Accelerated Neighbor Discovery in Bluetooth Based Personal Area Networks

Gergely V. Záruba
The University of Texas at Arlington
Department of Computer Science and Engineering
Arlington, TX
zaruba@uta.edu

Abstract

The recent industry standard and specification Bluetooth promises low cost replacement of communication cabling with moderate symbol-rate, short-range wireless links. The same specification also addresses the establishment of point-to-multipoint *piconets* and the interconnection of several of these piconets into *scatternets*, enabling Bluetooth to be used as a technology for realizing personal area networks (PAN). In Bluetooth, to establish piconets, nodes have to go through three phases of link activation: i) neighboring device discovery or *inquiry*, ii) handshake with discovered devices or *paging*, and iii) negotiation of link parameters. In this paper we are going to investigate the first phase; point out why the current inquiry procedure is inefficient for PAN scenarios, and propose and investigate a novel but backwards Bluetooth compliant modification to the specification with which device discovery can be accelerated.

1. Introduction

Wireless networks gained significant acceptance and importance in the last decade of the previous century, while it is predicted that the next decade will bring even more recognition and users to wireless networking. This wireless revolution was ignited by 2nd generation cellular standards such as GSM and IS-95, but at the end of the last decade wireless technologies became widely accepted also for packetized data delivery by new wireless networking paradigms such as wireless local area networks (WLAN) and wireless personal area networks (WPAN). It is widely predicted that the next decade will bring never before seen increase in wireless devices and penetration.

One of the emerging short-range wireless networking technologies is the recent industry standard Bluetooth (BT) [1]. Bluetooth evolved from the need to replace annoying communication wires, e.g., cables between mobile handsets and their headsets or serial cables between computers and peripherals with short-range wireless links. Derivable from the maximum transmission power and receiver sensitivity published in the specification [1], BT has a transmission range of approximately 10 meters in a free propagation environment at a nominal ISM band frequency of 2.4GHz. BT employs frequency hopping over 79 carrier frequencies[1], each of these carriers being able to carry a symbol rate of 1Mbps. The nominal number of frequency hops in a second during communication is 1600 relating to 625μs of slot time, while the number of hops is raised to 3200 hops/second (312.5μs) during neighbor discovery and communication link establishment. The main communication structure of Bluetooth – called *piconet* – relies on a point-to-multipoint star topology, with a *master* node in the middle of the star communicating with its slave nodes at the perimeter. The specification also discusses about the interconnection of such piconets into a *scatternet*. The above features make BT a viable candidate to be used as an inexpensive technology to establish personal area networks (e.g., a network between a large amount PDAs in a meeting room to exchange data).

In order for BT devices to communicate with each other, they have to go through a three-phase link establishment procedure. In the first phase (when the units have just been turned on), devices have to scan and seek on a subset of the hopping frequencies to determine whether there are other devices in their transmission range. This *first phase* is also referred to as *neighbor discovery or inquiry*. In the next step devices that are already aware of each other's proximity will initiate a handshake process in which they exchange crucial information (e.g., ids and clock values) between each other; this phase is also referred to as *paging*. After the second phase devices have the

[1] The 2.4GHz ISM band is not equally available in all countries. For countries that have less available bandwidth in the 2.4GHz band, BT specifies another number of hopping frequencies, namely 32. In this paper we will consider the 79-hop system, yet our proposed approach can be easily adapted to the 32-hop system.

means to exchange packets between each other. The third phase is about setting up a virtual channel for further control information exchange and negotiating communication parameters relating to other management issues like authentication and security. The three-phase process outlined above also implies that although devices may be in each other's transmission-range they do not necessarily have a link established - or activated - between them, thus they may not be able to communicate with one and other. In this paper we will concentrate on the first phase – the inquiry – and show the limitations of the mandatory inquiry process and describe a novel way to accelerate device discovery. In the rest of this section we will describe the mandatory inquiry process of BT, look into previous work on BT device discovery, and briefly describe our approach.

