MAC LAYER ASSISTED LOCALIZATION IN WIRELESS ENVIRONMENTS WITH MULTIPLE SENSORS AND MULTIPLE EMITTERS

A Thesis
Presented to
The Academic Faculty

by

Paul W. Garver

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy in the
School of Electrical and Computer Engineering

Georgia Institute of Technology
December 2017

MAC LAYER ASSISTED LOCALIZATION IN WIRELESS ENVIRONMENTS WITH MULTIPLE SENSORS AND MULTIPLE EMITTERS

Approved by:

Dr. Edward J. Coyle, Advisor
School of Electrical and Computer Engineering
Georgia Institute of Technology

Dr. Randal T. Abler
School of Electrical and Computer Engineering
Georgia Institute of Technology

Dr. David Anderson
School of Electrical and Computer Engineering
Georgia Institute of Technology

Dr. Kishore Ramachandran
College of Computing
Georgia Institute of Technology

Dr. John Barry
School of Electrical and Computer Engineering
Georgia Institute of Technology

Date Approved: 25 September 2017

To my late mother, Francis Del Garver, for her persistent encouragement.

ACKNOWLEDGEMENTS

I am grateful to my advisor, Dr. Edward J. Coyle, for challenging me academically and providing guidance throughout this dissertation. His insights and useful discussion helped to frame my thinking and approach to problem solving. Thank you for the committee members for their constructive criticism. I am also grateful for the financial contributions by BIT Systems, especially Tom Ladd, and Harris Corporation to complete my studies.

The testbed in Bobby Dodd Stadium would not have been possible without the hard work of the Intelligent Digital Communications Vertically Integrated Projects (VIP) team and Dr. Randal T. Abler. Dr. Abler's implementation insight, networking experience, and equipment was invaluable to the success of the testbed. Undergraduate students Jaison George, Orin Lincoln, and Hayden Flinner made significant contributions for which I am grateful. Additionally, the resources of the Senior Design Laboratory staffed by James Steinberg and Kevin Pham were critical in constructing the testbed. Finally, Doc and others from Georgia Tech Athletics were instrumental in providing access and supporting our *living laboratory* in the stadium.

Finally, this dissertation would not have been possible without the tireless support, encouragement, and advice from my beautiful wife, Jill. Thank you for always being for me, and for all you have done for our family. Taking such great care of our boys, James and Caleb, working part time, and walking with me through graduate school was no easy task. I am truly grateful for you, Jill.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

SUMMARY

Extreme emitter density (EED) RF environments, defined as 10k-100k emitters within a footprint of less than 1 km^2, are becoming increasingly common with the proliferation of personal devices containing myriad communication standards (e.g. WLAN, Bluetooth, 4G, etc). Attendees at concerts, sporting events, and other such large-scale events desire to be connected at all times, creating tremendous spectrum management challenges, especially in unlicensed frequencies such as 2.4 GHz, 5 GHz, or 900 MHz Industrial, Scientific, and Medical (ISM) bands. In licensed bands, there are often critical communication systems such as two-way radios for emergency personnel which must be free from interference. Identification and localization of a non-conforming or interfering Emitter of Interest (EoI) is important for these critical systems.

In this dissertation, research is conducted to improve localization for these EED RF environments by exploiting side information available at the Medium Access Control (MAC) layer. The primary contributions of this research are: (1) A testbed in Bobby Dodd football stadium consisting of three spatially distributed, time-synchronized RF Sensor Nodes (RFSN) collecting and archiving complex baseband samples for algorithm development and validation. (2) A modeling framework and analytical results on the benefits of exploiting the structure of the MAC layer for associating physical layer measurements, such as Time Difference of Arrivals (TDoA), to emitters. (3) A three stage localization algorithm exploiting time between packets and a constrained geometry to shrink the error ellipse of the emitter position estimate. The results are expected to improve localization accuracy in wireless environments when multiple sensors observe multiple emitters using a known communications protocol within a constrained geometry.

CHAPTER I

INTRODUCTION

Extreme emitter density (EED) RF environments, defined as 10k-100k emitters within a footprint of less than 1 km^2, are becoming increasingly common with the proliferation of personal devices containing myriad communication standards (e.g. WLAN, Bluetooth, 4G, etc). Attendees at concerts, sporting events, and other such large-scale events desire to be connected at all times, creating tremendous spectrum management challenges, especially in unlicensed frequencies such as 2.4 GHz, 5 GHz, or 900 MHz Industrial, Scientific, and Medical (ISM) bands. In licensed bands, there are often critical communication systems such as two-way radios for emergency personnel which must be free from interference. Identification and localization of a non-conforming or interfering Emitter of Interest (EOI) is important for these critical systems.

To study this problem in depth, a joint experimental and analytical research approach was undertaken. A testbed initially consisting of three spatially distributed RF sensor nodes (RFSN) to capture raw RF spectrum samples from realistic EED environments has been designed and deployed in Bobby Dodd football stadium at the Georgia Institute of Technology. Over 30 Terabytes (TB) of raw IQ spectrum samples have been collected and archived during live football games. Chapter 2 describes this testbed in detail, as well as a more controlled laboratory version. One associated theoretical problem of interest is that with multiple emitters there is ambiguity in assigning a given sequence of physical layer measurements, such as Time-of-Arrival (ToA), from the sensors to one of the emitters. A novel idea is proposed for this *data association* problem by exploiting side information provided by the Medium Access Control (MAC) layer to improve the probability of correct association, even if the packets can not be decoded. Chapter 3 describes the approach and provides theoretical results suggesting the approach can scale well for the large number of emitters present in an EED environment. A novel three-strategy localization approach is

proposed in Chapter 4 that can lower the uncertainty of the position estimate. The approach uses packet timing information from the MAC layer, as well as geometry constraints. These chapters show the benefit of using MAC layer side information for EED RF environments. Finally, overall conclusions and future research directions are discussed in Chapter 5.

Chapter 2 discusses implementation details, including software and hardware, for the two EED testbeds which have been deployed. The chapter is divided into two parts. Section 2.1 investigates the ability of the sensors to process signals using Software Defined Radio (SDR). This study was undertaken to assess the abilities and limitations of various hardware before deployment into the stadium and the laboratory. It provides a comparison of sensor hardware capabilities against Size, Weight, Area, and Power (SWAP) requirements. Once the hardware was selected, Section 2.2 discusses two testbeds that were created. The first is a laboratory testbed, referred to as Laboratory LOC-EED, which provides a controlled experimentation environment. The second, Stadium LOC-EED, describes the stadium testbed deployment.

Software defined radios, that digitize RF spectrum and perform traditional receiver tasks in software, are becoming increasingly viable as an enabling technology for mobile networks and sensor networks. The concurrent rise in commercially available small form-factor, low-power, x86-based processors creates the possibility of incorporating General Purpose Processor (GPP) software radios into existing sensor networks. The eStadium VIP project is considering the addition of such nodes to sense digitized RF spectrum data in Bobby Dodd football stadium. The flexibility inherent in GPP software radio provides rapid algorithm testing; however, the hardware is often large, heavy, and power intensive. Due to the limited resources and practical considerations in the stadium, the trade-offs between SWAP requirements and SDR capabilities must be studied prior to deployment. A performance analysis across four PC form factors, including one suitable for embedded use, running realistic SDR applications is presented in Section 2.1. Case studies include FM radio with the BPSK modulated Radio Broadcast Data Service (RBDS), FM analog video, and distributed processing of digital video with QPSK modulation. Such studies provide valuable insight into SDR testbeds.

As the RF spectrum becomes increasingly congested, localization algorithms which are tolerant of high levels of interference become necessary. A unique opportunity exists to study these issues during any event in a large venue, such as a football game in a large stadium. Section 2.2 reports on the development of a RF sensor localization field deployment, LOC-EED, in the football stadium at Georgia Tech as well as a simplified laboratory testbed for controlled experimentation. During football games, cellphones, stadium personnel radios, media organization radios and wireless controlled devices, game official wireless headsets, etc. create an EED background that is a challenge to any algorithm attempting to identify and localize a single emitter. The laboratory testbed and field deployment to study this problem consists of RFSNs using wideband RF digitizers and general purpose processors to sense the RF environment. SDR is used as an enabling technology for the development of unique cross-layer localization techniques which are typically not realizable on specialized hardware, such as WLAN Access Points (AP). Additionally, a preliminary analysis of spectrum captures in the 2.4 GHz band during a live football game is provided. The analysis and a simulation of a simple cross-layer localization technique confirm both the need for, and ability to exploit, cross-layer information for localization.

Localization is especially challenging in EED environments, in part, due to ambiguity in associating physical layer measurements, such as the time of arrival, to the proper emitter. Typical approaches in the radar and network security literature use physical layer characteristics of the transmitters as features to aid in this *data association* problem. However, there is significant structure at OSI Layer 2 to be exploited for known communications protocols. Examples include the MAC protocol and packet-level correlations. This idea is explored in Chapter 3, in the context of IEEE 802.11g, by using knowledge of the packet exchange sequence (PES), virtual carrier sense, and CSMA/CA to lower the Probability of Association Error (PAE) compared to an SNR-based Layer 1 strategy. Analytical expressions are derived for the PAE on both a per packet and per packet exchange sequence basis. It is shown that while Layer 1 outperforms the Layer 2 strategy for a single packet at low SNR, on a per packet exchange sequence basis the Layer 2 approach is superior. While the results are specific to WLANs, the approach may be applied more broadly to any communications

protocol with a MAC layer.

Chapter 4 proposes a fast and precise three-stage localization algorithm which exploits the fact that many potential interferers in these EED environments follow known communications protocols and the Emitter of Interest (EoI) is typically contained within a small region. Stage I uses only sensors able to decode packets to estimate position. A Confidence Region (CR) is then computed. In Stage II, sensors unable to decode packets bound their Time Delay Estimates (TDE) using this CR. A new CR for Stage II is then computed. Stage III exploits packet timing information from the MAC layer to estimate a distance from an anchor node with a known location, such as an Access Point (AP), to the EoI. The final CR is the intersection of the CRs from all three stages. The principle contributions of this chapter are the three-stage algorithm derivation with simulated results, a novel Packet Time-Difference-of-Arrival (PTDoA) technique using the MAC layer information, and analytical results on TDE variance as a function of window size and Signal-to-Noise Ratio (SNR).

As a whole, this dissertation suggests that localization can greatly benefit from a cross-layer approach. A data association and localization algorithm have been proposed which exploit the side information provided by the MAC layer. However, the MAC layer, as well as higher-level OSI layers such as the transport and application layer, have a rich amount of side information which has yet to be exploited. The primary contribution, therefore, is to show two examples of how the MAC layer can be exploited to improve localization, as well as provide over 30 terabytes (TB) of RF spectrum field data in EED environments for future characterization and analysis.

CHAPTER II

TESTBED DEVELOPMENT AND DEPLOYMENT

2.1 Comparison of High Performance Software Radios

Software-Defined Radios (SDR), which digitize RF spectrum and perform traditional receiver tasks in software, are becoming increasingly viable as an enabling technology for mobile networks and sensor networks. The concurrent rise in commercially available small form-factor, low-power, x86-based processors creates the possibility of incorporating General Purpose Processor (GPP) software radios into existing sensor networks. The eStadium VIP project is considering the addition of such nodes to sense digitized RF spectrum data in Bobby Dodd football stadium. The flexibility inherent in GPP software radio provides rapid algorithm testing; however, the hardware is often large, heavy, and power intensive. Due to the limited resources and practical considerations in the stadium, the trade-offs between size, weight, area, and power (SWAP) requirements and SDR capabilities must be studied prior to deployment. A performance analysis across four PC form factors, including one suitable for embedded use, running realistic SDR applications is presented. Case studies include FM radio with the BPSK modulated Radio Broadcast Data Service (RBDS), FM analog video, and distributed processing of digital video with QPSK modulation. Such studies provide valuable insight into SDR testbeds. The eStadium VIP project [33, 91, 4, 90] is a *Living Lab* for the research, development and deployment of technology for the next generation of wireless communication systems for large-scale events. These events, such as large concerts and football games, involve 10K to 100K spectators who are located in a structure with a limited footprint, typically less than 1 km^2. The vast majority of these spectators now carry smartphones that support many communication protocols - 3G/4G cellular, WiFi, Bluetooth, etc. - that operate in both licensed and unlicensed bands. The venue in which they operate often has a number of wireless systems - DAS-based cellular systems, WiFi infrastructure, RF-ID systems, ZigBee-based sensor networks, etc. - to

support connectivity with/between spectators and for event operations. These events are thus extreme in both the types and volume of data that can be generated and in the types of communication infrastructure that must coexist and, if possible, collaborate with each other. The eStadium team has been developing an extensive testbed for wireless systems within Bobby Dodd Stadium, the football stadium at Georgia Tech. This testbed includes, but is not limited to:

- Web applications that enable on-demand access for spectators to multimedia content, including video-clips of all plays, visualization of game events, and current game/player stats [33, 91, 4, 90].
- Social networking applications that enable alumni of similar backgrounds to find and chat with each other in the stadium.
- A sensor network to monitor structural vibrations of the stadium, audio of the crowd, and algorithms to estimate the distance to transmitters [90, 5, 73, 82].

Bobby Dodd stadium includes a DAS-based cellular system and 4G multi-cast and broadcast capabilities are expected to be available in the next year or two. There is limited, for-pay WiFi access in some parts of the stadium's seating and concourse areas. The current sensor network operates a ZigBee-like protocol in the 2.4GHz ISM band and includes a TV white-space backhaul link. The team controls some of this wireless infrastructure and collaborates with organizations, such as AT&T, that control the licensed parts of it. We thus have a unique opportunity to identify opportunities to maximize the capacity available for communications of all types by determining: what content to multicast or broadcast instead of unicast; the level of interference due to WiFi APs or other RF infrastructure outside the stadium [82]; and when and how to shift capacity demands from one type of network to another.

Additional RF sensors are needed to perform these tasks. Due to the flexibility required, GPP software radios [55] are a natural extension of our existing sensor network. However, consideration must be given to field deployment. We work closely with GT Athletics to

deploy these systems in the stadium. The SWAP constraints on such a deployment are very strict and include appearance as well as economic considerations.

- Size, Weight and Area are critical system factors as large antennas or sensor enclosures will be distracting to fans. The expense and risk of mounting and securing systems in remote locations in the stadium increase dramatically with the size and weight of the system.

- Power is extremely limited throughout the stadium and installing additional power infrastructure is very expensive. The current eStadium sensor network, which is mounted throughout the steel framing that supports the stands, is therefore battery-powered and wireless. Power is an important consideration for deployment of additional RF sensor nodes.

Commodity GPP hardware of differing size, weight, and power were compared using both narrow and wide-band applications as a feasibility study prior to stadium deployment. The applications demonstrated range in sample rate from 1-25 MSPS and include FM radio with the BPSK modulated Radio Broadcast Data Service (RBDS), FM analog video, and distributed processing of digital video with QPSK modulation. The principle contributions of these experiments are:

1. Demonstrating the viability of a GPP SDR approach for SWAP-constrained sensor networks.

2. Comparing commercially available hardware platforms under realistic SDR applications.

3. Providing recommendations for similar testbeds.

A brief review of SDR architectures in the literature is given in Section 2.1.1. Details of the hardware tested are given in Section 2.1.2, while the software configuration is described in Section 2.1.3. The objective was to categorize these GPP platforms with respect to their SWAP profile and sensing capability. To that end, Section 2.1.4 discusses test procedures

and metrics used in evaluation. Results are presented in Section 2.1.5. These case studies help inform the hardware and software architecture of the eStadium testbed. Recommendations for a commodity hardware platform and other implementation considerations are described in Section 2.1.6.

2.1.1 Background

Current SDR architectures can be classified into six approaches: general purpose processors, co processor, processor centric, configurable units, programmable blocks, and distributed [25]. Only GPP and processor centric approaches will be reviewed, but in general there is an attempt to balance flexibility and code portability while maximizing computational capacity. All approaches except for GPP use specialized hardware to perform DSP tasks.

GPP SDR platforms consist of a commodity computer with a DSP software suite and an RF digitizer. Such systems are the most flexible and the DSP code most generic since specialized hardware is minimized. For the same reason, they also tend to consume the most power. SORA [78] is one platform which uses a custom RF digitizer board called a SORA RCB. The supporting DSP software exploits multiple cores and vector instructions [78]. SORA was demonstrated by implementing a real-time WiFi and LTE stack on a 2.67 GHz Intel Core Duo 2 and a 2.67 GHz Core i7-920, respectively. GNURadio is a popular open source SDR platform with many standard receiver blocks available [34]. It uses C/C++ blocks for the implementation of core DSP algorithms and Python to connect the blocks and provide the control plane. These implementations often use the Vector-Optimized Library of Kernels (VOLK)[1] as a standard interface to vector processor instructions such as Intel's AVX. GNURadio supports many RF digitizers, but is often paired with an Ettus USRP [81].

Other approaches, including processor centric, consist of specialized hardware to perform computations rather than a general purpose processor. Compared to the GPP approach, these systems are more power efficient but the code is less portable. Additionally, the learning curve for developers is typically steeper. Examples of this architecture include

[1] `http://www.libvolk.org`

Table 1: System SWAP Comparison

Sys	Manufacturer	Model	Form Factor	Dim. (cm)	Wt. (lbs)	TDP (W)
A	Intel	DC3217BY	UCFF	11.6x11.2x3.9	2	17
B	Intel	S1200KPR	Mini-ITX	22x17.7x28.6	8.25	77
C	Dell	Optiplex 990	Custom	29x9.3x31.2	12.5	95
D	SuperMicro	X7DWE	ATX	63x48x9	15.75	160
E	Intel	DC53427HYE	UCFF	11.6x11.2x3.9	2	17

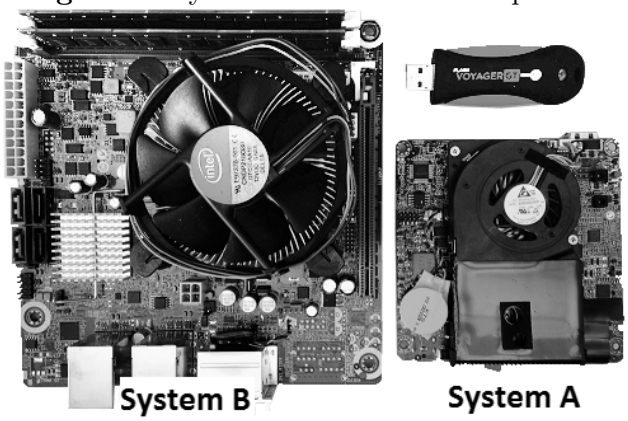
the SODA platform [51] and corresponding commercial prototype, Ardbeg. Ardbeg has algorithm specific hardware, while SODA does not [87]. Such an approach can be very power efficient; Ardbeg uses less than a tenth of a watt to process DVB-T at 5Mbps [87]. FPGA-based hardware includes the Wireless Open Access Research Platform (WARP) [85], the Nutaq Perseus 6010-based system [9], and Rutgers' WiNC2R [54]. Such hardware specialization is power efficient but may preclude rapid prototyping of new algorithms.

2.1.2 Hardware Configuration

Five commodity hardware platforms with different SWAP attributes were selected for evaluation. Table 1 provides the power requirements, size, and model information for each platform. Identification is either based on the motherboard or system if sold as a single unit. The form factor includes the chassis. System processors, RAM, and the most advanced vector instruction set supported for each platform are given in Table 2. The CPU frequencies given are nominal and do not include such features as Intel Turbo Boost, which can increase the frequency for a period of time if certain physical system constraints are met. Systems B, C, and D will be benchmarked and compared. Systems A and E, which are practical for implementation in the stadium, are used to demonstrate some proof-of-concept sensor network applications. Figure 1 illustrates the size of Systems A and B in comparison to a USB flash drive.

System A is an Intel Next Unit of Computing (NUC) platform with an Ivy-Bridge processor consisting of dual 1.8 GHz cores. This platform was equipped with 16GB RAM, which is the maximum amount supported on the motherboard. The Ivy-Bridge class processors support the Advanced Vector Extensions (AVX) single instruction multiple data (SIMD) instruction set, which increases the efficiency of signal processing operations. AVX

Figure 1: System A and B Size Comparison



consists of 256 bit instructions which can operate on floating point data. The processor is designated as Ultra-Low Voltage (ULV) by Intel and is intended to be used in mobile computing applications. System A is the smallest form factor tested with a total volume of 507 cm^3.

System B consists of the Intel S1200KPR Mini-ITX motherboard with a server class CPU in contrast to the mobile ULV processor of System A. This system also supports AVX instructions. In contrast to System A, it has a Max TDP approximately 3.5 times higher at 77W and is a quad-core. This system was equipped with 16 GB of RAM, the maximum supported. This processing platform has a volume of 11137 cm^3.

A typical desktop computer was included for comparison as System C. The platform is a Dell Optiplex 990 with a quad core i7 desktop processor. System C also supports AVX instructions, but is only equipped with 8GB of RAM. The volume for this platform is 8415 cm^3.

An older server-class platform on a conventional ATX motherboard was included as System D. Unlike all other systems which have a single physical CPU, this system has two quad-core Xeon CPUs. Since these processors are from 2007, the most advanced vector instruction set supported is SSSE3. The total volume of this platform is 27216 cm^3.

System E is the next generation of System A, with a Core i5 vs. i3 processor. A crucial hardware advantage of System E over A is the ability to increase the processor clock frequency from 1.8 to 2.8 GHz as needed. During evaluation, it was confirmed with the

Table 2: System Processor Comparison

Sys	Processor	CPUs	Cores	Clk (GHz)	Mem (GB)	SIMD	Date
A	Intel Core i3-3217U	1	2	1.8	16	AVX	Q2 2012
B	Intel Xeon E3-1275v2	1	4	3.5	16	AVX	Q2 2012
C	Intel Core i7-2600	1	4	3.4	8	AVX	Q1 2011
D	Intel Xeon E5472	2	4	3	8	SSSE3	Q4 2007
E	Intel Core i5-3427U	1	2	1.8	16	AVX	Q2 2012

Linux utility *turbostat* that both processors were clocked at 2.6 GHz. Without the clock increase, the digital video test would not be possible with only two nodes. Like System A, the volume is 507 cm^3.

An RF digitizer was used to convert signals to complex baseband. A maximum of 25 MHz of analog bandwidth with center frequencies up to 6 GHz can be captured. These signals are then sampled and transported to the host system via gigabit ethernet. Receiver tuning, sample rates, and gains are controllable via the host.

2.1.3 Software Configuration

Systems A, B, D, and E were configured with Red Hat Enterprise Linux (RHEL) 6.3, while System C used RHEL 5.5. An internally developed software radio suite (SRS) was used as an SDR platform. The SRS consists of DSP blocks written in C/C++, or Fortran. These blocks can be connected to each other either by a custom language or Python. Conceptually, these connections are very similar to UNIX-style pipes between processes. Each block typically runs as a separate process on the operating system, enabling a performance evaluation of each individual DSP function in the processing chain. Some computationally intensive blocks may be threaded, but each block is always a single process. The most common blocks are:

- **RFDRX**: Receives a packet over a gigabit NIC from the RF Digitizer. Each packet contains a header as well as complex data samples
- **PSPLIT**: Splits packet into header information, which contains fields such as sample time, and the payload samples.
- **FMDM**: Performs frequency demodulation on samples of a signal by differentiating

the phase to obtain instantaneous frequency

- ***_FILT**: Filters, decimates, and mixes multiple input signals
- **FFT**: Performs the Fast Fourier Transform (FFT) operation on input signals, using the Intel Math Kernel Library (IMKL) implementation

All processing blocks are compiled with Intel C/C++ and Fortran compilers version 12.1.0. The SRS uses Intel Performance Primitives (IPP) and IMKL extensively for core signal processing functions. For example, all FFT operations are performed with the IMKL FFT function. IMKL and IPP dynamically launch optimized library versions depending on the target hardware and use SIMD instructions.

2.1.4 Methodology

Performance metrics are captured by running the Linux program *nmon* [58] in the background and recording the data to a file. Nmon collects PC statistics such as memory, CPU, and network usage. CPU statistics are collected by reading */proc/pid/stat*, where *pid* is a given process id. This path contains the time the given process was scheduled for user and kernel space execution in units of $\frac{1}{100}$ of a second. Denote the jth sample of the user space and kernel space time of process as U_{ij} and K_{ij}, respectively. Let Δ_t represent the elapsed time between consecutive readings. The fraction of total physical cores per process, T_{ij} is calculated as in Equation 1.

$$T_{ij} = \frac{U_{i(j+1)} - U_{ij} + K_{i(j+1)} - K_{ij}}{100\Delta_t} \tag{1}$$

N samples are averaged to yield the average physical core fraction, T_i as given in Equation 2.

$$T_i = \frac{1}{N-1} \sum_{j=1}^{N-1} T_{ij} \tag{2}$$

With a multi-core architecture, T_i can exceed one since multiple processes can be scheduled concurrently. To calculate total clock cycles used, T_i is multiplied by the number of physical cores, P, and the nominal clock frequency F (Hz). Systems A, B, C, and E have hardware threading; System D does not. Only physical cores are counted since a single physical core

with two hardware threads can not run those threads concurrently. C_i, the average number of clock cycles used for process i, is given in Equation 3. This metric does implicitly assume that a scheduled process can always be parallelized, which is likely not true in practice. For example, a processor with two physical cores executing a serial process may be scheduled for equal amounts of user space time as a single core. This argument notwithstanding, this calculation provides a simple first-order metric.

$$C_i = T_i * F * P \tag{3}$$

All tests are presumed to be CPU-limited, with the exception of the process reading from the NIC. Since each DSP block corresponds to a particular process, it is clear which DSP operations use the majority of CPU time. Processes which do not consume significant CPU or are not critical in the signal processing algorithm are not shown for clarity. The testing procedure is as follows:

1. Start SRS for the specific test: analog video, FM radio with RBDS, or digital video.
2. Start the analysis software. Sample CPU utilization at 1/2 Hz for 200 seconds.
3. Note any anomalies such as data discontinuities or warnings.
4. Wait until the performance analyzer (nmon) has all required samples and close programs.

The next section discusses each test in detail. Video signals were selected to show real-time wideband signal processing. The analog video test compares systems B, C, and D. A digital video test distributes receiver tasks over multiple nodes and shows four out of every eight seconds of video to the user. The FM Radio with RBDS test demonstrates the possibility of running useful SDR applications on a single node.

2.1.4.1 Analog Video

This program processes standard definition analog video in either NTSC or PAL formats which has been frequency modulated and displays the video. Figure 2 provides a high-level overview of the processing steps involved. Processing blocks not directly relevant to video,

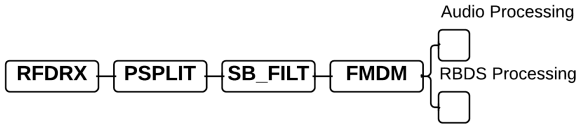
Figure 2: Analog Video Processing Block Diagram

or of low computational complexity, have been omitted for clarity. *RFDRX* and *PSPLIT* receives packet data and formats it appropriately. Wideband FM demodulation is performed by the *FMDM* block over 25 MHz, which requires arctangent and derivative operations. The *VID_FILT* block filters out the 6 MHZ wide video signal. Next, the *VSYNC* block frame synchronizes the video. *LC_FILT* filters the Luma and Chroma subcarriers, and *PHADJ* performs phase adjustment for color video extraction. With some code optimizations and optional signal processing operations turned off, it was eventually possible to run this test on System E. The effect of hardware threading on SDR performance was also analyzed.

2.1.4.2 FM Radio With RBDS

In this test, System A was used to receive a broadcast FM signal. This signal includes mono FM radio as well as the Radio Broadcast Data Service (RBDS). RBDS is a differentially encoded Binary Phase Shift Key (BPSK) modulated bit stream which contains digital information such as current time, station ID, or the name of the current song and artist. A block diagram of the receiver is shown in Figure 3. Audio and RBDS processing are performed but do not use significant CPU resources due to low sample rates and therefore are not shown.

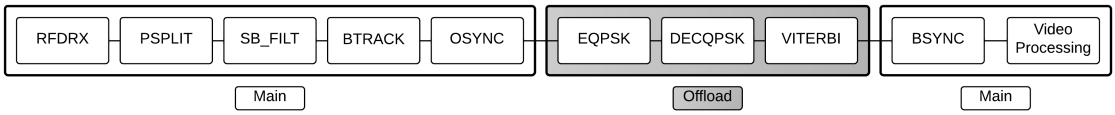
Figure 3: FM Receiver



2.1.4.3 Digital Video

Two System E platforms are required to process digital video, connected by gigabit ethernet. This test constructed a receiver for a video embedded in an MPEG2 transport stream using

Figure 4: Distributed Digital Video Processing Block Diagram



Quadrature Phase Shift Key (QPSK). The resulting color video is displayed to the user. Due to the processing requirements, a block decimation of two was used for an effective duty cycle of 50%. While this does not provide the user an adequate viewing experience, it does show that processing of a high data rate test signal is possible. The system could also be used for processing snapshots if full rate video is not required.

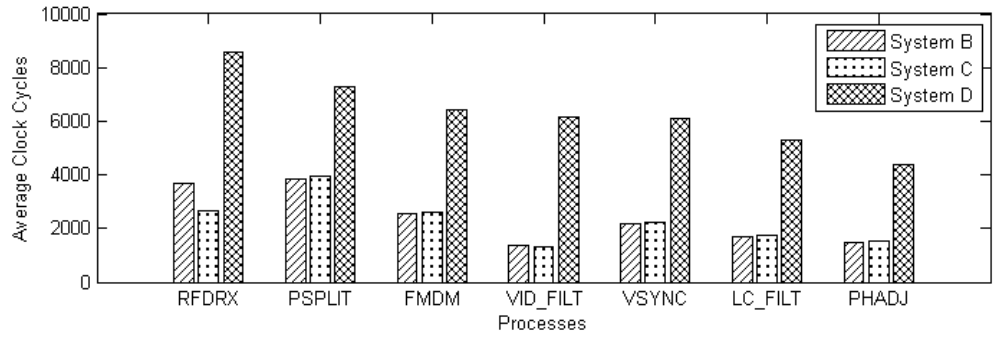
The processor groups the input samples into blocks of 100 megasamples and processes every other block. Figure 4 provides a block diagram of key functions in the software receiver. The main node receives spectrum samples from the RF digitizer, *SB_FILT* low-pass filters the signal with a 16 MHz cutoff , and *BTRACK* interpolates to produce one sample at the center of each symbol. Next, *OSYNC* performs symbol synchronization. The output of *OSYNC* is sent to the offload processing node via TCP/IP with a rate of 550 Mbps. The offload node receives the samples and performs channel equalization (*EQPSK*), symbol decoding, (*DECQPSK*), and performs the Viterbi algorithm (*VITERBI*). The output of this block is then sent back to the main node with a rate of 69 Mbps, where variable length frame synchronization occurs (*BSYNC*). Finally, the MPEG transport stream is processed.

A key design decision is identifying DSP blocks to be offloaded. Early in the chain the data typically has a high sample rate making offloading difficult due to bandwidth requirements. However, these blocks are also usually the ones which are most processor intensive. It is also possible to identify a group of DSP blocks which are good candidates for offloading due to their close interaction and independence from the rest of the processing. This is primarily a limitation of the SRS architecture due to tight coupling between the control and data planes. Other SDR software such as GNURadio may not suffer from this limitation, although wideband offloading will remain a challenge due to inter-system bandwidth constraints.

2.1.5 Results

2.1.5.1 Analog Video

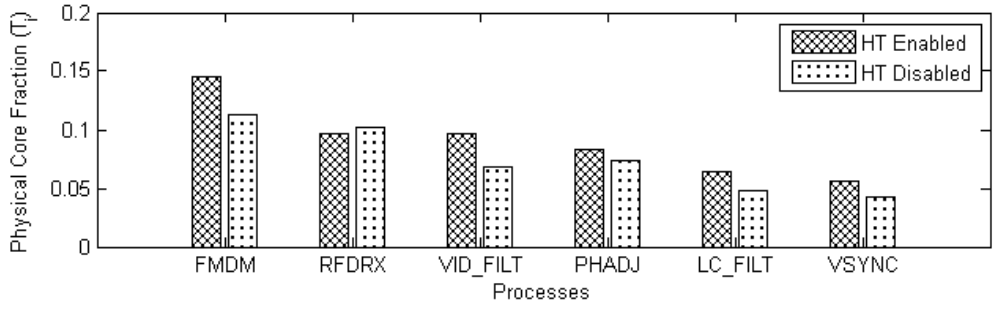
Figure 5: Analog Video Benchmark



The analog video test was run on Systems B, C, and D and C_i was calculated and plotted in Figure 5. Only these blocks described in the block diagram will be shown in the performance benchmark results. As can be seen, the RFDRX and PSPLIT blocks use significant resources compared to DSP operations. These blocks are responsible for high-throughput IO. Importing the samples over UDP requires significant overhead.

Once the samples are gathered, the DSP operations are remarkably efficient. The cost of gathering the samples for System B is 39 percent more than System C; however, both systems use the Intel 82579LM gigabit ethernet controller. This difference may be due to operating system differences as System C used RHEL 5.5 instead of RHEL 6.3. An

Figure 6: Hardware Threading Performance on System C

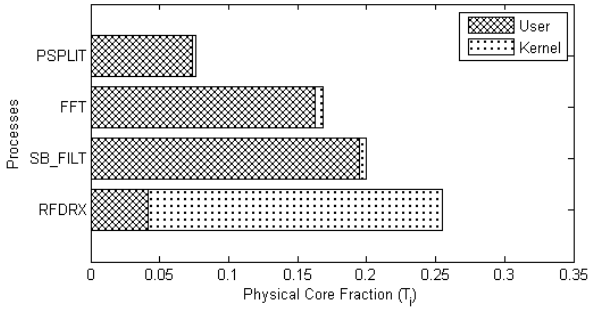


evaluation was also performed on System C to determine the impact of hardware threading on SDR software performance. Figure 6 illustrates an average *decrease* of 18% by disabling hardware threading. RFDRX exhibited little improvement, which is to be expected since

it is primarily IO bound. On multi-core systems, hardware threading may add unnecessary overhead since the SDR system relies on functional parallelism. Hardware threading may yield better performance gains for a large number of simple processes rather than a small number of computationally intensive ones.

2.1.5.2 FM Receiver

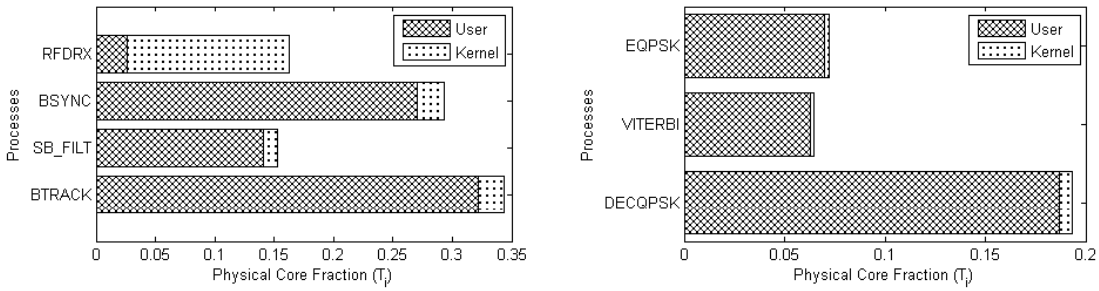
Figure 7: FM Radio Receiver



FM Radio receiver performance running on System A is shown in Figure 7, split by user and kernel processing time. About 80 percent of clock cycles for RFDRX is spent in kernel space, presumably retrieving samples from the RF Digitizer. The FFT is used to display spectrum to the user; this could be removed in a production application. This result shows a 10cmx10cm general purpose processor can perform meaningful SDR tasks, using approximately a single physical core to do so.

2.1.5.3 Digital Video

The digital video test ran for 20 minutes without any data discontinuities (other than introduced by the duty cycle) and the video was captured to disk. Figure 8a plots the physical core fraction for the main processing node, while Figure 8b shows this metric for the offload node. The most expensive receiver operations on the main node are interpolation to the center of the QPSK symbol and the variable frame length bit synchronizer. For the offload node, symbol decoding was almost twice as expensive as the Viterbi decoder or the channel equalizer. The main node used 1.46 physical cores while the offload node used 0.88. These benchmarks suggest that for GPP SDR tasks, a simple metric to evaluate



(a) Main Node (b) Offload Node

Figure 8: Main and Offload Node Video Processing

performance is the physical core-Hertz product. Due to the functional parallelism inherent in SDR applications, any single process which exceeds 100 percent of a core creates a bottleneck and must be optimized further. Multi-threading can be beneficial in this case, but not all processes can effectively use parallelism. Another consideration is the width and availability of SIMD operations on the GPP. All five platforms tested used Intel x86 family processors, four of which implement SIMD using AVX instructions.

Load balancing the cores is a secondary consideration. Hardware threading, such as Intel Hyperthreading, is beneficial when there are large number of non-compute-bound processes, but may actually impede performance in special cases of the reverse. SDR platforms will typically be of the later type since a few compute-bound processes such as interpolation or channel equalization at high sample rates dominate overall system performance.

2.1.6 Conclusions

Five different commodity PCs in four form factors were evaluated for GPP SDR applications. To demonstrate high data rate processing, software receivers were designed and tested for analog video, FM Radio with RBDS, and digital video signals. These tests demonstrate that GPP SDR is a real possibility in SWAP-constrained environments which require reconfigurability. The latest mobile x86 processors have adequate vector instructions, clock frequency, and number of physical cores to be used as a GPP hardware platform for experimentation in sensor networks. The eStadium team is currently building and testing an RF sensor network based on a GPP SDR architecture. Specifically, the next generation of

Intel's NUC hardware with the i5-4250U has been selected due to its relatively low power consumption (15W Max TDP). Next steps include the physical packaging of the system with an Ettus USRP and deployment in Bobby Dodd stadium. The network will be comprised of six nodes to collect and process spectrum data. Due to the flexibility of GPP SDRs, rapid prototyping and experimentation will be possible.

2.2 Extreme Emitter Density Testbed

To facilitate the prototyping and development of novel OSI cross-layer localization algorithms in an EED environment, the Intelligent Digital Communications (IDC) Vertically Integrated Projects (VIP) team has created and deployed a software radio sensor network testbed, LOC-EED, in Bobby Dodd Stadium. In parallel we have also developed and deployed a simplified laboratory version for controlled experimentation. The VIP program [23] is an engineering education program consisting of multidisciplinary teams of undergraduates, graduate students, and faculty advisors who collaborate on long term projects beneficial to current research. The undergraduate students help deploy and maintain the testbed while learning the associated theory and gain exposure to the latest research topics. Graduate students and advisors develop new theory and algorithms which can be validated in field experiments.

IDC is particularly interested in spectrum utilization, security, and localization in EED environments using software radio as the enabling technology. Therefore, the team has created LOC-EED which consists of RFSNs using wideband RF digitizers and general purpose processors to sense the RF environment. Each sensor is capable of recording and time-tagging RF spectrum samples at 25 complex MSPS. Captured spectrum data is stored on a central server for analysis and experimentation of localization algorithms.

