Analysis of Bluetooth Device Discovery and Some Speedup Mechanisms*

Jehn-Ruey Jiang§, Bing-Rong Lin†, and Yu-Chee Tseng†

§Department of Computer Science and Information Engineering
National Central University, Taiwan

†Department of Computer Science and Information Engineering
National Chiao-Tung University, Taiwan

jrjiang@hcu.edu.tw, linnbiro@csie.nctu.edu.tw, yctseng@csie.nctu.edu.tw

Abstract

The device discovery time of Bluetooth is prohibitively long. This may significantly impact many mobile applications. In this work, we start by analyzing the frequency-matching delay of Bluetooth for both versions V1.1 and V1.2. We then propose three schemes to speed up the device discovery procedure of Bluetooth. The result is a significant reduction of average frequency-matching time from 23.55 seconds to 11.38 seconds.

Keywords: Bluetooth, device discovery, frequency-hopping spread spectrum (FHSS), inquiry and scan, wireless network.

1 Introduction

Bluetooth [2] is a promising technology for short-range, low-power wireless communications. Operating in the 2.4GHz license-free ISM (Industrial, Scientific-Medical) band, Bluetooth adopts a 79-channel *Frequency Hopping Spread Spectrum (FHSS)*[1] technology with a hopping rate of 1600 hops per second. In Bluetooth, before any two devices can communicate with each other, they must go through a device discovery procedure which consists of two steps, *inquiry* and *paging*. The former is for devices to find each other, while the latter is to establish actual connections. According to the specification [2], the inquiring procedure may take 10.24 seconds or longer, and the paging, 7.68 seconds or longer. This long connection setup time is fine for static applications, but is intolerable for mobile applications demanding quick and short connections, such as multi-media name card exchange [4] and pedestrian surroundings information retrieval [9]. Consequently, many approaches [1, 4, 5, 6, 7, 8, 9] have been proposed to speed up the Bluetooth device discovery procedure.

One major component in the discovery delay is the long *frequency-matching* time. Bluetooth adopts a master-slave architecture. To establish a connection between two devices, a potential master should be in the *inquiry* state

*Y. C. Tseng's research is co-sponsored by the MOE Program for Promoting Academic Excellence of Universities under grant number 89-E-FA04-1-4, by NSC of Taiwan under grant numbers NSC92-2213-E009-076 and NSC92-2219-E009-013, by the Institute for Information Industry and MOEA, R.O.C, under the Handheld Device Embedded System Software Technology Development Project, and by the Lee and MTI Center of NCTU.

[1]The number of channels may be reduced to 23 in certain countries.

to periodically send consecutive *ID* packets on some predefined 32 channels (or frequencies[2]), and a potential slave should be in the *inquiry scan* state trying to catch an *ID* packet from the right channel at the right time. Only when a frequency-matching occurs, i.e., the slave correctly receives an ID packet, can the inquiry-paging procedure be started.

A lot of works [3, 4, 6, 7, 8, 9, 10] have addressed the Bluetooth device discovery speedup problem. Some [4, 6, 7, 9] suggest to modify the device discovery parameters, some [3, 10] suggest to use auxiliary devices, while some [8] relies on device cooperation to assist device discovery. The recent Bluetooth specification V1.2 also proposes a "faster connection" based on the concept of interlaced inquiry scan frequencies.

In this work, we start by analyzing the frequency-matching time of Bluetooth, the major component of delay in its device discovery, for both versions V1.1 and V1.2. We show through analysis that the average delay is about 23.55 seconds. This motivates us to search for schemes to shorten the frequency-matching time. In this paper, three schemes are proposed. The reduction is shown to be significant.

The rest of this paper is organized as follows. Section 2 presents some backgrounds. In Section 3, we analyze the frequency-matching delay of Bluetooth's device discovery. Section 4 presents our schemes. Concluding remarks are drawn in Section 5.