The Bluetooth Inquiry Process

As described previously, the inquiry process' function is for nodes in each other's transmission range to become aware of each other's proximity. More precisely, a successful inquiry "handshake" between two nodes results in the initiating node acquiring knowledge about the responding node's identity and clock value. Nodes executing the inquiry process can be in either one of the following two states: inquiry (initiating) and inquiry scan (responding).

In the inquiry state, devices continuously transmit very short (68μs) so-called ID packets. The short duration of these ID packets not only enables a receiving node to efficiently correlate its receiver to this packet, but also makes the division of a normal 625μs slot into two 312.5μs "half-slots" possible. Consequently, in an even numbered slot the inquiring node will send out two ID packets at two different frequencies, determined by only the clock of the inquiring node. In an odd numbered slot, the inquiring node will tune its receiver to the corresponding frequencies of the previous two transmission frequencies; thus in 1250 μs there are two inquiry messages sent and two "waiting for response" periods. The number of different inquiry frequencies is limited to 32 (32 different and unique inquiring and 32 different and unique response frequencies) compared to the overall 79 frequencies. Furthermore, these 32 frequencies are further subdivided in two 16-frequency trains: the A- and the B-train. According to the mandatory inquiry scheme a single train has to be repeated for at least 256 times before changing to the other train.

In the inquiry scan state a node is listening for at least 18 slots on one of the 32 different inquiry scan frequencies (with a frequency change every 1.28s) waiting to overhear an ID packet from an inquiring node (again the selection of the listening frequency only depends on the devices own clock). If indeed an ID packet is overheard, the node generates a random number *b* from the interval [0,1023] and suspends the inquiry process for a duration of *b* slots. This later process has been introduced to avoid collision of responses in the (unlikely) case when more than one nodes are listening on the same frequency.

Once the timeout generated by *b* has expired, the device reenters the inquiry scan mode and responds to the first ID packet it overhears (the response should start exactly 625μs after the first bit of the ID packet has been received). The response packet is a so-called FHS packet containing the id and clock values of the responding node. In the case of the inquiring node overhearing an FHS response, it records the clock and ID value of the responding node and either continues the inquiry process or initiates a paging process (or goes back to its original state).

It can be shown that by selecting proper inquiry and inquiry scan state holding times, this process results in two devices discovering each other in less than 10 seconds yet nothing is said about the case where there are more devices competing. In a PAN case, there are two obvious and three hidden problems that slow down the inquiry process. The first problem corresponds to the fact that each time an ID packet is overheard the nodes spend a random time from a static interval in a back-logged state. The second problem corresponds to the infrequent change of the inquiry trains (2.56s), which reduces the probability that an inquiring node is transmitting on the same frequency a scanning node is listening to. The less obvious problems include: i) the situation when by the time a node returns to the scan mode after backing-off, the corresponding inquiring node has changed its state, thus no ID exchange can happen, ii) the situation when a node returns from the back-off to the inquiry scan state and replies to the first ID packet it overhears, which may be transmitted by a different device than before, and iii) since the ID packet is a predefined bit pattern, it is impossible for devices to know who issued that packet (relates to the previous problem) and thus it can easily happen that scanning nodes will reply to the same inquiring node over and over again.

Previous Work

Previous research work on Bluetooth has been mostly focused on distributed scatternet formation in PAN and ad hoc environments (e.g. [2,4] respectively), on scheduling packets in pico- and scatternets, and on performance and interference measurement in the case of the presence of other piconets or interfering electro-magnetic forces (e.g., 802.11b networks, or microwaves). Most scatternet formation approaches assume (e.g., [4]), that device discovery has already been taken place and all nodes are aware of all other neighboring devices, this is one of the

strong motivating forces behind our work. The authors in [3] derive strategies and equations based on different state holding probability distribution for the case when there is one inquiring and one scanning node present and assuming that inquiry trains change after each train.

