Formation of Bluetooth Scatternet and Its Delay Analysis

Debasish Chakraborty, G. Chakraborty†,
Sagar Naik ‡, B.B.Bista† and Norio Shiratori ††

Tohoku JGN-II Research Center, † Iwate Prefectural University, ‡ University of Waterloo,
†† RIEC, Tohoku University

Abstract

Bluetooth device discovers other Bluetooth devices, and establishes connections with those devices. Device discovery is achieved by the master running the Inquiry protocol and slaves running the Inquiry Scan protocol. The master and a slave establish a connection using the Page and Page Scan protocols, respectively. Once a connection is established between a master and a slave, upper level data can be exchanged between the two devices. Thus, device discovery and connection establishment are fundamental to communication between two Bluetooth devices. While a device is in the process of discovering other devices, there is a random delay. We analyze the discovery time for different situations. We further show the delay-distribution for varied number of devices. The results were obtained by simulation which is thorough to every details of the protocol specification. We explain reasons of occasional long discovery time, and propose means to improve it.

1. Introduction

Bluetooth (BT) specification version 1.1, published in 2001, is an industry standard for providing low power, low-cost, short-range wireless radio link to connect devices like a head phone, a printer, an Internet access point etc. to a host like a mobile PC or a Palm. Alternatively, it could be like gathering information about restaurants, game-centers, in a departmental stores floor, using a palm or a mobile-phone equipped with BT module. Device discovery simplifies the task of finding and connecting to different services available locally. It helps to organize a mobile ad-hoc network tailored to the requirement. Alternately, it simplifies the task of building and maintaining a network, especially when new services are introduced. With rising number of such services to be linked dynamically and without cable, efficient node discovery plays an important role in overall performance of the BT wireless network. There have been innovative efforts in using the technology to form personal area network and multi-hop ad hoc networks [3, 5].

The baseband protocol takes care of the basic tasks of device discovery and subsequent connection. The Inquiry protocol runs at the discovering end, also called the Master. It tries to locate devices in its vicinity. Inquiry scan protocol is executed by those devices which offer services and wait to be discovered. They are called slaves. Once the discovering device discovers a slave, the two devices can go to the connection state through Paging protocol. Both these protocols run at the baseband level.

Most of the previous research works in BT service discovery are in the higher layer at Service Device Protocol (SDP), to enhance its functionalities or compare its functionalities with competitive technologies. To our knowledge, no work has yet been done, or reported any result about the performance of device discovery at the baseband level. But, device discovery at the baseband level is the core task, on which SDP is based. We implemented a complete BT network environment by software simulation, with full details of the baseband protocols, to do different performance experiments. We made the observation that the device discovery depends on the density of the devices and the random backoff time. We also indicate how to improve the discovery delay.

The rest of the paper is as follows. The procedures of *Inquiry* and *Inquiry scan*, which are the basis for finding other BT nodes, is described in Section 2. Related work has been discussed in Section 3. An estimation of the delay in device discovery is done in Section 4. Simulation results are described in Section 5, and we conclude the paper in Section 6.

2. Inquiry and Inquiry Scan Procotols

The device discovery protocol consists of two parts, namely Inquiry and Inquiry Scan. A discovering device (or, master) runs the Inquiry protocol and a device willing to be discovered (or, slave) runs the Inquiry Scan protocol. The two protocols have been explained in text form in the Bluetooth baseband specification [4]. However, for the purpose of clear understanding, we presented them in the form of state-transition diagram.

In a cluster of BT enabled devices, before any communication or connection is established, all nodes are in **Standby** state. For a device to change from an uncommunicating **Standby** state to a communicating **Connection** state, it has to go through two substates, namely **Inquiry** and **Page**. In fact, a device can go to **Inquiry** state from **Connection** state too, time to time, to discover newer nodes. Even then, the basic steps of device discovery are the same. The inquiry

