Bluetooth Frequency Hop Selection Kernel Impact on “Inter-Piconet” Interference

by

Jason Ballagh

Thesis submitted to the Faculty of the
Virginia Polytechnic Institute and State University
in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE
In
Electrical Engineering

Dr. Dennis Sweeney, Chair
Dr. Timothy Pratt
Dr. William Baumann

April 29th, 2003
Blacksburg, VA

Keywords: Bluetooth, Interference, Piconet, Synchronization, Inter-piconet,
Hop Selection Kernel

Bluetooth Frequency Hop Selection Kernel Impact on "Inter-Piconet" Interference

Jason Ballagh

(ABSTRACT)

The Bluetooth wireless transmission standard provides a low-power data link between electronic devices over relatively short ranges. These links, also known as piconets, transmit using frequency hopping spread spectrum (FHSS) to send information over the air. As more applications for Bluetooth technology become available, the number of Bluetooth devices will continue to increase. With this increase in use, there will be a decrease in performance that can be attributed to Bluetooth "inter-piconet" interference. To date, very little has been published on the subject of inter-piconet interference. Previous studies have derived mean packet error rates for an increase in the number of piconets present. To come up with the mean rate, many papers make the assumption that the probability of a Bluetooth device hopping to a channel is random. However, making this assumption does not explain what happens in real time.

This research gives some insight into what really happens when multiple piconets are interfering in real time. Bluetooth devices actually use a frequency hopping algorithm to determine the hopping sequence. This algorithm has been implemented in software to test various aspects of inter-piconet interference. Previous studies have shown that synchronizing the clocks among neighboring piconets will result in an increase in performance. This study shows that there are cases where synchronization alone will not provide sufficient improvement. Experimental testing has been conducted to validate some of the simulated results. Adjacent channel interference was observed during experimentation. This contradicts previous research, which has assumed that adjacent channel interference is insignificant.

Acknowledgements

I would like to start by thanking my advisor, Dr. Dennis Sweeney for his patience and continued support of this research. His useful insight, invaluable advice, and sense of humor have made working on this thesis enjoyable. I would like to thank the CWT for providing me with the resources necessary in conducting my research. And I would like to thank my committee members, Dr. Timothy Pratt and Dr. William Baumann, for reviewing this thesis work.

I would like to thank the students at the CWT for their support, especially Tom Rondeau. Without his dedicated support of the Bluetooth Lab, this thesis would have never come about. I would also like to thank him for taking time to exchange ideas with me about Bluetooth.

I would also like to recognize my family for their inspiration. Without them, I would not be writing this acknowledgement right now. My parents, Jim and Barbara Ballagh, have encouraged me throughout my life to set high goals for myself. Finally, I would like to thank my significant other, Angela. She provided me with motivation and support during times where I struggled.

Table of Contents

List of Figures

List of Tables

Chapter I: Introduction

1.1 Introduction

The Bluetooth wireless transmission standard provides a low-power data link between electronic devices over relatively short ranges. These links, also known as piconets, refer to a connection between a single master device and up to seven slave devices. As more applications for Bluetooth technology become available, the number of Bluetooth devices will continue to increase. With this increase in use, there will be a decrease in performance that can be attributed to Bluetooth piconets interfering with one another.

Bluetooth radios transmit using a frequency hopping spread spectrum (FHSS) technique. This technique forces transmissions to switch between 79 different frequency channels at 1600 hops per second. At any given time, the master and up to seven slaves within a piconet will hop together in frequency. A hopping sequence is generated by a "frequency hop selection kernel" that uses the address and clock from the master as inputs.   This sequence determines the "pseudo-random" hopping order at which transmissions occur ("pseudo" meaning that the hopping order repeats after some time). Keep in mind that the hop sequence for each piconet is "pseudo-random" and there are a limited number of frequency channels to hop to. Therefore, if two or more piconets are within range of one another, the frequencies at which they transmit may match up during some time slots. This overlap in frequency and time will cause packet collisions and loss of throughput in these piconets. This means that Bluetooth technology will actually interfere with itself. This interference will be referred to as "inter-piconet" interference.

The Bluetooth specification [1] already describes methods to help reduce the amount of Bluetooth interference. Some of these methods include changing the data packet length, adding forward error correction, and the use of adaptive power control. Adaptive power control keeps Bluetooth devices from radiating more power then necessary. This thesis will focus mostly on the Bluetooth frequency hop selection kernel. Various characteristics of this hopping algorithm could be exploited to help reduce inter-piconet interference.

1.2 Previous Research

So far, very little has been published regarding "inter-piconet" interference. Much of what has been published is strictly theoretical and represents only the mean worst-case and best-case scenarios. A few papers have been found where the frequency hopping algorithm was incorporated into their simulations. The subject matter of each of these papers will be summarized in this section. There will be an emphasis on the portions of these papers that pertain specifically to "inter-piconet" interference.

One of the earlier papers researched was titled "Frequency Lookahead and Link State History Based Interference Avoidance in Wireless Pico-cellular Networks." The paper proposed a method of looking at a channel before transmitting in order to avoid interference. With this method, a link state history (LSH) table would be used to keep track of the "bad" channels on a Bluetooth link. A Bluetooth piconet would use this LSH table to mask out any transmissions on channels having a high probability of error. This method is useful for avoiding interference coming from signals at fixed frequencies. A portion of this paper briefly mentions simulating the Bluetooth frequency hop selection kernel. However, there was no reference in this paper to avoiding interference from neighboring Bluetooth piconets [2].

Another paper that was researched was titled "Performance Evaluation of the Bluetooth-based Public Internet Access Point." The authors discuss the performance of an Internet access point consisting of one Bluetooth device connecting to multiple Bluetooth equipped notebook computers. The authors then model several scheduling policies to show the performance capabilities of such a Bluetooth link. These policies were simulated in order to compare the throughput and link delay associated with each. The paper also examines the use of multiple Bluetooth radios in an access point. When multiple Bluetooth piconets are used to serve a single access point, then Bluetooth interference starts to appear. There is also some throughput analysis based on the number of piconets present. It should be noted that this analysis was based on the assumption that the probability of hopping to a channel was 1/79. The aggregate throughput was plotted for an increase in the number of users. One of the conclusions was that the maximum aggregate throughput occurs with 40 piconets present. This result was based on the assumption that each transmission occupied a single time slot and that the channels were aligned with each other in time. This alignment in time can be visualized in Figure 1.2.1 by comparing the packets in the top row to the packets in row (b). Notice that the time slots are aligned. This paper also discusses what happens when these channels become misaligned in time. The worst-case scenario can be seen in Figure 1.2.1 in row (c). With such a misalignment, there are now two packets in the same time slot as the "data" packet on the top row. Either of these two packets now has the potential to collide with the "data" packet. Keep in mind that an actual collision would also require one of these two packets to share the same frequency as the "data" packet. With this time misalignment, the probability of a collision occurring has been doubled [3].

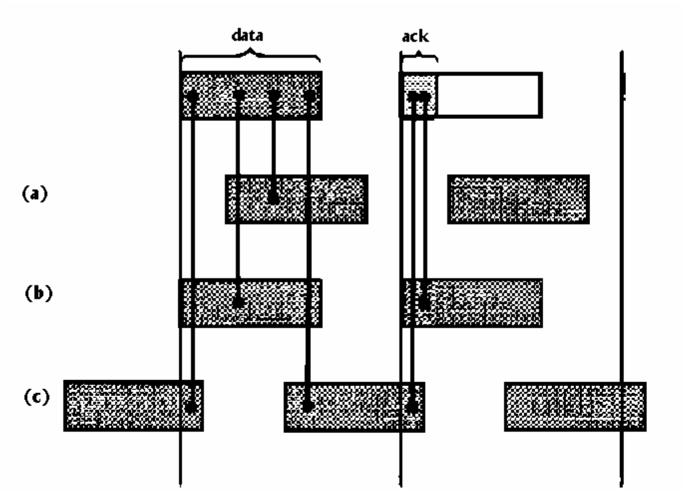


Figure 1.2.1 Different alignments of time slots between two Bluetooth piconets (© 2001 IEEE) [3]

The next two papers, "Packet Error Rate due to Interference between Bluetooth Networks – Probabilistic Upper Bound and Simulation Results" and "Interference Between Bluetooth Networks – Bound on the Packet Error Rate," will be discussed together because the same author wrote them and they are almost identical in content. These papers provide a probabilistic study of the upper and lower bounds of mean packet error rate for a Bluetooth device in the presence of multiple Bluetooth piconets.

Theoretical statistics for the upper bound were developed using the best-case scenario for time slot alignment as shown in Figure 1.2.1 (b). With this alignment, only one packet from an interfering piconet dwells in an intended time slot. This means that there is only a single threat from an interfering piconet on each intended packet. Statistics for the lower bound were developed using the worst-case scenario for time slot alignment as shown in Figure 1.2.1 (c). In this case of misalignment, recall that there are now portions of two packets from an interfering piconet dwelling in each intended time slot. This means that there is now a double threat from an interfering piconet and the probability of a collision occurring is greatly increased. The theoretical results were then validated through simulation in OPNET. It is important to note that in this study, the author also assumed that the probability of hopping to a channel was 1 in 79. The author of this paper concluded that synchronizing the time slots of neighboring Bluetooth piconets could be used as a method to reduce interference. This type of synchronization would be similar to that of slotted ALOHA [5, 6].

Another paper was titled "Piconet Interference Modeling and Performance Evaluation of Bluetooth MAC Protocol." The paper covers a study of the performance of the medium access control (MAC) protocol performance in the presence of interference from multiple Bluetooth piconets. A complex mathematical model was developed and then validated using the Network Simulator-2 software package. The study pays particular attention to the performance of different layers of the Bluetooth protocol architecture [4].

The last two papers that were researched, gave some insight into the behavior of the frequency hopping algorithm. The first of these papers was titled "Scenario Driven Evaluation and Interference Mitigation Proposals for Bluetooth and High Data Rate Bluetooth Enabled Consumer Electronic Devices." The paper describes the development of an indoor propagation model used to test the effect of inter-piconet interference on throughput. The previous studies described had derived their results from the assumption that the probability of hopping to a channel was 1 in 79. However, the model developed in this study used the Bluetooth frequency hop selection kernel to determine the frequency channel of each packet during a given time slot. Collision statistics were simulated by averaging the results of several different collision tests. Each test was run with piconets producing a different hop pattern. This hop pattern was varied between tests by changing the master address and clock inputs to the frequency hopping algorithm. By showing only the average results, the study did not give any insight as to how the collision statistics behave over time. The paper used the model to compare the throughput performance when using 3 different PSK modulation schemes [9].

The final paper that was researched is titled "Frequency Hop Selection in the Bluetooth Radio System." The paper describes the Bluetooth frequency hop selection kernel and also shows some analysis of its performance. The paper was discovered while this thesis study was well under way. It gave some helpful insight into the behavior of the Bluetooth frequency hop selection kernel. In particular, the paper did some analysis on the cross-correlation properties of two Bluetooth hopping sequences. Some of the observations made by the author validated some of the results in this research. One of these observations was that the cross-correlation of two hopping sequences is periodic as the two sequences are shifted apart in time [8].

1.3 Research Goals

The main objective of this thesis is to gain a better understanding of inter-piconet interference. The first goal of this thesis is to develop some background information necessary in understanding inter-piconet interference. This information will be covered in Chapter 2 of this thesis.

Chapter 3 will cover the second goal of this thesis, which is to use a software simulation to explore the frequency hop selection kernel in greater detail. The first step in doing this will be to implement the frequency hop selection kernel in software. The software kernel will then be used to develop a much larger simulation. The simulation will be capable of performing various interference tests. These tests will be performed to gather an upper-limit of collision rate under several different scenarios. All simulations will maintain the basic assumption that all nodes of piconets being tested are within interfering range of one another. The other assumptions are that the Bluetooth radios are transmitting at

100% capacity and that they are transmitting at a power sufficient to cause co-channel interference yet low enough to avoid adjacent channel interference. Co-channel interference refers to interference caused by unintended transmission on the same frequency channel. Adjacent channel interference refers to interference caused by unintended transmission in a neighboring frequency channel. In summary, all interference variables will be fixed so that the focus is on the behavior of the frequency hop selection kernel itself. The collision rate statistics will be analyzed as a function of clock time offset between piconets. These statistics will also be analyzed as multiple interfering piconets hop along in time.

This thesis plans to show how beneficial "synchronization" is between piconets. All previous studies show that "synchronization" is beneficial when the mean statistics are being analyzed. However, this thesis will show that there are special cases where synchronization of time slots alone will not improve throughput.

Chapter 4 will cover the third goal of this thesis, which is to verify some simulation results through experimentation with real hardware. The plan is to conduct several experiments using a Tektronix Protocol Analyzer to measure interference on a per-packet basis. A method will be introduced to resolve the inputs to the frequency hopping algorithm used during these experiments. With these inputs, each individual experiment can be recreated using the computer simulation. Hopefully, the simulated experimental results match up in a way that will give further insight into "inter-piconet" interference.

Chapter 5 will cover the final goal of this thesis. That goal is to develop conclusions as well as give some description of future work that could result from this research.

Chapter II: Background Information

2.1 Introduction

This chapter of the thesis will detail some of the background information necessary to understand the issues involved with "inter-piconet" interference. Section 2.2 will review the Bluetooth hopping sequence. Section 2.3 will cover the different types of packet transmissions that were studied. Section 2.4 will show how multiple piconets can interfere with one another. Finally, in Section 2.5, theoretical collision and throughput statistics due to "inter-piconet" interference will be derived.

2.2 Bluetooth Hopping Sequence

To further explain "inter-piconet" interference, the hopping sequence must be viewed in greater detail. The channel on which Bluetooth transmits at any point in time is determined by the master's address and clock. The address and clock values are fed into a "frequency hop selection kernel" in order to select the appropriate channel for transmitting and receiving. The address is used to determine the exact hop sequence to be used, while the clock simply determines the phase in that sequence [1]. The master's address and clock are transmitted to slaves during an Inquiry process. During this process, the slave calculates the clock offset between its clock and that of the master. With this offset and the master's address, the slave is able to synchronize its hop sequence with that of the master.

Figure 2.2.1 below shows a block diagram of how the frequency hop selection kernel operates. The values for the inputs to this block diagram can also be found below in Table 2.2.1. Clock bit 0 is not used in this operation for the reason that each time slot covers two clock cycles. However, Clock bit 0 is used when determining the clock offset between the master and slave, allowing two or more devices to synchronize. Clock bit 1 (used in inputs Y1 and Y2) is used to alternate between transmit and receive modes every two clock cycles. This means that each time slot is two cycles long or 625 μs.

The permutation operation shown in the block diagram uses Y1, C, and D as controls for a butterfly operation. This butterfly operation interleaves the input bits calculated from X, A, and B. The output of the final addition block is applied to a modulo 79 operation. The output of this operation is a control word used to select the frequency channel from a register. The frequency channels in the register are ordered in such a way that the first 40 code words select the even channels and the last 39 code words select the odd channels. For example code words 1, 2, and 3 would correspond to frequency channels 0, 2 and 4. Likewise, code words 41, 42, 43 and so on correspond to frequency channels 1, 3, 5, etc. This was done to ensure that adequate spreading could be achieved over a short time interval. To further understand this, try to imagine what happens in the algorithm between frequency hops. Clock bit 1 is the first bit that will cause the phase of the hopping sequence to progress. An offset of 32 x "Clock bit 1" (Y2 in Figure 2.2.1) is added at the output of the algorithm. As Clock bit 1 changes, an offset of 32 is either added or removed from the output. The 5-bit output of the permutation operation is the only other variable and has a maximum decimal value of 32. The input "E" into the final adder is fixed as it is dependent only on the master's address. So, the addition of an offset of 32 every other time slot, in addition to the arrangement of channels in the register helps the sequence to spread transmission over 80% of the 79 MHz band over very short time intervals.

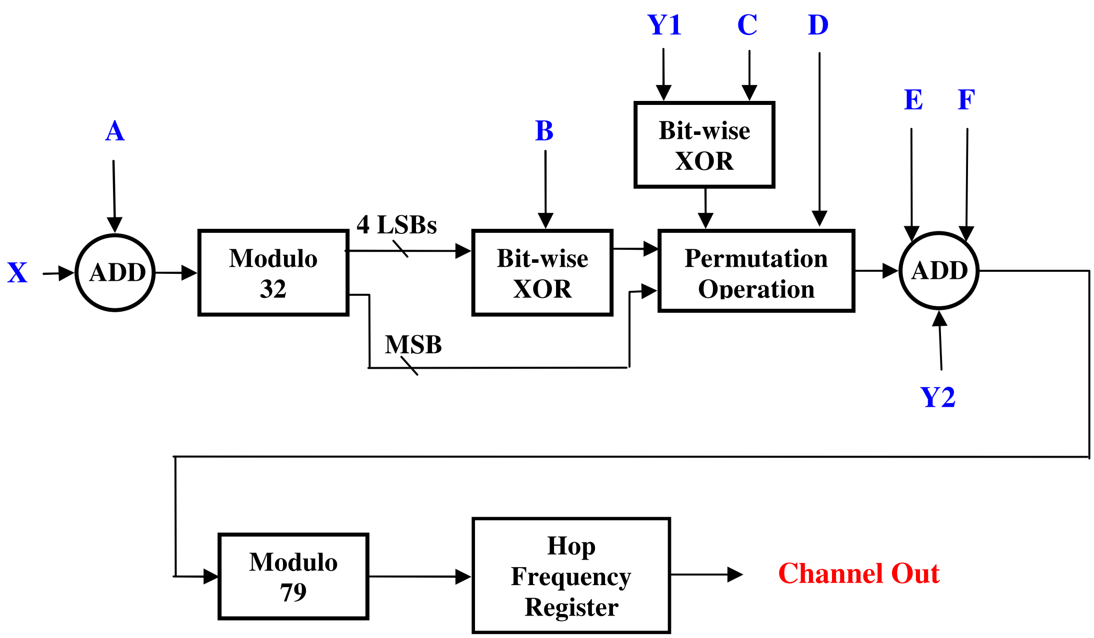


Figure 2.2.1 Hop Selection Kernel as defined in the Bluetooth Specification

Input Label	Value In
A	Address bits 27-23 XOR Clock bits 25-21
B	Address bits 22-19
C	Address bits (8,6,4,2,0) XOR Clock bits 20-16
D	Address bits 18-10 XOR Clock bits 15-7
E	Address bits (13,11,9,7,5,3,1)
F	(16 x Clock bits 27-7) mod 79
X	Clock bits 6-2
Y1	Clock bit 1
Y2	32 x Clock bit 1

Table 2.2.1 Inputs to Hop Selection Kernel as defined in the Bluetooth Specification

The frequency hop selection kernel specified above does not operate in a true random manner. In fact, the kernel operates by hopping sequentially through 32-hop segments. Clock bits 6-2 (Input X) determine the hop position within each of these 32-hop segments. As a 32-hop segment ends, Clock bit 7 will increment and bits 6-2 will reset to zero, marking the beginning of the next segment. During each particular 32-hop segment, no frequencies will be repeated. This means that over short intervals of time, the probability of hopping to any of the 79 channels is not truly random in nature. This also means that the probability of two piconets colliding will not be random over short intervals of time either.

2.3 Packet Transmissions

There are several different types of links that can be defined as Bluetooth data transmissions. This section will cover the main concepts necessary in understanding the behavior of the links analyzed in this thesis. This section is separated into three subsections. Section 2.3.1 will briefly describe some of the different types of packets. Section 2.3.2 explains the ARQ scheme employed in some Bluetooth transmissions. And finally, Section 2.3.3 defines the difference between symmetric and asymmetric transmissions.

2.3.1 Types of Packets

There are three main types of packets: Link control packets, Asynchronous Connection-Less (ACL) data packets, and Synchronous Connection-Oriented (SCO) voice packets. Chapter 3 of this thesis will mainly study links using link control and ACL data packets. The types of ACL packets analyzed in this thesis employ an ARQ scheme. Chapter 5 of this thesis focuses on hardware experimentation performed using SCO voice packets. Due to the inability of the current hardware to transmit ACL data packets at 100% capacity, experimentation was instead performed with the use of SCO voice packets. The following packets will be analyzed throughout this thesis.

NULL This packet is used for Bluetooth link control. During data transmission, NULL packets are used to acknowledge receipt of a transmission when no data is available to send in response. These packets have a length of 126 bits.

DM# DM packets are data packets with Forward Error Correction (FEC) codes. Specifically a rate 2/3 FEC code is used in order to improve BER performance. DM packets can be 1, 3 or 5 time slots long, with each time slot spanning 625 μs. A longer time slot will provide a larger payload and higher data throughput. These packets also employ a Cyclic Redundancy Check (CRC) to detect errors in the payload.

DH# DH packets are very similar in nature to DM packets. They are also data packets. However, DH packets do not use FEC in exchange for an increase in payload size. These packets can also be 1, 3 or 5 time slots in length. Like DM packets, DH data packets also employ CRC.

HV# HV packets are voice packets that span 1 time slot in length. Unlike data packets, voice packets do not use CRC to check for payload errors. There are three different types of HV packets, each utilizing a different level of FEC. HV1 packets carry a payload size of 10 bytes and use rate 1/3 FEC. HV2 packets carry 20 bytes in each payload and use rate 2/3 FEC. HV3 packets carry 30 bytes of payload data and use no FEC.

2.3.2 ARQ Scheme

The ACL links which are studied within this thesis use a cyclic redundancy check (CRC) code in order to determine if the payload in the received packet is correct or not. If the payload in the received packet is correct, an acknowledgement (ACK) is returned to the device that sent the packet. This ACK can be sent in the header of the next return packet. The return packet can be either a data packet or a NULL packet that has no payload. If no ACK is received, then the transmission will be resent.

Using the ARQ scheme, every transmission must be accompanied by the receipt of an ACK in order for transmission to be successful. Therefore, a loss of either the initial transmission or the successive ACK will result in a complete retransmission.

2.3.3 Symmetric/Asymmetric Transmission

Bluetooth ACL transmissions can occur with a wide variety of packet combinations. The two main classes of links are known as symmetric and asymmetric. A symmetric link occurs when both the master and slave in a piconet are transmitting the same sized packet. An example of a symmetric link is shown below in Figure 2.3.1. Notice that the figure does not show the packets filling up the entire time slot. This is because there is some settling time required after the switching of the synthesizer. This time is specified to be a minimum of 224.5 μs [1]. Although not drawn to scale, the gap between packet transmissions in Figures 2.3.1 and 2.3.2 represents this synthesizer settling time.

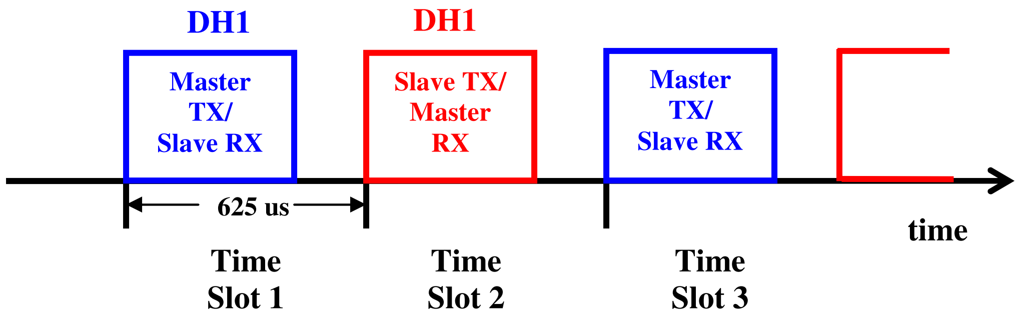


Figure 2.3.1 Graphical example of Symmetric Transmission

An asymmetric link occurs when the master sends one size packet and receives a different size packet as a response from the slave. For example, a DH1 (1-slot) packet could be transmitted from the master with the slave returning DH5 (5-slot) packets. This example of an asymmetric link is shown below in Figure 2.3.2.