Our Approach

In the following sections we will describe a novel approach to reduce device discovery times by measuring the number of idle slots during inquiry scans and by estimating the number of contending nodes by this measurement. The number of contending nodes is important in order to make the random backoff selection adaptive, thus reducing the number of "wasted" slots during inquiry, which can have a great impact on the speed of the inquiry process.

The rest of the paper is organized as follows: in the next section we are going to present our accelerated service discovery technique using a top-down approach. In the succeeding section we will show using simulation results, what effect the proposed measurement's resolution has on the estimated number of contending nodes. Finally we draw our conclusions and briefly outline current work in progress to further evaluate our proposed technique.

2. Accelerating Inquiry

Looking at the BT inquiry procedure, we can state that its inefficiency comes mainly from the static maximum backoff period *B* that is set to 1023. In order to accelerate the inquiry process, *B* should reflect the number of nodes actually contending for the attention of the inquiring node. In this section we are going to show how this number can be made adaptive and how it should be calculated to achieve lower device discovery times. We will look at scenarios where all nodes are in each other's proximity (typical PAN scenario) and no other interfering sources are present. We will use a top-down approach in which assumptions to derive equations will be relaxed in the respective next steps.

Calculating the Best Backoff Value - B

Let us assume that we have knowledge on the number of nodes that are in inquiry scan mode on the same frequency and denote this population by n_{sf}. Then the probability $P_b(i)$ that exactly i nodes have chosen the same random number can be determined using an (n_{sf},$1/B$) parameter binomial distribution:

$$P_b(i) = \binom{n_{sf}}{i} \left(\frac{1}{B}\right)^i \left(1-\frac{1}{B}\right)^{n_{sf}-i}$$

In order to minimize the collisions while reducing the number of wasted replying opportunities, the probability that exactly 1 node has chosen a given frequency - $P_b(1)$ - should be maximized for B. This can be done by simply differentiating according to B, making it zero and solving the acquired equation for B.

$$\frac{dP_b(i=1)}{dB} = \frac{d\,\frac{n_{sf}}{B}\left(1-\frac{1}{B}\right)^{n_{sf}-1}}{dB} = \frac{n_{sf}\,(n_{sf}-B)\left(\frac{B-1}{B}\right)^{n_{sf}}}{B(B-1)^2} = 0$$

$$B = n_{sf} \quad \text{to maximize } P_b(1)$$

Thus, the probability of a wasted (idle) slot, successful inquiry, and collision is respectively:

$$P_b(0) = \left(1-\frac{1}{n_{sf}}\right)^{n_{sf}} \cong \frac{1}{e} \cong 38\% \qquad P_b(1) = \frac{n_{sf}}{n_{sf}}\left(1-\frac{1}{n_{sf}}\right)^{n_{sf}-1} \cong \frac{1}{e} \cong 38\%$$

$$\sum_{i=1}^{\infty} P_b(i) = 1 - P_b(1) - P_b(0) \cong 1 - \frac{2}{e} \cong 26\%$$

Calculating the number of nodes scanning on the same frequency

Here we make the final assumption that each inquiring node is changing the train sequence after every train, thus inquiring not only on 16 frequencies but on all $I_s=32$ designated inquiry frequencies. Nodes in the inquiry scan mode can listen on any of these I_s frequencies, the exact frequency determined only by their native clock. Thus the probability $P_{is}(k)$, that there are exactly k nodes listening on the same frequency, assuming the knowledge on the number of nodes n_s in the inquiry scan mode, can be derived from a (ns,$1/ I_s$) parameter binomial distribution:

$$P_{is}(k) = \binom{n_s}{k}\left(\frac{1}{I_s}\right)^k\left(1-\frac{1}{I_s}\right)^{n_s-k} \quad \text{where } I_s = 32$$

