A Simple Neighbour Discovery Procedure for Bluetooth Ad Hoc Networks

Miklós Aurél Rónai
Budapest University of Technology and Economics
Department of Telecommunications and Media Informatics
High Speed Networks Laboratory
Magyar tudósok körútja 2., H-1117 Budapest, Hungary
Email: ronai@szit.bme.hu

Eszter Kail
Budapest University of Technology and Economics
Department of Telecommunications and Media Informatics
High Speed Networks Laboratory
Magyar tudósok körútja 2., H-1117 Budapest, Hungary
Email: kail@hit.bme.hu

***Abstract*— Bluetooth was designed to replace cables between electronic devices, but it can also be used to build ad hoc networks. In the cable replacement scenario nodes can discover each other using the inquiry procedure. This has been designed to satisfy the requirements of cable replacement applications so that it discovers all neighbours in a fixed amount of time. On the other hand, the inquiry procedure is not well suited for nodes in a Bluetooth ad hoc network, where we found three main weaknesses. First, inquiry takes a lot of time and therefore it requires too much overhead if used regularly. Second, it is very inefficient to transmit data simultaneously with the inquiry. Third, the inquiry assumes asymmetric roles, which is not well suited to an ad hoc network of peer nodes. To resolve these problems we propose the Simple Neighbour Discovery (SND) procedure. We evaluate it using analytical and simulation methods and show that it is configurable in the trade-off between discovery time and overhead. The results show that the SND procedure is more efficient in an ad hoc network of peer Bluetooth nodes than the inquiry procedure.**

I. INTRODUCTION

The primary goal of Bluetooth [1] is to replace cables between electronic devices. Cable replacement needs little flexibility in neighbour discovery, due to the static and pair wise connections between nodes. In this case it does not cause a problem if the discovery process – called inquiry in Bluetooth [2] – takes relatively long time and if the data transmission is stopped or performed on very low bit rate during the discovery procedure.

On the other hand, when Bluetooth is used in a dynamically changing environment like ad hoc networks, where the nodes are continuously moving, appearing and disappearing, we found three weaknesses of Bluetooth's current neighbour discovery procedure. First, inquiry takes a lot of time and therefore it requires too much overhead if used regularly. Second, it is very inefficient to transmit data simultaneously with the inquiry. Third, the inquiry assumes asymmetric roles, which is not well suited to an ad hoc network of peer nodes.

We propose the Simple Neighbour Discovery (SND) procedure for Bluetooth that assumes symmetric roles, it is faster than the inquiry and it enables more bandwidth for data transmission during the discovery period. The solution is based on beacon packets, which are sent by a node regularly with the appropriate information to establish a connection. Other nodes can scan for these beacon packets to discover the neighbour.

The paper is organized as follows. In section II we give a short overview of related work. In section III we describe the SND procedure. In section IV we analyze the SND with analytical and simulation methods. In section V we compare the inquiry and the SND procedures. In section VI we summarize our work.

II. RELATED WORK

To provide efficient support for ad hoc networks with Bluetooth the technology needs to be enhanced with scatternet (a network of Bluetooth devices) formation, scheduling and neighbour discovery mechanisms. These key issues of using Bluetooth in ad hoc networks are investigated by Guérin et al. [3].

Several papers consider the problem of scatternet formation. Záruba et al. [4] propose protocols to establish scatternets as trees of Bluetooth nodes (Bluetrees). Wang et al. [5] propose a scatternet formation scheme called Bluenet to build efficient scatternets. Félegyházi [6] proposes two solutions for scatternet formation. Stojmenovic [7] addresses the problem of scatternet formation and maintenance for multi-hop Bluetooth ad hoc networks from the graph theory point of view.

A scheduling mechanism is investigated by Rácz [8]. The authors propose the Pseudo-Random Coordinated Scatternet Scheduling (PCSS) algorithm to perform the scheduling of both intra- and inter-piconet communication. Baatz et al. [9] propose an adaptive distributed approach to scatternet scheduling which is capable to handle the dynamics of network traffic.

