A Mechanism for Quick Bluetooth Device Discovery*

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

§Department of Information Management
Hsuan-Chuang University, Taiwan

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

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 of Bluetooth, the major component of delay in its device discovery, for both versions V1.2 and V1.1 specifications. We then propose a new scheme called Dual Inquiry Scan to reduce the delay. The result is a reduction of average frequency matching from 22.98 seconds to 20.63 seconds in typical cases. If a master continuously sends ID packets, the delay can be further reduced from 1.075 seconds to 0.005 seconds.

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

1 Introduction

Bluetooth [2] is a promising technology to enable 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 multimedia 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 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 the 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 of Bluetooth, the major component of delay in its device discovery, for both versions V1.2 and V1.1 specifications. We show that the average delay is about 22.98 seconds. Even if we allow a master to continuously send ID packets, the delay is still as long as 1.075 seconds. This motivates us to propose a scheme, called *Dual Inquiry Scan*, which is compatible with the original Bluetooth specification and can shorten the average frequency-matching time to around 20.63 seconds. The delay can be further reduced to 0.005 seconds if the master is willing to send ID packets continuously.

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 dual inquiry scan scheme, including an analysis and a comparison to the original Bluetooth scheme. Concluding remarks are drawn in Section 5.

*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.

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

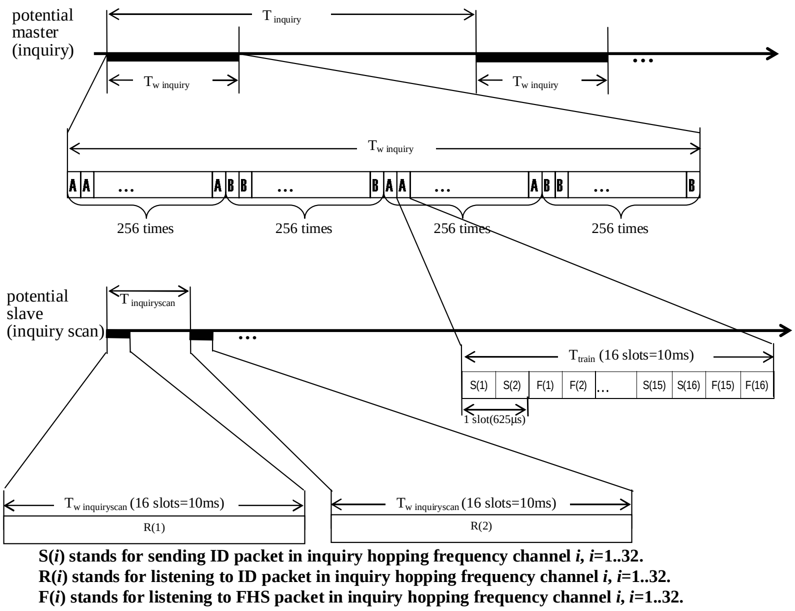

Figure 1. Bluetooth inquiry procedure.

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 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 two sets of 16 channels exclusively. In an $T_{w\_inquiry}$ interval, $N_{inquiry}$ A trains, $N_{inquiry}$ B trains, $N_{inquiry}$ A trains, and $N_{inquiry}$ B trains of ID packets will be 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. The Bluetooth specification suggests 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 the 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 $T_{w\_inquiryscan} \geq T_{train}$ 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.

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 modify the device discovery parameters [4, 6, 7, 9], some use auxiliary devices [3, 10], while some relies on device cooperation to assist the device 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

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

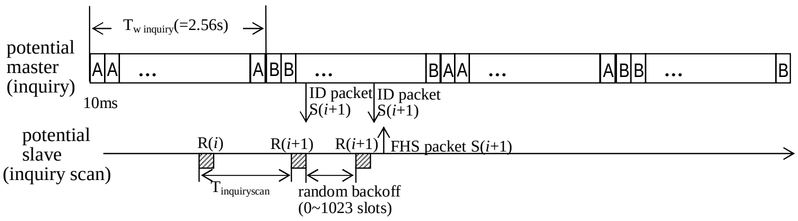

Figure 2. The backoff procedure for a slave to reply a FHS packet.

cooperative device discovery scheme is proposed to allow devices to exchange their knowledge of nearby devices, such as DB address 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. For example, an interlaced inquiry scan hopping sequence may be f_1, f_{18}, f_3, f_{20},..., where f_1 and f_3 belong to A train and f_{18} and f_{20} belong to B train.

3 Analyses for Bluetooth Device Discovery

In this section, we analyze the frequency matching delay of Bluetooth, the major component of delay in its device discovery, for both versions V1.2 and V1.1 specifications. We start with the analysis for Bluetooth V1.1. Note that we say that a device is working in A (or B) train if it is sending or receiving ID packets at a frequency in A (or B) train.