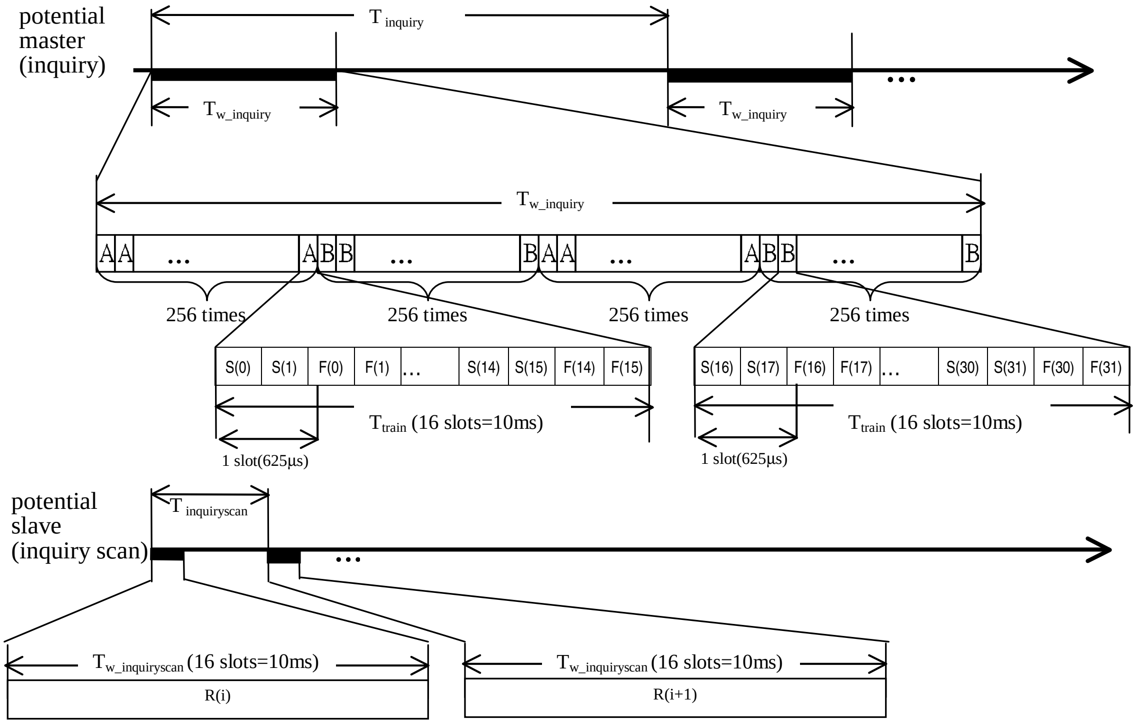
2 Backgrounds

2.1 Inquiry and Paging Procedures of Bluetooth

The device discovery in Bluetooth involves two steps: *inquiry* and *paging*. The inquiry procedure is asymmetric. A potential master must enter the INQUIRY state first, and a potential slave must enter the INQUIRY SCAN state. The master will periodically broadcast ID packets in every $T_{inquiry}$ interval (refer to Fig. 1). These ID packets are hopping on 32 common channels. These 32 channels are divided into two sets, each with 16 channels. ID packets are grouped into *A trains* and *B trains*, each using one of the two sets of 16 channels exclusively. In a $T_{w\_inquiry}$ interval, $N_{inquiry}$ A trains, followed by $N_{inquiry}$ B trains, $N_{inquiry}$ A trains, and $N_{inquiry}$ B trains of ID packets are sequentially transmitted, where $N_{inquiry} = 256$. Each train consists of 16 slots (of length $T_{train} = 10$ ms). Two ID packets on two different channels are placed in one 625-μs slot. So there are 8 slots of ID packets interleaved by 8 response slots reserved for slaves to reply. Consequently, $T_{w\_inquiry}$ takes up to 10.24 seconds to complete (4×256 of A/B trains, each of 10 ms), unless the master has collected enough ($\geq N_{inquiry\_responses}$) responses and determines to abort the INQUIRY procedure earlier. For example, one commonly selected setting is that masters enter the INQUIRY state every one minute, i.e., $T_{inquiry}$=60 sec.

A potential slave should enter the INQUIRY SCAN state to listen to ID packets (refer to Fig. 1). It sequentially hops on the aforementioned 32 channels, but at a much slower speed. It takes $T_{inquiryscan}$ seconds to hop from one channel to another. In each hop, it only enters the listening status for $T_{w\_inquiryscan}$=10 ms. Note that it is necessary that $T_{w\_inquiryscan} \geq T_{train}$ so as to guarantee that the slave can catch an ID packet from the master. The Bluetooth specification suggests that $T_{inquiryscan}$ be no longer than 2.56 seconds, which equals the length of $N_{inquiry}$ A/B trains. Note that many vendors set $T_{inquiryscan} = 1.28$ seconds, which will also be adopted in this paper. Table 1 summarizes all the above timing parameters.

[2] In this paper, the word "channel" and the word "frequency" are used interchangeably.



S(*i*) stands for sending ID packet in inquiry hopping frequency channel *i*, *i*=0..31.
R(*i*) stands for listening to ID packet in inquiry hopping frequency channel *i*, *i*=0..31.
F(*i*) stands for listening to FHS packet in inquiry hopping frequency channel *i*, *i*=0..31.

Figure 1. Bluetooth inquiry procedure.

Table 1. Timing parameters of inquiry and inquiry scan.

Parameter	Description	Recommended value
$T_{inquiry}$	inquiry interval	60s
$T_{w\_inquiry}$	inquiry window length	10.24s
$T_{inquiryscan}$	inquiry scan interval	1.28s
$T_{w\_inquiryscan}$	inquiry scan window length	10ms
T_{train}	length of a train	10ms
$N_{inquiry}$	train repetition number	≥ 256

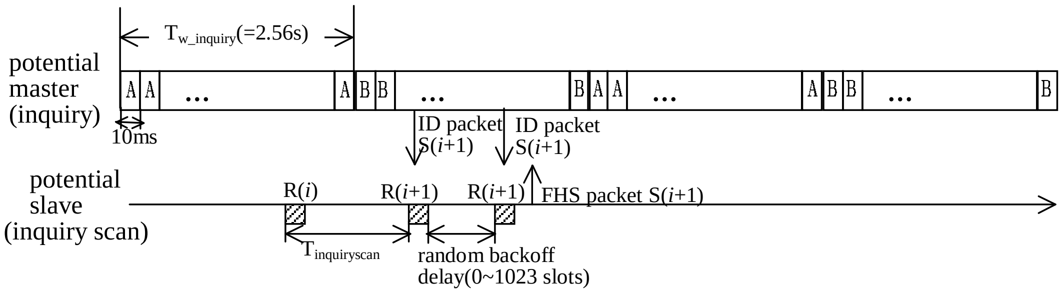


Figure 2. The backoff procedure for a slave to reply a FHS packet.

Upon receiving an ID packet from some channel, say i, a slave should take a random backoff and then reply a *Frequency Hopping Synchronization (FHS)* packet via the same channel. The backoff value is between 0 to 1023 slots to avoid possible collisions with other slaves. After the backoff, the slave should continuously listen to channel i and reply a FHS immediately after the first ID packet (also on channel i) is heard. Fig. 2 illustrates this procedure. Note that the average backoff value is 512 slots, which equals 32 trains. This explains why A/B trains need to be repeated so many times.

2.2 Related Work

Several methods have been proposed to improve the Bluetooth device discovery procedure [3, 4, 6, 7, 8, 9, 10]. Some schemes try to modify the device discovery parameters [4, 6, 7, 9]. Some schemes propose to use auxiliary devices [3, 10], while some relies on device cooperation to assist the discovery [8].

In [9], three methods are proposed. The first method tries to decrease or even eliminate the random backoff in INQUIRY SCAN, the second method uses one single 32-frequency train to replace the two 16-frequency trains in INQUIRY, and the last method is a hybrid one to combine the first two methods. According to [9], these methods can improve the connection setup time up to 75% without deteriorating the overall system performance. A hardware empirical testbed is developed to verify these methods in [6]; the result suggests that a single train with no backoff has the best performance. In [4, 7], each device is assumed to alternate between "potential master" and "potential slave" modes in a random fashion. Analysis and simulation results show that the connection establishment latency can be reduced to be 80 ms with a probability of 0.95. In [3, 10], it is suggested to use auxiliary devices, such as IrDA interfaces or RFID transponders, to facilitate connection setup. In [8], a cooperative device discovery scheme is proposed to allow devices to exchange their knowledge of nearby devices, such as BD addresses and clocks, to speed up device discovery. The recent Bluetooth specification V1.2 also proposes a mechanism which requires a device to perform inquiry scan with interlaced hopping frequency in A and B trains.