The expected value of an (*n*,*p*)-parameter binomial distribution is *np*, while the most likely value is $\lfloor (n+1)p \rfloor$. Thus, a good approximation for the number of listening nodes and the maximum backoff *B* is:

$$n_{sf} = \left\lfloor \frac{n_s + 1}{I_s} \right\rfloor$$

Determining the number of nodes in inquiry scan mode

Let us denote the number of nodes in the inquiry state by n_i. Let us assume that all devices work according to the same inquiry strategy and assume that by knowing this strategy a probability distribution function $P_{IQ}(s)$ can be derived that corresponds to the probability that $s=n_s/n_i$. Let us also assume that $P_{IQ}(s)$ has a computable and finite expected value $E(P_{IQ}(s))=Q$. Now if nodes in the inquiry scan mode can determine or estimate the number of nodes in the inquiry mode - n_i, then they can estimate the number of nodes in the inquiry state mode they are contending with by: $Q*n_i$, thus:

$$B = \left\lfloor \frac{Q * n_i + 1}{I_s} \right\rfloor$$

Determining the number of nodes in the inquiry mode

Let us make the assumption that frequency of nodes changing state from inquiry to inquiry scan and vice versa is much smaller than the frequency of two train changes, thus that a change in number of nodes in the inquiry while two consecutive inquiry trains is negligible (i.e., the number of nodes performing a state change is small in a $t_{TR}=625\mu s*32=20ms$ period). This assumption can be made valid by choosing appropriate mean values for inquiry and inquiry scan durations. If nodes that just start the inquiry scan operation spend the first 32 slots by listening at the channel only, and there are no collisions among inquiry transmissions of inquiring nodes, then they could count the number of ID packets received, which would determine n_i. Yet this latest assumption is likely to not hold, thus a good estimation on n_i allowing collision should be sought. To be able to reduce the complexity of this problem, let us assume that all nodes are synchronized, i.e., there is no drift between clocks and clock ticks are synchronized. Later on we will show how this assumption may be relaxed. If a node is listening first for t_{TR} on a given frequency, then bit zero in its clock will change $s=64$ times. Devices in the inquiry mode will start their transmission exactly on these "half-slot" boundaries. The listening node will be able to tell in how many of these *s* slots there was radio energy present on the channel, thus be able to determine how many of these *s* slots were idle s_i. The question is: can the number of inquiring nodes n_i be estimated using by knowing *s* and measuring s_i? For the synchronized case a relatively simple estimation can be given.

Let us reverse the problem and assume knowledge on n_i and *s* thus attempting to determine s_i. The probability ($P_e(k)$) that a slot was used for exactly *k* transmissions can be determined by a (n_i, *1/s*) parameter binomial distribution:

$$P_e(k) = \binom{n_i}{k}\left(\frac{1}{s}\right)^k\left(1-\frac{1}{s}\right)^{n_i-k}$$

Thus the probability that a randomly selected slot is empty is $P_e(0)=(1-1/s)^{n_i}$. The probability $P_s(i)$ that exactly *i* slots are idle can be modeled again by a binomial distribution with parameters: (s,$P_e(0)$):

$$P_s(i) = \binom{s}{i}\left(1-\frac{1}{s}\right)^{i*n_i}\left(1-\left(1-\frac{1}{s}\right)^{n_i}\right)^{s-i}$$

The expected value of this function is the product of its parameters: $E(P_s(i))=s*(1-1/s)^{n_i}$. Thus it is expected that if n_i nodes have to contend for s slots, there will be $s*(1-1/s)^{n_i}$ slots empty, providing with a good estimation for s_i. Now let us solve the above equation for n_i, Thus calculating a good estimate on the number of inquiring nodes employing channel measurements with synchronized clocks:

$$n_i = \frac{\ln(s_i / s)}{\ln\left(1 - \frac{1}{s}\right)}$$

The synchronization assumption can be relaxed by enabling/enhancing nodes to measure the time t_E in which energy is present on the listening radio channel. Depending on the sophistication of the node's hardware, the resolution r_e of the measurement of t_E can be from a couple of microseconds up to the duration of an ID packet $t_{ID}=68\mu s$. A measurement resolution of $1\mu s<r_e<10\mu s$ can be easily achieved without generating a major cost increase of Bluetooth chips. In this case nodes measure energy on the channel during s slots with r_e resolution, i.e., in t_{TR}/r_e micro-slots. If the radio energy on the channel during a micro-slot does not go above a given threshold, then s_i will be incremented. The selection of the resolution time r_e has a significant impact on the accuracy of n_i's estimation as we are going to see in the next section. The estimated value of n_i with r_e resolution of measurements and s_m idle micro-slots can be given as:

$$n_i = \frac{\ln(s_m * r_e / t_{TR})}{\ln(1 - r_e / t_{TR})} * \frac{r_e}{t_{ID}}$$

Now the optimal backoff B can be determined by:

$$B = \left\lfloor \frac{Q * \frac{\ln(s_m * r_e / t_{TR})}{\ln(1 - r_e / t_{TR})} * \frac{r_e}{t_{ID}} + 1}{I_s} \right\rfloor$$

Leaving Q as the only variable that yet needs to be determined.

Determining the Proportion of Nodes in the Inquiry Scan State

Obviously, the inquiry policy will have a significant impact on the proportion of nodes in the inquiry state versus nodes in the inquiry scan state. Thus the value of Q will depend on this policy. Here we will calculate Q for a probabilistic inquiry scheme relying on more or less uniform distribution of state holding times. In [3] it has been shown that deterministic state holding times will result in device discovery times with an infinite expected value.

Let us determine the value of Q for a simple state holding time distribution. It is necessary for the previous assumptions to hold, that each node has a lower bound on the number of slots it spends in each of the inquiry states. Obviously, the more time they spend in the individual states at a time, the more likely it will be that they discover each other. On the other hand, since the device discovery procedure is asymmetric (i.e., only because node i discovered node j, node j will not have knowledge on node i's identity), nodes have to change states frequently. Additionally, the only way to avoid continuously overlapping transmissions of different inquiring nodes at the same frequency (of which the likelihood is growing with the population) is to make sure they eventually switch states (and do that independently). The determination of the best state holding times and distribution is beyond the scope of this paper but efforts are underway to published it in a subsequent article.

In our simple model each node is required to spend k_0 time slots in inquiry state and then an additional k_1 slots that is randomly chosen from the interval $[0, 2K]$. Furthermore every node has to assume the inquiry scan state for l_0 time slots and then an additional l_1 slots that is randomly chosen from the interval $[0, 2L]$. Another simplified way of describing this model is that after spending k_0 , and l_0 time slots in the appropriate states, each node is changing to the other state with a probability of K and L into the inquiry scan/inquiry states respectively. Thus in average a node spends k_0+K-1 slots in the inquiry and l_0+L-1 slots in the inquiry scan state. Consequently the estimated value for the ratio of the state holding times can be given as: $Q=(l_0+L-1)/(k_0+K-1)$. Therefore the optimal backoff value can be estimated by:

$$B = \left\lfloor \frac{\frac{l_0 + L - 1}{k_0 + K - 1} * \frac{\ln(s_m * r_e / t_{TR})}{\ln(1 - r_e / t_{TR})} * \frac{r_e}{t_{ID}} + 1}{I_s} \right\rfloor \qquad [1]$$

Note, that if all nodes work according to the described inquiry strategy, then in the above equation all variables are known after the first time a device spends t_{TR} time in the inquiry scan state.

3. Evaluation of Accelerated Inquiry

In this section we are going to investigate what impact the resolution of the slot measurements has on the accuracy of the prediction of the population of nodes in the inquiry state. Evaluation of the proposed inquiry acceleration technique in a simulated Bluetooth environment will be given in a subsequent paper.

