

Air Force Institute of Technology

AFIT Scholar

Theses and Dissertations

Student Graduate Works

9-2004

Device Discovery in Frequency Hopping Wireless Ad Hoc Networks

Brian S. Peterson

Follow this and additional works at: <https://scholar.afit.edu/etd>



Part of the [Digital Communications and Networking Commons](#)

Recommended Citation

Peterson, Brian S., "Device Discovery in Frequency Hopping Wireless Ad Hoc Networks" (2004). *Theses and Dissertations*. 3900.

<https://scholar.afit.edu/etd/3900>

This Dissertation is brought to you for free and open access by the Student Graduate Works at AFIT Scholar. It has been accepted for inclusion in Theses and Dissertations by an authorized administrator of AFIT Scholar. For more information, please contact AFIT.ENWL.Repository@us.af.mil.



DEVICE DISCOVERY IN FREQUENCY HOPPING WIRELESS AD HOC
NETWORKS

DISSERTATION

Brian S. Peterson, Major, USAF

AFIT/DS/ENG/04-06

DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY

AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

Research sponsored primarily by the Air Force Office of Scientific Research and secondarily by the Air Force Research Laboratory, Human Effectiveness Directorate, Air Force Materiel Command, USAF. The United States Government is authorized to reproduce and distribute reprints notwithstanding any copyright notation thereon. The views and conclusions contained in this dissertation are those of the author and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of the Air Force Office of Scientific Research, the Air Force Research Laboratory, Department of Defense, or the United States Government.

AFIT/DS/ENG/04-06

DEVICE DISCOVERY IN FREQUENCY HOPPING WIRELESS
AD HOC NETWORKS

DISSERTATION

Presented to the Faculty
Graduate School of Engineering and Management
Air Force Institute of Technology
Air University
Air Education and Training Command
In Partial Fulfillment of the Requirements for the
Degree of Doctor of Philosophy

Brian S. Peterson, B.S., M.S.
Major, USAF

June 2004

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

Abstract

This dissertation develops a method for efficient discovery of wireless devices for a frequency hopping spread spectrum, synchronous, ad hoc network comprised of clustered sub-networks. Bluetooth serves as a reference protocol. An analytical model characterizing the interference to network traffic by inquiring devices is developed and demonstrates that interference becomes significant with multiple inquirers. A mathematical model describing the probability mass function of packet error rates between arbitrary pairs of networks is used to show that the interference is substantial and that packet collision avoidance methods are needed. The interference model illustrates the superiority of a proposed collision avoidance method, the Avoidance Forward Inspection Technique, compared to an existing method. The Bluetooth discovery time probability distribution is rigorously derived for all Bluetooth discovery methods described in the specification. Additionally, an alternative discovery method used in industry is fully characterized. All analytical models are compared to simulation models and to measured data when available. Three scatternet outreach methods are developed and compared. The two methods that actively inquire at random intervals yield lower goodput, increased mean packet delay, consume more power, and cause significant discovery delays compared to the passive outreach approach. Hence, it is recommended that scatternets use a passive outreach method.

Acknowledgements

I would like to thank several people for their support and assistance in this research. First and foremost is my advisor, Dr. Rusty Baldwin. While keeping a watchful eye to prevent me from straying too far off course, the leeway he gave me to investigate paths that were of interest me made this research an enjoyable and rewarding experience. His guidance ensured that the research is well supported and statistically significant. I would also like to thank his multiple red pens that surrendered their lives in attempting to make the presentation of the research clear and concise. I would also like to thank Dr. Jeff Kharoufeh for his demand for meticulous precision in stochastic variable and event definitions, terminology, and derivation. His high standards significantly raised the quality of the research and the papers derived from it. Thanks also go to Dr. Rick Raines and Dr. Mike Temple for their helpful comments on the drafts of this dissertation.

I would also like to thank Mr. Tim Lacey in the AFIT Wireless Laboratory, Mr. Charlie Powers, and Ms. LaVonne Allen for their logistical support in obtaining the space, equipment, and supplies needed for the research. Mr. Tom Kundert, Director of Intellectual Property Law, Air Force Material Command, deserves great thanks in preparing the classified patent associated with the classified portion of the research not presented in this dissertation. Additionally, I would like to thank the appropriate contacts at the National Security Agency who sponsored the research, providing financial support while leaving great leeway in the emphasis of the research.

Finally, I would like to thank, and dedicate this dissertation to, my wife who has stood by me through all of my endeavors in my academic and military career (despite considering much of it too “geeky” for normal human interest).

Brian S. Peterson
Sep 2004

Table of Contents

	Page
Abstract	iv
Acknowledgements	v
List of Figures	xi
List of Tables	xiv
List of Abbreviations	xv
I. Introduction	1
1.1 Background	1
1.2 Research Objectives	1
1.3 Document Overview	2
II. Background and Literature Survey	4
2.1 Clustered Synchronous Ad Hoc Frequency Hopping Spread Spectrum Wireless Networks	4
2.1.1 Clustered Networks	4
2.1.2 Ad Hoc Networks	6
2.1.3 Synchronous Frequency Hopping Spread Spec- trum	6
2.2 The Bluetooth Protocol	9
2.2.1 Standby state	10
2.2.2 Connection state	10
2.2.3 Inquiry Substate	16
2.2.4 Inquiry scan/Inquiry response substates	20
2.2.5 Page/Page scan substates	24
2.2.6 Low Power States	24
2.2.7 Power consumption	24
2.3 Bluetooth Packet Error Rate	24
2.4 Bluetooth Inquiry Time	27
2.5 Bluetooth Scatternets	28
2.5.1 Scatternet Organization	29
2.5.2 Scatternet Scheduling	31
2.5.3 Intra-piconet Scheduling	35
2.5.4 Outreach	35
2.5.5 Collision Avoidance	36
2.6 Summary	40

	Page
III. Analysis Techniques	41
3.1 Introduction	41
3.2 Stochastic Analysis	41
3.2.1 Law of Total Probability	42
3.2.2 Random Variable Addition	42
3.3 Probability Distribution Comparisons	42
3.3.1 Cramer von-Mises Test	43
3.3.2 Kolmogorov-Smirnov Test	43
3.4 Summary	43
IV. Research Objectives and Methodology	45
4.1 Research Objectives	45
4.2 System Boundaries and Assumptions	47
4.2.1 Baseband Layer	47
4.2.2 Propagation Delay	48
4.2.3 Standard Configurations	48
4.2.4 Scatternet Organization	48
4.2.5 Scatternet Scheduling	48
4.3 System Performance Metrics	49
4.3.1 Discovery Time	49
4.3.2 Throughput/Goodput	50
4.3.3 Mean Delay	50
4.3.4 Packet Error Rate	50
4.3.5 Buffer Size	51
4.3.6 Packet Generation Time pmf	51
4.4 System Parameters	51
4.4.1 Network Topology	51
4.4.2 Packet Distribution	52
4.4.3 Outreach Methods	53
4.4.4 Packet Length	54
4.4.5 Collision Avoidance	55
4.4.6 Range	55
4.4.7 Bridge Dwell Time	55
4.4.8 Inquiry Timing	56
4.4.9 Inquiry Scan Type	56
4.4.10 Device Arrival Time	57
4.5 Performance Metric and Parameter Characterization	57
4.6 Performance Metric Improvements	57
4.7 Outreach Method Proposals and Evaluation	58
4.8 Summary	58

	Page
V. Inquiry Substate Interference	59
5.1 Inquiry Interference Probability	59
5.1.1 Direct Interference Probability	64
5.1.2 Indirect Interference Probability	67
5.1.3 Total Interference Probability	68
5.2 Multiple Inquiry Nodes	69
5.2.1 Direct Interference Probability with m Inquiry Nodes	69
5.2.2 Indirect Interference Probability with m Inquiry Nodes	72
5.2.3 Total Interference Probability with m Inquiry Nodes	73
5.3 Expected Interference	73
5.4 Summary	74
VI. Packet Error Rate (PER)	77
6.1 Packet Error Rate Characterization	77
6.1.1 MSI/SSI Relationship	78
6.1.2 MSI/SSI Relative Change	79
6.1.3 Placement within the MSI/SSI	80
6.1.4 Temporal Packet Relationship	87
6.2 PER Distribution Derivation Necessities	93
6.2.1 Frequency Sharing Rate	93
6.3 PER Derivation	98
6.3.1 PER in T_1	98
6.3.2 PER in T_2	99
6.3.3 PER in T_3	101
6.3.4 PER in T_4	103
6.3.5 PER in T_5	103
6.3.6 PER in T_6	104
6.3.7 PER in T_7	105
6.3.8 PER in T_8	106
6.4 PER Probability Mass Function (pmf)	107
6.5 PER for Multiple Piconets	109
6.6 Goodput Derivation	110
6.6.1 Goodput in T_1	111
6.6.2 Goodput in T_2	111
6.6.3 Goodput in T_3	112
6.6.4 Goodput in T_4	112

	Page	
6.6.5	Goodput in T_5	112
6.6.6	Goodput in T_6	113
6.6.7	Goodput in T_7	113
6.6.8	Goodput in T_8	113
6.7	Goodput Probability Mass Function	114
6.8	Goodput with Multiple Piconets	115
6.9	Summary	115
VII.	Collision Avoidance Techniques	117
7.1	Avoidant Forward Inspection Technique (AFIT)	118
7.2	Analysis Necessities	120
7.2.1	Time Slot Frequency Sum (TSFS)	120
7.3	Goodput Probability Mass Function	122
7.3.1	Goodput in T_1	122
7.3.2	Goodput in T_2	123
7.3.3	Goodput in T_3	124
7.3.4	Goodput in T_4 and T_5	125
7.3.5	Goodput in T_6	126
7.3.6	Goodput in T_7	126
7.3.7	Goodput in T_8	127
7.4	Avoidance Technique Comparisons	128
7.5	Goodput in Multiple Piconets	132
7.6	Collision Avoidance with Multi-slot Packets	133
7.7	Summary	138
VIII.	Inquiry Time	140
8.1	Inquiry Interference Probability	140
8.2	Random Variable and Event Definitions	141
8.3	Components of Inquiry Time pdf	148
8.4	Distribution of Back-off Period Completion time, T_P	154
8.5	Conditional Inquiry Time Probability Densities	158
8.5.1	Conditional density given E_{H1}	160
8.5.2	Conditional density given E_{H2a}	166
8.5.3	Conditional density given E_{H2b}	168
8.5.4	Conditional density given E_{H3}	169
8.5.5	Conditional density given E_{H4}	170
8.5.6	Conditional density given E_{H5}	170
8.6	Results	170

	Page
8.6.1 Simulated and Measured Results	172
8.6.2 Inquiry Scan using Specification v1.2	174
8.7 Noise and Other Disrupting Factors	177
8.8 Collision Avoidance	177
8.9 Multiple Inquiring Nodes	179
8.10 Multiple Inquiring Nodes and Collision Avoidance	181
8.11 Summary	182
IX. Simplification in the Inquiry Substate	184
9.1 CLK-Driven Inquiry Analysis	184
9.1.1 Probability of E_{Hi} , $i = \{1, 2, 3, 4, 5\}$	186
9.1.2 Conditional Densities for T_B Completion	187
9.1.3 Conditional Densities for the Inquiry Time	191
9.2 Results	195
9.3 Clk-Driven Inquiry with Standard Inquiry Scan	199
9.4 Summary	200
X. Network Models	201
10.1 Outreach Method Elimination	201
10.2 Simulation Model	202
10.3 Method Comparison	205
10.3.1 Goodput Measure	205
10.3.2 PER	206
10.3.3 Mean Delay	206
10.3.4 Buffer Size	208
10.3.5 Discovery Time	210
10.4 Packet Generation Time pmf	211
10.4.1 Non-bridge slave nodes	213
10.4.2 Bridge slave nodes	215
10.4.3 Master nodes	215
10.5 Summary	216
XI. Conclusions and Recommendations	218
11.1 Conclusions and Contributions	218
11.2 Recommendations for Future Research	220
Bibliography	221

List of Figures

Figure		Page
2.1.	Examples of network structures.	5
2.2.	Bluetooth packets: a) POLL/NULL packet b) full payload single-slot packet c) full payload three-slot packet d) full payload five-slot packet.	12
2.3.	Bluetooth packet exchange with varying packet size.	13
2.4.	Bluetooth packet exchange with a disruption.	14
2.5.	Bluetooth Frequency Hop Selection Kernel in connection state. [Blu03]	15
2.6.	Shifting of the MSI [PBR04b].	16
2.7.	Graphical depiction of the Bluetooth discovery process [PBK04]. 17	
2.8.	Shifting of the inquiry trains [PBR04a].	20
2.9.	The inquiry substate [PBR04a].	21
2.10.	Inquiry scan frequency remaining in a train [PBK04].	23
2.11.	PER simulation data for a) packets temporally overlapped by one interfering packet b) packets overlapped by two interfering packets [Bal03].	26
2.12.	Estimated probability density functions for v1.1 inquiry scan time a) simulated data [KaP02] b) measured data [Leo03].	28
2.13.	Measured probability density for standard inquiry scan [KaL01].	29
2.14.	Collision avoidance using PIAM a) slave packets collision b) master-slave packet collision c) inefficiency due to master-slave packet collision d) inefficiency due to identical TSFSs.	38
2.15.	Inefficient collision avoidance using PIAM.	39
4.1.	Scatternet configuration used for outreach comparisons.	52
5.1.	Alignment between piconet and inquiring node where a) Master packet is vulnerable only to the packet from the Lower Range (with a frequency F_1) and b) both Master and Slave are vulnerable to disruption.	60
5.2.	Probability of direct interference versus time of a) a Master packet b) a Slave packet c) Master or Slave packet where $n = \lfloor \frac{t}{40ms} \rfloor$	66
5.3.	Pattern of direct or indirect interference probability against any packet.	69
5.4.	Pattern of interference probability with a) five inquiring nodes b) an infinite number of interfering nodes.	74
5.5.	Probability of interference: multiple inquiring nodes with unknown Δ	75

Figure		Page
6.1.	Aligned MSI/SSIs.	79
6.2.	Determining the MSI/SSI spectral relationship, $N_{i,j}$	79
6.3.	Determining the MSI/SSI relationship: $N_{i,j}$ and $n_{i,j}$	80
6.4.	Determining the MSI/SSI temporal relationship $n_{i,j}$	81
6.5.	Portions of the Frequency Hop Selection Kernel [Blu03].	82
6.6.	Impact of placement within the MSI on likelihood of sharing a channel when $N_{1,2} = 24$, $n_{1,2} = 1$, $A1 = 7$, $A2 = 12$, and $B1 = B2 = 0$	84
6.7.	Effect of the master-slave frequency selection dependence on packet disruption.	87
6.8.	Determining $\Delta_{i,j}$, the packets' temporal relationship.	90
6.9.	Overlap of packet with a) $L_{i,S} = 5$, b) $L_{j,S} = 3$ and overlaps i 's packet with two packets with just one bit at the end of the packet on the left, and c) with just one bit at the beginning of the packet on the right.	92
6.10.	Probability mass function of PER between two random piconets. 108	
6.11.	Probability mass function of PER between m random piconets.	110
6.12.	Probability mass function of goodput between two random piconets.	114
7.1.	Derived probability mass function of goodput between two random piconets.	129
7.2.	Simulated probability mass function of goodput between two random piconets.	131
7.3.	Goodput between m random piconets.	133
7.4.	Maximum packet size using AFIT.	135
8.1.	First inquiry packet is received prior to T_R	141
8.2.	Train membership change during a scan window.	144
8.3.	Events E_P and E_Y occur and a) the scan window begins between $t = -10$ ms and $t = -1.25$ ms so the first packet is received after 1.28 s, b) the scan window begins between $t = -1.25$ ms and 0 s, c) the scan window begins between $t = -11.25$ ms and -10 ms.	145
8.4.	Possible events during back-off a) $E_{S\bar{T}}$, when the scan frequency changes but train membership does not and b) $E_{\bar{S}T}$, when train membership changes but the scan frequency does not.	146
8.5.	Probability density function of a) $T_R * T_1$ and b) $T_F E_{H1}$	150
8.6.	Conditional probability density functions for T_P	158
8.7.	Unconditional probability density for inquiry scan using v1.1 of the specification, $f_{T_I}(t)$ a) full scale b) magnified view.	171
8.8.	Difference in $f_{T_I}(t)$ between derived and simulated mean pdf using specified inquiry process.	173

Figure	Page	
8.9.	Probability density estimate for standard inquiry scan a) simulated [KaP02] b) measured [Leo03].	174
8.10.	Measured probability density estimate for standard inquiry scan [KaL01].	175
8.11.	Unconditional probability density for the standard inquiry scan, $f_{T_I}(t)$	176
8.12.	Inquiry time pmf with collision avoidance for three neighboring piconets.	179
8.13.	Inquiry time pmf with a) one other inquiring node b) four other inquiring nodes.	181
9.1.	Inquiry time pdf, $f_{T_I}(t)$, using the CLK-driven inquiry.	195
9.2.	Difference in $f_{T_I}(t)$ between derived and simulated mean pdf estimates using CLK-driven inquiry.	196
9.3.	Measured inquiry times compared to derived and simulated inquiry times using the CLK-driven inquiry substate.	197
9.4.	Measured inquiry times compared to simulated inquiry times using BT specification v1.1.	198
9.5.	Inquiry time for the standard inquiry scan using CLK-Driven Inquiry, $f_{T_I}(t)$	199
10.1.	Benchmark scatternet configuration used for outreach comparison.	203
10.2.	Average piconet goodput (kbps).	206
10.3.	Packet Error Rate due to a) Direct Collision b) Packet repeated due to Indirect Collision.	207
10.4.	Mean packet delay (ms).	207
10.5.	Mean buffer size for non-bridge slaves.	208
10.6.	Buffer size for bridge slaves a) average b) maximum.	209
10.7.	Buffer size for master nodes a) a single master node during a simulation run b) all master nodes across all simulation runs.	210
10.8.	Buffer size for master nodes a) average b) maximum.	211
10.9.	Discovery time by a) first piconet b) second piconet c) third piconet.	212
10.10.	Non-bridge slave node packet generation time pmfs a) Piconets A-D b) Piconet E.	214
10.11.	Bridge node packet generation time pmfs a) Piconets A-D b) Piconet E.	216
10.12.	Master node packet generation time pmfs.	217

List of Tables

Table		Page
2.1.	Bluetooth system power consumption at 3.3V [KaL01].	25
5.1.	Piconet packet vulnerability.	62
5.2.	Piconet packet header vulnerability.	63
5.3.	Probability of interference for unknown Δ	76
6.1.	Piconet packet vulnerability.	89
7.1.	Expected goodput.	130
7.2.	Expected efficiency.	132
7.3.	Goodput with multi-slot packets.	138
7.4.	Expected efficiency with multi-slot packets.	138
8.1.	Devices discovered by inquiry time.	172
8.2.	Inquiry time using collision avoidance.	179
8.3.	Inquiry time with multiple inquirers.	180

List of Abbreviations

Abbreviation		Page
AC	Access Code	11
AFH	Adaptive Frequency Hopping	25
AFIT	Avoidant Forward Inspection Technique	118
BT	Bluetooth	9
CDF	Cumulative Distribution Function	155
<i>CLK</i>	Free-Running Counter of a BT device	9
DIAC	Dedicated Inquiry Access Code	17
DSSS	Direct Sequence Spread Spectrum	1
FHSK	Frequency Hop Selection Kernel	14
FHSS	Frequency Hopping Spread Spectrum	1
GIAC	General Inquiry Access Code	17
GPS	Global Positioning System	8
ISOM	Inquiring Slave Outreach Method	202
MSI	Master Selection Interval	14
MSTS	Master-slave Time Slot	10
NB	Non-bridge Slave	212
NISOM	Non-bridge Inquiring Slave Outreach Method	202
PAN	Personal Area Network	2
pdf	Probability Density Function	42
PER	Packet Error Rate	2
PH	Packet Header	11
PIAM	Popovski Interference Avoidance Method	37
QoS	Quality of Service	31
SBOM	Scan-Based Outreach Method	202
SINCGARS	Single-Channel Ground And Airborne Radio System	8
SSI	Slave Selection Interval	15
TSFS	Time Slot Frequency Sum	37

DEVICE DISCOVERY IN FREQUENCY HOPPING WIRELESS AD HOC NETWORKS

I. Introduction

1.1 *Background*

As electronic devices become entrenched in even the most minute aspects of our daily lives, the need for wireless data transfer is essential. From controlling and monitoring military operations down to an individual soldier, to activating a clothes dryer on the drive home, wireless devices are engrained in our society. Several wireless protocols have been developed during the transition to wireless systems. These protocols use different methods for optimizing frequency use, noise suppression, and security including frequency hopping spread spectrum (FHSS) and direct sequence spread spectrum (DSSS). While forming ad hoc networks to transfer information in mobile scenarios, devices sometimes interfere with nearby networks and systems. An efficient discovery protocol should minimize collateral interference while retaining the ability to locate new members searching for a wireless network to join. The process of discovering and accepting nodes into such a network is called *outreach*.

1.2 *Research Objectives*

The primary objective of this research is the development of a method of outreach for a frequency hopping, synchronous ad hoc network consisting of clustered structures that can discover arriving nodes quickly while limiting interference to the network clusters. Examples of such networks include small combat units with full inter-connectivity between the unit members with one or two links to the larger network infrastructure and swarm technologies with clusters of units devoted to

separate tasks. To that end, the Bluetooth [Blu01] [Blu03] Personal Area Network (PAN) protocol is used as a representative protocol for such systems since it is one of the few such protocols currently in use. The observations, conclusions, and methods that apply to the Bluetooth model generally apply to other systems. The smallest network structure, or cluster, in a Bluetooth PAN is a piconet. Piconets can be ordered into a larger network structure, or scatternet.

A secondary objective of this research is to expand the body of knowledge surrounding the Bluetooth protocol. First presented in 1999, Bluetooth is a relatively protocol whose knowledge base is limited to simulation models, measured data, and predictive models based on largely simplistic assumptions. The interactions between packets of neighboring piconets, the interactions between piconets and nodes in the discovery process, and the time needed for discovery have not been fully characterized and no accurate predictive models are available. Although the protocol for communication within the individual clusters, or piconets, in a Bluetooth network is well-defined in the Bluetooth standard, the protocol for efficiently maintaining the larger network structure, or scatternet, remains under development [Blu03]. The models and methods presented in this research will be useful in further development of the scatternet protocol.

1.3 Document Overview

This chapter is a brief introduction to the objectives of this research.

Chapter II provides background information on clustered, FHSS ad hoc networks, with general definitions and examples as well as specific discussion of the Bluetooth protocol. In addition to a summary of the Bluetooth operational states and the hop sequences used in each state, research relevant to packet error rate (PER), inquiry time, scatternet scheduling, organization, collision avoidance, and outreach is presented. Chapter III discusses the Law of Total Probability and statis-

tical tests used in analyzing performance characteristics of Bluetooth networks while Chapter IV presents the methodology used to meet the research objectives. Goals and assumptions are discussed along with their effect on performance.

An analysis of the interference problem posed by inquiring devices to piconet traffic is presented in Chapter V. The PER between piconet packets is completely characterized in Chapter VI. Using similar analysis constructs as those in Chapter VI, proposed and new packet collision avoidance methods are characterized and compared in Chapter VII.

The inquiry time distribution is derived in Chapter VIII for all implementations of the Bluetooth discovery process. Additionally, the impact on inquiry time due to multiple inquiring devices and collision avoidance with piconets is presented. An analysis on the inquiry time using a modified inquiry process is provided in Chapter IX. In Chapter X, several outreach methods are evaluated and the three most promising methods are compared. Finally, research conclusions and recommendations for future research are presented in Chapter XI.

II. Background and Literature Survey

Clustered synchronous ad hoc frequency hopping wireless networks are defined in Section 2.1 with actual and potential applications presented. Typical methods for synchronizing to a network's frequency hopping pattern are also addressed. The protocol of the most typical example of such networks, Bluetooth, is summarized in Section 2.2. Current data and research on packet collisions between piconets in the normal communication state is discussed in Section 2.3. Section 2.4 is dedicated to current research on the Bluetooth discovery process and inquiry time is presented. Bluetooth scatternets are discussed, including a summary of proposed packet scheduling, organization, and packet collision avoidance methods in Section 2.5. Finally, proposed methods for discovering and allowing the scatternet to be discovered, or outreach, are presented.

2.1 Clustered Synchronous Ad Hoc Frequency Hopping Spread Spectrum Wireless Networks

2.1.1 Clustered Networks

A *clustered* network is defined as a group of distinct, identifiable, highly-connected subnetworks with a maximum of one link connecting individual subnetworks. Since inter-cluster throughput requirements are typically considerably less than intra-cluster throughput requirements, a single link between clusters is sufficient. If this is not the case, the clusters should be re-apportioned or a mesh network used. Each cluster may have unique network identifier codes, unique communication protocols, or any other characteristic which clearly segregates the clusters. The clusters have a common protocol, interpreter, or the ability to change cluster affiliations such that any cluster can transfer information to any other cluster. A mesh network, on the other hand, allows for direct connection from any node to any other

node within range. An example of a clustered network and a mesh network is shown in Figure 2.1.

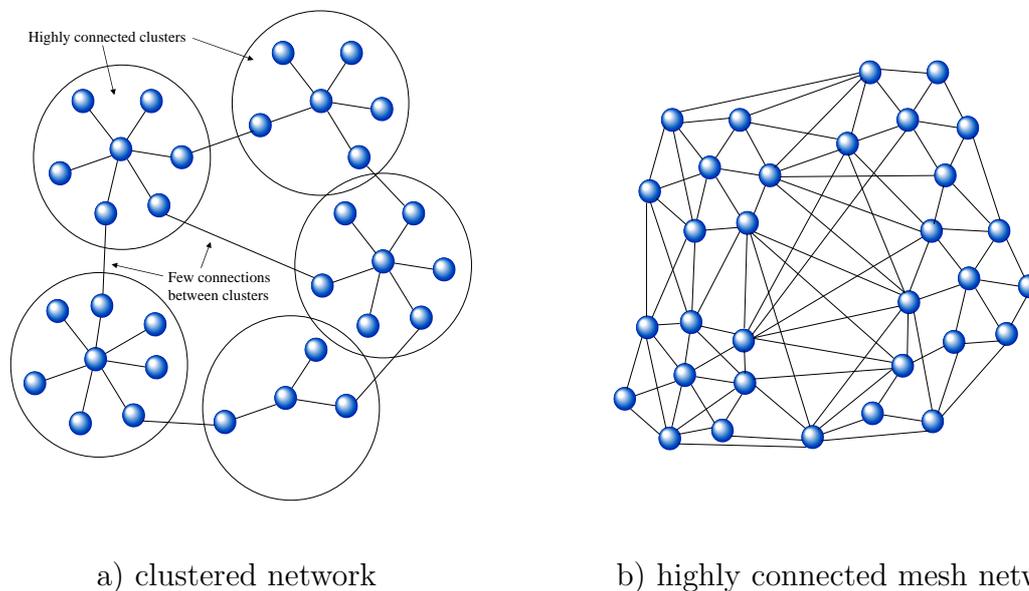


Figure 2.1: Examples of network structures.

Clustered networks are useful in situations where nodes communicate regularly with other nodes in the cluster and need only sporadic access to the other clusters in the network. This is the situation in the Army's Land Warrior program [NCR97]. The Land Warrior system is designed to integrate data support into an individual soldier's uniform. Tactical data is transferred to the soldier via a wireless link while data, such as position, heart rate, respiratory rate, and intelligence, can be sent to higher levels of command. Via this wireless link, an individual soldier maintains contact with the other members in his squad. A squad leader can contact the platoon leaders which can contact each other and as well as higher headquarters. If a squad leader is unable to communicate, another member of the unit can transfer the data to/from the platoon leader. The Bluetooth protocol which can support the scenario described, was evaluated by Army researchers for use in the Land Warrior radio system but rejected due to the limited range of the Bluetooth protocol [DFS01].

This cluster structure may also be useful in a swarm of nodes where groups of workers attend to a common task. In such a structure, workers need to be strongly connected to other workers in the cluster for coordination, but only need limited links to the rest of the swarm to update status and receive information pertinent to their collective task.

2.1.2 Ad Hoc Networks

An ad hoc network is a network that is formed as needed without any assistance from an existing infrastructure [Per01]. For this reason, node membership in the network at a particular time is generally unpredictable. A node's arrival to join the network and departure from the network may occur at any time without warning or notice.

2.1.3 Synchronous Frequency Hopping Spread Spectrum

A Frequency Hopping Spread Spectrum (FHSS) transmission scheme chooses a transmission frequency from a set of multiple frequency bands which collectively span the network's available bandwidth. If the hop sequence used by multiple networks are orthogonal, the networks can operate with no mutual interference. If interference only occurs over a small number of frequency bands, performance still may not degrade significantly since the nodes only dwell in the affected band for short durations. A pseudo-random hop sequence provides some security from packet interception since the transmission frequency is unknown to an eavesdropping node. Synchronous FHSS network nodes hop to the next frequency at the same time. Fast hopping FHSS systems change frequency at least once during a symbol transmission while slow hopping FHSS systems remain on the same frequency for the entire duration of a symbol and possibly for the duration of multiple packets [PZB95].

To hop with a FHSS network, each node must synchronize to the hop sequence. However, a "key limitation of protocols based on code assignments is that

senders and receivers have to find each others' codes before communicating with each other [TzG00].” Synchronization to a hop sequence can be achieved by a variety of methods. Synchronization is typically a function of the sequence length, the number of hop frequencies, and the network structure. The hop sequence is usually known a priori and it is the current phase of the sequence that must be determined. However, there are exceptions [WRV86]. In some protocols, such as Bluetooth, synchronization with the network is not sufficient to be included in the network. Additional acceptance protocols may be required for other members or the controlling entity of the network to recognize a node as a member.

Various methods for synchronizing to a hop sequence are described in the following subsections. Methods that passively synchronize or synchronize on a dedicated acquisition frequency impose no interference in the system (other than on the dedicated frequency) and their effect does not need to be mitigated. Methods that require active transmission on frequencies used by the network during normal operation may affect the target network or neighboring networks' performance. These methods require additional collision mitigation to improve network throughput while retaining member acquisition capabilities.

Brute Force - One of the simplest ways to recover frequency-hopped data is to ignore the hop sequence and simultaneously receive on all possible frequencies. Having circuitry for each possible frequency allows data reception regardless of what frequency the hop pattern uses [Rap88]. Of course this method is hardware intensive and inefficient since only one receiver/transmitter is used at any given instant. Additionally, since the next frequency in the sequence is generally unknown until the entire pattern has been determined, the node must also transmit packets on all frequencies. Likewise, if multiple networks are transmitting in the vicinity of the receivers, packets from all networks are received and must be analyzed to determine if any are destined for the receiving device.

Fixed Synchronization Frequency - Some systems have a dedicated fixed synchronization frequency that new nodes use to transmit requests to the system and monitor for responses. Occasionally, network members monitor the same frequency and transmit information new members need to synchronize to the hopping sequence [VaE87]. Acquisition time is dependent on how often the network members monitor the synchronization frequency. This method does not interfere with the rest of the network and does not require mitigation.

Short Known Pattern - Some schemes use relatively short, known hopping patterns. A shifted version of one published pattern is used by IEEE 802.11b, or WiFi [IEE99]. HomeRF is similar, using one of 75 possible patterns over 75 frequencies that repeats every 1.5 seconds [ZGE01]. To synchronize, the entire sequence can be captured and replicated if the system hops slowly enough to allow the receiver to sequentially check all frequencies. If the hop sequence does not repeat frequencies in an iteration of the hop sequence, the receiver can dwell on a frequency until that frequency is used, at which time the phase of the sequence is discovered. Both methods are passive and do not require mitigation to reduce interference.

Long Known Pattern with Shared Time Knowledge - Some schemes use long hop patterns which are known. For example, the Commercial Communications Associates HF-90 radio set uses a hop pattern that repeats every 457 million years [CCA04]. Generally, the hop pattern is known only to those who are permitted access to the network. A secret parameter is given a priori to authorized network participants. The pattern phase is determined by a second parameter that all nodes have access to. For example, Have Quick and the Single-Channel Ground And Airborne Radio Systems (SINCGARS) have Network Identification numbers and Words-of-the-Day that are used to generate the pattern [HAV99] [SIN96]. The Global Positioning System (GPS) time, or a locally generated time on a known frequency, is used by all members to determine the correct phase of the frequency hopping sequence.

Two-Level Scheme - Some schemes use a long hop sequence for normal data transfer and a short sequence for coarse acquisition. The short sequence is generally based on a subset of frequencies used by the system. If the short sequence is known, methods mentioned above may be used for coarse acquisition. Otherwise, a more complex protocol may be required. (Such is the case with the Bluetooth protocol as discussed in Section 2.2.) Once coarse acquisition is achieved, the devices communicate using the short sequence and the information needed for synchronization with the long sequence is exchanged [GTP86].

Other Synchronization Methods - Other research has been conducted to characterize an unknown hop sequence in a noisy environment, but such methods are of little value since the entire pattern must usually be recorded twice to counter noise effects [BBB00]. In the case of a long pattern, this is often not feasible for timely communication.

2.2 The Bluetooth Protocol

Bluetooth (BT) is a low-power, open standard for implementing PANs [Blu01] [Blu03]. It is a popular protocol with 40 million Bluetooth-enabled phones shipped worldwide and over 1,000 new BT products being developed by more than 2,000 companies [GNL03]. It uses a slow hop frequency hopping spread spectrum scheme with 79 1-MHz frequency slots (23 in some countries) in the 2.4 GHz band. Each device has a 28 bit 3.2kHz free-running counter (*CLK*) which cannot be set or reset. Members of a BT network, or piconet, hop together among the 79 frequencies (numbered 0-78) with a sequence that is a function of the master's free-running counter and the first 28 bits of the master's 48 bit address. The piconet master coordinates time-division duplex transmissions of up to seven *active* slaves by alternating between master and slave transmissions in 625 μ s time slots. Master packets always begin on *even* time slots and can use one, three, or five time slots. Slave packets always begin on *odd* time slots and can also be one, three, or five time slots. Time

slots are designated as even or odd based on bit 1 of the master's CLK, or CLK_1 , which toggles every $625 \mu s$. The $1250 \mu s$ containing an even (master) time slot and the odd (slave) slot immediately following it are denoted as a *master-slave time slot* (MSTS) [PBR04b].

Up to 256 nodes can be associated with the piconet in a power conservation mode where activity is temporarily suspended. A device can be a master in one piconet and a slave in other piconets thereby forming what is called a scatternet. A BT device can be in either a *standby* or *connection* state or one of seven substates: *inquiry*, *inquiry scan*, *inquiry response*, *page*, *master response*, *page scan*, or *slave response*. The substates are part of the process used in the discovery of prospective piconet members and the transfer of the master's clock and address information to accepted members. Each are described in what follows

2.2.1 *Standby state*

No communication occurs in the standby state and the device is in the lowest power consumption mode. This is the default state of BT devices.

2.2.2 *Connection state*

In BT devices, routine data traffic is transferred in the *connection* state. Piconet members hop together among the 79 frequencies with a sequence that is a function of the master's counter and the first 28 bits of the master's address. Slave devices add an offset to their resident CLK to match the master's CLK and generate the hop sequence.

BT packets are transmitted over ACL (asynchronous) or SCO (synchronous) links. The BT definition for "synchronous" is different than the standard definition of "synchronous" given above. A BT SCO link uses scheduled time slots for periodic data transfer between the master and a specific slave. These are typically

used for voice transmission. A BT device can support three SCO links. With an ACL link, packets are not scheduled and are selected for transmission by the scheduling protocol. Using the standard definition then, both SCO and ACL links are synchronous since each node in the piconet hops to the next frequency in the hop pattern simultaneously.

All packets, except inquiry packets, contain a 72-bit Access Code (AC) to identify the piconet and a 54-bit Packet Header (PH). The Access Code identifies the piconet a packet is associated with. Inquiry packets, as discussed in Section 2.2.3, are comprised of an AC without the four trailer bits normally attached to an AC. Thus, Inquiry packets contain only 68 bits. The PH contains packet type, acknowledgement and sequence information as well as a 3-bit slave device address (LT_ADDR) which is assigned by the master when the slave is accepted into the piconet. POLL and NULL packets have no payload. A POLL packet is sent by the master to a slave to verify its active status in the piconet, retain synchronization, and allow the slave an opportunity to transmit a packet to the master. If the slave has no data to send, it responds with a NULL packet.

As shown in Figure 2.2, a single-slot packet can have up to 240 payload bits, leaving 259 μs of the 625 μs time slot for oscillator re-tuning. Three- and five-slot packets leave at least 253 μs to re-tune the oscillator for the next frequency in the hop sequence. ACL packets have two levels of error correction which can be used for the payload while SCO packets have three levels of error correction.

Multi-slot packets do not change frequencies during transmission. After transmission is complete, the device hops to the current frequency generated by the address/CLK combination. Thus, during the transmission of multi-slot packets, some frequencies in the hop sequence are skipped. By jumping back to the scheduled frequency once the multi-slot packet is complete, all active devices can determine the frequency the master will transmit on in a given slot. This pattern is seen in

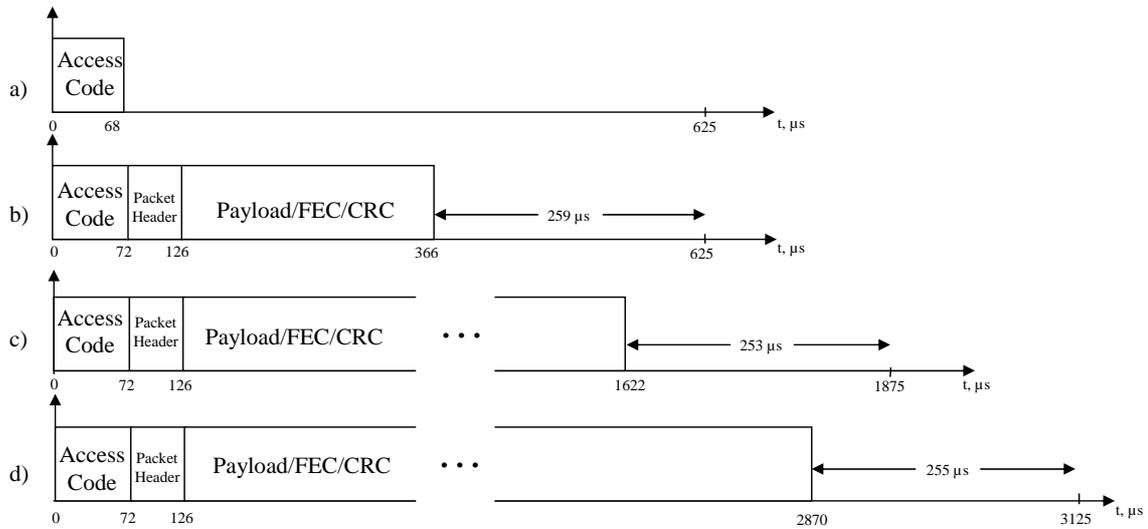


Figure 2.2: Bluetooth packets: a) POLL/NULL packet b) full payload single-slot packet c) full payload three-slot packet d) full payload five-slot packet.

Figure 2.3. In Slot 1, the master begins a 5-slot packet to Slave 3 on Freq 1. After the transmission, Slave 3 responds in Slot 6 on Freq 6 rather than Freq 2.

Master devices always begin transmission to slaves on even time slots. These packets are called *Master* packets. A slave that receives a Master packet responds in the subsequent odd time slot with a *Slave* packet. When the master transmits a packet to a slave, the slave must respond even if it has no data to send. This acknowledges reception of the master packet as well as confirms link status and active participation in the network. All active slave devices listen for the master at the beginning of even time slots to determine if they are being addressed. Within the Packet Header, the slaves receive the packet type indicator which informs them of the packet length and destination slave. If a slave is not being addressed, it drops to a low power mode until the beginning of the next even slot after the master's packet is complete. For multi-slot packets, this may be four or six slots. If the slave device responds to the master with a multi-slot packet transmission, the other devices detect that the master did not send a packet at the expected hop frequency and switch to low power mode until the next even time slot.

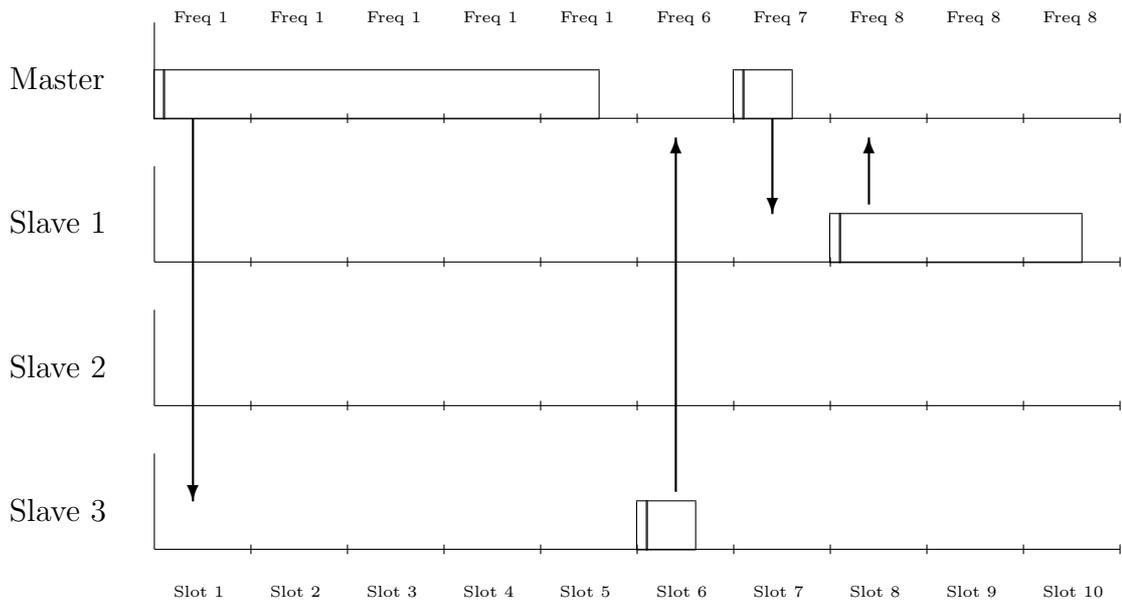


Figure 2.3: Bluetooth packet exchange with varying packet size.

If the master's packet header is disrupted, no slave recognizes it is the intended recipient and therefore none will respond. Such an occurrence can be seen in Figure 2.4. In Slot 1, the master sends a single-slot packet which is received by Slave 1. The slave responds with a single-slot packet in Slot 2 which is received by the master. In Slot 3, the master transmits a packet which is disrupted. As indicated in the figure, it is unclear which slave was the intended recipient and since the header is disrupted, no slave recognizes it was addressed. Therefore, no slave responds in Slot 4. If the PH is received successfully and only the payload data is corrupted, the addressed slave will still respond in the next odd slot after the master has completed packet transmission.

2.2.2.1 Connection State Frequency Hop Pattern

Over time, frequencies used in the connection state are uniformly selected from the 79 BT frequencies. However, the instantaneous probability that a frequency is selected is based on 32-frequency partitions of the BT frequency band. The hop

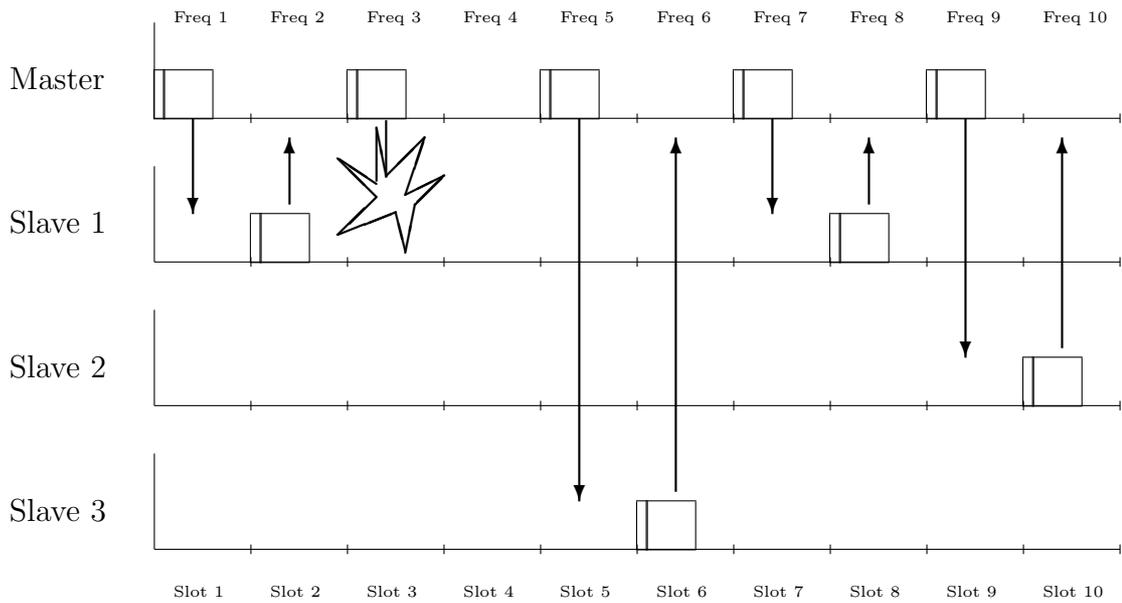
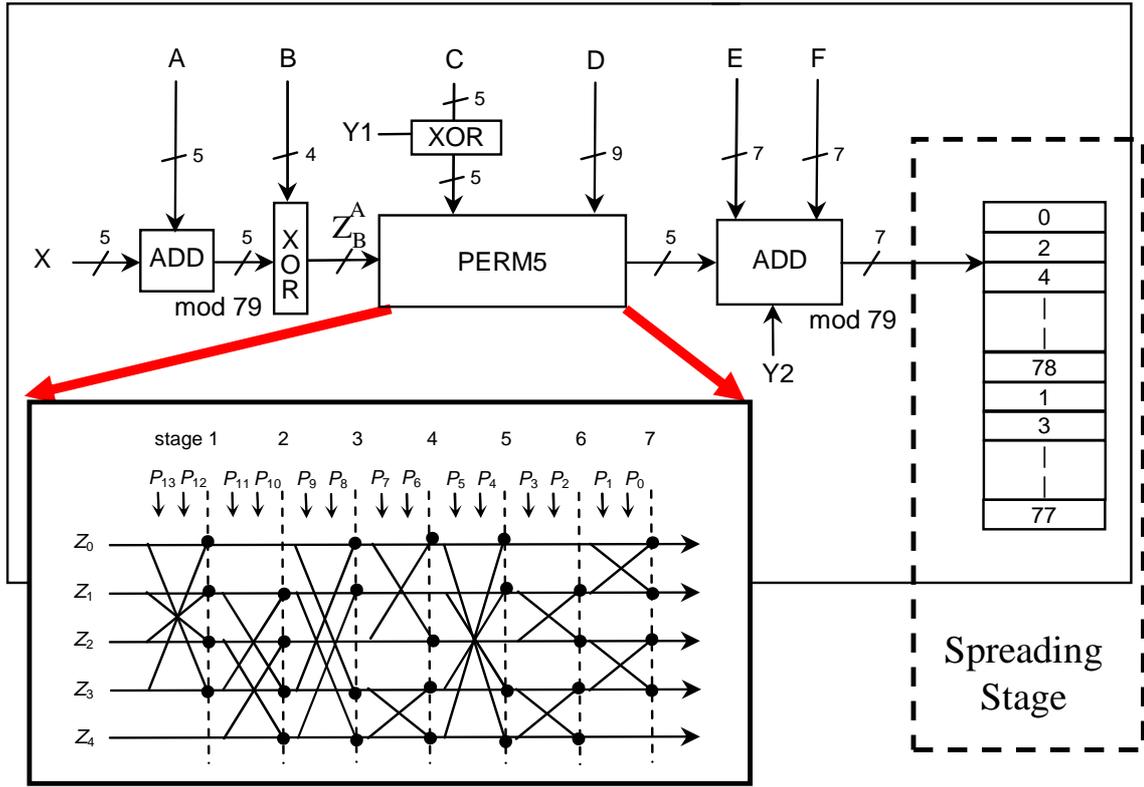


Figure 2.4: Bluetooth packet exchange with a disruption.

sequence is generated by the Frequency Hop Selection Kernel (FHSK) shown in Figure 2.5. The FHSK is described in detail in Section 6.1.3. The FHSK uses a group of 32 frequencies called the Master Selection Interval (MSI) [PBK03] to select a frequency for Master packets. In the BT spectrum, the frequencies in a MSI are not consecutive. The final stage of the BT hop frequency selection process shown in Figure 2.5 spreads the frequency selections across the BT spectrum by doubling the Bluetooth frequency number modulo 79 [Blu01] [Blu03]. Since interference from adjacent frequencies is assumed to be insignificant, placement within the spectrum is irrelevant. Thus, the spreading stage is ignored and the 32 frequencies in the MSI are treated as consecutive as shown in Figure 2.6. The MSI is determined by the CLK and address values of the master device in the piconet. Each unique 28-bit address input to the FHSK generates a unique hop sequence. The sequence phase is a function of the free-running counter. Since the order the frequencies chosen within the MSI are scrambled by the FHSK, the selection appears random to an outside observer and is considered pseudo-random.



$$X = \text{CLK}_{6-2}$$

$$Y1 = \text{CLK}_1$$

$$Y2 = 32 \times \text{CLK}_1$$

$$A = A_{27-23} \oplus \text{CLK}_{25-21}$$

$$B = A_{22-19}$$

$$C = A_{8,6,4,2,0} \oplus \text{CLK}_{20-16}$$

$$D = A_{18-10} \oplus \text{CLK}_{15-7}$$

$$E = A_{13, 11, 9, 7, 5, 3, 1}$$

$$F = 16 \times \text{CLK}_{27-7} \bmod 79$$

Figure 2.5: Bluetooth Frequency Hop Selection Kernel in connection state. [Blu03]

When using single-slot packets, the slave transmits on a frequency selected from the Slave Selection Interval (SSI), a 32 frequency partition immediately to the right of the MSI modulo 79 [PBK03]. Once each of the 32 frequencies in the MSI and SSI have been used exactly once, both intervals shift right by 16 frequencies modulo 79 as shown at $(t + 40)$ ms in Figure 2.6. Therefore, each piconet's MSI/SSI shifts every 32 MSTs. A *MSI cycle* is defined as the 32 MSTs a piconet uses before shifting by 16 frequencies [PBR04b]. In other words, a MSI cycle begins when the MSI/SSI shifts and ends 64 time slots later when the MSI/SSI shifts once again, beginning a new MSI cycle. Note that when multi-slot packets are used, it is

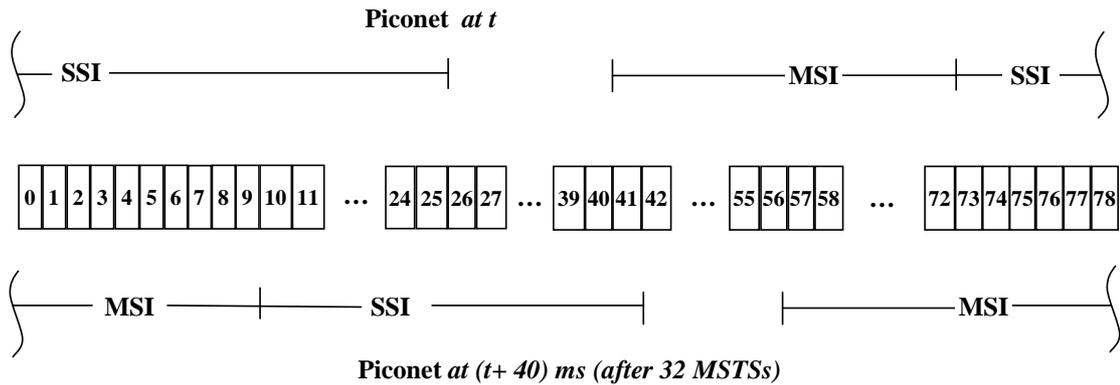


Figure 2.6: Shifting of the MSI [PBR04b].

possible the MSI/SSI will shift during transmission of the master packet, resulting in the frequency used by slave packet being selected from a SSI that is 48, rather than 32, frequencies to the right of the MSI used by the master packet.

The MSI/SSI shifts such that it uses all 79 MSI/SSI combinations before repeating a MSI/SSI. Thus, the frequency selection is uniform across the 79 frequencies in the limit. Note that in Figure 2.5, $\text{mod}(E + F, 79)$ determines the beginning of the MSI and $Y2$ provides the 32 frequency offset of the SSI from the MSI.

Before a slave has the master's CLK and address values needed to generate the hop sequence, the slave must be discovered by the master, paged to join the piconet, and assigned a 3-bit piconet address, LT_ADDR . This discovery process uses the *inquiry*, *inquiry scan*, and *inquiry response* substates.

2.2.3 Inquiry Substate

A node enters the *inquiry* substate for a given period of time to discover other nodes to form a piconet; the inquiring node acts as the master. BT devices typically use the specification recommended inquiry time of 10.24 s [Blu03] [WJC02]. A node in the *inquiry scan* substate, in contrast, searches for nodes in the inquiry substate to form a piconet; the scanning node acts as a slave.

A device in the inquiry substate transmits inquiry packets on two pseudo-random frequencies during a normal packet time slot as shown in step 1 of Figure 2.7. Inquiry packets consist of a 68 bit General Inquiry Access Code (GIAC) or Dedicated Inquiry Access Codes (DIAC) that do not include the four trailer bits normally associated with an AC. The GIAC is used when searching for any BT device. A node may use one of several DIACs to search for devices with specific characteristics [Blu01] [Blu03]. The inquiring device waits for a response $625\mu\text{s}$ later on the same frequencies from a prospective slave device in the *inquiry scan* substate. The inquiring device continues this process while collecting responses until the inquiry period is complete or an acceptable number of devices have been discovered. The device may leave the inquiry substate to service SCO links or to immediately *page* a discovered device. The page process uses the response from a discovered device to contact prospective slaves and incorporate them into the piconet (cf. Section 2.2.5). An inquiring device may also wait until the inquiry period is complete to page devices that are accepted into the piconet.

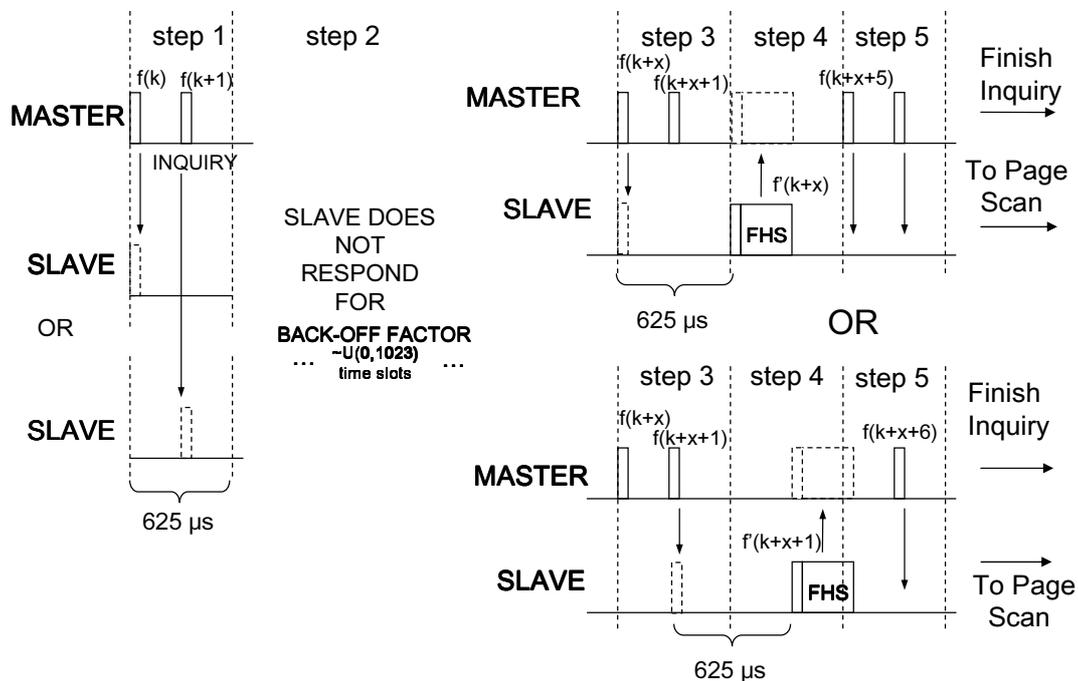


Figure 2.7: Graphical depiction of the Bluetooth discovery process [PBK04].

Although some have recommended that a device enter the inquiry substate on a regular basis [Blu03] [SBT01] [ZaG04], many implementations do not. For example, in their default mode, TDK and Anycom PC BT cards enter the inquiry mode only when explicitly commanded [PBR04b].

2.2.3.1 Inquiry Substate Frequency Hop Pattern

The inquiry substate uses a 32-frequency partition of the 79 frequencies similar to that used by a piconet in the connection state [Blu03]. Unlike in the connection state, however, this partition remains constant. A node uses an address associated with the GIAC ($9E8B33_{16}$) or one of the DIACs to generate the frequency hopping sequence. This address determines the 32-frequency subset used by the inquiry/inquiry scan substates.

Although generated using the resident CLK and the GIAC or DIAC address, to an outside observer the hop sequence within the partition appears random. The 32 frequencies used by the GIAC hop sequence are also spread across the BT spectrum by the spreading process in the final stage of the Frequency Hop Selection Kernel as shown in Figure 2.5 [Blu03]. Again, placement within the spectrum is irrelevant for analysis and the spreading stage is ignored. Thus, the frequencies used by the inquiry substate for the GIAC address are designated as 0-5 and 53-78, rather than the BT frequency spectrum designations used after the spreading process which doubles the frequency number modulo 79 (i.e., 0,2,4,6,8,10,27,29,31,...77) [Blu03].

This set of 32 frequencies is further segmented by the inquiry procedure into two 16-frequency trains, *A* and *B*. A device in the inquiry substate chooses the *A* or *B* train for initial transmission and switches between the trains every 2.56 s. The train used initially is not significant [Blu03]; the initial train selection process is implementation specific. Remaining in the inquiry substate for 10.24 s as the specification recommends allows four train changes. The frequencies within these trains change over time as shown in Figure 2.8 where the *A* train are the frequencies

in the white boxes and the B train are the frequencies in the shaded boxes. The 16 frequencies in a train at a given instant are called the train's *membership* [PBR04a]. A frequency entering the A train is shown as a thick bordered box and a frequency switching to the B train is shaded with a left-hash. The trains exchange one member every 1.28 s based on bit changes in the free-running counter, completely swapping membership every 20.48 s. It is possible, although unlikely, that a device will enter the inquiry substate exactly as the train membership changes. Therefore, assuming a device enters the inquiry state between t and $t + 1.28$ s in Figure 2.8 and uses train A , it begins using a train that includes frequencies 67 and 75. However, by the time the device changes the train it is using 2.56 s later, train A no longer contains frequencies 67 and 75, but rather contains 63 and 71 due to membership changes. Even so, each train always will have eight frequencies in the 53-68 range and eight frequencies in the 69-78 and 0-5 range. These are called the Lower and Upper Ranges, respectively [PBR04a].

An inquiring device transmits an inquiry packet at the beginning of an *even* time slot, when its two least significant clock bits are zero ($\text{CLK}_{1,0} = 00_2$). An *even* time slot occurs every 1250 μs . The frequency used is pseudo-randomly selected from frequencies in the current train, A or B , from the Lower Range [PBR04a]. This frequency is called F_1 . Another inquiry packet frequency, F_2 , is selected from the Upper Range in the current train and is used 312.5 μs later when $\text{CLK}_{1,0} = 01_2$. During odd time slots, the device listens for responses from neighboring devices in the inquiry scan substate. For example, in Figure 2.9, the device transmits an inquiry packet on frequency F_1^0 in the even time slot followed by a second packet on F_2^0 312.5 μs later. Since responses are transmitted 625 μs after an inquiry packet is received and are transmitted on the same frequencies, F_1 and F_2 , as per Section 11.3.4 of the specification [Blu03], the device listens on frequencies F_1 and F_2 in the odd slot. Thus, in Figure 2.9, the device then listens for a response on F_1^0 at the beginning of the odd time slot and on F_2^0 312.5 μs later. In the subsequent even

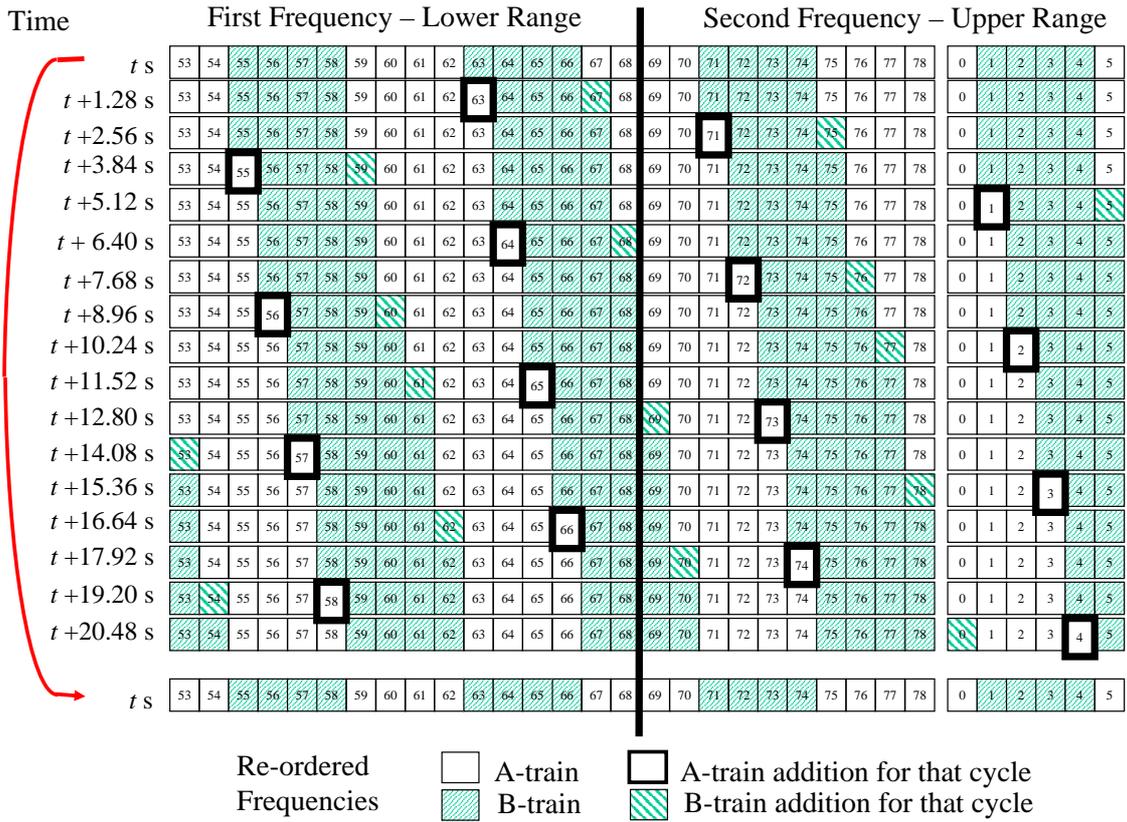


Figure 2.8: Shifting of the inquiry trains [PBR04a].

time slot, the inquiring device then selects new frequencies, F_1^1 and F_2^1 , and repeats the process, collecting data from as many neighboring devices as possible during the duration of the inquiry substate.

