RELATIVE POSITIONING, NETWORK FORMATION, AND ROUTING IN ROBOTIC WIRELESS NETWORKS

by

Pradipta Ghosh

A Dissertation Presented to the
FACULTY OF THE USC GRADUATE SCHOOL
UNIVERSITY OF SOUTHERN CALIFORNIA
In Partial Fulfillment of the
Requirements for the Degree
DOCTOR OF PHILOSOPHY
(ELECTRICAL ENGINEERING)

August 2018

Dedication

To my beloved family and friends.

Acknowledgment

I would like to express my sincere gratitude to many significant individuals for their continuous support, help, and contributions to my Ph.D. studies.

It is with immense gratitude that I acknowledge the support and help of my advisor, Professor Bhaskar Krishnamachari. He is an exemplary advisor that a Ph.D. student can only wish for. He guided me through all stages of my Ph.D., my success, my failures, and my ups and downs; with equal patience and consideration. He has been very welcoming towards investigating any interesting promenades both in academic research and life beyond school. He motivated me to pay equal attention to work and personal life without neglecting either. His immense dedication towards research and teaching has always been inspirational to me. I also want to thank him for having faith in me on numerous occasions when I lacked the confidence and courage. I cannot imagine going through the Ph.D. without having him as my guide, mentor, and teacher – a true advisor. He is and always will be someone I look up to as a mentor, researcher, teacher, and above all a kind, patient, admirable person.

Besides my advisor, I would like to thank Professor Nora Ayanian for her invaluable guidance and advice on many aspects of my thesis works. She has helped me on numerous occasions to clear doubts regarding robotics related problems. I would also like to thank Professor Murali Annavaram and Professor Ramesh Govindan for their guidance and advice on academia as well as their insightful comments and contributions to my thesis. I would also like to thank Professor Andrea Gasparri for his guidance and advice on a significant portion of my thesis research. He has been one of the persons who helped me determine my course of actions towards kick-starting my thesis research.

My sincere thanks also go to other knowledgeable researchers: ANRG members and Dr. Ranjan Pal. First of all, I want to thank my fellow ANRG member, colleague and dear friend, Jason A. Tran for all of his hard work and contributions towards my thesis research as well as for his invaluable support and advice. It was not possible to conceive some parts of my thesis research without his significant contributions and support. I also had a valuable experience in our joint effort towards mentoring undergraduate and masters students. His organizational and coding skills are something I always look up to. I also want to acknowledge all other ANRG members for all the academic as well as fun non-academic discussions. I will cherish every moment I had with you over past couple of years. I really admire and love the culture we have at ANRG where everyone interacts and bonds like a big

family. I would also like to acknowledge Dr. Ranjan Pal for his valuable opinions and support throughout my Ph.D.

In addition, I want to extend my sincere thanks to my friends both at USC and outside USC. I want to particularly thank my very close friend, Pratik Shah, for being the friend with whom I can talk about almost everything, for always having my back, for being such a good listener, and always giving me great advises. I also want to thank my other close friends and roommates: Md. Nasir, Arindam Jati, Subhayan De, Vishnu Ratnam, and Kosha Talati Shah. I also want to thank Amrita Kundu for being there in my tough times and supporting me constantly. I am truly blessed to have such a group of wonderful people as my friend whom I consider as a part of my family.

The works in this dissertation were supported in part by funding from NSF and DARPA. I would also like to express my appreciation to the Provost Fellowship Program and the Ming Hsieh Department of Electrical Engineering for supporting my tuition and offering stipends during my Ph.D. studies. I would also like to extend my gratitude to the Ming Hsieh Institute and the USC Graduate Student Government for supporting my conference travel expenses.

Lastly but most importantly, I want to dedicate this thesis to my family: my parents, Dr. Pijush Kanti Ghosh and Dr. Chhabi Ghosh, and my brother, Dr. Prattay Ghosh. Life is very uncertain and short. The family is the most important thing to me. Thanks for supporting me and helping me become who I am today.

Without your support, it would not have been possible for me to come this far let alone complete my Ph.D. studies. I love you.

Table of Contents

List Of Figures

List Of Tables

Abstract

Robotic Wireless Networks (RWN) is one of the cutting-edge domain of research that focuses on a range of collaborative autonomous operations such as exploration of an unknown terrain, fire-fighting, temporary wireless communication backbone deployments, and extending existing communication infrastructures. Such application contexts with a group of robots impose a diverse set of stringent requirements to the system designers that include but not limited to efficient infrastructure-less localization, positioning, movement control, connectivity maintenance, and lightweight communication protocols. In this thesis, we identify five key problems in the field of RWN and provide the necessary solutions to meet the end-goal of building a scalable, self-sustained, energy efficient, and controllable RWN system.

The first research problem that we focus on is related to the localization of robots in any random deployment arena. A real-world deployment of an RWN relies on the availability of a scalable localization system in the application arena for effective operation. While GPS can provide sufficiently accurate positioning in open outdoor setting, effective operations of an RWN demands an alternative and sufficiently precise relative or absolute localization scheme for an indoor cluttered

setting where GPS based global positioning is very flaky and unreliable. Most of the existing RF based alternative indoor localization schemes rely on the presence of an infrastructure for trilateration based localization which, thereafter, restricts the RWN application domain. To this end, we propose Autonomous RSSI based RElative poSitioning and Tracking (ARREST), an RSSI based relative localization and proximity maintenance system for RWN that does not require any pre-deployed infrastructure. To demonstrate the practicality as well as analyze the performance, we have developed a real prototype and performed extensive experimentation.

The second key problem that we look into is related to the application of an RWN to support a temporary communication backbone with certain communication guarantees in terms of throughput, loss rate, and latency in a dynamic environment. It is often presumed in such contexts that a "sufficient" number of robots are present in the network. However, to our knowledge, none of the existing works has addressed this crucial RWN system parameter. In our second study, we first derive an upper bound on the spacing between any transmitter-receiver pair by exploiting the properties of Carrier Sense Multiple Access (CSMA) and thereafter, map this bound to a lower bound on the number of robots to deploy.

In the course of our third study, we look into the classic problem of routing but with the sole focus on fulfilling the RWN specific requirements in terms of efficiency, reliability, timeliness, and scalability. Only a few existing solutions can fulfill all these requirements (mainly the delay requirement) imposed upon the networking

protocol stack of an RWN. To this end, we propose the Heat Diffusion Collection Protocol(HDCP), a backpressure based routing protocol that uses the classic equations related to the heat flow from a highly heated region to a less heated region towards queue based dynamic packet routing. Through an extensive set of real-world experiments, we demonstrate that the HDCP algorithm can guarantee lower delay and higher throughput compared to the existing contenders under a diverse set of conditions.

The robots in an RWN also inherently work under limited communication bandwidth and heavy power constraints which in turn put constraints on the communications hardware and protocols. Therefore, in our fourth study, we concentrate on the design of the Robotic Overlay coMmunicAtioN prOtocol (ROMANO), an efficient lightweight application layer communication protocol with low bandwidth and energy requirement for sensing data collection and control. ROMANO is overlaid on top of the well-known Message Queuing Telemetry Transport for Sensor Nodes (MQTT-SN) publish-subscribe protocol to provide a simple and unified abstraction of control and sensing data communication. Through a set of real-world experiments with real prototypes, we demonstrate different features of ROMANO as well as analyze the performance.

In the fifth and final study, we look into a problem of passive RF mapping by employing the hardware developed for ARREST system experimentation. The goal of this study is to exploit a received directional RSSI pattern from an omnidirectional

transmitter towards passive localization of unknown RF reflecting surfaces/objects in an unknown environment. To this end, we propose the concept of miniRadar which is a low power IEEE 802.15.4 based bi-static radar type system for RWN.

In summary, we have developed systems and algorithmic solutions for an RWN that address five cutting-edge problems in an RWN. The proposed solutions and systems will guide the design of an integrated real-world RWN system with self-sustained localization and relative position control, efficient routing with lower delay, simple abstraction of control and communication, and passive RF sensing.

Chapter 1

Introduction

Robotics has been a very important and active field of research over the last couple of decades with the main focus on seamless integration of robots in human lives to assist and to help human in difficult, cumbersome jobs such as search and rescue in disastrous environments [3, 4]. Recent advancements in affordable technology and hardware miniaturization have partly materialized this goal. Following this trend, researchers have been motivated to look into the collaborative aspects where a group of robots can work in synergy to perform a set of diverse tasks such as collaboratively moving an object [5, 6]. To this end, networked/swarm robotics [5], robot augmented wireless communication backbones [7, 8], and robotic wireless networks [9, 10] has become cutting-edge fields of research. The controllability of the wireless nodes has opened up a whole new dimension to the networks and wireless communication researchers. The applicability of such systems includes but not limited to fire fighting [3, 11], disaster management [12, 13], search and rescue missions to sense environments with limited visibility [4, 14, 15], and smart home

environments [16]. This cutting-edge research domain is called by many different names such as *"Wireless Robotic Networks"*, *"Robotic Wireless Sensor Networks"*, *"Wireless Automated Networks"*, and *"Networked Robots"*. In this thesis, we will refer to this field as *"Robotic Wireless Networks (RWN)"*.

This new area of research has posed a plethora of research questions related to localization, coordination, consensus, maintaining link qualities and connectivity, and distributed decision making and control. To design a fully functional RWN system, one needs to take each of these questions into consideration. In the following section, we provide a brief overview of these research problems along with a formal categorization. Moreover, we provide sufficient background to understand the novelty and importance of the works presented in this thesis.

1.1 Robotic Wireless Networks

First of all, let us define and illustrate the field of Robotic Wireless Network (RWN). According to the IEEE Society of Robotics and Automation's Technical Committee [17]: *"A 'networked robot' is a robotic device connected to a communications network such as the Internet or LAN. The network could be wired or wireless and based on any of a variety of protocols such as TCP, UDP, or 802.11. Many new applications are now being developed ranging from automation to exploration. There are two subclasses of Networked Robots: (1) Tele-operated, where human supervisors send commands and receive feedback via the network. Such systems support*

research, education, and public awareness by making valuable resources accessible to broad audiences; (2) Autonomous, where robots and sensors exchange data via the network. In such systems, the sensor network extends the effective sensing range of the robots, allowing them to communicate with each other over long distances to coordinate their activity. The robots, in turn, can deploy, repair, and maintain the sensor network to increase its longevity, and utility. A broad challenge is to develop a science base that couples communication to control to enable such new capabilities." We define an *RWN* as an autonomous networked multi-robot system that aims to achieve sensing and mainly communication-related *goals* while fulfilling certain *communication performance requirements* via cooperative control, learning, and adaptation.

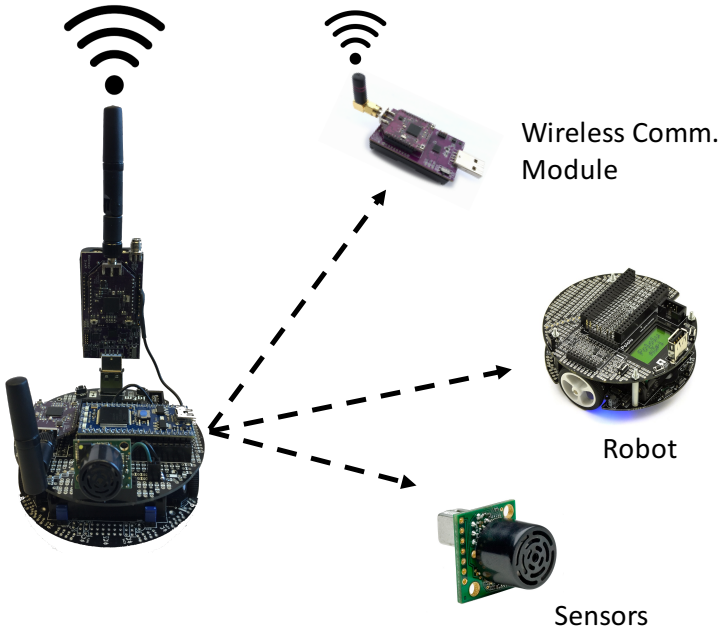


Figure 1.1: Illustration of a Robotic Wireless Router

Ideally, each node of a typical RWN should have controlled mobility, a set of sensors, and wireless communication capabilities (as illustrated in Figure 1.1). We refer to such nodes (devices) as *"Robotic Wireless Routers"*. Nonetheless, an RWN can also have some nodes with just sensing/control and wireless communication capabilities but without mobilities. Note that, every node of an RWN must have wireless communication capabilities. Moreover, an RWN is typically expected to be able to fulfill or guarantee certain communication performance requirements enforced by the application contexts such as minimum achievable bit error rate (BER) in every link of the network.

The existing research works related to RWN can be subdivided into two broader genres. The first genre focuses on generic multi-robot sensing systems with realistic communication channels (i.e., including the effects of fading, shadowing etc.) between the robots. *To clarify, these are mostly the existing works in the robotics literature on multi-robot systems but with practical wireless communication and networking models.* One application context of such an RWN is in robot-assisted firefighting where the robots are tasked to sense the unknown environments inside rubble to help the firefighters navigate. Now, if the robots are not able to maintain a good connectivity among themselves or to a mission control station, the whole mission is voided. Refer to Figure 1.2 for an illustration of such contexts where a group of robots is sensing an unknown environment to guide the human movements. In Figure 1.2, the network consisting of five robots and two firefighters needs to

be connected all the time and also needs to have properties such as reliability and lower packet delays. To make the problem more complex, the robots might also need to continually maintain certain proximity to the firefighters. Thus, we need a class of multi-objective motion control algorithms that will optimize the sensing and exploration task performance, and will also ensure the connectivity, proximity, and the performance of the network. Some of the main identifiable keywords in this genre of works are connectivity maintenance, efficient routing to reduce end to end delay of packets, proximity maintenance, localization, and multi-objective motion control and optimization.

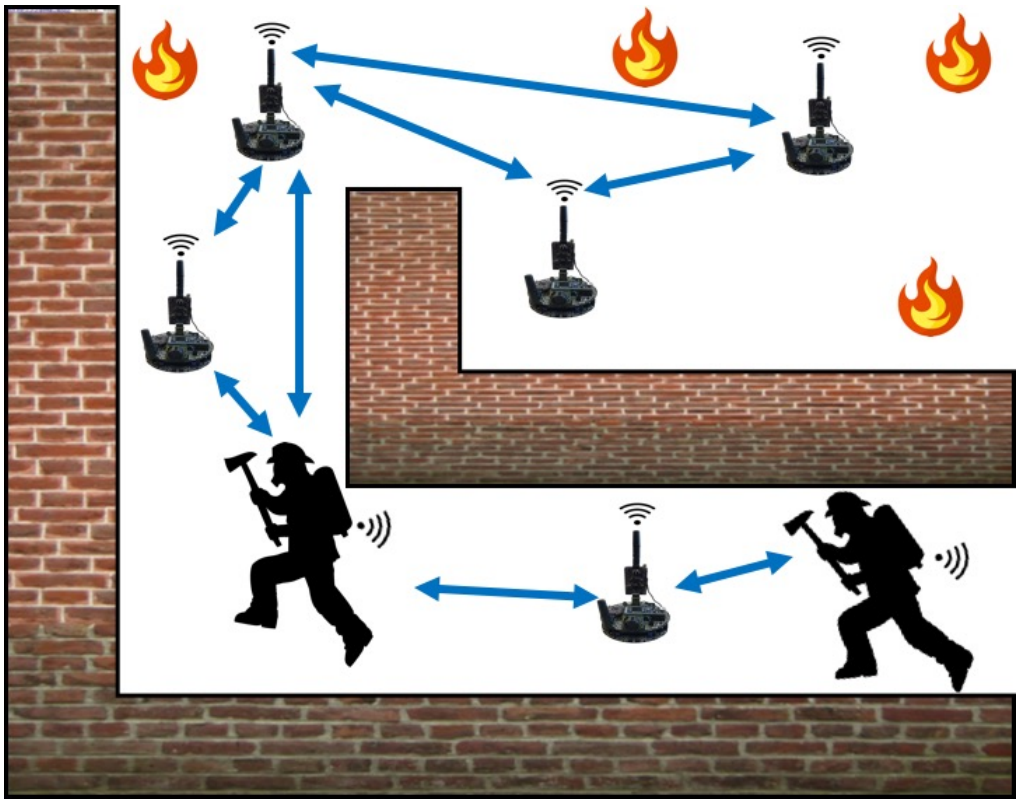


Figure 1.2: Illustration of an RWN where a group of five robots is sensing the environment around the firefighters to guide them in firefighting while also providing connectivity

The second genre of RWN research focuses on the application of robotic wireless routers to create and support a temporary communication backbone between a set of communicating entities. The main theme of these works is to exploit the controlled mobility of the robotic routers to support certain communication-related goals such as deploying a temporary communication backbone. In Figure 1.3, we present an example illustration where a set of two robotic routers form a communication relay path between two humans (e.g., two fire-fighters) who are unable to communicate directly. There exists a vast literature on multi-agent systems in robotics and control community that apply *simple disk models* for communication modeling and, subsequently, apply graph theory to solve different known problems such as connectivity and relay/repeater node placements. However, most of these existing works lack the inclusion of the effects of fading and shadowing in the communication models which is likely to significantly increase the complexity of the problems as well as the solutions. Some of the main challenges in this genre of RWN research are: link performance guarantee (in terms of Signal to Interference plus Noise Ratio, SINR, or Bit Error Rate, BER), optimized robotic router placements and movements in a dynamic network, non-linear control dynamics due to inclusion of network performance metrics into control loop, and localization. A special case of this would be robot-assisted static relay deployments, where the robots act as carriers of static relay nodes and smartly place/deploy them to form a communication path/backbone.

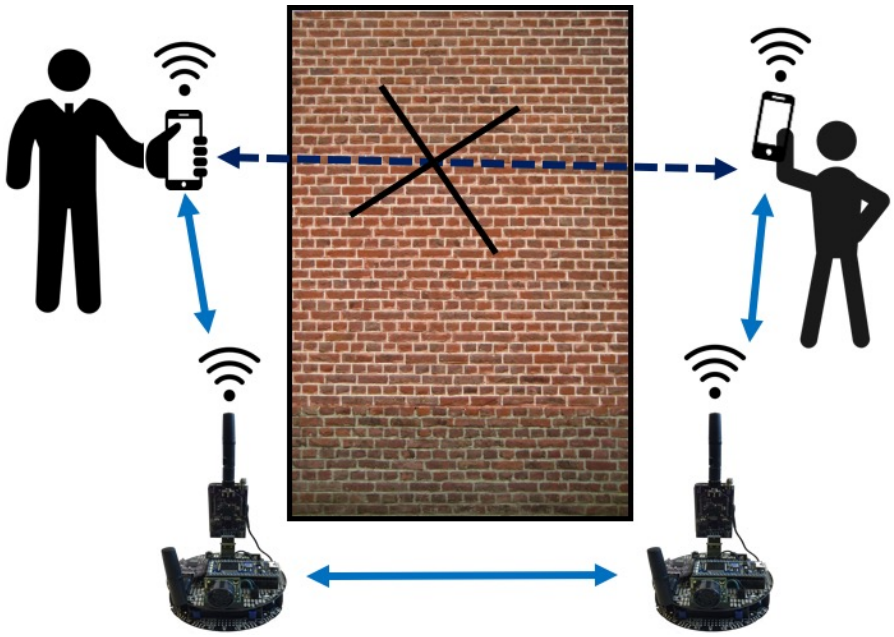


Figure 1.3: Illustration of robotic routers where the two humans are not able to communicate directly due to presence of a wall or some other blocking object

Next, we identify a set of major research problems and areas related to RWN as follows. Note that, it is by far not the exclusive set of research problems in RWN.

1.1.1 RSSI Models, Measurements, and RF mapping

RF measurement-based modeling, sensing, and mapping is an important topic of research related to an RWN. In an RWN, it is often important to estimate and monitor the quality of the communication links between the nodes (in terms of Bit Error Rate (BER), Signal to Noise plus Interference Ratio (SINR) etc.) in order to satisfy the communication-related requirements.[1] For practicality, these estimations should be either partly or fully based on online RF sensing such as temporal

[1]Note that RF-based communication is the standard mode of communication in RWN for obvious reasons.

RSSI measurements in a deployment. In some application contexts of RWN, the sole goal of an RWN is to sense and formulate an RF map of an environment to be processed or exploited later on [8].

1.1.2 Communication Protocols

Similar to any wireless networks and sensor networks, efficient communication protocols are of utmost importance in an RWN. Following the standard five layered Internet model for communication, we present a summary of the communication protocol research in RWN.

1.1.2.1 Media Access Control Layer

Media Access Control (MAC) protocols deal with proper distributed access to the physical medium among wireless devices, device to device communication, framing, and error corrections. While most of the classical MAC layer modules and protocols [18] are applicable to RWN, the extra dimension of controllability has opened up opportunities for a new class of MAC protocols in RWN. Hollinger *et al.* [19] presented such a MAC protocol for robotic sensor networks in acoustic environments. There also exist works related to mobile sensor networks which use predicted mobility patterns such as pedestrian mobility or vehicular mobility to

design efficient and application-context-specific MAC protocols [20, 21, 22]. In contrast, the mobility of the nodes in RWN are controlled and, thus, can be exactly known or predicted with higher accuracy.

1.1.2.2 Network Layer

The main protocols concerning network layer of an RWN pertain to routing and data collection. Routing in an RWN can be considered same as in Mobile Adhoc Networks (MANET) but with an extra advantage of controllability. While the existing well-known algorithms in MANET such as the Ad-hoc On-demand Distance Vector (AODV) [23, 24], Dynamic Source Routing (DSR) [25, 26], Optimized Link State Routing (OLSR) [27], and B.A.T.M.A.N. [28] can be applied to an RWN, they do not take advantage of the controlled mobility feature in RWN. The concept of RWN has opened up the door to a new class of routing protocols that incorporates the controlled mobility of the nodes in the routing decisions for more effective communication. Moreover, the end to end delay reduction (mainly for control packets) and reliability improvement have become of prime interests. Delayed or missing packets can result in an improper collaborative movement control and task completion in an RWN. To this extent, some researchers have modified existing routing solutions to adapt to a robotic network and proposed completely new routing solutions as well [29]. Nonetheless, it remained to be one of the less explored areas in RWN.

1.1.2.3 Transport Layer

In the field of RWN, the researchers are yet to significantly focus on the transport layer protocols. Till date, researchers employed traditional transport layer protocols such as TCP, UDP, or some MANET transport layer protocols for robotic networks, without taking advantage of the controlled mobility. Conversely, controllability requires highly reliable, low delay, and error-free communication between robots, which is not possible using original TCP or UDP and requires special transport layer protocols. The work of Douglas W. Gage on the MSSMP Transport layer protocol (MTP) [30] is mentionable in this context. In summary, the transport layer related research on RWN is an open area with a major focus on reliability and delay performances.

1.1.2.4 Application Layer

For proper operation of an RWN, one requires efficient application layer protocols that can efficiently handle control related traffic as well as regular traffic (e.g., sensor data). Some key requirements from such application layer protocols are a lower delay for control traffic, lightweight and low bandwidth consumption for scalability, and low power consumption for energy efficiency. Moreover, the abstraction provided to the user of the RWN should be simple and modular such that an end user can easily use and customize it for different applications contexts. The most popular state-of-the-art application layer method for effective control of robots and

effective collection of sensor data relies on the Robot Operating System (ROS) [31]. However, the traditional ROS-based solutions require high enough compute power to run a full-fledged Linux OS. Moreover, ROS uses XML-RPC which relies on HTTP, a protocol with large header sizes that in turn consume more bandwidth and power. This makes ROS unsuitable and unoptimized for a network of battery-operated robots operating with low-power embedded processors on a *shared wireless communication channel*. Thus, application layer protocol research is an important and promising area of research for building RWN systems.

1.1.3 Connectivity Maintenance

In any collaborative network of robots, it is important to maintain a steady communication path (direct or multi-hop) between any pair of nodes in the network for an effective operation. This problem, traditionally referred to as connectivity maintenance problem, in very well studied by the robotics research community. In the connectivity maintenance problem, the main goal is to guarantee the existence of end-to-end paths between every pair of nodes. The interaction between pairs of robots is usually encoded by means of a graph, and the existence of an edge connecting a pair of vertexes represents the fact that two robots can exchange information either through sensing or communication capabilities. *Notably, the connectivity of the interaction graph represents a fundamental theoretical requirement for proving the convergence of distributed algorithms in a variety of tasks, ranging*

from distributed estimation [32, 33, 34] to distributed coordination and formation control [35, 36, 37]. In the context of robotic networks, where the connectivity of the interaction graph is strictly related to the motion of the robots, a fundamental challenge is the design of distributed control algorithms which can guarantee that the relative motions of the robots do not result in a network partitioning, by relying only on local information exchange. Two possible versions of the connectivity maintenance problem can be considered: local connectivity and global connectivity. The local version of the connectivity maintenance problem focuses on the preservation of the original set of links of the graph encoding the pairwise robot-to-robot interactions to ensure its connectedness [38, 39, 40]. The global version of the connectivity maintenance problem focuses on the preservation of the overall graph connectedness, i.e., links can be added or removed as long as this does not prevent the interaction graph to remain connected over time [41, 42, 43, 44, 10].

1.1.4 Communication Aware Robot Positioning and Movement Control

As mention earlier, one of the application contexts of RWN is in supporting temporary communication backbones [45, 46]. One such application scenario involves the use of a set of *'robotic routers/relays'* to form a communication path between a set of *'communication endpoints'*. The communication endpoints are pairs of nodes/devices in the network that are willing to communicate but unable to reach

each other. The communication endpoints can be mobile or the environment can be dynamic with changing communication link properties. The communication endpoints might have certain communication requirements such as min achievable data rate, high throughput, and lower delay [7, 47]. The most important research question in such contexts is to devise a control system that adapts the positions of the robotic routers throughout the period of deployment to optimize the network performance while optimizing the movements as well. Therefore, the main goal of this class of work is continuous joint optimization of the robotic movements and the wireless network's performance. Moreover, the router placement controller should also be able to support network dynamics such as node failures and change in the set of communication endpoints. This problem involves direct relations with many other research pieces of RWN such as connectivity maintenance, communication link modeling, and localization.

1.1.5 Localization and Relative Positioning

The problem of localization is very well-known in the contexts of sensor networks and distributed robotics. The state-of-the-arts on localization are very mature [48, 49, 50, 51]. Since the field of RWN mainly deals with the sensing of the environment as well as positioning of the robotic routers, localization is definitely one of the major problems in the field of RWN. The concept of *'localization'* is to locate a node in a deployment arena with respect to a reference frame or a reference

location. A commonly used system called the Global Positioning System (GPS) localizes objects in terms of their latitudes and longitudes. However, GPS is known to not work properly in cluttered or indoor environments. Thus, a bulk of the target application contexts of RWN require an alternate and efficient localization scheme for indoor environments such as RF-based localization. Nonetheless, to make the problem more intense, in some of the application scenarios of an RWN such as fire-fighting in a dynamic environment, there might not exist any fixed infrastructure to support traditional alternative RF based trilateration solution for localization. While absolute locations are much important, a relative localization between the nodes in the network is sufficient in many of such contexts. For example, consider a scenario where a group of robotic routers is employed to connect a moving target with a base station. In such contexts, the robots form a chain where each robot positions itself with respect to its neighboring nodes only. Relative positions with respect to the neighboring nodes are enough for a node's movement control decisions in this context. The relative position is also of utmost importance in application contexts that require proximity maintenance between nodes/devices where proximity can be defined as maintaining certain thresholded distance. Therefore, the main focus of localization research in RWN is towards relative localization in an arena with none or limited localization infrastructure.

1.1.6 Edge Computing

Most of the robots/drones nowadays come with low power compute devices such as Raspberry Pi3. This has made possible the extension of edge or fog computing [52] to a network of robots. Edge computing focuses on exploiting all the devices near end users to comply with the skyrocketing demand for computationally intensive applications such as image processing and voice recognition towards autonomy and personalized assistance. To avoid unnecessary network overhead and delays, edge computing systems execute the required computation either partially or fully in the edge devices rather than transferring the responsibility to a remote central cloud. With drones and personal robots becoming ubiquitous near end users, it has opened up the opportunity of employing a network of robots (mainly drones) as the edge cloud [53] for application such as real-time video processing and streaming [54]. This demands for distributed algorithms and frameworks for edge computing that can optimally leverage the available compute nodes in an RWN for timely processing of the data.

1.2 Our Contributions

In Section 1.1, we discussed six major research areas in the cutting-edge field of robotic wireless networks. In this thesis, we contribute to five key problems pertaining to robotic wireless networks that fall along these areas of research:

- RSSI Based Relative Position Control
- Interference Power Bound Analysis for Network Formation Control
- Routing and Data Collection Protocol
- Unified Communication Protocol for Control and Sensing
- Passive RF Sensing

1.2.1 RSSI Based Relative Position Control

In our first study, we delve into the problem of relative location estimation and tracking/following of a moving target. *This problem pertains to the research areas of localization, relative positioning, and movement control.* For a collaborative work environment, the robots in an RWN need to properly localize and position themselves with respect to each other or a human. There exist many scenarios in indoor and cluttered environments where traditional GPS signals are limited, such as disaster operations in large cities or underground operations. While there are many camera and range-finder based systems for localization and tracking of moving objects [55, 56], the effectiveness of these sensors crumbles when visibility deteriorates or direct line of sight does not exist [57]. Moreover, the use of these types of sensors as well as the processing of their data, namely image processing, increase the form factor and power consumption of the robots which inherently always work

under power constraints. Thus, we need an alternative, cheap, and scalable solution with low processing power requirements to tackle low visibility contexts and cluttered environments. Among the alternatives, RF-based localization techniques are very popular [58]. However, most of these localizations schemes require a set of static reference nodes which might not exist in the target application contexts. To this extent, we propose Autonomous RSSI based RElative poSitioning and Tracking (ARREST), *the first-ever (to our knowledge) pure RSSI based single node RF sensing system* for relative location estimation (with decimeter level accuracy) and tracking/following of a moving object that can be implemented using commodity hardware. In our proposed system, the target, which we refer to as the *Leader*, carries an RF-emitting device that sends out periodic beacons. The tracking robot, which we refer to as the *TrackBot*, employs an off-the-shelf directional antenna, novel relative position and speed estimation algorithms, and a Linear Quadratic Gaussian (LQG) controller to measure the RSSI of the beacons and control its maneuvers. This system is further detailed in Chapter 4. This system is useful in a wide range of RWN application contexts such as fire-fighting or smart home, where the robots stay within the proximity of a human or another robot to provide a range of service such as sensing, data streaming, guiding, and packer relaying. By extending to multiple pairs of chained leader-follower robots, where each node can act either or both as leader and follower, this system can also be employed

to maintain certain distances between router nodes in a context of robotic router deployment.

1.2.2 Interference Power Bound Analysis for Network Formation Control

In our second study, we focus on answering one key question in the design of a multi-robot system that can self-organize to maintain a dynamic wireless network with acceptable link qualities: *is there a lower bound on the number of robots needed to deploy to guarantee the link quality requirements such as minimum Bit-Error-Rate (BER) and minimum acceptable Signal to Interference-Noise Ratio (SINR)? This problem directly falls under the category of communication aware robot position in RWN.* Based on our literature survey, we discover that while there exist some works on robotic routers [7, 59], none of them (to the best of our knowledge) answer this question which is fundamental to the deployment of an RWN. In this study, we first prove that in any CSMA based RWN, the interference power is upper bounded while achievable SINR is lower bounded. Next, we use the interference and SINR bounds to show that the required number of robots is also lower bounded given the target link performance requirements. We detail the process of mapping target link performance requirements into this bound in Chapter 5. This work will guide system developer to choose an appropriate number of robots to deploy as well as to get an idea of the performance limits of the deployed networks.

1.2.3 Routing and Data Collection Protocol

In the third study, we explore the problem of data collection and routing in an RWN. While the links are controllable, they still suffer from traditional issues such as interference, varying link qualities, and node failures. Moreover, in a network of robotic routers, the set of active links changes constantly. We focus on a Back-pressure routing algorithm called the Heat Diffusion (HD) algorithm [2], due to its promising performance guarantees without maintaining a routing table, and its ability to easily adapt to mobile environments. The core of this algorithm lies in the classical Heat transfer equations where heat flows from a heated region to a less heated or cold region. The HD algorithm uses the analogy of the heat propagation in collecting data from a source to a destination. To this end, we implement a practical version of the theoretical HD algorithm in Contiki OS [60], which we refer to as the Heat Diffusion Collection Protocol (HDCP) and analyze its performance for a varying set of network conditions in static settings, explained in Chapter 6. The real world experiments demonstrate promising routing properties such as lower delay compared to its alternatives and adaptability in the dynamic environment which makes HDCP a competent routing protocol for RWN.