Intuitively, the higher the resolution of the measurement is (i.e., the smaller t_r is), the better the estimation of the population - n_i of inquiring nodes will be. To study the effect of the resolution on n_i and to evaluate the estimation given by Equation 1, we have simulated N number of nodes accessing a channel randomly during t_{TR}=*20ms* duration. For each *N* and t_r value, we have run enough simulations to be able to claim, that we are 95% sure, that our results have less than 5% error of margin. In the following figures, we will draw the real and estimated number of nodes in function of the empty micro-slots: s_m. Figure 1a depicts the case where t_r was set to 1μs, a value significantly less than the duration of an ID packet t_{ID} , the figure clearly shows that Equation 1 provides a good estimation in this case.

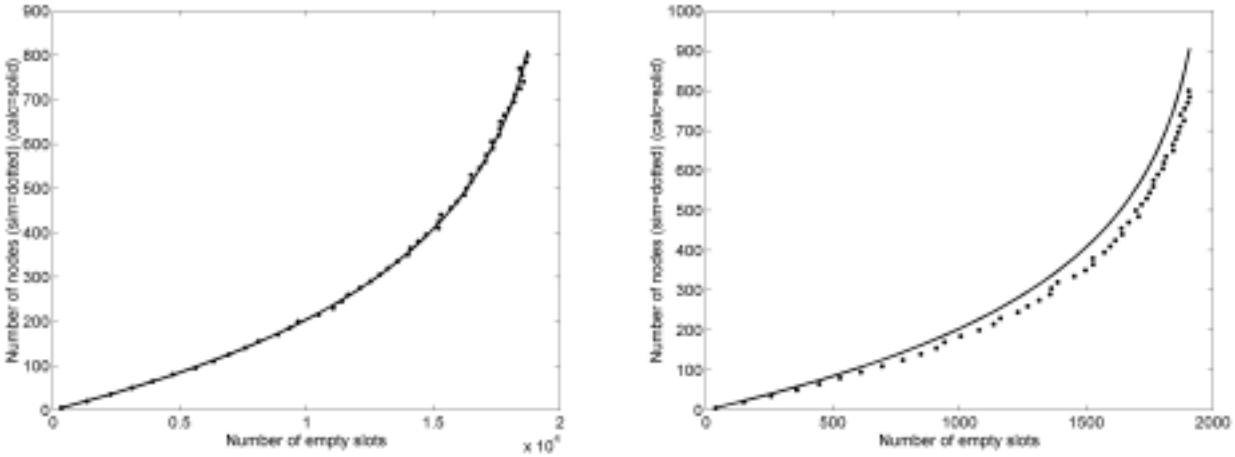

Figure 1. Simulated and Calculated Populations for a) 1μs and b) 10μs Measurement Resolution

In Figure 1b we are showing results for t_r=10μs and yet we can state that the estimation does not significantly differ from the real population value. Figure 3 shows the situation where $t_r = t_{ID}$, in this case the error of the estimation can be as much as the population. Thus it is instrumental that the measurement time t_r be at least approximately an order of magnitude less than the duration of an ID packet. This can also be seen in Figure 4, where the difference between the calculated and real value of the population is shown in function of both the resolution and a normalized value of idle slots. Furthermore we can state that since t_{TR} has a fixed value, the measurement method will require the number of nodes to be below approximately a thousand nodes.

