Improved Time to Connect for Bluetooth

LENNART ISAKSSON, MARKUS FIEDLER
Department of Telecommunication Systems
Blekinge Institute of Technology
Campus Gräsvik, Karlskrona
Sweden

Abstract: - Bluetooth has from its beginning used Frequency Hopping (FH) from version 1.0A to the current version 1.2. The so-called selection box uses a complex scheme to calculate the next correct frequency in time. This makes transmissions between Bluetooth devices both robust and secure. Several attempts have been carried out to improve the time to connect, due to the relatively high time to connect given by the Bluetooth specification. Many have not fully understood the FH for Bluetooth and have therefore suggested sub-optimized solutions to improve the time to connect. In this paper, a presentation of the problem that causes the poor time to connect performance is given. This knowledge is needed for an improvement of the time to connect for Bluetooth. To accomplish a reduction of the time to connect by a factor of 100, three main parameters have to be changed: First, the A_{train} and B_{train} should be switched directly after each other. Second, the phase should be switched faster, which is based on the clock. Finally, a reduction of the random backoff parameter needs to be considered. These changes, together with other parameters, like time out values, have to be changed/reduced accordingly. Together with these changed parameters a simulation of the time to connect for Bluetooth is presented.

Key-Words: - Bluetooth, Frequency Hopping

1 Introduction

In June 10, 1941, a patent for Frequency Hopping (FH) using 88 frequencies called the "Secret Communications System" was requested by Hedy Kiesler Markey and George Antheil [1]. In August 11, 1942, the patent 2,292,387 was approved. However, it was not until the Cuba blockade in 1962 that the FH was used for the first time. Nowadays FH is used in cellular telephones and Personal Digital Assistants (PDAs), etc. The FH itself is very strong in secure communication as compared to other technologies like the Direct Sequence (DS) which uses only one dedicated channel for its transmission. Bluetooth uses a very large part of the 2.400 – 2.4835 GHz Industrial, Scientific, and Medical (ISM)-band. Bluetooth uses 79 frequencies between 2.402 GHz and 2.480 GHz and changes its frequency 1600 times per seconds after having established a connection between two devices.

From the very start, the idea of Bluetooth was to replace all wired connections between computers and other connected devices with wirelessness. The first versions of the Bluetooth devices 1.0A and 1.0B, could only do a point-to-point connection which is the smallest ad-hoc wireless network possible and called piconet. The next versions, 1.1 [2] and further [3], could later do a point-to-multipoint connection. Still, according to the Bluetooth specification an inquiry could take up to 10 s to create a connection and in some situations even longer. For different types of moving vehicle scenarios this is not fast enough. According to Nielsen [4] different time thresholds are perceived differently by the user and to measure these thresholds a framework has to be set up as suggested by Rajamony *et al.* [5].

In which types of scenario could Bluetooth be used? One possible scenario could be the use of the Bluetooth technique when driving through a toll system that collects a fee for driving on a specific road, lane or bridge. Another scenario could be the driving past a meteorological station mounted near the motor way using Bluetooth. In this manner important information regarding the weather forecast could be provided. A moving vehicle is used in both scenarios. How, then, would Bluetooth function in these scenarios? First one needs to calculate the time necessary for Bluetooth to work properly in these types of scenarios.

To calculate the time spent in range, depending on the power used by the device, a few parameters have to be known, see Figure 1. First, the speed [km/h] of the moving vehicle has to be known. Second, the range/power used by the device r [m], and finally the length x [m] between the two Bluetooth devices have to be found. These calculations are based on x = 2 m which is the length from the origin o from the moving object. Now, the time spent in the range/power could be calculated using the distance $2 \cdot y$ [m], which later could be expressed in seconds, see Figure 2. In this figure an illustration of the time needed to create a connection, to send over a message, and finally disconnect at a specific speed vs. range/power is presented. The first power level (20 dBm) has a range of $\sim$100 m and is suitable for different types of scenarios when using a maximum speed of 50 km/h. The second power level (4 dBm) has a range of $\sim$10 m and is suitable for moving activities under a few km/h.