1.2.4 Unified Communication Protocol for Control and Sensing

In the course of our fourth study, we focus on another important research problem in RWN that pertains to the application layer protocol research. For efficient operation of an RWN that consists of a wide range of battery-constrained heterogeneous robots operating on a shared limited bandwidth channel, we require lightweight, low-power. low bandwidth consuming application layer protocols. To this end, we noticed that the traditional solutions, mainly based on the well-known Robot Operating System (ROS), fall short due to the dependency on heavy bandwidth and power consuming protocols such as XML-RPC. To this end, we propose the Robotic Overlay coMmunicAtioN prOtocol (ROMANO) which is a novel *lightweight overlay networking protocol* for sensing and control of a set of heterogeneous robots that build upon the cutting-edge lightweight publish-subscribe Internet of Things (IoT) protocol called the Message Queuing Telemetry Transport for Sensor Nodes (MQTT-SN) [61]. ROMANO employs the concept of "topics" from the MQTT-SN communication protocol to create an overlay network of robots where each robot subscribes/publishes to a set of control and sensing related topics (e.g., gyroscope, proximity, location, speed, movement control instructions, etc.). ROMANO can change and control the topic subscriptions of different robots to control the ROMANO communication endpoints for different types of communication: one to one, one to many, many to many, or many to one. This is further detailed in Chapter 7.

1.2.5 Passive RF Sensing

In the fifth and final study, we look into the problem of passive RF sensing in an unknown environment using the concept of a bistatic radar [62]. Researchers have applied RF signal in many sensing and mapping contexts including but not limited to robotic mapping of unknown areas [63], indoor localization [48], and see through capabilities [8]. In this study, we propose a passive RF mapping system with a single rotating directional antenna that scans the environment for different orientation of the antenna and collects a set of directional RSSI samples from a single omnidirectional transmitter. Next, we feed these directional RSSI samples in form of an RSSI vector to an MLE based estimation module that iterates through different potential combinations of the transmitter and reflectors locations to find the location combination that is most probable to generate the collected RSSI vector. We employ the TrackBot prototype from Study 1 to perform a set of simulation and real-world experiments to prove the concept and to analyze the performance of the MLE based estimation algorithm. This system is detailed in Chapter 8. This passive sensing technique is meant to be combined with the RF-based relative localization and positioning method from Study 1 to make of use of the same hardware for a dual purpose: localization and mapping.

Chapter 2

Background

In this chapter, we explain some key concepts and preliminaries that are required to better understand the studies presented in this thesis.

2.1 Kalman Filter

In this section, we briefly present the process of Kalman Filtering which is adapted from [1, 64]. The Kalman filter is one of the most ubiquitous filter in statistical noise filtering for long-term measurement based estimation of a set of unknown variables. The diverse range of application domains for the Kalman Filtering includes but not limited to robotics, wireless sensor networks, signal processing, and control. The core of the Kalman Filtering lies in two recursive steps: Prediction and Estimation. Kalman filter keeps track of the estimations of the system state and the noise over time and uses them to predict/estimate the next state. Briefly speaking, Kalman filter first predicts the state at time n as $\hat{\mathcal{S}}_{n|n-1}$ by using the last state estimate

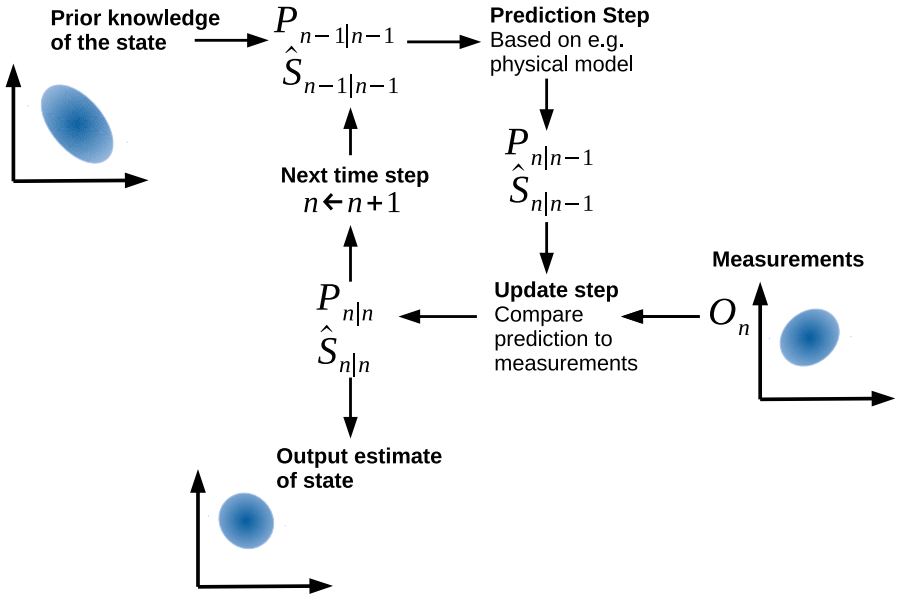


Figure 2.1: Illustration of the Kalman Filtering (This figure is adapted from [1])

$\hat{\mathcal{S}}_{n-1|n-1}$ and the system model. Similarly, it predicts the uncertainty or noise state at time n, $P_{n|n-1}$, from the respective estimations at time $n-1$, $P_{n-1|n-1}$. Next, the Kalman filter applies the observation/measurement at time n, O_n, to update the state and uncertainty estimates for time n, $\hat{\mathcal{S}}_{n|n}$ and $P_{n|n}$ respectively. This process, illustrated in Fig. 2.1, is repeated throughout the duration of the experiments.

2.2 Linear Quadratic Gaussian Control

A Linear Quadratic Gaussian (LQG) controller is a combination of a Kalman Filter and a Linear Quadratic Regulator (LQR) that is proven to be the *optimal controller* for linear systems with Additive White Gaussian Noise (AWGN) and incomplete

state information [65]. The linear system equations for any discrete LQG problem can be written as:

$$\begin{aligned} \mathcal{S}_{n+1} &= A_n \mathcal{S}_n + B_n \mathbf{U}_n + \mathbf{z}_n \\ \mathbf{O}_n &= C_n \mathcal{S}_n + \mathbf{w}_n \end{aligned} \tag{2.1}$$

where $\mathcal{S}_n$ is the state vector, A_n and B_n are the state transition matrices, $\mathbf{U}_n$ is the LQG control vector, $\mathbf{z}_n$ is the system noise with covariance Z_n, $\mathbf{O}_n$ is the LQG system's observation vector, C_n is the state-to-observation transformation matrix, and $\mathbf{w}_n$ is the observation noise with covariance W_n at time n. An LQG controller first predicts the next state based on the current state and the signals generated by the LQR. Next, it applies the system observations using Kalman filtering to update the estimates further and generates the control signals based on the updated state estimates. For a finite time horizon LQG problem [66] with N being the horizon, the cost function can be written as:

$$J = \mathbb{E}\left(\mathcal{S}_N^T \mathbf{F} \mathcal{S}_N + \sum_{n=0}^{N-1} \mathcal{S}_n^T \mathbf{Q}_n \mathcal{S}_n + \mathbf{U}_n^T \mathbf{H}_n \mathbf{U}_n \right) \tag{2.2}$$

where $\mathbf{F} \geq 0, \mathbf{Q}_n \geq 0, \mathbf{H}_n > 0$ are the weighting matrices.

The discrete time LQG controller for this optimization problem is:

$$\begin{aligned} &\hat{\mathcal{S}}_{n+1} = A_n \hat{\mathcal{S}}_n + B_n \mathbf{U}_n + K_{n+1}(\mathbf{O}_{n+1} - C_{n+1}\{A_n \hat{\mathcal{S}}_n + B_n \mathbf{U}_n\}) \\ &\mathbf{U}_n = -L_n \hat{\mathcal{S}}_n \qquad \text{and} \qquad \hat{\mathcal{S}}_0 = \mathbb{E}[\mathcal{S}_0] \end{aligned} \tag{2.3}$$

where $\hat{}$ denotes estimates, K_n is the Kalman gain which can be solved via the algebraic Riccati equation [67], and L_n is the feedback gain matrix. The Kalman gain can be calculated as:

$$\begin{aligned} K_n &= P_n C_n^T (C_n P_n C_n^T + W_n)^{-1} \\ P_{n+1} &= A_n \left(P_n - P_n C_n^T \left(C_n P_n C_n^T + W_n \right)^{-1} C_n P_n \right) A_n^T + Z_n \\ P_0 &= \mathbb{E} \left(\mathcal{S}_0 - \hat{\mathcal{S}}_0 \right) \left(\mathcal{S}_0 - \hat{\mathcal{S}}_0 \right)^T \end{aligned} \tag{2.4}$$

The feedback gain matrix equals

$$\begin{aligned} L_n &= (B_n^T R_{n+1} B_n + H_n)^{-1} B_n^T R_{n+1} A_n \\ R_n &= A_n^T \left(R_{n+1} - R_{n+1} B_n \left(B_n^T R_{n+1} B_n + R_n \right)^{-1} B_n^T R_{n+1} \right) A_n + Q_n \\ R_N &= F. \end{aligned} \tag{2.5}$$

2.3 Backpressure Routing

The general idea behind dynamic queue-aware routing algorithms such as the Backpressure (BP) [68] algorithm and the Heat Diffusion (HD) algorithm [2] is that they do not require any explicit path computation. Instead, the next-hop for each packet depends on queue-differential weights that are functions of the local queue occupancy information and link state information at each node. It is a general assumption in (theoretical) queue-aware routing algorithms such as BP that the networks operate in slotted time. Furthermore, a wireless network is represented as

a graph $(\mathcal{V}, \mathcal{E})$ with vertices $\mathcal{V}$ and edges $\mathcal{E}$. However, the adjacent links or edges of a wireless network cannot be used simultaneously due to many constraints such as interference. In that context, a *maximal schedule* is defined as a set of links such that no two links interfere with each other and no other link can be added to that set without causing interference. We will denote a maximal schedule as: $\pi = \{\pi_{ij} | i \neq j \text{ and } \ i, j \in \mathcal{V}\} \in \{0,1\}^{|\mathcal{E}|}$, where $\pi_{ij} = 1$ if the link ij (or link ji) is included in the schedule. The set of all such maximal schedule is referred to as a *scheduling set*, denoted as Π. Next, we briefly discuss the Backpressure routing algorithm, first proposed by Tassiulas and Ephremides [68], and extended by Neely *et al.* [69, 70]. The original BP routing algorithm [68] consists of three major steps: BP weighing, BP scheduling, and BP forwarding. The BP algorithm uses the information about estimated channel capacities $\mu_{ij}(n)$ and the queue backlogs $q_i(n)$ to make the routing decisions at each time slot n. This follows a brief description of the BP routing steps including the penalty optimization extension introduced by Neely *et al.*

2.3.1 BP Weighing

For each link ij in the network, find the queue differential, $q_{ij}(n) = q_i(n) - q_j(n)$. Next, assign some weights to the links based on the queue differential as follows.

$$w_{ij}(n) = \mu_{ij}(n) q_{ij}(n) \tag{2.6}$$

In the original BP, only the queue stabilities are considered. To incorporate the routing cost into the BP, the drift-plus-penalty approach [69, 70] was proposed, which we refer to as the V-parameter BP algorithm. In this approach, a route usage cost is added as a negative penalty in the weight calculation as follows.

$$w_{ij}(n) = \mu_{ij}(n)(q_{ij}(n) - V.\theta_{ij}) \tag{2.7}$$

where $V \in [0, \infty)$ determines the importance of the link penalty and θ_{ij} is the link penalty which depends on the link utility or cost function along with some penalty functions.

2.3.2 BP Scheduling

Find or choose a scheduling vector $\pi \in \Pi$ that maximizes the sum of the weights of the activated links. In other words, choose a scheduling vector π such that

$$\phi(n) = \arg\max_{\pi \in \Pi} \sum_{ij \in \mathcal{E}} \pi_{ij} w_{ij}(n) \tag{2.8}$$

In case of ties, the scheduling vector is chosen randomly from the solution set.

2.3.3 BP Forwarding

Based on the scheduling vector from the BP scheduling step, if a link is active at time-slot n i.e., $\pi_{ij}(n) = 1$, and if the link weight $w_{ij}(n) > 0$, then transmit packets

on that link at full capacity $\mu_{ij}(n)$. Null packets are sent if a node does not have enough packets to send.

2.4 Publish Subscribe Protocol and MQTT

In this section, we briefly explain the core concepts of MQTT-SN and MQTT to better understand ROMANO. Message Queuing Telemetry Transport (MQTT) is a publish-subscribe based machine to machine application layer networking protocol for the Internet of Things (IoT). The core idea is that a set of "subscriber" nodes are connected to a set of "publisher" nodes via a "broker" and the concept of a "topic". Each publisher publishes its messages to the broker by tagging them with a topic ID. The broker receives the published messages and sends them to all the subscribers of that topic. When there are multiple subscribers to a topic, the broker dispatches copies of a published message via sequential unicast. This is further illustrated in Fig. 2.2.

MQTT works on top of the Transmission Control Protocol (TCP) which guarantees reliable message transfer. MQTT also provides its own Acknowledgments (ACKs) for additional reliability via three different Quality of Service (QoS) modes numbered in the order of increasing complexity: QoS 0, QoS 1, and QoS 2. The QoS feature of MQTT handles the required message retransmissions to provide message delivery guarantee in an unreliable lossy network with frequent packet losses. In that regard, the QoS 0 refers to at most once message transfer guarantee, the QoS

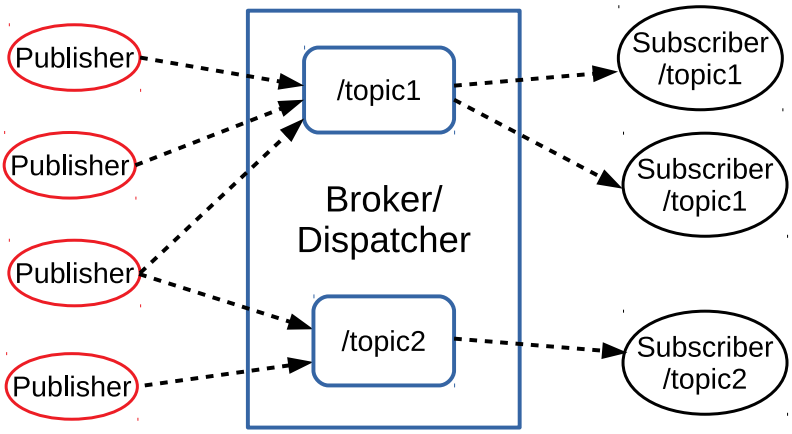


Figure 2.2: Illustration of Publish-Subscribe Communication: A set of publisher nodes publishes their data to a broker by associating the data with a topic. The broker relays the data to all the subscribers of that topic

1 refers to at least once message transfer guarantee, and the QoS 2 refers to exactly once message transfer guarantee.

MQTT for Sensor Nodes (MQTT-SN) is a variant of MQTT that is focused on resource-constrained devices such as battery powered sensors. MQTT- SN uses the User Datagram Protocol (UDP) rather than TCP and has smaller message headers to reduce the overall communication overhead. However, MQTT-SN still maintains reliability through the QoS levels used in MQTT. The typical format of an MQTT-SN Publish message is as follows.[1]

Table 2.1: MQTT-SN Publish Message Format

length (octet 0)	Msg Type (1)	Flags (2)	Topic id (3-4)	MsgId (5-6)	Data (7:n)

[1]For a more detailed description of MQTT-SN, interested readers are referred to `http://mqtt.org/new/wp-content/uploads/2009/06/MQTT-SN_spec_v1.2.pdf`.

The fields Length, Msg Type, Flags, and Msd Id are used by MQTT-SN for proper message delivery whereas the “Topic id” determines the receiving endpoints of a message and the “Data” field contains the actual message.

Chapter 3

Related Works

Till last decade, there was no significant focus on the RWN related research topics, mostly due to lack of proper technologies and hardware. Thus, most of the existing works on RWN topics are very recent. However, there exist a range of works related to collaborative robotics and wireless sensor networks that are very much relevant to our studies. In this section, we present a summary of the existing works related to each of the studies.

3.1 Relative Localization and Tracking

In this section, we present the existing works related to our proposed ARREST system for RSSI based relative localization and positioning. This set of related works can be subdivided into three major classes: Tracking related works, Relative Localization related works, and LQG based control related works.

3.1.1 Tracking

Localization and Tracking of moving targets have been a very active field of research in the robotics research domain. Over last two decades, researchers have proposed a range of interesting techniques for tracking different moving objects such as humans, robots, and animals. To this extent, the most common architectures are based on Vision and Laser Range Finder systems. Researchers have proposed a class of efficient sampling and filtering algorithms for vision-based tracking such as the Kalman filtering and the particle filtering [56, 71]. Among the pioneer works, the work of Papanikolopoulos, Khosla, and Kanade [55] in combining vision algorithms with different controller schemes, ranging from a PID to complicated LQG controller, is mentionable. Among recent works, Jung and Sukhatme [56] proposed an adaptive particle filter based approach for tracking a moving object in the image space with a single camera based robot while omnidirectional camera-based tracking system is proposed in [72]. In [73], a Laser Range Finder based tracking of multiple moving objects that relies on a sequence of temporal snapshots is proposed for an autonomous wheelchair guidance system. There also exist some works that combine vision with rangefinder such as the work of Lindström and Eklundh [57], and Kleinehagenbrock *et al.* [74]. However, any camera/vision based approach has many limitations such as the visibility requirement, limited field of vision of traditional cameras, larger form-factor of the robots, and costly image processing software requirements. On the other hand, while the laser range based

methods do not suffer from the visibility problem, they are limited to direct line of sight between the target and the tracker and require complicated processing. *In contrast, our proposed ARREST architecture employs RF signals for localization and tracking, which can be developed with low-cost, small form-factor hardware, and can be applied in scenarios with limited visibility and non-line-of-sight such as inside cluttered indoor environments, rubbles, forests, and undergrounds.*

3.1.2 Relative Localization

Another class of related works lies within the large body of works in the field of RF Localization in wireless sensor network [58] where robots are employed for localizing static nodes. The work of Graefenstein *et al.* [75] that employs a rotating antenna on a mobile robot to map the RSSI of a region and exploit the map to localize the static nodes, is mentionable in this context. Similar works have been proposed in the context of tracking fish or wild animals [76, 77], where the target is assumed to be a static radio-tagged fish/animal. The main problem addressed in these works are proper path planning and sensing location selection for sufficient data collection. On a related topic, hardware for the localization and tracking of a moving target (fish) are proposed in [78] where the navigation of the tracker robots rely on a combination of GPS and Compass. Similar work for wildlife tracking using aerial robots is shown in [79]. Related to RF-based relative positioning works, Zickler and Veloso [80] proposed a data-driven approach that relies completely on the Robot's

received RSSI for localization and tethering purpose. In their discrete grid based Bayesian probabilistic approach, the target communicates its odometer reading with the tracker while the tracker moves to multiple positions relative to the target and collects multiple RSSI measurements. Oliveira *et al.* [81] proposed an RSSI based, anchor-less, relative localization system for a group of mobile robots, which relies on pairwise RSSI measurements between the robots, and applies Kalman filter and the Floyd–Warshall algorithm for relative localization. Some researchers have also employed infrared [82] and ultrasound devices [83] for relative localization. Vasisht, Kumar, and Katabi [84] have applied a MIMO-based system to relatively localize a single node. Simulation of an RSSI based constant distance following technique is demonstrated in [85] where the leader movement path is predetermined and known to the Follower. *However, unlike these works, the TrackBot in the ARREST system relies solely on RSSI data not only for the localization of the mobile Leader with unknown movement pattern but also for autonomous motion control with the goal of maintaining a bounded distance.* The closest state-of-the-art related to our work is presented in [86]. In this work, the authors developed a system that follows the bearing of a directional antenna for effective communication. However, to our knowledge, the maintenance of guaranteed close proximity to the Leader was not discussed in [86], which is the most important goal in our work. Moreover, this work employs both RSSI and sonar to determine the orientation of the transmitter

antenna along with comparatively costly and high power consuming processing hardware with the larger form factor.

3.1.3 LQG Based Control

On LQG related works, Bertsekas [66] has demonstrated that an LQG controller can provide the optimal control of a robot along a known/pre-calculated path when the uncertainty in the motion, as well as the observation noise, are Gaussian. Extending this concept, Van Den Berg, Abbeel, and Goldberg [87] proposed an LQG based robotic path planning solution to deal with uncertainties and imperfect state observations. In [88], Van den Berg *et al.* proposed yet another LQG based control system for obstacle avoidance in robotic motions. There exist many other LQG based robotic path planning and control systems [89, 90]. *To the best of our knowledge, we are the first to combine the RSSI based relative position, angle, and speed estimation with LQG control for localization and tracking of a moving RF-emitting Object.*

3.2 Interference Power Bound for CSMA Based Wireless Network

In our venture for a generic model (during the second study) to estimate the number of robots to deploy (by estimating the maximum allowed inter-node distance

to maintain the target SINR), we explored the existing literature in search for a proper model of interference and Signal to Interference plus Noise Ratio (SINR) range analysis in a CSMA/CA based wireless network. There exist a large body of works that characterize the mean interference power distribution in CSMA networks ([91, 92]) by employing the concepts of point process such as Poisson point process, Mat'ern hard-core process, and simple sequential inhibition [93]. *The basic idea of this class of work is to represent the locations of the interferers as spatial point processes, more specifically, hard-core point processes where the nodes fulfill a criterion of being a certain distance apart to take into account CSMA among themselves.* Through the application of different point process properties such as thinning and superpositions, researchers [91, 94, 92, 95] have estimated the probability distributions of the mean interference powers in the presence of CSMA/CA. Interested readers are referred to [96] for a detailed survey on this class of works. Among the other class of works, the work of Hekmat and Van Mieghem [97] is the most relevant to us. They demonstrated that the interference power in the presence of CSMA is actually upper bounded and can be best estimated by use of hexagonal lattice structure. *However, this work as well as most of the other works include some assumptions such as the receiver being located at the center of a contention region, which is only acceptable if the devices follow the 802.11 RTS/CTS standards [98]. Interestingly, in practice, very few commercially available products actually employ the RTS/CTS mechanism. Furthermore, the Internet*

of Things (IoT) and Wireless Sensor Network (WSN) standard 802.15.4, which is also a standard choice for robotic network platforms, does not use RTS/CTS mechanism, in order to avoid inefficiencies. Thus, it is actually the transmitter that employs the CSMA and should be located at the center of the contention region, whereas, the receiver is free to be anywhere inside the transmitter's communication range. In such cases, the SINR and the interference mean values as well as the bounds for a link are, in fact, functions of the separation distance (d) between the endpoints of the link. *In our second study, unlike the existing works, we characterize the SINR or the interference as a function of the separation distance (d) and apply the modified bounds to estimate the number of robots to be deployed to satisfy the communication performance goals.*

3.3 Data Collection Routing Protocols

In this section, we present a brief survey of the existing literature that is directly related to the proposed HDCP routing protocol for RWN in our third study. In the existing network theory literature, there exists a range of throughput optimal policies [99, 100, 101] alongside the well-known Backpressure routing [69] algorithm. The HD algorithm also provides the same throughput optimality guarantee in theory. However, what motivated us to implement HD were the striking additional expected performance capabilities (based on the theoretical results)— that it also offers a Pareto-optimal trade-off between routing cost and queue congestion.

There have also been several reductions of Backpressure routing to practice in the form of distributed protocols, pragmatically implemented and empirically evaluated for different types of wireless networks [102, 103, 104]. Most relevant to the present work is the Backpressure Collection Protocol (BCP) developed by Moeller *et al.* [102], the first-ever implementation of dynamic queue-aware routing in wireless sensor networks. Our third study is informed by the BCP approach in implementing the Backpressure routing in a distributed manner. We also directly compare the performance of the new HDCP protocol with BCP.

Besides BCP, there are a number of other prior works on routing and collection protocols for wireless sensor networks, including the Collection Tree Protocol (CTP) [105], Glossy [106], Dozer [107] and Low-power Wireless Bus [108]. We provide a side by side comparison of HDCP with the well-known CTP and BCP protocols. We believe this provides a meaningful comparison with a state of the art minimum cost quasi-static routing protocol as well as a state of the art queue and cost-aware dynamic routing protocol.

In recent years there has been a significant focus on developing networking protocols that are IP-friendly, such as RPL [109]. While this study does not focus on providing an IP-compliant version of HD, there is prior work on extending BCP to handle IP packets [110] and we believe that a similar approach could be adopted to enable IP operation for HDCP.

In the prior works, Banirazi *et al.* have presented the idealized Heat Diffusion routing algorithm [2, 111]. All of these are network theory papers that spell out a centralized algorithm, assume global synchronization, assume that at each time step an NP-hard Maximum Weight Independent Set problem can be solved and that all queues are of an unlimited size, and under these assumptions prove various properties of the HD algorithm. The only evaluations presented in these works are idealized MATLAB simulations. Our third study is clearly inspired by and built upon the earlier works on HD routing, but is the first to develop and implement it as a realistic distributed protocol (HDCP) and evaluate it on a real testbed.

3.4 Pub-Sub Based Robotic Communication and Control

In our fourth study, we propose an overlay communication protocol as an application layer protocol for an RWN that uses the well-known MQTT-SN as the protocol underneath. The widespread alternatives to MQTT-SN are MQTT, Constrained Application Protocol (CoAP) [112], and XML-RPC (ROS). Amaran *et al.* [113] presented a comparison of these protocols, showing that MQTT-SN and CoAP have similar performance and advantages over MQTT and XML-RPC (ROS). Additionally, they showed that MQTT-SN messages are slightly more efficient than CoAP,

which motivates our choice of MQTT-SN. Recently, a middleware named Data Distribution Service (DDS) is gaining popularity in the space of Industrial IoT due to its delay and throughput guarantees [114]. However, the delay and throughput guarantees are materialized at a cost of increased bandwidth consumption and control traffic overhead [115]. On contrary, MQTT-SN has significantly less bandwidth consumption and control overhead which we believe to be a key communication requirement in dense robotic swarms. Nonetheless, since ROMANO is an overlay protocol, it can be easily ported to DDS. Another mentionable protocol for high bandwidth and low latency systems is the Lightweight Communication and Marshalling (LCM) protocol [116] with the key difference with MQTT-SN being the "high bandwidth" requirement. Thus, we opt for the MQTT-SN protocol instead.

While MQTT-SN has been proposed and used in the context of sensing in static IoT sensor deployments, MQTT-SN has not yet been used for a network of robots. Many researchers have envisioned/proposed utilizing MQTT in the context of robotic control. Aroon [117] demonstrated the feasibility of remotely controlling a single robot over a cloud platform via MQTT. Kazala *et al.* [118] have also presented a proof of concept implementation of using the basic functionality of MQTT for data exchange among multiple robots. However, to the best of our knowledge, there are no publications presenting a low power MQTT-SN based overlay protocol that allows nodes to easily facilitate communication endpoints (one-to-one, one-to-many, many-to-one, many-to-many) among a robot swarm. SENORA, proposed

in [119], includes an inter-robot communication protocol which takes into account robot location for medium access and peer-to-peer communication, but the authors do not detail how generic communication would be facilitated among robots (e.g. one-to-one or one-to-many) which our protocol addresses. The authors of [120] proposed a messaging architecture for inter-robot communication with the target application specifically for integration into a surveillance system. In contrast, our work targets generic multi-robot systems along with various application-specific real implementation details. Sauer *et al.* [121] presented the concept of an overlay protocol built on top of the CoAP but do not present any implementation details or performance evaluation. The lack of details makes it impossible for us to replicate and compare with the existing work. On the other hand, ROS messages are widely used to connect robot swarms. For example, Yan *et al.* [122] presented a prototype system built on ROS messages for robot communication and described the ease of scaling their system, although they do not include any performance evaluation. However, while ROS messages are powerful, their packet header overhead and the computation requirement to run ROS do not make it ideal or bandwidth-efficient for low capacity robots consisting of only microcontrollers using IEEE 802.15.4 radios. Nonetheless, it should be still possible to tunnel ROMANO messages through ROS, if so desired by an application developer.

3.5 RF Mapping

RF signal based localization and mapping have been extensively studied in the existing literature. In this context, the work of Mostofi *et al.* [8, 123] on the mapping of an environment using two moving RF transceivers robots is relevant. In their works, the robots follow a predetermined set of paths for collecting a set of RF samples which are later processed using the concept of compressible sampling to map obstacles. They exploit the attenuation introduced by different obstacles to map them. *Our goal is slightly different as we are interested in passively localizing the objects/surfaces by extracting multipath components from the received signal at a single receiver while the transmitter and receiver remain in the line of sight with respect to each other. This also separates our work from the standard RF Sensor Network based passive localization works [124] where a fixed network of RF devices monitor changes in the RF communication channel properties in order to passively localize an object.* There also exist some passive localization works that employ UWB radios and MIMO systems to localize objects. Aditya and Molisch [125] presented one such solution where a set of transmitters and receivers use the blocking characteristics of pairwise communication links to passively localize objects. Gulmezoglu, Guldogan, and Gezici have proposed a similar UWB radio based solution in [126]. There also exist some works that use RFID [127] and multiple receiving antennas for localization. The work of Tan, Chetty, and Jamieson [128] on using 8-element uniform circular phased arrays for through-wall passive sensing and

mapping is also relevant. The proposed system, called TrueMapper, was built on top of costly, power consuming USRP radios. Fadel *et al.* [129, 130] have worked on a custom solution called RF-capture that employs a directional antenna array to sense human motion. They capture the reflections of a human body with each antenna transceiver in the array, which are later processed jointly to generate an image. Chetty, Smith, and Woodbridge [131] also presented a wifi based multi-static radar system for through-the-wall passive sensing. There also exist some systems [132, 45] that use phase information of the RF signal along with RSSI to implement synthetic aperture radar (SAR) based imaging and localization. However, most of the cheap, commercially available RF modules do not provide access to the phase information. *What separates our work is the application of a single RF transmitter-receiver antenna pair instead of multiple antennas or antenna arrays to achieve passive localization of reflecting objects with acceptable performance that can be implemented with low-cost off-the-shelf hardware.*

Chapter 4

RSSI Based Relative Position Control for RWN

In this study, we focus on a class of relative localization and tracking problems in RWN where the term **"tracking"** refers to the relative positioning and control of a robot that is required to stay in the bounded proximity of an uncontrolled/controlled moving target, which can be a leader robot or a human.[1] We present Autonomous Rssi based RElative poSitioning and Tracking (ARREST), a new robotic mobile sensing system for tracking and following a moving RF-emitting object, which we refer to as a Leader, solely based on signal strength information. This kind of system can expand the horizon of autonomous mobile tracking, distributed robotics, and robotic routers into many scenarios with limited visibility such as night time or adverse weather operations, forests that are dense or on fire, cluttered environments, and underground settings [4, 11, 3, 12, 136]. Our proposed tracking agent, which we refer to as the TrackBot, uses a single rotating, off-the-shelf, directional antenna, novel angle, and relative speed estimation algorithms,

[1]The material in this chapter is based in part on the works in [133, 134, 135].

and Kalman filtering to continually estimate the relative position of the Leader with decimeter level accuracy (which is comparable to a state-of-the-art multiple access point based RF-localization system) and the relative speed of the Leader with accuracy on the order of 1 m/s. The TrackBot feeds the relative position and speed estimates into a Linear Quadratic Gaussian (LQG) controller to generate a set of control outputs to control the orientation and the movement of the TrackBot with the objective of staying withing a threshold distance, D_{th} of the leader. We perform an extensive set of real-world experiments with a full-fledged prototype to demonstrate that the TrackBot is able to stay within 5m of the Leader with: (1) more than 99% probability in line of sight scenarios, and (2) more than 70% probability in no line of sight scenarios, when it moves 1.8X faster than the Leader. For ground truth estimation in real-world experiments, we also developed an integrated Time Difference of Arrival (TDoA) based distance and angle estimation system with centimeter-level relative localization accuracy in the line of sight scenarios.

