

Simplified Bluetooth Device Discovery – Analysis and Simulation

Gergely V. Záruba, and Vishant Gupta

Center for Research in Wireless Mobility and Networking (CReWMaN)

Department of Computer Science and Engineering (CSE@UTA)

The University of Texas at Arlington

zaruba@uta.edu, vishantgupta@hotmail.com

Abstract

Bluetooth is a promising wireless technology enabling (portable) devices to form short-range star-shaped wireless networks (or wireless personal area networks - PAN). Bluetooth relies on a frequency hopping physical layer, implying that hosts are not able to communicate unless they have previously discovered each by synchronizing their frequency hopping patterns. This further implies, that even if all nodes are within direct communication range, only those nodes that are synchronized within the personal area network can overhear a transmission. To support any-to-any communication such personal area networks (or in Bluetooth terminology piconets and possibly scatternets) need to be established among nodes. Thus, it is of utmost importance to be able to model or predict the time required for nodes to discover each other, i.e., to synchronize them. This paper provides a mathematical analysis and corresponding simulation results for the Bluetooth discovery (inquiry) time in a fully connected (personal area network) situation with the population as a parameter. The simplification to the analysis and simulation is due to the intermediate train changes in the inquiry process as it is outlined in the paper.

1. Introduction

Bluetooth is a promising technology aimed to support wireless connectivity among cell phones, headsets, PDAs, digital cameras, and laptop computers. The initial intention of the technology was to define a short-range wireless replacement for peripheral cables, but in due time solutions for point-to-multipoint and multi-hop networking over Bluetooth may/will evolve.

Most research in inexpensive wireless networks – may they be infrastructure-based such as wireless LANs or infrastructure-less such as ad hoc networks – rely on the assumption that once nodes are in each others communication range they can exchange

packets among themselves. Since in these cases a single channel is used throughout the network, the topology of the network is implicitly (and uniquely) determined by proximity among the participating nodes. The IEEE802.11 standard (and its extensions) further supports such assumptions by its nature¹.

On the other hand, Bluetooth employs a polling based medium access control (MAC) sub-layer with an underlying fast-frequency hopping physical layer. To be able to communicate, nodes have to synchronize themselves to the polling coordinator node (the master), which also determines the frequency hopping sequence. This entails that in Bluetooth nodes that are in each other's communication range may not even know about the other nodes' existence, i.e., physical proximity does not imply the existence of communication links. In fact, Bluetooth devices need to undergo a three-phase device discovery/link establishment procedure before they can be placed in such a master-slave polling structure (called a *piconet*).

1.1. Previous Works and Our Contribution

According to the best of our knowledge there have been no previous works or attempts to model the device discovery (inquiry) phase of Bluetooth for arbitrary number of nodes. The only three papers dealing with Bluetooth device discovery to date are [3,4] and [5].

In [3] the authors derive a probability density function for the Bluetooth inquiry time assuming that there are only two nodes that need to discover each other. These results are then used to outline that the duration of the inquiry and inquiry scan states need to be probabilistic for a non-infinite expected inquiry time.

In [4] we have shown that the major drawback of the Bluetooth discovery process is due to the static random back-off period built into the Bluetooth

¹ The Distributed Coordination Function (DCF) of 802.11 is a CSMA/CA based MAC protocol.

specification. We have also outlined a solution to adaptively (to the population of the network) change the back-off period to near its optimal value.

The authors of [5] have also considered inquiry issues in their paper although the focus was more on the scatternet formation issue. They have identified the random back-off as the major bottleneck for multi-node operation. Furthermore, [5] proposed a synchronous inquiry procedure, where if one node discovered another node, the other node would also gain knowledge about the first node. This was done by establishing communication (immediate paging) after a successful inquiry and thus informing each other about all required device properties.

In this paper we are concentrating on the original (asynchronous) inquiry scheme of Bluetooth and derive formulas to model the total inquiry time of node-sets with an arbitrary population. We assume that all nodes are in each other's proximity and that nodes change between A- and B-trains immediately after a train is over. We derive two formulas:

a formula for the total inquiry time with a certain confidence (parameter) that all nodes have discovered all the other nodes and

a formula for the total inquiry time where a certain percentage (parameter) of all inquiries have taken place.

We then validate our formulas via simulation results obtained by a discrete event C++ simulation developed to closely model the Bluetooth inquiry process.

1.2. Introduction to Bluetooth Inquiry

The *Bluetooth Baseband Specification* [1,2] defines the Bluetooth point-to-point connection establishment as a three-step procedure. First neighborhood information is collected through the *inquiry* Procedure. A *paging* procedure may be subsequently used to establish connections between neighboring devices, while in the final phase piconet properties are negotiated upon. Both the inquiry and paging procedures are asymmetric processes. During inquiry, transmitting nodes discover and collect neighborhood information provided by responding nodes that are in an inquiry scan state.