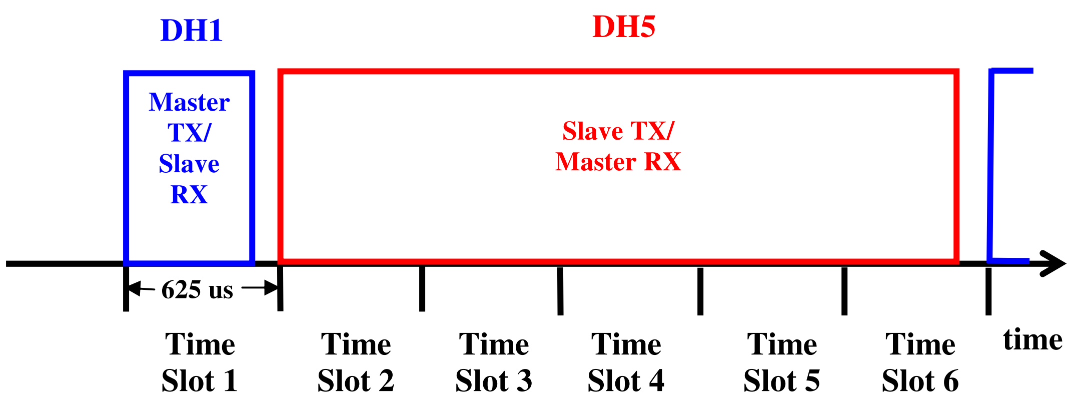


Figure 2.3.2 Graphical example of Asymmetric Transmission

The different types of packets that will be analyzed are tabulated below in Table 2.3.1.

Packet Type	Payload Header (bytes)	User Payload (bytes)	FEC rate	CRC	Symmetric Max. Rate (kb/s)	Assymetric Max. Rate (kb/s)	
						Forward	Reverse
DM1	1	0-17	2/3	yes	108.8	108.8	108.8
DH1	1	0-27	no	yes	172.8	172.8	172.8
DM3	2	0-121	2/3	yes	258.1	387.2	54.4
DH3	2	0-183	no	yes	390.4	585.6	86.4
DM5	2	0-224	2/3	yes	286.7	477.8	36.3
DH5	2	0-339	no	yes	433.9	723.2	57.6
HV1	na	10	1/3	no	64	na	na
HV2	na	20	2/3	no	64	na	na
HV3	na	30	no	no	64	na	na
NULL	na	na	na	na	na	na	na

Table 2.3.1 Characteristics of some Bluetooth packets [1]

2.4 Introduction to "Inter-piconet" Interference

In order to describe inter-piconet interference, a few terms must be identified. This section has been divided into three subsections. Section 2.4.1 will briefly define the term "piconet of interest". Section 2.4.2 will identify what an "interfering piconet" is and how it occurs. And finally, Section 2.4.3 will explain what the difference is between "synchronized" and "unsynchronized" piconets. The three ideas covered here in Section 2.4 of the thesis are fundamental in understanding inter-piconet interference.

2.4.1 Piconet of Interest

In order to study the interference caused from multiple Bluetooth piconet transmissions, a point of reference must be taken. The piconet used as a point of reference will always be referred to as the "piconet of interest" throughout this thesis. Other piconets will be introduced to the piconet of interest's environment as interferers. The piconet of interest is the piconet that is subject to interference. The actual amount of interference will be measured by examining the decrease in performance from the piconet of interest.

2.4.2 Interfering Piconet

Any piconet that transmits within range of the "piconet of interest" will be referred to as an "interfering piconet." The interfering piconet is named so for the simple reason that it is now a threat to interfere with the piconet of interest. Interference, however, is not guaranteed by just the interferer's presence. Interference can only occur when the piconet of interest and at least one interfering piconet hop to the same frequency during the same time slot. In order for this to occur, both piconets must be in the process of transmitting and receiving. Interference is also dependent on the power levels transmitted by the piconets involved. The requirement in the Bluetooth specification for avoiding co-channel interference is that the carrier to interferer (C/I) ratio be a minimum of 11 dB. If the C/I falls below 0 dB (specification for $C/I_{adjacent\ 1\ MHz}$), then adjacent channel interference may occur as well [1].

Figure 2.4.1 below shows an example of two piconets interfering with one another. Note that "Packet 3" in the diagram is on channel 78 during the same time slot for both piconets. These two packets will collide with each other, resulting in a retransmission of both packets.

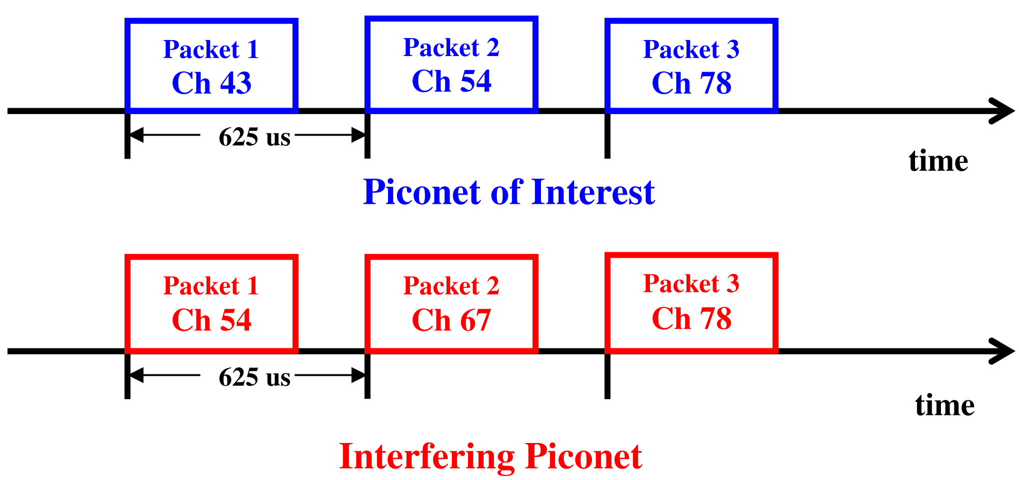


Figure 2.4.1 Example of inter-piconet interference. Note that packet 3 will interfere.

2.4.3 "Synchronized" and "Unsynchronized" Piconets

The term "synchronized" often refers to things that are matched up in some fashion. In this thesis, the term "synchronized" will refer to two piconets that are aligned in such a way that their time slots are matched up within a few 10's of microseconds. In real-life Bluetooth data transmissions, separate piconets will not be intentionally synchronized to one-another. Therefore, it is important to understand what occurs when packets from different piconets are not aligned in the time domain. Figure 2.4.1 above shows a graphical representation of how the packets might look in the time domain if their time slots were synchronized. Based upon this representation, for there to be a collision, packets from both piconets must hop to the same frequency during the same time slot. It is possible that the time slots between piconets can be slightly misaligned, keeping the collision statistics the same. This slight misalignment would allow for the two piconets to still be referred to as "synchronized" piconets.

If the two piconets instead have unsynchronized time slots, the potential for interference will increase. Figure 2.4.2 below shows a graphical representation of how packets from Figure 2.4.1 might look in the time domain if their time slots were "unsynchronized." In Figure 2.4.2, there is a time misalignment. In Figure 2.4.1, only "packet 3" from the two piconets collided. Now due to the misalignment, "packet 2" from the piconet of interest and "packet 1" from the interfering piconet are now also both on the same channel (Channel 54) at the same time. "Unsynchronization" causes this additional interferer that would not be present if the two piconets were synchronized as they were in Figure 2.4.1.

In the "unsynchronized" case, each packet from the interfering piconet is a "double threat" to interfere with the piconet of interest. These packets are now twice as likely to cause interference because they each overlap in time two packets from the piconet of interest. Keep in mind that in order for interference to occur, packets from two piconets must occur on the same frequency channel at the same time.

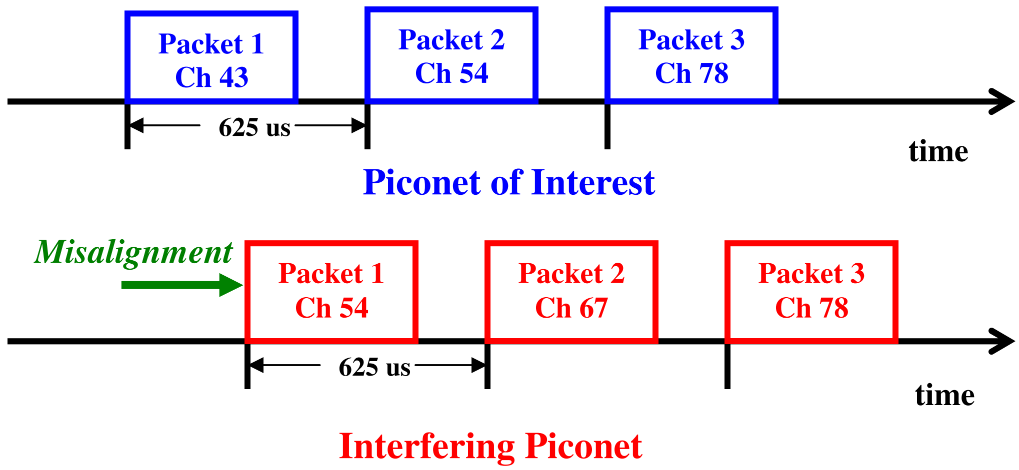


Figure 2.4.2 Piconets with "unsynchronized" time slots

2.5 Theoretical Collision and Throughput Statistics

Most of the previous research conducted up to this point on Bluetooth "inter-piconet" interference has been theoretical in nature. In a real-life Bluetooth piconet, the frequency hop selection kernel described in Section 2.2 determines the probability of interference. In theoretical studies, the probability of hopping to a particular channel is assumed to be equally likely for all 79 channels. This means that at any given time, the assumption is made that the probability of hopping to a given channel is 1 in 79. This is one of the assumptions made in previous studies to develop mean collision and throughput statistics in a multiple Bluetooth piconet environment. This section of the thesis has been divided into three subsections. Section 2.5.1 will review the assumptions that were made in this theoretical study. Section 2.5.2 will show the derivation of an equation for packet collision probability. And finally, Section 2.5.3 will go through the derivation of an equation for piconet throughput.

2.5.1 Assumptions

A few assumptions must be made in order to derive the equations for Bluetooth interference statistics. First, all Bluetooth devices are assumed within range of one another. This means that the only criteria for a collision to occur will be that multiple piconets dwell on the same frequency channel during the same time slot. This assumption is physically possible, but it assumes that the power levels received by the piconet of interest are within a C/I window of 0 and 11 dB. This is so that the interferer power is high enough to cause interference, yet low enough to avoid causing adjacent channel interference.

The second assumption is that the probability of hopping to a frequency channel at any point in time is assumed to be 1/79. This assumption is unrealistic over a finite time interval. This is due to the fact that the hopping sequence is not random in nature. In real hardware, the Bluetooth frequency hop selection kernel determines the hopping sequence.

The third assumption is that all transmissions are symmetric. This is possible in real Bluetooth transmissions, but it requires that constant streams of data are needed in both directions of the link. Voice transmissions rely on this type of transmission in order to talk and listen simultaneously. However, many applications require asymmetric transmission where the flow of data is heavier in one direction than the other. The assumption made here does not take into account these applications where asymmetric transmission is used.

And finally, all piconets are assumed to be transmitting at 100% capacity. This assumption is also possible, but it requires information to be sent at all times. When voice transmissions are active, this assumption is very likely. However, there are many applications for Bluetooth where data is sent intermittently.

All of the assumptions made here are somewhat unrealistic. All are possible under specific circumstances, with the exception of the assumption that the probability of hopping to a channel is 1 in 79. Although the assumptions are unrealistic, they have been chosen to simplify the analysis of "inter-piconet" interference.

2.5.2 Deriving Collision Probability

This subsection will go through the derivation of the theoretical calculation for packet collision rate. Multiple collisions against a packet in each particular time slot will only be counted as one. This way, the number of collisions will be representative of the number of packets corrupted in the "piconet of interest."

At first, the collision rate for two synchronized piconets ("piconet of interest" plus one interferer) will be analyzed. Time must be fixed in order to analyze the statistics involved. Imagine that the "piconet of interest" has already switched to its frequency channel. Then, the probability that the "interfering piconet" will hop to the same channel is 1/79. Therefore, the probability of two piconets' packets colliding is 1/79.

Next, the collision rate for multiple synchronized piconets ("piconet of interest" plus *n* interferers) will be analyzed. The key to deriving this equation is to recall that multiple collisions against a packet in one time slot must be counted as one collision. In this derivation, the packet from the "piconet of interest" will be referred to as "A." The packets from the "interfering piconets" will be referred to as B, C, D, etc. A collision will occurs in a time slot if (A channel = B channel) or (A channel = C channel) and so on. The equation for theoretical packet collision rate is derived using this statement. The probability of synchronized piconets colliding during each packet transmission is

$$Pcollision_{sync} = 1 - (1 - P_1)^n \qquad \{2.1)$$

where P_1=(1/79) is the probability of hopping to any of the 79 Bluetooth channels, and *n* is the number of interfering piconets [5].

The equation for "unsynchronized" piconets will now be derived. When piconets become unsynchronized, the possibility for interference will change. Looking back at Figure 2.4.2, there are two bordering time slots that are now a threat to interfere with each packet in the "piconet of interest." Equation 2.1 has to be modified to consider potential interferers from both threatening time slots. The equation used to calculate the theoretical packet collision rate for unsynchronized piconets becomes

$$Pcollision_{unsync} = 1 - (1 - P_1)^{2n} \qquad (2.2)$$

where P_1=(1/79) is the probability of hopping to any of the 79 Bluetooth channels, and n is the number of interfering piconets [5]. A plot of the packet collision rate versus the number of interferers can be seen below in Figure 2.5.1.

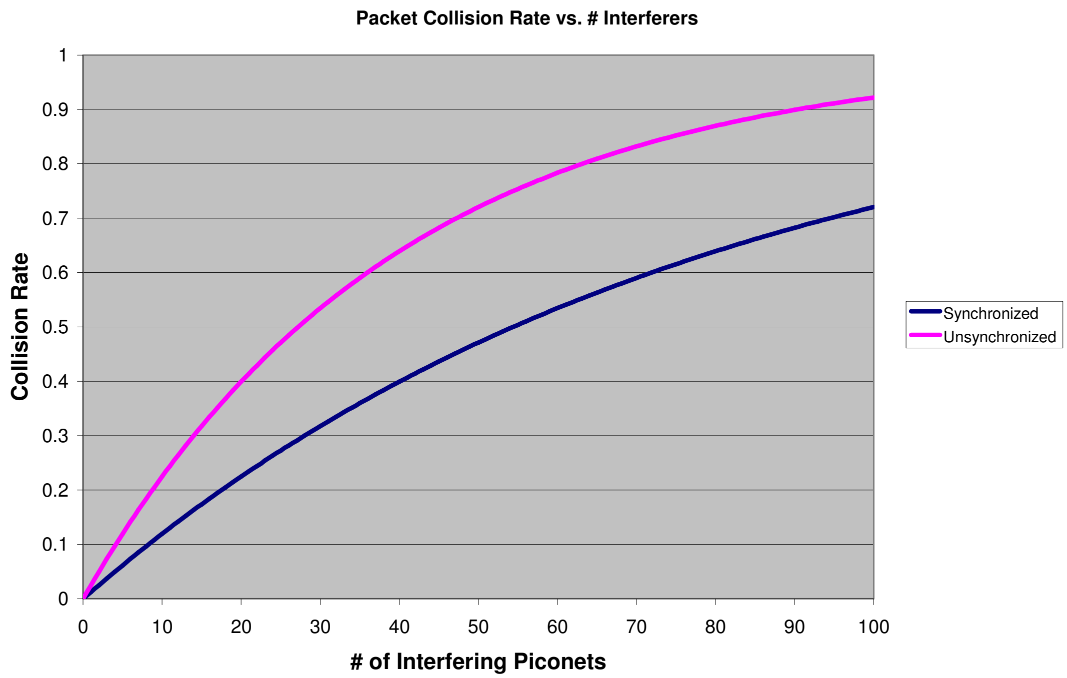


Figure 2.5.1 Packet collision rate versus number of interferers, *n*.

2.5.3 Deriving Piconet Throughput

In order for packets to be received successfully, the transmitted packet as well as the acknowledgment packet must not suffer from a collision. If a collision were to occur on either of these packets, then a complete retransmission will be required. Therefore, the hit on throughput is going to be higher than the packet collision rate specified above. In fact, the hit on throughput is simply the probability that one in every pair of consecutive packets is in error. The throughput probability for multiple "synchronized" piconets is

$$Pthroughput_{sync} = (1 - P_1)^{2n} \qquad (2.3)$$

where P_1=(1/79) is the probability of hopping to any of the 79 Bluetooth channels, and *n* is the number of interfering piconets [5]. The throughput probability for multiple "unsynchronized" piconets is

$$Pthroughput_{unsync} = (1 - P_1)^{4n} \tag{2.4}$$

where P_1=(1/79) is the probability of hopping to any of the 79 Bluetooth channels, and *n* is the number of interfering piconets [5]. A plot showing throughput versus the number of interferers can be seen below in Figure 2.5.2.

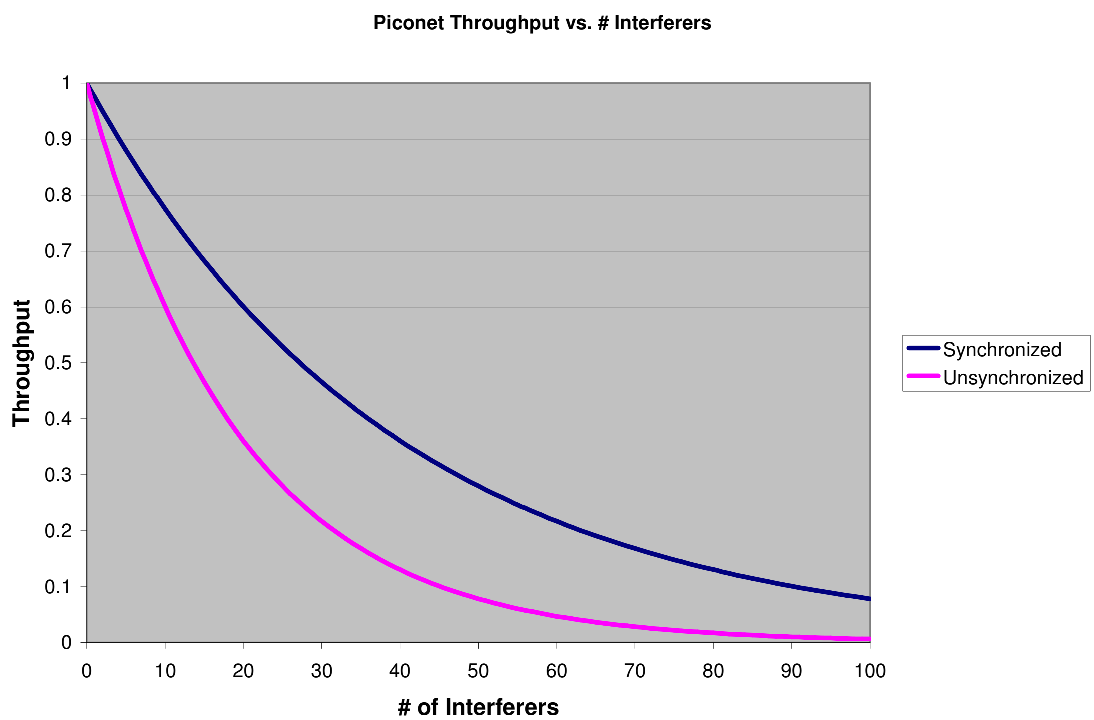


Figure 2.5.2 Piconet throughput versus number of interferers, n.

Figure 2.5.2 shows how the throughput of a piconet will decrease as the number of interfering piconets increase. Likewise, Figure 2.5.1 showed how collisions increased as the number of interfering piconets went up. Also, both figures show an improvement in performance when the piconets are synchronized.

Chapter III: Inter-piconet Interference Simulations

3.1 Introduction

In Section 2.5 of this thesis, theoretical statistics for collision and throughput rates were examined. Some theoretical equations were derived under the assumption that there is a 1 in 79 chance that a Bluetooth piconet will hop to a specific frequency. However, this is not the case at any given instant in time. In fact, Bluetooth technology uses the channel hop selection algorithm as specified in Section 2.2. This algorithm causes the radio to hop once through each channel in a 32-hop segment sequentially before switching to the next 32-hop segment. No channel is repeated during each 32-hop segment. Therefore, over a short period of time, the probability of hopping to any one of the 79 channels will not be 1 in 79. For example, the probability is 1 in 32 for the given 32-hop segment and zero elsewhere [1].

This chapter of the thesis covers the development of a simulation that uses the "frequency hop selection kernel" to test the effects of "inter-piconet' interference. Section 3.2 of this chapter will focus on the development of the simulation. Several sections have also been included in this chapter to describe the results of several different simulations. Section 3.3 of this chapter will show how collision statistics vary over time and as the time slot offset between piconets is increased. Section 3.4 will analyze the statistical distribution of interference data. And finally, Section 3.5 will show the effect of different types of interfering packets on a particular transmission as the number of interfering piconets is increased.

3.2 Development of Simulation

This section of the thesis focuses on the development of an inter-piconet interference simulation. The initial simulation was developed to analyze how 1-slot transmissions in multiple piconets would interfere with one another in time. There are three subsections covering the development stages and results of this initial simulation. Section 3.2.1 covers the implementation and verification of the frequency hop selection kernel in software. Section 3.2.2 will show how the hop selection kernel was used to develop a simulation to test for collisions among two piconets. Results from this simulation will be verified by comparison to the operation of real Bluetooth hardware. And finally, Section 3.2.3 will involve an analysis of the simulation results.