4.1 Problem Formulation

In this section, we present the details of our tracking problem and our mathematical formulation based on both a 2D global frame of reference, $\mathcal{R}_G$, and the TrackBot's 2D local frame of reference at time t, $\mathcal{R}_F(t)$. Let the location of the *Leader* at time t be represented as $\mathbf{X}_L(t) = (x_L(t), y_L(t))$ in $\mathcal{R}_G$. The *Leader* follows an unknown path, $\mathcal{P}_L$. Similarly, let the position of the TrackBot at any time instant

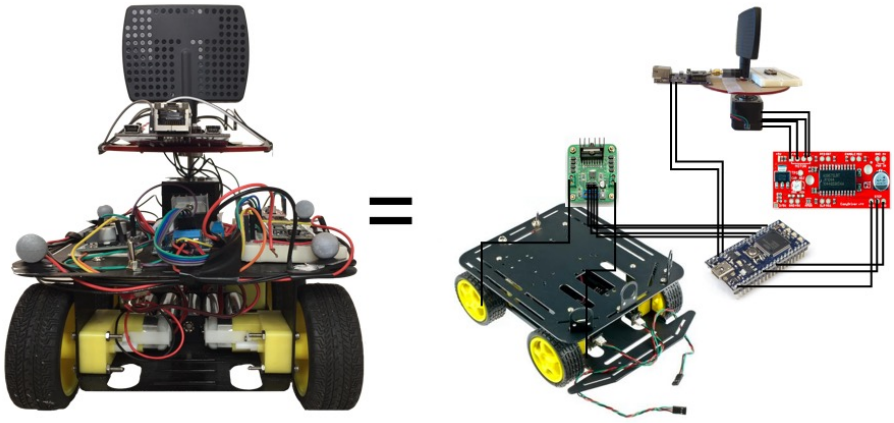


Figure 4.1: The TrackBot Prototype. This prototype system is built using commercial off-the-shelf products. The black lines illustrate the wire connections between different hardware components.

t be denoted by $\mathbf{X}_F(t) = (x_F(t), y_F(t))$. The maximum speeds of the Leader and the TrackBot are v_L^{max} and v_F^{max}, respectively. For simplicity, we discretize the time with steps of $\delta t > 0$ and use the notation n to refer to the n^{th} time step i.e., $t = n \cdot \delta t$. Let $d[n] = ||\mathbf{X}_L[n] - \mathbf{X}_F[n]||_2$ be the distance between the TrackBot and the Leader at time-slot n, where $||.||_2$ denotes the L_2 norm. Then, with D_{th} denoting the max distance allowed between the Leader (L) and the TrackBot (F), the objective of tracking is to plan the TrackBot's path, $\mathcal{P}_F$, such that $\mathbb{P}\left(d[n] \leq D_{th}\right) \approx 1 \quad \forall n$ where $\mathbb{P}(.)$ denotes the probability.

However, realistic deployment scenarios typically do not have a global frame of reference. Thus, we formulate a local frame of reference, $\mathcal{R}_F[n]$, with the origin representing the location of the TrackBot, $\mathbf{X}_F[n]$. Let the robot's forward and backward movements at any time instant n be aligned with the X-axis of $\mathcal{R}_F[n]$. Also, let the direction perpendicular to the robot's forward and backward movements be

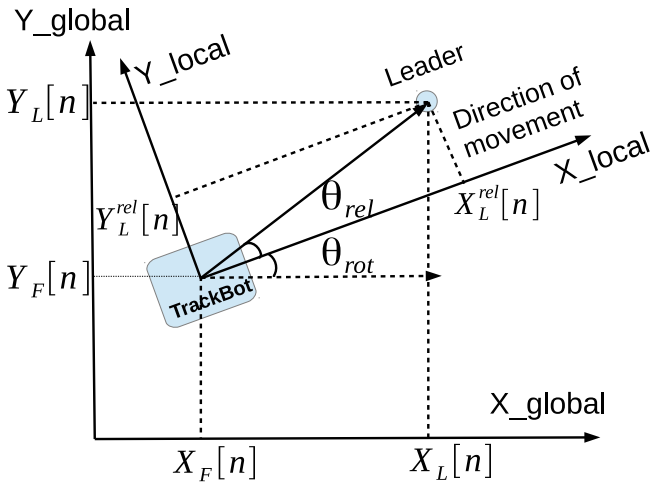


Figure 4.2: Global and Local Coordinate System Illustration

aligned with the Y-axis of $\mathcal{R}_F[n]$. This local frame of reference is illustrated in Fig. 4.2. Note that in our real system all measurements by the TrackBot are in $\mathcal{R}_F[n]$. In order to convert the position of the Leader in $\mathcal{R}_F[n]$ from $\mathcal{R}_G$ or vice versa for simulations and emulations, we need to apply coordinate transformations. Let the relative angular orientation of $\mathcal{R}_F[n]$ with respect to $\mathcal{R}_G$ be $\theta_{rot}[n]$ and the position of the Leader in $\mathcal{R}_F[n]$ be $\mathbf{X}_L^{rel}[n] = (x_L^{rel}[n], y_L^{rel}[n])$. Then:

$$\begin{bmatrix} x_L[n] \\ y_L[n] \\ 1 \end{bmatrix} = \begin{bmatrix} \cos(\theta_{rot}[n]) & -\sin(\theta_{rot}[n]) & x_F[n] \\ \sin(\theta_{rot}[n]) & \cos(\theta_{rot}[n]) & y_F[n] \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_L^{rel}[n] \\ y_L^{rel}[n] \\ 1 \end{bmatrix} \tag{4.1}$$

and $\theta_{rel}[n] = \arctan(y_L^{rel}[n]/x_L^{rel}[n])$ is the Leader's direction in $\mathcal{R}_F[n]$. To restate the objective of tracking in terms of the local coordinates, $\mathbb{P}\left(d[n] \leq D_{th}\right) \approx 1 \quad \forall t$ where $d[n] = ||\mathbf{X}_L^{rel}[n]||_2 = (x_L^{rel}[n]^2 + y_L^{rel}[n]^2)^{1/2}$.

4.2 The ARREST System

In this section, we discuss our proposed system solution for RSSI based relative position sensing and tracking. In the ARREST system, the Leader is a robot or a human carrying a device that periodically transmits RF beacons, and the TrackBot is a robot carrying a directional, off-the-shelf RF receiver. As shown in Fig. 4.3, the ARREST architecture consists of three layers: Communication ANd Estimation (CANE), Control And STate update (CAST), and Physical RobotIc ControllEr (PRICE). In order to track the Leader, the TrackBot needs sufficiently accurate estimations of both the Leader's relative position ($\mathbf{X}_L^{rel}$) and relative speed (v_{rel}). Thus, at any time instant n, we define the state of the TrackBot as a 3-tuple: $\mathcal{S}[n] = \left[d^e[n],\ \ v_{rel}^e[n],\ \ \theta_{rel}^e[n]\right]^T$ *where the superscript e refers to the estimated values,* $d^e[n] = ||\mathbf{X}_L^{rel}[n]||_2$ refers to the estimated distance at time n, $v_{rel}^e[n]$ refers to the relative speed of the TrackBot along the X-axis of $\mathcal{R}_F[n]$ with respect to the Leader, and $\theta_{rel}^e[n]$ refers to the angular orientation (in radians) of the Leader in $\mathcal{R}_F[n]$.

CANE: The function of the CANE layer is to measure RSSI values from the beacons and approximate the Leader's position relative to the TrackBot, (i.e., $d^e[n]$ and $\theta_{rel}^e[n]$). The CANE layer is broken down into three modules: Wireless Communication and Sensing, Rotating Platform Assembly, and Relative Position Estimation. At the beginning of each time slot, n, the Wireless Communication and Sensing module and the Rotating Platform Assembly perform a 360° RSSI sweep

by physically rotating the directional antenna while storing RSSI measurements of successful beacon receptions into the vector $\mathbf{r_v}[n]$. The Relative Position Estimation module uses $\mathbf{r_v}[n]$ to approximate the relative position of the Leader by leveraging pre-estimated directional gains of the antenna, detailed in Section 4.3.

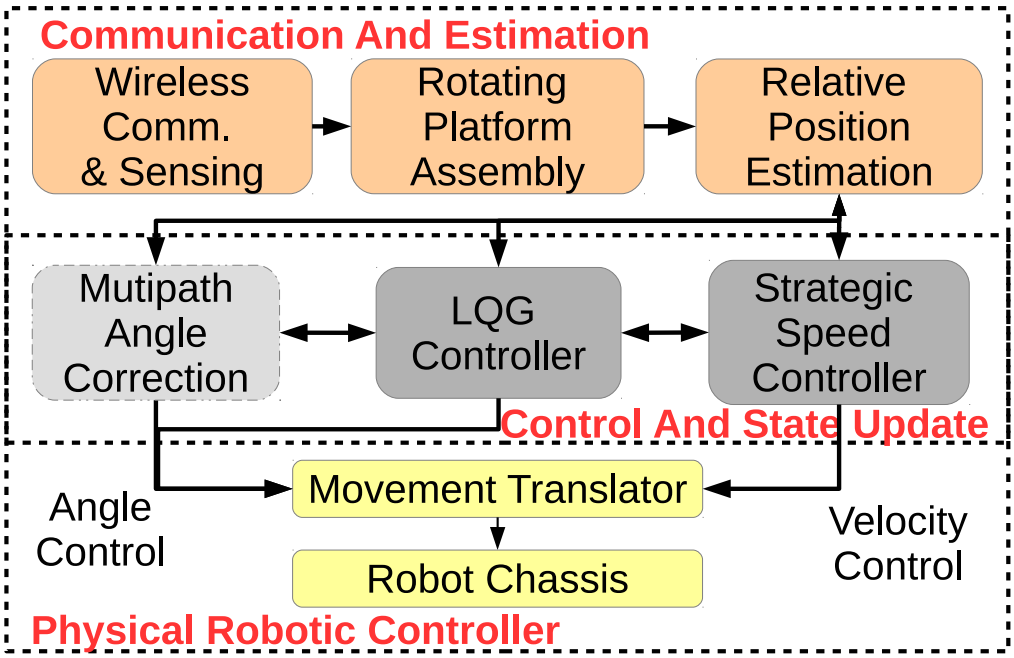


Figure 4.3: The ARREST Architecture

CAST: The functions of the CAST layer is to maintain the 3-tuple state estimates and to generate control commands based on current and past observations to send to the PRICE layer. The CAST layer consists of two main modules: the Linear Quadratic Gaussian (LQG) Controller and the Strategic Speed Controller. *The CAST layer also includes a special, case-specific module called Multipath Angle Correction for severely cluttered environments (explained further in Section 4.5.3.4).*

The Strategic Speed Controller estimates the relative speed of the Leader by exploiting past and current state information and generates the speed control signal in conjunction with the LQG controller. The term *"Strategic"* is used to emphasize that we propose two different strategies, Optimistic and Pragmatic, for the relative speed approximation as well as speed control of the TrackBot (detailed in Section 4.3.3). The LQG controller, detailed in Section 4.2.1, incorporates past state information, past control information, and relative position and speed approximations to (1) generate the system's instantaneous state, (2) determine how much to rotate the TrackBot itself, and (3) determine what should be the TrackBot's relative speed. The state information generated by the LQG controller is directly sent to the Strategic Speed Controller to calculate the absolute speed of the TrackBot.

PRICE: The goal of the PRICE layer is to convert the control signals from the CAST layer into actual translational and rotational motions of the TrackBot. It consists of two modules: Movement Translator and Robot Chassis. The Movement Translator maps the control signals from the CAST layer to a series of platform-specific Robot Chassis motor control signals (detailed in Section 4.4).

4.2.1 Proposed LQG Formulation

In our proposed solution, we first formulate the movement control problem of the TrackBot as a discrete time Linear Quadratic Gaussian (LQG) control problem. An LQG controller is a combination of a Kalman Filter with a Linear Quadratic

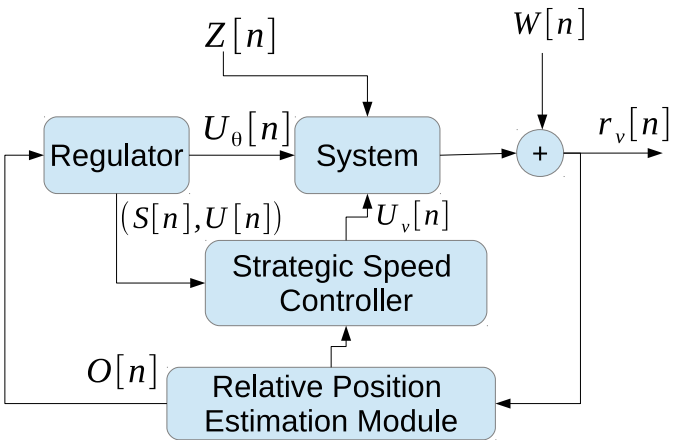


Figure 4.4: Proposed LQG Controller System

Regulator (LQR) that is proven to be the *optimal controller* for linear systems with Additive White Gaussian Noise (AWGN) and incomplete state information [65]. The linear system equations for any discrete LQG problem can be written as:

$$\begin{aligned} \mathcal{S}[n+1] &= A_n\mathcal{S}[n] + B_n\mathbf{U}[n] + \mathbf{Z}[n] \\ \mathbf{O}[n] &= C_n\mathcal{S}[n] + \mathbf{W}[n] \end{aligned} \tag{4.2}$$

where A_n and B_n are the state transition matrices, $\mathbf{U}[n]$ is the LQG control vector, $\mathbf{Z}[n]$ is the system noise, $\mathbf{O}[n]$ is the LQG system's observation vector, C_n is the state-to-observation transformation matrix, and $\mathbf{W}[n]$ is the observation noise at time n. A LQG controller first predicts the next state based on the current state and the signals generated by the LQR. Next, it applies the system observations to update the estimates further and generates the control signals based on the updated state estimates. *In our case,* $\mathbf{O}[n] = \left[d^m[n],\ \ v^m_{rel}[n],\ \ \theta^m_{rel}[n]\right]^T$ *(the superscript m refers to measured values).* Moreover, in our case, the state transition matices

$A_n = A$, $B_n = B$, $C_n = C$ are time invariant and the time horizon is infinite as we do not have any control over the Leader's movements. For a infinite time horizon LQG problem [66], the cost function can be written as:

$$J = \lim_{N\to\infty} \frac{1}{N}\mathbb{E}\left(\sum_{n=0}^{N} \mathcal{S}[n]^T \mathbf{Q}\mathcal{S}[n] + \mathbf{U}[n]^T \mathbf{H}\mathbf{U}[n]\right) \tag{4.3}$$

where $\mathbf{Q} \geq 0, \mathbf{H} > 0$ are the weighting matrices. The discrete time LQG controller for this optimization problem is:

$$\begin{aligned} &\hat{\mathcal{S}}[n+1] = A\hat{\mathcal{S}}[n] + B\mathbf{U}[n] + K(\mathbf{O}[n+1] - C\{A\hat{\mathcal{S}}[n] + B\mathbf{U}[n]\}) \\ &\mathbf{U}[n] = -L\hat{\mathcal{S}}[n] \qquad \text{and} \qquad \hat{\mathcal{S}}(0) = \mathbb{E}(\mathcal{S}(0)) \end{aligned} \tag{4.4}$$

where $\hat{}$ denotes estimates, K is the Kalman gain which can be solved via the algebraic Riccati equation [67], and L is the feedback gain matrix. In our system, the state transition matrix values are as follows:

$$\mathbf{A} = \begin{bmatrix} 1 & -\delta t & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \mathbf{B} = \begin{bmatrix} 0 & -\delta t & 0 \\ 0 & 1 & 0 \\ 0 & 0 & -1 \end{bmatrix} \mathbf{C} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{4.5}$$

where δt is the time granularity for the state update. Ideally, within δt, the TrackBot executes one set of movement control decisions while it also scans RSSI for the next set of control decision (detailed in Sections 4.5.1 and 4.5.2). Note that, to solve

this optimization problem, we also require the covariance data for the noise, i.e., $\Sigma_{WW} = \mathbb{E}(\mathbf{W}\mathbf{W}^T)$, and $\Sigma_{ZZ} = \mathbb{E}(\mathbf{Z}\mathbf{Z}^T)$. We assume the system noise, $Z[n]$, to be Gaussian and the measurement noise, $W[n]$, to be approximated as Gaussian.

Furthermore, we tweak the LQG controller to send out a rotational control signal after a state update and before generating the LQR control signals, $\mathbf{U}[n]$. The rotational control signal rotates the TrackBot assembly by $\theta^e_{rel}[n]$ and sets $\theta^e_{rel}[n] = 0$. This is performed to align the robot toward the estimated direction of the Leader before calculating the movement speed. Thus, we use only the Kalman Filtering part of the LQG controller for angle/orientation control. The reason behind not using the full LQG controller for TrackBot's orientation control lies in the fact that the LQG controller considers a sudden rapid change in direction ($\approx 180°$) as a noise and takes a while to correct the course of the TrackBot. A block diagram of our LQG control system model is presented in Fig. 4.4.

4.3 RSSI Based Relative Position and Speed Observations

In this section, we discuss our methodologies to map the observed RSSI vector, $\mathbf{r_v}[n]$, into the controller observation vector, $\mathbf{O}[n]$.

4.3.1 Distance Observations

The RSSI is well known to be a measure of distance if provided with sufficient transceiver statistics such as the transmitter power, the channel path loss exponent, and the fading characteristics. One of the standard equations for calculating the received power for an omnidirectional antenna is as follows [18]:

$$\begin{aligned} P_{r,dBm} &= P_{t,dBm} + G_{dB} - \mathcal{L}_{ref} - 10\eta \log_{10} \frac{d^m[n]}{d_{ref}} + \psi \\ P^{ref}_{r,dBm} &= P_{t,dBm} + G_{dB} - \mathcal{L}_{ref} + \psi \\ \implies \frac{d^m[n]}{d_{ref}} &\approx 10^{\frac{\left(P^{ref}_{r,dBm} - P_{r,dBm}\right)}{10\cdot\eta}} \end{aligned} \tag{4.6}$$

where $P_{r,dBm}$ is the received power in dBm, $P_{t,dBm}$ is the transmitter power in dBm, G_{dB} is the gain in dB, $\mathcal{L}_{ref}$ is the path loss at the reference distance d_{ref} in dB, η is the path loss exponent, $d^m[n]$ is the distance between the transmitter and receiver, ψ is the random shadowing and multipath fading noise in dB, and $P^{ref}_{r,dBm}$ is the received power at reference distance, d_{ref}. Eqn. (4.6) is also valid for the average received power for a directional antenna with an average gain of G_{dB}. To calculate the received power for a particular direction θ, we just need to replace G_{dB} in (4.6) with the directional gain of the antenna, $G_{dB}(\theta)$. To apply (4.6) in ARREST, the TrackBot needs to learn the channel parameters such as the η, $\mathcal{L}_{ref}$, and d_{ref}. In our proposed system, we assume that the TrackBot has information about the initial distance to the Leader, $d^m(0)$. Furthermore, $G_{dB}(\theta)$ and $P_{t,dBm}$ are known as a part of the system design process. Upon initialization of ARREST,

the TrackBot performs an RSSI scan by rotating the antenna assembly to generate $\mathbf{r_v}(0)$ and harnesses the average received power ($P_{r,dBm}$) information to estimate the environment's η as follows.

$$\eta = \frac{P^{ref}_{r,dBm} - P_{r,dBm}}{10 \log_{10} \frac{d^m(0)}{d_{ref}}} \tag{4.7}$$

Next, the TrackBot applies the estimated η and $P_{r,dBm} = avg\ \{\mathbf{r_v}[n]\}$ on (4.6) to map $\mathbf{r_v}[n]$ to the observed distance to the Leader, $d^m[n]$.

4.3.2 Angle Observations

One of the main components of our ARREST architecture is the observation of the Angle of Arrival (AoA) of RF beacons solely based on the RSSI data, $\mathbf{r_v}[n]$. There exist three different classes of RF-based solutions to determine the AoA. The first class, *antenna array based approaches*, employs an array of antennas to determine the AoA by leveraging the phase differences among the signals received by the different antennas [137]. The main difficulty of implementing this class is that very few multi-antenna off-the-shelf radios provide access to the phase information. The second class, *multiple directional antenna based approaches*, employs at least two directional antennas oriented in different directions [138] to determine AoA. In this class, the differences among RSSI values from all antennas are utilized to determine the AoA. However, utilizing current off-the-shelf antenna arrays or multiple directional antennas increases the cost, form factor, and complexity of a

TrackBot implementation. We avoid the multiple directional antenna based option also because it requires separate radio drivers for each antenna as well as proper time synchronizations. Thus, we develop methods contributing to the third class of solutions, which is the use of a single, rotating antenna and the knowledge of the antenna's directional gain pattern to approximate the AoA of RF beacons. The core of these methods, called *pattern correlation*, is to correlate the vector of RSSI measurements, $\mathbf{r_v}[n]$, with another vector representing the antenna's known, normalized gain pattern, $\mathbf{g_{abs}}$. At the beginning of each time slot n, the TrackBot performs a 360° sweep of RSSI measurements to generate the vector, $\mathbf{r_v}[n]$. Then, $\mathbf{r_v}[n]$ is normalized: $\mathbf{g_m} = \mathbf{r_v}[n] - \max(\mathbf{r_v}[n])$. The TrackBot also generates different θ shifted versions of $\mathbf{g_{abs}}(\theta)$ as follows.

$$\begin{aligned}
\mathbf{r_v}[n] &= [r_{-180}, r_{-178.2}, \cdots, r_{-1.8}, r_0, r_{1.8}, \cdots r_{178.2}] \\
\mathbf{g_m} &= [r'_{-180}, r'_{-178.2}, \cdots, r'_{-1.8}, r'_0, r'_{1.8}, \cdots r'_{178.2}] \\
\mathbf{g_{abs}}(\theta) &= [g_{(-180+\theta)}, \cdots, g_{(0+\theta)}, \cdots, g_{(178.2+\theta)}]
\end{aligned} \tag{4.8}$$

where r_ϕ refers to the RSSI measurement, g_ϕ refers to the antenna gain, and $r'_\phi = r_\phi - \max\{\mathbf{r_v}\}$ refers to the observed gain for the antenna orientation of ϕ° with respect to the X-axis of $\mathcal{R}_F[n]$. The step size of 1.8° is chosen based on our hardware implementation's constraints. *Thus, the possible antenna orientations (ϕ) are limited to* $\Theta = \{-180, \cdots, -1.8, 0, \cdots, 178.2\}$. Next, the TrackBot employs different pattern correlation methods for the AoA observation. Below, we

describe three methods in increasing order of complexity. The first method was originally demonstrated by [75]. Through real-world experimentation, we develop two additional improved methods.

4.3.2.1 Basic Correlation Method

The first method (originally demonstrated by [75]) of determining AoA correlates $\mathbf{g_m}$ with $\mathbf{g_{abs}}(\theta)$ $\forall\theta \in \Theta$ and calculates the respective L_2 distances. The observed AoA is the θ at which the L_2 distance is the smallest:

$$\theta^m_{rel} = \arg\min_{\theta\in\Theta} \sum_{k\in\Theta} \omega_k \cdot ||r'_k - g_{(k+\theta)}||_2 \cdot \mathbb{I}_{r'_k} \tag{4.9}$$

where the indicator function $\mathbb{I}_{r'_k}$ indicates whether the sample r'_k exists or not to account for missing samples in real experiments, and $\omega_k = 1$ is a constant.

4.3.2.2 Clustering Method

While the first method works well if enough uniformly distributed samples (≥ 100 in our implementation) are collected within the 360° scan, it fails in scenarios of sparse, non-uniform sampling due to packet loss. In real experiments (mainly indoors), the collected RSSI samples can be uniformly sparse or sometimes batched sparse (samples form clusters with large gaps ($\approx 30°$) between them).

Definition 4.3.1. *An angular cluster (Λ) is a set of valid samples for a contiguous set of angles:* $\Lambda = \{k|\mathbb{I}_{r'_k} = 1 \forall k \in \{\phi_f, \phi_f + 1.8, \cdots, \phi_l\} \subset \Theta\}$.

To prevent undue bias from large cardinality clusters that can cause errors in estimating the correlation, we assign a weight (ω_k) to each sample (k) and use the pattern correlation method as in Eqn. (4.9). In our weighting scheme, we assign $\omega_k = \frac{1}{|\Lambda|}$ where $k \in \Lambda$.

4.3.2.3 Weighted Average Method

Based on real-world experiments, we find that the angle observation based on the basic correlation method, say θ_m^1, gives reasonable error performance if the average cluster size, denoted by λ_a, is greater than the average gap size between clusters, μ_a. Conversely, the angle observation based on the clustering method, say θ_m^2, is better if $\lambda_a << \mu_a$. Thus, as a trade-off between both the basic correlation method and the clustering method, we propose a weighted averaging method described below.

$$\theta_{rel}^m = \begin{cases} \frac{\lambda_a}{\mu_a} \cdot \theta_m^1 + (1 - \frac{\lambda_a}{\mu_a}) \cdot \theta_m^2 & \text{if } \lambda_a \leq \mu_a \\ \theta_m^1 & \text{if } \lambda_a > \mu_a \end{cases} \tag{4.10}$$

In the rest of the study, we use the weighted average method for angle observations.

4.3.3 Speed Observations

To fulfill the tracking objective, the TrackBot needs to adapt its speed of movement ($v_F[n]$), according to the Leader's speed ($v_L[n]$). In our ARREST architecture, the Strategic Speed Controller uses the relative position observations ($d^m[n], \theta_{rel}^m[n]$)

from the CANE layer and the past LQG state estimates to determine the current relative speed, $v^m_{rel}[n]$, as well as the Leader's speed, $v^m_L[n]$. In this context, we employ two different observation strategies. The first strategy, which we refer to as the *Optimistic strategy*, assumes that the Leader will be static for the next time slot and determines the relative speed as follows:

$$\begin{aligned} v^m_{rel}[n] &= v^e_{rel}[n] - \frac{(d^m[n] - d^e[n] \cdot \cos\theta^m_{rel}[n])}{\delta t} \\ v^e_L[n+1] &= 0 \end{aligned} \tag{4.11}$$

On the other hand, the *Pragmatic Strategy* assumes that the Leader will continue traveling at the observed speed, $v^m_L[n]$. This strategy determines the relative speed as follows:

$$\begin{aligned} v_L[n] &= \frac{\left((d^e[n] - d^m[n] \cdot \cos\theta^m_{rel}[n])^2 + (d^m[n] \cdot \sin\theta^m_{rel}[n])^2\right)^{1/2}}{\delta t} \\ \theta_v[n] &= \arctan\frac{d^m[n] \cdot \sin\theta^m_{rel}[n]}{d^m[n] \cdot \cos\theta^m_{rel}[n] - d^e[n]} - \theta^m_{rel}[n] \\ v^e_L[n+1] &= v^m_L[n] = v_L[n] \cdot \cos(\theta_v[n]) \\ v^m_{rel}[n] &= v_F[n] - v^m_L[n] \end{aligned} \tag{4.12}$$

For an illustration of different components of this process, please refer to Fig. 4.5. Next, the LQG controller uses the observation vector $\mathbf{O}[n]$ to decide the next state's relative speed, $v^e_{rel}[n+1]$ which is used by the Speed Controller to generate the TrackBot's actual speed for next time step, $v_F[n+1] = v^e_L[n+1] + v^e_{rel}[n+1]$. Note that the speed of the TrackBot, $v_F[n]$, is exactly known to itself at any time n. In

addition to the different assumptions about the Leader's speed, the two strategies also differ in how the noise is modeled in the correlation between distance and speed estimations: the Optimistic Strategy assumes that the noise in speed observations are uncorrelated with the noise in distance observations, whereas the Pragmatic strategy assumes strong correlation between distance and speed estimation noise. We compare the performance of both strategies based on emulation and real world experiments in Sections 4.5.1.1 and 4.5.2.1, respectively.

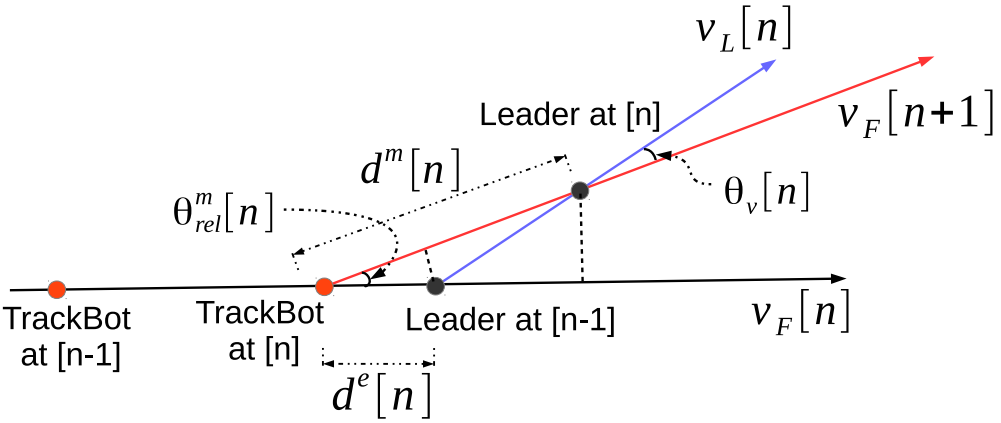


Figure 4.5: Illustration of Different Components for Relative Speed Observation

4.4 TrackBot Prototype

4.4.1 Hardware

We implemented a TrackBot with our ARREST architecture inside a real, low-cost robot prototype presented in Fig. 4.1. For a concise description of our prototype, we list the hardware used for implementation of each of the ARREST components

in Table 4.1. We also discuss details of the Time Difference of Arrival (TDOA) based localization system integrated with our ARREST architecture for ground truth estimation separately in Section 4.5.3.

Table 4.1: ARREST Hardware Implementation

	Module	Hardware
CANE	Wireless Communication and Sensing	OpenMote[139]; Rosewill Directional Antenna (Model RNX-AD7D)
	Rotating Platform Assembly	Nema 17 (4-wire bipolar Stepper Motor); EasyDriver - Stepper Motor Driver; mbed NXP LPC1768 [140]
	Relative Position Estimation	mbed NXP LPC1768 [140]
CAST		mbed NXP LPC1768 [140]
PRICE	Movement Translator	mbed NXP LPC1768 [140]
	Robot Chassis	Baron-4WD Mobile Platform, L298N Stepper Motor Driver Controller Board, HC-SR04 Ultrasonic Sensor [141]

OpenMote [139]	TI 32-bit CC2538 @ 32 MHz with 512KB Flash memory, 32KB RAM, 2.4GHz IEEE 802.15.4-based Transceiver connected via SMA plug
mbed NXP-LPC1768 [140] μ-processor	32-bit ARM Cortex-M3 core @ 96MHz, 512KB FLASH, 32KB RAM; **Interfaces:** built-in Ethernet, USB Host and Device, CAN, SPI, I2C, ADC, DAC, PWM and other I/O interfaces
Rosewill RNX-AD7D Directional Antenna	**Mode 1: Frequency:** 2.4GHz, **Max Gain:** 5dBi, **HPBW:** 70° **Mode 2: Frequency:** 5GHz, **Max Gain:** 7dBi, **HPBW:** 50°
Nema 17 Stepper Motor	**Dimension:** $1.65'' \times 1.65'' \times 1.57''$, **Step size:** 1.8 degrees (200 steps/rev), **Rated current:** 2A, **Rated resistance:** 1.1 Ohms
HC-SR04 [141]	**Operating Voltage:** 5V DC, **Operating Current:** 15mA, **Measure Angle:** 15°, **Ranging Distance:** 2cm - 4m

In the TrackBot prototype, the directional antenna and the OpenMote are mounted on top of a stepper motor using a plate. While we use two microprocessors (the OpenMote and the mbed), the system can be implemented using one microprocessor. We choose to use two in this prototype to work around wiring issues and work around the lack of sufficient GPIO pins on the OpenMote. The

OpenMote is only used for RF sensing while the mbed is used to implement the rest of the ARREST modules. For programming of the mbed, we use the mbed Real-Time Operating System [142]. The mbed sends control signals to the stepper motor to rotate it in precise steps of 1.8°. *Each consecutive* 360° *antenna rotations alternate between clockwise and anti-clockwise because this: (1) prevents any wire twisting between the mbed and OpenMote and (2) compensates for the stepper motor's movement errors.* The mbed communicates with other H/W components via GPIO pins and High-level Data Link Control (HDLC) Protocol [143] based reliable serial line communication.