The principle contributions of this section are:

- Architecture and practical deployment of an EED laboratory testbed and field deployment
- EED RF spectrum during a football game

- Simulation of a cross-layer localization technique

A brief description of previous testbeds is provided in Section 2.2.1. Design and deployment decisions, including both hardware and software, are detailed in Section 2.2.2. A preliminary data analysis of a WiFi channel during a football game is provided in Section 2.2.3. This analysis motivates the simulation of a simple cross-layer localization technique. Conclusions are discussed in Section 2.2.4.

2.2.1 Background

Other localization testbeds have been developed, but we are not aware of any specifically focusing on EED RF environments with a laboratory and field deployment. In [3], the authors consider only a single emitter whereas our laboratory testbed supports three. Additionally, LOC-EED laboratory uses cables to connect the software radios so the true time delay can be known. He et al. developed a testbed to experiment with indoor multipath localization using ToA for a single emitter [37]. Given the emitter and node geometry in the stadium, multipath conditions aren't as significant of a concern. However, additional data should be collected to verify this assumption. An RSSI approach for Wireless LAN is presented in [57], but dedicated hardware is used to process the signals making raw RF samples unavailable. Additionally, RSSI is not robust to RF environments due to the difficulties in modeling RF propagation [5]. Bhatti et. al. performed TDoA using software radios on two emitters. A WLAN TDoA system was presented in [72] but it is not clear the system has the flexibility of an SDR testbed or that Layer-2 information can be correlated with Layer-1 information.

2.2.2 Design and Deployment

LOC-EED consists of a laboratory testbed and field deployment; The former allows arbitrary geometries and interference situations to be simulated in a controlled manner, while the stadium version provides realistic field data. We utilize an iterative algorithm development approach. Algorithms are first simulated in software such as MATLAB. Next, the algorithm is implemented in the laboratory testbed with known inputs and then, if successful, deployed to the stadium nodes. Both the testbed and field deployment consist of

near identical node hardware. The primary differences are replacing free-space loss, sensor geometry, and wireless channels with attenuators, cabling, and splitters/combiners. The hardware and software design of the nodes are first presented and then the overall testbed architecture is discussed.

2.2.2.1 Hardware Design

Each RF sensor node consists of a direct-conversion RF digitizer, general purpose x86-based processor (GPP), Ethernet power relay, GPS Disciplined Oscillator, and a 2.4/5 GHz panel antenna. While there are many choices for implementing software radios, a GPP architecture was chosen because it has the key advantage of rapid algorithm prototyping [25]. The principle disadvantage of such an architecture is the limitation in processing power and bandwidth. However, it has been shown that small form factor GPPs are capable of processing up to 25 MHz of analog bandwidth for a variety of realistic tasks [31]. Additionally, since each GPP runs a standard Linux distribution, remote monitoring and maintenance tasks are simpler than on specialized DSP hardware. The specific parts used to build each RFSN is provided in Table 3.

The target deployment area for LOC-EED is in the stadium, typically in an outdoor location which is not readily accessible. For example, the first sensor was deployed on top of a 15 foot tall concession stand requiring an extension ladder for service. This creates the additional requirements of weatherproofing, small form factor, and remote monitoring for health and status. All components of each RFSN are placed inside an NEMA-rated enclosure with watertight connectors, as shown in Figure 9. For remote monitoring, a temperature sensor was placed inside the enclosure. The Ethernet power relay provides a method to cycle power should the node have any issues.

A narrowband antenna was selected due to the direct-conversion architecture of the RF digitizer. We discovered during testing that broadband antennas, while much more flexible, can not be used without a suitable RF front-end. When attempting to use broadband antennas to capture 2.4/5 Ghz spectrum, the SINR was insufficient for signal processing due to the lack of front-end analog filters in the receiver to reduce strong out-of-band

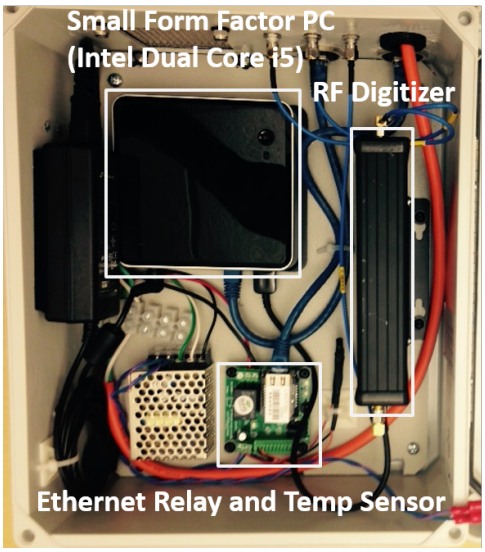


Figure 9: RFSN Components. The GPSDO is inside the RF digitizer enclosure

Table 3: RFSN Components

Manufacturer	Model	Description
Nat'l Instruments	782980-01	RF Digitizer
Nat'l Instruments	783454-01	GPS Oscillator
Intel	BOXD54250WYK	Haswell i5 NUC PC
Samsung	MZ-MTE1T0BW	1TB Solid State Disk
Crucial	BLS2K8G3N169ES4	16GB DDR3 RAM
Nat'l Control Devices	R110PL_ETHERNET	Ethernet Relay
L-COM	HG2458-20P	2.4/5GHz Antenna

signals. This hardware limitation reduces the range of frequencies which can be studied in the testbed. However, given the abundance of interesting signals in the band selected, this should not be a significant limitation. An alternative is to purchase RF digitizers which have a superheterodyne receive architecture and suitable RF front-end filtering, but this is outside of the current project budget.

2.2.2.2 Software Design

Each RFSN is installed with Ubuntu 14.04 LTS and GNUradio (GR)[2]. The disk is partitioned into an EXT4 and XFS partition, for applications and recording storage, respectively.

[2] `http://www.gnuradio.org`

Ubuntu 14.04 LTS was selected for its excellent consumer hardware and community support. GR is an open source platform for signal processing which has many common filters, demodulators, and other useful algorithms. It is particularly suited for wideband real-time processing by exploiting SIMD processor instructions and efficient DSP algorithms.

Support for data analysis is still under development in GR. We are currently developing gr-analysis, a module for GR which contains the following additional tools to record and analyze data. In the future we plan to make the module available to other researchers as well as the GR community.

- *specrec*: Recording utility capable of 30 MSPS on RFSNs
- *metadata_to_csv*: Convert metadata structure to CSV
- *gr_mkheader*: Add metadata to existing raw data records
- *gr_fileman*: Convert file formats, select recording subsection

The data recording utility, *specrec*, was developed out of a desire to investigate WiFi localization techniques. Due to RFSN size and power constraints, a RAID0 configuration for data storage is impractical. The file recording program example in GR, *uhd_rx_cfile*, drops samples due to Linux kernel buffering causing write bursts. When the bursts write to disk the maximum write speed is insufficient to maintain the required average. For RFSN hardware, *uhd_rx_cfile* begins to drop samples between 15-20 MSPS, while *specrec* can write 30 MSPS with no data loss. *uhd_rx_cfile* was passed the *-m* option to record inline metadata, whereas *specrec* uses a separate file to store the metadata (detached headers). *uhd_rx_cfile* also drops samples at 30 MSPS without writing any metadata.

specrec implements a producer-consumer multi-threaded architecture with a circular buffer. The writes from each thread are a multiple of the system page size. All pages associated with the subsection of the circular buffer to be written to disk are flushed using the *sync_file_range* kernel system call. The end result is a constant write speed at the expense of some additional CPU utilization. This recording program is Linux-only, but can increase write speeds by roughly a factor of two.

Health and status monitoring is provided the widely available CACTI software. In addition to the monitoring of the CPU temperatures, hard disk space, and other sensors of interest, the ambient temperature is monitored with a thermocouple and displayed on a webpage. Additionally, software can power cycle the node via the ethernet relay.

2.2.2.3 Laboratory LOC-EED

Figure 10 depicts the laboratory LOC-EED setup. Each box represents an RFSN, which consists of the hardware described in Section 2.2.2.1 except for the panel antenna and GPSDO. The GPSDO is replaced with a Jackson Labs' LC-XO providing 10 MHz and 1PPS outputs for receiver synchronization. The Splitter/Combiner (S and C) used is a Minicircuits ZX10-4-27+. With this setup, different TDoAs can be simulated. The TDoA between sensor j and k from emitter i is given by

$$\tau_{jk}^{(i)} = \frac{L_{ij} - L_{ik}}{v} = \frac{1}{v}\left(\|\boldsymbol{q}_j - \boldsymbol{p}_i\|_2 - \|\boldsymbol{q}_k - \boldsymbol{p}_i\|_2\right) \tag{4}$$

v is the propagation velocity of the wave which is cable-specific and $\boldsymbol{p}_i$, $\boldsymbol{q}_j$, and $\boldsymbol{q}_k$ are the position vectors of emitter i, sensor j, and sensor k, respectively. L_{ij} and L_{ik} are the cable lengths from emitter i to sensors j and k. The matrix $\boldsymbol{A} \in \mathcal{R}^{MxN}$ can control the sensor geometry, where M is the number of emitters and N is the number of unique TDoAs. Physically, these delays will be created by using cables of appropriate lengths.

$$\boldsymbol{A} = \begin{bmatrix} \tau_{12}^{(1)} & \tau_{13}^{(1)} \\ \tau_{12}^{(2)} & \tau_{13}^{(2)} \\ \tau_{12}^{(3)} & \tau_{13}^{(3)} \end{bmatrix} \tag{5}$$

A major advantage of using software radio nodes as opposed to specialized hardware in the testbed is the ability to change physical operating parameters such as the center frequency, modulation type, bandwidth, etc. Consider the case of localizing 20 MHz OFDM WiFi with IEEE 802.15.4 interference. One might ask how some physical layer parameters affect WiFi localization accuracy. This is easily simulated using the gr-ieee802-11 [13] and the gr-ieee802-15-4 [12] GNURadio modules. Other sources of localization error such as receiver synchronization are also easy to replicate. Thus the laboratory version of LOC-EED

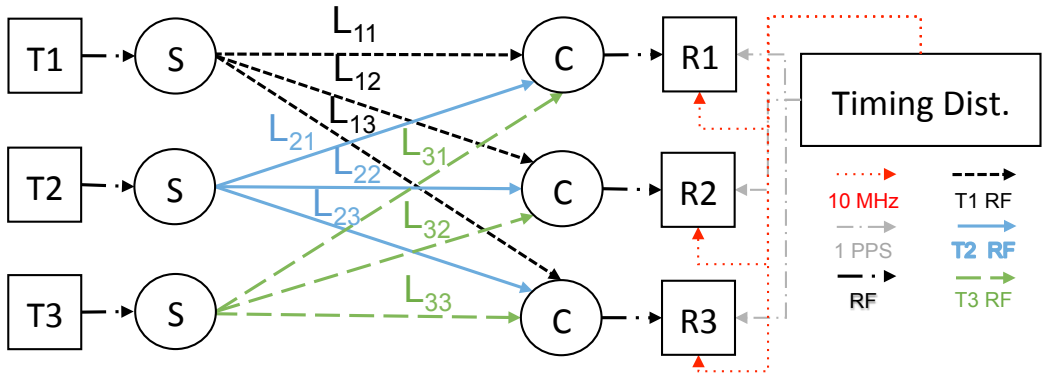


Figure 10: Laboratory LOC-EED. T1-T3 represent RFSNs which are cabled to splitters, labeled S. The cable lengths from emitter i to sensor k are L_{ik}. The combiner, C, sums the signals from all transmitters into R1-R3, which are also RFSNs. 10 MHz and 1 PPS references are distributed to all nodes for time synchronization.

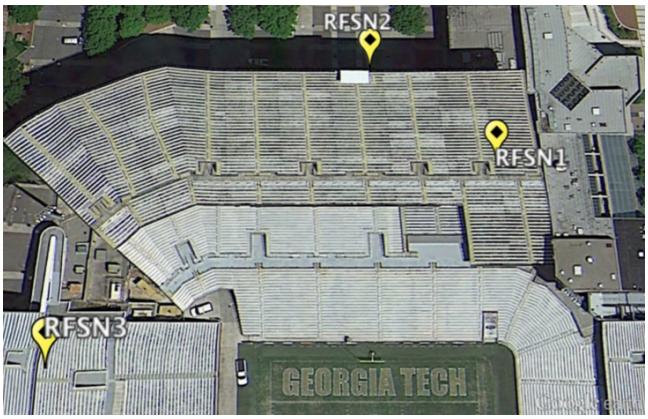


Figure 11: Stadium LOC-EED. RFSN1 is currently deployed. Google Earth.

provides a controlled environment for experimentation of algorithms with known inputs before they are applied to realistic field data.

2.2.2.4 Stadium LOC-EED

The sensor network within Bobby Dodd is shown in Figure 11. Currently, only RFSN1 is deployed. RFSNs 2 and 3 will be deployed in time for the upcoming football season. A particular challenge in the stadium is identifying mounting locations as the nodes require both gigabit ethernet for tasking and data backhaul as well as 120 VAC outlet power. Additionally, the antennas must be located relatively close to the nodes and have an acceptable field of view. These practical constraints impose sub-optimal sensor geometries.

RFSN1 was deployed on top of a concession stand, where power and a gigabit campus network connection was available. Figure 12a shows RFSN1 as the enclosure on the left connected to the router on the right. The antenna was mounted on a concrete support angled out over the field, as seen in Figure 12b using 50' of LMR-400. The antenna has since been enclosed by an RF-transparent billboard. No studies have been undertaken to assess the performance difference but it is assumed minimal as the AT&T Distributed Antenna System (DAS) operates under the same conditions.

(a) Deployment on top of a concession stand. RFSN1 is on the left, while the enclosure on the right houses an AP for connectivity to campus network .

(b) RFSN1 2.4/5GHz antenna mounted in the stadium

Figure 12: RFSN1 Deployment

2.2.3 Analysis and Simulation

Figure 13 is the spectrogram of channel 6 WiFi (2437 MHz) on gameday. The received spectrum is dense and highly non-stationary. The wideband signals present are indeed WiFi but the narrowband signals have not been identified. This data capture can be categorized as multiple known emitters with multiple unknown narrowband emitters. How can a particular WiFi signal be isolated to perform a localization technique such as TDoA? This preliminary data motivates a simulation.

Consider a simplified simulated test case where two WLAN emitters, E_1, E_2 and two sensors, S_1, S_2 are present. S_1 receives a sampled complex baseband spectrum of a WLAN

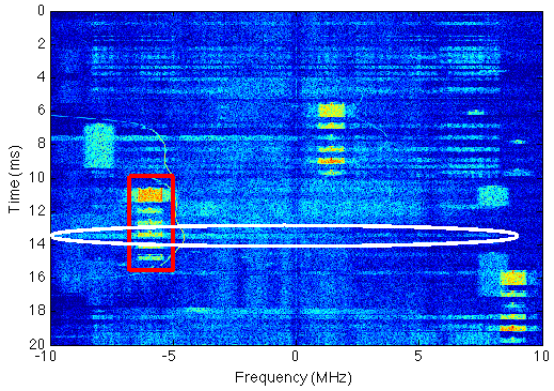


Figure 13: Ch. 6 WiFi (2437 MHz) during a football game. The white circle is an OFDM-modulated WLAN packet. The red square represents an unknown narrowband interferer. A variable frequency sinusoid can also be seen from [6,15] ms and [-10,-5] MHz.

signal in additive white Gaussian noise. Assume the real and imaginary noise are statistically independent.

$$r_1[n] = \sum_{i=1}^{2} s_i[n] + e_1[n], n = 0, \ldots, N-1 \tag{6}$$

where $e_1[n] \sim \mathcal{CN}(0, \sigma_1^2 \boldsymbol{I}_N)$ and $s_i[n]$ are the sampled WLAN signals at baseband. For this simulation, the signal is OFDM with an MCS of 0 (BPSK with coding rate = 1/2) and 20 MHz channel spacing. Each signal will contain a unique transmitter MAC address and it will further be assumed the signals share the channel without interfering with one another. S_2 receives the same signal with a delay m due to sensor geometry. For simplicity, an assumption has been made that $m = 10$, implying an integer delay.

$$r_2[n] = \sum_{i=1}^{2} s_i[n+m] + e_2[n] \tag{7}$$

Here, $e_2[n] \sim \mathcal{CN}(0, \sigma_2^2 \boldsymbol{I}_N)$. Assume the noise powers are such that $\sigma_1^2 < \sigma_2^2$ and S_1 can correctly demodulate the signal while S_2 has insufficient SNR. For the test scenario the SNRs were 13 dB and 3 dB, at S_1 and S_2, respectively.

The autocorrelation method given in [13] was used to identify the start of WLAN packets. Each time the autocorrelation method exceeded the threshold, the sample number, n_{ac} at which the peak occurred was recorded. If the packet was successfully demodulated, the transmitter MAC address is used to label the particular emitter as E_i. n_{ac} is associated with this label in the form of a tuple (n_{ac}, E_i) The fusion of Layer-1 and Layer-2 information

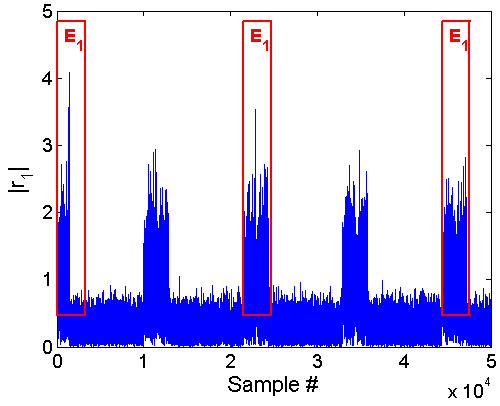


Figure 14: $|r_1[n]|$ with Emitter E_1 identified. When the cross-correlation exceeded a threshold, the sample number n_{ac} was recorded. The WLAN packet was subsequently decoded and the emitter labeled based on MAC address. This information was associated as a tuple (n_{ac}, E_i). The left side of the red box is placed at n_{ac} and labeled accordingly.

allows the ith known emitter to be labeled in the time domain plot. Figure 14 provides an example. This method does not require all MAC addresses of the emitters to be known and they are guaranteed to be unique provided no MAC address spoofing is present.

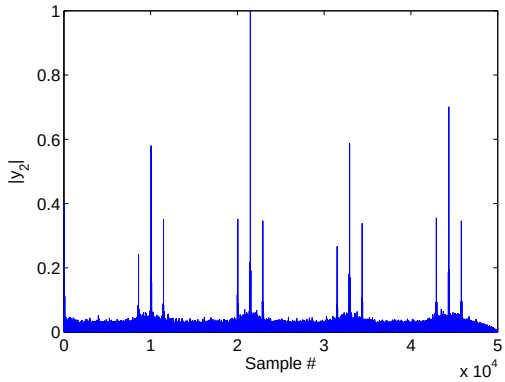
Although S_2 is unable to demodulate the signal, it is possible to uniquely identify E_1 in the received signal, $r_2[n]$. Consider the second E_1 transmission in Figure 14. Use the samples associated with this packet to create a matched filter, $p[n]$. $p[n]$ is then used to cross-correlate with $r_2[n]$, as in Equation 8 with i=2. Figure 15a plots $|y_2[m]|$, while Figure 15b graphs y_1 and y_2 around the maximum in Figure 15a. The difference between the two peaks is $m = 10$ samples, which is what was expected. This algorithm shows a particular receiver with insufficient SNR to demodulate an emitter can still uniquely identify it with help from another sensor's cross-layer information to generate the matched filter.

$$y_i[n] = \sum_{m=0}^{M} r_i[n+m]p^*[m], i = 1, 2 \tag{8}$$

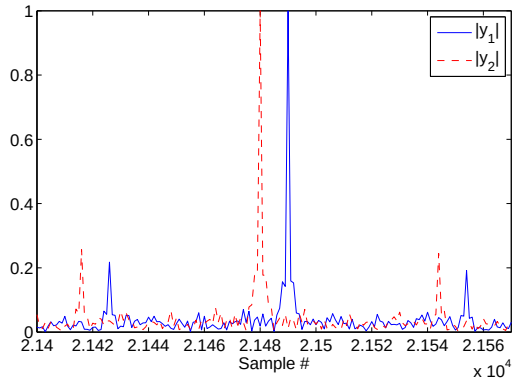
This simple example illustrates the power of cross-layer techniques to isolate a particular emitter. The MAC addresses of a WLAN signal are contained in the payload which is scrambled and has forward error correction. Because of this encoding it is necessary to perform the full demodulation to identify a particular emitter since the MAC address bits can't be linearly mapped to a sample position. Therefore, identification of a particular

transmitter by MAC address *requires* Layer-2 information. One can not simply cross-correlate certain samples at the physical layer to uniquely identify the transmission.

Additional Layer-2 techniques are possible. For example, a challenge in using the MUSIC algorithm for TDoA estimation is determining the number of emitters. Using Layer-2 information such as the number of unique MAC addresses, or number of clients connected to an AP can inform this Layer-1 algorithm. Stationary WLAN emitters may be identified by locating APs. This information could directly inform the TDoA solution since it is unlikely there is a Doppler shift. These possibilities should be investigated to create localization algorithms robust to interference.



(a) Cross-Correlation of the received signal r_2 and the template p. Since every WLAN packet contains the same short and long preamble, every packet has some degree of correlation which can clearly be seen on the graph. However, the maximum is still located at the correct packet and emitter.



(b) Close-up of $y1$ and $y2$ at the sample corresponding to the maximum cross-correlation of y_1. The difference between these peaks is 10 samples, which is the simulated delay.

Figure 15: Layer-1/Layer-2 Correlation

2.2.4 Conclusions

Two localization testbeds for EED RF environments were described in detail: A laboratory version and a system deployed in the football stadium. Additionally, the software architecture was discussed, including the custom *gr-analysis* module for data analysis. A spectrogram from a live football game was shown, illustrating the spectrum density of 2.4 GHz as well as the presence of narrowband interferers with wideband WLAN signals. Finally, a simulation showing the possibilities of cross-layer techniques was presented. Future

work should investigate exploiting Layer-2 information to create robust localization algorithms in EED RF environments.

CHAPTER III

MAC ASSISTED DATA ASSOCIATION

With multiple emitters, there is ambiguity in assigning a given sequence of physical layer measurements (e.g. time of arrivals) from the sensors to one of the emitters. A similar problem exists in radar when multiple targets are present. In the radar literature, the problem is known as *data association*. Typical techniques relying on target kinematics and position are of minimal use as the kinematic cost matrix elements are virtually identical in high target density environments [53]. To solve this problem, more recent research proposes feature-assisted tracking for radar, using such measurements as radar cross section (RCS) to aid in data association [28].

Of course, in radar there are no such concepts as the OSI model as there are for RF emitters operating under a specific communication protocol. This work studies the performance improvement of using OSI Layer 2 (L2) information as features in the data association problem compared to Layer 1 (L1) alone. Using such information blurs the line between a traditional radar, signal processing, and networking problem. The presentation that follows should be one familiar to signal processing engineers, although we try to add clarity for radar engineers where necessary.

The core idea is to exploit the structure of the MAC layer, as well as packet level correlations at Layer 2 and above, to associate physical layer measurements with emitters. For WLANs, these are the CSMA/CA protocol and packet exchange sequences, respectively. We assume packet decodability at each sensor is random and may be quite low. To our knowledge, using Layer 2 information as features in the data association problem is novel.

The main contributions of this chapter are:

1. Problem formulation and introduction of a Markov model to couple physical layer measurements with higher level side information.

2. Analysis of an RTS/CTS PES using the IEEE 802.11g standard for both single and

multiple clients. The PAE over the entire PES is lower for strategies employing the MAC layer.

3. For multiple AP clients, the Layer 2 strategy may be viable in EED RF environments.. The PAE over an entire PES does not vanish for a large number of clients.

Although the RTS/CTS PES for IEEE 802.11g is used for analysis, the approach is applicable to other PESs and, more broadly, communication standards employing a MAC layer.

Section 3.1 provides a brief summary of background material and relevant literature. The system model and assumptions are given in Section 3.2. Section 3.3 formulates the problem. The performance analysis is performed in two sections. Section 3.4 considers a single AP client, while Section 3.5 analyzes multiple clients as well as the asymptotic behavior in the number of clients. Section 3.6 provides a hypothesis test for model validity. Finally, conclusions are drawn in Section 3.7.

3.1 Background

There are two primarily threads of literature from two separate communities which are useful for emitter-measurement association. Using the entire protocol stack blurs the line between a traditional radar, signal processing, and networking problem. Therefore it is necessary to understand the existing work from both contexts for a complete survey. The DSP and networking problem of RF fingerprinting is discussed in Section 3.1.1. Radar engineers typically discuss associating measurements to targets as the data association problem in the context of multitarget, multisensor target tracking. Research in this area is discussed in Section 3.1.2. The problem formulations and applications are slightly different, but both are especially relevant and provide needed insight into the proposed research problem.

3.1.1 RF Fingerprinting

RF fingerprinting is the idea of using either channel or emitter specific characteristics for identification. Radiometric identification refers to the latter, using physical imperfections and process variations of emitter electronics for identification [17]. Another term, specific

emitter identification (SEI), appears to be used primarily in the defense community and may predate the radiometric identification literature [48]. A high level overview of SEI systems to identify emitters of interest is given by Talbot *et al* [77]. Typically the focus with RF fingerprinting is for network security. The argument is that layer 2 information which uniquely identifies devices (e.g. MAC address for WLANs) is susceptible to spoofing and may not be trustworthy. The goal of RF fingerprinting is therefore to uniquely identify devices solely based on layer 1 information for user validation and security.

Channel-based RF fingerprinting techniques leverage the fact that the RF channel between two emitters is likely different. One simple measurement which can be used is the received signal strength indicator (RSSI). Faria and Cheriton consider a vector of RSSI measurements, termed *signalprints*, from multiple APs to uniquely identify transmissions [29]. A matching algorithm is described based on acceptable RSSI bounds and evaluated in an office environment using IEEE 802.11b/g APs. Sheng *et al.* [74] extends RSSI measurement identification using Gaussian Mixture Models to more recent IEEE 802.11 standards employing antenna diversity. Emitters are identified statistically using a likelihood ratio test. If the emitters are in a rich multipath environment, the full channel response may be used. Patwari and Kasera [61] considered a minimum proximity function clustering method using the $\mathcal{L}_2$ dissimilarity measure based on channel impulse response features. Le *et al.* use channel tap power in the context of cognitive radio to distinguish between a primary user and a malicious secondary user. Notability, a cross-layer algorithm is proposed which combines the physical layer measurements with higher level authentication [49].

Radiometric Identification literature includes both machine learning and model-based algorithms. Either way, physical layer features are chosen to discriminate between emitters. Both techniques can also use transient or steady-state transmitter behavior. Only steady state is considered here; for more information on RF transient behavior for characterization see [80, 65].

Many authors have applied machine learning algorithms to signal features in order to identify emitters. Brik *et al.* [17] developed the PARADIS system using K-Nearest Neighbors (KNN) and Support Vector Machines (SVM) to classify WLAN cards. PARADIS

uses physical layer features including RF center frequency and I/Q offset, among others. Frequency offset and SYNC correlation were the most effective features. The learning algorithms were evaluated on 138 identical NICs and achieved an accuracy greater than 99 percent. However, such techniques from a security perspective are vulnerable to replay attacks, especially by software radios, as shown by Danev *et al.* [24]. In Candore *et al.* [18] transmitter features including frequency, magnitude, and I/Q offset were used to train a classifier. A histogram from training data was calculated and the features were combined using a voting-based algorithm. Similarly, Tomko *et al.* [79] also uses estimates of the feature probability distributions (including frequency offset) for IEEE 802.11b devices. A Gaussian distribution was fit to the smoothed estimates. If the fit coefficients change sufficiently over time it is assumed the MAC address has been spoofed.

Another approach is to explicitly assume a mathematical model for the imperfections in the emitter electronics. In Dolatshahi *et al.* [27] non-linear input and output characteristics of RF power amplifiers are modeled and subsequently used for radiometric identification. The likelihood (LRT) and generalized likelihood (GLRT) ratio tests are used to distinguish between two emitters. This work was extended to include the digital-to-analog (DAC) imperfections and additional experimental data [62]. The robustness of the technique to nefarious symbol modifications is shown analytically and experimentally in Polak and Goeckel [63]. A technique applicable to `802.11b` using the envelope profile of a preamble is presented by Yuan and Hu [88]. Vo-huu *et al.* consider the scrambling seed, sampling and carrier frequency offset, and frame transient as features to distinguish `IEEE 802.11g` devices, but use statistical methods tailored to each feature (e.g. KL Divergence) [84]. Fundamental limits of RF fingerprint authentication from an information theoretic perspective are discussed by Gungor and Koksal [35].

3.1.2 Data Association

Data association has been well studied in the radar community as a part of the target tracking problem. A summary of the problem and prior work in the context of radar is discussed in this section. For the following section emitters can be considered targets.

Additional background information on radar fundamentals may be found in [66, 53]. A good reference for multitarget multisensor tracking is Bar-Shalom and Li [7].

At each coherent processing interval (CPI) a list of detections is created. Detections typically consist of a measurement, measurement error estimate, timestamp, and possibly other metadata. If the system consists of multiple sensors, the first step is measurement-to-measurement association. This step associates measurements from each sensor to create a composite measurement (e.g. a vector of time of arrival measurements). Next, the composite measurement must be assigned an existing track, or a new one created. This is the measurement-to-track data association problem. The goal is to associate detections with tracks [66]. Each track consists of a state vector and corresponding covariance matrix $\boldsymbol{D}$ representing the state error. The measurement error is usually assumed to be Gaussian.

Typically, a cost matrix $\boldsymbol{C}$ is first populated with the negative log likelihood of assigning detection i to track j. The rows of $\boldsymbol{C}$ represent existing tracks, as well as a new track ϕ. Similarly, a specific column represents the i^{th} measurement with the last column labeled ϕ signifying a track which will not be updated for the current CPI.

A gating step, consisting of coarse and fine filters, is used to eliminate extremely unlikely associations. Coarse gates can either be spherical or rectangular and ensure measurement-track pairs are bounded within some desired radius or rectangle. Next, a fine filter is applied by computing the log likelihood for measurement i and track j as given in Equation 9 [53]. A Kalman Filter is used on the tracks for smoothing and prediction.

$$\Lambda_{ji} = -\frac{1}{2}\ln\left(\det\left(2\pi \boldsymbol{S}_{ji}\right)\right) - \frac{1}{2}\tilde{\boldsymbol{z}}_{ji}^{T}\boldsymbol{S}_{ji}\tilde{\boldsymbol{z}}_{ji} \tag{9}$$

In Equation 9, $\tilde{\boldsymbol{z}_{ji}} = \boldsymbol{z}_i - h(\boldsymbol{x}_j)$ is the innovation vector from the Kalman Filter, $\boldsymbol{z}_i$ is the measurement state, $\boldsymbol{x}_j$ is the predicted track state, and $h()$ is a function which transforms a track state into the measurement space. The covariance matrix of the innovations is $\boldsymbol{S}_{ji}$. The second term is essentially the Mahalanobis distance. Tracks with larger innovation variances are penalized by the first term.

The cost matrix is populated by using the negative log likelihood $-\Lambda_{ji}$. Such a matrix is referred to as the kinematic assignment matrix. A data association algorithm is

then used to find the most likely associations. In [68], the authors provide a summary of current techniques including Nearest Neighbor (NN), joint probabilistic data association (JPDA), multiple hypothesis tracking (MHT), and multidimensional assignment (MDA). In the assignment formulation of the problem, discrete optimization is used to associate measurements with targets [26]. The technique was further extended to correlated measurements such as TDoA in [67].

Other algorithms for data association have been proposed. *Bhatti et. al.* developed a phase closure method [10] to associate the physical measurements with a particular emitter. However, there are ambiguities as to the position of the emitter using this approach. Another method relies on separating the measurements with some *a priori* information about the expected measurement range [8].

In dense target environments, the kinematic cost matrix elements may be nearly identical due to the target density [53]. For such a scenario a feature-based term can be added to the likelihood function in Equation 9 to aid in association. For example, the signal-to-noise (SNR) ratio is considered as a feature for radar [28]. Other tracking examples from radar include using amplitude [50] and local target motion [39] as features.

3.2 System Model

Suppose $S_1, S_2, \ldots, S_M$ are M spatially distributed, time synchronized RF sensor nodes that can communicate with one another. N stationary emitters $E_1, E_2, \ldots, E_N$ transmit a signal using a known standard but are non-collaborative with the sensors. Non-collaborative implies the sensors and AP/clients do not share information, but does not necessarily mean the AP/clients are actively attempting to disrupt measurements. Future work could explore other relationships. Each sensor has the capability to measure time of arrival (ToA) and, given sufficient SNR, decode packets. The decodability of each packet at each sensor is assumed to be probabilistic.

The objective is to localize the emitters. In radar, this translates to the tracking of the (x, y) cartesian coordinates of all emitters. Without proper data association, only a single

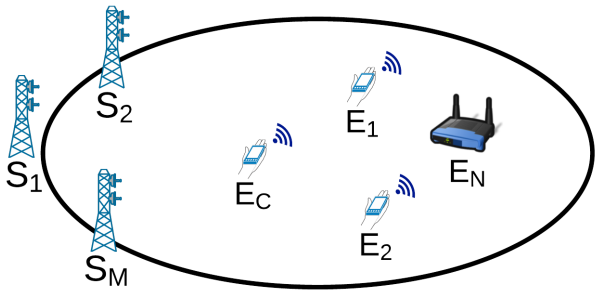


Figure 16: System Diagram. The AP communicates with C clients using IEEE 802.11g. The AP and clients do not collaborate with sensors $S_1, \ldots, S_M$. The sensors use Layer 1 and, when possible, Layer 2 information to localize emitters $E_1, E_2, \ldots, E_N$.

measurement can be used with confidence. Under the assumption of no measurement-to-measurement errors and no missed detections, a position estimate using a single measurement from the i^{th} detection can be made. However, multiple measurements can not be used to decrease localization error without data association because the measurements may be from different emitters. In this case, the extra measurements from other sensors would likely *increase* the localization error. It is highly probable that multiple measurements will be collected from each emitter.

Specifically, an IEEE 802.11g network in Infrastructure BSS mode using the Distributed Coordination Function (DCF) is considered. The MAC mechanism is therefore CSMA/CA and the PHY is chosen as OFDM. Error-free transmissions between the access point (AP) and each client are assumed. Furthermore, no emitter enters the exponential backoff procedure and each client has the same amount of data to send.

The consequences of these assumptions are that a PES, also referred to as a frame exchange sequence in the standard [41], always succeeds and all clients are equally likely to capture the channel. For this work, a single AP is associated with C clients, thus $N = C+1$. Each sensor passively observes the WLAN channel spectrum and is not associated with the AP. Figure 16 provides an illustration of the setup.

The N_d measurements collected by each sensor are complex baseband samples with timestamps. For the m^{th} sensor's samples, the i^{th} packet arrival time estimate is denoted

as $\hat{t}_m[i]$ and calculated by a time delay estimation algorithm such as maximizing the cross-correlation.

$$\hat{\boldsymbol{t}}_i \triangleq \begin{bmatrix} \hat{t}_1[i] & \hat{t}_2[i] & \dots & \hat{t}_M[i] \end{bmatrix}^T, i = 0, 1, \dots, N_d - 1 \tag{10}$$

If the WLAN packet is decodable by the sensor, then the packet P_i is associated with the corresponding ToA. If the packet detection threshold is crossed but the packet is undecodable, then a dummy packet is inserted for that particular ToA measurement. By taking the time difference between a reference sensor ToA, arbitrarily S_1, and all other sensors, the measurement TDoA vector is formed. These correspond to measurements in radar terminology.

$$\widehat{\boldsymbol{\Delta t}}_i \triangleq \begin{bmatrix} \hat{t}_{21}[i] & \hat{t}_{31}[i] & \dots & \hat{t}_{M1}[i] \end{bmatrix}^T, \hat{t}_{m1} \triangleq \hat{t}_m[i] - \hat{t}_1[i] \tag{11}$$

A single detection is defined as the two-tuple: $\left(\widehat{\boldsymbol{\Delta t}}_i, P_i\right)$.

The emitters only send a single RTS/CTS PES for the present analysis. As the primary purpose of this work is to explore the benefits of L2 information compared with L1, not all possible PESs are considered. Further analysis will be required for practical implementation.

It is assumed that there is no measurement-to-measurement association error. This is reasonable given a constrained geometry such that the maximum difference between ToA measurements is much smaller than the DCF interframe spacing (DIFS). In our application, the maximum TDoA possible in the stadium is 750ns, while the minimum DIFS is 28 μs. Thus measurement-to-measurement ambiguity should be of little concern.

Furthermore, it is assumed there are no missed detections. For practical application in the stadium, this will need to be relaxed. The primary objective of this paper is to explore the benefits of the MAC Layer in the data association problem compared to the physical layer. We avoid this additional complication for now and reserve it for future work.

Let N_a be the number of packets per PES. For example, in RTS/CTS $N_a = 4$ provided no packet fragmentation occurs. Given a detection sequence of length N_d,

$$(\widehat{\boldsymbol{\Delta t}}_0, P_0), (\widehat{\boldsymbol{\Delta t}}_1, P_1), \dots, (\widehat{\boldsymbol{\Delta t}}_{N_d-1}, P_{N_d-1}) \tag{12}$$

the goal is to associate emitters $E_1, E_2, \dots, E_N$ to each detection. Under these assumptions, $N_a = N_d$ as there are no missed detections and only a single PES is considered. The

kinematic-only cost matrix $\boldsymbol{C}$ has nearly identical elements, implying the cost function has virtually no discriminatory ability. Therefore, the data association problem depends only on features.

The approach is to make use of a Homogeneous Markov Chain model to couple a particular E_n with a packet. That is, each Markov model state is a two-tuple, or three-tuple for $C > 1$, consisting of a L2 packet type and emitter for a given PES. The time index i in the Markov model is the detection index. Transition probabilities and states depend on the PES and the CSMA/CA algorithm. Defining the model in this way allows packet level correlations only available at L2 to be exploited in the data association problem. The following sections formulate the problem and analyze the RTS/CTS PES.

3.3 Problem Formulation

Regardless of the packet sequence under consideration, the general problem is formulated and notation defined before discussing the specifics of the RTS/CTS sequence in Sections 3.4 and 3.5. Table 4 provides a summary of the notation used throughout this paper, in order of appearance. Upper-case bold symbols denote matrices, while lower-case bold symbols are vectors. Bars over symbols indicate averages, while hats denote estimates. Script upper-case letters are sets. $\mathbb{E}\{\}$ is the expected value of a random variable, and $\mathcal{P}\{\}$ is the probability.