Suppose there is already a device turned on to be a master performing the scan procedure. According to whether or not the master is sending ID packets, we divide time axis into *inquiry window* and *non-inquiry window*. We also suppose there is a device turned on at a later time to be a slave. The frequency-matching delay, denoted by D, can be measured by the elapsing time starting from the time the slave starts inquiry scan to the time 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 the inquiry window with probability $\frac{T_{w\_inquiry}}{T_{inquiry}}$, or in the non-inquiry window with probability $\frac{T_{inquiry}-T_{w\_inquiry}}{T_{inquiry}}$. We have

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

where X (resp., Y) is the expected delay for the slave starting

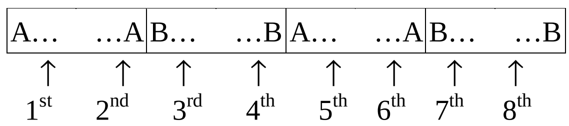

Figure 3. Eight possible cases for the slave to start its inquiry scan.

inquiry scan in the inquiry window (resp., within the first $T_{inquiryscan}$ of the inquiry window). The reader can check that when the slave starts its inquiry scan in the non-inquiry window, it is expected to wait $\frac{T_{inquiry}-T_{w\_inquiry}}{2}$ time to be in the next inquiry window.

As shown in Table 1, $T_{inquiry}$ and $T_{w\_inquiry}$ are suggested to be 60 and 10.24 seconds[3], respectively. The parameter $T_{w\_inquiry}$ is usually fixed. However, $T_{inquiry}$ may be as small as 10.24, which corresponds to the case that the master is always sending ID packets. For such a case, D has the minimal value.

Below, we perform the analyses for X and Y. We also adopt the parameter settings of Table 1. Specifically, we set $T_{inquiryscan}$ to be $\frac{N_{inquiry} \times T_{train}}{2}$, where $N_{inquiry} = 256$ and T_{train}=0.01. We have $T_{inquiryscan} = 128 \times T_{train} = 1.28$ and there are two inquiry scans during the period of 256 consecutive A (or B) trains of ID packets sent. Thus, there are 8 possible cases for the slave to start its inquiry scan in the inquiry window (refer to Fig. 3). Note that for simplicity, we omit the subtle cases that the slave's inquiry scan covers the duration that the master working in A (or B) train changes to work in B (or A) train.

For the 1^{st} case, the expected delay for the slave to receive master's ID packet is approximately

$$\frac{1}{2}(u \times 1.28 + v \times 5.12 + w \times 2.56) \quad (2)$$

, where $u = \frac{1}{16}, v = \frac{1}{16}$, and $w = 1 - u - v = \frac{14}{16}$.

[3] We will omit the word "seconds" in the following analysis.

A… …A	B… …B	A… …A	B… …B

	↑	↑	↑	↑	↑	
Possibility 1:	B	A				(with prob. *u*)
Possibility 2:	B	B	A	A	A	(with prob. *v*)
Possibility 3:	B	B	B			(with prob. *w*)

Figure 4. Three situations for Eq. (2).

In Eq. (2), u is the probability that under the condition of the slave working in B train, the slave works in A train for the next inquiry scan. For such a possibility, the frequency-matching will occur 1.28s later (refer to Fig. 4). As to v in Eq. (2), it is the probability that the salve performs inquiry scan at the frequencies of B, B, A, A, A,... trains. For such a possibility, the frequency-matching will occur 5.12s later (also refer to Fig. 4). If neither the first nor the second possibility takes place, frequency-matching will occur 2.56s later.

For the 2^{nd} case, the expected delay is approximately

$$\frac{1}{2}(p \times 3.84 + q \times 1.28). \tag{3}$$

, where $p = \frac{1}{16}$ is the probability that under the condition of the slave working in B train, the slave works in A train for the next inquiry scan, and $q = 1 - p = \frac{15}{16}$.

In Eq. (3), the term $\frac{1}{2}$ is for the probability that the slave works in B train. For such a case, if the slave works in A train for the next inquiry scan, the frequency-matching will occur 1.28s later; otherwise, the frequency-matching will occur 3.84s later.

The reader can check that Eq. (2) can also be applied to the 3^{rd} case, and Eq. (3), the 4^{th} case. Below, we analyze the 5^{th} case.