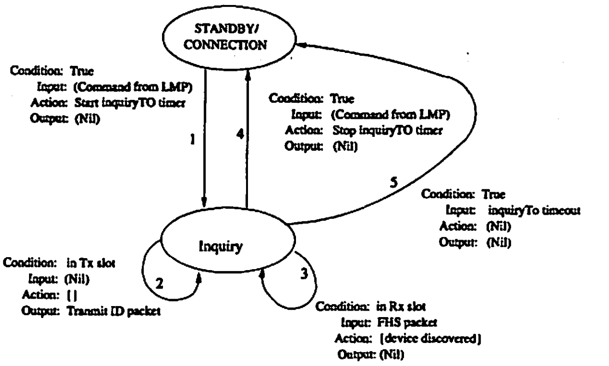

Figure 1: State transition diagram of *Inquiry* Protocol

procedure enables a unit to discover which other units are within the short range of BT communication. Paging procedure is to enable the actual connection. We will only discuss the steps involved in discovering other devices. We will also explain details using timing diagrams of different packets transmitted and received.

2.1 The Inquiry Protocol

The state-transition diagram of the Inquiry protocol is shown in Fig.1. The protocol consists of two states: STANDBY or CONNECTION and INQUIRY. Initially, a device could be in the STANDBY or the CONNECTION state. It is the Link Manager Protocol (LMP) layer that decides when the baseband layer may initiate the Inquiry protocol. At this point, the LMP layer asks the baseband layer to enter the INQUIRY state (#1). In #1, the baseband layer starts a timer called ***inquiryTO***. The baseband layer moves from the INQUIRY state to the original STANDBY/CONNECTION state in two ways, by transition #4, occurs when it is asked to do so by LMP and #5, occurs when the ***inquiryTO*** timer expires. Two kinds of state transitions occur depending on whether it is in a Tx (transmission) slot or a Rx (reception) slot, as shown in transition #2 and transition #3, respectively. In Tx slot the discovering device transmits two ID packets, one at the beginning and the other at the middle of the slot. In Rx slot, transition #3 takes place, when the reception of an FHS packet signals the discovery of a device willing to be discovered. In transition #3, the baseband layer informs the LMP of the discovery of a device, and the master device remains in the INQUIRY state. It may be noted that the discovering device does not acknowledge the reception of an FHS packet.

In inquiry hopping sequence a set of 32 frequencies is partitioned into two subsets of 16 frequencies each, and two trains, namely train A and train B, are constructed from those two subsets. They are different permutation of different frequencies. As an ID packet is transmitted twice in each Tx slot, it takes eight Tx slots to cover the whole train of 16 frequencies. Because the Tx and Rx

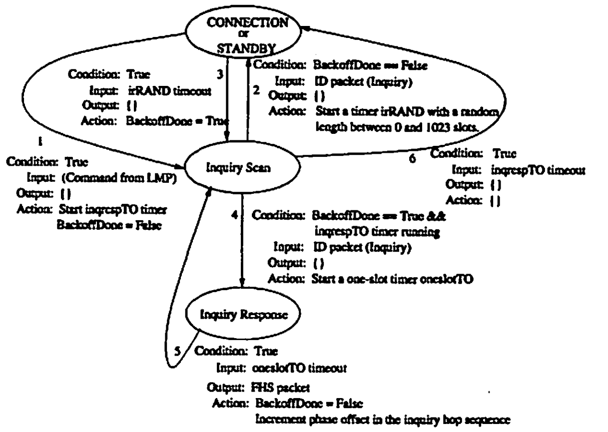

Figure 2: State transition diagram of *Inquiry Scan* Protocol

slots are interleaved, the total length of time to cover each train is 16 slots, or $10ms (= 16 \times 625\mu sec, 625\mu sec$ being the duration of one time-slot). A single train is repeated for at least $N_{inquiry} = 256$ times before the other train is used. In order to collect all responses in an error-free manner, at least four trains or in other words three train switches must have taken place. Therefore, the INQUIRY substate may have to last for 10.24 sec. ($256 \times 10 \times 4msec$). If desired, the inquirer can also prolong the inquiry substate to increase the probability of receiving all responses in an error-free manner.