2.2.4 Inquiry scan/Inquiry response substates

A device enters the inquiry scan substate to make itself available to discovery by an inquiring device. To account for the hop sequence randomness, the scanning device only changes frequency every 1.28 seconds. Since the scan frequency changes every 1.28 s, and the train changes every 2.56 s, most implementations only scan for 11.25 ms [KaL01] [Blu03] and then move to the *connection* (i.e., normal operation) or a *standby* state for the remainder of the 1.28 s. Using a scan of length 11.25 ms

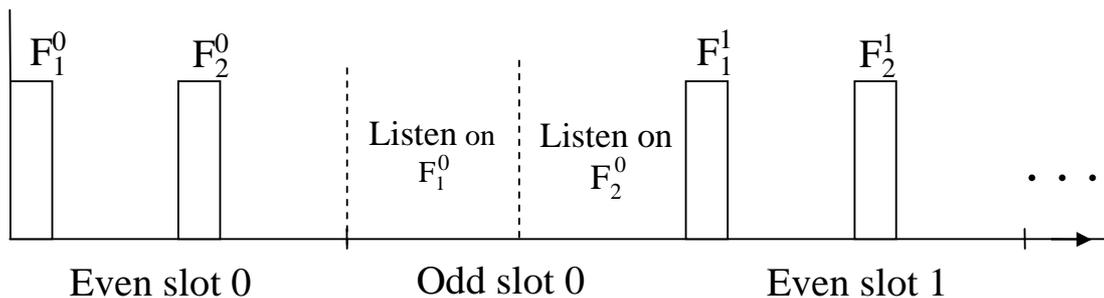


Figure 2.9: The inquiry substate [PBR04a].

rather than 10 ms compensates for any timing misalignment and allows the scanning device to receive at least one full inquiry train.

When an inquiry packet is received, a scanning device using version 1.1 of the BT specification drops out of scan mode for an integer number of time slots uniformly distributed between 0 and 1023 [Blu01]. This corresponds to a back-off time of 0 to 639.375 ms. This is Step 2 in Figure 2.7. This delay is designed to reduce collisions between multiple devices receiving the same inquiry packet. After the back-off period elapses, the device returns to scan mode. After receiving a second inquiry packet (Step 3), the scanning device waits 625 μ s and enters the inquiry response substate to return a FHS packet to the master (Step 4). The FHS packet contains the slave device address and *CLK* values. In Step 5, the inquiring device either continues transmitting packets for the duration of the inquiry substate to find other neighboring devices or jumps to the page substate to immediately page the scanning device before continuing to inquire. After transmitting the FHS packet, the scanning device advances the scan frequency by adding a 1.28 s offset to the *CLK* and re-enters the inquiry scan substate. Before doing so, it is allowed to enter the page scan substate in case the inquiring device immediately pages it.

Using the *standard* inquiry scan from BT specification v1.2, the requirement to receive a second inquiry packet is removed, effectively beginning the scan process at Step 3 [Blu03]. The back-off interval is utilized between transmission of the FHS

packet and reentry into the inquiry scan substate where a scan window is once again opened.

The *interlaced* scan introduced in v1.2 of the BT specification is similar to the standard inquiry scan except that immediately following a scan window, a second scan window is opened using a scan frequency from the train not used in the first scan window [Blu03].

The inquiring device discovers neighboring devices and collects information to incorporate devices into the piconet. The inquiring device continues transmitting inquiry packets and discovering devices until the specified duration of the inquiry substate is complete or a specified number of devices have been discovered.

When not inquiring, TDK and Anycom BT PC cards remained in the inquiry scan substate [PBK04a]. Since a scan window requires only 11.25 ms, the remainder of the 1.28 s between scan windows can be spent performing other operations. If the device is not a member of any piconet, the time between scan windows can be spent in the low-power standby state. If the device is an active slave in another piconet, entering the inquiry scan substate allows it to remain available for acceptance by another master, thus forming or expanding a scatternet. Since the scan window is so small, the device is unavailable for normal communication only 0.8% of the time. Therefore, it is expected when a user initiates an inquiry, most neighboring devices will be in inquiry scan and open a scan window every 1.28 s. Since the scan window openings and scan frequencies are independent and the devices have equal probability of their scan frequency being in train A or train B, the probability of two scanning devices receiving the same inquiry packet is $0.5/2048 = 0.00024$ for a standard scan and $0.5/3072 = 0.00016$ using the v1.1 of the specification. Thus, the number of scanning devices can be quite large before interference from FHS responses becomes an issue [PBK04a].

2.2.5 Page/Page scan substates

Once a master has received a FHS packet from a discovered device containing the address and CLK values of a prospective slave device, the page substate is used to accept the device into the piconet. The process is similar to that of the inquiry/inquiry page substates except the knowledge of the prospective slave's address and CLK enables the two devices to quickly find a common frequency. The master then provides the slave with the CLK and address values needed to synchronize with the piconet and assigns it *LT_ADDR* [Blu03].

2.2.6 Low Power States

A slave device in the connection state has several low activity modes such as *sniff*, *hold*, and *park* in addition to the *active* mode that allows a slave to be an active participant in a piconet. In sniff mode, the duty cycle of the slave's listening activity is reduced to agreed upon intervals. In hold mode, the slave's listening activity is suspended for an agreed upon interval. In park mode, the slave's status as an active piconet member is suspended and the 3-bit slave address is replaced by an 8-bit Parked Member address [Blu03].

2.2.7 Power consumption

Power consumption must be a consideration in developing a process for adding new nodes to a network. The power consumed by the device is dependent on the BT device's state/substate as shown in Table 2.1.

2.3 Bluetooth Packet Error Rate

As BT devices become more common, the probability of BT networks, or piconets, sharing the transmission space increases. Since piconets typically share the same spectrum, packets from piconets may share the same channel, or collide, and

Table 2.1: Bluetooth system power consumption at 3.3V [KaL01].

CPU powered down, Bluetooth detached	<11 mW
CPU running, Bluetooth detached	29 mW
CPU running, Bluetooth standby	50 mW
CPU running, Bluetooth inquiry-scan mode	100 mW
CPU running, Bluetooth inquiry mode	200 mW
Bluetooth transmit mode	94 mW
Bluetooth receive mode	94 mW

prevent successful packet reception. Similarly, piconets are expected to form larger networks, or scatternets (cf., Section 2.5), where the piconets are within transmission range by design. Thus, the packet error rate (PER) due to collisions must be characterized to determine the necessity of collision avoidance techniques in scatternet organization and design. To potentially lower PERs, Adaptive Frequency Hopping (AFH) was added to BT standard v1.2 which allows devices to exclude certain frequencies. Piconets may not use the full spectrum if they limit themselves to a subset it through AFH. Establishing the PER between piconets will enable educated decisions for piconet placement in the BT spectrum.

Current estimates of the PER assume frequencies in the hop sequence are independently and uniformly selected. This produces an expected PER of 0.0148 [ElH01] [HKZ02]. Thus far, neither the distribution nor the variance of the PER between arbitrary piconets has been derived.

Although some simulated PER data has been collected and an attempt was made to explain the distribution, the complex interaction between the piconet hop sequences have not been adequately defined or analyzed [Bal03] [HaZ02]. Rather, only a general description of the nature of the BT spectrum partition and an estimate of the upper bound on the PER is available [Bal03] [HaZ02] [PBK03]. Periodic patterns in the correlation between hop sequences were observed via simulation and the expected PER with BT hop sequences was shown to be similar to but distinguishable from that achieved by a uniform random hop sequence [HaZ02]. Available

simulation PER data was broken into *synchronized* and *unsynchronized* cases, using yet a third definition of ‘synchronous’ [Bal03]. Due to temporal alignment and the length of packets, a single-slot packet from a piconet may be vulnerable to collision by either one or two packets from a neighboring piconet. In this case, ‘synchronized’ means a BT packet is vulnerable to collision from a one packet from a neighboring piconet rather than two packets. The distributions from the simulation data are shown in Figure 2.11a and b.

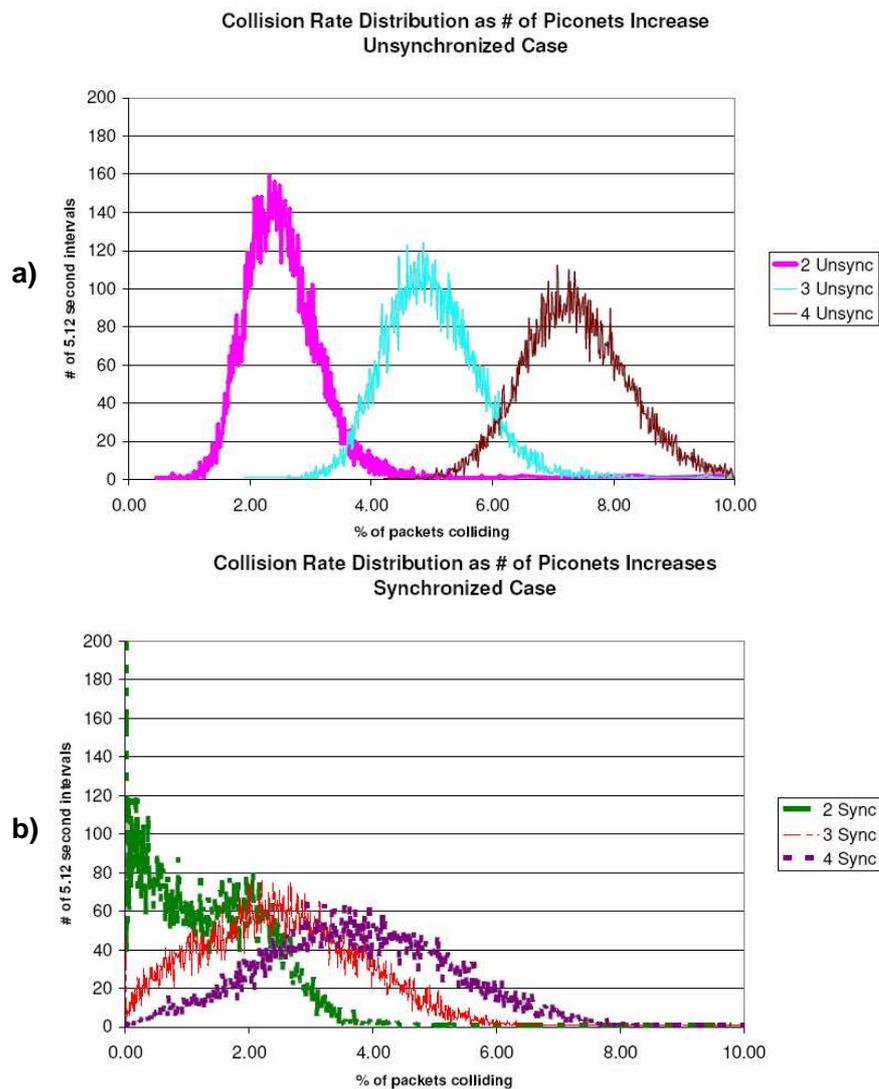


Figure 2.11: PER simulation data for a) packets temporally overlapped by one interfering packet b) packets overlapped by two interfering packets [Bal03].

No empirical PER distribution data has been published. Empirical data is difficult to obtain since parameters that dictate PER for different piconet pairs, the clock phase and address, remain constant for each piconet device. As shown in Chapter VI, there are 828 billion combinations of relationships which can affect the PER. Therefore, it is very difficult to collect enough data to create an accurate distribution estimate.

2.4 Bluetooth Inquiry Time

The Bluetooth standard [Blu03] recommends a device remain in the inquiry substate for 10.24 s. Current estimates on the time required to discover a device using v1.1 of the specification are based on simplifications that underestimate the needed inquiry time [SBT00] [SBT01] [ZaG04] [KaL01]. For example, an inquiry time estimate uniformly distributed on [1.25, 659.375] ms assumes a scanning device is continually receiving and a single frequency train is used. This assumption is not consistent with current implementations [KaL01]. Even assuming the scanning device periodically receives during an 11.25 ms window, the estimated inquiry time is uniformly distributed on [0.001, 1.94]s [KaL01]. Additionally, simulation models and experiments on inquiry time probabilities have been conducted [KaP02] [Leo03]. Some of the results were in rough agreement, such as Figures 2.12a and 2.12b, while those in Figure 2.13 were quite different. However, they all suggest that an inquiry time of 10.24 s is excessive and that the mean inquiry time is approximately 2.2 s. Furthermore, researchers recognized that a rigorous analysis of the probability distribution of inquiry time would be difficult but useful [KaP02] [Leo03]. It has been shown that the discovery time can be reduced by reducing the back-off time [ZaC02].

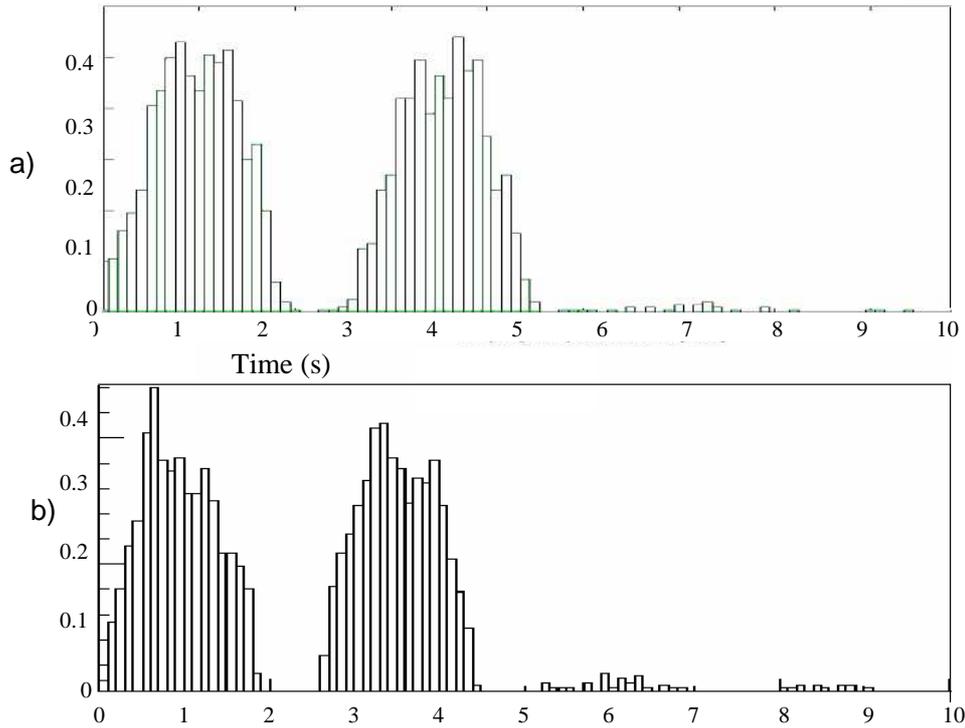


Figure 2.12: Estimated probability density functions for v1.1 inquiry scan time a) simulated data [KaP02] b) measured data [Leo03].

2.5 Bluetooth Scatternets

Although not well defined in v1.2 of the Bluetooth specification [Blu03], piconets can form a clustered structure called a scatternet. A node can be a member of multiple piconets, acting as the master in one at most. In the rest, the node acts as a slave. The formation, maintenance, and packet routing in scatternets are not addressed in the specification.

Current research focuses primarily on the organization and scheduling of scatternets for maximum throughput but does little to address the process of inducting members into the network after initial formation. Most algorithms presume the outreach problem has been solved, yet an actual solution is currently lacking as noted in [Sto02] [LiS02] [SBT01].

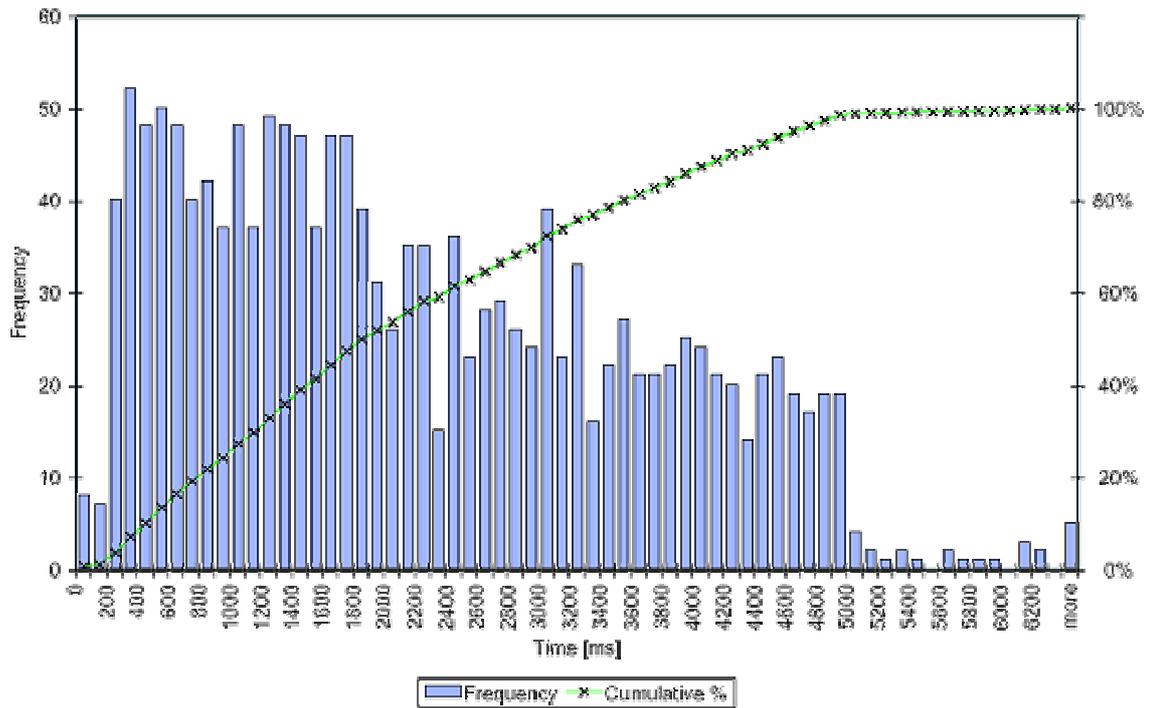


Figure 2.13: Measured probability density for standard inquiry scan [KaL01].

2.5.1 Scatternet Organization

Four basic structures have been proposed for scatternet architectures and are presented below. The structure of a scatternet should have little effect on an efficient outreach process unless each slave in a piconet is also a master in another piconet. In such cases, the option of designating slaves with little Bluetooth traffic as primary components of the outreach process is limited since some outreach methods may not allow master devices to participate in the outreach process. The four structures presented are representative of those found in current research; but there is limited research on outreach procedures or their impact on network throughput. A common assumption is all nodes are within range of each other. This may be a realistic scenario in a conference room, but unlikely in a large office space or auditorium; BT devices typically have a range of 10 meters (Class 2 devices) but it can be as much as 100 meters (Class 1 devices) [YiN02] [Blu03]. A master regulates its power based

on what power level is needed. For example, if a master determines its power level is sufficient to reach all slaves, it will not use a higher power level.

2.5.1.1 Spanning Tree

A ‘Bluetree’ (i.e., a spanning tree) has a central root which acts as a master for other branches which are, in turn, masters of piconets or masters of yet other branches [ZBC01]. The only slaves that are not also masters of a branch are the leaf nodes of the tree. Distributed Bluetrees (interconnected, smaller Bluetrees) can alleviate bottlenecks associated with tree architectures. Since most slaves are also masters, the options for outreach processes are limited.

2.5.1.2 Ring

Two ring architectures, both called ‘BlueRings’, were published almost simultaneously [LTC03] [FoC02]. One proposes a topology similar to a token ring; each master has one slave which is also a master of a slave and so on. This pattern continues until a ring is formed [FoC02]. This structure limits the options in an outreach process since every node is a master. The other architecture proposes a slightly less inefficient ring structure that allows masters to maintain full piconets, but only two of the slaves can be members of other piconets [LTC03]. Slave connections form a ring, with the entire scatternet coordinated to recognize one direction as upstream, and one direction downstream. If a packet is sent through the downstream slave to another piconet and the same packet returns through the upstream slave, it is assumed the destination is no longer reachable.

2.5.1.3 Strand

A strand topology assumes all nodes are within range of each other and form a structure similar to a BlueRing [LTC03]; however the two ends of the strand are not connected to form a ring [LaS01]. The strand is constructed such that all but one

piconet contains seven slaves. In all but the two piconets at the ends of the strand, two of the slaves act as bridges to the neighboring piconets. Although considerable restructuring is required when nodes are added to the scatternet, the development of the outreach process is not limited by structural constraints.

2.5.1.4 Mesh

Mesh (or web) structures are typically more complex and harder to coordinate than the previously mentioned architectures, but have the advantage of a much lower mean delay. In one of the first mesh architectures proposed for Bluetooth [SBT01], all nodes are assumed to be within range of each other. The node with the largest device address collects information on all of the available nodes and coordinates the construction of a mesh network. Other methods using meshes do not require nodes to be within range [BaP02]. These mesh methods also construct the network on the basis of device addresses. If a device has the highest address in an area, it becomes the master of as many devices as it is capable of. ‘Bluenets’, on the other hand, are constructed by requiring random masters to acquire as many slaves as possible while forbidding slaves from joining piconets that already contain a member of any of the other piconets it is already a member of [WTH02].

Mesh structures have been proposed that are organized according to quality of service (QoS) requirements [CMB03]. In these types of structures, each node advertises specific characteristics desired in a piconet. Masters with the desired characteristic add that node to the network until its throughput capacity has been reached. Nodes can, for example, seek a high quality link or speed.

2.5.2 Scatternet Scheduling

Scheduling in Bluetooth scatternets is difficult because of the limitations on piconet nodes. A node can only be present in one piconet at a time since each piconet has an independent frequency hopping pattern. If a node is a slave in several piconets,

it not only has to divide its time between each, but it has to devote two time slots for synchronization with each piconet and may have to perform maintenance functions such as inquiry. Most scheduling and organization algorithms do not allow masters to be bridge nodes between piconets. If a master leaves to participate in a piconet in which it is a slave, the piconet it is the master of ceases all communication since the master is no longer present to coordinate traffic. This inefficient situation is generally avoided. High throughput with minimum buffering occurs when the master piconet has timely access to each bridge node. However, the master must not be consumed with servicing the bridge nodes; it must also communicate with the other slaves in the piconet. A bottleneck occurs if several bridge slave nodes arrive in a piconet at the same time with buffered data from the bridged piconet(s) to pass on. The master cannot serve them all simultaneously and a bridge node may waste considerable time waiting to access the master. If scheduled correctly, the bridge node can spend additional time in other piconets which are waiting to use them for data transfer or in their bridge capacity rather than wait for contact in a busy piconet.

Scheduling algorithms may use Bluetooth's park, sniff, or hold modes to allow a member to temporarily leave a piconet to join another. In park mode, the master retains knowledge that a slave is in the area, but the slave gives up its active status and its 3-bit identifier with the master. In the sniff and hold mode, a slave retains its active status, but the master remains out of contact with the slave for an agreed upon period of time. The specific synchronization algorithm may impact an outreach procedure, but impacts are unlikely to be significant when comparing outreach methods. Since most scheduling algorithms depend on scheduled meetings or appointments, scheduled meetings typically have priority over any outreach duties a node may be tasked with.

Most synchronization algorithms concentrate on determining the duration of time spent outside each piconet or, given a fixed duration, whether to actually switch to a different piconet at the expected time. A credit scheme based on last piconet

visit time and number of packets transmitted to a bridge device has been proposed [BFK01]. If the current piconet has more credits than a piconet with a scheduled sniff/hold meeting, the sniff/hold meeting is ignored. If a link is relatively unused, it distributes its ‘extra’ credits to those with more data to transmit. A simulation of the scheme used a scatternet organized by a tree structure with three piconets and five nodes. The goodput was measured using credits adjusted exponentially and linearly. Although the exponential adjustment performed better, especially with bursty traffic, packet size was not specified, making it difficult to interpret the significance of the 300 kbps goodput. However, goodput is adversely affected when the dwell time within a piconet falls below 5 MSTs (i.e., 10 time slots).

Randomly generated appointment times between each bridge slave and all of the masters it serves have been proposed [RMK01]. When an appointment is kept, packets are exchanged until neither have any remaining packets to exchange or one has to leave due to another appointment. The frequency of appointments is based on the utilization between the two devices. If a master/slave pair have packets to exchange for the duration of every meeting, the algorithm increases the frequency of appointments. If one of the devices fails to arrive at scheduled appointments or the devices have few packets to exchange, appointments become less frequent. A simulation model compares the method to an “ideal” schedule where all nodes have complete knowledge of the buffer contents of all other nodes as well as a random schedule. A tree structure with a root (Network Access Point) and up to seven branch piconets (laptops), each with one slave (a mouse) was used. Bulk data sent from the Network Access Point to each laptop and 2.56 kbps of data sent from each mouse resulted in a throughput of 460 kbps. A packet size was not specified, making it difficult to interpret the significance of the throughput performance.

A variant of this approach is called ‘Rendezvous’ windows [JAJ01]. Masters track when Rendezvous windows occur for each bridge node. At the beginning of a window, each master checks to see if the slave is present. If so, the slave will then

be present until the next window. If not, the slave is in a different piconet until the next window. The slave may use a credit-type method to determine which piconet to switch to for each Rendezvous window. No performance data is available for this approach.

The Load Adaptive Algorithm is more complex since it uses several factors to determine when to switch piconets [HKZ02]. Like [RMK01], it switches to a new piconet if neither device in the current piconet has packets to send. However, it considers the queue size of all of the piconets, as well as how long it has been since it has visited individual piconets. Before leaving a piconet, it uses queue lengths to calculate the maximum time it will be absent from the current piconet. It is similar to the appointments in [RMK01] but it is not random and the bridge may return before the appointment time. Mean delay for packets was determined using an OPNET simulation of a scatternet with two piconets connected by a slave bridge for a total of nine nodes. Single-slot packets were generated with equal likelihood of being destined for any node in the network according to a Poisson arrival process. Plots showed the change in delay with varying appointment times and arrival rates. When stable, the mean delay rose slightly as the arrival rate rose. With an appointment of 32 time slots, the mean delay rose from approximately 45 time slots (28.125 ms) to 60 time slots (37.5 ms) as the arrival rate rose from 10 packets/second to approximately 75 packets/second. With the appointment time extended to 112 time slots, the mean delay time rose from approximately 85 time slots (53.125 ms) to 200 time slots (125 ms) as the arrival rate rose from 10 packets/second to approximately 100 packets/second.

No outreach methods or consideration for packet collisions were evaluated in any of the scheduling methods.

2.5.3 *Intra-piconet Scheduling*

Several scheduling methods have been proposed for scheduling packet delivery within a piconet. The methods include Round Robin polling, Exhaustive polling which allows a master/slave to exchange all packets, and E-Limited Exhaustive which puts a slot limit on an otherwise exhaustive exchange between a master/slave [CKG01] [CMM04] [MiM03]. Additional methods include those that look at the number of packets buffered between a master/slave pair to schedule packet transmission [CMM04]. Others focus on using traffic levels to determine T_{poll} , the maximum time that a master allows an active slave to go uncontacted. Those with less traffic may be contacted less often [RBK01]. It has been shown throughput is maximized if both the master and slave packets contain data, avoiding POLL/NULL packets which contain no user data [KBS99].

2.5.4 *Outreach*

Although no current research specifically addresses outreach (i.e., the process that a scatternet uses to discover nodes and be discovered), some do mention the topic. Many propose a method for initially establishing a piconet that consists of alternating between inquiry and inquiry scan substates so that all nodes consistently search for and are available to discovery by other nodes and neighboring scatternets [BaP02] [WTH02] [Sto02] [LiS02] [ZaG04] [SBT01]. The common theme for outreach is that the outreach process continues throughout the operation of the scatternet in a manner similar to that used when forming the scatternet. Thus far, [Sto02], [LiS02], and [SBT01] directly acknowledge the difficulty of maintaining throughput while entering the inquiry or inquiry substate for outreach. They also acknowledge a solution has not yet been found.

2.5.5 Collision Avoidance

It is shown in Chapters V and VI the interference between an inquiring node and a piconet as well as the inter-piconet interference is significant. Thus, developing an efficient outreach protocol includes avoiding collisions between inquiring nodes in the scatternet and any piconets operating in the connection state. This results in both higher throughput and less energy wasted transmitted inquiry and connection packets which will be disrupted due to a collision with another packet. This can be accomplished by using the AFH mechanism introduced in v1.2 of the BT specification whereby piconets avoid the frequencies used by the discovery process [Blu03]. However, this limits the available frequencies, increasing inter-piconet collisions as well as reducing the security and noise-reduction advantages of FHSS.

Piconets in a scatternet can avoid collisions by further partitioning the spectrum. However, this further limits the FHSS advantages and makes frequency management an additional layer of complexity in organizing the scatternet, to include de-allocating partitions when piconets leave the scatternet. This may also violate FCC directives which requires frequencies receive equal usage, especially if a piconets leaves without reallocating frequencies to neighboring piconets [FCC98]. Other collision avoidance procedures proposed for piconets in connection state are described below.

2.5.5.1 Packet Length Assignment

Using packet length information, certain frequencies in the hop sequence can be avoided [AnK00]. Since multi-slot packets use a single frequency, some frequencies in the hop sequence can be skipped and packet scheduling can be used to avoid collisions. For example, if a 3-slot packet is scheduled to be transmitted by the master and the slave will respond on a frequency known to be noisy, the master can instead transmit a single- or 5-slot packet from the buffer. The slave, then, will not transmit on the error-prone channel. Likewise, slave packet length can be

adjusted such that the next master packet avoids transmission on a noisy channel. This method is limited by the packets available in the buffer, knowledge of channel conditions, and how far the devices are willing to look into the future to orchestrate the multi-slot packet hop sequence.

2.5.5.2 Hop Sequence Knowledge

Popovski has proposed an interference avoidance method, referred to as the Popovski Interference Avoidance Method (PIAM). PIAM uses knowledge of neighboring piconets' hop sequence in a scatternet to prevent packet collisions [PGR03]. By using knowledge of other piconets' hop sequence, throughput is increased and energy is saved by not transmitting packets doomed to collision [PGR03]. In a scatternet, bridge devices can pass the address, clock, and timing offset data from one piconet to the master of another, giving it the capability to calculate what frequencies the other piconet will be using for single-slot packets. By calculating the frequencies used by other piconets, a priority scheme based on the sum of the frequencies used by the master and slave packets determines which device is authorized to use a channel with conflict potential. If a collision is predicted, PIAM compares the sum of the frequencies of the master/slave packets involved [PGR03]. This sum is called the time slot frequency sum (TSFS) [PBR04c]. For packets subject to an imminent collision, the piconet with the larger TSFS transmits while others remain silent.

For example, a collision is scheduled between Piconet 1 and Piconet 2's Slave packets on frequency 47 in Figure 2.14a. Since Piconet 1's TSFS of 79 is greater than Piconet 2's TSFS of 58, Piconet 1 is authorized to use the channel and transmits the Master packet. Piconet 2 does not transmit the Master packet on frequency 11 since the slave is not permitted to respond on frequency 47. Similarly, in 2.14b, Piconet 1's Master will collide with Piconet 2's Slave Packet. Since Piconet 1's TSFS of 79 is greater than Piconet 2's TSFS of 42, Piconet 2 again does not transmit the Master packet. Piconet 2 may transmit the next Master packet on frequency 12

even though it overlaps temporally with Piconet 1's Slave packet since the packets are transmitted on different frequencies.

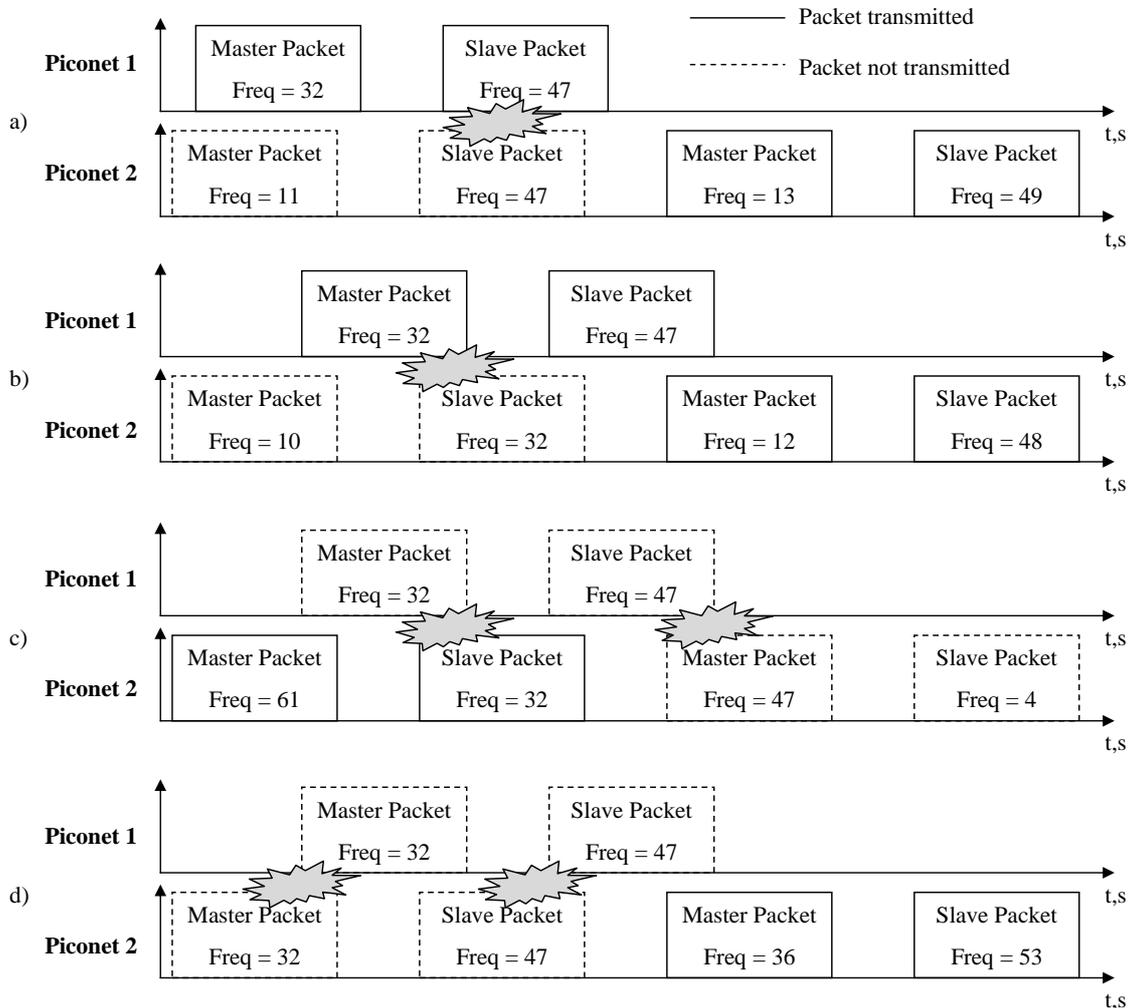


Figure 2.14: Collision avoidance using PIAM a) slave packets collision b) master-slave packet collision c) inefficiency due to master-slave packet collision d) inefficiency due to identical TSFSs.

In 2.14c, PIAM's inefficiency is illustrated. Although Piconet 1 does not transmit its packets due to having a lower TSFS than Piconet 2 when comparing the master/slave pairs containing frequency 32, Piconet 2 does not transmit the second master packet due to the lower TSFS when comparing the master/slave pair containing frequency 47. Even though the interfering packet from Piconet 1 is not transmitted, Piconet 2 refrains from transmitting due to the collision avoidance algo-

rithm. Likewise, in 2.14d neither Piconet 1 nor Piconet 2 transmit due to a scheduled conflict and equivalent TSFSs.

Although unlikely, a MSTs change result in no packets being transmitted as shown in Figure 2.15. As in Figure 2.14d, neither Piconet 1 nor Piconet 2 transmit due to a scheduled conflict and equivalent TSFSs. However, a MSI shift causes Piconet 2's next Master packet to share Piconet 1's frequency and have a lower TSFS.

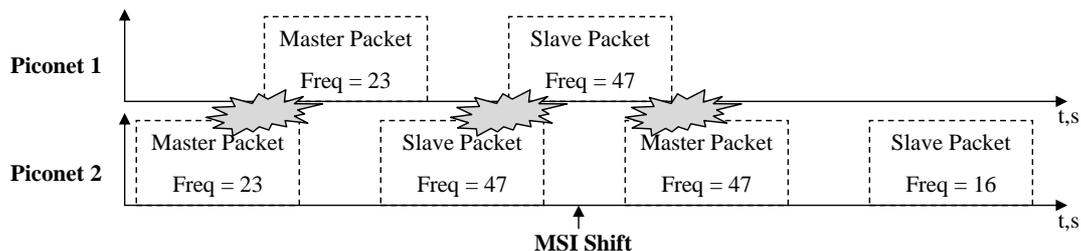


Figure 2.15: Inefficient collision avoidance using PIAM.

Limiting the piconet to single-slot packets, however, places an unreasonable limit on throughput. PIAM can be extended to multi-slot packets if the packet lengths of neighboring piconets are known. Such information is generally unavailable, rendering the method infeasible for widespread use. Likewise, the method is inefficient since neither device is authorized to use the channel when the frequencies in a time slot are the same for both piconets since the frequency sums are equivalent.

As the method generally resolves conflicts between two packets, PIAM generally halves the probability that a packet is transmitted by not successfully received [PGR03]. As multiple piconets are introduced and conflicts are occasionally resolved between multiple packet vying for use of the channel, the probability that a transmitted packet is disrupted is reduced below half.

Additionally, the network becomes more efficient as node do not transmit packets doomed to failure. Defining efficiency as the number of packets successfully received divided by the number transmitted, the efficiency increases by approximately 110% when 6 piconets are present [PGR03].

2.6 Summary

In this chapter, FHSS, clustered, ad hoc networks are defined. The baseband layer of the BT protocol is presented with descriptions of the states and substates used by BT devices. Hop sequence generation for each state/substate is explained. Current research into the methods for scatternet organization are presented. Proposed methods for scheduling bridge dwell times and coordination of piconets are presented with published performance metric data. Additionally, current methods for collision avoidance including using AFH, packet length tailoring, and the PIAM are presented.

III. Analysis Techniques

3.1 Introduction

Although standard queueing network analysis techniques are a valuable tool to determine the performance and operating characteristics of many systems, they are not suitable for all applications. While the arrival process to each node can be modelled using well known stochastic distributions (i.e., as a Poisson or Pareto process), much of a BT network cannot. Much of the analysis in Chapters V through IX models performance metrics within a BT network by directly deriving them using knowledge of the BT protocol. Thus, stochastic analysis techniques such as the convolution of distributions, is required. Simulation models validate the resulting performance metric distributions. Derived, simulated, and measured (when available) distributions are tested for statistical similarity using the Kolmogorov-Smirnov and Cramer von-Mises tests.

In Chapter X, a simulation model is used to present general comparisons of outreach techniques. The deterministic hop patterns in the connection state and inquiry/inquiry scan substates coupled produces a large variance in the relational factors between an arbitrary pair of nodes, making collision modelling difficult in a scatternet performing outreach. Likewise, the multiple substates needed for outreach in a scatternet do not conform to standard network analysis techniques. Although the simulation data used in Chapter X is not complete due to the large number and wide variance in piconet and nodal relationships, confidence intervals are used to bound the expected results.

3.2 Stochastic Analysis

The derivations in this research rely heavily on the Law of Total Probability as the characteristics of BT performance metrics are distinctly partitioned by

parameters such as the 1.28 s intervals between scan windows and intervals of temporal relationships between packets. Additionally, when calculating inquiry time, the random variables representing the time required for various stages in the discovery process add to determine the overall inquiry time.

3.2.1 Law of Total Probability

The probability of a random event occurring can be determined by combining all conditional probabilities of the event occurring under the Law of Total Probability.

Let B_1, B_2, \dots, B_n be mutually exclusive and exhaustive events. Then for an event $A \in \mathcal{F}$ [Kul95]

$$P\{A\} = \sum_{i=1}^n P\{A|B_i\}P\{B_i\} \quad (3.1)$$

where \mathcal{F} is “a set of all possible events of interest” [Kul95].

3.2.2 Random Variable Addition

The probability density function (pdf) of the sum, Z , of two independent random variables, X and Y , is the convolution of their respective pdfs, or [Kul95]

$$f_Z(z) = f_X(x) * f_Y(y) = \int_{-\infty}^{\infty} f_X(x')f_Y(z - x')dx'. \quad (3.2)$$

3.3 Probability Distribution Comparisons

Although it is impossible to definitively determine the population from which data samples have been drawn, the Kolmogorov-Smirnov and Cramer von-Mises tests provide a metric that indicates whether two distributions are statistically equivalent.

$$H_0 : C_1(x) = C_2(x) \quad \forall x \in \mathfrak{R} \quad (3.3)$$

Hence, both test equivalence of the CDFs.

3.3.1 Cramer von-Mises Test

Cramer von-Mises test compares a data sample consisting of N events whose cumulative distribution is $S_N(x)$ with a hypothesis function whose cumulative distribution is $F(x)$. For $N > 3$, the test statistic is

$$W^2 = \int_{-\infty}^{\infty} (S_N(x) - F(x))^2 f(x) dx \quad \forall x. \quad (3.4)$$

The null hypothesis that the sampled distribution is not the same as the hypothesis function is rejected if $NW^2 < 0.461$ at a 95% confidence level. For 90% and 99% confidence, the test statistic is compared to 0.347 and 0.743, respectively.

3.3.2 Kolmogorov-Smirnov Test

Kolmogorov-Smirnov test is a simpler test which is also used to compare a data sample consisting of N events whose cumulative distribution is $S_N(x)$ with a hypothesis function whose cumulative distribution is $F(x)$. For $N > 80$, the test statistic

$$D_N = \max(|S_N(x) - F(x)|) \quad \forall x. \quad (3.5)$$

Using a 95% confidence level, the null hypothesis that the sampled distribution is not the same as the hypothesis function is rejected if $D_N\sqrt{N} < 1.36$. For 90% and 99% confidence, the test statistic is compared to 1.22 and 1.63, respectively.

3.4 Summary

The fundamental tools for analysis and derivation have been presented. Due to the complex nature of the BT discovery process, traditional network analysis

models cannot be used. Many of the performance metrics used to determine an appropriate outreach method are derived in Chapters V through IX using traditional probability theory. These are verified by comparing them to distributions from simulation models using the Cramer von-Mises and Kolmogorov-Smirnov tests.

IV. Research Objectives and Methodology

This chapter presents the objectives and methodology used throughout this research. Much of the research consists of characterizing performance metrics and interference between clusters, or piconets, in the larger network as well as between the piconets and nodes in the discovery process. Characterization of these performance metrics is necessary for outreach method analysis, but the characterizations are either unavailable or unreliable in literature. Once these performance metrics are characterized, the outreach method performance is evaluated. The steps used in this research (listed below) are discussed in this chapter.

1. State research thesis and objectives
2. Define system boundaries and assumptions
3. Define and select performance metrics of interest
4. List parameters of the system
5. Characterize the impact of the relevant performance metrics and parameters.
6. Propose improvements developed in characterizing performance metrics and parameters
7. Propose outreach methods
8. Compare competing outreach methods using simulation model

4.1 Research Objectives

The fundamental thesis of this research is that the throughput and time needed to locate new nodes can be improved by allowing the nodes in a FHSS, clustered network to passively scan for nodes arriving in an area rather than actively searching for them on a regular basis.

The objective of this research is to develop a more efficient method for scatternet outreach, including an inquiry/inquiry scan schedule for nodes if appropriate, while improving network throughput, packet delay, and the time to find arriving nodes.

To meet this objective, this research addresses the following specific areas:

- a) The interference between an inquiring node and BT piconet is characterized. Since the impact of an inquiring node on piconet performance has not heretofore been determined, the impact of proposed outreach methods on piconet throughput cannot be adequately assessed.
- b) The time needed to discover a BT device is characterized. This is essential for determining the duration that a node is in the inquiry substate and unavailable for data transmission as well as how long a node should transmit inquiry packets during the outreach process.
- c) The time needed to discover a BT device is analyzed when packet collision avoidance methods are used. Using knowledge of neighboring piconets' hop sequence allows inquiring nodes to avoid transmissions that may disrupt data packets. This may delay discovery time and affect the time a node remains in the inquiry state. Additionally, the impact of multiple inquiring nodes on the discovery of a node is characterized.
- d) Packet interference between piconets in the connection state is characterized. Since packet error rates are critical to determining throughput, characterizing the relationship between the hop sequences in neighboring piconets is crucial to formulating schemes to maximize throughput through collision avoidance. Although models for the interference have been proposed, none capture the actual relationships between piconets or provide an accurate performance analysis of collision avoidance techniques.

- e) A packet collision avoidance methodology is developed by extending proposed interference avoidance methods. Using knowledge of the neighboring piconets' hop sequence allows piconets to avoid packet transmission which may disrupt other data packets. A complete performance analysis of the method is presented and compared to the performance of a previously proposed collision avoidance method.
- f) Finally, an outreach method is proposed to increase throughput and decrease packet delay time as well as time to discover arriving node. The performance of method is compared to that of currently proposed methods.

4.2 System Boundaries and Assumptions

The system considered in this research consists of a BT scatternet with an arriving node that needs to be discovered to join the scatternet. The following are the boundaries and assumptions for the system.

4.2.1 Baseband Layer

This research is limited to impacts on the BT v1.2 Baseband layer. Noise sources which may affect the physical layer of the protocol other than packets from BT devices are assumed to be negligible. Similarly, the packets are considered to be transmitted at a power level sufficient to be properly received unless disrupted by another BT packet. Although built into the simulation model to allow nodes to have limited range, the extensive simulation time and limited benefit of including range limitations warranted its exclusion from analysis.

Likewise, higher layers of the protocol, including packet processing and packaging as well as scatternet organization is assumed to be present.

Inquiry time is characterized for v1.1 of the specification in addition to v1.2 of the specification since many current devices are based on v1.1 and the distribution

of the v1.2 inquiry time is one of the stages in the derivation of the inquiry time for v1.1.

4.2.2 Propagation Delay

Since BT devices are assumed to have a typical range of 10 meters, with a maximum of 100 meters, propagation delay for packet transmission time is assumed to be zero.

4.2.3 Standard Configurations

The BT standard allows variation in some parameters, such as the time between scan windows and maximum back-off time. Unless otherwise noted, default parameters are used. The scan windows used in the inquiry scan substate are 11.25 ms in duration and begin every 1.28 s. The minimum back-off duration in both v1.1 and v1.2 is 1023 time slots. Adaptive Frequency Hopping is not used.

4.2.4 Scatternet Organization

It is assumed that a protocol to organize the scatternet efficiently is used. Once an arriving node is discovered and accepted into the piconet, this organization algorithm is responsible for assigning the node to an appropriate piconet or directing the creation of a new piconet within the scatternet. The focus of the research is to efficiently discover the arriving node, not the organizational responsibilities once the node has been discovered. Considerable research has been devoted to scatternet organizations, some of which is reviewed in Chapter 2.

4.2.5 Scatternet Scheduling

Similarly, research devoted to bridge and packet scheduling within the piconet has produced multiple and sometimes complex scheduling algorithms. For simplicity,

bridge dwell times are assumed to change according to a uniform random variable. Due to piconet synchronization delays when changing piconets, the minimum piconet dwell time is established at 5 MSTs to allow a bridge node to synchronize and reside in a piconet long enough to justify the change.

Intra-piconet scheduling uses a prioritized FIFO method within the master with a maximum contact interval, T_{poll} . Although individual packets are assumed to be of equal priority, slave devices in the piconet not contacted within T_{poll} have the highest priority. If several slave nodes have not been contacted in T_{poll} , the node with the oldest buffered packet is contacted first. If all slaves currently in the piconet have been contacted within T_{poll} , the next priority is the oldest packet going to a slave that has packets buffered for the master. This maintains throughput while reducing delay as the number of packets going in both directions (master-slave, slave-master) is maximized. If no such conditions exist, the master transmits the oldest buffered packet. If it has no buffered packets, the master generates a packet to transmit. Slaves transmit the oldest packet destined for the master of the piconet in which it is active. If no packets are buffered, the slave generates a packet. It is assumed that the slave uses a bit in the *LT_ADDR* field to return a flag to the master indicating whether it has more packets buffered for the master.

4.3 System Performance Metrics

4.3.1 Discovery Time

The most relevant metric in evaluating an outreach method is the time required for a scatternet to discover an arriving node. However, this is not the most important metric in the overall system. Discovery time must be balanced with the impact on other, more important performance metrics. Since a scatternet consists of clustered networks, or piconets, discovery time by multiple piconets is relevant in addition to initial discovery by a single node in the scatternet.

4.3.2 Throughput/Goodput

A common system performance metric used in network analysis is the bits transmitted each second, or throughput. Since BT allows transmission of packets consisting only of overhead bits, a more useful metric is goodput, or the number of user data bits. Depending on the application, system goodput or nodal goodput may be more useful in comparing systems. Assuming a saturated network, goodput is dependent on the PER.

4.3.3 Mean Delay

The time required for packet delivery to the destination node after acceptance by its source node, or mean delay, is also a relevant performance metric. An efficient outreach method should not significantly increase mean delay from that experienced with no outreach method in place. Delay is dependent on the source and destination of the packet. For example, master-to-slave and slave-to-master packets within a piconet only require one time-slot delay. However, slave-to-slave packets within a piconet require a minimum of two time slots since the packet must be passed to the master before being delivered to the destination slave. Packets leaving a piconet are subject to buffer delays and bridge change times in piconets in the route to the destination node.

4.3.4 Packet Error Rate

Since a perfect channel is assumed, packet errors only occur due to collisions with other BT packets. PER is significant in determining the goodput, although the two metrics are not necessarily directly related. Since a successfully received packet from the master to a slave must be repeated if the header of the slave response is disrupted, disrupting the ACK to the master, a single collision may effectively disrupt two packets.

4.3.5 Buffer Size

The average and maximum size of the packet buffer at each node is needed to determine adequate buffering resources at each node for each outreach method.

4.3.6 Packet Generation Time pmf

Using the packet generation scheme described in Section 4.2.5, which places the network in a saturated condition, the pmf of packet generation times is observed and analyzed for the different types of nodes (i.e., master, bridge, slave).

4.4 System Parameters

In a BT scatternet, there are a large number of parameters that can significantly affect performance.

4.4.1 Network Topology

A mesh structure of five piconets with a total of twenty-four nodes using single-slot packets shown in Figure 4.1 is used for outreach method comparison in Chapter X. Although the outreach methods are applicable to any size scatternet, this size was chosen to allow enough nodes to pose congestion and interference problems without requiring overwhelming processing power for simulation. Additionally, each piconet is connected to all other piconets. Each bridge device connects two piconets. The mesh structure is a representative model as it provides a similar number of master and bridge nodes as strand or ring structures. Spanning trees are unlikely to be widely used if master devices must also serve as bridge nodes. Further, it requires fewer hops than the other structures, which produces higher goodput.

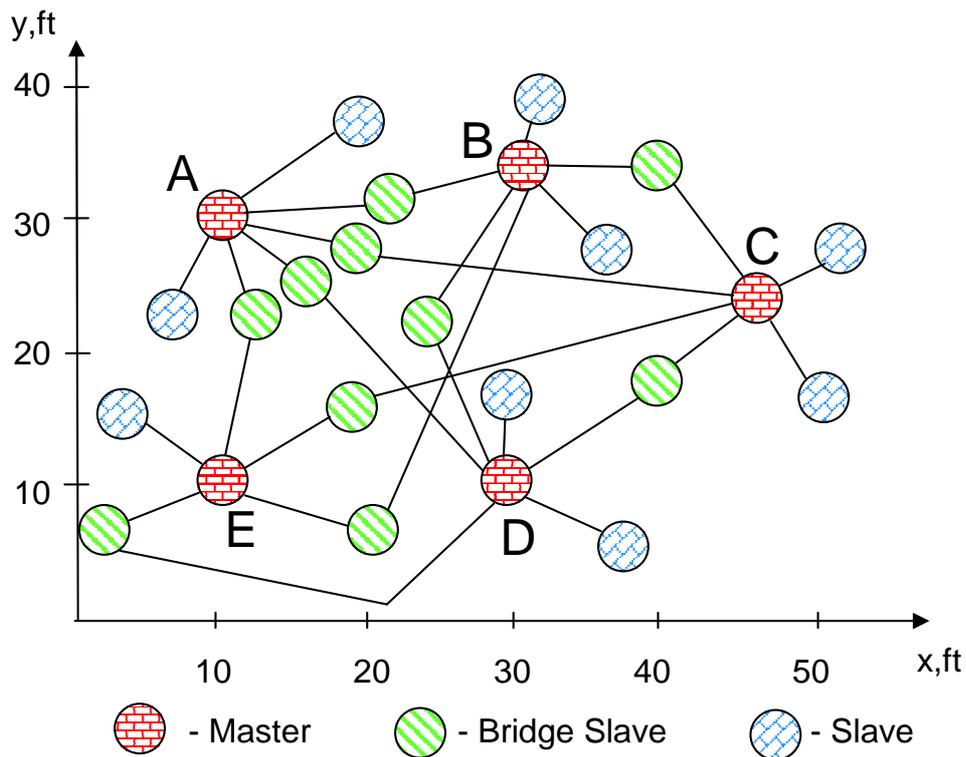


Figure 4.1: Scatternet configuration used for outreach comparisons.

4.4.2 Packet Distribution

In a scatternet, each application and scenario produces different distributions of master-slave/slave-master and slave-slave packets within a piconet as well as packets leaving a piconet. Additionally, POLL/NULL packets may be common in some applications. Since there is no standard distribution for BT packet characteristics, a distribution was chosen to produce buffering of packets at the master and bridge nodes without overwhelming them. If a large number of packets are generated which leave the piconet, bridge nodes may be overwhelmed and be unable to generate packets without seriously affecting network performance.

Initially, it was anticipated that NULL packets from slaves may be necessary to prevent master buffers from being overwhelmed as some packets are disrupted and must be repeated. Preliminary tests indicated this was not the case. Therefore

POLL/NULL packets are not used. Thus, packet arrival distributions that saturates the network can be determined.

In order to exercise, but not overwhelm bridge nodes, 70% of the generated packets are destined for nodes outside the piconet. Of those remaining in the piconet, 70% of slave packets are destined for other slaves in the piconet, leaving 30% with the master as the destination. All master packets destined within the piconet are transmitted directly to the destination slave.

4.4.3 Outreach Methods

At the onset of this research, prior to characterization of the interference and inquiry time, a wide variety of outreach methods were anticipated. The expected possibilities included

1. A commonly proposed method with all nodes alternating between inquiry and inquiry scan substates; arriving nodes are discovered in either the inquiry or inquiry scan substate; disjoint scatternets quickly find each other
2. Limit the inquiry substate to slaves only, all nodes scan regularly
3. Limit the inquiry substate to slaves only, only slaves scan regularly
4. Limit the inquiry substate to non-bridge slaves only, all nodes scan regularly
5. Limit the inquiry substate to non-bridge slaves only, only slaves scan regularly
6. Nodes in the scatternet rarely enter inquiry substate, all nodes scan regularly
7. Nodes in the scatternet rarely enter inquiry substate, only slaves scan regularly; arriving nodes must inquire

Collision avoidance between piconets was a consideration as part of the outreach method. If feasible, it was expected that inquiring devices could avoid collisions with piconet data packets. Additionally, since the interference effects of an inquiring node (with no collision avoidance) was unknown, a possible option also included

attempting to detect inquiring nodes that were not part of the scatternet to immediately notify them to cease inquiry operations as they degrade scatternet performance. The scatternet would then inquire for the arriving node. Since the option to notify an inquiring node to cease inquiry operations is not in the BT protocol, it would have to be added to the specification.

It was also anticipated that other options might arise as the protocol was characterized. However, once the inquiry time and interference were characterized, the options listed above were reduced to three: limiting the inquiry substate to all slaves while all slaves also scan regularly, limiting the inquiry substate to non-bridge slaves while all slaves scan regularly, and finally, allowing slave devices to scan regularly while nodes in the scatternet rarely enter the inquiry substate. The rationale for selecting these options is presented in Chapter X.

4.4.4 Packet Length

Using multi-slot packets can greatly increase throughput as the ratio of user data to overhead bits and oscillator re-tuning time increases. However, since the radio remains on a single frequency for the duration of the packet, it is more vulnerable to disruption than single-slot packets.

Only single-slot packets are used to characterize the impact of inquiry packets on piconet traffic. Although this provides the best-case scenario, the impact on multi-slot packets can be easily extrapolated as discussed in Chapter VI.

To characterize packet interference between piconets, all combinations of packets are used: single-slot packets in one piconet with single-slot packets in another piconet, single-slot packets in one piconet with three-slot packets in another piconet, etc. Since no typical distribution of single-, three- and five- slot packets has been published and packet size varies by implementation, mixtures of packet sizes are not used. Thus, the interference performance is characterized for piconets which consistently use each of the combinations of packet sizes.

The impact of packet size within the scatternet can have a significant effect on goodput and PER. However, the impact on the comparison of outreach methods is expected to be minimal due to the drastic differences in the tested methods. Therefore, only full single-slot packets are used in comparing outreach methods.

4.4.5 Collision Avoidance

Although the effect of two collision avoidance methods are characterized in this research, collision avoidance is not used to compare the outreach methods. The effects of collision avoidance can be extrapolated, regardless of outreach method. Therefore, to accelerate the simulation model, collision avoidance methods are not included.

4.4.6 Range

All devices are within transmission range of each other. Although this is not always a valid assumption in practice, it removes ambiguity when assessing the effects of specific devices on the performance of the system.

4.4.7 Bridge Dwell Time

Complex scheduling algorithms for slave-bridge dwell time have been developed to maximize throughput (cf., Section 2.5.2). The dwell time is uniformly distributed on [10,50] time slots. A minimum dwell time of 5 MSTSs is selected to ensure the device has adequate time to synchronize and exchange packets with the master [BFK01]. A periodic dwell time is not used since several bridge devices may consistently arrive to a piconet nearly simultaneously. It is unlikely the master can service each of the bridge nodes adequately in such cases. A maximum of 25 MSTSs is selected since the delay time for packets passing through bridge devices increases as the bridge dwell time increases.

4.4.8 Inquiry Timing

Inquiring nodes within the scatternet remain in the inquiry substate for 3.84 s. Although the specification recommends an inquiry substate dwell time of 10.24 s, it is shown in Chapter V that the inquiry time must be as short as possible while it is shown in Chapter VIII that 99% of devices can be located in 3.84 s using the standard inquiry scan (1.28 s using interlace inquiry scan) in a noiseless environment. With multiple devices transmitting, the channel is not necessarily noiseless. However, with multiple devices in the inquiry substate, discovery of the arriving node by members of the scatternet is likely.

The time until re-entry into the inquiry substate is uniformly distributed on [0,40] s. Although specification suggests entering the inquiry substate every 60 s, even 40 s is considered a long delay for an arriving device to be discovered if it is relying entirely on inquiry scans to be discovered. Arriving nodes that inquire enter the inquiry substate for 3.84 s and do not re-enter.

4.4.9 Inquiry Scan Type

The standard inquiry scan from v1.2 of the specification is used to evaluate outreach methods. As few v1.2 compatible devices have been produced, it is unknown which inquiry scan type will be most prevalent. The type of scan should have little effect on the goodput of a piconet with several slave devices or the comparison of the outreach procedure. Using the interlaced inquiry scan may reduce discovery time but it requires additional time in the inquiry substate where data transmissions cannot occur.

4.4.10 Device Arrival Time

In the outreach method comparison, devices arrive to the scatternet at a time uniformly distributed on [6.52, 12.5] s, thereby allowing the network to stabilize before discovery is attempted.

4.5 Performance Metric and Parameter Characterization

In Chapter V, the interference posed by an inquiring device to a piconet transmitting single-slot packets is characterized via analytical and simulation models. Additionally, the models are extended to include multiple inquiring nodes.

In Chapter VI, the PER and goodput for multiple piconets is characterized via analytical and simulation models.

In Chapter VIII, the time for an inquiring device to find an arbitrary device in the inquiry scan substate is characterized via analytical and simulation models. The inquiry scan from v1.1 of the specification, as well as the standard and interlaced inquiry scans from v1.2 of the specification are included. The impact of multiple inquiring devices on inquiry time is also characterized as well as the impact of a collision avoidance method for inquiring devices attached to a pico- or scatternet.

Since the inquiry time models match most, but not all available measures of inquiry time data, the variation is explained in Chapter IX. A specification-compatible simplification to the inquiry substate is presented and fully characterized via analytical and simulation models.