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 Bluetooth clock value. Thus, 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 (68 μ s) so-called ID packets as the Inquiry Access Code (IAC). The short duration of these ID packets does not only enable the receiving node to efficiently correlate its receiver to this packet, but it will also make the division of a normal 625 μ s slot into two 312.5 μ s "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 μ s there are two inquiry messages sent and two "waiting for response" periods. The number of different inquiry frequencies is limited² 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. To simplify our analysis in this paper, we will assume that a custom inquiry process is used, where train changes occur after each train sequence, i.e., none of the trains is repeated – an A train follows a B train immediately and vice versa.

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 device's own clock). If indeed an ID packet is overheard, the node generates a random number b from the closed 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 node is 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 625 μ s 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

² In this paper we restrict ourselves to the U.S. (and most European Countries) version of Bluetooth, which uses 79 hopping frequencies among which 32 can be used for inquiry and another 32 for inquiry response.

ID value of the responding node and either continues the inquiry process or initiates a paging process (or goes back to its original state). Given two units in the two complementary inquiry states respectively, we refer to the term *frequency synchronization delay* (or *FS delay*) as the time until the sender transmits at the frequency the receiver is currently listening on. In [4] several minor and major drawbacks of the inquiry process have been identified; the interested reader is referred to that publication.

The rest of the paper is organized as follows. In section 2 we provide with the inquiry model used in our analysis and present two different analytical methods to derive inquiry duration formulas. Section 3 presents simulation results on the total inquiry time comparing these results to the analytical outcomes. Section 4 concludes the paper and outlines our current research directions.

2. Simplified Model of the Bluetooth Inquiry Procedure

In our analytical model we assume that nodes enter either the inquiry or inquiry scan states randomly (as outlined in [3]). As it was described in the previous section, there is an initial FS delay until the sender (the node in the inquiry state) synchronizes to the frequency the receiver (the node in the inquiry scan state) is listening to. Upon receiving the IAC packet, the receiver backs off for an amount of time that is discrete uniformly distributed between 0 and 639.375ms (with a 625 μ s resolution corresponding to a [0,1023] slot delay). This happens in order to prevent the contention problem that would arise if there were more than one receivers listening on the same hop frequency. If both of them responded immediately, the response message would get garbled and thus the sender would not receive any of these messages. (To make the matter even worse, the nodes that were listening on the same frequency would likely keep listening on the same frequencies again at later times). Let us denote the duration of the receiver's back-off as the *random back-off delay* (or *RB delay*). When the receiver unit is awakening from the back-off, it starts listening again on a frequency determined by its own Bluetooth clock. After approximately another FS delay, a second IAC packet is received from the sender (assuming that the sender is still in the inquiry state). Consequently, the receiver will reply with an FHS packet containing:

The receiver's address used by the sender to derive the device access code (DAC) of the receiver and the page hopping sequence it will use later in order to page the receiver.

The receiver's clock value - used to estimate the phase of the receiver thus eliminating the FS delay during the paging procedure that follows.

The timing diagram in Figure 1 summarizes the point-to-point connection establishment procedure between two units. The dashed arrows symbolize events on each unit's timeline and each event is numbered in the order it happens during the connection establishment procedure.

After a successful inquiry, nodes may continue with a paging process to establish a piconet. The timing diagram shows the receiver entering the PAGE SCAN state after having sent the inquiry response FHS packet to the sender. When the sender receives the FHS packet it enters the PAGE state and uses the clock information in the FHS packet to send a DAC packet on the frequency the receiver is listening to in the PAGE SCAN state. Then, the receiver responds immediately with a DAC packet and the sender sends an FHS packet to the receiver. The receiver uses the FHS information to determine the channel hopping sequence and the phase of the sender and becomes the slave of the point-to-point connection. It then acknowledges the FHS packet with another DAC packet. As soon as the sender receives the acknowledgment, it becomes the master of the connection and may start exchanging data with the synchronized receiver-slave.

By observing Figure 1 we can easily identify the link formation delay components. The inquiry procedure delay consists of a first FS delay, the RB delay and a second FS delay that is taking place when the receiver waits for the second IAC packet after it wakes up. Thus we can approximate the inquiry delay R using the following equation:

$$R = 2 * FS + RB , \quad (1)$$

where FS and RB are uniform random variables in $[0, T_{\text{coverage}}]$ and $[0, r_{\text{max}}]$ respectively. According to equation 1, the link formation delay can be at most $2T_{\text{coverage}} + r_{\text{max}} = 40\text{ms} + 639.375\text{ms} = 679.375$. Since both FS and RB are uniform random variables, the mean of the random variable representing their sum is the sum of their means [6].