2.2 The Inquiry Scan Protocol

The state-transition diagram of the Inquiry Scan and Response protocol is shown in Fig.2. If a device wants to be discovered by others, then the LMP layer of the device issues a command to the baseband layer to move from its present STANDBY/CONNECTION state to the Inquiry Scan substate, which is shown as transition #1. In this transition the device starts the inqrespTO timer, and initializes a ***BackoffDone*** flag to false. Here we explicitly specify the backoff mechanism for a clear understanding of the protocol. If the device receives an ID packet and the backoff has not been done, then it starts an inquiry response timer irRAND, as we name it, with a duration randomly set between 0 and 1023 slots and moves to its original state. If the device receives a second ID packet in the Inquiry Scan state, then it starts another one-slot timer we call oneslotTO, after expiry of which it moves to the Inquiry Response state. After waiting for exactly one slot in the Inquiry Response state, the device transmits a FHS packet, reinitializes the BackoffDone flag to false, and moves to the Inquiry Scan state. After the inqrespTO timer expires, the device moves from the Inquiry Scan state to its original STANDBY/CONNECTION state.

The idea behind having the responding devices to wait for a random period is to avoid collision of FHS packets from devices that receive the same first ID packet from a discovering device. When many devices are simultaneously competing to be discovered, there is a finite probability of this to happen.

3. Related Research

Delay analysis of the device discovery protocol and proposals to modify the protocol to reduce the delay have recently been a topic of research interest [1, 2, 6]. Zaruba and Gupta [10] simplify the analysis of the discovery protocol by assuming that none of the trains – A or B–is repeated before a train switch. Connection establishment between two mutually unknown devices takes 5.76 seconds in a typical case and may take as long as 23 seconds in the worst case [1]. Thus, for Bluetooth technology to be useful in a mobile environment, there is a need to speed up the discovery time. Welsh, Murphy and Frantz [9] propose three possible changes to the Bluetooth specification in order to reduce the discovery time. These suggestions are: (i) eliminate or decrease the random backoff delay in the Inquiry Scan protocol, (ii) use a single frequency train instead of two A and B–in the Inquiry protocol, and (iii) combine the first two. Salonidis, Bhagwat and Tassiulas [7] have shown that device discovery time can be reduced by having devices alternate between two roles, namely, a discoverer and one willing to be discovered.

4. Analysis of Discovery Time

In this section, we will analyze the discovery time.

T_{ID1} : This is the average time taken by a slave to receive the first ID. This time is measured from the instant the master starts transmitting ID packets to the instant when a slave receives an ID.

T_b : This is the average of the random backoff time after the slave receives the first ID packet.

T_{ID2} :This is the average time taken by the slave to receive a second ID. This time is measured from the instant the slave comes out of its random backoff to the instant when it receives an ID.

1-slot : This is the 1-slot wait by the slave between receiving a second ID and transmitting an FHS packet. We will ignore this 1-slot component from further consideration.

Therefore, discovery time, denoted by T, is expressed as follows. $T = T_{ID1} + T_b + T_{ID2}$.

The Bluetooth specification states that a random backoff interval be uniformly chosen from the range [0, 1023] slots. Thus, the average value of Tb is 512 slots. There is a similarity between T_{ID1} and T_{ID2}, because both these intervals are associated with the same condition. For T_{ID1} we have assumed that a slave enters its Inquiry Scan state no later than the master entering the Inquiry state. Essentially, what this assumption implies is that by the time the master starts transmitting ID packets, a slave has already begun or just begins scanning the medium to receive an ID packet. We measure T_{ID2} from the instant a slave enters Inquiry Scan for a second time to when it receives a second ID. Thus, T_{ID1} and T_{ID2} can be computed in the same way. S, we can write,

$T = 2 * T_{ID} + T_b$,

where $T_{ID} = T_{ID1} = T_{ID2}$ represents the average time taken by a slave to receive an ID packet from the instant when the master is transmitting ID packets and the slave is scanning the medium. Thus, to give an estimation of T, we need to estimate T_{ID}.

Next, we discuss the impact of the value of k (phase difference) on discovery time, and will show that this phase plays an important role in determining the delay. Let $F = A \cup B$ be the set of 32 frequencies used in the discovery protocol. Members of F are denoted by f_i, where $1 \leq i \leq 32$. A slave scans the medium on the same frequency, say f_i for 2048 slots. A+ and B+ represent repetition of the A train and the B train for 256 times. Since a slave scans the medium on the same frequency for 2048 slots and the length of each of A+ and B+ is 4096 slots, the range of phase difference to be considered is [0, 2046] slots. In other works, we have $0 \leq k \leq 2046$. Moreover, we consider only the even values of k in the range [0, 2046]. This is because the master transmits ID packets in its Tx slots, and remains in receive mode in the Rx slots to catch an FHS packet.