3 Analyses for Bluetooth Device Discovery

In this section, we analyze the frequency-matching time of Bluetooth V1.1 and V1.2, which is the major component of delay in its device discovery. We start with the analysis for Bluetooth V1.1. Suppose that there is already a master device performing the scan procedure. According to whether or not the master is sending ID packets, we divide the time axis into *inquiry windows* and *non-inquiry windows*. Now suppose that there is a slave device tuning to the inquiry scan procedure and starting with an inquiry scan window. We are interested in the frequency-matching delay, denoted by D, measured by the elapsed time from the time when the slave starts inquiry scan to the time when it successfully receives an ID packet from the master.

By investigating the timing diagram of Fig. 1, the slave may start its inquiry scan in an inquiry window with probability $\frac{T_{w\_inquiry}}{T_{inquiry}}$, and in a non-inquiry window with probability $\frac{T_{inquiry}-T_{w\_inquiry}}{T_{inquiry}}$. So we have

$$D = \frac{T_{w\_inquiry}}{T_{inquiry}} \times X + \frac{T_{inquiry} - T_{w\_inquiry}}{T_{inquiry}} \times (\frac{T_{inquiry} - T_{w\_inquiry}}{2} + \frac{T_{inquiryscan}}{2} + Y), \qquad (1)$$

where X is the expected delay after the slave starts its inquiry scan and Y is the expected delay after the slave's first inquiry scan encounters the master's first inquiry window. Note that in the second case, the slave has to wait

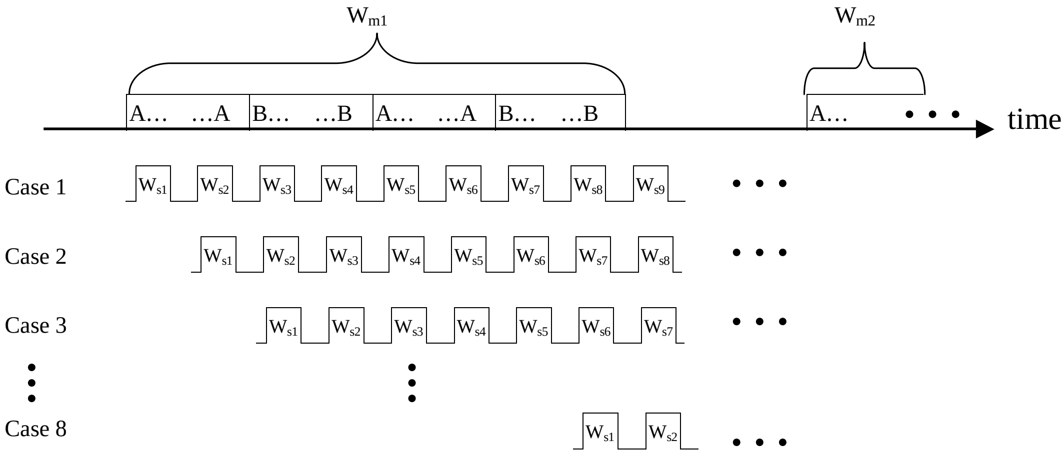


Figure 3. Eight possible cases for slave to start its inquiry scan. W_{mi} **is the i-th inquiry window of the master, and** W_{si} **is the i-th inquiry scan window of the slave.**

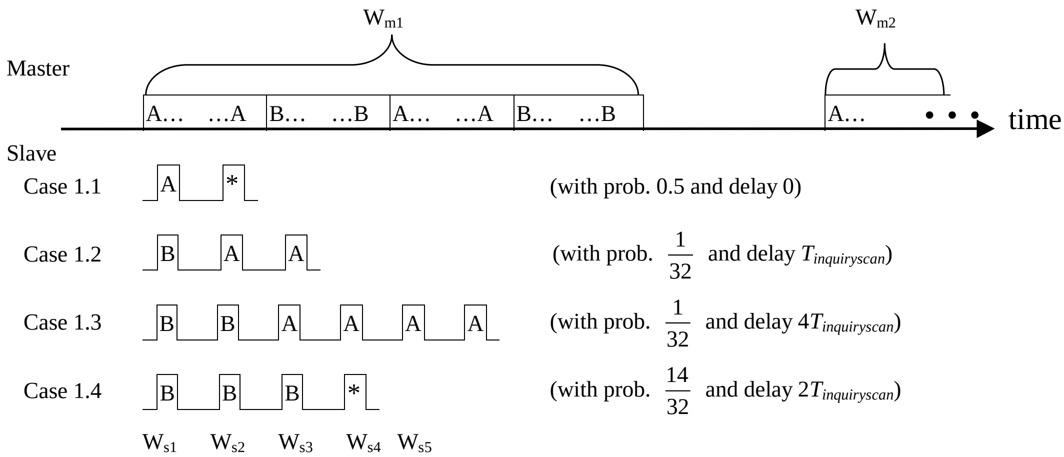


Figure 4. Illustration of Eq. (2), which contains four subcases of case 1 for frequency-matching between a master and a slave. A "*" means a "Don't Care" frequency, because a matching has already appeared in the previous inquiry scan window.

$\frac{T_{inquiry}-T_{w\_inquiry}}{2}+\frac{T_{inquiryscan}}{2}$ time in average before its first inquiry scan window encounters the master's first inquiry window.

In the following analysis, we follow the recommended values of Bluetooth that the length of one inquiry scan interval is one half of a sequence of 256 A/B trains. Therefore, the slave has two chances to match with the frequencies on which the master sends ID packets. Now, to calculate the expected value of X, we have to consider all possible locations where the first inquiry scan window of the slave (denoted by W_{s1}) appears in the first inquiry window of the master (denoted by W_{m1}). Basically, we evenly divide the window W_{m1} into 8 partitions, as illustrated in Fig. 3. There are 8 cases to consider, which are discussed in the following.