3.2.1 Software Implementation of Frequency Hop Selection Kernel

The first step in developing the "inter-piconet interference" simulation was to implement the frequency hop selection kernel in software. This was done previously by Mark D'Souza in his study on "Microwave Oven Interference on Bluetooth Transmissions"[7]. He developed a program in MATLAB to generate a Bluetooth hopping sequence given any Bluetooth address and clock start time. I have modified this code into a subroutine in C++ to generate a single hop frequency given the Bluetooth address and a single clock time. The subroutine, *fh_channel*, is represented in C++ as

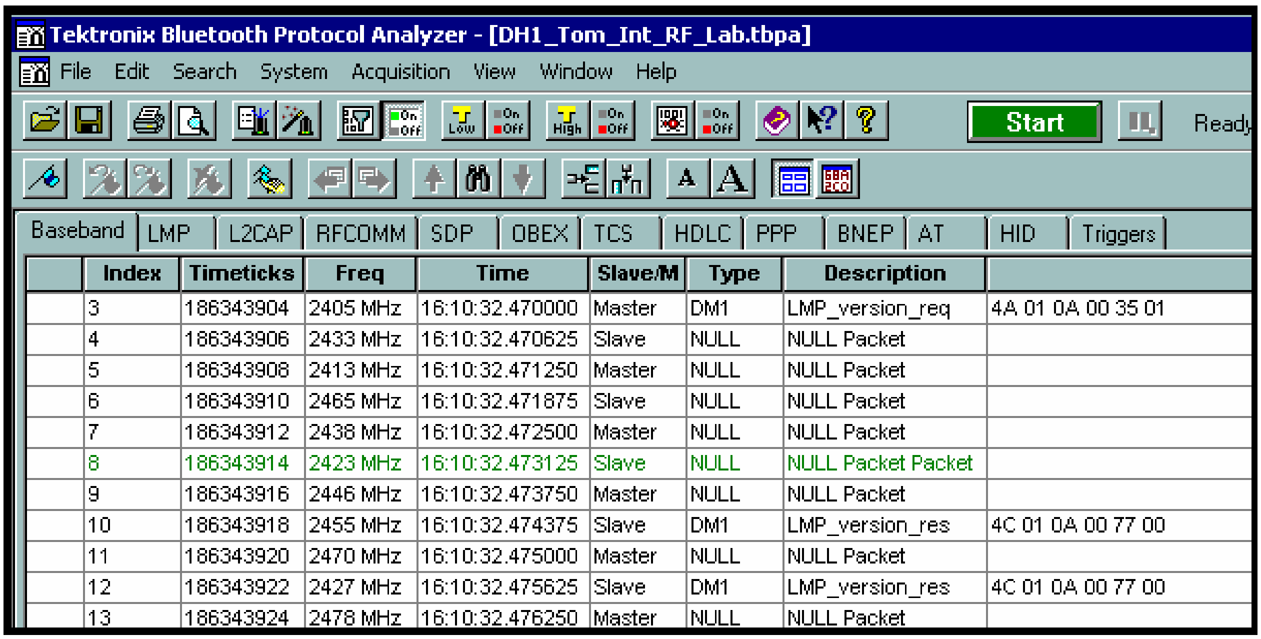
fh_channel(address,clock,index_array)

This subroutine outputs the hopping frequency corresponding to the two inputs "address" and "clock." The "address" and "clock" inputs correspond to the address and clock of the master in a piconet. In order to develop a sequence of frequency hops, the subroutine *fh_channel* must be repeatedly called as the clock is incremented. Recall that in Section 2.2 of this thesis it was mentioned that each time slot is 2 clock cycles long. Therefore, when transmitting 1-slot packets, the hopping frequency will change every 2 clock cycles. The following loop is used to generate a Bluetooth hop sequence

```
for (int j=0;j<2000;j++){
freq = fh_channel(address,clock,index_array)
a_file<<clock<<freq<<endl;        //Outputs each line of hop sequence to file
clock=clock+2;                    //Increment to next time slot
}
```

Keep in mind that before this loop is used, the master "address" and initial "clock" time must be defined in the code. This fragment of code will generate a hop sequence spanning 2000 time slots as specified in the *for* loop descriptor.

After developing the C++ code to generate the hopping sequence, it was necessary to verify its operation against actual Bluetooth hardware. This was done by first setting up two Bluetooth devices to operate as a single piconet. Once this was set up, a Tektronix "Bluetooth Protocol Analyzer" was used to capture all packets transmitted within the piconet. An example of the computer output from the Protocol Analyzer is shown below in Figure 3.2.1.



	Index	Timeticks	Freq	Time	Slave/M	Type	Description	
	3	186343904	2405 MHz	16:10:32.470000	Master	DM1	LMP_version_req	4A 01 0A 00 35 01
	4	186343906	2433 MHz	16:10:32.470625	Slave	NULL	NULL Packet	
	5	186343908	2413 MHz	16:10:32.471250	Master	NULL	NULL Packet	
	6	186343910	2465 MHz	16:10:32.471875	Slave	NULL	NULL Packet	
	7	186343912	2438 MHz	16:10:32.472500	Master	NULL	NULL Packet	
	8	186343914	2423 MHz	16:10:32.473125	Slave	NULL	NULL Packet Packet	
	9	186343916	2446 MHz	16:10:32.473750	Master	NULL	NULL Packet	
	10	186343918	2455 MHz	16:10:32.474375	Slave	DM1	LMP_version_res	4C 01 0A 00 77 00
	11	186343920	2470 MHz	16:10:32.475000	Master	NULL	NULL Packet	
	12	186343922	2427 MHz	16:10:32.475625	Slave	DM1	LMP_version_res	4C 01 0A 00 77 00
	13	186343924	2478 MHz	16:10:32.476250	Master	NULL	NULL Packet	

Figure 3.2.1 Sample output from Tektronix Bluetooth Protocol Analyzer

Notice in the Protocol Analyzer output that there are columns labeled "Timeticks" and "Freq." These two columns were important in verifying the operation of the C++ program. After recording the "real" Bluetooth data, it was then necessary to attempt to duplicate this data in software. The first "Timetick" recorded by the Protocol Analyzer was used as the initial "clock" time in the program. The same master address was also used as an input to the program. In this particular case, the master address was 5B001913 in hexadecimal. After entering the correct inputs, a 2,000 hop sequence was generated using the code. A sample output from the C++ program is shown below in Figure 3.2.2.

For comparison purposes, the same range of timeticks as in Figure 3.2.1 has been shown. Over this small sample it is easily seen that the frequency hop selection kernel in C++ functions exactly the same as real Bluetooth hardware.

Index	Timeticks	Freq (MHz)
3	**186343904**	**2405**
4	**186343906**	**2433**
5	**186343908**	**2413**
6	**186343910**	**2465**
7	**186343912**	**2438**
8	**186343914**	**2423**
9	**186343916**	**2446**
10	**186343918**	**2455**
11	**186343920**	**2470**
12	**186343922**	**2427**
13	**186343924**	**2478**

Figure 3.2.2 Sample output from Frequency Hop Sequence C++ Program

To further verify the preciseness of the Frequency Hop Sequence Program, it was necessary to compare much larger samples. This was done by comparing samples of 40,000 time slots. Using a Microsoft Excel spreadsheet, I was able to verify that the experimental sample and that generated in C++ were 100% identical.

3.2.2 Simulating Collisions between Piconets

The second step in developing an “inter-piconet” interference simulation was to test for collisions between two piconets. Before writing the program to test for this, a few assumptions had to be made. First of all, the piconets were assumed to be within range of one another. This means that the only criterion for interference is that both piconets dwell on the same frequency channel during the same time slot. It was also assumed that both piconets were transmitting 1-slot symmetric data packets at 100% throughput. This meant that there was a potential for interference in each time slot. And the final assumption was that both piconets were “synchronized” together as described in Section 2.4.3.

In order to develop a program to test for collisions between piconets, only a few modifications had to be made to the code used to generate a Bluetooth hop sequence. First, hop sequences for two separate piconets had to be generated simultaneously. During each time slot of the hop sequence, the two piconets were tested to see if they were on the same frequency. When a match in frequency occurred, a "collision counter" was incremented. This "collision counter" was used to tally the total number of collisions that took place in the "piconet of interest." The following fragment of code is used to test for collisions between two piconets.

```
for (int j=0;j<50000;j++){
freqa = fh_channel(addressa,clocka,index_array)
freqb = fh_channel(addressb,clockb,index_array)
if (freqa==freqb){collisioncount=collisioncount+1;}          //test for collision
clocka=clocka+2;                                              //Increment to next time slot
clockb=clockb+2;
}
a_file<<collisioncount<<endl;        //Outputs number of collisions  to file
```

In the actual C++ code developed, the collision rate is also calculated. In the previous fragment of code, the collision rate would simply be the number of collisions in the "collision counter" divided by the number of time slots tested. In the fragment of code shown, 50,000 time slots were tested as shown in the *for* loop descriptor.

After developing a program to test for collisions between two piconets, it was necessary to verify its operation. To do so, the code was once again compared to the output of real Bluetooth hardware as measured by the Protocol Analyzer. It would have been ideal to run two piconets simultaneously while monitoring one of them for errors in the Protocol Analyzer. However, the capability to operate two piconets simultaneously at 100% throughput did not exist at the time. Instead, each piconet was run individually with its hopping sequence recorded by the Protocol Analyzer. The "timetick" and frequency channel data was stored from each piconet into an Excel spreadsheet. A sample of the spreadsheet used is shown below in Figure 3.2.3. The spreadsheet was used to test the two piconets for collisions over 50,000 time slots. The “Delta Freq” column calculates the difference in frequency between the two piconets for each time slot. The shaded row marks an occurrence of a collision in frequency. The “Number of Collisions” column counts each instance where there is a “0” value in the “Delta Freq” column and thus the total number of collisions. Only a small sample from the 50,000 time slots is actually shown in the figure below. The number of collisions was calculated by the Excel spreadsheet to be 308. 308 collisions out of 50,000 time slots correspond to a collision rate of 0.62%.

	Piconet #1		Piconet #2			
Time Slot	**Timetick**	**Freq(MHz)**	**Timetick**	**Freq(MHz)**	**Delta Freq**	**Number of Collisions**
125	104004208	2402	186950708	2417	-15	**308**
126	104004210	2417	186950710	2475	-58	
127	104004212	2426	186950712	2449	-23	
128	104004214	2468	186950714	2404	64	
129	104004216	2473	186950716	2419	54	
130	104004218	2464	186950718	2428	36	
131	104004220	2438	186950720	2451	-13	
132	104004222	2423	186950722	2436	-13	
133	104004224	2438	186950724	2468	-30	
134	104004226	2453	186950726	2453	0	
135	104004228	2462	186950728	2421	41	

Figure 3.2.3 Spreadsheet used to predict number of collisions between two piconets using recorded data from Bluetooth hardware

After generating collision data for real Bluetooth hardware, the collision program had to be validated. To do this, the same parameters were used from the hardware test. “Piconet #1” and “Piconet #2” had addresses of 961F92AE and B001913 respectively. The start time for “Piconet #1” was 104003960 and the “inter-piconet offset” between the two piconets was calculated from Figure 3.2.3 to be 82946500. With these four values as inputs, the collision test program was run.

A sample of the output from the collision test program is shown below in Figure 3.2.4. The same interval from Figure 3.2.3 is shown for comparison purposes. Notice that the two samples of data are identical.

	Piconet #1		Piconet#2	
Index	Timetick	Freq (MHz)	Timetick	Freq (MHz)
125	**104004208**	**2402**	**186950708**	**2417**
126	**104004210**	**2417**	**186950710**	**2475**
127	**104004212**	**2426**	**186950712**	**2449**
128	**104004214**	**2468**	**186950714**	**2404**
129	**104004216**	**2473**	**186950716**	**2419**
130	**104004218**	**2464**	**186950718**	**2428**
131	**104004220**	**2438**	**186950720**	**2451**
132	**104004222**	**2423**	**186950722**	**2436**
133	**104004224**	**2438**	**186950724**	**2468**
134	**104004226**	**2453**	**186950726**	**2453**
135	**104004228**	**2462**	**186950728**	**2421**

Figure 3.2.4 Sample Output from Collision Test C++ Program

A thorough comparison of the two sets of data was made with the use of an Excel spreadsheet. It was found that there were zero differences in the data over the entire 50,000 time slots that were tested. The collision test program also output overall collision statistics into a separate file. This output stated ***"There were 308 errors out of 50000 corresponding to a collision rate of 0.00616."*** These collision statistics are identical to those calculated previously using real data collected from the Bluetooth Protocol Analyzer. This process of verifying the collision test program was repeated using addresses from several different Bluetooth devices.

3.2.3 Analysis of Simulation Results

Using the equation for theoretical collision probability from Section 2.5.2, the collision rate with one interfering piconet was found to be .0127. However, the collision test program produced a value for collision rate of 0.00616. This difference in collision statistics prompted a rerun of the simulation using different input parameters (clock start time, "inter-piconet offset" or addresses). Varying the inputs proved to vary the results of the simulation. There were times where the statistics were much worse or much better than the theoretical 1.27 % collision rate. These results can be attributed to the "non-random" nature of the Bluetooth hopping sequence as described in Section 2.2.

Bluetooth devices hop sequentially over 32 hop segments. This means that at any given time slot, the probability of hopping to a particular frequency channel is not random as assumed in the theoretical case. This variance in collision statistics will be further analyzed in Section 3.3.

3.3 Changing Collision Rate

In order to better understand the inconsistency between the theoretical and simulated collision rates, it was necessary to analyze the changes in collision rate. Two key simulations are covered in this section of the chapter. They were developed by making small modifications to the simulation program described in Section 3.2. The first simulation tested for changes in collision rate over time. This was done by measuring the collision rate between two piconets over several short intervals in time. The second simulation was used to analyze the change in collision rate as the "inter-piconet offset" between two piconets was increased.

This section has been divided into four subsections. Section 3.3.1 covers the simulation of collision rate changes over time. Section 3.3.2 is about the simulation of collision rate changes with increasing "inter-piconet time offset." Section 3.3.3 gives a more in-depth description of the Bluetooth frequency hop selection kernel. This description is given during this part of the thesis in order to further explain the results from Sections 3.3.1 and 3.3.2. And finally, Section 3.3.4 compares collision statistics from synchronized and unsynchronized piconets operating under identical conditions.

3.3.1 Collision Rate Changes Over Time

In order to better understand inter-piconet interference, it was necessary to modify the simulation program to test for collision rate changes over time. In Section 3.2, it was shown that the collision rate with one interfering Bluetooth piconet varied from the theoretical value. Hopefully, the process of analyzing the collision rate between two piconets as a function of time will help to reveal the cause of this phenomenon.

In setting up a simulation for inter-piconet interference over time, it was necessary to first decide on the parameters of the test. First of all, the test was run with two piconets, one as the “piconet of interest” and the other as the “interfering piconet.” Both piconets were assumed to be transmitting symmetric 1-slot packets in “synchronized” time slots. The “inter-piconet offset” between piconets was fixed at a constant value. Recall that the “inter-piconet offset” is the difference in the two piconets’ clock integer values (As time progresses in a real situation, this offset would remain constant so long as the clocks do not drift). Collision rates were recorded every 2^{15} transmissions, in other words ever 20.48 seconds. This program was run over an entire cycle of the Bluetooth clock. Recall that this means that there will be 2^{27} transmissions spanning approximately 23 hours. This means that 2^{12} or 4096 collision rate samples will be recorded. And finally, the program had to output the mean collision rate over this entire cycle.

After running the program, the 4096 collision rate samples were analyzed. Figure 3.3.1 below shows a plot of the collision rate versus time. Notice in the figure how the plot appears to be very periodic in nature. The period of 1024 20.48 second intervals is noted first. This corresponds to a span of 2^{25} time slots or ¼ the duration of the entire clock cycle. This periodic nature can be attributed to the Bluetooth hopping algorithm. Bits 26 and 27 happen to be the two most significant bits and are used in determining the phase of the hop sequence. These two bits vary at a period of 2^{25} time slots, just like the period identified previously. The periodic nature makes even more sense when considering the mathematical impact of bits 26 and 27 of the clock. These two bits are only used to supply a phase offset in the addition function at the end of the hopping algorithm. This can be seen in Figure 2.2.1 and Table 2.2.1.

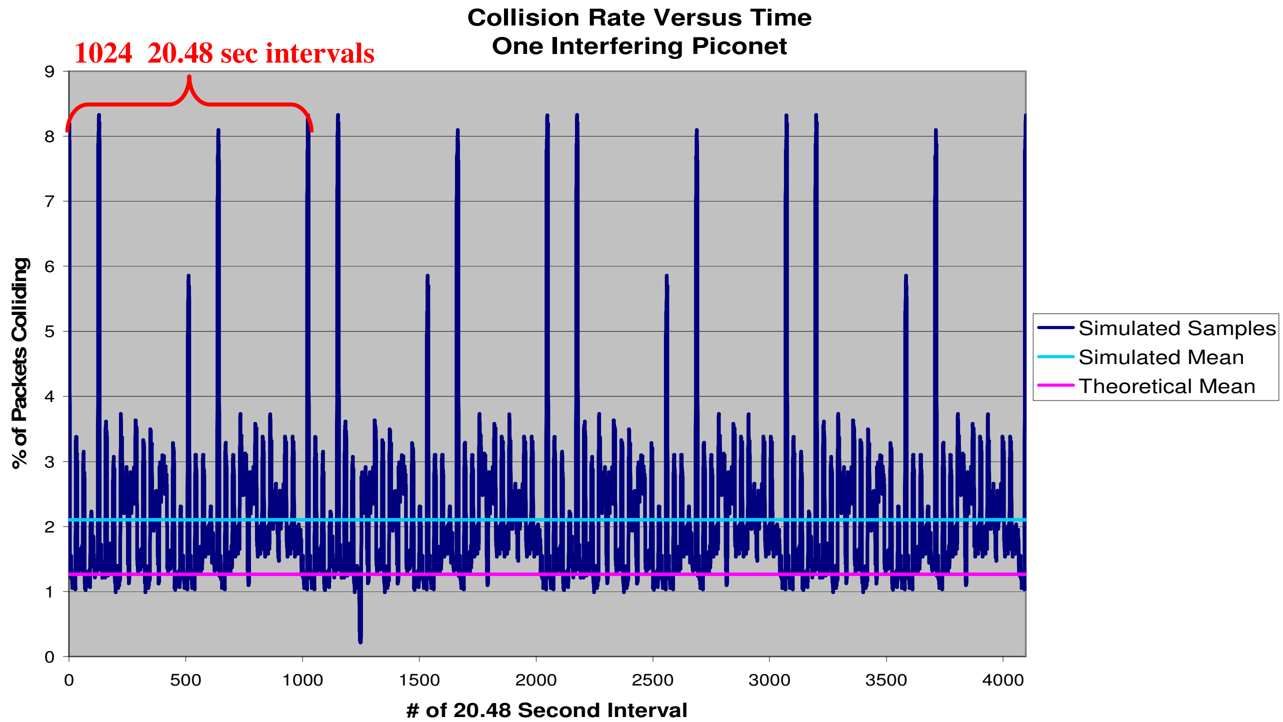


Figure 3.3.1 Collision rate over all time – two 1-slot symmetric "synchronized" piconets

After analyzing the collision rate versus time plot as shown in Figure 3.3.1, a smaller portion of the data was used to get a "zoomed-in" view of the collision rate behavior over time. Figure 3.3.2 shows only the first 200 20.48 second intervals from the simulation. Notice that even more periodic trends can be detected from this figure. Two obvious periods exist at 32 and 128 20.48 second intervals. The first of these periods corresponds to a span of 2^{22} time slots or 1/32 of the entire clock cycle. The second period corresponds to a span of 2^{20} time slots or 1/128 of the entire clock cycle. Both of these periods can also be attributed to the cyclic nature of the frequency hopping algorithm. The period of 32 20.48 second intervals or 2^{20} time slots occurs as the first 20 bits of the clock reset. The frequency hopping algorithm must be explored in order to better understand why this causes a periodic behavior in collision rate. Bits 21 through 25 are fed into an addition at the beginning of the algorithm as shown in Figure 2.2.1 and Table 2.2.1. They are modified only at a period of 2^{20} time slots, which is when the first 20 bits of the clock will reset. Each time these 20 bits reset on the "piconet of interest," the bits on the "interfering piconet" will be the same as they were upon the last reset. Of course, this is under the assumption that their clocks do not drift apart.

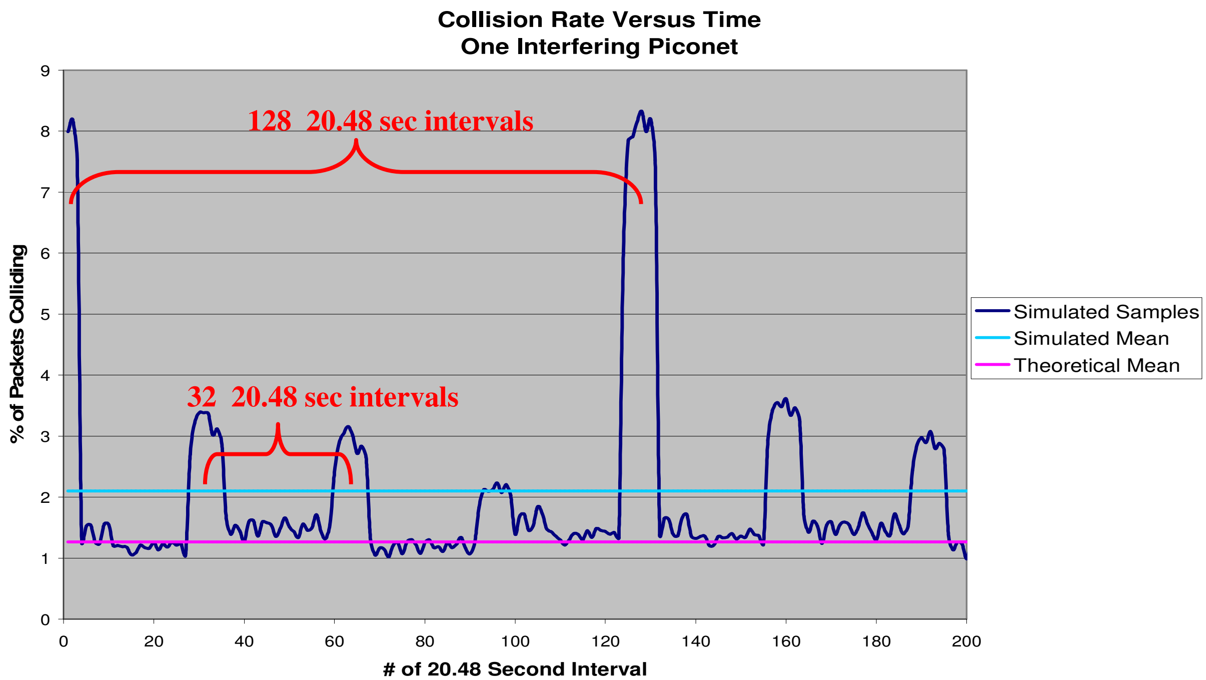


Figure 3.3.2 Smaller time window of collision rate versus time – two 1-slot symmetric "synchronized" piconets

It can also be seen in Figure 3.3.1 and 3.3.2 that the mean collision rate is somewhat higher than the theoretical collision rate for the conditions chosen. Different addresses and "inter-piconet" offsets were used in rerunning the simulation. After repeating the simulation for collisions versus time using different "inter-piconet" time offsets, it became apparent that the mean collision rate varied. The simulation program had to be modified to test collisions versus increasing "inter-piconet" time offset in order to better understand this effect.

3.3.2 Change in Collision Rate with Increasing "Inter-piconet Offset"

After analyzing the changes in collision rate with time, it was necessary to take a look at the effects of changing the "inter-piconet" time offset. The simulation program was once again modified in order to conduct such a test. Before modifying the program, several test parameters were chosen. Once again, the test was run with two piconets, one as the "piconet of interest" and the other as the "interfering piconet". Both piconets were also assumed to be transmitting symmetric 1-slot packets in "synchronized" time slots.

Collision rate samples were taken over a fixed interval of 2^{13} time slots or 5.12 seconds. The "inter-piconet time offset" was incremented by one time slot between samples. In total, this process was repeated to produce 16,384 samples.

The results of the simulation revealed a lot about the interference that exists between neighboring piconets. Figure 3.3.3 below shows the entire set of 16,384 samples beginning with an "inter-piconet" time offset of 466,866 clock ticks. This means that one piconet has a clock integer value that is 466,866 clock ticks larger than the second piconet. It is apparent from the plot that the collision rate between two piconets is periodic with the "inter-piconet" time offset between the two different Bluetooth hopping sequences. The plot below shows that collision rate is periodic as the time offset is increased 10112 clock cycles. Recall that 10112 clock cycles is equivalent to 5056 time slots. Another important observation was that there were certain offsets which gave really high collision rates. An offset of 494216 between the two piconets' clocks produced a collision rate of 8.51% over the 5.12 second interval tested. This is well above the theoretical value for two "synchronized" piconets. There are also several offsets which provide 0 collisions over the 5.12 second interval. This particular simulation over all 16,384 different samples produced a mean collision rate of 1.266%, matching the theoretical rate for this case.