4.6 Performance Metric Improvements

A collision avoidance method designed to prevent collisions between single- and multi-slot packets is presented in Chapter VII and compared to a proposed collision avoidance method for single-slot packets.

In Chapter IX, an improvement to the inquiry substate is characterized. Although not advertised by the device manufacturers, the improvement was implemented in tested devices. The change simplifies the inquiry substate implementation and decreases the inquiry time.

4.7 Outreach Method Proposals and Evaluation

Outreach methods mentioned in Section 4.4.3 are evaluated in Chapter X. The candidate methods are reduced to three based upon the results of the performance metrics characterization. The performance metrics of interest of three proposed outreach methods are compared via simulation models.

4.8 Summary

The objectives and methodology used in this research were presented in this chapter. Section 4.1 listed the research thesis and objectives. System boundaries and assumptions were presented in Section 4.2. The performance metrics used in evaluating the Outreach methods were discussed in Section 4.3. Section 4.4 contained descriptions of the system parameters used in the research. A description of the analytically characterized performance metrics was presented in Section 4.5. Section 4.6 listed the methods for improving metric performance which are presented in this research. Finally, Section 4.7 described the method for reducing and evaluating the proposed outreach methods.

V. Inquiry Substate Interference

The key to successfully establishing and maintaining a BT scatternet is the inquiry procedure that discovers BT devices within range. While a piconet master in the scatternet is in the inquiry substate, other piconets may experience interference from concentrated transmissions in the inquiry frequency partition. Likewise, if each piconet master delegates its inquiry process to one or more slave nodes, all the piconets may experience significant interference. Further interference can be caused by nodes arriving in the vicinity entering the inquiry substate in an attempt to join a pico- or scatternet.

In this chapter, an analytical model characterizing the interference between a piconet and a single inquiring node is derived. The model is subsequently extended to determine the interference of m inquiring devices. Analysis shows a single inquiring node is expected to disrupt 1.3% of the packets in a neighboring piconet using single-slot packets with interference approaching 73% as the number of inquiring nodes increases. Likewise, the probability of disruption increases for multi-slot packets as the packets are vulnerable to twice and three times as many inquiry packets.

5.1 *Inquiry Interference Probability*

The alignment of time slots between a piconet and an inquiring device is significant; it determines the packet's vulnerability to interference. If a piconet master and inquiring device both begin their even time slots at the same instant, the inquiring device will transmit two packets which have a possibility of *directly* interfering with the piconet's Master packet. On the other hand, it is not possible for the piconet's Slave packet to be directly disrupted since the inquiring device listens for a response from devices in the inquiry scan substate while the piconet transmits its Slave packet. Likewise, if the piconet's even slots begin at the same instant as the inquiring devices odd slots, only the piconet's Slave packets are vulnerable to interference. Since BT

device clocks are permitted very little drift, the temporal relationship between the piconet and the inquiring device is essentially constant. The offset of the beginning of an inquiring device's even slots from the piconet's even slots, Δ , is distributed uniformly between 0 and 1250 μs . Thus, the expected vulnerability of the packets can be determined. However, the relationship of Δ to F_1 and F_2 , the frequencies from the Upper and Lower Ranges in the inquiry frequency partition (See Section 2.2.3), is also important. Consider an inquiring device whose even packets are offset by 200 μs (i.e., $\Delta = 200 \mu\text{s}$) as shown in Figure 5.1a. The Master packet is only vulnerable to interference if the MSI overlaps the Lower Range since it can only be affected by the packet transmitted on F_1 . Even if the MSI overlaps the Upper Range, inquiry packets at those frequencies will only be transmitted while the piconet devices are re-tuning their oscillators. In contrast, when $\Delta = 257 \mu\text{s}$ as shown in Figure 5.1b, both the Master and Slave packets are vulnerable to interference. If the SSI overlaps the Upper Range, the Slave packet can be affected by the inquiry packet transmitted on F_2 .

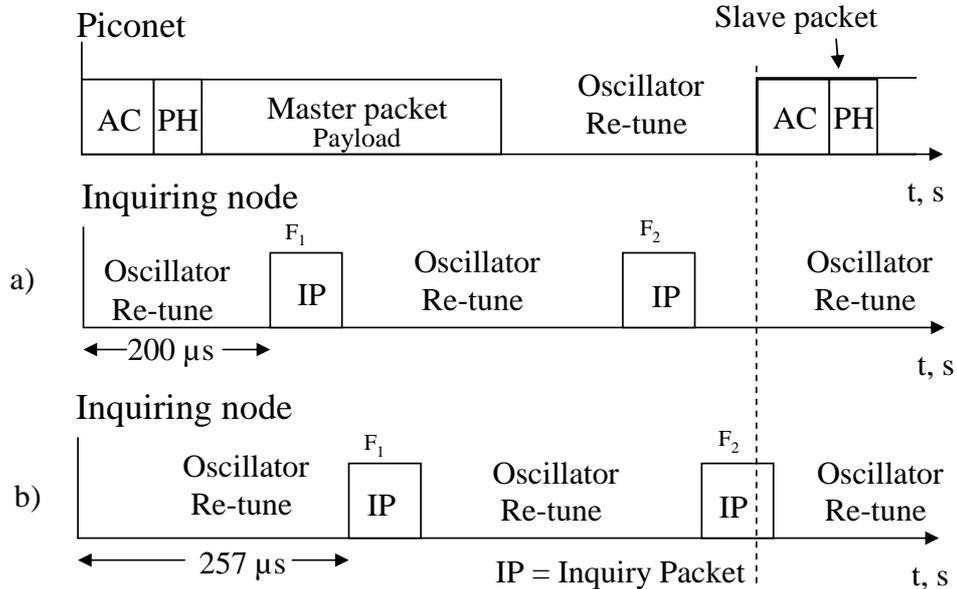


Figure 5.1: Alignment between piconet and inquiring node where a) Master packet is vulnerable only to the packet from the Lower Range (with a frequency F_1) and b) both Master and Slave are vulnerable to disruption.

The vulnerability of packets to interference as a function of the inquiry device offset, Δ , is indicated in Table 5.1 using $I_{\Delta,Intersection}$, where the *Intersection* values {ML,MU,SL,SU} indicate whether a Master (M) or Slave (S) piconet packet can be affected by an inquiry packet from the Lower (L) or Upper (U) Range. For example, if $I_{\Delta,ML} = 1$, the Master packet is disrupted if it simultaneously transmits on the same frequency used by the inquiring device in the Lower Range. Conversely, if $I_{\Delta,ML} = 0$, the Master packet is *not* disrupted even when it simultaneously transmits on the same frequency used by the inquiring device in the Lower Range. The expected number of bits in the payload and the number of bits in the Access Code/Payload Header which must be affected to consider the packet disrupted are denoted by x and y , respectively. The number of bits, x and y , are not constant due to different error correction in single-slot packet types. Generally, $y \cong 12$ due to the Hamming distance of 14 provided by the error correction in the Access Code and constructive/destructive interference [Blu03] [Sk101]. A Hamming distance of 14 allows for error correction of up to 6 bits. However, it is assumed that in half of the cases, the overlap of bits will cause constructive interference and not impact reception of the correct bit. Therefore, 12 bits must be overlapped for the packet to be disrupted. There are cases where the overlap of only 7 bits will cause the packet to be disrupted. Likewise, if the inquiring device is not fully within the range of the piconet, the interference may not be significant enough to disrupt all conflicting, overlapping bits.

When $x = 1$, $y = 12$, and $\Delta = 200\mu s$ as in Figure 5.1a, $I_{200,ML} = 1$ (from the second row in Table 5.1) indicates the Master packet is vulnerable to collision from an inquiry packet transmitted from the Lower Range at F_1 . All other entries in this row are zero, indicating that the Master packet is not vulnerable to a packet at F_2 and the Slave packet is not vulnerable to interference by the inquiring device. Likewise in Figure 5.1b, $I_{257,ML} = 1$ and $I_{257,SU} = 1$ while $I_{257,MU} = 0$ and $I_{257,SL} = 0$. Table

Table 5.1: Piconet packet vulnerability.

$\Delta(\mu\text{s})$			$I_{\Delta,ML}$	$I_{\Delta,MU}$	$I_{\Delta,SL}$	$I_{\Delta,SU}$
0	to	(54-x)	1	1	0	0
(54-x)	to	(245+y)	1	0	0	0
(245+y)	to	(366-x)	1	0	0	1
(366-x)	to	(557+y)	0	0	0	1
(557+y)	to	(679-x)	0	0	1	1
(679-x)	to	(870+y)	0	0	1	0
(870+y)	to	(991-x)	0	1	1	0
(991-x)	to	(1182+y)	0	1	0	0
(1182+y)	to	1250	1	1	0	0
M = Master Packet			U = Upper Range			
S = Slave Packet			L = Lower Range			
x = number of bits interfering with Payload to cause an error						
y = number of bits interfering with Header to cause an error error						

5.1 contains the indicator values for $I_{\Delta,ML}$, $I_{\Delta,MU}$, $I_{\Delta,SL}$, and $I_{\Delta,SU}$ for all possible values of Δ .

If Δ is unknown, $E[I_{\Delta,Intersection}]$ can be used and is calculated by averaging $I_{\Delta,Intersection}$ across all possible values of Δ . For example, if $x = 1$, $y = 12$, $E[I_{\Delta,Intersection}] = 0.337$ for $Intersection \in \{ML, MU, SL, SU\}$. This is obtained by dividing the range of values of Δ which cause $I_{\Delta,Intersection}$ to equal 1 by the range of Δ (*i.e.*, 421/1250).

Even if a Slave packet is not vulnerable to direct disruption, a Slave packet can be *indirectly* lost if the preceding Master packet's AC or PH is disrupted. A slave device cannot determine it has been addressed and thus will not respond to the master. Likewise, if a Slave packet's AC or PH is disrupted, the master will not recognize the response from the slave and will likely retransmit at the next opportunity. Thus, a Master packet can also be lost *indirectly*. Since the impact of

indirect interference is also significant, the vulnerability of the AC and PH is shown in Table 5.2.

Table 5.2: Piconet packet header vulnerability.

$\Delta(\mu\text{s})$			$I_{\Delta,\text{MHL}}$	$I_{\Delta,\text{MHU}}$	$I_{\Delta,\text{SHL}}$	$I_{\Delta,\text{SHU}}$
0	to	(126-y)	1	0	0	0
(126-y)	to	(245+y)	0	0	0	0
(245+y)	to	(439-y)	0	0	0	1
(439-y)	to	(557+y)	0	0	0	0
(557+y)	to	(751-y)	0	0	1	0
(751-y)	to	(870+y)	0	0	0	0
(870+y)	to	(1064-y)	0	1	0	0
(1064-y)	to	(1182+y)	0	0	0	0
(1182+y)	to	1250	1	0	0	0
MH = Master Packet Header			U = Upper Range			
SH = Slave Packet Header			L = Lower Range			
y = number of bits interfering with Header to cause an error						

The probability of interference changes as the MSI/SSI shifts and change its intersection with the Lower and Upper Ranges. The first frequency in the MSI determines the probability of interference. The position of the MSI/SSI is periodic, repeating every 3.16 seconds (i.e., 79·40ms) and can be calculated relative to a known reference point since it shifts by 16 frequencies modulo 16 every 40 ms. Thus, the first frequency in the MSI, $\beta(n)$, is

$$\beta(n) = (16n) \bmod 79 \quad (5.1)$$

where n is the number of times the MSI/SSI has shifted since the first frequency in the MSI was zero.

Let $M_{\beta(n),L}$ and $S_{\beta(n),L}$ be the number of frequencies in the MSI and SSI that overlap the Lower Range at MSI shift n . Likewise, $M_{\beta(n),U}$ and $S_{\beta(n),U}$ are the

number of frequencies in the MSI and SSI that overlap the Upper Range at MSI shift n , respectively, then

$$M_{\beta(n),L} = \begin{cases} n - 21 & 22 \leq \beta(n) \leq 36 \\ 16 & 37 \leq \beta(n) \leq 53 \\ 69 - n & 54 \leq \beta(n) \leq 68 \\ 0 & \textit{otherwise} \end{cases}, \quad (5.2)$$

$$M_{\beta(n),U} = \begin{cases} 6 - n & 0 \leq \beta(n) \leq 5 \\ n - 37 & 38 \leq \beta(n) \leq 52 \\ 16 & 53 \leq \beta(n) \leq 69 \\ 85 - n & 70 \leq \beta(n) \leq 78 \\ 0 & \textit{otherwise} \end{cases}, \quad (5.3)$$

$$S_{\beta(n),L} = M_{(\beta(n)+32) \bmod 79,L}, \quad (5.4)$$

and

$$S_{\beta(n),U} = M_{(\beta(n)+32) \bmod 79,U}. \quad (5.5)$$

5.1.1 Direct Interference Probability

Assuming the MSI overlaps the Lower Range by $M_{\beta(n),L}$ frequencies and the Master packet is vulnerable to interference by the Lower Range ($I_{\Delta,ML} = 1$), the probability of a Master packet being transmitted on a frequency in the Lower Range is $M_{\beta(n),L}/32$ since any frequency in the MSI generally has a $1/32$ chance of being used. There are dependencies between frequencies of Master packets since frequencies in a MSI are used only once but these dependencies are assumed to be insignificant.

If a frequency in the Lower Range is used by the Master Packet, the probability that the frequency is in the train being used by the inquiring device is 1/2 since only 8 of the 16 frequencies in the Lower Range are in the train. Finally, if the frequency is in the train, there is still only a 1/8 chance of interference since any of the eight frequencies in the train in the Lower Range can be selected. A similar analysis applies to the Upper Range as well. Thus, the probability of a Master packet being *directly* (D) disrupted by an inquiry packet is

$$P\left(X_{M,D}(n, \Delta)\right) = I_{\Delta,ML} \frac{M_{\beta(n),L}}{512} + I_{\Delta,MU} \frac{M_{\beta(n),U}}{512} \quad (5.6)$$

where $X_{M,D}(n, \Delta)$ is the event the Master packet was directly disrupted in the n th MSI/SSI after the MSI began at zero ($t = 0$) by a inquiry packet from a node whose even time slots begin $\Delta \mu s$ after the piconet's. Note the event that a packet is disrupted by the inquiry packet transmitted in in the Upper and Lower Range are mutually exclusive. Assuming Δ is not known but uniformly distributed, $E[I_{\Delta,ML}]$ can be used instead of $I_{\Delta,ML}$ for a specific piconet/inquiring device pair. It was determined earlier that $E[I_{\Delta,Intersection}]$ was 0.337 for $Intersection \in \{ML, MU\}$, and $M_{\beta(n),L}$ and $M_{\beta(n),U}$ are added to form $M_{\beta(n)}$. Thus, the probability a packet is vulnerable is

$$P\left(E[X_{M,D}(n)]\right) = \frac{0.337}{512} M_{\beta(n)}. \quad (5.7)$$

The probability of Master packet disruption is periodic and a complete period ($t = 3.16$ s) is shown in Figure 5.2a for $n = \lfloor \frac{t}{40ms} \rfloor$ and an unknown Δ .

The above analysis can be applied to the Slave packet with the probability a slave packet is directly disrupted being

$$P\left(X_{S,D}(n, \Delta)\right) = I_{\Delta,SL} \frac{S_{\beta(n),L}}{512} + I_{\Delta,SU} \frac{S_{\beta(n),U}}{512} \quad (5.8)$$

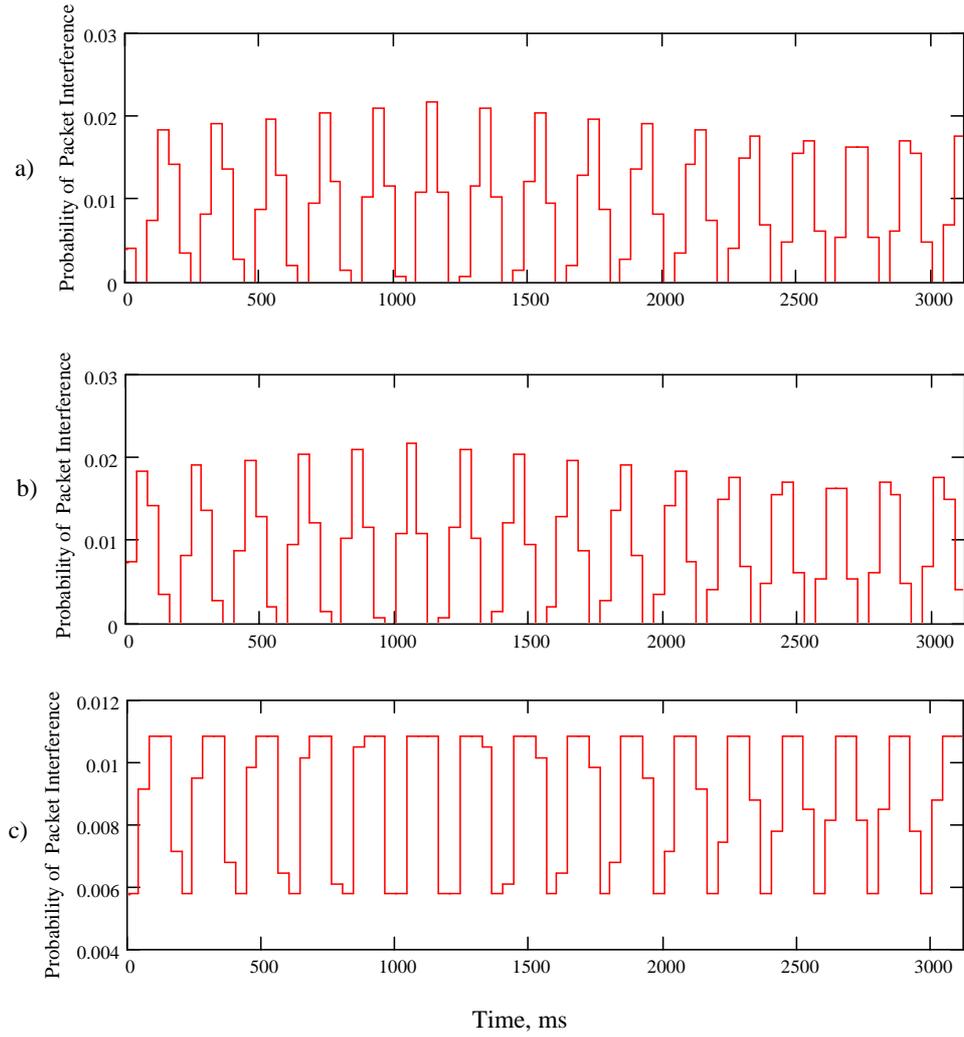


Figure 5.2: Probability of direct interference versus time of a) a Master packet b) a Slave packet c) Master or Slave packet where $n = \lfloor \frac{t}{40ms} \rfloor$.

when Δ is known and

$$P\left(E[X_{S,D}(n)]\right) = \frac{0.337}{512} S_{\beta(n)} \quad (5.9)$$

when Δ is unknown. One period of the probability of direct Slave packet disruption with an unknown Δ is shown in Figure 5.2b.

During the 40 *ms* a MSI/SSI is used, half of the packets transmitted are Master packets and half are Slave. Thus, the overall probability of a packet from a piconet being directly disrupted is

$$P\left(X_D(n)\right) = \frac{P\left(X_{M,D}(n)\right) + P\left(X_{S,D}(n)\right)}{2} \quad (5.10)$$

and is shown in Figure 5.2c.

5.1.2 Indirect Interference Probability

The expression for the probability of packets being lost due to *indirect* interference, \bar{D} , is similar. The vulnerabilities differ and indirect disruption of a packet is caused by direct disruption to the opposite type packet. That is, the probability of a Master packet being indirectly lost is dependent on the vulnerability of the Slave packet's AC/PH. If the Slave packet cannot be disrupted, no Master packets can be indirectly lost. Therefore, the probability that an inquiry packet will indirectly interfere with a Master packet using the same notation as before is

$$P\left(X_{M,\bar{D}}(n, \Delta)\right) = I_{\Delta,SHL} \frac{S_{\beta(n),L}}{512} + I_{\Delta,SHU} \frac{S_{\beta(n),U}}{512}. \quad (5.11)$$

Note that $S_{\beta(n),L}$ and $S_{\beta(n),U}$ are the same as in the direct interference case. Likewise, the analysis when Δ is unknown is similar to the previous cases. The expected vulnerability, $E[I_{\Delta,Intersection}]$, is 0.136 for $Intersection \in \{MHL, MHU, SHL, SHU\}$ when $x = 1$ and $y = 12$. Therefore, the probability a slave packet is indirectly disrupted due to header disruption is

$$P\left(X_{M,\bar{D}}(n)\right) = \frac{0.136}{512} S_{\beta(n)}. \quad (5.12)$$

5.1.3 Total Interference Probability

The probability that a Master packet is disrupted is the combination of the interference due to direct and indirect interference. If the packet is not transmitted due to indirect interference, it cannot be directly disrupted. Therefore, the total probability of the Master packet in the n th MSI/SSI being lost due to an inquiry packet is

$$P(X_M(n)) = P(X_{M,D}(n)) \left(1 - P(X_{M,\bar{D}}(n))\right) + P(X_{M,\bar{D}}(n)). \quad (5.13)$$

Similarly for a Slave packet,

$$P(X_S(n)) = P(X_{S,D}(n)) \left(1 - P(X_{S,\bar{D}}(n))\right) + P(X_{S,\bar{D}}(n)). \quad (5.14)$$

Again, since there is equal likelihood of a packet being Master or Slave, the probability of a piconet packet being disrupted by an inquiring device is

$$P(X(n)) = \frac{P(X_M(n)) + P(X_S(n))}{2}. \quad (5.15)$$

One period of interference probability, $P(X(n))$, is shown in Figure 5.3.

This result is compared to a simulation study using 1500 piconet/inquirer pairs with random clock, address, and alignment values. Each piconet generated packets for all slots (both master and slave) as the MSI cycles through all 79 possibilities. Collision with packets from the inquiring device were recorded. The analytical result fell within the 95% confidence interval for all values shown in Figure 5.3.

The interference for a specific piconet/inquiring node pair where Δ is known can be derived in a similar fashion using

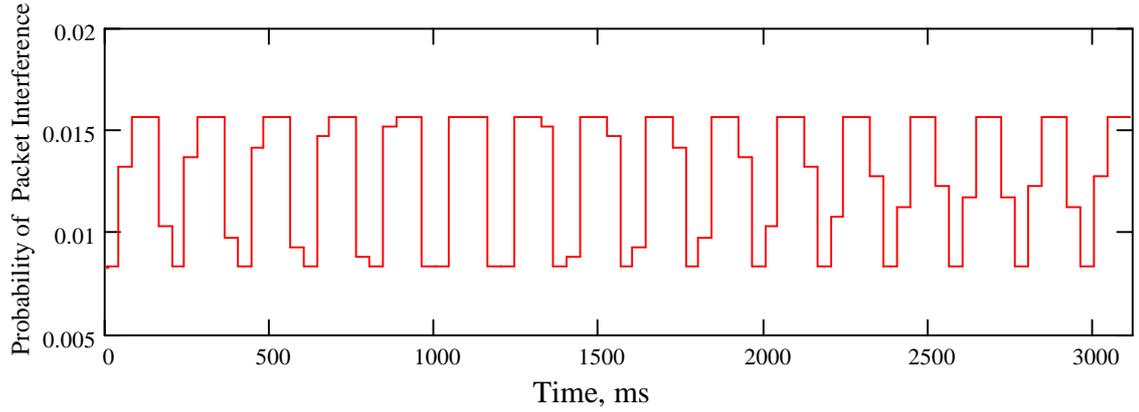


Figure 5.3: Pattern of direct or indirect interference probability against any packet.

$$\begin{aligned}
 P(X(n, \Delta)) &= \frac{1}{2} \left(P(X_{S,\bar{D}}(n, \Delta)) + P(X_{S,D}(n, \Delta)) \left(1 - P(X_{S,\bar{D}}(n, \Delta)) \right) \right) + \\
 &\quad \left(P(X_{M,\bar{D}}(n, \Delta)) + P(X_{M,D}(n, \Delta)) \left(1 - P(X_{M,\bar{D}}(n, \Delta)) \right) \right) \quad (5.16)
 \end{aligned}$$

instead of (5.15).

5.2 Multiple Inquiry Nodes

In a scatternet, there may be more than one node in the inquiry substate at a time, depending on the outreach method. In what follows, the probability of interference is extended to account for this.

5.2.1 Direct Interference Probability with m Inquiry Nodes

The probability a Master packet is disrupted in the n th MSI/SSI by any of m inquiry devices is

$$\begin{aligned}
P\left(X_{M,D}^m(n)\right) = & \\
& \left(P\left(L(M, n)\right) \sum_{i=1}^m P\left(V(M, L, m) = i\right) + P\left(U(M, n)\right) \sum_{i=1}^m P\left(V(M, U, m) = i\right) \right) \times \\
& \sum_{j=0}^i P\left(T(i) = j\right) P\left(X_{M,D}^m(n) | T(i) = j\right) \tag{5.17}
\end{aligned}$$

where

$L(M, n)$ is the event the Master packet is transmitted on a frequency in the Lower Range in the n^{th} MSI/SSI,

$U(M, n)$ is the event the Master packet is transmitted on a frequency in the Upper Range in the n^{th} MSI/SSI,

$V(L, m)$ is the random number of m inquiry nodes to which the Master packet is vulnerable in the Lower Range,

$V(U, m)$ is the random number of m inquiry nodes to which the Master packets are vulnerable in the Upper Range, and

$T(i)$ is the random number of i inquiry nodes using the train containing the frequency used by the piconet packet assuming the piconet uses a frequency in the Upper or Lower Range.

Since each frequency in an MSI is equally likely to be used at a given time,

$$P\left(L(M, n)\right) = \frac{M_{\beta(n),L}}{32} \tag{5.18}$$

and

$$P\left(U(M, n)\right) = \frac{M_{\beta(n),U}}{32}. \tag{5.19}$$

If Δ for each inquiring node is known, $P\left(V(M, L, m) = i\right)$ and $P\left(V(M, U, m) = i\right)$ can be computed directly. For example, when $m = 5$ and the inquiring nodes have Δ s of 200, 300, 600, 900, and 1200 μ s,

$$P(V(M, L, m) = i) = \begin{cases} 1 & i = 3 \\ 0 & i \in \{1, 2, 4, 5\} \end{cases} \quad (5.20)$$

and

$$P(V(M, U, m) = i) = \begin{cases} 1 & i = 2 \\ 0 & i \in \{1, 3, 4, 5\} \end{cases} \quad (5.21)$$

using Table 5.1. If the Δ s are unknown, $E[I_{\Delta, Intersection}]$ is used giving

$$P(V(M, L, m) = i) = P(V(M, U, m) = i) = \binom{m}{i} 0.337^i 0.337^{m-i} \quad (5.22)$$

for $x = 1$ and $y = 12$.

The probability that j of i inquiring nodes are using the train which contains the frequency used by the piconet's packet is

$$P(T(i) = j) = \binom{i}{j} 0.5^j 0.5^{i-j}. \quad (5.23)$$

Finally, the probability at least one of the inquiring nodes use the frequency of the piconet packet, assuming the piconet packet is vulnerable and in the same train and Range, is

$$P(X_{M,D}^m(n) | T(i) = j) = 1 - \left(\frac{7}{8}\right)^j. \quad (5.24)$$

5.2.2 Indirect Interference Probability with m Inquiry Nodes

The probability a Master packet is indirectly disrupted in the n th MSI/SSI is

$$\begin{aligned}
 P\left(X_{M,\bar{D}}^m(n)\right) = & \\
 & \left(P\left(M, LH(n)\right) \sum_{i=1}^m P\left(V(M, LH, m) = i\right) + \right. \\
 & \left. P\left(M, UH(n)\right) \sum_{i=1}^m P\left(V(M, UH, m) = i\right) \right) \times \\
 & \sum_{j=0}^i P\left(T(i) = j\right) P\left(X_{M,D}^m(n) | T(i) = j\right) \tag{5.25}
 \end{aligned}$$

where

$LH(M, n)$ is the event the preceding Slave packet was transmitted on a frequency in the Lower Range in the n^{th} MSI/SSI,

$UH(M, n)$ is the event the preceding Slave packet was transmitted on a frequency in the Upper Range in the n^{th} MSI/SSI,

$V(M, LH, m)$ is the random number of m inquiry nodes to which the preceding Slave packet was vulnerable in the Lower Range, and

$V(M, UH, m)$ is the random number of m inquiry nodes to which the preceding Slave packet was vulnerable in the Upper Range.

As before, indirect interference is a result of the Header of the preceding packet being disrupted, giving

$$P\left(LH(M, n)\right) = \frac{S_{\beta(n),L}}{32} \tag{5.26}$$

and

$$P\left(UH(M, n)\right) = \frac{S_{\beta(n),U}}{32}. \tag{5.27}$$

Likewise $P\left(V(MH, L, m) = i\right)$ and $P\left(V(MH, U, m) = i\right)$ can be computed directly using Table 5.2 or the Expected Vulnerability using the data for the Slave packet.

5.2.3 Total Interference Probability with m Inquiry Nodes

As in (5.13), the probability that one or more of m inquiring devices will disrupt a Master packet is

$$P(X_M^m(n)) = P(X_{M,D}^m(n))(1 - P(X_{M,\bar{D}}^m(n))) + P(X_{M,\bar{D}}^m(n)). \quad (5.28)$$

Applying the same approach used to derive $P(X_S^m(n))$, the overall probability of interference on a packet is

$$P(X^m(n)) = \frac{P(X_M^m(n)) + P(X_S^m(n))}{2}. \quad (5.29)$$

The resulting probability of interference for $m = 5$ and $m = \infty$ are shown in Figures 5.4a and 5.4b, respectively. Note that the probability of interference has a bipolar distribution. For example, when $m = 5$ the probability of interference is above 0.073 44% of the time even though the mean probability of interference is only 0.06.

A simulation study was run with 2, 3, 4 and 5 inquiring devices for verification and again, the derived probability of interference fell within the 95% confidence interval of the simulated result in all cases.

5.3 Expected Interference

When the cyclic probability pattern is averaged over the possible MSI's (i.e., $0 \geq n \leq 78$), the expected interference for a single inquiring node is 0.013. However, this quickly rises to significant levels as the number of inquiring nodes increases as shown in Figure 5.5. Including both direct and indirect interference, it approaches 0.73 since packets transmitted in the Upper or Lower Range are always disrupted and indirectly cause the next packet to be lost. For example, every Master packet transmitted in the Upper/Lower Range has its Header disrupted which indirectly disrupts the subsequent Slave packet, regardless of the Slave frequency's relation to

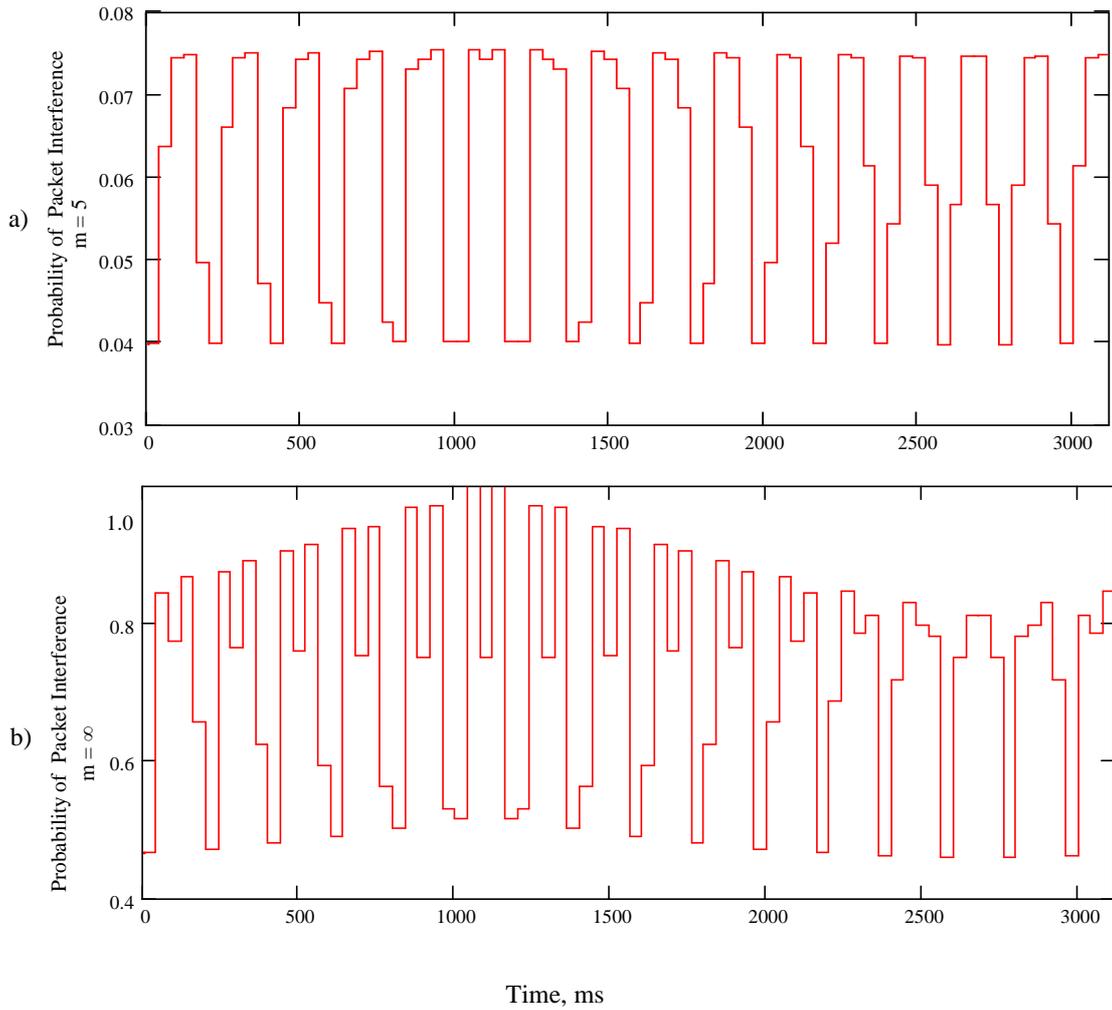


Figure 5.4: Pattern of interference probability with a) five inquiring nodes b) an infinite number of interfering nodes.

the Upper/Lower Ranges. If only direct interference is considered, the probability approaches 0.41. The data for the expected interference rate for $1 \leq m \leq 10$ are shown in Table 5.3.

5.4 Summary

Although often considered insignificant, we have demonstrated that, in fact, the probability of packet disruption from inquiring nodes is significant. All nodes

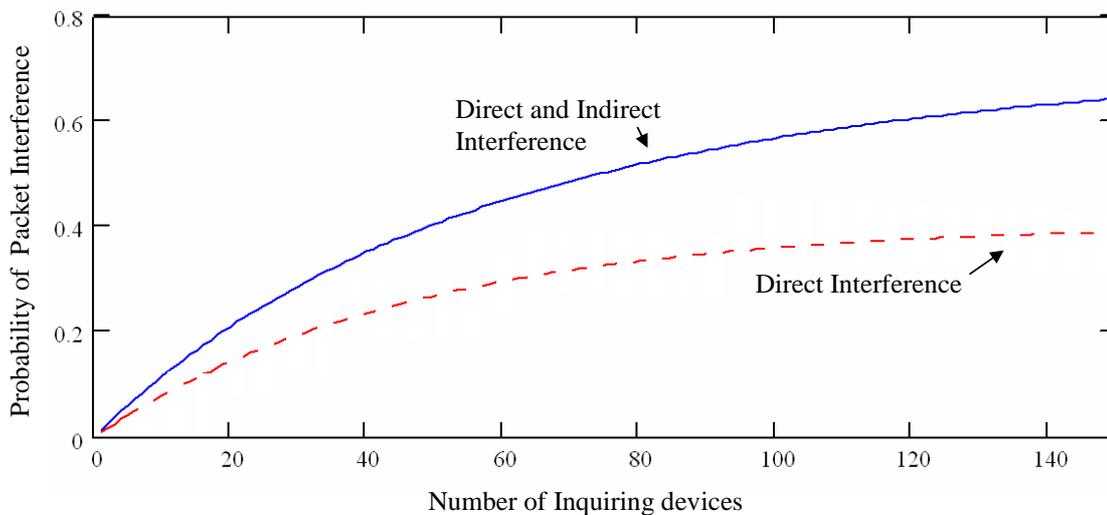


Figure 5.5: Probability of interference: multiple inquiring nodes with unknown Δ .

within range will be affected by the inquiring node, with the pattern of interference repeating every 3.16 seconds. With five inquiry nodes, the probability of packet disruption a neighboring piconets suffers exceeds 7.3% approximately half of the time. Thus, the maintenance and growth of scatternets in a Bluetooth network must be well regulated to avoid such interference.

In previously proposed models where nodes continually alternate between the inquiry and inquiry scan substates, it is clear the probability of packet disruption from inquiring nodes is significant. To properly address a resolution to the interference, the interference rate between piconets in the connection state is characterized in Chapter VI.

Table 5.3: Probability of interference for unknown Δ .

Inquiring nodes, m	Direct and Indirect Effects	Direct Effects Only
1	0.013	0.009
2	0.025	0.017
3	0.037	0.026
4	0.049	0.034
5	0.060	0.042
6	0.072	0.050
7	0.083	0.058
8	0.094	0.065
9	0.105	0.072
10	0.116	0.080

VI. Packet Error Rate (PER)

As BT devices become more common, the probability of piconets sharing the transmission space increases. Since piconets typically share the same spectrum, packets from piconets may share the same channel, or collide, and prevent successful packet reception. Similarly, piconets sometimes form scatternets where the piconets are within transmission range by design. Thus, the PER due to collisions is characterized to determine the necessity of collision avoidance techniques in scatternet organization and design. Additionally, adaptive frequency hopping was added in the BT standard v1.2 which allows devices to exclude certain frequencies. Establishing the PER between piconets enables educated decisions for piconets placement in the BT spectrum.

6.1 Packet Error Rate Characterization

A collision occurs when two piconets transmit packets on the same frequency with temporal overlap sufficient to cause the packet to be disrupted, or received with uncorrectable errors. It has been shown the probability of packet collision can range from 0 to 1/32 for a pair of neighboring piconets assuming the frequency selection within the respective MSI/SSI is uniformly distributed and the MSI/SSIs in the two piconets shift simultaneously [PBK03]. However, frequency selection is not uniform within the MSI/SSI and the probability of packet collision can be even greater. The PER depends on the relationship between MSI/SSIs, the time at which MSI/SSIs change, the frequency selection pattern within the MSI/SSI, and the temporal alignment of packets. To determine the actual probability of packet collisions, these relationships are defined.

6.1.1 MSI/SSI Relationship

The relationship between the MSI/SSIs of piconets is a significant factor in the probability of error [PBK03]. For example, if the MSI/SSIs have complete spectral overlap as shown in Figure 6.1, the probability of master packets using the same channel is unconditionally $1/32$. If the master packets are temporally aligned (cf. Section 6.1.4) and the piconet masters share a channel, a collision will result. Since the slave packets will also overlap temporally and the SSIs are also spectrally aligned as in Figure 6.1, the expected slave PER is also $1/32$. Now consider a different scenario where the MSI/SSIs of two piconets are spectrally aligned as in Figure 6.1 but have a temporal alignment such that the master packet of one piconet only overlaps the slave packet of the other (cf. Section 6.1.4). In this case, the PER for master and slave is zero since the master and slave devices alternate between sending packets and thus will never share a transmission frequency with the other piconet. That is, the MSI and SSI are mutually exclusive and the master packet transmits on a frequency from the MSI while the slave transmits on a frequency from the SSI.

The spectral relationship between the MSI/SSIs of Piconets i and j , $N_{i,j}$, is the number of frequency slots j 's MSI is offset to the right from i 's MSI during the transmission of j 's first master packet during i 's MSI cycle. Figure 6.2 shows the spectral offset of Piconets 1 and 2, $N_{1,2}$. Piconet 1 begins a new MSI cycle at t_1 as its MSI/SSI shifts from beginning at frequency 63 to begin at frequency 0 in MSTs 0 of the new cycle. Piconet 2's MSI begins with frequency 19 at t_2 , thus $N_{1,2} = 19$ since $\text{mod}(19 - 0, 79) = 19$. Note that $N_{1,2}$ is based on the spectral beginning of Piconet 2's MSI used by the first master packet after t_1 (i.e., frequency 19) rather than the beginning of Piconet 2's MSI at t_1 (i.e., frequency 3). In Figure 6.3 however, $N_{1,3} = 3$ since Piconet 3's MSI begins at frequency 3 for the master packet sent in the MSTs following t_1 (MSTs 31). Since $N_{i,j}$ is determined during the first MSTs in a piconet's MSI cycle, both $N_{2,1}$ and $N_{3,1}$ equal 60. In Figure 6.2, Piconet 1's MSI begins at frequency 0 at t_3 in the first MSTs of Piconet 2's MSI cycle that begins at

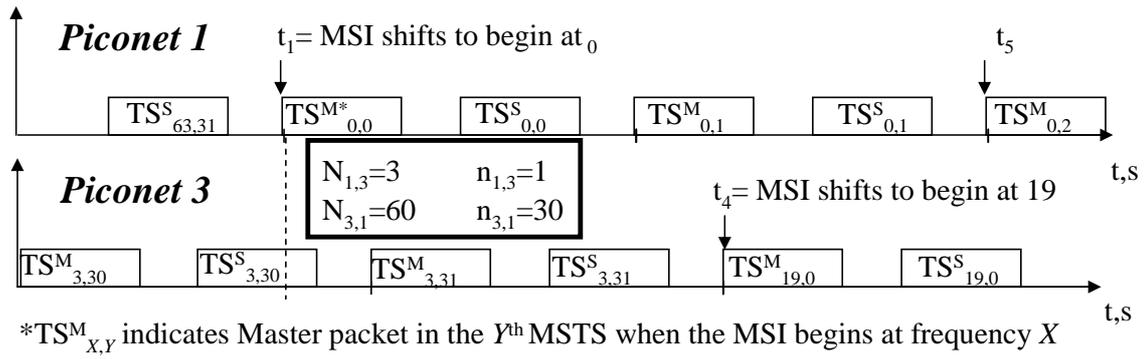
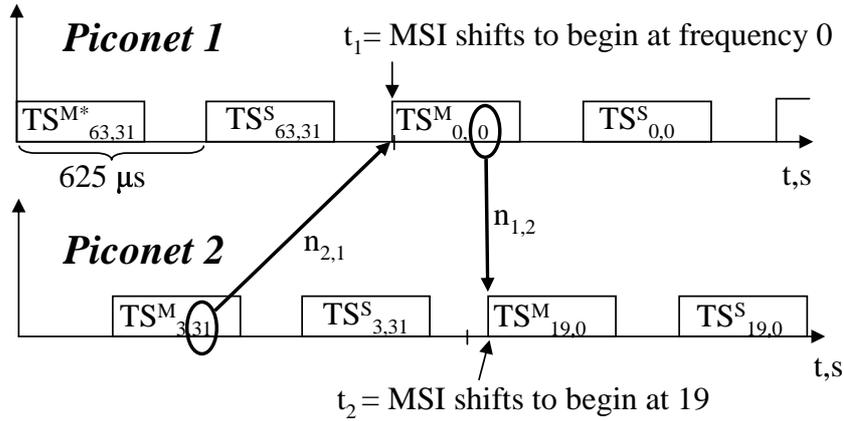


Figure 6.3: Determining the MSI/SSI relationship: $N_{i,j}$ and $n_{i,j}$.

and yet remain aligned for 30 MSTSs before one of the MSIs shift, the PER will be different than when one of the MSIs shift after only being spectrally aligned for one MSTS. This relationship is denoted $n_{i,j}$, the number of MSTSs that occur in Piconet i before Piconet j begins a new MSI cycle. In Figure 6.4, $n_{1,2} = 0$ since t_2 (the beginning of a new MSI cycle for Piconet 2) lies in the first MSTS after Piconet 1 begins a MSI cycle at t_1 . Likewise, Piconet 1 begins a new MSI cycle in the MSTS just before t_2 , which is 31 MSTSs after the previous cycle began, so $n_{2,1} = 31$. In Figure 6.3, t_4 (the beginning of a new MSI cycle for Piconet 3) lies in the second MSTS of Piconet 1's MSI cycle that begins at t_1 , giving $n_{1,3} = 1$. On the other hand, Piconet 1 doesn't begin a new cycle until after 30 of Piconet 3's MSTSs in a MSI cycle, therefore $n_{3,1} = 30$.

6.1.3 Placement within the MSI/SSI

A third factor in determining the PER between piconets is the pseudo-random choice of frequencies within the MSI and SSI. Recall that each of the 32 frequencies in the MSI/SSI is selected exactly once during an MSI cycle. By assuming the frequencies within the MSI or SSI are uniformly and independently chosen, the probability of any of the 32 frequencies being chosen is $1/32$. However, the *placements*, or spectral offsets, within the MSI and SSI are neither uniformly distributed nor independent. The Frequency Hop Selection Kernel (FHSK) consists of three stages pertinent to

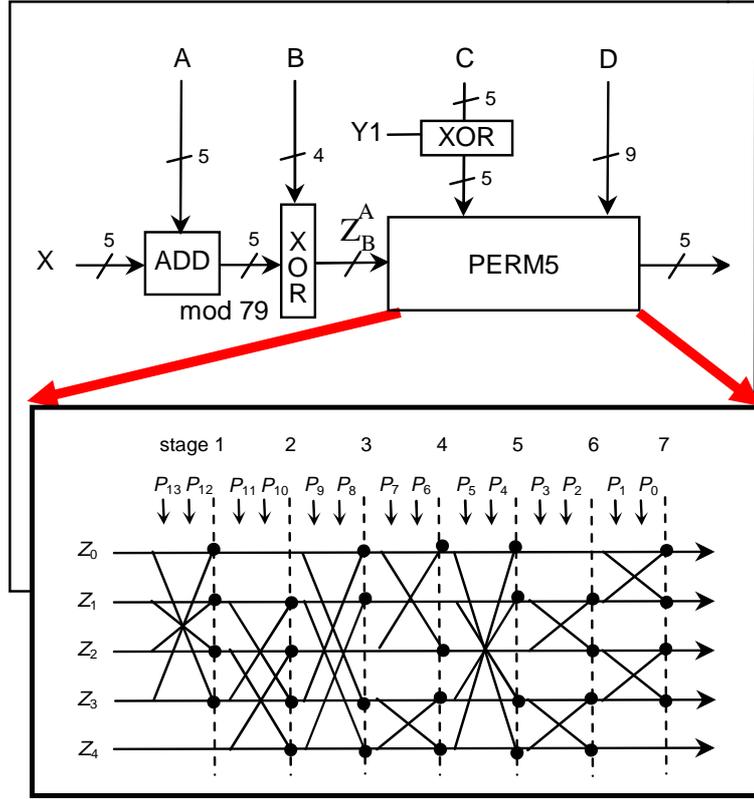


* $TS^M_{X,Y}$ indicates Master packet in the Y^{th} MSTS when the MSI begins at frequency X

Figure 6.4: Determining the MSI/SSI temporal relationship $n_{i,j}$.

this analysis and are shown in Figure 6.5. The only inputs which changes during the 32 MSTSs in a MSI cycle is the 5-bit X value which counts from 0 to 31, changing every MSTS (i.e., $1250 \mu\text{s}$), and $Y1/Y2$ which change the placement for the slave packet and shift it into the SSI. A phase shift of A is introduced to the X sequence in the first block. This phase shift increases by one every 11 seconds. The second block reorders the X sequence to one of 16 possible sequences. The new pattern, which we call Z , is created by a bitwise XOR between the four lower bits of X (i.e., $X_0 - X_3$) and the corresponding address bits in B (i.e., $A_{19}-A_{22}$). Thus, since the address used in the FHSK remains constant, the Z pattern remains constant for each piconet. Some of the 16 possible Z patterns remain very similar to X , including X itself when $B = 0$. Note that since only the lower four bits are inverted, the first 16 values in sequence Z are always between 0 and 15 while the final 16 are always greater than 15. With the one-to-one relation between sequences, the phase shift in X causes an identical phase shift in Z .

Finally, the Perm5 stage is a butterfly permutation that shuffles the order of the bits in Z without changing the number of '1's and '0's. The pattern of shuffling changes with every MSI cycle. The placement within the MSI is the result of the



$$\begin{aligned}
 X &= \text{CLK}_{6-2} & A &= A_{27-23} \oplus \text{CLK}_{25-21} \\
 Y1 &= \text{CLK}_1 & B &= A_{22-19} \\
 Y2 &= 32 \times \text{CLK}_1 & C &= A_{8,6,4,2,0} \oplus \text{CLK}_{20-16} \\
 & & D &= A_{18-10} \oplus \text{CLK}_{15-7}
 \end{aligned}$$

Figure 6.5: Portions of the Frequency Hop Selection Kernel [Blu03].

Perm5 operation. The difference between the placement within the MSI and SSI is due to inversion of the C input to the Perm5 operation during the odd (slave) time slots. Therefore, the $Z_B^A(X)$ input to the Perm5 operation and placements within the MSI and SSI always have the the same number of set, or ‘1’, bits.

For example, the placements with the MSI and SSI are contained in the set $\{1, 2, 4, 8, 16\}$ if the input into the Perm5 operation, $Z_B^A(X) = 1$. Likewise, if $Z_B^A(X) = 0$, both the MSI and SSI placement must be zero. When $X = 0$, $A = 7$, and $B = 9$, the first input into the Perm5 operator, $Z_9^7(0)$ is 14 ($00111_b \oplus 01001_b = 01110_b$). Thus, the placement within the MSI for the first packet has three set bits and the

possible placement selections from the set $\{7, 11, 13, 14, 19, 21, 22, 25, 26, 28\}$ are all equally likely. The placement within the SSI will also have three set bits in this case. However, due to a peculiarity in the Perm5 operator, the placement within the SSI cannot be the same as the MSI placement if the placement has two or three set bits.

Since the first 16 values in Z are less than 16 and are less likely to have many set bits in that portion of the sequence, the number of set bits and, therefore, the placements are not uniformly distributed. Thus, Z_B^A for the two piconets, denoted $Z_{B_i}^{A_i}$ and $Z_{B_j}^{A_j}$, are significant when determining the probability that two piconets share a frequency. Consider the case where $N_{12} = 47$, $n_{1,2} = 1$, $A_1 = 7$, and $A_2 = 12$. For simplicity, we assume Piconet 1's MSI begins at frequency zero, placing Piconet 2's MSI at frequency 47 for the first MSTs. Also, we assume $B_1 = B_2 = 0$, which sets Z to equal X in both piconets. The frequencies that can be used by each piconet for the first four MSTs's in Piconet 1's MSI cycle are shown in Figure 6.6. Note that, Z does not correspond directly to X since each piconet has a different phase shift, A .

In the first MSTs in its MSI cycle, Piconet 1's MSI includes frequencies 0 to 31 while Piconet 2's MSI include frequencies 47 to 78. Since the MSI's have no overlap, it is impossible to share the same frequency. In the second MSTs, however, Piconet 2's MSI shifts by 16 frequencies as $n_{1,2} = 1$ and X^2 (i.e., the X value in Piconet 2) scrolls back to zero. The possible placements within Piconet 1's MSI all have a single set bit since $Z_0^7(1) = 8$, which has a single bit. Due to its phase shift $Z_0^{12}(0) = 12$ and has two set bits. Note that Piconet 2 uses X^2 and Piconet 1 uses X^1 . Thus, four of Piconet 1's five possible MSI placements in that time slot (i.e., 1, 2, 4, and 8) are on one of the ten frequencies that may be used by the other piconet, (i.e., frequencies 1, 2, 4 or 8). The bold frequency blocks are those which may be shared by the two piconets. Each of Piconet 1's five placements have a 0.2 probability of being used. For four of the five, the probability that Piconet 2 uses

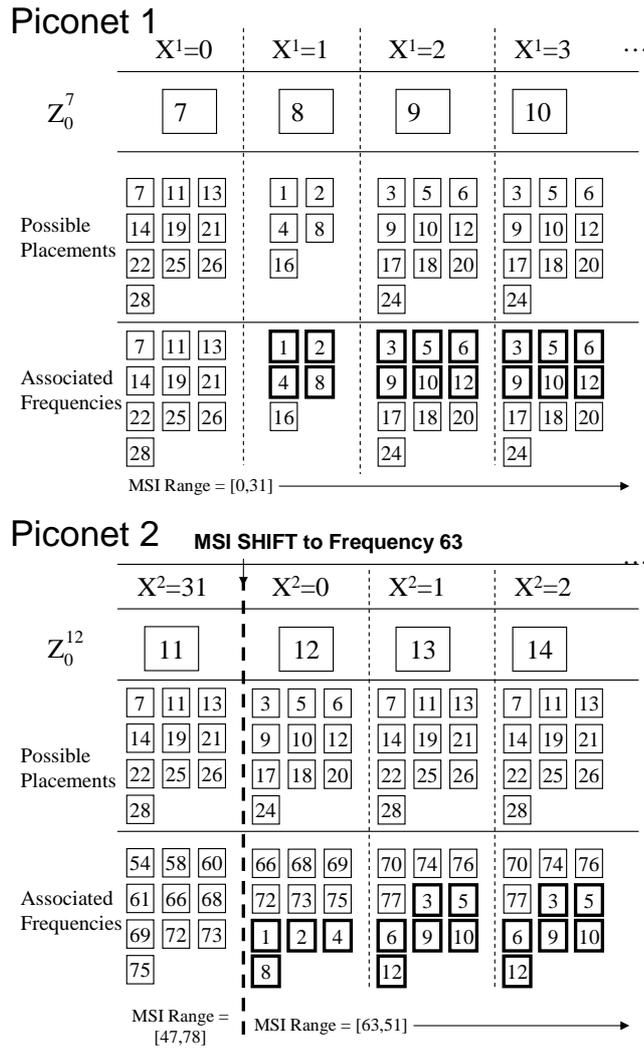


Figure 6.6: Impact of placement within the MSI on likelihood of sharing a channel when $N_{1,2} = 24$, $n_{1,2} = 1$, $A_1 = 7$, $A_2 = 12$, and $B_1 = B_2 = 0$.

the same frequency is 0.1. Thus, when $X^1 = 1$, the probability of the master packets sharing a frequency in the master slot is $4 \times 0.2 \times 0.1 = 0.08$. When $X^1 = 2$, six of the possible placements can be used by both piconets, making the probability of such an occurrence $6 \times 0.1 \times 0.1 = 0.06$. With this set of relational parameters (i.e., $N_{1,2} = 24$, $n_{1,2} = 1$, $A_1 = 7$, $A_2 = 12$, and $B_1 = B_2 = 0$), the probability that the piconets share frequencies is nonzero for 16 of the 32 X values in the MSI cycle. Since each X value is equally likely in a MSI cycle, the probability of master packets

sharing a frequency 0.044 when the piconets have the relationship specified. The expression for this probability can be found in Section 6.2.1.

The probability that the master packets share a frequency is sensitive to the relationship between the piconets. For example, if $n_{1,2}$, $A1$, or $B2$ were increased by one slot, or $A2$ were decreased by one, the probability of master packets sharing a frequency drops by 80% to 0.0088. If $N_{1,2}$ was shifted by one frequency, the probability drops to 0.016. Finally, if $B1$ were increased by one, the probability drops to 0.0069. Thus, all parameters (i.e., $n_{i,j}$, $N_{i,j}$, Ai , Bi , Aj , and Bj) are essential to capturing the large variation between piconets with subtle relational differences. Although the relationship between slave packets is similar, their collision properties also depend on the interaction between master packet frequencies.

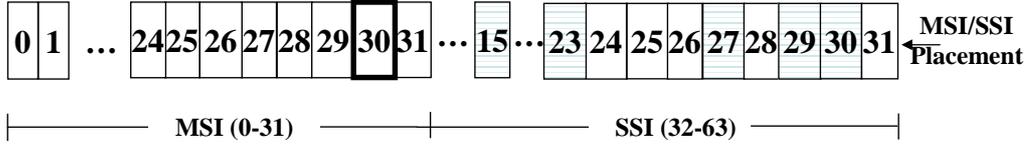
6.1.3.1 Master-Slave Frequency Dependence

The dependence between the placement of the master and slave frequencies links the probability of both packets being disrupted by another piconet. When MSI/SSIs are spectrally aligned as in Figure 6.1 (i.e., $N_{1,2} = 0$) and the first frequency in the MSI is used (i.e., the placement is zero), master and slave packets are both disrupted if the master packet is disrupted. Since the first frequency in the MSI is used, the placement within the SSI is also zero. Since the master packet is disrupted only when the placement within the MSI of both piconets is zero, the placement in both piconets' SSI must also be zero. That is, since the MSI/SSIs are spectrally aligned, neither the master nor the slave packets can collide if one piconet uses a MSI placement of 0 and the other does not. Similarly, if the master packet is disrupted with a MSI placement of 1, the probability of the slave packet also being disrupted is 0.2. Since the placement within the MSI is 1 in both MSIs, the placement in the SSIs is chosen from the set $\{1, 2, 4, 8, 16\}$ for each piconet. Each of these placements are equally likely in each piconet. Therefore, the probability of a collision is $5 \times (0.2)^2 = 0.2$.

The more complex case of $N_{1,2} = 24$ is shown in Figure 6.7. Determining the probability of slave packets sharing a frequency is more difficult when Piconet 2's MSI is located 24 frequencies to the right of Piconet 1's MSI. Piconet 1 using frequency 30 corresponds to the placement of 30 within the MSI. Since 30 (11110_b) is a placement with four set bits, the possible SSI placements for Piconet 1 include all five-bit numbers with four set bits, or $\{15, 23, 27, 29, 30\}$. Thus, the probability of Piconet 1 transmitting the slave packet on frequencies 47, 55, 59, 61, or 62 is 0.2. On the other hand, Piconet 2 has a MSI placement of 6 (00110_b), which has two set bits. Therefore, the possible SSI placements are $\{3, 5, 9, 10, 12, 17, 18, 20, 24\}$, making the probability of Piconet 2's slave packet being transmitted on frequency 59, 61, 65, 66, 68, 73, 74, 76 or 1 each 0.111. Note the placement within the SSI cannot be 6 since the placement within the MSI and SSI cannot be the same when the placement contains two or three set bits. A placement of 3 maps to frequency 59 since the SSI is 32 frequencies to the right of the MSI and $N_{1,2} = 24$. Thus, the probability of both piconets transmitting on frequency 59 or 61 is $2 \times 0.111 \times 0.2 = 0.0444$. If the placement within the SSI were independent, the probability of using the same frequency in the spectrally overlapped range of 56-63 would only be 0.04.

This difference is significant due to the polling nature of the Bluetooth protocol. If the payload of the master packet in a piconet is disrupted, the slave still receives the packet and responds with a packet and a NACK indicator to inform the master that the payload needs to be re-sent. However, when the header of the master packet is disrupted, the intended slave recipient will not recognize it has been addressed and will not respond with a packet [Blu03]. Therefore, when a master packet header is vulnerable to a master packet from another piconet, the probability that the slave packet is disrupted by the subsequent slave packet is dependent on whether the master packet's header is disrupted.

Piconet 1



Frequency in Bluetooth Spectrum

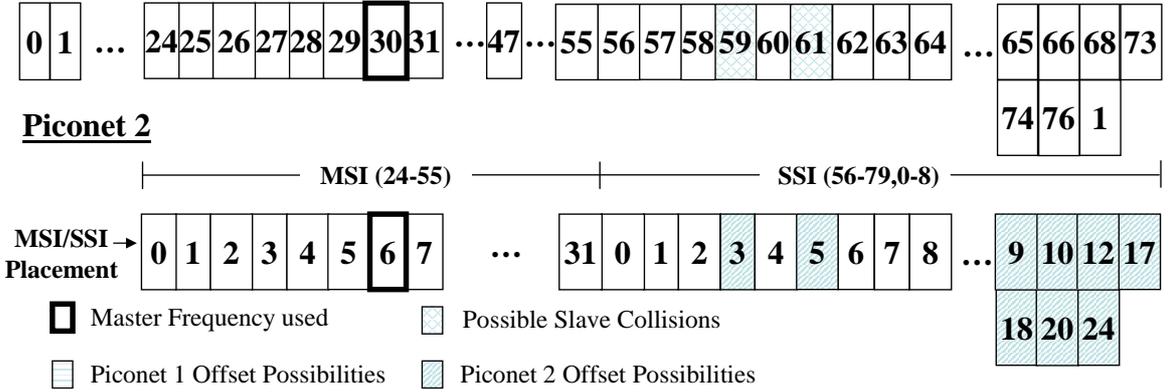


Figure 6.7: Effect of the master-slave frequency selection dependence on packet disruption.

6.1.4 Temporal Packet Relationship

The final factor which determines PER between a pair of piconets is temporal alignment between piconet packets. This relationship is defined as the difference between the beginning of the master packet slot in Piconet i and the beginning of the next master packet slot in Piconet j and is again denoted by $\Delta_{i,j}$. The value of $\Delta_{i,j}$ can range from 0 to 1250 μs and is assumed to remain constant due to limited drift in piconet clocks. Two piconets can have their time slots aligned such that a packet from one piconet may be vulnerable to, or may collide with, both a master and slave packet from the other piconet. The probability of this occurring for single-slot packets is

$$P(H) = 2r - 1 \quad (6.1)$$

where r is the ratio of packet and slot duration [ElH01].

Assuming each single-slot packet contains a full payload so that $r = 366/625$, there are eight significant intervals of $\Delta_{i,j}$ which must be characterized for single-slot PER analysis. Table 6.1 shows the intervals that $\Delta_{i,j}$ can fall within and the packet vulnerability for each interval. In the table, x is the number of bits that a packet must overlap the payload of another packet for an error to occur. The value for x varies according to the type of packet and the Forward Error Correction (FEC) used. We assume the worst case and use $x = 1$. Since the BT standard specifies that the AC and PH incorporate a error correction capability, a different number of bits must overlap for the header (consisting of the AC and PH) to be disrupted. Let y be the number of bits required to cause an error in the header. Note the temporal alignment may be such that the beginning of a packet from one piconet temporally overlaps the end of the payload of a packet from another piconet. If this occurs, the errors in the header may be correctable while the payload of the other packet is disrupted and uncorrectable. Since the PH uses 1/3 FEC, repeating the 8-bit PH word 3 times, we assume $y = 8$.