Consider a single PES with associated detection set. The task is to assign an $E_n, n \in \{1, 2, \ldots, N\}$ to every detection within the set. Let Y_i be a random variable representing the true emitter index at detection i

$$\Omega_Y = \{E_1, E_2, \ldots, E_N\}, Y_i(\omega) = n \text{ if } \omega = E_n \tag{13}$$

and $\hat{Y}_i$ be the emitter index estimate for the i^{th} detection. To assess performance, consider a simple 0/1 loss in associating an emitter to the i^{th} detection. This is the per packet PAE.

$$\epsilon_i = 1 \text{ if } \hat{Y}_i \neq Y_i, 0 \text{ if } \hat{Y}_i = Y_i \tag{14}$$

The number of errors in a PES, Q_x, is therefore $Q_x = \sum_{i=1}^{N_a} \epsilon_i$. Define the per packet exchange sequence PAE as $\mathcal{P}\{Q_x > 0\}$. This is the error in making any association mistake

Table 4: Data Association Notation

Symbol	Description
S_m	m^{th} sensor
M	No. of sensors
N	No. of emitters
E_n	n^{th} emitter
C	No. of AP clients
N_d	No. of detections
$t_m[i]$	i^{th} ToA at S_m
P_i	i^{th} Packet
$\widehat{\boldsymbol{\Delta t}}_i$	TDoA estimate for i^{th} packet
N_a	# of packets per PES
Y_i	Emitter index R.V. at packet i
Ω_Y	Emitter index sample space
ϵ_i	Association error indicator for P_i
Q_x	No. of association errors per PES
$\mathbb{E}\{\epsilon_i\}$	Per packet PAE
$\mathcal{P}\{Q_x > 0\}$	Per sequence PAE
X_i	State R.V. at time i
Ω_X	State sample space
$\mathcal{B}$	Set of possible MAC packet types
$\omega_X(j)$	A particular state j
$\varnothing$	Dummy emitter in state label
$p_j(i)$	Probability of $\omega_X(j)$ at detection i
$\boldsymbol{p}(i)$	Unconditional state probability vector at time i
$\boldsymbol{P}$	Transition probability matrix
D_i	Decodability of P_i
p	Global sensor probability of decoding a packet
γ_n	SNR per symbol for emitter E_n.
$\Delta\gamma$	Difference in SNRs per symbol
ξ	Ratio of $\Delta\overline{\gamma}$ to $\overline{\gamma}$
K	No. of decodable packets in a PES
p_0	Local sensor probability of decoding a packet
q	Probability AP initiates RTS/CTS
B	R.V. representing MAC packet type
π_0	Bernoulli R.V. representing an RTS collision
L_{RTS}	Length of an RTS packet
L_{CTS}	Length of a CTS packet

over an entire sequence. In order to calculate this error, the probability of a particular emitter index n for detection i, $\mathcal{P}\{Y_i = n\}$ is required. This quantity can be calculated using the Markov model.

For one client ($C = 1$), it is sufficient to define X_i as a R.V. representing the Markov state at detection i with the sample space given by

$$\Omega_X = \{(a_1, a_2) | a_1 \in \mathcal{B}, a_2 \in \Omega_Y\} \tag{15}$$

where $\mathcal{B}$ is a set consisting of all possible MAC packet types in a given PES (e.g. RTS, ACK, etc). Particular states are enumerated and denoted as $\omega_X(j)$ where $\omega_X(j) \in \Omega_X$, $j = 1, 2, \ldots, |\Omega_X|$.

For $C > 1$, a three-tuple state label is required as the state transitions depend on whether the AP or another $E_n, n = 1, 2, \ldots C$ transmitted the first packet. In this case,

$$\Omega_X = \{(a_1, a_2, a_3) | a_1 \in \mathcal{B}, a_2 \in \Omega_Y, a_3 \in \Omega_C\}, \Omega_C = \{\varnothing, E_1, E_2, \ldots, E_C\} \tag{16}$$

where $\varnothing$ is a dummy emitter indicating the third element is not necessary to define a particular state. Denote $p_j(i)$ as the unconditional probability of a given state $\omega_X(j)$ at detection $i, i = 0, 1, \ldots, N_d - 1$. That is, $p_j(i) = \mathcal{P}\{X_i = \omega_X(j)\}$. The unconditional state probability vector is defined as $\boldsymbol{p}(i) \triangleq \begin{bmatrix} p_1(i) & p_2(i) & \ldots & p_{|\Omega_X|}(i) \end{bmatrix}^T$. By Chapman-Kolmogorov,

$$\boldsymbol{p}(i) = \left(\boldsymbol{P}^i\right)^T \boldsymbol{p}(0) \tag{17}$$

where $\boldsymbol{P}$ is the $|\Omega_X|$x$|\Omega_X|$ one-step state transition probability matrix and $\boldsymbol{p}(0)$ is the initial state probability vector [60]. The j, kth element of $\boldsymbol{P}$ is

$$\boldsymbol{P}[j, k] = \mathcal{P}\{X_{i+1} = \omega_X(k) | X_i = \omega_X(j)\} \tag{18}$$

To find the probability of a particular emitter index n,

$$\mathcal{P}\{Y_i = n\} = \sum_{\omega_X \in \Theta(n)} \mathcal{P}\{X_i = \omega_X\} = \mathbb{1}_{\Theta(n)}^T \left(\boldsymbol{P}^i\right)^T \boldsymbol{p}(0) \tag{19}$$

where $\mathbb{1}_{\Theta(n)}$ is the $|\Omega_X|$x1 indicator vector. That is, $\mathbb{1}_{\Theta(n)}[j] = 1$ if $\omega_X(j) \in \Theta(n)$ and $\Theta(n)$ is the set of all states containing the n^{th} emitter. For $C = 1$ and $C > 1$, respectively,

$$\Theta(n) = \{\omega_x = (a_1, a_2) \in \Omega_X | a_2 = E_n\}, \Theta(n) = \{\omega_x = (a_1, a_2, a_3) \in \Omega_X | a_2 = E_n\} \tag{20}$$

Table 5: MAC Frame ID, AP sends RTS

Packet Type	Client Known
RTS	✓
CTS	×
DATA	✓
ACK	×

3.4 Single Client Analysis

Consider the Request-to-Send/Clear-to-Send (RTS/CTS) PES without fragmentation; hence $N_a = 4$ packets. It is assumed that both the AP and client use RTS/CTS. In reality it is much more likely that the client uses RTS and the AP does not. Since the AP is associated with the clients, it can presumably hear all clients and therefore RTS/CTS is of little benefit. This assumption can be relaxed in future work.

The state diagram for $C = 1$ client is shown in Figure 17, where $AP = E_N$ for notational convenience. Thus $Y_i = N$ corresponds to the AP being assigned to the i^{th} detection. While the models are simple and do not allow for failed transmissions, they provide a starting point for exploring how packet level correlations can assist in the data association problem. Relaxing the assumptions to allow for failed transmissions and multiple sequences requires modeling of the binary exponential backoff procedure, which is reserved for future work. This may be complicated and lead to a state space explosion as there are many possibilities [11].

The possible MAC packets for RTS/CTS are $\mathcal{B} = \{RTS, CTS, DATA, ACK\}$. Not all MAC packets contain the MAC addresses of the two emitters which are communicating. Suppose a client and the AP are communicating, with AP sending the RTS. Table 5 describes which emitter identities are known in each packet.

From the illustration in Figure 17, $\boldsymbol{P}$ and $\boldsymbol{p}(0)$ can be defined. Suppose the state random variable sequence is X_1, X_2, X_3, X_4 for the PES.

3.4.1 PHY-Only (L1)

Without knowledge of the CSMA/CA MAC protocol or PES, a decision should be made separately at each packet. The L1 feature considered is the SNR. The rational for this

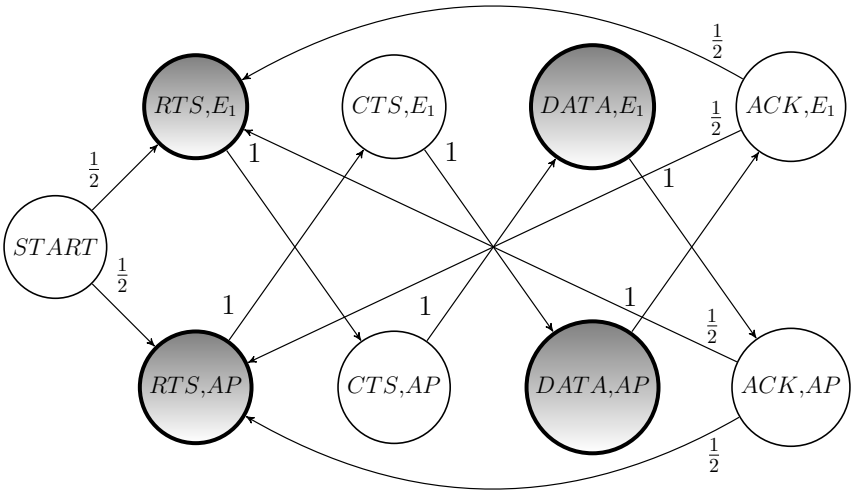


Figure 17: RTS/CTS packet exchange sequence state diagram for one client. Extra thick and shaded circles are states with corresponding MAC packets containing both emitter IDs (c.f. Table 5). State order is: (RTS, E_1), (RTS, AP),$(CTS, E1)$, ...

selection is that it is commonly provided by commercial APs and channel models are well-studied in the communications literature. Other physical layer features could be used as discussed in Section 3.1.1, which likely yield better performance. This analysis is restricted to showing the benefits of L2 and hence not all physical layer features are considered.

If the packet is undecodable, then the strategy is a simple hypothesis test to associate the detection to an emitter. It will be assumed the average SNRs are known for the emitters.

For decodable packets, the MAC addresses are used for association. It is reasonable to allow the L1 strategy to use the MAC addresses since the bits are known at the physical layer. The contrast with the L2 strategy is that knowing the i^{th} association does not imply anything about the $(i+1)^{th}$ association. In other words, packet level correlations are ignored. Additionally, using the MAC addresses at L1 provides a more insightful reference strategy to compare against L2. It more fully captures the advantages gained by exploiting packet level correlations and the CSMA/CA MAC protocol. It also normalizes for the fact that the emitter identification is sometimes contained in the packet itself.

The task is to calculate the per packet PAE for the i^{th} packet, $\mathcal{P}_{L1}\{Y_i \neq \hat{Y}_i\} = \mathbb{E}_{L1}\{\epsilon_i\}$, where the subscript L1 is a reminder that the calculation is restricted to using only L1 information. This will then be extended, assuming independent errors, to calculate the

PAE over the entire PES. First, condition on the decodability of packet P_i as the strategy pursued depends on this quantity. Suppose $D_i \sim$ bernoulli(p) is the R.V. representing the decodability of packet P_i in the exchange sequence. Assume the D_i are i.i.d over all packets. Here p represents the *global* probability of decoding P_i among all M sensors. For now, consider $M = 1$.

$$\mathcal{P}_{L1}\{Y_i \neq \hat{Y}_i\} = \mathcal{P}\{Y_i \neq \hat{Y}_i | D_i = 0\}\mathcal{P}\{D_i = 0\} + \mathcal{P}\{Y_i \neq \hat{Y}_i | D_i = 1\}\mathcal{P}\{D_i = 1\} \tag{21}$$

If $D_i = 1$, then P_i is decodable and $\mathcal{P}\{Y_i \neq \hat{Y}_i | D_i = 1\} = 0$ as the MAC addresses are used and hence there is no possibility of making an error in association. Even if the packet type is a CTS or an ACK, both emitters are still known as the other one can be inferred. For $C > 1$, this will not be the case.

Per the assumption, $\mathcal{P}\{D_i = 0\} = 1 - p$. To evaluate $\mathcal{P}\{Y_i \neq \hat{Y}_i | D_i = 0\}$, the probability of error needs to be computed for the SNR hypothesis test. To do so, a channel model and SNR distribution must be assumed. Consider a Ricean flat fading channel model with additive white Gaussian noise (AWGN). The *Nakagami* distribution can be used to approximate the Rice distribution and has a form which is often easier to work with analytically [76]. The instantaneous SNR per symbol γ has a distribution given by

$$p_\gamma(\gamma; \overline{\gamma}) = \frac{m^m \gamma^{m-1}}{\overline{\gamma}^m \Gamma(m)} e^{-\frac{m\gamma}{\overline{\gamma}}}, \gamma \geq 0, m = \frac{(1 + K_0)^2}{1 + 2K_0} \tag{22}$$

where K_0 is the Rice factor which controls the ratio of line-of-sight (LoS) to scatterer power [75], $\overline{\gamma}$ is the average SNR, and $\Gamma(\cdot)$ is the Gamma function. Assume emitters E_1 and E_2 have average per-symbol SNRs $\overline{\gamma}_1$ and $\overline{\gamma}_2$, respectively. Without loss of generality, suppose $\overline{\gamma}_2 > \overline{\gamma}_1$.

$$\mathcal{H}_0 : \gamma \sim p_\gamma(\gamma; \overline{\gamma}_1), \mathcal{H}_1 : \gamma \sim p_\gamma(\gamma; \overline{\gamma}_2) \tag{23}$$

The hypothesis are equally likely by Figure 17. The resulting test is given in Equation 24.

$$\gamma \underset{\mathcal{H}_0}{\overset{\mathcal{H}_1}{\gtrless}} \eta, \eta \triangleq \frac{\overline{\gamma}_1 \overline{\gamma}_2 \ln\left(\frac{\overline{\gamma}_2}{\overline{\gamma}_1}\right)}{\overline{\gamma}_2 - \overline{\gamma}_1} \tag{24}$$

Next, calculate the per packet conditional PAE.

$$\mathcal{P}\{Y_i \neq \hat{Y}_i | D_i = 0\} = \frac{m^m}{2\Gamma(m)} \frac{1}{\overline{\gamma_2}^m} \int_0^\eta \gamma^{m-1} e^{-\frac{m\gamma}{\overline{\gamma}_2}} d\gamma + \frac{m^m}{2\Gamma(m)} \frac{1}{\overline{\gamma_1}^m} \int_\eta^\infty \gamma^{m-1} e^{-\frac{m\gamma}{\overline{\gamma}_1}} d\gamma \tag{25}$$

Rewrite the average SNRs as $\overline{\gamma}_2 = \overline{\gamma}_1 + \Delta\overline{\gamma}, \Delta\overline{\gamma} \geq 0$. Then define $\xi \triangleq \frac{\Delta\overline{\gamma}}{\overline{\gamma}_1}$. The per packet conditional PAE will only depend on this "aspect" ratio. Noting $\Gamma(a,x) = \int_x^\infty t^{a-1}e^{-t}dt$ is the upper incomplete gamma function and defining $\mathcal{P}_e(\xi) \triangleq \mathcal{P}\{Y_i \neq \hat{Y}_i | D_i = 0\}$, the final result is given in Equation 26.

$$\mathcal{P}_e(\xi) = \frac{1}{2} - \frac{\Gamma\left(m, \frac{m\ln(\xi+1)}{\xi}\right)}{2\Gamma(m)} + \frac{\Gamma\left(m, \frac{m(\xi+1)\ln(\xi+1)}{\xi}\right)}{2\Gamma(m)} \tag{26}$$

Equation 26 gives the per packet conditional PAE as a function of ξ. Notice $\lim_{\xi\to\infty}\mathcal{P}_e(\xi) = 0$ since the second sum term's numerator becomes $\Gamma(m,0) = \Gamma(m)$ and the third vanishes.

There is a helpful physical interpretation of Equation 26. Consider SNR as some measure of distance. However, we are careful to not to assume any explicit mapping as the accuracy of such RSSI-based techniques are often poor in practice and highly environment dependent [5, 38]. For a fixed separation between emitters corresponding to a fixed $\Delta\overline{\gamma}$, ξ is large for small $\overline{\gamma}_1$. This implies as the sensor moves further away from the pair of emitters, the PAE decreases. If the distance between the sensor and first emitter is fixed, then consider $\overline{\gamma}_1$ constant, implying ξ is large for large $\Delta\overline{\gamma}$. The interpretation is that larger separation between emitters lowers the PAE. A visualization is a triangle with E_1, E_2 and S_1 as vertices and $\overline{\gamma}_1$ and $\Delta\overline{\gamma}$ as edges. Substituting Equation 26 into Equation 21 gives the unconditional per packet PAE.

$$\mathbb{E}_{L1}\{\epsilon_i\} = \mathcal{P}_{L1}\{Y_i \neq \hat{Y}_i\} = (1-p)\,\mathcal{P}_e(\xi) \tag{27}$$

Next, the per sequence PAE is derived for the Layer One strategy. Recall that $Q_x \sim$ binomial $(4, \mathbb{E}\{\epsilon_i\})$ if the "successes" (incorrect assignments) are independent since ϵ_i is a Bernoulli R.V. This is a justifiable assumption because knowledge of previous assignment correctness should not influence the current emitter guess unless L2 provides that side information. Also, although the other detections may have corresponding packets which are decodable, this should not influence the per packet PAE. Allowing these packet correlations implies knowledge of the PES and CSMA/CA, which subtlety violates the assumption of using only L1 information. Equation 28 gives the per sequence PAE using only L1

information.

$$\mathcal{P}_{L1}\{Q_x > 0\} = 1 - \mathcal{P}_{L1}\{Q_x = 0\} = 1 - \binom{4}{0}\mathbb{E}\{\epsilon_i\}^0(1 - \mathbb{E}\{\epsilon_i\})^4 = 1 - (p\mathcal{P}_e(\xi))^4 \quad (28)$$

3.4.2 MAC-Only (L2)

The key realization for exploiting L2 information is that observing any of the X_i makes the other three known due to the structure of the PES. Therefore, decoding a single packet is sufficient to correctly assign all packets to emitters. This is unique to the single client case because CTS and ACK MAC packets only contain a single emitter ID. For $C > 1$, exactly which packet was decoded is of importance.

Recall D_i represents the decodability of the i^{th} packet. Assuming the D_i are i.i.d., then $K \sim$ binomial(N_a, p) represents the number of decodable packets in a PES. Using the Markov model of the MAC, first compute the complementary probability. Note the subscript L2 indicates the use of only L2 information.

$$\begin{aligned}\mathcal{P}_{L2}\{Q_x = 0\} &= \mathcal{P}\{Q_x = 0|K \geq 1\}\mathcal{P}\{K \geq 1\} + \mathcal{P}\{Q_x = 0|K = 0\}\mathcal{P}\{K = 0\} \\ &= 1 - (1-p)^4 + (1-p)^4\mathcal{P}\{Q_x = 0|K = 0\} \end{aligned} \quad (29)$$

This follows because decoding at least one packet leaves no assignment ambiguity.

Although the packet is not decodable, the structure of the PES suggests that sequences E_1, AP, E_1, AP or AP, E_1, AP, E_1 should be guessed with equal probability. The guess is correct with probability 0.5 since by Equation 19 $\mathcal{P}\{Y_i = n\} = \frac{1}{2}$ for all i and n.

$$\mathcal{P}_{L2}\{Q_x > 0\} = \frac{1}{2}(1-p)^4 \quad (30)$$

The final result is given by Equation 30. Compare this strategy with the L1 approach, where in the absence of L2 knowledge, emitters are guessed independently for each measurement.

For comparison to the L1 strategy, it will be helpful to have a per packet probability of association error. As shown above, there are correlations between detections which influence the probability of error. As such, consider an average per packet PAE, $\overline{\mathcal{P}}_{L2}\{Y_i \neq \hat{Y}_i\}$.

$$\mathcal{P}_{L2}\{Y_0 \neq \hat{Y}_0\} = \mathcal{P}_{L2}\{Y_0 \neq \hat{Y}_0|D_0 = 0\}\mathcal{P}\{D_0 = 0\} + \mathcal{P}_{L2}\{Y_0 \neq \hat{Y}_0|D_0 = 1\}\mathcal{P}\{D_0 = 1\} \quad (31)$$

The second sum term is zero as the packet is decodable. Note $\mathcal{P}\{Y_0 = n\} = \frac{1}{2} \forall i, n$.

$$\mathcal{P}_{L2}\{Y_0 \neq \hat{Y}_0 | D_0 = 0\} = \frac{1}{2}\mathcal{P}_{L2}\{\hat{Y}_0 \neq 1 | D_0 = 0, Y_0 = 1\} + \frac{1}{2}\mathcal{P}_{L2}\{\hat{Y}_0 \neq 2 | D_0 = 0, Y_0 = 2\} \quad (32)$$

Guess the emitters with equal probability and the wrong assignment is made with probability $\frac{1}{2}$.

$$\mathcal{P}_{L2}\{Y_0 \neq \hat{Y}_0\} = \frac{1-p}{2} \quad (33)$$

By similar arguments, $\mathcal{P}_{L2}\{Y_1 \neq \hat{Y}_1\} = (1-p)\,\mathcal{P}_{L2}\{Y_1 \neq \hat{Y}_1 | D_1 = 0\}$. Since L2 information is available, information from the previous detection can be used for the current association. Specifically, if P_0 was decodable, then no mistake is made.

$$\frac{\mathcal{P}_{L2}\{Y_1 \neq \hat{Y}_1 | D_1 = 0\}}{1-p} = \mathcal{P}_{L2}\{Y_1 \neq \hat{Y}_1 | D_1 = 0, D_0 = 0\} \quad (34)$$

If P_0 was not decodable, then the PES and CSMA/CA suggests an association. Condition on the previous guess $\hat{Y}_0$, then guess the other emitter.

$$\mathcal{P}_{L2}\{\hat{Y}_1 = 2 | \hat{Y}_0 = 1, D_0 = 0\} = 1, \mathcal{P}_{L2}\{\hat{Y}_1 = 1 | \hat{Y}_0 = 2, D_0 = 0\} = 1 \quad (35)$$

The conditional probability becomes

$$\begin{aligned}\mathcal{P}_{L2}\{Y_1 \neq \hat{Y}_1 | D_1 = 0, D_0 = 0\} = {} & \mathcal{P}_{L2}\{Y_1 = 1 | D_1 = 0, D_0 = 0, \hat{Y}_0 = 1\}\mathcal{P}\{\hat{Y}_0 = 1 | D_0 = 0\} \\ & + \mathcal{P}_{L2}\{Y_1 = 2 | D_1 = 0, D_0 = 0, \hat{Y}_0 = 2\}\mathcal{P}\{\hat{Y}_0 = 2 | D_0 = 0\} \quad (36)\end{aligned}$$

Note by previous work $\mathcal{P}\{\hat{Y}_0 = 1 | D_0 = 0\} = \mathcal{P}\{\hat{Y}_0 = 2 | D_0 = 0\} = \frac{1}{2}$ and by Equation 19 this conditional probability evaluates to $\frac{1}{2}$. Therefore, the per packet PAE is

$$\mathcal{P}_{L2}\{Y_1 \neq \hat{Y}_1\} = \frac{(1-p)^2}{2} \quad (37)$$

By similar arguments, $\mathcal{P}\{Y_2 \neq \hat{Y}_2\} = \frac{1}{2}(1-p)^3$ and $\mathcal{P}\{Y_3 \neq \hat{Y}_3\} = \frac{1}{2}(1-p)^4$. The average per packet PAE for the L2 strategy is given in Equation 38.

$$\overline{\mathcal{P}}_{L2}\{Y_i \neq \hat{Y}_i\} = \frac{1}{8}\sum_{i=1}^{4}(1-p)^i \quad (38)$$

3.4.3 MAC-Assisted (L1/L2)

This strategy combines the physical and MAC layers to lower the PAE.

$$\min_{\hat{\boldsymbol{Y}}} \sum_{i=1}^{4} \mathbb{E}\{\epsilon_i\} = \min_{\hat{\boldsymbol{Y}}} \mathcal{P}\{Y_i \neq \hat{Y}_i\}, \hat{\boldsymbol{Y}} \triangleq \begin{bmatrix} \hat{Y}_0 & \dots & \hat{Y}_3 \end{bmatrix}^T \quad (39)$$

Since $\mathbb{E}\{\epsilon_i\} \geq 0$, it is sufficient to minimize the equation term-by-term.

$$\min \mathbb{P}\{Y_0 \neq \hat{Y}_0\} = 0 \text{ if } \cup_{i=1}^{4} D_i \geq 1, \min\left(\frac{1}{2}, \mathcal{P}_e(\xi)\right) \text{ o.w.}$$

$$\begin{aligned}\mathcal{P}\{Y_0 \neq \hat{Y}_0\} &= \mathcal{P}\{Y_0 \neq \hat{Y}_0 | \cup_{i=1}^{4} D_i \geq 1\}\mathcal{P}\{\cup_{i=1}^{4} D_i \geq 1\} \\ &\quad + \mathcal{P}\{Y_0 \neq \hat{Y}_0 | \cup_{i=1}^{4} D_i = 0\}\mathcal{P}\{\cup_{i=1}^{4} D_i = 0\} = (1-p)^4 \mathcal{P}_e(\xi) \end{aligned} \tag{40}$$

For subsequent guesses, use the L1 decision from the first association if packets are undecodable.

$$\min \mathcal{P}\{Y_i \neq \hat{Y}_1\} = \begin{cases} 0 & \cup_{i=1}^{4} D_i \geq 1 \\ Z = \mathcal{P}\{Y_1 \neq \hat{Y}_1 | Y_0 = \hat{Y}_0\}\mathcal{P}\{Y_0 = Y_0\} & \\ +\mathcal{P}\{Y_1 \neq \hat{Y}_1 | Y_0 \neq \hat{Y}_0\}\mathcal{P}\{Y_0 \neq Y_0\} & \text{o.w.} \end{cases} \tag{41}$$

Note that $\mathcal{P}\{Y_1 \neq \hat{Y}_1 | Y_0 = \hat{Y}_0\} = \mathcal{P}\{Y_1 \neq \hat{Y}_1 | Y_0 \neq \hat{Y}_0\} = 0$. Knowing if the first association is correct (or incorrect) is all the information which is required to make an error free association for the sequence. The final result is given by Equation 42.

$$\mathcal{P}_{L1/L2}\{Q_x > 0\} = \mathcal{P}_e\left(\xi\right)\left(1-p\right)^4 \tag{42}$$

Assuming the per symbol SNR random variables γ are i.i.d. on a per packet basis, then this result carries over for using any one of four L1 decisions. Other MAC layer assisted strategies can be imagined, such as using all four L1 SNR measurements.

3.4.4 Probability of Error as a Function of SNR

One reasonable question to investigate is how the L1 strategy compares to L2 as a function of average SNR received at the sensor from E_1 packets. At a sufficiently high SNR, one expects the both association errors to vanish, whereas in the low SNR region the physical layer approach may be superior. What is not clear, however, is the performance for the moderate SNR region when packets are occasionally decodable. Consider the independent variable as $\overline{\gamma}_{1(dB)} = 10\log_{10}\left(\gamma_1\right)$ and compare the two strategies in terms of average per packet PAE and per sequence PAE.

For the following comparisons, the two emitters are assumed to use a Modulation and Coding Scheme (MCS) of 2, representing QPSK modulation with a rate $R = \frac{1}{2}$. For QPSK,

the energy per symbol is equal for all data symbols due to phase modulation [64]. The difference in energy of the preamble and signal fields, which may have a different MCS than the data, is ignored.

To map $\overline{\gamma}_{1(dB)}$ to an average probability of decoding a packet $\bar{p}$, simulation is employed due to the complexities of analysis. Other simulation options are available [59, 47, 44].

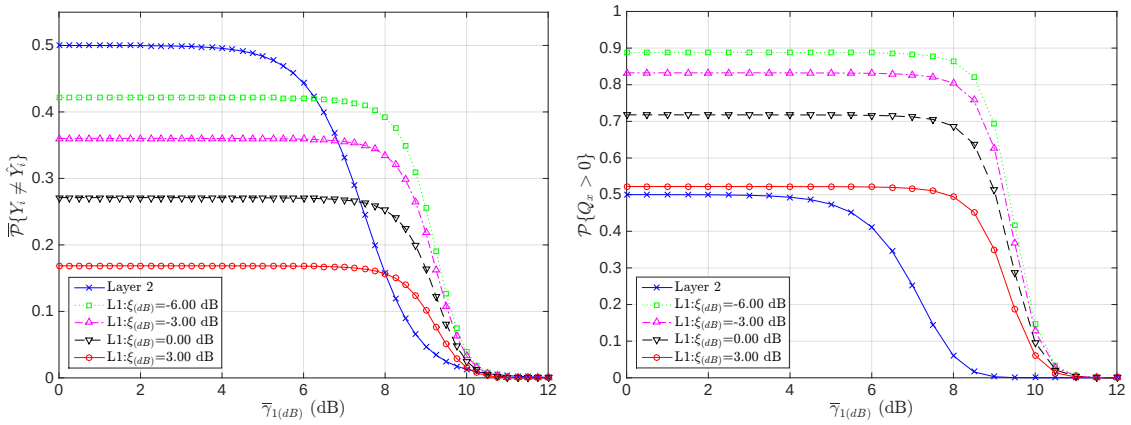
The simulation itself was performed with the `GNURadio` software radio toolkit using the `gr-ieee80211` module [14]. The flat fader channel model with a Rice Factor of $K_0 = 5$ and AWGN were added to the IEEE 802.11g PHY. For each SNR, the probability of receiving a packet was averaged over 20 trials of 100 packets each. A generalized logistic function

$$\bar{p}\left(\overline{\gamma}_{1(dB)}\right) = \frac{1}{1 + \exp\left(-\alpha\left(\overline{\gamma}_{1(dB)} - \beta\right)\right)} \tag{43}$$

with parameters α and β was fit to the simulated average probability of decoding a packet over a range of SNRs using the *nlinfit* non-linear regression `MATLAB` function. The root mean square error (RMSE) for the fit is 6.23×10^{-3}. The fit parameters for QPSK 1/2 are $\alpha = 1.32$ and $\beta = 8.27$. Mappings can easily be found for other modulations and values of K_0.

The L1 and L2 strategies are compared in terms of the per packet PAE. Substituting Equation 43 into Equations 27 and 38 for p gives these functions in terms of $\overline{\gamma}_{1(dB)}$ and ξ. The per packet PAE is plotted in Figure 18a for various $\xi_{(dB)} = 10\log_{10}\xi$. The L1 strategy is superior on a per detection basis until $\overline{\gamma}_{1(dB)}$ is between 6.5 and 8 dB, with the exact intersection dependent upon $\xi_{(dB)}$. A larger ξ produces a lower per packet PAE. Recall the L1 strategy does allow for choosing associations based on MAC addresses for the present detection if the packet is decodable. For the sufficiently high SNR region where packets are sometimes decodable, it is clear exploiting the structure of L2 is advantageous over and above simply decoding the MAC addresses of the emitters and using them for data association.

The advantage of the L2 strategy becomes more significant when looking at the per



(a) (b)

Figure 18: The average per packet and per packet exchange sequence probability of association error is shown in 18a and 18b, respectively for a single client and sensor. The independent variable is average SNR per symbol (dB) received at the sensor from emitter E_1. The various $\xi_{(dB)}$ curves represent the ratio of the SNR difference (in dB) received at the sensor between emitters to the SNR from E_1. The packet decode probability mapping to SNR assumes QPSK $\frac{1}{2}$ in a Ricean channel with $K_0 = 5$.

sequence PAE. Substituting Equation 43 into Equation 28 and using the relevant fit parameters yields the L1 PAE as a function of $\overline{\gamma}_{1(dB)}$ and $\xi_{(dB)}$.

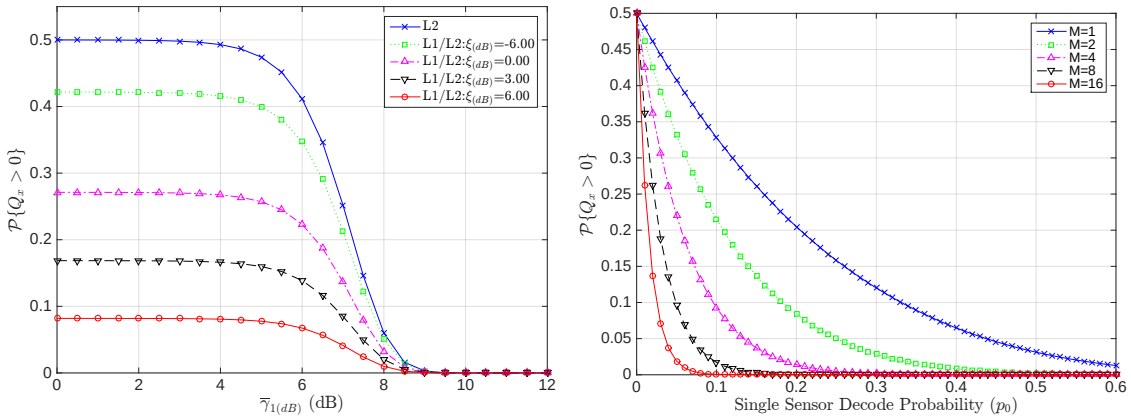
$$\mathcal{P}_{L1}\{Q_x > 0\} = 1 - \left(\frac{\mathcal{P}_e\left(10^{\xi_{(dB)}/10}\right)}{1 + \exp\left(-\alpha\left(\overline{\gamma}_{1(dB)} - \beta\right)\right)} \right)^4 \tag{44}$$

Similarly, substitute Equation 43 into Equation 30 for p.

$$\mathcal{P}_{L2}\{Q_x > 0\} = 1 - \left(\frac{1}{1 + \exp\left(\alpha\left(\overline{\gamma}_{1(dB)} - \beta\right)\right)} \right)^4 \tag{45}$$

Figure 18b plots Equations 44 and 45 as a function of $\overline{\gamma}_{1(dB)}$ with $\xi_{(dB)}$ fixed at various values. As expected, at high SNR the packets are decodable and there are no errors in association. At low SNR, the performance of the L1 strategy improves as $\xi_{(dB)}$ increases. The most interesting SNR region is when $\overline{\gamma}_{1(dB)} \in [5, 11]$ dB. At $\overline{\gamma}_{1(dB)} = 5$ dB, the average decode probability is $\overline{p} = 0.0134$. At $\overline{\gamma}_{1(dB)} = 11$ dB, the packets are almost always decodable with $\overline{p} = 0.9733$. If some packets are occasionally decodable, using L2 information can significantly outperform L1.

It is also interesting to compare Figure 18b to the per packet PAE shown in Figure 18a. Although the L1 strategy outperforms L2 on a per packet basis for sufficiently small $\overline{\gamma}_{1(dB)}$,



(a) (b)

Figure 19: 19a shows the per packet exchange sequence probability of association error for one client with fixed ξ. L1/L2 and L2 represent the MAC-Only and MAC-Assisted strategies, respectively. The various $\xi_{(dB)}$ curves represent the ratio of the SNR difference (in dB) received at the sensor between emitters to the SNR from E_1. 19b plots the per packet exchange sequence probability of association error for one client and multiple sensors using Layer 2 information.

on a per sequence basis the L2 strategy is superior for all $\overline{\gamma}_{1(dB)}$ for the $\xi_{(dB)}$ considered here. These figures suggest L2 side information is most helpful in the data association problem when associations must be made over entire packet exchange sequences.

The combined L1/L2 strategy per sequence PAE using a physical layer decision for a single measurement is plotted in Figure 19a. Comparison with Figure 18b demonstrates the benefits of using cross-layer information. The PAE can likely be lowered by using all available SNR measurements to make the physical layer decision. That analysis is reserved for future work.

3.4.5 Multiple Sensors

Recall p is the *global* probability that P_i is decodable. Given M sensors, each with *local* decode probability p_0, only a single sensor need decode P_i as it is assumed the sensors can share the decoded bits from the packet. Assume the decodability of the P_i are independent.

$$p = 1 - (1 - p_0)^M \tag{46}$$

If this decode probability is substituted in Equation 30, then the relation between the number of sensors and the per sequence PAE can be quantified.

$$\mathcal{P}_{L2}\{Q_x > 0\} = \frac{1}{2}\left(1 - p_0\right)^{4M} \tag{47}$$

Figure 19b plots the relation for multiple sensors as a function of p_0. With a realistic number of sensors, $M = 16$, it is possible to achieve a per sequence PAE of less than 0.01 with a single sensor decode probability of 0.06. Inspecting Equation 47 we observe that although $1 - p_0$ may be close to one, it quickly vanishes. This is because the number of sensors is multiplied by a factor of four due to the fact that only a single packet must be decoded for correct association over the entire PES. Also, for $M >> 1$, a thresholding effect is evident with respect to p_0. For smaller M, the performance improvement is more gradual. Another observation is that increasing the number of sensors leads to diminishing returns in terms of the performance metric. This implies that it is sufficient to have around 10 to 20 sensor nodes deployed in our application.

For deployments, it may be useful to calculate the number of sensors required for a specified PAE. That is, $\mathcal{P}\{Q_x > 0\} \leq \delta$. Equation 47 can be rearranged to give the required number of sensors for the specified error bound, where $\lceil \cdot \rceil$ is the ceiling function.

$$M \geq \left\lceil \frac{\ln\left(2\delta\right)}{4\ln\left(1 - p_0\right)} \right\rceil \tag{48}$$

The interpretation is that if M is chosen according to Equation 48, then the probability an error is made in associating emitters to measurements for single PES is less than δ.

3.5 Multi Client Analysis

This section analyzes the advantages of using L2 information as a function of the number of clients. Intuitively, L2 should be superior as CSMA/CA and the PES provide side information. The L1 comparison is omitted and the focus is on how L2 scales with the number of clients.

For the single client case, additional CTS, Data, and ACK states must be introduced, otherwise the state transition probabilities depend on whether the AP or another $E_n, n = 1, \ldots, C$ sends the RTS. The total number of states is $7C+1$. Furthermore, $q \triangleq \mathcal{P}\{Y_0 = N\}$

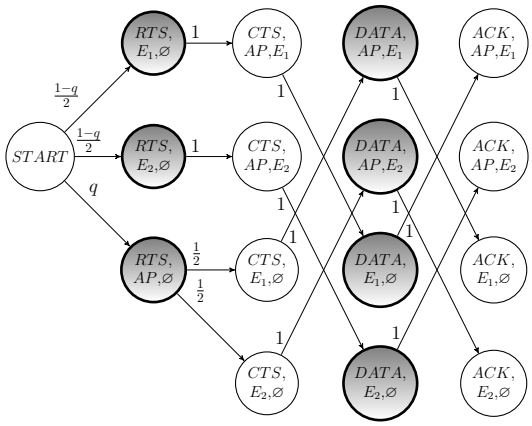


Figure 20: RTS/CTS packet exchange sequence state diagram for two clients. State transitions are not shown from the ACK state for diagram clarity. All ACK states return to $(RTS, E_n), n = 1, \ldots, N-1$ with probability $\frac{1-q}{2}$ and (RTS, E_N) with probability q. Extra thick and shaded circles signify states with corresponding MAC packets containing both emitter IDs (c.f. Table 5). $\varnothing$ is a dummy emitter indicating the third element is not required to fully define the state.

and $\mathcal{P}\{Y_0 \neq N\} = \frac{1-q}{C}$. That is, the AP initiates the sequence with probability q. If the AP does not initiate, then all other emitters have an equal probability of initiating the sequence. This state diagram is shown in Figure 20 for $C = 2$. Extension for $C > 2$ is straightforward.