In the current prototype, the maximum speed of the robot is $30cm/s$. Due to synchronization issues on the mbed when trying to simultaneously rotate the antenna and move the robot chassis, the antenna assembly sometimes does not return to its initial position after a complete rotation. To solve this issue while avoiding complex solutions (e.g., via a feedback-based offset control mechanism), the TrackBot instead first performs an RSSI scan and then moves the chassis. Ideally, the antenna can rotate 360° in $1s$ while collecting 200 samples. However, we choose to slow the scan down to a duration of $2s$ to cope with the occasional occurrence of sparse RSSI samples. Moreover, to keep the movement simple, the TrackBot first rotates to the desired direction and then moves straight with the desired speed. The wheels of the robot are controlled using PWM signals from the mbed with a period of $2s$. We choose a $2s$ period for robot rotation as one $2s$

pulse width equates to a chassis rotation amount of $\approx 180°$. We also choose the same period length ($2s$) for forward movement which caps the speed of the robot at $60/6 = 10cm/s$ (including $2s$ of RSSI scan). The whole system is powered by five AA batteries which can run for a total of $\approx 3-4$ hours. We also implemented a very simple obstacle avoidance mechanism by employing a single HC-SR04 rangefinder in the front bumper of the chassis and protection bumpers on the other sides. While moving forward, if the ultrasound detects an object at a distance less than 10cm, it stops the TrackBot's movement immediately.

The Leader node is currently implemented as an OpenMote transmitting beacons with the standard omnidirectional antenna and a transmit power of $7dBm$. For programming of the OpenMotes, we use the RIOT operating systems [144, 145]. The Leader implementation is capable of transmitting 200 packets/second.

4.4.2 ARREST System Parameter Setup

4.4.2.1 Cost Parameters Setup

In the cost function of our LQG formulation, the matrix $\mathbf{Q}$ is a 3×3 positive definite diagonal matrix: $\mathbf{Q} = diag\{Q_d, Q_v, Q_\theta\}$. Our main goal is to keep the distance as well as the relative angle to be as low as possible while keeping emphasis on the distance. From this perspective, we perform a set of experiments to find a good trade-off between Q_v, Q_θ, and Q_d where we vary one parameter while keeping the rest of them fixed. Based on these experiments, we opt for the following settings:

$Q_v = 0.1$, $Q_\theta = 1$, and $Q_d = 10 \cdot v_L^{max}$ where v_L^{max} is the maximum speed of the Leader. With these settings, our system performs better than any other explored settings. Furthermore, $\mathbf{H}$ is chosen to be a 3×3 Identity matrix.

4.4.2.2 Noise Covariance Matrix Parameters Setup

In our implementation, the system noises are assumed to be i.i.d normal random variables with Σ_{ZZ} being a 3×3 identity matrix. On the other hand, the observation noise covariance matrix requires separate settings for the different strategies. For the Optimistic strategy, we assume that the observation noises are uncorrelated, whereas, for the Pragmatic strategy, the distance estimation errors and the relative speed estimation errors are highly correlated with variances proportional to v_L^{max}. A set of empirically determined values of Σ_{WW} for the Optimistic and the Pragmatic strategies are as follows.

$$\Sigma_{WW}^{Op} = \begin{bmatrix} 4 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \qquad \Sigma_{WW}^{Pg} = \begin{bmatrix} 1 & v_L^{max} & 0 \\ v_L^{max} & (v_F^{max})^2 & 0 \\ 0 & 0 & 0.1 \end{bmatrix} \tag{4.13}$$

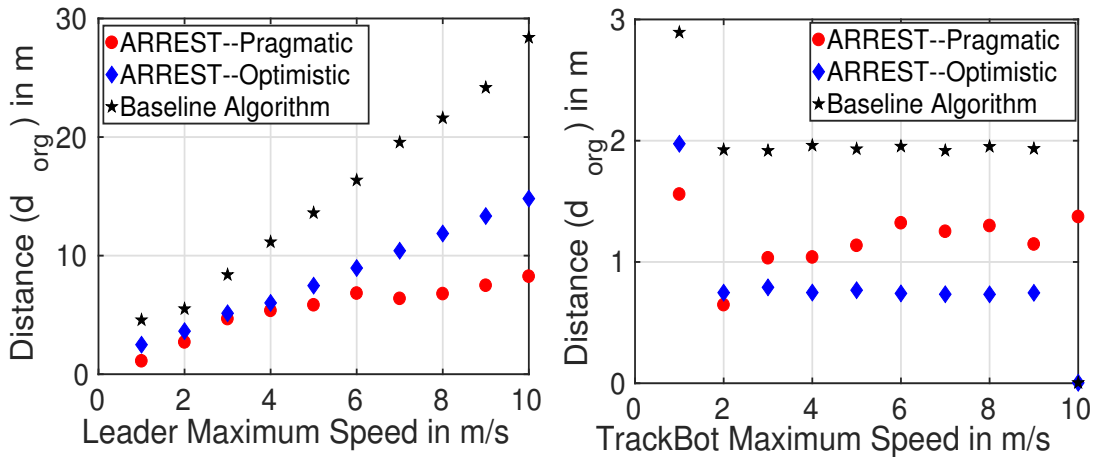
where Op and Pg refers to the Optimistic and the Pragmatic strategies, respectively.

4.5 Experiments and Performance Analysis

4.5.1 Baseline Analysis via Emulation

In this section, we perform a thorough evaluation and setup the different parameters such as the LQG covariance matrix (discussed in Section 4.4.2) of the ARREST architecture via a set of emulation experiments. We use the emulation experiment results as a baseline for our real-world experiments.

We employ our hardware prototypes, discussed in Section 4.4, to collect sets of RSSI data in cluttered indoor and outdoor environments for a set of representative distances, $\mathcal{D}$, and angles Θ. Next, we use the collected samples to interpolate the RSSI samples for any random configuration $\mathcal{C} = (d, \theta_{rel})$, where $d \in \mathbb{R}^+$ and $\theta_{rel} \in [-180, 180)$, as follows: $r^e = r^s - 10 \cdot \eta \cdot \log_{10}(d/d_{near}) + \mathcal{N}(0, \sigma^2)$, where r^s is a random sample for configuration $\mathcal{C}_{near} = (d_{near}, \theta_{near})$ such that $d_{near} = \arg\min_{d_i \in \mathcal{D}} |d_i - d|$ and $\theta_{near} = \arg\min_{\theta_i \in \Theta} |\theta_i - \theta_{rel}|$. Note that we add an extra noise of variance $\sigma^2 = 2$ on top of the noisy samples (with $\sigma^2 \approx 4$) for configuration $\mathcal{C}_{near}$. To estimate the η, we use (4.7) to calculate η_{ij} for each pair of distances, $d_i, d_j \in \mathcal{D}$ and take the average of them. We choose a value of $\delta t = 1s$ in (4.5) to match the maximum achievable speed of our stepper motor as, ideally, the interval between any two consecutive movement control decisions could be $1s$ where the TrackBot carries out any movement control decision within the respective $1s$ interval.



(a) Varying Leader Speed

(b) Varying TrackBot's Speed

Figure 4.6: (a)-(b)Tracking Performance Comparison Among Different Speed Estimation Strategies

4.5.1.1 The Optimistic Strategy vs. The Pragmatic Strategy

In this section, we compare the performance among the two proposed strategies, Optimistic and Pragmatic, and a Baseline algorithm. In the *Baseline algorithm*, the TrackBot estimates the relative position via the basic correlation method (discussed in 4.3.2.1). Once the direction is determined, the TrackBot rotates to align itself toward the estimated direction and then moves with a speed of $\min\{v_F^{max}, \frac{d^e[n]}{\delta t}\}$. In Fig. 4.6a, we compare the average distance between the TrackBot and the Leader for varying v_L^{max} while setting $v_F^{max} = 1.8 \cdot v_L^{max}$. Figure 4.6a clearly demonstrates that the Pragmatic strategy performs better than the Optimistic strategy as well as the Baseline algorithm, due to adaptability and accuracy of the speed information. The poor performance of the Optimistic strategy is due to its indifference towards the actual speed of the Leader which causes the TrackBot to lag behind for higher velocities. Conversely, we compare the average distance between the

Leader and the TrackBot for varying v_F^{max}, while the Leader's maximum speed is fixed at $v_L^{max} = 1m/s$. The experiment outcomes, presented in Fig. 4.6b, show that the performance of both strategies are comparable, while the Optimistic strategy outperforms the Pragmatic strategy for $v_F^{max} \geq 3 \cdot v_L^{max}$. The reason behind this is the Leader is constantly changing movement direction while the TrackBot always travels along the straight line joining the last estimated position of the Leader and the TrackBot which may not be the same as the Leader's direction of movement. This results in oscillations in the movement pattern for the Pragmatic strategy while the Optimistic strategy avoids oscillations since it assumes the Leader to be static. The worst performance of the Baseline approach is attributed to lack of speed adaptation by taking past observation into account.

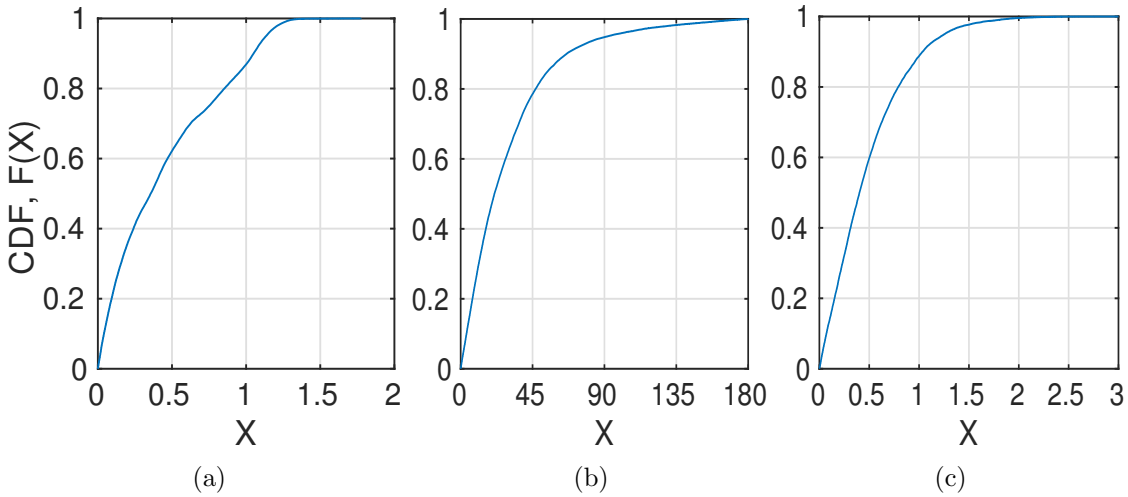
One more noticeable fact from Fig. 4.6b is that if $v_F^{max} = v_L^{max}$, the tracking performance is the worst. This is quite intuitive because for this speed configuration, the TrackBot is unable to compensate for any error or initial distance while the Leader constantly moves at a speed close to v_L^{max}. Thus, the relative speed needs to be positive for proper tracking. In order to find a lower bound on the TrackBot's speed requirement, we perform another set of experiments by varying v_F^{max} from v_L^{max} to $3 \cdot v_L^{max}$. Based on the results, we conclude that for $v_F^{max} \leq 1.6 \cdot v_L^{max}$, the tracking system fails and the distance increases rapidly. On the other hand, for $v_F^{max} > 1.6 \cdot v_L^{max}$ the performance remains the same. Thus, in our experimental setup, we opt for $v_F^{max} = 1.8 \cdot v_L^{max}$.

4.5.1.2 Absolute Distance Statistics

One main focus of our ARREST architecture is to guarantee $\mathbb{P}\left(||\mathbf{X}_L[n] - \mathbf{X}_F[n]||_2 \leq D_{th}\right) \approx 1 \ \ \forall n$. The value of D_{th} could be chosen as a function of v_L^{max}. However, according to our target application context, we select $D_{th} = 5m$ as we consider a distance more than 5 meters to be large enough to lose track in an indoor environment. With this constraint, we find that our present implementation of the ARREST system fails in the tracking/following objective if the Leader moves faster than 3m/s. In order to verify whether our ARREST architecture can guarantee the distance requirement for Leader with $v_L^{max} \leq 3m/s$, we perform a set of emulations with $\delta t = 1s$, where the Leader travels along a set of random paths. In all cases, the instantaneous distances between the TrackBot and the Leader during the emulation are less than $5m$ with probability ≈ 1. The nonzero probability of distances higher than $5m$ is due to randomness in the Leader's motion including a complete reversal of movement direction.

4.5.1.3 Estimation Errors

In order to learn the statistics of different estimation errors, we perform a range of emulation experiments, where the Leader follows a set of random paths and $v_L^{max} \leq 3m/s$. In Fig. 4.7a, we plot the empirical CDF of the absolute errors in the distance estimates maintained by our system. Figure 4.7a clearly illustrates that the instantaneous errors are less than $100cm$ with very high probability ($\approx 90\%$) and



(a) (b) (c)

Figure 4.7: Emulation Based Performance: (a) Absolute Distance Estimation Errors (in m), (b) Absolute Angle Estimation Errors (in degrees), and (c) Absolute Speed Estimation Errors (in m/s)

that the absolute error values are bounded by $1.5m$. These statistics are reasonable for pure RSSI-based estimation systems (explained further in Section 4.5.4.1). We also plot the CDF of the absolute angle estimation errors over the duration of the emulations in Fig. 4.7b. It can be seen that the absolute angle errors are less than 40° with high ($\approx 80\%$) probability, which is justified as the Half Power Beam Width (HPBW) for the antenna we are using is approx 70°. Further improvements may be possible by using an antenna with greater directionality or other radios (such as UWB radios). The non-zero probability of the angle error being more than 40° is again due to the random direction changes in the Leader's movements. Similarly, we analyze the absolute speed estimation errors in terms of CDF, illustrated in Fig. 4.7c. The absolute errors in the speed estimations of the Leader are less than $1m/s$ with $\approx 90\%$ probability.

Table 4.2: Summary of Emulation Results

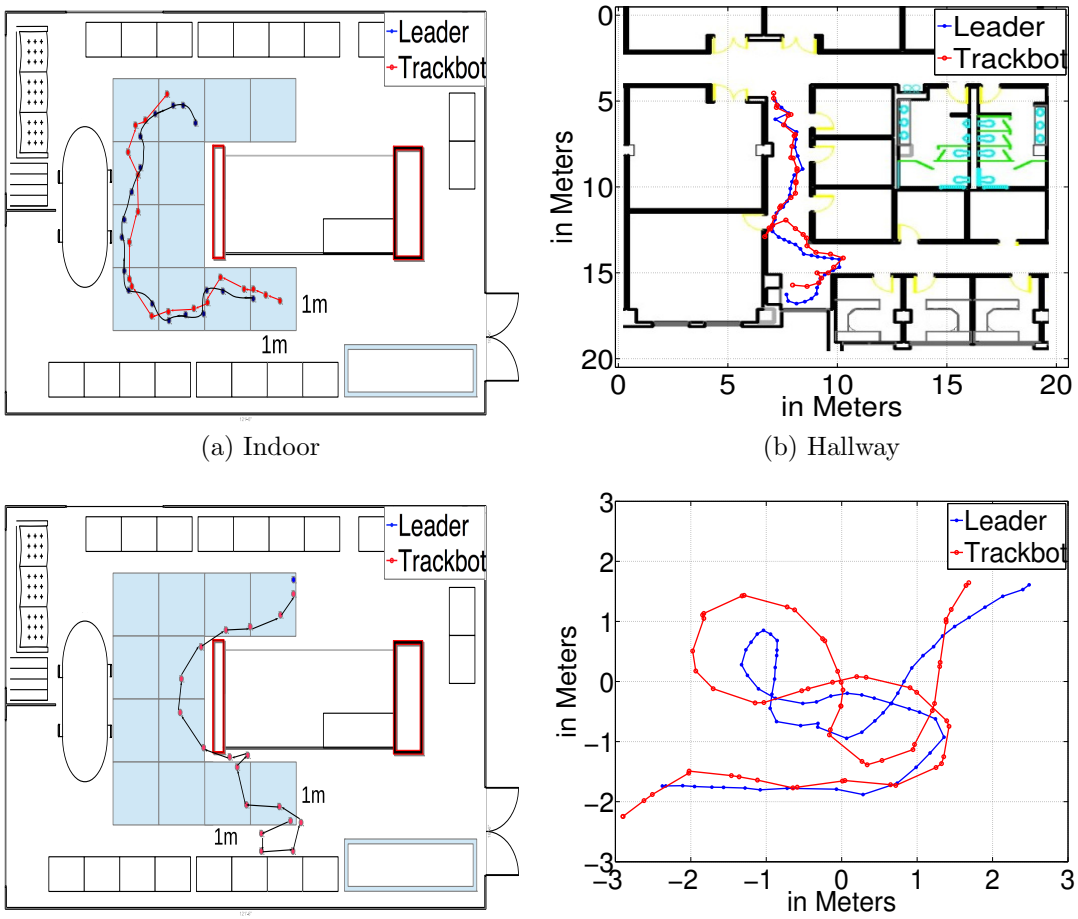
☐ Pragmatic Strategy performs best for $1.6 \cdot v_L^{max} < v_F^{max} < 3 \cdot v_L^{max}$ while Optimistic Strategy performs best for $v_F^{max} \geq 3 \cdot v_L^{max}$
☐ The ARREST system fails if $v_L^{max} > 3m/s$.
☐ For $v_L^{max} \leq 3m/s$ and $v_F^{max} = 1.8 \cdot v_L^{max}$, the TrackBot stays within $5m$ of the Leader with probability $\approx 100\%$.
☐ Absolute distance estimation errors are $< 100cm$ with probability $\approx 90\%$ and $< 150cm$ with probability $\approx 100\%$.
☐ Absolute angle estimation errors are $< 40°$ with probability $\approx 80\%$.
☐ Absolute speed estimation errors are less than $1m/s$ with probability $\approx 90\%$.

4.5.2 Real Experiment Results: Small Scale

To analyze the performance of the ARREST architecture, we use the TrackBot prototype to perform a set of small-scale experiments, followed by a range of large-scale experiments. In this section, we present the results of our small-scale real-world experiments.

Based on the valuable insights from the emulation results, we choose TrackBot's speed to be at least 1.8X the Leader's speed. The TrackBot makes a decision every $6s$. Between each decision, the TrackBot takes $2s$ for both the antenna rotation and RSSI scan, $2s$ for the chassis rotation, and $2s$ for the chassis translation. However, in the state update equations, $\delta t = 4s$ because the actual chassis movement takes place for only $4s$. With this setup, we perform a set of real tracking experiments in three different environments:

☐ A cluttered office space, illustrated in Fig. 4.8a ($\approx 10m \times 6m$), with a lot of office desks, chairs, cabinets, and reflecting surfaces.



(a) Indoor

(b) Hallway

(c) Indoor (No Line of Sight)

(d) VICON System

Figure 4.8: Full Path Traces from Small Scale Real World Experiments

☐ A hallway, illustrated in Fig. 4.8b ($\approx 18m$ long and $3m$ wide), with pillars as well as sharp corners.

☐ A VICON camera localization [146] based robot experiment facility, illustrated in Fig. 4.8d ($\approx 6m \times 6m$).

For the first two environments, we use manual markings on the floor to localize both the Leader and the TrackBot. For the last environment, the VICON facility provides us with camera-based localization at millimeter scale accuracy. We

perform a set of experiments in each of these environments for an approximate total period of one month with individual run lasting for 30 minutes during different times of the day. For these experiments, the Leader is a human carrying an OpenMote transmitter.

4.5.2.1 The Optimistic Strategy vs. The Pragmatic Strategy

Similar to our emulation based analysis, we perform a real system based comparison of the proposed speed adaptation strategies as well as the *Baseline Algorithm* (introduced in Section 4.5.1.1). However, in this set of experiments, we do not vary the maximum speed of the TrackBot or the Leader due to prototype hardware limitations. Instead, we compare the absolute distance CDF statistics of these three strategies in Fig. 4.9a for $v_F^{max} = 10cm/s$ and $v_F^{max} = 1.8 \cdot v_L^{max}$. Figure 4.9a validates that Pragmatic strategy performs best among all three strategies when $v_F^{max} = 1.8 \cdot v_L^{max}$. Moreover, the baseline strategy performs the worst due to lack of speed adaptation as well as lack of history incorporation. In summary, our real experiment based results concur with the emulation results.

4.5.2.2 Estimation Errors

To analyze the state estimation errors in our ARREST architecture similar to the emulations, we perform a range of prototype based experiments, where the $v_F^{max} = 1.8 \cdot v_L^{max}$ and the Leader follows a set of random paths. In Fig. 4.9b, we plot the empirical CDF of the absolute errors in the distance estimates maintained

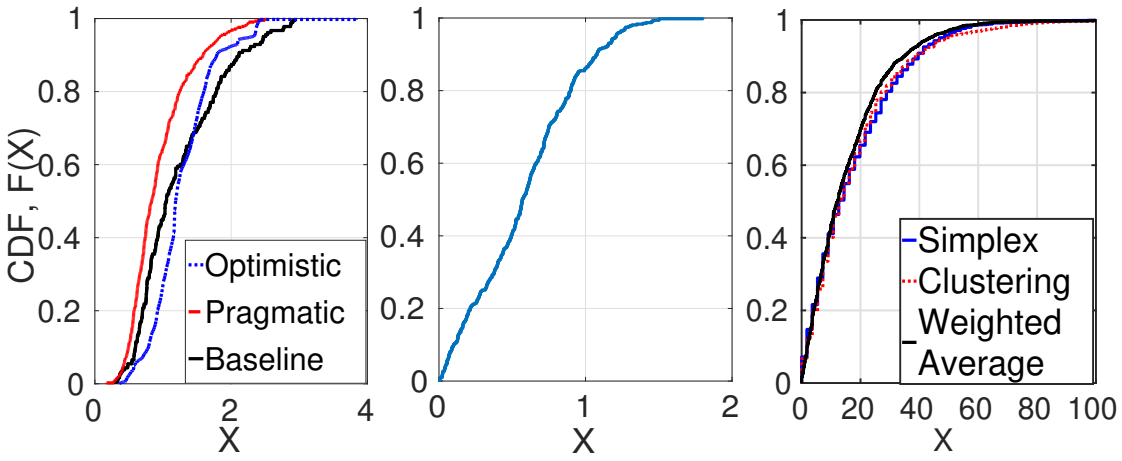


Figure 4.9: Real Experiment Based Performance for Small Scale: (a) Absolute Distance in Meters, (b) Absolute Distance Estimation Error in Meters, and (c) Absolute Angle Estimation Error in Degrees

by our TrackBot. Figure 4.9b clearly illustrates that the instantaneous absolute errors in our distance estimates are $\leq 100cm$ with very high probability ($\approx 90\%$), and are bounded by $1.5m$. These statistics are also reasonable for pure RSSI based estimation systems and concur with the emulation results. Next, in Fig. 4.9c, we compare the angle estimation error performance of the TrackBot for all three AoA observation methods introduced in Section 4.3.2 where we intentionally introduce random sparsity in the RSSI measurements. Figure 4.9c illustrates that our proposed *clustering method* and *weighted average method* perform significantly better than the *basic correlation method* which is expected since the first two take into account the clustered sparsity (Detailed in Section 4.3.2). The instantaneous absolute angle errors are less than 40° with high probability ($\approx 90\%$) for all three

methods which is justified because the HPBW specification for the antenna is approx 70°. Figure 4.9c also illustrates that the weighted angle observation method slightly outperforms the clustering method for AoA observation. The apparent similarity between the performance of the clustering method and the weighted average method is attributed to the consistent lower cluster sizes compared to the gap sizes ($\lambda_a << \mu_a$) in our experiments.

4.5.2.3 Tracking Performance

In Fig. 4.8a, we present a representative path trace from the experiments in the indoor scenario. Similarly, in Fig. 4.8b we present a real experiment instance in the Hallway. Lastly, Fig. 4.8d illustrates an example trace from the VICON system. All three figures illustrate that our system performs quite well in the respective scenarios and stays within $\approx 2m$ from the Leader for the duration of the experiments. These results suggest that our system works equally well in different environments: cluttered and uncluttered. To verify that further, we perform a set of experiments with a static Leader not in the line of sight of the TrackBot for $\geq 50\%$ of the TrackBot's path. *Our TrackBot was able to find the Leader in* 75% *of such experiments.* In Fig. 4.8c, we present one instance of such experiment. The main reason behind this success lies in the TrackBot's ability to leverage a good multipath signal (if exists). In absence of a direct line of sight, the TrackBot first follows the most promising multipath component and by doing so it eventually comes in a line of

sight with the Leader and follows the direct path from that point on. *In most of these experiments ($\geq 90\%$), the TrackBot travels a total distance of less than 2X the distance traveled by the Leader. This implies that our system is efficient in terms of energy consumption due to robotic maneuvers.*

Nonetheless, these small real-world experiments also point out that our current system does not work if there exists no strong/good multipath signal in NLOS situations where "strong multipath" implies that one multipath signal's power is significantly higher than other multipath signals. We detail multipath related problems and our method of partly circumventing it in Section 4.5.3.4.

Table 4.3: Summary of Small Scale Real-World Experiments

☐ Pragmatic Strategy performs best for $1.8 \cdot v_L^{max} = v_F^{max}$.
☐ Absolute distance estimation errors are $< 100cm$ with probability $\approx 90\%$ and $< 150cm$ with probability $\approx 100\%$.
☐ Absolute angle estimation errors are $< 40^\circ$ with probability $\approx 90\%$.
☐ Weighted average AoA observation method performs the best.
☐ The TrackBot stays within $2m$ of the Leader with probability $\approx 98\%$ in line of sight contexts.
☐ The ARREST system works with probability $\approx 75\%$ for NLOS contexts, although it fails if no "strong multipath" exists.

4.5.3 Real Experiment Results : Large Scale

The small-scale experiments, presented in Section 4.5.2, were limited in terms of deployment region (≤ 60 sq. meters) due to the dimensions of the VICON system

and the effort plus time required for large-scale experiments with manual measuring/markings. To perform large scale and long duration experiments based evaluations, we integrated a version of a well known Time Difference of Arrival based localization [147] ground truth system in our TrackBot. This helped us avoid the need for tedious manual markings and measurements. For more efficient experiments, we also developed a robotic leader (illustrated in Figure 4.10), which we will refer to as the *LeaderBot* in this section, to act as both Leader as well as the reference node for the TDoA localization system.

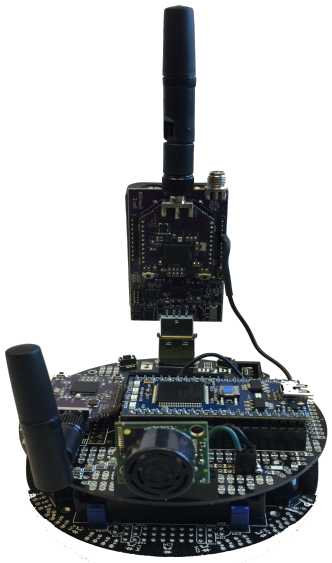


Figure 4.10: Illustration of a 3pi LeaderBot. This robot is used as the leader robot as well as the TDoA localization anchor for large-scale experiments.

The main idea behind TDoA systems is to use a reference node that transmits two different types of signals, say RF and Ultrasound, simultaneously. Now, the localizing/receiver node receives these two signals at different instances of time due to the propagation speed difference between RF and Ultrasound, say Δc. With

proper timestamps, the receiver can now calculate the time difference of arrival of these two signal, say Δt, to estimate the distance as $\Delta c \cdot \Delta t$. We extend this concept slightly further by placing both the receiver RF antenna and the ultrasound on the TrackBot's rotating platform. We rotate the platform in steps of 18° (just a design choice) and perform TDoA based distance estimation for each orientation of the assembly. The TDoA system returns a valid measurement if and only if the assembly is oriented toward a direct line of sight or a reflected signal path. Assuming that there exists a line of sight, the orientation with the smallest TDoA corresponds to the actual angle between the LeaderBot and the TrackBot, and value of the smallest TDoA corresponds to the distance.

4.5.3.1 LeaderBot and TDoA Ranging

The LeaderBot is built upon the commercially available small Pololu 3pi robot [148]. *In our LeaderBot, we use two Openmotes: one Openmote acts as the Leader beaconer* (Beacon Mote) *and operates on 802.15.4 channel 26; the other Openmote* (Range Mote) *is used to remotely control the 3pi robot's movements and to perform the TDoA based localization on 802.15.4 channel 25.* We use two Openmotes for cleaner design as well as to avoid operation interference between remote controlling and beaconing. We use a MB 1300 XL-MaxSonar-AE0 [149] as the ultrasound beaconer, powered by the 3pi robot. The LeaderBot is illustrated in Fig. 4.10. On the TrackBot, we also add a MB 1300 XL-MaxSonar-AE0 [149] ultrasound on the

rotating platform along side with the directional antenna to receive the ultrasound beacons. In these experiments, the Openmote on the TrackBot switches between *Tracking mode* and the *Ranging mode* for ground truth estimation by switching its operating threads as well as the Openmote channel. Step by step method of ranging is as follows.

1. Before ranging, the TrackBot and the LeaderBot finish up their last movement step and stops.

2. TrackBot switches channel from 26 (Tracking channel) to 25 (Ranging Channel).

3. TrackBot's Openmote sends a ranging request (REQ) packet to the Leader's RangeMote.

4. Upon receipt of the REQ packet, the RangeMote and the LeaderBot prepares for ranging by temporarily switching off the remote control feature and sends a Ready (RDY) packet to the TrackBot.

5. Upon receiving RDY packet, the TrackBot's Openmote turns ON the ultrasound-RF ping receiving mode by setting some flags in the MAC layer to prepare for interception of the packet and sends a GO packet.

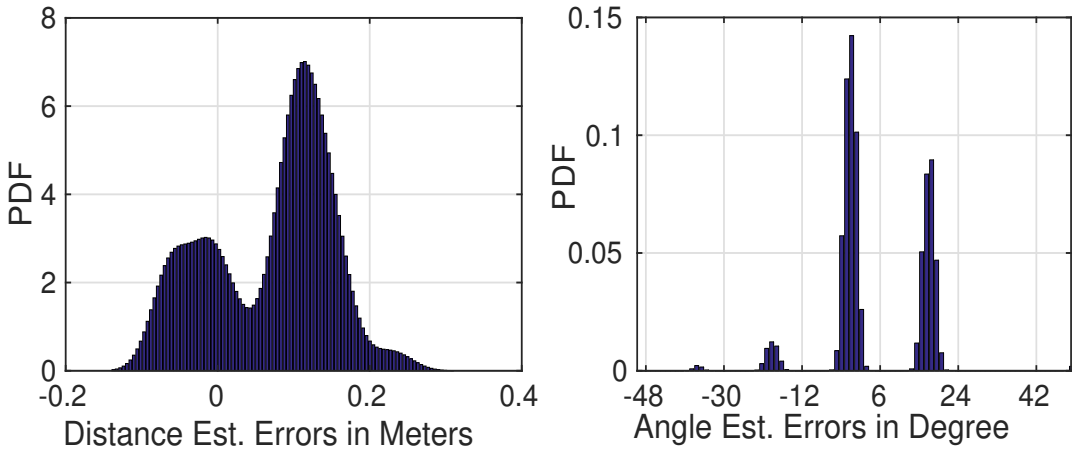
6. Upon receiving the GO packet, the RangeMote on the 3pi sends exactly one RF packet and exactly one ultrasound ping @42kHz.

7. If both transmissions are received, the TrackBot's Openmote estimates the TDoA and sends it to the mbed which then rotates the platform to the next orientation. If the TDoA process fails, the Openmote timeout and returns 0 to the mbed.

8. After rotating the platform by one step, the mbed controls the Openmote to repeat the procedure from Step 3 to Step 7.

9. If a full 360° rotation of the platform is complete, the mbed processes the TDoA data to estimate the angle and the distance. The TrackBot's Openmote switches back to channel 26 for Tracking mode.

Before evaluating the ARREST system on the basis of the TDoA ground truth system, we first evaluate the performance of the TDoA system. We found that the worst case distance estimation errors in TDoA systems are in the order of $10-20$ cm, as illustrated in Fig. 4.11a. The angle estimation statistics presented in Fig. 4.11b demonstrates highly accurate performance in angle estimations. The slight chances of getting an error of 18° are justifiable by our choice of ranging rotation step size of 18°. Thus, our TDOA system is accurate enough to be considered as a ground truth in the line of sight situations. Nonetheless, we monitor the ranging outputs to trigger retries in case of very inaccurate outputs or momentary failures. Moreover, in non-line of sight situations, we still rely on manual measurements as the TDoA system fails in such scenarios.