Case 1: (W_{s1} in the first $\frac{1}{8}$ window of W_{m1}) In this case, the delay will depend on the frequencies on which the slave is waiting for the master's ID packets. Recall that the slave will repeatedly scan all frequencies of train A in 16 consecutive inquiry scan windows, followed by all frequencies of train B in 16 consecutive inuiry scan windows. So there are 32 possibilities where the slave can catch an ID packet on the right frequency from the master. These

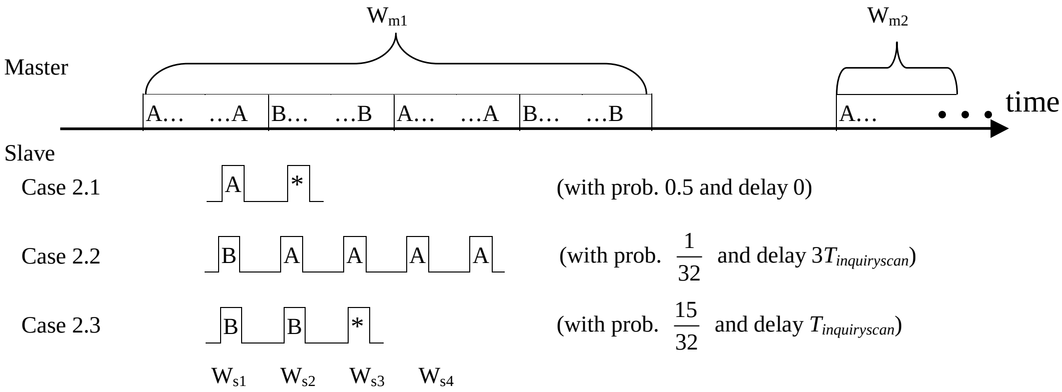


Figure 5. Illustration of Eq. (3), which contains three subcases of case 2 for frequency-matching between a master and a slave.

possibilities can be classified into 4 subcases, as illustrated in Fig. 4. So the expected value of X in this case can be approximated by

$$X_1 = \frac{1}{32} \times T_{inquiryscan} + \frac{1}{32} \times (4 \times T_{inquiryscan}) + \frac{14}{32} \times (2 \times T_{inquiryscan}). \tag{2}$$

Note that there is no delay for case 1.1 in Fig. 4. For case 1.2, the delay is one inquiry scan interval, as reflected in the first term of Eq. (2). Similarly, there are four and two inquiry scan intervals of delays for cases 1.3 and 1.4, respectively.

Case 2: (W_{s1} in the second $\frac{1}{8}$ window of W_{m1}) As described in case 1, the slave hops on 32 frequencies repeatedly. Similarly, there are also 32 possibilities where the slave can catch an ID packet on the right frequency from the master. These possibilities can be classified into 3 subcases, as illustrated in Fig. 5 So the expected value of X in this case can be approximated by

$$X_2 = \frac{1}{32} \times (3 \times T_{inquiryscan}) + \frac{15}{32} \times T_{inquiryscan}. \tag{3}$$

Note that there is no delay for case 2.1 in Fig. 5. For cases 2.2 and 2.3, the delays are three and one inquiry scan interval, respectively.

The next two cases are similar to the above two cases. So we omit the explanations.

Case 3: (W_{s1} in the third $\frac{1}{8}$ window of W_{m1})

$$X_3 = \frac{1}{32} \times T_{inquiryscan} + \frac{1}{32} \times (4 \times T_{inquiryscan}) + \frac{14}{32} \times (2 \times T_{inquiryscan}). \tag{4}$$

Case 4: (W_{s1} in the fourth $\frac{1}{8}$ window of W_{m1})

$$X_4 = \frac{1}{32} \times (3 \times T_{inquiryscan}) + \frac{15}{32} \times T_{inquiryscan}. \tag{5}$$

Case 5: (W_{s1} in the fifth $\frac{1}{8}$ window of W_{m1}) The 32 frequency-matching possibilities of case 5 can be classified into four subcases, as shown in Fig. 6. All subcases are similar to earlier discussions, except subcase 5.3, where the frequency-matching will occur in next inquiry window W_{m2}. The slave thus has to wait $\lceil \frac{T_{inquiry} - T_{inquiryscan} \times 4}{T_{inquiryscan}} \rceil \times T_{inquiryscan}$ for window W_{m2} to appear. In the following analysis, we assume that $T_{inquiry}$ is a multiple of

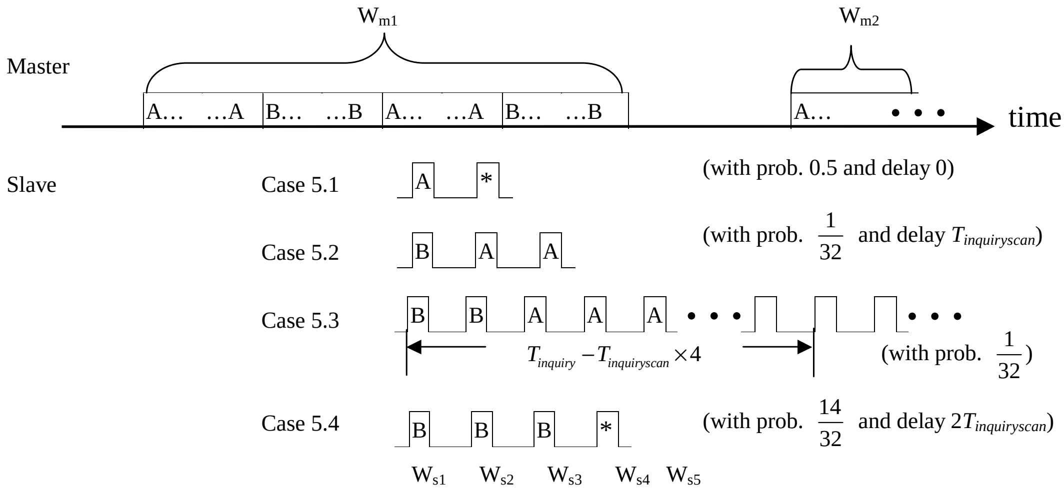


Figure 6. Illustration of Eq. (6), which contains four subcases of case 5. The frequency-matching of case 5.3 will occur in window W_{m2}

$T_{inquiryscan}$ for simplicity. So the waiting time is simplified to be ($T_{inquiry} - T_{inquiryscan} \times 4$). After the waiting, it will take X_1 time more for frequency-matching. So the expected value of X in this case can be approximated by

$$X_5 = \frac{1}{32} \times T_{inquiryscan} + \frac{1}{32} \times (T_{inquiry} - 4 \times T_{inquiryscan} + X_1) + \frac{14}{32} \times (2 \times T_{inquiryscan}). \tag{6}$$

The next three cases are similar to case 5. So we omit the explanations.