2.1. Mathematical Model for Inquiry Duration

In order to analyze inquiry process of Bluetooth we will use the following denotations and assumptions:

Instead of having 2 trains (A and B) we consider only one train with 32 hopping frequencies. Thus, duration of 32 hops of an inquiry node will be $32 \times 625 \times 10^{-6} = 20\text{ms}$. (Note, that this is in line with the explanation in the previous section where trains A and B are immediately following each other.)

Probabilistically, half of the time a node is in the inquiry scan state while in the other half in the inquiry state, i.e., the probability that at any time a node is in the inquiry scan state is $P_{scan}=0.5$ and the probability that at any time a node is in the inquiry state is

$P_{inq}=1-P_{scan}=0.5$. Note, that we assume that nodes are not involved in anything else but in the inquiry process (no paging or communication is enabled while the device discovery lasts). We will denote the population of nodes by n .

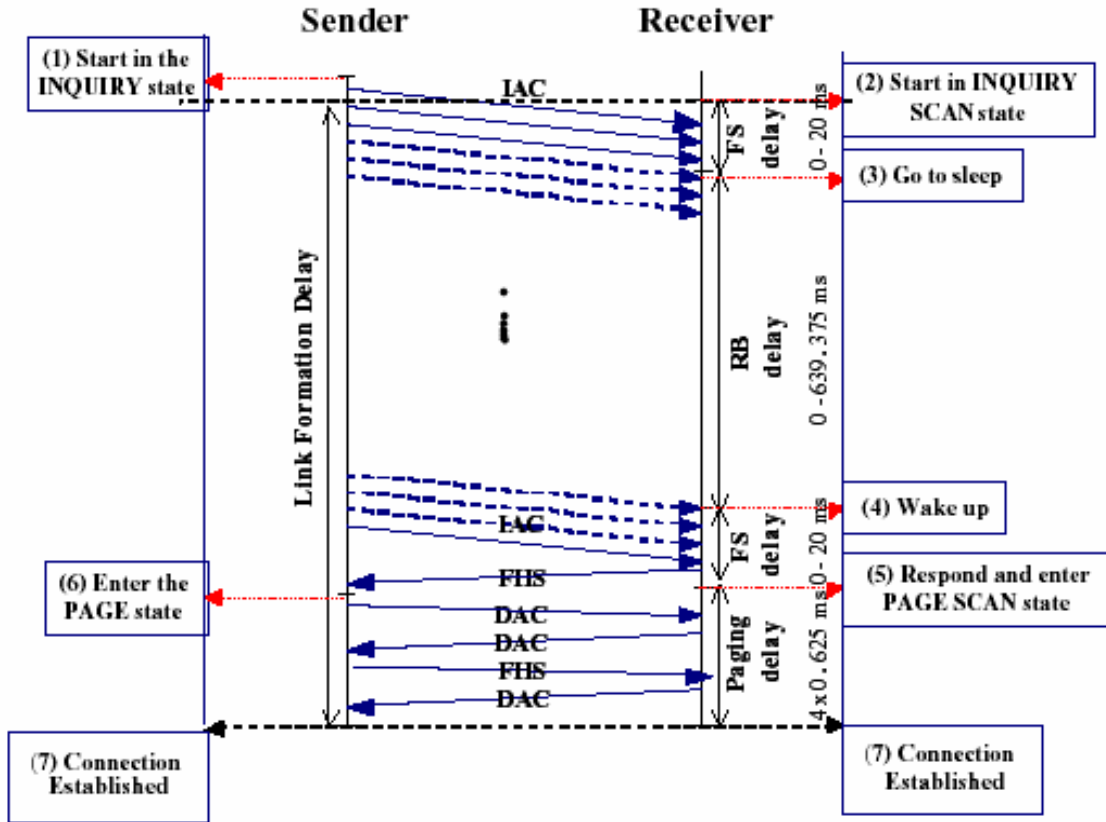


Figure 1. The Bluetooth asymmetric link formation protocol.

Duration of states (inquiry scan or inquiry) are modeled by I.I.D. exponential random variables with parameter λ . (Since $P_{inq}=P_{scan}$ the parameters of the “state holding times” for both states are equal.) We denote such a Poisson process by X .

The average time for any node to be discovered by all other nodes is referred to as the *Average Total Inquiry Time* or T_c .

RB and FSD denote the duration of the random back-off and the frequency synchronization delay respectively. Maximum values of RB and FSD are given by $RB_{max}=1023 \times 312.5 \times 10^{-6}=319.6875ms$ and $FSD_{max}=\text{Duration of 32 frequency hops of an inquiring node}=20ms$.