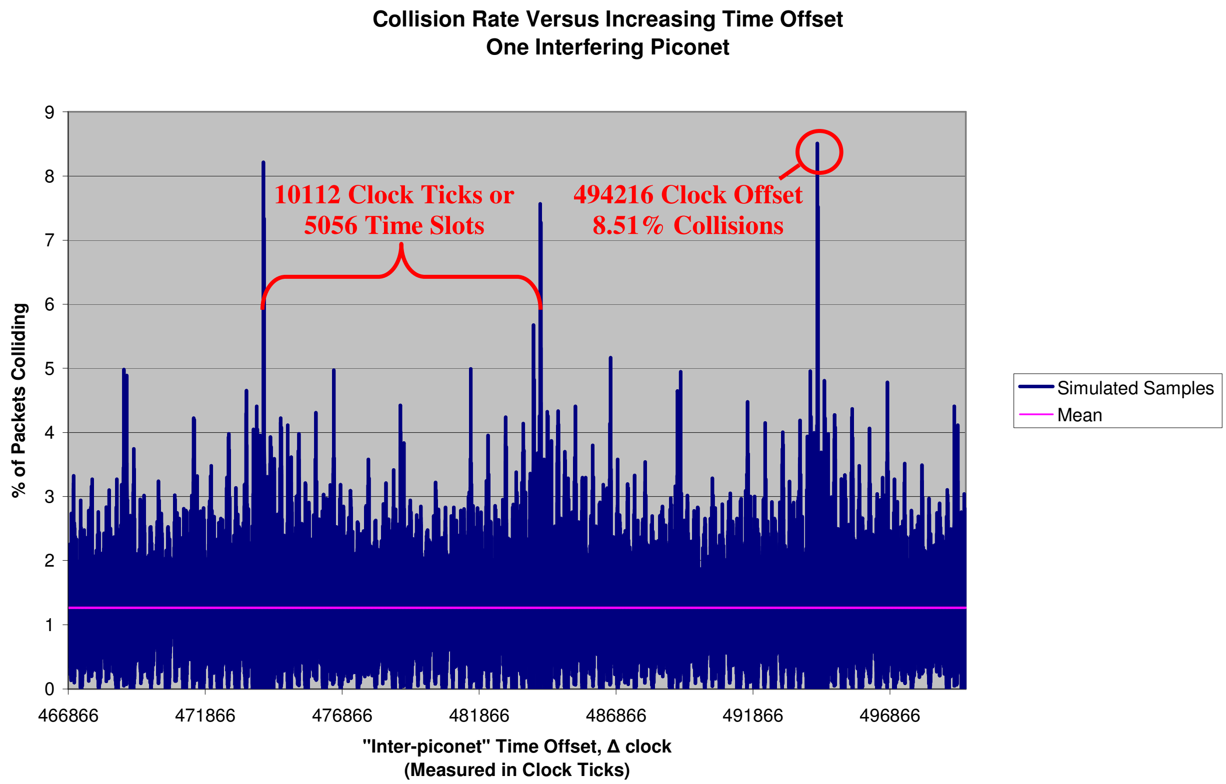


Figure 3.3.3 Collision rate versus "Inter-piconet" time offset – two 1-slot symmetric "synchronized" piconets

By taking a closer look at the data in Figure 3.3.3, a period of 316 clock ticks of offset, or in other words 158 time slots, was detected. An even closer view of the data is shown below in Figure 3.3.4. An important observation from this plot is that the collision rate tends to vary drastically each time the "inter-piconet" time offset is incremented. Keep in mind that the x-axis of the plot represents the time tick difference in the piconets clocks. Two time ticks correspond to one change in time slot offset. The plot shows that for most of the samples, each increase in offset causes the collision rate to alternate above and below the mean rate of 1.266%. In order to fully understand "inter-piconet" interference, the observations made from this simulation must be explained. This means that the frequency hopping algorithm will have to be explored in greater detail than in the Background section of this thesis. This detailed explanation of the Bluetooth Hopping Algorithm will be explained in Section 3.3.3 below.

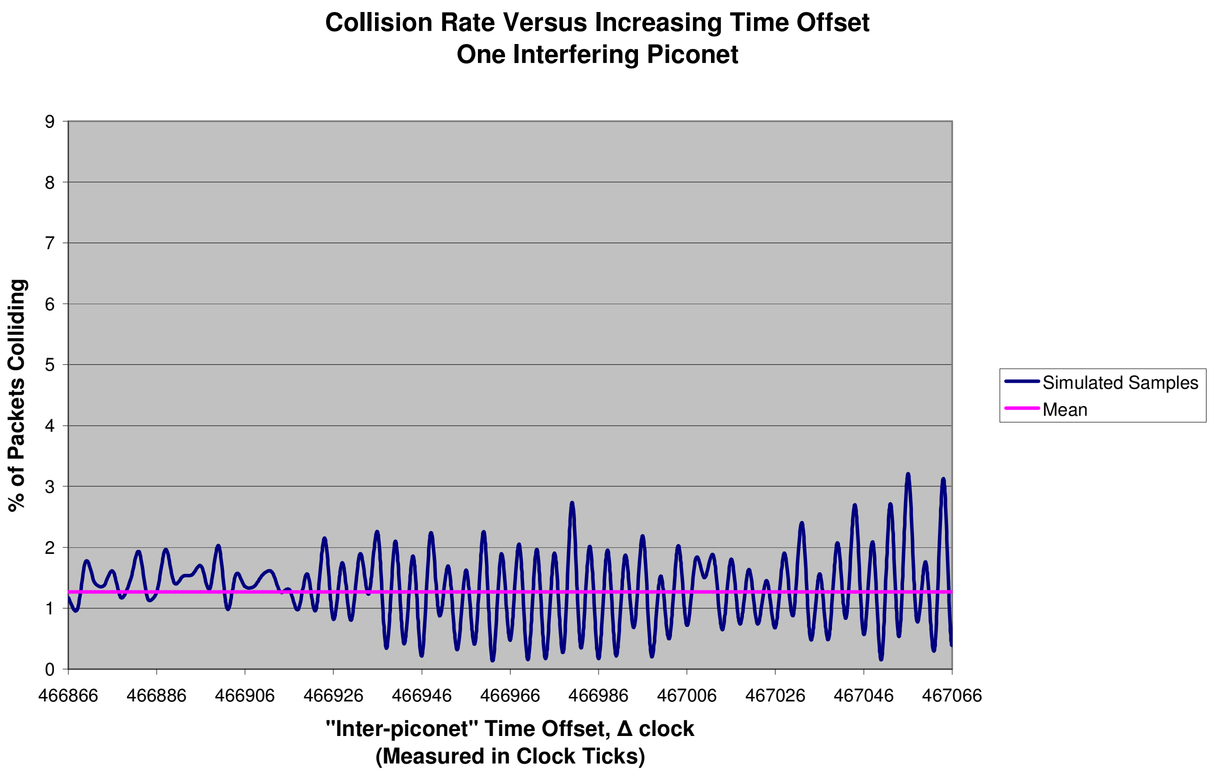


Figure 3.3.4 Closer view of collision rate versus "Inter-piconet" time offset – two 1-slot symmetric "synchronized" piconets

3.3.3 Better Understanding the Bluetooth Hopping Algorithm

In order to fully understand inter-piconet interference, it is necessary to fully understand the behavior of the Bluetooth frequency hopping algorithm. In Section 2.2 of this thesis, the algorithm was briefly described so that the reader could acquire a general understanding of its non-random nature. Now, the algorithm will be broken down into its finer details in an effort to fully explain the behavior of two Bluetooth piconets interfering with one another. This analysis will show why the collision rate is a periodic function of the clock time and the offset in the number of clock cycles between piconets.

The first thing to do is to revisit the diagram of the frequency hopping algorithm depicted in Figure 2.2.1 and Table 2.2.1. Keep in mind that for any given piconet, the "master address" in a piconet will remain constant. This means that the address portions of all interfering piconets can be ignored in this analysis. The clocks of the piconets are of the most significance to this analysis. Clock bit 1 has the most significance to the algorithm. It is changed every other clock cycle, alternating so that the piconet switches between transmit and receive modes. When clock bit 1 is a ‘0’, master-to-slave transmission will occur. For this analysis, the master of the piconet is used as the point of interest, so we will refer to a clock bit 1 of ‘0’ as “transmit.” When the clock bit 1 is a ‘1’, slave-to-master transmission occurs. For this analysis, we will refer to a clock bit 1 of ‘1’ as “receive.”

Now let’s look at the second most significant portion of the clock. Clock bits 2 through 6 are used to determine the position among a 32-hop segment. Recall that in Section 2.2, it was stated that the algorithm causes the devices to hop sequentially through 32-hop segments. This will now be further clarified. Two interleaved 32-hop segments will exist in the piconet at any point in time. One of these is for “transmit” and the other for “receive.” Remember that there are 32 possible values for Clock bits 2 through 6. And for each of these values Clock bit 1 can be a ‘0’ or ‘1’, switching the piconet between “transmit” and “receive” modes. The Bluetooth specification still holds true in stating that no frequency channel is repeated in a given 32-hop segment. However, a common frequency channel may occur between consecutive “transmit” and “receive” segments.

It is now important to look at the frequency hopping algorithm as two separate blocks as shown below in Figure 3.3.5. The first block can be referred to as the “ordering” portion of the algorithm [8]. This block includes everything through the output of the permutation operation. It is during this portion of the algorithm that the “pseudo-random” order for the 32-hop segments is generated. The output of “ordering” block is a 5 bit integer that will range from decimal value 0 to 32. It is now important to look at the second portion of the frequency hopping algorithm. This block, referred to as “mapping,” is shown below in Figure 3.3.5. The “mapping” block is the part of the algorithm responsible for selecting the 32-hop segment to be used. The inputs to this block are listed below in Table 3.3.1. These inputs are all added together in the “add” block before being sent through a modulo-79 operation. The final output is an index used to select the appropriate channel from the frequency selection register.

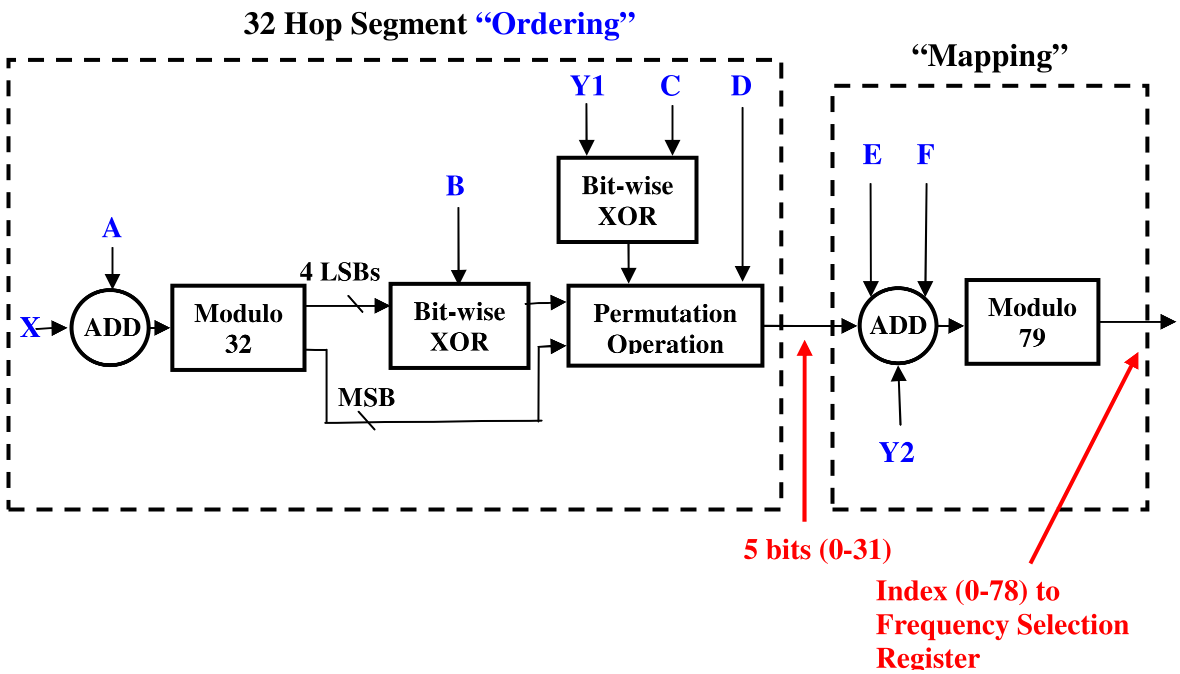


Figure 3.3.5 Hop Selection Kernel as defined in the Bluetooth Specification (© 2002 IEEE) [1,8]

Input	Value In	Comments
Output from "Ordering"		Ouptut from perumation is 5 bits, decimal values 0 to 31
E	Address bits (13,11,9,7,5,3,1)	Fixed Value
F	(16 x Clock bits 27-7) mod 79	Varies only after TX and RX 32-hop segments complete (that's when Clk bit 7 inrements)
Y2	32 x Clock bit 1	Selects between TX or RX segments, adds an offset to "Mapping" portion of 0 or 32

Table 3.3.1 Inputs to "Mapping" portion of Hop Selection Kernel

Now that the frequency hopping algorithm has been further analyzed, it is necessary to understand how this algorithm affects the collisions between two piconets. First, we must consider the frequency selection register indexed by the output of the algorithm. This register was briefly described in Section 2.2 of this thesis. Recall, that the frequency channels in the register are ordered in such a way that the first 40 code words select the even frequency channels and the last 39 code words select the odd channels. The inputs

“E” and “F” in the “mapping” block are used to select the 32-hop segment of channels within this register. The input “Y2,” when high, adds an offset of 32 to separate the 32-hop receive segment away from the transmit segment. The 32-hop segments in the register are represented graphically below in Figure 3.3.6. It is the “ordering” block of the algorithm which determines the hopping order of these 32-hop segments. As the 32-hop segments complete their cycle, input “F” (Clock bit 7) is incremented resulting in an increase of 16 index positions as shown in Figure 3.3.6. Columns A, B, and C in the figure represent single increments of this input. Between the algorithm itself and Figure 3.3.6, it is obvious that there are 79 possible “mapping” outputs. Therefore, there are 79 different 32-hop segment positions.

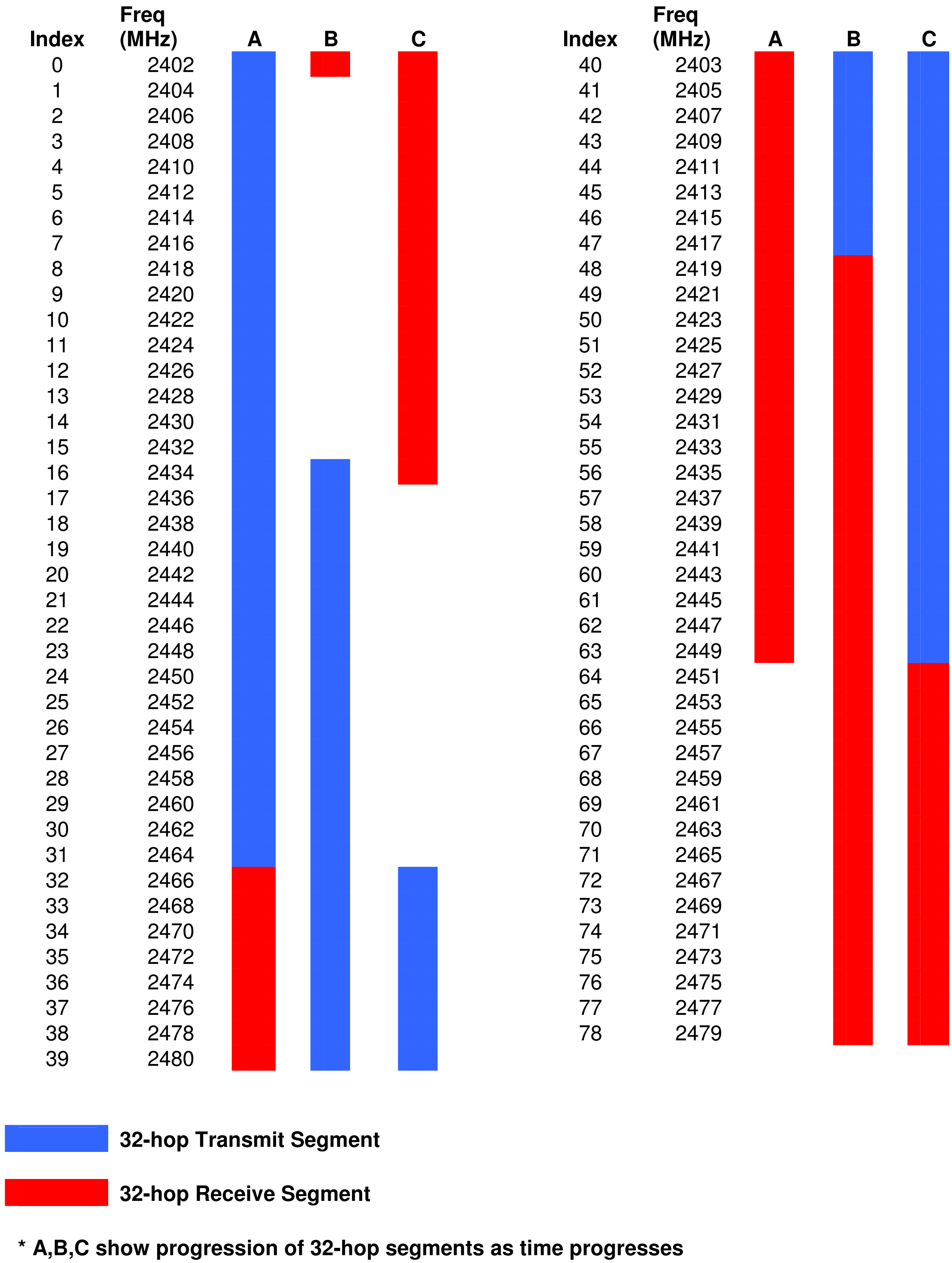


Figure 3.3.6 "Mapping" of 32-hop segments as time progresses

It is now necessary to use this analysis of the frequency hopping algorithm to explain the observations of Section 3.3.2. The first observation from Section 3.3.2 was that the collision rate was periodic with increasing "inter-piconet" time offset. The two periods observed were of 158 and 5056 time slots. These two values happen to both be multiples of 79. The periods exist because there are only 79 different 32-hop segments. As the "inter-piconet" time offset is increased by 79 time slots, the mapping of 32-hop segments in the two piconets is repeated. This is why the collision probability appears to be periodic over multiples of this interval.

The second observation was that the statistics often varied greatly with each single increment of "inter-piconet" time offset, as shown in Figure 3.3.4. For a given 32-hop segment in the piconet of interest, an increase in "inter-piconet" time offset between itself and the interfering piconet will result in a change in collision probability. This can be better understood by analyzing Figure 3.3.6. Let's assume that the piconet of interest is "mapped" to use the 32-hop segments in "Column A." If the interfering piconet uses the same mapping and the "inter-piconet" time offset is such that the two piconets transmit at the same time, then the probability of collision will be very high. This would be a worst-case scenario for collision probability. However, if the "inter-piconet" time offset is incremented by 1 time slot, then the interfering piconet will be transmitting while the piconet of interest is receiving. Since the 32-hop transmit and receive segments do not overlap at all, there will be zero interference. This corresponds to the best-case scenario for collision probability. Incrementing the "inter-piconet" time offset by another single time slot will cause the two piconets to once again transmit simultaneously. This will cause the statistics to rise once again. Keep in mind that continuing to increase the time slot offset will eventually cause the interfering piconet to change the position of its 32-hop segment relative to the piconet of interest.

3.3.4 Changes in Collision Rate for Unsynchronized Piconets

Most of the analysis in this section was done under the assumption that the two piconets had "synchronized" time slots. This does not explain what will happen in real Bluetooth devices where time slots between piconets will often be "unsynchronized." There is no way to control the alignment of time slots using the current Bluetooth specification. Therefore, changes in collision rate for "unsynchronized" piconets must be analyzed as well. If you recall Section 2.4 of this thesis, "unsynchronized" piconets face a double threat for interference. This means that each transmission can potentially interfere with two transmissions in the opposing piconet. One way to calculate the collision rate for unsynchronized piconets would be to add the number of collisions that occur due to two consecutive "inter-piconet" time offsets. A simulation of two "unsynchronized" piconets versus increasing time offset was run to prove this theory. A sample of the output data is shown in Figure 3.3.7. The figure shows that the "unsynchronized" collision curve tends to follow the envelope of the "synchronized" curve. It is very apparent from this figure that the "unsynchronized" collision rate is the sum of collisions from two consecutive inter-piconet time offsets. This figure also shows that the "unsynchronized" mean is equal to the theoretical mean when averaged over several samples taken at different time offsets. It should also be noted that the improvement gained by synchronization is negligible for some offsets. Figure 3.3.8 shows just how little improvement is achieved at one particular fixed "inter-piconet" time offset. This particular simulation was run using the inter-piconet time offset producing worst-case interference for two synchronized piconets. This time offset was taken at the peak level of collision percentage found in Figure 3.3.3. This suggests that for two piconets, the synchronization of the piconets' time slots is advantageous only if the inter-piconet time offset is controlled as well.