Case 6: (W_{s1} in the sixth $\frac{1}{8}$ window of W_{m1})

$$X_6 = \frac{1}{32} \times (T_{inquiry} - 5 \times T_{inquiryscan} + X_1) + \frac{15}{32} \times T_{inquiryscan}. \tag{7}$$

Case 7: (W_{s1} in the seventh $\frac{1}{8}$ window of W_{m1})

$$X_7 = \frac{1}{32} \times T_{inquiryscan} + \frac{15}{32} \times (T_{inquiry} - 6 \times T_{inquiryscan} + X_1). \tag{8}$$

Case 8: (W_{s1} in the eighth $\frac{1}{8}$ window of W_{m1})

$$X_8 = \frac{1}{2}(T_{inquiry} - 7 \times T_{inquiryscan} + X_1). \tag{9}$$

We can now get the expected value of X as follows:

$$X = \frac{1}{8}\sum_{i=1}^{8} X_i. \tag{10}$$

Next, we derive the value of Y. It is not hard to see that the calculation is similar to the case 1 of X. Therefore, the expected value of Y is

$$Y = \frac{1}{32} \times T_{inquiryscan} + \frac{1}{32} \times (4 \times T_{inquiryscan}) + \frac{14}{32} \times (2 \times T_{inquiryscan}). \tag{11}$$

Below, we analyze the frequency-matching delay for the interlaced inquiry scan which is proposed in Bluetooth V1.2. Bluetooth V1.2 tries to interlace the inquiry scan hopping sequence of V1.1. Specifically, let $f_0, f_1, ..., f_{31}$

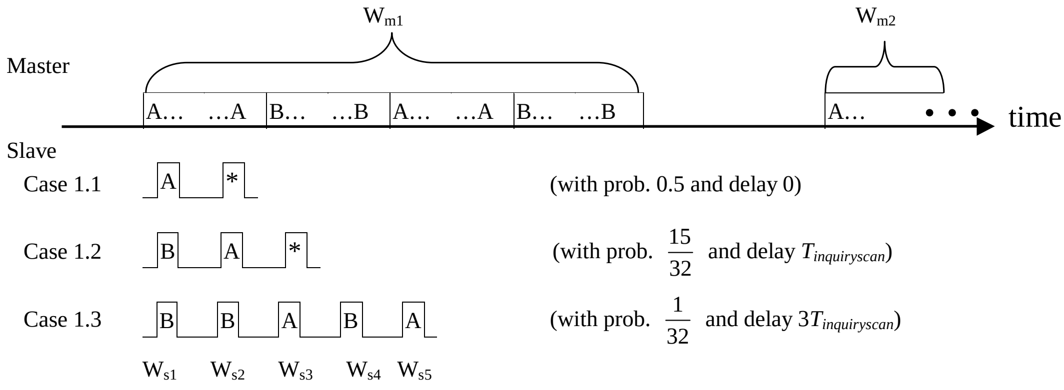


Figure 7. Three subcases of case 1 for Bluetooth V1.2.

be the hopping sequence in V1.1. Then the hopping sequence in V1.2 will replace f_i for each odd *i* by $f'_i = f_{i+16 (mod\ 32)}$. Therefore, the Eq. (1) can still be applied to Bluetooth V1.2. We only need to recalculate the values of X and Y.

There are also 8 cases for analyzing X, as discussed below.

Case 1: (W_{s1} in the first $\frac{1}{8}$ window of W_{m1}) The analysis is similar to the case 1 of X in Bluetooth V1.1. There are also 32 possibilities where the slave can catch an ID packet on the right frequency from the master. These possibilities can be classified into 4 subcases, as illustrated in Fig. 7. Note that in Fig. 7, a frequency in $f_0, f_1, ..., f_{15}$ is denoted by an "A", and a frequency in $f_{16}, f_{17}, ..., f_{31}$ is denoted by a "B". Also note that case 3.1 happens when frequencies $f_{31}, f_{16}, f_1, f_{18}$ and f_3 appear in windows $W_{s1}, W_{s2}, W_{s3}, W_{s4}$ and W_{s5}, respectively. So the expected value of X in this case can be approximated by

$$X_1 = \frac{15}{32} \times T_{inquiryscan} + \frac{1}{32} \times (3 \times T_{inquiryscan}). \tag{12}$$

Case 2: (W_{s1} in the second $\frac{1}{8}$ window of W_{m1}) This case is shown in Fig. 8.

$$X_2 = \frac{1}{32} \times T_{inquiryscan} + \frac{14}{32} \times (2 \times T_{inquiryscan}) + \frac{1}{32} \times (4 \times T_{inquiryscan}). \tag{13}$$

Case 3: (W_{s1} in the third $\frac{1}{8}$ window of W_{m1}) This case is similar to case 1.

$$X_3 = \frac{15}{32} \times T_{inquiryscan} + \frac{1}{32} \times (3 \times T_{inquiryscan}). \tag{14}$$

Case 4: (W_{s1} in the fourth $\frac{1}{8}$ window of W_{m1}) This case is similar to case 2.

$$X_4 = \frac{1}{32} \times T_{inquiryscan} + \frac{14}{32} \times (2 \times T_{inquiryscan}) + \frac{1}{32} \times (4 \times T_{inquiryscan}). \tag{15}$$

Case 5: (W_{s1} in the fifth $\frac{1}{8}$ window of W_{m1}) This case is similar to case 1.

