect and co-prime based also exist [14, 15, 16].

3. PRELIMINARIES

In this section some basic ideas and assumptions about neighbour discovery is presented. How two nodes can discover one another in a wireless sensor network is also discussed here.

3.1 Node

Wireless sensor network consists of wireless sensor nodes. Each node has transmitter and receiver to broadcast or receive signal from other nodes and also each node has limited power supply. Due to limited power capacity each node cannot always stay active to transmit or receive data. So every node has to combine its active or inactive periods to effectively minimise energy consumption. There are some protocols in which every node either activate its receiver or transmitter during active state [10]. In collective neighbour discovery we assume that in active state both transmitter and receiver are active to transmit and receive simultaneously.

3.2 Time

We assume that time is discretized in slots for unit interval of fixed length. The slot length is ample for communication. We have indexed these interval as 1,2,3 ... to define time units. From the figure 1, 2, 3 we can get the idea. Note that there is no global clock so that all the nodes work according to their local clock. Consequently, the local time index for a given global time may differ from each other. For this reason, the concept of global clock and local clock are introduced. The time variation of each node is defined as time offset so that every node has different time offset to define their local time. Every protocol has their own time scheduling algorithm to determine when a node will be active or inactive according to their local time slots.

3.3 Time Period

Every node has a time period consisting of fixed number of time slots and the number of slots per period may vary. For example, we may take 10 slots per period or 20 slots per period according to the circumstances. Under same network, number of slots per period of every node must be the same. Time period is also referred as window size.

3.4 Duty Cycle

At any time a node can be in active or dormant state. To measure how long a node is in the active state, the term duty cycle is used. It is the ratio of time a node spends as active state vs. the total elapsed time [4].

3.5 Neighbour Discovery

If two nodes are active at the same moment and if one node's transmitted message can reach another node's receiver then they discover one another as their neighbour. Otherwise, they remain undiscovered. Moreover, every node has fixed transmission range.

4. SYSTEM MODEL

In this protocol we assume that one node can communicate with other nodes within a finite distance and also each of the nodes only has local clock without any access to any global clock. Time is represented as distinct slotted units that are presumed to be real time period to consent communication. The number of time slots in each period is defined as window size. Every node can either be active or inactive in any given time. For scheduling active slots randomness is adapted. That is, if the window size is n in each period, k slots are selected uniformly randomly from n slots as active slots for a node. When a node is in active state it can transmit and receive at the same time. During active period each



Fig. 3. Collective neighbour discovery (CND)process.

node broadcasts a message including its current neighbours and their next wake up time and also receives messages from others simultaneously. Figure 3 illustrates an example of collective neighbour discovery. For window size 10, node A will be active at slot 1, 4, 9 and node B will be active at slot 2, 4, 9 while node C will at slot 1, 5 and 8. These nodes have different time offset due to absence of clock synchronisation. Node A and B both are active at global time = 5 and they are in transmission range of one another. So they discover themselves and also retain record of each others next active schedule according to their own local time. At time = 7, node B discovers node C and recommends its neighbour A and its next active time to C. After that C will be active at the proposed neighbour's next active slot. At time = 10, C is active and perceives node A if they are in communication range then adds A as neighbour. Otherwise C includes node A to its out of range list so that in future if any node recommend node A to C it will ignore the recommendation time unless this slot is already selected for node C to be active.

Now we are going to discuss more elaborately. Suppose, there are seven nodes {A, B, C, D, E, F, G}. Node A has neighbours {B, C, G} and node D has neighbours {G, E, F}. Both nodes A and D also know their neighbours' next wake up time. In this model when two nodes are active in the same time slot their transmitter and receiver are also work in parallel and each time slot is sufficient for bidirectional communication. At time t, node A and D both are active. So node A transmits a message containing its existing neighbours' next wake up schedule and also node D transmits a message containing its neighbours' next wake up time simultaneously and they are in their communication range. Then node A discovers node D and gets the next wake up schedule of node D's neighbours G, E, F as recommended neighbours. Then node A appends the next wake up time of node E and F to its wake up schedule as additional active slot. It does not add node G's next active time because node G already is in node A's neighbour list. Similarly, node D appends the next wake up time of node B and C as extra active slots. After that node A and D will be active in these extra active slots and discover recommended neighbours if they are in range.