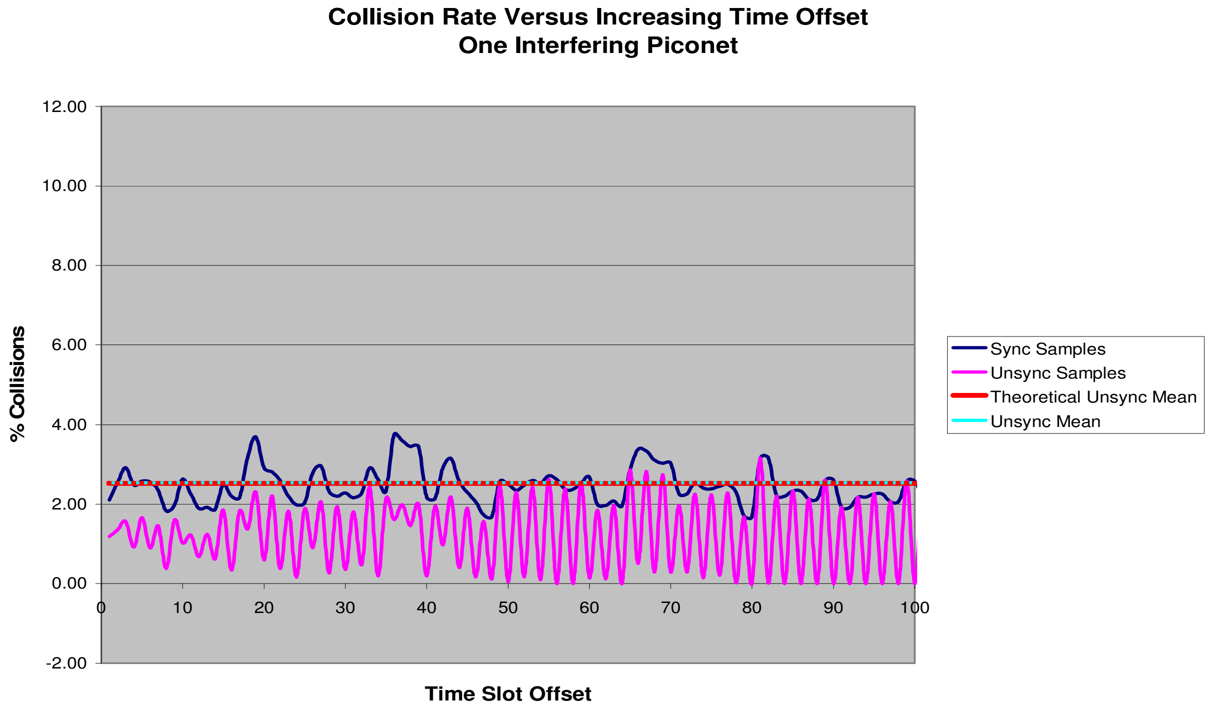


Figure 3.3.7 Collision rate versus increasing time offset for 2 unsynchronized and 2 synchronized piconets

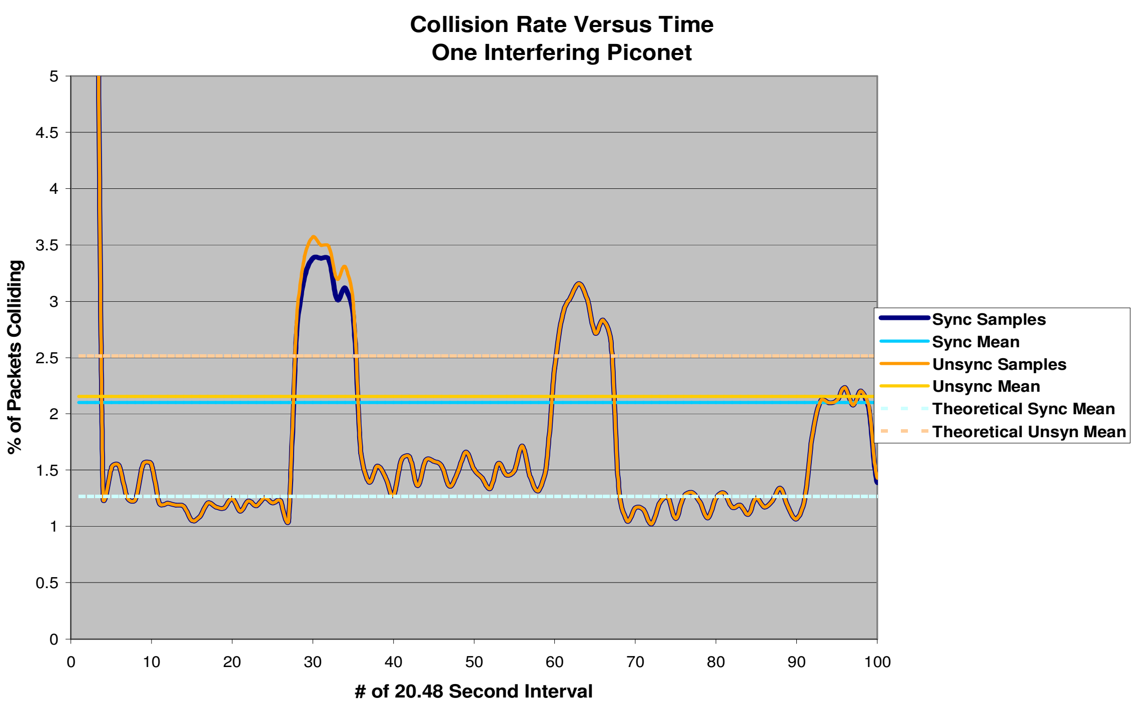


Figure 3.3.8 Collision rate versus time for 2 unsynchronized and 2 synchronized piconets w/ Inter-piconet time offset of 247,108 time slots

The inter-piconet time offset of 247,108 time slots (494,216 clock cycles) used to collect the data for Figure 3.3.8 was incremented by one to test for improvement. Using this new time offset of 247,109 time slots, and keeping all other variables constant, the data in Figure 3.3.9 was produced. This data shows that increasing the offset has little improvement on the collision rate over time for two unsynchronized piconets. The mean collision rate only went up from 2.15% to 2.24%. Both of these values are still below the theoretical mean of 2.52% for two unsynchronized piconets. However, the data does show drastic improvement in collision rate for two synchronized piconets. From one offset to the next, the mean collision rate for two synchronized piconets drops from 2.10% to 0.14%, as represented between Figures 3.3.8 and 3.3.9. Keep in mind that the time offset or the worst-case collision rate was used initially. This means that the analysis here shows the best possible improvement that can be achieved by synchronization in addition to control of the inter-piconet time offset.

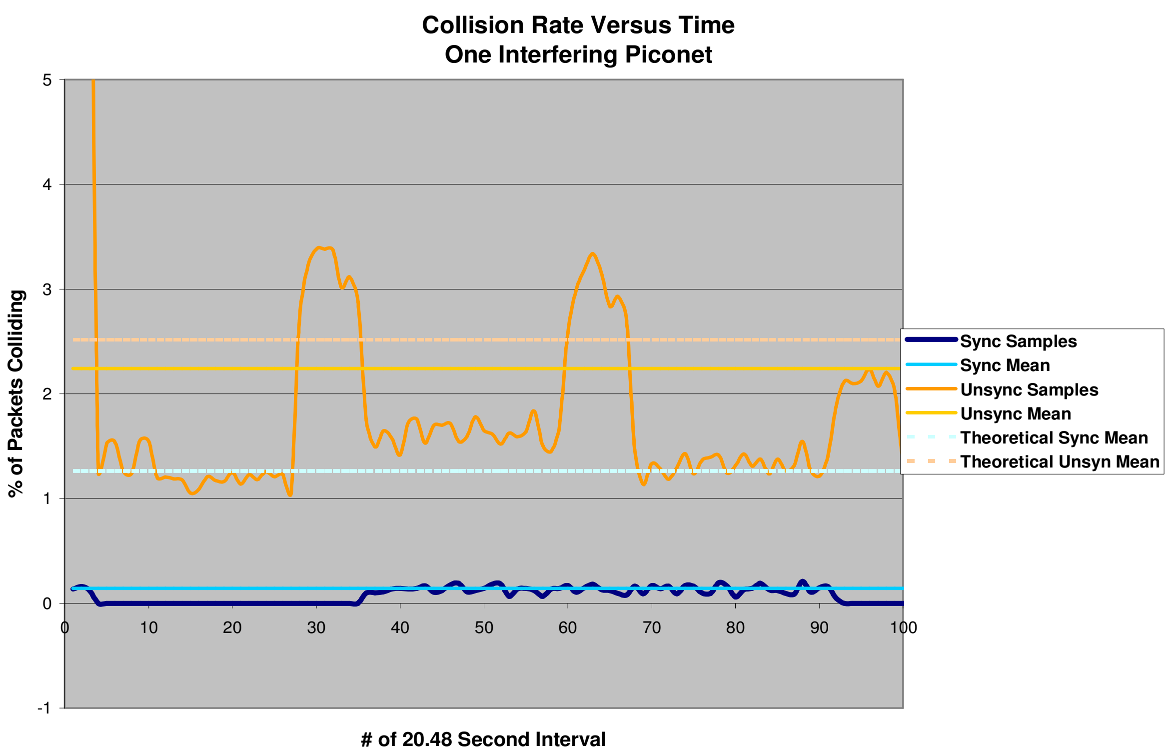


Figure 3.3.9 Collision rate versus time for 2 unsynchronized and 2 synchronized piconets w/ increased Inter-piconet time offset of 247,109 time slots

3.4 Distribution of Collision Rate Statistics

It was shown in the previous section that collision rate changes as a function of the inter-piconet time offset. In this section, the distribution of these statistics will be briefly analyzed. The data used in this section was collected the same way as in Section 3.3.2. Samples were once again taken over 5.12 second intervals as the inter-piconet time offset was increased. After collection, the samples were sorted in Microsoft Excel to represent the statistical distribution of collision rates over a sample of different inter-piconet time offsets. Figure 3.4.1 below shows the distribution of collision rates for 2, 3, and 4 unsynchronized piconets. Notice the variance in each of these distributions. The mean and variance increases as the number of piconets is increased.

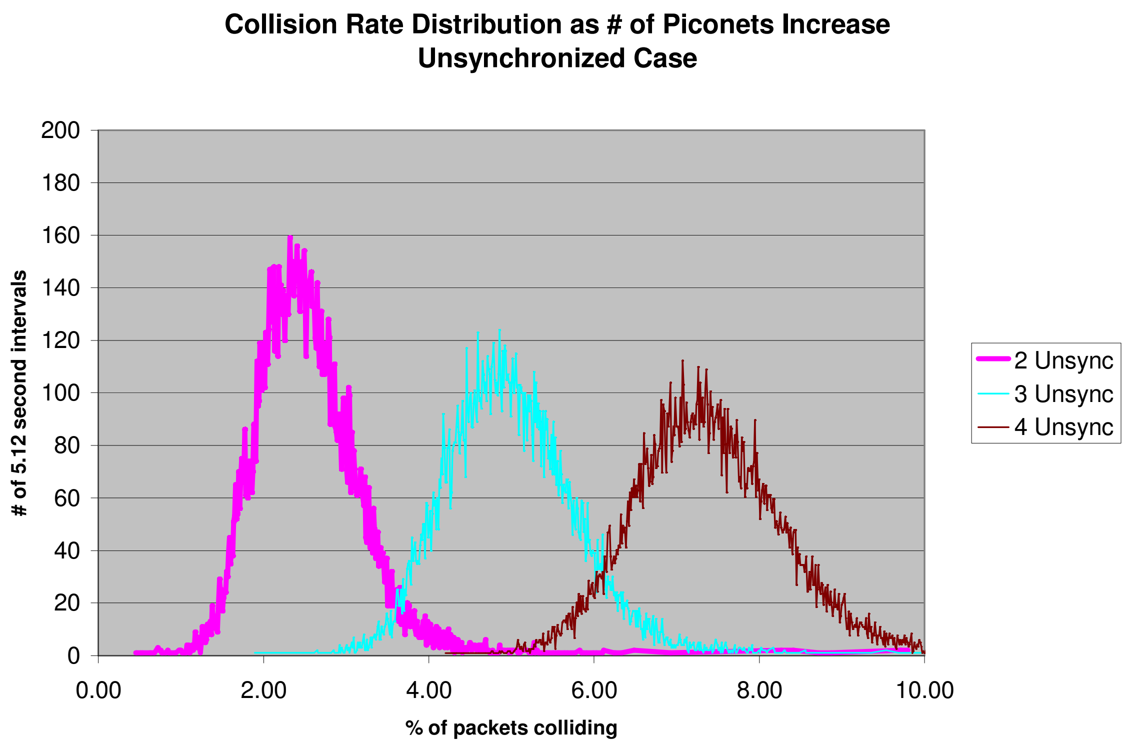


Figure 3.4.1 Collision rate distribution as number of interfering piconets increases – unsynchronized case

The analysis was then repeated for synchronized piconets. Figure 3.4.2 shows the statistical distribution of collision rates for 2, 3, and 4 synchronized piconets. The mean and variance also increase with the number of piconets.

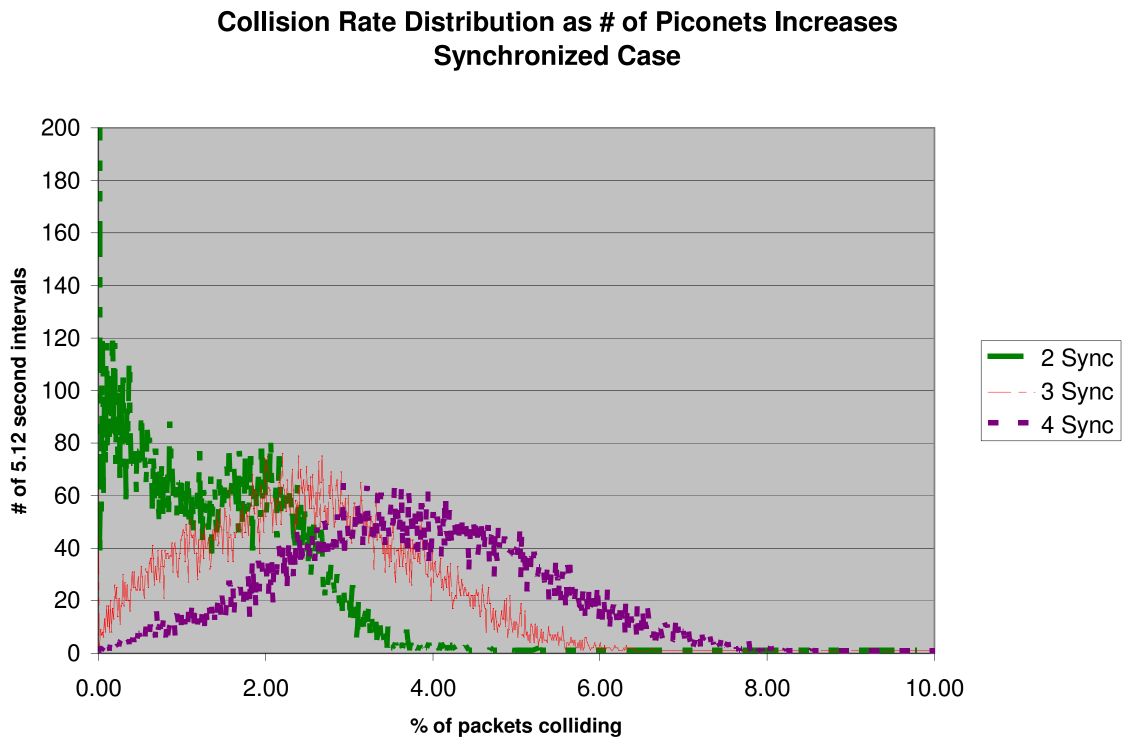


Figure 3.4.2 Collision rate distribution as number of interfering piconets increases – synchronized case

It is important to note how much larger the statistical variance is for synchronized piconets then for unsynchronized piconets. This occurs because during the synchronized case, transmissions from the interfering piconet are not a double threat. This makes the case of zero interference, explained in Section 3.3.3, possible. Table 3.4.1 compares the mean of these distributions to the theoretical collision rate calculated using equations 2.1 and 2.2. The measured variance is also listed in this table. It is important to recognize that the mean taken from the results of testing collision rate versus inter-piconet time offset is equivalent to the theoretical mean. This shows that the collision rate distribution is a function of this offset, and the mean can be found by averaging several different samples taken at random offsets. The next section will use this finding to simulate the mean collision rate as the number of piconets increases.

	# Piconets	Mean Collision Rate % Simulated	Theoretical	Variance
Sync	2	1.266	1.266	1.266
	3	2.514	2.516	1.665
	4	3.750	3.750	2.378
Unsync	2	2.532	2.516	0.411
	3	4.995	4.968	0.663
	4	7.399	7.359	0.857

Table 3.4.1 Collision statistics with increasing number of neighboring piconets

3.5 Collision Rate with Increasing Number of Piconets

Up to this point, a thorough analysis of collision statistics has been completed for a small number of piconets. This section will show how the throughput performance of a piconet is hampered as the number of interfering piconets is increased. Several different simulations were performed in order to represent the effects of using different packet sizes, symmetric/asymmetric and synchronized/unsynchronized data packets.

3.5.1 Comparison of Throughput to Theoretical Equations

The simulation program was modified once again in order to simulate the decrease in throughput as a result of increasing the number of interfering piconets. The data was collected by taking 1000 5.12 second samples that were each captured using a different inter-piconet time offset. This result represents the mean collision rate, considering the random inter-piconet time offset. The aggregate throughput was calculated by multiplying the throughput per piconet by the total number of piconets and the maximum data rate of 345.6 kbps (2-way) for DH1 transmissions. The throughput was calculated for the case where the time slots for all interferers were either synchronized or unsynchronized. Equations 2.3 and 2.4, equations for calculating the theoretical throughputs, were used to compare the simulated and theoretical data. Figure 3.5.1 shows the simulated and theoretical aggregate throughputs as a function of the total number of interfering piconets. In this figure, the "single-interfering" curves represent transmissions where the time slots between piconets were synchronized. The "double-interfering" curves represent the worst possible case of unsynchronization, where every interfering piconet poses a double threat to interfere with the piconet of interest. In actual practice, the throughput would fall somewhere between these two extremes.

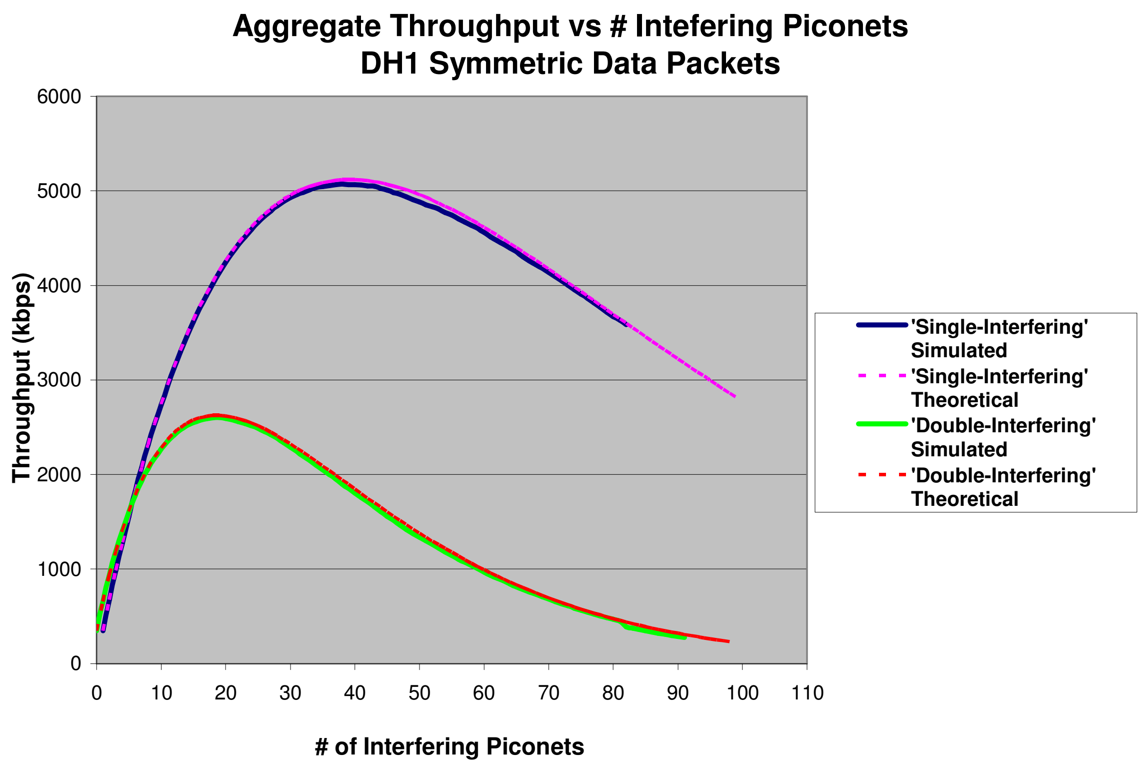


Figure 3.5.1 Aggregate throughput versus number of interfering piconets – all transmitting DH1 symmetric data packets

From the figure, it is even more apparent that each mean throughput calculated by varying the inter-piconet time offset is equal to the theoretical predictions. The curves also show the improvement that can be achieved by synchronizing time slots among neighboring piconets. With unsynchronized interferers, the maximum aggregate throughput is achieved with about 20 piconets. When the piconets are all synchronized, the aggregate throughput continues to increase until there are about 40 piconets. The maximum aggregate throughput doubles from the unsynchronized case to the synchronized case.

3.5.2 Effect of Different Types of Interferers on 1-slot Transmissions

After analyzing the effect of 1-slot interferers on a one slot transmission, it was necessary to take a look at the effects of different types of interferers. The simulation program was modified to simulate interference cause by twelve different types of interference. The twelve different types of interference accounted for every combination of packet size, symmetric/asymmetric transmission and synchronized/unsynchronized time slots. For asymmetric transmission, 3 or 5 slot transmissions are followed by a 1 slot response, while 1 slot transmissions were followed by a Null response. Once again, each data point was collected by taking 1000 5.12 second samples that were each captured using a different inter-piconet time offset. Figure 3.5.2 shows the effects of these different types of interfering transmissions on a Bluetooth piconet transmitting 1-slot symmetric data packets. To review the characteristics of the different types of packet transmissions, see Table 2.3.1.

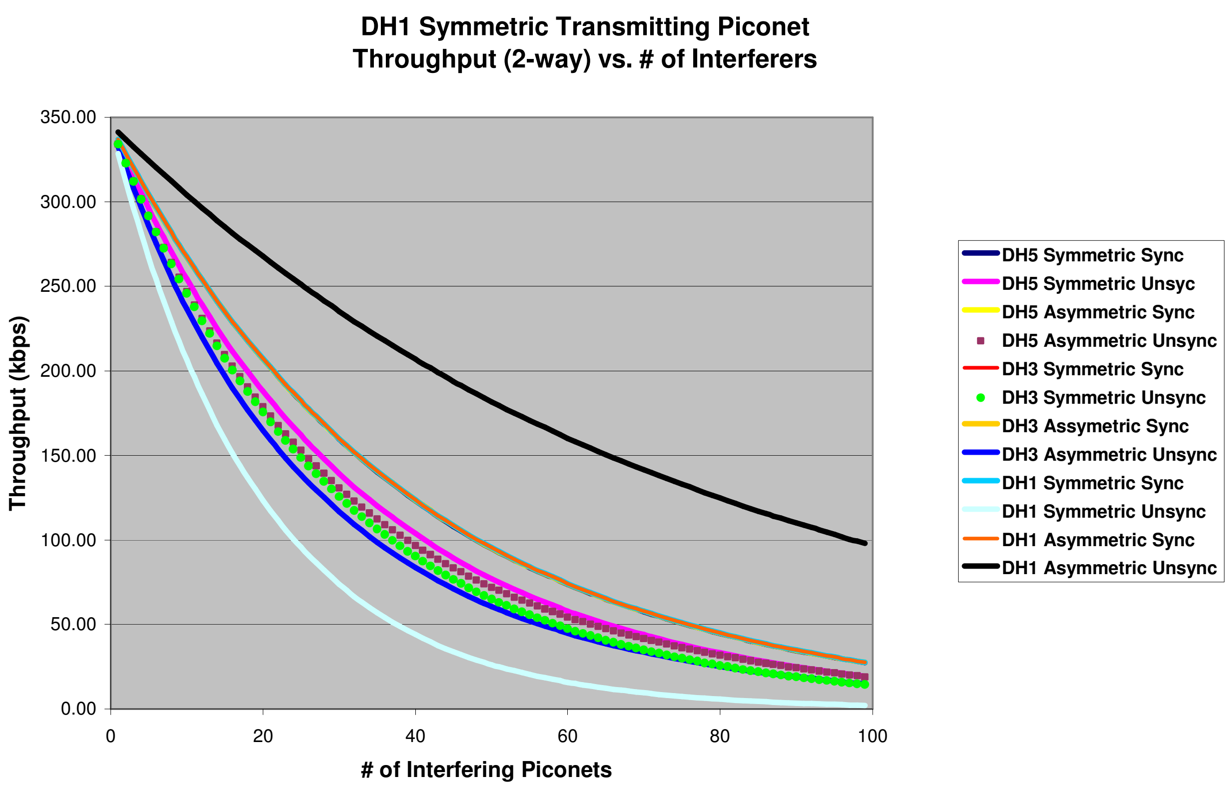


Figure 3.5.2 DH1 symmetric transmission throughput versus number of interfering piconets with different types of transmissions

From Figure 3.5.2, it is apparent that the least amount of interference occurs when the interfering piconet is transmitting DH1 asymmetric packets that are unsynchronized. This case represents the most likely (and most beneficial) unsynchronized case where the transmit packet is a potential interferer, however the null response is not. Figure 3.5.3 shows how the transmissions must be aligned in time to produce this result. This figure shows that "Packet 1" from the interfering piconet is a threat to interfere with "Packet 1" in the piconet of interest. However, the Null response in the interfering piconet is not a threat to interfere at all. This specific type of interference produces the best case throughput curve shown in Figure 3.5.2

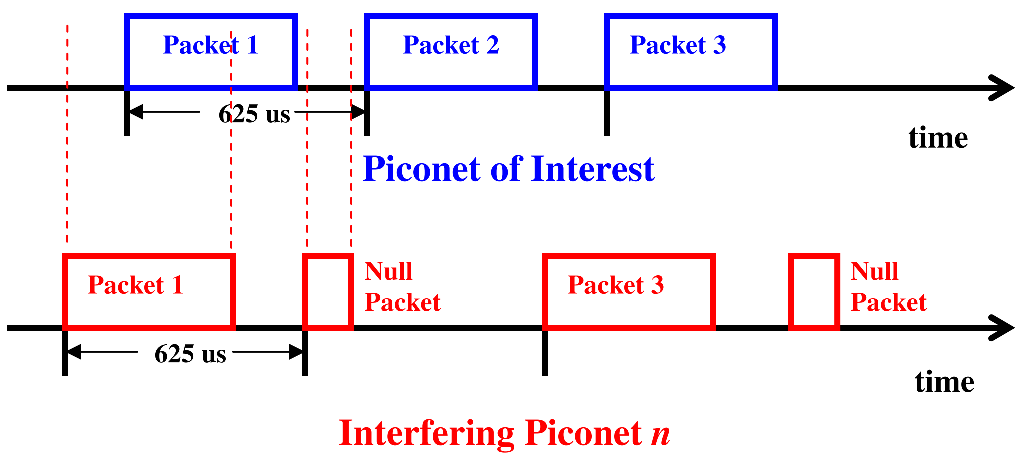


Figure 3.5.3 Example of unsynchronized piconets where half of the transmission is not a threat to interfere

The next best throughput performance was achieved when the piconet of interest was in the presence of interfering piconets that had synchronized time slots. In fact, when the time slots between piconets are synchronized, the same amount of interference occurs on a DH1 transmission regardless of the interferers' packet size. This can be seen on the second curve from top in Figure 3.5.2. This curve actually shows the curves from all six synchronized cases overlapping one-another.