Bluetooth's FH technique of today is not fast enough according to these findings. How, then, can this be im-

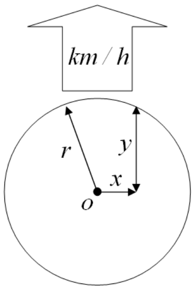

Fig. 1: A moving object which intersects with stationary object along the corda of the object.

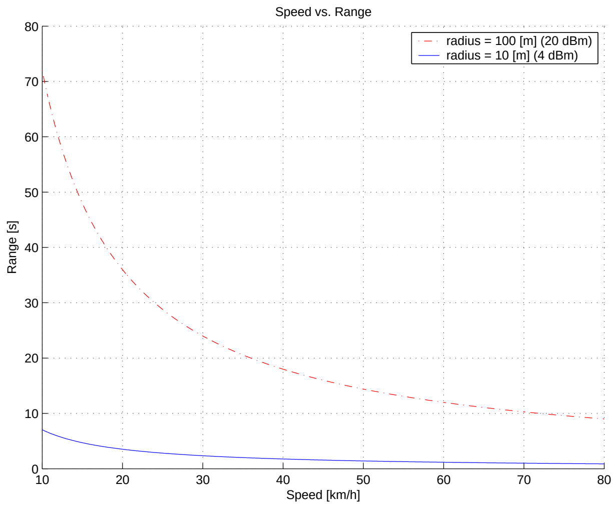

Fig. 2: A plot showing the speed vs. time spent in range of the moving object.

proved, and is it possible to reduce the time to connect by a factor of 100 which is necessary to fulfill the scenarios with highly moving vehicles? This paper shows that this is doable.

The paper is organized as follows: Section 2 describes related work and research on improving the time to connect for Bluetooth and on differently perceived time thresholds by the user. Section 3 describes the issue of the insufficient time to connect scheme used. Section 4 describes the improved time to connect. In Section 5, a description of the simulation results based on the new parameters used is presented. In Section 6, conclusions and future work are given.

2 Related Work

Záruba *et al.* [6] have investigated the inquiry procedure. They have identified five problems, two out of which are of particular interest for this paper. First, they identified that the random backoff (RB) spent after receiving an identity (ID) packet could be time consuming and that this could be reduced. Second, they studied the infrequent change of the Bluetooth train which reduces the probability when finding the same channel. According to simulations Záruba *et al.* are able to reduce the inquiry procedure. More simulation results are presented that use more than two devices simultaneously. However, Záruba *et al.* have not investigated how long the entire time to connect could take and they have not presented the parameters used to achieve these results. Their simulation model is, furthermore, not validated.

Welsh *et al.* [7] have looked at the time to connect for Bluetooth and how to reduce this time spent during a link set-up. They have done empirical measurements and simulations on a connection between two Bluetooth devices. Three possible changes are presented. First, a reduction of the random backoff parameter or elimination is suggested. Second, a single frequency train is proposed instead of two. Finally, a combination of both ideas is also proposed. The problem with the elimination of the random backoff sub state eliminates the idea of avoiding a collision in the air when several devices are responding to the same identity packet by simultaneous sending the Frequency Hop Synchronization (FHS) packet back to the inquiry device. A second problem is that all simulations done are based on a non-validated simulation model.

3 Time to Connect

In this section a basic description of the original connection scheme for Bluetooth is presented.

3.1 Selection Box

The selection box, which derives the next proper channel, depends on five input parameters. The first parameter is the state machine's substate. Each substate is represented by a unique algorithm which is denoted by X. The notation X_{a-b} refers to which bits a to b are produced by the algorithm X, see algorithms (1) and (2). The algorithm (1) is used for the inquiry substate and the algorithm (2) is used for the inquiry response and inquiry scan substate.

$$\mathrm{Xi}_{4-0}^{(79)} = [\mathrm{CLKN}_{16-12} + k_{\mathrm{offset}} + (\mathrm{CLKN}_{4-2,0} - \mathrm{CLKN}_{16-12}) \mod 16] \mod 32 \qquad (1)$$

$$\mathrm{Xir}_{4-0}^{(79)} = [\mathrm{CLKN}_{16-12} + \mathrm{N}] \mod 32 \qquad (2)$$