Additionally, for $C > 1$ it is possible to decode packets in the sequence but still need to guess the client (c.f. $C = 1$). This is true if the P_i decodable contains only CTS or ACK MAC packets because these packets only contain the address of the RTS transmitter. However, if the client sends the RTS, then decoding any packet is sufficient to identify the client as shown in Table 5.

$$\boldsymbol{P} = \left[\begin{array}{c|c|c|c} \mathbf{0}_{NxN} & \boldsymbol{F} & \mathbf{0}_{Nx2C} & \mathbf{0}_{Nx2C} \\ \hline \mathbf{0}_{2CxN} & \mathbf{0}_{2Cx2C} & \boldsymbol{G} & \mathbf{0}_{2Cx2C} \\ \hline \mathbf{0}_{2CxN} & \mathbf{0}_{2Cx2C} & \mathbf{0}_{2Cx2C} & \boldsymbol{G} \\ \hline \boldsymbol{H} & \mathbf{0}_{Nx2C} & \mathbf{0}_{Nx2C} & \mathbf{0}_{Nx2C} \end{array}\right] \boldsymbol{F} = \left[\begin{array}{c|c} \mathbf{0}_{CxC} & \boldsymbol{I}_C \\ \hline \frac{1}{C}\mathbf{1}_C^T & \mathbf{0}_C^T \end{array}\right] \boldsymbol{G} = \left[\begin{array}{c|c} \mathbf{0}_{CxC} & \boldsymbol{I}_C \\ \hline \boldsymbol{I}_C & \mathbf{0}_{CxC} \end{array}\right]$$

$$\boldsymbol{H} = \left[\begin{array}{c|c} \frac{1-q}{C}\mathbf{1}_{2CxC} & q\mathbf{1}_{2C} \end{array}\right] \boldsymbol{p}(0) = \left[\begin{array}{c|c|c} \frac{1-q}{c}\mathbf{1}_C^T & q & \mathbf{0}_{6C}^T \end{array}\right]^T \tag{49}$$

From Figure 20, define the matrices in Equation 49. $\mathbf{0}_{NxN}$ is an NxN matrix of zeros, $\boldsymbol{I}_N$ is the NxN identity matrix, $\mathbf{1}_C$ or $\mathbf{1}_{NxC}$ is a Cx1 vector or NxC matrix of 1's, respectively.

$\mathbf{0}_N$ is the Nx1 column vector of zeros.

The initial state probabilities are given as $\boldsymbol{p}(0)$. Next, calculate $\mathcal{P}_{L2}\{Q_x > 0\}$, which depends critically on the packet type, number of decodable packets, and data direction.

$$\mathcal{P}_{L2}\{Q_x > 0\} = \sum_{k=0}^{4} \mathcal{P}_{L2}\{Q_x > 0 | K = k\}\mathcal{P}\{K = k\} \tag{50}$$

From Table 5, if $K \geq 3$ then $\mathcal{P}\{Q_x > 0 | K \geq 3\} = 0$ an RTS or DATA packet is received.

If $K = 2$, then there are a total of $\binom{4}{2} = 6$ different MAC packet types. Of these, only CTS or ACK packets makes the client unknown, provided the AP sends the RTS. Recall $D_i \sim$ bernoulli(p) represents the decodability of the i^{th} measurement. Let CA $\triangleq D_0 = 0 \cap D_1 = 1 \cap D_2 = 0 \cap D_3 = 1$ be an event indicating the two decodable packets were CTS/ACK and $\mathbb{1}_{CA}$ an indicator R.V. for the event. Note that $\mathcal{P}\{CA | K = 2\} = \frac{1}{6}$.

$$\mathcal{P}\{Q_x > 0 | K = 2\} = \frac{5}{6}\mathcal{P}\{Q_x > 0 | K = 2, \mathbb{1}_{CA} = 0\} + \frac{1}{6}\mathcal{P}\{Q_x > 0 | K = 2, \mathbb{1}_{CA} = 1\} \tag{51}$$

However, $\mathcal{P}\{Q_x > 0 | K = 2, \mathbb{1}_{CA} = 0\} = 0$ because the packet type will either be RTS or DATA. Further condition on the probability of the AP sending data to a client, $\mathcal{P}\{Y_0 = N\} = q$.

$$\begin{aligned}\mathcal{P}\{Q_x > 0 | K = 2, \mathbb{1}_{CA} = 1\} &= q\mathcal{P}\{Q_x > 0 | K = 2, \mathbb{1}_{CA} = 1, Y_0 = N\} \\ &+ (1 - q)\mathcal{P}\{Q_x > 0 | K = 2, \mathbb{1}_{CA} = 1, Y_0 \neq N\}\end{aligned} \tag{52}$$

If $Y_0 \neq N$, then $\mathcal{P}\{Q_x > 0 | K = 2, \mathbb{1}_{CA} = 1, Y_0 \neq N\} = 0$ because the CTS and ACK contain the client MAC address. Therefore, the other emitter is the AP due to the assumption of infrastructure BSS mode. If $Y_0 = N$, then the strategy is to uniformly guess a client and it is correct with probability $\frac{1}{C}$.

$$\mathcal{P}\{Q_x > 0 | K = 2\} = \frac{q(C - 1)}{6C} \tag{53}$$

For $K = 1$, the problem again is decoding either a CTS or ACK as correct association is always possible otherwise. Let random variable B represents the decodable MAC packet type.

$$\mathcal{P}\{Q_x > 0 | K = 1\} = \frac{1}{4}\mathcal{P}\{Q_x > 0 | K = 1, B = CTS\} + \frac{1}{4}\mathcal{P}\{Q_x > 0 | K = 1, B = ACK\} \tag{54}$$

The last two probabilities are identical as can be seen from the Markov state diagram so it is sufficient to consider $B = CTS$. Again, condition on $Y_0 = N$.

$$\mathcal{P}\{Q_x > 0|K = 1, B = CTS\} = q\mathcal{P}\{Q_x > 0|K = 1, B = CTS, Y_0 = N\} + (1-q)\mathcal{P}\{Q_x > 0|K = 1, B = CTS, Y_0 \neq N\} \quad (55)$$

As in the $K = 2$ case, if $Y_0 = N$ uniformly associate a client. The association is correct with probability $\frac{1}{C}$; otherwise correct association is always possible.

$$\mathcal{P}\{Q_x > 0|K = 1\} = \frac{q(C-1)}{2C} \quad (56)$$

If no packets are decodable, then choose the RTS emitter and the particular client. The strategy is to choose the clients with equal probability and the order which maximizes the probability.

Suppose $q > 0.5$, then consider the complementary probability

$$\begin{aligned}\mathcal{P}\{Q_x = 0|K = 0\} &= \mathcal{P}\{\cap_{i=0}^{3}\hat{Y}_i = Y_i\} = \mathcal{P}\{\cap_{i=1}^{3}\hat{Y}_i = Y_i|\hat{Y}_0 = Y_0\}\mathcal{P}\{\hat{Y}_0 = Y_0\} \\ &= q\mathcal{P}\{\cap_{i=1}^{3}\hat{Y}_i = Y_i|Y_0 = N\} \end{aligned} \quad (57)$$

The last line follows because the association for Y_0 is correct. This implies $Y_0 = N$ because of the assumption of $q > 0.5$, implying $\hat{Y}_0 = N$. Then, guess the client and are correct with probability $\frac{1}{C}$. Thus for $q > 0.5$, $\mathcal{P}\{Q_x > 0|K = 0\} = 1 - \frac{q}{C}$.

Suppose $q < 0.5$, then consider the complementary probability

$$\mathcal{P}\{Q_x = 0|K = 0\} = \mathcal{P}\{\cap_{i=1}^{3}\hat{Y}_i = Y_i|\hat{Y}_0 = Y_0\}\mathcal{P}\{\hat{Y}_0 = Y_0\} = \frac{1-q}{C} \quad (58)$$

The strategy is to guess $\hat{Y}_0 = 1, 2, \ldots, C$ with equal probability. If correct, then the client is known and the other emitter must be the AP. The implication is that there is no ambiguity in the association for the conditional probability term.

For $q < 0.5$, $\mathcal{P}\{Q_x > 0|K = 0\} = 1 - \frac{1-q}{C}$. The condition on q can be incorporated as

$$\mathcal{P}\{Q_x > 0|K = 0\} = 1 - \frac{\max\{q, 1-q\}}{C} \quad (59)$$

Combining equations 50, 53, 56 together with Equation 59 yields the final result. Equation 60 gives the L2 per sequence PAE as a function of the decode probability, number of clients,

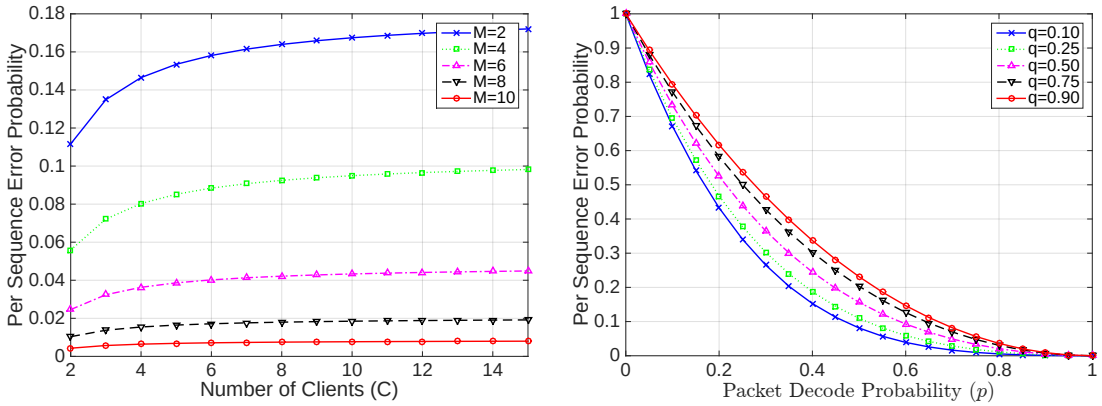


Figure 21: The per packet exchange sequence probability of association error using Layer 2 information. 21a plots this for various numbers of sensors, M. Each sensor has a local packet decode probability of $p_0 = 0.2$ and $q = 0.75$, where q is the probability the RTS packet is sent by the AP. 21b plots this probability as the number of clients, C, tends to ∞ for various q.

and probability the AP transmits the RTS.

$$\mathcal{P}_{L2}\{Q_x > 0\} = \frac{p^2\,(1-p)^2\,q(C-1)}{C} + \frac{2p\,(1-p)^3\,q(C-1)}{C} + \frac{(1-p)^4\,(C-\max\{q,1-q\})}{C} \tag{60}$$

The asymptotic behavior of this equation is interesting. Equation 61 gives the limit as the number of clients tends to infinity. Critically, it is not zero and only depends on the packet decode probability and data flow direction.

$$\lim_{C\to\infty} \mathcal{P}_{L2}\{Q_x > 0\} = q\left(p^2\,(1-p)^2 + 2p\,(1-p)^3\right) + (1-p)^4 \tag{61}$$

Figure 21a plots Equation 60. Figure 21b plots Equation 61 as a function of p for various values of q. It can be seen that client to AP transactions have lower PAE than AP to client.

3.6 Model Validity

The state diagrams of Figs. 17 and 20 are quite restrictive since no failed transmissions are allowed. However, the model can be applied appropriately if packet collisions can be detected. The following analysis demonstrates measuring the interframe spacing can detect a packet collision correctly with high probability.

In the multi client case, note that collisions only happen on an RTS. From Equation 10,

$$\hat{\boldsymbol{t}}_i = \widehat{\boldsymbol{\Delta t}}_i + \mathbf{1}_{M-1}\hat{t}_1[i] \tag{62}$$

Let $\alpha_i \triangleq \hat{t}_1[i]$. Then, the m^{th} element of $\hat{\boldsymbol{t}}_i$ can be approximated.

$$\hat{t}_m[i] = \hat{t}_{m1}[i] + \alpha_i \approx \alpha_i, m = 2, \ldots, M \tag{63}$$

For a constrained geometry such as a stadium (100mx200m), the maximum possible TDoA is 750nS, which is much shorter than the shortest possible interframe spacing of 9μS.

Suppose the measurements are $\Delta\alpha_i \triangleq \alpha_i - \alpha_{i-1}$. Let $F \sim \text{bernoulli}(\pi_0)$ be an R.V. indicating an RTS collision, with $\pi_1 = 1 - \pi_0$. Assume conditional Gaussian distributions dependent on π_0. For any of the N emitters,

$$\mathcal{H}_0 : p_F(\Delta\alpha_i | F = 1) \sim \mathcal{N}\left(\mu_0(N_s), \sigma_0^2\right), \mathcal{H}_1 : p_F(\Delta\alpha_i | F = 0) \sim \mathcal{N}\left(\mu_1, \sigma_1^2\right)$$
$$\mu_0(N_s) = L_{RTS} + SIFS + L_{CTS} + DIFS + N_s T_{slot}, \mu_1 = L_{RTS} + SIFS \tag{64}$$

L_{RTS} and L_{CTS} are the lengths of the RTS and CTS packets, respectively. SIFS and DIFS are the short and DCF interframe spacing times, respectively. T_{slot} is the length of a slot in the IEEE 802.11g standard [41], which is either 9μS or 20μS. $N_s \sim \text{unid}(0, CW_{min})$ is the slot number, which is a discrete uniform R.V. from 0 to CW, where CW is the contention window size. To simplify the problem, suppose

$$N_s^* = \min_{N_s} \|\Delta\alpha_i - \mu_0(N_s)\|_2^2 \tag{65}$$

Then, $p_F(\Delta\alpha_i | F = 1)$ becomes Gaussian with mean $\mu_0 \triangleq \mu_0(N_s^*)$. The approximation is that errors are only made to adjacent slots. Without this assumption, the distribution under $\mathcal{H}_0$ is a Gaussian mixture, which does not have a closed form solution for a simple hypothesis test.

$$\Delta\alpha_i^2\left(\sigma_1^2 - \sigma_0^2\right) - 2\Delta\alpha_i\left(\mu_0\sigma_1^2 - \mu_1\sigma_0^2\right) \underset{\mathcal{H}_0}{\overset{\mathcal{H}_1}{\gtrless}} 2\sigma_0^2\sigma_1^2 \ln\left(\frac{\sigma_1\pi_0}{\sigma_0\pi_1}\right) - \sigma_1^2\mu_0^2 + \sigma_0^2\mu_1^2 \tag{66}$$

Assuming $\sigma_0^2 = \sigma_1^2$ and noting $\mu_0 > \mu_1$, this simplifies to

$$\Delta\alpha_i \underset{\mathcal{H}_0}{\overset{\mathcal{H}_1}{\gtrless}} \gamma, \gamma \triangleq \frac{\sigma_0^2 \ln(\pi_0/\pi_1)}{\mu_1 - \mu_0} + \frac{\mu_1 + \mu_0}{2} \tag{67}$$

The probability of error is calculated as follows, where $\Phi(\cdot)$ is the Gaussian CDF.

$$\begin{aligned} P_E &= \pi_1 \int_\gamma^\infty p_F(\Delta\alpha_i | F = 0)\, d\Delta\alpha_i + \pi_0 \int_{-\infty}^\gamma p_F(\Delta\alpha_i | F = 1)\, d\Delta\alpha_i \\ &= \pi_1 \Phi\left(\frac{\mu_1 - \gamma}{\sigma_0}\right) + \pi_0 \Phi\left(\frac{\gamma - \mu_0}{\sigma_0}\right) \end{aligned} \tag{68}$$

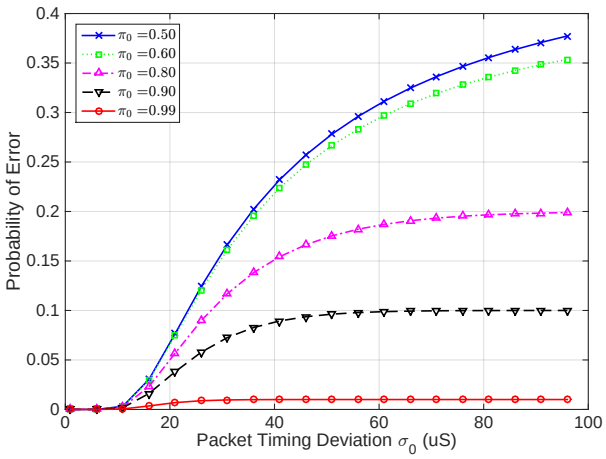


Figure 22: Approximate probability of error for detecting an RTS collision using interframe spacing. The curves represent various π_0, the probability of an RTS collision. The signal is assumed to be 20 MHz channel-spaced OFDM with QPSK $R = \frac{1}{2}$ modulation and short timeslot $T_{slot} = 9\mu$S.

Figure 22 plots P_E as a function of σ_0 for a 20 MHz channel-spaced OFDM IEEE 802.11g signal with QPSK $R = \frac{1}{2}$ and short interframe spacing. As the probability of an RTS collision increases, P_E decreases. Note that the standard specifies error bounds on SIFS as SIFS $\pm 10\% =$ (pg. 827,[41]). To a first order approximation, this implies $\sigma_0 \approx 2\mu$S. From the figure, P_E is virtually zero under these worst-case assumptions. Using a hypothesis test on the interframe spacing can accurately detect if an RTS collision has occurred. The result is that the data association strategies discussed can be applied with greater accuracy.

3.7 Conclusions

This chapter explored using Layer 2 knowledge such as MAC protocol and packet level correlations as features in the data association problem for extreme emitter density RF environments. A Markov model was introduced as an analysis technique to couple physical layer measurements with side information available at higher levels of the protocol stack.

Analysis for an RTS/CTS packet exchange sequence under ideal channel conditions demonstrated that a reasonably small number of sensors with low local packet decode probability can correctly associate emitters to measurements with high probability. For more than two clients, the direction of data transfer affects the probability of association error. That is, clients uploading to an access point lowers the probability of association

error compared to downloading. Most importantly, exploiting Layer 2 knowledge can yield a lower per packet exchange sequence PAE compared to the Layer 1 strategy, although the per packet PAE may be higher.

Future work should compare other PHY information known at the sensors such as frequency offset of the detections to the Layer 1 strategy as well as Layer 2. Additionally, other packet exchange sequences and standards could be analyzed. The channel model should be relaxed to incorporate the possibility that a packet transmission fails and the emitter enters the exponential backoff procedure. However, the current analysis is sufficient to show exploiting Layer 2 structure can yield significant performance advantages in the data association problem.

CHAPTER IV

MAC ASSISTED LOCALIZATION

Localization in EED environments is of critical importance for spectrum management and security. One example of such an environment is Bobby Dodd football stadium on game-day. Fans' smartphones, Bluetooth headsets, wearables, etc. may interfere with critical communication systems such as mobile ticket scanners or coach-to-coach headsets. Fast and precise localization of a rogue client or other non-conforming Emitter of Interest (EoI) is needed to secure the spectrum for these critical systems. Localization can be performed using sensors to capture RF spectrum. An example scenario is illustrated in Figure 23 for two emitters and M sensors mounted on the perimeter of the stadium.

In such a situation, the emitters are typically constrained to be within a fixed ellipse, and their transmissions follow a known communications protocol. Both of these assumptions can be exploited to decrease the localization estimate's uncertainty. In EED environments, this is especially important due to the high density of devices. Returning to the football stadium example, this implies inconveniencing fewer fans when finding the interfering emitter.

In this chapter, a three-stage algorithm is presented for lowering the uncertainty of emitter position. The stages are coupled by the confidence regions (CR) generated from their position estimates. The first stage uses only sensors which can decode the packet sent from the emitter to solve for an initial position estimate. Since these sensors can decode the packet, they have localized the signal in time. A confidence region for the position estimate is then computed. In the second stage, sensors which can't decode the packet then use the Time-of-Arrival (ToA) estimates from Stage I and the geometry to bound their ToAs. This is done by restricting the range of the cross-correlation lags based on geometry and the initial Stage I estimate. Finally, Stage III exploits the protocol-specified time between packets to estimate the distance from a node with known position, such as an Access Point (AP), to the EoI. For each stage, a confidence region is computed based on a

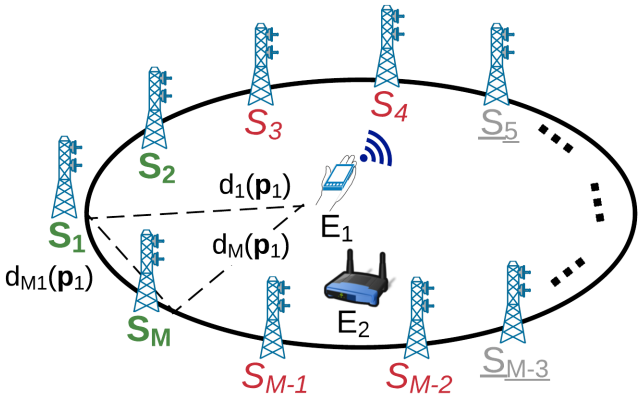


Figure 23: System Diagram. Emitter E_1 transmits a signal to E_2. Sensors $S_m, m = 1, 2, \ldots, M$ have known position vectors $\boldsymbol{q}_m$ and attempt to localize Emitter E_1 with unknown position vector $\boldsymbol{p}_1$ using TDoA. E_2 has a known position. Distances from E_1 to S_m are denoted as $d_m(\boldsymbol{p}_1)$. Sensors able to decode the packet ($S_m \in \Gamma_{dec}$) are labeled in bold green, non-decoding ($S_m \in \Gamma_{ndec}$) in italicized red, and non-participating sensors ($S_m \in \Gamma_{np}$) in underlined gray text.

specified confidence level. The region from the previous stage is used by the next stage to bound the emitter location. In choosing the intersection of the error sets, we have chosen a specific sensor fusion approach. There are other possible approaches, such as Covariance Intersection (CI), and our choice may not be optimal, but should be fast. [1] The analysis of other fusion methods is reserved for future work.

The primary contributions of this chapter are:

1. A three-stage localization algorithm for emitters operating under a known communications standard within a constrained geometry. The technique in shown in simulation to significantly reduce the error area in the localization estimate by exploiting knowledge of the MAC layer, as well as windowing the timing estimate based on geometry for low Signal-to-Noise Ratio (SNR) sensors.

2. A novel technique, Packet Time-Difference-of-Arrival (PTDoA), to estimate the distance between an emitter of interest and an emitter with a known position using the Time-Difference-of-Arrival (TDoA) between packets. The theoretical variance of the

[1] In fact, if the cross-covariance is known between the ellipses, then the optimal ellipse is a subset of the intersection [21, 45].

estimate is derived as a function of packet timing jitter, SNR, and the length of the Packet Exchange Sequence (PES).

3. Analytical results on the probability of choosing the correct integer lag as a function of SNR and window size for an ideal impulse signal auto-correlation function. The results are compared with the typical uniform distribution assumption. The derived distribution variance is less than the uniform distribution variance for moderate SNRs.

Background material is discussed in Section 4.1. Section 4.2 describes the system model and assumptions. The three-stage algorithm is explained by stage in Sections 4.3, 4.4, and 4.5. Simulations are discussed in Section 4.6. The paper concludes with Section 4.7.

4.1 Background

4.1.1 Time Delay Estimation

Localization using TDoA requires a TDE technique for discrete time data. Typically, there are two steps. First, the coarse estimate locates the delay to within an integer sample. Next, a fine estimate performs interpolation or optimization to locate the sub-sample delay. The coarse estimate can include maximization of the cross-correlation function, minimizing the average square difference function (ASDF), or minimizing the average magnitude difference function (AMDF). Jacovitti compares these techniques under the assumption that the signal is random [43]. ASDF and ADMF are found to have lower variance at high SNRs than the cross-correlation approach.

Multiple algorithms also exist for the fine TDE. One popular approach is to perform a parabolic interpolation around the maximum magnitude cross-correlation sample. However, this is known to be a biased estimator, where the bias is a function of the sub-sample displacement [56, 15]. Other approaches include spline-based interpolation [83] and sync interpolation by zero-padding the Discrete Fourier Transform [52]. Finally, the TDE problem can also be formulated as one of estimating coefficients of a FIR filter [19].

For complex cross-correlation functions, some modifications are necessary. The Maximum Likelihood Estimate is a maximization of the real part of the cross-correlation function [66]. At the true time delay, the cross-correlation magnitude is maximized, and the

cross-correlation phase has a zero crossing. Therefore, one can linearly interpolate the phase to estimate the y-intercept and achieve sub-sample accuracy [89]. In simulation, this technique is significantly less biased than parabolic interpolation on the magnitude of the cross-correlation. The parabolic interpolation technique can also be adapted for real data to significantly reduce the bias [42]. Finally, Agrez provides another three-sample interpolation technique [2].

Many theoretical TDE variance bounds exist, but some only apply in certain SNR intervals. At sufficiently high SNR, the Cramer-Rao Lower Bound (CRLB) provides the asymptotic performance for an unbiased estimator. At sufficiently low SNR, the estimator variance can be approximated by the variance of a uniform random variable [66]. This is because large errors can occur in picking the cross-correlation peak due to the large noise variance [40]. Composite bounds have been derived which cover a range of SNRs including Ziv-Zakai [92] and Weiss [86].

4.1.2 Round-Trip Time-of-Flight

The third stage of the proposed algorithm uses MAC protocol knowledge; specifically, the interframe spacing, to improve emitter localization. This idea is ostensibly similar to some existing round-trip-time localization schemes; however, there are significant differences which will now be discussed.

In RTOF, a client estimates its distance to an AP by measuring the number of clock cycles between sending a data frame, for example, and receiving an acknowledgement frame. The Time-of-Flight (ToF) is then calculated as a function of the RTT. Schauer et al. [70] provide a summary of the problem, and proposes a new technique based on the NULL-ACK PES. Gunther and Hoene [36] conducted two experiments using three different IEEE 802.11 b/g chipsets and measured the Round Trip Time (RTT) for an ICMP ping. The packet timestamps were provided by the WLAN card drivers to microsecond resolution, limiting the accuracy of the distance estimation. Ciurana et al. performed a similar experiment, again for an ICMP ping, but using external hardware to capture timing signals directly from the WLAN IC [22].

Bahillo et al. [6] also used a test harness to capture timing signals directly from the WLAN IC, but using the RTS/CTS PES. The authors note these signals are synchronous with the clock. This implies the time delay estimation resolution is limited by the clock frequency of the WLAN IC. Measurements were taken in three different environments, and the RTT was estimated as the mean of a Gaussian distribution.

The CAESAR system [32] uses the DATA/ACK PES for the RTT calculation. ToAs are calculated from the hardware driver. Histograms of the MAC idle time are shown demonstrating a bimodal distribution. This is hypothesized to be a result of the frame detection algorithm and automatic gain control (AGC) adjustments. Additionally, WLAN IC manufacturers have different interframe spacing bias compared to the IEEE 802.11g standard, with variances on the order of nanoseconds [16]. These factors can be accounted for as the MAC layer provides information on the WLAN chipset from the MAC address.

In all of these works, the client measures the RTT. This requires cooperative localization, wheres the proposed system works with non-collaborative clients. Additionally, the client is likely to have a consumer-grade oscillator with a frequency accuracy on the order of ± 25 parts per million (ppm). This may significantly degrade the accuracy of the ToF when converted from clock cycles to seconds. In contrast, a sensor node, such as a software defined radio (SDR), with a GPS-Disciplined Oscillator can easily have nanosecond-level timestamps of complex baseband samples with frequency accuracies on the order of ± 0.5 parts per billion (ppb). Because the clock is used for the timestamp, no interpolation is possible in RTOF. However, in our algorithm the sensors receive complex baseband samples. This enables the use of a range of TDE techniques, including cross-correlation and sub-sample interpolation.

4.2 System Model

Consider M spatially distributed, time synchronized sensors $\mathcal{S} = \{S_m\}_{m=1}^{M}$ with *known* 2-D[2] position vectors $\boldsymbol{q}_m = \begin{bmatrix} x_m & y_m \end{bmatrix}^T$ which can communicate reliably with one another. It is assumed the sensor clocks are perfectly synchronized. A stationary emitter E_1 with

[2]The 3-D case is a straightforward extension.

unknown position vector $\boldsymbol{p}_1 = \begin{bmatrix} x & y \end{bmatrix}^T$ transmits a signal $s(t)$ using a known standard to another emitter, E_2, with known position $\boldsymbol{p}_2$. This communication is observed by the RF sensor network. The emitter position, $\boldsymbol{p}_1$, is to be estimated using received signals at each sensor and is assumed to lie within the convex hull of $\{\boldsymbol{q}_m\}_{m=1}^M$. The setup is depicted in Figure 23. One practical example of such a geometry is a football stadium with sensors mounted on its perimeter.

In the testbed which has been developed in Bobby Dodd football stadium as part of this research, the sensors have GPS-Disciplined Oscillators (GPSDO) for clock synchronization [30]. These GPSDOs provide a 10 MHz clock, as well as a 1 Pulse Per Second (1 PPS) signal to the sensors. However, each sensor's 1 PPS signal is typically only within ± 50 ns of UTC time. For our testbed to more closely resemble the proposed system model, additional hardware can be procured with tighter time tolerances for additional cost.

Although E_1 transmits a signal using a known standard, it is non-collaborative with the sensors. Non-collaborative as defined here implies that the emitter does not share information explicitly with the sensors, but it is not actively attempting to disrupt sensor measurements. The sensors only passively observe signal samples. Furthering the stadium example, the emitter may be a particular cell phone transmitting under a WLAN, Bluetooth, or a cellular standard. In such a scenario, the sensors observe signals conforming to a known standard, but can not communicate with the emitter. The non-collaborative assumption coupled with the desire for a simple RF front-end suggests using TDoA for position estimation.

Some additional constraints are placed on $s(t)$, the signal transmitted from E_1. $s(t)$ is restricted to be a digital modulation; hence, knowing the data symbols is sufficient to reconstruct the transmitted signal. It is assumed that the communications protocol has a medium access control (MAC) layer, and that the data is packetized. Additionally, there are *Packet Exchange Sequences* (PES) that are defined by the protocol. These assumptions are sufficient to increase the accuracy of TDoA estimation, while being sufficiently general to apply more broadly. WIFI is one example of a conforming protocol, but there are many others.

Each packet consists of a preamble, a MAC header, and a payload (data) section. The preamble is known *a-priori* to all sensors. The MAC header is unknown, but some information may be inferred for undecodable packets due to packet-level correlations, as shown in Chapter 3. The data symbols are unknown and assumed random with equal probability for each symbol.

Sensor S_m receives a complex baseband signal[3] $s(t)$ attenuated by $\alpha_m \in \mathcal{R}$ and delayed by t_m.

$$s_m(t) \triangleq \alpha_m s(t - t_m) \tag{69}$$

The receiver noise is assumed to be stationary complex additive white Gaussian noise. $s(t)$ is of duration T_{sig} seconds and effectively band-limited to β Hz. The observed signal is $y_m(t)$ of duration T_{meas} seconds. It is assumed that $T_{meas} > 2T_{sig}$. We attempt to estimate $\{t_m\}_{m=1}^{M}$ as $\{\hat{t}_m\}_{m=1}^{M}$.

$$y_m(t) = \begin{cases} s_m(t) + w_m(t) & 0 \le t \le T_{sig}, m = 1, 2 \ldots, M \\ w_m(t) & \text{otherwise} \end{cases} \tag{70}$$

Following ([66], Ch. 7.2.1), and sampling at the Nyquist rate $T_s = \frac{1}{\beta}$ yields the discrete-time formulation where $s_m[n] \triangleq s_m(nT_s)$, $N_{sig} = \lfloor T_{sig}/T_s \rfloor$, $N_{meas} = \lfloor T_{meas}/T_s \rfloor$, and $N_m \approx t_m/T_s$ is the delay at S_m in samples.

$$y_m[n] = \begin{cases} w_m[n] & 0 \le n \le N_m - 1 \\ s_m[n] + w_m[n] & N_m \le n \le N_m + N_{sig} - 1 \\ w_m[n] & N_m + N_{sig} \le n \le N_{meas} - 1 \end{cases} \tag{71}$$

Assume the noise is i.i.d. in time and space. The $w_m[n]$ are independent zero mean complex random variables with variance $\sigma_w^2 = \beta\sigma_r^2$ where $w_m(t)$ has power spectral density σ_r^2 W/Hz. That is, $w_m[n] \sim \mathcal{CN}\left(0, \sigma_w^2\right) \forall m, n$.

The sensors can be divided into three sets such that $\mathcal{S} = \Gamma_{dec} \cup \Gamma_{ndec} \cup \Gamma_{np}$.

- Γ_{dec}: Sensors which *can decode* the packet and participate in the localization.

[3]We are ignoring the phase offset between the sensor LO and the emitter center frequency during down-conversion, which causes a phase delay [52].

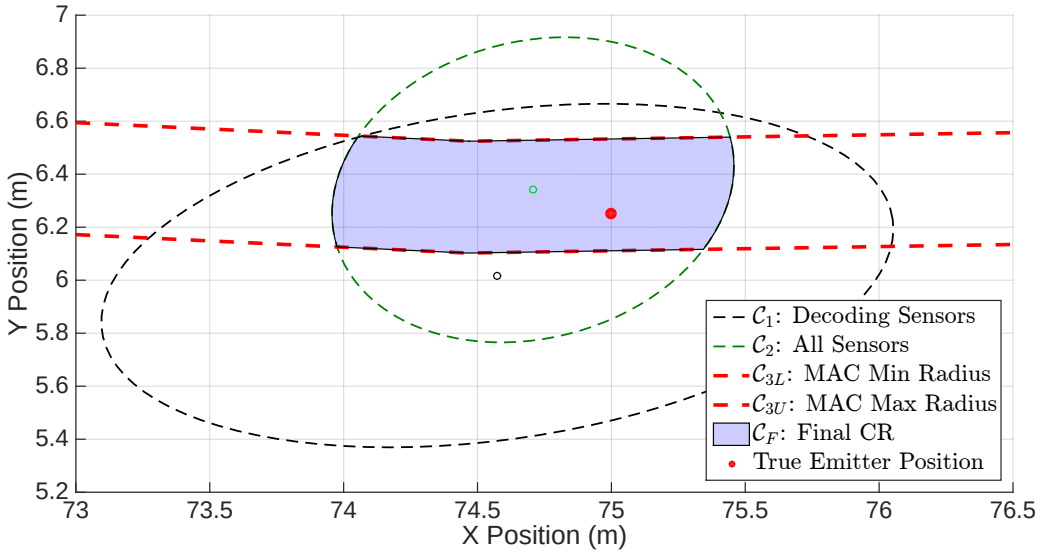


Figure 24: Three-stage positioning algorithm example result. The dotted black and green lines are Stage I and II confidence regions, $\mathcal{C}_1$ and $\mathcal{C}_2$, respectively. The dotted red lines represent $\mathcal{C}_{3L}$ and $\mathcal{C}_{3U}$, the confidence region generated from estimating the distance from Emitter E_1 to Emitter E_2, whose position is known. The open dots are position estimates with the colors representing their respective strategies. The red-filled dot is the true emitter position. The final intersection of all the confidence regions, $\mathcal{C}_F$, is shaded blue, which is a sliver containing the true position. No timing jitter on the interframe spacing is shown.

- Γ_{ndec}: Sensors which *cannot decode* the packet but participate in the localization.
- Γ_{np}: Sensors *not participating* in the localization.

Decoding sensors are shown in bold green, non-decoding in italicized red, and non-participating sensors in underlined gray text in Figure 23.

Assume at least three sensors, arbitrarily $S_1, S_2, \ldots, S_{M_d}$, can decode the packet, where $M_d = |\Gamma_{dec}|, M_d \geq 3$ is the size of the decodable sensor set. S_1 will denote the reference sensor. This will typically be the sensor with the highest SNR. Then, $S_m \in \Gamma_{dec}$ perform TDE.

The following section describes the three-stage iterative localization algorithm. For each stage, a different algorithm is used and a corresponding CR is computed for a given confidence level. Some example error sets are depicted in Figure 24. This multi-stage approach makes the localization fast. Stage I involves position estimation using only sensors able to decode the packet. This CR is shown as dotted black lines. In Stage II, all sensors

participate in the localization. However, sensors which can not decode the packet have a restricted range in which to choose ToAs based on the initial Stage I CR and the constrained geometry. In the figure, this corresponds to the green dotted lines. Finally, Stage III uses knowledge from the MAC layer, specifically the packet exchange sequence and inter-packet timing, to determine a CR. The corresponding CR is between the dotted red lines. The true position of E_1 is the red-filled circle, and the open circles correspond to position estimates of the stage with the same color.

4.3 Stage I - Decoding Sensors

For $S_m \in \Gamma_{dec}$, the data symbols can be recovered. Furthermore, because these sensors can decode the packet, they have localized the packet in time to within some small window. A ToA estimate is made at each sensor using a TDE technique and the variance is computed. Then, an initial position estimate is made.

At S_m, the cross-correlation is computed using the decoded symbols to reconstruct $s[n]$.

$$R_m[l] \triangleq \sum_{n=0}^{N_{meas}-1} y_m[n]s^*[n-l], 0 \leq l \leq N_{meas} + N_{sig} - 2 \tag{72}$$

For this algorithm, the coarse TDE is determined by selecting the integer sample that maximizes the real part of the cross-correlation. Then, sub-sample interpolation is performed. As mentioned in Section 4.1.1, there are many options for sub-sample interpolation.

The estimator variance is computed using the CRLB. We argue this is appropriate because the sensors have sufficient SNR to decode the packet. The assumption is that this implies operation in the "high SNR" region such that the CRLB is an accurate lower bound on the variance of the estimator. The use of certain sub-sample interpolation schemes, such as parabolic interpolation, does add some bias, but it is small in the range of SNRs considered. The CRLB for any unbiased estimator of t_m, denoted $\hat{t}_m$, is from [66],

$$\sigma^2_{\hat{t}_m} \geq \frac{1}{8\pi^2 \chi_m \beta^2_{RMS}} \sec^2 \tag{73}$$

where $\chi_m \triangleq \mathcal{E}_m/\sigma^2_w$ is the *energy* SNR at S_m and β_{RMS} is the RMS bandwidth of the signal. The equation assumes the Nyquist sampling rate. $\mathcal{E}_m$ can be written as a function

of $\mathcal{E}_{sig}$, the energy in the transmitted signal $s[n]$.

$$\mathcal{E}_m \triangleq \sum_{n=0}^{N_{sig}-1} |s_m[n]|^2 = \alpha_m^2 \mathcal{E}_{sig} \tag{74}$$

The $S_m \in \Gamma_{dec}$ communicate their respective ToA estimate to a central processor.

$$\boldsymbol{\theta}_{ToA} \triangleq \begin{bmatrix} t_1 & t_2 & \dots & t_{M_d} \end{bmatrix}^T, \hat{\boldsymbol{\theta}}_{ToA} \triangleq \begin{bmatrix} \hat{t}_1 & \hat{t}_2 & \dots & \hat{t}_{M_d} \end{bmatrix}^T \tag{75}$$

The TDoA estimates are formed as,

$$\hat{\boldsymbol{\theta}}_{TDoA} \triangleq \boldsymbol{H}_t \hat{\boldsymbol{\theta}}_{ToA} = \begin{bmatrix} \Delta\hat{t}_{21} & \Delta\hat{t}_{31} & \dots & \Delta\hat{t}_{M_d 1} \end{bmatrix}^T, \boldsymbol{H}_t = \left[\begin{array}{c|c} -\mathbf{1}_{M-1} & \boldsymbol{I}_{M-1} \end{array} \right] \tag{76}$$

where $\Delta\hat{t}_{m1} \triangleq \hat{t}_m - \hat{t}_1, m = 2, \dots, M_d$ and $\boldsymbol{H}_t$ is a matrix of dimension $(M_d - 1)\text{x}M_d$.