In Figure 6.8, $\Delta_{1,2} = 240 \mu s$ since Piconet 2 begins its master packet 240 μs after Piconet 1 begins its master packet, placing it in Interval 2 (T_2). From Table 6.1, M_0 in the Master Payload (MP) column indicates that Piconet 1's master packet's payloads are vulnerable to interference from a Piconet 2 master packet begun in the same MSTS (M_0). Piconet 1's master packet payload will be disrupted if M_0 is transmitted on the same frequency. Likewise, Piconet 1's Slave packet's Payloads (SP) are vulnerable to interference from slave packets from Piconet 2's that occur in the same MSTS (S_0) in which Piconet 1's master packet begins. Conversely, $\Delta_{2,1} = 1010 \mu s$ since Piconet 1 begins its master slot 1010 μs after Piconet 2. As a result, it is in T_8 . From Table 6.1, M_{-1} in the Master Header (MH) and MP columns indicates that the master packet transmitted by Piconet 1 in Piconet 2's *previous* MSTS can disrupt Piconet 2's master packet's header and payload. Likewise, Piconet

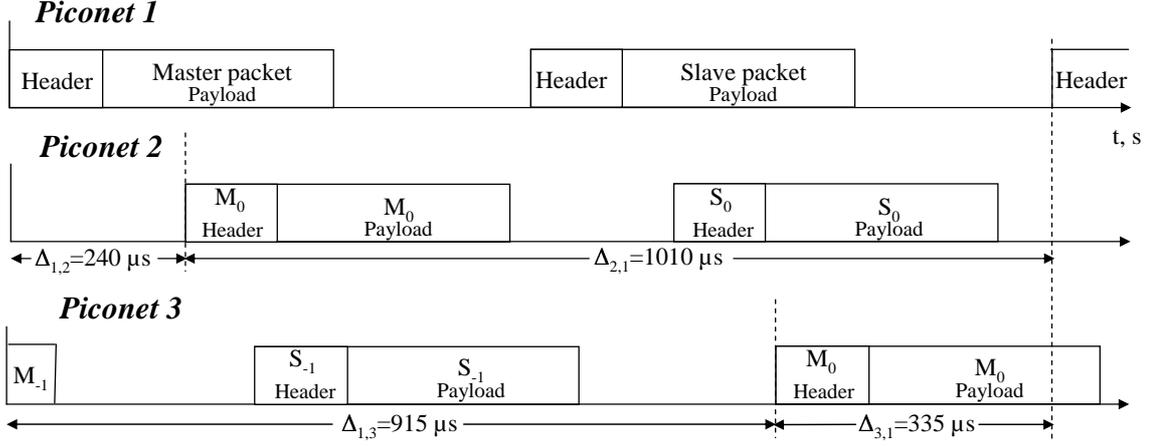


Figure 6.8: Determining $\Delta_{i,j}$, the packets' temporal relationship.

$\Delta_{3,1} = 335 \mu s$, placing it in T_3 . From Table 6.1, the Piconet 3's MH is vulnerable to S_{-1} , its MP and SH are vulnerable to M_0 , and its SP is vulnerable to S_0 .

6.1.4.1 Interference Between Piconets with Multi-slot Packets

Even though the PER analysis is restricted to single slot packets, (6.1) is generalized to multi-slot packet cases and different payload sizes between Piconets i and j for future analysis. The expression for packet length is defined as the sum of the number of time slots completely filled by the packet and the portion of the packet filled in the last time slot. For example, if Piconet i fills 3.9 times slots during a five-slot packet, the number of slots completely filled, $L_{i,F}$, is 3 and the portion of the last time slot filled, $L_{i,P}$, is 0.9. Define $L_{i,S}$ as the number of slots reserved for a packet. For a 5-slot packet, $L_{i,S} = 5$. Letting H be the number of packets from Piconet j that can overlap Piconet i 's packet, and define H_o as the minimum number of packets that can overlap a packet from Piconet i then

$$H_o = \begin{cases} \left\lfloor \frac{L_{i,F} + L_{j,F}}{L_{j,S}} \right\rfloor & L_{i,S} > L_{j,S} \\ 0 & otherwise \end{cases} \quad (6.2)$$

for $\lfloor x \rfloor$ indicating the largest integer not exceeding x .

Letting $L_{SUM} = L_{i,F} + L_{i,P} + L_{j,F} + L_{j,P}$ and E_U be the event that $\{L_{SUM} > L_{j,S}\} \cap \{L_{i,P} + L_{j,P} \geq 1\}$, the probability a packet from Piconet i is in danger of disrupt by H of Piconet j 's packets is

$$P(H = H_o) = I_{\overline{E_U}} \left(1 + H_o - \frac{L_{SUM}}{I_S} \right), \quad (6.3)$$

$$P(H = H_o + 1) = I_{\overline{E_U}} \left(\frac{L_{SUM}}{I_S} - H_o \right) + I_{E_U} \left(2 + H_o - \frac{L_{SUM}}{I_S} \right), \quad (6.4)$$

and

$$P(H = H_o + 2) = I_{E_U} \left(\frac{L_{SUM}}{I_S} - 1 - H_o \right). \quad (6.5)$$

The indicator function, I_{E_U} , equals 1 when E_U occurs and 0 otherwise. For example, when both piconets use single slot packets and using the notation from [ElH01] where r is the fraction of a slot used by the packet, $L_{i,S} = L_{j,S} = 1$, $L_{i,F} = L_{j,F} = 0$, $L_{i,P} = L_{j,P} = r$, and $H_o = 0$. Assuming $r = 366/625$, $I_{E_U} = 1$. This gives

$$P(H = 0) = 0, \quad (6.6)$$

$$P(H = 1) = 2(1 - r), \quad \text{and} \quad (6.7)$$

$$P(H = 2) = 2r - 1 \quad (6.8)$$

which matches the result from [ElH01]. In a more complex example, let Piconet j use three-slot packets with $L_{j,S} = 3$, $L_{j,F} = 2$, $L_{j,P} = 0.2$. Generally the higher layers of the protocol will fill all packet payloads when possible, but the general expression should not require this. Let Piconet i use a five slot packet with $L_{i,S} = 5$, $L_{i,F} = 3$, $L_{i,P} = 0.9$ which gives $L_{SUM} = 6.1$ and $H_o = 1$. This produces

$$P(H = 1) = 0, \quad (6.9)$$

$$P(H = 2) = \frac{2.9}{3}, \quad \text{and} \quad (6.10)$$

$$P(H = 3) = \frac{0.1}{3}. \quad (6.11)$$

The packet from Piconet i is overlapped by three of Piconet j 's packets in all temporal alignments between the two positions in Figures 6.9 b and c. Therefore, three packets only overlap in 0.1 time slots in the three time slots that the packet could be temporally positioned. The remainder of the possible temporal alignments only cause i 's packet to overlap two of j 's packets.

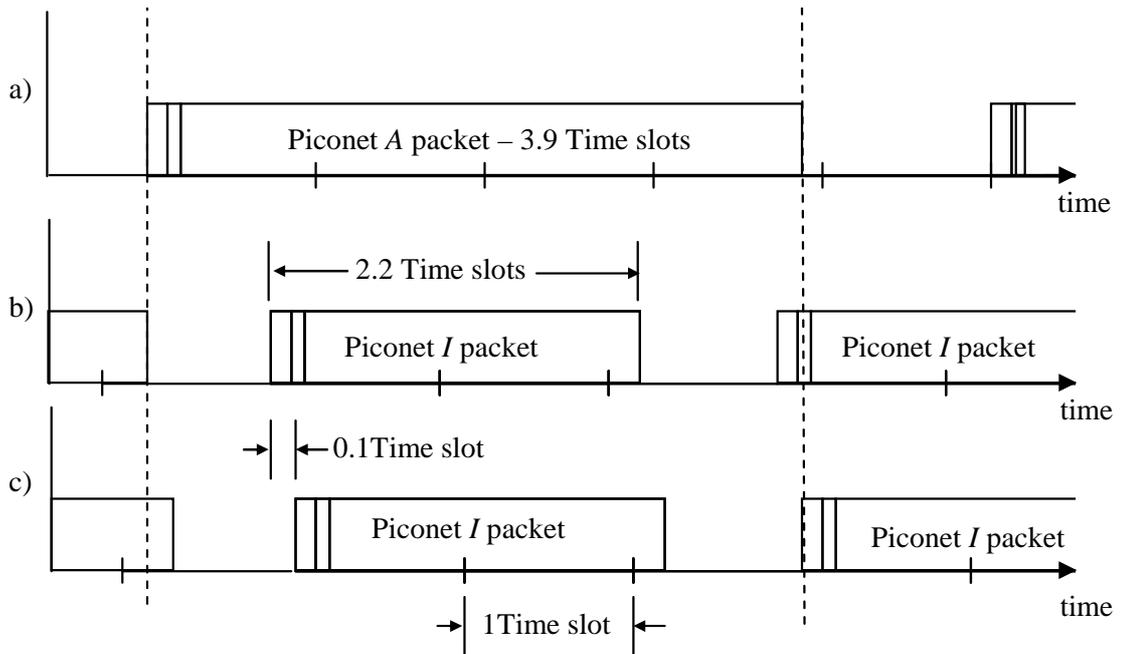


Figure 6.9: Overlap of packet with a) $L_{i,S} = 5$, b) $L_{j,S} = 3$ and overlaps i 's packet with two packets with just one bit at the end of the packet on the left, and c) with just one bit at the beginning of the packet on the right.

6.2 PER Distribution Derivation Necessities

Since the relational parameters $\Delta_{i,j}$, $n_{i,j}$, $N_{i,j}$, Ai , Bi , Aj , and Bj between a pair of piconets remains relatively constant, the PER can be explicitly determined for a piconet pair. Note that the phase shifts, Ai and Aj , advance by one every 11 seconds. Therefore, when analyzing the PER between a specific pair of piconets, the phase shifts should be considered. For this analysis, the phase shifts are assumed to remain constant. With a random pairing of piconets, the relational parameters, $\Delta_{i,j}$, $n_{i,j}$, $N_{i,j}$, Ai , Bi , Aj , and Bj are uniformly distributed on $[0, 1250 \mu\text{s}]$, $\{0, 31\}$, $\{0, 78\}$, $\{0, 31\}$, $\{0, 15\}$, $\{0, 31\}$, and $\{0, 15\}$ respectively. By assuming $\Delta_{i,j}$ has μs granularity, the number of possible PERs is finite since $n_{i,j}$ and $N_{1,2}$ are integers. Thus, a probability mass function (pmf) for the PER can be derived for an arbitrary pair of piconets using single slot packets. Let Φ be the random variable representing the PER for a random piconet pair. To derive the pmf for the PER, $f_{\Phi}(\phi)$, the probability that piconets share the same frequency is now determined.

6.2.1 Frequency Sharing Rate

Assuming the placements within an MSI are uniformly distributed and $n_{i,j} = 0$, the expression for the probability that a master packet shares a frequency with the packet of another piconet is

$$P(M|N^{\theta}) = \begin{cases} \frac{1}{32^2}(32 - N^{\theta}) & 0 \leq N^{\theta} < 32 \\ \frac{1}{32^2}(N^{\theta} - 47) & 47 < N^{\theta} \leq 78 \\ 0 & \textit{otherwise} \end{cases} \quad (6.12)$$

where $N^{\theta} = \text{mod}(N_{i,j} + 16\theta, 79)$, and θ represents the shift by 16 frequencies of the MSI/SSI interval the interfering packet is coming from. Recall that $N_{i,j}$ is the spectral alignment of the MSIs between two piconets. For example, if a master packet from Piconet i is vulnerable to the master packet from a Piconet j , the probability

that Piconet i 's master packet is disrupted is $P(M|N^0)$. If Piconet i 's master packet is instead vulnerable to the slave packet, the probability that Piconet i 's master packet is disrupted is $P(M|N^2)$ since the slave packet comes from the SSI that is 32 frequencies to the right (i.e., $\theta = 2$) of the MSI.

Since placements are not uniformly distributed, this expression is the expected value of the probability. The probability is dependent not only on $N_{i,j}$, but also Z_{Bi}^{Ai} , Z_{Bj}^{Aj} and $n_{i,j}$. For an accurate distribution of the PER, the expression is expanded to include $n_{i,j}$, Ai , Bi , Aj , and Bj . Due to the polling nature of the protocol, the probability must be partitioned to include the probability of the slave frequency being shared as well. Although it is possible to derive a complex expression for the probability, it is more expedient to determine the probabilities by enumerating the possible collisions. For simplicity of notation, α , will represent Ai , Bi , Aj , and Bj . Likewise, N and n represent $N_{1,2}$ and $n_{1,2}$, respectively. The sequences Z_{B1}^{A1} and Z_{B2}^{A2} are represented by $Z1$ and $Z2$, respectively. Defining the arrays

$$\begin{aligned}
S_0 &= [0] \\
S_1 &= [1, 2, 4, 8, 16] \\
S_2 &= [3, 5, 6, 9, 10, 12, 17, 18, 20, 24] \\
S_3 &= [7, 11, 13, 14, 19, 21, 22, 25, 26, 28] \\
S_4 &= [15, 23, 27, 29, 30] \\
S_5 &= [31] \\
L &= [1, 5, 10, 10, 5, 1] \\
L_S &= [1, 5, 9, 9, 5, 1]
\end{aligned}$$

and function

$$b_v = \text{number of set bits in scalar } v,$$

the probability a master packet shares its frequency with another piconet's master packet and the subsequent slave packet shares its frequency with the other piconet's slave packet is

$$\begin{aligned}
P(M \cap S | N^\theta, n, \alpha) = & \\
& \sum_{X=0}^{31} \sum_{i=0}^{(L(b_{Z1(X)})-1)} \sum_{j=0}^{(L(b_{Z2(X-n)})-1)} \frac{1}{32L(b_{Z1(X)})L(b_{Z2(X-n)})} \times \\
& \sum_{k=0}^{(L(b_{Z1(X)})-1)} \sum_{r=0}^{(L(b_{Z2(X-n)})-1)} \frac{1}{L_S(b_{Z1(X)})L_S(b_{Z2(X-n)})} \times \\
& I_{FM1=FM2} I_{(FM1 \neq FS1) \cup (L_S(b_{Z1(X)}) \neq 9)} \times \\
& I_{FS1=FS2} I_{(FM2 \neq FS2) \cup (L_S(b_{Z2(X-n)}) \neq 9)}
\end{aligned} \tag{6.13}$$

where

$$\begin{aligned}
FM1 &= S_{b_{Z1(X)}}(i), \\
FS1 &= S_{b_{Z1(X)}}(k), \\
FM2 &= \text{mod} \left(S_{b_{Z2(\text{mod}(X-n, 32))}}(j) + N^\theta + 16I_{X \geq n}, 79 \right), \\
FS2 &= \text{mod} \left(S_{b_{Z2(\text{mod}(X-n, 32))}}(r) + N^\theta + 16I_{X \geq n}, 79 \right),
\end{aligned}$$

and I_z is the indicator function such that $I_z = 1$ when z is true and $I_z = 0$ otherwise. To simplify notation, n is used in the expression where $\text{mod}(n-1, 32) + 1$ should be used. In other words, n should be set to 32 when $n = 0$. The summations include all possible placements between two piconets for each X input to the FHSK.

This complex expression is most easily explained by applying it to a pair of piconets such as those shown in Figure 6.6. Recall that, for that example, $N_{1,2} = 47$,

$n_{1,2} = 1$, $A1 = 7$, $A2 = 12$, and $B1 = B2 = 0$. Assume master packets are only vulnerable to master packets from the interfering piconet, so $\theta = 0$. When $X = 0$, $Z_{B1}^{A1}(X) = Z_0^7(0) = 7$. Since $b_7 = 3$ and $L(3) = 10$, the integer i counts from zero to nine and is used to count through the 10 possible placements within the first piconet's MSI when $X = 0$. Similarly, $Z_{B2}^{A2}(X - n) = Z_0^{12}(-1)$. Note that if $n > X$, $\text{mod}(X - n, 32)$ must be used, giving $Z_0^{12}(31) = 11$. Therefore, $b_{11} = 3$ and $L(3)=10$ and the integer j , which counts through the possible placements within the second piconet's MSI, also counts between zero and nine.

The counters for the placements within the first and second piconets' SSI (i.e., k and r) are identical to i and j . Even though there are only nine possible placements within each SSI since the placements have three set bits, all ten possible placements are checked and one which cannot be used (since it matches the MSI placement) is removed by the indicator functions.

The probability of each set of placement pairs being used is $1/(10 \times 10 \times 9 \times 9)$ since each of the ten possible MSI placements and nine possible SSI placements in each piconet are equally likely. This partial probability is added to $P(M \cap S | N_\theta, n, \alpha, X = 0)$ for each case the master packet frequencies and slave packet frequencies match in a possible combination. For example, when $i = j = k = r = 0$, placements within each of the intervals is seven. This equates to a master frequency of $FM1 = 7$ in the first piconet. The master frequency of the second piconet is $FM2 = 54$ since $N^0 = 47$ and $16I_{X \geq n} = 0$. Since both slave frequencies are shifted by 32, the shift is ignored. Therefore, the slave frequencies, $FS1$ and $FS2$, are also 7 and 54 respectively. In this case, $I_{(FM1=FM2)} = 0$, $I_{(FS1=FS2)} = 0$, $I_{(FM1 \neq FS1) \cup (L_S(b_{Z1(X)}) \neq 9)} = 0$, and $I_{(FM2 \neq FS2) \cup (L_S(b_{Z2(X)}) \neq 9)} = 0$, so it clearly does not add to $P(M \cap S | N_\theta, n, \alpha, X = 0)$. Note that even if $I_{FM1=FM2} = I_{FS1=FS2} = 1$, it is not a possible combination if $I_{(FM1 \neq FS1) \cup (L_S(b_{Z1(X)}) \neq 9)} = 0$ or $I_{(FM2 \neq FS2) \cup (L_S(b_{Z2(X)}) \neq 9)} = 0$ since both placements have three set bits and the $FM1$

cannot equal $FS1$. When $X = 0$, there are no cases where $FM1 = FM2$ in this piconet pair since the MSI's do not overlap. However, when $X = 1$, it is possible.

When $X = 1$, $Z_0^7(1) = 8$. Since $b_7 = 1$ and $L(3) = 5$, the integer i counts from zero to 4. Similarly, $Z_0^{12}(0) = 12$. Therefore, $b_{11} = 2$ and $L(2)=10$. Again, in most cases, $FM1 \neq FM2$. However, when $i = 0$ the placement with Piconet 1's MSI is 1 which places it at frequency 1. When $j = 6$, Piconet 2's MSI placement is 17, $N^\theta = 47$ and $16I_{X \geq n}=16$, placing it at frequency 1 since $mod(80, 79) = 1$. Thus, the master packets share the same frequency and $I_{(FM1=FM2)} = 0$. When $k = 0$ the Piconet 1's slave frequency is 1. In practice, this is shifted by 32 but again, the shift by 32 frequencies for the slave packets is ignored in this expression since the slave frequencies are shifted for both. When $r = 6$, Piconet 2's SSI placement is also 1. However, since Piconet 2's placements have two set bits, the placement cannot be the same in both intervals and $I_{(FM2 \neq FS2) \cup (L_S(b_{Z2(X)}) \neq 9)} = 0$. However, when $i = 1$, Piconet 1 uses frequency 2 for the slave. Likewise, when $j = 7$, the placement in Piconet 2's SSI is 18, placing at also at frequency 2. Therefore, all four indicator functions equal one and the probability that the master and slave frequencies share the same frequency contributes to the sum when $X = 4$, $i = 0$, $j = 6$, $k = 1$, and $r = 7$ and contribute to the overall probability. Including all cases that cause the master and slave frequencies to match, $P(M \cap S|N^\theta, n, \alpha) = 0.00453$. Note that this does not match the 0.044 probability present earlier for this example. This is explained by the fact that $P(M|N^\theta, n, \alpha)$ is the sum of $P(M \cap S|N^\theta, n, \alpha)$ and $P(M \cap \bar{S}|N^\theta, n, \alpha)$.

Equation (6.13) is modified to compute $P(M \cap \bar{S}|N^\theta, n, \alpha)$ by changing the equal sign in $I_{(FS1=FS2)}=0$ to $I_{(FS1=FS2)} \neq 0$. Then, $P(M \cap \bar{S}|N^\theta, n, \alpha) = 0.0392$, giving $P(M|N^\theta, n, \alpha) = 0.044$ as before. Likewise, $P(\bar{M} \cap S|N^\theta, n, \alpha)$ and $P(\bar{M} \cap \bar{S}|N^\theta, n, \alpha)$ are computed by changing $I_{(FM1=FM2)}$ to $I_{(FM1 \neq FM2)}$. All four probabilities are needed to derive the PER and goodput.

6.3 PER Derivation

The pmf for the PER of a random pairing of piconets is determined by weighting the possible PERs with their likelihood. Thus, an expression for the PER conditioned on the combination of $\Delta_{i,j}$, $n_{i,j}$, $N_{i,j}$, and α must be derived. The temporal relationship between packets is significantly different for the seven intervals, so the expression is developed piece-wise on $\Delta_{i,j}$. Throughout the derivation

$$P(M|N^\theta, n, \alpha) = P(M \cap S|N^\theta, n, \alpha) + P(M \cap \bar{S}|N^\theta, n, \alpha) \quad (6.14)$$

and

$$P(S|N^\theta, n, \alpha) = P(M \cap S|N^\theta, n, \alpha) + P(\bar{M} \cap S|N^\theta, n, \alpha). \quad (6.15)$$

6.3.1 PER in T_1

In the T_1 interval, the MH is vulnerable to M_0 and SH is vulnerable to S_0 . If the header of a packet is disrupted, it is irrelevant that the payload may also be disrupted since the intended receiver will not attempt to receive the payload without properly receiving the header. Denoting D_{MH} as the event that the MH is disrupted

$$P(D_{MH}|T_1, n, N, \alpha) = P(M|N^0, n, \alpha). \quad (6.16)$$

The probability that SH is disrupted, $P(D_{SH}|T_1, n, N, \alpha)$, must account for the dependence that the Master packet header be received correctly. If the MH is disrupted, the slave packet is not transmitted and cannot be disrupted. Thus, the probability of SH being disrupted is

$$P(D_{SH}|T_1, n, N, \alpha) = \frac{P(\bar{M} \cap S|N^0, n, \alpha)}{P(\bar{M}|N^0, n, \alpha)}. \quad (6.17)$$

Since each packet is only vulnerable to one packet from the interfering piconet, the probability that the master packet is disrupted, $P(D_M|T_1, n, N, \alpha)$, is

$$P(D_M|T_1, n, N, \alpha) = P(D_{MH}|T_1, n, N, \alpha). \quad (6.18)$$

Likewise, the probability the slave packet is disrupted is

$$P(D_S|T_1, n, N, \alpha) = P(D_{SH}|T_1, n, N, \alpha). \quad (6.19)$$

The PER is the probability that either a master or slave packet packet is disrupted.

Denoting the PER as $\phi(T_k, n, N, \alpha)$,

$$\begin{aligned} \phi(T_k, n, N, \alpha) = & \\ & \frac{P(D_M|T_k, n, N, \alpha) + P(\overline{D_{MH}}|T_k, n, N, \alpha)P(D_S|T_k, n, N, \alpha)}{1 + P(\overline{D_{MH}}|T_k, n, N, \alpha)} \end{aligned} \quad (6.20)$$

for $1 \leq k \leq 8$ assuming that a single-slot master packet is transmitted in every master slot. This ensures that a slave packet will be generated for each master packet whose header is not disrupted. Thus,

$$\phi(T_1, n, N, \alpha) = \frac{P(M|N^0, n, \alpha) + P(\overline{M} \cap S|N^0, n, \alpha)}{1 + P(\overline{M}|N^0, n, \alpha)}. \quad (6.21)$$

6.3.2 PER in T_2

In T_2 , the M_0 packet no longer affects the MH, but only the MP. Therefore,

$$P(D_{MH}|T_2, n, N, \alpha) = 0 \quad (6.22)$$

and

$$P(D_{MP}|T_2, n, N, \alpha) = P(M|N^0, n, \alpha). \quad (6.23)$$

Since MH is not vulnerable, the intended slave packet always recognizes it is the intended recipient and returns a packet. However, since the interfering piconet's master packet header is disrupted whenever the master packets collide as shown in Piconet 2 of Figure 6.8, the interfering piconet's (i.e., Piconet 2's) slave packet is not transmitted. Thus

$$P(D_{SH}|T_2, n, N, \alpha) = 0 \quad (6.24)$$

and

$$P(D_{SP}|T_2, n, N, \alpha) = P(\overline{M} \cap S|N^0, n, \alpha). \quad (6.25)$$

Since each packet is only vulnerable to one packet from the other piconet,

$$P(D_M|T_2, n, N, \alpha) = P(D_{MP}|T_2, n, N, \alpha) \quad (6.26)$$

and

$$P(D_S|T_2, n, N, \alpha) = P(D_{SP}|T_2, n, N, \alpha). \quad (6.27)$$

As with all of the intervals, T_1 through T_8 , (6.20) applies when determining the PER. Thus,

$$\phi(T_2, n, N, \alpha) = \frac{P(M|N^0, n, \alpha) + P(\overline{M} \cap S|N^0, n, \alpha)}{2}. \quad (6.28)$$

6.3.3 PER in T_3

Determining the PER in T_3 is more difficult since each packet is vulnerable to two packets from the other piconet. Packet M_0 may still disrupt MP, but S_{-1} may also disrupt MH. However, S_{-1} can only disrupt MH if S_{-1} is transmitted. This does not occur if the previous MP was disrupted, also disrupting the header of M_{-1} . If S_{-1} is transmitted, the probability of a collision is similar to that between the two master packets except that the SSI is shifted to the right by 32 frequencies in most cases. The exception occurs if $n = 0$, when the first S_{-1} is transmitted on a frequency from the SSI used before the interfering piconet shifts its MSI/SSI cycle. Therefore, the first S_{-1} is from a SSI only 16 frequencies to the right. Furthermore, since S_{-1} is from the previous MSTS of the interfering piconet, the effective n is shifted one MSTS to the right for all values of n . Thus,

$$P(D_{MH}|T_3, n, N, \alpha) = P(M|N^{1+I_n > 0}, n + 1, \alpha)P(\overline{D_{MP}}|T_3, n, N, \alpha) \quad (6.29)$$

where

$$P(D_{MP}|T_3, n, N, \alpha) = P(M|N^0, n, \alpha). \quad (6.30)$$

Similarly, S_0 may still disrupt SP, but M_0 may also disrupt SH. As in T_1 and T_2 , the packet can only be transmitted and disrupted if MH is not disrupted. Since SH can be disrupted by M_0 , the probability of disruption is similar to that of being disrupted by a slave packet except the interval is shifted 32 frequencies to the left. Therefore,

$$P(D_{SH}|T_3, n, N, \alpha) = P(S|N^{-2}, n, \alpha) \quad (6.31)$$

and

$$\begin{aligned}
P(D_{SP}|T_3, n, N, \alpha) = & \\
& \frac{P(\overline{M} \cap S|N^0, n, \alpha)P(\overline{D_{MH}}|T_3, n, N, \alpha)}{P(\overline{M}|N^0, n, \alpha)}. \tag{6.32}
\end{aligned}$$

The events that MH and MP are disrupted are generally mutually exclusive. Since the slave and master packets come from mutually exclusive intervals, both packets generally cannot use the same frequency. However, when the interfering packet's interval shift (at the beginning of its MSI cycle), it is possible that the master packet in the new MSI can select the same frequency as the last slave packet from the previous SSI since the new MSI and old SSI overlap by 16 frequencies. However, this is rare and the events are assumed to be mutually exclusive so

$$\begin{aligned}
P(D_M|T_3, n, N, \alpha) = & P(M|N^0, n, \alpha) + \\
& P(M|N^{1+I_n>0}, n+1, \alpha)P(\overline{M}|N^0, n, \alpha). \tag{6.33}
\end{aligned}$$

Likewise, the event that SH and SP are disrupted is mutually exclusive without exception, giving

$$\begin{aligned}
P(D_S|T_3, n, N, \alpha) = & P(S|N^{-2}, n, \alpha) + \\
& \frac{P(\overline{M} \cap S|N^0, n, \alpha)P(\overline{D_{MH}}|T_3, n, N, \alpha)}{P(\overline{M}|N^0, n, \alpha)}. \tag{6.34}
\end{aligned}$$

Again, (6.20) applies in determining $\phi(T_3, n, N, \alpha)$.

6.3.4 PER in T_4

In T_4 , MH is only vulnerable to S_{-1} and SH is only vulnerable to M_0 . Since the interfering piconet's master packet headers are not vulnerable, S_{-1} is always transmitted. Therefore,

$$P(D_{MH}|T_4, n, N, \alpha) = P(M|N^{1+I_n>0}, n+1, \alpha), \quad (6.35)$$

$$P(D_{SH}|T_4, n, N, \alpha) = P(S|N^{-2}, n, \alpha), \quad (6.36)$$

and

$$P(D_{MP}|T_4, n, N, \alpha) = P(D_{SP}|T_4, n, N, \alpha) = 0. \quad (6.37)$$

Since each packet is only vulnerable to a single packet from the interfering piconet,

$$P(D_M|T_4, n, N, \alpha) = P(D_{MH}|T_4, n, N, \alpha), \quad (6.38)$$

and

$$P(D_S|T_4, n, N, \alpha) = P(D_{SH}|T_4, n, N, \alpha). \quad (6.39)$$

As before, (6.20) applies in determining $\phi(T_4, n, N, \alpha)$.

6.3.5 PER in T_5

The most complex PER case lies in interval T_5 . As in T_4 , MH is only vulnerable to S_{-1} and SH is only vulnerable to M_0 . However, the headers of the interfering packets are also disrupted when a collision occurs. Therefore, when SH is disrupted, it is not possible for the next MH to be disrupted since the interfering piconet's

slave packet is not transmitted. Likewise, when MH is disrupted, the subsequent slave packet is not transmitted, guaranteeing that interfering piconet's master packet header is not disrupted. After a short derivation,

$$P(D_{MH}|T_5, n, N, \alpha) = \frac{P(M|N^{1+I_{n>0}}, n+1, \alpha)P(\bar{S}|N^{-2}, n, \alpha)}{1 - P(M|N^{1+I_{n>0}}, n+1, \alpha)P(S|N^{-2}, n, \alpha)}. \quad (6.40)$$

Since the interfering piconet's master packet is always transmitted and there is no strong dependent relationship between the slave packet frequency from one MSTS and the master packet frequency from the next

$$P(D_{SH}|T_5, n, N, \alpha) = P(S|N^{-2}, n, \alpha). \quad (6.41)$$

As in T_4 ,

$$P(D_M|T_5, n, N, \alpha) = P(D_{MH}|T_5, n, N, \alpha), \quad (6.42)$$

$$P(D_S|T_5, n, N, \alpha) = P(D_{SH}|T_5, n, N, \alpha), \quad (6.43)$$

and (6.20) applies in determining $\phi(T_5, n, N, \alpha)$.

6.3.6 PER in T_6

In the interval T_6 , the same packets are affected as in T_5 . However, only the interfering packet headers are affected. Thus,

$$P(D_{MH}|T_6, n, N, \alpha) = P(D_{SH}|T_6, n, N, \alpha) = 0 \quad (6.44)$$

and

$$P(D_{MP}|T_6, n, N, \alpha) = P(M|N^{1+I_{n>0}}, n+1, \alpha)P(\bar{S}|N^{-2}, n, \alpha). \quad (6.45)$$

As in the previous interval, the interfering piconet's master packet is always transmitted, giving

$$P(D_{SP}|T_6, n, N, \alpha) = P(S|N^{-2}, n, \alpha). \quad (6.46)$$

Since only the packet payloads are vulnerable,

$$P(D_M|T_6, n, N, \alpha) = P(D_{MP}|T_6, n, N, \alpha) \quad (6.47)$$

and

$$P(D_S|T_6, n, N, \alpha) = P(D_{SP}|T_6, n, N, \alpha). \quad (6.48)$$

Equation (6.20) applies in determining $\phi(T_6, n, N, \alpha)$.

6.3.7 PER in T_7

The PER in T_7 is similar to that in T_3 ; each packet is vulnerable to two packets from the other piconet as shown in Piconet 3 of Figure 6.8. Packet M_{-1} may disrupt the MH and S_{-1} may disrupt the MP. However, S_{-1} is only transmitted if M_{-1} 's header does not collide with the previous SP. Thus,

$$P(D_{MH}|T_7, n, N, \alpha) = P(M|N^{-1+I_{n>0}}, n+1, \alpha), \quad (6.49)$$

$$\begin{aligned}
P(D_{MP}|T_7, n, N, \alpha) &= \\
&P(M|N^{1+I_n>0}, n+1, \alpha)P(\overline{D_{SP}}|T_7, n, N, \alpha),
\end{aligned} \tag{6.50}$$

$$\begin{aligned}
P(D_{SH}|T_7, n, N, \alpha) &= \\
&\frac{P(\overline{M} \cap S|N^{-1+I_n>0}, n+1, \alpha)P(\overline{D_{SP}}|T_7, n, N, \alpha)}{P(\overline{M}|N^{-1+I_n>0}, n+1, \alpha)},
\end{aligned} \tag{6.51}$$

and

$$\begin{aligned}
P(D_{SP}|T_7, n, N, \alpha) &= \\
&P(S|N^{-2}, n, \alpha)P(\overline{M}|N^{1+I_n>0}, n+1).
\end{aligned} \tag{6.52}$$

6.3.8 PER in T_8

Finally, in interval T_8 , packet headers are vulnerable to disruption, but a packet is only vulnerable to a single packet from the interfering piconet. Packet M_{-1} may disrupt the MH and S_{-1} may disrupt the SH. The interfering piconet's packet headers are not vulnerable. Thus,

$$P(D_{MH}|T_8, n, N, \alpha) = P(M|N^{-1+I_n>0}, n+1, \alpha), \tag{6.53}$$

$$\begin{aligned}
P(D_{SH}|T_8, n, N, \alpha) &= \\
&\frac{P(\overline{M} \cap S|N^{-1+I_n>0}, n+1, \alpha)}{P(\overline{M}|N^{-1+I_n>0}, n+1, \alpha)},
\end{aligned} \tag{6.54}$$

and

$$\phi(T_8, n, N, \alpha) =$$

$$\frac{P(M|N^{-1+I_n>0}, n+1, \alpha) + P(\overline{M} \cap S|N^{-1+I_n>0}, n+1, \alpha)}{1 + P(\overline{M}|N^{-1+I_n>0}, n+1, \alpha)}. \quad (6.55)$$

6.4 PER Probability Mass Function (pmf)

The pmf, $f_{\Phi}(\phi)$, for the PER between a random pair of piconets is comprised of the values of $\phi(T_k, n, N, \alpha)$ for all $\Delta_{i,j}$, $n_{i,j}$, $N_{i,j}$, $A1$, $B1$, $A2$, and $B2$ and is

$$f_{\Phi}(\phi) = \sum_{k=1}^8 \sum_{n=0}^{31} \sum_{N=0}^{78} \sum_{A1=0}^{31} \sum_{A2=0}^{31} \sum_{B1=0}^{15} \sum_{B2=0}^{15} \frac{P(K=k)\delta(\phi = \phi(T_k, n, N, \alpha))}{662,700,032}. \quad (6.56)$$

The probability of each combination of relational parameters is $32 \times 32 \times 32 \times 16 \times 16 \times 79 = 662,700,032$. The probability $P(K=k)$ is taken from Table 6.1. For example,

$$P(K=2) = \frac{(258+y) - (127-y)}{1250} \quad (6.57)$$

where y is the number of bits that packets must overlap in order to interfere with the Header.

The expected PER, $E[\phi(T_k, n, N)]$, is 0.01454 with a standard deviation of 0.00953 and a maximum of 0.1875. The expected PER is slightly below the PER of 0.0148 obtained by assuming the frequencies in the hop sequence are uniformly distributed over the 79 Bluetooth frequencies [ElH01]. This is expected since the

probability of the slave packet being disrupted is lower than the probability that the master packet is disrupted in certain cases due to the dependance and the fact that the slave packet is often not transmitted at a time when it is likely to be disrupted. In network design and analysis, assuming the PER for each piconet pair lies at 0.0148 is not only inaccurate, it also fails to account for bimodal nature of the mass function. With one mode centered at at 0.0215 and the other near zero, using 0.0148, or even 0.01454, results in inaccurate piconet performance predictions.

A MATLAB[®] simulation was developed and compared to the distribution. Each PER analysis consisted of packet collision analysis of 10 MSI cycles (i.e., 0.4 seconds) between piconets with randomly generated address, clock and δ parameters. The resultant pmf of 2400 simulated PERs is shown in Figure 7.1. The null hypothesis that our simulated distribution is statistically equivalent to the derived distribution is not rejected at the 0.05 level for either the Kolmogorov-Smirnov ($KS = 0.721$, $Critical = 1.358$) or Cramér-von Mises ($CV = 0.279$, $Critical = 0.461$) tests.

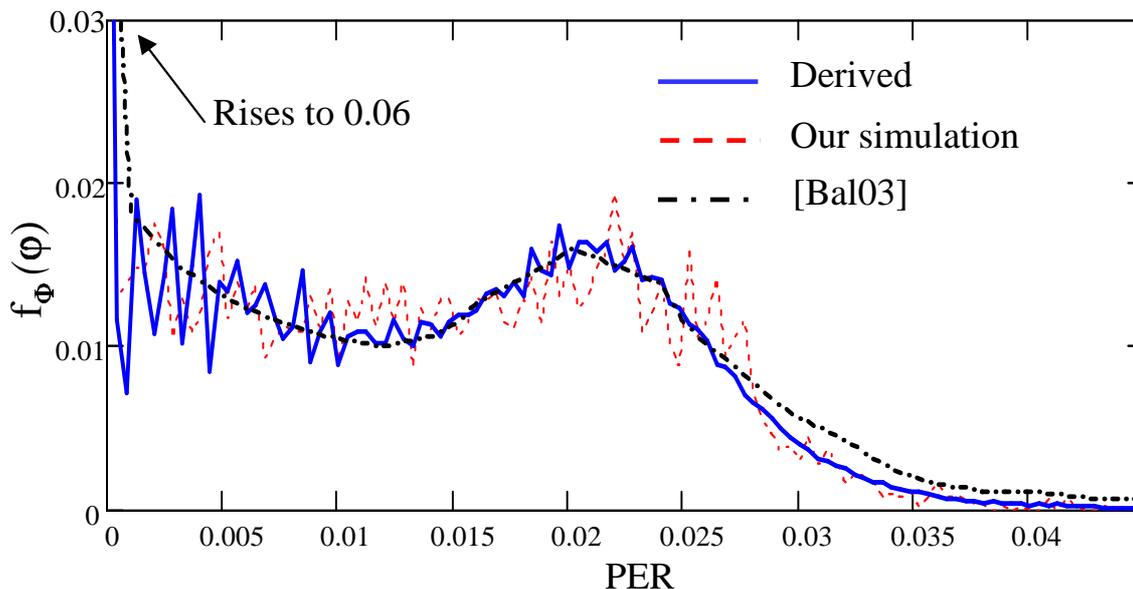


Figure 6.10: Probability mass function of PER between two random piconets.

An independently developed simulation was used in a previous inter-piconet PER analysis [Bal03]. Multiple replications were run to establish a PER, but a

complete distribution was not developed which integrated all possible temporal relationships. Although the original data was no longer available, the distributions were manually extracted from the published plots and combined to form the approximate pmf shown in Figure 7.1. The mean PER produced was 0.01483. Although acknowledging the impact of the FHSK on the PER, an analytical expression for the distribution was not developed. However, simulation showed that the collisions are bursty in nature with up to 20 seconds between bursts [Bal03]. This is understandable given the nature of the placement process. Since placements are not repeated in a MSI cycle, a second collision is more likely to occur when a first collision occurs. A collision reduces the possible number of placements which can prevent a collision. For example, if two piconets are spectrally aligned and have placement possibilities of 1, 2, 4, 8, and 16 and both use placement 1 (which occurs with probability 0.2), the probability that a second collision will occur within the placement set rises to 0.25. If that first collision had not occurred, the probability of a collision on the second placement is only 0.1875 since each piconet has a possible placement in the set which cannot possibly cause a collision.

6.5 PER for Multiple Piconets

Since the clock and address for the master of a piconet is independent of all other piconet masters, their relationships, and therefore their PERs are also independent. Therefore, the pmf for the PER for m neighboring piconets can be found by multiplying $f_{\Phi}(\phi)$ by itself $m - 1$ times (since $f_{\Phi}(\phi)$ is the PER pmf when $m = 1$). However, the number of unique PERs in the pmf increases exponentially. Denoting the pmf for the PER with m neighboring piconets as $f^m_{\Phi}(\phi)$,

$$f^m_{\Phi}(\phi) = \sum_{\phi_{\alpha} \in S_{m-1}} \sum_{\phi_{\beta} \in S_1} f^{m-1}_{\Phi}(\phi_{\alpha}) f^1_{\Phi}(\phi_{\beta}) \delta(1 - (1 - \phi_{\alpha})(1 - \phi_{\beta}) - \phi). \quad (6.58)$$

where S_m is the set of non-zero PER values in $f^m\Phi(\phi)$. The pmfs for $m = 2, 5, 10$, and 100 are shown in Figure 6.11.

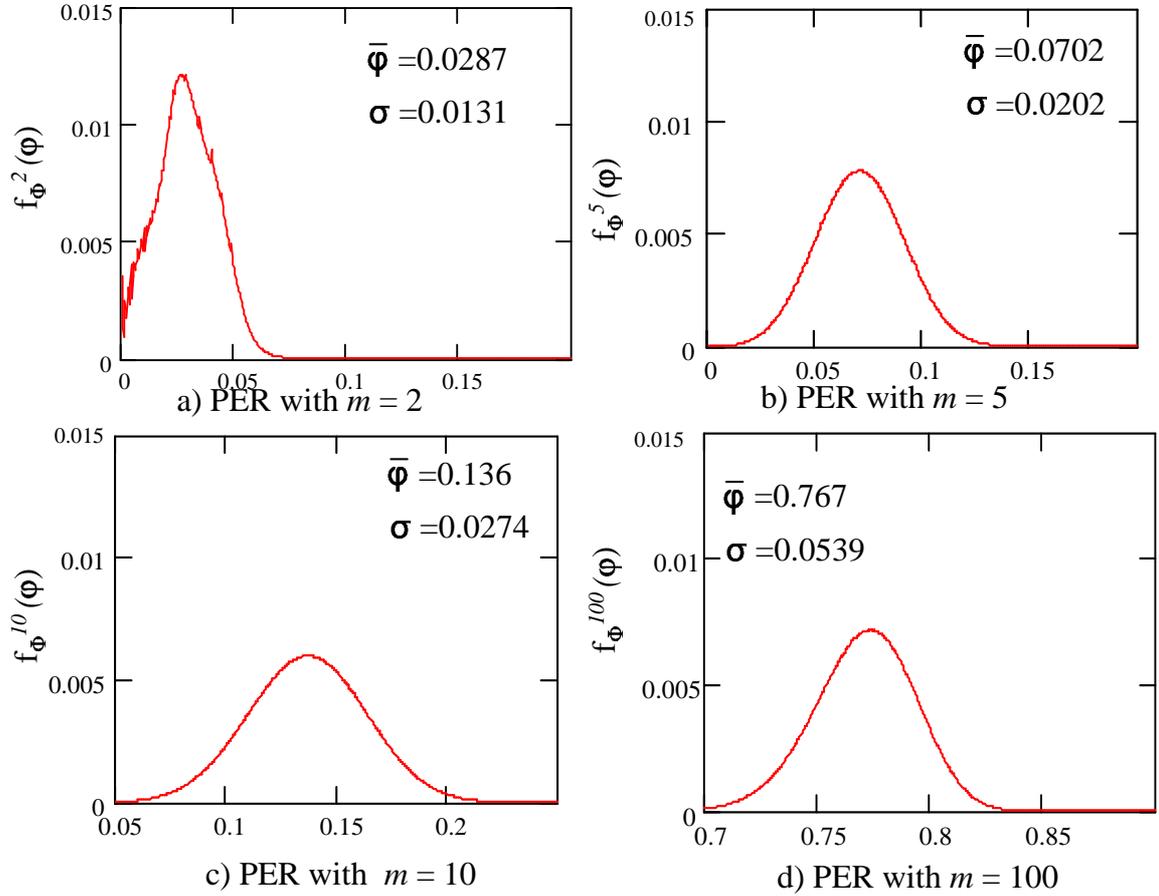


Figure 6.11: Probability mass function of PER between m random piconets.

6.6 Goodput Derivation

The process used to derive the pmf for the expected goodput, $g(T_k, n_{i,j}, N_{i,j}, \alpha)$, for a piconet pair is similar to that used to derive the PER since the two quantities are related. The goodput cannot be directly extrapolated from the PER, however, since some master packets may be correctly received by the slave but must be retransmitted. If the header of the slave packet following a successfully received master packet is disrupted, the acknowledgement (ACK) of the reception of the master's packet

will not be received by the master. The goodput derivation is therefore partitioned by T_k where $1 \leq k \leq 8$.

6.6.1 Goodput in T_1

In the T_1 interval, the MH and SH are both vulnerable. Good data is only received when neither the master nor slave packets are disrupted. If the master packet is disrupted, the slave will not transmit data. However, if the master packet is successfully received but the slave packet is disrupted, the master does not receive the ACK included in the SH and therefore re-transmits the preceding packet. Effectively, neither the master nor slave packets are successfully received if *either* share a channel with another piconet. Therefore, $g(T_1, n_{1,2}, N_{1,2}, \alpha)$, the goodput of a piconet with a single neighboring piconet in interval T_1 , is

$$g(T_1, n, N, \alpha) = 2G_{max}P(\overline{M} \cap \overline{S}|N^0, n, \alpha). \quad (6.59)$$

The maximum goodput for a single device (master or slave), G_{max} , is achieved when a master and slave packet are successfully received in every MSTs. Since each single-slot packet with no FEC contains 216 data bits and 800 MSTs occur each second, $G_{max} = 172.8$ kbps. Thus the maximum goodput a piconet may have is $2G_{max}$, or 345.6 kbps.

6.6.2 Goodput in T_2

In interval T_2 , only the payload of packets are subject to interference, so the goodput is directly related to the PER. Assuming no other interference or packet errors

$$g(T_2, n, N, \alpha) = G_{max}((P(\overline{M}|N^0, n, \alpha) + P(\overline{S}|N^0, n, \alpha))). \quad (6.60)$$

6.6.3 Goodput in T_3

A packet's header and payload are vulnerable to different packets in interval T_3 . Again, if the MH is disrupted, the subsequent slave packet is not transmitted. Since the MH is vulnerable to the slave from the previous MSTs, the probability of a transmitted slave packet being disrupted is independent of whether the MH is disrupted. Also, the master packet must be repeated if the SH is disrupted. Thus, the master packet delivers good data when it is not disrupted and when the subsequent SH is not disrupted. The slave packet delivers good data when it is not disrupted which only occur if the preceding MH is not disrupted, giving

$$g(T_3, n, N, \alpha) = G_{max} (P(\overline{D_M}|T_3, n, N, \alpha)P(\overline{D_{SH}}|T_3, n, N, \alpha) + P(\overline{D_S}|T_3, n, N, \alpha)P(\overline{D_{MH}}|T_3, n, N, \alpha)). \quad (6.61)$$

6.6.4 Goodput in T_4

In interval T_4 , packets are only vulnerable to one packet, so

$$g(T_4, n, N, \alpha) = 2G_{max}P(\overline{D_{MH}}|T_4, n, N, \alpha)P(\overline{D_{SH}}|T_4, n, N, \alpha). \quad (6.62)$$

6.6.5 Goodput in T_5

In interval T_5 , the same configuration exists as in T_4 , giving

$$g(T_5, n, N, \alpha) = 2G_{max}P(\overline{D_{MH}}|T_5, n, N, \alpha)P(\overline{D_{SH}}|T_5, n, N, \alpha). \quad (6.63)$$

6.6.6 Goodput in T_6

The expression for the goodput in interval T_6 is very similar to T_5 except only payloads are affected. Packets will be sent in all packet slots and no successfully received master packet must be repeated. The goodput is

$$g(T_6, n, N, \alpha) = G_{max}(P(\overline{D}_M|T_6, n, N, \alpha) + P(\overline{D}_S|T_6, n, N, \alpha)). \quad (6.64)$$

6.6.7 Goodput in T_7

Data are only successfully transmitted in T_7 when neither the master nor slave packet headers are disrupted. Even then, the payload of the packets are vulnerable. Therefore

$$g(T_7, n, N, \alpha) = G_{max}(P(\overline{M} \cap \overline{S}|N^{-1+I_n>0}, n+1, \alpha)P(\overline{D}_{MP}|T_7, n, N, \alpha) + P(\overline{M} \cap \overline{S}|N^{-1+I_n>0}, n+1, \alpha)P(\overline{D}_{SP}|T_7, n, N, \alpha)). \quad (6.65)$$

6.6.8 Goodput in T_8

In interval T_8 , conditions are similar to those in T_7 except that packets are only vulnerable to a single packet. Thus,

$$g(T_8, n, N, \alpha) = 2G_{max}P(\overline{M} \cap \overline{S}|N^{-1+I_n>0}, n+1, \alpha). \quad (6.66)$$

6.7 Goodput Probability Mass Function

The pmf, $f_G(g)$, for the goodput between a random pair of piconets is comprised of the values of $g(T_k, n, N, \alpha)$ for all $\Delta_{i,j}$, $n_{i,j}$, $N_{i,j}$, $A1$, $B1$, $A2$, and $B2$. Using the same values for $P(K = k)$ as in (6.56), it is

$$f_G(\gamma) = \sum_{k=1}^7 \sum_{n=0}^{31} \sum_{N=0}^{78} \sum_{A1=0}^{31} \sum_{A2=0}^{31} \sum_{B1=0}^{15} \sum_{B2=0}^{15} \frac{P(K = k)\delta(\gamma = g(T_k, n, N, \alpha))}{662,700,032}. \quad (6.67)$$

Equation (6.67) produced the pmf shown in Figure 6.12 with a expected piconet goodput of 337.32 kbps and a standard deviation of 5.93 kbps.

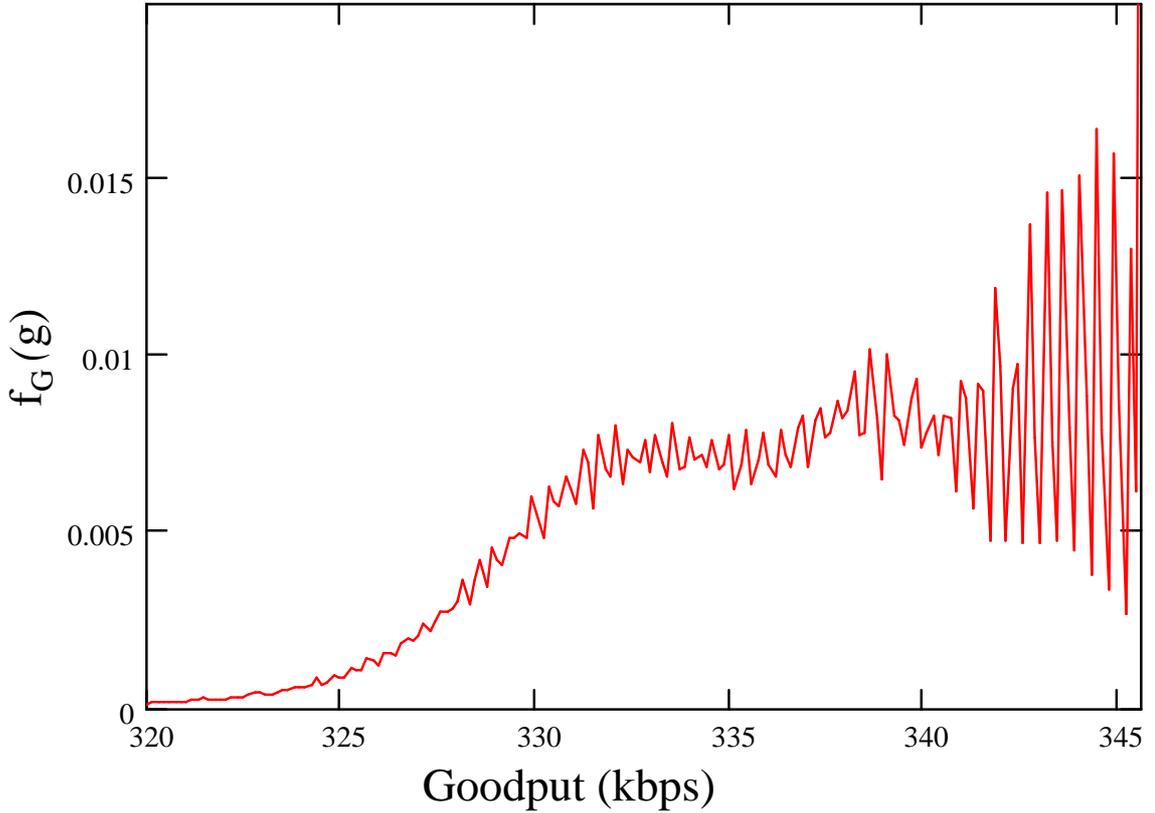


Figure 6.12: Probability mass function of goodput between two random piconets.

6.8 Goodput with Multiple Piconets

Although it is possible to derive the pmf for the goodput with multiple neighboring piconets, it too is quite cumbersome. With 8 possible intervals, the expression for the goodput has 64 possible combinations of vulnerabilities when two interfering piconets are used. Therefore, the goodput analysis for multiple interferers if necessary, should be accomplished on an individual basis when the intervals of each piconet is known.

6.9 Summary

The frequency selection in Bluetooth is often assumed to be uniform over the 79 frequencies, producing a packet error rate of 0.0127 for those piconets whose packets overlap only one piconet and 0.252 for those piconets whose packets overlap two packets in a neighboring piconet. Weighted by the likelihood of each occurring, this produces an expected PER of 0.0148. Although the frequency selection is uniform in the limit, the localized pattern of packet selection and the relation between two piconets can drive the PER as high as 0.1875. The distribution of the PER for a random pairing of piconets is bimodal in nature, a mode near zero and one centered at 0.021 with 99% of its mass lying between zero and 0.036. The shape of the distribution is important in BT network analysis and using the mean value of 0.01454 seriously underestimates the PER of many of the possible piconet pairings. This becomes significant in scatternet design and determining appropriate collision avoidance techniques. For example, switching to a different device as master of a piconet can markedly improve or degrade the PER. Likewise, knowledge of the other piconet's clock and address values can allow piconets to determine if it is necessary to implement collision avoidance methods such as adaptive frequency hopping on negotiated partition of the spectrum or active avoidance using knowledge of the other piconets hop sequence. Since the PER is often above 0.02 and possibly as high as

0.1875, a collision avoidance technique is developed in Chapter VII and compared to an existing collision avoidance technique.

VII. Collision Avoidance Techniques

Piconets sometimes form larger networks, or scatternets, where neighboring piconets are deliberately within transmission range. As noted in the previous chapter, the probability of interference between two piconets can be as large as 0.1875. When multiple piconets form a scatternet, the expected interference can become significant and must be mitigated if possible. Using protocol information within the scatternet, it is possible to predict the frequencies used by neighboring piconets to de-conflict such interference. In a scatternet, bridge devices can pass the address, clock, and timing offset data from one piconet to the master of another, giving it the capability to calculate what frequencies the other piconet will be using for single-slot packets. By reducing packet collisions, throughput is increased and energy is not wasted on packets doomed to collide. The BT protocol for scatternets has not been fully specified and no collision avoidance method has been adopted. Therefore, even slight improvements to over proposed methods may be useful.

In addition to AFH proposed in v1.2 of the BT standard which can partition the BT spectrum and prevent collisions between the packets in a piconet [Blu03], PIAM prevents collisions between packets sending only single-slot packets (See Section 2.5.5.2). Recall that PIAM establishes a priority scheme based on the sum of the frequencies used by the master and slave packets to determine which device is authorized to use a channel with conflict potential. In addition to being limited to single-slot packets, the method is inefficient since neither device is authorized to use the channel when the frequencies in a time slot are the same for both piconets since the frequency sums are equivalent.

PIAM compares the sum of the master/slave frequencies, or TSFS, for packets scheduled to collide to determine which piconet is authorized to use the channel. This produces inefficiencies – time slots may go unused by all piconets in the scatternet.

Thus, a method for collision avoidance, the Avoidant Forward Inspection Technique (AFIT), uses elements of a collision avoidance algorithm based on link history and the likelihood of a noisy channel. By considering hop frequency when selecting packet size, a BT hop sequence can be manipulated to avoid certain channels such that packet collisions are less likely to occur [AnK00]. Since the precise temporal relationship between piconets can be determined, POLL/NULL packets can be used to transmit data in one direction of a master/slave packet exchange when it is impossible to transmit effectively in both directions.

Analysis of collision avoidance effectiveness is difficult due to the complicated temporal and spectral relationship between piconet hop sequences. The goodput for each piconet pair is dependent on the relationships between the piconets and can vary widely between piconet pairings. Therefore, using an expected goodput measure does not adequately predict goodput that can be achieved between an arbitrary pair of piconets. In this chapter, a detailed analysis of collision avoidance methods is presented along with the derivation of a goodput probability mass function (pmf) for each collision avoidance method assuming piconets use single-slot packets. The derived pmfs are compared to simulated pmfs. Additionally, it is shown that AFIT provides slightly greater goodput than PIAM when using single-slot packets and significantly increases goodput when applied to piconets using multi-slot packets, even when the packet-length of neighboring piconets is unknown.

7.1 Avoidant Forward Inspection Technique (AFIT)

AFIT uses knowledge of neighboring piconets' hop sequence and dynamic packet size selection to maximize throughput while looking forward in time to avoid collisions. Using AFIT, a master must only avoid collisions with packets in MSTs which begin after the master packet. For example, in Figure 6.8, Piconet 1 must only avoid collisions between its slave packet and M_0 in Piconet 2. Piconet 2 is responsible for avoiding collisions between the master packet and M_{-1} and S_{-1} as

well as between the slave packet and S_{-1} since $MSTS_{-1}$ began prior to the $MSTS$ containing Piconet 1's master and slave packet. However, Piconet 1, must avoid collision with both M_0 and S_0 from Piconet 3.

Using AFIT, piconets avoid collisions with as little impact to throughput as possible. If only the payload of a piconet's packet is scheduled to collide, the collision can be avoided if the piconet transmits a POLL/NULL packet, which consists of a header with no payload. Therefore, the associated master/slave packets can still transmit data and, in the case of the slave packet, successfully return an ACK so the master does not resend its data. In Figure 6.8, Piconet 1's slave could avoid an imminent collision with Piconet 2's M_0 by sending a NULL packet or even by sending a payload of 17 bytes (rather than the full 28 bytes), which allows the packet to end 4 μs prior to the beginning of the M_0 packet. This means the master packet can be received and successfully acknowledged. To simplify the analysis, only single-slot full payloads or POLL/NULL packets are considered as options for transmission.

Returning a NULL packet may have an impact on some scheduling algorithms which assume that slaves that return NULL packets have no buffered data to send. This can be countered by using a bit in the `LT_ADDR` field of the Packet Header to inform the master that a NULL was sent for collision avoidance purposes and that the slave does have buffered data to transmit. The `LT_ADDR` field is only used during master packet transmissions and is available during slave packet transmissions. Similarly, if only the master packet's payload is scheduled to collide, the master can transmit a POLL packet to the slave, which allows the slave to respond while still avoiding the collision. Scheduling impacts are not an issue in this case since the master performs the scheduling.

Due to the dynamic packet scheduling, the goodput must be determined to compare performance between AFIT and PIAM. In the initial presentation of PIAM, only the probability of successful transmission of master/slave pairs was determined [PGR03]. Furthermore, the available performance analysis on PIAM assumed that

the master and slave frequencies were uniformly chosen across the 79 frequencies in the BT spectrum with the exception that a given frequency cannot be used for two consecutive packets [PGR03]. Although simplifying the analysis, this is not accurate and the probability of frequency collisions could be much greater. To determine the goodput probability mass function, the frequency hop process must be accurately characterized.

7.2 *Analysis Necessities*

7.2.1 *Time Slot Frequency Sum (TSFS)*

The TSFS for a piconet pair must be calculated. The placement of the MSI/SSIs within the spectrum, as well as $N_{i,j}$, is significant in determining the TSFS. For example, if $N_{i,j} = 2$, one might expect the probability that Piconet j 's TSFSs are greater than Piconet i 's to be slightly greater than 0.5 since its MSI/SSI are always two frequencies to the right of Piconet i 's. However, such an estimation not only fails to account for the cases where the TSFSs are equal, but also the fact that Piconet i has a numerical advantage when the MSIs shift such that Piconet j 's interval scrolls beyond frequency 78, scrolling to frequencies with smaller numerical values. Therefore, the relations are dependent on the specific placements of the intervals, which are uniformly distributed over time. The expected probability the TSFS of a piconet is greater than another piconet's should be used. This expectation varies with $N_{i,j}$, $n_{i,j}$, A_i , B_i , A_j , and B_j . Since the numerical value of the Bluetooth frequency of the 232 placement combinations are fairly random when the MSI is shifted across the spectrum and spread by the final stage of the FHSK, only $N_{i,j}$ is used in deriving $P(H|N)$, the probability that Piconet i 's TSFS is larger than Piconet j 's, where $N = N_{i,j}$. If $n_{1,2}$ is included determining $P(H)$, all other parameters must also be included. This complicates the expression while adding little to the accuracy of the derivation. Using the same notation as in (6.13),

$$\begin{aligned}
P(H|N) = & \sum_{M=0}^{78} \sum_{i1=0}^{31} \sum_{j1=0}^{(L(b_{i1})-1)} \sum_{i2=0}^{31} \sum_{j2=0}^{(L(b_{i2})-1)} \frac{1}{4,252,096} \times \\
& I_{(FM1+FS1)>(FM2+FS2)} I_{(i1 \neq S_{b_{i1}}(j1)) \cup (L_S(b_{i1}) \neq 9)} \times \\
& I_{(i2 \neq S_{b_{i2}}(j2)) \cup (L_S(b_{i2}) \neq 9)}
\end{aligned} \tag{7.1}$$

where

$$\begin{aligned}
FM1 &= \text{mod}(2 \times (i1 + M), 79), \\
FS1 &= \text{mod}(2 \times (S_{b_{i1}}(j1) + M + 32), 79), \\
FM2 &= \text{mod}\left(2 \times (S_{b_{Z2(\text{mod}(X-n,32))}}(j) + M + N^\theta), 79\right) \\
FS2 &= \text{mod}\left(2 \times (S_{b_{Z2(\text{mod}(X-n,32))}}(r) + M + N^\theta), 79\right).
\end{aligned}$$

This expression simply generates a comparison for every master/slave frequency pair that can occur within each MSI cycle for each of two piconets and sums the probability of each occurring when the first is larger than the second. Since there are 232 combinations of master/slave pairs for each piconet at each of the 79 MSI/SSI intervals, there are $232^2 \times 79 = 4,252,096$ individual combinations. Note the frequencies are doubled, mod 79, in $FM1$, $FS1$, $FM2$, and $FS2$ to incorporate the spreading stage of the FHSS that is ignored in the remainder of the analysis. Since the sum of frequencies is based on the frequency used in the BT spectrum, the spreading stage cannot be ignored in the TSFS.