For the 5^{th} case, the expected delay for the slave to receive master's ID packet is approximately

$$\frac{1}{2}(u \times 1.28 + v \times \lceil \frac{T_{inquiry} - 1.28 \times 4}{1.28} \rceil \times 1.28 + w \times 2.56) \tag{4}$$

, where $u = \frac{1}{16}, v = \frac{1}{16}$, and $w = 1 - u - v = \frac{14}{16}$. The situation of the 5^{th} case is similar to that of the 1^{st} case. However, frequency-matching will occur in next inquiry window for the third possibility with probability v. The slave thus has to wait $\lceil \frac{T_{inquiry} - 1.28 \times 4}{1.28} \rceil \times 1.28$ for the next inquiry window. Note that we omit the insignificant delay for the master and the slave to work in the same train in the next inquiry window. The omitted delay is usually less than $T_{inquiryscan}$ (1.28s) and is seldom of the maximum value about $5 \times T_{inquiryscan}$ (6.4s). With the same senses, we analyze the 6^{th}, the 7^{th}, and the 8^{th} cases as follows:

For the 6^{th} case, the expected delay is approximately

$$\frac{1}{2}(p \times \lceil \frac{T_{inquiry} - 1.28 \times 5}{1.28} \rceil \times 1.28 + q \times 1.28) \tag{5}$$

For the 7^{th} case, the expected delay is approximately

$$\frac{1}{2}(p \times 1.28 + q \times \lceil \frac{T_{inquiry} - 1.28 \times 6}{1.28} \rceil \times 1.28) \tag{6}$$

For the 8^{th} case, the expected delay is approximately

$$\frac{1}{2}(\lceil \frac{T_{inquiry} - 1.28 \times 7}{1.28} \rceil \times 1.28) \tag{7}$$

A… …A	B… …B	A… …A	B… …B

	↑	↑	↑	↑	
Possibility 1:	B	B			(with prob. *u*)
Possibility 2:	B	A	A	B	(with prob. *v*)
Possibility 3:	B	A	B		(with prob. *w*)

Figure 5. Three situations for Eq. (9).

When we sum up the products of probabilities and expected delays for all the eight cases, we get the value of X, which is 7.43 for $T_{inquiry}$=60. As to the value of Y, it can be calculated by Eq. (2), which is 1.32. The time delay D of frequency-matching is thus 22.98. The reader can check that D is 1.075 when we take $T_{inquiry}$ as 10.24.

Below, we analyze the frequency-matching delay for Bluetooth specification V1.2. The Eq. (1) can also be applied to Bluetooth V1.2. Thus, we just need to figure out the values of X and Y as follows.

There are 8 cases for the slave to start its inquiry scan. For the 1^{st}, the 3^{rd} and the 5^{th} cases, the expected delay for the slave to receive master's ID packet is approximately

$$\frac{1}{2}(p \times 1.28 + q \times 3.84) \tag{8}$$

, where $p = \frac{15}{16}$ is the probability that under the condition of the slave working in B train, the slave works in A train for the next inquiry scan, and $q = 1 - p = \frac{1}{16}$.

For the 2^{nd}, 4^{th} and the 6^{th} cases, the expected delay is approximately

$$\frac{1}{2}(u \times 1.28 + v \times 3.84 + w \times 2.56) \tag{9}$$

, where $u = \frac{1}{16}$ stands for the probability of the occurrence of possibility 1 in Fig. 5; $v = \frac{1}{16}$, the possibility 2, and $w = 1 - u - v = \frac{14}{16}$, the possibility 3.

For the 7^{th} case, the expected delay is approximately

$$\frac{1}{2}(p \times 1.28 + q \times \lceil \frac{T_{inquiry} - 1.28 \times 6}{1.28} \rceil \times 1.28) \tag{10}$$

For the 8^{th} case, the expected delay is approximately

$$\frac{1}{2}(\lceil \frac{T_{inquiry} - 1.28 \times 7}{1.28} \rceil \times 1.28) \tag{11}$$

Likewise, we also omit the delay for the master and the slave to work in the same train in the next inquiry window for the 7^{th} and 8^{th} cases. When we sum up the products of probabilities and expected delays for all the eight cases, we get the value of X, which is 4.23 for $T_{inquiry}$=60. As to the value of Y, it can be calculated by Eq. (8), which is 0.72. The time delay D of frequency-matching is thus 21.95 for $T_{inquiry} = 60$. The reader can check that D is 0.919 when we take $T_{inquiry}$ to be 10.24.