$$X_5 = \frac{15}{32} \times T_{inquiryscan} + \frac{1}{32} \times (3 \times T_{inquiryscan}). \tag{16}$$

Case 6: (W_{s1} in the sixth $\frac{1}{8}$ window of W_{m1}) This case is similar to case 5 in V1.1. The slave may need to wait ($T_{inquiry} - T_{inquiryscan} \times 5$) for window W_{m2} to appear.

$$X_6 = \frac{1}{32} \times T_{inquiryscan} + \frac{14}{32} \times (2 \times T_{inquiryscan}) + \frac{1}{32} \times (T_{inquiry} - 5 \times T_{inquiryscan} + X_1). \tag{17}$$

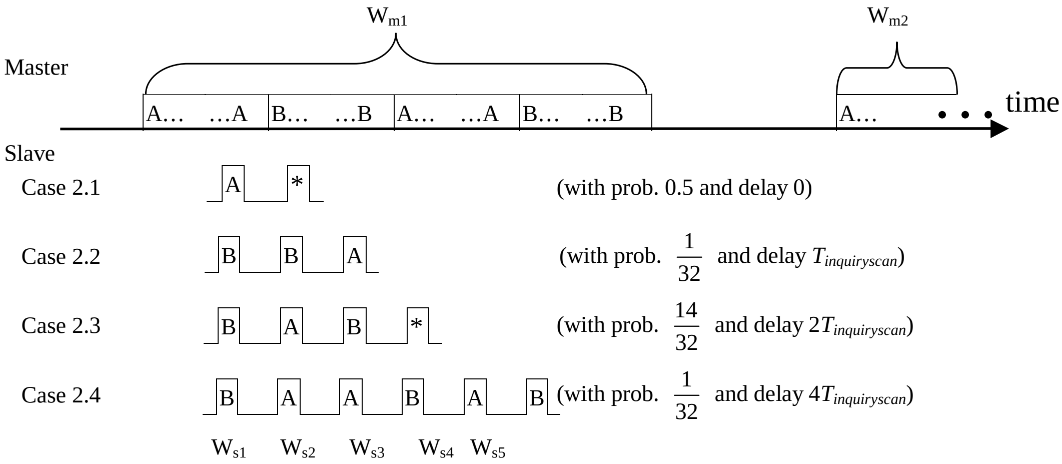


Figure 8. Four subcases of case 2 for Bluetooth V1.2.

The next two cases are similar to the above cases. So we omit the details.

Case 7: (W_{s1} in the seventh $\frac{1}{8}$ window of W_{m1})

$$X_7 = \frac{15}{32} \times T_{inquiryscan} + \frac{1}{32} \times (T_{inquiry} - 6 \times T_{inquiryscan} + X_1). \tag{18}$$

Case 8: (W_{s1} in the eighth $\frac{1}{8}$ window of W_{m1})

$$X_8 = \frac{1}{2} \times (T_{inquiry} - 7 \times T_{inquiryscan} + X_1). \tag{19}$$

We can now get the expected value of X as follows:

$$X = \frac{1}{8} \sum_{i=1}^{8} X_i. \tag{20}$$

Next, we want to calculate Y. It is not hard to see that the calculation is similar to the case 1 of X. Therefore, the expected value of Y is

$$Y = \frac{15}{32} \times T_{inquiryscan} + \frac{1}{32} \times (3 \times T_{inquiryscan}). \tag{21}$$

If we set $T_{inquiry} = 60$ and $T_{w\_inquiry} = 10.24$ seconds according to Table 1, we get the frequency-matching time $D = 23.55$ and 22.53 for Bluetooth V1.1 and V1.2, respectively. If we look in further details, we find that $X = 7.58$ and 4.47 and $Y = 1.32$ and 0.72 for V1.1 and V1.2, respectively. The interlaced inquiry scan indeed speeds up the frequency-matching but overall the improvement does not seem to be significant. The reason is because the value of $T_{inquiry}$ is too large. Thus, in Section 4 we propose some methods to speed up the bluetooth device discovery.

4 Speedup Schemes for Bluetooth Device Discovery

In this section, we propose three methods for speeding up the Bluetooth device discovery.

4.1 Half Inquiry Interval (HII)

From the analysis in Section 3, especially in Eq. (1), we note that the frequency-matching time is dominated by $T_{inquiry}$. Thus, we recommend that $T_{inquiry}$ be halved. In order to keep the same ratio of inquiry time, we

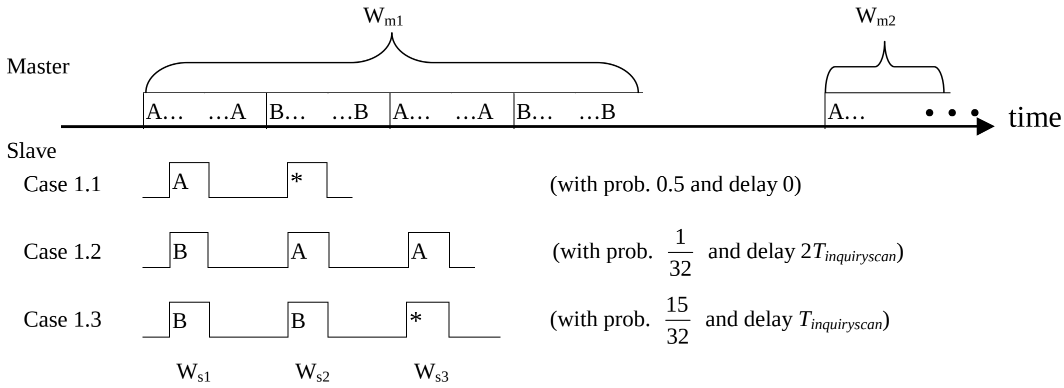


Figure 9. The case 1 of the HII method.

also recommend that $T_{w\_inquiry}$ be halved. As a result of this, the slave has only one chance to match with the frequencies on which the master sends ID packets during a sequence of 256 A/B trains. Note that here we do not use the interlacing technique in V1.2.

Below, we analyze the new frequency-matching time due to these changes. Eq. (1) is still applicable. However, there are only four cases of X, as discussed below.