In the following, the input X determines the phase which is a set of channels/frequencies used based on the built-in clock. The variable $k_{\mathrm{offset}} \in \{8, 24\}$ is used to toggle between the $\mathrm{A}_{\mathrm{train}}$ and $\mathrm{B}_{\mathrm{train}}$. Each train is a sequence of frequencies used for scanning parts of the ISM-band for new units. These trains are groups of frequencies to be visited in a predefined order. Equation (1) produces a sequence of sub-channels to form the contents of the $\mathrm{A}_{\mathrm{train}}$ and $\mathrm{B}_{\mathrm{train}}$ depending on k_{offset}. Phase 0 in the $\mathrm{A}_{\mathrm{train}}$ includes the frequencies $f(k-8)\ldots f(k)\ldots f(k+7)$ and the phase 0 in $\mathrm{B}_{\mathrm{train}}$ includes the frequencies $f(k+8)\ldots f(k+15)$, $f(k-16)\ldots f(k-9)$, see Table 1. Corresponding entries for the $\mathrm{B}_{\mathrm{train}}$ are obtained by adding 16 and taking

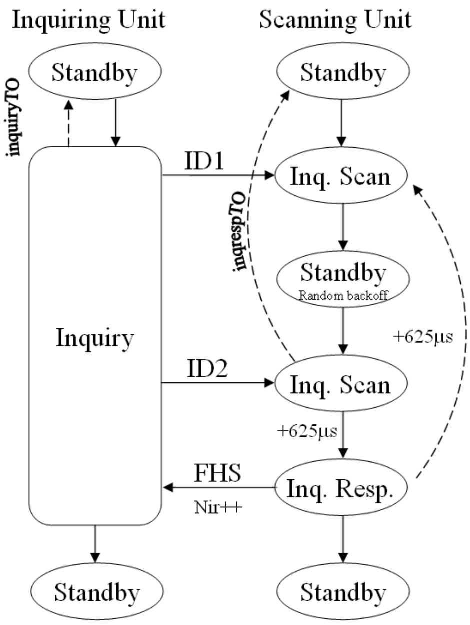

Fig. 3: Inquiry procedure.

the result modulo 32. For the A_{train}, phase 0 consists of the following sequence of numbers: 24, 25, 24, 25, 26, 27, 26, 27 $\cdots$ 6, 7, 6, 7. For the B_{train}, phase 0 consists of the following sequence of numbers: 8, 9, 8, 9, 10, 11, 10, 11 $\cdots$ 22, 23, 22, 23. The variable N counts the number of successfully received FHS packets, by default this is set to 1. The second parameter is the number of trains used before switching to the other train and to compensate the number of used slots by the Synchronous Connection Oriented (SCO) link. The third parameter is the clock, the fourth is the Bluetooth address, and finally, the 23/79 mode.

3.2 Inquiry, Page, and Connection Procedure

For a complete connection setup between two Bluetooth devices a three step procedure must be completed which includes inquiry, page and connection procedure. Bluetooth also uses a faster FH during the time to connect procedure which involves 3200 hops per seconds.

To start the synchronization between two devices an ID packet is broadcasted which is only 68 bits short (68 μs), see Figure 3. This ID packet is transmitted during a half slot (312.5 μs) using a pre-calculated channel. In the next half slot, a new pre-calculated channel is used. To increase the probability 64 channels are used during synchronization. These channels are divided into one A_{train} and one B_{train} which each consists of 32 channels. As even slots are for transmitting and odd slots are for receiving, each train uses 16 channel for each task. Every train must be repeated $N_{inquiry}$ = 256 times (N_{page} = 128) and takes $0.3125\ \mathrm{ms/freq} \cdot 32\ \mathrm{freq} = 10$ ms to be transmitted. In total, there is a period of $256 \cdot 10\ \mathrm{ms} = 2.56$ s for each type of train for the inquiry procedure and $128 \cdot 10\ \mathrm{ms} = 1.28$ s for each type of train for the page procedure. If a SCO link is present one has to increase the train repetition to $N_{inquiry}$ = 512 times. If two SCO links are present one has to increase the train repetition to $N_{inquiry}$ = 768 times.