7.3 Goodput Probability Mass Function

The pmf for the goodput of a piconet with a random interferer, each using single-slot packets, is determined by weighting the possible goodput values with their likelihood. Thus, an expression for the goodput for each combination of $\Delta_{i,j}$, $n_{i,j}$, $N_{i,j}$, and α must be derived. The temporal relationship between packets is significantly different for the seven intervals, so the expression is developed piece-wise on $\Delta_{i,j}$. Throughout the derivation, $P(M|N^\theta, n, \alpha) = P(M \cap S|N^\theta, n, \alpha) + P(M \cap \bar{S}|N^\theta, n, \alpha)$ and $P(S|N^\theta, n, \alpha) = P(M \cap S|N^\theta, n, \alpha) + P(\bar{M} \cap S|N^\theta, n, \alpha)$.

7.3.1 Goodput in T_1

In the T_1 interval, the MH and SH are both vulnerable. Good data is received when neither the master nor slave packets are disrupted. If the master packet is disrupted, the slave will not transmit data. However, if the master packet is successfully received but the slave packet is disrupted, the master does not receive the ACK included in the SH and therefore retransmits the preceding packet. Effectively, neither the master nor slave packets are successfully received if *either* share a channel with another piconet.

Therefore, using AFIT, a master packet is only transmitted when neither the master nor the slave shares a frequency with the interfering piconets master or slave. The goodput of a piconet using AFIT with a single neighboring piconet in interval T_1 , $g^A(T_1, n_{1,2}, N_{1,2}, \alpha)$, is

$$g^A(T_1, n, N, \alpha) = 2G_{max}P(\bar{M} \cap \bar{S}|N^0, n, \alpha). \quad (7.2)$$

The maximum goodput within a piconet using single-slot packets, G_{max} , is achieved when a master and slave packet are successfully received in every MSTs. Since each single-slot packet with no FEC contains 216 data bits and 800 MSTs

occur each second, $G_{max} = 172.8$ kbps. Thus the maximum goodput a piconet may have is $2G_{max}$, or 345.6 kbps.

When using PIAM, however, packets will be sent when a collision is imminent and the TSFS is greater than that of the other piconet. Therefore, the goodput of a piconet using PIAM with a single neighboring piconet in interval T_1 , $g^P(T_1, n_{1,2}, N_{1,2}, \alpha)$, is

$$g^P(T_1, n, N, \alpha) = 2G_{max} \left(P(\overline{M} \cap \overline{S} | N^0, n, \alpha) + (1 - P(\overline{M} \cap \overline{S} | N^0, n, \alpha)) P(H|N) \right). \quad (7.3)$$

7.3.2 Goodput in T_2

In interval T_2 , AFIT prevents packet transmissions only when both the master and slave are to be disrupted since only the payload of packets are subject to interference. When one of the payloads will collide, a POLL/NULL packet is sent so the other full packet of data in the MSTs is transmitted. For example, if Piconet 3's M_0 in 6.8 shares a frequency with Piconet 1's master packet, a collision is avoided if Piconet 1 sends a POLL packet which allows the slave device in Piconet 1 to respond. Thus,

$$g^A(T_1, n, N, \alpha) = 2G_{max} (P(\overline{M} \cap \overline{S} | N^0, n, \alpha) + P(\overline{M} \cap S | N^0, n, \alpha)). \quad (7.4)$$

Note that in all cases, $P(\overline{M} \cap S | N^0, n, \alpha) = P(M \cap \overline{S} | N^0, n, \alpha)$.

Since the vulnerability of the header versus the payload is irrelevant in PIAM,

$$g^P(T_2, n, N, \alpha) = g^P(T_1, n, N, \alpha). \quad (7.5)$$

7.3.3 Goodput in T_3

The first case where each packet is vulnerable to collision by two separate packets occurs in T_3 . Since the MH is vulnerable to the slave from the previous MSTS, the probability of the slave packet colliding with another packet is independent of whether the MH is disrupted. Since AFIT only looks ahead, the vulnerability of the MH is not a factor. The other piconet insures that collision won't occur either by sending a NULL packet for S_{-1} or no packets at all in the MSTS. Thus, if the SH is not on the same frequency as M_0 and neither payloads will collide (i.e., $P(\overline{M} \cap \overline{S} | N^0, n, \alpha)$), both master and slave packets will have payloads. If SH shares a frequency with M_0 (i.e., $P(S | N^{-2}, n, \alpha)$), neither packet is transmitted. When only one packet is to collide (i.e., $P(\overline{M} \cap S | N^0, n, \alpha)$ or $P(M \cap \overline{S} | N^0, n, \alpha)$) one payload is sent while the other packet is a NULL/POLL. Recall that the SSI is 32 frequencies to the right of the MSI and N is based on the alignment of the MSIs between piconets. Thus, N^{-2} must be used to determine the overlap of a SSI with the MSI of the other piconet. Note that the event that the slave packet shares a frequency with M_0 is mutually exclusive of either of the payloads being disrupted. Therefore,

$$g^A(T_3, n, N, \alpha) = 2G_{max}(P(\overline{M} | N^0, n, \alpha) - P(S | N^{-2}, n, \alpha)). \quad (7.6)$$

PIAM's scheme is slightly more complicated. The possibility of the MH sharing a frequency with S_{-1} must also be considered. In addition, it is possible that the the MH and SH both collide. Since they collide with packets from different MSTSs in the other piconet, the TSFS must be greater than the TSFS of both of the other MSTSs in order to transmit. Since the MH is vulnerable to a packet which began before the MH began, n is effectively shifted to the right by one time slot

for that MSTs. Note that, to simplify the expression, $n + 1$ is used when it should properly be $\text{mod}(n + 1, 32)$. Also, as before, the MSI shift, $n_{i,j}$ is ignored in the use of $P(H|N)$. Thus,

$$\begin{aligned}
g^P(T_3, n, N, \alpha) &= 2G_{max} \times \\
&\left((P(\overline{M} \cap \overline{S}|N^0, n, \alpha) - P(S|N^{-2}, n, \alpha))P(\overline{M}|N^{1+I_n>0}, n + 1, \alpha) + \right. \\
&\quad (1 - P(\overline{M} \cap \overline{S}|N^0, n, \alpha) + P(S|N^{-2}, n, \alpha))P(\overline{M}|N^{1+I_n>0}, n + 1, \alpha)P(H|N) + \quad (7.7) \\
&\quad (P(\overline{M} \cap \overline{S}|N^0, n, \alpha) - P(S|N^{-2}, n, \alpha))P(M|N^{1+I_n>0}, n + 1, \alpha)P(H|N) + \\
&\quad \left. (1 - P(\overline{M} \cap \overline{S}|N^0, n, \alpha) + P(S|N^{-2}, n, \alpha))P(M|N^{1+I_n>0}, n + 1, \alpha)(1 - P(\overline{H}|N)^2) \right).
\end{aligned}$$

7.3.4 Goodput in T_4 and T_5

In interval T_4 , each packet is vulnerable to one packet from the neighboring piconet. However, the packets that affect the master and slave are from different MSTs and therefore are independent. The goodput for both avoidance techniques is the same for T_4 and T_5 . The intervals were not combined to retain consistency with the notation in Chapter VI. Again, since AFIT only looks forward, it must only de-conflict collisions with the SH, giving

$$\begin{aligned}
g^A(T_4, n, N, \alpha) &= g^A(T_5, n, N, \alpha) \quad (7.8) \\
&= 2G_{max}(1 - P(S|N^{-2}, n, \alpha)).
\end{aligned}$$

PIAM, on the other hand, must de-conflict all collisions, producing

$$\begin{aligned}
g^P(T_4, n, N, \alpha) &= g^P(T_5, n, N, \alpha) \\
&= 2G_{max} \times \\
&\quad \left((P(\bar{S}|N^{-2}, n, \alpha))P(\bar{M}|N^{1+I_n>0}, n+1, \alpha) + \right. \\
&\quad (P(S|N^{-2}, n, \alpha))P(\bar{M}|N^{1+I_n>0}, n+1, \alpha)P(H|N) + \\
&\quad (P(\bar{S}|N^{-2}, n, \alpha))P(M|N^{1+I_n>0}, n+1, \alpha)P(H|N) + \\
&\quad \left. (P(S|N^{-2}, n, \alpha))P(M|N^{1+I_n>0}, n+1, \alpha)(1 - P(\bar{H}|N)^2) \right).
\end{aligned} \tag{7.9}$$

7.3.5 Goodput in T_6

The relationships in interval T_6 are similar to those in T_4 and T_5 except only the payloads are vulnerable to collision instead of the headers. This allows AFIT to transmit data in the master packet even when the slave packet payload cannot be transmitted and a NULL packet must be transmitted. Therefore,

$$g^A(T_6, n, N, \alpha) = G_{max}(2 - P(S|N^{-2}, n, \alpha)) \tag{7.10}$$

and

$$g^P(T_6, n, N, \alpha) = g^P(T_5, n, N, \alpha). \tag{7.11}$$

7.3.6 Goodput in T_7

The second interval where packets are vulnerable to two packets from the other piconet is T_7 . Looking forward, AFIT must again only avoid collisions with the SP. Therefore

$$g^A(T_7, n, N, \alpha) = G_{max}(2 - P(S|N^{-2}, n, \alpha)). \quad (7.12)$$

However, PIAM must avoid collision with all other packets. Again, when collisions are imminent from packets in separate MSTs, the TSFS must be greater than the TSFSs from both of the MSTs in order to allow transmission of the master/slave pair.

Thus,

$$\begin{aligned}
g^P(T_7, n, N, \alpha) = & 2G_{max} \times \\
& \left((P(\overline{M} \cap \overline{S}|N^{-1+I_n>0}, n+1, \alpha) - P(M|N^{1+I_n>0}, n+1, \alpha))P(\overline{S}|N^{-2}, n, \alpha) + \right. \\
& (1 - P(\overline{M} \cap \overline{S}|N^{-1+I_n>0}, n+1, \alpha) + P(M|N^{1+I_n>0}, n+1, \alpha)) \times \\
& \quad P(\overline{S}|N^{-2}, n, \alpha)P(H|N) + \\
& (P(\overline{M} \cap \overline{S}|N^{-1+I_n>0}, n+1, \alpha) - P(M|N^{1+I_n>0}, n+1, \alpha)) \times \\
& \quad P(S|N^{-2}, n, \alpha)P(H|N) + \\
& (1 - P(\overline{M} \cap \overline{S}|N^{-1+I_n>0}, n+1, \alpha) + P(M|N^{1+I_n>0}, n+1, \alpha)) \times \\
& \quad \left. P(S|N^{-2}, n, \alpha)(1 - P(\overline{H}|N)^2) \right). \quad (7.13)
\end{aligned}$$

7.3.7 Goodput in T_8

In interval T_8 , conditions are similar to those in T_7 except packets are only vulnerable to a single packet. There are no packets which can disrupt packets using AFIT. Thus,

$$g^A(T_8, n, N, \alpha) = 2G_{max} = 345.6 \text{ kbps}. \quad (7.14)$$

Using PIAM, the probability of collisions is very similar to T_1 except n is shifted by one slot to the right, giving

$$g^P(T_8, n, N, \alpha) = 2G_{max} \left((P(\overline{M} \cap \overline{S} | N^{1+I_n > 0}, n+1, \alpha)) + (1 - P(\overline{M} \cap \overline{S} | N^{1+I_n > 0}, n+1, \alpha)) P(H|N) \right). \quad (7.15)$$

7.4 Avoidance Technique Comparisons

Recall from Chapter VI, the pmf for the derived goodput within a piconet with a random pair neighboring of piconets with no collision avoidance, $f_G(\gamma)$, is comprised of the values of $g(T_k, n, N, \alpha)$ for all $\Delta_{i,j}$, $n_{i,j}$, $N_{i,j}$, $A1$, $B1$, $A2$, and $B2$. Since goodput is dependent on the interval containing $\Delta_{i,j}$ rather than the value of $\Delta_{i,j}$, the expression is simplified by combining device pair in the same interval, T_k . Thus, pmfs for the goodput using AFIT or PIAM with a single neighboring piconet are

$$f_G^A(\gamma) = \sum_{k=1}^8 \sum_{n=0}^{31} \sum_{N=0}^{78} \sum_{A1=0}^{31} \sum_{A2=0}^{31} \sum_{B1=0}^{15} \sum_{B2=0}^{15} \frac{P(K=k) \delta(\gamma = g^A(T_k, n, N, \alpha))}{662,700,032} \quad (7.16)$$

and

$$f_G^P(\gamma) = \sum_{k=1}^8 \sum_{n=0}^{31} \sum_{N=0}^{78} \sum_{A1=0}^{31} \sum_{A2=0}^{31} \sum_{B1=0}^{15} \sum_{B2=0}^{15} \frac{P(K=k) \delta(\gamma = g^P(T_k, n, N, \alpha))}{662,700,032}. \quad (7.17)$$

The probability of each combination of relational parameters is $32 \times 32 \times 32 \times 16 \times 16 \times 79 = 662,700,032$. The probability $P(K=k)$ is taken from Table 6.1. For example,

$$P(K = 2) = \frac{(258 + y) - (127 - y)}{1250}. \quad (7.18)$$

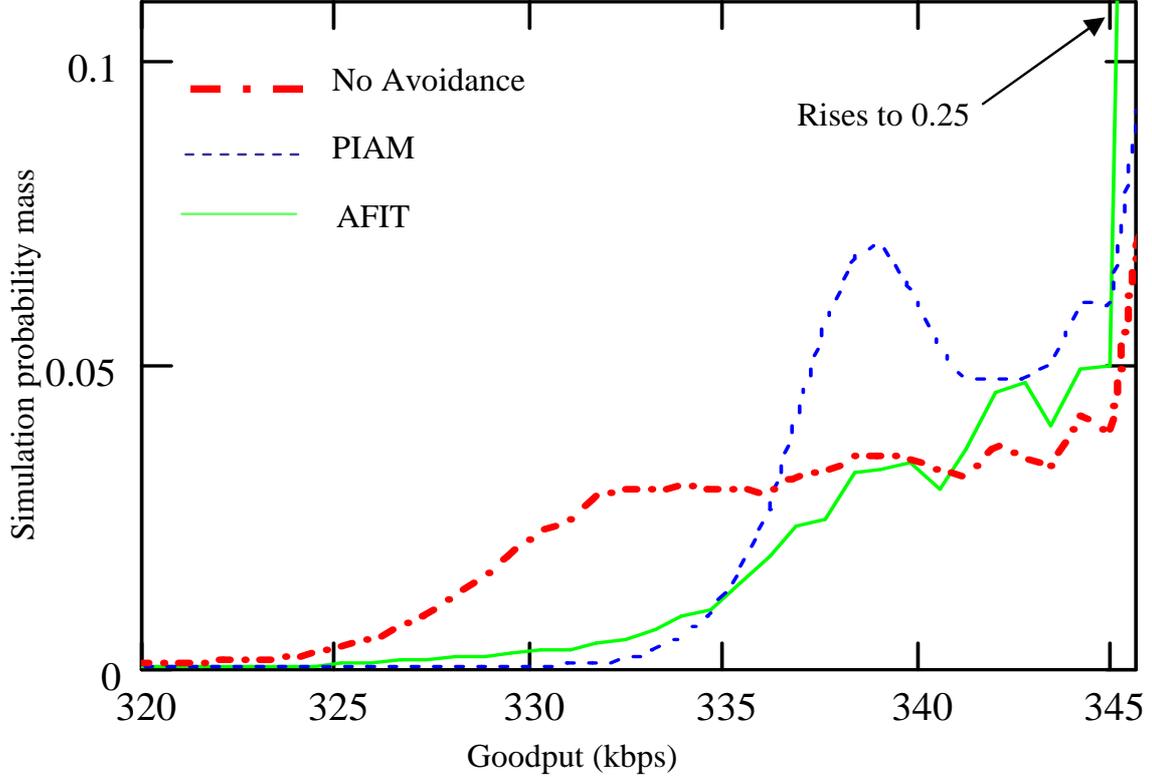


Figure 7.1: Derived probability mass function of goodput between two random piconets.

Since the pmfs are comprised of discrete points, a plot containing the three pmfs is difficult to interpret. Therefore, the points in the pmfs in Figure 7.1 are summed across equal intervals for easier comparison. PIAM produces an expected goodput 1% greater than using no collision avoidance, with an expected goodput of 340.60 kbps, as listed in Table 7.1. However, AFIT provides even better goodput with an expected 341.82 kbps. Although the improvement over PIAM is a mere 0.36%, the benefit increases as m increases and as the packet size extends beyond single-slot packets. The PIAM pmf contains 29.4% of its mass in the interval ranging from 99.8% to 100% of the maximum goodput. By comparison, AFIT contains only

10.6% and the baseline, with no collision avoidance, contains only 7.6% of its mass in the same interval.

Table 7.1: Expected goodput.

	Derived Goodput (kbps)	Derived Improvement (%)	Simulated Goodput (kbps)	Simulated Improvement (%)
Maximum	345.60	-	345.60	-
Baseline	337.32	-	337.75	-
PIAM	340.60	0.97	340.71	0.87
AFIT	341.82	1.33	341.97	1.25

More importantly, energy is conserved since no packets are transmitted which are doomed to failure. Defining efficiency, η , as the number of successfully transmitted user data bits divided by the number of bits transmitted, PIAM is more efficient than AFIT since only full packets are transmitted. Devices using PIAM will always maximize efficiency. AFIT, on the other hand directs the transmission of POLL/NULL packets which contain no user data in order to maximize goodput. Although the efficiency may be derived using methods presented in Section 7.3 and incorporating the probability that a POLL/NULL packet is transmitted, in this research it is analyzed only through simulation.

A MATLAB[®] simulation was developed to compare the techniques and verify the derived result. Each configuration was run for 6.32 s, which allowed the MSI to shift twice through all 79 beginning frequencies. The same seed was used for each data set to ensure that the baseline and each collision avoidance technique faced the same piconet relationships. The simulation was initially run 700 times with both piconets using single-slot packets, producing the goodput pmfs in Figure 7.2. The mean piconet simulated goodput for each technique and the baseline shown in Table 7.1 were all higher than the predicted value, although the derived means fell within the 95% confidence interval. Additionally, none of the confidence intervals overlapped. The consistent difference indicates that the simulation data set had a mean goodput slightly better than that expected with larger sample set. Saturation

was assumed and each device in the piconet transmitted a full single-slot packet whenever possible.

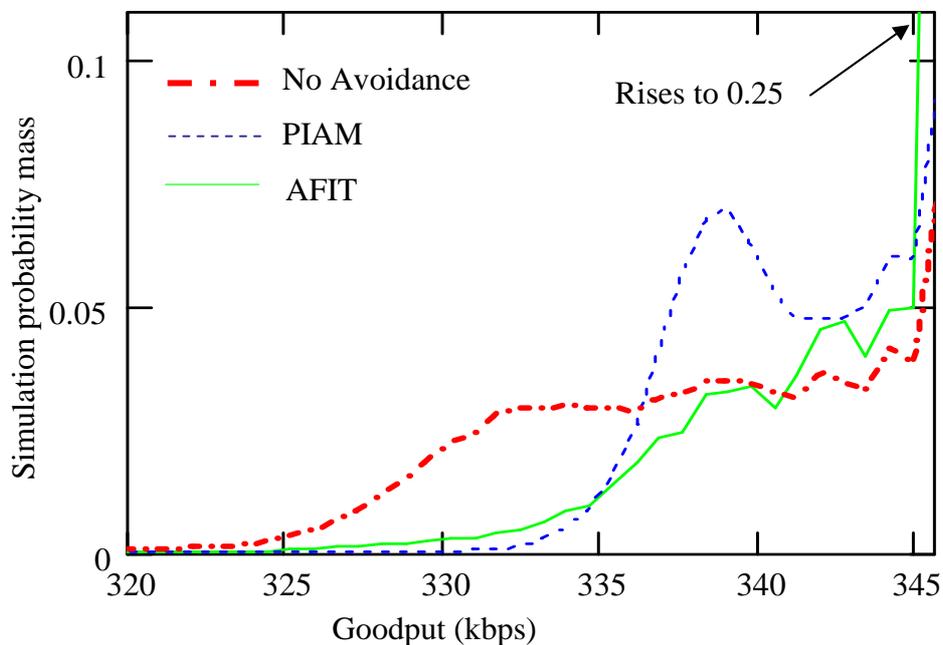


Figure 7.2: Simulated probability mass function of goodput between two random piconets.

As expected, PIAM was very efficient, producing up to 9% improvement in efficiency over the baseline as shown in Table 7.2. The efficiency is normalized to the maximum efficiency that can be produced using single slot packet, 59.1%. AFIT produce similar improvements despite using POLL/NULL packets.

When additional piconets using single-slot packets were introduced, the performance difference between the collision avoidance techniques increased as shown in Figure 7.3. With 5 neighboring piconets, PIAM produced a 4.6% improvement while AFIT showed a 6.5% increase in piconet goodput.

Table 7.2: Expected efficiency.

Piconets	Normalized Baseline η (%)	Normalized PIAM η (%)	PIAM Improvement (%)	Normalized AFIT η (%)	AFIT Improvement (%)
2	98.1	1	1.98	99.9	1.88
3	96.1	1	4.05	99.8	3.85
4	94.4	1	5.98	99.7	5.68
5	92.6	1	7.89	99.7	7.53
6	90.9	1	9.92	99.6	9.45

7.5 Goodput in Multiple Piconets

Since the relational parameters between multiple piconets are independent, the probability that a master/slave packet pair is not transmitted due to an imminent packet collision is also independent when using PIAM. Therefore, the goodput with m neighboring piconets, g_m^A , is

$$g_m^A = 2G_{max} \left(1 - \prod_{i=1}^m \left(1 - \frac{g^P(T_i, n_{1,i+1}, N_{1,i+1}, \alpha_i)}{2G_{max}} \right) \right). \quad (7.19)$$

However, since AFIT uses POLL/NULL packet transmission to maximize goodput, the probability that a master/slave pair is transmitted is not independent between interfering piconets. For example, if a master packet's payload will be disrupted by one piconet and the slave packet's payload disrupted by the other, the master/slave packet pair will not be transmitted at all. Independently avoiding collisions with each neighboring piconet, the technique allows packets to be transmitted since only one payload is affected by each neighboring piconet, but with both imminent collisions, there is no reason to transmit any packets. Therefore, the derived goodput must be conditioned on the relational parameters from each set of interfering piconets. Due to the number of combinations, this is not feasible to condition on each combination

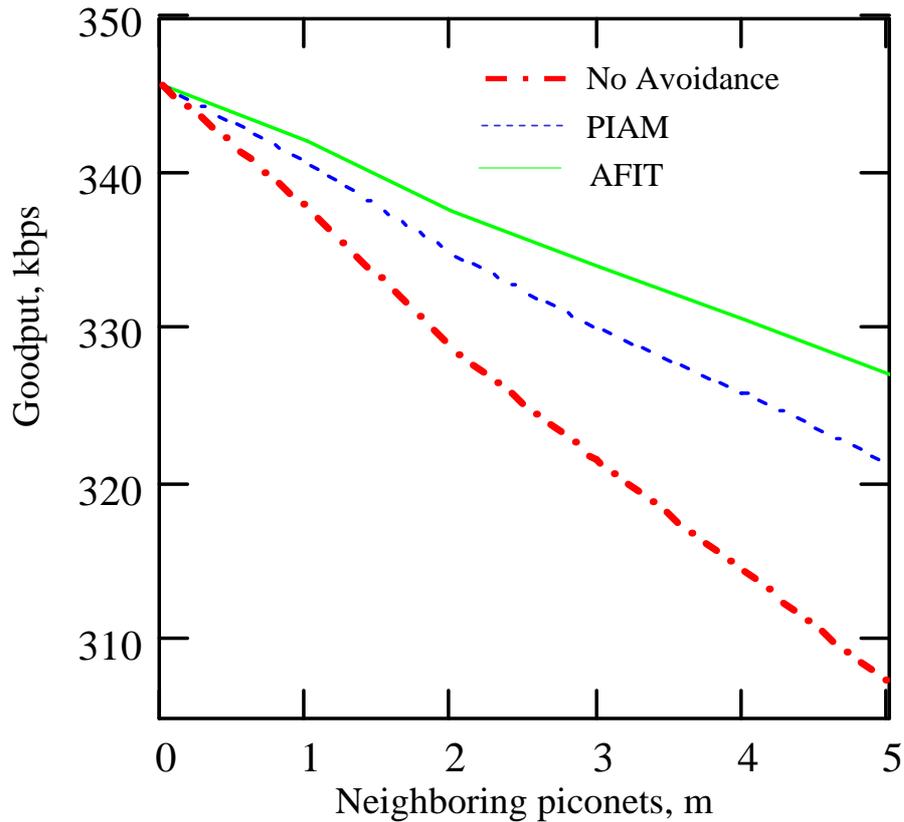


Figure 7.3: Goodput between m random piconets.

for a general expression but must be determined on a case-by-case basis where the parameters are known.

7.6 Collision Avoidance with Multi-slot Packets

Limiting a piconet to single-slot packets is a handicap to the system and unrealistic in normal use. Single-slot packets allocate only 34.6% of the available slot time to user data transmission with the remainder being used for overhead transmission and oscillator re-tuning. Three- and five-slot packets, on the other hand, allocate 78.1% and 86.8%, respectively, for user data. Since multi-slot packets remain on a single transmit frequency for the duration of the packet, no transmission time is lost to oscillator re-tuning. Therefore, it is much more efficient to use multi-slot packets for larger files. Note that the packet subsequent to a multi-slot packet uses

the frequency in the hop sequence that is generated by the FHSK for its time slot. Therefore, when multi-slot packets are used, some frequencies in the single-slot hop sequence are skipped.

Although it is possible to adapt PIAM to allow for multi-slot packets if each piconet has knowledge of packet length for each neighboring piconet, it is unrealistic due to the uncertainty in amount of data to be transmitted by each piconet and the necessity of single-slot packets for SCO, or synchronous, links [Blu03]. Therefore, it is difficult to develop a logical extension to PIAM which includes multi-slot packets without unreasonably degrading goodput.

However, such an extension is feasible for AFIT. Since the technique requires that a piconet only look forward, it can detect all future collisions and modify packet size accordingly. The master packet must be de-conflicted with all piconets before determining the slave packet length since the frequency used by the slave packet is dependent on the length of the master packet. For example, in Figure 7.4 assume that Piconet 1's master and target slave devices have enough data to fill 5-slot packets at each opportunity. With no collision avoidance mechanism, Piconet 1's master attempts to send a 5-slot packet on frequency F_1 . If a master packet header is properly received, the slave will attempt to respond with a 5-slot packet on frequency F_6 . However, if Piconet 2 in the vicinity, the Piconet 1's MP may be disrupted by Piconet 2 in slot 3 while its SP may be disrupted in slot 5 in Figure 7.4a. The collision is not imminent as Piconet 2 may also be using multi-slot packets which cause F_1 and/or F_6 to be skipped in the hop sequence. However, since 5-slot packets are transmitted on a single frequency, the packets are vulnerable to 5 or 6 single-slot packets on different frequencies, greatly increasing the probability that it will be disrupted. If Piconet 2 uses single-slot packet's the collision is guaranteed to occur.

If AFIT is implemented, however, these collisions can be avoided. Since Piconet 1 has no knowledge of Piconet 2's packet size, it must assume the worst case; Piconet 2 is using single slot packets. Thus, the maximum length of the master

H = Header PL = Payload RT = Retune M = Master S = Slave

Piconet 1's hop sequence

Slot 0	Slot 1	Slot 2	Slot 3	Slot 4	Slot 5	Slot 6	Slot 7
H PL RT							
F ₀	F ₁	F ₂	F ₃	F ₄	F ₅	F ₆	F ₇

Piconet 2's hop sequence

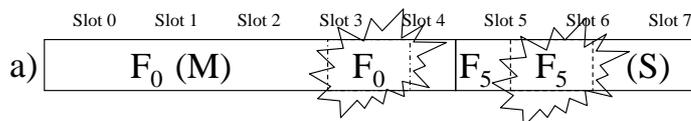
F ₂	F ₄	F ₆	F ₀	F ₁	F ₅	F ₃	F ₇
----------------	----------------	----------------	----------------	----------------	----------------	----------------	----------------

Piconet 3's hop sequence

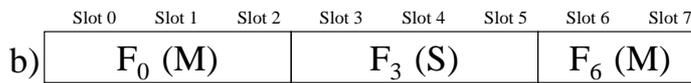
F ₀	F ₃	F ₆	F ₅	F ₇	F ₂	F ₁	F ₄
----------------	----------------	----------------	----------------	----------------	----------------	----------------	----------------

Piconet 4's hop sequence

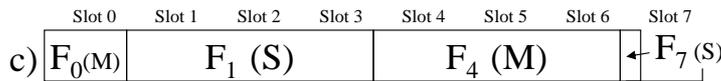
F ₄	F ₁	F ₀	F ₇	F ₆	F ₅	F ₃	F ₂
----------------	----------------	----------------	----------------	----------------	----------------	----------------	----------------



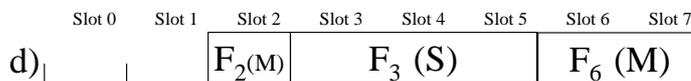
Largest Packets with Piconet 1 not avoiding Piconet 2



Largest Packets with Piconet 1 avoiding Piconet 2



Largest Packets with Piconet 1 avoiding Piconets 2, 3



Largest Packets with Piconet 1 avoiding Piconets 2, 3, 4

Figure 7.4: Maximum packet size using AFIT.

packet without a collision in slot 3 is 3 slots as in Figure 7.4b. The slave packet begins in the fourth slot on F_3 . However, if the slave uses a 5-slot packet, the SP will be disrupted in slot 6. Therefore, the maximum slave packet length is also 3 slots, in which case the next master/slave pair begins in slot 6. In this scenario, Piconet 1 successfully transmits 4,052 user data bits in the eight time slots. Although, not as successful as the 5424 user data bits which would have been transmitted had no interference occurred, it is much better than the baseline in which all data was disrupted.

When a second neighboring piconet, Piconet 3, is introduced in Figure 7.4c, Piconet 1 must consider both piconets when determining master packet length. Piconet 3 may use frequency F_0 beginning in slot 0 which prevents Piconet 1 from transmitting a multi-slot packet. However, since Piconet 3's packets begin during Piconet 1's re-tuning period, Piconet 1's master can send a full single-slot packet. The subsequent slave packet is limited to 3 slots due to a possible collision with Piconet 2 on F_1 in slot 4. The next master packet is then constrained to 3 slots as well due to a possible collision with Piconet 3 on F_4 in slot 7. Finally, the responding slave packet is limited to a POLL packet since Piconet 2 may transmit a packet on F_7 during the Piconet 1's payload portion of the time slot. This allows, 3144 bits of user data to be transmitted in the eight slots.

When Piconet 4 is introduced in Figure 7.4d, Piconet 1's master packet is unaffected. However, the subsequent SH is subject to disruption. Since the master packet size cannot be reduced to change the frequency on which the slave packet is transmitted, there is no reason to attempt to transmit the first master slave pair. Piconet 1 attempts to transmit the next master packet beginning in slot 2. A 3-slot packet is the largest that can be used due to a possible collision with Piconet 3 in slot 5. However, if a 3-slot packet is used, the SH of the subsequent slave is subject to collision with Piconet 4 in slot 5 on F_5 . Thus, a single slot master packet is used. The subsequent slave packet is then transmitted on F_3 rather than F_5 and can successfully use a 3-slot packet. If a 5-slot packet is attempted, a collision in slot 6 with Piconet 2 may occur. Finally, the next master packet can be transmitted on F_6 , allowing 2804 user data bits to be transmitted.

In AFIT, different combinations of packet sizes may increase goodput for one of the devices in the piconet. For example, instead of both the master and slave using 3-slot packets, conditions may allow the master to use a single-slot packet while the slave device transmits a 5-slot packet. The simplest method, however, is to use the maximum master packet size that is both needed and feasible. If the

data available to transmit only fills a 3-slot packet, there is no reason to attempt to transmit a 5-slot packet. The master device must calculate all frequencies used by the neighboring devices during the transmission of the master packet with the maximum desired packet length. If a collision is possible, the master packet length is reduced as to avoid such a collision. If the MH is subject to collision, a master packet should not be transmitted. When the master packet length has been determined, the process must be repeated for the slave packet. If the SH is subject to collision and a multi-slot master packet is planned, the master packet length must be reduced (from 5 to 3 or from 3 to 1 slots, as appropriate). The maximum slave packet length must again be determined. If the planned master packet is single-slot and the SH is still subject to collision, the master packet must not be transmitted.

Since the master must perform the collision avoidance calculations for both the master and slave, it would be ideal to transmit that information to the slave. Unless information containing the maximum packet length the slave may use is transmitted to the slave, it must also compute its maximum packet length upon receipt of the master packet.

A MATLAB[®] simulation was conducted using a single neighboring piconet for all possible packet sizes. Again, saturation was assumed - piconet devices sent full packets of the desired length when possible. Note that PIAM was not modified and is therefore at a disadvantage when multi-slot packets are used. The methods produced the results presented in Table 7.3. The goodput is normalized to the maximum goodput possible with no interference with the specified packet lengths. Also, the resulting improvement that each avoidance method provided over the baseline goodput, with no collision avoidance, is presented. As expected, AFIT achieved the greatest gain when one piconet attempted to transmit 5-slot packets while the other attempted to transmit only single-slot packets since the technique was designed to perform best in just such a worst-case scenario.

Table 7.3: Goodput with multi-slot packets.

Piconet 1 Desired Packet Length (slots)	Piconet 2 Desired Packet Length (slots)	Max Goodput (kbps)	Baseline (%)	PIAM (%)	PIAM Improvement (%)	AFIT (%)	AFIT Improvement (%)
1	1	345.6	97.8	98.6	0.88	99.0	1.25
3	1	672.0	80.5	80.9	0.48	83.1	3.24
5	1	780.8	73.2	73.7	0.62	77.1	5.36
3	3	780.8	97.3	97.2	-0.91	99.2	2.00
3	5	835.2	95.6	95.4	-0.20	98.0	2.55
5	5	867.8	97.3	97.2	-0.03	99.4	2.17

Table 7.4: Expected efficiency with multi-slot packets.

Piconet 1 Desired Packet Length (slots)	Piconet 2 Desired Packet Length (slots)	Max efficiency (%)	Normalized Baseline (%)	Normalized AFIT (%)	AFIT Improvement (%)
1	1	59.0	98.1	99.9	1.88
3	1	77.6	96.4	99.8	3.58
5	1	80.7	94.5	99.9	5.69
3	3	90.3	97.5	99.7	2.30
3	5	94.0	95.4	98.0	2.84
5	5	94.5	97.4	99.4	2.46

Likewise, the largest gains in efficiency were evident when a device attempting 5-slot packets was paired with a device attempting single-slot packets. Since PIAM was not designed for multi-slot packets, its efficiency data is not included in Table 7.4.

7.7 Summary

Although no collision avoidance method has been dictated by the BT standard, it is clear from Chapter VI that one is needed in a scatternet with collocated piconets. Although PIAM is effective for single slot packets, AFIT is more effective at increasing goodput and can be used with multi-slot packets. It conserves energy by not transmitting packets doomed to disruption. Moreover, the goodput gain with multi-slot packets in two piconets can be as high as 5.4%. Although the expected goodput gain with single-slot is a mere 1.3% with two piconets, this grows significantly, rising to 6.5% with 5 neighboring piconets. The derivation of the pmf for goodput using both collision avoidance methods provide accurate knowledge of

the nature of the goodput between arbitrary BT devices as well as giving tools for deriving the goodput pmf for other collision avoidance methods.

Interference from inquiring nodes can also significantly impact performance so similar collision avoidance methods may be useful for inquiring nodes within a scatternet. However, such methods may impact the time needed to discover BT nodes. This is investigated in Chapter VIII.

VIII. Inquiry Time

The Bluetooth discovery process requires use of the inquiry substate which not only consumes significant power, but prevents normal data traffic flow and simultaneously acts as a noise source for neighboring networks. Therefore, the inquiry substate dwell time should be limited to that needed to expect to discover an acceptable percentage of neighboring devices. The BT specification recommends an inquiry substate duration of 10.24 s, but the complex temporal and spectral interactions between two devices make inquiry time estimation difficult. Additionally, collision avoidance methods to reduce the negative impact of inquiring devices on piconet goodput may affect the inquiry time. In this chapter, the complex interactions which may occur between hop sequences in the discovery process are defined and detailed analytical expressions for the probability distribution of the inquiry time for a BT-enabled device that strictly follows v1.1 of the BT standard.

Analysis shows inquiry substate durations of 5.12 and 2.56 seconds will locate 99% of all devices within transmission range using the standard inquiry scan mode and the interlaced inquiry scan, respectively. Likewise, analysis of the standard and interlaced discovery process in v1.2 of the specification shows 99% of neighboring devices can be located in 3.84 s and 1.28 s, respectively. Substantial inquiry time reduction results in reduced power requirements and increased throughput by increasing data traffic and reducing interference with neighboring piconets. The results are compared to existing simulation models and measurement studies. Simulation models are also used to study the impact of a collision avoidance method on the inquiry time distribution. Although the impact is minimal, it is shown that the presence of multiple inquiring nodes significantly impacts the inquiry time.

8.1 Inquiry Interference Probability

The probability distribution for the inquiry time is a function of

1. the time required for a device in the inquiry scan substate to scan for the first time,
2. the number of scans required to receive the first inquiry packet,
3. the duration of the back-off period, and
4. the number of scans required to receive the second packet.

The number of scan windows required to receive an inquiry packet is a function of the relationship between the scan frequency, the frequency trains and the changes in both. In the next section, the analytic expressions for the inquiry time probability distribution are derived.

8.2 Random Variable and Event Definitions

The probability distribution of the inquiry time is the key to selecting the appropriate inquiry substate duration. When a master enters the inquiry substate (denoted $t = 0$), the time until a scanning device *re-enters* the scan substate and begins a 11.25 ms scan window is uniformly distributed on $(0, 1.28)$ s. This random variable is T_R . Note that an inquiry packet may have been received prior to T_R if $T_R > 1.26875$ as shown in Figure 8.1. Since a device in the inquiry scan substate is assumed to be scanning before $t = 0$, a scan window begins prior to, but yet overlaps, $t = 0$ in such a case. Thus, an inquiry packet may be received prior to T_R . This will be significant in developing the equation for the first received inquiry packet. If the scanning frequency is in the inquiry train, the scanning device receives

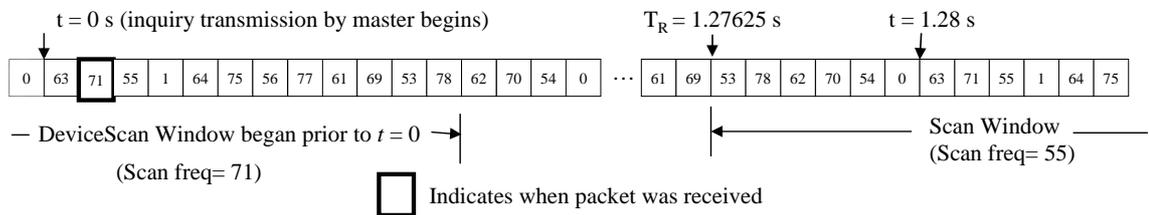


Figure 8.1: First inquiry packet is received prior to T_R .

an initial inquiry packet in a time distributed on $(0, 11.25)$ ms, depending on the position of the scan frequency within the train once the scan window begins. Let this random variable be T_1 . Since the temporal alignment between the inquiring and scanning devices is uniform on $(0, 1250)$ μs , T_1 is also uniformly distributed. If the scanning frequency is not in the train, the scanning device drops out of scan mode but returns exactly 1.28 s after the beginning of the previous scan window using a different scan frequency. Let the random variable N_1 be the number of unsuccessful scan attempts before receipt of the first inquiry packet. Each time the scanning device returns to scan mode, the trains will have either swapped a frequency, or the inquiring device may have switched the train on which it is transmitting. Let T_F be the time the first inquiry packet is received by the scanning device. One might expect $T_F = T_R + T_1 + 1.28N_1$. However, recall that packets may be received prior to T_R if $T_R > 1.26875$ s. Thus, the probability density for T_F is not a simple convolution of the probability densities of T_R, T_1 , and $1.28N_1$. The probability density for T_F is developed in Section 8.3.

Once an initial packet has been received at T_F , the scanning device drops out of the inquiry scan substate for a time T_B , a discrete random variable whose samples space consists of 1024 values spaced every 625 μs between $t = 0$ and $t = 639.375$ ms. After the back-off time has elapsed, the inquiry is completed in time T_2 discretely distributed on $(0, 11.25)$ ms if the scanning frequency is in the current train. The random variable T_2 is not uniformly distributed due to the transmission pattern of an inquiring device. Furthermore, due to T_B and the possibility of scan frequency and train changes, T_2 may not be equivalent to T_1 as is sometimes assumed [SBT00] [SBT01] [KaL01]. Moreover, the scanning frequency may not be in the correct train after the back-off. Thus, the random variable N_2 is defined as the number of unsuccessful attempts before receipt of the second inquiry packet after the back-off time has elapsed.

Since T_2 is not uniformly distributed, a probability density function for the inquiry time including T_2 is complex. Assuming $T_2 = 0$ simplifies the model with little loss in accuracy. The maximum error induced by this simplification is 11.25 ms. Since the system has a mean inquiry time on the order of seconds, this is considered negligible.

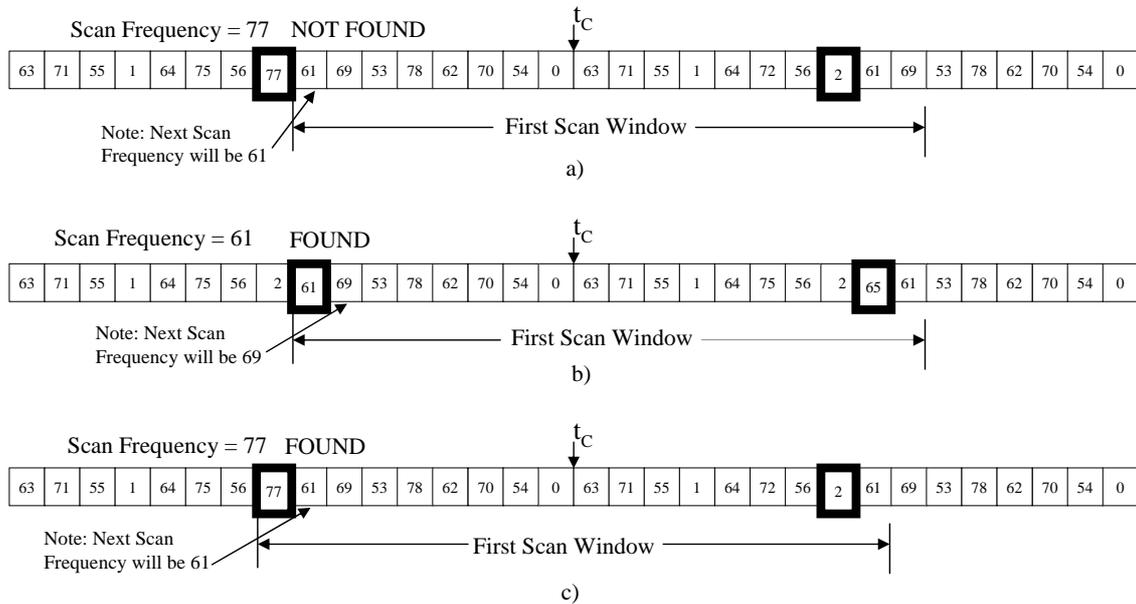
The inquiry time, T_I , then, is the sum

$$T_I = T_F + T_B + 1.28N_2. \quad (8.1)$$

The relationship between the inquiry trains and the scan frequency determine the distribution for the random variables T_F , T_B , and N_2 . Thus, several events are used in deriving the probability density for T_I . These events are:

E_M : the first scan frequency is in the current inquiry train when the first scan window begins. For example, in Figure 8.2a, E_M occurs since the scan frequency is 77; 77 is in the train when the scan window begins. Note that even though 77 is in the train when the scan window begins, a membership change in the train during the scan window prevents 77 from being used by the inquiring device within the scan window. On the other hand, in Figures 8.2b and c, E_M also occurs because 61, and 77 are in the inquiry trains when the scan window begins and are used.

E_B : the membership of the train changes (i.e., swaps one frequency with the other train) during the scan window which takes place *before* the back-off period (i.e., every 1.28 s until the first packet is received). Since both the scan windows and frequency changes occur every 1.28 s, E_B will occur in all scan windows until the first inquiry packet is received if E_B occurs in the first scan window.



□ Frequency changing membership $t_c =$ Start of train with changed membership

Figure 8.2: Train membership change during a scan window.

E_A : the membership of the train changes (i.e., swaps one frequency with the other train) during the scan window which takes place *after* the back-off period, T_B . Since both the scan windows and frequency changes occur every 1.28 s, E_A will occur in all scan windows until the second inquiry packet is received if E_A occurs in the first scan window after the back-off period.

E_L : the scan frequency is the frequency which will swap trains at the next membership change. Since the change in the train membership follows the same pattern as the selection of the scan frequency, this event continues throughout the entire inquiry process. For example, suppose frequencies 77 and 2 in Figure 8.2a are the two frequencies which switch trains at the next membership change. Note that in Figure 8.2, only the membership of the trains change, not the train being used by the inquiring device. The subsequent scan frequency,

and thus the next frequency to leave the train, is 61. By examining 8.2b, it can be determined that 61 is swapped with 65; the position once held by 61 in Figure 8.2a now contains 65, just as 77 was replaced by 2 in Figure 8.2a.

E_Y : the first scan window overlaps $t = 1.28$ s (i.e., $T_R > 1.26875$ s). This means a scan window overlapped $t = 0$ and will for integer multiples of $t = 1.28$ s.

E_P : the scan frequency used at $t = 0$ s is the frequency in the train just prior to $t = 0$ s. For example, in Figure 8.3a-c, the scan frequency is 0, which is the frequency in the train just before the inquiry packets are transmitted at $t = 0$ s. This only occurs if E_M occurs since, otherwise, the scan frequency is not in the train at $t = 0$ s.

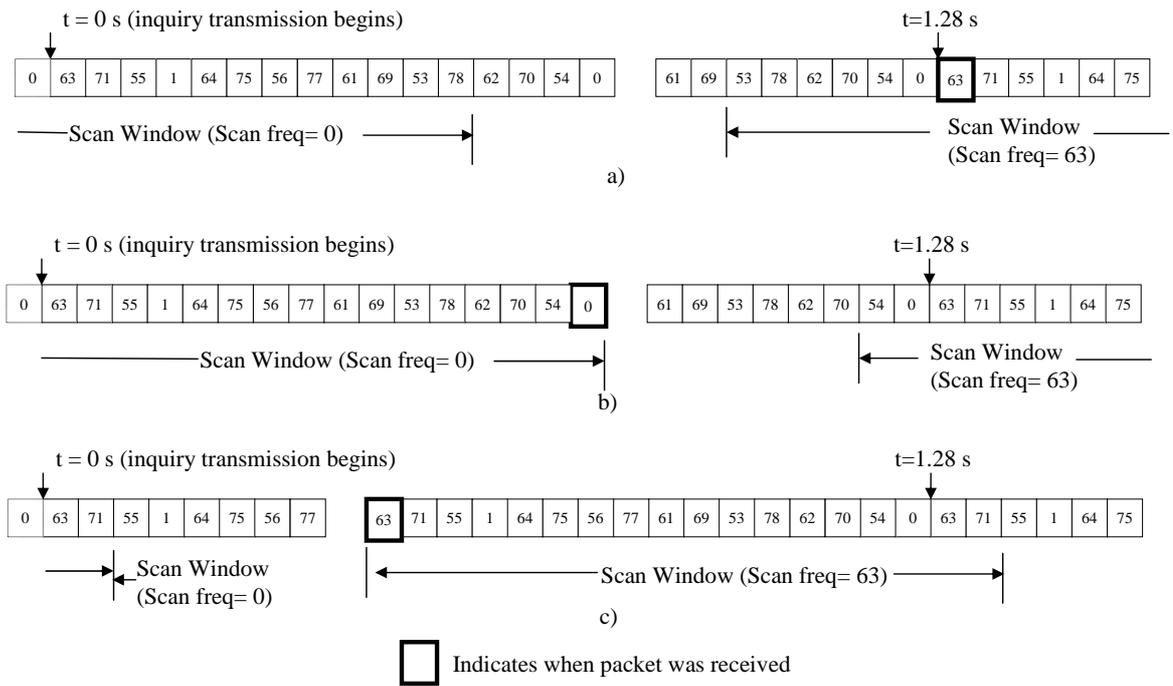


Figure 8.3: Events E_P and E_Y occur and a) the scan window begins between $t = -10$ ms and $t = -1.25$ ms so the first packet is received after 1.28 s, b) the scan window begins between $t = -1.25$ ms and 0 s, c) the scan window begins between $t = -11.25$ ms and -10 ms.

E_K : the scan frequency at $t = 2.56$ s is the frequency in the train used at $t = 2.56$ s just prior to $t = 2.56$ s. This can only occur when $\overline{E_M}$ occurs.

$E_{S\overline{T}}$: the scan frequency changes but the train does not change membership during the back-off period. It is possible for the scan frequency to effectively change trains when this occurs. Such a case is shown in Figure 8.4a. Had membership in the train also changed during the back-off period, the scan frequency would have remained in the same train used when receiving the first packet.

$E_{\overline{S}T}$: the train membership changes but the scan frequency does not change during the back-off time. Again, this may effectively result in a train change as shown in Figure 8.4b.

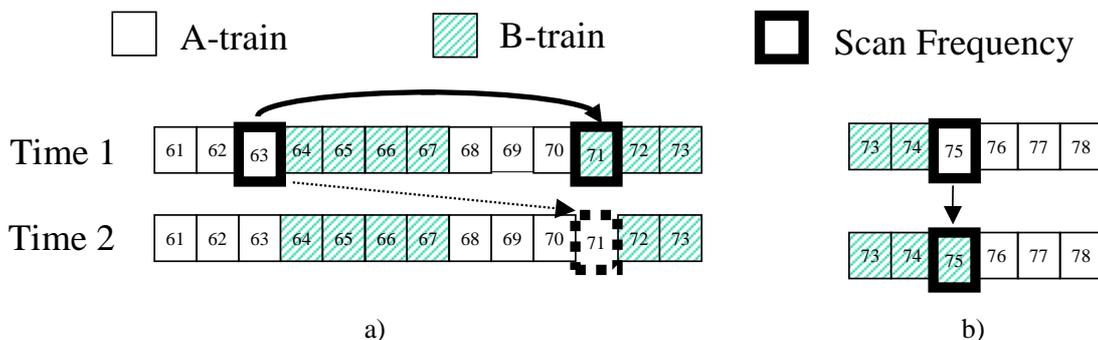


Figure 8.4: Possible events during back-off a) $E_{S\overline{T}}$, when the scan frequency changes but train membership does not and b) $E_{\overline{S}T}$, when train membership changes but the scan frequency does not.

E_{ST} : the train membership and the scan frequency both change during the back-off period.

$E_{\overline{S}\overline{T}}$: neither the train membership changes nor the scan frequency changes during back-off.

E_W : the back-off period overlaps a time (t) that is an integer multiple of 1.28 s.

E_{Hi} : the first inquiry packet is received in the i th 1.28 s interval of the inquiring process, where $i \in \{1, 2, 3, 4, 5\}$. Since the first packet can only be received in one interval, E_{Hi} can only occur for one value of i , $i \in \{1, 2, 3, 4, 5\}$. The occurrence of E_{Hi} is dependent on E_M , E_B , E_L , E_Y , and E_P .

The events E_M , E_B , E_L , and E_Y are mutually independent. Events E_P and E_K are mutually exclusive and depend on E_M . The event E_W depends on E_Y . Events $E_{S\bar{T}}$, $E_{\bar{S}T}$, E_{ST} and $E_{\bar{S}\bar{T}}$ are mutually exclusive but depend on E_B . The event E_A is mutually exclusive of E_B , $E_{\bar{S}T}$ and E_{ST} and therefore depends on $E_{S\bar{T}}$ and $E_{\bar{S}\bar{T}}$.

If E_M occurs, the second scan frequency is in the train being transmitted when the second scan window begins, assuming the first packet was not received in the first scan window. However, due to the train change at $t = 2.56$ s, the third and fourth scan frequencies are not in the train being transmitted when the third and fourth scan window begins, assuming the first packet was not received in the first or second scan window. At $t = 5.12$ s, the train switches again and the fifth and sixth scan frequencies are again in the train used.

The marginal probabilities of events E_M , E_B , E_L , and E_Y are

$$P(E_M) = 0.5, \tag{8.2}$$

$$P(E_B) = P(E_Y) = \frac{11.25 \times 10^{-3}}{1.28} = 0.0088, \text{ and} \tag{8.3}$$

$$P(E_L) = 1/16 = 0.0625. \tag{8.4}$$

The conditional probabilities of E_P and E_K are

$$P(E_P|E_M) = P(E_K|\bar{E}_M) = 1/16 = 0.0625 \text{ and} \tag{8.5}$$

$$P(E_P|\bar{E}_M) = P(E_K|E_M) = 0. \tag{8.6}$$

Due to the large number of events that may occur in the interaction of the train and the scanning frequency, it is cumbersome to derive the probability density function of the inquiry time by conditioning on all combinations of events. The probability density function may be more simply derived by conditioning on the events E_{Hi} , $i = 1, 2, 3, 4, 5$. The probability density function of T_I can be written by using the law of total probability as

$$f_{T_I}(t) = \sum_{i=1}^5 f_{T_I}(t|E_{Hi})P(E_{Hi}). \quad (8.7)$$

The probability distribution of $T_I|E_{Hi}$ is dependent on the density of the random variable $T_F|E_{Hi}$. Its density is derived in Section 8.3 along with the marginal probabilities $P(E_{Hi})$, $i = 1, 2, 3, 4, 5$ as the first step in determining the probability density function of (8.7).

8.3 Components of Inquiry Time pdf

If the first scan frequency is in the train being transmitted when the first 11.25 ms scan window begins (i.e., E_M occurs), the inquiry packet will be received in the first 1.28 s interval in most cases (i.e., E_{H1} will occur). An exception occurs when $E_P \cap E_Y \cap E_M$. For example, in Figure 8.3a the scan frequency is 0 with 63 being the first transmitted frequency and the scan window overlaps $t = 0$. If E_P , E_Y , and E_M jointly occur, the inquiry packet will only be received after 1.28s if the scan window begins between -10 ms and -1.25 ms rather than the -11.25 ms to 0 s range for which E_Y occurs. This is due to the length of the scan window being longer than the train. As seen in Figures 8.3b-c, the repetition of the first two frequencies allows packets at those frequencies to be received prior to 1.28s. Thus,

$$P(\overline{E_{H1}}|E_Y \cap E_P \cap E_M) = 14/18. \quad (8.8)$$

If $E_B \cap E_L$ occurs, the scan frequency can effectively change trains. If the inquiry train changes membership during the 11.25 ms of a scan window, a frequency which was in the train being transmitted when the scan began may be missed. For example, in Figure 8.2a, E_L occurs since the scan frequency is 77. In the next membership change, frequency 77 is replaced with 2, causing the inquiry packet to be missed. The position of the scan frequency in the train determines whether the frequency will be missed. Half of the possible positions are expected to cause an effective train change. In Figure 8.2b and c the position of the scan frequency is such that the packet is still received. Likewise, effective train changes allow E_{H1} to occur even if $\overline{E_M}$ occurs. For example, in Figure 8.2a, the packet can be received if the scan frequency is 2, even though 2 is not in the train when the scan window begins. Therefore,

$$\begin{aligned}
P(E_{H1}|E_B \cap E_L \cap E_M) &= P(\overline{E_{H1}}|E_B \cap E_L \cap E_M) = P(E_{H1}|E_B \cap E_L \cap \overline{E_M}) \\
&= P(\overline{E_{H1}}|E_B \cap E_L \cap \overline{E_M}) = 0.5
\end{aligned} \tag{8.9}$$

and the probability the scanning device will receive a packet in the first 1.28 s interval is

$$\begin{aligned}
P(E_{H1}) &= P(E_M) - P(\overline{E_{H1}}|E_B \cap E_L \cap E_M)P(E_B)P(E_L)P(E_M) + \\
&\quad P(E_{H1}|E_B \cap E_L \cap \overline{E_M})P(E_B)P(E_L)P(\overline{E_M}) - \\
&\quad P(\overline{E_{H1}}|E_Y \cap E_P \cap E_M)P(E_P)P(E_Y)P(E_M) \\
&= 0.5 - 0.000137 + 0.000137 - .0002136 = 0.4997864.
\end{aligned} \tag{8.10}$$

Note that (8.10) not only includes the case when the train's membership changes during the scan window and causes the packet to be missed, but also the case when the scan frequency is not initially in the train and the membership change allows the packet to be received.

The conditional distribution of $T_F|E_{H1}$ is approximately continuously uniform. Recall that T_R is distributed uniformly on $(0, 1.28)$ s, and T_1 distributed uniformly on $(0, 11.25)$ ms. When E_{H1} occurs, the time the first packet is received is the sum of the two random variables T_R and T_1 as shown in Figure 8.5a. However, the distribution fails to account for the case when E_Y occurs. For example, if the scan window begins at $t = 1.275$ s, a scan window had also begun at $t = -5$ ms and extended to $t = 6.25$ ms. Thus, the probability measure (in the density sense) extending past $t = 1.28$ s represents packets which would be received between $t = 0$ and 11.25 ms, returning the distribution to approximately $U(0,1.28)$ s as shown in Figure 8.5b. The distribution of $T_F|E_{H1}$ is approximately uniform since 0.042% of the packets are not received until after 1.28 s when $E_P \cap E_Y$ occurs. Therefore, assuming the distribution is uniform results in an error of up to 0.042% in the unconditional distribution of $T_F|E_{H1}$ until $t = 1.28$ s.

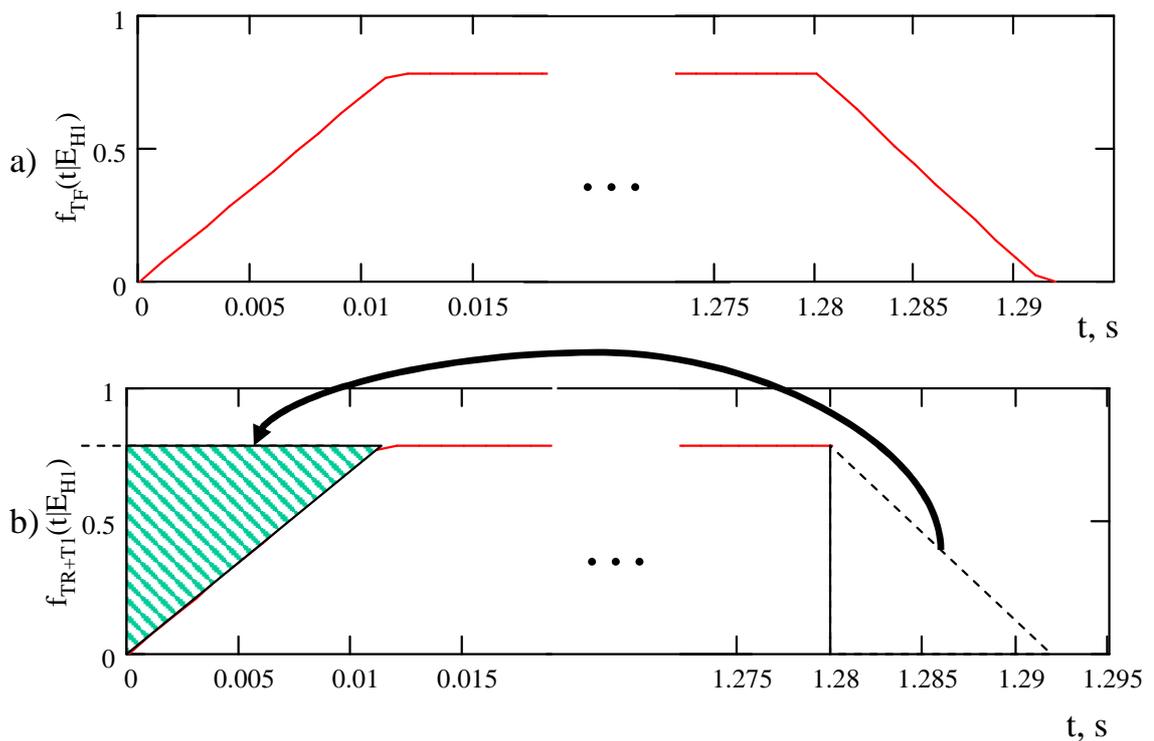


Figure 8.5: Probability density function of a) $T_R * T_1$ and b) $T_F|E_{H1}$.