(a) (b)

Figure 4.11: TDOA based Localization System Performance: (a) Distance Estimation Errors and (b) Angle Estimation Errors

4.5.3.2 Different Experimental Settings

With the aforementioned setup, we performed a range of experiments over months of duration with each run lasting for $1-2$ hours. For the ARREST setup, we use the Pragmatic policy with the weighted average angle estimation because of its superior performance in our emulations and small-scale experiments. The LQG setup is also kept same as the small-scale experiments. To diversify the situation we have performed experiments in four different classes of settings.

☐ Large ($\geq 15m \times 10m$) office rooms with lots of computers, reflective surface, and cluttered regions.

☐ Long hallways ($\approx 200m$ long and $5-10m$ wide) with lots of turns.

☐ Open ground floor spaces ($\approx 30m \times 30m$) with pillars.

☐ Homelike environments with couches, furniture, and obstacles.

4.5.3.3 Performance Analysis

In Fig. 4.12a, we present the statistics of the absolute distance between the TrackBot and the LeaderBot throughout the duration of the experiments in all four scenarios. Figure 4.12a shows that the absolute distance is bounded by 3.5 meters in all four scenarios which further verifies our small scale experiment results presented in Fig. 4.9. Another noticeable fact from the figure is that ARREST system performs worst in the cluttered office scenarios which is justifiable due to presence of a lot of reflecting surfaces as well as obstacles.

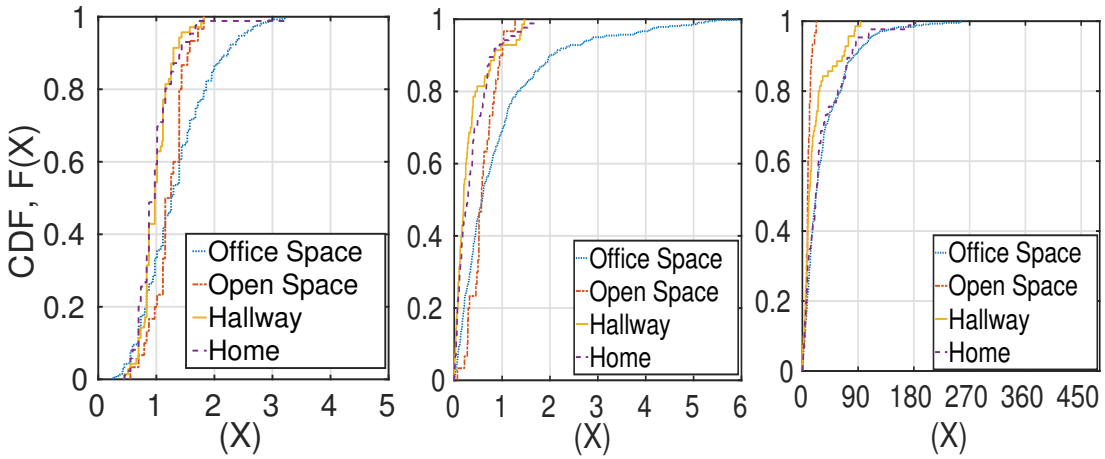


Figure 4.12: Real Experiment Based Performance for Large Scale: (a) Absolute Distance in Meters, (b) Absolute Distance Estimation Error in Meters, and (c) Absolute Angle Estimation Error in Degrees

Similar statistics can be seen in the absolute LQG distance error plot presented in the Fig. 4.12b. Figure 4.12b shows that the instantaneous absolute distance errors are $\leq 100cm$ with $\approx 90\%$ probability, except in the office scenario ($\approx 70\%$). The comparatively higher distance errors for office scenarios is due to overestimation of

distances in NLOS scenarios and in presence of strong multipath signals. However, this does not affect the performance much as the temporarily predicted higher distance tends to only lead to a temporary higher velocity of the TrackBot. In summary, the distance error statistics is also similar to the distance error statistics from the small-scale experiments. Similar pattern can be observed in the angle estimation error plots presented in Fig. 4.12c. Again the office space performance is worst. The open space performance is prominently better than the other scenarios due to the absence of any sort of multipath signals. The instantaneous angle errors are less than 40° with high probability ($\approx$ 85%) in *overall statistics.* However the scenario specific errors statistics (error being less than 40°) vary from $\approx$ 75% probability in indoor setting to $\approx$ 100% probability in the outdoor settings. This slight discrepancy between small scale and large scale angle error performance is mainly due to different environment settings as evident from the Fig. 4.12c itself. In Fig. 4.13, we present a sample illustrative trace of a large scale hallway experiment, drawn based on manual reconstruction from a video recording and markings on the floor.

4.5.3.4 Multipath Adaptation

Similar to small-scale experiments, we perform a set of experiments with a static Leader not in the line of sight of the TrackBot for $\geq$ 50% of the TrackBot's path. Due to the TrackBot's ability to leverage a good multipath signal, the TrackBot

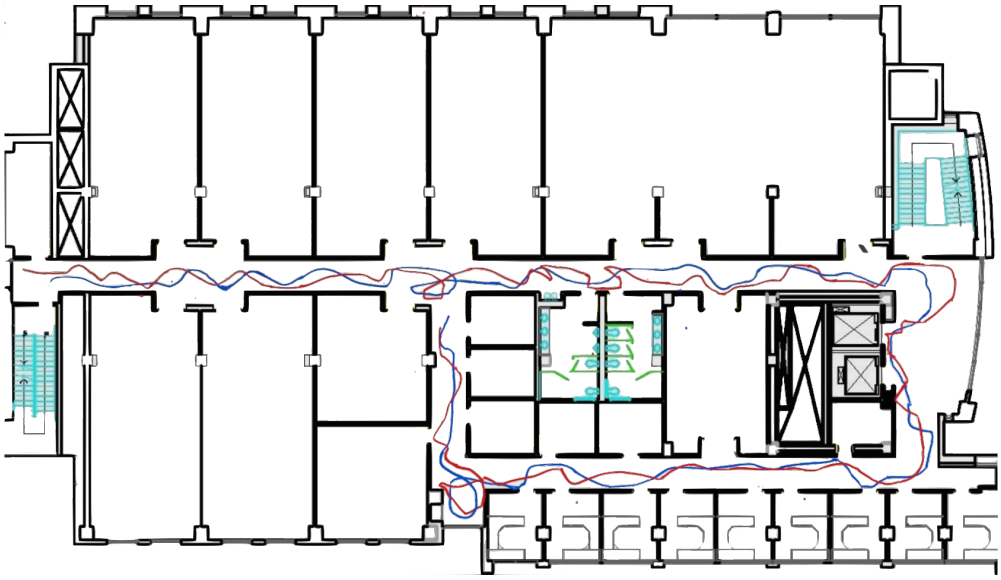


Figure 4.13: Full Path Trace for a Sample Large Scale Experiment (Blue $\Longrightarrow$ Leader, Red $\Longrightarrow$ TrackBot)

was able to find the Leader in 70% of the cases. However, we also notice that it fails dramatically if the TrackBot falls into a region with no direct path as well as no strong multipath signals (i.e., there exist multiple similar strength multipath signals). To overcome that, we add a *Multipath Angle Correction* module in the CAST layer (refer to Fig. 4.3). This module triggers a randomized movement for a single LQG period if: (1) the TrackBot hits an obstacle for $3-4$ consecutive LQG periods or, (2) the LQG estimated distance to the transmitter doesn't change much over $3-4$ consecutive periods. This policy basically leads the TrackBot to a random direction with the hope of getting out of such region. However, we noticed that if the TrackBot keeps following randomized direction for consecutive LQG periods, it harms the tracking performance. Thus, we have set a minimum time duration (Five LQG periods in our implementation) between any two consecutive

randomized movements. Note that, all these timing choices are made empirically via a range of real experiments. *With this strategy, we noticed an improvement on the TrackBot's success rate from* $\approx 70\%$ *to a success rate of* $\approx 95\%$ *in such scenarios. However, the trade-off in such context is that the convergence in case of a far away Leader* ($\geq 8m$) *is now slower by* $\approx 15\%$.

Table 4.4: Summary of Large Scale Real-World Experiments

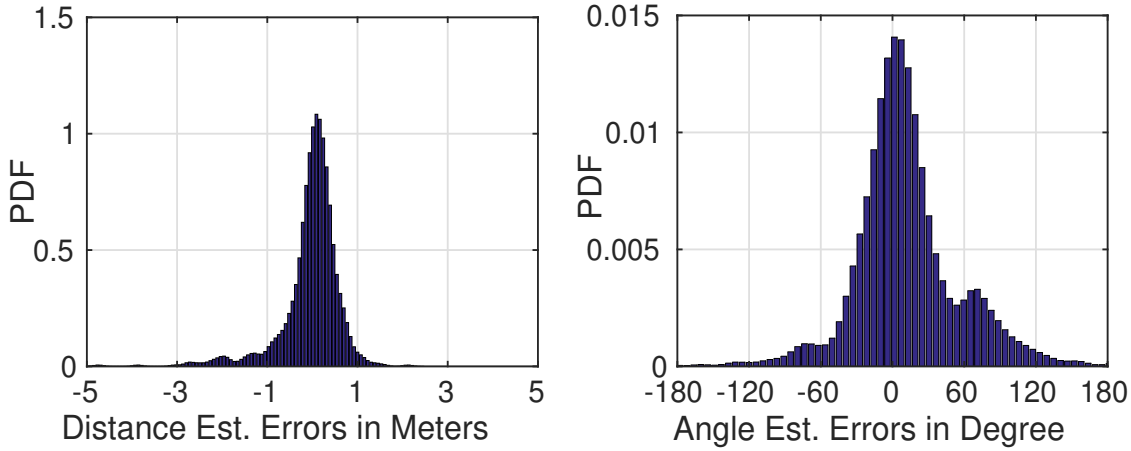
☐ Absolute distance estimation errors are $< 100cm$ with probability $\approx 90\%$ except in the case of cluttered office environments.
☐ Average Absolute angle estimation errors are $< 40^\circ$ with probability $\approx 85\%$.
☐ The TrackBot stays within $3.5m$ of the Leader with probability $\approx 100\%$ in all scenarios of tracking.
☐ In NLOS scenarios, addition of a conditional randomization improves the success rate from 70% to 95% but slows the converges by $\approx 15\%$ for static far-away Leader.

4.5.4 Miscellaneous

4.5.4.1 Raw RSSI Data Analysis

Based on all our evaluations, we conclude that the presence of multipath signals does not hamper the performance if there exists a direct line of sight. To justify this further and to gather more insights on the system's performance, we perform a raw RSSI data analysis and calculate the unfiltered error statistics. In Fig. 4.14a, we plot the RSSI pattern based distance estimation error statistics which demonstrates that the accuracy of the directional antenna pattern based distance estimations are in the order of less than 1 meter with 90% probability. On the other hand, Fig. 4.14b shows that the RSSI pattern based angle estimation error are less than 40° with

very high probability ($\approx 80\%$) with some deviations due to multipath and random changes in movement directions. Again, note that the error up to 40° is acceptable due to our choice of directional antenna. We also verify the performance of the RSSI based estimation for varying sampling rate. For these set of experiments, we fix the distance and angle between the TrackBot and the Leader and properly set the channel parameters before each experiment. Figures 4.15a and 4.15b present the average distance errors and average angle estimation errors with 95% confidence interval for varying sampling rate. Figure 4.15b shows that the angle estimation performance deteriorates as the sampling rate is decreased which is self-justified. The distance estimation actually does not vary much with the sampling rate.

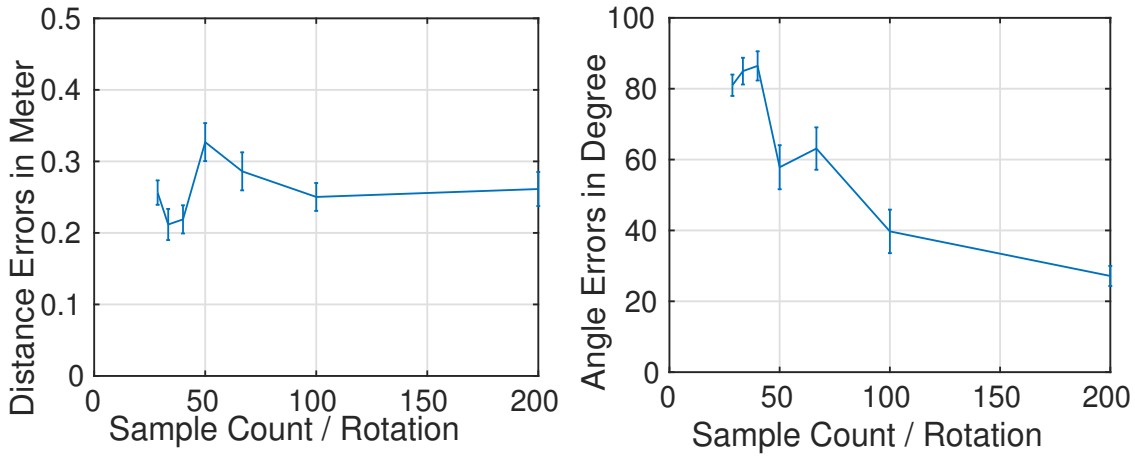


(a) (b)

Figure 4.14: Raw Data Analysis: (a) Distance Estimation Errors and (b) Angle Estimation Errors

Our numbers may even appear to be better than those typically reported for RSSI based localization (where typical accuracies are $\approx 2m - 5m$ or higher), but this is attributed to the fact that the distance estimates use the average of $40 -$

200 samples, one from each sample's respective antenna orientation. This analysis also suggests that we can use sampling rate of 100 samples/rev to achieve similar performance. Nonetheless, we stick with 200 sample/rev as we notice a loss of maximum $70 - 90$ samples per revolution in severe scenarios.



(a) (b)

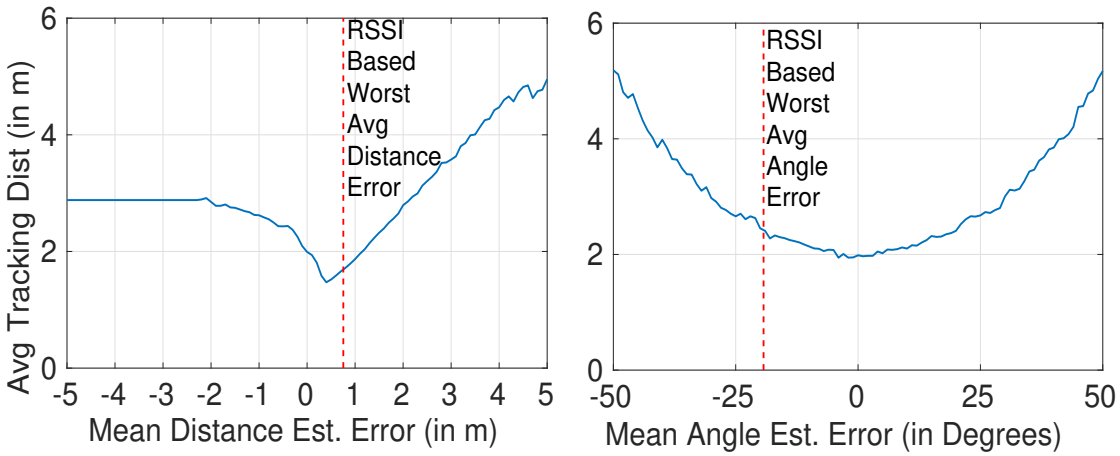
Figure 4.15: Estimation Performance for Varying Sampling Rate: (a) Distance Estimation Errors and (b) Angle Estimation Errors

4.5.4.2 Different Sensing Modalities

While our proposed ARREST architecture employs pure RSSI based distance, angle, and speed estimations, the same architecture can be easily adapted to use other technologies such as cameras or infrared sensors. In such cases, we just need to modify the CANE layer of the ARREST architecture and feed the relative position approximations to the CAST layer. Now, each of these estimation technologies i.e., camera-based or RF based estimations, have different accuracies in terms of

distance and angle estimations. To analyze the tracking performance of the AR-REST system, oblivious to the actual technology used in CANE layer, we perform a set of simulation experiments where we control the average errors in the distance and the angle estimations. Figure 4.16a illustrates one instance of such experiments where we fix the average angle error (0 in this case) and vary the average distance estimation error. Figure 4.16a shows that the effect of positive estimation errors ($d_{org} - d^e > 0$, where d_{org} is the actual distance) have a more detrimental effect on the tracking performance than negative errors. This is justified as positive distance estimation errors imply always falling short in the movements, whereas, negative errors imply over-estimations and more aggressive movements. It is also noticeable that there exists an optimal value of average distance estimation error. The value of this optimal distance error depends on the maximum Leader speed as well as average angle error. Next, we plot the relation between average tracking distance and average angle error while the average distance error is kept to be 0 in Fig. 4.16b. It is obvious and quite intuitive that the best tracking performance is obtained for an average angle estimation error of 0. Note that, we do not control the speed error separately as it is directly related to the angle and distance estimations. This analysis demonstrates the versatility of our ARREST architecture to tolerate a large range of estimation errors. More specifically, it tolerates up to $5m$ average distance error and 45° average absolute angle error in a successful tracking application. This analysis also shows that while RSSI based system is not optimum, it has reasonable

performance compared to the best possible ARREST system (with zero distance and angle estimation error).



(a) (b)

Figure 4.16: Performance of the ARREST System in Terms of Controlled Estimation Errors: (a) Fixed Angle Estimation Error, and (2) Fixed Distance Estimation Error

4.6 Discussion

In this study, we proposed ARREST, a pure RF based relative localization and tracking system, for autonomously following an RF-emitting object. The proposed ARREST system employs a rotating directional antenna to approximate the relative distance, orientation, and speed, that is further utilized by an LQG controller to improve the estimates and to autonomously control the motion of the following agent i.e., the TrackBot. Through a set of emulations and real-world experiments using our TrackBot prototype, we demonstrated the performance and error statistics of our system. Moreover, for ground truth evaluation we developed another

system for relative localization in line-of-sight situations by using the concept of TDoA for simultaneous RF and Ultrasound transmissions. Note that while the problem formulation of this study deals with only close bounded proximity ($\leq D_{th}$) maintenance, it can be easily switched to a different yet similar objective where the Trackbot is required to stay at a distance, $D_{th}^{l} \leq d \leq D_{th}^{h}$ with D_{th}^{l} and D_{th}^{h} being the lower and upper bounds of the distance, respectively. One simple method for doing that would be to replace $d[n]$ with $(d[n] - D_{th}^{l})$ and D_{th} with $(D_{th}^{h} - D_{th}^{l})$. Based on the application context, one can switch between these two types of formulations. Nonetheless, there exist some potential directions for further research such as incorporating game theory and optimality conditions into the system.

Chapter 5

Interference Power Bound Analysis for Network Formation Control in RWN

In this study, we consider a fundamental problem concerning the deployment of a robotic wireless network: to fulfill various end-to-end performance requirements, a "sufficient" number of robotic relays must be deployed to ensure that links are of acceptable quality. Prior work has not addressed how to find this number.[1] We use the properties of Carrier Sense Multiple Access (CSMA) based wireless communication to derive an upper bound on the spacing between any transmitter-receiver pair, which directly translates to a lower bound on the number of robots to deploy. We focus on SINR-based performance requirements due to their wide applicability. We show that the bound can be improved by exploiting the geometrical structure of a network, such as a linearity in the case of flow-based robotic router networks. Furthermore, we also use the bound on robot count to formulate a lower bound

[1]The material in this chapter is based in part on the work in [150].

on the number of orthogonal codes required for a high probability of interference-free communication. We demonstrate and validate our proposed bounds through simulations.

5.1 Problem Description

In this section, we detail our problem formulations. For compactness, we list the symbols used for base problem formulation in Table 5.1 and symbols related to our goals in Table 5.2, respectively. Say, we have a transmitter node T and a receiver node X that are placed at d distance apart, alongside with a larger number of interfering wireless nodes. Each node of this interference limited network (i.e., the interference dominates over noise) employs *Channel Sense Multiple Access with Collision Avoidance* (CSMA/CA) [18] for wireless media access and has a transmission power of P_t. The radio range of each node is subdivided into three regions, centered at the node's location: a circular *connected/contention region* of radius D_1, an annular *transition region* with inner radius D_1 and outer radius D_2 (including the boundaries), and a *disconnected region* which is the entire region outside the circle with radius $D_2 > D_1$; where the values of D_1 and D_2 depend on the actual RSSI thresholds of the devices used [151]. Undoubtedly, in the presence of fading, the regions are not so nicely structured, nonetheless, can be approximated by proper choice of D_1 and D_2. Now, the CSMA restricts simultaneous transmissions from the nodes in the contention region of T, while the nodes in the transition

region are aware of T's transmission with very low probabilities and, therefore, are the potential interferers. However, only a subset of the nodes in the transition region can be active simultaneously, due to CSMA among themselves, which requires any two simultaneous interferers to be at least D_1 distance apart. The interference power from a node in the disconnected region is considered insignificant.

Table 5.1: General Parameters

Symbol	Description
T	Transmitter
X	Receiver
d_{ij}	Distance between node i and j
d	Distance between T and X i.e., d_{TX}
η	Path Loss Exponent
$\psi \sim \mathcal{N}(0, \sigma^2)$	Log normal Fading Noise with variance σ^2
P_t	Transmitted Signal Power
P_r	Received Signal Power
$P_{\mathcal{I}}$	Received Interference Power
$\mathcal{I}^C$	Interference Set Cover
M	Number of Flows

Table 5.2: System Parameters

Symbol	Description
SIR_{th}	The Target Minimum SIR
$SIR_X(d)$	Minimum Achievable SIR at X for d separation
D_1	Contention Region Outer Radius
D_2	Transition Region Outer Radius
γ	Required Probability of $SIR \geq SIR_{th}$
κ	Minimum probability of interference free communication
d_{max}	Maximum distance allowed between T and X
$N_{\mathcal{O}}$	Number of Orthogonal Codes
$N_{\mathcal{I}}^{max}$	Maximum Number of Interfering Nodes
$N^{\mathcal{R}}$	Minimum Number of Nodes to Deploy

Definition 5.1.1. *A set of interfering nodes ($\mathcal{I}^C$) such that $D_2 \geq d_{iT} \geq D_1$ and $d_{ij} \geq D_1 \ \forall \ i, j \in \mathcal{I}^C$, is referred to as an* Interference Set Cover*.*

Now, there are four main objectives of this work as follows.

Objective 5.1.1. *Find a mapping between d and the minimum achievable SIR at X, $SIR_X(d)$.* □

Objective 5.1.2. *Find the range, $0 < d \leq d_{max}$, such that the outage probability i.e., $\mathbb{P}(SIR_X(d) < SIR_{th}) < \gamma$ where $0 \leq \gamma \leq 0.5$ is the choice of the designer.* □

Now, one can employ a set of orthogonal codes to further restrict the interference in a CSMA network. In such cases, the maximum value of interference power decreases, based on the number of codes employed, possibly leading to near zero interference. In this context, our goal is as follows.

Objective 5.1.3. *Characterize $SIR_X(d)$ as a function of the number of orthogonal codes ($N_{\mathcal{O}}$) employed for concurrent transmissions, and find a bound $N'_{\mathcal{O}}$ such that $\mathbb{P}(\mathbb{1}_{\mathcal{I}0} = 1) \geq \kappa \ \forall N_{\mathcal{O}} > N'_{\mathcal{O}}$, where the indicator function $\mathbb{1}_{\mathcal{I}0}$ refers to interference free communication and $\kappa \geq 0.5$ is a designer choice.* □

For our SIR and Interference bound analysis, we consider two different scenarios in this study. In the *first scenario*, the node pair in focus is placed in a *"dense"* network, where a countably many *uncontrollable wireless nodes* are co-located in the area of interest. *Secondly*, we consider our target application of robotic router placement, where the goal is to place a set of robots such that they form multihop

links between a set of maximum M concurrent communication end-point pairs. This application context restricts the possible configuration of the interfering nodes within a class of network formations, such as straight line formation, that voids the earlier dense network assumption. At any time instance, we associate a set of routers with each flow $i \in \{1, 2, \cdots M\}$ that form a chain between the communication endpoints.

For a fixed set of communication endpoints of a flow i, the minimum number of nodes ($N_i^{\mathcal{R}}$) to be allocated to flow i depends on d_{max} which in turn controls the minimum number of nodes to be deployed, $N^{\mathcal{R}} \geq \sum_{i=1}^{M} N_i^{\mathcal{R}}$.

Objective 5.1.4. *Find a better and tighter bound on interference as well as SIR by exploiting the application specific restrictions on the network configurations. Next, analyze the improvement in the number of robots required, with this improved bound.*

□

5.2 Outline of the Proposed Solution

In this section, we summarize our methodologies for achieving the target objectives while the details are discussed later on.

5.2.1 Methodology for Mapping from d to SIR_X

For a fixed value of the separation distance d between T and X, we estimate the maximum feasible interference as well as minimum feasible SIR, by exploiting

the geometry of the connectivity region and transition region. For received power modeling, we opt for the standard lognormal fading model [18], where the received power is distributed lognormally with mean power calculated using simple path loss model. Thus, the received power can be represented as:

$$P_r(d) = Q.P_t d^{-\eta} 10^{\frac{\psi}{10}} \tag{5.1}$$

where Q is some constant. Next, we introduce the following claim as our whole estimation process revolves around this claim.

Claim 5.2.1. *In presence of Independent and Identically Distributed (I.I.D) fading noise, the Interference Set Cover (see Definition 5.1.1) with maximum mean power as well as the maximum number of interferers will give us better stochastic bound than any other Interference Set Cover.*

Justification. This claim is justified by the fact that, if the fading noise is I.I.D, the Interference Set Cover with the maximum number of nodes will give the highest variance. Thus, the Interference Set Cover with highest mean as well as highest number of nodes will be a better bound than any other Interference Set Cover. □

Now, the main steps for representing SIR_X as a function of d are as follows.

Step 5.2.1. *We first identify the* Interference Set Cover(s) *($\mathcal{I}^C$) that will potentially give us the best estimate of the maximum feasible mean interference power, for a fixed d, using the greedy algorithm.*

Step 5.2.2. *We estimate the maximum cardinality of an* Interference Set Cover*, $N_{\mathcal{I}}^{max}$. This number is used in a later step to compensate for non-optimal greedy solutions from Step 5.2.1.*

Step 5.2.3. *To get the maximum interference power, we add up the interference powers of the nodes of the Interference Set Covers selected in Step 5.2.1, according to Eqn* (5.1)*. Thus the total interference power at X is a sum of lognormal variables as follows.*

$$P_{\mathcal{I}^C}(d) = Q. \sum_{j \in \mathcal{I}^C} P_t d_{jX}^{-\eta} 10^{\frac{\psi}{10}} \tag{5.2}$$

Step 5.2.4. *We multiply the interference power estimate in Step 5.2.3 by a correction factor $\zeta = \max\{1, \frac{N_{\mathcal{I}}^{max}}{|\mathcal{I}^C|}\}$, where $|.|$ denotes the cardinality of a set, to account for the Interference Set Covers with less than $N_{\mathcal{I}}^{max}$ number of nodes, i.e., $|\mathcal{I}^C| < N_{\mathcal{I}}^{max}$. Now, the modified interference power is:*

$$P_{\mathcal{I}^C}(d) = \zeta.Q. \sum_{j \in \mathcal{I}^C} P_t d_{jX}^{-\eta} 10^{\frac{\psi}{10}} \tag{5.3}$$

Step 5.2.5. *We calculate the SIR value for each of the Interference Set Covers selected in Step 5.2.1 in dB, as follows.*

$$SIR_X(d) = 10 \log_{10} \left(\frac{P_t d^{-\eta} 10^{\frac{\psi}{10}}}{\zeta. \sum_{j \in \mathcal{I}^C} P_t d_{jX}^{-\eta} 10^{\frac{\psi}{10}}} \right) \tag{5.4}$$

5.2.2 Methodology for Selecting d_{max}

In order to properly select d_{max}, first of all, we need to estimate the distribution of the $SIR_X(d)$ using Eqn (5.4), which is not very straightforward as it involves division and summation of a large set of lognormal random variables. The traditional log normal summation methods involve sampling and filtering to fit the distribution into an approximated log normal [152]. We opt for similar approach where we collect a good number of samples, say 50000, from each of the contributing log normal distributions, for a fixed d, to generate the SIR samples ($SIR_X(d)$) and use the SIR samples to determine the mean, $\mu_{SIR_X(d)}$, the variance of the SIR, $\sigma^2_{SIR_X(d)}$ and the empirical probability distribution function (PDF) of the $SIR_X(d)$. *Note that in presence of fading, using simple path loss model, we can easily get the mean powers received from each interferer, which can be used to estimate* $\frac{\mathbb{E}(P_r)}{\mathbb{E}(P_{\mathcal{I}})}$*, but, not the mean SIR, i.e.,* $\mathbb{E}(SIR) = \mathbb{E}\left(\frac{P_r}{P_{\mathcal{I}}}\right) \neq \frac{\mathbb{E}(P_r)}{\mathbb{E}(P_{\mathcal{I}})}$.

Step 5.2.6. *To properly select* d_{max}*, we first choose an acceptable value for* SIR_{th} *and* γ*. Next, we use the samples of* $SIR_X(d)$ *to estimate the outage probability* $\Gamma(d) = \mathbb{P}(SIR_X(d) < SIR_{th})$*, for a uniformly selected values of* $d \in [0, D_1]$*. The highest value of* d *that satisfies* $\Gamma(d) < \gamma$ *is the estimated* d_{max}*.*

5.2.3 Orthogonal Code Bound For Interference Free Network

First of all, say, $N_{\mathcal{O}}$ number of orthogonal codes are used and each node chooses a code randomly (all codes are equally likely to be chosen) and independently. The

new code specific interference power bound for a randomly selected Interference Set Cover ($\mathcal{I}^C$) will be:

$$\begin{aligned} P_{\mathcal{I}^C}(d|\mathcal{O}_T) &= \sum_{j=1}^{|\mathcal{I}^C|} P^j_{\mathcal{I}^C} \times \mathbb{1}_{\{\mathcal{O}_j=\mathcal{O}_T\}} \\ \mathbb{E}(P_{\mathcal{I}^C}(d)) &= \frac{1}{(N_{\mathcal{O}})} \sum_{j=1}^{|\mathcal{I}^C|} \mathbb{E}(P^j_{\mathcal{I}^C}) \end{aligned} \tag{5.5}$$

where $\mathcal{O}_T$ is the code chosen by the transmitter in focus (T), $P^j_{\mathcal{I}^C}$ denotes the interference power due to j^{th} interferer in $\mathcal{I}^C$, and the indicator function $\mathbb{1}_{\{\mathcal{O}_j=\mathcal{O}_T\}}$ denotes whether the j^{th} interferer have chosen same code as the transmitter i.e., $\mathcal{O}_T$. *Notice that, the Interference Set Cover with maximum mean interference power will still give us the maximum mean estimated interference power in presence of orthogonal codes.*

Step 5.2.7. *We use the estimated Interference Set Cover from Step 5.2.1 to determine the new SIR bounds as follows.*

$$SIR_{\mathcal{I}^C}(d|\mathcal{O}_T) = \frac{P_t d^{-\eta} 10^{\frac{\psi}{10}}}{\zeta . \sum_{j \in \mathcal{I}^C} \left(P_t d_{jX}^{-\eta} 10^{\frac{\psi}{10}} \right) . \mathbb{1}_{\{\mathcal{O}_j=\mathcal{O}_T\}}} \tag{5.6}$$

Now, at any time instance, maximum $N^{max} = (N^{max}_{\mathcal{I}} + 1)$ number of nodes can be active simultaneously. Given that $N_{\mathcal{O}} \geq N^{max}$, we deduce that:

$$\mathbb{P}(\mathbb{1}_{\mathcal{I}0} = 1) \geq \prod_{i=1}^{N^{max}} \left(1 - \frac{i-1}{N_{\mathcal{O}}} \right) \tag{5.7}$$

Proof. Say, at any time instance, the number of active interferers is $N_{\mathcal{I}} \in [0, N_{\mathcal{I}}^{max}]$. Given that $N_{\mathcal{I}}$ number of nodes are active and $N_{\mathcal{O}} \geq N_{\mathcal{I}}$, the probability of interference free communication is as follows.

$$\begin{aligned} \mathbb{P}(\mathbb{1}_{\mathcal{I}0} = 1 | N_{\mathcal{I}}) &= \frac{^{N_{\mathcal{O}}}C_{N_{\mathcal{I}}} \times N_{\mathcal{I}}!}{(N_{\mathcal{O}})^{N_{\mathcal{I}}}} = \prod_{i=1}^{N_{\mathcal{I}}} \left(1 - \frac{i-1}{N_{\mathcal{O}}}\right) \\ \implies \mathbb{P}(\mathbb{1}_{\mathcal{I}0} = 1 | N_{\mathcal{I}}^1) &\geq \mathbb{P}(\mathbb{1}_{\mathcal{I}0} = 1 | N_{\mathcal{I}}^2) \quad \text{if } N_{\mathcal{I}}^1 \leq N_{\mathcal{I}}^2 \leq N_{\mathcal{O}} \end{aligned} \tag{5.8}$$

Thus, the probability of interference free transmission for $N_{\mathcal{O}} \geq N^{max}$, where $N^{max} = N_{\mathcal{I}}^{max} + 1$, can be expressed as follows.

$$\begin{aligned} \mathbb{P}(\mathbb{1}_{\mathcal{I}0} = 1) &= \sum_{j=0}^{N^{max}} \mathbb{P}(\mathbb{1}_{\mathcal{I}0} = 1 | N_{\mathcal{I}} = j) \mathbb{P}(N_{\mathcal{I}} = j) \\ &= \sum_{j=0}^{N^{max}} \prod_{i=1}^{j} \left(1 - \frac{i-1}{N_{\mathcal{O}}}\right) \mathbb{P}(N_{\mathcal{I}} = j) \\ &\geq \prod_{i=1}^{N^{max}} \left(1 - \frac{i-1}{N_{\mathcal{O}}}\right) \sum_{j=0}^{N^{max}} \mathbb{P}(N_{\mathcal{I}} = j) \quad \text{Using (5.8)} \\ &\geq \prod_{i=1}^{N^{max}} \left(1 - \frac{i-1}{N_{\mathcal{O}}}\right) \end{aligned} \tag{5.9}$$

□

From Eqn (5.7), we can see that for $N_{\mathcal{O}} \geq N^{max}$, $\prod_{i=1}^{N^{max}} \left(1 - \frac{i-1}{N_{\mathcal{O}}}\right)$ is a strictly increasing function of $N_{\mathcal{O}}$.