To show the impact of the value of k on discovery time in an informal manner, we consider two values of k, such as k = 0 and k = 16. These two cases are discussed in the following. On the one hand, if k = 0, a slave scans the medium on two different frequencies during the period when the master transmits ID packets using A+ frequencies. Thus, if $f_{r0} \in A$ or $f_{r1} \in A$, the ID packet will be received by the slave. On the other hand, if $k > 0$, a slave scans the medium on three different frequencies during the period when the master transmits ID packets using A+ frequencies. This is evident from Fig.3(b). Thus, device discovery time is expected to be smaller in the latter case. We can further identify another important value of k, namely k = 16, as follows. Let us assume that k takes on a value less than 16, say 14. In 14 consecutive slots there are 7 Tx slots and 7 Rx slots. Since the master can transmit two ID packets in each Tx slot on two different frequencies, only 14 out of 16 A train frequencies will be potential match for f_{r2} as shown in Fig.3(b) (if it is not already matched during A-train). Thus the probability of receiving an ID packet due to f_{r2} matching with one of the frequencies in the period Δt is smaller for smaller values of k in the range [2, 8]. A similar argument can be used to show that the two ranges of k, namely [2, 8] and [2034, 2046], lead to similar discovery time. When $16 \leq k \leq 2032$, f_{r2} can scan all the

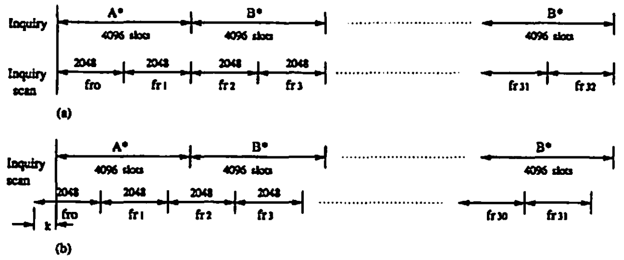

Figure 3: Situations with different Δt

A as well as B train of frequencies, and the match is certain at this stage, if not before. Therefore, we need to consider four distinct ranges of k as follows to give a model of T_{ID}.
(i) $k = 0$
(ii) $2 \leq k \leq 14$
(iii) $16 \leq k \leq 2032$
(iv) $2034 \leq k \leq 2046$
Cases (ii) and (iv) lead to similar delay behavior.

Now we obtain an expression for T_{ID} by combining the expressions for T_{ID}, (k=0), T_{ID}, $(2 \leq k \leq 14)$, and T_{ID}, $(16 \leq k \leq 2032)$. We can write an expression for T_{ID} as follows:

$$T_{ID} = \frac{1}{1024} \times \left(T_{ID,(k=0)} + 2 \times \sum_{k=2}^{14} T_{ID,(2 \leq k \leq 14)} + \sum_{k=16}^{2032} T_{ID,(16 \leq k \leq 2032)} \right) \quad (1)$$

It is possible to calculate the average delay for different values of k for different combinations of A-train, B-train, and slave scanning frequencies with their corresponding probabilities. The details is beyond the scope of this paper, but the final value of T_{ID} is calculated to be approximately 1141 time slots.

5. Simulation Result

We have developed a Bluetooth simulator to study the performance of the device discovery protocol. In the simulator, we have implemented the frequency hopping kernel box specified in Chapter 11 of the Bluetooth specification [4]. From the analysis in Section 4, we obtained discovery time for different ranges of values for k, namely $k = 0, 2 \leq k \leq 14, 16 \leq k \leq 2032$, and $2034 \leq k \leq 2046$, where the second and the fourth blocks of k values are treated alike.

We also wanted to compare the analytical model with simulated data. Using the expression for T_{ID} given in Equation (1), we computed T_{ID} to be 1141 slots. If the random backoff range is [0,1023], the average value of random backoff values chosen from the range is 512 slots. Thus, the average value of discovery time, when there are only 2 BT devices, one master and one slave, could be computed as $T = 2 \times 1141 + 512 = 2794$ slots.