For unsynchronized piconets, the best throughput was achieved when the interferers transmitted larger packet sizes. This is because when the interferers are larger packet sizes, double threats will occur less frequently on a 1-slot transmission. This is shown graphically below in Figure 3.5.4. Notice in the figure that both “packet 1” and “packet 2” in the “DH5 interfering piconet” pose a potential to interferer with “packet 1” in the “piconet of interest.” However, only “packet 2” in the “DH5 interfering piconet” can interfere with the next four packets in the “piconet of interest.” Now notice that each transmission in the unsynchronized “DH1 interfering piconet” is a double threat to interfere. This shows that double threats occur less frequently when the interfering piconets are transmitting larger packet sizes.

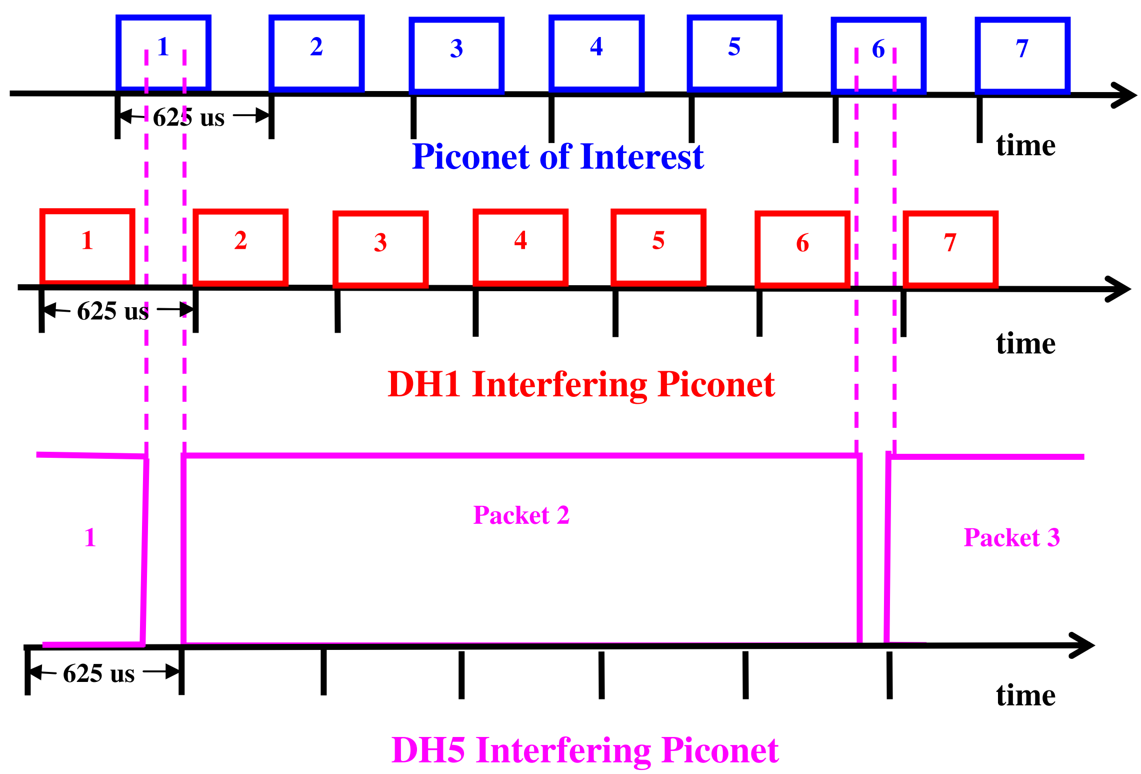


Figure 3.5.4 Example of larger packet sizes posing as a smaller interference threat to DH1 transmission.

The final observation from Figure 3.5.2 is that asymmetric interferers are more of a threat than symmetric interferers. This holds true for the same reasoning that was just given.

Asymmetric transmissions respond with smaller packets, therefore will have a smaller average packet length then symmetric transmissions. Double threats for interference will occur more frequently because of this.

3.5.3 Effect of Different Types of Interferers on 5-slot Transmissions

The simulation from Section 3.5.2 was repeated to test for the effect of different types of interferers on 5-slot symmetric and asymmetric transmissions. The results from this simulation are shown in Figure 3.5.5 and Figure 3.5.6 respectively.

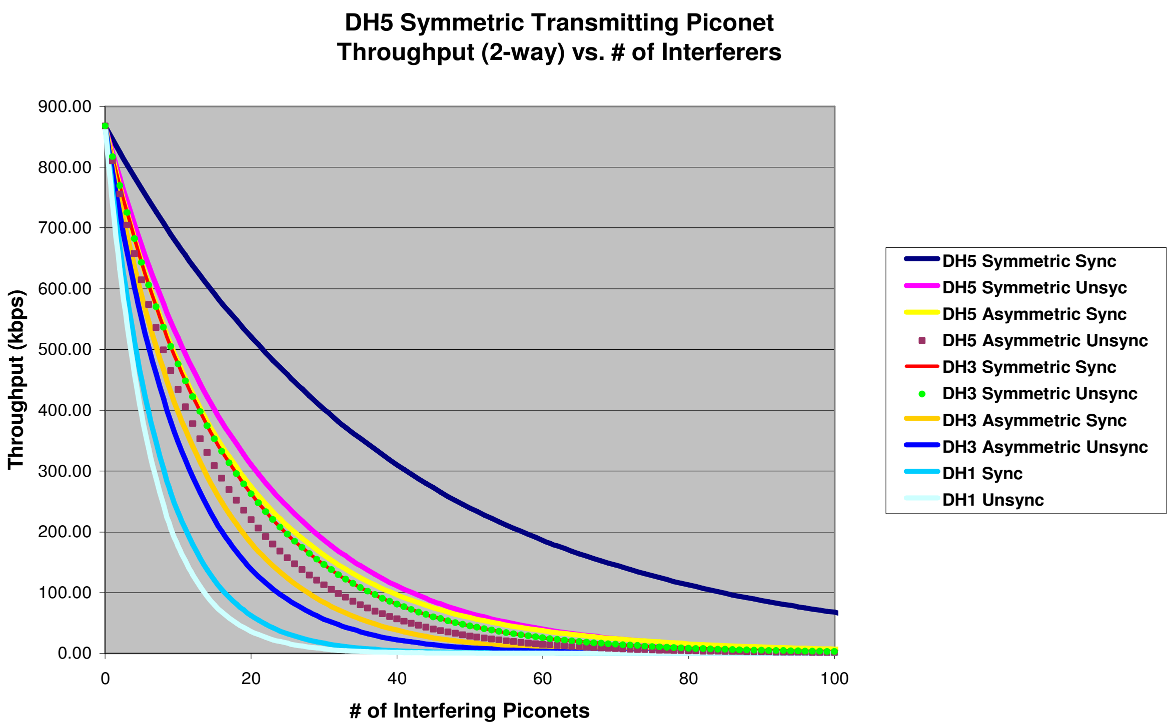


Figure 3.5.5 DH5 symmetric throughput vs. number of interfering piconets with different types of transmissions

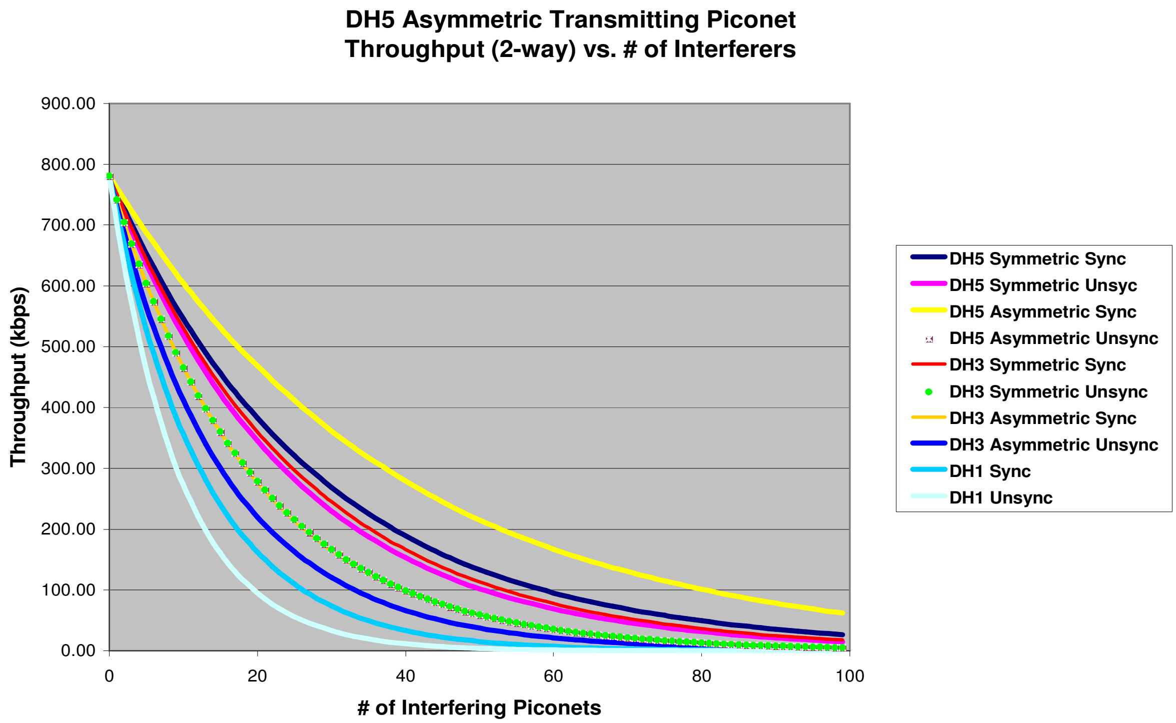


Figure 3.5.6 DH5 asymmetric throughput vs. number of interfering piconets

These results show that DH5 transmissions were also more susceptible to smaller packet sized interferers. The best throughput was achieved when the interfering piconets used the same type of transmission as the piconet of interest. Synchronizing time slots among piconets once again showed improvement for each type of interferer.

3.5.4 Throughput Comparison for Particular Types of Interferers

Up to this point, it has been shown that certain types of Bluetooth transmissions are more likely to cause inter-piconet interference. This section will compare types of transmissions to see which is the most vulnerable to a particular type of interferer. The data used in this analysis was extracted from the simulations described in Section 3.5.3. Each plot here represents how DH1, DH5 symmetric and DH5 asymmetric transmissions each hold up against particular types of interferers. Figure 3.5.7 shows the effect of DH1 interference on these three types of packets.

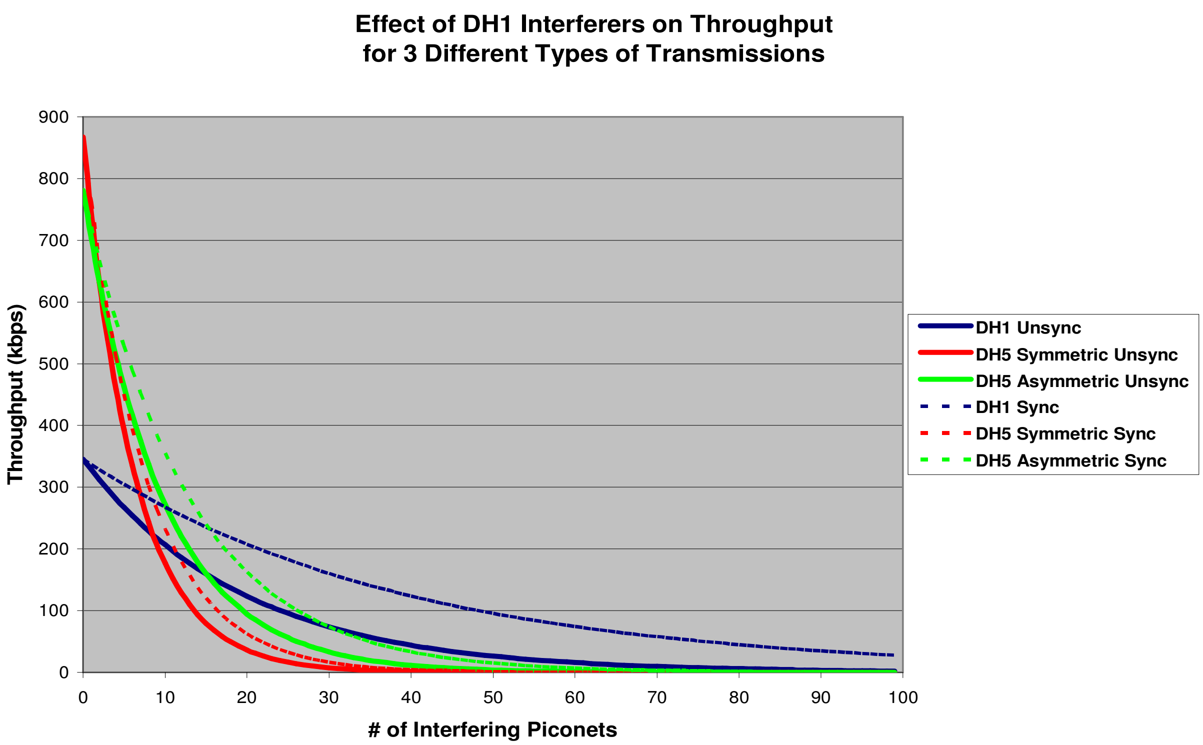


Figure 3.5.7 Effect of DH1 interferers on throughput for 3 types of transmissions

The curves in Figure 3.5.2 show that as the number of interfering piconets increase, different types of transmissions can provide higher throughput. Larger data packets are more susceptible to data collisions with 1-slot interferers. However, maximum throughput will occur with 5 slot transmissions until there is a significant amount of interference. When there are more than two DH1 interferers present, the total 2-way throughput becomes greater through the use of DH5 asymmetric data packets. When the number of DH1 interfering piconets surpasses 15, better throughput is achieved by using DH1 transmissions. Figure 3.5.7 also uses dotted lines to represent the improvement achieved through time slot synchronization. When the time slots are synchronized, there is still a point where throughput is larger when transmitting DH5 asymmetric packets. This occurs when there are more than 18 synchronized DH1 interfering piconets present.

When the interferers are DH5 symmetric data packets, similar results can be found. The results from this analysis are shown in Figure 3.5.8. Maximum 2-way throughput can be achieved using symmetric DH5 transmission until there are more than 10 unsynchronized interfering piconets. After this point is surpassed, a larger throughput is achieved using DH5 asymmetric transmissions. However, in this case it isn't very advantageous (in terms of 2-way throughput) to use DH1 transmissions. Once again, shaded lines are used to represent the improvement gained by using time slot synchronization. When time slot synchronization is used, the maximum throughput is always achieved when transmitting DH5 symmetric packets, regardless of the number of interfering piconets. A 2-way throughput of 100 kbps can be achieved with up to 80 interfering piconets present.

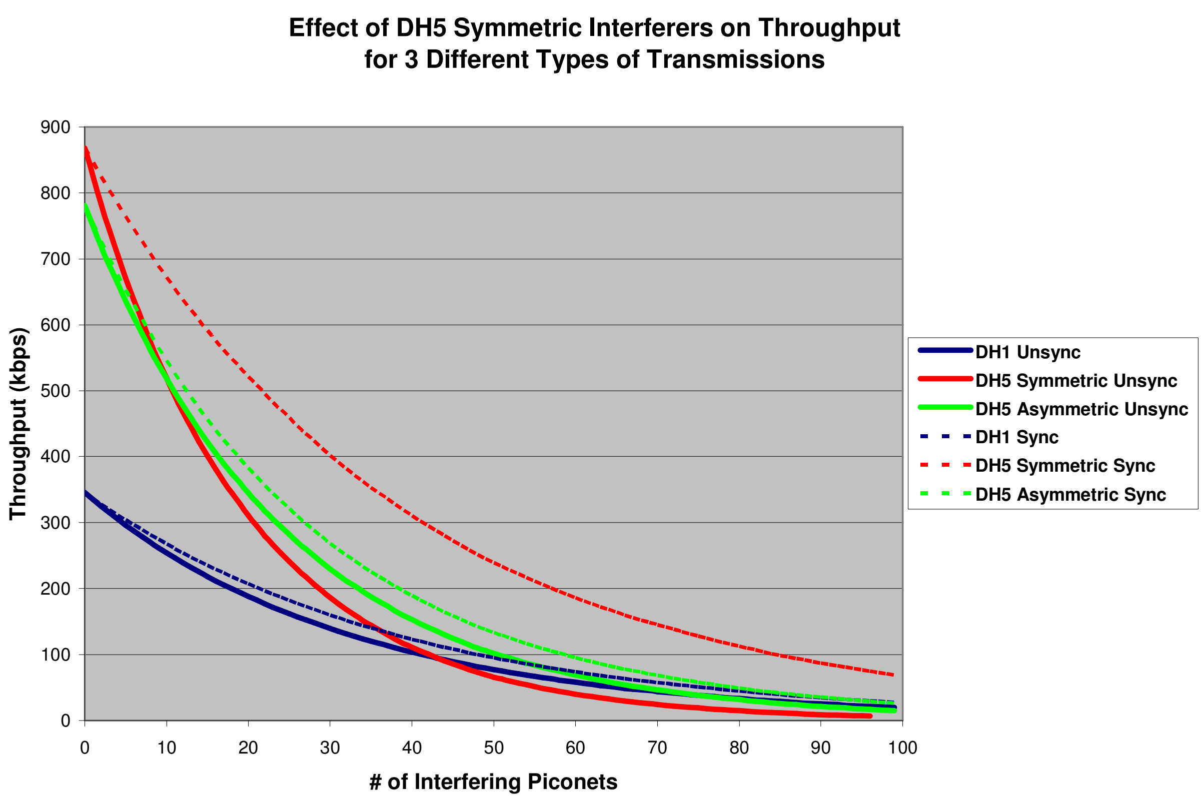


Figure 3.5.8 Effect of DH5 symmetric interferers on throughput for 3 types of transmissions

This section provided some insight into the improvement that can be achieved by changing the data packet size. This section also makes a case for time slot synchronization between piconets. Figure 3.5.8 showed that with time slot synchronization, the maximum throughput can be achieved when all piconets are using

DH5 symmetric transmissions. However, one must keep in mind that when the number of interferers gets to be very large, achieving this throughput will be very inefficient.

Chapter IV: Hardware Testing

4.1 Introduction

The previous two chapters of this thesis were used to evaluate the statistical nature of inter-piconet interference. Many assumptions were made during each of these chapters. The theoretical equations given in chapter two made the assumption that the probability of hopping to a channel at any given point in time was 1 in 79. It was later shown that this was not true for real Bluetooth devices transmitting over finite intervals of time. The third chapter simulated several Bluetooth interference scenarios using the actual Bluetooth frequency hopping algorithm. These simulations were performed under the assumption that the only criterion for interference was that two or more piconets had to dwell on the same frequency channel during the same time slot. It was also assumed that both piconets were transmitting 1-slot symmetric data packets at 100% throughput. This meant that there was a potential for interference in each time slot.

This chapter of the thesis covers the development of an experiment using real Bluetooth hardware to validate some of the simulation results. This meant that the assumptions made during the simulation had to be duplicated in the experiment as best possible. Section 4.2 will discuss the process of setting up the test. Section 4.3 in this chapter will cover the experimental procedure that was followed. And finally, Section 4.4 will review some experimental results and compare them to simulated data.

4.2 Experimental Setup

In order to perform an experiment to validate the simulation results, care had to be taken in the setup. Several assumptions were made in the simulation that had to be duplicated in a real test environment. The experiment also had to be set up in way that interference could be measured. There are three subsections covering the setup of this "inter-piconet" interference experiment. Section 4.2.1 covers the conception of the physical test setup. Section 4.2.2 describes how the hardware was configured for testing. And finally, Section 4.2.3 outlines some of the limitations of this test setup.

4.2.1 Conception of Initial Test Setup

The first step in setting up the test was to determine what type of test to run. It was decided that an experiment would be conducted where there were two piconets present in a given area. Specifically, the experiment was set up to test the actual interference from one piconet (interfering piconet) onto another piconet (piconet-of-interest).

After choosing the type of test, the appropriate equipment had to be chosen. Four Bluetooth devices are required to set up the two piconets in this experiment. The main equipment requirement was that the interference had to be measured in one of the nodes in the piconet-of-interest. A Tektronix BPA100 Bluetooth Protocol Analyzer was chosen to operate as this node in the piconet-of interest. The BPA100 has the capability to participate in a piconet as a master or slave. It also has the ability to record and analyze the packet data sent over a Bluetooth piconet. The BPA100 Protocol Analyzer was chosen for this reason. The BPA100 contains a class 1 Bluetooth device and it has been developed according to the Bluetooth Specification v1.1 [1].