In the inquiry procedure the master is dedicated to look for new devices and to receive the slave's information from the FHS packet which contains information such as the Bluetooth address and the clock to be used for the page and the connection procedure. During this stage the slave device could be situated in the inquiry scan substate listening for the ID packet from the master device. The slave device does not listen all the time as this is power-consuming. For that reason the device uses a window of 11.25 ms for listening for the ID packet and saves power during the rest of the time which uses an interval of 1.28 s by default. When the FHS packet is delivered the page procedure could take place. The page procedure is made for fine-tuning the synchronization and to deliver an FHS packet containing information of the master to the slave device. Finally, the connection procedure is reached and executed. The master sends over a POLL packet that has to be confirmed and the slave device will answer with a NULL packet. All three procedures together typically run approximately in mean 3.7 s, see Figure 4.

In the inquiry procedure the slave is dedicated to listening for other devices. The channels that need to be used are decided by the different phases, see Table 1. The parameter `clk12to16` changes each time the Bluetooth clock reaches bit 12, i.e. every $4096 \cdot 0.3125$ ms = 1.28 s. Without using a train switch, the sets of channels are completely changed after 16 phases as if one had actually switched train.

Why does it take so long to set up a connection between two Bluetooth devices? Assume that the master uses the A_{train} with its set of 32 channels and that the slave uses the B_{train} with its set of 32 channels and for that reason a match does not occur, in this case during a period of 2.56 s. In Figure 5 (top), each column is represented by a train used

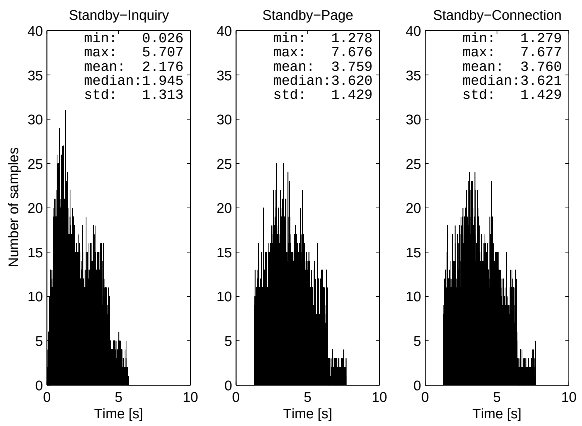

Fig. 4: Histogram plot based on the original parameters from the Bluetooth specification and simulations [8].

Table 1: Sequence of sub-channels used during different phases for the A_{train} and B_{train}.

Phase	0	···	7	8	···	14	15
0	24	···	31	0	···	6	7
1	8	⋮	⋮	⋮	⋮	⋮	⋮
⋮	⋮	⋱	⋮	⋮	⋮	⋮	⋮
8	⋮	⋮	15	⋮	⋮	⋮	⋮
9	⋮	⋮	⋮	16	⋮	⋮	⋮
⋮	⋮	⋮	⋮	⋮	⋱	⋮	⋮
15	⋮	⋮	⋮	⋮	⋮	22	⋮
16	⋮	⋮	⋮	⋮	⋮	⋮	23
17	24	⋮	⋮	⋮	⋮	⋮	⋮
⋮	⋮	⋱	⋮	⋮	⋮	⋮	⋮
24	⋮	⋮	31	⋮	⋮	⋮	⋮
25	⋮	⋮	⋮	0	⋮	⋮	⋮
⋮	⋮	⋮	⋮	⋮	⋱	⋮	⋮
31	⋮	⋮	⋮	⋮	⋮	6	⋮