Step 5.2.8. *To find the optimum value of $N_{\mathcal{O}}$, we estimate $\prod_{i=1}^{N^{max}} \left(1 - \frac{i-1}{N_{\mathcal{O}}}\right)$ for increasing value of $N_{\mathcal{O}}$ (starting from N^{max}), and select the minimum value of $N_{\mathcal{O}}$ such that $\prod_{i=1}^{N^{max}} \left(1 - \frac{i-1}{N_{\mathcal{O}}}\right) \geq \kappa$.*

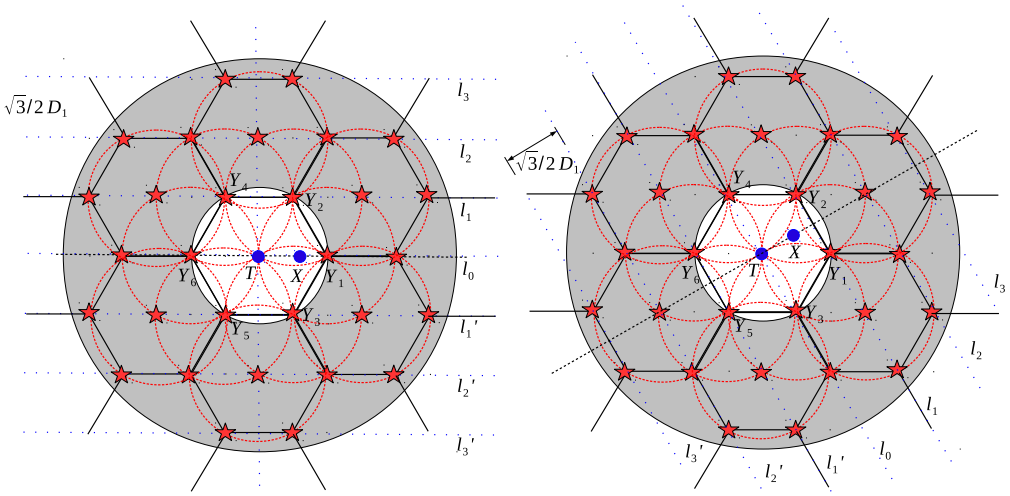
5.3 Identification of Maximum Power Interference Set Cover

In the section, we identify the Interference Set Covers that result in the highest total interference power at a given receiver location, X, for both scenarios i.e., dense random network and robotic router network.

5.3.1 Dense Random Network

In [97], Hekmat and Van Mieghem showed that the mean interference power in CSMA Network is bounded by the interferers located along the hexagonal rings centered at the receiver's location, where the i^{th} ring with each side length equal to $i \times D_1$ contains $6 * i$ nodes. While the assumption of putting the receiver at the center is valid in the presence of RTS/CTS mechanism in CSMA, in the reality, RTS/CTS mechanism is *NOT* employed in most of the enterprise wireless networks as well as the Internet of Things (IoT) networks. *In such cases, the transmitter is the node to be located at the center of the rings while the receiver is free to be located anywhere in the connected region of T and, as a result, the maximum feasible interference is actually dependent on the separation distance (d).* However, hexagonal packing is known to be the densest packing in circular spaces which leads us to believe that the distance-dependent interference is also bounded by the interference power of the set of interferers located at hexagonal rings (similar to [97] but in an

annular ring) around the transmitter's location. With this assumption, our focus becomes restricted to all possible sets of locations that form such hexagonal packing. *Moreover, we prove that with the separation distance $d > 0$, we only need to consider two different angular orientations of such hexagonal packing, as illustrated in Figures 5.1a and 5.1b.*



(a) Configuration1 (b) Configuration2

Figure 5.1: Illustration of the Interference Set Covers For Estimation of Interference Upper Bound in a Dense Network

In the first type of configuration, which we refer to as *Configuration 1*, the closest interferer is located at the intersection of the inner boundary of the annulus and the line joining T and X (Illustrated in Figure 5.1a). *This configuration is generated by taking a greedy iterative approximation approach, where we start with an empty $\mathcal{I}^C$ and, in each iteration, we select a point on the annulus that is closest to the receiver X and is not located in the connected regions of the nodes already added to $\mathcal{I}^C$.* In the second configuration, which we refer to as *Configuration 2*

(illustrated in Figure 5.1b), the number of closest interferers is two and they are exactly D_1 distance apart from each other as well as from the transmitter. *With this new initial condition, we can find the rest of the nodes, again, using the greedy approach.* The positions of the interfering nodes for both of these Interference Set Covers are detailed in Table 5.3. These two configurations form the bound of the interference power for any configuration within same class i.e, with the similar relative position between nodes with hexagonal corner positioning. Next, we calculate the interference and SIR for these two configurations according to Eqn. (5.3) and (5.4). Then, we choose the maximum of these two interference estimates as our interference estimate and the minimum of these two SIR estimates as our SIR estimate. We perform this using the sampling method discussed in Section 5.2.2, where *we collect a large number of pairs of samples from these two configurations and take the highest interference power sample (or lowest SIR sample) from each pair as a sample for our estimated bound.*

However, since this is a greedy solution, the resulting Interference Set Cover combination may not include the maximum number of interferer and, therefore, does not guarantee maximum possible interference power. Now say the greedy logic includes n interferes. Then according to the greedy logic, it is guaranteed that the top n interfering nodes of the maximum power Interference Set Cover will have less or equal interference power compared to the interference power from the greedily found Interference Set Cover. To guarantee that our estimated interference

Table 5.3: Interference Set Cover Node Locations for a Dense Network

Line Number (Illustrated in Figs 5.1a and 5.1b)	Configuration 1	Configuration 2	
l_0	$\{(\pm jD_1, 0)\}$ $\forall j \in \{1, 2, \cdots, N_0 + 1\}$	$\{(0, \pm jD_1)\}$ $\forall j \in \{1, 2, \cdots, N_0 + 1\}$	$N_0 = \lfloor \frac{D_2 - D_1}{D_1} \rfloor$
l_k where k is odd $\forall k \in \{1, \lfloor \frac{2D_2}{\sqrt{3}D_1} \rfloor\}$	$\{(\pm \frac{D_1(1+2\times j)}{2}, \frac{\sqrt{3}}{2} kD_1)\}$ $\forall j \in \{0, 1, \cdots, N_k\}$	$\{(\frac{\sqrt{3}}{2} kD_1, \pm \frac{D_1(1+2\times j)}{2})\}$ $\forall j \in \{0, 1, \cdots, N_k\}$	$N_k = \lfloor \frac{\left(D_2^2 - \frac{3}{4}k^2 D_1^2\right)^{\frac{1}{2}}}{D_1} \rfloor$
l'_k where k is odd $\forall k \in \{1, \lfloor \frac{2D_2}{\sqrt{3}D_1} \rfloor\}$	$\{(\pm \frac{D_1(1+2\times j)}{2}, -\frac{\sqrt{3}}{2} kD_1)\}$ $\forall j \in \{0, 1, \cdots, N_k\}$	$\{(-\frac{\sqrt{3}}{2} kD_1, \pm \frac{D_1(1+2\times j)}{2})\}$ $\forall j \in \{0, 1, \cdots, N_k\}$	$N_k = \lfloor \frac{\left(D_2^2 - \frac{3}{4}k^2 D_1^2\right)^{\frac{1}{2}}}{D_1} \rfloor$
l_k where k is even $\forall k \in \{1, \lfloor \frac{2D_2}{\sqrt{3}D_1} \rfloor\}$	$\{(\pm jD_1, \frac{\sqrt{3}}{2} kD_1)\}$ $\forall j \in \{0, 1, \cdots, N_k\}$	$\{(\frac{\sqrt{3}}{2} kD_1, \pm jD_1\}$ $\forall j \in \{0, 1, \cdots, N_k\}$	$N_k = \lfloor \frac{\left(D_2^2 - \frac{3}{4}k^2 D_1^2\right)^{\frac{1}{2}}}{D_1} \rfloor$
l'_k where k is even $\forall k \in \{1, \lfloor \frac{2D_2}{\sqrt{3}D_1} \rfloor\}$	$\{(\pm jD_1, -\frac{\sqrt{3}}{2} kD_1)\}$ $\forall j \in \{0, 1, \cdots, N_k\}$	$\{(-\frac{\sqrt{3}}{2} kD_1, \pm jD_1\}$ $\forall j \in \{0, 1, \cdots, N_k\}$	$N_k = \lfloor \frac{\left(D_2^2 - \frac{3}{4}k^2 D_1^2\right)^{\frac{1}{2}}}{D_1} \rfloor$

power is no less than the maximum possible interference power, we multiply our estimated interference power by a correction factor, ζ (discussed in Step 5.2.4). We determine the maximum number of concurrent interfering nodes ($N_{\mathcal{I}}^{max}$) by formulating the problem as a circle packing problem [153] as follows. Note that, there exists a range of approximation solution to the circle packing problem [153], which can be directly applied to solve this problem. In this study, we do not discuss any circle packing solution.

Definition 5.3.1. ***Pack Problem:*** *Maximize the number of circles with radius* $\left(\frac{D_1}{2}\right)$ *that can be packed inside an annulus with inner and outer radius:* $\left(\frac{D_1}{2}\right)$ *and* $\left(D_2 + \frac{D_1}{2}\right)$*, respectively.*

Lemma 5.3.1. *The cardinality of the solution to the Pack Problem is also the maximum cardinality of an Interference Set Cover.*

Proof. A valid solution to the Pack Problem can be directly mapped to a valid Interference Set Cover. To prove that, let us consider the set of centers, S_P, for the circles in the Pack Problem solution. For any valid solution to the Pack Problem, the distance between the centers of the circles are at least R_1 which satisfies the Interference Set Cover distance condition. Now, the center of any circle to be packed must lie in the annulus with radius R_1 and R_2 as the radius of the circles are $\frac{R_1}{2}$. Thus S_P is a valid Interference Set Cover. Next, assume the solution to the pack problem, n, does not contain the maximum number of interferers. So there must exist an Interference Set Cover with more than n interferer. However, if we formulate a set of circles with the centers to be same as the Interference Set Cover but with a radius equal to $\frac{R_1}{2}$, it is also a valid circle packing solution with higher cardinality. This is a contradiction. Thus the earlier assumption is not true. Conversely, say that the solution to the Pack problem has a higher cardinality than the max cardinality of Interference Set Cover, we can always map the Pack problem solution to a new Interference Set Cover with higher cardinality than the earlier solution. This is also a contradiction, thus, proves the lemma.

□

5.3.2 Interference Estimation for Robotic Router Network

In this section, we focus on the interference estimation for our application specific context of a robotic wireless network in an obstacle-free environment. Before that, we make an assumption, based on two related works [7, 9], as follows.

Assumption 5.3.1. *For a flow based robotic network in an obstacle-free environment, if the goal is to optimize the flow performance in terms of SIR, the best configuration of robots allocated to that flow is to stay on the straight line joining the static endpoints.*

This assumption is justified by the work presented in [7] which shows that the best configuration of robots in order to optimize packet reception rate (which is directly related to SIR) of a flow-based network is to evenly place them along the line segment joining the static endpoints. The work of Yan and Mostofi [9] further justify the linear arrangement of same flow nodes for Signal to Noise Ratio (SNR) based optimization goal. In our analysis, we employ Assumption 5.3.1 to restrict the feasible positions of the interfering nodes, thereby, leading to better and tighter bounds on interference. In this context, we divide the interference into two components: Intra-flow interference and Inter-flow interference. These two components refer to the interference power from the nodes in the same flow as the transmitter T and interference power from the nodes of different flows, respectively.

Intra-Flow Interference Our intra-flow interference estimation is based on the following lemma.

Lemma 5.3.2. *The maximum expected intra-flow interference power for a link corresponds to the sum of interference powers from nodes located at distances* $\{D_1, 2 \cdot D_1, \cdots k \cdot D_1\}$ *from the transmitter node* T *along the line segment joining the flow endpoints, where* $k.D_1 \leq D_2$ *(illustrated in Fig. 5.2).*

Proof. Assumption 5.3.1 states that in the final configuration, the routers should be placed along the line segment joining the sink and source, say l_{opt}. Now, assume that the first interferer in the worst case interference combination is located at $D_1 + \delta$ distance from the source, along l_{opt}, instead of D_1 where $0 < \delta < (D_2 - D_1)$. Since, the distance between two interferer have to be greater than D_1 for concurrent transmission, the resulting set of interferers are located at $\mathcal{I}_1 = \{D_1 + \delta, 2 * D_1 + \delta, \cdots, k * D_1 + \delta\}$ where $k * D_1 + \delta \leq D_2$. Now, the Interference Power is inversely proportional to distance, more specifically $d^{-\eta}$ where $2 \leq \eta \leq 6$ is the path loss exponent. Now say, the receiver is located at distance d from the transmitter on the same side as the interferers. Therefore, the power of the interferer located at $D_1 + \delta$ is less the the power of interferer located at D_1 as $\frac{1}{(D_1 - d)^\eta} \geq \frac{1}{(D_1 + \delta - d)^\eta}$. Similarly if the receiver is located at distance d from the transmitter on the other side i.e, the distance between the first interferer and the receiver is $D_1 + \delta + d$, the power of the interferer located at $D_1 + \delta$ is less the the power of interferer located at D_1 as $\frac{1}{(D_1 + d)^\eta} \geq \frac{1}{(D_1 + \delta + d)^\eta}$ Thus, if we exchange the first interferer position with D_1 i.e,

$\mathcal{I}_2 = \{D_1, 2*D_1+\delta, \cdots, k*D_1+\delta\}$ where $k*D_1+\delta \leq D_2$ then we get set of location with total interference power higher than that of I_1. This is a contradiction. Thus the earlier assumption is wrong, thus, proves the lemma. □

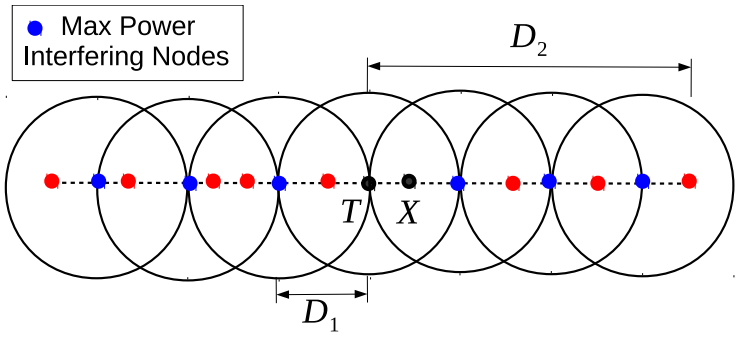
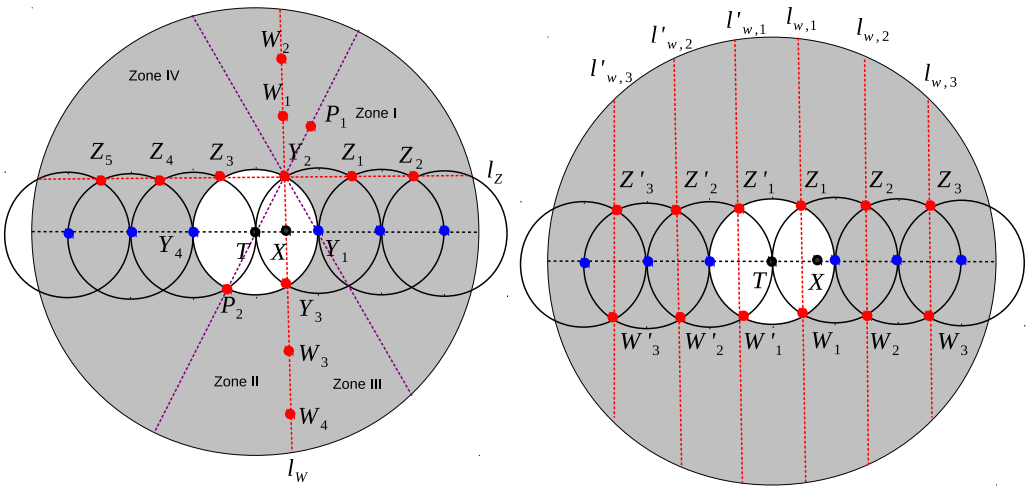


Figure 5.2: Illustration of the Highest Power Intra-Flow Interference Set Cover

Therefore, the maximum number of intra-flow interferers is $2\left(\lfloor\frac{D_2-D_1}{D_1}\rfloor+1\right)$, where the factor 2 accounts for both sides.

Inter-Flow Interference In realistic scenarios, there will be more than one flows in the network where robots assigned to different flows can interfere as well. We refer to such interference as the *Inter-flow interference*. Now, the interferers can be located in the annular transition region around the transmitter, while the nodes allocated to same flow stay on the straight line joining the endpoints of the respective flow (according to Assumption 5.3.1). In this section, we start the bound estimation for a two flow network, followed by a network with M flows. In this context, we make a key assumption about the maximum power Interference Set Cover for the multi-flow scenario, as follows.

Assumption 5.3.2. *For any transmitter-receiver node pair of a flow, the intra-flow maximum power Interference Set Cover estimated in section 5.3.2 is always part of the maximum power Interference Set Cover in presence of multiple flows.* □



(a) Two flow Case (b) M Flow Case

Figure 5.3: Illustration of the Multiple Flow Interference Estimation (Blue Nodes: Intra-Flow Interferer, Red Nodes: Inter-Flow Interferer)

The reason behind this assumption is mainly the fact that in practical deployment, some node-pairs might not have any inter-flow interference at all (e.g., single flow network). Therefore, neglecting any of the intra-flow interfering nodes will lead to an incorrect estimate of the interference in such cases. Under the given assumption, our next step is to find another line segment that will generate the maximum inter-flow interference power, for two flow cases. In general case with M flows, we need to find $M-1$ other line segments such that carefully placed set of interferers on those segments result in the highest inter-flow interference power. Now, following the greedy approach mentioned in the Section 5.3.1, the second flow should

contain Y_2 or Y_3 or both, in Figure 5.3a, since they are the next closest points to X after the Intra-flow interference set cover nodes are accounted for. We prove this by introducing two lemmas as follows.

Lemma 5.3.3. *The length of the chords of an annulus with inner radius D_1 and outer radius D_2, located at $d_r < D_1$ distance from the centre increases monotonically with d_r.*

Proof. Lets take a random chord of the annulus, located at d_r distance from the center with $d_r < D_1$. Then the length of the chord is equal to $g(d_r) = \sqrt{D_2^2 - d_r^2} - \sqrt{D_1^2 - d_r^2}$. Now taking derivative of $g(.)$ as follows.

$$\begin{aligned} g'(d_r) &= -\frac{d_r}{\sqrt{D_2^2 - d_r^2}} + \frac{d_r}{\sqrt{D_1^2 - d_r^2}} \\ &= -\frac{1}{\sqrt{(\frac{D_2}{d_r})^2 - 1}} + \frac{1}{\sqrt{(\frac{D_1}{d_r})^2 - 1}} \\ &> 0 \text{ as } D_2 > D_1 \text{ and } d_r < D_1 \end{aligned} \tag{5.10}$$

This implies that $g(.)$ is a strictly increasing function of d_r, which proves our lemma. □

Lemma 5.3.4. *Among the possible line segments through Y_2 or Y_3 or both, we just need to consider $l_{\mathcal{Z}}$ and $l_{\mathcal{W}}$ in Figure 5.3a for estimating the bound on the interference power for two flow case.*

Proof. WLOG, we assume that Y_2 must be part of the interfering set cover. Next, for proving this claim, we subdivide the angular region around point Y_2 into four

regions, demonstrated in Figure 5.3a. For region I and IV we can show that the maximum interference power from any flow, placed along any line in that region, is upper bounded by the interference power from a flow located on $l_{\mathcal{Z}}$ as shown in Figure 5.3a. For any random line l in Zone I, the next interfering nodes on either side of Y_2 are, say, P_1 and P_2 while the same for l_Z are Z_1, Z_3, respectively. From triangular geometry, $||XP_1|| \geq ||XZ_1||$ as $||Y_2P_1|| = ||Y_2Z_1|| = D_1$ whereas $||XP_2|| \geq ||XZ_3||$ (Due to the presence of node Y_4). Thus the interference power from Z_1 is greater than or equal to P_1, and the interference from Z_3 is greater or equal to the interference from P_2. This way we can show that the maximum interference power from a flow located along $l_{\mathcal{Z}}$ is always ahead of the same for l with the same number of interferers on either side of Y_2. Furthermore, using the properties of an annulus along with Lemma 5.3.3, it can be easily shown that the length of l is less than the length of $l_{\mathcal{Z}}$ and therefore can support less number of simultaneously interfering nodes than $l_{\mathcal{Z}}$. Thus, the maximum interference power from a flow on l is less than the maximum interference power from a flow on $l_{\mathcal{Z}}$. Due to symmetry, we can similarly prove that the interference power from a flow located along any line l in Zone IV is always upper bounded by the maximum interference power from a flow located along $l_{\mathcal{Z}}$.

Now, for region II and III, we claim that interference power from a flow located along any random line segment l is always upper bounded by the maximum interference power of a flow located along $l_{\mathcal{W}}$. In such cases, the power from P_2 is

less than the power from Y_3, whereas the power from P_1 is greater than the power from W_1, or vice versa. Thus, there is no straightforward dominance of the power from either line segment. Instead, the sum of the power dominates for $l_{\mathcal{W}}$. To show this, we perform a brute force simulation algorithm where we first add up the total interference power from Y_3 and W_1, and P_1 and P_2, respectively, which verified that the former is always higher than later. Similarly, we perform a simulation to show that the maximum interference power from a flow along l is always upper bounded by the maximum interference power from a flow along the line $l_{\mathcal{W}}$. □

Table 5.4: Interference Set Cover Node Locations for a Flow Based Network

Line Number (Illustrated in Figures 5.3b)	
$l_{W,k}$ $\forall k \in \{0, \lfloor \frac{(D_2-D_1)}{2D_1} \rfloor\}$	$\{((2k+1)\frac{D_1}{2}, \pm(\frac{\sqrt{3}}{2}D_1 + jD_1))\}$ $\forall j \in \{0, 1, \cdots, N_{W,k}\}$
$l'_{W,k}$ $\forall k \in \{0, \lfloor \frac{(D_2-D_1)}{2D_1} \rfloor\}$	$\{(-(2k+1)\frac{D_1}{2}, \pm(\frac{\sqrt{3}}{2}D_1 + jD_1))\}$ $\forall j \in \{0, 1, \cdots, N_{W,k}\}$
$N_{W,k} = \lfloor \frac{\left(D_2^2 - \frac{\{(2k+1)D_1\}^2}{4}\right)^{\frac{1}{2}} - \frac{\sqrt{3}}{2}D_1}{D_1} \rfloor$	

Using lemma 5.3.4 and the greedy logic, we can find the set of nodes on $l_{\mathcal{W}}$ (or $l_{\mathcal{Z}}$) that will result in highest interference power, detailed in Table 5.4. Surprisingly, the maximum power interference set cover node locations on l_Z are same as the line l_1 of *Configuration 1.* Now, the inter flow interference power is $\max\{P_{\mathcal{I}}^{l_{\mathcal{W}}}(d), P_{\mathcal{I}}^{l_{\mathcal{Z}}}(d)\}$, where $P_{\mathcal{I}}^{l_{\mathcal{W}}}$ and $P_{\mathcal{I}}^{l_{\mathcal{Z}}}$ denotes the total maximum interference power for nodes in line $l_{\mathcal{W}}$ and $l_{\mathcal{Z}}$, respectively. Next, we extend this concept to M flow scenario i.e.,

maximum $M-1$ interfering flows. For a fixed pair of transmitter and receiver node of a flow with $M-1$ interfering flows, we need to consider two class of configurations. The mean inter-flow interference power bound of the *first class of configurations* is calculated by summing up the total interference power of the first M' lines from the set $\{l_1, l'_1, l_2, l'_2, \cdots, l_K, l'_K\}$ in Figure 5.1a, where $K = \lfloor \frac{2D_2}{\sqrt{3}D_1} \rfloor$ and $M' = \min\{M-1, 2K\}$. Now, for the bound estimation of the *second class of configurations*, we consider the line segment joining the closest pair of nodes at any point in time. More precisely, we choose M' pairs of nodes from the pairs illustrated in Figure 5.3b as $\{(Z_1, W_1), (Z_2, W_2), (Z'_1, W'_1), (Z_3, W_3), \cdots, (Z'_K, W'_K)\}$ where $K = \left(\lfloor \frac{(D_2 - D_1/2)}{D_1} \rfloor + 1 \right)$, $M' = \min\{M-1, 2K\}$, and the pairs are sorted in terms of the respective distances to the receiver. Thus, the flows situated along lines $l_{\mathcal{W},i}$ and $l'_{\mathcal{W},i}$, $i \in \{1, 2, \cdots, K\}$ determine the second type of interference bound in our estimation. Next, we compare these two bounds and take the maximum of them as the estimated interference power bound.

5.4 Simulation Results

In this section, we verify our proposed d dependent bounds on the interference and SIR, through a set of MATLAB 8.1 based experiments performed on a machine with 3.40 GHz Intel i7 processor and 12GB RAM. For this set of experiments, we fix the values of the transmitter powers and the path loss exponent at $P_t = 1Watt$ and $\eta = 2.2$, respectively. The value of $\eta = 2.2$ is motivated by our experiences

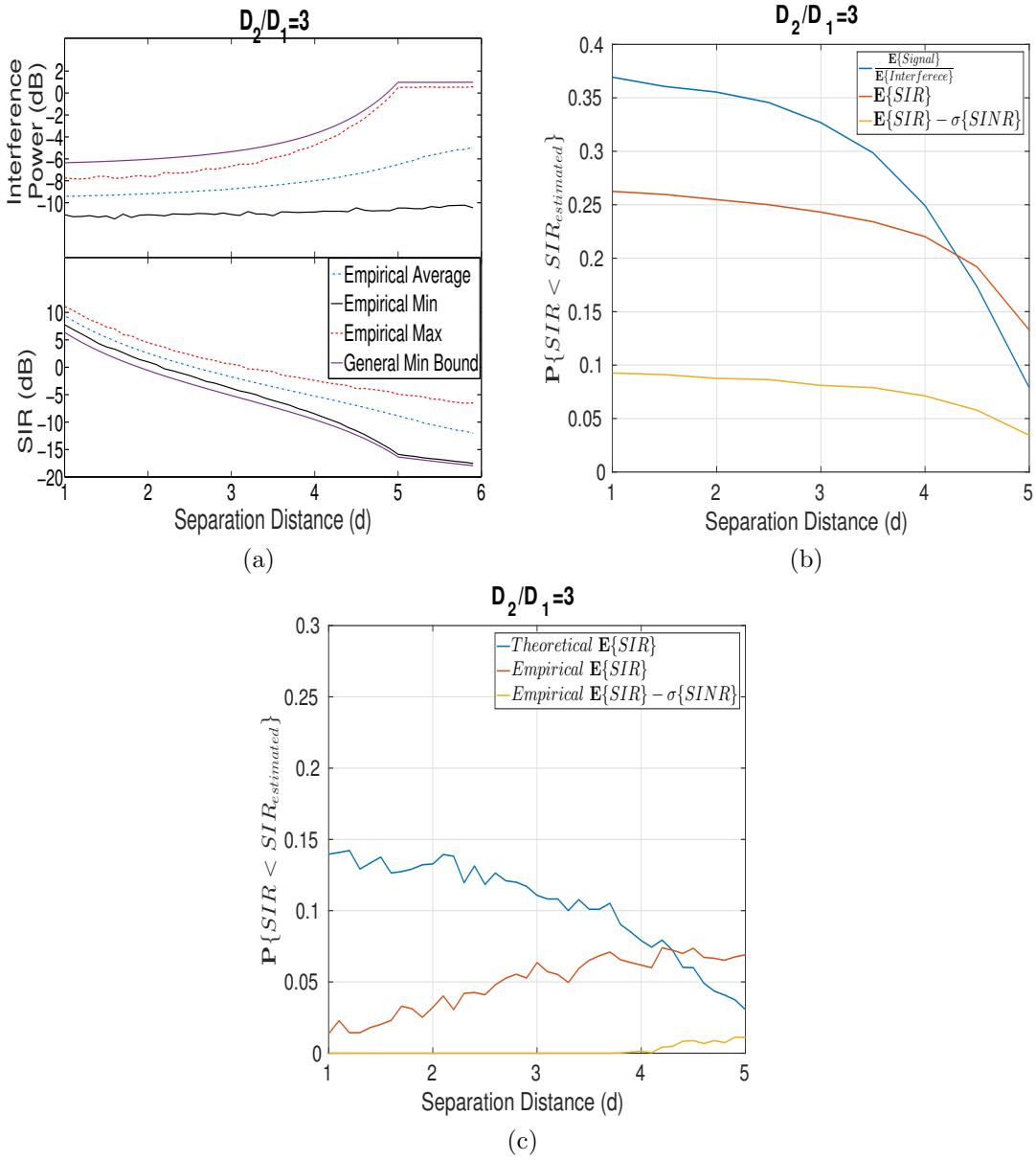


Figure 5.4: (a) Validation of Estimated Interference Power (Top) and SIR (Bottom) Bounds in dB, for Dense Network with No Fading (b) Probability that Actual SIR is Lower than the Estimated Minimum SIR in Presence of Log-Normal Fading with Variance $\sigma^2 = 4$ (c) Probability that Actual SIR is Lower than the Estimated Minimum SIR with NO Fading but in Presence of 10 Orthogonal Codes

from real-world experiments presented in Chapter 4. As a measure of the annular transition region area, we choose the ratio of $\frac{D_2}{D_1} = \{3, 6\}$ as the typical RSSI

CCA thresholds are separated by 10dB to 15dB [151]. The absolute value of D_1 is randomly selected to be $6m$ as *the major factors that controls the performance is the* $\frac{D_2}{D_1}$ *ratio, not the absolute values of* D_1 *and* D_2.