The simulator takes input parameters such as the number of devices and backoff limit. Each data point

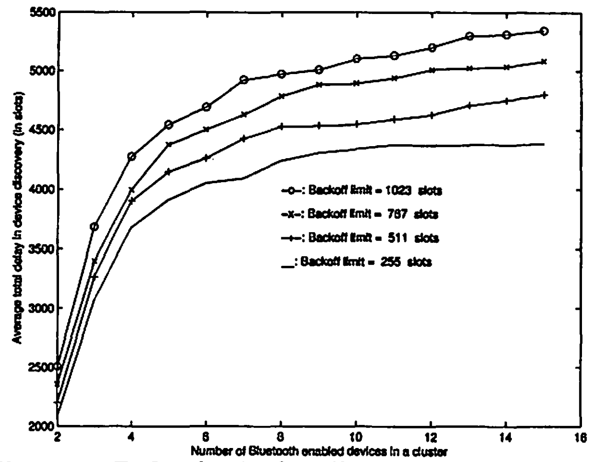

Figure 4: Delay in device discovery as a function of cluster size

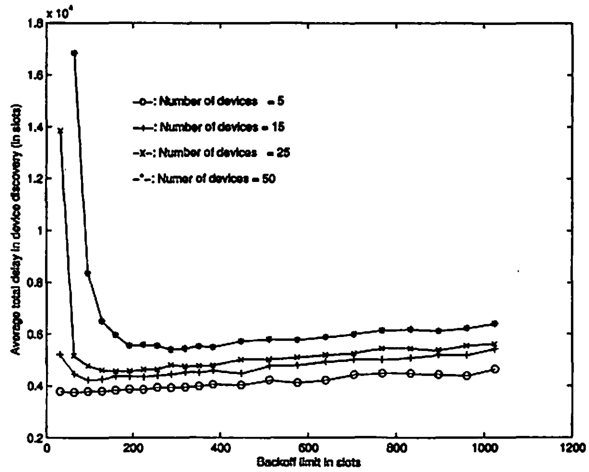

Figure 5: Delay in device discovery as a function of backoff limit

in Fig.4 and 5 are averages of 1000 independent experiments. In Fig.4, we assume that there are N number of devices within the radio propagation area of each other. One device performs the Inquiry protocol and the rest N-1 devices perform Inquiry Response. We have performed experiments for different values of the backoff period. The y-axis shows the average time to discover a device as a function of N. For N=2, i.e., one Inquirer and one Inquiry scanner situation, the discovery delay conforms with our theoretical analysis.

As predicted, the average discovery time increases rapidly when the number of devices is increased. Also, it is observed that when the backoff time is more, the delay increases. When backoff time is reduced to very low value, the probability of collision from two or more FHS packets increases, resulting in longer discovery delay. This is shown in Fig.5.

In Fig.5, the y-axis shows the average time of device discovery as a function of the backoff limit for different values of N as explained above. Though a backoff limit of 1024 has been suggested in the specification to reduce the possibility of collision, it simply leads to a

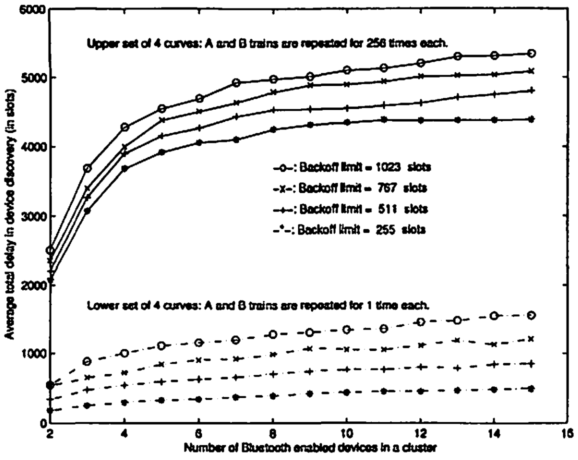

Figure 6: Average total delay in device discovery as a function of total number of devices

larger discovery time for smaller values of N. Discovery delay improves with decreasing backoff time up to some point, after which collision of FHS becomes the predominant factor and the discovery delay increases. For N=50 the optimum delay is obtained when backoff time is ≡ 250, whereas for N=5 the optimum delay is obtained when backoff is ≡ 70.