Case 1: (W_{s1} in the first $\frac{1}{4}$ window of W_{m1}) There are 32 possibilities, which can be classified into 3 subcases as illustrated in Fig. 9. Note that there is only one chance for frequency matching during a sequence of 256 A/B trains. The delay is:

$$X_1 = \frac{1}{32} \times (2 \times T_{inquiryscan}) + \frac{15}{32} \times T_{inquiryscan}. \tag{22}$$

Case 2: (W_{s1} in the second $\frac{1}{4}$ window of W_{m1}) This case is similar to case 1.

$$X_2 = \frac{1}{32} \times (2 \times T_{inquiryscan}) + \frac{15}{32} \times T_{inquiryscan}. \tag{23}$$

Case 3: (W_{s1} in the third $\frac{1}{4}$ window of W_{m1}) The 32 frequency-matching possibilities can be classified into three subcases, as shown in Fig. 10. All subcases are similar to earlier discussions, except subcase 3.2, where the frequency-matching will occur in next inquiry window W_{m2}. Recall that we assume that $T_{inquiry}$ is a multiple of $T_{inquiryscan}$, so the waiting time is ($T_{inquiry} - T_{inquiryscan} \times 2$). The expected value of X in this case can be approximated by

$$X_3 = \frac{1}{32} \times (T_{inquiry} - 2 \times T_{inquiryscan} + X_1) + \frac{15}{32} \times T_{inquiryscan}. \tag{24}$$

Case 4: (W_{s1} in the fourth $\frac{1}{4}$ window of W_{m1}) This case is similar to case 3.

$$X_4 = \frac{1}{2} \times (T_{inquiry} - 3 \times T_{inquiryscan} + X_1). \tag{25}$$

We can now get the expected value of X as follows:

$$X = \frac{1}{4} \sum_{i=1}^{4} X_i. \tag{26}$$

The calculation of Y is similar to the case 1 of X. The expected value of Y is

$$Y = \frac{1}{32} \times (2 \times T_{inquiryscan}) + \frac{15}{32} \times T_{inquiryscan}. \tag{27}$$

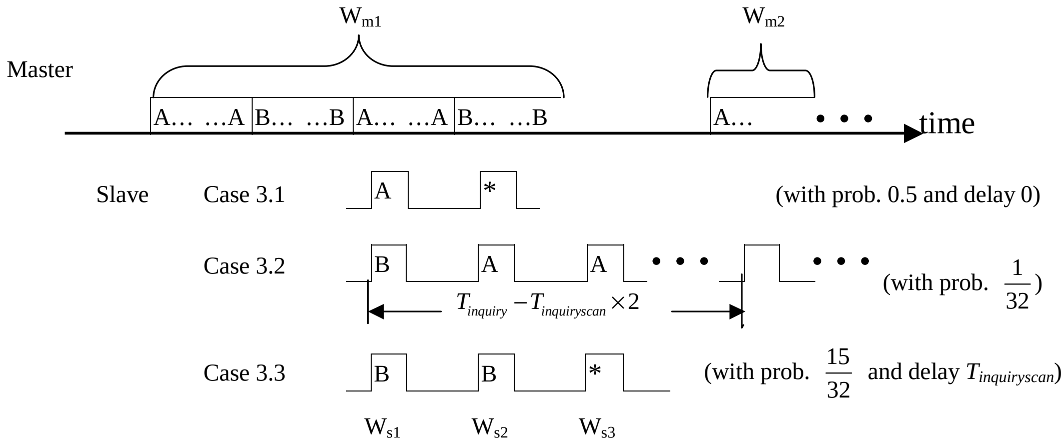


Figure 10. The case 3 of the HII method.

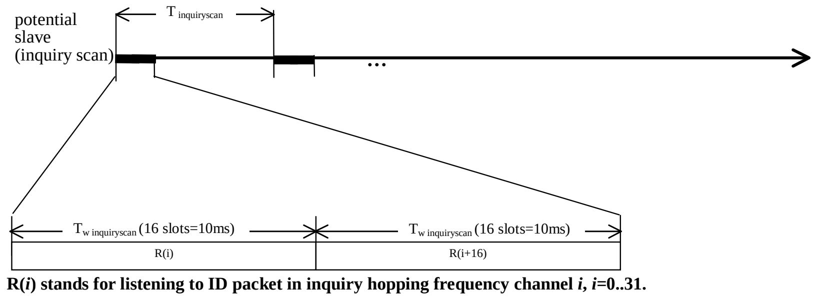


Figure 11. The proposed DIS scheme.

For example, if we set $T_{inquiry} = 30$ and $T_{w\_inquiry} = 5.12$ seconds, then we get the frequency-matching time D = 12.11 seconds. The reduction is significant. In this case, X is 4.06 seconds and Y is 0.68 seconds. So the reduction is mainly contributed by the reduction of $T_{inquiry}$.

4.2 Dual Inquiry Scan (DIS)

In this scheme, we hope that once an inquiry scan window of a slave encounters an inquiry window of a master, a frequency matching will occur as long as there is sufficient overlapping between these two windows. Toward this goal, the *Dual Inquiry Scan* (*DIS*) scheme requires the slave to perform inquiry scan on dual frequencies, one in A train and the other in B train. To be more precise, for every $T_{inquiryscan}$ period, the slave should perform inquiry scan on two frequencies, f_i and f_{i+16}, each for a duration of $T_{w\_inquiryscan}$ (refer to Fig. 11). Note that the value of i is increased by 1 (with modulo 32) after each inquiry scan window. As a result, frequency-matching will occur on either f_i or f_{i+16} with a high probability. In order to keep the same ratio of inquiry scan time, we recommend that $T_{inquiryscan}$ be doubled.

Below, we analyze the frequency-matching delay for the *DIS* scheme. Eq. (1) can also be applied to the analysis except that Y is replaced by X. That is, we have

$$D = \frac{T_{w\_inquiry}}{T_{inquiry}} \times X + \frac{T_{inquiry} - T_{w\_inquiry}}{T_{inquiry}} \times \left(\frac{T_{inquiry} - T_{w\_inquiry}}{2} + \frac{T_{inquiryscan}}{2} + X\right), \quad (28)$$

where X is the expected delay after the slave starts an inquiry scan window during an inquiry window. When the master is sending an A/B trains which is sufficiently covered by the slave's inquiry scan window, frequency-matching will occur with no delay with a probability of about $\frac{1}{2}$ and with $T_{train}(=0.01)$ delay with a probability of about $\frac{1}{2}$. Thus, we have $X \approx 0.005$, which gives $D \approx 21.71$ seconds.