The probability a device will receive its first inquiry packet in the second 1.28 s is limited to three cases: i) $E_B \cap E_L \cap E_M$ occurs and the packet is not received on the first scan as shown in Figure 8.2a, but is received on the second due to the position of the scan frequency in the train as in Figure 8.2b or c, ii) $E_B \cap E_L \cap \overline{E_M}$ occurs and the packet is not received on the first scan but is on the second, or iii) $\overline{E_{H1}} \cap E_Y \cap E_P \cap E_M$. Since the first two and the latter occurrences result in different distributions of T_F , they are treated as two mutually exclusive occurrences of E_{H2} : E_{H2a} and E_{H2b} , respectively, where E_{H2} is the event that E_{H2a} or E_{H2b} occur. Thus,

$$\begin{aligned} P(E_{H2a}) &= P(E_{H2a}|E_B \cap E_L \cap E_M)P(E_B)P(E_L)P(E_M) + & (8.11) \\ &P(E_{H2a}|E_B \cap E_L \cap \overline{E_M})P(E_B)P(E_L)P(\overline{E_M}) = 6.866 \times 10^{-5} \end{aligned}$$

where

$$P(E_{H2a}|E_B \cap E_L \cap \overline{E_M}) = P(E_{H2a}|E_B \cap E_L \cap E_M) = 1/8 \quad (8.12)$$

since this event only occurs when the first scan frequency falls outside the scan window as shown in Figure 8.2a, and the second falls within the scan window as shown in Figure 8.2b. When E_{H2a} occurs, T_F is uniformly continuous between 1.28 and 2.56 s. Note that neither the event $E_Y \cap E_P \cap E_L \cap E_B \cap \overline{E_M}$ or $E_Y \cap E_P \cap E_L \cap E_B \cap E_M$ has been addressed. Both events are possible, but the probability of each is only 1.88×10^{-6} and are therefore considered to be negligible. The probability that E_{H2b} occurs is

$$P(E_{H2b}) = P(\overline{E_{H1}}|E_Y \cap E_P \cap E_M)P(E_Y)P(E_P)P(E_M) = 0.000214. \quad (8.13)$$

If E_{H2b} occurs, T_F is uniformly distributed between 1.28 and 1.2809375 s but will be considered a point mass at $t = 1.28$ s for simplicity.

The event E_{H3} will generally occur when the scan frequency belongs in the train which is first used at $t = 2.56$ s, i.e., if $\overline{E_M}$ occurs. Exceptions to this include the cases where E_B and E_L occur and the packet can be received before the train change when $\overline{E_M}$ occurs or causes the first two scans to miss the packet when E_M occurs.

As before, if E_P and E_Y occur, the packet may not be received until shortly past $t = 3.84$ s and $P(\overline{E_{H3}}|E_Y \cap E_K \cap \overline{E_M}) = 14/18$ just as $P(\overline{E_{H1}}|E_Y \cap E_P \cap E_M) = 14/18$. Additionally, if $E_B \cap E_L \cap E_M$ occurs, E_{H3} occurs for the remaining positions of the scan frequencies in the train except in a special case. It is possible that E_B , E_L , and E_M occur and the position of the scan frequency within the train does not allow the packet to be received in the first two attempts but would have on the third attempt if the train doesn't change. In such a case, the train change dictates that the packet is not received until the train changes again at 5.12 s. Thus,

$$\begin{aligned}
P(E_{H3}) = & P(\overline{E_M}) - P(E_{H1} \cap E_B \cap E_L \cap \overline{E_M}) - P(E_{H2a} \cap E_B \cap E_L \cap \overline{E_M}) \\
& - (P(\overline{E_{H3}}|E_Y \cap E_P \cap \overline{E_M}) \times P(E_Y)P(E_P)P(\overline{E_M})) \\
& + P(E_{H3}|E_B \cap E_L \cap E_M)P(E_B)P(E_L)P(E_M).
\end{aligned} \tag{8.14}$$

From (8.10),

$$P(E_{H1} \cap E_B \cap E_L \cap \overline{E_M}) = 0.000137$$

and from (8.11)

$$P(E_{H2a} \cap E_B \cap E_L \cap \overline{E_M}) = 3.433 \times 10^{-5}.$$

Of the sixteen possible positions of the scan frequency within the train when $E_B \cap E_L \cap E_M$ occur, eight will cause E_{H1} to occur, one will cause E_{H2a} to occur, and one will cause E_{H5} to occur. Thus $P(E_{H3}|E_B \cap E_L \cap E_M) = 6/16$ and $P(E_{H3}) =$

0.4997348. Using the same reasoning as with E_{H1} and shown in Figure 8.5, the distribution time for T_F is uniformly distributed between 2.56 and 3.84 s.

The event E_{H4} only occurs when $E_Y \cap E_K \cap \overline{E_M}$ occurs. Similar to E_{H2b} occurring, this happens when the inquiry packet is in the train used at $t = 2.56$ s but the scan window overlaps $t = 2.56$ s and the packet using the scan frequency would have been transmitted just before $t = 2.56$ s had the train already changed. Due to the scan frequency change between scan windows as in Figure 8.3, the inquiry packet is not received until immediately after $t = 3.84$ s. Thus,

$$P(E_{H4}) = P(\overline{E_{H3}}|E_Y \cap E_P \cap \overline{E_M})P(E_Y)P(E_P)P(\overline{E_M}) = 0.000214. \quad (8.15)$$

and the distribution for the time at which the first inquiry packet is received is considered to be a point mass at 3.84 s.

Finally, $T_F|E_{H5}$ is uniformly continuous on (5.12, 6.4) s and only occurs when $E_B \cap E_L \cap E_M$ occurs. It occurs when the packet is missed in the first two scans but would have been received in the third scan had the train not changed. Since $P(E_{H5}|E_B \cap E_L \cap E_M) = 1/16$,

$$P(E_{H5}) = P(E_{H5} \cap E_B \cap E_L \cap E_M) = 1.717 \times 10^{-5}. \quad (8.16)$$

Once the first packet is received, the inquiring device leaves the inquiry scan substate for a random duration, T_B using v1.1 of the specification. In Section 8.4, T_B is added to the conditional probability densities for T_F to determine the density for the time at which the inquiring device leave re-enters the inquiry scan substate after the back-off period.

8.4 *Distribution of Back-off Period Completion time, T_P*

Recall that T_B is discretely uniform between 0 and 639.375 ms and its sample space contains 1024 points. The pdf for the completion time of the back-off period, T_P , is the convolution of the pdfs for T_B and T_F since they are independent random variables.

Since the probability density for T_F is dependent on E_{Hi} , the density for T_P must be conditioned on E_{Hi} . Since T_B is independent of E_{Hi} , the pdf for T_P generally is

$$f_{T_P}(t|E_{Hi}) = f_{T_B}(t) * f_{T_F}(t|E_{Hi}) \quad (8.17)$$

where $*$ denotes the convolution operator. However, the inquiry time is dependent on the events $E_{S\bar{T}}$, $E_{\bar{S}T}$, E_{ST} , and $E_{\bar{S}\bar{T}}$ which are dependent on T_B . As T_B increases, the probability of events $E_{S\bar{T}}$, $E_{\bar{S}T}$, or E_{ST} increases. Therefore, f_{T_P} is conditioned on these events as well such that

$$\begin{aligned} f_{T_P}(t|E_{Hi}) = & f_{T_P}(t|E_{Hi} \cap E_{S\bar{T}})P(E_{S\bar{T}}|E_{Hi}) + f_{T_P}(t|E_{Hi} \cap E_{\bar{S}T})P(E_{\bar{S}T}|E_{Hi}) + \\ & f_{T_P}(t|E_{Hi} \cap E_{ST})P(E_{ST}|E_{Hi})f_{T_P}(t|E_{Hi} \cap E_{\bar{S}\bar{T}})P(E_{\bar{S}\bar{T}}|E_{Hi}). \end{aligned} \quad (8.18)$$

Recall that since the scan frequency is based on the free running counter of the scanning device, the time until a change in scan frequency is distributed continuously uniform on $[0, 1.28)$ s and is denoted S_S . This time is independent of the beginning of the scan window, and represents the frequency generating subsystem changing the scan frequency to be used when the scan window begins.

Likewise, the time until the next membership change in the train, denoted S_T , is distributed uniformly on $[0, 1.28)$ s except when there is knowledge of the change. If E_B occurs, a membership change occurs just before the first inquiry packet is received, the next change will occur 1.28 s later. Since the maximum back-off time

is 639.375 ms, the membership cannot change during the back-off time when E_B occurs. Thus, the cumulative distribution functions (CDF) of S_S and S_T , $F_S(t)$ and $F_T(t|\overline{E_B})$ respectively, are

$$F_S(t) = F_T(t|\overline{E_B}) = \begin{cases} 0 & t < 0 \\ \frac{t}{1.28s} & 0 \leq t \leq 1.28s \\ 1 & otherwise \end{cases} \quad (8.19)$$

The conditional probabilities in (8.18) are

$$\begin{aligned} P(E_{S\overline{T}}|E_{Hi}) &= P(\{S_S \leq T_B\} \cap \{S_T > T_B\}|E_{Hi}), \\ P(E_{\overline{S}T}|E_{Hi}) &= P(\{S_S > T_B\} \cap \{S_T \leq T_B\}|E_{Hi}), \\ P(E_{ST}|E_{Hi}) &= P(\{S_S \leq T_B\} \cap \{S_T \leq T_B\}|E_{Hi}), \text{ and} \\ P(E_{\overline{S}\overline{T}}|E_{Hi}) &= P(\{S_S > T_B\} \cap \{S_T > T_B\}|E_{Hi}). \end{aligned}$$

The mutual independence of S_S and S_T implies that

$$P(E_{S\overline{T}}|E_{Hi}) = \sum_{n=0}^{1023} F_S(nT_{slot}) (1 - F_T(nT_{slot}|\overline{E_B} \cap E_{Hi})) P(\overline{E_B}|E_{Hi}) \frac{1}{1024}. \quad (8.20)$$

Similarly,

$$P(E_{\overline{S}T}|E_{Hi}) = \sum_{n=0}^{1023} (1 - F_S(nT_{slot})) F_T(nT_{slot}|\overline{E_B} \cap E_{Hi}) P(\overline{E_B}|E_{Hi}) \frac{1}{1024}, \quad (8.21)$$

$$P(E_{ST}|E_{Hi}) = \sum_{n=0}^{1023} F_S(nT_{slot}) F_T(nT_{slot}|\overline{E_B} \cap E_{Hi}) P(\overline{E_B}|E_{Hi}) \frac{1}{1024}, \quad (8.22)$$

and

$$P(E_{\overline{ST}}|E_{Hi}) = \sum_{n=0}^{1023} \left(1 - F_S(nT_{slot})\right) \left(1 - F_T(nT_{slot}|\overline{E_B} \cap E_{Hi})P(\overline{E_B}|E_{Hi})\right) \frac{1}{1024}. \quad (8.23)$$

To derive the pdf for $(T_P|E_{\overline{ST}} \cap E_{Hi})$, $f_{T_P}(t|E_{\overline{ST}} \cap E_{Hi})$ is convolved with $f_{T_B}(t|E_{\overline{ST}} \cap E_{Hi})$. Since $E_{\overline{ST}}$ is independent of T_F , $f_{T_F}(t|E_{\overline{ST}} \cap E_{Hi}) = f_{T_F}(t|E_{Hi})$. Therefore,

$$\begin{aligned} f_{T_B}(t|E_{\overline{ST}} \cap E_{Hi}) &= \frac{\frac{d}{dt}P(\{S_S \leq T_B\} \cap \{S_T > T_B\} \cap E_{\overline{ST}}|E_{Hi})}{P(E_{\overline{ST}}|E_{Hi})} \\ &= \frac{\delta(t - nT_{slot})}{1024} \left(\frac{F_S(nT_{slot}) (1 - F_T(nT_{slot})P(\overline{E_B}|E_{Hi}))}{P(E_{\overline{ST}}|E_{Hi})} \right) \end{aligned} \quad (8.24)$$

for $n = 0, 1, 2, \dots, 1023$ where $\delta(t)$ is the impulse function and

$$\begin{aligned} f_{T_P}(t|E_{\overline{ST}} \cap E_{Hi}) &= \\ &\begin{cases} \frac{1}{1.28} \int_0^t f_{T_B}(\tau|E_{\overline{ST}} \cap E_{Hi})d\tau & 0 < t \leq 639.375ms \\ \frac{1}{1.28} & 639.375ms < t \leq 1.28s \\ \frac{1}{1.28} \left(1 - \int_0^{t-1.28} f_{T_B}(\tau|E_{\overline{ST}} \cap E_{Hi})d\tau\right) & 1.28 < t \leq 1.9139375s \end{cases} \\ &= f_{T_F}(t|E_{Hi}) * f_{T_B}(t|E_{\overline{ST}} \cap E_{Hi}). \end{aligned} \quad (8.25)$$

Likewise,

$$\begin{aligned} f_{T_B}(t|E_{\overline{ST}} \cap E_{Hi}) &= \\ &\frac{\delta(t - nT_{slot}) (1 - F_S(nT_{slot})) F_T(nT_{slot}) P(\overline{E_B}|E_{Hi})}{1024 P(E_{\overline{ST}}|E_{Hi})}, \end{aligned} \quad (8.26)$$

$$f_{T_B}(t|E_{ST} \cap E_{H_i}) = \frac{\delta(t - nT_{slot}) F_S(nT_{slot}) F_T(nT_{slot}) P(\overline{E_B}|E_{H_i})}{1024 P(E_{ST}|E_{H_i})}, \quad (8.27)$$

and

$$f_{T_B}(t|E_{\overline{ST}} \cap E_{H_i}) = \frac{\delta(t - nT_{slot}) \left(1 - F_S(nT_{slot})\right) \left(1 - F_T(nT_{slot}) P(\overline{E_B}|E_{H_i})\right)}{1024 P(E_{\overline{ST}}|E_{H_i})} \quad (8.28)$$

for $n = 0, 1, 2, \dots, 1023$.

Since $f_{T_F}(t|E_{H_i})$ for $i \in \{2a, 3, 5\}$ are time shifted versions of $f_{T_F}(t|E_{H_1})$ (i.e., uniform, continuous, and spanning 1.28s), $f_{T_P}(t|E_{\overline{ST}} \cap E_{H_i})$ for $i = 2a, 3, 5$ can then be derived by shifting by $1.28k$ s and substituting the applicable $P(\overline{E_B}|E_{H_i})$ and $P(E_{\overline{ST}}|E_{H_i})$ where $k = 1, 2, 4$ for $i = 2a, 3, 5$, respectively. Likewise, $f_{T_P}(t|E_{\overline{ST}} \cap E_{H_i})$, $f_{T_P}(t|E_{ST} \cap E_{H_i})$, and $f_{T_P}(t|E_{\overline{ST}} \cap E_{H_i})$ for $i \in \{1, 2a, 3, 5\}$ are determined by replacing $E_{\overline{ST}}$ in (8.25) with $E_{\overline{ST}}$, E_{ST} , and $E_{\overline{ST}}$, respectively. For example,

$$f_{T_P}(t|E_{\overline{ST}} \cap E_{H_5}) = \begin{cases} \frac{1}{1.28} \int_0^{t-5.12s} f_{T_B}(\tau|E_{\overline{ST}} \cap E_{H_5}) d\tau & 5.12 < t \leq 5.759375s \\ \frac{1}{1.28} & 5.759375 < t \leq 6.4s \\ \frac{1}{1.28} \left(1 - \int_0^{t-5.12} f_{T_B}(\tau|E_{\overline{ST}} \cap E_{H_5}) d\tau\right) & 6.4 < t \leq 7.039375 s \end{cases} \\ = f_{T_F}(t - 5.12|E_{H_1}) * f_{T_B}(t|E_{\overline{ST}} \cap E_{H_5}). \quad (8.29)$$

These conditional densities, as well as $f_{T_P}(t|E_{\overline{ST}} \cap E_{H_i})$, are depicted in Figure 8.6 for $i = 1$. Note that the likelihood of a change in scan frequency, train membership, or both, occurs during the back-off period increases as the back-off period increases.

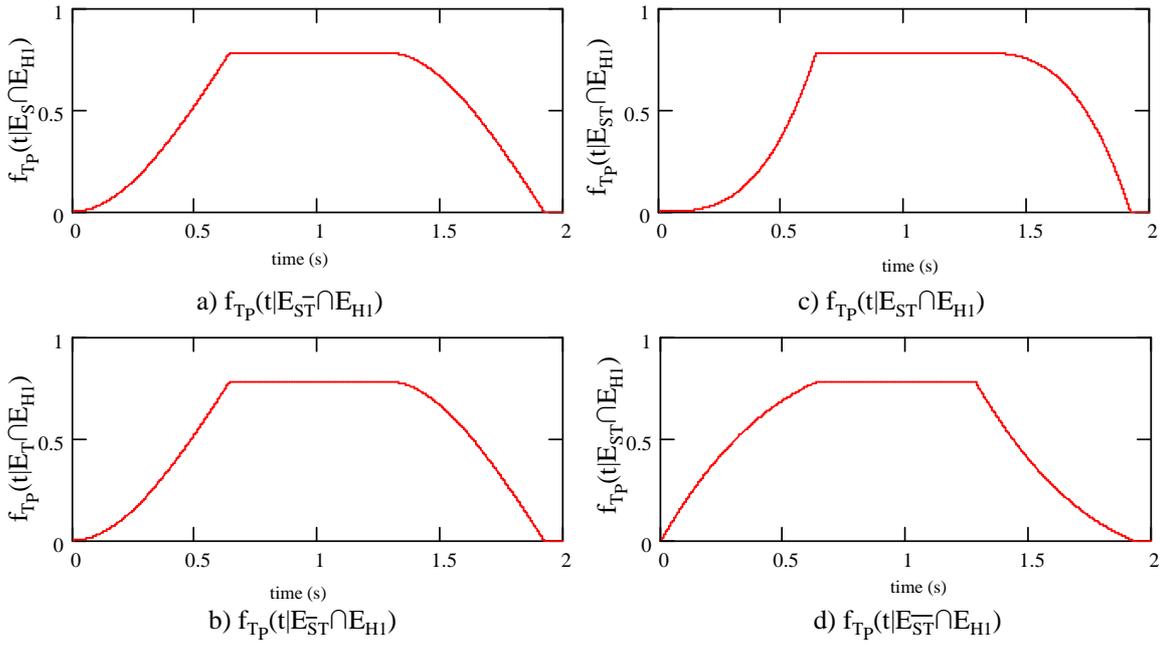


Figure 8.6: Conditional probability density functions for T_P .

When E_{H2b} or E_{H4} occur, the conditional pdf for T_F is treated as a point mass. Since the conditional pdf for T_P is the convolution of the conditional density for T_B with this point mass, the resultant density is $f_{T_B|E_{H_i}}$, $i = 2b, 4$ respectively, shifted to the value of t at which the point mass occurs.

8.5 Conditional Inquiry Time Probability Densities

Combining (8.1) and (8.17),

$$T_I = T_P + 1.28N_2. \quad (8.30)$$

The number of unsuccessful scan frequencies attempted after the back-off time elapses and before receipt of the second inquiry packet, N_2 , is a function of T_P as well as the events $E_{S\bar{T}}$, $E_{\bar{S}T}$, E_{ST} , and $E_{\bar{S}\bar{T}}$. Therefore, the pdf is not derived

by convolving the pdfs of T_P and N_2 . Conditional arguments must be used showing that

$$f_{T_I}(t) = \sum_{i=1}^5 \sum_{j=0}^4 f_{T_I}(t|E_{H_i} \cap \{N_2 = j\})P(\{N_2 = j\}|E_{H_i})P(E_{H_i}). \quad (8.31)$$

Recall that E_{H_2} is the combination of $E_{H_{2a}}$ or $E_{H_{2b}}$ occurring. To derive the needed conditional pdfs, the probability of the events E_A and E_W must be derived. The back-off time can only overlap an integer multiple of 1.28 s if T_F is within 639.375 ms of the next integer multiple of 1.28 s. For example, if $E_{H_{2b}}$ occurs, $P(E_W|E_{H_{2b}}) = 0$ since T_F is modelled as occurring at $t = 1.28$ s and the maximum T_B is 639.375 s. Therefore T_P cannot be greater than 2.56 s when $E_{H_{2b}}$ occurs, thus

$$P(E_W|E_{H_i} \cap E_X) = \begin{cases} \int_{1.28k}^{0.639375+1.28k} f_{T_P}(\tau|E_X \cap E_{H_i})d\tau & i \in \{1, 2a, 3, 5\} \\ 0 & i \in \{2b, 4\} \end{cases} \quad (8.32)$$

where $E_X \in \{E_{S\bar{T}}, E_{\bar{S}T}, E_{ST}, E_{\bar{S}\bar{T}}\}$ and $k = 1, 2, 3, 5$ for $i = 1, 2a, 3, 5$, respectively.

Recall that E_A is the event that the train changes membership during the scan windows which take place after the back-off period. The event E_A is mutually exclusive of E_B , $E_{\bar{S}T}$, and E_{ST} since the membership change cannot happen in the scan window if the membership change occurred during the back-off period or within the 11.25 ms prior to the back-off period. Therefore, since $E_{\bar{S}T}$, $E_{S\bar{T}}$, E_{ST} , and $E_{\bar{S}\bar{T}}$ are also mutually exclusive and

$$\begin{aligned}
P(E_A|E_{H_i}) &= \frac{11.25ms}{1.28s} P(\overline{E_B} \cap (E_{S\overline{T}} \cup E_{\overline{S}T}) \cap E_{H_i}) \\
&= \frac{11.25ms}{1.28s} P(\overline{E_B}|E_{H_i}) (P(E_{S\overline{T}}|E_{H_i}) + P(E_{\overline{S}T}|E_{H_i})). \quad (8.33)
\end{aligned}$$

Additionally, the knowledge that E_{H_i} occurs determines the probability that E_L occurs. For example, $P(E_L|E_{H_5}) = 1$ since E_{H_5} can only occur if E_L occurs. Therefore,

$$P(E_L|E_{H_1}) = P(E_L|E_{H_{2b}}) = P(E_L), \quad (8.34)$$

$$P(E_L|E_{H_{2b}}) = P(E_L|E_{H_5}) = 1, \quad \text{and} \quad (8.35)$$

$$\begin{aligned}
P(E_L|E_{H_3}) &= P(E_L|E_{H_4}) \\
&= \frac{P(E_L) \left(1 - P(E_{H_1}) - P(E_{H_{2b}})\right) - P(E_{H_{2a}}) - P(E_{H_5})}{P(E_{H_3}) + P(E_{H_4})} \quad (8.36)
\end{aligned}$$

Likewise, the probability $\overline{E_B}$ occurs given E_{H_i} is

$$P(\overline{E_B}|E_{H_1}) = P(\overline{E_B}|E_{H_{2b}}) = P(\overline{E_B}), \quad (8.37)$$

$$P(\overline{E_B}|E_{H_{2b}}) = P(\overline{E_B}|E_{H_5}) = 0, \quad \text{and} \quad (8.38)$$

$$P(\overline{E_B}|E_{H_3}) = P(\overline{E_B}|E_{H_4}) = \frac{P(\overline{E_B}) \left(1 - P(E_{H_1}) - P(E_{H_{2b}})\right)}{P(E_{H_3}) + P(E_{H_4})}. \quad (8.39)$$

In the following sections, the above conditional probabilities are used to explicitly derive the conditional density functions.

8.5.1 Conditional density given E_{H_1}

If the scan frequency and train under which the first inquiry packet was received do not change during the back-off period and E_B did not occur, the inquiring device will receive a second inquiry packet when it re-enters the inquiry scan substate.

However, it is clear from Section 8.4 that the relationship between the scan frequency and inquiry train can change. If $E_{ST} \cap E_{H1}$ occurs, the second packet will be received immediately because the scan frequency will still be in the inquiry train and there is no chance that the train membership will change during the scan window implying that

$$P(N_2 = 0 | E_{ST} \cap E_{H1}) = 1. \quad (8.40)$$

If $(E_{\overline{ST}} \cap E_{H1})$ occurs, the second packet will be immediately received unless $E_A \cap E_L$ occur at which time the probability of receiving the second packet is 0.5 since the train will shift from containing the scan frequency in half of the possible configurations giving

$$P(N_2 \neq 0 | E_A \cap E_L \cap E_{\overline{ST}} \cap E_{H1}) = 0.5. \quad (8.41)$$

If $(E_{\overline{ST}} \cap E_{H1})$ occurs, the second packet will be received when $N_2 = 0$ except when E_L occurs, which causes the scanning frequency to effectively change trains and the packet will not be received until the train change at $t = 2.56$ s. A similar effect is seen when $E_{\overline{ST}}$ occurs except that $(E_A \cap E_L)$ may also occur, at which time the probability that the packet is received with $N_2 = 0$ is 0.5 in a situation similar to that described in (8.9).

Given the event $E_{H1} \cap N_2 = 0$,

$$\begin{aligned} f_{T_I}(t | E_{H1} \cap \{N_2 = 0\}) &= \left(1/P(E_{H1} \cap \{N_2 = 0\}) \right) \times \\ &\left(f_{T_P}(t | E_{ST} \cap E_{H1} \cap \{N_2 = 0\})P(\{N_2 = 0\} | E_{ST} \cap E_{H1})P(E_{ST} | E_{H1})P(E_{H1}) + \right. \\ &f_{T_P}(t | E_{\overline{ST}} \cap E_{H1} \cap \{N_2 = 0\})P(\{N_2 = 0\} | E_{\overline{ST}} \cap E_{H1})P(E_{\overline{ST}} | E_{H1})P(E_{H1}) + \\ &f_{T_P}(t | E_{\overline{ST}} \cap E_{H1} \cap \{N_2 = 0\})P(\{N_2 = 0\} | E_{\overline{ST}} \cap E_{H1})P(E_{\overline{ST}} | E_{H1})P(E_{H1}) + \\ &\left. f_{T_P}(t | E_{\overline{ST}} \cap E_{H1} \cap \{N_2 = 0\})P(\{N_2 = 0\} | E_{\overline{ST}} \cap E_{H1})P(E_{\overline{ST}} | E_{H1})P(E_{H1}) \right) \quad (8.42) \end{aligned}$$

which is

$$\begin{aligned}
f_{T_I}(t|E_{H1} \cap \{N_2 = 0\}) &= \left(1/P(E_{H1} \cap \{N_2 = 0\})\right) \times \\
&\left(f_{T_P}(t|E_{ST} \cap E_{H1})P(E_{ST}|E_{H1})P(E_{H1}) + \right. \\
&f_{T_P}(t|E_{\overline{ST}} \cap E_{H1})P(E_{\overline{ST}}|E_{H1})P(E_{H1})\left(1 - 0.5P(E_L|E_{H1})P(E_A|E_{H1})\right) + \\
&f_{T_P}(t|E_{\overline{ST}} \cap E_{H1})P(E_{\overline{ST}}|E_{H1})P(E_{H1})P(\overline{E_L}|E_{H1}) + \\
&\left.f_{T_P}(t|E_{\overline{ST}} \cap E_{H1})P(E_{\overline{ST}}|E_{H1})P(E_{H1})P(\overline{E_L}|E_{H1})\left(1 - 0.5P(E_A|E_{H1})\right)\right)
\end{aligned} \tag{8.43}$$

where

$$\begin{aligned}
P(E_{H1} \cap \{N_2 = 0\}) &= \\
&P(E_{ST}|E_{H1})P(E_{H1}) + P(E_{\overline{ST}}|E_{H1})P(E_{H1})P(\overline{E_L}|E_{H1}) + \\
&P(E_{\overline{ST}}|E_{H1})P(E_{H1})\left(1 - 0.5P(E_L|E_{H1})P(E_A|E_{H1})\right) + \\
&P(E_{\overline{ST}}|E_{H1})P(E_{H1})P(\overline{E_L}|E_{H1})\left(1 - 0.5P(E_A|E_{H1})\right).
\end{aligned} \tag{8.44}$$

Note that $f_{T_P}(t|E_X \cap E_{H1} \cap \{N_2 = 0\})$ is, in fact, independent of N_2 or any other events which occur after the back-off is complete and therefore equals $f_{T_P}(t|E_X \cap E_{H1})$, where $E_X \in \{E_{ST}, E_{\overline{ST}}, E_{ST}, E_{\overline{ST}}\}$.

The occurrence of $\{N_2 = 1\} \cap E_{H1}$ must be conditioned on E_W due to the train change at $t = 2.56$ s. The probability that the membership changes within the scan window and causes the first scan frequency after the back-off to miss but allows packet reception in the subsequent window is $1/16$ when $E_{H1} \cap E_L \cap E_A \cap \overline{E_W} \cap E_{\overline{ST}}$ or $E_{H1} \cap E_L \cap E_A \cap \overline{E_W} \cap E_{\overline{ST}}$. Note that under similar conditions when $E_{H1} \cap E_L \cap E_A \cap E_W \cap (E_{\overline{ST}} \cup E_{\overline{ST}})$ occurs, $N_2 = 3$ or 4 .

If $E_{H1} \cap E_W$ occurs when $E_{\overline{ST}}$, $E_{\overline{ST}}$, or $E_{\overline{ST}} \cap E_A \cap E_L$ occur and cause the scan frequency to effectively change trains to the train used after the change at $t = 2.56$ s, N_2 may equal 1. If $E_{\overline{ST}} \cap E_{H1} \cap E_A \cap E_L \cap E_M \cap E_W$ occurs, seven of the possible locations of the scanning frequency within the train will cause the

packet to be received since eight of the positions cause $N_2 = 0$ and the remaining one position causes $N_2 = 3$. This one position would have allowed the packet to be received between $t = 2.56$ and 3.84 s except that the train change cause receipt to be delayed until after $t = 5.12$ s. Thus, $P(\{N_2 = 1\}|E_{\overline{ST}} \cap E_{H1} \cap E_M \cap E_W) = P(E_A|E_{H1})P(E_L|E_{H1}) \times 7/16$. Similarly, if $E_{\overline{ST}} \cap E_L \cap E_{H1} \cap E_W$ occurs, $N_2 = 1$ except if E_A also occurs. If this and E_M occur, eight of the possible locations of the scanning frequency allow the packet to be received when $N_2 = 0$ and one would have allowed the packet to be received between $t = 2.56$ s and $t = 3.84$ s had the train not changed, leaving $P(\{N_2 = 1\}|E_{\overline{ST}} \cap E_{H1} \cap E_M \cap E_W) = P(E_L)(1 - 9P(E_A|E_{H1})/16)$. Finally, if $E_{\overline{ST}} \cap E_L \cap E_{H1}$ occurs, $N_2 = 1 \cap E_{H1}$ will occur regardless of E_W . Thus,

$$\begin{aligned}
f_{T_I}(t|E_{H1} \cap \{N_2 = 1\}) &= \left(1/P(E_{H1} \cap \{N_2 = 1\})\right) \times \\
&\left(f_{T_P}(t - 1.28s|\{N_2 = 1\} \cap E_{\overline{ST}} \cap E_A \cap E_L \cap E_{H1})P(E_L|E_{H1})P(E_{H1}) \times \right. \\
&\quad P(\{N_2 = 1\}|E_{\overline{ST}} \cap E_A \cap E_L \cap E_{H1})P(E_{\overline{ST}}|E_{H1})P(E_A|E_{H1})u(2.56 - t) + \\
&f_{T_P}(t - 1.28s|\{N_2 = 1\} \cap E_{\overline{ST}} \cap E_{H1})P(\{N_2 = 1\}|E_{\overline{ST}} \cap E_{H1}) \times \\
&\quad P(E_{\overline{ST}}|E_{H1})P(E_{H1})u(t - 2.56) + \\
&f_{T_P}(t - 1.28s|\{N_2 = 1\} \cap E_{\overline{ST}} \cap E_A \cap E_L \cap E_{H1})P(E_L|E_{H1})P(E_{H1}) \times \\
&\quad P(\{N_2 = 1\}|E_{\overline{ST}} \cap E_A \cap E_L \cap E_{H1})P(E_{\overline{ST}}|E_{H1})P(E_A|E_{H1})u(2.56 - t) \\
&f_{T_P}(t - 1.28s|\{N_2 = 1\} \cap E_{\overline{ST}} \cap E_{H1})P(\{N_2 = 1\}|E_{\overline{ST}} \cap E_{H1}) \times \quad (8.45) \\
&\quad P(E_{\overline{ST}}|E_{H1})P(E_{H1})u(t - 2.56) + \\
&f_{T_P}(t - 1.28s|\{N_2 = 1\} \cap E_{\overline{ST}} \cap E_L \cap E_{H1}) \times \\
&\quad \left. P(\{N_2 = 1\}|E_{\overline{ST}} \cap E_L \cap E_{H1})P(E_L|E_{H1})P(E_{\overline{ST}}|E_{H1})P(E_{H1})u(t - 2.56) \right)
\end{aligned}$$

where $u(t)$ is the unit step function. After some simplification, this becomes

$$\begin{aligned}
f_{T_I}(t|E_{H1} \cap \{N_2 = 1\}) &= \frac{P(E_L|E_{H1})P(E_{H1})}{16P(E_{H1} \cap \{N_2=1\})} \times \\
&\left(f_{T_P}(t - 1.28s|E_{S\bar{T}} \cap E_{H1})P(E_{S\bar{T}}|E_{H1})P(E_A|E_{H1}) \times \right. \\
&\quad \left(u(2.56 - t) + \left(\frac{16}{P(E_A|E_{H1})} - 9 \right) u(t - 2.56) \right) + \\
f_{T_P}(t - 1.28s|E_{\bar{S}T} \cap E_{H1})P(E_{\bar{S}T}|E_{H1})P(E_A|E_{H1}) &\left(u(2.56 - t) + 7u(t - 2.56) \right) + \\
16f_{T_P}(t - 1.28s|E_{\bar{S}T} \cap E_{H1})P(E_{\bar{S}T}|E_{H1}) &u(t - 2.56) \left. \right) \quad (8.46)
\end{aligned}$$

where

$$\begin{aligned}
P(E_{H1} \cap \{N_2 = 1\}) &= P(E_L|E_{H1})P(E_{H1})P(E_A|E_{H1}) \times \\
P(E_{S\bar{T}}|E_{H1}) &\left(P(\bar{E}_W|E_{S\bar{T}} \cap E_{H1})/16 + \left(\frac{1}{P(E_A|E_{H1})} - 9/16 \right) P(E_W|E_{S\bar{T}} \cap E_{H1}) \right) + \\
P(E_{\bar{S}T}|E_{H1}) &\left(P(\bar{E}_W|E_{\bar{S}T} \cap E_{H1})/16 + 7P(E_W|E_{\bar{S}T} \cap E_{H1})/16 \right) + \\
P(E_{\bar{S}T}|E_{H1})P(E_W|E_{\bar{S}T} \cap E_{H1}) &/P(E_A|E_{H1}). \quad (8.47)
\end{aligned}$$

If $N_2 > 1$ and E_{H1} occurs, $N_2 = 2$ in all cases except if $E_M \cap E_A \cap E_L \cap \bar{E}_W \cap (E_{S\bar{T}} \cup E_{\bar{S}T})$ occurs and the position of the scan frequency in the train was such that the packet would be received between $t = 2.56$ and 3.84 s if the train does not change. In this case, $N_2 = 4$. Thus

$$\begin{aligned}
f_{T_I}(t|E_{H1} \cap \{N_2 = 2\}) &= \frac{P(E_L|E_{H1})P(E_{H1})u(3.84 - t)}{16P(E_{H1} \cap \{N_2 = 2\})} \times \\
&\left(f_{T_P}(t - 2.56s|E_{S\bar{T}} \cap E_{H1})P(E_{S\bar{T}}|E_{H1}) \left(16 - 10P(E_A|E_{H1}) \right) + \right. \\
&\quad 6f_{T_P}(t - 2.56s|E_{\bar{S}T} \cap E_{H1})P(E_{\bar{S}T}|E_{H1})P(E_A|E_{H1}) + \\
&\quad \left. 16f_{T_P}(t - 2.56s|E_{\bar{S}T} \cap E_{H1})P(E_{\bar{S}T}|E_{H1}) \right) \quad (8.48)
\end{aligned}$$

where

$$\begin{aligned}
P(E_{H1} \cap \{N_2 = 2\}) &= P(E_L|E_{H1})P(E_{H1}) \times \\
&\left(6P(E_{\overline{ST}}|E_{H1})P(E_A|E_{H1})P(\overline{E_W}|E_{\overline{ST}} \cap E_{H1})/16 + P(E_{\overline{ST}}|E_{H1})P(\overline{E_W}|E_{\overline{ST}} \cap E_{H1}) + \right. \\
&\left. P(E_{\overline{ST}}|E_{H1})P(\overline{E_W}|E_{\overline{ST}} \cap E_{H1})\left(1 - 10P(E_A|E_{H1})/16\right)\right). \tag{8.49}
\end{aligned}$$

Recall that only one of the sixteen positions the scanning frequency can take in the train results in $N_2 = 3$, giving

$$\begin{aligned}
f_{T_I}(t|E_{H1} \cap \{N_2 = 3\}) &= \left(1/P(E_{H1} \cap \{N_2 = 3\})\right) \times \\
&\left(f_{T_P}(t - 3.84s|\{N_2 = 3\} \cap E_{\overline{ST}} \cap E_A \cap E_L \cap E_{H1})P(E_L|E_{H1})P(E_{H1}) \times \right. \\
&\quad P(\{N_2 = 3\}|E_{\overline{ST}} \cap E_A \cap E_L \cap E_{H1})P(E_{\overline{ST}}|E_{H1})P(E_A|E_{H1})u(t - 5.12) + \\
&f_{T_P}(t - 3.84s|\{N_2 = 3\} \cap E_{\overline{ST}} \cap E_A \cap E_L \cap E_{H1})P(E_L|E_{H1})P(E_{H1}) \times \tag{8.50} \\
&\quad \left. P(\{N_2 = 3\}|E_{\overline{ST}} \cap E_A \cap E_L \cap E_{H1})P(E_{\overline{ST}}|E_{H1})P(E_A|E_{H1})u(t - 5.12)\right),
\end{aligned}$$

or

$$\begin{aligned}
f_{T_I}(t|E_{H1} \cap \{N_2 = 3\}) &= \frac{(P(E_A|E_{H1})P(E_L|E_{H1})P(E_{H1})u(t - 5.12))}{16P(E_{H1} \cap \{N_2 = 3\})} \times \\
&\left(f_{T_P}(t - 3.84s|E_{\overline{ST}} \cap E_{H1})P(E_{\overline{ST}}|E_{H1}) + \right. \\
&\quad \left. f_{T_P}(t - 3.84s|E_{\overline{ST}} \cap E_{H1})P(E_{\overline{ST}}|E_{H1})\right), \tag{8.51}
\end{aligned}$$

where

$$\begin{aligned}
P(E_{H1} \cap \{N_2 = 3\}) &= P(E_L|E_{H1})P(E_{H1})P(E_A|E_{H1}) \times \tag{8.52} \\
&\left(P(E_{\overline{ST}}|E_{H1})P(E_W|E_{\overline{ST}} \cap E_{H1})/16 + P(E_{\overline{ST}}|E_{H1})P(E_W|E_{\overline{ST}} \cap E_{H1})/16\right).
\end{aligned}$$

This leaves

$$f_{T_I}(t|E_{H1} \cap \{N_2 = 4\}) = \frac{P(E_L|E_{H1})P(E_{H1})P(E_A|E_{H1})u(6.4 - t)}{16P(E_{H1} \cap \{N_2 = 4\})} \times \quad (8.53)$$

$$\left(f_{T_P}(t - 5.12s|E_{S\bar{T}} \cap E_{H1})P(E_{S\bar{T}}|E_{H1}) + f_{T_P}(t - 5.12s|E_{\bar{S}T} \cap E_{H1})P(E_{\bar{S}T}|E_{H1}) \right)$$

where

$$P(E_{H1} \cap \{N_2 = 4\}) = P(E_L|E_{H1})P(E_{H1})P(E_A|E_{H1}) \times \quad (8.54)$$

$$\left(P(E_{S\bar{T}}|E_{H1})P(\bar{E}_W|E_{S\bar{T}} \cap E_{H1}) + P(E_{\bar{S}T}|E_{H1})P(\bar{E}_W|E_{\bar{S}T} \cap E_{H1}) \right) / 16.$$

The probability density for the inquiry time when E_{H1} occurs is

$$f_{T_I}(t|E_{H1}) = \sum_{j=0}^4 f_{T_I}(t|E_{H1} \cap \{N_2 = j\})P(\{N_2 = j\}|E_{H1})P(E_{H1}). \quad (8.55)$$

8.5.2 Conditional density given E_{H2a}

The event $E_B \cap E_L$ must have occurred if E_{H2a} occurs and therefore E_A cannot occur, the train cannot change membership during the back-off period, and $P(\bar{E}_B|E_{H2a}) = 0$. However, if $E_M \cap E_{H2a}$ occurs, due to the nature of the relationship between the train change and scan frequency, the scan frequency will not be in the train used immediately after the back-off period unless the scan frequency changes during the back-off period. The converse is true if $E_{\bar{M}} \cap E_{H2a}$ occurs. If the packet could have been received in the first window but is not received because E_W occurs, $N_2 = 3$. Therefore,

$$\begin{aligned}
f_{T_I}(t|E_{H2a} \cap \{N_1 = 0\}) &= \left(1/P(E_{H2a} \cap \{N_1 = 0\})\right) \\
&f_{T_P}(t|E_{S\bar{T}} \cap E_{H2a} \cap E_M)P(E_M|E_{H2a})P(E_{S\bar{T}}|E_{H2a})u(2.56 - t) + \\
&f_{T_P}(t|E_{S\bar{T}} \cap E_{H2a} \cap \overline{E_M})P(\overline{E_M}|E_{H2a})P(E_{S\bar{T}}|E_{H2a})u(t - 2.56) + \\
&f_{T_P}(t|E_{S\bar{T}} \cap E_{H2a} \cap \overline{E_M})P(\overline{E_M}|E_{H2a})P(E_{S\bar{T}}|E_{H2a})u(2.56 - t) + \\
&f_{T_P}(t|E_{S\bar{T}} \cap E_{H2a} \cap E_M)P(E_M|E_{H2a})P(E_{S\bar{T}}|E_{H2a})u(t - 2.56), \quad (8.56)
\end{aligned}$$

$$\begin{aligned}
f_{T_I}(t|E_{H2a} \cap \{N_1 = 1\}) &= \left(1/P(E_{H2a} \cap \{N_1 = 1\})\right) \\
&f_{T_P}(t - 1.28|E_{S\bar{T}} \cap E_{H2a} \cap \overline{E_M})P(\overline{E_M}|E_{H2a})P(E_{S\bar{T}}|E_{H2a})u(3.84 - t) + \\
&f_{T_P}(t - 1.28|E_{S\bar{T}} \cap E_{H2a} \cap E_M)P(E_M|E_{H2a})P(E_{S\bar{T}}|E_{H2a})u(3.84 - t), \quad (8.57)
\end{aligned}$$

and

$$\begin{aligned}
f_{T_I}(t|E_{H2a} \cap \{N_1 = 3\}) &= \left(1/P(E_{H2a} \cap \{N_1 = 3\})\right) \\
&f_{T_P}(t - 3.84|E_{S\bar{T}} \cap E_{H2a} \cap E_M)P(E_M|E_{H2a})P(E_{S\bar{T}}|E_{H2a})u(t - 5.12) + \\
&f_{T_P}(t - 3.84|E_{S\bar{T}} \cap E_{H2a} \cap \overline{E_M})P(\overline{E_M}|E_{H2a})P(E_{S\bar{T}}|E_{H2a})u(t - 5.12) \quad (8.58)
\end{aligned}$$

where

$$\begin{aligned}
P(E_{H2a} \cap \{N_1 = 0\}) &= \\
&P(E_{S\bar{T}}|E_{H2a})\left(P(E_M|E_{H2a})P(\overline{E_W}|E_{H2a}) + P(\overline{E_M}|E_{H2a})P(E_W|E_{H2a})\right) + \\
&P(E_{S\bar{T}}|E_{H2a})\left(P(\overline{E_M}|E_{H2a})P(\overline{E_W}|E_{H2a}) + P(E_M|E_{H2a})P(E_W|E_{H2a})\right), \quad (8.59)
\end{aligned}$$

$$\begin{aligned}
P(E_{H2a} \cap \{N_1 = 1\}) &= P(E_{S\bar{T}}|E_{H2a})P(\overline{E_M}|E_{H2a})P(\overline{E_W}|E_{H2a}) + \\
&P(E_{S\bar{T}}|E_{H2a})P(E_M|E_{H2a})P(\overline{E_W}|E_{H2a}), \quad (8.60)
\end{aligned}$$

and

$$\begin{aligned}
P(E_{H2a} \cap \{N_1 = 3\}) = & P(E_{\overline{ST}}|E_{H2a})P(E_M|E_{H2a})P(E_W|E_{H2a}) + \\
& P(E_{\overline{ST}}|E_{H2a})P(\overline{E_M}|E_{H2a})P(E_W|E_{H2a}). \quad (8.61)
\end{aligned}$$

8.5.3 Conditional density given E_{H2b}

Since the event E_{H2b} is essentially the event E_{H1} extended across $t = 1.28$ s, the conditions under which $N_2 = j$, $j = 0, 1, 2, 3, 4$, are similar to those when E_{H1} occurs. The only significant difference is that the density function for $T_F|E_{H2b}$ is modelled as a point mass making $T_P(t|E_{H2b}) = T_B(t - 1.28|E_{H1})$. Since E_W cannot occur and there is only one scan window before the train change at $t = 2.56$,

$$\begin{aligned}
f_{T_i}(t|E_{H2b} \cap \{N_2 = 0\}) = & \left(1/P(E_{H2b} \cap \{N_2 = 0\})\right) \times \\
& \left(f_{T_B}(t|E_{ST} \cap E_{H2b})P(E_{ST}|E_{H2b})P(E_{H2b}) + \right. \\
& f_{T_B}(t|E_{\overline{ST}} \cap E_{H2b})P(E_{\overline{ST}}|E_{H2b})P(E_{H2b})(1 - 0.5P(E_L|E_{H2b})P(E_A|E_{H2b})) + \\
& f_{T_B}(t|E_{\overline{ST}} \cap E_{H2b})P(E_{\overline{ST}}|E_{H2b})P(E_{H2b})P(\overline{E_L}|E_{H2b}) + \quad (8.62) \\
& \left. f_{T_B}(t|E_{\overline{ST}} \cap E_{H2b})P(E_{\overline{ST}}|E_{H2b})P(E_{H2b})P(\overline{E_L}|E_{H2b})(1 - 0.5P(E_A|E_{H2b}))\right)
\end{aligned}$$

where

$$\begin{aligned}
P(E_{H2b} \cap \{N_2 = 0\}) = & P(E_{ST}|E_{H2b})P(E_{H2b}) + P(E_{\overline{ST}}|E_{H2b})P(E_{H2b})P(\overline{E_L}|E_{H2b}) + \\
& P(E_{\overline{ST}}|E_{H2b})P(E_{H2b})(1 - 0.5P(E_L|E_{H2b})P(E_A|E_{H2b})) + \\
& P(E_{\overline{ST}}|E_{H2b})P(E_{H2b})P(\overline{E_L}|E_{H2b})(1 - 0.5P(E_A|E_{H2b})). \quad (8.63)
\end{aligned}$$

As with $\{N_2 = 2\} \cap E_{H1}$,

$$\begin{aligned}
f_{T_I}(t|E_{H2b} \cap \{N_2 = 1\}) &= \left(P(E_L|E_{H2b})P(E_{H2b})/P(E_{H2b} \cap \{N_2 = 1\}) \right) \times \\
&\left(f_{T_B}(t - 1.28s|E_{\overline{S}T} \cap E_{H2b})P(E_{\overline{S}T}|E_{H2b})(1 - 9P(E_A|E_{H2b})/16) + \right. \\
&7f_{T_B}(t - 1.28s|E_{\overline{S}T} \cap E_{H2b})P(E_{\overline{S}T}|E_{H2b})P(E_A|E_{H2b})/16 + \\
&\left. f_{T_P}(t - 1.28s|E_{\overline{S}T} \cap E_{H2b})P(E_{\overline{S}T}|E_{H2b}) \right) \quad (8.64)
\end{aligned}$$

where

$$\begin{aligned}
P(E_{H2b} \cap \{N_2 = 1\}) &= \left(P(E_L|E_{H2b})P(E_{H2b}) \right) \left(P(E_{\overline{S}T}|E_{H2b}) \left(1 - 9P(E_A|E_{H2b})/16 \right) + \right. \\
&\left. 7P(E_{\overline{S}T}|E_{H2b})P(E_A|E_{H2b})/16 + P(E_{\overline{S}T}|E_{H2b}) \right). \quad (8.65)
\end{aligned}$$

Likewise,

$$\begin{aligned}
f_{T_I}(t|E_{H2b} \cap \{N_2 = 3\}) &= \\
&\frac{f_{T_B}(t - 5.12s|E_{\overline{S}T} \cap E_{H2b})P(E_{\overline{S}T}|E_{H2b}) + f_{T_B}(t - 5.12s|E_{\overline{S}T} \cap E_{H2b})P(E_{\overline{S}T}|E_{H2b})}{P(E_{\overline{S}T}|E_{H2b}) + P(E_{\overline{S}T}|E_{H2b})}. \quad (8.66)
\end{aligned}$$

The probability density for the inquiry time when E_{H1} occurs is created by using the method used in (8.55) and combining the portions of the density as

$$f_{T_I}(t|E_{H2}) = \frac{f_{T_I}(t|E_{H2a})P(E_{H2a}) + f_{T_I}(t|E_{H2b})P(E_{H2b})}{P(E_{H2a}) + P(E_{H2b})}. \quad (8.67)$$

Note that, although discrete, $f_{T_I}(t|E_{H2b})$ is scaled to continuous time by dividing it by T_{slot} .

8.5.4 Conditional density given E_{H3}

The probability density function for $T_I|E_{H3}$ is similar to $f_{T_I}(t|E_{H1})$. The only differences are the distribution is shifted to the right by 2.56 s and $P(\overline{E_B}|E_{H3})$ is used instead of $P(\overline{E_B}|E_{H1})$. The densities are almost identical except for the 2.56 s shift and the events E_M and $\overline{E_M}$ are reversed in (8.40) through (8.55).

8.5.5 Conditional density given E_{H4}

The probability density for the inquiry time when E_{H4} occurs is very similar to the probability density for E_{H2b} , except that $P(\overline{E_B}|E_{H4})$ must be used instead of $P(\overline{E_B}|E_{H2b})$.

8.5.6 Conditional density given E_{H5}

The probability for the inquiry time when E_{H5} occurs is similar to the probability density for the inquiry time when E_{H2a} occurs. All conditions are identical except that the train switch does not occur until at least 640.625 ms after $T_P|E_{H5}$ and E_M must have occurred, thus making the occurrence of E_W inconsequential. Therefore,

$$f_{T_I}(t|E_{H5} \cap \{N_1 = 0\}) = f_{T_P}(t|E_{S\overline{T}} \cap E_{H5}) \quad (8.68)$$

and

$$f_{T_I}(t|E_{H5} \cap \{N_1 = 2\}) = f_{T_P}(t - 2.56|E_{S\overline{T}} \cap E_{H5}). \quad (8.69)$$

8.6 Results

By applying (8.10) through (8.16) and (8.44) through (8.69) to (8.31),

$$f_{T_I}(t) = \sum_{i=1}^5 \sum_{j=0}^4 f_{T_I}(t|E_{H_i} \cap \{N_2 = j\})P(\{N_2 = j\}|E_{H_i})P(E_{H_i}),$$

the unconditional inquiry time density for discovery using v1.1 of the BT specification. The density is shown in Figure 8.7.

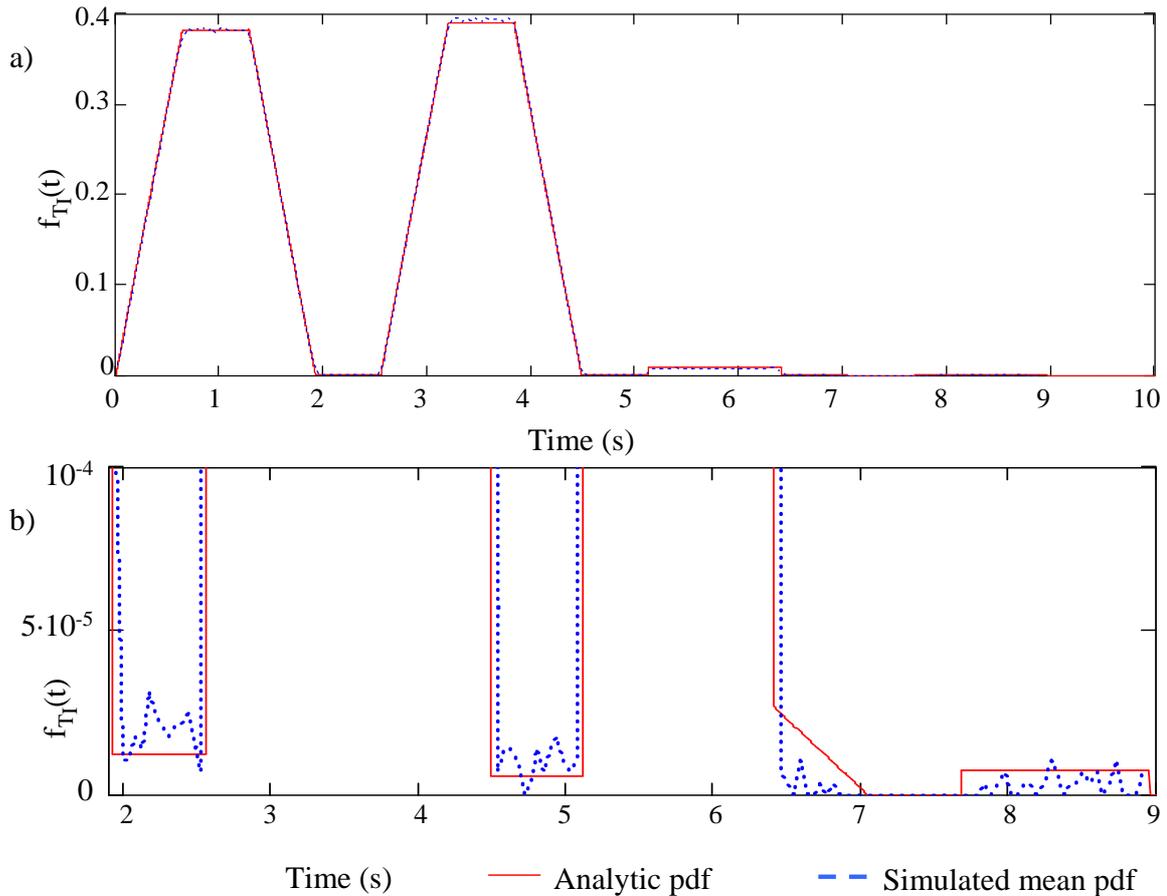


Figure 8.7: Unconditional probability density for inquiry scan using v1.1 of the specification, $f_{T_I}(t)$ a) full scale b) magnified view.

Using the derived density, the expected inquiry time is 2.292 s. Since the time spent in the inquiry substate must be a multiple of 1.28 s [Blu03] determined by the BT variable *Inquiry_Length*, the percentage of devices expected to be discovered are listed by possible inquiry duration times in Table 8.1. Assuming a perfect channel, 99% of the packets are received when the inquiry duration is 5.12 seconds. When the duration is extended to 6.4 s, 99.98% of inquiry packets are received. By remaining in the inquiry substate for the additional 1.28 s, only an additional 1% of the devices are discovered while 2048 packet slots for regular traffic are lost. Additionally, remaining in the inquiry substate consumes twice as much power as the connection

state [KaL01]. It is clear that devices should remain in the inquiry state for 5.12 s when it is expected that the scanning devices are using the standard inquiry scan.

Table 8.1: Devices discovered by inquiry time.

Inquiry_Length	Inquiry Duration	% Discovered
1	1.28 s	36.71
2	2.56 s	48.96
3	3.84 s	86.71
4	5.12 s	98.95
5	6.4 s	99.98

The expected inquiry time from the derived pdf of 2.292 s is considerably larger than 329.7 ms derived in [SBT00] [SBT01] [ZaG04] where it was assumed that the devices continuously receives when in the inquiry scan substate and a single train is used. Even when small scan windows were considered [KaL01], failure to account for the two trains of inquiry frequencies causes the derived expected inquiry time to only equal 964.1 ms. The difference between the expected inquiry time and mean inquiry time collect via experiment of 2.22 ms was explained by possible channel noise in the experiment.

8.6.1 Simulated and Measured Results

A MATLAB[®] simulation was written to emulate a device measurement experiment which characterized the density of inquiry time between two BT devices [KaL01] one meter apart. Inquiry time data was collected for a device that remained in the inquiry scan substate. The second device entered the inquiry substate for 12.8 seconds and recorded the time the scanning device was discovered. The inquiring device then entered the standby state for a time uniformly distributed on (0, 12.8) seconds to in an attempt to remove “synchronization artifacts” and then re-entered the inquiry substate. Note that the relationship between scan frequency changes and train membership changes remain constant through the 1500 replications of the

experiment; they both change every 1.28 s and maintain the same relation to one another, regardless of when data collection begins. This makes synchronization artifacts impossible to remove. The inquiry process was repeated 1500 times. Our simulation repeated the experiment 4,000 times. Since each replication of the experiment produced a different inquiry time pdf based on the relationship between the clock/addresses of the devices involved, the simulation yielded a mean pdf shown in Figure 8.7. The derived pdf falls within the 95% confidence interval of the simulated pdf. The difference between the analytical and simulated mean density is shown in Figure 8.8.

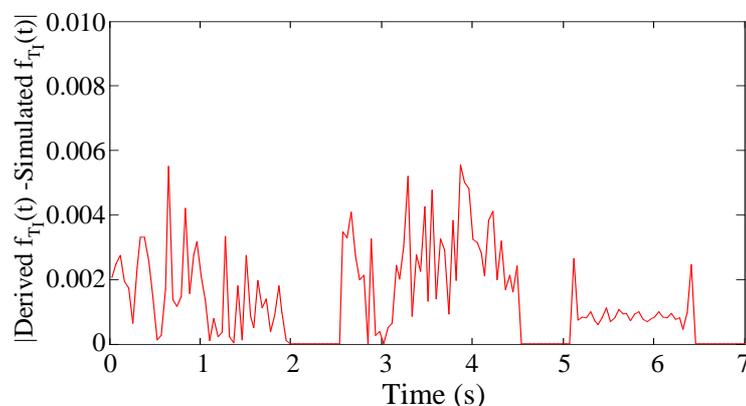


Figure 8.8: Difference in $f_{T_I}(t)$ between derived and simulated mean pdf using specified inquiry process.

Likewise, the distribution closely resembles an inquiry time simulation perform using NS with BlueHoc shown in Figure 8.9a [KaP02]. The null hypothesis that the simulated distribution is statistically equivalent to the derived distribution is not rejected at the 0.05 level for either the Kolmogorov-Smirnov ($KS = 0.8851$, $Critical = 1.358$) or Cramér-von Mises ($CV = 0.1281$, $Critical = 0.461$) tests. The derived distribution is also statistically similar to the data received from an experiment that attempted to duplicate the experiment of [KaL01] using Brain Boxes PCMCIA cards shown in Figure 8.9b [Leo03]. The null hypothesis that the simulated distribution is statistically equivalent to the derived distribution is not rejected at

the 0.05 level for either the Kolmogorov-Smirnov ($KS = 1.347$, $Critical = 1.358$) or Cramér-von Mises ($CV = 0.2985$, $Critical = 0.461$) tests.

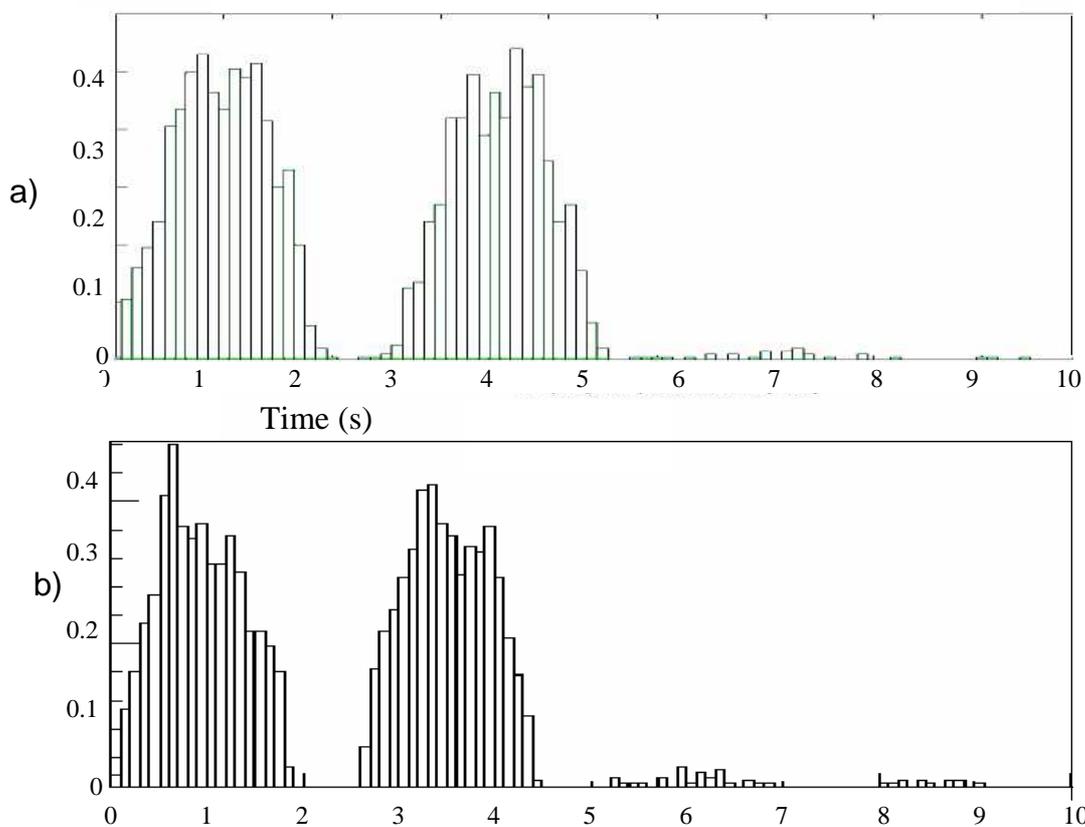


Figure 8.9: Probability density estimate for standard inquiry scan a) simulated [KaP02] b) measured [Leo03].