We define the term *Successful Inquiry Attempt* as an event where an inquiring node has successfully attempted to discover a scanning node (i.e., an inquiry handshake has taken place).

Figure 2 depicts 4 nodes alternating between inquiry and inquiry scan states according to exponential “state holding” times. The merged process X is also depicted. Note, that X has a change in state whenever there is a state change in any node of the population. Since according to our assumptions, the state-changes in the nodes are exponentially distributed, a state change in the merged process will also be exponentially distributed [6].

A *Successful Inquiry Attempt* between two nodes in complementary states is made if and only if the scanning node remains in the scanning state for 2 synchronization delays plus a random back-off delay, i.e., $2 \times FSD + RB \leq X$. Let us denote the random variable on the left side of the previous equation by R , i.e., $R=2 \times FSD + RB$. Since both FSDs and RBs are I.I.D. uniform random variables in $[0, FSD_{max}]$ and $[0, RB_{max}]$ respectively, therefore R can be described as a distribution in $[0, 2 \times FSD_{max} + RB_{max}]$ or $[0, r_{max}]$, where $r_{max}=2 \times FSD_{max} + RB_{max}=359.6875 ms$.

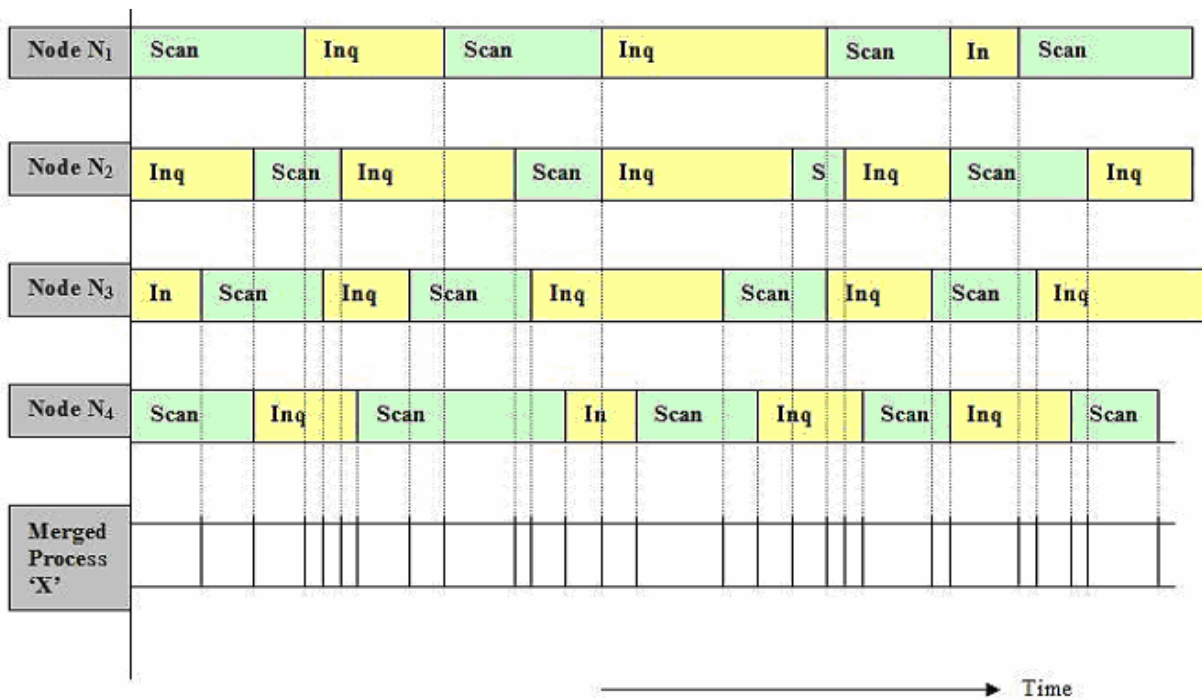


Figure 2. Alternating inquiry states of 4 nodes with a merged state process.