First, we verify the bounds for a general dense network, where the interfering nodes are uniformly distributed over the annular transition region around T. To verify the bounds, we randomly generate 1000 sets of interfering nodes, for a fixed value of d, using Algorithm 1. In Figure 5.4a, we compare our estimated interference power and estimated SIR, with the interference powers and SIR of the generated $\mathcal{I}^S$ sets, for no fading scenario and $\frac{D_2}{D_1} = 3$. Figure 5.4a clearly validates our d dependent interference and SIR bounds for a general dense network in absence of fading. Next, we perform similar experiments but in the presence of lognormal fading of variance $\sigma^2 = 4$ and $\frac{D_2}{D_1} = 3$. In this set of experiments, we empirically compute the probabilities, $\mathbb{P}(SIR_{\mathcal{I}^S} < \mu_{SIR_X(d)})$, $\mathbb{P}(SIR_{\mathcal{I}^S} < \mu_{SIR_X(d)} - \sigma_{SIR_X(d)})$ and $\mathbb{P}(SIR_{\mathcal{I}^S} < \frac{\mathbb{E}(Signal)}{\mathbb{E}(Interference}))$ and plot the results in Figure 5.4b. Figure 5.4b shows that the estimated SIR mean (from Eqn (5.4)) is higher than the actual SIR for around 25% of the cases, while $\mu_{SIR_X(d)} - \sigma_{SIR_X(d)}$ is higher than the actual SIR for only 10% of the case. Thus, if we were to choose a deterministic value for the bound rather than a distribution, $\mu_{SIR_X(d)} - \sigma_{SIR_X(d)}$ is considered as a good estimate. Next, we validate our orthogonal code based SIR bounds for a code alphabet of cardinality 10, while the maximum number of simultaneously interfering node is 38 (For $D_2/D_1 = 3$). The results are plotted in Figure 5.4c, which shows

that our proposed bound also works well in presence of orthogonal codes for log normal fading scenario.

Algorithm 1 Generate a random set of Interferer

procedure GENERATE( )

Initialize a Dense Set of Nodes: $\mathcal{I}^D$

Initialize $\mathcal{I}^S$ as a empty set

while $\mathcal{I}^D$ is not Empty **do**

Randomly select $v \in \mathcal{I}^D$

$\mathcal{I}^S = \mathcal{I}^S \cup v$

$\mathcal{B}_v = \{i | i \in \mathcal{I}^D \ \& \ d_{iv} < D_1\}$

$\mathcal{I}^D = \mathcal{I}^D \setminus \mathcal{B}_v$

end while

end procedure

Similar to the generic dense wireless network, we perform a set of bound tests for the robotic network scenario for $\frac{D_2}{D_1} = 3$. In this case, we randomly select two pairs of endpoints (i.e., we consider a 3 flow network) along the circumference of the outer circle with radius D_2, which are the flow endpoint for the two other flows. Next, we place a dense set of points along each of the randomly selected flow segments as well as the line segment joining the transmitter T and the receiver X to include the intra-flow interference. Then, we use Algorithm 1 to generate 1000 sets

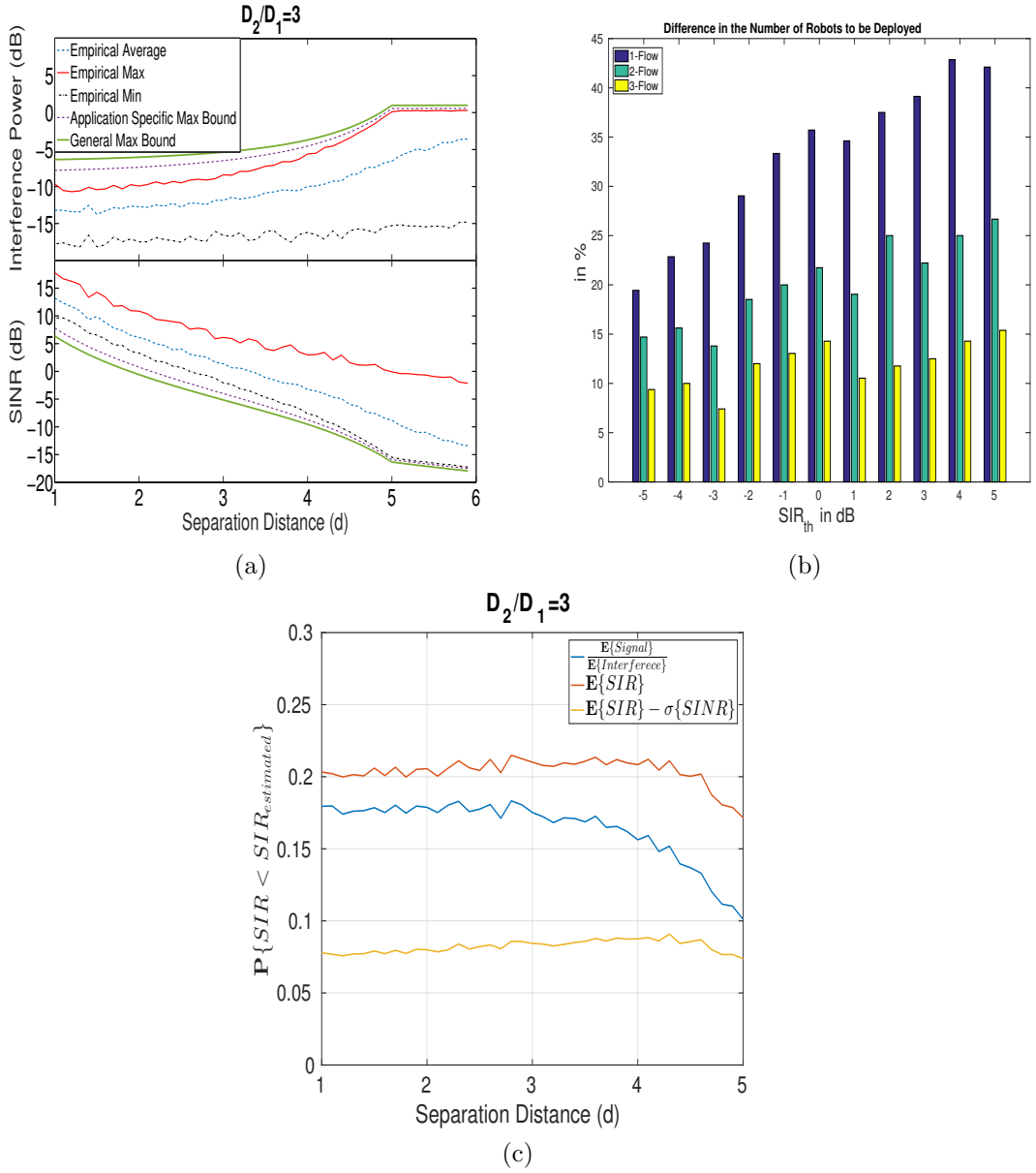


Figure 5.5: For a 3 Flow Network: (a) Validation of Estimated Interference Bound (Top) and SIR Bound (bottom) with No Fading (b) Illustration of Less Number of Robots to be Deployed with Our Application Specific Bound (No Fading) (c) Probability that Actual SIR is Lower than the Estimated Minimum SIR with Log-Normal Fading (Variance $\sigma^2 = 4$)

of interfering nodes for each value of d and for each of the 500 randomly generated sets of flow endpoints. In all cases, the total interference power is bounded

by our proposed theoretical maximum interference power, for no fading scenario, as illustrated in Figure 5.5a. Figure 5.5a also shows that our application specific bounds are much tighter than the generic bound. In order to illustrate the impact of this improvement, we also plot the difference in the number of robots required to cover a distance of $100m$ for different values of $SIR_{th} \in [-5dB, 5dB]$ in Figure 5.5b for $\frac{D_2}{D_1} = 3$. Figure 5.5b clearly illustrates that with our improved bound, the required number of robots to guarantee some target SIR requirements, is significantly lower than the generic bound based number of robots estimations, ranging from a maximum of $\sim 45\%$ for single flow network to a minimum of $\sim 10\%$ for a three flow network. The improvement is significant for less number of flows, as for a higher number of flows ($\sim 6-7$ flows) the general dense network bound becomes dominant, which is quite intuitive. Next, similar to the generic bound, in Figure 5.5c we illustrate the SIR bound in presence of fading to show that the estimated $\mu_{SIR_X(d)} - \sigma_{SIR_X(d)}$ is higher than the actual SIR for only 10% of the case, for $\frac{D_2}{D_1} = 3$.

5.5 Discussion

In this study, we proposed a method for estimation of the maximum interference and minimum achievable SIR for a link of length d in an unknown environment while CSMA-CA or equivalent MAC layer protocols are employed. First, we demonstrate a strong dependency of these bounds on the transmitter-receiver separation

distance d. Next, by considering two different scenarios: generic dense network and robotic router network; we demonstrate that we can formulate better and tighter bounds by exploiting the network topology structure which in fact improves our main goal of estimating the number of nodes to be deployed for our robotic router network in order to guarantee some network performance. We also performed a set of MATLAB based simulation results that validate our findings. The analysis and bounds presented in this chapter will guide an RWN system designer in the proper selection of some of the deployment parameters such as the number of robots required to fulfill the communication demands.

Chapter 6

Routing and Data Collection Protocol for RWN

In this chapter, we present our third study which is focused on efficient routing and data collection in a network of robots.[1] While there exists a large body of routing related works in mobile ad-hoc networks [155, 156], most of these works do not take into account the controllability as a new routing dimension. Moreover, the controllability demands an efficient routing protocol with delay guarantees. We want to fill in this gap of the controllability aware routing and data collection by building on top of a recently-developed queue-aware backpressure routing policy for multi-hop wireless networks, inspired by Thermodynamics, called the Heat Diffusion (HD) algorithm [2]. To this end, we present the first-ever decentralized version on the implementation of the Heat Diffusion (HD) algorithm [2] for data collection in RWN. We refer to this distributed version of HD protocol as the Heat Diffusion Collection Protocol (HDCP). We choose a back-pressure based routing for our study as backpressure protocols are proven to be effective for data collection with

[1]The material in this chapter is based in part on the work in [154].

low overhead in wireless networks with dynamic link qualities [102]. This leads us to believe that it is also an effective choice of routing protocol for a network of robots. We choose HD algorithm in specific as it minimizes the overall queue congestion of the network among the class of throughput optimal algorithms that make decision based on only current queue occupancies and channel statistics as well as provides additional significant improvements in average queue sizes (delay) and average routing costs (such as ETX) compared to traditional Backpressure routing [68].

Table 6.1: List of Symbols

Variables		
Symbol	Value Set	Description
q_i	$\mathbb{Z}_{\geq 0}$	The Queue Backlog at Node i
q_{ij}	$\mathbb{Z}$	The Queue Differential Between Node i and Node j
μ_{ij}	$\mathbb{R}_0^+$	Capacity of Link ij
ω_{ij}	$\mathbb{R}$	The Calculated Weight of Link ij
f_{ij}	$\mathbb{Z}_{\geq 0}$	The Number of Packets to be Sent for HD
Parameters		
Symbol	Value Set	Description
β	$[0, 1]$	Pareto Optimal Trade-off Control Parameter for HD

6.1 The Heat Diffusion Routing: Theory and Concepts

In this section, we detail the basic concepts behind the Heat Diffusion Routing algorithm by building on top the concepts presented in Section 2.3. The Heat Diffusion (HD) [2] routing has been derived from the combinatorial analogue of

classical Heat-Diffusion equation in Thermodynamics. It uses the information about estimated link capacities, $\mu_{ij}(n)$, link cost factor, $\rho_{ij}(n)$, and queue backlogs, $q_i(n)$ $\forall i \in \mathcal{V}$, to make the routing decisions to put f_{ij} packets at each time slot n. The optimization goal of the HD algorithm for a wireless network in theory can be described as follows [2, 111]:

$$\begin{aligned} &\text{Minimize: } (1-\beta)\overline{Q} + \beta\overline{R} \\ &\text{Subject to: (1) Throughput optimality, and} \\ &\qquad\qquad\quad \text{(2) Network constraints} \end{aligned} \tag{6.1}$$

where $\overline{R} = \sum_{ij\in\mathcal{E}} \overline{\rho_{ij}(f_{ij})^2}$ is the Dirichlet average routing cost, $\overline{Q} = \sum_{i\in\mathcal{V}} \overline{q_i}$ is the average network queue size, and $\beta \in [0, 1]$ is the control parameter to determine the trade-off between these two optimization goals. Note that β is the only controllable parameter in the HD formulation. *Throughput optimality* for a routing algorithm refers to its ability to maintain all queues stable for all sets of arrival rates for which it is possible by an omniscient router to maintain stable queues. Network constraints include constraints on link rates as well as interference constraints. Next, we give more concrete details on the HD routing algorithm. For clarity, we summarize the symbols used in Table 6.1. Similar to the classic BP routing algorithm, the HD routing algorithm has three steps: Link Weighing, Scheduling, and Forwarding.

6.1.1 Link Weighing

Calculate the number of packets the link will transmit if it is activated at time-slot n. In the original literature, this quantity is denotes by $\widehat{f_{ij}}(n)$, calculated as follows:

$$\begin{aligned} \widehat{f_{ij}}(n) &= \min\{\phi_{ij}(n)q_{ij}(n)^+, \mu_{ij}(n)\} \\ \phi_{ij}(n) &= (1-\beta) + \beta/\rho_{ij}(n) \end{aligned} \tag{6.2}$$

where the Lagrange parameter β is defined in Eqn (6.1). Now, the link weights are calculated as follows:

$$w_{ij}(n) = 2\phi_{ij}(n)q_{ij}(n)\widehat{f_{ij}}(n) - \widehat{f_{ij}}(n)^2 \tag{6.3}$$

6.1.2 Scheduling

Find a scheduling vector $\pi \in \Pi$ such that:

$$\Gamma(n) = \arg\max_{\pi\in\Pi} \sum_{ij\in\mathcal{E}} \pi_{ij} w_{ij}(n) \tag{6.4}$$

and the ties are broken randomly.

6.1.3 Forwarding

At this step, send $\widehat{f_{ij}}(n)$ number of packets over the link ij if $\pi_{ij}(n) = 1$ and $w_{ij}(n) > 0$. However, $\widehat{f_{ij}}(n)$ may be a fractional number. Therefore the actual number of packets transmitted is:

$$f_{ij}(n) = \begin{cases} \lceil \widehat{f_{ij}}(n) \rceil & \text{if } \pi_{ij}(n) = 1 \\ 0 & \text{otherwise} \end{cases} \tag{6.5}$$

Note that, unlike BP routing, a node running HD routing sends $f_{ij}(n)$ number of packets rather than transmitting at the full capacity, $\mu_{ij}(n)$.

In Table 6.2, we summarize the side by side comparison of the BP and the HD algorithms. For a more detailed comparative analysis of the theoretical BP and HD algorithms, interested readers are referred to the original HD paper [2].

Table 6.2: Contrasting HD policy with V-parameter BP policy (adapted from [2])

Weighing	$\widehat{f_{ij}}(n)$	BP	$\min\{\mu_{ij}(n),\ q_i(n)\}$
		HD	$\min\{(1-\beta+\beta/\rho_{ij}(n))q_{ij}(n)^+,\ \mu_{ij}(n)\}$
	$w_{ij}(n)$	BP	$\mu_{ij}(n)(q_{ij}(n) - V\rho_{ij}(n)\,\mu_{ij}(n))^+$
		HD	$2(1-\beta+\beta/\rho_{ij}(n))q_{ij}(n)\widehat{f_{ij}}(n) - \widehat{f_{ij}}(n)^2$
Scheduling	$\Gamma(n) = \arg\max_{\pi\in\Pi} \sum_{ij\in\mathcal{E}} \pi_{ij} w_{ij}(n)$		
Forwarding	$f_{ij}(n) = \begin{cases} \lceil \widehat{f_{ij}}(n) \rceil & \text{if } \pi_{ij}(n) = 1 \\ 0 & \text{otherwise} \end{cases}$		

6.2 The Heat Diffusion Collection Protocol: From Theory to Reality

In this section, we detail the Heat Diffusion Collection Protocol (HDCP) and the modifications made to the theoretical HD protocol for practical implementation.

6.2.1 Predecessors

Before detailing the HDCP, we present a brief overview of two of the well-known data collection routing protocols for a side by side comparison: Backpressure Collection Protocol (BCP) and Collection Tree Protocol (CTP). Moreover, these two protocols gave us insights on molding the promising theoretical HD algorithm into a real implementation.

6.2.1.1 The Collection Tree Protocol

The Collection Tree Protocol (CTP) [105] is a tree based, best-effort, anycast data collection protocol that was first introduced in [157]. There have been many practical implementations of CTP among which CTP Noe, presented in [105], is the most popular one. The main idea of CTP is to maintain minimum cost trees to a set of nodes that advertise themselves as the data sink/tree roots. The distance/cost used in this context is in terms of the well-known metric called ETX. Each node

calculates the shortest distance to a sink/root in terms of ETX and uses the respective next hop neighbor to send the data. In CTP, there exist two types of packets: data packets and routing packets. While the data packets are used for actual data transmission, the routing packets are used solely to setup/update the tree. For setting up the tree, CTP uses a variant of the Trickle algorithm [158]. CTP uses an adaptive beaconing technique to identify the neighbors, to calculate the shortest path, and to adapt to node failures or link quality changes. To avoid routing loops, CTP uses a datapath validation technique. In this technique, if a node receives a packet from a node with lower or equal distance/cost (in terms of ETX) to a root, it triggers a router repair phase and retries after a timeout. In contrast, neither BCP nor HDCP relies on a predetermined routing path.

6.2.1.2 The Backpressure Collection Protocol

The Backpressure Collection Protocol (BCP) [102] is a distributed dynamic routing protocol which practically implements the idealized V-parameter BP algorithm without the need for a global max weight scheduling. In this protocol, the link penalty, $\rho_{ij}(n)$, in (2.7) is replaced by $\overline{ETX_{ij}}(n)$, which is the ETX estimate for link ij at time-slot n. Therefore, the modified weighing function is as follows:

$$w_{ij}(n) = \mu_{ij}(n)(q_{ij}(n) - V.\overline{ETX_{ij}}(n)) \tag{6.6}$$

In this distributed protocol, each node calculates the weight for each of its outgoing links locally and chooses the neighbor with the maximum positive weight, if any, to forward the next packet. It is shown in [102] that a last-in-first-out (LIFO) queue implementation of BCP is better than first-in-first-out (FIFO) in terms of delay performance. Also, floating or virtual queues are implemented to deal with the problem of limited buffer size.

6.2.2 The Heat Diffusion Collection Protocol

The original HD algorithm is a centralized protocol where at each time slot the optimum non-interfering schedule must be computed. In our distributed implementation of the Heat Diffusion Collection Protocol (HDCP), every node decides the next hop locally and greedily based on the weight calculations. *Moreover, following the logic of BCP the penalty/cost factor* $\rho_{ij}(n)$ *in* (6.2) *is replaced by* $\overline{ETX_{ij}}(n)$ *which is the estimated* ETX *of the link* ij *at time-slot* n. Thus, the modified equations to calculate the link weights are as follows:

$$\begin{aligned} \widehat{f_{ij}}(n) &= \min\{\phi_{ij}(n)q_{ij}(n)^+, \mu_{ij}(n)\} \\ \phi_{ij}(n) &= (1-\beta) + \beta/\overline{ETX_{ij}}(n) \\ w_{ij}(n) &= 2\{(1-\beta) + \beta/\overline{ETX_{ij}}(n)\}q_{ij}(n)\widehat{f_{ij}}(n) - \widehat{f_{ij}}(n)^2 \end{aligned} \tag{6.7}$$

Now, each node calculates the weight for each of its outgoing links and chooses the link with the maximum positive weight. Note that, most of the variables

in the weight calculation can be estimated or calculated during the operation if provided with a value of β. We explain the choice of β in the next section, followed by detailed descriptions of other components of the HDCP such as distributed weight calculation, link ETX estimation, and our proposed link switching method to improve the link qualities. Moreover, unlike the centralized algorithm's NP-hard maximum weight independent set time scheduling to avoid interference, our distributed protocol handles interference by adaptive retransmissions and CSMA based MAC access.

6.2.2.1 The β Parameter

In theory, for different choices of β, we should get different performance for HDCP as the optimization goal changes for different values of β (Note that this parameter is not part of the CTP and BCP formulations). If we choose $\beta = 0$, Eqn. (6.7) will be simplified to:

$$\widehat{f_{ij}}(n) = \min\{q_{ij}(n)^+, 1\} \qquad w_{ij}(n) = 2q_{ij}(n)\widehat{f_{ij}}(n) - \widehat{f_{ij}}(n)^2 \tag{6.8}$$

Now, for $q_{ij}(n) > 0$, $\widehat{f_{ij}}(n) = 1$ as the queue differential can take only integer values. Therefore, Eqn (6.8) can be rewritten as follows:

$$w_{ij}(n) = \begin{cases} 2q_{ij}(n) - 1 & \text{if } q_{ij}(n) > 0 \\ 0 & \text{Otherwise} \end{cases} \tag{6.9}$$

Therefore, the optimization goal becomes similar to the goal in the original "pure" BP routing (by Tassiulas and Ephremides [68]) and it does not include the minimization of ETX. Also, as the link weights solely depend on the queue differentials, the delay performance should be better *provided that the links are all very good*[2]. On the contrary, since this protocol does not try to minimize the ETX, it can choose a bad link[3] if the queue gradient on that link is the largest. As a consequence, the average number of retransmissions faced by a packet also increases which is directly translated to larger end to end delay. Therefore, if the overall path costs in terms of ETX is the dominant factor in the end to end delay calculation, HDCP with $\beta = 0$ may perform poorly in practice.

On the other hand, if $\beta = 1$, Eqn. (6.7) will be as follows:

$$\widehat{f_{ij}}(n) = \min\{\frac{q_{ij}(n)^+}{\overline{ETX_{ij}}(n)}, 1\} \qquad w_{ij}(n) = 2\frac{q_{ij}(n)}{\overline{ETX_{ij}}(n)}\widehat{f_{ij}}(n) - \widehat{f_{ij}}(n)^2 \quad (6.10)$$

Similar to the previous case, we can simplify (6.10) as follows:

$$w_{ij}(n) = \begin{cases} 2\left(\frac{q_{ij}(n)}{\overline{ETX_{ij}}(n)}\right) - 1 & \text{if} \frac{q_{ij}(n)}{\overline{ETX_{ij}}(n)} \geq 1 \\ \left(\frac{q_{ij}(n)}{\overline{ETX_{ij}}(n)}\right)^2 & \text{if } 0 < \frac{q_{ij}(n)}{\overline{ETX_{ij}}(n)} < 1 \\ 0 & \text{Otherwise} \end{cases} \quad (6.11)$$

[2]A link is very good/perfect if the ETX of the link is 1 which implies that every packet transmission is successful on that link.

[3]We consider a link to be very bad if the ETX is ≥ 5 i.e., one successful transmission per every five transmissions.

In this case the optimization goal is mainly the reduction of overall path costs in terms of ETX. Thus the overall ETX for a path should be improved for this case. However, this might result in a slight increase in hop counts if the links are very lossy. The first and last cases in Eqn. (6.11) correctly fulfill our routing requirement. But the second case causes inefficiency in the real testbed experiments. In such cases, even if the ETX cost is very high for a link and the queue differential is as low as 1, a node will try to send the packet to that link according to the original HD rule. Moreover, in practical experiments, the probability of falling under such a situation is very high. Thus, it will negatively affect the overall performance of the HDCP and needs to be avoided. Furthermore, provided that we have avoided any such situations and have a good link with $ETX = 1$, a queue differential of 1 will still result in a positive weight thereby causing the protocol to forward the packet. This results in an absence of a steady state queue gradient on such links. This can potentially increase the number of hops traversed by the packets and also deteriorates the goodput. In order to avoid both of these situations, we replace the $\rho_{ij}(n)$ in Eqn (6.2) by $\mathbf{V} \times \overline{ETX_{ij}}(n)$ which modifies Eqn. (6.7) as follows:

$$w_{ij}(n) = 2\{(1-\beta) + \frac{\beta}{\mathbf{V} \times \overline{ETX_{ij}}(n)}\} q_{ij}(n) f_{ij}(n) - f_{ij}(n)^2 \qquad (6.12)$$

where $f_{ij}(n) = \lceil \widehat{f_{ij}}(n) \rceil$.

By setting $\mathbf{V} \geq 2$, we make it certain that there exists a steady state queue gradient towards the sink. Therefore, a node will consider a link only if $q_{ij}(n)$ is

greater than $\overline{ETX_{ij}}(n)$. Thus, for a link with very high ETX, the queue differential has to be higher in order to consider that link. Furthermore, this strategy also satisfies the Backpressure criterion as for $q_{ij}(n) < 0 \implies w_{ij}(n) < 0$. In Section 6.4.2, we present a practical experiment based analysis of the performance improvement as a result of this change in weight calculation.

6.2.2.2 Updating Weights

In order to calculate the weights in a distributed manner, each node requires updated information about the queue sizes of its neighboring nodes without affecting the performance of the routing task. In our distributed implementation of HDCP, we employ two techniques to do that. **First,** during a long period of inactivity, each node periodically broadcasts a beacon with its current queue status similar to common wireless access points. If a neighboring node receives this broadcast, it will update its locally stored queue differential information. **Second,** when a node sends a data packet, it includes its current queue state in that packet's header. Due to the nature of wireless links, every packet is received by all the neighboring nodes (we assume that no advanced MAC protocol is employed that schedules nodes to communicate in pairs at different times). Once a data packet is received by a node, it sniffs the header of the packet to extract the queue information and updates the local queue information database, even if the respective node is not the destination of the packet. Note that, BCP follows similar technique for updating link weights.

In CTP, there is no concept of queue differential based link weights. Rather, CTP uses beaconing based ETX information to predetermine the routing paths.

6.2.2.3 Queue Implementation

Similar to BCP, the practical implementation of the HDCP can have a FIFO queue or a LIFO queue implementation. Based on the observation in [102] that LIFO queue implementation has a significantly better performance in terms of end-to-end delay (which we also observed empirically), we present only the LIFO queue implementation of the HDCP protocol in this study. We also adopt the virtual "floating" queue approach proposed in [102] to prevent packet buffer overflows due to the steady-state queue gradient.

6.2.2.4 Link Metric Estimation

One of our contributions in this study is to propose a new method of ETX calculation for implementations of dynamic routing. *Initially, we opted to follow the ETX calculation technique from original BCP paper [102]. In that implementation the estimation of* $\overline{ETX_{ij}}$ *for link* ij *is performed in an online manner where the metric is updated by taking an exponential weighted moving average of the number of retransmission attempts of the most recently transmitted packet.* This is a very effective way of ETX estimation for routing protocols that do not switch next hop during retransmission i.e., use the same link ij for all the retransmission attempts.

However, for Backpressure-based dynamic routing protocols, the next hop calculation is performed before each retransmission, for path diversity. In such cases, we have to be very careful in calculating moving average since the same link may not be used for all the retransmissions. Therefore, an attempt to update the ETX for the most recently used link with the total number of retransmission attempts might lead to an erroneous ETX estimation. To avoid this flaw, we can keep track of all the links used as well as the number of tries on that link and update either only the last used link or all the links after a successful packet transmission or a packet drop.

As an alternative, we propose a 2-state discrete-time Markov Chain based ETX estimation. In this method, we assume that each link can be either good ('1') or bad ('0') at a certain point in time. With each state, we associate two transition probabilities: good to good (p_{11}), good to bad (p_{10}), bad to good (p_{01}) and bad to bad (p_{00}). Now the $\overline{ETX_{ij}}$ can be calculated as $\frac{1}{p_{01}}$ when the last state observed was a 0, and as $\frac{1}{p_{11}}$ when the last state was 1.

We maintain four counters associated with each routing table entry to keep track of different state transitions, denoted as c_{00}, c_{01}, c_{10} and c_{11}. We also add a Boolean variable to keep track of the last state of the link, i.e., if the value is *true*, the last known state was good. We initialize the c_{01} and c_{11} to be 1 and the others to be 0. Now, every time a packet is transmitted (or retransmitted), the algorithm waits for a certain period of time to receive the acknowledgment (ACK).

If received, the state of the link is set to be good ('1') otherwise it is set to be bad ('0'), and then based on the last state, the respective counter is increased by one. The counters may be reset after reaching a maximum value, to keep the ETX estimates fresh. In our experiments, we have not done this as it did not appear to affect the performance.

Now, based on the counters, the ETX is calculated (it can be shown that this corresponds to a maximum likelihood estimate of the underlying Markov Chain parameters) as follows:

$$\overline{ETX_{ij}} = \begin{cases} \frac{c_{00}+c_{01}}{c_{01}} & \text{if last State} = 0 \\ \\ \frac{c_{10}+c_{11}}{c_{11}} & \text{if last State} = 1 \end{cases} \tag{6.13}$$

All the results presented in this study are based on this new, more justifiable method of ETX calculation, which we apply to both BCP and HDCP for a fair comparison.

6.2.2.5 Link Switching

In this study, we propose an enhancement of HDCP by introducing link switching. The main concept of link switching is to maintain an ordered set of best (in terms of the weights) K neighbors (K can be any positive integer) at each node. When a packet is sent, it is first sent to the first neighbor on this list. If the transmission fails, the retransmission attempt is made immediately to the next neighbor in the

list and so on. If the list is exhausted during retransmissions, the process restarts again from the first neighbor in the list. A node should fulfill some selection criterion to be included in the list such as the link weight should be within some threshold of the best link. In our experiments, we set a threshold on the ETX and weight i.e., if a positively weighted link's ETX is no worse than the best link's ETX + 1, we add that link to the list. In section 6.4.2, we present a practical experiment based analysis of the performance improvement as a result of this change. However, we introduce this switching in HDCP only because we empirically found that it does not help to improve the performance of BCP. Note that the concept of link switching is not part of the existing BCP and CTP implementations.

6.3 Implementation Details

Similar to any data collection routing protocol, a number of common routing parameters need to be set properly in the real implementation of HDCP (in our case in the Contiki OS implementation) such as maximum queue size and the maximum number of retransmissions. In this section, we discuss the choices of such parameters and the reason behind them in details. First of all, we set the value of $\mu_{ij}(n)$ in Eqn. (6.2) to be 1 as a node cannot send more than one packet simultaneously.

6.3.1 Retransmission

Retransmission is very crucial for the performance of any wireless network. For effective retransmission, the parameters such as retransmission timeout and maximum number of retransmissions have to be properly chosen. Retransmission is also directly related to the acknowledgment mechanism and the choice of ARQ. Since the choice of ARQ affects the HDCP, the BCP, and the CTP algorithms equally, we have implemented a simple Stop and Wait ARQ mechanism where a node can send only one packet at a time and wait for its acknowledgment before moving to the next packet. If the acknowledgment is not received within a certain time, commonly referred as retransmission timeout, the node retransmits the same packet. Now, the value of this retransmission timeout directly affects the goodput of the system and needs to be properly chosen. Note that, the ARQ mechanism is employed on top of the existing hardware level acknowledgment mechanism that tries a maximum of 3 times to properly transfer the packet to the next hop in case of unicast transmissions (e.g., software acknowledgments). We do not remove the hardware level acknowledgment (One key feature of the CTP algorithm) for a fair comparison as well as to avoid the unreliability issues in pure software acknowledgments.

In our experiments, the transmission and propagation time for a packet is in the order of tens of milliseconds. It would then perhaps be expected that the best setting for the retransmission should be on the order of around 10ms or so. Nevertheless, we empirically found that it is best to set the timeout for retransmitting a

lost packet to be chosen randomly between 10 to 200 ms. We believe that this large range is needed because of the link coherence time in our testbed which is located in a busy office building environment. For instance, in [159], it is indicated that the coherence time for IEEE 802.15.4 radios can be about 175ms. Retransmitting a lost packet quicker than the coherence time runs a higher risk of seeing another packet loss. Furthermore, we use CSMA/CA as the link access protocol, which also introduces some delay.

The maximum number of retransmission attempts is set to 5 based on the original BCP code, which we empirically observed to perform well on our testbed. After five retransmission attempts, if a packet is not acknowledged, the node will drop it and move to the next packet.

6.3.2 Retry

Whenever a node generates or receives a packet, it tries to send it immediately (after about $4 - 5$ ms) if no other packet is being transmitted or waiting in the queue. However, when the node wants to transmit the packet, there might not be any suitable neighbor (in terms of having a positive weight) to forward the packet. In that case, the node needs to decide how much time should it wait before retrying. We refer to this wait time as the Retry time. One viable option is to constantly keep trying which is not efficient in terms of energy consumption due to radio wake times. Also once this situation happens, it might take a while to have a good neighbor.