Neighbour discovery is investigated by Salonidis et al. [10], who suggest a symmetric way to establish connections for Bluetooth units. They propose to use the Bluetooth inquiry procedure, but instead of predefined roles, the nodes are performing the inquiry and inquiry-scan in an alternating fashion. Law and Siu [11] propose a scatternet formation algorithm, which is based on the idea that every node performs inquiry and inquiry-scan with a certain probability. However these proposals solve only the problem of the asymmetry.

III. Simple Neighbour Discovery Procedure

The main concept of the SND procedure is that every node sends beacon messages regularly at pseudo-randomly selected slots, which contain all the basic information about the node that is necessary for its neighbours to initiate connection establishment. If a node wants to discover its neighbours or it just wants to update its information about them, it performs scanning and scans for these beacon packets. The scanning does not need to be continuous. It can be done for short periods of time when the node does not receive or transmit data.

A. Sending beacon packets

Our goal was to create a flexible neighbour discovery procedure that can be configured according the need of the nodes. Therefore we introduce a tunable parameter called beacon period. Beacon periods are consecutive periods of equal length. In every beacon period the node sends a one slot long beacon packet. The beacon packet contains the necessary information for connection establishment: the node's clock, address and the length of the beacon period. If the node wants to send beacons more often, then it chooses a shorter beacon period; otherwise it uses a longer one. To make beacons predictable the timeslot of the beacon in the beacon period is chosen pseudo randomly using the clock and the address of the node.

To allow the devices to take beacons of their neighbours into account, beacon packets are given the priority over baseband data packets. This means that nodes will not start transmitting a data packet if this transmission overlaps with the timeslot of the beacon.

The master can take into account that slaves have to send beacons. In this case the master can skip polling the slave that is currently unavailable due to beacon sending and it can poll another one. If the master does not take this into account, then it will not receive any answer to its poll message from the slave that is currently sending a beacon and it will poll the slave next time.

Since beacon packet prediction is just an option it may happen that a data packet will be lost, because the source node did not take into account that the destination has to send a beacon. In this case the packet loss is corrected by the automatic retransmission query (ARQ) mechanism.

The transmission frequency of the beacon is selected pseudo randomly and also calculated from the clock and the address of the node. To ensure faster discovery of nodes not every Bluetooth frequency is used as beacon frequency. For example in the case of 79 hop carriers only 32 frequencies are used to send beacons. Beacon packets should be sent with one of the 63 DIAC access code [2], so that the nodes which are aware of SND and are in scanning mode can receive the beacons.

Fig. 1 illustrates the timing of the beacon periods and beacon packets. It can be seen that one data packet is lost, because the destination node had to send a beacon.

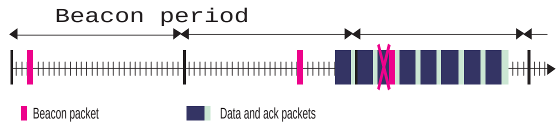

Fig. 1. Timing of the beacon packets

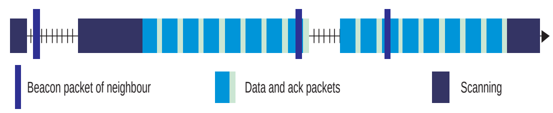

Fig. 2. Timing of the scanning

B. Scanning

To discover the neighbours or to update the status information about its neighbours a node performs scanning. This means that for a period of time the node does not send or receive data packets, just listens for beacon packets. To ensure flexibility the length and the timing of the scan periods are not fixed and can be tuned according the node's need. The more often and the longer the node scans, the quicker it can discover its neighbours. The node randomly selects frequencies for scanning and during a scan period the node listens only on one frequency.

Since beacon packets are predictable and have priority over baseband data packets, if a node wants to refresh its information about a specific neighbouring device, it derives the frequency and the time of scanning from the neighbour device's clock and address. This way the nodes can check very easy, whether the already known neighbours are still in radio range. If a node does not receive any beacon packet from a certain node for a given amount of time, then this node can be considered as to be moved away or turned off.

Fig. 2 shows a case when a node performs scanning meanwhile in a time multiplex fashion it transmits and receives data packets. In the figure we can see the beacon packets of a neighbour. Since the beacon slots have not yet coincided with the scanning, the node has not yet discovered this neighbour so far.