Assuming the ToA measurements are uncorrelated, the asymptotic error distribution can be computed for $\hat{\boldsymbol{\theta}}_{TDoA}$. Suppose $\boldsymbol{J}_{ToA} \triangleq \boldsymbol{I}_{ToA}^{-1}$, where $\boldsymbol{I}_{ToA}$ is the Fischer Information Matrix.

$$\boldsymbol{J}_{ToA}(i) = \text{Diag}\left(\sigma_{\hat{t}_1}^2, \sigma_{\hat{t}_2}^2, \dots, \sigma_{\hat{t}_{M_d}}^2\right) \tag{77}$$

Using the Vector Parameter CRLB Transform ([46],pg. 45), the CRLB for the TDoA set can be derived as

$$\boldsymbol{J}_{TDoA} = \frac{\partial \boldsymbol{g}(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}} \boldsymbol{J}_{ToA} \frac{\partial \boldsymbol{g}(\boldsymbol{\theta})^T}{\partial \boldsymbol{\theta}} \tag{78}$$

with $\boldsymbol{g}(\boldsymbol{\theta}) = \boldsymbol{H}_t \hat{\boldsymbol{\theta}}_{ToA}$. Estimator efficiency is maintained for affine transformations ([46], pg. 37) such as Equation 76. The vector partial derivative is defined as

$$\frac{\partial \boldsymbol{g}(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}} \triangleq \begin{bmatrix} \frac{\partial g_1(\boldsymbol{\theta})}{\partial \theta_1} & \frac{\partial g_1(\boldsymbol{\theta})}{\partial \theta_2} & \dots & \frac{\partial g_1(\boldsymbol{\theta})}{\partial \theta_M} \\ \frac{\partial g_2(\boldsymbol{\theta})}{\partial \theta_1} & \frac{\partial g_2(\boldsymbol{\theta})}{\partial \theta_2} & \dots & \frac{\partial g_2(\boldsymbol{\theta})}{\partial \theta_M} \\ \vdots & \dots & \ddots & \vdots \\ \frac{\partial g_N(\boldsymbol{\theta})}{\partial \theta_1} & \frac{\partial g_N(\boldsymbol{\theta})}{\partial \theta_2} & \dots & \frac{\partial g_N(\boldsymbol{\theta})}{\partial \theta_M} \end{bmatrix} \tag{79}$$

where θ_m and $g_n(\boldsymbol{\theta})$ denotes the m^{th} parameter and n^{th} function, respectively. Using Equation 78 and noting $\frac{\partial \boldsymbol{g}(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}} = \boldsymbol{H}_t$, the CRLB for the TDoA estimate is:

$$\boldsymbol{J}_{TDoA} = \boldsymbol{H}_t \boldsymbol{J}_{ToA} \boldsymbol{H}_t^T \tag{80}$$

With the TDoA estimate and asymptotic covariance, the emitter position $\boldsymbol{p}_1$ can be estimated. While there are many options to compute position from TDoA measurements,

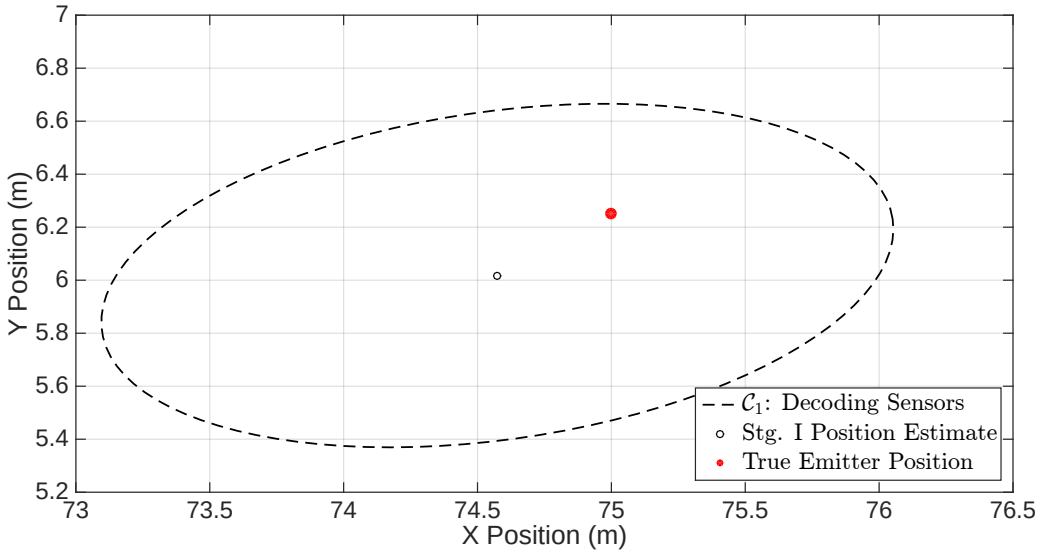


Figure 25: Example Stage I simulation. The black circle represents the initial position estimate $\hat{p}$. The true emitter position is the red dot. The CRLB is computed and used to determine a confidence region with confidence coefficient $\delta_{pos1} = 0.95$ centered at $\hat{p}$.

the approach by Chan and Ho is used [20]. Appendix A describes the calculations and provides a clarification to their original paper. Since the emitter position estimate is made only by sensors in the decodable set, $M = M_d$, the asymptotic covariance of Equation 80 is used to construct the error ellipse and the subsequent set $\mathcal{C}_1$. An example realization is shown in Figure 25.

4.4 Stage II: All Participating Sensors

Stage II involves all participating sensors solving for emitter position. For decoding sensors, the TDE is identical to Stage I. These sensors have high SNR, but the position estimate is likely to have poor resolution in one direction due to sensor and emitter geometry. The non-decoding sensors have a favorable geometry, but low SNR, resulting in a non-negligible probability of choosing the wrong integer lag of the cross-correlation peak. To minimize these large errors, the ToA is restricted to a range based on geometry and the reference sensor's TDE variance.

4.4.1 Non-Decodable Sensor ($S_m \in \Gamma_{ndec}$) Time Delay Estimation

The initial position estimate and associated confidence region allows the non-decodable sensors to window their cross-correlation estimates. That is, they are restricted to choosing lags which are a function of the maximum and minimum TDoA within the CR of Stage I. It is assumed that the $S_m \in \Gamma_{ndec}$ are operating in the low SNR region since the sensors are unable to decode the packet.

Suppose $\hat{t}_1$ is used to seed the cross-correlation peak search. We wish to bound the time delay using the confidence region from Stage I, as well as the variance of the estimator $\hat{t}_1$. If t_0 is the time of transmission, then the true ToA at S_1, t_1, can be written as

$$t_1 = \frac{\|\boldsymbol{q}_1 - \boldsymbol{p}_1\|_2}{v} + t_0 \tag{81}$$

where v is the signal propagation velocity. Of course, the true Emitter position $\boldsymbol{p}_1$ is unknown, but a probabilistic bound may be derived. Let $t_m^{max} \geq t_m$ be the largest possible ToA at S_m, and $t_m^{min} \leq t_m$ be the smallest. Then Equation 82 gives the minimum and maximum ToAs at S_m. $\mathcal{C}_1$ is a set of valid (x, y) coordinates for the emitter based on the Stage I confidence region.

Set $\mathcal{C}_1$ can be constructed a few different ways. If multiple position estimates are made, a sample covariance matrix can estimated. Then, a confidence region can be computed. For a single measurement, the Fischer Information Matrix (FIM) for the TDoA estimates can be computed as given in Equation 73, and then used as the TDoA covariance matrix in the position FIM[4]

$$t_m^{min} = \min_{\boldsymbol{p}_1 \in \mathcal{C}_1} \frac{d_{m1}(\boldsymbol{p}_1)}{v} + t_1 \qquad t_m^{max} = \max_{\boldsymbol{p}_1 \in \mathcal{C}_1} \frac{d_{m1}(\boldsymbol{p}_1)}{v} + t_1$$

$$d_{m1}(\boldsymbol{p}) \triangleq d_m(\boldsymbol{p}) - d_1(\boldsymbol{p}), m = 2, \ldots, M \qquad d_k(\boldsymbol{p}) \triangleq \|\boldsymbol{q}_k - \boldsymbol{p}\|_2, k = 1, \ldots, M \tag{82}$$

An optimization problem is formed to find the position maximizing and minimizing the

[4] A minor correction to Eq. 33 of [20]: G_l^0 should read G_t^0, which agrees with the derivation in the Appendix of [20].

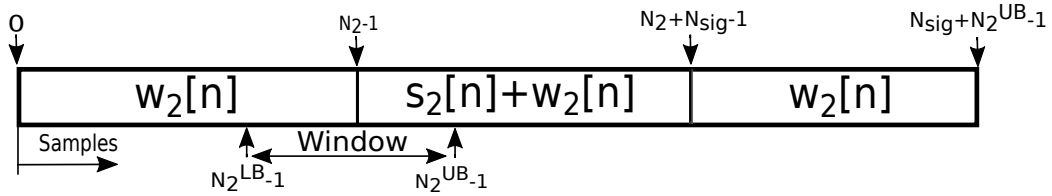


Figure 26: Diagram of samples received at sensor S_2. The window, $[N_2^{LB}-1, N_2^{UB}-1]$, shows where to search for N_2 with probability δ based on the ToA estimate at S_1, $\hat{N}_1$. Units are in samples.

TDoA, $\boldsymbol{p}_{max}(m)$ and $\boldsymbol{p}_{min}(m)$, respectively.

$$\boldsymbol{p}_{min}(m) = \arg\min_{\boldsymbol{p}_1 \in \mathcal{C}_1} d_{m1}(\boldsymbol{p}_1) \qquad \boldsymbol{p}_{max}(m) = \arg\max_{\boldsymbol{p}_1 \in \mathcal{C}_1} d_{m1}(\boldsymbol{p}_1) \tag{83}$$

In practice, S_m only has knowledge of the estimate $\hat{t}_1$, not the true value t_1. Suppose the estimate has the distribution given below.

$$\hat{t}_1 \sim \mathcal{N}\left(t_1, \sigma^2_{\hat{t}_1}\right) \tag{84}$$

Asymptotically, this is the distribution assuming the MLE estimate is used. $\sigma^2_{\hat{t}_1}$ is given by Equation 73. This is an asymptotic bound; for finite realizations, the distribution of $\hat{t}_1$ may be different and therefore the bound may not hold.

Suppose N_e estimates are made, and a confidence interval is constructed based on the sample mean.

$$\bar{t}_1 \triangleq \frac{1}{N_e}\sum_{i=1}^{N_e} \hat{t}_1 \sim \mathcal{N}\left(t_1, \frac{\sigma^2_{\hat{t}_1}}{N_e}\right) \tag{85}$$

We wish to select a lower bound T_m^{LB} and upper bound T_m^{UB} such that $\bar{t}_m^{max} \leq T_m^{UB}$ with probability δ_t and $\bar{t}_m^{min} \geq T_m^{LB}$ with probability δ_t.

$$\delta_t = \mathcal{P}\{\bar{t}_m^{max} \leq T_m^{UB}\} \qquad \delta_t = \mathcal{P}\{\bar{t}_m^{min} \geq T_m^{LB}\} \tag{86}$$

Sensor S_m selects a cross-correlation lag within this window $[T_m^{LB}, T_m^{UB}]$. Figure 26 illustrates this concept in units of samples for Sensor S_2. In units of samples, $N_m = t_m/T_s$,$N_m^{UB} = T_m^{UB}/T_s$, and $N_m^{LB} = T_m^{LB}/T_s$. Since $\bar{t}_m^{min}$ and $\bar{t}_m^{max}$ rely on $\hat{t}_1$, they are random variables as well.

$$\bar{t}_m^{min} = \frac{d_{m1}(\boldsymbol{p}_{min}(m))}{v} + \bar{t}_1 \qquad \bar{t}_m^{max} = \frac{d_{m1}(\boldsymbol{p}_{max}(m))}{v} + \bar{t}_1 \tag{87}$$

The upper and lower ToA bounds at sensor S_m are then given by Equation 88, where $\Phi(\cdot)$ is the standard normal cumulative distribution function.

$$T_m^{LB} = \bar{t}_1 + \frac{d_{m1}(\boldsymbol{p}_{min}(m))}{v} - \Phi^{-1}(\delta_t)\frac{\sigma_{\hat{t}_1}}{\sqrt{N_e}} \quad T_m^{UB} = \bar{t}_1 + \frac{d_{m1}(\boldsymbol{p}_{max}(m))}{v} + \Phi^{-1}(\delta_t)\frac{\sigma_{\hat{t}_1}}{\sqrt{N_e}} \tag{88}$$

Then, set a probability δ that both bounds are met. This is the confidence coefficient.

$$\begin{aligned}
\delta &= \mathcal{P}\left\{\left(\bar{t}_m^{max} \le T_m^{UB}\right) \cap \left(\bar{t}_m^{min} \ge T_m^{LB}\right)\right\} \\
&= \mathcal{P}\left\{\left(\frac{1}{v}d_{m1}(\boldsymbol{p}_{max}(m)) + \bar{t}_1 \le T_m^{UB}\right) \cap \left(\frac{1}{v}d_{m1}(\boldsymbol{p}_{min}(m)) + \bar{t}_1 \ge T_m^{LB}\right)\right\} \\
&= \mathcal{P}\left\{T_m^{LB} - \frac{1}{v}d_{m1}(\boldsymbol{p}_{min}(m)) \le \bar{t}_1 \le T_m^{UB} - \frac{1}{v}d_{m1}(\boldsymbol{p}_{max}(m))\right\} \\
&= \Phi\left(\frac{\sqrt{N_e}\left(T_m^{UB} - \frac{1}{v}d_{m1}(\boldsymbol{p}_{max}(m)) - t_1\right)}{\sigma_{\hat{t}_1}}\right) - \Phi\left(\frac{\sqrt{N_e}\left(T_m^{LB} - \frac{1}{v}d_{m1}(\boldsymbol{p}_{min}(m)) - t_1\right)}{\sigma_{\hat{t}_1}}\right) \\
&= \delta_t - \Phi\left(-\Phi^{-1}(\delta_t)\right) = \delta_t - (1-\delta_t) = 2\delta_t - 1
\end{aligned} \tag{89}$$

The last line follows by substitution of Equation 88. The final bounds are given, with the interpretation that the true value t_m is within $[T_m^{LB}, T_m^{UB}]$ with probability δ. This is essentially a confidence interval, but is not necessarily symmetric about $\bar{t}_1$ due to the values of $\boldsymbol{p}_{min}(m)$ and $\boldsymbol{p}_{max}(m)$.

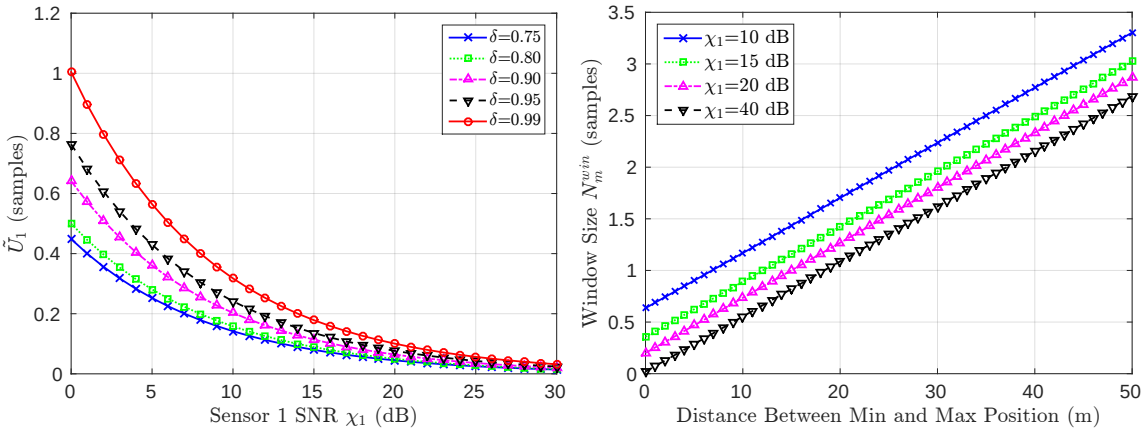
$$\begin{aligned}
T_m^{LB} &= \bar{t}_1 - \Phi^{-1}\left(\frac{\delta+1}{2}\right)\frac{\sigma_{\hat{t}_1}}{\sqrt{N_e}} + \frac{d_{m1}(\boldsymbol{p}_{min}(m))}{v} \\
T_m^{UB} &= \bar{t}_1 + \Phi^{-1}\left(\frac{\delta+1}{2}\right)\frac{\sigma_{\hat{t}_1}}{\sqrt{N_e}} + \frac{d_{m1}(\boldsymbol{p}_{max}(m))}{v}
\end{aligned} \tag{90}$$

Equation 91 is in units of samples, where $U_1 \triangleq \Phi\left(\frac{\delta+1}{2}\right)^{-1}\sigma_{\hat{t}_1}\beta\frac{1}{\sqrt{N_e}}$.

$$N_m^{LB} = \bar{N}_1 - U_1 + \frac{\beta}{v}d_{m1}(\boldsymbol{p}_{min}(m)) \qquad N_m^{UB} = \bar{N}_1 + U_1 + \frac{\beta}{v}d_{m1}(\boldsymbol{p}_{max}(m)) \tag{91}$$

If the signal spectrum is approximately rectangular, $\beta_{RMS}^2 = \beta^2/12$ [66], so U_1 can be simplified. This is a reasonable approximation for many digital modulations such as PSK, QAM, and OFDM (Sec. 4.9, [76]) with raised cosine pulse shaping. The window size, N_m^{win}, is then found.

$$N_m^{win} \triangleq N_m^{UB} - N_m^{LB} = 2\tilde{U}_1 + \frac{\beta}{v}\left(d_{m1}(\boldsymbol{p}_{max}(m)) - d_{m1}(\boldsymbol{p}_{min}(m))\right) \quad \tilde{U}_1 = \frac{\sqrt{3}\Phi^{-1}\left(\frac{\delta+1}{2}\right)}{\pi\sqrt{2\chi_1 N_e}} \tag{92}$$



(a) (b)

Figure 27: The overall search window size is shown in Figure 27b, with the contribution from the estimator uncertainty, in samples, shown in Figure 27a. δ is the probability the true ToA is within the window. Figure 27b plots the window size, N_m^{win}, as a function of the difference between maximum and minimum emitter positions, $d_{m1}\left(\boldsymbol{p}_{max}(m)\right) - d_{m1}\left(\boldsymbol{p}_{min}(m)\right)$ for $\delta = 0.99$ and one measurement ($N_e = 1$). See Figure 23 for a visualization of $d_{m1}(\boldsymbol{p})$. The signal bandwidth is $\beta = 16$ MHz and a rectangular spectrum is assumed.

Equation 92 suggests there are two components which contribute to the window size. $\tilde{U}_1$ is the estimator variance from S_1. The geometric component depends on the maximum and minimum emitter positions, which are a function of sensor geometry and the confidence region from Stage I. The quantity $d_{m1}(\boldsymbol{p})$ can be visualized from Figure 23. Figures 27a and 27b plot $\tilde{U}_1$ and N_m^{win}, respectively, assuming a rectangular spectrum in units of samples for one measurement.

To make use of these bounds, the cross-correlation is performed as in Equation 72. Then, $R_m[l]$ is restricted such that $l \in [N_m^{LB}, N_m^{UB}]$ and the lag maximizing the real part of the discrete cross-correlation is chosen as l_0. This is not necessary the true time delay, which occurs at lag l^*. No sub-sample interpolation is performed. In general, N_m^{LB} and N_m^{UB} will not be integers. One option which appears to work well in our simulations is rounding the bounds to the nearest integer.

$$l_0 = \arg\max_l \mathbb{R}\{R_m[l]\} \text{ such that } l \in [N_m^{LB}, N_m^{UB}] \tag{93}$$

4.4.2 Non-Decodable Sensor Estimator Variance

The position computation, as well as performance analysis, will require a variance for the windowed TDE approach of Equation 93. One typical assumption is that the cross-correlation lag selected follows the uniform distribution [66]. Under this case, the variance of the estimate is simply that of a discrete uniform random variable.

However, the discrete uniform distribution is the maximum entropy distribution for bounded discrete support; it is likely to be pessimistic in our model where the signal auto-correlation function is known. Additionally, this assumption gives equal weight to all TDoA measurements from non-decodable sensors. But some sensors in the non-decodable set may have a much higher probability of choosing the correct maximum integer lag than others due to SNR differences across sensors. If the probability of choosing the correct maximum integer lag can be written as a function of window size and SNR, then a commensurate weighting of the TDoA estimates can be used. Intuitively, this should lead to an increase in localization performance.

For notational simplicity, we drop the subscript "m" on the received samples $y_m[n]$ and the noise $w_m[n]$. The cross-correlation at a single non-decoding sensor is considered and the $w_m[n]$ are assumed to be i.i.d. so the result can be generalized for all sensors. Also, $N_0 \triangleq N_m$, is the particular time delay of interest. The cross-correlation distribution under these assumptions is given by Lemma 1.

Lemma 1 (Cross-Correlation Distribution)**.** *Suppose* $\boldsymbol{s} \in \mathbb{C}^{N_{sig}}$ *is a known signal vector and* $\boldsymbol{y} = \left[\begin{array}{c|c|c} \boldsymbol{w}_1 & \boldsymbol{s}+\boldsymbol{w}_2 & \boldsymbol{w}_3 \end{array} \right]^T \in \mathbb{C}^{N_{meas}}$ *with independent noise vectors* $\boldsymbol{w}_1 \sim \mathcal{CN}\left(\boldsymbol{0}_{N_0}, \boldsymbol{C}_1\right)$, $\boldsymbol{w}_2 \sim \mathcal{CN}\left(\boldsymbol{0}_{N_{sig}}, \boldsymbol{C}_2\right)$, *and* $\boldsymbol{w}_3 \sim \mathcal{CN}\left(\boldsymbol{0}_{N_{meas}-N_0-N_{sig}}, \boldsymbol{C}_3\right)$. *Then the cross-correlation vector* $\boldsymbol{k}_{ys} = \boldsymbol{D}\boldsymbol{y}$, *where* $\boldsymbol{D}$ *is a Toeplitz convolution matrix constructed from the matched*

filter of $\boldsymbol{s}$*, denoted* $\boldsymbol{h}$*, and has the following distribution.*

$$\boldsymbol{k}_{ys} \sim \begin{cases} \mathcal{CN}\left(\mathbf{0}_{N_0}, \boldsymbol{A}_{1,n}\boldsymbol{C}_1\boldsymbol{A}_{1,n}^H\right) & 0 \le l \le N_0 - 1 \\ \mathcal{CN}\left(\boldsymbol{A}_{2,o}\boldsymbol{s}, \begin{bmatrix}\boldsymbol{A}_{1,o} & \boldsymbol{A}_{2,o}\end{bmatrix}\begin{bmatrix}\boldsymbol{C}_1 & \mathbf{0} \\ \mathbf{0} & \boldsymbol{C}_2\end{bmatrix}\begin{bmatrix}\boldsymbol{A}_{1,o}^H \\ \boldsymbol{A}_{2,o}^H\end{bmatrix}\right) & N_0 \le l \le N_{sig} + N_0 - 2 \\ \mathcal{CN}\left(\boldsymbol{A}_{2,n}\boldsymbol{s}, \boldsymbol{A}_{2,n}\boldsymbol{C}_2\boldsymbol{A}_{2,n}^H\right) & l = l^* = N_{sig} + N_0 - 1 \\ \mathcal{CN}\left(\tilde{\boldsymbol{A}}_{2,o}\boldsymbol{s}, \begin{bmatrix}\tilde{\boldsymbol{A}}_{2,o} & \boldsymbol{A}_{3,o}\end{bmatrix}\begin{bmatrix}\boldsymbol{C}_2 & \mathbf{0} \\ \mathbf{0} & \boldsymbol{C}_3\end{bmatrix}\begin{bmatrix}\tilde{\boldsymbol{A}}_{2,o}^H \\ \boldsymbol{A}_{3,o}^H\end{bmatrix}\right) & N_{sig} + N_0 \le l \le 2N_{sig} + N_0 - 2 \\ \mathcal{CN}\left(\mathbf{0}, \boldsymbol{A}_{3,n}\boldsymbol{C}_3, \boldsymbol{A}_{3,n}^H\right) & 2N_{sig} + N_0 - 1 \le l \le N_{meas} + N_{sig} - 2 \end{cases}$$

$\boldsymbol{A}_1$*,*$\boldsymbol{A}_2$*,*$\boldsymbol{A}_3$*,*$\boldsymbol{A}_{1,n}$*,*$\boldsymbol{A}_{2,o}$*,*$\boldsymbol{A}_{2,n}$*,*$\boldsymbol{A}_{3,o}$*, and* $\boldsymbol{A}_{3,n}$ *are constructed from rows of* $\boldsymbol{D}$ *and are defined in Equation 140 of Appendix B.1.*

Proof. See Appendix B.1. □

To gain some insight into how the SNR and window size affect the probability of choosing the correct cross-correlation lag, we consider an ideal impulse auto-correlation function of Equation 94 with simplified noise covariances. For digital modulations with random symbols, this idealized auto-correlation function is reasonable except for the contribution of the band-limited pulse shape around l^*. While the uniform distribution leads to a pessimistic variance at higher SNRs, this assumption is optimistic. In practice, the peak-to-sidelobe ratio of realistic auto-correlation functions will be lower, resulting in a higher probability that the neighboring samples around the true maximum l^* are chosen. Nonetheless, this assumption leads to some useful analytical results and insight. Lemma 1 can always be used to numerically calculate the probabilities for a particular autocorrelation function.

Lemma 2 (Ideal Cross-Correlation Distribution)**.** *Suppose* $\boldsymbol{C}_1 = \sigma^2\boldsymbol{I}_{N_0}, \boldsymbol{C}_2 = \sigma^2\boldsymbol{I}_{N_{sig}}$*,*

$\boldsymbol{C}_3 = \sigma^2 \boldsymbol{I}_{(N_{meas}-N_0-N_{sig})}$, *and the signal auto-correlation,* $\tilde{\boldsymbol{k}}_{ss}$, *is given below.*

$$\tilde{\boldsymbol{k}}_{ss} = \begin{bmatrix} h[0] & 0 & 0 & \dots & 0 \\ h[1] & h[0] & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ h[N_{sig}-1] & h[N_{sig}-2] & \dots & \dots & h[0] \\ 0 & h[N_{sig}-1] & h[N_{sig}-2] & \dots & h[1] \\ 0 & 0 & \dots & 0 & h[N_{sig}-1] \end{bmatrix} \boldsymbol{s} = \begin{bmatrix} \boldsymbol{0}_{N_{sig}-1} \\ \mathcal{E}_{sig} \\ \boldsymbol{0}_{N_{sig}-1} \end{bmatrix} \tag{94}$$

Under these assumptions, the ideal cross-correlation vector is distributed as follows.

$$\tilde{\boldsymbol{k}}_{ys} \sim \begin{cases} 0 \le l \le N_0 - 1 \\ \mathcal{CN}\left(\boldsymbol{0}_{N_0}, \sigma^2 \operatorname{Diag}\left(|h[0]|^2, |h[0]|^2 + |h[1]|^2, \dots, \sum_{n=0}^{N_0-1} |h[n]|^2\right)\right) \\ N_0 \le l \le N_{sig} + N_0 - 2 \\ \mathcal{CN}\left(\boldsymbol{0}_{N_{sig}-1}, \sigma^2 \operatorname{Diag}\left(\sum_{n=0}^{N_0} |h[n]|^2, \dots, \sum_{n=0}^{N_{sig}-1} |h[n]|^2 = \mathcal{E}_{sig}, \dots, \mathcal{E}_{sig}\right)\right) \\ l = l^* = N_{sig} + N_0 - 1 \\ \mathcal{CN}\left(\mathcal{E}_{sig}, \sigma^2 \mathcal{E}_{sig}\right) \\ N_{sig} + N_0 \le l \le 2N_{sig} + N_0 - 2 \\ \mathcal{CN}\left(\boldsymbol{0}_{(N_{sig}-1)}, \sigma^2 \mathcal{E}_{sig} \boldsymbol{I}_{N_{sig}-1}\right) \\ 2N_{sig} + N_0 - 1 \le l \le N_{meas} + N_{sig} - 2 \\ \mathcal{CN}\left(\boldsymbol{0}_{(N_{meas}-N_{sig}-N_0)}, \sigma^2 \operatorname{Diag}\left(\sum_{n=0}^{N_{sig}-1} |h[n]|^2, \dots, |h[N_{sig}-1]|^2\right)\right) \end{cases}$$

Proof. See Appendix B.2. □

The cross-correlation is windowed around $\hat{N}_1$ for a particular sensor S_m. The window is a subset of the distribution defined in Lemma 2. To make the analysis easier, assume a_m and b_m are chosen such that the cross-correlation distribution is identical on either side of $\hat{N}_1$. The requirement is that b_m is restricted to $0 \le b_m \le N_{sig} - 1$, and $0 \le a_m \le N_m$ as there are N_m $\sigma^2 \mathcal{E}_{sig}$ terms for $l \in [N_m, N_{sig} + N_m - 2]$. Then Lemma 2 simplifies. The factor of one half is due to the variance splitting equally between the real and imaginary

random variables (pg. 307, [66]).

$$\boldsymbol{g}_{ys}[l] = \mathbb{R}\{\tilde{\boldsymbol{k}}_{ys}[l]/\sigma^2\} \sim \begin{cases} \mathcal{N}\left(0, \frac{\chi}{2}\right) & l \neq l^* \\ \mathcal{N}\left(\chi, \frac{\chi}{2}\right) & l = l^* \end{cases} \qquad 0 \leq a_m \leq N_m, 0 \leq b_m \leq N_{sig} - 1$$

$$l \in [\hat{N}_1 - a_m, \hat{N}_1 + b_m] = \mathcal{W} \qquad \mathcal{P}\{l^* \in \mathcal{W}\} = \delta \tag{95}$$

Theorem 1. *Suppose $l^* \in \mathcal{W}$, $\boldsymbol{g}_{ys}[l]$ is distributed as in Equation 95, $\hat{N}_1$ is an integer, and the joint amplitude random variables are independent conditioned on $\boldsymbol{g}_{ys}[l^*]$. Then the probability p that the true lag l^* is the maximum within the window, $l_0 = l^*$, is given as a function of the SNR χ_m and window bounds a_m, b_m.*

$$p \triangleq \mathcal{P}\{l_0 = l^* = \arg\max_l \boldsymbol{g}_{ys} \mid l, l^* \in \mathcal{W}\} = \int_{-\infty}^{\infty} \Phi^{a_m+b_m}\left(\frac{\sqrt{2}\alpha}{\sqrt{\chi_m}}\right) f_{\boldsymbol{g}_{ys}[l^*]}(\alpha) d\alpha$$

$$f_{\boldsymbol{g}_{ys}[l^*]}(\alpha) = \frac{\sqrt{2}}{\sqrt{\pi}\chi_m} \exp\{-2(\alpha - \chi_m)^2/\chi_m^2\}$$

Proof. See Appendix C.1. □

Theorem 2. *Suppose $l^* \in \mathcal{W}$, $\boldsymbol{g}_{ys}[l]$ is distributed as in Equation 95, $\hat{N}_1$ is an integer, and the joint amplitude random variables are independent conditioned on $\boldsymbol{g}_{ys}[l_0]$. Then the probability $\tilde{p}$ that another lag, $l_0 = l^* + k, k \neq 0$, is the maximum within the window, is given as a function of the SNR χ_m and window bounds a_m, b_m.*

$$\tilde{p} \triangleq \mathcal{P}\{l_0 = l^* + k = \arg\max_k \boldsymbol{g}_{ys}[l^* + k] \mid l^* \in \mathcal{W}\}, k \in [-a_m, -a_m + 1, \ldots, -1, 1, \ldots, b_m]$$

$$= \int_{-\infty}^{\infty} \Phi\left(\frac{\sqrt{2}\,(\alpha - \chi_m)}{\sqrt{\chi_m}}\right) \Phi\left(\frac{\sqrt{2}\alpha}{\sqrt{\chi_m}}\right)^{a_m+b_m-1} f_{\boldsymbol{g}_{ys}[l^*+k]} d\alpha$$

$$f_{\boldsymbol{g}_{ys}[l^*+k]}(\alpha) = \frac{\sqrt{2}}{\sqrt{\pi}\chi_m} \exp\{-2\alpha^2/\chi_m^2\}$$

Proof. See Appendix C.2. □

Interestingly, the distribution of lag $l = l^* + k, k \neq 0$ given in Equation 95 is identical to the "signal absent" sufficient statistic distribution, and the $l = l^*$ lag is identical to the "signal present" sufficient statistic distribution for signal detection in coherent radar receivers using the Likelihood Ratio Test (pg. 309, [66]).

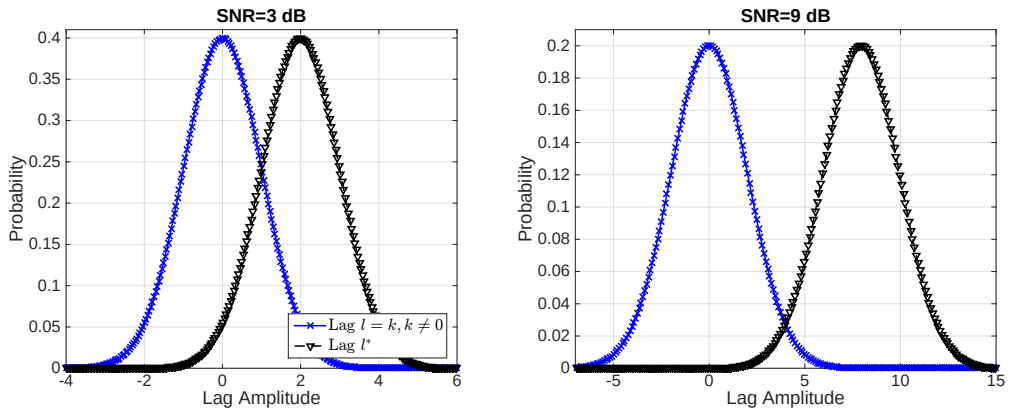
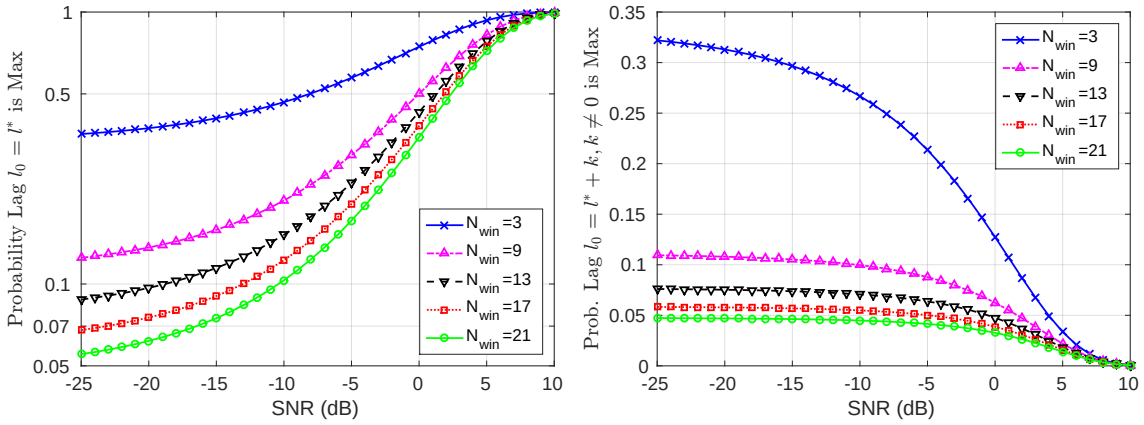


Figure 28: Distribution of the real part of the cross-correlation function normalized by the noise variance σ^2 for 3 and 9 dB SNR. As the SNR increases, the overlapping area between the two distributions decreases. This implies a decrease in the probability that the wrong cross-correlation lag is selected. The signal auto-correlation is assumed to be an impulse.

Figure 28 plots the distribution of the normalized cross-correlation amplitude $\tilde{\boldsymbol{g}}_{ys}[l]$ for two different SNRs. As expected, as the SNR increases, the distribution means move farther apart, decreasing the probability that the wrong cross-correlation lag is selected. The variances are identical.

Next, we numerically evaluate the probabilities that a particular lag l_0 is the maximum as a function of window size and SNR. Let $a_m = b_m$. The window size is then defined as $N_{win} = 2a_m + 1, N_{win} \in \{1, 2, \ldots, 2N_0 + 1\}$. The integrals of Theorem 1 and Theorem 2 are evaluated numerically and plotted as a function of window size and SNR in Figure 29. Numerically, this figure demonstrates that as the linear SNR approaches zero, the distribution converges to uniform. Figure 30a plots the probabilities of the lags in a three sample window, with a comparison to the uniform distribution assumption. This plot illustrates the uniform assumption is pessimistic until the SNR is exceptionally low, around -15 dB. This is significant for the algorithm because it provides a better bound of the variance than the uniform assumption for sensors which can not decode the packet but have moderate SNR.

Finally, the variance of the estimator can be calculated. Assume $a_m \neq b_m$ and consider the random variable L. Without loss of generality, and to simplify the calculations, assume



(a) (b)

Figure 29: Probability of choosing a particular lag l in a windowed cross-correlation as a function of SNR and window size. 29a plots the probability that the true cross-correlation lag, l^*, is the maximum lag l_0 in a window of size N_{win}, given that l^* exists in the window. 29b plots the probability that another lag, $l_0 = l^* + k, k \neq 0$, is the maximum in the window. The signal is assumed to have an impulse auto-correlation function.

$l^* = 0$.

$$L(l) \sim \begin{cases} p & l = 0 \\ \tilde{p} & l \neq 0 \end{cases} \qquad l \in [-a_m, b_m], a \geq 0, b \geq 0 \tag{96}$$

L represents the probability that lag l_0 is the maximum in the cross-correlation window. After some calculations, Equation 97 gives the mean, and Equation 98 gives the variance. The calculations are provided in Appendix D.

$$\mathbb{E}\{L\} = \mu_L = \tilde{p}(b - a) \tag{97}$$

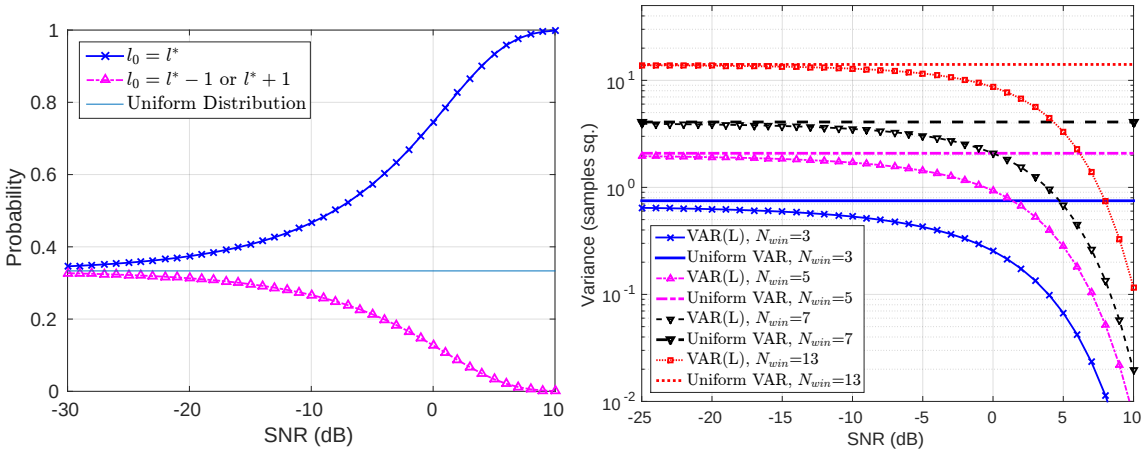
$$\text{VAR}\{L\} = \frac{\tilde{p}}{6}\left[a(a+1)(2a+1) + b(b+1)(2b+1)\right] - \tilde{p}^2(b-a)^2 \tag{98}$$

4.4.3 Decodable Sensor ($S_m \in \Gamma_{dec}$) Time Delay Estimation

For sensors which can decode the packet, the procedure is identical to Stage I.