5. SIMULATION AND ANALYSIS

The simulation setting covers 3000 *3000 unit square area and the transmission range is defined by a threshold. Figure 4 shows the comparison between collective neighbour discovery and random active slot scheduling discovery. Both of these have initial 3 active slots that are chosen randomly, window size is 10 and the threshold is 200 units. Only difference between these two processes is that collective neighbour discovery takes recommendation from its neighbours' but random active slot scheduling does not. The number of nodes of this comparison is 300 and we have plotted the moving average of data set. It is evident from the graph collective neighbour discovery reduces the completion time significantly. So it performs faster than random

active slot scheduling discovery in the same simulation environment. Figure 5 shows the comparison between collective neighbour discovery and searchlight. In this simulation number of nodes are 200, window size is 20 and threshold is 450. Initially searchlight performs well but when the number of nodes increases the performance of collective neighbour discovery is much better than searchlight. The discovery latency is less than searchlight when the node density is higher. In the figure when the number of nodes is greater than 110 the completion time of collective neighbour discovery is decreased noticeably. If the node density is higher collective neighbour discovery takes advantage by getting more recommendations from neighbours since, when the density is higher the probability of recommendation is also higher. Collective neighbour discovery is also applied to searchlight and figure 6 shows the comparison of discovery rate between searchlight and searchlight with collective neighbour discovery. For this simulation the window size is 20 and threshold is 450 unit. The percentage of discovery rapidly increases when collective neighbour discovery is applied and it also reduces the discovery latency by 33.33% compared to searchlight.

Next, we evaluate the characteristics of collective neighbour discovery. Figure 7 shows the comparison of duty cycle vs. number of nodes for different thresholds in collective neighbour discovery. If the threshold is increased the duty cycle also gets increased because when the transmission range is higher node density also is higher and more active slot of recommended neighbours will be added. So the threshold can be adjusted according to the energy saving needs. In figure 8, we compare the completion time vs. number of nodes by varying different number of initial active slots per window. In this simulation the window size is 20 and threshold is 200. Completion time is decreased if the number of slots in a period is increased and data are plotted as moving average. Figure 9 shows the comparison of completion time vs. number of nodes in collective neighbour discovery by varying threshold. Here the window size is 10 and the number of node is 300. In this result when the threshold is higher the discovery latency is decreased remarkably because the chance of getting recommendation is also higher.

6. FUTURE DIRECTION

Besides wireless sensor networks, collective neighbour discovery may be implemented in mobile computing for arbitrary moving nodes. There is a scope to implement collective neighbour discovery in interesting research areas like self-organising swarm robotics [17]. Collective neighbour discovery may be applied to communicate among mobile robots [18]

7. CONCLUSION

Different types of simulation results are presented in this paper. The characteristics of collective neighbour discovery are also shown. This protocol offers a way to adjust the latency and energy efficiency by varying different parameters for different environment. The flexibility of this protocol is that it can be applied with existing protocols like searchlight. From the analysis of the simulation result we can conclude that collective neighbour discovery performs effectively to make the discovery process faster and reduces the delay time notably.

8. ACKNOWLEDGEMENT

We thank Md. Tanvirul Islam for his valuable discussion and comments.



Fig. 4. Comparison between collective neighbour discovery and random active slot scheduling discovery.



Fig. 5. Comparison between collective neighbour discovery and searchlight.

9. REFERENCES

- [1] Sudarshan Vasudevan, Micah Adler, Dennis Goeckel, and Don Towsley. Efficient algorithms for neighbor discovery in wireless networks. *Networking, IEEE/ACM Transactions on*, 21(1):69–83, 2013.
- [2] Daniele Puccinelli and Martin Haenggi. Wireless sensor networks: applications and challenges of ubiquitous sensing. *Circuits and Systems Magazine*, IEEE, 5(3):19–31, 2005.
- [3] Sudarshan Vasudevan, Donald Towsley, Dennis Goeckel, and Ramin Khalili. Neighbor discovery in wireless networks and the coupon collector's problem. In *Proceedings of the 15th*



Fig. 6. Comparison of discovery rate between searchlight and searchlight with collective neighbour discovery.



Fig. 7. Comparison of duty cycle vs. number of nodes for different thresholds in collective neighbour discovery.

annual international conference on Mobile computing and networking, pages 181–192. ACM, 2009.