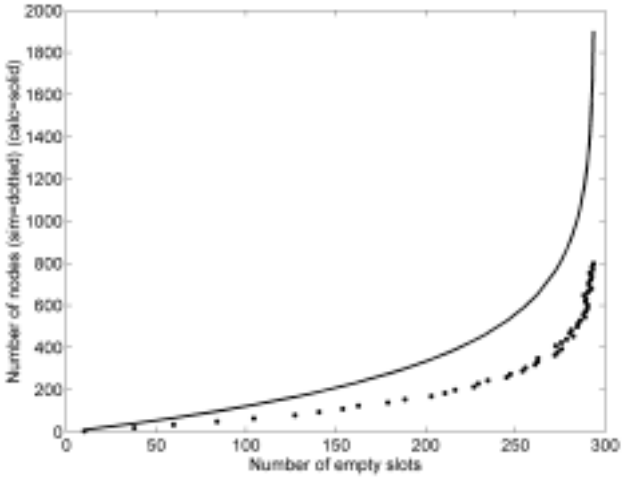

Figure 3. Simulated and Calculated Populations for 68μs Measurement Resolution

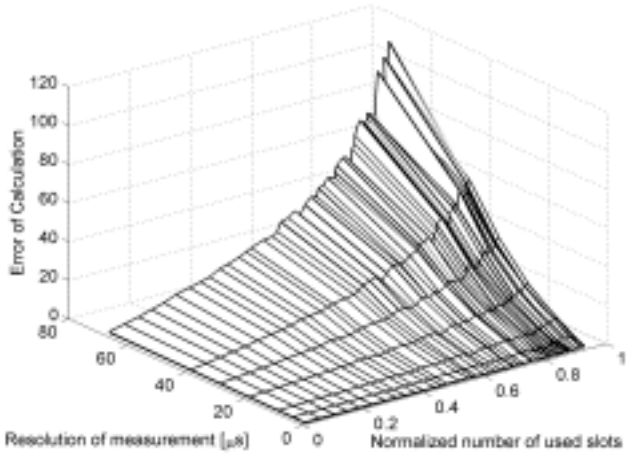

Figure 4. Difference Between Calculated and Simulated Populations

4. Conclusions and Future Work

In this paper we presented a novel way to reduce the device discovery times in full-proximity Bluetooth PANs, where all nodes are assumed to be in the other nodes transmission range. Our technique is based on the observation that in the original inquiry specification, the backoff value does not reflect the number of contending nodes on the channel, but is instead set statically to an extremely high value. We have shown by a top-down analytical approach, that the backoff value can indeed be made adaptive to the population by a small modification to the nodes, where they are enabled to measure the channel they are listening on. We have shown the impact of the resolution of the measurement on the accuracy of the estimation by simulations.

Work is underway to simulate the proposed inquiry acceleration technique in a Bluetooth environment and to calculate and compare device discovery times achieved by our technique and the original mandatory inquiry scheme. Future work also include considerations of listening nodes changing channels, in which case they have to keep track of which nodes they already have replied to – another situation where a higher resolution of measurement can provide with reduced device discovery times. We will also evaluate what impact symmetric discovery (in which nodes that complete an inquiry handshake initiate a connection set-up right away) will have on device discovery times.

5. Acknowledgments

We express our sincere gratitude to Dr. Farhad Kamangar of CSE@UTA, who was always available to brainstorm on the topics addressed by this paper.

6. References

[1] Bluetooth Special Interest Group, "*Specification of the Bluetooth System, Core Version 1.1*", Specification Volume 1, *http://www.bluetooth.com*, February 2001.

[2] T. Salonidis, P. Bhagwat, L. Tassiulas, and R. LaMaire, "*Distributed Topology Construction of Bluetooth Personal Area Networks*," Proceedings of the IEEE INFOCOM 2001, pp.1577-1586, Anchorage, Alaska, April 2001.

[3] T. Salonidis, P. Bhagwat, L. Tassiulas, and R. LaMaire, "*Proximity Awarness and Ad Hoc Network Establishment in Bluetooth*," Institute of Systems Research (ISR), University of Maryland Technical Report, 2001.

[4] G.V. Záruba, S. Basagni, and I. Chlamtac, "*Bluetrees-scatternet formation to enable Bluetooth-based ad hoc networks*," Proceedings of the IEEE International Conference on Communications, ICC 2001, vol. 1, pp.273-277, Helsinki, Finland, June 2001.