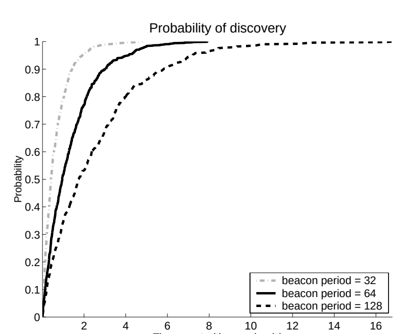

Fig. 3. The effect of the length of the beacon period on the performance of the neighbour discovery procedure

IV. Performance of the SND procedure

We implemented the SND procedure in a discrete event driven, object oriented simulator environment, called Plasma [12], [13]. In our simulations we placed two nodes in the system. The intervals between scanning periods were chosen uniformly between 0 and 1000 slots. The duration of the scanning was 10 slots in all cases and the frequency used for scanning was selected randomly from the 32 channels with uniform distribution.

In the first experiment we studied the effects of the beacon period's length and time spent with scanning. One node performed scanning and the other node sent beacon packets. The results can be seen in Fig. 3. The figure shows the probability of the neighbour discovery as a function of the overall time spent with scanning in three different cases, where the beacon period was 32, 64, 128 slots long. It can be seen that if a node sends beacons more often, the probability of discovery is higher.

The figure suggests that the discovery probabilities follow an exponential curve. In order to prove this assumption, we carried out analytical calculations to investigate how much time is needed for discovering a neighbour with a certain probability.

It is assumed, that a node performs scanning on a given frequency for a period of T_{SCAN}, where the value of T_{SCAN} is at least $2T_S$ and T_S is the length of a timeslot. This requirement is necessary because the native clocks of the nodes may not be synchronized, and so the beginnings of the slots may not coincide in different nodes. Therefore a total time of $2T_S$ is necessary to detect a beacon of length T_S. In the following we assume that T_{SCAN} is smaller than T_{BCN}.

A neighbour node sends a beacon in every T_{BCN} long beacon period, while the node performing the neighbour discovery has a beacon period of length T_{bcn}. We assume that the number of beacon and scanning frequencies is N_{BCN}.

The probability P_1 expresses that the reception of a beacon packet at a scanning node is successful.

$$P_1 = \left(\frac{1}{N_{BCN}}\right)\left(1 - \frac{2T_S}{T_{bcn}}\right)(1 - P_{err}), \qquad (1)$$

where $\left(\frac{1}{N_{BCN}}\right)$ is the probability of using the same frequency

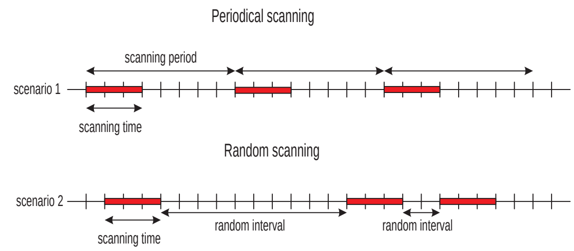

Fig. 5. Scanning Schemes

for scanning as for sending the beacon. The $(1 - \frac{2T_S}{T_{bcn}})$ factor takes into account with a probability of $\frac{2T_S}{T_{bcn}}$ that the beacon packet cannot be detected because the scanning node has to interrupt the scanning in order to send a beacon. $(1 - P_{err})$ gives the probability of having received the beacon successfully, if interference occurs with a probability of P_{err}. Since we assumed that $T_{SCAN} < T_{BCN}$ the probability of having a beacon in a scan period is approximately $\frac{T_{SCAN}}{T_{BCN}}$.

Assuming that the scan periods are independent, we get that the probability of discovering

$$P_{disc} = 1 - \left[1 - \frac{T_{SCAN}}{T_{BCN}} P_1\right]^{\frac{T_{tot}}{T_{SCAN}}} \qquad (2)$$

where T_{tot} is the total amount of time spent with scanning. The limiting case of this formula is

$$P_{disc} = 1 - e^{-P_1 \frac{T_{tot}}{T_{BCN}}} \qquad (3)$$

If we expand the two formulas 2 and 3 in Taylor series, we can see that the difference between these two formulas is very small, so henceforth we can use the formula 3 instead of 2.