4 Speeding Device Discovery for Bluetooth

In this section, we propose a scheme, called *Dual Inquiry Scan*, for speeding up the Bluetooth device discovery. The

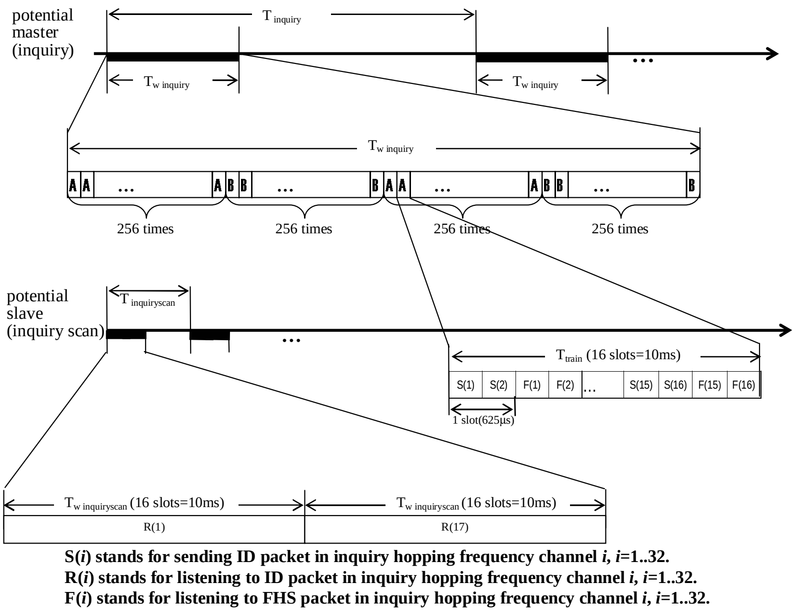

Figure 6. The proposed dual inquiry scan scheme.

proposed scheme requires a potential slave to perform inquiry scan by receiving ID packets on dual hopping frequencies, one in A train and the other in B train. To be more precise, for every $T_{inquiryscan}$ period, a potential slave should perform inquiry scan on two channels, f_i and f_{i+16}, each for a duration of $T_{w\_inquiryscan}$ (refer to Fig. 6). If the slave is receiving while the master is sending ID packets, frequency-matching occurs on either f_i or f_{i+16}. Thus, the frequency-matching delay time is shorten significantly. The proposed scheme also has the merit that it is compatible with the original Bluetooth specification.

Below, we analyze the proposed scheme and compare it with original Bluetooth device discovery. The time delay of frequency-matching is the same as Eq. (1) except that Y is replaced by X. That is, we have

$$D = \frac{T_{w\_inquiry}}{T_{inquiry}} \times X + \frac{T_{inquiry} - T_{w\_inquiry}}{T_{inquiry}} \times \left(\frac{T_{inquiry} - T_{w\_inquiry}}{2} + X\right), \quad (12)$$

where X is the expected delay for the slave starting inquiry scan at any instance of the inquiry window. When the master is sending and the slave is receiving ID packets simultaneously, frequency-matching occurs with no delay of probability about $\frac{1}{2}$ and with $T_{train}(= 0.01)$ delay of probability about $\frac{1}{2}$. Thus, we have $X \approx 0.005$. We can now calculate D to have $D = 20.63$ for $T_{inquiry} = 60$, and D=0.005 for $T_{inquiry} = 10.24$. The proposed scheme is compared with original Bluetooth specifications in terms of frequency-matching delay in Table 2.

5 Conclusion

In this paper, we have analyzed the frequency matching of Bluetooth, the major component of delay in its long device discovery, for both versions V1.2 and V1.1 specifications. We have also proposed a scheme, called *Dual Inquiry Scan*, to shorten the frequency-matching delay. As we have shown, the expected frequency-matching delay is reduced from 22.98 seconds to 20.63 seconds. When the potential master continuously sends ID packets, the delay is further reduced from 1.075 seconds to 0.005 seconds. In the future, we plan to derive the frequency-matching delay by simulations and/or experiments for both of the original Bluetooth and the proposed scheme.

References

[1] S. Basagni, R. Bruno, and C. Petrioli. Device Discovery in Bluetooth Networks. *Proc. IEEE Networking 2002*, 2002.

[2] Bluetooth Special Interest Group. Bluetooth specification version 1.1. *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*, 2002.

[4] J. Ma and J. Lee. A Random Inquiry Procedure using Bluetooth. *Proc. of International Conference on Communication in Computing (CIC)*, 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*, 2002.

Table 2. Comparison of frequency-matching delays for the Dual Inquiry Scan Scheme and Bluetooth Specifications V1.1 and V1.2.

The Scheme	Time Delay of Frequency-matching ($T_{inquiry} = 60$s)	Time Delay of Frequency-matching ($T_{inquiry} = 10.24$s)
Bluetooth V1.1	22.98s	1.075s
Bluetooth V1.2	21.95s	0.919s
Dual Inquiry Scan	20.63s	0.005s

[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)*, 2000.

[8] F. Siegemund and M. Rohs. Rendezvous Layer Protocols for Bluetooth-Enabled Smart Devices. *Proc. 1st International Conference on Architecture of Computing Systems*, pages 256–273, 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, Communcations and Computer Sciences (ICFS)*, 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*, 2002.