4.4.4 Position Estimation Using All Sensors

The position estimate can now be computed from the algorithm in Appendix A with $M_d = M$ and an appropriate covariance matrix. For decodable sensors, the ToA variance is simply



(a) (b)

Figure 30: Probabilities and variance of maximum lag estimate random variable L versus the typical uniform assumption. 30a shows the probability that lag $l^* - 1$, l^*, and $l^* + 1$ is the maximum in the cross-correlation function for a three sample window. The horizontal line shows the uniform distribution assumption for comparison. 30b compares the variance of L with the uniform distribution variance, assuming $a_m = b_m$. It is assumed the signal has an impulse auto-correlation function.

the CRLB. For non-decodable sensors, either the uniform variance or the variance of L given by Equation 98 can be used. In practice, this should be computed using the true signal auto-correlation function.

$$\boldsymbol{J}_{ToA}[m,m] = \begin{cases} \frac{T_s^2}{8\pi^2 \chi_m \beta_{RMS}^2} & m \in \{a | S_a \in \Gamma_{dec}\} \\ \frac{(T_s N_{win})^2}{12} \text{ or } T_s^2 \text{VAR}\{L\} & m \in \{a | S_a \in \Gamma_{ndec}\} \end{cases}$$

$$\boldsymbol{J}_{ToA}[m,k] = 0, m \in \{1,2,\ldots,M\}, k = \{1,2,\ldots,M\}, k \neq m \tag{99}$$

As in Stage I, the position error ellipses $\mathcal{C}_2$, are computed from the covariance matrix. The intermediate Stage I and II CR is the intersection of the sets, as shown in Figure 31. In general, the intersection is not simply $\mathcal{C}_2$.

$$\mathcal{C}_2' = \mathcal{C}_1 \cap \mathcal{C}_2 \tag{100}$$

4.5 Stage III: MAC-Assisted Positioning

Except for using the MAC address to associate TDoA measurements to a particular emitter, Stages I and II do not leverage the MAC Layer. Additionally, they consider TDoA only

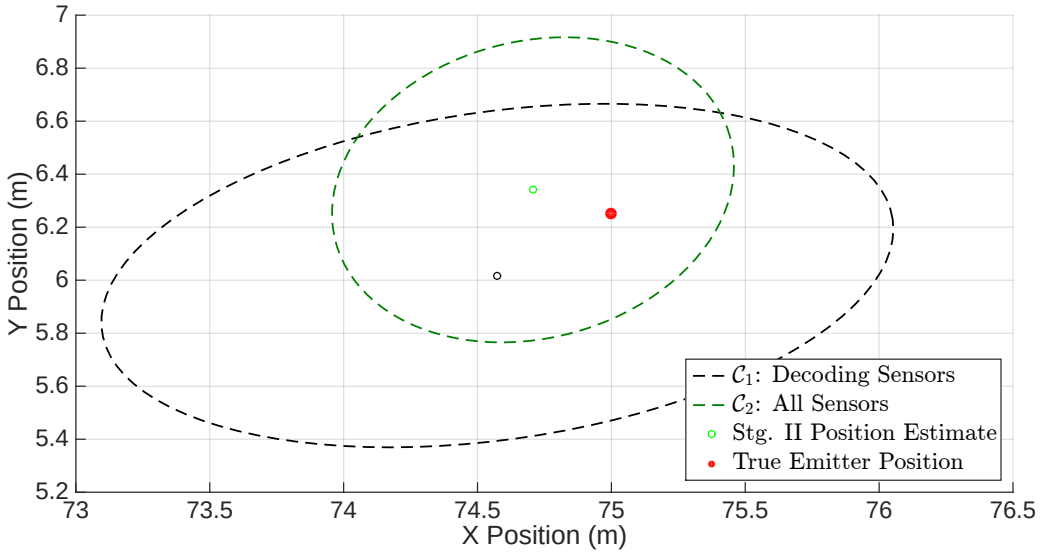


Figure 31: Example Stage I and II algorithm simulation. The decoding sensors in Stage I compute confidence region $\mathcal{C}_1$, depicted with black dashed lines. This confidence region was used to refine the position estimate in Stage II with associated confidence region $\mathcal{C}_2$, shown with green dashed lines.

on a per packet basis. In Stage III, the interframe (packet) spacing is used to augment the position estimate.

To proceed with analysis, a particular standard must be chosen. In the system model of Figure 23, E_2 is now considered the AP and E_1 is the client. An IEEE 802.11g network in Infrastructure BSS mode using the Distributed Coordination Function (DCF) is assumed. The MAC mechanism is therefore CSMA/CA and the PHY is chosen as OFDM. However, the technique described is sufficiently broad that it could be applied to other communications protocols with fixed interframe spacing time and packet exchange sequences. An RTS/CTS PES is considered for this analysis. Figure 32 illustrates the packet sequence without packet fragmentation.

4.5.1 Analysis of MAC-Assisted Positioning

Consider taking the ToA difference between two *packets* at S_m. The distance between E_1 and E_2 can be estimated using the packet timing, further refining the position estimates from Strategies One and Two. Notation used in this section is listed in Table 6 by order of appearance.

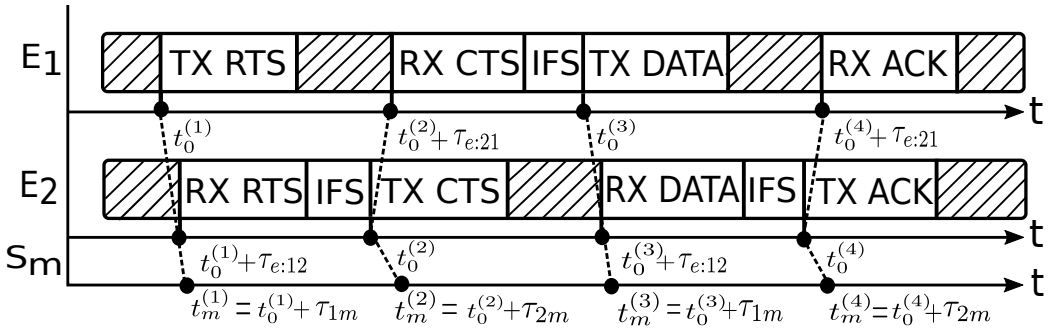


Figure 32: Packet Timing Diagram for an IEEE 802.11g RTS/CTS Packet Exchange Sequence. E_1 is the emitter of interest, E_2 is an AP, and S_m is the m^{th} sensor node. The interframe spacing timing jitter is not shown for clarity.

Table 6: Notation

Symbol	Description
$\Delta t_m^{(ij)}$	Packet TDoA between Packet i and Packet j at Sensor S_m (s)
$t_m^{(i)}$	Time of Arrival for packet i (sec),$m \in \{1, 2, \ldots, M\}$
$t_0^{(i)}$	Time of transmission for packet i
τ_{nm}	Time of Flight from Emitter E_n to Sensor S_m (s)
$\tau_{e:nk}$	True Time of Flight from Emitter E_n to E_k (s)
$\hat{\tau}_{e:nk}(i,m)$	Est. ToF by S_m from E_n to E_k using Packets i and 1 (s)
T_{CTS}	Length of a CTS Packet (s)
T_{IFS}	Standard-defined Interframe Spacing Time (s)
T_{RTS}	Length of a RTS Packet (s)
T_{JIFS}	Interframe Spacing Time Random Variable (s)
b_{IFS}	Interframe Spacing bias from T_{IFS} (s)
σ^2_{IFS}	Interframe spacing variance (sec^2)
v	Propagation Velocity in the medium (m/s)
R_{nk}	True Distance beteween Emitters E_n and E_k
$\hat{R}_{nk}(i,m)$	Distance estimate by S_m between E_n and E_k using Packets i and 1 (s)
$\sigma^2_{\hat{\tau}_m^{(i)}}$	Variance of timing estimate for Packet i received at Sensor S_m (sec^2)
$\sigma^2_{\hat{R}_{nk}(i,m)}$	Variance distance estimate between emitters E_n and E_k (sec^2)
$\chi_m^{(i)}$	Linear SNR of i^{th} packet for sensor S_m
α_m	Combined SNR received at sensor S_m for RTS and DATA packets.
T_{DATA}	Length of a DATA Packet (s)
T_{ACK}	Length of an ACK Packet (s)
σ^*_{IFS}	IFS std. dev. where required SNR approaches infinity (s)
δ_R	Confidence coefficient of the radius estimate $\hat{R}_{nk}(i,m)$ (meters)
N_R	Number of sensors participating in Stage 3 localization
$\bar{R}_{nk}$	Best Linear Unbiased Estimate (BLUE) of R_{nk} (meters)
R_{nk}^{LB} / R_{nk}^{UB}	Lower/upper bounds on R_{nk} using $\bar{R}_{nk}$ (s)

$$\begin{aligned}\Delta t_m^{(31)} = t_m^{(3)} - t_m^{(1)} &= \left(t_0^{(3)} + \tau_{m1}\right) - \left(t_0^{(1)} + \tau_{m1}\right) = \left(t_0^{(2)} + \tau_{e:21} + T_{CTS} + T_{IFS}\right) - t_0^{(1)} \\ &= \left(t_0^{(1)} + \tau_{e:12} + T_{RTS} + T_{IFS}\right) + \tau_{e:21} + T_{CTS} + T_{IFS} - t_0^{(1)} \\ &= 2\tau_{e:12} + 2T_{IFS} + T_{RTS} + T_{CTS} \end{aligned} \tag{101}$$

Equation 101 assumes no interframe spacing jitter. That is, the emitters precisely follow the interframe spacing time, T_{IFS} given in the standard. In practice, the interframe spacing should be considered a random variable, T_{JIFS}, not necessarily having a mean of T_{IFS} [32, 16]. We assume independent, identically distributed Gaussian random variables with mean $T_{IFS} + b_{IFS}$ and variance σ_{IFS}^2 on both emitters. The bias b_{IFS}, although it may be a function of the manufacturer of the WLAN IC [32], can be estimated and removed. More critical is the variance of the estimates. Bourchas et al. [16] shows that the deviation of the median from the first estimate is between 10-20 ns. A more comprehensive study for various WLAN IC manufacturers should be undertaken. The packet lengths T_{RTS} and T_{CTS}, and propagation velocity v are known exactly. Then R_{12}, the distance between emitters E_1 and E_2, can be estimated.

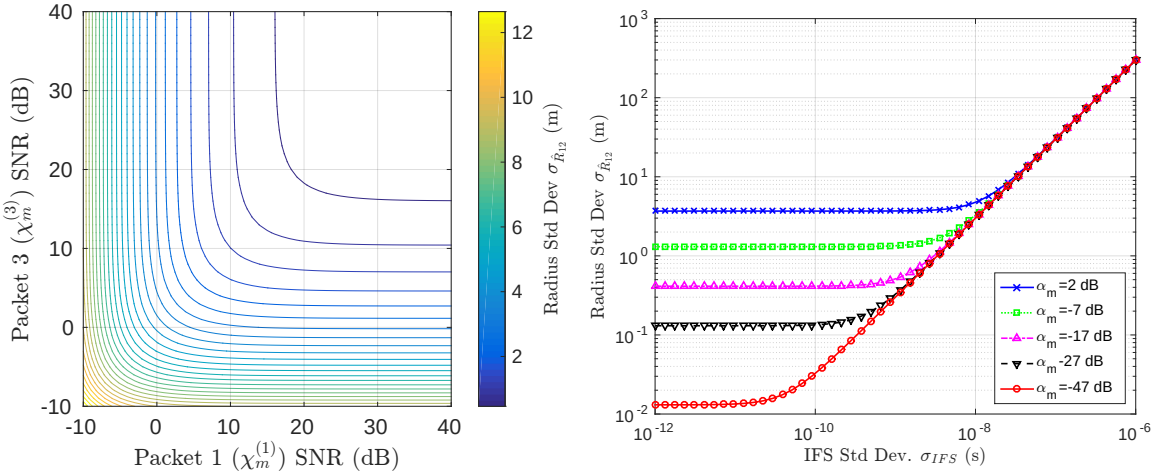
$$T_{JIFS} \sim \mathcal{N}\left(T_{IFS} + b_{IFS}, \sigma_{IFS}^2\right) \tag{102}$$

With this substitution, R_{12} is computed. $\tau_{e:12}$ is the Time of Flight (ToF) estimate between E_1 and E_2.

$$R_{12} = v\tau_{e:12} = \frac{v}{2}\left(\Delta t_m^{(31)} - 2T_{JIFS} - T_{RTS} - T_{CTS}\right) \tag{103}$$

Of course, the distance, and by extension, the ToF, between stationary emitters, does not change. However, the sensors must estimate this quantity. The estimate itself is a random variable which is a function of both the sensor index m and packet number i. We explicitly show the dependence of the estimates $\hat{R}_{12}(i,m)$ and $\hat{\tau}_{e:12}(i,m)$ in our notation to make this clear. For the present analysis, only the time difference between the RTS and DATA packets are considered. The DATA packet is third in the sequence, therefore $\hat{R}_{12}(m) \triangleq \hat{R}_{12}(3,m)$ and $\hat{\tau}_{e:12}(m) \triangleq \hat{\tau}_{e:12}(3,m)$. If the AP sends the RTS, then the CTS and ACK packets could be used with identical results.

The variance of $\hat{R}_{12}(m)$ is then computed. It is assumed that only sensors which have sufficiently high SNR such that the CRLB applies attempt to estimate $\Delta t_m^{(31)}$. These are



(a) (b)

Figure 33: Figure 33a is a contour plot of the standard deviation of the distance estimate between E_1 and E_2, $\sigma_{\hat{R}_{12}(m)}$ without IFS jitter. The independent variables are the SNRs of the packets, $\chi_m^{(1)}$ and $\chi_m^{(3)}$, respectively, received at sensor S_m in dB. Figure 33b plots this distance standard deviation versus the interframe spacing deviation σ_{IFS} for various combined SNRs $\alpha_m \triangleq \frac{\chi_m^{(1)}+\chi_m^{(3)}}{\chi_m^{(1)}\chi_m^{(3)}}$. The signal bandwidth is $\beta = 20$ MHz and propagation velocity was the speed of light in a vacuum, $v = c$.

likely to be sensors in the decodable set, but could also include some in the non-decodable set, depending on the location of the emitters. Assume $\hat{\Delta}t_m^{(31)}$ is statistically independent from T_{JIFS}. Then asymptotically, $\hat{\Delta}t_m^{(31)} \sim \mathcal{N}\left(\Delta t_m^{(31)}, \sigma^2_{\hat{\tau}_m^{(3)}} + \sigma^2_{\hat{\tau}_m^{(1)}}\right)$ and Equation 104 gives the distribution of the ToF estimate between the AP and E_1.

$$\hat{\tau}_{e:12}(m) \sim \mathcal{N}\left(\frac{1}{2}\Delta t_m^{(31)} - \mu, \frac{1}{4}\left(\sigma^2_{\hat{\tau}_m^{(3)}} + \sigma^2_{\hat{\tau}_m^{(1)}}\right) + \sigma^2_{IFS}\right), \mu \triangleq T_{IFS} + b_{IFS} + \frac{1}{2}T_{RTS} + \frac{1}{2}T_{CTS} \tag{104}$$

Ultimately, the variance of the distance estimate, $\hat{R}_{12}(m)$ is the statistic of interest. Under the rectangular signal spectrum assumption, this is given in Equation 105.

$$\text{VAR}\left\{\hat{R}_{12}(m)\right\} = \sigma^2_{\hat{R}_{12}(m)} = \frac{v^2}{4}\left(\sigma^2_{\hat{\tau}_m^{(3)}} + \sigma^2_{\hat{\tau}_m^{(1)}}\right) + v^2\sigma^2_{IFS} = \frac{3v^2}{8\pi^2\beta^2}\left(\frac{\chi_m^{(1)}+\chi_m^{(3)}}{\chi_m^{(1)}\chi_m^{(3)}}\right) + v^2\sigma^2_{IFS} \tag{105}$$

If the combined packet SNR at sensor S_m is defined as $\alpha_m \triangleq \frac{\chi_m^{(1)}+\chi_m^{(3)}}{\chi_m^{(1)}\chi_m^{(3)}}$, then Equation 105 can be used to plot the standard deviation of the radius estimate. Figure 33a plots the standard deviation of $\hat{R}_{12}(m)$ as a function of packet one and packet three SNRs, $\chi_m^{(1)}$ and $\chi_m^{(3)}$, respectively without IFS jitter. Figure 33b plots $\sigma_{\hat{R}_{12}(m)}$ as a function of the standard

deviation of the interframe spacing time σ_{IFS} on a log-log plot.

Figure 33b illustrates the piecewise nature of $\sigma_{\hat{R}_{12}(m)}$. For sufficiently small σ_{IFS}, $\sigma_{\hat{R}_{12}(m)}$ is dominated by the ToA estimator variance. This is the flat region of the function on the left. Conversely, σ_{IFS} dominates for sufficiently small α_m, and by extension, $\sigma_{\hat{t}_m^{(3)}} + \sigma_{\hat{t}_m^{(1)}}$. This is the linear region of the function on the right. For $\alpha_m \approx -17$ dB, $\sigma_{\hat{R}_{12}(m)}$ is about half a meter for σ_{IFS} around a nanosecond. For equal packet SNRS $\chi_m^{(1)} = \chi_m^{(3)}$, this corresponds to a packet SNR of 20 dB. This analysis demonstrates it is theoretically possible to significantly increase localization accuracy by using the PES timing to estimate the distance from the emitter of interest to an AP with a *known* position.

Next, a DATA/ACK PES with packet fragmentation case is considered. It will be shown that longer packet exchange sequences result in lowering the distance estimator variance. Consider the PTDoA between packet i and packet 1, where T_{DATA} and T_{ACK} are the length of a DATA and ACK packet, respectively.

$$\begin{aligned}\Delta t_m^{(i1)} &= t_m^{(i)} - t_m^{(1)} = \left(t_0^{(i)} + \tau_{1m}\right) - \left(t_0^{(1)} + \tau_{1m}\right), i = 3, 5, \ldots \\ &= (i-1)\,\tau_{e:12} + (i-1)\,T_{JIFS} + \frac{1}{2}(i-1)\,T_{DATA} + \frac{1}{2}(i-1)\,T_{ACK} \end{aligned} \tag{106}$$

Then, rearrange in terms of $\tau_{e:12}(i,m)$, where i denotes the packet number used for the PTDoA.

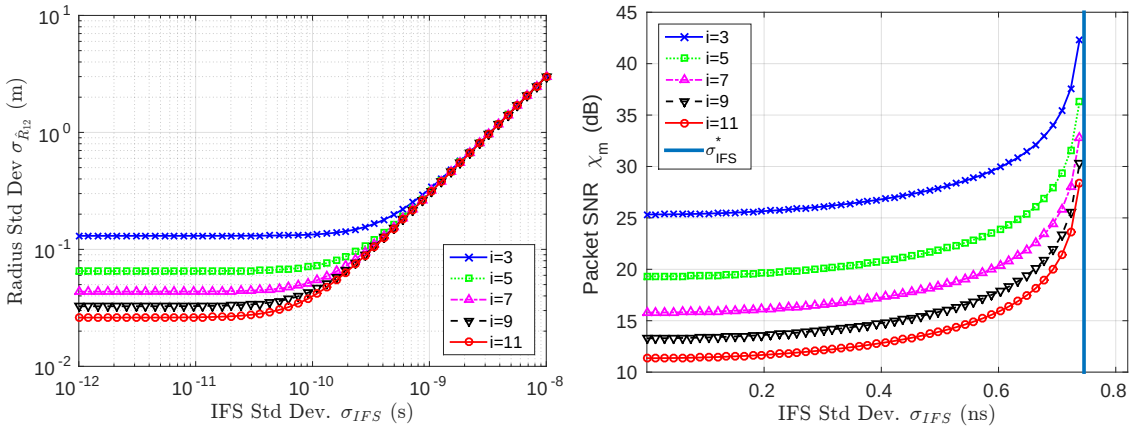
$$\tau_{e:12}(i,m) = \frac{\Delta t_m^{(i1)}}{i-1} - T_{JIFS} - \frac{1}{2}(T_{DATA} - T_{ACK}) \tag{107}$$

The distribution of $\hat{\tau}_{e:12}(i,m)$ is then found as Equation 108.

$$\begin{aligned}\hat{\tau}_{e:12}(i,m) &\sim \mathcal{N}\left(\frac{\Delta t_m^{(i1)}}{i-1} - \mu_f, \frac{\sigma^2_{\hat{\tau}_m^{(i)}} + \sigma^2_{\hat{\tau}_m^{(1)}}}{(i-1)^2} + \sigma^2_{IFS}\right) \\ \mu_f &\triangleq T_{IFS} + b_{IFS} + \frac{1}{2}(T_{DATA} + T_{ACK}) \end{aligned} \tag{108}$$

Finally, the variance of the radius estimate is derived as Equation 109. Critically, as the number of packets in the sequence increases, the variance approaches $v^2\sigma^2_{IFS}$. To an approximation, the variance in the timing estimate decreases proportional to the inverse square of the length of the PES.

$$\mathrm{VAR}\{\hat{R}_{12}(i,m)\} = \frac{3v^2}{2\pi^2\beta^2\,(i-1)^2}\left(\frac{\chi_m^{(1)} + \chi_m^{(3)}}{\chi_m^{(1)}\chi_m^{(3)}}\right) + v^2\sigma^2_{IFS}, i = 3, 5, \ldots \tag{109}$$



(a) (b)

Figure 34: The effect of interframe spacing jitter on the distance estimate is shown in Figure 34a, and SNR in Figure 34b. 34a is the standard deviation of the distance estimate between E_1 and E_2, $\hat{R}_{12}(i)$ as a function of the interframe spacing standard deviation σ_{IFS} and the length of the packet exchange sequence i. The combined signal SNR is $\alpha_m = -27$ dB. 34b plots the required packet SNR in dB as a function of σ_{IFS} for various packet exchange sequence lengths i. The required radius estimator variance is $\sigma^2_{\hat{R}_{12}} = 0.05$ m^2. The σ^*_{IFS} line represents the value of σ_{IFS} where the packet SNR approaches infinity. Packets are assumed to have identical SNR. The signal bandwidth is $\beta = 20$ MHz and propagation velocity was the speed of light in a vacuum, $v = c$

Figure 34a plots Equation 109 as a function of IFS timing standard deviation and PES length i for a fixed bandwidth, propagation velocity, and combined SNR α_m. For a fixed α_m, the standard deviation of the distance estimate between E_1 and E_2 can be significantly lowered if the PTDoA is taken between packets which are the furthest apart in time from one another. In other words, it is possible to mitigate the effects of higher variance ToA estimates by taking the PTDoA across a larger time span. However, the variance will never be lower than the variance of the interframe spacing.

It is insightful to look at the packet SNR required for a fixed radius estimator variance $\sigma^2_{\hat{R}_{12}(i,m)}$. Suppose packets have equal SNR, that is $\chi_m^{(i)} = \chi_m^{(1)} = \chi_m$. Then solve Equation 109 as a function of χ_m.

$$\chi_m = \frac{6v^2}{2\pi^2\beta^2 (i-1)^2 \left(\sigma^2_{\hat{R}_{12}(i,m)} - v^2\sigma^2_{IFS}\right)} \qquad \chi_m = \chi_m^{(i)} = \chi_m^{(1)} \tag{110}$$

Notice that when $\sigma^2_{\hat{R}_{12}(i,m)} = v^2\sigma^2_{IFS}$, χ_m approaches infinity. This is the point where the interframe spacing variance is sufficiently high such that the radius estimator will not be

less than the required value, regardless of the SNR received at the sensor. Denote this value σ^*_{IFS}.

$$\sigma^*_{IFS} = \frac{\sigma_{\hat{R}_{12}(i,m)}}{v} \tag{111}$$

Figure 34b plots the required SNR as a function of interframe spacing standard deviation. The radius estimate variance is fixed at 0.05 m^2. Taking the PTDoA between the packets which are the furthest apart significantly lowers the required SNR for the same estimator variance. For example, taking the PTDoA between packets 7 and 1 lowers the required SNR by 10 dB over packets 3 and 1 for $\sigma_{IFS} = 0.6$ ns.

It is important to remember the limitations of increasing the PES length i. The ToA estimator variance begins to diverge from the CRLB around $\chi_m = 15$ dB. Therefore, one must be cautious in assessing the performance gains shown by Figure 34b. Essentially, 15 dB is a minimum required SNR bound and increasing the PES length can not lower this bound. Fragmentation can significantly lower the SNR required at the sensor *provided* the devices have sufficiently low IFS timing jitter and the packet SNR remains above 15dB.

4.5.2 MAC-Assisted Position Estimation

Suppose there are N_R sensors which have sufficiently high SNR such that the CRLB applies. It is likely $N_R \geq N_{dec}$, but this is geometry dependent. If PTDoA is performed on the RTS/CTS, then there are N_R independent estimates of R_{12}. The estimates can be combined using the Best Linear Unbiased Estimator (BLUE).

$$\bar{R}_{12} = \sigma_{\bar{R}_{12}} \sum_{m=1}^{N_R} \frac{\hat{R}_{12}(m)}{\sigma^2_{\hat{R}_{12}(m)}} \qquad \mathbb{E}\{\bar{R}_{12}\} = R_{12}, \sigma^2_{\bar{R}_{12}} = \left(\sum_{m=1}^{N_R} \frac{1}{\sigma^2_{\hat{R}_{12}(m)}} \right)^{-1} \tag{112}$$

In Stage III, the maximum and minimum radius of a circle is computed such that the true radius R_{12} is within the bounds with probability δ_R. This is simply a confidence interval computation.

$$R^{LB}_{12} = \bar{R}_{12} - \Phi^{-1}\left(\frac{\delta_R + 1}{2}\right)\sigma_{\bar{R}_{12}} \qquad R^{UB}_{12} = \bar{R}_{12} + \Phi^{-1}\left(\frac{\delta_R + 1}{2}\right)\sigma_{\bar{R}_{12}} \tag{113}$$

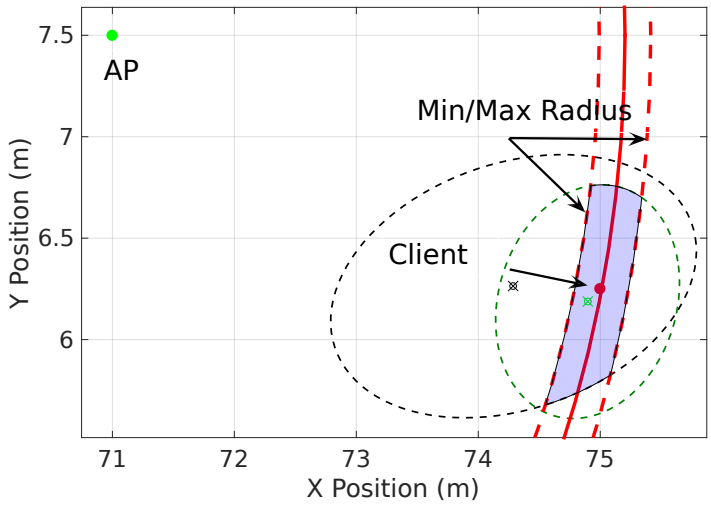


Figure 35: Example simulation result for Stage III. The minimum and maximum radius R_{12}^{LB} and R_{12}^{UB}, respectively, comprising the Stage III confidence region, are shown as dotted red lines. The final intersection of the confidence regions from all three stages, $\mathcal{C}_F$, is shaded in blue.

For Positioning, Equation 113 is used to draw circles centered at E_2 with radii R_{12}^{LB} and R_{12}^{UB}. Define the set of coordinate with these circles.

$$\mathcal{C}_{3L} = \{(x,y) \in \mathbb{R}^2 | x^2 + y^2 \geq R_{12}^{LB}\}, \mathcal{C}_{3U} = \{(x,y) \in \mathbb{R}^2 | x^2 + y^2 \leq R_{12}^{UB}\} \tag{114}$$

Then, the updated position estimate is within the set $\mathcal{C}_3$ with probability based on the confidence level selected.

$$\mathcal{C}_F = (\mathcal{C}_{3U} - \mathcal{C}_{3L}) \cap \mathcal{C}_2' = (\mathcal{C}_{3U} - \mathcal{C}_{3L}) \cap \mathcal{C}_2 \cap \mathcal{C}_1 \tag{115}$$

This is very intuitive when visualized, as shown in Figure 35. The dotted red lines represent $\mathcal{C}_{3U}$ and $\mathcal{C}_{3L}$, dotted black lines Stage I confidence region $\mathcal{C}_1$, and dotted green lines confidence region $\mathcal{C}_2'$. The open dots are position estimates with the colors representing their respective strategies. The red filled dot is the true emitter position. Finally, $\mathcal{C}_3$ is represented by the intersection of all the lines, which is a small sliver containing the true position. The cross-correlation has been normalized by the theoretical mean.

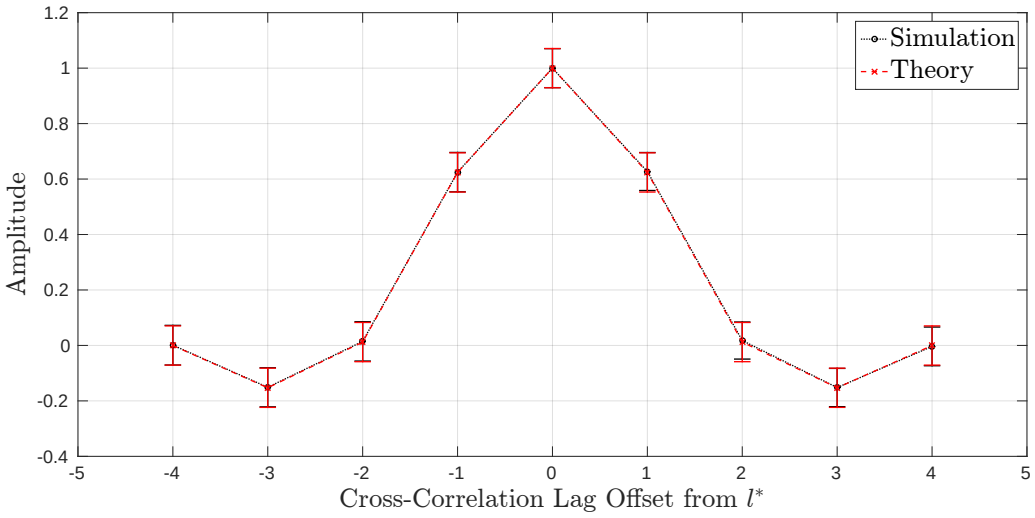


Figure 36: Simulated Cross-Correlation Distribution at 20 dB SNR. The x-axis represents the cross-correlation lag index relative to the true maximum lag l^*. The lines represent the sample and theory mean, respectively. Finally, the error bars represent the sample and theory standard deviation at each respective lag.

4.6 Simulation

4.6.1 Cross-Correlation Distribution

To verify the cross-correlation distribution of Lemma 1 in Section 4.4.2, a 1000 trial simulation was performed using a Binary Phase Shift Key (BPSK) signal with a raised cosine pulse having excess bandwidth parameter $\beta_0 = 0.35$. Two samples per symbol were used, and the pulse was truncated after six symbols. The simulation was performed at 20 dB SNR, with a 9 sample window. That is, $a_m = b_m = 4$. Figure 36 plots the result. The x-axis represents the cross-correlation lag index relative to the true maximum lag l^*. The lines represent the sample and theory mean, respectively. Finally, the error bars represent the sample and theory standard deviation at each respective lag.

4.6.2 Three-Stage Algorithm

Due to the nature of the proposed algorithm, convergence and performance analysis is extremely challenging, therefore simulations were conducted in MATLAB. The M sensors were equally placed on a circle of radius $R_c = 100$ meters. The client position was chosen as $\boldsymbol{p}_1 = \begin{bmatrix} 0.75R_c & 0.0625R_c \end{bmatrix}^T$, and the AP position as $\boldsymbol{p}_2 = \begin{bmatrix} 0.75R_c & 0.4R_c \end{bmatrix}^T$. All sensors

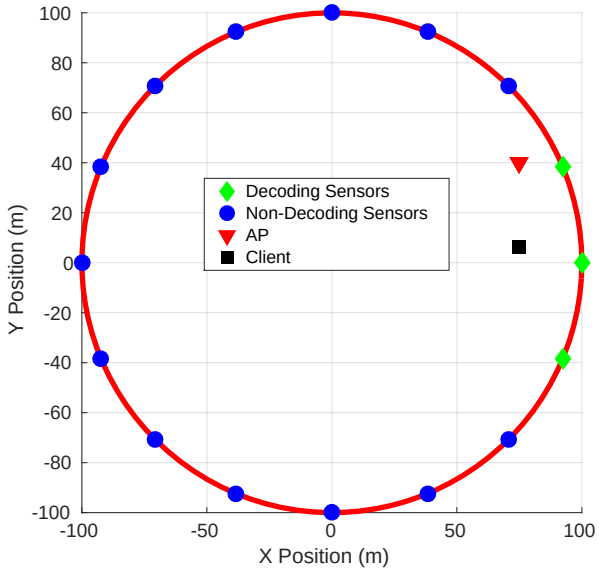


Figure 37: An overview of the simulation geometry. Sensors are placed on a circle with a 100m radius and divided into sets based on whether or not they can decode the transmitted packet. The AP has a known position and the position of the client is to be estimated.

participated in the localization. A decoding sensor was defined to be one with a received SNR above 21 dB. Figure 37 depicts the geometry. Exponential path loss was used for the channel model.

A BPSK signal was chosen with a raised cosine pulse having excess bandwidth parameter $\beta_0 = 0$. This value was chosen so that the spectrum is approximately rectangular. The filter was truncated to 6 symbols. 1024 data symbols were chosen from a PN sequence generated with a linear feedback shift register (LFSR). This was implemented using MATLAB's `comm.PNSequence` function. The signal bandwidth β was set to 20 MHz and the signal was sampled at $T_s = \frac{1}{4\beta}$. Oversampling was necessary to ensure the group delay from the Farrow fractional delay filter was constant as a function of frequency. For all stages, Sinc interpolation was performed by zero-padding the DFT using an upsampling factor of 64 and a window of 64 samples.

In Stage I, three decoding sensors used the maximum lag of the real part of the cross-correlation function. The confidence coefficient for the CR was selected as $\delta_{pos1} = 0.95$. The inverse FIM was used as the covariance matrix. The centroid of the CR was centered at the mean of the position estimates.

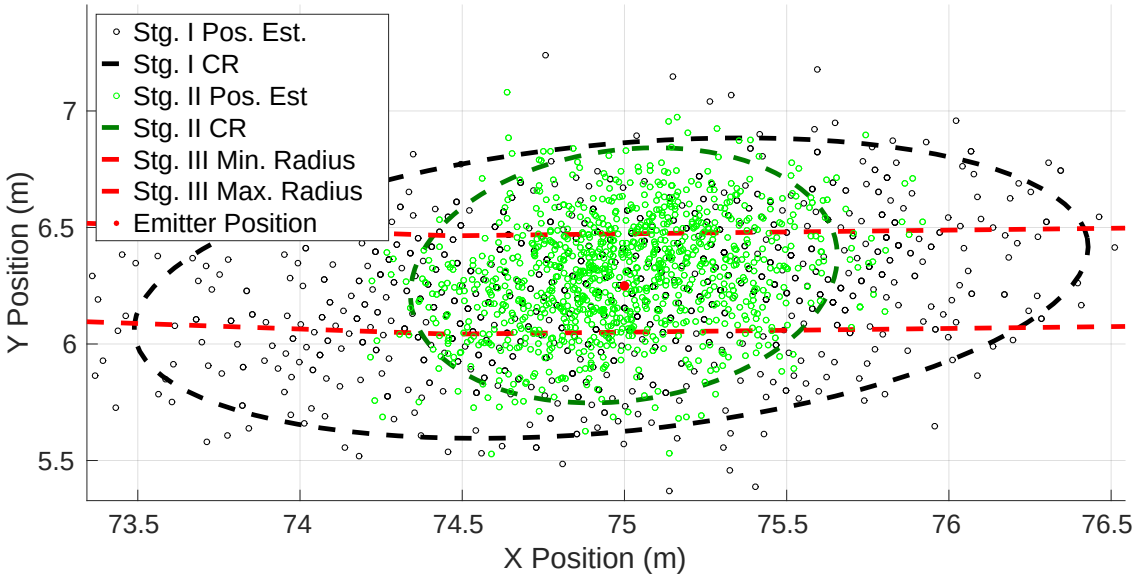
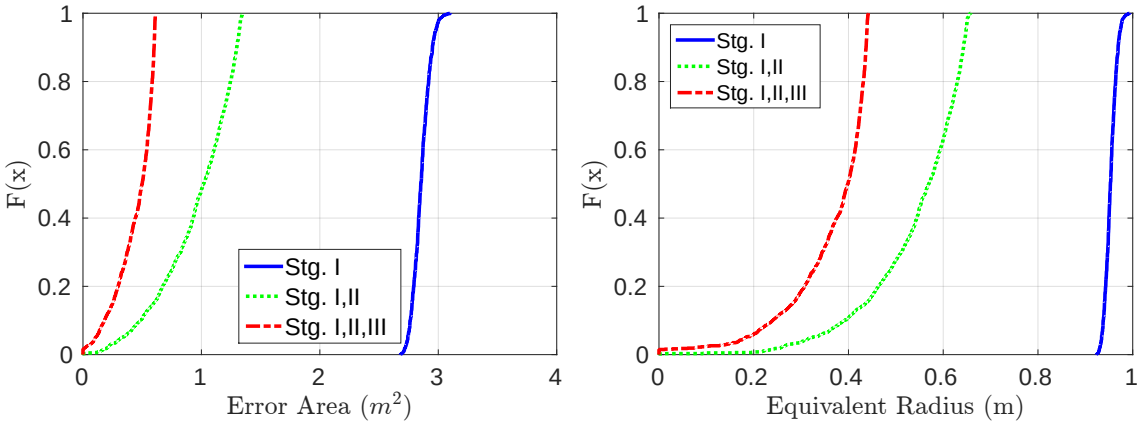


Figure 38: Asymptotic performance simulation for the three-stage localization algorithm. 1000 trials per stage were simulated. The position confidence coefficients were chosen as $\delta_{pos} = 0.95$ for all stages. Confidence regions are depicted with dashed lines. Position estimates are shown as open circles, where the color indicates the stage at which they were made.