In order to reduce discovery time, there are proposals to avoid repetition of the individual trains for 256 times [8]. Thus, rather than use a sequence of the form A+ B+ A+ B+..., the suggested sequence is A B A B However, the effect of the proposal on discovery time has not been studied in detail. To observe how the proposal affects discovery time, we simulated delay behavior when frequency trains are not repeated, and are switched alternatively. The results are in Fig.6, where we show how the discovery time varies with increasing number of potential slaves for the two cases of inquiry sequence, namely A+ B+ A+ B+ and A B A B, for different fixed values of the backoff limit.

There is another proposal to improve discovery time by eliminating or reducing the random backoff time. However, it is clear from Fig.5 that it is not a good idea to eliminate the concept of random backoff. Instead, a small backoff limit, say, 200 to 300 slots, can be used for an overall low discovery time, even when there are large number of contending slaves.

6. Discussion and Conclusion

We performed a large number of experiments to obtain insights into the distribution of discovery time. We observed how the total discovery time varies with the number of potential slaves when the backoff limit is constant. We also observed how the total discovery time varies with the backoff limit when the number of potential slaves is constant. With small values of backoff limit, discovery time could be too long, if there are more than 10 potential slaves in the same area. By performing additional simulations we observed the effectiveness of two proposed improvements to device discovery. Specifically, it is useful to avoid repetitions of the individual A and B trains before a train switch. On the other hand, complete elimination of the idea of random backoff is not a good idea as discovery time will be too long. Rather, choosing a small backoff limit, say, 200 to 300 slots, is useful in reducing discovery time even when the number of potential slaves in the same area is large.

References

[1] A. Busboom, I. Herwono, M. Schuba, and G. Zavagli. Unambiguous device identification and fast connection setup in bluetooth. In Proc. of European Wireless 2002, Florence, Italy, Feb., 2002.

[2] G. Chakraborty, K. Naik, D. Chakraborty, N. Shiratori, and D. Wei. Delay analysis and improvement of the device discovery protocol in bluetooth. In IEEE Vehicular Technology Conference, Fall 2004, Los Angeles, USA, Sept., 2004.

[3] A. Conti, D. Dardari, G. Pasolini, and O. Andrisano. Bluetooth and ieee 802.11b coexistence: Analytical performance evaluation in fading channels. IEEE Journal on Selected Areas in Communications, Vol. 21, No.2, pages 259–269, Feb. 2003.

[4] The Bluetooth Special Interest Group. Baseband specification version 1.1. (www.bluetooth.com), 2001.

[5] Y. Liu, M.J. Lee, and T.N. Saadawi. A bluetooth scatternet-route structure for multihop ad hoc networks. IEEE Journal on Selected Areas in Communications, Vol. 21, No.2, pages 229–239, Feb. 2003.

[6] P.Murphy, E.Welsh, and J.P. Frantz. Using bluetooth for short-term ad hoc connections between moving vehicles: A feasibility study. In Proc. of IEEE Vehicular Technology Conference, Spring 2002, pages 414–418, May, 2002.

[7] T.Salonidis, P.Bhagwat, and L.Tassiulas. Proximity awareness and fast connection establishment in bluetooth. In Proc. of Mobile and Ad Hoc Networking and Computing (MobiHoc), Boston, pages 141–146, August, 2000.

[8] F. Siegemund and M. Rohs. Rendezvous layer protocols for Bluetooth-enabled smart devices. In Proc. of Intl. Conference on Architectures of Computing Systems, Karlsruhe, Germany, pages 256–273, April, 2002.

[9] E.Welsh, P. Murphy, and J.P. Frantz. Improving connection times for bluetooth devices in mobile environment. In Proc. of Intl. Conference on Fundamentals of Electronics, Communications, and Computer Science, March, 2002.

[10] G.V. Zaruba and V. Gupta. Simplified Bluetooth device discovery – analysis and simulation. In Proc. of the 37th Hawaii International Conference on System Sciences, 2004.