However, neither the simulated nor the derived pdf are similar to the measured inquiry time distribution shown in Figure 8.10. This discrepancy is discussed and resolved in Chapter IX.

8.6.2 Inquiry Scan using Specification v1.2

It has been shown that the discovery time can be reduced by reducing the back-off time [ZaC02]. Back-off was put into the specification to reduce collisions when multiple devices have open scan windows and an inquiring device begins transmitting inquiry packets. In version 1.2 of the Bluetooth specification, the back-off

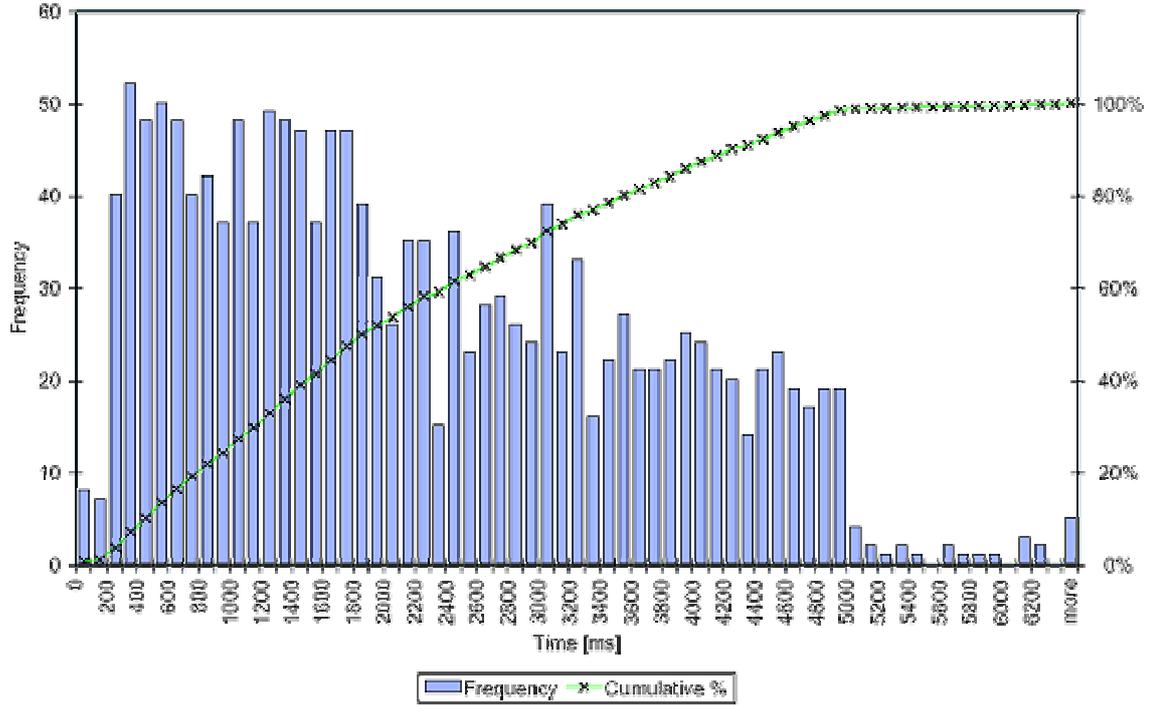


Figure 8.10: Measured probability density estimate for standard inquiry scan [KaL01].

time and second inquiry packet reception requirement prior to FHS transmission is removed [Blu03]. The probability of FHS packets colliding for two random scanning devices is 0.00024 without the back-off, compared to 0.00016 using the back-off. With the back-off scheme removed, the inquiry time pdf is

$$f_{T_I}(t) = \sum_{i=1}^5 f_{T_F}(t|E_{H_i})P(E_{H_i}). \quad (8.70)$$

Thus 99% of the devices can be discovered in 3.83 s as shown in Figure 8.11.

Additionally, in v1.2 of the specification, an interlaced inquiry scan is introduced. The interlaced inquiry scan consists of an 11.25 ms scan window like that used in the v1.1 inquiry scan, immediately followed by a second 11.25 ms scan window using a scan frequency from the other train. Therefore, if the inquiry packet is not received in the first 11.25ms, it is received in the second 11.25 ms. Likewise,

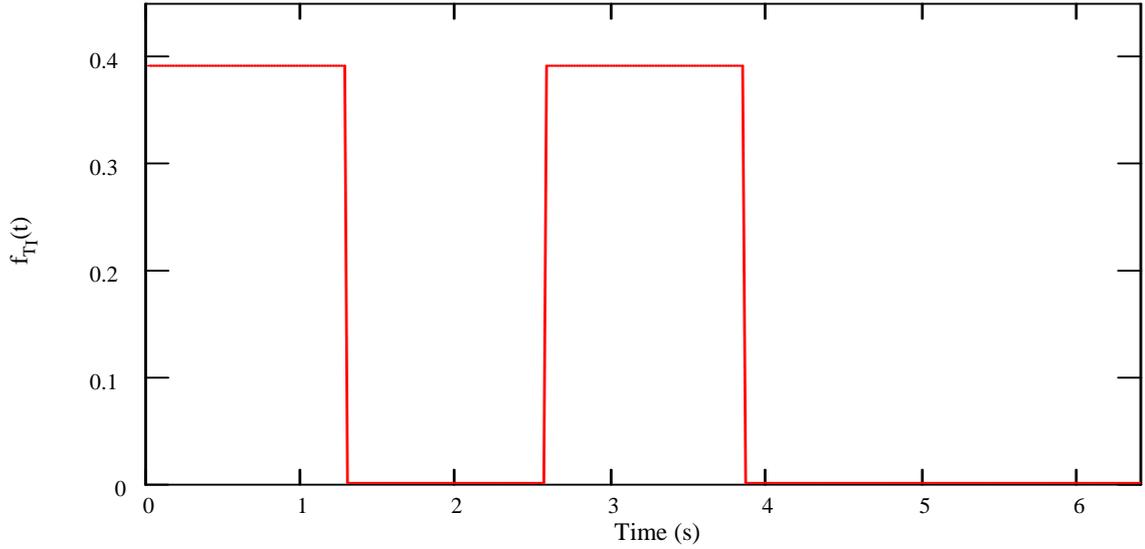


Figure 8.11: Unconditional probability density for the standard inquiry scan, $f_{T_I}(t)$.

the second inquiry packet will be received in the first 22.5 ms after the the back-off completes. The only exception is if the membership of the train shifts during the one of the two scan windows, the scan frequency is the frequency which changes from the second window, and the change occurs after the frequency from the first train which changes trains has been transmitted. The probability of this occurring is $P(E_B)P(E_L)P(\overline{E_M}) = 2.75 \times 10^{-4}$. If this does occur the inquiry completion is delayed by 1.28 s. Therefore, the interlaced inquiry scan inquiry time pdf is essentially

$$f_{T_I}(t) = \frac{1}{2} \left(f_{T_F}(t) + f_{T_F}(t - 11.25ms) \right) \quad (8.71)$$

and is effectively uniform on $[0, 1.28s]$. If devices remain in the interlaced inquiry scan mode opening 11.25ms windows every 1.28s, the devices remain constantly available for discovery while only removing them from the connection or standby state 1.75% of the time. Thus, there is very little impact on power consumption or scatternet throughput while discovering 99% of devices within 1.28s.

8.7 *Noise and Other Disrupting Factors*

When synchronous (SCO) channels are used in either the inquiring or scanning devices or the channel is noisy, a greatly increased inquiry time can result. A missed inquiry packet extends the inquiry time by at least 1.28 s while two consecutively missed inquiry packets can delay the inquiry by 5.12 s due to the train changes in a standard inquiry scan. However, these effects can be mitigated by extending the duration of the scan window. For example, if the first inquiry packet is disrupted or not sent due to SCO link requirements, but the scan window remains open for 22.5 ms, the inquiry packet can be received in the second repetition of the train within the window.

Although the distribution was validated with measurement and simulation data, a set of measured data shows a different distribution [KaL01]. The authors of the experiment offer no explanation other than possible noise. This is unlikely since the measured devices were 1 meter apart. Additionally, if noise caused the distortion in the distribution, it would be expected that both modes of the distribution would be spread to the right as the inquiry time was extended. However, this was not the case; the expected value for the measure inquiry time was lower than the expected time of the validated model. To fully investigate the inquiry time, it was assumed that the manufacturers deviated from the specification. It was theorized that the manufacturers had used CLK_{13} to determine the which train was used in the inquiry. This theory is investigated and confirmed in Chapter IX.

8.8 *Collision Avoidance*

For inquiring nodes which are connected to a pico- or scatternet, collision avoidance techniques can be implemented to limit the impact of inquiry packet transmissions on piconet traffic. As with PIAM and AFIT, knowledge of the neighboring piconet hop sequences enable the inquiring device to predict which inquiry packet

will interfere with the piconet traffic. Since the piconet traffic is a higher priority, the inquiring device should always skip its packet transmission if a collision is expected. Additionally, inquiry packets should also be skipped which, if received by a scanning device, will result in an FHS packet that will collide with piconet traffic. There is little use in allowing an inquiry packet to be transmitted knowing the response will likely be destroyed. The scanning device would then enter a back-off period before re-entering the inquiry scan substate, delaying discovery.

As shown in this chapter and Chapter V, the derivations for inquiry packet interference effects and inquiry time are quite complex. Combining the two to determine the inquiry time with collision avoidance is not feasible. Therefore, the impact on the inquiry time distribution is determined via simulation.

A MATLAB[®] simulation model was developed that determined the inquiry time required for an inquiring device to discover a node in the default inquiry scan substate (i.e., opens an 11.25 ms scan window every 1.28 s) while avoiding collisions with neighboring piconets. As Table 8.2 shows, the impact on inquiry time is minimal. With 2500 simulation runs, the inquiry time was delayed by 1.28 s when avoiding a single piconet in 1.8% of the cases. This occurs when the inquiry packet is not transmitted on the scan frequency due to collision avoidance and the scanning device does not keep the scan window open long enough to receive the next transmission on that frequency. In 0.16% of the cases, the inquiry packet occurs at the beginning of the scan window, is not transmitted due to collision avoidance, but is transmitted and received at the end of the scan window. The 11.25 ms permits the scanning device to receive the second copy of the inquiry packet on the scanning frequency in some cases, as it is transmitted every 10 ms.

As the number of piconets increases, additional inquiry packets are skipped as more piconet packets must be avoided. Note that if the scanning device misses packets in two consecutive windows, the train change in the inquiring device typically does not allow the the inquiry packet to be received until the train changes again,

Table 8.2: Inquiry time using collision avoidance.

# of piconets	Mean inquiry time	% delayed by 10 ms	% delayed by 1.28 s	% delayed by 5.12 s
0	1.90	0	0	0
1	1.92	0.16	1.84	0
2	1.95	0.40	3.68	0.04
3	1.99	0.56	5.80	0.28
4	2.02	0.84	7.44	0.52

causing at least a 5.12 s extension of the inquiry time. These delays are evident in the pmf of the inquiry time with three neighboring piconets shown in Figure 8.12. The mean inquiry time increases as the number of piconets increases and the number of skipped inquiry packets increases.

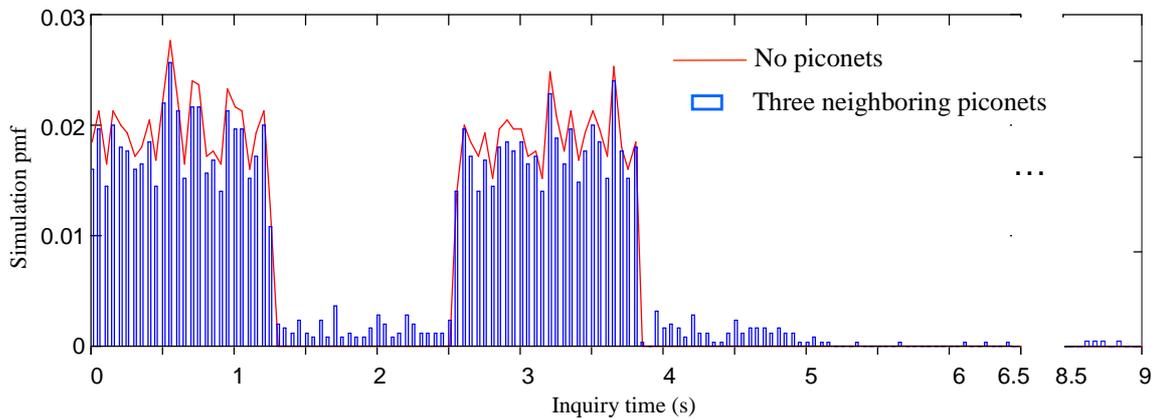


Figure 8.12: Inquiry time pmf with collision avoidance for three neighboring piconets.

8.9 Multiple Inquiring Nodes

The impact of multiple inquiring nodes is important since several nodes may be inquiring in a scatternet scenario in some of the possible outreach scenarios. Again, an analytical model describing the inquiry time which incorporates multiple inquiring nodes is infeasible.

The standard inquiry scan is used since it is likely to be the most commonly used scan type for future BT devices. The presence of multiple inquiring devices presents several challenges in device discovery. It is not feasible for inquiring devices to avoid collisions with other inquiring devices as there may be a large number of them in a scatternet and it is impractical for each node to monitor the activities of every other node in the scatternet. Fortunately, the short duration of inquiry packets makes the probability of packet collisions relatively small ($2 \frac{68}{625} \frac{1}{32} = 0.068$).

Table 8.3: Inquiry time with multiple inquirers.

Inquiring Devices	Mean Inquiry Time	% not found after 15 s
1	1.90	-
2	2.40	1.1
3	2.77	1.4
4	3.24	3.5
5	4.07	6.0

However, as Table 8.3 shows, the mean inquiry time increases significantly as the number of inquiry nodes rises. The inquiry times were collected using 2500 runs of the Matlab simulation model used in Section 8.8. The model recorded the time required for a primary inquiring node to discover a scanning device in the presence of secondary inquiring nodes. In some cases, when the primary node is the first to discover the scanning node and the inquiry packet is not disrupted, the inquiry time is no different from that required when no other inquiry nodes are present. However, when the primary inquiring piconet is not the first to discover the scanning node, the scanning node transmits a response FHS packet to the secondary inquiring node that discovered it, closes the scan window for the back-off period, advances its scan frequency by adding a 1.28 s offset to the CLK, and then opens a scan window to repeat the process. This may occasionally permit the inquiry time for the primary inquiring node to decrease if the advance in the scan frequency shifts the scan frequency into a different train. More likely, however, the inquiry time for the primary inquiring device increases as shown in Figure 8.13.

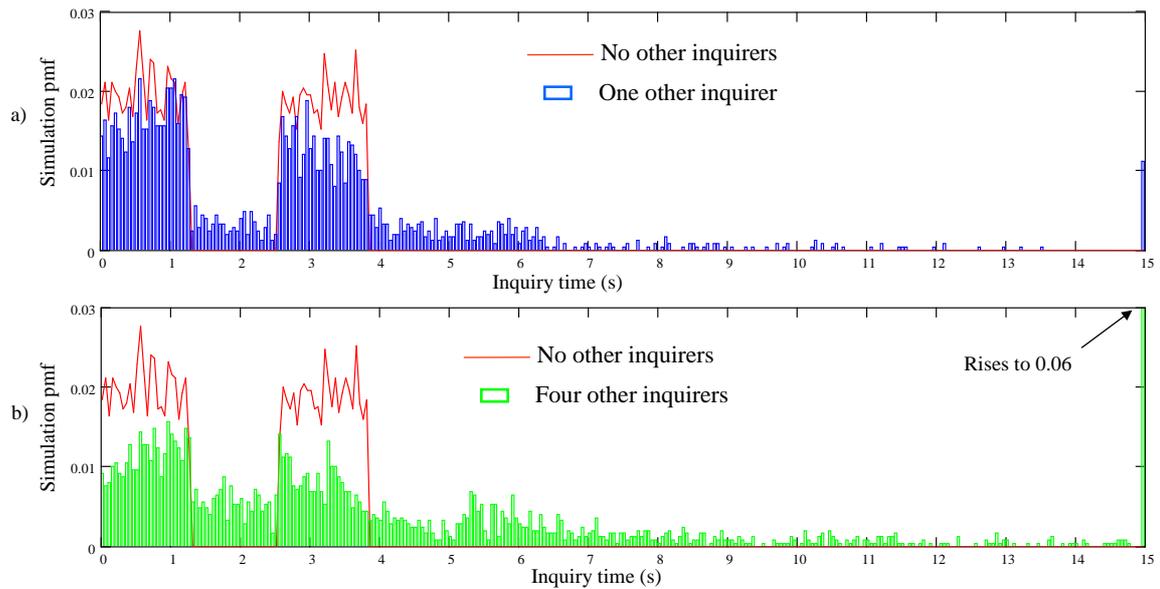


Figure 8.13: Inquiry time pmf with a) one other inquiring node b) four other inquiring nodes.

In some cases, when the trains used by the inquiring devices are similar but slightly different in phase, it is possible that a secondary device consistently transmits an inquiry packet on the scan frequency before the primary device transmits its inquiry packet on that scan frequency during a scan window. Thus, until the secondary device ceases its inquiry process, the primary inquiring device *never* discovers the scanning node. Thus, the simulation was ended after 15 seconds of discovery attempts. For example, with one neighboring inquirer, 1.1% of the primary nodes did not find the scanning node within 15 s. The mean inquiry time listed in Table 8.3 includes only those devices discovered within 15 s.

8.10 Multiple Inquiring Nodes and Collision Avoidance

It is apparent that the presence of additional inquirers have a much greater effect on inquiry time than inquiry packet collision avoidance with piconets. Although collision avoidance may cause a secondary inquirer's packet to be skipped, allowing the scanning device to receive the primary inquirer's packet when it would not have

otherwise, this is unlikely. The simulation was run 2500 times with one secondary inquirer and one piconet. Both inquirers avoided collisions with the piconet's packets. In the presence of a piconet, the percentage of devices not discovered within 15 seconds dropped from 1.1% to 0.9%. The resultant mean inquiry time for the devices discovered within 15 seconds was 2.43 s, which is very similar to the mean inquiry time when the only the secondary inquirer was present. As expected, the presences of multiple inquirers dominates the impact on the inquiry time.

8.11 Summary

In this chapter, a rigorous derivation of the inquiry time pdf for a BT device which periodically opens an inquiry scan window was presented. The possible interactions between the inquiry trains and the scan frequencies in the Bluetooth discovery process are numerous and complex. By identifying these interactions, the pdf for the inquiry time was derived for multiple devices in standard inquiry scan substate in a noiseless environment. Understanding these interactions easily lends itself to developing the pdf for the inquiry time under different situations and implementations of the BT discovery process such as the standard and interlaced inquiry scans. The derived distributions provide the information needed to reduce the inquiry substate dwell time, which in turn reduces power requirements, interference, and increase piconet throughput.

The 10.24 s inquiry substate duration recommended in the specification is much longer than required with little gain in device discovery capability. With the inquiry scan from v1.1 of the specification, the discovery time can be cut in half. The impact is even more significant when using the standard and interlaced inquiry scan from v1.2 of the specification; 99.99% of devices are discovered when the inquiry time is reduced by 63.5% and 87.5%, respectively.

Collision avoidance can be used to remove the impact of inquiring devices in a scatternet on neighboring piconets without significantly affecting the inquiry time. However, the presence of multiple inquiring node can significantly increase the inquiry time.

In Chapter IX, the discrepancy between the measured data in [KaL01] and the inquiry time pdf derived and validated in this Chapter is investigated and resolved.

IX. Simplification in the Inquiry Substate

In Chapter VIII, one set of measurements does not match the derived and validated models using v1.1 of the BT specification. Thus, a likely deviation in the specification is explored in this chapter. In the pdf obtained via measurement shown in Figure 8.10, a large number of sample points lie between $t = 1.92$ and $t = 2.56$ seconds. In the derived and simulated densities as well as other measured densities [KaP02] [Leo03], very few devices have an inquiry time between $t = 1.92$ and $t = 2.56$ seconds. Given this difference, it was theorized that the train of the inquiring device used in the experiment changed based on bit 13 of the inquiring device's CLK rather than using a separate 2.56 s counter. This simplifies the Bluetooth protocol implementation while retaining compatibility with devices that more strictly follow the standard. It is shown through the derived and simulation distribution models that some manufacturers have chosen to simplify the inquiry substate in a manner compatible with the specification which slightly improves performance. This modified inquiry substate is referred to as a *CLK-driven inquiry*.

9.1 CLK-Driven Inquiry Analysis

Complex interactions cause special, but unlikely, situations which make the derivation of the inquiry time pdf cumbersome. Two such cases occur when the train membership changes during a scan window or when a scan window overlaps $t = 0$ s. The interactions are even more complex when using the CLK-driven inquiry since it is possible the train in use (A or B) also changes during a scan window. To simplify the situation, it is assumed the train membership changes do not occur during the scan window. This introduces an error of 0.88% (11.25 ms/1.28 s) into the pdf but simplifies the density function significantly. Additionally, it is assumed when E_{Hi} occurs, the reception time of the first inquiry packet, T_F , is uniform on $\left((i-1)1.28, 1.28i\right)$, $i \in 1, 2, 3$. This introduces an error of 0.042% but simplifies the

analysis substantially. To derive the inquiry time pdf, much of the notation used in Chapter VIII is used. As in Chapter VIII,

$$f_{T_I}(t) = \sum_{i=1}^5 \sum_{j=0}^4 f_{T_I}(t|E_{Hi} \cap \{N_2 = j\})P(\{N_2 = j\}|E_{Hi})P(E_{Hi}), \quad (9.1)$$

although the conditional distributions and probabilities differ.

During the back-off period, the scan frequency generated by the scanning device may change, which is denoted as S in the following event definitions. Also, the inquiry train membership may change by itself (T) or in conjunction with a changing of the inquiry used by the inquiring device (C). When the train used changes, the train memberships also change since they are also tied to the CLK. It is possible for the scan frequency to effectively change trains when the scan frequency changes or the train membership changes if E_L also occurs. The possible event combination are:

$E_{S\overline{TC}}$:the scan frequency changes but the train does not change membership during the back-off period, thus the train used by the inquiring device also does not change due to train choice being driven by CLK_{13} . This is shown in Figure 8.4a.

E_{ST} : the train membership and the scan frequency both change during the back-off period but the train used does not change, leaving the scan frequency in the same train as before the back-off period.

E_{SC} : the train membership, the scan frequency, and the train used all change during the back-off period.

$E_{\overline{STC}}$:neither the train membership nor the scan frequency change during the back-off period which implies the train being used also does not change.

$E_{\overline{ST}}$: the train membership changes but neither the scan frequency nor the train used changes during the back-off period. Again, this may effectively result in a train change as shown in Figure 8.4b.

$E_{\overline{SC}}$: the train being used changes but the scan frequency does not change during the back-off period.

Using the above events, the unconditional and conditional probabilities needed to derive the inquiry time pdf are developed below, beginning with the probability the first inquiry packet is received in the i th 1.28 s interval.

9.1.1 Probability of E_{H_i} , $i = \{1, 2, 3, 4, 5\}$

With our assumptions

$$P(E_{H1}) = P(E_M) = 0.5. \quad (9.2)$$

The scan frequency has a 0.5 probability of being in the first train used by the inquiring device with a random pairing of Bluetooth devices. As mentioned previously, $T_F|E_{H1}$ is uniformly distributed on $(0, 1.28)$ s. The event E_{H2} occurs when $\overline{E_M}$ occurs and the train changes between 0 and 1.28 s. Since the time of the first train change, T_C , is uniformly distributed on $(0, 2.56)$ s and is independent of the scan frequency's relationship to the current train,

$$P(E_{H2}) = P(\overline{E_M})P(\{0 \leq T_C \leq 1.28\}) = 0.25. \quad (9.3)$$

Likewise, E_{H3} occurs if the train change occurs between 1.28 and 2.56 seconds, giving

$$P(E_{H3}) = P(\overline{E_M})P(\{1.28 \leq T_C \leq 2.56\}) = 0.25. \quad (9.4)$$

The reception time of the first packet given E_{H2} or E_{H3} is uniformly distributed on (1.28, 2.56) s and (2.56, 3.84) s, respectively. With our assumptions, E_{H4} and E_{H5} cannot occur.

9.1.2 Conditional Densities for T_B Completion

Recall that T_B is discretely uniform between 0 and 639.375 ms and its sample space contains 1024 points. The pdf of the back-off period completion time, T_P , is the convolution of the pdfs for T_B and T_F —the sum of the two independent random variables.

The pdf of T_F is dependent on E_{Hi} , therefore, the density function of T_P must be conditioned on E_{Hi} , $i = 1, 2, 3$. However, T_B is independent of E_{Hi} . Thus, the conditional pdf for T_P is

$$f_{T_P}(t|E_{Hi}) = f_{T_B}(t) * f_{T_F}(t|E_{Hi}). \quad (9.5)$$

The inquiry time is dependent on events $E_{\overline{STC}}$, $E_{\overline{STC}}$, $E_{\overline{ST}}$, $E_{\overline{SC}}$, E_{ST} , and E_{SC} which are further dependent on T_B . As T_B gets larger, the probability of $E_{\overline{STC}}$ decreases. Therefore, f_{T_P} is conditioned on these events as well with

$$f_{T_P}(t|E_{Hi}) = \sum_X f_{T_P}(t|E_{Hi} \cap E_X)P(E_X|E_{Hi})P(E_{Hi}) \quad (9.6)$$

where $X \in \{\overline{STC}, \overline{STC}, \overline{ST}, \overline{SC}, ST, SC\}$ and $i = 1, 2, 3$. These conditional probabilities and probability densities are defined below.

Recall that the scan frequency is based on the free running counter of the scanning device. Therefore, the time until a change in scan frequency is distributed continuously uniform on (0, 1.28) s and is denoted S_S . Note that this time is not dependent on the beginning of the scan window, but rather represents when the frequency generating subsystem changes the current scan frequency.

Likewise, the time until the next membership change in the train, denoted as S_T , is distributed uniformly on $(0, 1.28)$ s. Thus, the cumulative distribution functions (CDF) of S_S and S_T , $F_S(t)$ and $F_T(t)$ respectively, are

$$F_S(t) = F_T(t) = \begin{cases} 0 & t < 0 \\ \frac{t}{1.28s} & 0 \leq t \leq 1.28s \\ 1 & \textit{otherwise} \end{cases} . \quad (9.7)$$

When train membership changes during the back-off period, the inquiring device changes trains with a probability of 0.5 given E_{H1} . This is due to the train change being tied to the same clock driving the membership change with the membership changing every 1.28 s but the train being used changing with every other membership change. Denote the event that the train changes during the back-off period given that the train changes membership during that same time as E_C . When E_{H2} or E_{H3} occur, it is known that the inquiring device changed trains at some point in the 1.28 s preceding reception of the first packet and, therefore, the train will not change during the back-off period, giving

$$P(E_C|E_{H1}) = 0.5 \quad (9.8)$$

and

$$P(E_C|E_{H2}) = P(E_C|E_{H3}) = 0. \quad (9.9)$$

Thus, the conditional probabilities in (9.6) are

$$P(E_{S\overline{TC}}|E_{Hi}) = P(\{S_S \leq T_B\} \cap \{S_T > T_B\}), \quad (9.10)$$

$$P(E_{\overline{STC}}|E_{Hi}) = P(\{S_S > T_B\} \cap \{S_T > T_B\}), \quad (9.11)$$

$$P(E_{SC}|E_{H1}) = P(E_{ST}|E_{H1}) = 0.5P(\{S_S \leq T_B\} \cap \{S_T \leq T_B\}), \quad (9.12)$$

$$P(E_{\overline{SC}}|E_{H1}) = P(E_{\overline{ST}}|E_{H1}) = 0.5P(\{S_S > T_B\} \cap \{S_T \leq T_B\}), \quad (9.13)$$

$$P(E_{SC}|E_{H2}) = P(E_{SC}|E_{H3}) = 0, \quad (9.14)$$

and

$$P(E_{\overline{SC}}|E_{H2}) = P(E_{\overline{SC}}|E_{H3}) = 0. \quad (9.15)$$

The mutual independence of S_S and S_T results in

$$P(E_{S\overline{TC}}|E_{Hi}) = \frac{1}{1024} \sum_{n=0}^{1023} F_S(nT_{slot}) \left(1 - F_T(nT_{slot})\right) \quad (9.16)$$

where T_{slot} is the duration of one Bluetooth time slot, $625\mu s$.

Similarly,

$$P(E_{\overline{STC}}|E_{Hi}) = \frac{1}{1024} \sum_{n=0}^{1023} \left(1 - F_S(nT_{slot})\right) \left(1 - F_T(nT_{slot})\right), \quad (9.17)$$

$$P(E_{ST}|E_{Hi}) = \frac{1 - P(E_C|E_{Hi})}{1024} \sum_{n=0}^{1023} F_S(nT_{slot}) F_T(nT_{slot}), \quad (9.18)$$

and

$$P(E_{\overline{ST}}|E_{Hi}) = \frac{1 - P(E_C|E_{Hi})}{1024} \sum_{n=0}^{1023} \left(1 - F_S(nT_{slot})\right) F_T(nT_{slot}). \quad (9.19)$$

for $i = 1, 2, 3$. To derive the pdf for $T_P|E_{S\overline{TC}} \cap E_{Hi}$, $f_{T_P}(t|E_{S\overline{TC}} \cap E_{Hi})$ is convolved (*) with $f_{T_B}(t|E_{S\overline{TC}})$. Since $E_{S\overline{TC}}$ is independent of T_F , $f_{T_P}(t|E_{S\overline{TC}} \cap E_{Hi}) = f_{T_P}(t|E_{Hi})$. Therefore,

$$\begin{aligned}
f_{T_B}(t|E_{S\overline{T}\overline{C}} \cap E_{H1}) &= \frac{\frac{d}{dt}P(\{S_S \leq T_B\} \cap \{S_T > T_B\}|E_{H1})}{P(E_{S\overline{T}\overline{C}}|E_{H1})} \\
&= \frac{\delta(t - nT_{slot})}{1024} \left(\frac{F_S(nT_{slot})(1 - F_T(nT_{slot}))}{P(E_{S\overline{T}\overline{C}}|E_{H1})} \right) \quad (9.20)
\end{aligned}$$

for $n = 0, 1, 2, \dots, 1023$ and,

$$\begin{aligned}
f_{T_P}(t|E_{S\overline{T}\overline{C}} \cap E_{H1}) &= \begin{cases} \frac{1}{1.28} \int_0^t f_{T_B}(\tau|E_{S\overline{T}\overline{C}})d\tau & 0 < t \leq 639.375ms \\ \frac{1}{1.28} & 639.375ms < t \leq 1.28s \\ \frac{1}{1.28} \left(1 - \int_0^{t-1.28} f_{T_B}(\tau|E_{S\overline{T}\overline{C}})d\tau \right) & 1.28 < t \leq 1.9139375s \end{cases} \\
&= f_{T_F}(t|E_{H1}) * f_{T_B}(t|E_{S\overline{T}\overline{C}} \cap E_{H1}) \quad (9.21)
\end{aligned}$$

where $\delta(t)$ is the impulse function. The probability $P(E_{S\overline{T}\overline{C}}|E_{H1})$ is used in (9.20) since the probability distribution for $T_B|E_{S\overline{T}\overline{C}}$, is the same is the same for all E_{Hi} , $i = 1, 2, 3$. The same is true for all E_X , $X \in \{\overline{S\overline{T}\overline{C}}, \overline{S\overline{T}\overline{C}}, \overline{S\overline{T}}, \overline{S\overline{C}}, \overline{S\overline{T}}, \overline{S\overline{C}}\}$, except when E cannot occur. In those instances where E_X cannot occur, $f_{T_B}(t|E_X)$ is zero for all time. However, those terms will be driven to zero in the overall expression (9.1) by the probability of the events occurring. Thus, for brevity, $T_B|E_X$ is assumed independent of E_{Hi} .

Therefore,

$$f_{T_B}(t|E_{S\overline{T}\overline{C}}) = \frac{\delta(t - nT_{slot})}{1024} \frac{(1 - F_S(nT_{slot}))(1 - F_T(nT_{slot}))}{P(E_{S\overline{T}\overline{C}}|E_{H1})}, \quad (9.22)$$

$$f_{T_B}(t|E_{ST}) = f_{T_B}(t|E_{SC}) = \frac{\delta(t - nT_{slot})}{1024} \frac{(F_S(nT_{slot}))(F_T(nT_{slot}))}{P(E_{ST}|E_{H1})}, \quad (9.23)$$

and

$$f_{T_B}(t|E_{\overline{ST}}) = f_{T_B}(t|E_{\overline{SC}}) = \frac{\delta(t - nT_{slot})}{1024} \frac{(1 - F_S(nT_{slot}))F_T(nT_{slot})}{P(E_{\overline{ST}}|E_{H1})} \quad (9.24)$$

for $n = 0, 1, 2, \dots, 1023$.

The densities for T_P when $E_{\overline{STC}}$, E_{ST} , E_{SC} , $E_{\overline{ST}}$, and $E_{\overline{SC}}$ occur are determined by replacing $E_{\overline{STC}}$ in (8.25) with $E_{\overline{STC}}$, E_{ST} , E_{SC} , $E_{\overline{ST}}$, and $E_{\overline{SC}}$, respectively as well as shifting the distribution appropriately in time. Since $T_F|E_{H_i}$, $i = 2, 3$ are distributed identically to $T_F|E_{H1}$ but are time shifted by $1.28(i - 1)$ s,

$$f_{T_P}(t|E_X \cap E_{H_i}) = f_{T_P}(t - 1.28(i - 1)|E_X \cap E_{H1}) \quad (9.25)$$

for $X \in \{\overline{STC}, \overline{STC}, \overline{ST}, \overline{SC}, ST, SC\}$.

For example,

$$f_{T_P}(t|E_{\overline{STC}} \cap E_{H3}) = \begin{cases} \frac{1}{1.28} \int_0^{t-2.56s} f_{T_B}(\tau|E_{\overline{STC}})d\tau & 2.56 < t \leq 3.199375s \\ \frac{1}{1.28} & 3.199375 < t \leq 3.84s \\ \frac{1}{1.28} \left(1 - \int_0^{t-3.84s} f_{T_B}(\tau|E_{\overline{STC}})d\tau \right) & 3.84 < t \leq 4.479375s \end{cases} \quad (9.26)$$

9.1.3 Conditional Densities for the Inquiry Time

The inquiry time is dependent on N_2 such that

$$T_I = T_P + N_2. \quad (9.27)$$

The random integer N_2 is dependent on the relationship between the train and the scan frequency which is a function of the event E_{H_i} , $i = 1, 2, 3$.

If E_{H1} occurs, the second packet is received during the first scan window after completion of the back-off period (i.e., $N_2 = 0$) unless the scan frequency changes such that it is no longer in the train being used or the train changes such that the scan frequency no longer resides in it. If $E_{\overline{ST}} \cap E_L \cap E_{H1}$ occurs, $N_2 = 1$ since the train membership change causes the scan frequency to be outside the train during the first scan window after the back-off period. Between the first and second window, however, it is known that the inquiring device will change trains. Likewise, if $E_{\overline{STC}} \cap E_L \cap E_{H1}$ occurs, $N_2 = 1$ if the train membership change also includes a train change between the first and second windows. This happens with probability 0.5. Otherwise, $N_2 = 2$. Finally, $N_2 = 2$ if $E_{\overline{SC}} \cap E_{H1}$ or $E_{SC} \cap \overline{E_L} \cap E_{H1}$ occurs. The train changes such that the scan frequency is no longer in the train and does not fall within the train again until the train change between the second and third windows. This yields

$$\begin{aligned}
f_{T_I}(t|E_{H1} \cap \{N_2 = 0\}) &= \left(\frac{1}{P(E_{H1} \cap \{N_2 = 0\})} \right) \times \\
&\left(f_{T_P}(t|E_{\overline{STC}})P(E_{\overline{STC}}|E_{H1})P(E_{H1}) + f_{T_P}(t|E_{\overline{ST}})P(E_{\overline{ST}}|E_{H1})P(\overline{E_L})P(E_{H1}) + \right. \\
&f_{T_P}(t|E_{ST})P(E_{ST}|E_{H1})P(E_{H1}) + f_{T_P}(t|E_{\overline{ST}})P(E_{\overline{ST}}|E_{H1})P(\overline{E_L})P(E_{H1}) + \\
&\left. f_{T_P}(t|E_{SC})P(E_{SC}|E_{H1})P(E_L)P(E_{H1}) \right), \tag{9.28}
\end{aligned}$$

$$\begin{aligned}
f_{T_I}(t|E_{H1} \cap \{N_2 = 1\}) &= \left(\frac{P(E_L)P(E_{H1})}{P(E_{H1} \cap \{N_2 = 1\})} \right) \times \tag{9.29} \\
&\left(0.5f_{T_P}(t - 1.28|E_{\overline{STC}})P(E_{\overline{STC}}|E_{H1}) + f_{T_P}(t - 1.28|E_{\overline{ST}})P(E_{\overline{ST}}|E_{H1}) \right),
\end{aligned}$$

and

$$\begin{aligned}
f_{T_I}(t|E_{H1} \cap \{N_2 = 2\}) &= \left(\frac{P(E_{H1})}{P(E_{H1} \cap \{N_2 = 0\})} \right) \left(f_{T_P}(t - 2.56|E_{\bar{S}C})P(E_{\bar{S}C}|E_{H1}) + \right. \\
&\quad \left. 0.5f_{T_P}(t - 2.56|E_{\bar{S}T\bar{C}})P(E_{\bar{S}T\bar{C}}|E_{H1})P(E_L) + f_{T_P}(t - 2.56|E_{SC})P(E_{SC}|E_{H1})P(\bar{E}_L) \right)
\end{aligned} \tag{9.30}$$

where

$$\begin{aligned}
P(E_{H1} \cap \{N_2 = 0\}) &= P(E_{H1}) \left(P(E_{SC}|E_{H1})P(E_L) + \right. \\
&\quad \left. P(E_{\bar{S}T\bar{C}}|E_{H1}) + P(E_{\bar{S}T\bar{C}}|E_{H1})P(\bar{E}_L) + P(E_{ST}|E_{H1}) + P(E_{\bar{S}T}|E_{H1})P(\bar{E}_L) \right),
\end{aligned} \tag{9.31}$$

$$P(E_{H1} \cap \{N_2 = 1\}) = P(E_{H1})P(E_L) \left(0.5P(E_{\bar{S}T\bar{C}}|E_{H1}) + P(E_{\bar{S}T}|E_{H1}) \right), \tag{9.32}$$

and

$$\begin{aligned}
P(E_{H1} \cap \{N_2 = 2\}) &= P(E_{H1}) \left(P(E_{SC}|E_{H1})P(\bar{E}_L) + \right. \\
&\quad \left. 0.5P(E_{\bar{S}T\bar{C}}|E_{H1})P(E_L) + P(E_{\bar{S}C}|E_{H1}) \right).
\end{aligned} \tag{9.33}$$

If E_{H2} occurs, the situation is different. The train being used cannot change during the back-off period since it changed in the 1.28 s prior to the first packet being received. Likewise, if the train changes membership during the back-off period, the train being used changes between the first and second scan window. Otherwise, the train being used changes between the second and third scan window.

Thus,

$$\begin{aligned}
f_{T_I}(t|E_{H_2} \cap \{N_2 = 0\}) &= \left(\frac{1}{P(E_{H_2} \cap \{N_2 = 0\})} \right) \times \\
&\left(f_{T_P}(t - 1.28|E_{\overline{STC}})P(E_{\overline{STC}}|E_{H_2})P(E_{H_2}) + \right. \\
&f_{T_P}(t - 1.28|E_{\overline{STC}})P(E_{\overline{STC}}|E_{H_2})P(\overline{E_L})P(E_{H_2}) + \\
&f_{T_P}(t - 1.28|E_{ST})P(E_{ST}|E_{H_2})P(E_{H_2}) + \\
&\left. f_{T_P}(t - 1.28|E_{\overline{ST}})P(E_{\overline{ST}}|E_{H_2})P(\overline{E_L})P(E_{H_2}) \right), \tag{9.34}
\end{aligned}$$

$$f_{T_I}(t|E_{H_2} \cap \{N_2 = 1\}) = f_{T_P}(t - 2.56|E_{\overline{ST}}), \tag{9.35}$$

and

$$f_{T_I}(t|E_{H_1} \cap \{N_2 = 2\}) = f_{T_P}(t - 2.56|E_{\overline{STC}}) \tag{9.36}$$

where

$$\begin{aligned}
P(E_{H_2} \cap \{N_2 = 0\}) &= P(E_{H_2}) \left(P(E_{\overline{STC}}|E_{H_2}) + P(E_{\overline{STC}}|E_{H_2})P(\overline{E_L}) + \right. \\
&\left. P(E_{ST}|E_{H_2}) + P(E_{\overline{ST}}|E_{H_2})P(\overline{E_L}) \right), \tag{9.37}
\end{aligned}$$

$$P(E_{H_2} \cap N_2 = 1) = P(E_{\overline{ST}}|E_{H_2})P(E_L)P(E_{H_2}), \tag{9.38}$$

and

$$P(E_{H_2} \cap N_2 = 0) = P(E_{\overline{STC}}|E_{H_1})P(E_L)P(E_{H_2}). \tag{9.39}$$

Since the conditions under which E_{H_3} occurs are identical to those under which E_{H_2} occurs (except for a 1.28 s time shift),

$$f_{T_I}(t|E_{H_3} \cap \{N_2 = j\}) = f_{T_I}(t - 1.28|E_{H_2} \cap \{N_2 = j\}) \tag{9.40}$$

for $j = 0, 1, 2$.

Substituting all required probabilities and conditional probabilities into densities (8.31)

$$f_{T_I}(t) = \sum_{i=1}^4 \sum_{j=0}^5 f_{T_I}(t|E_{Hi} \cap \{N_2 = j\})P(\{N_2 = j\}|E_{Hi})P(E_{Hi}). \quad (9.41)$$

The resultant density can be determined and is shown in Figure 9.1.

The simulation model was modified to change inquiry trains when the inquiring device's CLK_{13} bit changed, producing the pdf also shown in Figure 9.1. The simulation was replicated 4,000 times with 1500 inquiries per replication and random address/clock values for each replication. The difference between the analytical and simulated mean density is shown in Figure 9.2.

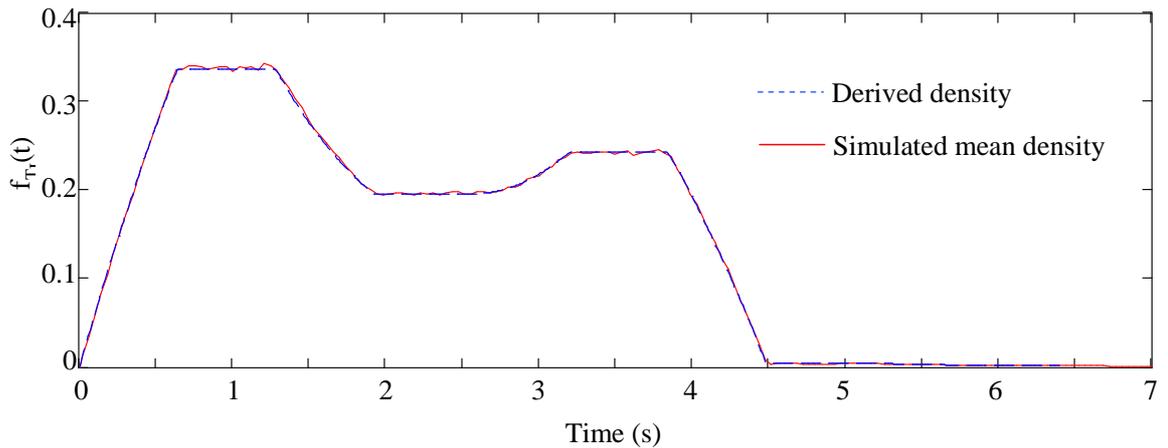


Figure 9.1: Inquiry time pdf, $f_{T_I}(t)$, using the CLK-driven inquiry.

9.2 Results

The simulated mean inquiry time using the CLK-driven inquiry is 2.112 s compared to 2.292 s using the inquiry substate as specified in the standard [Blu03]. The mean inquiry time for the measured devices is 2.221 s [KaL01].

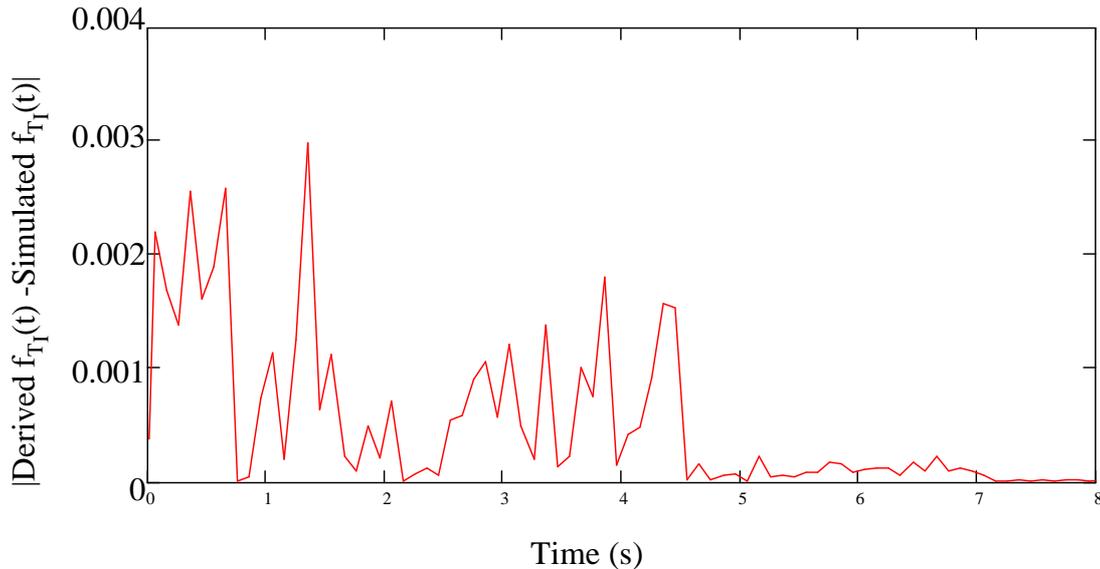


Figure 9.2: Difference in $f_{T_I}(t)$ between derived and simulated mean pdf estimates using CLK-driven inquiry.

The measured distribution closely matches the derived distribution until $t = 3$ s, after which it diverges slightly. However, the null hypothesis that the measured distribution is statistically equivalent to the derived distribution is not rejected at the 0.05 level for either the Kolmogorov-Smirnov ($KS = 0.909$, $Critical = 1.358$) or Cramér-von Mises ($CV = 0.08$, $Critical = 0.461$) tests. However, the tests are somewhat contrived since the derived distribution is not conditioned upon the address/clock values for the device pairs whereas the measured distribution is conditioned on a specific address/clock pair. Although not precisely matching the derived CDF, the measured CDF lies well within the range of CDFs generated by the simulation as seen in Figure 9.3. Since the measured experiment represents a single replication of the experiment from which the simulated distributions were derived, it is likely that the measured devices used the CLK-driven inquiry process. For comparison, the measured data and the range of CDFs generated when following v1.1 of the specification is shown in Figure 9.4.

Minimizing the inquiry time is important for several reasons. The inquiry substate uses twice as much power as the connection state [KaL01]. Remaining in the inquiry state longer than necessary reduces the potential to be in the connection state

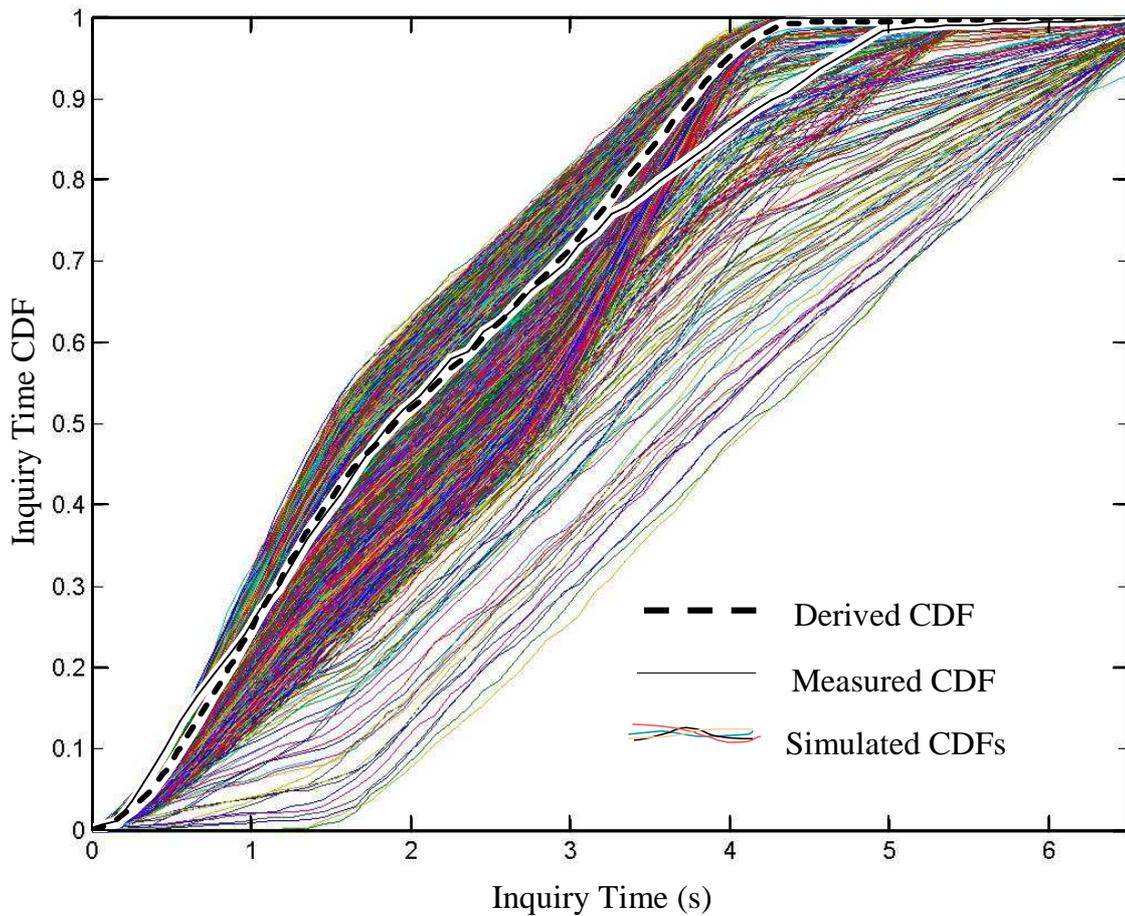


Figure 9.3: Measured inquiry times compared to derived and simulated inquiry times using the CLK-driven inquiry substate.

and, therefore, network throughput. Additionally, nodes in the inquiry state can cause significant interference to the neighboring piconets. The CLK-driven process simplifies the implementation while maintaining compatibility with devices more strictly adhering to the standard. The time spent in the inquiry substate must be a multiple of 1.28 s [Blu03]. Using the specification and an inquiry state duration of 5.12 seconds (half of that recommended in the specification), 99.0% of the packets are received assuming a perfect channel. By comparison, 99.6% of the packets are received in 5.12 s using the CLK-driven inquiry substate. Thus, an inquiry duration of 5.12 s is sufficient using the inquiry process in the standard but is more successful

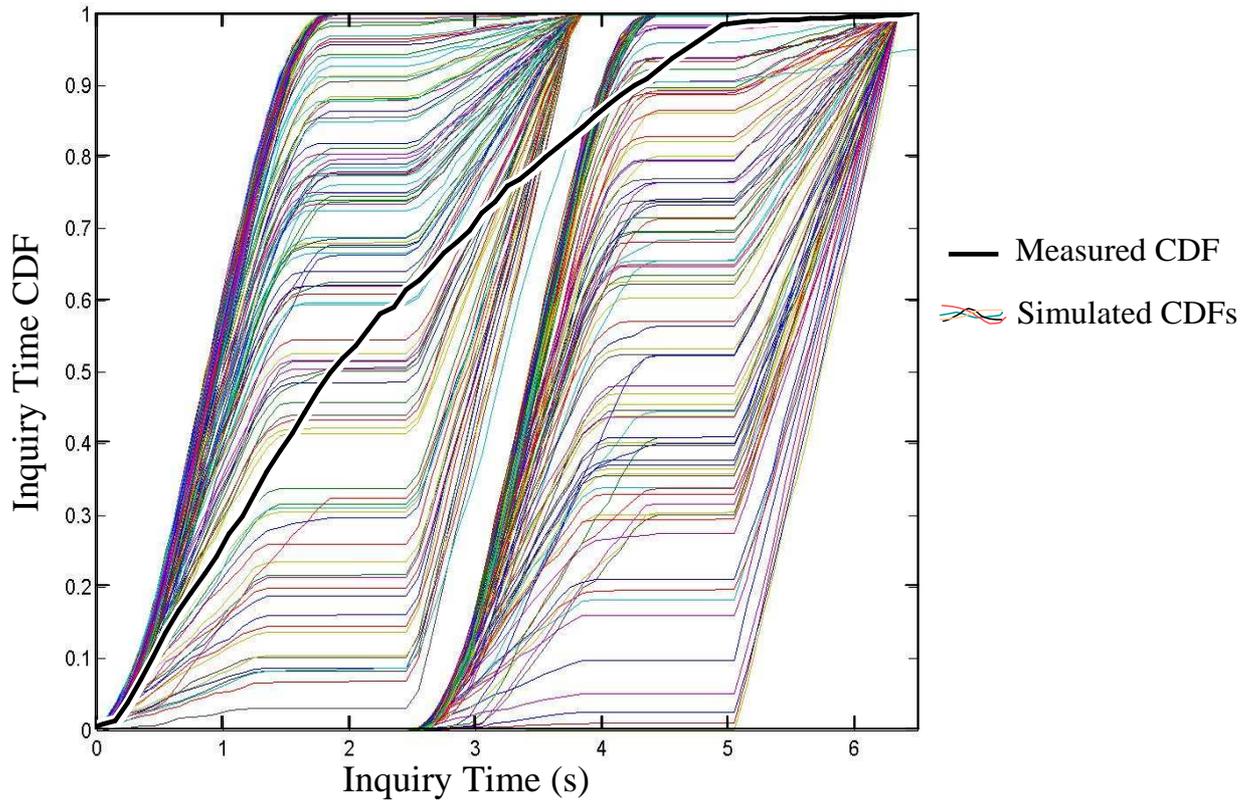


Figure 9.4: Measured inquiry times compared to simulated inquiry times using BT specification v1.1.

when using the CLK-driven inquiry. However, the most important advantage of the CLK-driven inquiry is that it eliminates the need for a separate counter, register, and supporting algorithm to control train use. No hardware/software is needed for train selection since it is controlled by a CLK bit.

Although the empirical inquiry data was not collected from a perfect channel, it is unlikely that noise had a significant impact on the inquiry time since the two devices were only one meter apart. However, if a noisy environment is detected, noise effects can be countered by increasing the scan window length. By doubling the scan window length, the inquiry time distribution remains almost identical. Since the train is simply repeated 128 times between membership changes, it allows a scanning node a second opportunity to receive the inquiry packet if receipt of the first transmission is disrupted by noise. If the noise is due to packets from other

piconets being transmitted on the same frequency or sporadic background noise, the noise may not be present at the second attempt to receive the inquiry packet. This delays packet reception by only 10 ms in a distribution spread across multiple seconds. This almost doubles the time that a scanning node spends in scan windows, from 0.88% to 1.66%, but has a relatively small impact on node throughput.

9.3 *Clk-Driven Inquiry with Standard Inquiry Scan*

As in Chapter VIII,

$$f_{T_I}(t) = \sum_{i=1}^5 f_{T_F}(t|E_{H_i})P(E_{H_i}). \quad (9.42)$$

as shown in Figure 9.5.

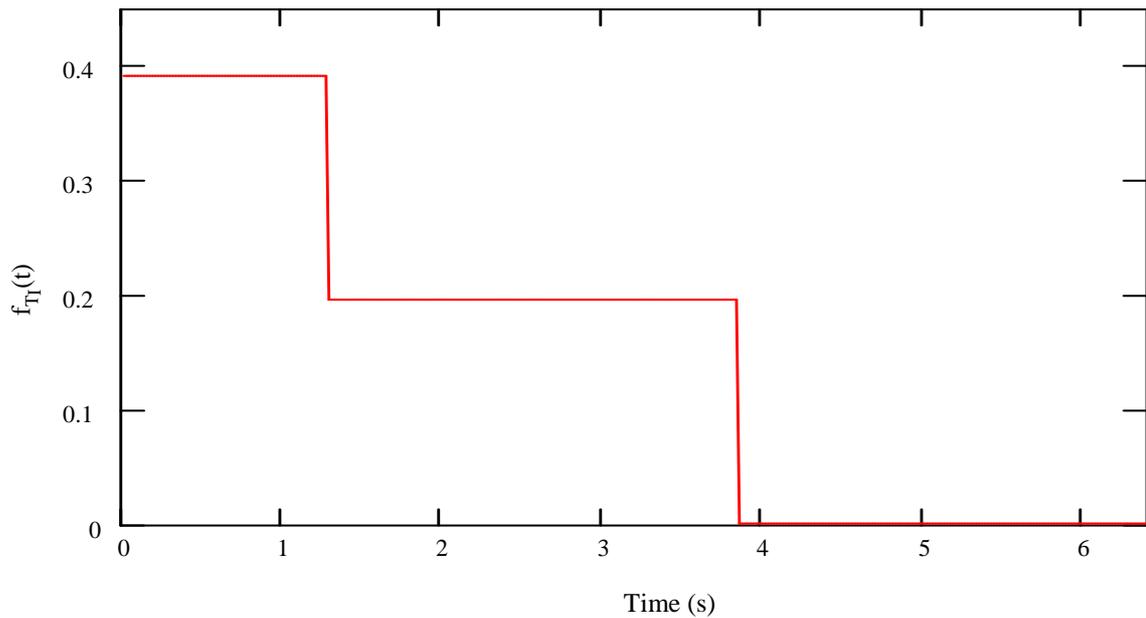


Figure 9.5: Inquiry time for the standard inquiry scan using CLK-Driven Inquiry, $f_{T_I}(t)$.

9.4 *Summary*

The BT inquiry time density function is the result of a complex and dynamic relationship between the inquiry train and the scan frequency of devices in the inquiry scan substate. The mean inquiry time from the simulation closely matches that derived in Chapter VIII for the inquiry time when specification v1.1 is followed. However, it is evident from experimental data that some implementations do not strictly adhere to the specification [KaL01]. By using the thirteenth CLK bit to determine the inquiry train being used, the inquiry substate implementation can be simplified while retaining compatibility with the specification and reducing actual inquiry time. Although the apparent use by the measured devices shows the CLK-driven inquiry is not a new idea, the probability distribution for its inquiry time gives insight into the effects of the change. The simulated and analytical inquiry time densities for the CLK-driven inquiry closely follow the probability density for the collected data.

Using the metric characterizations derived in the previous five chapters, the outreach methods are evaluated and compared in Chapter X.

X. Network Models

In this chapter, the proposed outreach methods presented in Section 4.4.3 is reduced to three using general observations from the preceding chapters. The performance metrics of the three remaining methods are compared, resulting in a final recommendation for a scatternet outreach method. The most effective outreach methods include those directing all slave devices to periodically scan for new devices. Devices in the scatternet do not inquire unless specifically directed.

10.1 Outreach Method Elimination

Using the performance metric characterizations developed in the Chapters V-IX, several of the initially proposed outreach methods are eliminated as clearly inferior. Since a minimum of 3.84 s is required for adequate discovery using the standard inquiry scan, master devices should not enter the inquiry substate unless directed by the user. Traffic within the piconet stops when the master is in the inquiry substate. Furthermore, with multiple inquirers, a 3.84 s inquiry substate dwell time is not effective. Additionally, when all slaves scan regularly, there is little value in the master scanning as well. When a slave device discovers, or is discovered by, an arriving node, that node can be added to the scatternet and placed into the scatternet structure by the process governing scatternet organization.

Additionally, as a single inquiring device has little impact on goodput ($<1.5\%$), attempting to detect an arriving inquiring device to direct it to cease its inquiring is of little benefit. The overhead needed for this operation also makes it undesirable. Thus, the candidate outreach methods of Section 4.4.3 are reduced from

1. Currently proposed method with all nodes alternating between inquiry and inquiry scan substates; arriving nodes are discovered in either the inquiry or inquiry scan substate; disjoint scatternets quickly find each other,

2. Limit the inquiry substate to slaves only, all nodes scan regularly,
3. Limit the inquiry substate to slaves only, only slaves scan regularly,
4. Limit the inquiry substate to non-bridge slaves only, all nodes scan regularly,
5. Limit the inquiry substate to non-bridge slaves only, only slaves scan regularly,
6. Nodes in the scatternet rarely enter inquiry substate, all nodes scan regularly,
7. Nodes in the scatternet rarely enter inquiry substate, only slaves scan regularly; arriving nodes must inquire, to
 - a) Limit the inquiry substate to slaves only, only slaves scan regularly, or Inquiring Slave Outreach Method (ISOM),
 - b) Limit the inquiry substate to non-bridge slaves only, only slaves scan regularly, or Non-bridge Inquiring Slave Outreach Method (NISOM), or
 - c) Nodes in the scatternet rarely enter inquiry substate, only slaves scan regularly; arriving nodes must inquire, or Scan-Based Outreach Method (SBOM).