In Stage II, the non-decoding sensors use the mean estimate for $\hat{t}_1$ from Stage I. Then, the cross-correlation was windowed. The confidence coefficient for the window was selected as $\delta = 0.99$. Since an integer lag must be selected, the lower and upper bounds N_m^{LB} and N_m^{UB} are rounded to the nearest integer. No fractional delay estimate is performed for these sensors. The ToAs associated with the non-decoding sensors use the uniform distribution assumption to compute the variance of the position estimate. The decoding sensors use the TDoA CRLB. The position was estimated and the CRs constructed with confidence coefficient $\delta_{pos2} = 0.95$. As in Stage I, the centroid of the CR was centered at the mean of the position estimates. In Stage III, no bias was assumed ($b_{IFS} = 0$) for the interframe spacing, since this can easily be estimated and corrected. The maximum integer lag of was selected using the real part of the cross-correlation. The confidence coefficient for the radius estimate was chosen as $\delta_R = 0.95$. Finally, the BLUE was computed from the radius estimates using the CRLB as the estimator variance.



(a) (b)

Figure 39: Simulation results for the three stage algorithm using a single observation of a packet, or packet exchange sequence, per stage. Figure 39a plots the sample Cumulative Distribution Function (CDF) of the error area over 1000 simulations as a function of stage. Figure 39b plots the equivalent radius. No timing jitter was simulated.

4.6.2.1 Asymptotic Performance

To check the asymptotic analysis, 1000 i.i.d. realizations were performed at each stage. Figure 38 illustrates the result. The centroid of the confidence regions, which are ellipses in the first two stages, are very close to the true emitter position as expected. The error ellipses shown are computed using the inverse FIM with a confidence coefficient of 0.95. This simulation helps to verify the asymptotic analysis.

4.6.2.2 Single Observation Performance

The goal of the algorithm is to be fast, precise, and significantly reduce the error area. In this experiment, a single packet, or packet exchange sequence, is generated at each stage. The algorithm is run 1000 times. For each run, the CR area is computed, as well as if the true emitter position is within the CR. Figure 39a demonstrates the reduction in the CR area by plotting the sample CDFs. Figure 39b plots the equivalent CR radius. This is representative of the performance bounds assuming the IFS jitter is very small compared to the ToA estimator variance. No IFS timing jitter was simulated for these plots.

Figure 40 simulates various timing jitters and compares to the Stage II CR error area. This plot demonstrates that Stage III can significantly reduce the area of the CR *provided*

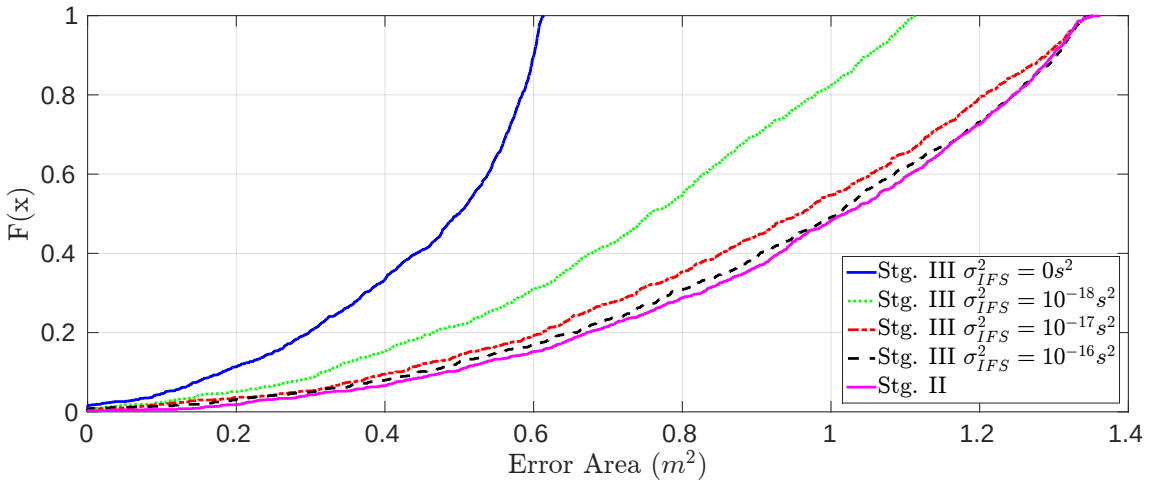


Figure 40: Sample CDF of the confidence region error area for various IFS timing jitters. Stage II error area is shown for comparison.

Table 7: Probability the Emitter Lies Within the Confidence Region for a Confidence Coefficient of 0.95 over 1000 Trials. Variance units are seconds squared.

Stage	$\sigma^2_{IFS} = 0$	$\sigma^2_{IFS} = 10^{-18}$	$\sigma^2_{IFS} = 10^{-17}$	$\sigma^2_{IFS} = 10^{-16}$	Conf. Coeff.
I	0.938	0.934	0.934	0.947	$\delta_{pos1} = 0.95$
II	0.946	0.930	0.922	0.933	$\delta_{pos2} = 0.95$
III	0.937	0.981	0.987	0.991	$\delta_R = 0.95$
I,II	0.888	0.870	0.863	0.883	$\sim 0.95^2 = 0.90$
I,II,III	0.829	0.854	0.852	0.875	$\sim 0.95^3 = 0.86$

the IFS timing jitter is sufficiently small compared to the ToA estimator variance. If the timing jitter is too large, then Stage III does not reduce the area of the CR.

Table 7 provides simulation results to estimate the probability that the true emitter position is within the CR for a confidence coefficient of 0.95. The simulated results agree well with the analysis to within about three points. Furthermore, it can be seen that the probability that the true emitter position is within Stage I and II is approximately δ^2_{pos} for $\delta_{pos1} = \delta_{pos2} = \delta_{pos}$. Similarly, the probability the emitter is within all three CRs is approximately δ^3_{pos} assuming $\delta_{pos3} = \delta_{pos}$. This empirical observation is a starting point for understanding the probability the true emitter position is within the intersection of all three confidence regions.

4.7 Conclusion

In this chapter, a fast and precise three-stage localization algorithm was proposed. Stage I provided an initial confidence region using only sensors able to decode the packet. Non-decodable sensors in Stage II used this information to window the cross-correlation function to prevent large errors in the time delay estimation. Decodable sensors estimated the ToA as in Stage I. Stage III exploited the interframe spacing time specified in the MAC layer to estimate the distance between the emitter of interest and an AP with known position. Asymptotic performance analysis was conducted which guided and inspired the single observation algorithm. Simulation demonstrates the algorithm performs well for a single observation, with a final confidence region equivalent radius of around 0.4 meters with high probability. For our stadium application, this implies that the emitter of interest can be located to within a single seat, instead of two.

Packet Time Difference of Arrival is of greatest benefit when the emitters have highly stable, low jitter clocks over the packet exchange sequence. Although current WLAN standards may not have a sufficiently strict requirement, future standards should consider it to increase client localization accuracy. Besides WLAN, the technique can be useful for other cellular or custom communications protocols where non-collaborative localization is required.

For practical applications, more comprehensive research should be performed to better characterize common WLAN ICs. Specifically, the bias of the IFS with respect to the standard should be studied by manufacturer. The variance of the IFS as a function of time, as well as manufacturer, is also of interest. With this knowledge, the model can be adapted by substitution of b_{IFS} and σ^2_{IFS}. based on manufacturer ID in the MAC layer, to assess theoretical performance.

CHAPTER V

CONCLUSION AND FUTURE DIRECTIONS

Three significant efforts were undertaken in this dissertation research. Chapter 2 detailed two localization testbeds: Laboratory LOC-EED and Stadium LOC-EED. Over 30 TB of complex baseband data has been collected from the three RF sensor nodes deployed in Bobby Dodd Stadium to further research on EED environments. These data will enable algorithm development, algorithm validation, and spectrum characterization, among other research objectives. Packet level correlations known by the MAC layer were used in Chapter 3 to improve data association in EED environments. Notably, it was shown that for a large number of emitters, the probability of correctly associating all measurements in a packet exchange sequence is non-zero using MAC layer side information. Finally, Chapter 4 described a three-strategy localization algorithm exploiting the constrained geometries typically found in EED environments, as well as the packet timing specified by the MAC layer. These contributions significantly further research in this area and suggest a cross-layer approach to localization is necessary in EED environments.

There are many interesting future directions for cross-layer localization in EED environments. This dissertation only explored two pieces of side information: packet level correlations and inter-packet timing, provided by the MAC layer. There is likely other side information which can contribute significantly to improving localization. The difficult part is identifying that information and showing that it is indeed useful. Future research directions should consider other MAC layer information to augment the physical layer measurements required for localization.

In the OSI protocol stack, the MAC layer is only the second of seven distinct layers, including the Network, Transport, and Application Layer, specified by ISO/IEC 7498-1. Additional research should be conducted to investigate how the other layers can improve localization. For example, applications may have unique network traffic characteristics

which are beneficial for localization. In Chapter 4 it was shown analytically that packet fragmentation can improve localization accuracy. Therefore, applications which are likely to fragment packets may be more useful for localization than those which do not. Other uses of the structure provided by these layers should be envisioned.

There is also a need for additional applied research. The first chapter detailed the deployment of a sensor network testbed in Bobby Dodd football stadium. At the time of publication, it is believed this is the first persistent EED testbed to capture complex baseband samples and archive them for future analysis. Other testbeds should be deployed in similar environments for spectrum characterization and data analysis. Additional studies should also be conducted into the implementation of inter-packet timing on commercially available WLAN IC hardware. A few papers exist, but none are comprehensive enough to assess the viability of implementing the third stage of the three-strategy localization algorithm described in Chapter 4.

In closing, this dissertation provides an initial step into both applied and theoretical research involving wireless environments with multiple sensors and multiple emitters. Two algorithms were presented which exploited the MAC layer to improve localization, either using data association or directly shrinking the position estimate error ellipses. A three-sensor testbed has been collecting data the past few football seasons and will continue to provide future researchers additional data for analysis and algorithm validation. With the continued proliferation of emitters such as smartphones, wearables, and cars and a desire to be connected at all times, a cross-layer approach to localization in dense emitter environments will remain relevant.

APPENDIX A

POSITION SOLVER CALCULATIONS

This appendix explains how to estimate emitter position using the method of Chan and Ho [20]. The intent is to clarify the calculation by explaining assumptions and organizing them into a step-by-step process. There are two cases:

1. $M = 3$: Three sensors (Section II-A-1 in [20])
2. $M > 3$: More than three sensors (Section II-A-2 in [20])

In both cases it is assumed the emitter is close. The $M = 3$ case is also derived by Schau [69]. First, the TDoA estimate is transformed to a distance difference.

$$\hat{\boldsymbol{r}} = v\hat{\boldsymbol{\theta}}_{TDoA} = \begin{bmatrix} r_{21} & r_{31} & \dots & r_{M1} \end{bmatrix}^T \in \mathbb{R}^{M-1} \tag{116}$$

A.1 Three Sensors ($M = 3$)

1. Form $\boldsymbol{G}_3$ and $\boldsymbol{g}_3$ using known sensor positions $\boldsymbol{q}_m = \begin{bmatrix} x_m & y_m \end{bmatrix}^T, m = 1, \dots, M$. Compute $\boldsymbol{G}_3^{-1}$ and $\det(\boldsymbol{G}_3)$.

$$\boldsymbol{G}_3 = \begin{bmatrix} x_{21} \triangleq x_2 - x_1 & y_{21} \triangleq y_2 - y_1 \\ x_{31} \triangleq x_3 - x_1 & y_{31} \triangleq y_3 - y_1 \end{bmatrix}, \boldsymbol{g}_3 = \frac{1}{2} \begin{bmatrix} \hat{\boldsymbol{r}}[1]^2 - \|\boldsymbol{q}_2\|_2^2 + \|\boldsymbol{q}_1\|_2^2 \\ \hat{\boldsymbol{r}}[2]^2 - \|\boldsymbol{q}_3\|_2^2 + \|\boldsymbol{q}_1\|_2^2 \end{bmatrix} \tag{117}$$

2. Compute $\overline{x}$ and $\overline{y}$.

$$\overline{x} = \left(y_{21}\hat{\boldsymbol{r}}[2] - y_{31}\hat{\boldsymbol{r}}[1]\right) / \det(\boldsymbol{G}_3), \overline{y} = \left(x_{31}\hat{\boldsymbol{r}}[1] - x_{21}\hat{\boldsymbol{r}}[2]\right) / \det(\boldsymbol{G}_3) \tag{118}$$

3. Compute $\boldsymbol{b} = \boldsymbol{G}_3^{-1}\boldsymbol{g}_3$ and form a quadratic equation in $\tilde{r}_1$. Solve and take the root in the region of interest.

$$a\tilde{r}_1^2 + b\tilde{r}_1 + c = 0, a = \overline{x}^2 + \overline{y}^2 - 1 \tag{119}$$

$$b = -2\left(\overline{x}\left(x_1 + \boldsymbol{b}[1]\right) + \overline{y}\left(y_1 + \boldsymbol{b}[2]\right)\right), c = \|\boldsymbol{q}_1\|_2^2 + \boldsymbol{b}[1]^2 + 2\left(x_1\boldsymbol{b}[1] + y\boldsymbol{b}[2]\right) \tag{120}$$

4. Obtain the position estimate, $\hat{\boldsymbol{p}}_0$.

$$\hat{\boldsymbol{p}}_0 = -\boldsymbol{G}_3^{-1}\left(\tilde{r}_1\hat{\boldsymbol{r}} + \boldsymbol{g}_3\right) \tag{121}$$

A.2 More Than Three Sensors ($M > 3$)

1. Compute $\boldsymbol{G}_a$ and $\boldsymbol{h}$.

$$\boldsymbol{G}_a = -\begin{bmatrix} x_2 - x_1 & y_2 - y_1 & \hat{\boldsymbol{r}}[1] \\ x_3 - x_1 & y_3 - y_1 & \hat{\boldsymbol{r}}[2] \\ \vdots & \vdots & \vdots \\ x_M - x_1 & y_M - y_1 & \hat{\boldsymbol{r}}[M-1] \end{bmatrix}, \boldsymbol{h} = \frac{1}{2}\begin{bmatrix} \hat{\boldsymbol{r}}[1]^2 - \|\boldsymbol{q}_2\|_2^2 + \|\boldsymbol{q}_1\|_2^2 \\ \hat{\boldsymbol{r}}[2]^2 - \|\boldsymbol{q}_3\|_2^2 + \|\boldsymbol{q}_1\|_2^2 \\ \vdots \\ \hat{\boldsymbol{r}}[M-1]^2 - \|\boldsymbol{q}_M\|_2^2 + \|\boldsymbol{q}_1\|_2^2 \end{bmatrix} \tag{122}$$

2. Compute $\boldsymbol{z}_{a1}$ using the Fischer Information Matrix for the TDoA estimate, $\boldsymbol{I}_{TDoA} = \boldsymbol{J}_{TDoA}^{-1}$.

$$\boldsymbol{z}_{a1} = \begin{bmatrix} \hat{x} & \hat{y} & \hat{r}_1 = \|\boldsymbol{q}_1 - \hat{\boldsymbol{p}}\|_2 \end{bmatrix}^T \approx \left(\boldsymbol{G}_a^T\boldsymbol{I}_{TDoA}\boldsymbol{h}\right)\boldsymbol{G}_a^T\boldsymbol{I}_{TDoA}\boldsymbol{h} \tag{123}$$

3. Use $\hat{\boldsymbol{p}} = \begin{bmatrix} \hat{x} & \hat{y} \end{bmatrix}^T$ from $\boldsymbol{z}_{a1}$ to compute $\boldsymbol{B}$. v is the propagation velocity of the signal.

$$\boldsymbol{B} = \text{Diag}\left(\|\boldsymbol{q}_2 - \hat{\boldsymbol{p}}\|_2, \|\boldsymbol{q}_3 - \hat{\boldsymbol{p}}\|_2, \ldots, \|\boldsymbol{q}_M - \hat{\boldsymbol{p}}\|_2\right) \tag{124}$$

4. Compute $\boldsymbol{\Psi}$ and $\boldsymbol{\Psi}^{-1}$ using the TDoA Covariance and $\boldsymbol{B}$.

$$\boldsymbol{\Psi} = v^2\boldsymbol{B}\boldsymbol{J}_{TDoA}\boldsymbol{B} \tag{125}$$

5. Compute $\boldsymbol{z}_{a2} = \begin{bmatrix} \tilde{x} & \tilde{y} & \tilde{r}_1 \end{bmatrix}^T$

$$\boldsymbol{z}_{a2} = \left(\boldsymbol{G}_a^T\boldsymbol{\Psi}^{-1}\boldsymbol{G}_a\right)^{-1}\boldsymbol{G}_a^T\boldsymbol{\Psi}^{-1}\boldsymbol{h} \tag{126}$$

6. Calculate $\boldsymbol{B}'$ using estimates from $\boldsymbol{z}_{a2}$.

$$\boldsymbol{B}' = \text{Diag}\left(\tilde{x} - x_1, \tilde{y} - y_1, \tilde{r}_1\right) \tag{127}$$

7. Calculate $\boldsymbol{\Psi}'$ using $\boldsymbol{G}_a$,$\boldsymbol{\Psi}$, and $\boldsymbol{B}'$ as estimates for their respective true matrices.

$$\boldsymbol{\Psi}' = 4\boldsymbol{B}'\left(\boldsymbol{G}_a^T\boldsymbol{\Psi}^{-1}\boldsymbol{G}_a\right)^{-1}\boldsymbol{B}' \tag{128}$$

8. Construct $\boldsymbol{h}'$ using the estimates from $\boldsymbol{z}_{a1}$ in Equation 123.

$$\boldsymbol{h}' = \begin{bmatrix} (\hat{x} - x_1)^2 & (\hat{y} - y_1)^2 & \hat{r}_1^2 \end{bmatrix}^T \tag{129}$$

9. Calculate $\boldsymbol{z}'_a$.

$$\boldsymbol{z}'_a = \begin{bmatrix} (x - x_1)^2 \\ (y - x_1)^2 \end{bmatrix} = \left(\boldsymbol{G}_a'^T \boldsymbol{\Psi}'^{-1} \boldsymbol{G}'_a\right)^{-1} \boldsymbol{G}_a'^T \boldsymbol{\Psi}'^{-1} \boldsymbol{h}', \boldsymbol{G}'_a = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \end{bmatrix}^T \tag{130}$$

10. The position estimate, $\hat{\boldsymbol{p}}_0$, is computed by solving for x and y in Equation 130. There are four possible solutions; the one in the desired region of interest must be selected. This is a clarification of Equation 24 in Chan and Ho [20].

11. The estimated position covariance matrix of $\hat{\boldsymbol{p}}_0$ can also be calculated.

$$\begin{aligned} \boldsymbol{K}_{pos} &= v^2 \left(\boldsymbol{B}'' \boldsymbol{G}_a'^T \boldsymbol{B}'^{-1} \boldsymbol{G}_a^T \boldsymbol{B}^{-1} \boldsymbol{I}_{TDoA} \boldsymbol{B}^{-1} \boldsymbol{G}_a \boldsymbol{B}'^{-1} \boldsymbol{G}'_a \boldsymbol{B}''\right)^{-1} \\ \boldsymbol{B}'' &= \text{Diag}\left(\hat{\boldsymbol{p}}_0[1] - x_1, \hat{\boldsymbol{p}}_0[2] - y_1\right) \end{aligned} \tag{131}$$

There are two possible positions for $M = 3$, whereas the $M > 3$ case has four. Additionally, the TDoA covariance matrix $\boldsymbol{J}_{TDoA}$ is only required in the $M > 3$ case. While there a quite a number of steps involved in solving for TDoA position, it is a closed-form, non-iterative solution.

APPENDIX B

DISTRIBUTION OF THE CROSS-CORRELATION FUNCTION

B.1 General Distribution

Define $\boldsymbol{w} = \left[\begin{array}{c|c|c} \boldsymbol{w}_1 & \boldsymbol{w}_2 & \boldsymbol{w}_3 \end{array}\right]^T$. Denote the n^{th} elements of vector $\boldsymbol{x}$ as $\boldsymbol{x}[n]$. Define the signal matched filter as $h[n] = s^*[-n]$. The cross-correlation function can be written as

$$\begin{aligned} R[l] = &\sum_{n=0}^{N_{meas}-1} y[n]h[l-n] = \sum_{n=0}^{N_0-1} w[n]h[l-n] \\ &+ \sum_{n=N_0}^{N_0+N_{sig}-1} (s[n]+w[n])\,h[l-n] + \sum_{n=N_0+N_{sig}}^{N_{meas}-1} w[n]h[l-n] \end{aligned} \tag{132}$$

The three different convolutions can be analyzed separately.

$$\begin{aligned} &R_{ys}[l] = K_1[l] + K_2[l] + K_3[l] && K_1[l] \triangleq \sum_{n=0}^{N_0-1} w[n]h[l-n] \\ &K_2[l] \triangleq \sum_{n=N_0}^{N_0+N_{sig}-1} (s[n]+w[n])\,h[l-n] && K_3[l] \triangleq \sum_{n=N_0+N_{sig}}^{N_{meas}-1} w[n]h[l-n] \end{aligned} \tag{133}$$

Next, write the convolutions as matrix-vector multiplies, where $\boldsymbol{k}_1 \in \mathbb{C}^{N_{sig}+N_0-1}$, $\boldsymbol{w}_1 \in \mathbb{C}^{N_0}$, and $\boldsymbol{A}_1$ is a complex matrix of size $(N_{sig}+N_0-1)\text{x}N_0$.

$$\begin{aligned} \boldsymbol{k}_1 &= \begin{bmatrix} K_1[0] & K_1[1] & \dots & K_1[N_{sig}+N_0-2] \end{bmatrix}^T = \boldsymbol{A}_1\boldsymbol{w}_1 \\ &= \begin{bmatrix} h[0] & 0 & 0 & \dots & 0 \\ h[1] & h[0] & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ h[N_0-1] & h[N_0-2] & \dots & h[1] & h[0] \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ h[N_{sig}-1] & h[N_{sig}-2] & \dots & \dots & h[N_{sig}-N_0] \\ 0 & h[N_{sig}-1] & h[N_{sig}-2] & \dots & h[N_{sig}-N_0+1] \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & h[N_{sig}-1] \end{bmatrix} \begin{bmatrix} w[0] \\ w[1] \\ \vdots \\ w[N_0-1] \end{bmatrix} \end{aligned} \tag{134}$$

By linearity, $K_2[l]$ can be separated.

$$K_2[l] = \sum_{n=N_0}^{N_0+N_{sig}-1} s[n]h[m-n] + \sum_{n=N_0}^{N_0+N_{sig}-1} w[n]h[l-n] = K_{2s}[l] + K_{2w}[l] \tag{135}$$

In vector form, $\boldsymbol{k}_2 = \boldsymbol{k}_{2s} + \boldsymbol{k}_{2w}$, where $\boldsymbol{k}_{2s}$ and $\boldsymbol{k}_{2w} \in \mathbb{C}^{2N_{sig}-1}$. $\boldsymbol{A}_2$ is a complex matrix of size $(2N_{sig}-1)$xN_{sig}, and $\boldsymbol{s}$,$\boldsymbol{w}_2 \in \mathbb{C}^{N_{sig}}$.

$$\begin{aligned}
\boldsymbol{k}_{2s} &= \begin{bmatrix} K_{2s}[0] & K_{2s}[1] & \dots & K_{2s}[2N_{sig}-2] \end{bmatrix}^T = \boldsymbol{A}_2\boldsymbol{s} \\
&= \begin{bmatrix}
h[0] & 0 & 0 & \dots & 0 \\
h[1] & h[0] & 0 & \dots & 0 \\
\vdots & \vdots & \vdots & \vdots & \vdots \\
h[N_{sig}-1] & h[N_{sig}-2] & \dots & \dots & h[0] \\
0 & h[N_{sig}-1] & h[N_{sig}-2] & \dots & h[1] \\
0 & 0 & \dots & 0 & h[N_{sig}-1]
\end{bmatrix}
\begin{bmatrix} s[0] \\ s[1] \\ \vdots \\ s[N_{sig}-1] \end{bmatrix} \\
\boldsymbol{k}_{2w} &= \begin{bmatrix} K_{2w}[0] & K_{2w}[1] & \dots & K_{2w}[2N_{sig}-2] \end{bmatrix}^T = \boldsymbol{A}_2\boldsymbol{w}_2 \\
\boldsymbol{w}_2 &= \begin{bmatrix} w[N_0] & w[N_0+1] & \dots & w[N_{sig}+N_0-1] \end{bmatrix}^T
\end{aligned} \tag{136}$$

K_3 in matrix form is straightforward, save for getting the correct indexing. $\boldsymbol{A}_3$ is a complex matrix of size $(N_{meas}-N_0-1)$x$(N_{meas}-N_0-N_{sig})$, $\boldsymbol{k}_3 \in \mathbb{C}^{N_{meas}-N_0-1}$ and

$\boldsymbol{w}_3 \in \mathbb{C}^{N_{meas}-N_0-N_{sig}}$.

$$\boldsymbol{k}_3 = \begin{bmatrix} K_3[0] & K_3[1] & \dots & K_3[N_{meas}-N_0-2] \end{bmatrix}^T = \boldsymbol{A}_3\boldsymbol{w}_3$$

$$= \begin{bmatrix} h[0] & 0 & 0 & \dots & 0 \\ h[1] & h[0] & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ h[N_{meas}-N_0-N_{sig}-1] & \dots & \dots & h[1] & h[0] \\ h[N_{meas}-N_0-N_{sig}] & \dots & \dots & \dots & h[1] \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ h[N_{sig}-1] & h[N_{sig}-2] & \dots & \dots & \dots \\ 0 & h[N_{sig}-1] & h[N_{sig}-2] & \dots & \dots \\ 0 & 0 & \dots & \dots & h[N_{sig}-1] \end{bmatrix} \boldsymbol{w}_3 \quad (137)$$

$$\boldsymbol{w}_3 = \begin{bmatrix} w[N_0+N_{sig}] & w[N_0+N_{sig}+1] & \dots & w[N_{meas}-1] \end{bmatrix}^T \quad (138)$$

Zero-padding the cross-correlation functions yields the composite result. The vectors overlap, so the next step is to write $\boldsymbol{k}$ as a piecewise function.

$$\boldsymbol{k}_{ys} = \begin{bmatrix} \boldsymbol{k}_1 \\ \hline \boldsymbol{0}_{(N_{meas}-N_0)} \end{bmatrix} + \begin{bmatrix} \boldsymbol{0}_{N_0} \\ \hline \boldsymbol{k}_{2s}+\boldsymbol{k}_{2w} \\ \hline \boldsymbol{0}_{(N_{meas}-N_{sig}-N_0)} \end{bmatrix} + \begin{bmatrix} \boldsymbol{0}_{(N_0+N_{sig})} \\ \hline \boldsymbol{k}_3 \end{bmatrix} \quad (139)$$

Let $[\boldsymbol{A}]_a^b$ denote selecting rows a to b of matrix $\boldsymbol{A}$. The newly constructed matrix is denoted

as $\boldsymbol{A}_{i,o}$ if it overlaps with another lag range, or $\boldsymbol{A}_{i,n}$ if it does not. This is essentially overlap-add in vector form.

$$\boldsymbol{k}_{ys} = \begin{cases} \left[\boldsymbol{A}_1\right]_0^{N_0-1} \boldsymbol{w}_1 = \boldsymbol{A}_{1,n}\boldsymbol{w}_1 & 0 \leq l \leq N_0 - 1 \\ \left[\boldsymbol{A}_1\right]_{N_0}^{N_{sig}+N_0-2} \boldsymbol{w}_1 + \left[\boldsymbol{A}_2\right]_0^{N_{sig}-2} (\boldsymbol{s} + \boldsymbol{w}_2) & \\ = \boldsymbol{A}_{1,o}\boldsymbol{w}_1 + \boldsymbol{A}_{2,o} (\boldsymbol{s} + \boldsymbol{w}_2) & N_0 \leq l \leq N_{sig} + N_0 - 2 \\ \left[\boldsymbol{A}_2\right]_{N_{sig}-1}^{N_{sig}-1} (\boldsymbol{s} + \boldsymbol{w}_2) = \boldsymbol{A}_{2,n} (\boldsymbol{s} + \boldsymbol{w}_2) & l = N_{sig} + N_0 - 1 \\ \left[\boldsymbol{A}_2\right]_{N_{sig}}^{2N_{sig}-2} (\boldsymbol{s} + \boldsymbol{w}_2) + \left[\boldsymbol{A}_3\right]_0^{N_{sig}-2} \boldsymbol{w}_3 & \\ = \tilde{\boldsymbol{A}}_{2,0} (\boldsymbol{s} + \boldsymbol{w}_2) + \boldsymbol{A}_{3,o}\boldsymbol{w}_3 & N_{sig} + N_0 \leq l \leq 2N_{sig} + N_0 - 2 \\ \left[\boldsymbol{A}_3\right]_{N_{sig}-1}^{N_{meas}-N_0-2} \boldsymbol{w}_3 = \boldsymbol{A}_{3,n}\boldsymbol{w}_3 & 2N_{sig} + N_0 - 1 \leq l \leq N_{meas} + N_{sig} - 2 \end{cases} \tag{140}$$

For proper complex random vectors, affine transformations have easily derived distributions (pg. 508, [46]). Specifically, if $\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{b}$, where $\boldsymbol{A}$ is a complex full-rank matrix, $\boldsymbol{b}$ is a complex vector, and $\boldsymbol{x} \sim \mathcal{CN}(\boldsymbol{\mu}_x, \boldsymbol{C}_x)$, then $\boldsymbol{y} \sim \mathcal{CN}\left(\boldsymbol{A}\boldsymbol{\mu}_x + \boldsymbol{b}, \boldsymbol{A}\boldsymbol{C}_x\boldsymbol{A}^H\right)$. The notation $\mathcal{CN}(\cdot)$ symbolizes the proper, but not necessary circular, complex Gaussian random vector. More information on working with complex Gaussian random vectors is available [46, 1, 71]. Apply the affine transformation for each range of elements in Equation 140.

$$\boldsymbol{k}_{ys} \sim \begin{cases} \mathcal{CN}\left(\boldsymbol{0}_{N_0}, \boldsymbol{A}_{1,n}\boldsymbol{C}_1\boldsymbol{A}_{1,n}^H\right) & 0 \leq l \leq N_0 - 1 \\ \mathcal{CN}\left(\boldsymbol{A}_{2,o}\boldsymbol{s}, \begin{bmatrix}\boldsymbol{A}_{1,o} & \boldsymbol{A}_{2,o}\end{bmatrix} \begin{bmatrix}\boldsymbol{C}_1 & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{C}_2\end{bmatrix} \begin{bmatrix}\boldsymbol{A}_{1,o}^H \\ \boldsymbol{A}_{2,o}^H\end{bmatrix}\right) & N_0 \leq l \leq N_{sig} + N_0 - 2 \\ \mathcal{CN}\left(\boldsymbol{A}_{2,n}\boldsymbol{s}, \boldsymbol{A}_{2,n}\boldsymbol{C}_2\boldsymbol{A}_{2,n}^H\right) & l = N_{sig} + N_0 - 1 \\ \mathcal{CN}\left(\tilde{\boldsymbol{A}}_{2,o}\boldsymbol{s}, \begin{bmatrix}\tilde{\boldsymbol{A}}_{2,o} & \boldsymbol{A}_{3,o}\end{bmatrix} \begin{bmatrix}\boldsymbol{C}_2 & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{C}_3\end{bmatrix} \begin{bmatrix}\tilde{\boldsymbol{A}}_{2,o}^H \\ \boldsymbol{A}_{3,o}^H\end{bmatrix}\right) & N_{sig} + N_0 \leq l \leq 2N_{sig} + N_0 - 2 \\ \mathcal{CN}\left(\boldsymbol{0}, \boldsymbol{A}_{3,n}\boldsymbol{C}_3, \boldsymbol{A}_{3,n}^H\right) & 2N_{sig} + N_0 - 1 \leq l \leq N_{meas} + N_{sig} - 2 \end{cases}$$

This gives the distribution of the cross-correlation function, where $\boldsymbol{C}_1$,$\boldsymbol{C}_2$, and $\boldsymbol{C}_3$ are the covariance matrices of their respective noise vectors $\boldsymbol{w}_1$,$\boldsymbol{w}_2$, and $\boldsymbol{w}_3$. The autocorrelation function of the signal appears in the mean of the distributions as $\boldsymbol{A}_{2,o}\boldsymbol{s}$, $\boldsymbol{A}_{2,n}\boldsymbol{s}$, and $\tilde{\boldsymbol{A}}_{2,o}\boldsymbol{s}$. The true maximum lag occurs at $l^* = N_{sig} + N_0 - 1$.

B.2 Ideal Cross-Correlation Distribution

The cross-correlation vector $\tilde{\boldsymbol{k}}_{ys}$ for an ideal signal auto-correlation vector $\tilde{\boldsymbol{k}}_{ss}$ and covariance matrices $\boldsymbol{C}_1 = \sigma^2 \boldsymbol{I}_{N_0}$,$\boldsymbol{C}_2 = \sigma^2 \boldsymbol{I}_{N_{sig}}$, and $\boldsymbol{C}_3 = \sigma^2 \boldsymbol{I}_{(N_{meas}-N_{sig}-N_0)}$ is computed in this section.

$$\tilde{\boldsymbol{k}}_{ss} = \begin{bmatrix} h[0] & 0 & 0 & \dots & 0 \\ h[1] & h[0] & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ h[N_{sig}-1] & h[N_{sig}-2] & \dots & \dots & h[0] \\ 0 & h[N_{sig}-1] & h[N_{sig}-2] & \dots & h[1] \\ 0 & 0 & \dots & 0 & h[N_{sig}-1] \end{bmatrix} \boldsymbol{s} = \begin{bmatrix} \boldsymbol{0}_{N_{sig}-1} \\ \mathcal{E}_{sig} \\ \boldsymbol{0}_{N_{sig}-1} \end{bmatrix}$$

$$\boldsymbol{A}_{1,n}\boldsymbol{C}_1\boldsymbol{A}_{1,n}^H = \sigma^2 \,\mathrm{Diag}\left(|h[0]|^2, |h[0]|^2 + |h[1]|^2, \dots, \sum_{n=0}^{N_0-1} |h[n]|^2\right)$$

$$\boldsymbol{A}_{2,o}\boldsymbol{s} = \boldsymbol{0}_{N_{sig}-1}$$

$$\boldsymbol{A}_{1,o}\boldsymbol{C}_1\boldsymbol{A}_{1,o}^H = \sigma^2 \,\mathrm{Diag}\left(\sum_{n=1}^{N_0} |h[n]|^2, \sum_{n=2}^{N_0+1} |h[n]|^2, \dots, |h[N_{sig}-1]|^2\right)$$

$$\boldsymbol{A}_{2,o}\boldsymbol{C}_2\boldsymbol{A}_{2,o}^H = \sigma^2 \,\mathrm{Diag}\left(|h[0]|^2, |h[0]|^2 + |h[1]|^2, \dots, \sum_{n=0}^{N_{sig}-2} |h[n]|^2\right)$$

$$\boldsymbol{A}_{1,o}\boldsymbol{C}_1\boldsymbol{A}_{1,o}^H + \boldsymbol{A}_{2,o}\boldsymbol{C}_2\boldsymbol{A}_{2,o}^H = \sigma^2 \,\mathrm{Diag}\left(\sum_{n=0}^{N_0} |h[n]|^2, \sum_{n=0}^{N_0+1} |h[n]|^2, \dots, \sum_{n=0}^{N_{sig}-1} |h[n]|^2 = \mathcal{E}_{sig}, \dots, \mathcal{E}_{sig}\right)$$

$$\boldsymbol{A}_{2,n}\boldsymbol{s} = \mathcal{E}_{sig}, \boldsymbol{A}_{2,n}\boldsymbol{C}_2\boldsymbol{A}_{2,n}^H = \sigma^2\mathcal{E}_{sig}, \tilde{\boldsymbol{A}}_{2,0}\boldsymbol{s} = \boldsymbol{0}$$

$$\tilde{\boldsymbol{A}}_{2,0}\boldsymbol{C}_2\tilde{\boldsymbol{A}}_{2,0}^H = \sigma^2 \,\mathrm{Diag}\left(\sum_{n=1}^{N_{sig}-1} |h[n]|^2, \sum_{n=2}^{N_{sig}-1} |h[n]|^2, \dots, |h[N_{sig}-1]|^2\right)$$

$$\boldsymbol{A}_{3,o}\boldsymbol{C}_3\boldsymbol{A}_{3,o}^H = \sigma^2 \,\mathrm{Diag}\left(|h[0]|^2, |h[0]|^2 + |h[1]|^2, \dots, \sum_{n=0}^{N_{sig}-2} |h[n]|^2\right)$$

$$\tilde{\boldsymbol{A}}_{2,0}\boldsymbol{C}_2\tilde{\boldsymbol{A}}_{2,0}^H + \boldsymbol{A}_{3,0}\boldsymbol{C}_3\boldsymbol{A}_{3,0}^H = \sigma^2\mathcal{E}_{sig}\boldsymbol{I}$$

$$\boldsymbol{A}_{3,n}\boldsymbol{C}_3\boldsymbol{A}_{3,n}^H = \sigma^2 \,\mathrm{Diag}\left(\sum_{n=0}^{N_{sig}-1} |h[n]|^2, \sum_{n=1}^{N_{sig}-1} |h[n]|^2, \dots, |h[N_{sig}-1]|^2\right) \tag{141}$$

Substitution into the general formula of Lemma 1 yields $\tilde{\boldsymbol{k}}_{ys}$.

$$\tilde{\boldsymbol{k}}_{ys} \sim \begin{cases} 0 \leq l \leq N_0 - 1 \\ \mathcal{CN}\left(\mathbf{0}_{N_0}, \sigma^2 \operatorname{Diag}\left(|h[0]|^2, |h[0]|^2 + |h[1]|^2, \ldots, \sum_{n=0}^{N_0-1} |h[n]|^2\right)\right) \\ N_0 \leq l \leq N_{sig} + N_0 - 2 \\ \mathcal{CN}\left(\mathbf{0}_{N_{sig}-1}, \sigma^2 \operatorname{Diag}\left(\sum_{n=0}^{N_0} |h[n]|^2, \ldots, \sum_{n=0}^{N_{sig}-1} |h[n]|^2 = \mathcal{E}_{sig}, \ldots, \mathcal{E}_{sig}\right)\right) \\ l = N_{sig} + N_0 - 1 \\ \mathcal{CN}\left(\mathcal{E}_{sig}, \sigma^2 \mathcal{E}_{sig}\right) \\ N_{sig} + N_0 \leq l \leq 2N_{sig} + N_0 - 2 \\ \mathcal{CN}\left(\mathbf{0}_{(N_{sig}-1)}, \sigma^2 \mathcal{E}_{sig} \boldsymbol{I}_{N_{sig}-1}\right) \\ 2N_{sig} + N_0 - 1 \leq l \leq N_{meas} + N_{sig} - 2 \\ \mathcal{CN}\left(\mathbf{0}_{(N_{meas}-N_{sig}-N_0)}, \sigma^2 \operatorname{Diag}\left(\sum_{n=0}^{N_{sig}-1} |h[n]|^2, \ldots, |h[N_{sig}-1]|^2\right)\right) \end{cases}$$