4.3 Combination of HII and DIS

If we combine the above two strategies by adopting *HII* for the master and adopting *DIS* for the slave, then further reduction of D can be obtained. The analysis is similar and can be obtained from Eq. (28). By setting $T_{inquiry} = 30$ and $T_{w\_inquiry} = 5.12$ seconds, D can be reduced to be 11.38 seconds.

5 Conclusions

In this paper, we have analyzed the frequency-matching time of Bluetooth V1.1 and V1.2. The main component of delay in its long device discovery is the long waiting time for the appearance of inquiry windows from the master. The proposed *HII* scheme can reduce the aforementioned waiting time. The *DIS* scheme can further reduce the frequency-matching delay by scanning two frequencies back to back. If we combine these two schemes, the expected frequency-matching delay can be reduced from 23.55 seconds to 11.38 seconds. The ratio of time for performing inquiry and inquiry scan does not increased.

6 Acknowledgement

Y. C. Tseng's research is co-sponsored by the NSC Program for Promoting Academic Excellence of Universities under grant number 93-2752-E-007-001-PAE, by Computer and Communications Research Labs., ITRI, Taiwan, by Intel Inc., by the Institute for Information Industry and MOEA, R.O.C, under the Handheld Device Embedded System Software Technology Development Project and the Communications Software Technology Project, and by Chung-Shan Institute of Science and Technology under contract number BC93B12P.

References

[1] S. Basagni, R. Bruno, and C. Petrioli. "Device Discovery in Bluetooth Networks: A Scatternet Perspective," *Proc. of the Second IFIP-TC6 Networking Conference,Networking 2002*, Pisa, Italy, May 2002, pp. 1087-1092.

[2] Bluetooth Special Interest Group. Bluetooth specification version 1.1 and 1.2. *http://www.bluetooth.com*, 2001.

[3] A. Busboom, I. Herwono, M. Schuba, and G. Zavagli. "Unambiguous Device Identification and Fast Connection Setup in Bluetooth," *Proc. of the European Wireless 2002*, vol. 0, Florence, Italy, Feb 2002.

[4] K. Cheolgi, M. Joongsoo, L. Joonwon. "A Random Inquiry Procedure using Bluetooth," *Proc. of International Conference on Communication in Computing (CIC)*, Las Vegas, USA, Jun 2001.

[5] I. Maric. "Connection Establishment in the Bluetooth System," Masters Thesis, the State University of New Jersey, 2000.

[6] P. Murphy, E. Welsh, and J. P. Frantz. ”Using Bluetooth for Short-Term Ad-Hoc Connections Between Moving Vehicles: A Fesability Study,” *IEEE Vehicular Technology Conference*, vol. 1, no. 55, Birmingham, AL, May 2002, pp. 414-418.

[7] T. Salonidis, P. Bhagwat, and L. Tassiulas. ”Proximity awareness and fast connection establishment in Bluetooth,” *Proc. of Mobile and Ad Hoc Networking and Computing, 2000 (MobiHOC’00)*, Boston, Massachusetts, Aug. 2000, pp. 141-142.

[8] F. Siegemund and M. Rohs. ”Rendezvous Layer Protocols for Bluetooth-Enabled Smart Devices,” *Proc. 1st International Conference on Architecture of Computing Systems*, vol. 2299, Karlsruhe, Germany, pp. 256-273, Apr. 2002.

[9] E. Welsh, P. Murphy, and J. P. Frantz. Improving Connection Times for Bluetooth Devices in Mobile Environments. *Proc. of International Conference on Fundamentals of Electronics, Communications and Computer Sciences (ICFS)*, Mar. 2002.

[10] R. Woodings, D. Joos, T. Clifton, and C. D. Knutson. ”Rapid Heterogeneous Connection Establishment: Accelerating Bluetooth Inquiry Using IrDA,” *Proc. of the Third Annual IEEE Wireless Communications and Networking Conference (WCNC) 2002*, vol. 1, Orlando, Florida, Mar. 2002, pp. 342-349.

Biographies

Jehn-Ruey Jiang received his Ph. D. degree in Computer Science in 1995 from National Tsing-Hua University, Taiwan. He joined Chung-Yuan Christian University and Hsuan-Chuang University as an Associate Professor in 1995 and 1998, respectively. He is currently with the Department of Computer Science and Information Engineering, National Central University. He is a recipient of the Best Paper Award in Int’l Conf. on Parallel Processing, 2003. His research interests include distributed computing, mobile computing, peer-to-peer computing, distributed fault-tolerance, protocols for mobile ad hoc networks and wireless sensor networks.

Bing-Rong Lin received his B.S. degree in Computer Science from the National Chiao-Tung University, Taiwan, in 2002. His research interests include wireless network, sensor network and Bluetooth.

Yu-Chee Tseng received his B.S. and M.S. degrees in Computer Science from the National Taiwan University and the National Tsing-Hua University in 1985 and 1987, respectively. He worked for the D-LINK Inc. as an engineer in 1990. He obtained his Ph.D. in Computer and Information Science from the Ohio State University in January of 1994. He was an Associate Professor at the Chung-Hua University (1994 1996) and at the National Central University (1996 1999), and a Full Professor at the National Central University (1999 2000). Since 2000, he has been a Full Professor at the Department of Computer Science and Information Engineering, National Chiao-Tung University, Taiwan.

He is a two-time recipient of the Outstanding Research Award, National Science Council, ROC, in 2001-2002 and 2003-2005, and a recipient of the Best Paper Award in Int’l Conf. on Parallel Processing, 2003. Several of his papers have been chosen as Selected/Distinguished Papers in international conferences. He has guided students to participate in several national programming contests and received several awards. His research interests include mobile computing, wireless communication, network security, and parallel and distributed computing. Dr. Tseng is a member of ACM and a Senior Member of IEEE.