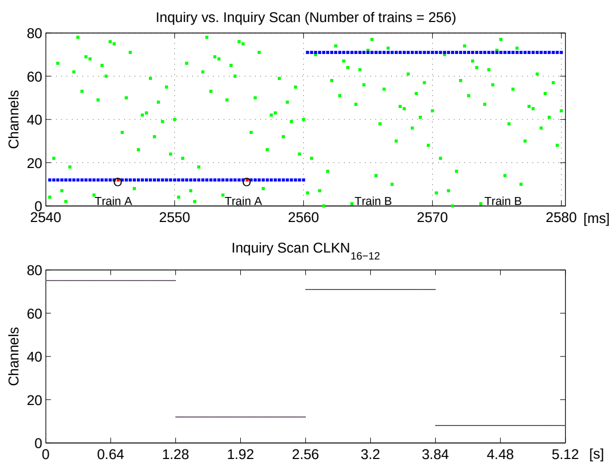

Fig. 5: Original FH scheme used by the master and slave device.

by the master (32 channels between 2.540 s and 2.550 s). Both devices have changed their phases twice during their first 2.56 s period, with the change occuring every 1.28 s. Then something remarkable occurs after 2.56 s. There is no match between the two devices until the next interval is reached at 5.12 s. To illustrate this strange behavior a plot is created in the transition seam at 2560 ms. The behavior is based on the fact that the algorithm (1) uses the train switch k_{offset} and the algorithm (2) does not. Each match between the two devices using the same channel, in time, is marked by a circle. In Figure 5 (top), the two first columns are based on the A_{train} and the last two columns are based on the B_{train} for the master which uses 32 different channels for each train. Both devices have also changed their train, but only two sub-channels/phases, see Table 1. In Figure 5 (bottom), between 1.28 s and 2.56 s, the slave uses the channel 12, and between 2.56 s and 3.84 s the slave uses the channel 72. The different channels used by the slave are plotted over the time period between time 0 s and 5.12 s.

4 Improved Time to Connect

In this section all necessary changes that drastically reduce the connection times for Bluetooth are described, see Table 2. Different plots and explanations are presented.

4.1 Train

First each train should be changed immediately after each other to increase the probability of locating each other. The variable k_{offset} is set to 1, N_{inquiry} = 1, and N_{page} = 1. In Figure 6 (top), during the period $20 - 40$ ms, both A_{train} and the B_{train} have been used. This increases the probability to 100 % to get a match between two Bluetooth devices during a period of 20 ms, which was analytically proven by Záruba *et al.* [6]. In this case there is a match in the first column which is represented by the A_{train}, marked by a circle, while there is no match in the second column. In Figure 6 (bottom), different channels used by the slave are plotted over the time period between time 0 s and 320 ms.

4.2 Phase

Both master and slave also change their channels faster due to the changed phase parameters, see Table 2. The parameter `clk12to16` is now changed to `clk7to11` which changes each time the Bluetooth clock reaches bit 7, i.e. every $128 \cdot 0.3125$ ms = 40 ms, see Equations (3) and (4). By both changing the train more rapidly and changing the phases faster the two devices are now able to find a matching channel considerably faster.

$$\mathrm{Xi}_{4-0}^{(79)} = [\mathrm{CLKN}_{11-7} + k_{\mathrm{offset}} + (\mathrm{CLKN}_{4-2,0} - \mathrm{CLKN}_{11-7}) \mod 16] \mod 32 \quad (3)$$

$$\mathrm{Xir}_{4-0}^{(79)} = [\mathrm{CLKN}_{11-7} + \mathrm{N}] \mod 32 \quad (4)$$

The change in phases for the slave and switching between the trains every time they are used for the master is the key to the high time to connect between two Bluetooth devices. By changing the representation of the bits nearer to the least significant bit (LSB), the master and the slave are able to change their channels more rapidly compared to the previous scheme. All 64 channels are now being used more rapidly. In Figure 7 (top), the slave uses the original scheme based on CLKN_{16-12}. In Figure 7 (bottom), the slave is using the improved scheme based on CLKN_{11-7}. In each column, 16 changes of the slave's channels have actually occurred.