APPENDIX C

WINDOWED INTEGER LAG PROBABILITIES

C.1 Probability $l_0 = l^*$ is the Maximum Lag

To simplify notation, define $k \triangleq l - \hat{N}_1$, $k^* \triangleq l^* - \hat{N}_1$, $X_k \triangleq \boldsymbol{g}_{ys}[k]$, and $f_{X_k}(x_k)$ as the PDF of X_k. By the assumption since $l^* \in \mathcal{W}, \exists k | k = k^*$.

$$\mathcal{P}\{l^* = \arg\max_l \boldsymbol{g}_{ys}[l] \mid l, l^* \in \mathcal{W}\} = \mathcal{P}\{\cap_{k=-a_m, k\neq k^*}^{b_m} X_k \leq X_k^*\}$$

The joint PDF can be written using the conditional distribution.

$$f_{\boldsymbol{X}}(\boldsymbol{x}) = f_{X_{-a_m}, X_{-a_m+1}, \ldots X_{b_m}}(x_{-a_m}, x_{-a_m+1}, \ldots, x_{b_m}) = f_{\boldsymbol{X}|X_{k^*}}(\boldsymbol{x} \mid x_{k^*}) f_{X_{k^*}}(x_{k^*})$$

Compute the marginal PDF with respect to X_{k^*}

$$f_{\tilde{\boldsymbol{X}}}(\tilde{\boldsymbol{x}}) = \int_{-\infty}^{\infty} f_{\boldsymbol{X}|X_{k^*}}(\tilde{\boldsymbol{x}}|\alpha) f_{X_{k^*}}(\alpha) d\alpha$$

Assume the conditional amplitude distribution random variables are independent. Once the amplitude value α at lag k^* is assumed, an independence assumption is reasonable with independent noise. The intuition is that the amplitude of lag k shouldn't affect another lag l, unless there is correlation in the noise.

$$= \int_{-\infty}^{\infty} \prod_{k=-a_m, k\neq k^*}^{b_m} f_{X_k}(x_k|\alpha) f_{X_{k^*}}(\alpha) d\alpha$$

Then, compute the CDF.

$$\begin{aligned} F_{\tilde{\boldsymbol{X}}}(\tilde{\boldsymbol{x}}) &= \int_{-\infty}^{\tilde{x}_{-a_m}} \cdots \int_{-\infty}^{\tilde{x}_{b_m}} \int_{-\infty}^{\infty} \prod_{k=-a_m, k\neq k^*}^{b_m} f_{X_k}(x_k|\alpha) f_{X_{k^*}}(\alpha) d\alpha dx_{-a_m} \ldots dx_{b_m} \\ &= \int_{-\infty}^{\infty} \left(\prod_{k=-a_m, k\neq k^*}^{b_m} \int_{-\infty}^{\tilde{x}_k} f_{X_k}(x_k|\alpha) dx_k \right) f_{X_{k^*}}(\alpha) d\alpha \end{aligned}$$

Evaluate the CDFs at the amplitude of lag k^*, $\tilde{\boldsymbol{x}} = \alpha\mathbf{1}$.

$$= \int_{-\infty}^{\infty} \left(\prod_{k=-a_m, k\neq k^*}^{b_m} \int_{-\infty}^{\alpha} f_{X_k}(x_k|\alpha) dx_k \right) f_{X_{k^*}}(\alpha) d\alpha$$
$$= \int_{-\infty}^{\infty} \prod_{k=-a_m, k\neq k^*}^{b_m} \Phi\left(\frac{\sqrt{2}\alpha}{\sqrt{\chi}}\right) f_{X_{k^*}}(\alpha) d\alpha$$
$$= \int_{-\infty}^{\infty} \Phi^{a_m+b_m}\left(\frac{\sqrt{2}\alpha}{\sqrt{\chi}}\right) f_{X_{k^*}}(\alpha) d\alpha = \mathcal{P}\{l^* = \arg\max_l \boldsymbol{g}_{ys}[l] \mid l, l^* \in \mathcal{W}\}$$

The probability distribution function at the maximum lag k^* is given below.

$$f_{X_{k^*}}(\alpha) = \frac{\sqrt{2}}{\sqrt{\pi}\chi} \exp\{-2(\alpha-\chi)^2/\chi^2\}$$

C.2 Probability $l_0 = l^* + k, k \neq 0$ is the Maximum Lag

$$\mathcal{P}\{l_0 = l^* + k = \arg\max_k \boldsymbol{g}_{ys}[l^* + k] \mid l^* \in \mathcal{W}\}, k \in [-a_m, -a_m + 1, \ldots, -1, 1, \ldots, b_m]$$
$$= \mathcal{P}\{\cap_{l=-a_m, l\neq l_0}^{b_m} \boldsymbol{g}_{ys}[\hat{N}_1 + l] \leq \boldsymbol{g}_{ys}[l_0] \mid l^* \in \mathcal{W}\}$$

Define some notation for simplicity. $n \triangleq l - \hat{N}_1$,$n^* \triangleq l^* - \hat{N}_1$, $n_0 \triangleq l_0 - \hat{N}_1$,$X_n \triangleq \boldsymbol{g}_{ys}[n]$, and $f_{X_n}(x_n)$ is the PDF of random variable X_n.

$$= \mathcal{P}\left\{\cap_{n=-a_m, n\neq n_0}^{b_m} X_n \leq X_{n_0}\right\}$$

The joint PDF can be written using the conditional distribution.

$$f_{\boldsymbol{X}}(\boldsymbol{x}) = f_{X_{-a_m}, X_{-a_m+1}, \ldots X_{b_m}}(x_{-a_m}, x_{-a_m+1}, \ldots, x_{b_m})$$
$$= f_{\boldsymbol{X}|X_{k^*}}(\boldsymbol{x} \mid x_{k^*}) f_{X_{k^*}}(x_{k^*})$$

Compute the marginal PDF with respect to X_{n_0}

$$f_{\tilde{\boldsymbol{X}}}(\tilde{\boldsymbol{x}}) = \int_{-\infty}^{\infty} f_{\boldsymbol{X}|X_{k^*}}(\tilde{\boldsymbol{x}}|\alpha) f_{X_{n_0}}(\alpha) d\alpha$$

Assume the conditional amplitude distribution random variables are independent. Once the amplitude value α at lag n_0 is assumed, an independence assumption is reasonable with independent noise. The intuition is that the amplitude of lag n shouldn't affect another lag n_0, unless there is correlation in the noise.

$$\begin{aligned} F_{\tilde{\boldsymbol{X}}}(\tilde{\boldsymbol{x}}) &= \int_{-\infty}^{\tilde{x}_{-a_m}} \cdots \int_{-\infty}^{\tilde{x}_{b_m}} \int_{-\infty}^{\infty} \prod_{n=-a_m, n\neq n_0}^{b_m} f_{X_n}(x_n|\alpha) f_{X_{n_0}}(\alpha) d\alpha dx_{-a_m} \ldots dx_{b_m} \\ &= \int_{-\infty}^{\infty} \left(\prod_{n=-a_m, n\neq n_0, n\neq n^*}^{b_m} \int_{-\infty}^{\tilde{x}_n} f_{X_n}(x_n|\alpha) dx_n \right) \times \\ &\int_{-\infty}^{\tilde{x}_{n^*}} f_{X_{n^*}}(x_{n^*}) dx_{n^*} f_{X_{n_0}}(\alpha) d\alpha \end{aligned}$$

Evaluate the CDFs at the amplitude of lag n_0, $\tilde{\boldsymbol{x}} = \alpha \mathbf{1}$.

$$\begin{aligned} &= \int_{-\infty}^{\infty} \prod_{n=-a_m, n\neq n_0, n\neq n^*}^{b_m} \Phi\left(\frac{\sqrt{2}\alpha}{\sqrt{\chi}}\right) \Phi\left(\frac{\sqrt{2}\,(\alpha-\chi)}{\sqrt{\chi}}\right) f_{X_{n_0}}(\alpha) d\alpha \\ &= \int_{-\infty}^{\infty} \Phi^{a_m+b_m-1}\left(\frac{\sqrt{2}\alpha}{\sqrt{\chi}}\right) \Phi\left(\frac{\sqrt{2}\,(\alpha-\chi)}{\sqrt{\chi}}\right) f_{X_{n_0}}(\alpha) d\alpha \end{aligned}$$

The probability distribution function at the maximum lag n_0 is given below.

$$f_{X_{n_0}}(\alpha) = \frac{\sqrt{2}}{\sqrt{\pi}\chi} \exp\{-2\alpha^2/\chi^2\}$$

APPENDIX D

MEAN AND VARIANCE FOR R.V. L

Suppose L is a random variable representing the probability of choosing lag l in a windowed cross-correlation function. If an ideal impulse auto-correlation function is used, then p and $\tilde{p}$ are the probabilities of selecting the true maximum lag l^*, or another lag, respectively. Without loss of generality, let $l^* = 0$.

$$L(l) \sim \begin{cases} p & l = 0 \\ \tilde{p} & l \neq 0 \end{cases} \qquad l \in [-a, b], a, b \geq 0$$

$$\mathbb{E}\{L\} = \mu_L = \sum_{l=-a}^{b} l p_l = \tilde{p} \left(\sum_{l=-a}^{-1} l + \sum_{l=1}^{b} l \right) = \tilde{p}(b - a)$$

$$\begin{aligned} \mathrm{VAR}\{L\} &= \sum_{l=-a}^{b} l^2 p_l - \mu_L^2 \\ &= \tilde{p} \left(\sum_{l=1}^{a} l^2 + \sum_{l=1}^{b} l^2 \right) - \mu_L^2 = \tilde{p} \left(\frac{a(a+1)(2a+1)}{6} + \frac{b(b+1)(2b+1)}{6} \right) \\ &= \frac{\tilde{p}}{6} \left[a(a+1)(2a+1) + b(b+1)(2b+1) \right] - \tilde{p}^2 (b-a)^2 \end{aligned}$$

APPENDIX E

TIME DELAY ESTIMATION SIMULATIONS

Parabolic, Zero-padded DFT (ZPD), and linear phase sub-sample interpolation on the cross-correlation function $R_m[l]$ were simulated and compared. Symbols were generated using MATLAB's `comm.PNSequence` PN sequence generator and modulated using M-PSK. Then, raised cosine pulse shaping with excess bandwidth parameter β_0 was applied.

The signal was delayed using MATLAB's `fdesign.fracdelay`. The Lagrange Method and a fractional delay filter order of 10 was used. One significant issue in using this filter is that the group delay is not a constant function of frequency, resulting in significant estimator bias. To overcome this issue, the CRLB must be modified to account for oversampling as Equation 73 assumes Nyquist sampling. The resulting equation is given below in units of samples, assuming k_{os} is the oversampling factor and $\tilde{\beta}_{RMS} = T_s\beta_{RMS}$. The SNR is still defined in terms of the signal energy and variance of a single noise sample, $\chi_m \triangleq \mathcal{E}_s/\sigma_w^2$, consistent with the previous definition.

$$\sigma^2_{\hat{N}_0} \geq \frac{1}{8\pi^2\chi_m k_{os}\tilde{\beta}^2_{RMS}} \text{ (samples)}^2 \tag{142}$$

Under the rectangular spectrum assumption, the equation simplifies.

$$\sigma^2_{\hat{N}_0} \geq \frac{3k^2_{os}}{2\pi^2\chi_m} \text{ (samples)}^2 \tag{143}$$

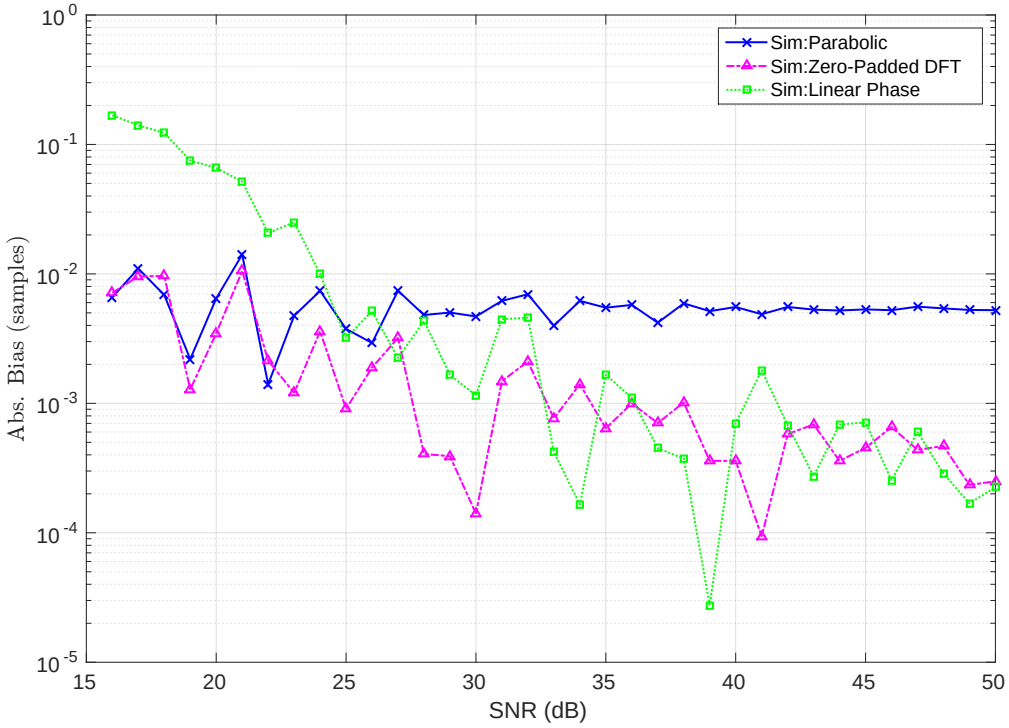
The Parabolic interpolation formula is given by Equation 144. A window of 64 samples around the integer sample maximum was selected for ZPD, with an interpolation factor of 64. Assuming a maximum magnitude integer lag of l^* samples, Equation 145 provides the linear phase ToA estimate, in samples. f_{if} is the non-zero IF frequency (Hz) of the signal, $F_s = \beta k_o s$ is the sampling rate, and $\hat{b}$ is the y-intercept of the phase. This could be calculated using $\hat{b} = \angle\left(R_m[l^*]\right)$, but linearly interpolating the phase of the cross-correlation function around l^* is more accurate. However, care most be taken to window the phase

around the true cross-correlation maximum l^* as the phase is only linear in a small region around this lag.

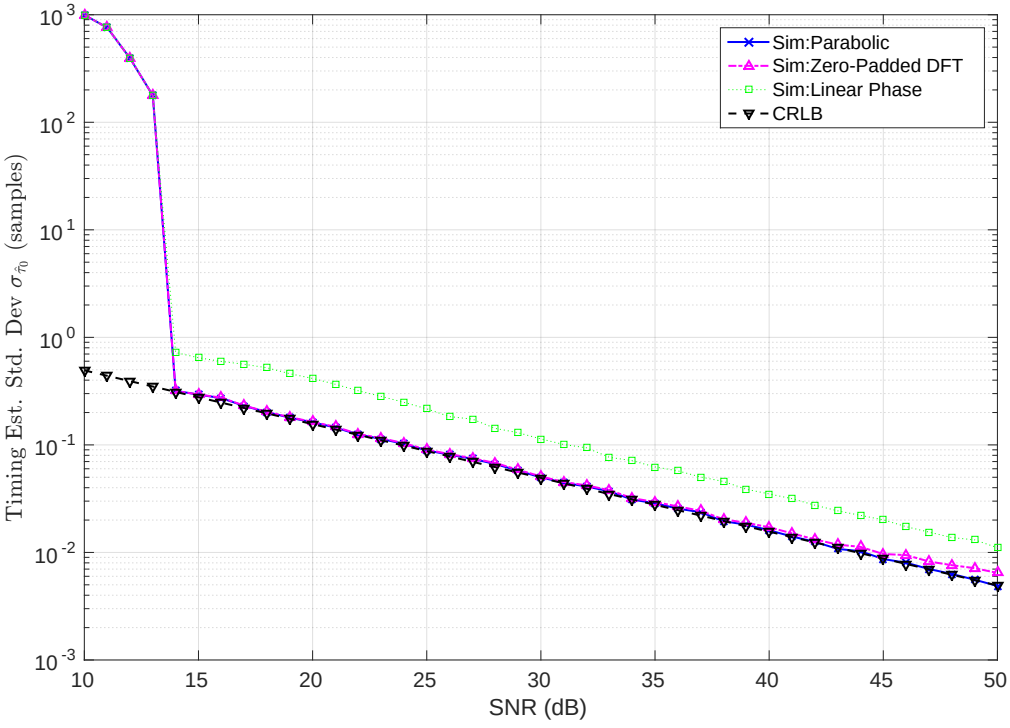
$$\hat{N}_m^{pb} = l_c + l_f, l_c = \arg\max_l \Re\{R_m[l]\}, l_f = \Re\left\{\frac{R_m[l_c+1] - R_m[l_c-1]}{4R_m[l_c] - 2R_m[l_c-1] - 2R_m[l_c+1]}\right\} \quad (144)$$

$$\hat{N}_m^{lp} = |R_m[l^*]| - \frac{F_s\hat{b}}{2\pi f_{if}} \text{ (samples)} \quad (145)$$

For these simulations $\beta_0 = 0$, $N_{sym} = 4096$,$M = 4$, 4 samples/symbol were used, and the pulse shape was truncated to 6 symbols. It is important to note that Equation 143 only applies for small β, since as $\beta \to 1$ the spectrum is no longer well approximated by a rectangle. The linear phase interpolation was performed with phases from lags $[l^*-3, l^*+4]$. The IF frequency was $0.05F_s$. Figure 41a compares the bias of the two estimators, which is ideally zero. It can be seen that parabolic is more biased than ZPD and linear phase interpolation. The parabolic interpolation bias is a function of the sub-sample displacement. Interestingly, the linear phase bias appears to be worse at lower SNRS than ZPD. Figure 41b compares the standard deviation of the estimators with the CRLB given in Equation 73 over 1000 trials for a true time delay of $N_0 = 10.2$ samples. Asymptotically in the number of observations, the SNR at Sensor m, χ_m, must be at least 15 dB to apply the CRLB. Otherwise, it becomes likely the wrong cross-correlation lag will be selected.



(a)



(b)

Figure 41: Time Delay Estimation CRLB Vs. Simulation. Figure 41a illustrates the estimator bias, while Figure 41b compares the estimator standard deviation to the CRLB. The signal was a BPSK-Modulated Pseudorandom Noise bit sequence with a Root-Raised Cosine pulse ($\beta = 0$). At each SNR, 1000 trials were performed. The true delay was 10.2 samples.

REFERENCES

[1] Adali, T., Schreier, P. J., and Scharf, L. L., "Complex-valued signal processing: The proper way to deal with impropriety," *IEEE Transactions on Signal Processing*, vol. 59, pp. 5101–5125, Nov 2011.

[2] Agrez, D., "Weighted multipoint interpolated dft to improve amplitude estimation of multifrequency signal," *IEEE Transactions on Instrumentation and Measurement*, vol. 51, pp. 287–292, Apr 2002.

[3] Akindoyin, A., Willerton, M., and Manikas, A., "Localization and array shape estimation using software defined radio array testbed," in *Sensor Array and Multichannel Signal Processing Workshop (SAM), 2014 IEEE 8th*, pp. 189–192, June 2014.

[4] Ault, A., Krogmeier, J., Dunlop, S., and Coyle, E., "estadium: The mobile wireless football experience," in *Internet and Web Applications and Services, 2008. ICIW '08. Third International Conference on*, pp. 644–649, June 2008.

[5] Ault, A., Zhong, X., and Coyle, E., "K-nearest-neighbor analysis of received signal strength distance estimation across environments," in *Workshop on Wireless Network Measurements (WiNMee 2005), Trentino, Italy, April 3, 2005*, April 2005.

[6] Bahillo, A., Prieto, J., Mazuelas, S., Lorenzo, R. M., Blas, J., and Fernandez, P., "Ieee 802.11 distance estimation based on rts/cts two-frame exchange mechanism," in *VTC Spring 2009 - IEEE 69th Vehicular Technology Conference*, pp. 1–5, April 2009.

[7] Bar-Shalom, Y. and Li, X.-R., *Multitarget-multisensor tracking : principles and techniques*. Storrs, CT : Yaakov Bar-Shalom, 1995.

[8] Benesty, J., Chen, J., and Huang, Y., "Time-delay estimation via linear interpolation and cross correlation," *Speech and Audio Processing, IEEE Transactions on*, vol. 12, pp. 509–519, Sept 2004.

[9] Bhatnagar, V., Ouedraogo, G., Gautier, M., Carer, A., and Sentieys, O., "An fpga software defined radio platform with a high-level synthesis design flow," in *Vehicular Technology Conference (VTC Spring), 2013 IEEE 77th*, pp. 1–5, June 2013.

[10] Bhatti, J., Humphreys, T., and Ledvina, B., "Development and demonstration of a tdoa-based gnss interference signal localization system," in *Position Location and Navigation Symposium (PLANS), 2012 IEEE/ION*, pp. 455–469, April 2012.

[11] Bianchi, G., "Performance analysis of the ieee 802.11 distributed coordination function," *IEEE Journal on Selected Areas in Communications*, vol. 18, pp. 535–547, March 2000.

[12] Bloessl, B., Leitner, C., Dressler, F., and Sommer, C., "A GNU Radio-based IEEE 802.15.4 Testbed," in *12. GI/ITG KuVS Fachgespräch Drahtlose Sensornetze (FGSN 2013)*, (Cottbus, Germany), pp. 37–40, September 2013.

[13] Bloessl, B., Segata, M., Sommer, C., and Dressler, F., "An IEEE 802.11a/g/p OFDM Receiver for GNU Radio," in *ACM SIGCOMM 2013, 2nd ACM SIGCOMM Workshop of Software Radio Implementation Forum (SRIF 2013)*, (Hong Kong, China), pp. 9–16, ACM, August 2013.

[14] Bloessl, B., Segata, M., Sommer, C., and Dressler, F., "An ieee 802.11a/g/p ofdm receiver for gnu radio," in *Proceedings of the Second Workshop on Software Radio Implementation Forum*, SRIF '13, (New York, NY, USA), ACM, 2013.

[15] Boucher, R. and Hassab, J. C., "Analysis of discrete implementation of generalized cross correlator," *Acoustics, Speech and Signal Processing, IEEE Transactions on*, vol. 29, pp. 609–611, Jun 1981.

[16] Bourchas, T., Bednarek, M., Giustiniano, D., and Lenders, V., "Poster abstract: Practical limits of wifi time-of-flight echo techniques," in *IPSN-14 Proceedings of the 13th International Symposium on Information Processing in Sensor Networks*, pp. 273–274, April 2014.

[17] Brik, V., Banerjee, S., Gruteser, M., and Oh, S., "Wireless device identification with radiometric signatures," in *Proceedings of the 14th ACM International Conference on Mobile Computing and Networking*, MobiCom '08, (New York, NY, USA), pp. 116–127, ACM, 2008.

[18] Candore, A., Kocabas, O., and Koushanfar, F., "Robust stable radiometric fingerprinting for wireless devices," in *Hardware-Oriented Security and Trust, 2009. HOST '09. IEEE International Workshop on*, pp. 43–49, July 2009.

[19] Chan, Y., Riley, J., and Plant, J., "Modeling of time delay and its application to estimation of nonstationary delays," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 29, pp. 577–581, Jun 1981.

[20] Chan, Y. T. and Ho, K. C., "A simple and efficient estimator for hyperbolic location," *IEEE Transactions on Signal Processing*, vol. 42, pp. 1905–1915, Aug 1994.

[21] Chen, L., Arambel, P. O., and Mehra, R. K., "Estimation under unknown correlation: covariance intersection revisited," *IEEE Transactions on Automatic Control*, vol. 47, pp. 1879–1882, Nov 2002.

[22] Ciurana, M., Barcelo-Arroyo, F., and Izquierdo, F., "A ranging system with ieee 802.11 data frames," in *2007 IEEE Radio and Wireless Symposium*, pp. 133–136, Jan 2007.

[23] Coyle, E., Allebach, J., and Krueger, J., "The vertically-integrated projects (vip) program: Fully integrating undergraduate education and graduate research," in *Proceedings of the 2006 ASEE Annual Conference and Ex- position*, June 2006.

[24] Danev, B., Luecken, H., Capkun, S., and El Defrawy, K., "Attacks on physical-layer identification," in *Proceedings of the Third ACM Conference on Wireless Network Security*, WiSec '10, (New York, NY, USA), pp. 89–98, ACM, 2010.

[25] Dardaillon, M., Marquet, K., Risset, T., and Scherrer, A., "Software defined radio architecture survey for cognitive testbeds," in *Wireless Communications and*

Mobile Computing Conference (IWCMC), 2012 8th International, pp. 189–194, Aug 2012.

[26] Deb, S., Yeddanapudi, M., Pattipati, K., and Bar-Shalom, Y., "A generalized s-d assignment algorithm for multisensor-multitarget state estimation," *Aerospace and Electronic Systems, IEEE Transactions on*, vol. 33, pp. 523–538, April 1997.

[27] Dolatshahi, S., Polak, A., and Goeckel, D. L., "Identification of wireless users via power amplifier imperfections," in *2010 Conference Record of the Forty Fourth Asilomar Conference on Signals, Systems and Computers*, pp. 1553–1557, Nov 2010.

[28] Ehrman, L. M. and Blair, W. D., "Using target rcs when tracking multiple rayleigh targets," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 46, pp. 701–716, April 2010.

[29] Faria, D. B. and Cheriton, D. R., "Detecting identity-based attacks in wireless networks using signalprints," in *Proceedings of the 5th ACM Workshop on Wireless Security*, WiSe '06, (New York, NY, USA), pp. 43–52, ACM, 2006.

[30] Garver, P. W., Abler, R., and Coyle, E. J., "Theory and development of cross-layer techniques for localization in environments with extreme emitter densities," in *Military Communications Conference, MILCOM 2015 - 2015 IEEE*, pp. 471–476, Oct 2015.

[31] Garver, P. W., Abler, R., Coyle, E. J., and Narayan, J., "Comparisons of high performance software radios with size, weight, area and power constraints," in *Proceedings of the 9th ACM International Workshop on Wireless Network Testbeds, Experimental Evaluation and Characterization*, WiNTECH '14, (New York, NY, USA), pp. 17–24, ACM, 2014.

[32] Giustiniano, D. and Mangold, S., "Caesar: Carrier sense-based ranging in off-the-shelf 802.11 wireless lan," in *Proceedings of the Seventh COnference on Emerging Networking EXperiments and Technologies*, CoNEXT '11, (New York, NY, USA), pp. 10:1–10:12, ACM, 2011.

[33] Glotzbach, R. J., Coyle, E. J., and Bingham, N., "e-stadium: Wireless football infotainment applications," in *ACM SIGGRAPH 2005 Web Program*, SIGGRAPH '05, (New York, NY, USA), ACM, 2005.

[34] GNURadio, "Gnuradio website." `http://www.gnuradio.org/`. accessed 03-August-2016.

[35] Gungor, O. and Koksal, C. E., "On the basic limits of rf-fingerprint-based authentication," *IEEE Transactions on Information Theory*, vol. 62, pp. 4523–4543, Aug 2016.

[36] Günther, A. and Hoene, C., "Measuring round trip times to determine the distance between wlan nodes," in *NETWORKING 2005. Networking Technologies, Services, and Protocols; Performance of Computer and Communication Networks; Mobile and Wireless Communications Systems: 4th International IFIP-TC6 Networking Conference*, 2005.

[37] He, J., Pahlavan, K., Li, S., and Wang, Q., "A testbed for evaluation of the effects of multipath on performance of toa-based indoor geolocation," *IEEE Transactions on Instrumentation and Measurement*, vol. 62, pp. 2237–2247, Aug 2013.

[38] Heurtefeux, K. and Valois, F., "Is rssi a good choice for localization in wireless sensor network?," in *Advanced Information Networking and Applications (AINA), 2012 IEEE 26th International Conference on*, pp. 732–739, March 2012.

[39] Hong, L., Cui, N., Pronobis, M., and Scott, S., "Local motion feature aided ground moving target tracking with gmti and hrr measurements," *IEEE Transactions on Automatic Control*, vol. 50, pp. 127–133, Jan 2005.

[40] Ianniello, J., "Time delay estimation via cross-correlation in the presence of large estimation errors," *Acoustics, Speech and Signal Processing, IEEE Transactions on*, vol. 30, pp. 998–1003, Dec 1982.

[41] IEEE, "Wireless lan medium access control (mac) and physical layer (phy) specifications," 2012.

[42] Jacobsen, E. and Kootsookos, P., "Fast, accurate frequency estimators [dsp tips tricks]," *IEEE Signal Processing Magazine*, vol. 24, pp. 123–125, May 2007.

[43] Jacovitti, G. and Scarano, G., "Discrete time techniques for time delay estimation," *Signal Processing, IEEE Transactions on*, vol. 41, pp. 525–533, Feb 1993.

[44] Judd, G. and Steenkiste, P., "A software architecture for physical layer wireless network emulation," in *Proceedings of the 1st International Workshop on Wireless Network Testbeds, Experimental Evaluation & Characterization*, WiNTECH '06, (New York, NY, USA), pp. 2–9, ACM, 2006.

[45] Julier, S. J. and Uhlmann, J. K., "A non-divergent estimation algorithm in the presence of unknown correlations," in *Proceedings of the 1997 American Control Conference (Cat. No.97CH36041)*, vol. 4, pp. 2369–2373 vol.4, Jun 1997.

[46] Kay, S. M., *Fundamentals of statistical signal processing.* Englewood Cliffs, N.J. : Prentice-Hall PTR, 1993.

[47] Lacage, M. and Henderson, T. R., "Yet another network simulator," in *Proceeding from the 2006 Workshop on Ns-2: The IP Network Simulator*, WNS2 '06, (New York, NY, USA), ACM, 2006.

[48] Langley, L. E., "Specific emitter identification (sei) and classical parameter fusion technology," in *WESCON/'93. Conference Record,*, pp. 377–381, Sep 1993.

[49] Le, T. N., Chin, W. L., and Kao, W. C., "Cross-layer design for primary user emulation attacks detection in mobile cognitive radio networks," *IEEE Communications Letters*, vol. 19, pp. 799–802, May 2015.

[50] Lerro, D. and Bar-Shalom, Y., "Interacting multiple model tracking with target amplitude feature," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 29, pp. 494–509, Apr 1993.

[51] Lin, Y., Lee, H., Woh, M., Harel, Y., Mahlke, S., Mudge, T., Chakrabarti, C., and Flautner, K., "Soda: A low-power architecture for software radio," in *Computer Architecture, 2006. ISCA '06. 33rd International Symposium on*, pp. 89–101, 2006.

[52] Marple, S. L., "Estimating group delay and phase delay via discrete-time analytic cross-correlation," *IEEE Transactions on Signal Processing*, vol. 47, pp. 2604–2607, Sep 1999.

[53] Melvin, W. L. and Scheer, J. A., *Principles of Modern Radar, Volume 2 - Advanced Techniques*. Institution of Engineering and Technology, 2013.

[54] Miljanic, Z., Seskar, I., Le, K., and Raychaudhuri, D., "The winlab network centric cognitive radio hardware platform - winc2r," in *Cognitive Radio Oriented Wireless Networks and Communications, 2007. CrownCom 2007. 2nd International Conference on*, pp. 155–160, Aug 2007.

[55] Mitola, J., I., "Software radios-survey, critical evaluation and future directions," in *Telesystems Conference, 1992. NTC-92., National*, pp. 13/15–13/23, May 1992.

[56] Moddemeijer, R., "On the determination of the position of extrema of sampled correlators," *IEEE Transactions on Signal Processing*, vol. 39, pp. 216–219, Jan 1991.

[57] Nafarieh, A. and Ilow, J., "A testbed for localizing wireless lan devices using received signal strength," in *Communication Networks and Services Research Conference, 2008. CNSR 2008. 6th Annual*, pp. 481–487, May 2008.

[58] "nmon for linux." http://nmon.sourceforge.net/pmwiki.php. (Visited on 05/21/2014).

[59] Papanastasiou, S., Mittag, J., Strom, E. G., and Hartenstein, H., "Bridging the gap between physical layer emulation and network simulation," in *2010 IEEE Wireless Communication and Networking Conference*, pp. 1–6, April 2010.

[60] Papoulis, Athanasios, P. U. S., *Probability, Random Variables, and Stochastic Processes*. McGraw-Hill, 4 ed., 2002.

[61] Patwari, N. and Kasera, S. K., "Robust location distinction using temporal link signatures," in *Proceedings of the 13th Annual ACM International Conference on Mobile Computing and Networking*, MobiCom '07, (New York, NY, USA), pp. 111–122, ACM, 2007.

[62] Polak, A. C., Dolatshahi, S., and Goeckel, D. L., "Identifying wireless users via transmitter imperfections," *IEEE Journal on Selected Areas in Communications*, vol. 29, pp. 1469–1479, August 2011.

[63] Polak, A. C. and Goeckel, D. L., "Identification of wireless devices of users who actively fake their rf fingerprints with artificial data distortion," *IEEE Transactions on Wireless Communications*, vol. 14, pp. 5889–5899, Nov 2015.

[64] Proakis, J. G. and Salehi, M., *Digital Communications*. McGraw-Hill, 2008.

[65] Rasmussen, K. B. and Capkun, S., "Implications of radio fingerprinting on the security of sensor networks," in *Security and Privacy in Communications Networks and the Workshops, 2007. SecureComm 2007. Third International Conference on*, pp. 331–340, Sept 2007.

[66] Richards, M. A., *Fundamentals of Radar Signal Processing.* McGraw-Hill Education, 2014.

[67] Sathyan, T., Sinha, A., and Kirubarajan, T., "Passive geolocation and tracking of an unknown number of emitters," *Aerospace and Electronic Systems, IEEE Transactions on*, vol. 42, pp. 740–750, April 2006.

[68] Sathyan, T., Sinha, A., Kirubarajan, T., McDonald, M., and Lang, T., "Mda-based data association with prior track information for passive multitarget tracking," *Aerospace and Electronic Systems, IEEE Transactions on*, vol. 47, pp. 539–556, January 2011.

[69] Schau, H. and Robinson, A., "Passive source localization employing intersecting spherical surfaces from time-of-arrival differences," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 35, pp. 1223–1225, Aug 1987.

[70] Schauer, L., Dorfmeister, F., and Maier, M., "Potentials and limitations of wifi-positioning using time-of-flight," in *International Conference on Indoor Positioning and Indoor Navigation*, pp. 1–9, Oct 2013.

[71] Schreier, P. J. and Scharf, L. L., *Statistical Signal Processing of Complex-valued Data : The Theory of Improper and Noncircular Signals.* Cambridge University Press, 2010.

[72] Schwalowsky, S., Trsek, H., Exel, R., and Kero, N., "System integration of an ieee 802.11 based tdoa localization system," in *Precision Clock Synchronization for Measurement Control and Communication (ISPCS), 2010 International IEEE Symposium on*, pp. 55–60, Sept 2010.

[73] Senel, M., Chintalapudi, K., Lal, D., Keshavarzian, A., and Coyle, E., "A kalman filter based link quality estimation scheme for wireless sensor networks," in *Global Telecommunications Conference, 2007. GLOBECOM '07. IEEE*, pp. 875–880, Nov 2007.

[74] Sheng, Y., Tan, K., Chen, G., Kotz, D., and Campbell, A., "Detecting 802.11 mac layer spoofing using received signal strength," in *INFOCOM 2008. The 27th Conference on Computer Communications. IEEE*, April 2008.

[75] Simon, M. K. and Alouini, M.-S., *Digital Communications over Fading Channels.* Hoboken: Wiley, 2005.

[76] Stuber, G. L., *Principles of mobile communication.* New York: Springer, 2011.

[77] Talbot, K. I., Duley, P. R., and H., H. M., "Specific emitter identification and verification," in *Technology Review Journal*, pp. 113–133, 2003.

[78] Tan, K., Liu, H., Zhang, J., Zhang, Y., Fang, J., and Voelker, G. M., "Sora: High-performance software radio using general-purpose multi-core processors," *Commun. ACM*, vol. 54, pp. 99–107, Jan. 2011.

[79] Tomko, A. A., Rieser, C. J., and Buell, L. H., "Physical-layer intrusion detection in wireless networks," in *MILCOM 2006 - 2006 IEEE Military Communications conference*, pp. 1–7, Oct 2006.

[80] Ureten, O. and Serinken, N., "Detection of radio transmitter turn-on transients," *Electronics Letters*, vol. 35, pp. 1996–1997, Nov 1999.

[81] "Ettus research." http://www.ettus.com. (Visited on 05/05/2014).

[82] V. Kapnadak, M. S. and Coyle, E., "Distributed iterative quantization for interference characterization in wireless networks," in *Digital Signal Processing, Vol. 22 No. 1*, pp. 96–105, Jan 2012.

[83] Viola, F. and Walker, W., "A spline-based algorithm for continuous time-delay estimation using sampled data," *Ultrasonics, Ferroelectrics, and Frequency Control, IEEE Transactions on*, vol. 52, pp. 80–93, Jan 2005.

[84] Vo-Huu, T. D., Vo-Huu, T. D., and Noubir, G., "Fingerprinting wi-fi devices using software defined radios," in *Proceedings of the 9th ACM Conference on Security & Privacy in Wireless and Mobile Networks*, WiSec '16, (New York, NY, USA), pp. 3–14, ACM, 2016.

[85] "Warp project - wireless open-access research platform." http://warp.rice.edu/trac/wiki/about. (Visited on 05/01/2014).

[86] Weiss, A. J., "Composite bound on arrival time estimation errors," *IEEE Transactions on Aerospace and Electronic Systems*, vol. AES-22, pp. 751–756, Nov 1986.

[87] Woh, M., Lin, Y., Seo, S., Mahlke, S., Mudge, T., Chakrabarti, C., Bruce, R., Kershaw, D., Reid, A., Wilder, M., and Flautner, K., "From soda to scotch: The evolution of a wireless baseband processor," in *Microarchitecture, 2008. MICRO-41. 2008 41st IEEE/ACM International Symposium on*, pp. 152–163, Nov 2008.

[88] Yuan, H. L. and Hu, A. Q., "Preamble-based detection of wi-fi transmitter rf fingerprints," *Electronics Letters*, vol. 46, pp. 1165–1167, August 2010.

[89] Zheng, F. and Ebbini, E., "A subsample estimator based on zero phase crossing in ultrasound," in *2012 IEEE International Ultrasonics Symposium*, pp. 1698–1701, Oct 2012.

[90] Zhong, X. and Coyle, E. J., "estadium: a wireless "living lab" for safety and infotainment applications," in *2006 First International Conference on Communications and Networking in China*, pp. 1–6, Oct 2006.

[91] Zhong, X., Chan, H.-H., Rogers, T. J., Rosenberg, C. P., and Coyle, E. J., "The development and estadium testbeds for research and development of wireless services for large-scale sports venues," in *2nd International Conference on Testbeds and Research Infrastructures for the Development of Networks and Communities, 2006. TRIDENTCOM 2006.*, 2006.

[92] Ziv, J. and Zakai, M., "Some lower bounds on signal parameter estimation," *IEEE Transactions on Information Theory*, vol. 15, pp. 386–391, May 1969.