As we can see formula 3 does not include T_{SCAN}, which means that the overall performance of the scanning basically depends only on the total time spent with scanning and it is independent of how this total time is divided into scan periods.

We also investigated the case, where T_{SCAN} has a value that is larger or equal to T_{BCN}. In this case we derived other formulas but at the end we got the same results, only the errors of the Taylor series were different.

Fig. 4 shows the simulation results compared to the curve of the analytical model. The dotted line represents the measured

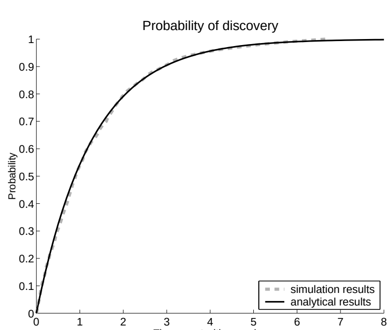

Fig. 4. The probability of discovering a node as a function of the time spent with scanning

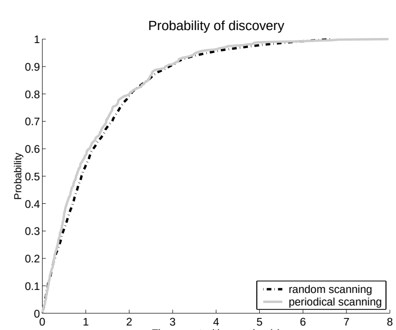

Fig. 6. The effect of the different scanning schemes on the performance of the neighbour discovery procedure

values of the simulations, while the black line corresponds to the calculated values. We can see that in the case of randomized scanning the measurements fit well to the analytical results. It can also be seen, that 2.9s is needed to discover a neighbour with 90% probability, and another 2.9s to discover it with 99%, and in 11.6s a neighbour is discovered with a 99,99% probability.

In the next step, we analyzed two different kinds of scanning schemes (Fig. 5). With these simulations we investigated the difference between deterministic and random scanning periods. In the first case the time between consecutive scans was a pre-defined constant value and in the second case the time between scanning periods was uniformly chosen between 0 and 1000 slots.

In Fig. 6 the probability of the neighbour discovery can be seen as a function of the overall time spent with scanning. The figure shows that in the case of the random scanning scheme (dotted curve) the probability of discovering a neighbour is the above-observed exponential function of the time spent with scanning. We can see that the results of the periodical scanning scheme (grey line) is very similar to the random scanning scheme. There is only a little difference between the two curves. This can be explained by the fact that in the deterministic case the discovery probabilities in consecutive scanning periods are not independent. This effect results in some slight fluctuations in the grey curve.

V. Comparison of SND and inquiry

In the previous sections we described and analyzed the SND procedure. In this section we compare SND with the Bluetooth inquiry (Fig. 7).

The first essential difference between SND and inquiry lies in the fact that while in the inquiry procedure a node that wants to update the information about its neighbours is the one who sends inquiry messages, in the SND solution the nodes that want to be discovered send beacon packets. The second difference is while in inquiry the nodes perform asymmetric roles, the SND algorithm can be carried out in an environment where peer nodes are communicating. The third distinction is that the inquiry is deterministic and guarantees

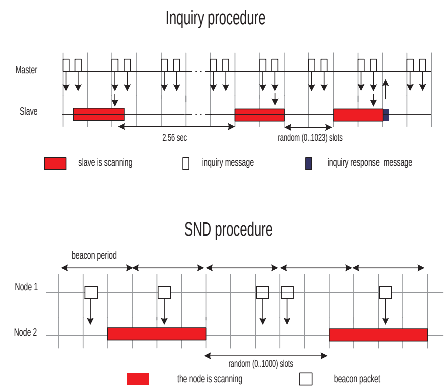

Fig. 8. Asymmetric roles during the neighbour discovery

100% probability of the discovery in a fixed time interval, the SND procedure is probabilistic and can be tuned taking the trade-off between the discovery time and overhead into account.

We compared the two procedures with simulations in two cases: in the case of asymmetric roles and in the case of symmetric roles (Fig. 8).