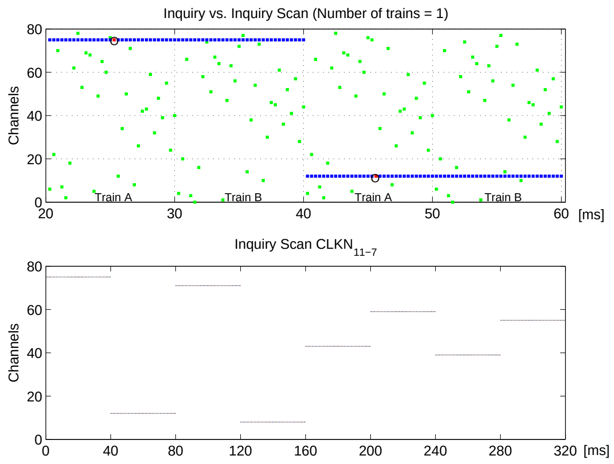

Fig. 6: Single train switch for the master device with faster sub-changes of the phases for both the devices.

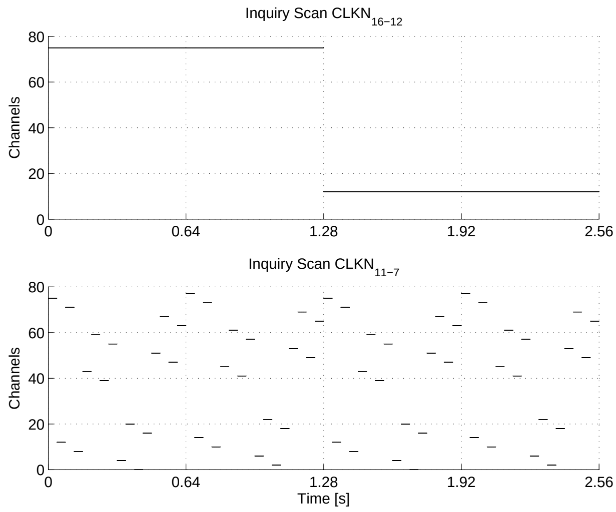

Fig. 7: Inquiry Scan CLKN_{16-12} vs. CLKN_{11-7}.

4.3 Inquiry Scan

The inquiry scan interval and the inquiry scan window are now changed to the same time period which indicates that the device is listening all the time. This is acceptable as the time to connect is faster as compared to the original scheme. These issues are discussed in detail in Section 5.

4.4 Random Backoff

The idea of using the random rackoff parameter is to decrease the probability that two or more devices respond simultaneously in case they receive the first ID packet at the same time. The rule has changed. The original value of the random backoff parameter could occupy a large time period of the total time to connect. By considerably reducing the value of the random backoff parameter the time to connect could easily be reduced to the mean value of 115 ms. See Section 5 for details.

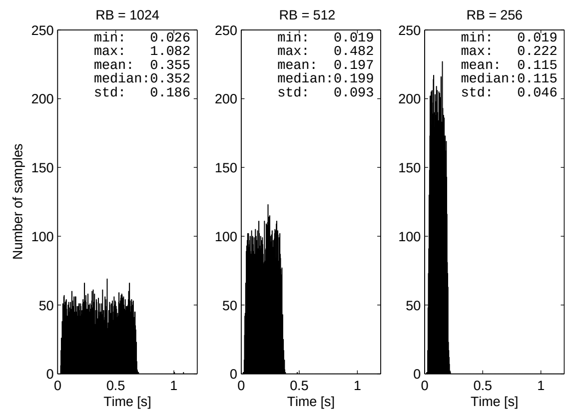

Fig. 8: Time to Connect with different random backoff values.

4.5 Time-Out

Other parameters used with different time-out values have to be reduced to reflect the improved performance of the other parameters. These issues will be discussed in detail in Section 5.

5 Simulation Results

In this section our simulation based on the new facts is presented.

Table 2: Parameters used in the simulation model.