Let us denote the probability that a *Successful Inquiry Attempt* is made by a scanning node by P_c . Then P_c can be given by the product of $P(X \geq R)$ and the probability of having one or more nodes in inquiry-scan state out of remaining $n-1$ nodes. The second member of the product can also be described as 1 minus the probability of having all other nodes in the inquiry state. Thus:

$$\begin{aligned}
 P_c &= P(R \leq X) * \left(1 - \frac{1}{2^{n-1}}\right) = \\
 &= \left(1 - \frac{1}{2^{n-1}}\right) * \left[\int_0^{\infty} P(R \leq x / X = x) * f_x(x) dx \right] = \\
 &= \left(1 - \frac{1}{2^{n-1}}\right) \left[\int_0^{r_{\max}} \frac{x * f_x(x) dx}{r_{\max}} + \int_{r_{\max}}^{\infty} f_x(x) dx \right] = \\
 &= \left(1 - \frac{1}{2^{n-1}}\right) \left[\int_0^{r_{\max}} \frac{x * \lambda * e^{-\lambda x} dx}{r_{\max}} + \int_{r_{\max}}^{\infty} \lambda * e^{-\lambda x} dx \right] = \\
 P_c &= \left(1 - \frac{1}{2^{n-1}}\right) \left[\frac{1}{r_{\max}} \left\{ \frac{1}{\lambda} - e^{-\lambda r_{\max}} \left(r_{\max} + \frac{1}{\lambda} \right) \right\} + e^{-\lambda r_{\max}} \right] \quad (2)
 \end{aligned}$$

As a node in the Inquiry scan state can make more than one *successful inquiry attempt* in a scan interval,

let us denote P_k as the probability of making exactly k *successful inquiry attempts* in a scan interval, where P_k can be given by, $P_k = (P_c)^k * (1 - P_c)$. Furthermore, let us denote the average number of *successful inquiry attempts* in a scan interval by N_{scan} and thus derive it as the estimated value of k :

$$N_{\text{scan}} = E[k] = \sum_{i=0}^{\infty} i * P_c^i (1 - P_c) = \frac{P_c}{(1 - P_c)} \quad (3)$$

If we substitute Equation 2 into Equation 3 we obtain N_{scan} to be:

$$\begin{aligned}
 &\left(1 - \frac{1}{2^{n-1}}\right) * \left[\frac{1}{r_{\max}} \left\{ \frac{1}{\lambda} - e^{-\lambda r_{\max}} \left(r_{\max} + \frac{1}{\lambda} \right) \right\} + e^{-\lambda r_{\max}} \right] \quad (4) \\
 &\left[1 - \left(1 - \frac{1}{2^{n-1}}\right) \left[\frac{1}{r_{\max}} \left\{ \frac{1}{\lambda} - e^{-\lambda r_{\max}} \left(r_{\max} + \frac{1}{\lambda} \right) \right\} + e^{-\lambda r_{\max}} \right] \right]
 \end{aligned}$$

One of the pathological situations in the Bluetooth inquiry process is the multiple re-inquiry phenomenon: since all nodes are transmitting the same IAC, nodes in the inquiry scan state do not know who the originator of the inquiry message is. Thus it can easily happen that two nodes re-discover each other over and over again. Obviously such successful inquiry attempts must not be considered towards the goal of finding the time for all possible inquires between all possible node pairs. In the next two subsections we explore two different ways to deal with these situations.

2.2. Average Total Inquiry Time with Varying Confidence

Let us introduce the term *successful inquiry* as a special case of a *successful inquiry attempt*, where the successful inquiry attempt happened between two nodes that have not have such a successful attempt before. Furthermore, let us denote the probability of having $(n-1)$ *successful inquiries* in k *successful inquiry attempts* via one scanning node by P_k . Evidently, if $k < n$ then $P_k = 0$. Then P_k can be derived as

$$P_k = \frac{(n-1)!}{n^k} \left[\sum_{l_j=0}^{n-1} (n-1)^{l_1} (n-2)^{l_2} \dots * 2^{l_{n-2}} * 1^{l_{n-1}} \right] \quad (5)$$

where $l_1 + l_2 + \dots + l_{n-1} = k - (n-1)$

Although the above expression is intuitive it is not in a well-computable form. With some mathematical tools and tricks, Equation 5 can be reformulated and shown to be equal to:

$$P_k = 1 - \sum_{j=1}^{n-2} \binom{n-1}{j} \left(\frac{n-1-j}{n-1} \right)^k * (-1)^{j+1} \quad (6)$$

Equation 6 is easily computable by simple numeric methods. To shed more light into the behavior of the asynchronous inquiry system, we have plotted the cumulative distribution function and the probability mass function of P_k versus the number of successful inquiry attempts for four different node populations in Figure 3. As a numerical example we can see that if we have $n=30$ nodes discovering each other, then the probability that a randomly picked node is discovered by all of its neighbors assuming that it was discovered 100 times is around 40%. We can also see that the more the population the less the slope of the main part of the curves become thus lengthening the inquiry process impropotionally. As another example if we want to have a 50% confidence that a node has been discovered by all of its neighbors (with a population of $n=50$) then one has to wait until this node has responded to around 210 inquiries.