10.2 Simulation Model

A MATLAB[®] simulation model was developed to simulate the communication within a scatternet with the various outreach methods. As mentioned in Section 4.3, the simulated network consisted of 5 piconets, all interconnected, as shown in Figure 10.1. No network configuration is consistently used in the literature. The network was chosen as a representative office or classroom scatternet configuration and is similar in size and piconets to those used in [LaS01] [SBT01]. The network goodput, PER, mean packet delay, buffer size, and packet arrival distribution were collected. Additionally, discovery time for arriving nodes was noted. Since it is beneficial for a node to be discovered by multiple piconets in the scatternet to quickly determine the most appropriate piconet to join, the discovery time for the first three unique piconets was recorded. If the arriving node was discovered by a bridge node,

it was assumed to have been discovered by both piconets the bridge resides in. For example, if a bridge node between Piconets A and B first discovered an arriving node, the second discover time was recorded when a node from piconets C, D, or E discovered the arriving node.

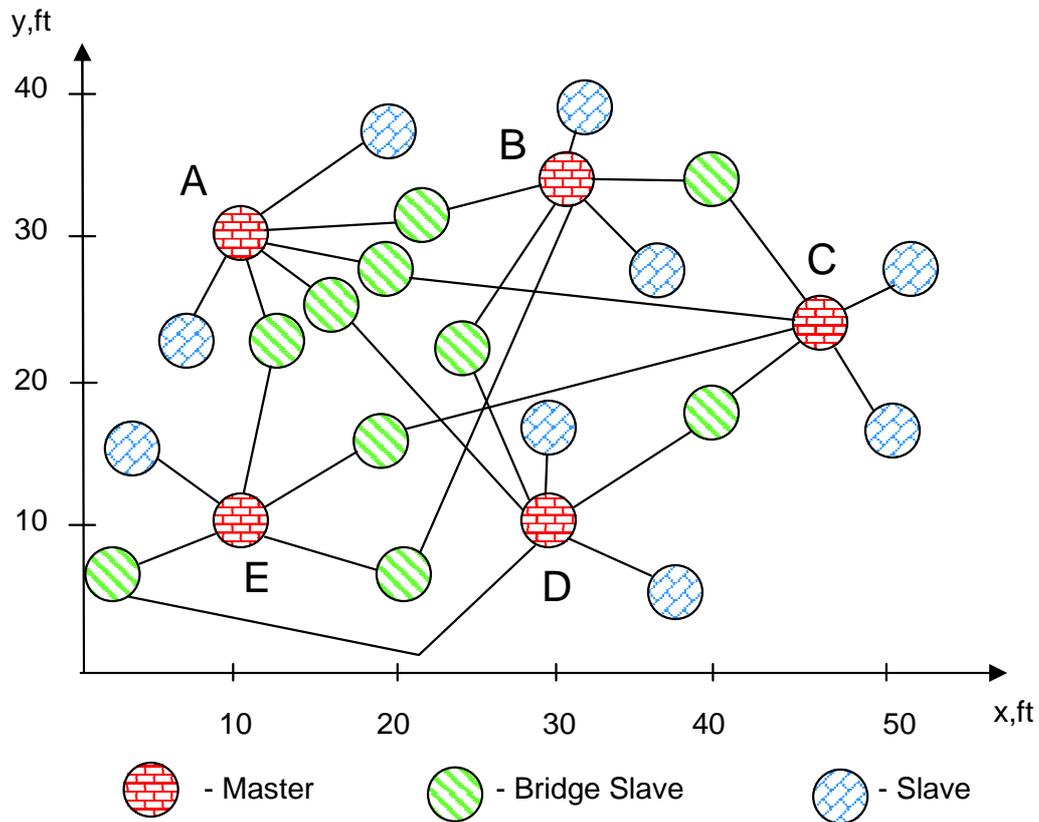


Figure 10.1: Benchmark scatternet configuration used for outreach comparison.

Additionally:

- All nodes are within range of one another
- No collision avoidance is implemented in the simulation
- Bridge nodes dwell in each piconet for a time uniformly distributed on $[5, 25]$ MSTSs
- The standard inquiry scan from specification v1.2 is used
- The inquiry substate duration is 3.84 seconds

- Each inquiring node enters the inquiry substate at a time uniformly distributed on $[0,40)$ seconds
- Packets are transmitted by the master with the following priority:
 1. Oldest buffered packet destined for a slave in the piconet that hasn't been contacted in T_{poll} . (A packet is generated if no packets are buffered for that node),
 2. Oldest buffered packet to a slave currently in the piconet that indicated packets in it's buffer destined for the master,
 3. Oldest buffered packet to a slave currently in the piconet without indication of buffered packets destined for the master, and finally
 4. Generated packet destined for a slave in the piconet.
- The oldest packets in the buffer are sent to the master of the piconet in which the slave resides. If none are buffered, a packet is generated.
- Master packets are single-slot and are generated with the following distribution:
 - 30% destined for a node outside the current piconet
 - 70% destined for a slave node with the current piconet
- Slave packets are single-slot and are generated with the following distribution:
 - 30% destined for a node outside the current piconet
 - 49% destined for a slave node within the current piconet
 - 21% destined for a the master of the current piconet
- Packet that are not acknowledged are re-transmitted
- A device arrives to the scatternet at a time uniformly distributed on $[6.52, 12.5]$ seconds, allowing the scatternet to stabilize before discovery is attempted. If no devices in the scatternet enter the inquiry substate, the arriving node enters the inquiry substate upon arrival.

A total of 20 simulation replications were conducted for each outreach method and data was collected for 48,000 MSTs, or 60 s.

10.3 Method Comparison

The three outreach methods (i.e., ISOM, NISOM, and SBOM) each have strengths and weaknesses. ISOM and NISOM are able to discover devices that only scan through inquiring. The scatternet can find other neighboring, but disjoint, scatternets or devices unaware that a scatternet is operating in the vicinity. However, devices expend far more energy than those using SBOM. Additionally, the inquiring devices cause additional interference (when collision avoidance is not used), will generally repeatedly discover nodes already in the scatternet, and remain out of contact with the piconet for the inquiry substate duration. ISOM may discover devices more quickly as more nodes are in the inquiry substate, but throughput and the mean delay are adversely affected since bridge nodes are in neither piconet when inquiring. SBOM, on the other hand, is much more power-efficient and does not have long periods where some nodes are inaccessible, but arriving nodes must inquire and disjoint scatternets may be slow to discover each other. The advantages SBOM provides in mean delay, buffer size, and discovery time make it a superior method over ISOM or NISOM.

10.3.1 Goodput Measure

SBOM produces the highest average goodput per piconet as shown in Figure 10.2. The goodput for NISOM is affected by the packet collisions with inquiring devices as well as with the inter-piconet collisions that also affected SBOM's performance. ISOM's goodput is affected to an even greater degree as even more devices are inquiring than in NISOM. If a collision avoidance method is used by inquiring devices to avoid inter-piconet interference, the goodput for the three outreach methods is expected to be statistically equivalent since masters do not leave the connection

state. However, the discovery time would increase for NISOM and ISOM. Also, using SBOM, the scatternet would experience unavoidable interference from an arriving inquiring node which scatternets using ISOM and NISOM will not experience, since arriving node need only scan. The arriving device cannot avoid the piconet packets since it does not have the address/clock information from the masters in the scatternet. Since the impact of the collisions is well understood, collision avoidance was not used in the simulation.

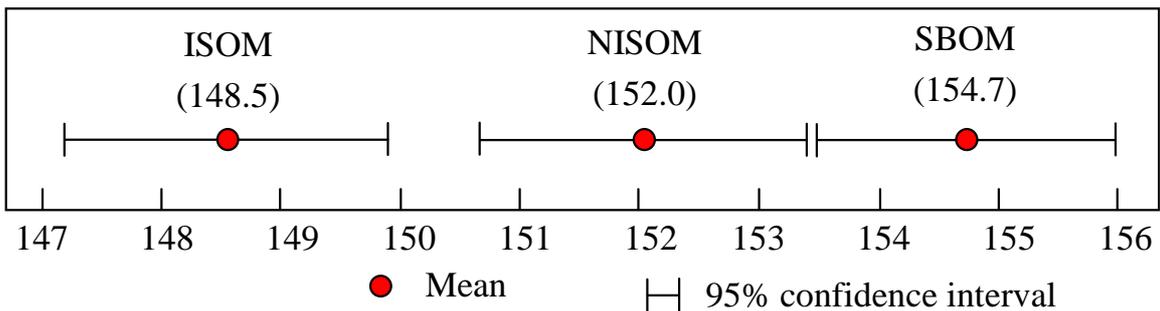


Figure 10.2: Average piconet goodput (kbps).

10.3.2 PER

As with goodput, the PER difference between outreach methods can be eliminated by using collision avoidance except for that cause by arriving inquiring nodes. Without collision avoidance, SBOM has the lowest PER, followed by NISOM and ISOM for both direct and indirect packet disruptions (See Chapter V) as shown in Figure 10.3. Although the PERs are not statistically different, the mean PER is consistent with the goodput data in Figure 10.2.

10.3.3 Mean Delay

Since delivery of packets to nodes in the inquiry substate is impossible, the mean delay for networks using the ISOM or NISOM is greater than in SBOM. Additionally, the mean delay for scatternets using ISOM is significantly higher as bridge nodes in the inquiry substate become a bottle neck for traffic between piconets. Flex-

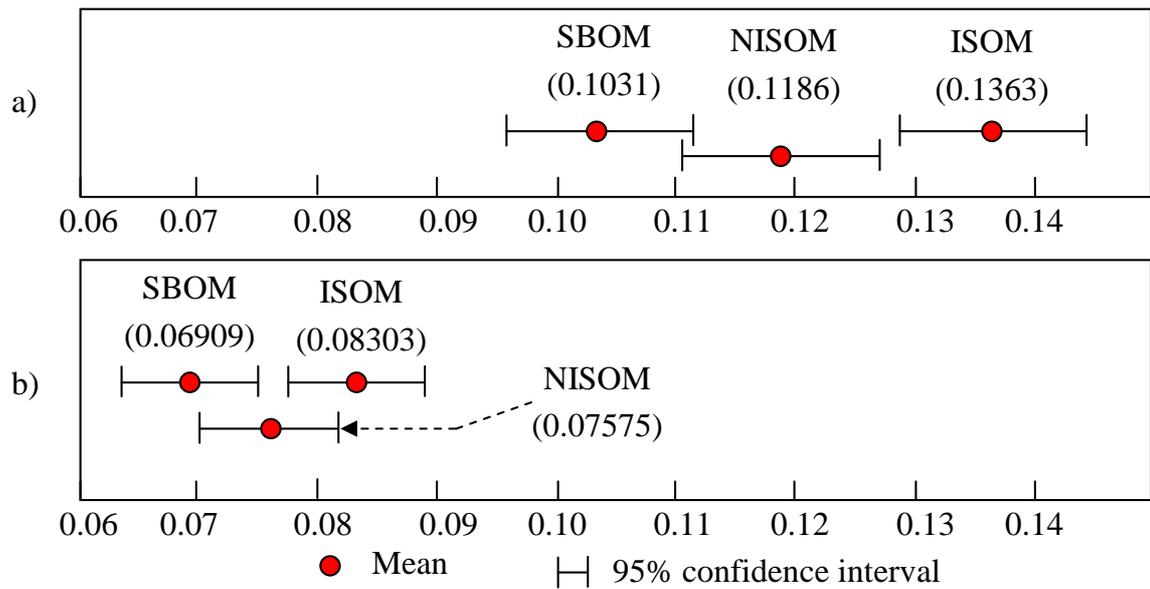


Figure 10.3: Packet Error Rate due to a) Direct Collision b) Packet repeated due to Indirect Collision.

ible routing using alternate paths may alleviate this delay, but was not incorporated into the simulation. Additionally, dynamic bridge assignment, or assigning bridge duties to a different node while the designated bridge enter the inquiry substate adds a level of complexity to the system, but may also significantly reduce mean delay. Since no devices in the scatternet are unavailable due to inquiry duties when using the SBOM, its mean delay is a third of NISOMs and less than a tenth of ISOM's as shown in Figure 10.4.

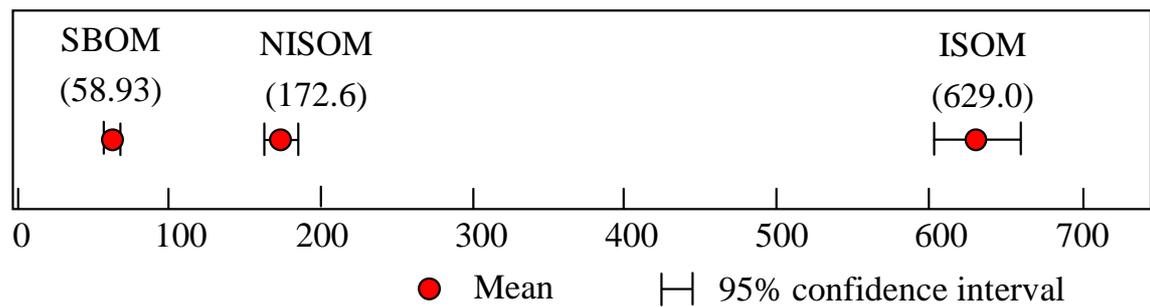


Figure 10.4: Mean packet delay (ms).

10.3.4 Buffer Size

The buffer size of a node is dependent on its role in the scatternet. Non-bridge slave devices rarely have any buffered packets since they only receive packets destined for them. In a real system, the most recently transmitted packet must always be buffered until an ACK is received. In the simulation, packets were buffered only when they were re-transmitted and therefore the buffer size never exceeded one packet. Thus, the mean buffer size is not statistically different at the 95% confidence level as shown in Figure 10.5. Due to the PER, NISOM causes packets to be buffered more often than SBOM, but not as often as ISOM. To compare this to the expected buffer size for a real system, the specified packet inter-arrival time must be compared to the packet inter-arrival time for the saturated system presented in Section 10.4.

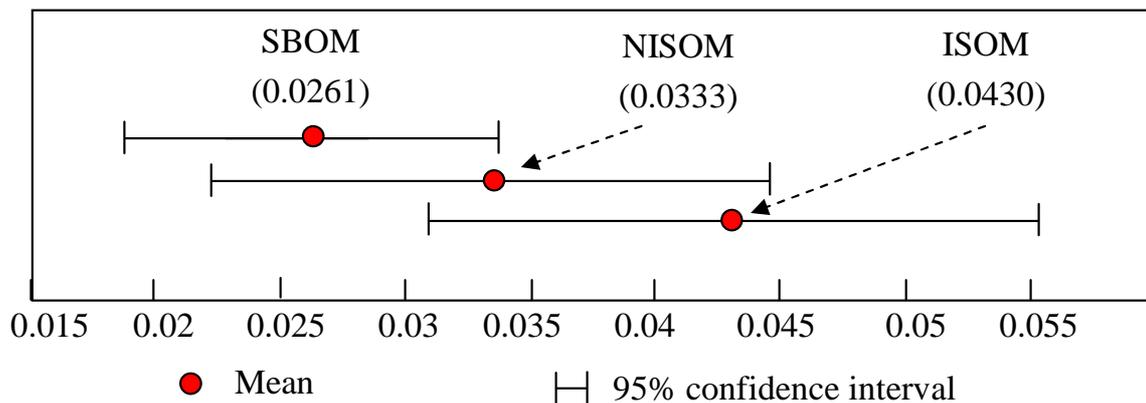


Figure 10.5: Mean buffer size for non-bridge slaves.

Bridge nodes not only buffer packets which must be re-transmitted, but also packets destined for the bridged piconet. Again, the mean buffer size is not statistically different between outreach methods since the buffered packets differ primarily due to the PER differences shown in Figure 10.6a. The buffer size is larger since a bridge nodes may be temporarily starved by the master when attempting to empty its buffer of older packets to a node that has been inquiring. Thus, the bridge buffers packets when in one piconet and is prevented from removing them from its buffer

in the other piconet. Thus, the maximum buffer size is statistically different at the 95% confidence level as shown in Figure 10.6b.

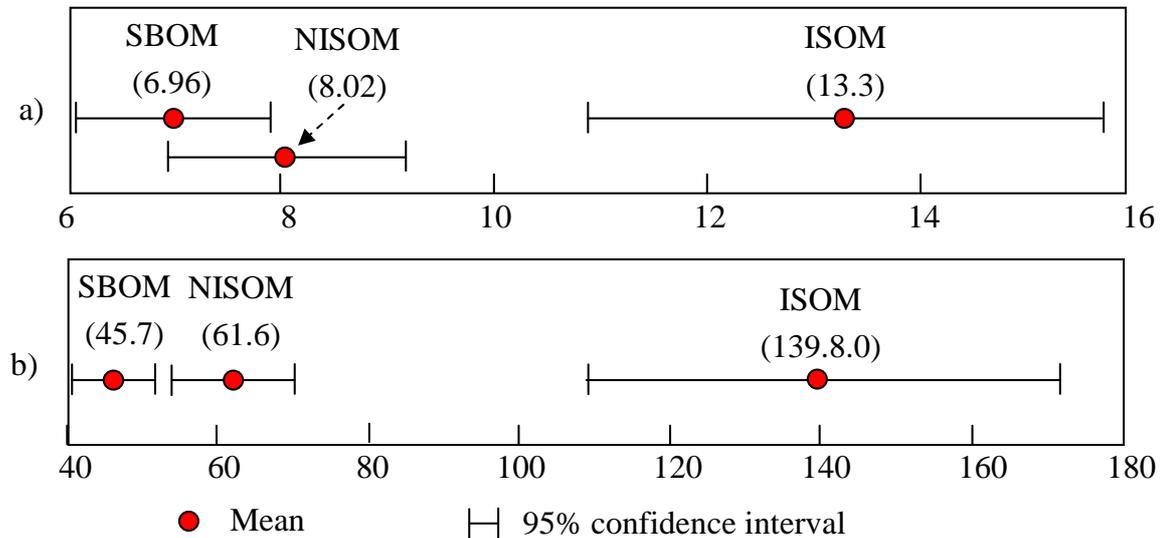


Figure 10.6: Buffer size for bridge slaves a) average b) maximum.

The buffer size of the master nodes, in contrast, depends on the outreach method. Since some slave nodes are unavailable in NISOM while the nodes inquire for 3.84 s, packets destined for the node must be buffered. When ISOM is used, bridge nodes also inquire and all packets destined for the bridge nodes, as well as the bridged piconet, must be buffered by piconets on both sides of the bridge while the bridge node inquires. A time-series history of a master node's buffer size is shown in Figure 10.7a. When using ISOM, it seems that the buffer size includes the same packets buffered when using NISOM as well as additional packets as expected. Note that the buffer remains small using SBOM since nodes are unavailable for only 11.25 ms intervals when the slave node enter the inquiry scan substate. The average buffer size of the master devices in the simulation is shown in Figure 10.7b.

The maximum and mean buffer size for each of the outreach methods is statistically different at the 95% confidence level as expected and shown in Figure 10.8. When using NISOM, the master nodes require a much larger buffer than SBOM, but much less than ISOM.

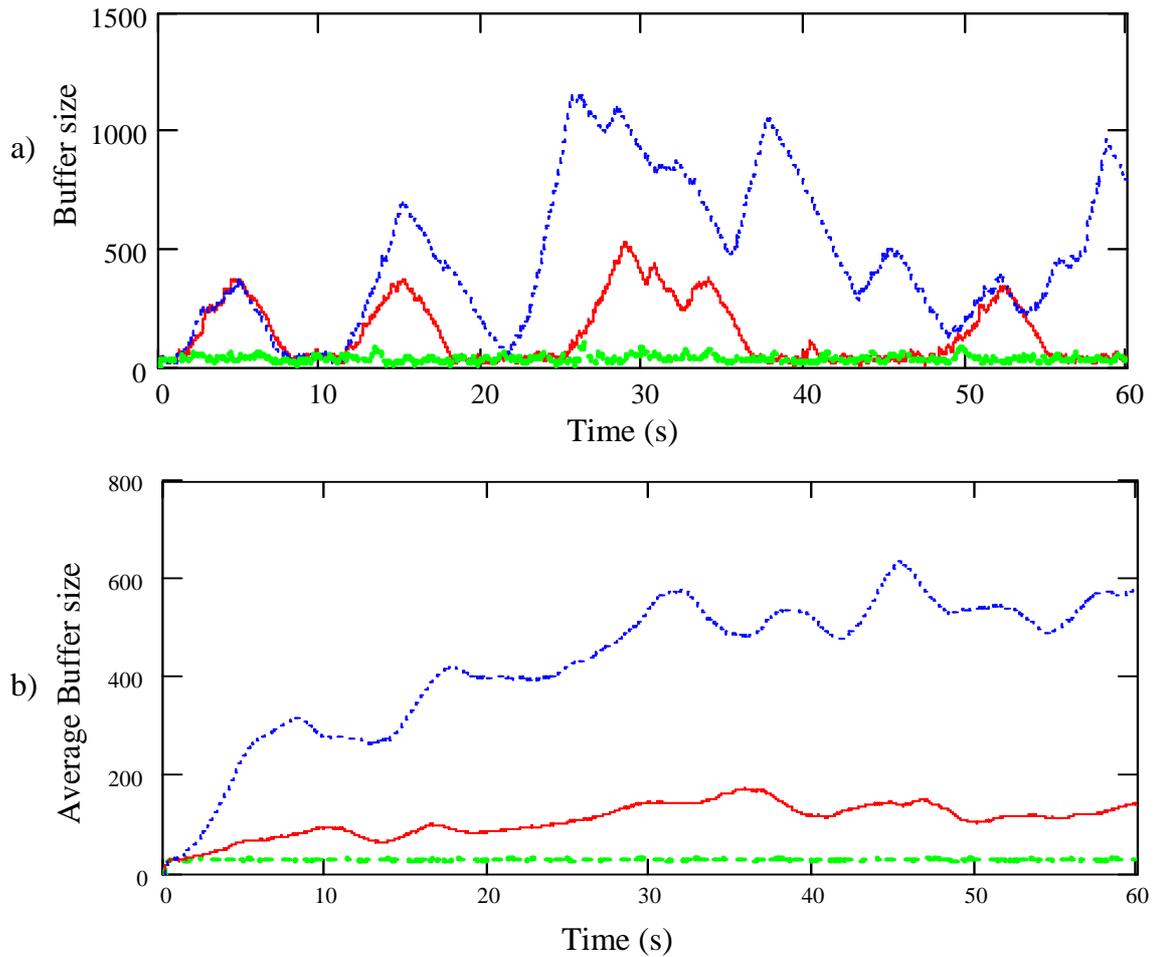


Figure 10.7: Buffer size for master nodes a) a single master node during a simulation run b) all master nodes across all simulation runs.

10.3.5 Discovery Time

One of the most important performance metrics of an outreach method is discovery time. Although discovery time for the three outreach methods is similar when an arriving node inquires (since the slave nodes scan in all three), the advantage of NISOM and ISOM is arriving nodes are not required to inquire. Thus, the discovery time for NISOM and ISOM assumes an arriving node does not inquire, but only scans. The arriving device may need to be discovered by all piconets within range for the scatternet organization algorithm to restructure the scatternet to include the new device. Thus, the time required for different piconets to discover the arriving

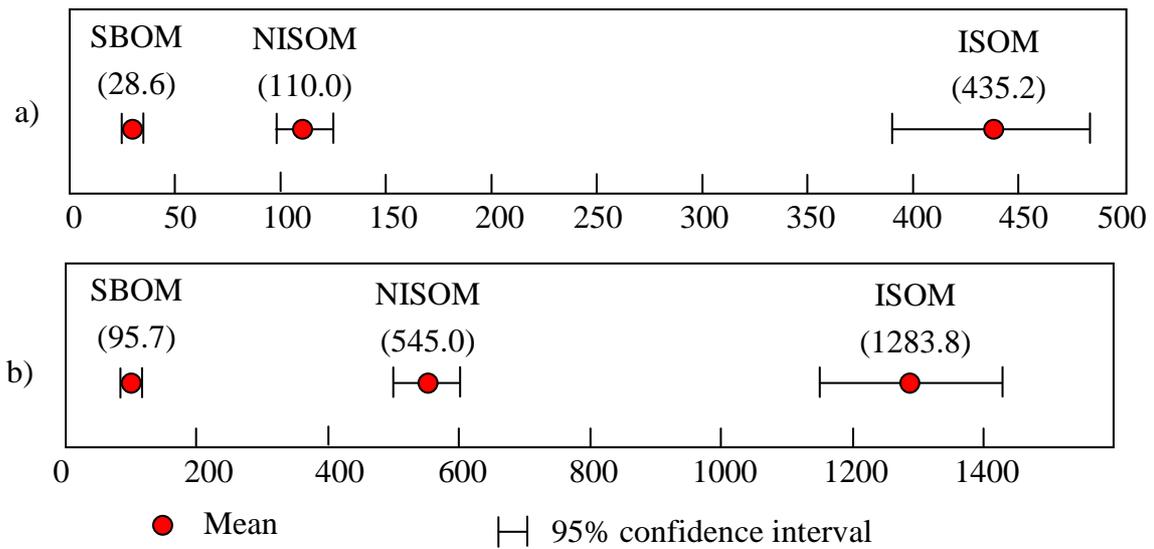


Figure 10.8: Buffer size for master nodes a) average b) maximum.

mode is presented. If the arriving device is discovered by a bridge node, it is assumed that both piconets know of the arriving device and the discovery time by the next piconet is discovery by neither of those piconets.

Since each of the nodes scan every 1.28 s and the train scanned by each device is independent, the arriving device is discovered quickly using SBOM as shown in Figures 10.9a-10.9c. The discovery time by the third unique piconet has a mean of 505 ms as shown in Figure 10.9c. This is statistically less than the discovery time for the NISOM and ISOM at the 95% confidence level. The discovery time for ISOM and NISOM is greater since devices enter the inquiry state on a time uniformly distributed on $[0, 40]$ s. The discovery time for NISOM is statistically greater at the 95% confidence level than that for ISOM after the first discovery since more devices inquire with ISOM.

10.4 Packet Generation Time pmf

Although not useful as a criteria for selecting an outreach method, packet generation time distributions derived from the simulation under saturation conditions is

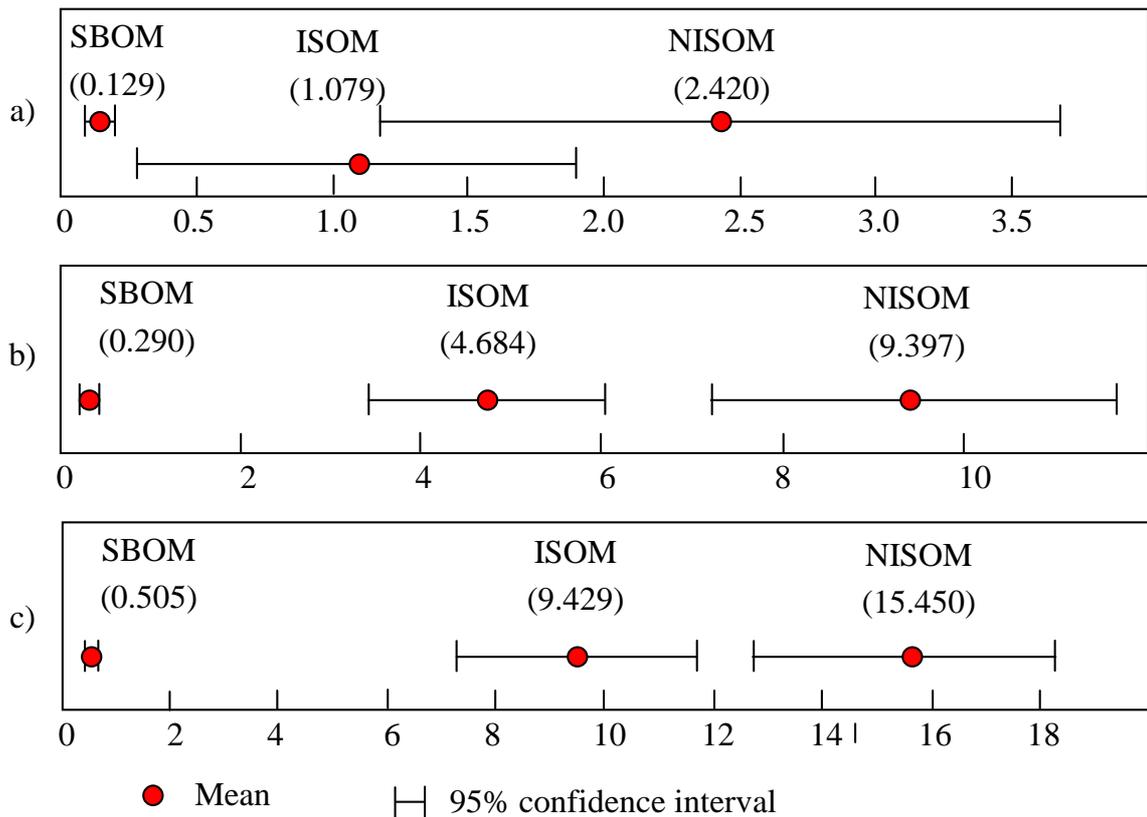


Figure 10.9: Discovery time by a) first piconet b) second piconet c) third piconet.

useful for future research. The packet generation time pmf is dependent on the type of node (i.e., non-bridge slave (NB), slave, master), the proportion of packets leaving the piconet, the proportion of packets destined for the master, and the balance of bridge/non-bridge slave nodes. If few packets leave the piconet, the bridge node resemble NB nodes and have similar packet generation distributions. However, if many packets are passed between piconets, the bridge nodes receive more attention from the master since a bridge accepts and passes packets between piconets and usually has buffered packets, placing the bridge node in higher priority than NB nodes. NB nodes only generate packets every T_{poll} or after receiving a packet. In piconets with several NB nodes, a packet transmitted by the master is more likely to go to a NB node than in nodes with few NB nodes, but the NB nodes must share the ‘excess’ MSTs when the bridge nodes do not have buffered data to transmit to the master.

With few NB nodes, even though the ‘excess’ MSTS must be divided between fewer NB nodes, it is less likely that a packet transmitted by the master is destined for a NB. Thus, if a piconet contains one NB node, the NB node will transmit less than three times as many packets as each NB node in a piconet with three NB nodes. The fewer bridge nodes in a piconet, the more ‘excess’ MSTSs are available for the NB nodes. In real scenarios, slave devices may get equal priority with bridge nodes since they generate and buffer packets independent of the transmission schedule.

Additionally, if most packets within a piconet are destined for the master, the master is able to generate more packets rather than just relaying packets to other nodes. Likewise, if few packets leave the piconet, the bridge nodes are not as busy relaying packets between piconets and have more opportunities to generate packets.

When a slave node is out of the piconet, either in another piconet or the inquiry/inquiry scan substates, the master buffers packets for the node. Thus, when the node returns to the piconet, it may receive several packets in consecutive MSTSs and be able to generate packets in subsequent MSTSs after quickly emptying its buffer.

10.4.1 Non-bridge slave nodes

Although the average rate of packet generation of the outreach methods is statistically similar at the 95% confidence level, non-bridge slaves using NISOM transmit fewer packets since no data packets are sent when inquiring. Although non-bridge packets using ISOM face a similar problem, the non-bridge nodes see an increase in activity when bridge nodes are inquiring and the master has fewer slave nodes to service.

The packet generation time pmfs (in MSTSs between packets) and mean packet generation rate for piconets A-D in Figure 10.1 are shown in Figure 10.10a along with an exponential distribution with a similar mean. Since Piconet E contains only five slave nodes, the master has fewer nodes to attend to, there are half as many

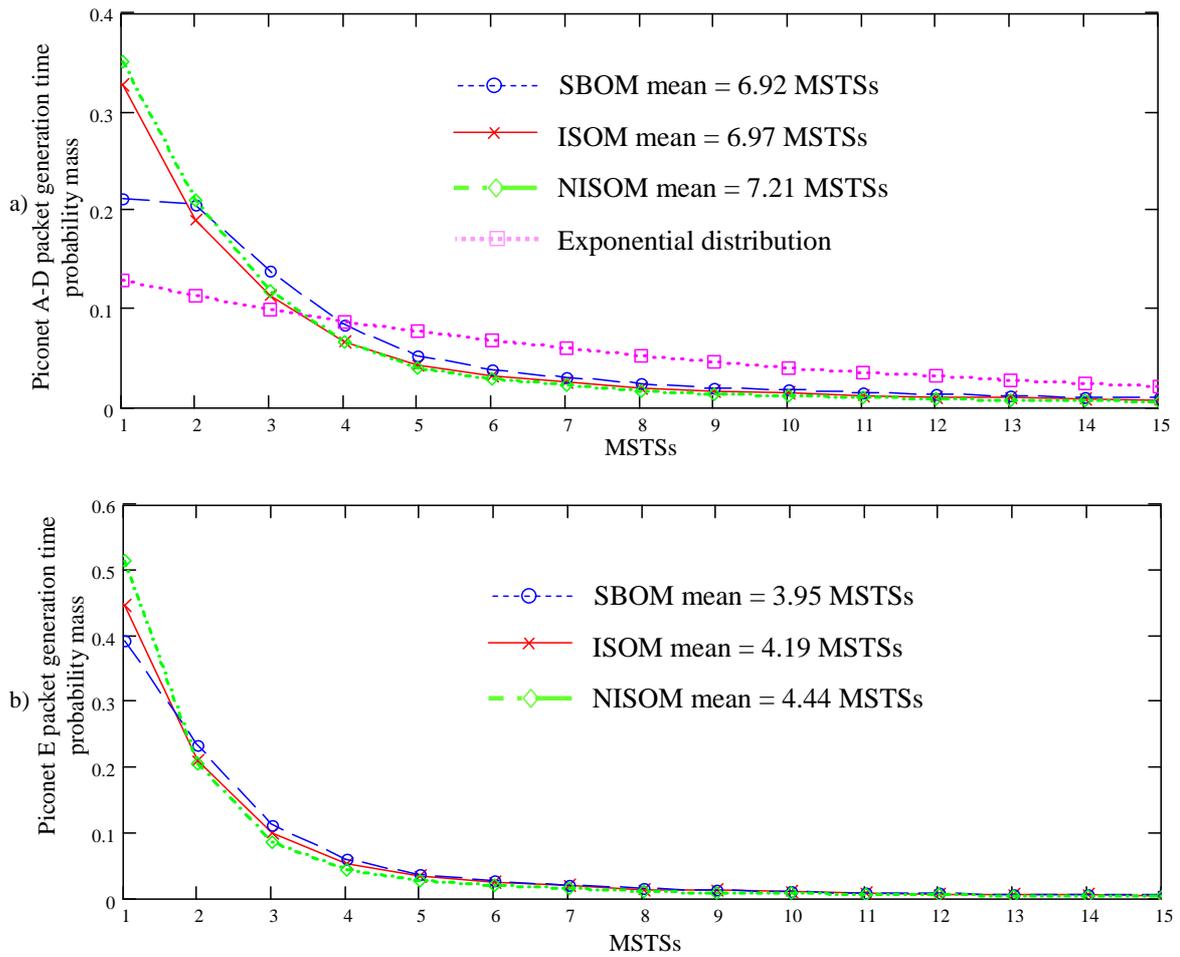


Figure 10.10: Non-bridge slave node packet generation time pmfs a) Piconets A-D b) Piconet E.

NB nodes as in other piconets, and slave nodes transmit more packets since they have fewer packets destined into the piconet. Thus, there are fewer nodes to divide the ‘excess’ MSTs between, but it is also less likely that a packet is sent to a NB node. Thus, the non-bridge slave in Piconet E generates packets almost twice as often as in other piconets. The packet generation time pmf for the non-bridge node in Piconet E is shown in Figure 10.10b.

The ISOM and NISOM pmfs have several outliers in the distribution (> 3072 time MSTs = 3.84 s) for packets generated after returning from inquiring for 3.84 s. Although the pmfs resemble the exponential distribution, the exponential distribu-

tion contains less mass for packet generation intervals less than 4 MSTSs and more than 40 MSTSs than those of the outreach method pmfs. Even so, all three pmfs for Piconets A-D contain 98% of their mass within 45 MSTSs. For Piconet E, 98% of the mass is contained within 23 MSTSs.

The NISOM and ISOM distributions contain greater mass than that of SBOM with one MSTS between packets. This is due to the master device emptying its buffer to the node in consecutive MSTSs when the node returns from inquiring since the buffered packets are older than all other buffered packets. Thus, the slave node quickly empties its buffer and has the opportunity to transmit packets in successive MSTSs.

10.4.2 Bridge slave nodes

Although, Piconet E has the same number of bridge nodes to the other piconets, it has fewer slaves. Thus, the bridge nodes in Piconet E are able to generate approximately 30% more packets than those in piconets with six slaves since the master has fewer nodes to service and the bridges have fewer nodes to relay packets for. The packet generation time pmfs are shown in Figure 10.11.

The bridge nodes using ISOM generates statistically fewer packets than those using SBOM and NISOM since only bridges using ISOM are unavailable for 3.84 s when inquiring.

10.4.3 Master nodes

The master nodes packet generation pmfs shown in Figure 10.12 were statistically identical at the 95% confidence level for the three outreach methods. Since Piconet E has fewer nodes to service, it received fewer packets from outside the piconet and was therefore able to transmit packets more often than relaying packets. Although the pmfs in Figure 10.12 look similar, the pmf for Piconet E has less mass

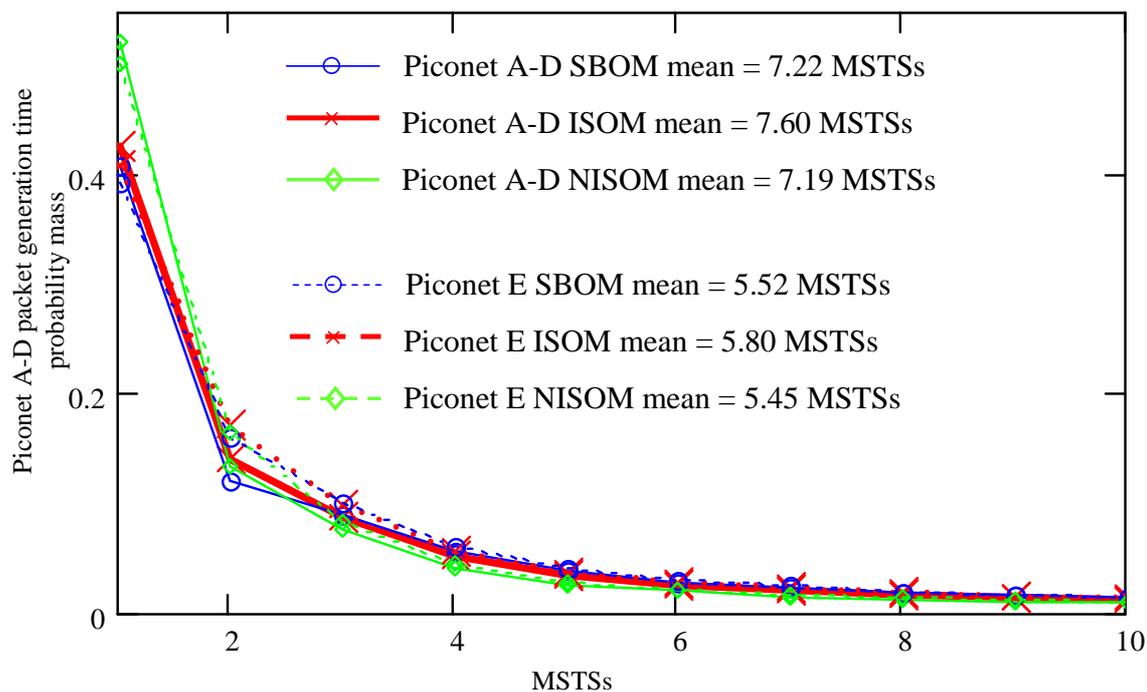


Figure 10.11: Bridge node packet generation time pmfs a) Piconets A-D b) Piconet E.

in the tail of the pmf. While 98% of Piconet E's pmf is contained within the first 48 MSTs, the other pmf requires 61 MSTs to capture 98% of the mass.

10.5 Summary

In this chapter, the performance metrics of the outreach methods were compared. Using knowledge from the preceding chapters, the field of proposed outreach methods was reduced to three; SBOM, NISOM, and ISOM. The parameters of the simulation model that generated the performance data was scribed in Section 10.2. In Section 10.3, it was shown that SBOM provides the greatest goodput and lowest PER, although the performance difference between methods is eliminated when collision avoidance is used. SBOM also had the smallest mean delay, mean and maximum buffer size, and discovery time. Finally, packet generation time pmfs were presented in Section 10.4.

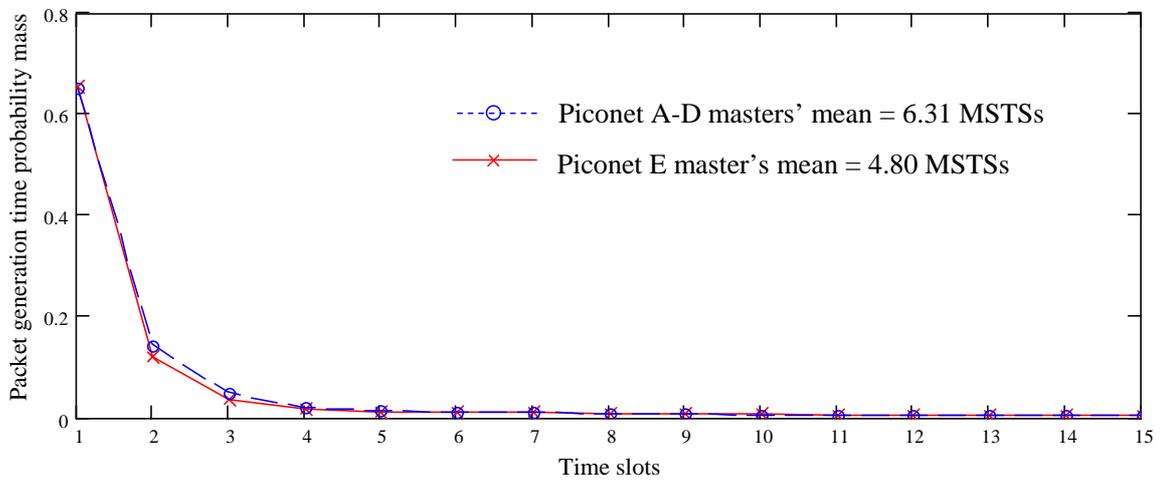


Figure 10.12: Master node packet generation time pmfs.

XI. Conclusions and Recommendations

This dissertation has developed a method for efficient discovery of wireless devices for a frequency hopping spread spectrum, synchronous, ad hoc network comprised of clustered sub-networks. The Bluetooth wireless protocol was used as the reference model. The development of a discovery or outreach method requires the characterization of performance metrics of Bluetooth piconets including the packet error rate between piconets and inquiring nodes, the PER between piconets, and the inquiry time. Additionally, improvements to the discovery process and packet collision avoidance were proposed.

11.1 Conclusions and Contributions

Although the NISOM and ISOM outreach methods discover devices arriving at a scatternet even if they do not enter the inquiry state, the methods are inferior to SBOM. ISOM and NISOM expend power (an extremely limited resource for portable devices) by inquiring on a regular basis, yet their discovery time is slow while simultaneously inflating buffer requirements and mean packet delay. In most scenarios presented in the specification [Blu03], devices do not enter a scatternet area often. Therefore, the inquiring devices in NISOM and ISOM usually discover nodes already in the scatternet. This adds an additional level of complexity since the inquiring devices must track which nodes are already in the scatternet and should be ignored. Thus, SBOM is the superior outreach method, providing quick discovery time by multiple piconets with no significant impact on goodput, buffer size, or mean delay.

Although discovery by a scatternet may be advantageous if a user unknowingly comes within range of a scatternet, users are generally aware of the devices with which they wish to communicate. Currently, users must initiate an inquiry to create a piconet. Using SBOM, devices in a scatternet can be directed by the entity organizing

the network to occasionally inquire to connect disjoint scatternets that have drifted together. However, as inquiries increase, the disadvantages found in NISOM and ISOM occur in SBOM as well.

In addition to the SBOM outreach method, the AFIT collision avoidance method (See Section 7.1) should be implemented in scatternets. It improves goodput and is useful with multi-slot packets.

With the introduction of the standard inquiry scan, the CLK-driven inquiry provides slight benefit in reducing inquiry time. Although it does not reduce inquire time using the interlaces inquiry scan, it does simplify implementation of the inquiry process with no adverse effects.

Although most conclusions are specific to the Bluetooth protocol, the impact of active discovery on PER, goodput, and mean delay can be applied to all clustered, FHSS ad hoc networks.

Research contributions include:

1. a complete analytical characterization of interference caused by inquiring nodes,
2. a complete analytical characterization of PER distribution between multiple piconets,
3. development and analytical characterization of the AFIT collision avoidance method,
4. analytical characterization of the PIAM collision avoidance method and its comparison to the AFIT method,
5. analytical characterization of the BT inquiry process from v1.1 as well as the standard and interlaced inquiry processes,
6. analytical characterization of the CLK-driven inquiry and verification of its implementation in measured devices, and

7. development and analysis of the SBOM, NISOM, and ISOM outreach methods using simulation models with the recommendation that SBOM be used in scatternets. Evidence that passive outreach methods are superior to those that actively search for arriving nodes is presented.

11.2 Recommendations for Future Research

This research has extended the current state-of-the-art in the discovery process of the Bluetooth protocol as well as the impact of neighboring Bluetooth devices and piconets on PER. However, the performance of outreach methods was studied using only a single scatternet configuration. Although, advantages of SBOM were significant enough to provide confidence that it is the best method in all scenarios where arriving nodes initiate an inquiry, it would be beneficial to investigate other scenarios and network configurations to:

1. determine what scenarios require the scatternet to perform inquiries for successful discovery and what pattern/interval permits reasonable discovery time with limited impact to scatternet performance.
2. further characterize the packet generation/inter-arrival time probability mass functions that saturate a scatternet under varying, likely conditions.
3. extend the packet generation/inter-arrival time to include the distribution of multi-slot packets and their generation rate.
4. investigate the impact on inquiry time when scanning devices have an SCO link which must be serviced, disrupting the scan window.
5. investigate the impact on inquiry time when inquiring devices have an SCO link which must be serviced, disrupting the inquiry train transmission.

Bibliography

- AnK00. D. Anvekar and M. Kapoor, "Frequency look-ahead and link state history based interference avoidance in wireless picocellular networks," *2000 IEEE International Conference on Personal Wireless Communications*, 17–20 Dec 2000.
- BaP02. S. Basagni and C. Petrioli, "A Scatternet Formation Protocol for Ad Hoc Networks of Bluetooth Devices," *Proceedings of the IEEE Vehicular Technology Conference*, Birmingham, AL, May 2002.
- Bal03. J. Ballagh, "Bluetooth Frequency Hop Selection Kernel Impact on Inter-Piconet Interference," Master's Thesis, Virginia Polytechnic Institute and State University, Apr 2003.
- BBB00. O. Berder, C. Boudier, and G. Burel, *Problems in Modern Applied Mathematics*, World Scientific and Engineering Society Press, 2000.
- BFK01. S. Baatz, M. Frank, C. Kuhl, P. Martini, and C. Scholz "Adaptive Scatternet Support for Bluetooth using Sniff Mode," *Proceedings of the IEEE Conference on Local Computer Networks, LCN 2001, Tampa, Florida*, Nov 2001.
- Blu01. "Specification of the Bluetooth System, Core Version 1.1," <http://www.bluetooth.com>, 1999.
- Blu03. "Specification of the Bluetooth System, Core Version 1.2," <http://www.bluetooth.com>, 2003.
- CCA04. Commercial Communication Associates HF PARA-MILITARY MAN PACK 1,6-30 MHZ, HF-90 product page, Available: <http://www.netangola.com/cca/pages/products/newproducts/1.htm>.
- CKG01. A. Capone, R. Kapoor, and M. Gerla, "Efficient Polling Schemes for Bluetooth Picocells," *Proceedings of IEEE International Conference on Communications ICC 2001*, Vol. 7, Helsinki, Finland, pp. 1990–1994, Jun 2001.
- CMB03. C. F. Chiasserini, M. Marsan, E. Baralis, P. Garza, "Towards Feasible Topology Formation Algorithms for Bluetooth-based WPANs," *Proceedings of the 36th Hawaii International Conference on System Sciences*, Hawaii, USA, Jan 2003.
- CMM04. K. L. Chan, J. Misić, and V. Misić, "Efficient Polling Schemes for Bluetooth Picocells Revisited," in *Proceedings of the 37th Hawaii International Conference on System Sciences*, Hawaii, USA, Jan 2004.
- DFS01. Deb, Ashok, K. Freberg, J. Surdu, A. Hall, and F. Maymi, "Modeling Communication Capabilities for Land Warrior 1.0," Presented at the Military Operations Research Society Symposium (MORSS) 2001, Annapolis, MD, 12–14 Jun 2001.

- ElH01. A. El-Hoiydi, "Interference Between Bluetooth Networks - Upper Bound on Packet Error Rate," *IEEE Communications Letters*, Vol. 5, pp. 245–247, Jun 2001.
- FCC98. Federal Communications Commission "Title 47, Code for Federal Regulations," Part 15, Oct 1998.
- FoC02. C. Foo and K. Chua, "BlueRings - Bluetooth Scatternets with Ring Structures," *Proceedings of the IASTED International Conference on Wireless and Optical Communication (WOC 2002)*, Banff, Canada, Jul 2002.
- GNL03. L. Godell, M. Nordan, T. Lapolla, and M. Mendez, "WLAN And Bluetooth Update: Beyond The Hype," Available: <http://www.forrester.com/ER/Research/Brief/Excerpt/0,1317,16960,00.html>.
- GTP86. S. Glisić, B. Todorović, and M. Popović-Lavrinsek, "A New Approach to Long Code Acquisition in Spread Spectrum Radio," *IEEE Global Telecommunications Conference Record (GLOBECOM '86)*, Vol. 5, pp. 1747–51, 1986.
- HaR02. D. Harris and M. Roberts, "Quantifying Bluetooth Piconet Mutual Interference," in *Proceedings of OPNETWORK 2002*, Washington D.C., Aug 2002.
- HAV99. "Draft Navy Training System Plan For The AN/ARC-182(V)Radio Set N88-NTSP-A-50-8115D/D." <http://www.avtechtra.navy.mil/pdf/ARC-182.PDF>, 1999.
- HaZ02. J. Haartsen and S. Zurbes, "Frequency Hop Selection in the Bluetooth Radio System," *IEEE 7th International Symposium on Spread-Spectrum Technology and Applications*, Sep 2002.
- HKZ02. L. Har-Shai, R. Kofman, G. Zussman, and A. Segall, "Inter-Piconet Scheduling in Bluetooth Scatternets," in *Proceedings of OPNETWORK 2002*, Washington D.C., Aug 2002.
- IEE99. "ANSI/IEEE Std 802.11, 1999 Edition, Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications," <http://standards.ieee.org/reading/ieee/updates/errata/802.11a-errata.pdf>, 2000.
- JAJ01. N. Johansson, F. Alriksson, and U. Jonsson, "JUMP Mode - A Dynamic Window-cased Scheduling Framework for Bluetooth Scatternets," *Proceedings of the ACM Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc 2001)*, Long Beach, California, Oct 2001.
- KBS99. M. Kalia, D. Bansal, and R. Shorey, "Data Scheduling and SAR for Bluetooth MAC," in *Proceedings Sixth IEEE International Workshop on Mobile Multimedia Communications (MOMUC'99)*, San Diego, CA, pp. 384–388, Nov 1999.

- KaL01. O. Kasten and M. Langheinrich, "First Experiences with Bluetooth in the Smart-It's Distributed Sensor Network," in *Proceedings of the International Conference on Parallel Architectures and Compilation Techniques, 2001*, Sept. 2001.
- KaP02. J. Karlsson and A. Persson, "Device and Service Discovery in Bluetooth Networks," Master's Thesis, Blekinge Institute of Technology, Available: http://www.inst-informatica.pt/v20/cid/biblioteca_digital/telecomunicacoes/200206BluetoothNetworks.pdf
- Kul95. V. Kulkarni, *Modeling and Analysis of Stochastic Systems*. London: Chapman and Hall, 1995.
- LaS01. C. Law and K. Siu, "Performance of a New Bluetooth Scatternets Formation Protocol." *Proceedings of the ACM Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc 2001)*, Long Beach, California Oct 2001.
- Leo03. M. Leopold, "Evaluation of Bluetooth communication: Simulation and Experiments," Technical Report, University of Copenhagen, Feb 2003.
- LiS02. X. Li and I. Stojmenovic, "Partial Delaunay Triangulation and Degree Limited Localized Bluetooth Scatternet Formation," *Proceedings AD-HOC Networks and Wireless (ADHOC-NOW)*, Fields Institute, Toronto, Sep 2002.
- LTC03. T. Lin, Y. Tseng, K. Chang, and C. Tu, "Formation, Routing, and Maintenance Protocols for the BlueRing Scatternet of Bluetooths," *Proceedings of the 36th Hawaii International Conference on System Sciences*, Hawaii, USA, Jan 2003.
- MiM03. J. Misić and V. Misić, "Modeling Bluetooth Piconet Performance," *IEEE Communications Letters*, Vol. 7, No. 1, Jan 2003.
- NCR97. NRC, National Research Council, *Tactical Display for Soldiers: Human Factor Considerations*, Washington D.C.: National Academy Press, 1997.
- Per01. C. Perkins, *Ad Hoc Networking*, Boston: Addison-Wesley, 2001.
- PBK04. B. Peterson, R. Baldwin, and J. Kharoufeh, "A Specification-Compatible Bluetooth Inquiry Simplification," in *Proceedings of the 37th Hawaii International Conference on System Sciences*, Hawaii, USA, Jan, 2004.
- PBK04a. B. Peterson, R. Baldwin, and J. Kharoufeh, "Bluetooth Inquiry Time Characterization and Selection," *submitted to IEEE Transactions on Mobile Computing*.
- PBK03. B. Peterson, R. Baldwin, J. Kharoufeh, and R. Raines, "Refinements to the Packet Error Rate Upper Bound for Bluetooth Networks," *IEEE Communications Letters*, Vol. 7, No. 8, pp. 382-384, August 2003.
- PBR04a. B. Peterson, R. Baldwin, and R. Raines, "Inquiry Packet Interference in Bluetooth Scatternets," *ACM Mobile Computing and Communications Review*, Vol. 8, No. 2, pp. 66-75, April 2004.

- PBR04b. B. Peterson, R. Baldwin, and R. Raines, "Packet Error Rate Distribution Between Random Bluetooth Piconet Pairs," *submitted to Wireless Personal Communications*.
- PBR04c. B. Peterson, R. Baldwin, and R. Raines, "Enhanced Interference Avoidance in a Bluetooth Scatternet," *submitted to AdHoc Networks*.
- PGR03. P. Popovski, L. Gavrilovska, T. Renier, H. Fathi, and R. Prasad, "Energy-efficient Interference Avoidance for Interconnected Bluetooth Personal Area Networks," *IEEE Vehicular Technology Conference VTC-Spring '03*, Vol. 7, No. 8, pp. 382-384, April 2003.
- PZB95. R. Peterson, R. Ziemer, and D. Borth, *Introduction to Spread Spectrum Communications*, Englewood Cliffs, New Jersey: Prentice-Hall, 1995.
- Rap88. S. Rappaport, "Rapid Code Acquisition Schemes for Spread Spectrum Packet Signals," *Proceedings of the SPIE -The International Society for Optical Engineering*. pp. 68-71. 1988.
- RBK01. R. Rao, O. Baux, and G. Kesidis, "Demand-based Bluetooth Scheduling," in *Proceedings of the 33rd IEEE Wireless LAN (WLAN) Conference*, Boston, MA, Sept. 2001.
- RMK01. A. Rácz, G. Miklos, F. Kubinsky, and A. Valko, "A Pseudo Random Coordinated Scheduling Algorithm for Bluetooth Scatternets," *Proceedings of the ACM Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc 2001)*, Long Beach, California, Oct 2001.
- SBT00. T. Salonidis, P. Bhagwat, and L. Tassiulas, "Proximity Awareness and Fast Connection Establishment in Bluetooth," in *Proceedings of the First Annual ACM Workshop on Mobile and Ad Hoc Networking and Computing 2000*, Boston, Massachusetts, Aug 2000.
- SBT01. T. Salonidis, P. Bhagwat, L. Tassiulas, and R. LaMaire, "Distributed Topology Construction of Bluetooth Personal Area Networks," in *Proceedings of IEEE INFOCOM 2001*, Anchorage, Alaska, Apr 2001.
- SIN96. "TALK II-SINGARS Multiservice Communications Procedures For The Single-Channel Ground And Airborne Radio System," Available: <http://www.fas.org/man/dod-101/sys/land/docs/singars.pdf>, 1996.
- Skl01. B. Sklar, *Digital Communications: Fundamentals and Applications, 2nd edition*, New Jersey: Prentice Hall, 2001.
- Sto02. I. Stojmenovic, "Dominating Set Based Bluetooth Scatternet Formation with Localized Maintenance," *Proceedings of the International Parallel and Distributed Processing Symposium: IPDPS 2002 Workshops*, Apr 2002.
- TzG00. A. Tzamaloukas and J.J. Garcia-Luna-Aceves, "A Channel-Hopping Protocol for Ad-Hoc Networks," *Proceedings of the IEEE International Conference*

on *Computer Communications and Networks (IC3N) '00*, Las Vegas, NV, Oct 2000.

- VaE87. I. Vajda and G. Einarsson, "Code Acquisition for a Frequency Hopping-System," *IEEE Transactions on Communications*, Vol. 37, No. 5, pp. 540–541, May 1987.
- WTH02. Z. Wang, R. Thomas, and Z. Haas, "Bluenet - a New Scatternet Formation Scheme," in *Proceedings of the 35th Hawaii International Conference on System Sciences*, Hawaii, USA, Jan, 2002.
- WRV86. N. Wilson, S. Rappaport, and M. Vasudevan, "A Rapid Acquisition Scheme for Spread Spectrum Radio in a Fading Environment," *IEE Proceedings-Radar, Sonar and Navigation Vol. 135*, pp. 95-104, 1988.
- WJC02. R. Woodings, D. Joos, T. Clifton, and C. Knutson, "Rapid Heterogeneous Ad Hoc Connection Establishment: Accelerating Bluetooth Inquiry Using IrDA," *Proceedings of the Third Annual IEEE Wireless Communications and Networking Conference (WCNC '02)*, Orlando, Florida, Mar 2002.
- YiN02. K. Yip and T. Ng, "Transmit-Power Reduction for Class 1 Bluetooth-enabled indoor cordless phones," *IEEE Transactions on Consumer Electronics* Vol. 48, No. 4, Nov, 2002.
- ZBC01. G. Zaruba, S. Basagni, and I. Chlamtac, "Bluetrees - scatternet formation to enable Bluetooth-based ad hoc networks," *Proceedings of the IEEE International Conference on Communications (ICC 2001)*, St. Petersburg, Jun 2001.
- ZaC02. G. Zaruba and I. Chlamtac, "Accelerating Bluetooth Inquiry for Personal Area Networks," in *Proceedings of the 2002 International Conference on Parallel and Distributed Processing Techniques and Applications (PDPTA'02)*, Las Vegas, NV, USA, June 2002.
- ZaG04. G. Zaruba and V. Gupta, "Simplified Bluetooth Device Discovery Analysis and Simulation," in *Proceedings of the 37th Hawaii International Conference on System Sciences*, Hawaii, USA, Jan, 2004.
- ZGE01. J. Zyren, T. Godfrey, and D. Eaton, "Does Frequency Hopping Enhance Security?" Available: <http://www.packetnexus.com/docs/>, 2001.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

The public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing the burden, to the Department of Defense, Executive Services and Communications Directorate (0704-0188). Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to any penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number.

PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ORGANIZATION.

1. REPORT DATE (DD-MM-YYYY) 01-09-2004	2. REPORT TYPE Doctoral Dissertation	3. DATES COVERED (From - To) Jul 2001-Sep 2004
---	---	---

4. TITLE AND SUBTITLE Device Discovery In Frequency Hopping Wireless Ad Hoc Networks	5a. CONTRACT NUMBER
	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S) Brian S. Peterson, Major, USAF	5d. PROJECT NUMBER JON 04-114
	5e. TASK NUMBER
	5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Air Force Institute of Technology Graduate School of Engineerin and Management (AFIT/EN) 2950 Hobson Way WPAFB OH 45433-7765	8. PERFORMING ORGANIZATION REPORT NUMBER AFIT/DS/ENG/04-06
---	---

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) National Security Agency Attn: Mr. William Kroah 9800 Savage Road Ft. Meade, MD 20755-6000 410-854-0061	10. SPONSOR/MONITOR'S ACRONYM(S)
	11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED

13. SUPPLEMENTARY NOTES

14. ABSTRACT
This research develops a method for efficient discovery of wireless devices for a frequency hopping spread spectrum, synchronous, ad hoc network comprised of clustered sub-networks. The Bluetooth wireless protocol serves as the reference protocol. The development of a discovery, or outreach, method for scatternets requires the characterization of performance metrics of Bluetooth piconets, many of which are unavailable in literature. Precise analytical models characterizing the interference caused to Bluetooth network traffic by inquiring devices, the probability mass function of packet error rates between arbitrary pairs of Bluetooth networks, and Bluetooth discovery time distribution are developed. Based on the characterized performance metrics, three scatternet outreach methods are developed and compared. Outreach methods which actively inquire on a regular basis, as proposed in literature, are shown to produce lower goodput, have greater mean packet delay, require more power, and cause significant delays in discovery. By passively remaining available for outreach, each of these disadvantages is avoided.

15. SUBJECT TERMS
wireless communications;communications networks;network analysis (management)

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT UU	18. NUMBER OF PAGES 242	19a. NAME OF RESPONSIBLE PERSON Rusty O. Baldwin, Ph.D.
a. REPORT U	b. ABSTRACT U	c. THIS PAGE U			19b. TELEPHONE NUMBER (include area code) (937) 255-6565 ext. 4445 (DSN 785)