Parameter	Value (Original)	Value (New)
Inquiry		
inquiryTO	20 s	20 s
inqrespTO	2.56 s	40 ms
T_inquiry_scan	1.28 s	8 ms
Tw_inquiry_scan	11.25 ms	8 ms
Random_backoff	0 – 640 ms	0 – 640 ms
		0 – 320 ms
		0 – 160 ms
N_inquiry	256	1
Page		
pageTO	2.56 s	40 ms
pagerespTO	5 ms	5 ms
T_page_scan	2.56 s	8 ms
Tw_page_scan	11.25 ms	8 ms
N_page	128	1
Connection		
newconnectionTO	20 ms	20 ms

Now, our simulations are changed according to the reduction of the time to connect based on the new parameters. First, the A_{train} is always used once before it changes to the B_{train}. Second, the phase parameter is changed from `clk12to16` to `clk7to11`.

For the inquiry procedure the $T\_inquiry\_scan$ and the $Tw\_inquiry\_scan$ parameters uses the same value. For the page procedure the T_page_scan and the Tw_page_scan parameter also use the same value. During the inquiry and the page procedure these parameters are no longer required. The $inquiryTO$ parameter is used to reset the inquiry procedure. The $inqrespTO$ is reduced to reflect the change of the phases for the master and slave which is 40 ms. The $pageTO$ is also reduced to 40 ms. The $pagerespTO$ uses the same value as the Bluetooth specification.

Figure 8 (left side) shows a simulation based on the random backoff parameter using a uniformed probability up to 1024 slots (640 ms) according to the Bluetooth specification. The mean value is low (0.335 s). Some values occur above 1 s. Figure 8 (middle) shows a simulation using a uniformed probability up to 512 slots (320 ms). The mean value is now lower (0.197 s). Some values do occur around 0.5 s. Finally, Figure 8 (right side) shows a simulation using a uniform probability up to 256 slots (160 ms). Now, we have not found any samples outside the main histogram and a very low mean value of 0.115 ms is established.

6 Conclusions and Future Work

Even if the Bluetooth devices in some senses satisfy the user by ”doing the job”, e.g. by creating a connection between two devices, this new approach could really attract the user by instantaneously connecting a device instead of waiting several seconds or in some cases having no connection at all.

With regards to the time to connect, there are plenty of possibilities left to be studied e.g. when one or two *Synchronous Connection Oriented* (SCO) channels are already established or when WLAN traffic is present.

Future work also implies verifying the co-existence between the two types of algorithms, one of the devices using the original parameters and the other device using the new improved parameters, i.e. to which extent it would be possible to combine devices with different configurations.

References:

[1] Hedy Kiesler Markey and George Antheil, *Secret Communication System*, Patent Serial No. 2,292,387, United States Patent Office, August 11, 1942. Application June 10, 1941, Serial No. 397,412.

[2] Specification of the Bluetooth System, Core, Version 1.1, February, 2001. http://www.bluetooth.com

[3] Specification of the Bluetooth System, Core, Version 1.2, November, 2003. http://www.bluetooth.com

[4] J. Nielsen, *Usability Engineering*. Morgan Kaufmann, San Francisco 1994.

[5] R. Rajamony and M. Elnozahy, *Measuring client-perceived response times on the www*. In Proceedings of the 3rd USENIX Symposium on Internet Technologies and Systems (USITS), San Francisco, March 2001.

[6] Záruba G. V. and Chlamtac I., *Accelerating Bluetooth Inquiry for Personal Area Networks*. In Proceedings of IEEE Global Telecommunications Conference, GLOBECOM 2003, Vol. 2, pp. 702–706.

[7] Welsh E., Murphy P. and Frantz J.P., *Improving Connection Time for Bluetooth Devices in Mobile Environments*. In Proceedings of the International Conference on Fundamentals of Electronics, Communications and Computer Sciences, 2002.

[8] L. Isaksson, M. Fiedler and Arne A. Nilsson, *Validation of Simulations of Bluetooth's Frequency Hopping Spread Spectrum Technique*. In Proceedings of the 2004 Design, Analysis, and Simulation of Distributed Systems, Arlington, Virginia, USA. pp. 156–165.