The average number of *Successful Inquiry Attempts* (N_{total}) by a single scanning node to have a 50% confidence that all other nodes discovered it, can be denoted and given by $P_{k|k=N_{total}} \geq 0.5$, i.e.,

$$\left[1 - \sum_{j=1}^{n-2} \binom{n-1}{j} \left(\frac{n-1-j}{n-1} \right)^k * (-1)^{j+1} \right]_{k=N_{total}} \geq 0.5$$

from which N_{total} can be evaluated.

Let N_{scan_int} be the average number of scan intervals observed by all other nodes. N_{scan_int} can be given by the average number of *successful inquiry*

attempts required by a scanning node to be discovered by all other nodes divided by the average number of *successful inquiry attempts* made by a scanning node in one scan interval:

$$N_{scan_int} = \frac{N_{total}}{N_{scan}} \quad (7)$$

If T_c is represented as the average total time required for a node to be discovered by all other nodes, then T_c can be given by the product of the double of the average number of scan intervals required and the average duration of a scan interval as shown in Equation 8.

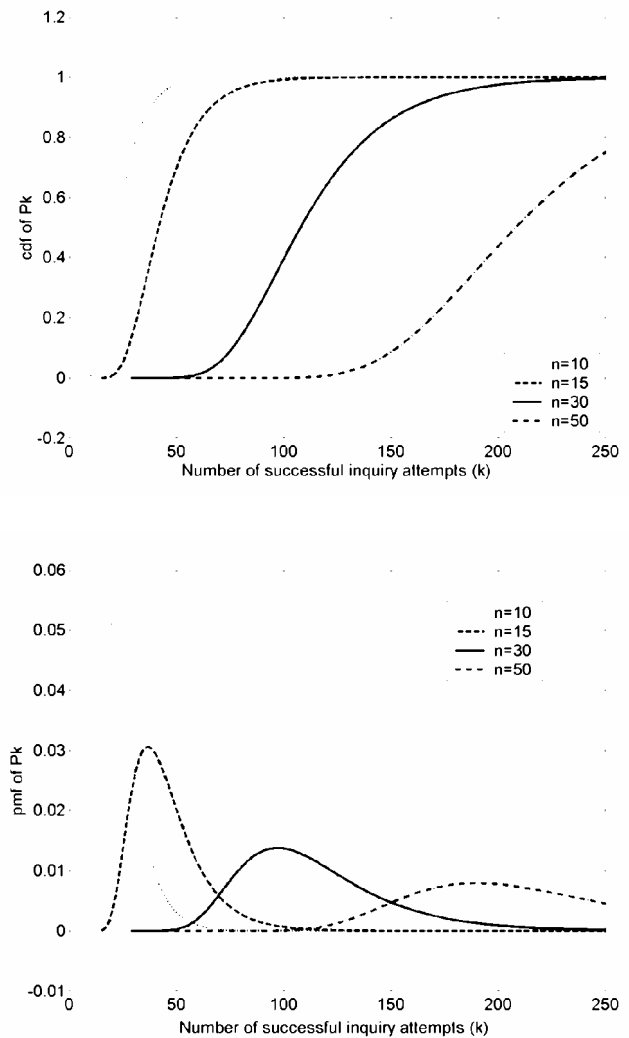


Figure 3. cdf and pmf of P_k versus the number of successful inquiry attempts.

Using Equation 8 we have evaluated T_c for different populations 'n' (with 50% confidence that all nodes have detected each other) as shown in Figure 4. Note, that both axes are logarithmically scaled. As a numerical example consider a population of a 100

nodes; if we want to be 50% confident that all nodes have discovered each other we have to wait for approximately 240 seconds.

2.3. Average Total Inquiry Time Varying Proportions of Nodes Discovered

In the previous subsection we have derived a formula with which we can calculate the total inquiry time if we want to have some level of certainty that all nodes have discovered each other. Our second method for indexing the total inquiry time allows us to determine how long it takes in average for a certain percent of all successful inquiries.

Let us refer to $P_{SC}(i)$ as the probability that the i^{th} *successful inquiry attempt* is indeed a *successful inquiry*. Furthermore, let us denote the average number of *successful inquiries* until i^{th} *successful inquiry attempt* by $S_c(i)$. Table 1 shows an intuitive iterative-method to calculate $P_{SC}(i)$ and $S_c(i)$ for different values of i , thus obtaining a generalized result.