The remaining three Bluetooth devices used are Uniwill Bluetooth modules that have been interfaced to USB ports on a PC. The "piconet-of-interest" uses one of these Uniwill devices as its master, while the BPA100 Protocol Analyzer acts as the slave. The "interfering piconet" consists of the remaining two Uniwill devices, one acting as master and the other as the slave. These modules contain the BlueCore™ 2 chip developed by Cambridge Silicon Radio (CSR). The Uniwill BTM2022-1 devices are also class 1 devices and have been developed according to the Bluetooth Specification v1.1 [1].

After deciding on the equipment to be used, care was taken to ensure that the experiment closely matched the assumptions that were made in the simulation. One of the assumptions made in the simulation was that all piconets were transmitting 1-slot data packets in all time slots. At the time, limitations in Bluetooth software and hardware would not enable us to transmit DH1 packets at 100% capacity. In order to transmit in all time slots, we instead decided to use SCO, or in other words voice packets (HV1). These packets are 1-slot packets and they can operate at 100% capacity. The main differences between the HV1 packets and the DH1 packets is that HV1 packets do not transmit using an ARQ scheme and there is increased forward error correction (FEC) in the HV1 packet. This means that lost packets will not be retransmitted and the data in each packet will have an increased chance of being corrected in the event that interference occurs.

Another assumption made in the simulation was that the only criterion for interference was that two or more piconets dwell on the same frequency channel during the same time slot. This means that in the experiment, the interfering piconet's master and slave must transmit a power level large enough to operate below the minimum specified carrier-to-interferer (C/I) ratio in the piconet-of-interest's receiver. Since the power levels are adjustable in the Bluetooth modules, the geometry of the physical setup was not that important. A diagram of the setup that was used is shown below in Figure 4.2.1.

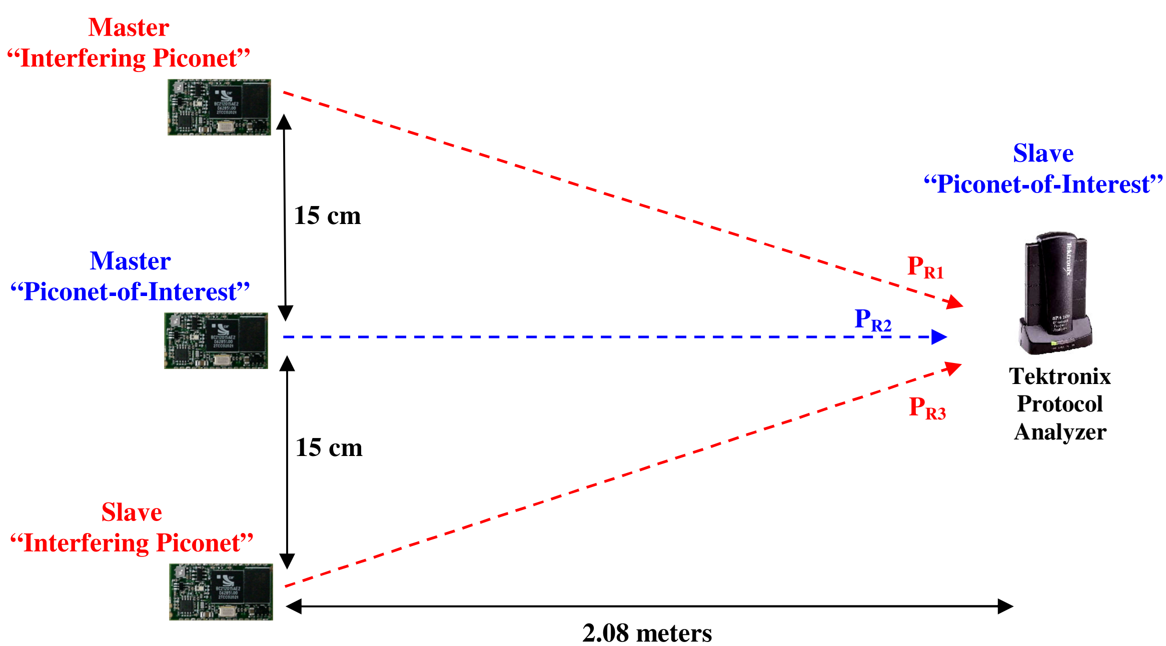


Figure 4.2.1 Physical setup of "Inter-piconet" interference experiment

The requirement in the Bluetooth specification for avoiding co-channel interference is that the C/I ratio be a minimum of 11 dB. In the experiment, the level of the interferers must be large enough to ensure that FEC does not correct an instance of interference, thus making the collision undetectable. However at the same time, the interfering piconet must not exceed a power level where adjacent channel interference begins to occur. The Bluetooth specification requires that Bluetooth radios perform with a C/I ratio of 0 dB with respect to an adjacent channel interferer [1]. This meant that the interferer power had to be chosen such that the C/I ratio would actually occur somewhere between 0 and 11 dB. For this experiment, the interfering piconet's powers were adjusted so that the C/I ≈ 1 dB at the Protocol Analyzer's receiver. This means that according to Figure 4.2.1, P_{R1} and P_{R3} have to be 1 dB less than P_{R2}. The Bluetooth specification states that interference performance should be "measured with the wanted signal 10 dB over the reference sensitivity level." Since the receiver sensitivity is specified to be -70 dBm, the receive power (P_{R2}) from the master in the "piconet-of-interest" were to be set at -60 dBm. The receive powers (P_{R1} and P_{R3}) for the master and slave in the "interfering piconet" were to be set at -61 dBm. The process of setting these power levels is covered in the next section, Section 4.2.2.

4.2.2 Configuring Hardware

After deciding on the physical setup for the experiment, care was taken to set up the power levels for the Bluetooth devices. The power levels were adjusted using two utilities included in CSR's Bluelab Development Kit. One of the utilities used is called "BlueTest." This utility allowed each Bluetooth module to be set into test mode. A picture of this utility is shown below in Figure 4.2.2. The "BlueTest" utility gives the user the capability of testing different External and Internal power amplifier settings (EPA and IPA respectively). Notice that the EPA, IPA, and frequency channels are adjustable inputs in the BlueTest utility shown in Figure 4.2.2.

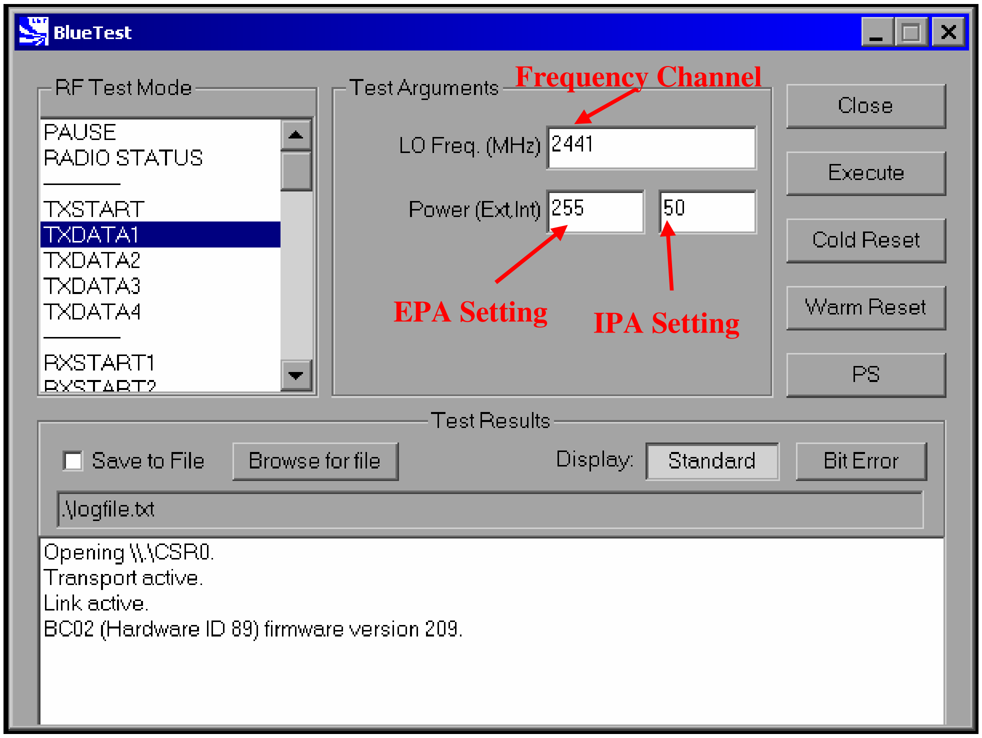


Figure 4.2.2 "BlueTest" utility used for testing different power amplifier values

In setting up this experiment, the single channel test mode was used. This test mode transmits a DH1 packet every 1250 µs at a single frequency specified by the user. It is important to note that in 1-slot transmission, each time slot spans 625 µs. Since a Bluetooth transmission alternates transmit/receive every time slot, then each end of a piconet will be transmitting once every 1250 µs. This single channel test mode is labeled as "TXDATA1" in Figure 4.2.2.

The desired output powers of the three Uniwill BTM2022-1 devices described at the end of Section 4.2.1 were calibrated while measuring the power levels at the location of the BPA100 Protocol Analyzer. To view the relative power level being received at the Protocol Analyzer, an Agilent 8594E Spectrum Analyzer was equipped with a ¼ wave monopole antenna. During calibration, the monopole antenna was put in place of the BPA100 Protocol Analyzer. The physical setup used during this calibration is shown in Figure 4.2.3.

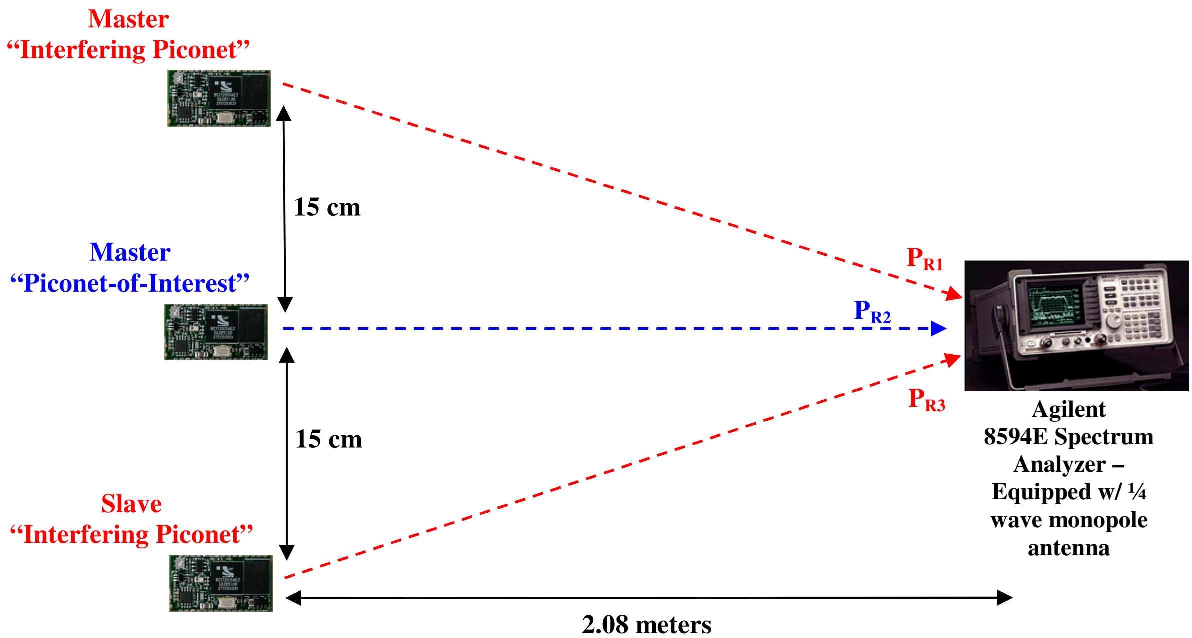


Figure 4.2.3 Physical setup used to calibrate received powers coming from three Uniwill BTM2022-1 devices

Three "BlueTest" utilities were opened in single channel test mode all at once, one for each of the three Uniwill devices. The "master interfering piconet" and "slave interfering piconet" were set to frequencies 2438 and 2444 MHz respectively. The "master piconet-of-interest" was set to frequency 2441 MHz. After setting the frequency channel for each of the three devices, the received power levels had to be calibrated. All three signals were viewed simultaneously using the 8594E Spectrum Analyzer centered at 2441 MHz with a span of 10 MHz. Because the signal from each Bluetooth device was sent only once every 1250 μs, the Spectrum Analyzer had to be set to "Max hold" in order to view the signals. A screenshot was captured from the Spectrum Analyzer during calibration using Labview software. This screenshot can be seen in Figure 4.2.4. During the process of calibration, different IPA settings were tested until the desired received power was measured. The EPA value for all three devices remained set at the value 255. Table 4.2.1 shows the BlueTest settings and resulting received power which produced the output in Figure 4.2.4.

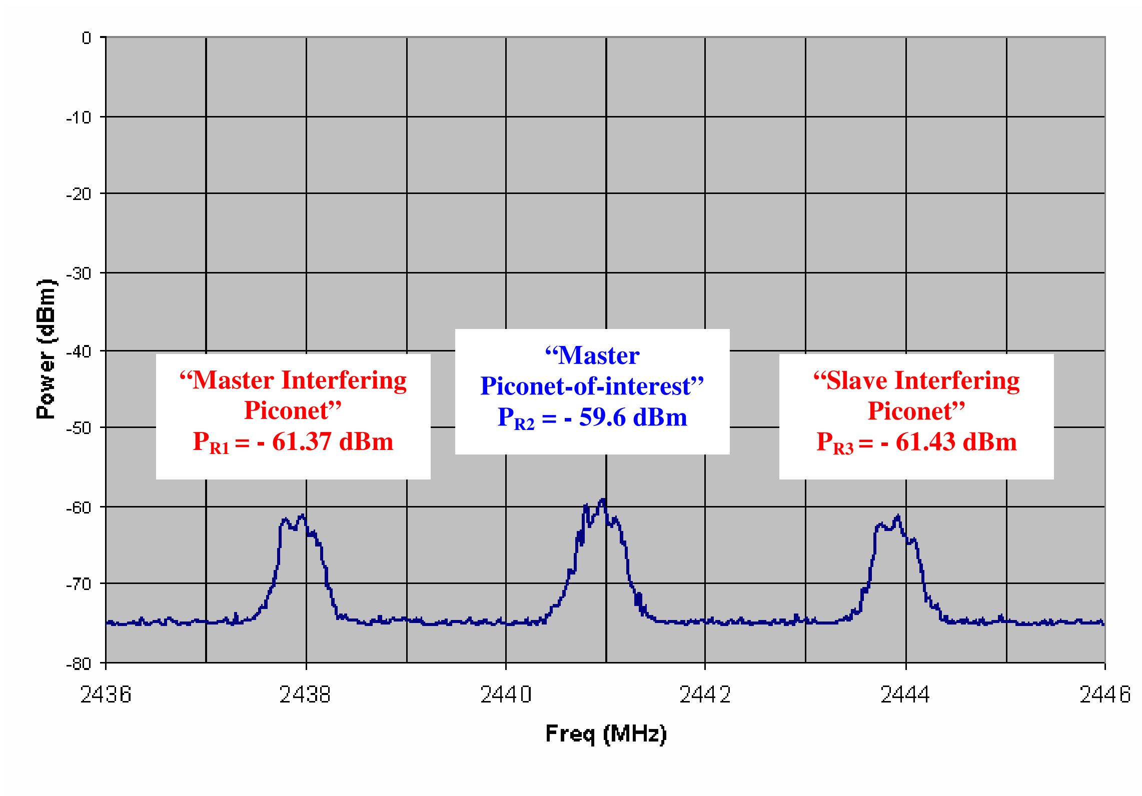


Figure 4.2.4 Spectrum Analyzer data captured during calibration of received powers

	IPA	EPA	Freq during Calibration	Pwr Received at Protocol Analyzer
Master (POI)	20	255	2441 MHz	-59.6 dBm
Master (IP)	20	255	2438 MHz	-61.37 dBm
Slave (IP)	7	255	2444 MHz	-61.43 dBm

Table 4.2.1 "BlueTest" settings used to produce calibrated received powers shown in Figure 4.2.4

After calibrating the power levels, the EPA and IPA values had to be set in the power table for each Bluetooth device. The "Persistent Store Tool" or "PSTool" was used to set these power table values. A picture of this utility is shown below in Figure 4.2.5. Notice that there is only one IPA and EPA setting for frequency band. Also notice that there is only one row in the power table. All other power level settings were removed so that the devices cannot change their power levels during this testing.

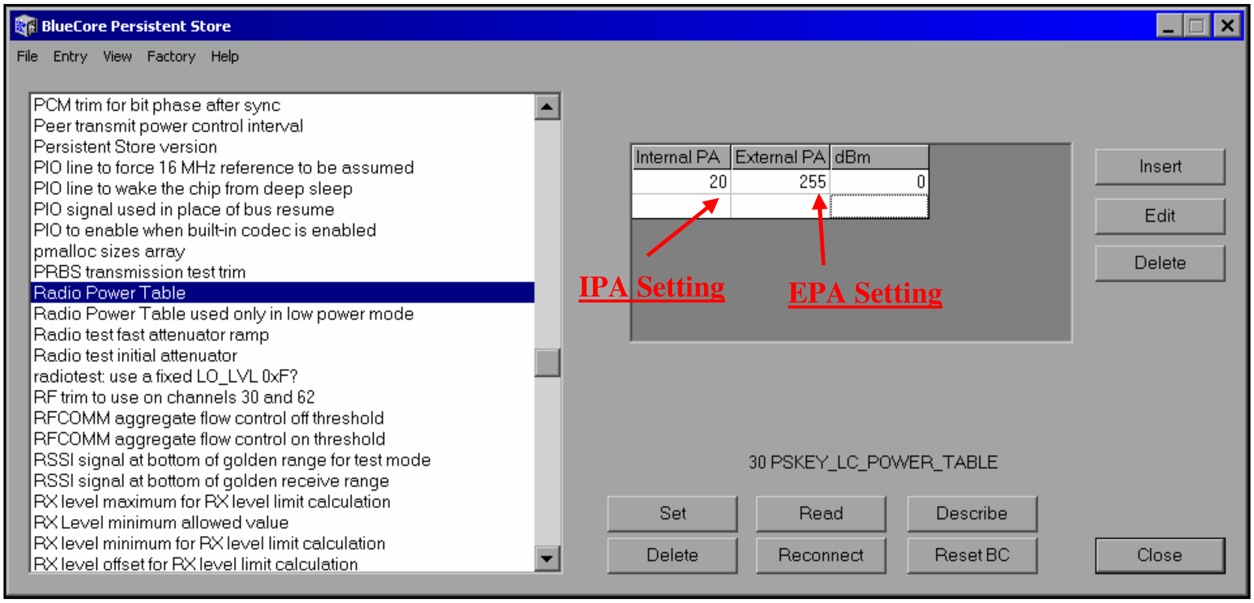


Figure 4.2.5 "PSTool" utility used for setting power amplifier values

4.2.3 Test Setup Limitations

There are a few limitations to the test setup that should be noted. First of all, the actual power levels received at the Protocol Analyzer cannot be detected. The Protocol Analyzer that we have does not have an external antenna input. If it did, then a more precise measurement of received power could be taken. Instead, an antenna connected to the spectrum analyzer had to be placed near the location of the Protocol Analyzer in order to capture the relative power levels of all Bluetooth devices. There is therefore no way to detect the exact C/I ratio at all frequencies and times. Fluctuations in power over time and frequency in this FHSS system may cause an interferer to drop in power relative to the intended signal at times. This may result in expected cases of interference not showing up. Also, the interferer power may increase relative to the intended signal. This will result in a decrease in the C/I. If the C/I ratio falls below the specification for minimum C/I relative to an adjacent channel interferer, then adjacent channel interference may occur.

Another limitation in this test setup is that only some of the interference will be measured. This is because the Protocol Analyzer will only detect errors in its own receiver. Since an ARQ scheme is not employed in HV1 packets, the Protocol Analyzer has no way of detecting whether or not the packets that it transmits are being interfered with. Therefore, only 1-way interference can be measured with this setup.

A third limitation is that by using voice (SCO) packets, only 1-slot transmissions can be experimented with. Due to the current inability to transmit data (ACL) packets at 100% capacity, voice packets had to be used. Voice packets only occupy single time slots during transmission.

And finally the fourth limitation is that HV1 packets use 1/3 rate FEC, meaning that some errors that would normally occur will instead be recoverable.

4.3 Test Procedure

After the test environment was set up, a good test procedure had to be developed. One of the main challenges in developing the test procedure was to find a way to capture the "inter-piconet" clock offset. Without this piece of information, it would be impossible to verify experimental results through simulation. There are three subsections covering the experimental procedure used to test for interference between Bluetooth piconets. Section 4.3.1 outlines the general procedure followed in the experiment. Section 4.3.2 describes in detail the process of capturing the "inter-piconet" clock offset. And finally, Section 4.3.3 shows how this test procedure has been automated.

4.3.1 General Test Procedure

The following general test procedure was developed in order to test for "inter-piconet" interference between two Bluetooth devices:

Step 1 Capture Clock Offset between the masters of two piconets

Step 2 Establish two piconets: "piconet-of-interest" and "interfering piconet"

Step 3	Have both piconets transmit voice packets (HV1)
Step 4	Record packets received by the Protocol Analyzer (Slave of "piconet-of-interest")
Step 5	Run simulation to analyze packet errors in recorded data

4.3.2 Capturing Clock Offset between Piconets

In order to analyze any experimental interference data, it was necessary to develop a dependable method for capturing the clock offset between two piconets. This piece of information, the clock start time of one piconets, and the address of each piconet's master enable the experiment to be recreated in simulation. Capturing the clock offset was the most challenging portion of this experiment. This was accomplished by having both piconet masters connect, one at a time, to the Protocol Analyzer. By connecting to the Protocol Analyzer, the clock offset between it each master could be captured. By comparing the clock offset between each master and the Protocol Analyzer, the actual clock offset between the two masters can be calculated. Figure 4.3.1 below shows a screenshot of Protocol Analyzer data that can be used to calculate the clock offset between the Protocol Analyzer and the master device that it connects to.

The screenshot shown in Figure 3.4.1 actually shows the connection state between a master device and the Protocol Analyzer (slave). Right before the connection is completed, the master sends an FHS packet to the Protocol Analyzer (slave). The slave then responds with an ID packet. The "timetick" of 70,430,245 for this event is the first one circled in the figure. At this point in time, the Protocol Analyzer is outputting the "timetick" values from its own clock. In the next time slot, the master responds with a POLL packet. Notice that the "timetick" of 218,587,500 at this point (identified in the figure by the second "timetick" circled) is a much larger value. This is because upon connection, the Protocol Analyzer begins to display the clock from the master device. So, to calculate the clock of the master relative to the clock in the Protocol Analyzer, the "timeticks" about this transition are subtracted. The result is a clock offset of 148,157,255. The master from the other piconet is then connected to the Protocol Analyzer to determine their clock offset as well. And finally, the two resulting clock offsets are subtracted from one another to get the clock offset between the masters of the two dueling piconets.