- [4] Wei Sun, Zheng Yang, Xinglin Zhang, and Yunhao Liu. Energy-efficient neighbor discovery in mobile ad hoc and wireless sensor networks: A survey. *Communications Surveys* & *Tutorials, IEEE*, 16(3):1448–1459, 2014.
- [5] Desheng Zhang, Tian He, Yunhuai Liu, Yu Gu, Fan Ye, Raghu K Ganti, and Hui Lei. Acc: generic on-demand accelerations for neighbor discovery in mobile applications. In Proceedings of the 10th ACM Conference on Embedded Network Sensor Systems, pages 169–182. ACM, 2012.



Fig. 8. Comparison of completion time vs. number of nodes by varying different number of initial active slots per window.



Fig. 9. Comparison of completion time vs. number of nodes in collective neighbour discovery by varying threshold.

- [6] Sudarshan Vasudevan, Jim Kurose, and Don Towsley. On neighbor discovery in wireless networks with directional antennas. In *INFOCOM 2005. 24th Annual Joint Conference* of the IEEE Computer and Communications Societies. *Proceedings IEEE*, volume 4, pages 2502–2512. IEEE, 2005.
- [7] Qiang Niu, Weiwei Bao, and Shixiong Xia. An improved group-based neighbor discovery algorithm for mobile sensor networks. *International Journal of Distributed Sensor Networks*, 2014, 2014.
- [8] Mehedi Bakht, Matt Trower, and Robin Kravets. Searchlight: helping mobile devices find their neighbors. In *Proceedings*

of the 3rd ACM SOSP Workshop on Networking, Systems, and Applications on Mobile Handhelds, page 9. ACM, 2011.

- [9] Mehedi Bakht, Matt Trower, and Robin Hilary Kravets. Searchlight: won't you be my neighbor? In *Proceedings of the* 18th annual international conference on Mobile computing and networking, pages 185–196. ACM, 2012.
- [10] Michael J McGlynn and Steven A Borbash. Birthday protocols for low energy deployment and flexible neighbor discovery in ad hoc wireless networks. In *Proceedings of the 2nd ACM international symposium on Mobile ad hoc networking & computing*, pages 137–145. ACM, 2001.
- [11] Shudong Fang, Stevan M Berber, and Akshya K Swain. Analysis of neighbor discovery protocols for energy distribution estimations in wireless sensor networks. In *Communications*, 2008. ICC'08. IEEE International Conference on, pages 4386–4390. IEEE, 2008.
- [12] Chung-Ming Own, Zhaopeng Meng, and Kehan Liu. Handling neighbor discovery and rendezvous consistency with weighted quorum-based approach. *Sensors*, 15(9):22364–22377, 2015.
- [13] Desheng Zhang, Tian He, Fan Ye, Raghu K Ganti, and Hui Lei. Eqs: Neighbor discovery and rendezvous maintenance with extended quorum system for mobile sensing applications. In *Distributed Computing Systems* (*ICDCS*), 2012 IEEE 32nd International Conference on, pages 72–81. IEEE, 2012.
- [14] Prabal Dutta and David Culler. Practical asynchronous neighbor discovery and rendezvous for mobile sensing applications. In *Proceedings of the 6th ACM conference on Embedded network sensor systems*, pages 71–84. ACM, 2008.
- [15] Arvind Kandhalu, Karthik Lakshmanan, and Ragunathan Raj Rajkumar. U-connect: a low-latency energy-efficient asynchronous neighbor discovery protocol. In Proceedings of the 9th ACM/IEEE International Conference on Information Processing in Sensor Networks, pages 350–361. ACM, 2010.
- [16] Lin Chen, Ruolin Fan, Kaigui Bian, Mario Gerla, Tao Wang, and Xiaoming Li. On heterogeneous neighbor discovery in wireless sensor networks. *arXiv preprint arXiv:1411.5415*, 2014.
- [17] Yudong Zhang, Praveen Agarwal, Vishal Bhatnagar, Saeed Balochian, and Jie Yan. Swarm intelligence and its applications. *The Scientific World Journal*, 2013, 2013.
- [18] M Yogeswaran and SG Ponnambalam. Swarm robotics: An extensive research review. Sciyo, 2010.