Obviously, if we want to be sure that all nodes have been discovered, then $S_c(N_{\text{total}}) = n-1$ but $n-1$ is the value to which $S_c(N_{\text{total}})$ converges when N_{total} grows to infinity ($\lim_{K \rightarrow \infty} S_c(N_{\text{total}}) = n-1$).

$$T_c = 2 * N_{\text{scan_int}} * \frac{1}{\lambda} = \frac{N_{\text{total}} * 2}{N_{\text{scan}} * \lambda} = \frac{N_{\text{total}}}{\left(1 - \frac{1}{2^{n-1}}\right) * \left[\frac{1}{r_{\text{max}}} \left\{ \frac{1}{\lambda} - e^{-\lambda * r_{\text{max}}} \left(r_{\text{max}} + \frac{1}{\lambda} \right) \right\} + e^{-\lambda * r_{\text{max}}} \right]} * \frac{2}{\lambda} \quad (8)$$

$$T_c = \frac{n * N_{\text{total}}}{\left(1 - \frac{1}{2^{n-1}}\right) * \left[\frac{1}{r_{\text{max}}} \left\{ \frac{1}{\lambda} - e^{-\lambda * r_{\text{max}}} \left(r_{\text{max}} + \frac{1}{\lambda} \right) \right\} + e^{-\lambda * r_{\text{max}}} \right]} * \frac{2}{\lambda} \quad (10)$$

We can now plot the total inquiry time versus the number of nodes as depicted in Figure 5. As a numerical example if in average we would like 99% of the nodes to be discovered by all other nodes at a population of a hundred nodes, we again have to wait around 240 seconds. The overlap between Figures 4

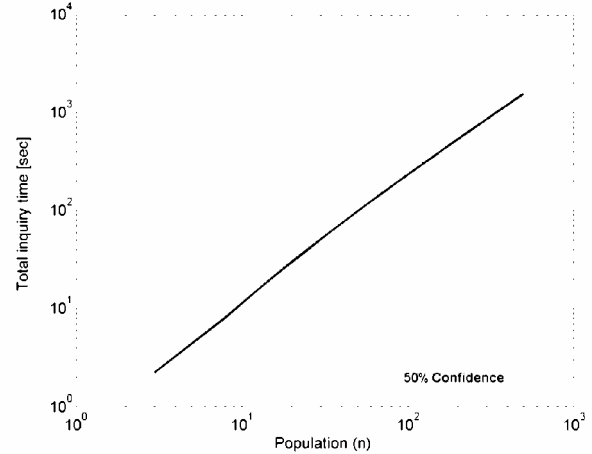


Figure 4. Average total inquiry time versus population (50% confidence).

Intuitively, if we want to be sure that all nodes have discovered each other we have to let the process run for infinite time. Let us therefore calculate N_{total} such that 99% nodes are detected, i.e.,

$$\sum_{j=1}^{N_{\text{total}}} P_{SC}(j) = 0.99 * (n-1) \quad (9)$$

If we solve Equation 9 for N_{total} numerically and substitute it into Equation 7 (the same way as Equation 8 was derived) we obtain our formula for the total inquiry time as:

and 5 indicates that the total inquiry time for a 50% of confidence that all nodes have discovered each other lies somewhere at the same level as the time required for 99% of the nodes to discover each other (for a population of a hundred nodes).

Table 1. Values of P_{SC}(i) and S_c(i).

| Index | P _{SC} (i) | S _c (i) |
|-------|--|--|
| 1 | 1 | 1 |
| 2 | $1 - \frac{1}{n-1}$ | $1 + \left[1 - \frac{1}{n-1}\right]$ |
| 3 | $1 - \frac{1 + \left[1 - \frac{1}{n-1}\right]}{n-1}$ | $1 + \left[1 - \frac{1}{n-1}\right] + \left[1 - \frac{1 + \left(1 - \frac{1}{n-1}\right)}{n-1}\right]$ |
| ⋮ | ⋮ | ⋮ |
| ⋮ | ⋮ | ⋮ |
| ⋮ | ⋮ | ⋮ |
| I | $1 - \frac{\sum_{j=1}^{i-1} P_{SC}(j)}{n-1}$ | $\sum_{j=1}^i P_{SC}(j)$ |

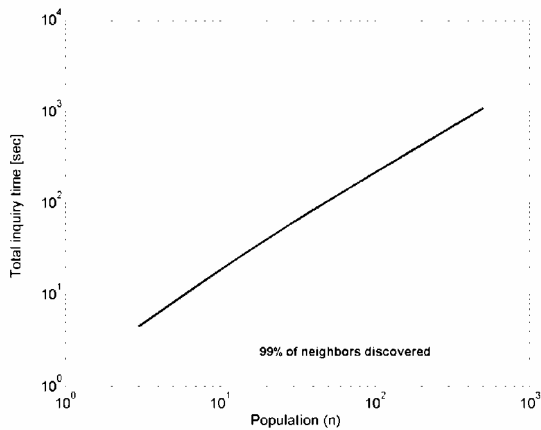


Figure 5. Total inquiry time versus population (99% of nodes discovered).