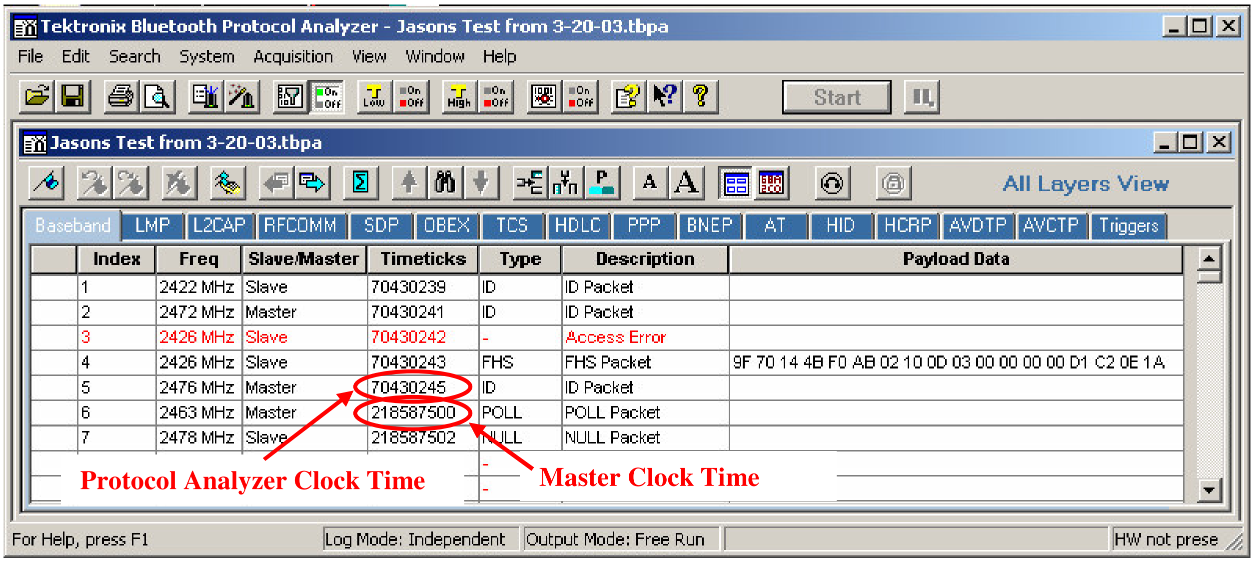


Figure 4.3.1 Calculating clock offset between Protocol Analyzer and master device

4.3.3 Automating the Test Procedure

It is important to note that the Bluetooth specification allows for a clock drift of up to 20 ppm [1]. Because the Bluetooth clock operates at 3200 cycles per second, this mean that any clock offset captured may only be accurate for a small amount of time. In fact if the drift is equivalent to the maximum allowed by the specification, the clock will drift one position every 16 seconds. Therefore it is important to automate the test procedure to operate very quickly.

Tom Rondeau wrote an application for this automated experiment. The application performs the following procedure:

Step 1	Connect Master from "interfering piconet" to Protocol Analyzer
Step 2	Disconnect devices
Step 3	Connect Master and Slave to form "interfering piconet"
Step 4	Connect Master from "piconet-of-interest" to Protocol Analyzer
Step 5	Have "interfering piconet" begin transmitting HV1 packets

Step 6 Have "piconet-of-interest" begin transmitting HV1 packets

Step 7 Transmit for 60 seconds then disconnect

Keep in mind that any connection to the Protocol Analyzer will cause it to record the packets sent back and forth. This means that in steps one and two above, the Protocol analyzer will automatically capture the connection states necessary to calculate the clock offsets. Also in step four when the "piconet-of-interest" is formed, all communication between the master and the Protocol Analyzer is being stored. In step five, the interference is turned on. Therefore when the "piconet-of-interest" is turned on in step six, interference is already being applied. Sixty seconds of interference data is then collected. The interface for this program developed by Tom Rondeau is shown below in Figure 4.3.2.

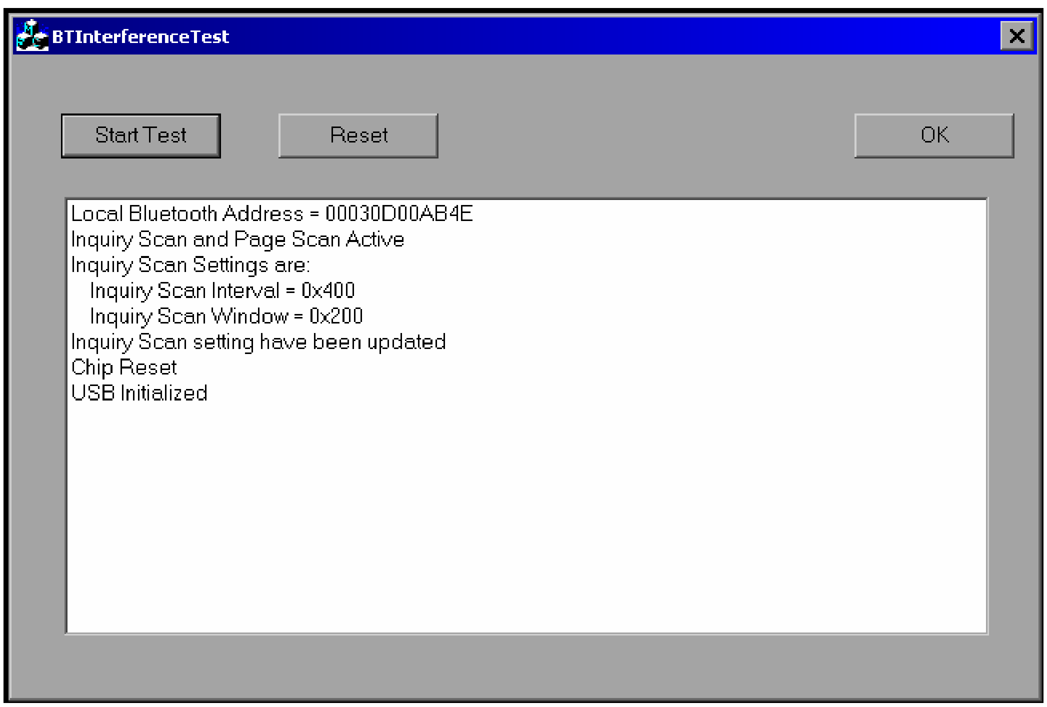


Figure 4.3.2 Windows interface for Bluetooth Interference Test written by Tom Rondeau

Before running the automated test, the Protocol Analyzer had to be set in "Piconet Mode." Then three instances of the program shown in Figure 4.3.2 were run simultaneously, one for each of the three Uniwill Bluetooth devices. From this point on, the procedure was automated. The next section of this chapter will involve an analysis of the resulting data.

4.4 Analysis of Experimental Results

After setting up the experiment and running each test, it was important to take time to analyze the results. This section of the chapter focuses on this analysis and has been separated into three subsections. Section 4.4.1 discusses the output of the Protocol Analyzer and describes how the errors were counted. Section 4.4.2 shows how the experimental data was verified by simulated data. And finally, Section 4.4.3 details the important experimental observations that were made.

4.4.1 Counting Errors in Experimental Data

After running the tests, the experimental data had to be sorted out in order to count the number of errors that had occurred. The Protocol Analyzer represents errors in a variety of different ways. There are two main groups of errors that were counted. The first group of errors is the recoverable errors. These are errors that occurred, but were corrected due to the 1/3 rate FEC. A large number of these errors were observed with no interferer present, and their cause cannot be determined. The important errors to be considered here are the non-recoverable errors. These are made up of access errors, payload errors, packet header errors, and also recoverable errors with corrupt data packets. Some recoverable errors show up with corrupt data packets because unlike data packets (DH1), the voice packets (HV1) do not employ a cyclic redundancy check (CRC).

After running each test, the errors were counted in the Protocol Analyzer output. The output was exported to a .csv file and then imported into Microsoft Excel. Five-second spans (8000 transmissions) were analyzed at a time using an Excel spreadsheet. This spreadsheet used several built-in functions to count the different types of errors existing in the data.

4.4.2 Verifying Results through Simulation

Care was taken in selecting these 5-second spans from the 60-second experimental data file. This was because each device's clock can drift up to one time tick every 16 seconds as described in Section 4.3.3. And depending on which way the frequencies of the two device's crystals are drifting, the two clocks could either drift together or drift apart. This means that the clock offset could vary at an even faster rate (up to once every 8 seconds.) With a clock offset that is constantly changing and while transmitting 1-slot packets, it is much less likely to detect the unsynchronized case. This is because the unsynchronized case is much less likely to occur than the synchronized case. Therefore, it was decided that only experimental cases where interference occurred in a synchronized manner would be analyzed. This experimental data was verified as synchronized by comparing the experimental and simulated hopping sequences using an Excel spreadsheet.

In order to verify these results, a simulation for two synchronized piconets was run. The clock start time and inter-piconet clock offset were collected from the Protocol Analyzer output and used as inputs to the simulation program. The addresses of the two master devices in the experiment were also used as inputs. The simulation had to be slightly modified to count only the errors occurring in the piconet-of-interest's slave receiver (this is because the experiment can only record one-way interference). And the number of iterations (or number of transmissions simulated) was set at 8000 as well to match the experimental dataset.

The results from several experiments were verified through the use of this simulation program. Twenty different tests were run consecutively to produce these results. Two of these tests were discarded because the two connection transitions necessary in deriving the clock offset did not appear on the Protocol Analyzer. The results from the remaining eighteen experiments have been tabulated below in Table 4.4.1. A different test is represented in each row of the table. Notice that the number of non-recoverable errors in the measured data is very close to the number of errors due to interference that were simulated. The average error rate in the experimental data was 1.571%. This is within

99.2% of the average simulated error rate of 1.583%. However, the simulation results in Table 3.4.1 and the theoretical Equation 2.1 show that two piconets with synchronized time slots should have an average collision rate of 1.27%. This value is the theoretical average for a sample of collision rates taken with random clock offsets. Unfortunately, the sample set taken did not produce this same value. Perhaps taking a larger sample set would produce a number closer to the theoretical average.

Clk Offset	Clk Start Time	Measured Data			Simulated Data	
		# Recoverable Errors	# Non-recoverable Errors	Collision %	# Errors	Collision %
182898	218448220	43	91	2.275	90	2.25
182896	218556220	56	0	0	0	0
347468	218933568	43	62	1.55	65	1.625
-5818340	224925964	73	60	1.5	68	1.7
4036	218313636	56	24	0.6	0	0
4062	220778332	76	78	1.95	84	2.1
4179	224145168	81	147	3.675	152	3.8
4206	226736824	65	85	2.125	79	1.975
4194	229232308	66	106	2.65	107	2.675
4376	235007812	74	120	3	122	3.05
-5258	218363524	82	50	1.25	49	1.225
-5252	218980036	95	62	1.55	68	1.7
-962	221748548	87	55	1.375	59	1.475
-876	218875788	93	37	0.925	46	1.15
-872	219205840	105	29	0.725	49	1.225
5062	218712868	75	12	0.3	0	0
-972	220830688	118	47	1.175	42	1.05
-882	218401896	84	66	1.65	60	1.5
Averages		**76.2**	**62.8**	**1.571**	**63.3**	**1.583**

Table 4.4.1 Comparison of experimental and simulated interference data for 2 synchronized piconets

Also, the number of recoverable errors measured seemed to be quite large. As mentioned in Section 4.4.1, some recoverable errors were observed with no interferer present. To test for this, the previous experiment was repeated without an interfering piconet present. Eighteen tests were run, resulting in an average of 84 recoverable errors per 8000 transmissions. This is in the same ballpark as the average number of recoverable errors reported with an interfering piconet present (76.2 recoverable errors per 8000 transmissions, as shown in Table 4.4.1). This suggests that most of the recoverable errors showing up in experimental data were not due to inter-piconet interference.

4.4.3 Experimental Observations

While conducting the experiment, several observations were made. The first observation was that the clock offset between piconets was constantly drifting. This could be seen in the initial experimental data when compared to the simulated output. At times, it was difficult to tell whether or not the clock offset captured from the Protocol Analyzer was correct. In order to directly compare experimental and simulated data, it was important to know exactly what the clock offset was in the data. Also it had to be determined whether the two piconets' time slots were in the synchronized or unsynchronized case. By comparing simulated output to actual data, the clock offset and the state of synchronization were discovered.

Another important observation based upon the clock drift was that the unsynchronized case was difficult to capture over a five second span. The two piconets drift in and out of the unsynchronized case so fast that it was difficult to detect their state in the resulting data. There were a few instances of time where interference occurred due to interferers in the same time slot and also the adjacent time slot. In other words, there were very few small time spans where the unsynchronized case could be observed. It should be noted that in the future, if we can get multi-slot data packets to transmit at 100%, then the unsynchronized case could be detected better. This is because multi-slot data packets span multiple time slots, and they will therefore spend more time in the unsynchronized case then 1-slot packets do.

Another observation from experimental testing was that at times adjacent channel interference would occur; yet at other times a predicted instance of interference would not occur at all. This suggested that the relative power between "interfering piconet" and "piconet-of-interest" transmissions might have been fluctuating over frequency and time. It is apparent from the data in Table 4.4.1 that at times the number of errors seen in the experiment was higher and other times lower than the number of errors simulated. However, the averages of all eighteen experiments matched very close in the end, suggesting that these effects balanced each other out.

And finally, the simulation program proved to be a very useful tool in finding out the cause of each error that showed up. The simulation program outputs a list of data that can be used to explain the cause of almost every occurrence of an error. In this particular experiment, I was able to verify some instances of adjacent channel interference and interference coming from an adjacent time slot.

Chapter V: Conclusions and Future Work

5.1 Introduction

This chapter concludes this study on “inter-piconet” interference. It has been divided into three major sections. Section 5.2 highlights the major conclusions from this thesis. Section 5.3 summarizes the contributions that have been made. And finally, Section 5.4 of this chapter will describe the future work that could stem from this research.

5.2 Conclusions

This section has been divided into three subsections covering the important conclusion categories. Section 5.2.1 contains a review of the research goals established at the beginning of the thesis. Section 5.2.2 describes some of the conclusions that were made according to simulation results. And finally, Section 5.2.3 describes the conclusions made as a result of experimentation.

5.2.1 Review of Research Goals

At the outset of this thesis research, three major goals were described. The first, and most important, goal was to gain a better understanding of “inter-piconet” interference. When this research was initialized, it was very rare to come across any related literature. What was available was mostly theoretical in nature, making the assumption that the probability of hopping to a channel was 1 in 79 at any given point in time. After studying the Bluetooth frequency hopping algorithm, it became apparent that this was not the case over short periods of time. This realization is what led to the second goal. The second goal was to simulate different aspects of “inter-piconet” interference by actually implementing the frequency hopping algorithm in software. Several programs were written in C++ to analyze different interference scenarios. And the final research goal was to verify some of the results through experimentation with actual Bluetooth hardware.

5.2.2 Simulation Conclusions

The following major conclusions were made as a result of running several different "inter-piconet" interference simulations:

I. The collision rate for a given instant in time is not truly random in nature. In fact, in Section 3.3.1 and 3.3.2 of this thesis, it was shown that the collision rate varies as a function of time as well as "inter-piconet" clock offset.

II. The theoretical statistics (equations 2.1 and 2.2) represent the mean of the collision rate as a function of "inter-piconet" clock offset. This can be seen by comparing the theoretical mean to the mean of the graph in Figure 3.3.3.

III. Collision rate is periodic as a function of "inter-piconet" clock offset. This was shown in Figures 3.3.3 and 3.3.4. It was also concluded that there is often a substantial difference in collision rate from one "inter-piconet" clock offset to the next. This is also shown in Figure 3.3.4. This suggests that interference improvement may be achieved by adjusting the master clock for a piconet.

IV. Forcing synchronization between piconets won't always result in a significant decrease in interference as suggested in the literature [5, 6]. In fact, an example of synchronization providing very little decrease in interference can be seen in Figure 3.3.8. However, if the "inter-piconet" offset is adjusted by one time slot, then the collision rate drops significantly as shown in Figure 3.3.9. This suggests that a combination of time slot synchronization and "inter-piconet" offset control results in better improvement than time slot synchronization alone.

V. The mean and variance of collision rate statistics as a function of "inter-piconet" clock offset will increase with the number of interfering piconets present. This is explained in Section 3.4 and is shown Table 3.4.1. Also shown in Table 3.4.1 is that the variance is much larger in the case of synchronized time slots.

VI. 1-slot data packets receive less interference from interfering piconets transmitting larger data packets. This is because double threats occur less frequently when the interfering piconets are transmitting these large data packets. This effect was explained in detail in Section 3.5.2.

VII. 5-slot data packets are more vulnerable to interfering piconets transmitting smaller data packets. An interfering piconet transmitting the identical type of 5-slot packet causes the least amount of interference with 5-slot transmission. This is shown graphically in Figures 3.5.5 and 3.5.6.

VIII. Throughput improvement may be achieved in the presence of interference by simply changing the packet size in the piconet-of-interest. Certain packets sizes can achieve higher throughput depending on the type and number of interferers. However, there are times where throughput is achieved at the cost of poor efficiency. Figures 3.5.7 and 3.5.8 each show throughput plots for different transmission types against a known type of interference.

5.2.3 Experimental Conclusions

The following experimental conclusions were made:

I. Experimental and simulated results matched up very well. The average collision rate for the eighteen tests run was within 99% of the simulated rate. These results were compared in Section 4.4.1.

II. A case was identified in the experimental data where the collision rate dropped significantly between consecutive "inter-piconet" clock offsets. The collision rate results for each of these offsets can be seen below in Figure 5.2.1. Notice in figure that the clock offset in one row is 182898 and in the second row it has shifted two clock cycles (or one time slot) to 182896. As the clock offset shifted by one time slot, the collision statistics dropped dramatically. This verifies the results from the simulation. The simulation conclusion numbered "III" in Section 5.2.2 describes the same effect. The clock from one piconet can be varied to improve interference performance.

Clk Offset	Clk Start Time	Measured Data			Simulated Data	
		# Recoverable Errors	# Non-recoverable Errors	Collision %	# Errors	Collision %
182898	218448220	43	91	2.275	90	2.25
182896	218556220	56	0	0	0	0

Figure 5.2.1 Improvement of collision statistics from one "inter-piconet" clock offset to the next

III. The "inter-piconet" clock offset is constantly changing due to the clock drift in the master of each piconet. This means that over long periods of time, the clock offset will cycle through several values, possibly causing the interference statistics to approach the mean. This constant drift in "inter-piconet" clock offset would require multiple piconets to continually correspond with one another in order to synchronize time slots.

5.3 Contributions

The following contributions were made in the area of "inter-piconet" interference:

I. A simulation was developed using the Bluetooth frequency hopping algorithm in order to research various aspects of "inter-piconet" interference. The results from several different interference scenarios have been presented that have been developed using this simulation.

II. A method was developed in which experimental instances of "inter-piconet" interference could be verified against simulated data. Actual measured data was verified against simulated data using this method.

III. The effect of different types of interferers on a particular type of transmission was analyzed. A comparison was given for collision rates produced by interferers with different variables such as asymmetric, symmetric, 1, 3, and 5-slot transmissions. This comparison also included both the synchronized and unsynchronized cases. These results can be seen in Sections 3.5.2 and 3.5.3. Previous literature has not been found that contains such an analysis.

IV. A throughput comparison was given for a fixed type of interferer against different types of transmission (asymmetric, symmetric, 1, and 5-slot). This was done for both the synchronized and unsynchronized cases. The results can be found in Section 3.5.4. Previous literature has not been found that contains such an analysis.

V. A detailed explanation of the Bluetooth hopping algorithm was given in Section 3.3.3. This section of the thesis showed that specific “inter-piconet” clock offsets exist between two piconets for which zero interference will occur. These are known as the best-case interference scenarios. Likewise, a worst-case interference scenario was described that is one offset value away from the best-case. Examples of these cases were shown in simulation (Figures 3.3.8 and 3.3.9) and also through experimentation (Figure 5.2.1). Based upon this particular research, it was determined that the synchronization of time slots in addition to the control of “inter-piconet” clock offsets will result in the best interference performance.

5.4 Future Work

The research conducted during the completion of this thesis has produced many new opportunities for future work. The bulk of the work reputed in this thesis was on the simulation of several different “inter-piconet” interference scenarios. Only the simplest of those scenarios was actually tested experimentally. A lot of the future work involves continuing to conduct experiments. First of all, it would be beneficial to repeat the same experimentation in an effort to see if taking many more samples will produce the theoretical or mean collision rate. It would also be useful to repeat the experiments using actual data packets instead of the voice packets that were used. The challenge in this is to enable a piconet to transmit data packets at 100% capacity. This will add the ability to experimentally verify the effect of different packet sizes. Larger packet sizes will also make it easier to detect cases where piconets have unsynchronized time slots. Another future area for experimentation would be to conduct tests with multiple interfering piconets.

Another area for future research would be to conduct a tradeoff study for implementing synchronization of time slots and controlling the “inter-piconet” clock offset. With time slot synchronization and a small number of piconets present, zero interference is definitely achievable. Controlling clock offset will also cause the adjacent channel interference to be reduced. The tradeoff however, is that it will take communication between multiple piconets in order to have synchronization. In addition, some overhead will be required in order to determine what the optimum clock offset is.

References

[1] “Specifications of the Bluetooth System-Core,” Version 1.1, http://www.bluetooth.org/specifications.htm

[2] Manika Kapoor, and Dinesh Anvekar, “Frequency lookahead and link state history based interference avoidance in wireless picocellular networks,” *IEEE International Conference on Personal Wireless Communications (ICPWC 2000),* Hyderabad, India, December 2000.

[3] Y. Lim, J Kim, S. Min, J. Ma, “Performance Evaluation of the Bluetooth-Based Public Internet Access Point,” *Proceedings from the 15th International Conference on Information Networking (ICOIN 2001),* pg. 643.

[4] C. de Morais Cordeiro, D. Sadok, D. P. Agrawal, “Piconet Interference Modeling and Performance Evaluation of Bluetooth MAC Protocol,” *Proceedings of IEEE GLOBECOM 2001,* San Antonio, TX, USA, November 2001.

[5] Amre El-Hoiydi, "Packet Error Rate due to Interference Between Bluetooth Networks – Probabilistic Upper Bound and Simulation Results", *2001 Virginia Tech MPRG Symposium on Wireless Personal Communication*, Blacksburg, USA, pp. 23-30, June 2001

[6] Amre El-Hoiydi, “Interference Between Bluetooth Networks-Upper Bound on the Packet Error Rate", *IEEE Communications Letters*, vol. 5, issue 6, June 2001, pp. 245-247.

[7] D’Souza, Mark, “Residential Microwave Oven Interference on Bluetooth Data Performance,” [This thesis has not yet been published].

[8] Jaap C. Haartsen, Stefan Zürbes, “Frequency Hop Selection in the Bluetooth Radio System,” *IEEE 7th Symposium on Spread-Spectrum Technology and Applications*, Prague, Czech Republic, Sept 2-5, 2002.

[9] A.K. Arumugam, A.R. Nix, P.N. Fletcher, S.M.D. Armour, B.S. Lee, “Scenario Driven Evaluation and Interference Mitigation Proposals for Bluetooth and High Data Rate Bluetooth Enabled Consumer Electronic Devices,” *IEEE Transactions on Consumer Electronics*, Vol. 48, No. 3, August 2002, pp. 754-764.

Vita

Jason E. Ballagh was born on May 18, 1978 in Laurel, Maryland. He graduated in 1996 from St. Vincent Pallotti High School in Laurel, Maryland. While an undergraduate at Virginia Tech, he CO-OPed for Ericsson in North Carolina. He graduated Cum Laude in electrical engineering in May 2001. After graduation, Jason worked as a summer intern for Northrop Grumman in College Park, Maryland. He returned to Virginia Tech in the fall of 2001 to begin working towards his Master's Degree. In the summer of 2002, Jason's interests in wireless communications led him to the Center for Wireless Telecommunications (CWT) at Virginia Tech. There he joined the Bluetooth Lab, where he worked on applications for Bluetooth and began his study on inter-piconet interference. Jason will have completed the requirements for his Master's Degree in May 2003.