First we investigated the asymmetric roles. In the case of inquiry this means that one node is performing inquiry and the other wakes up regularly for a given amount of timeslots on a randomly chosen frequency and performs inquiry-scan. In the case of SND one node sends beacons regularly – one beacon in every beacon period – and the other node performs scanning at random time intervals for a given amount of timeslots on a randomly chosen frequency. In our simulations the wake-up period was set to 2.56s, and the nodes scanned for 16 timeslots, which are according to the specification the minimum recommended values for Bluetooth units. The beacon period was 64 slots long, the length of a scanning period was 10 slots and the intervals between scanning periods were chosen uniformly between 0 and 1000 slots.

Fig. 9 shows the probability of discovering a neighbour as a function of the time. In the case of the SND we measure the timeslots spent with scanning and in the case of the inquiry procedure we measure the timeslots spent with the inquiry.

In Fig. 9 we can compare the results of SND and inquiry.

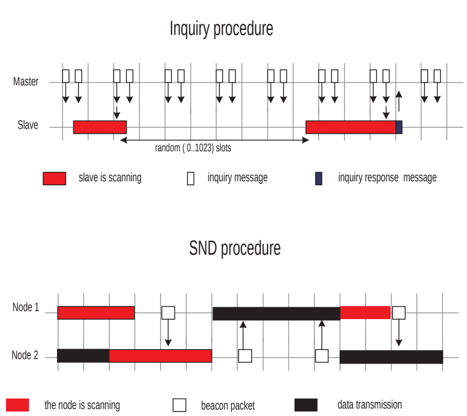

Fig. 7. Comparison of the Bluetooth inquiry and the SND procedure

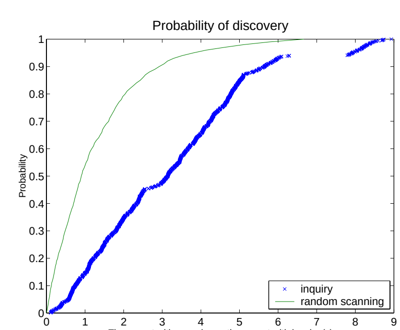

Fig. 9. Comparing the SND and the inquiry procedure in a system where asymmetric roles are assumed

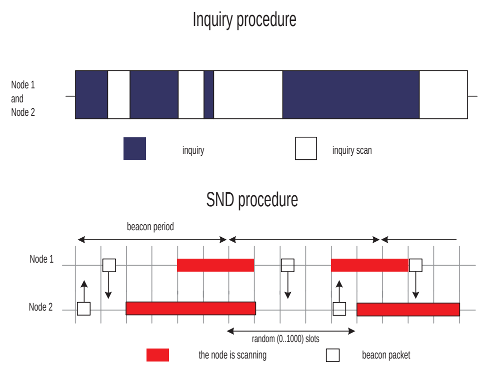

Fig. 10. Symmetric roles during the neighbour discovery

The curve with x-es represents the simulation results of the inquiry procedure and the continuous line represents the results of the SND procedure. In the figure we can see that in the case of the SND procedure the node can be discovered with a probability of 90% within 2.9s, while in the case of the inquiry procedure to discover a node with this probability needs 5.65s. Looking at the partner nodes, the node which performed inquiry scan spent on average 377 slots = 0.24s per minute with scanning, while the node that sent beacon packets spent $\frac{1}{64}$s = 0.94s per minute with this operation. This means that at a 90% probability of discovery the SND procedure consumes 2.9s + 0.94s = 3.84s per minute and the inquiry procedure needs 5.65s + 0.24s = 5.89s per minute. Therefore in the case of the SND procedure more time could be spent for transmitting data than in the case of the inquiry procedure.

In the case of symmetric roles (Fig. 10) the nodes using the inquiry procedure performed the inquiry and the inquiry scan substate in an alternating way – as described in [10] –, where the average value of the time spent with either inquiry or inquiry scan substate is 2000 slots which is approximately 1.2s. The nodes using the SND procedure sent beacons regularly and performed scanning at random time intervals, where the beacon period was 64 slots and the average time between two consecutive scanning was 1000 slots.

The results are shown in Fig. 11, where the curve with x-es represents the results of the inquiry procedure and the continuous line corresponds to the SND procedure. It can be seen that in this case the neighbours are also discovered faster with the SND procedure.