3. Comparison of Simulation and Analytical Results

In order to validate our analytical results we have created a object oriented, C++ discrete event simulation model closely following the specification of the Bluetooth inquiry. Simulation parameters (such as state holding times) were set to overlap with the parameters used in our analytical studies. For each simulation step, where the total inquiry time was sought for a given population, we have run enough simulation instances to be able to claim that we are 95% sure that our average results have less than 5% error (95-5 confidence interval).

As the reader may recall - in our first method we have selected a 50% confidence to be displayed corresponding to the median confidence that all nodes have been discovered. In our simulations we have used the average over all runs but observed that the difference between the average and the median was not significantly different from zero (except for higher population values). Thus, we can plot and compare all our results on a single figure. Figure 6 depicts the results of the previous analyses and the results of our simulation study with the solid line representing the latter. As we can observe the two separate efforts to model the Bluetooth inquiry lie close by thus supporting each other.

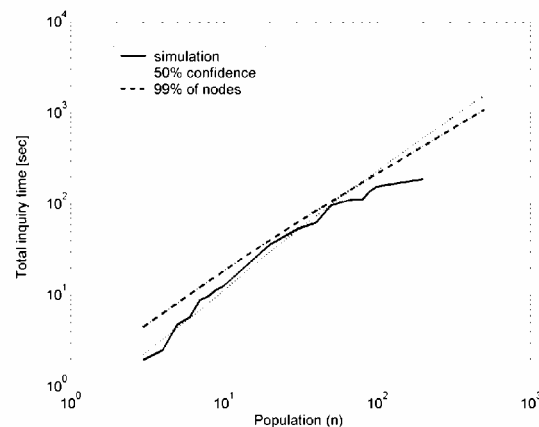


Figure 6. Simulation and analytical results for total inquiry time.

4. Conclusions and Future Work

Bluetooth is a promising wireless technology enabling devices to form short-range star-shaped wireless networks (or wireless personal area networks). Bluetooth relies on a frequency hopping physical layer, implying that hosts are not able to communicate unless they have previously discovered each other having synchronized their frequency hopping patterns. To support any-to-any communication with Bluetooth, personal area networks (or in Bluetooth terminology *piconets* and possibly *scatternets*) need to be established among nodes. Thus, it is of utmost importance to be able to model or predict the time required for nodes to discover each other, i.e., to synchronize them. In this paper we have provided with a mathematical analysis and corresponding simulation results for the discovery time in a fully connected (personal area network) situation. We have derived the total inquiry time for arbitrary populations using two different methods. In the first method we calculate the inquiry time required for a given (a parameter to the equation) confidence that all nodes have discovered each other. The second method calculates the time required for a given percentage for the entire population to discover each other. We note, that we required all nodes to change between the inquiry frequency trains immediately after a train is over.

We have also modeled the Bluetooth inquiry by a C++ discrete event simulation and obtained simulation results to compare with the outcomes of the mathematical analysis. Our simulation and analytical results have supported each other thus we conclude that our analytical models are viable and can be used for total inquiry time calculations.

Our future work is focused towards two major directions: i) on the inclusion of non-immediate train changes and ii) on the inclusion of multi-hop features where not all nodes will be required to be in each others' proximity. Our work will provide detailed insights to the feasibility studies on Bluetooth's viability to be used as an underlying ad hoc networking technology.

5. References

- [1] Bluetooth SIG, "*Bluetooth Baseband Specification*," version 1.1, www.bluetooth.com, 2000.
- [2] K. Fleming, "*Bluetooth Host Controller Interface Functional Specification*," version 1.1, www.bluetooth.com, 2000.
- [3] T. Salonidis, P. Bhagwat, L. Tassiulas, and R. LaMaire, "*Proximity Awareness and Ad Hoc Network Establishment in Bluetooth*," Institute of Systems Research (ISR), University of Maryland Technical Report, 2001.
- [4] G.V. Záruha, "*Accelerated Neighbor Discovery in Bluetooth Based Personal Area Networks*," Proceedings of the 2002 International Conference on Parallel and Distributed Processing Techniques and Applications (PDPTA'02), Las Vegas, NV, USA, June, 2002.
- [5] S. Basagni, R. Bruno, and C. Petrioli, "*Device Discovery in Bluetooth Networks: A Scatternet Perspective*," Proceedings of the IFIP-TC6 Networking Conference, Networking 2002, Pisa, Italy, May, 2002.
- [6] Leon-Garcia Alberto, "Probability and Random Processes for Electrical Engineering," 2nd Edition.