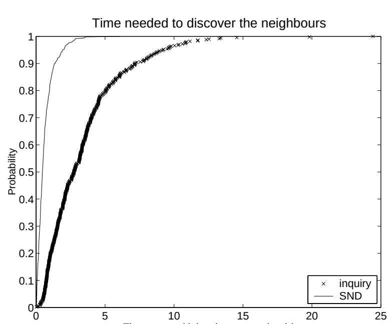

Fig. 11. Comparing the SND and the inquiry procedure in a system where symmetric roles are assumed

VI. Conclusion

In this paper we proposed a Simple Neighbour Discovery (SND) procedure for Bluetooth, which is well suited to systems where peer nodes are communicating. SND is a flexible mechanism that can be performed with low overhead during data transmissions. The nodes performing the SND procedure have a lot of freedom in deciding when they do and how much time they spend with the neighbour discovery. Consequently this procedure can be carried out during data transmission in a time multiplex fashion, taking the data traffic into account.

We investigated the SND procedure with simulation and analytical methods and we gave a comparison of the SND and the Bluetooth inquiry procedures. We found that in the case of SND the probability of discovering a neighbour depends only on the total time spent with scanning and it is independent of how this total time is divided into scan periods. The results show that both assuming asymmetric and symmetric roles among the nodes, the neighbour discovery is faster in the case of the SND procedure, this algorithm takes less time than the Bluetooth inquiry and it is configurable in the trade-off between discovery time and overhead. In the future we would like to make some more simulations with different traffic patterns.

References

[1] Jaap Haartsen: *Bluetooth - The universal radio interface for ad hoc, wireless connectivity*, Ericsson Review No.3, 1998, pp. 110-117

[2] *Bluetooth Baseband Specification version 1.1, http://www.bluetooth.com/*

[3] Roch Guérin, Enyoung Kim, Saswati Sarkar: *Bluetooth Technology Key Challenges and Initial Research*, Conference on Network and Distributed Simulations, 2002

[4] G.V. Záruba, I. Chlamtac, and S. Basagni: *Bluetrees - Scatternet Formation to Enable Bluetooth-Based Ad Hoc Networks*, in the Proceedings of the IEEE International Conference on Communications (ICC) 2001

[5] Z. Wang, R. Thomas, Z. Haas: *Bluenet – A New Scatternet Formation Scheme*, 5th Hawaii International Conference on System Science (HICSS-35), 2002

[6] Márk Félegyházi: *Development and Evaluation of a Dynamic Bluetooth Scatternet Formation Procedure*, Master's thesis, 2001, http://lcawww.epfl.ch/felegyhazi/

[7] Ivan Stojmenovic: *Dominating set based Bluetooth scatternet formation with localized maintenance*, IEEE Int. Parallel and Distributed Processing Symposium and Workshops, 2002

[8] András Rácz, György Miklós, Ferenc Kubinszky, András Valkó: *A Pseudo Random Coordinated Scheduling Algorithm for Bluetooth Scatternets*, MobiHOC 2001

[9] S. Baatz, M. Frank, C. K uhl, P. Martini, and C. Scholz: *Bluetooth Scatternets: An Enhanced Adaptive Scheduling Scheme*, in Proc. 21st Annual Joint Conference of the IEEE Computer and Communications Societies, INFOCOM 2002.

[10] Theodoros Salonidis, Pravin Bhagwat, Leandros Tassiulas, Richard LaMaire: *Distributed Topology Construction of Bluetooth Personal Area Networks*, Infocom 2001

[11] Ching Law, Kai-Yeung Siu: *A Bluetooth Scatternet Formation Algorithm*, in the Proceedings of the IEEE Symposium on Ad Hoc Wireless Networks 2001

[12] Zs. Haraszti, I. Dahlquist, A. Faragó, T. Henk: *PLASMA – An Integrated Tool for ATM Network Operation*,
Proceedings of International Switching Symposium, 1995.

[13] Miklós Aurél Rónai: *Packet–Level Simulation of the Bluetooth Physical Layer*, Master's Thesis, 2001, http://www.ronai.hu