In this work, we set the retry time to be chosen randomly between 50ms to 100ms. The intuition behind choosing this value is again the transmission time for a packet being in the order of tens of milliseconds. Based on our experiments, we have also observed that the typical packet transfer time (the time duration between the transmission and reception of a packet) is $\approx 10ms$. Therefore, by choosing a value between 50ms and 100ms, we give the neighboring nodes enough time to potentially transfer several packets which is likely to be enough to create a positive weight.

6.3.3 Queue Buffer

In practical low power low memory devices, the possible queue buffer allocations are severely restricted. We fixed the maximum queue size to be 25 as this is the highest possible number of queue buffer that our device can accommodate alongside other required memories. Along with this buffer, there also exists a small memory allocated to store only the recent packet for the retransmission purpose.

6.3.4 Beacon Timer

Beaconing is a very important part of the practical implementation of both HDCP and BCP. When a node has nothing to send for a long time, beacons are sent periodically, so that the neighboring nodes can keep their Backpressure database updated. Also, beaconing is mandatory for a sink node since it has nothing to send. Therefore we implement two different beaconing rates in our system. The first type

of beaconing is for source nodes and the period for that is around 5 seconds. The second type of beacons, which we refer to as the fast beacons, are used by the sink nodes and the period for that beacon is around 2 seconds. These values are chosen based on the original BCP code.

6.3.5 Inbound Packet Filtering

Inbound packet filtering is very important to improve the performance of both HDCP and BCP in the presence of retransmissions. If no filtering is used, a node might receive multiple copies of the same packet due to retransmissions. Therefore the node might have multiple copies of the same packet stored in the buffer simultaneously, which is not efficient. To avoid this kind of situations, we implement an inbound packet filter to drop any duplicate packets after sending proper acknowledgments. In our implementation, each node maintains a history (packet source information and the original sequence number) of 25 most recent packets received by the node. We choose this number to match the queue buffer size. Every time a node receives a packet, it checks the history, performs the necessary action such as packet drop or store, and updates the history.

Further, to prevent packet looping, we implement a TTL counter which decrements at each hop. In our experiments, sources set the initial TTL for each packet conservatively to 10 (the maximum hop distance from any node to the sink in our testbed is only 3).

6.3.6 End to End Delay Calculations

For calculating end to end delay of each packet, we maintain a separate field called $HDCP_{Delay}$ in the HDCP header, initialized with a value of 0. At the source node, the packet is timestamped at the generation (Say, A_{source}) and just before departure (Say, D_{source}), and the value $HDCP_{Delay}$ field is set to be ($D_{source} - A_{source}$). Similarly, we time-stamp the packet at each intermediate node, I_k: upon arrival (A_{I_k}) and just before departure (D_{I_k}); and add the time difference with the value of $HDCP_{Delay}$, i.e., $HDCP_{Delay} = HDCP_{Delay} + (D_{I_k} - A_{I_k})$. Thus, the value of the field $HDCP_{Delay}$ upon arrival on the sink denotes the end to end delay suffered by that packet. For illustration, assume that the travel path of a packet is $source \rightarrow I_1 \rightarrow \cdots \rightarrow I_M \rightarrow sink$, where $A_{I_1}, \cdots, A_{I_M}$ are the arrival times of the packet at the intermediate nodes, and $D_{I_1}, \cdots, D_{I_M}$ are the respective departure times. Then the end to end delay is: $\sum_{i=1}^{i=M} |D_{I_i} - A_{I_i}| + |D_{Source} - A_{source}|$. Note that, we do not add the propagation delays as the value of propagation delays are negligible compared to the queuing delay (which we measure) in our testbed setup.

6.3.7 Experimental Setup

To analyze the performance of HDCP in a real network and compare it with BCP and CTP, we have implemented the HDCP and the BCP algorithms on Contiki OS and used the CTP implementation available with the Contiki OS. We perform a set of evaluation experiments on an indoor wireless network testbed called Tutornet

[160] with forty-five IEEE 802.15.4-based Tmote-sky nodes distributed over a floor with roughly 80,000 sq.ft of an area. This testbed is also available for administered public use for approved research purposes including benchmarking protocols. The network topology is presented in Fig. 6.1 where the marked node is the sink and the rest of the nodes are the source nodes and the furthest node is three hops away from the sink. We use the channel number 26 with Tmote sky power level 31 for this purpose. The number of neighbors to each node varies from 19 to 35 with an average of 29. Nonetheless, typically only about 7-8 of the neighbors are connected via good links ($ETX \approx 1$). Thus the topology is very diverse with a considerable number of different paths between any two nodes in the network. On the negative side, a considerable amount of interference exists among the nodes, which limits the bandwidth. The data packets in our experiments are all 26 Bytes in size.

Figure 6.1: Real Experiment Testbed Topology

All the experiments are performed on weekdays during daytime with lots of moving people and physical objects around. Each experiment is performed for 35 min: the network settles down during the first 5 min and the data is collected during the next 30 min. Each experiment is repeated at least 10 times to improve the confidence levels. Note that, we discuss the experimental setup for low power stack and for node failures in Sections 6.4.5 and 6.4.7, respectively.

We evaluate HDCP's performance in terms of different values of β and different packet generation rates. We select the value of V to be 2 for both BCP and HDCP which has been empirically determined to be an efficient operating point for BCP in the original BCP paper (which we could also verify in our own experiments).[4]

6.4 Real Testbed Experiment Results and Analysis

In this section, we evaluate the HDCP protocol under different configurations (β values) and also compare them with LIFO BCP with virtual queue implementation and CTP.

[4]The source codes of these experiments are publicly available at `https://github.com/ANRGUSC/HDCP`.

6.4.1 Variation of the β Parameter

We perform a set of practical experiments on the testbed with different values of β and packet generation rates of 1 packet per 4 seconds per source (i.e., 0.25 PPS) as well as 1 packet per 2 seconds (0.5 PPS). The difference in performance is not prominent between the two source rates. Thus, we present the results only for the higher rate of 0.5 PPS in this section.

The goodput of each source node is defined to be the number of packets received by the sink from it over a one-second interval. For visual clarity, all the plots presented in this section are sorted in terms of the goodput of the individual nodes for the experiment with $\beta = 0$. The end to end delay calculation for each packet is performed by adding up all the queuing and processing delays in all intermediate nodes, as discussed in Section 6.3.6. This ignores the propagation times which in any case are negligible compared to the processing delays.

First, we analyze the goodput characteristics of HDCP for different choices of β. In Fig. 6.2a, we compare the goodput of each of the forty-four nodes for six different choices of β. Figure 6.2a clearly shows that the goodput for $\beta = 1$ and 0.8 are significantly better than the other choices of β. It also demonstrates that $\beta = 0$ results in a gradual decrease in the goodput to sink where only a few nodes are able to reach the maximum possible rate. Figure 6.2a also shows that the choice of $\beta \in \{0, 0.2, 0.4, 0.6\}$ does not significantly affect the goodput performance. Based on these observations, we hypothesize that the goodput performance of the network

is mostly dependent on the ETX of the path and, therefore, for higher β values the goodput performance metrics are better. Figure 6.2c, which shows that the average path costs for $\beta \in \{0.8, 1\}$ in terms of ETX for individual sources are significantly less than the average path costs for other choices of β, validates this hypothesis. In Fig. 6.2c, we also analyze the average hop counts observed by the packets. It shows a similar pattern as the path costs since total ETX of the path is proportional to the number of hops traversed by the packet.

In Fig. 6.2d (Bottom), we analyze the variation in the average end-to-end delay suffered by the packets generated from individual nodes for different values of β. It shows that the average delay performance for $\beta = 1$ is the best among different choices of β while any other choice of β results in a worse delay performance. Similar statistics are seen in Fig. 6.2d (Top) in terms of the average queue sizes for individual nodes. This Fig. 6.2d demonstrates that for $\beta = 1$ the average queue-sizes are almost three to four times smaller than that of the average queue sizes for $\beta \in \{0, 0.2, 0.4, 0.6\}$. We also plot the delay CDF in Fig. 6.2b for the packets generated from mote 38 in the testbed which is the mote farthest from the sink. It also shows that $\beta = 1$ is best in terms of end-to-end delay.

Summarizing all these results, we can say that HDCP performs really well if the value of β is close to 1. For lower values of β, we find the performance does not differ by too much from the performance when $\beta = 0$ (the reason for this is further discussed in section 6.5). Therefore, we only consider HDCP with $\beta = 1$ and $\beta = 0$

(the latter as a baseline scheme, which does not take into account ETX) for the rest of the study.

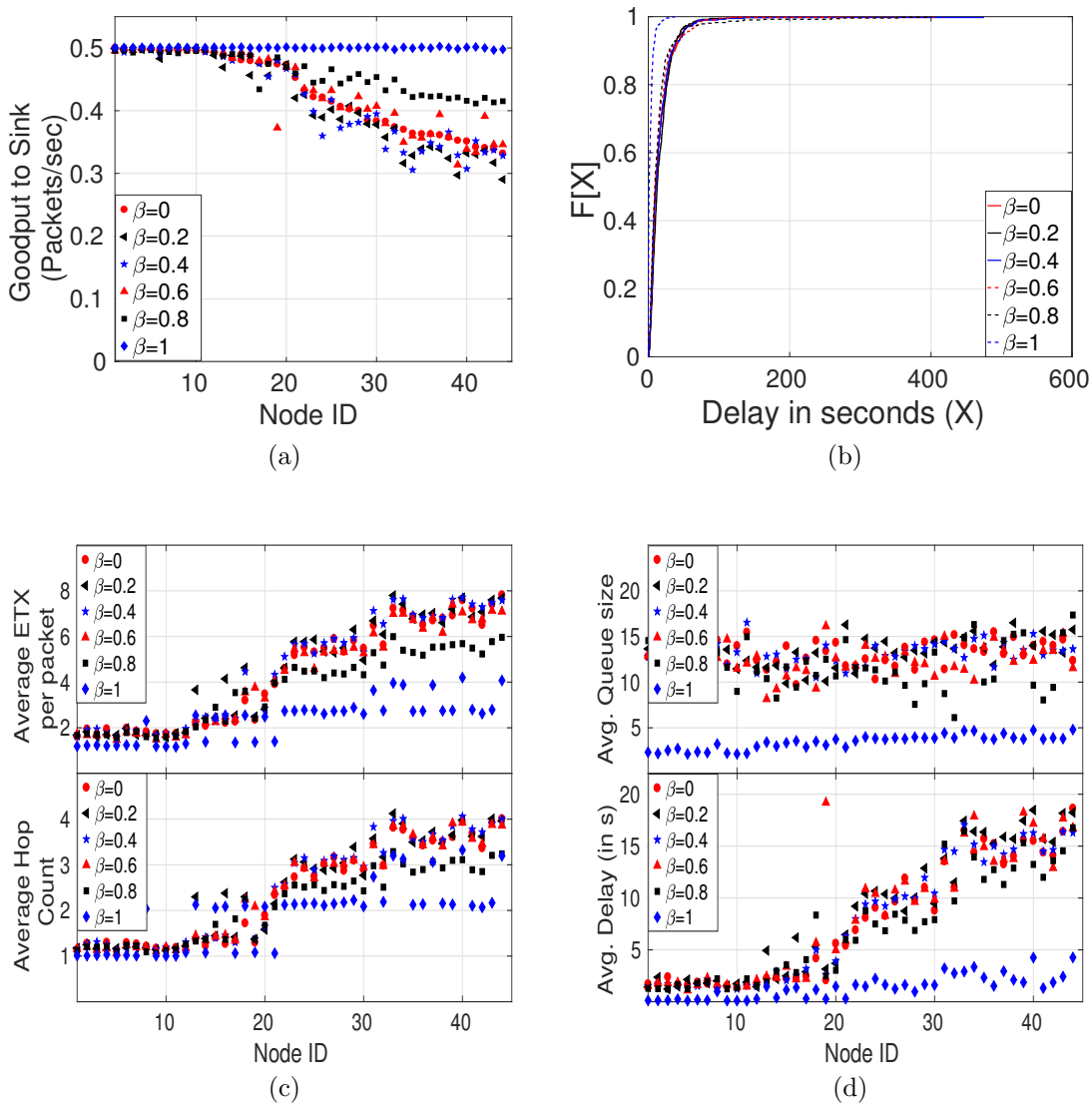


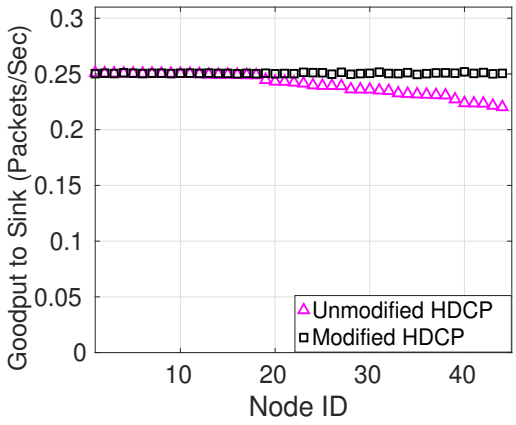
Figure 6.2: Performance Plots of HDCP Implementation for 0.5 PPS with Different Values of β: (a) Average Goodput to Sink (b) End-to-End Delay CDF Plot for Mote 38 (c) Average ETX per Packet (Top) and Average Hop Count (Bottom) (d) Average End-to-End Delay (Bottom) and Average Queue Size for Each Node (Top)

6.4.2 Modified HDCP vs Unmodified HDCP

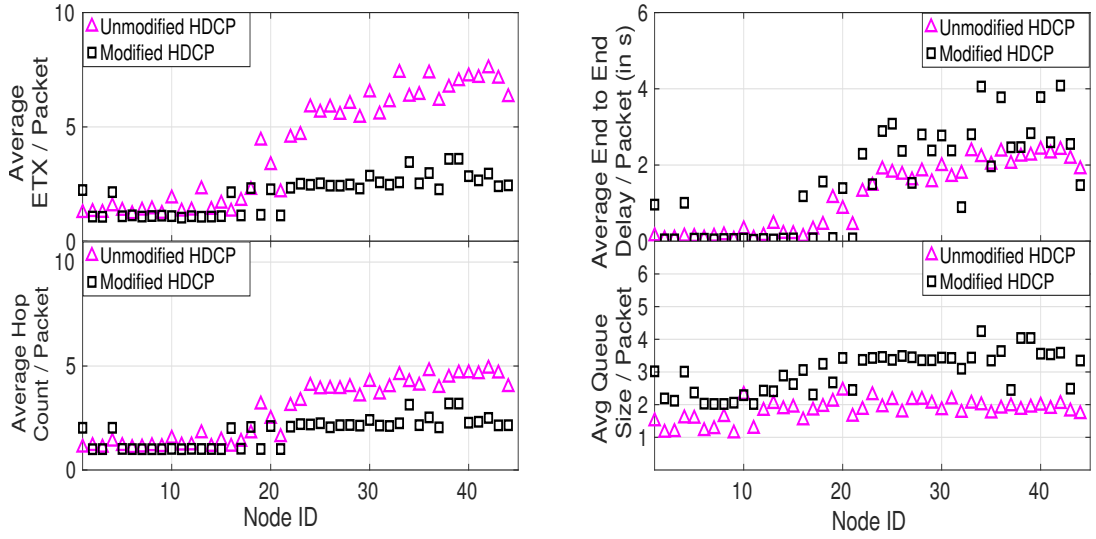
In this section, we present a comparison of an HDCP implementation based on the original weighing model suggested by the theory (Eqn. (6.7)) with our HDCP implementation where the link weight model is modified as in (6.12) as well as with our proposed link switching approach. For this purpose, we perform a set of experiments with both versions of HDCP for a fixed value of $\beta = 1$ and fixed packet generation rate of 0.25 PPS i.e., 1 packet per 4 seconds. Note that, for visual clarity, all the plots presented in this section are sorted in terms of the goodputs for unmodified HDCP implementation.

In Fig. 6.3a, we demonstrate that without the modifications we have proposed, the goodput performance of HDCP suffers significantly. This is mostly due to the selection of links with higher ETX as well as lack of proper queue gradient towards the sink, as discussed in Section 6.2.2.1. This is further verified by the Fig. 6.3b which clearly shows that the average path costs in terms of ETX for unmodified HDCP are very high compared to our HDCP implementation.

Next, we compare the performance of unmodified and modified HDCP in terms of average end to end delay as well as average queue size of individual nodes in Fig. 6.3c. It shows that the delay performance of unmodified HDCP is worse than modified HDCP for half of the nodes while it is better for the rest half of the nodes. Thus, on average, the modification does not attribute to any delay improvements. On the other hand, it is also clear from the Fig. 6.3c that there exists a steady



(a)



(b) (c)

Figure 6.3: Performance Comparison between Modified and Unmodified HDCP Implementation with $\beta = 1$ for 0.25 PPS: (a) Average Goodput (b) Average ETX per Packet (Top) and Average Hop Count (Bottom) (c) Average End-to-End Delay (Top) and Average Queue Size (Bottom)

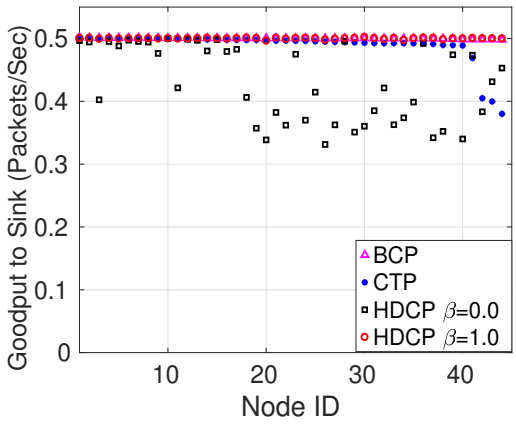
queue gradient in modified HDCP in contrary to the case of unmodified HDCP, where most of the nodes have queue size of 1 thereby lacking a proper queue gradient towards the sink. This also validates our justification for the modification of weights

in HDCP as indicated in Section 6.2.2.1. Thus, overall we improve the performance of HDCP by slightly compromising the average queue sizes.

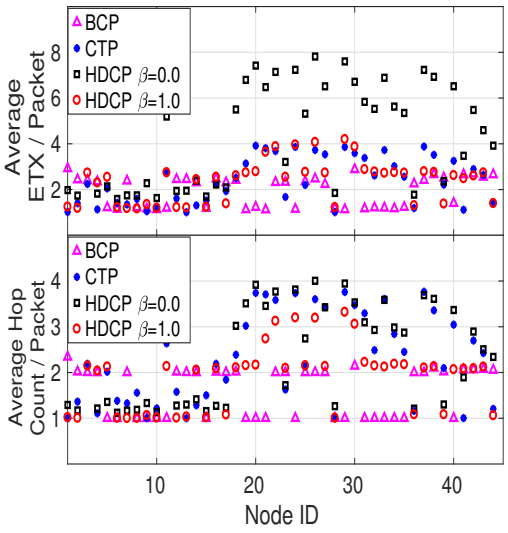
6.4.3 Performance Comparison with BCP and CTP for Fixed Packet Generation Rate

In this section, we compare the performance of HDCP with the performance of the BCP protocol and the CTP protocol for the fixed packet generation rate of 0.5 PPS, i.e., 1 packet per 2 seconds. Note that, for simplicity of presentation, all the plots presented in this section are sorted in terms of the goodputs for the BCP algorithm.

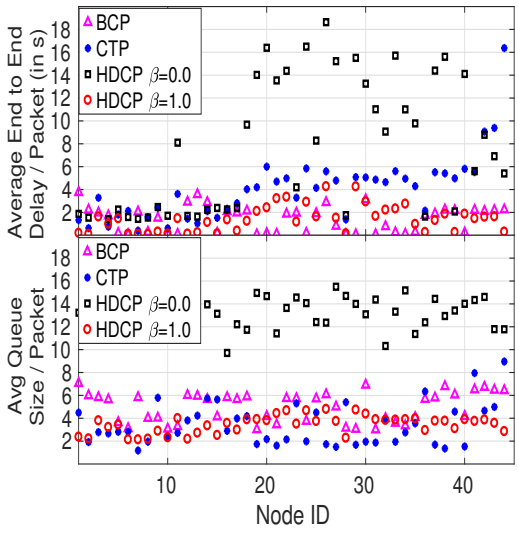
In Fig. 6.4a, we plot the goodputs for CTP, BCP and HDCP with $\beta = 0$ and 1, respectively. We observe that HDCP with $\beta = 1$ outperforms the CTP algorithm in terms of goodput while CTP outperforms HDCP for $\beta = 0$. However, BCP and HDCP with $\beta = 1$ perform almost identically. For $\beta = 0$, the weights of the links are fully determined by the queue differentials and it does not depend on ETX at all, resulting in bad performance. For CTP, a node relies on a single periodically calculated path to sink and does not take advantage of multiple available paths to sink thereby compromising the goodput for high packet generation rate such as 0.5 PPS. On the other hand, BCP and HDCP with $\beta = 1$ focus on reducing the total ETX cost of a source to sink path while not being restricted to a single pre-calculated path. Thus, the BCP and the HDCP algorithm with $\beta = 1$ appear



(a)



(b)



(c)

Figure 6.4: Comparison Plots between HDCP, BCP and CTP for 0.5 PPS: (a) Average Goodput to Sink (b) Average ETX (Top), Average Hop Count to Sink (Bottom) (c) Average End-to-End Delay (Top) and Average Queue Size (Bottom)

to be able to take advantage of the multiple paths available to the sink in order to cope with high packet generation rates thereby improving the throughput region.

Similar to the goodput analysis, we present the average hop count and average ETX of the entire path observed by the packets generated from individual sources in

Fig. 6.4b. It shows that, again, HDCP with $\beta = 1$ and BCP both slightly outperform CTP on average whereas HDCP with $\beta = 0$ performs the worst. This is also justified based on our discussion presented in the previous section. The performance of HDCP with $\beta = 1$ and the performance of BCP are again almost same. *Based on these results, we hypothesize that the similarity between BCP and HDCP with* $\beta = 1$ *is due to the similarity in their neighbor rankings, despite differences in the structure of the weight expression.* This is further explored in section 6.5.

We also compare the delay performance and queue size of HDCP with BCP and CTP in Fig. 6.4c. Figure 6.4c shows that the delay performance of HDCP for $\beta = 1$ is significantly better than HDCP with $\beta = 0$. However, based on Fig. 6.4c, the delay performance for BCP is almost same as HDCP with $\beta = 1$ while both of them outperforms CTP. The similarity between BCP and HDCP with $\beta = 1$ is justified based on the previous results. In Fig. 6.4c, we also demonstrate that the average queue size of HDCP with $\beta = 1$ is significantly low compared to BCP and HDCP with $\beta = 0$. The queue size of CTP seems to be the lowest for some nodes, however, we believe this is misleading as CTP experiences the most packet drops among the various protocols at this offered load. The packet drops in CTP occur partly due to retransmission packet drops caused by higher intra-network interference (reflected in the higher ETX and higher delay values), and partly due to some other parameters in its implementation such as forwarding packet lifetime and an in-built congestion control. However, for any higher packet generation rate,

we observe that the queue size for CTP increases rapidly (resulting in even more losses) as does its delay.

6.4.4 Varying Packet Generation Rate

In this section, we present and analyze the effects of the packet generation/source rates on the performance of HDCP and compare it with the performance of the BCP and CTP algorithms. We performed a set of experiments with six different packet generation rates: 1/12 PPS (i.e., 1 packet per 12 seconds), 1/8 PPS, 1/4 PPS, 1/2 PPS, 4/5 PPS, and 1 PPS. In Fig. 6.5a, we present the goodput variation due to the change in packet generation rate for HDCP with $\beta = 0$ and 1 as well as the goodput variations of the BCP and the CTP algorithms. It is clear from Fig. 6.5a that for lower packet generation/source rates, HDCP performs almost similar to the BCP and CTP algorithm in terms of goodput to sink. But, as we increase the offered load, HDCP and BCP gradually outperform the CTP algorithm. In our experiments, HDCP outperforms CTP in terms of goodput for packet generation rates higher than 1 packet per 4 seconds. From the Fig. 6.5a, we can estimate that the full throughput region (the maximum offered load at which the protocol is able to match the ideal curve) for HDCP is about 60 to 100% higher than that for CTP in this particular testbed and topology (of course the relative performance improvement is certainly likely to depend on the network topology.) Another thing to notice that, the average goodput for $\beta = 1$ is always higher than $\beta = 0$ which agrees with our

earlier findings and arguments concerning the inefficiencies introduced by ignoring the ETX costs of links. Yet again, the performance of BCP closely follows the performance of HDCP with $\beta = 1$ which is, again, due to the similarity in their neighbor rankings in terms of the weights. This is further explained in section 6.5.

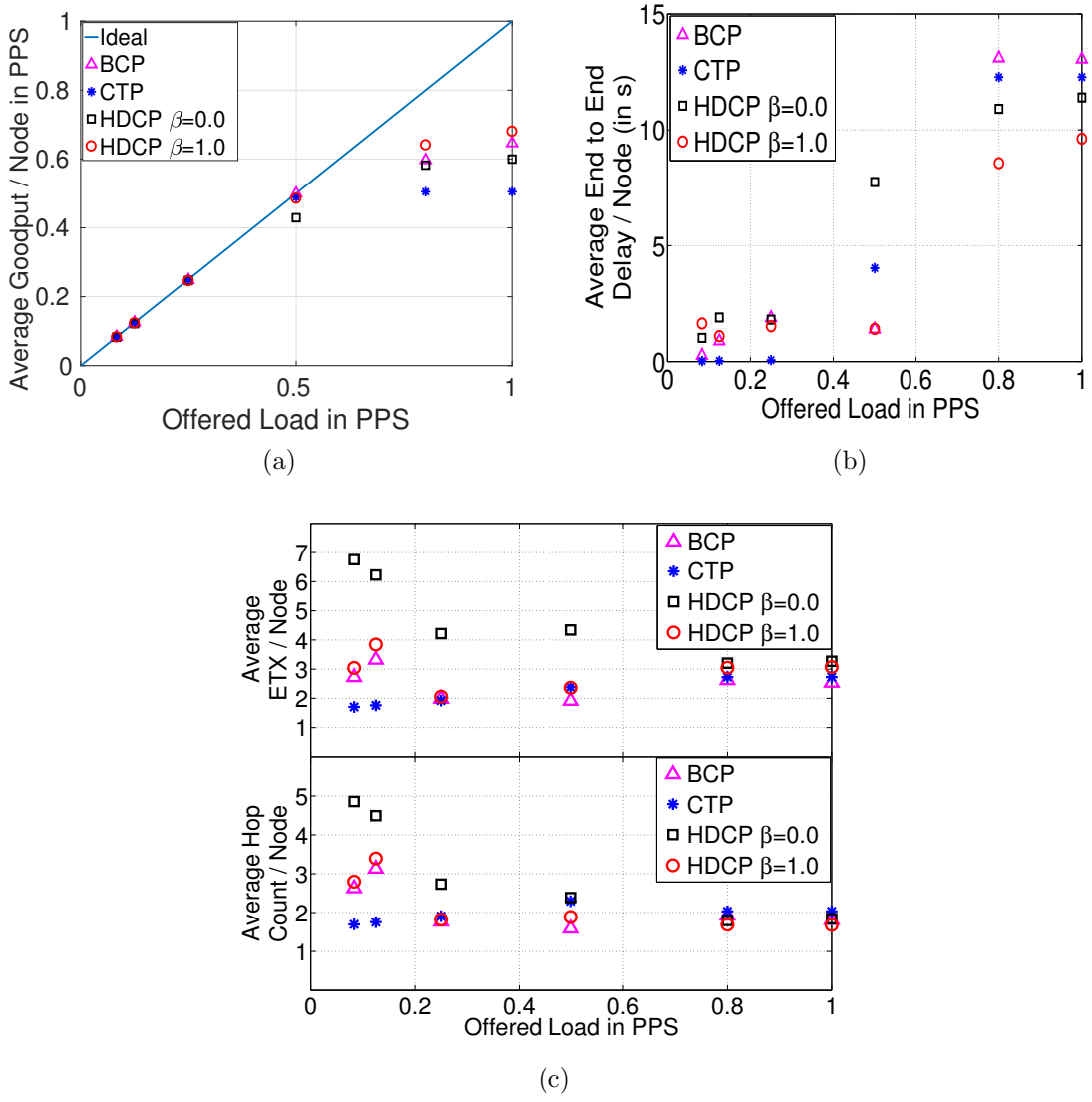


Figure 6.5: (a)Variation of Goodput for Varying Offered Load (b) Variation of Average End to End Delay for Varying Offered Load (c) Variation of Average Path Cost in Terms of ETX (Top) and Average Hop Count (Bottom) for Varying Offered Load

Next, we investigate the effects of increasing packet generation rates on the average path costs in terms of ETX and the average number of hops traversed by the packets. We plot the average ETX and average hop counts due to different source rates for HDCP, BCP and CTP in Fig. 6.5c. It is observable from Fig. 6.5c that for any packet generation rate overall path cost for HDCP with $\beta = 1$ is comparable to BCP while CTP outperforms both for packet generation rate lower than 0.25PPS and converges with them for higher rates. Moreover, the average path cost for HDCP with $\beta = 0$ is higher than $\beta = 1$ which is justified by our discussion in the previous section. Similar statistics are available from the plot of average numbers of hops encountered by each packet due to its direct relation with the overall path ETX. The similarity between HDCP with $\beta = 1$ and BCP is, again, justified based on our earlier discussions. The apparent 'good' performance of CTP is due to its increasing incapability of sending packets with long path costs to the sink as it encounters congestion drops.

Lastly, we analyze the effect of packet generation rate on the average delay in Fig. 6.5b. Although CTP and BCP outperform HDCP for source rates lower than 0.25PPS by a small margin, Fig. 6.5b demonstrates the superiority of the HDCP for $\beta = 1$ in overall delay performance as it continues to guarantee lower delay for higher packet generation rates. Another interesting fact to notice is that for BCP and HDCP, the delay increases steadily with packet generation rate whereas the delay for CTP increases rapidly with packet generation rate. This is likely because

BCP and HDCP can take advantage of multiple paths to sink whereas the CTP relies on only one path. Therefore, CTP reaches congestion earlier than BCP and HDCP which results in the rapid increase in delay. Again, the similarity between BCP and HDCP with $\beta = 1$ is due to the similarity in the neighbor ranking in terms of the weights.

To summarize, our experiments lead us to conclude that optimized combinations of queue-awareness and ETX (implemented in BCP and HDCP with $\beta = 1$) provide the best choice for routing, better than routing based on ETX alone (CTP), which in turn performs better than queue-aware routing alone (HDCP with $\beta = 0$).

6.4.5 Low Power Communication Stack Based Experiments

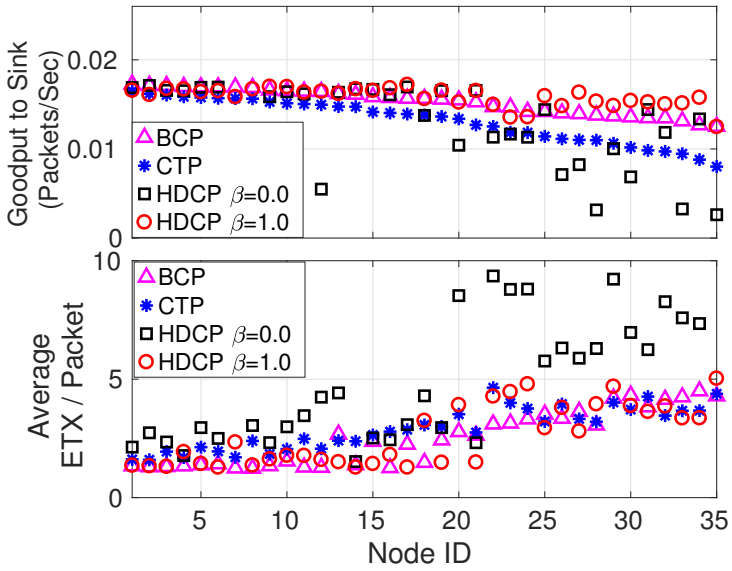


Figure 6.6: Performance Comparison of HDCP with BCP and CTP for a Low Power Communication Stack: (Top) Goodput to Sink, (Bottom) Average ETX Path Costs to Sink

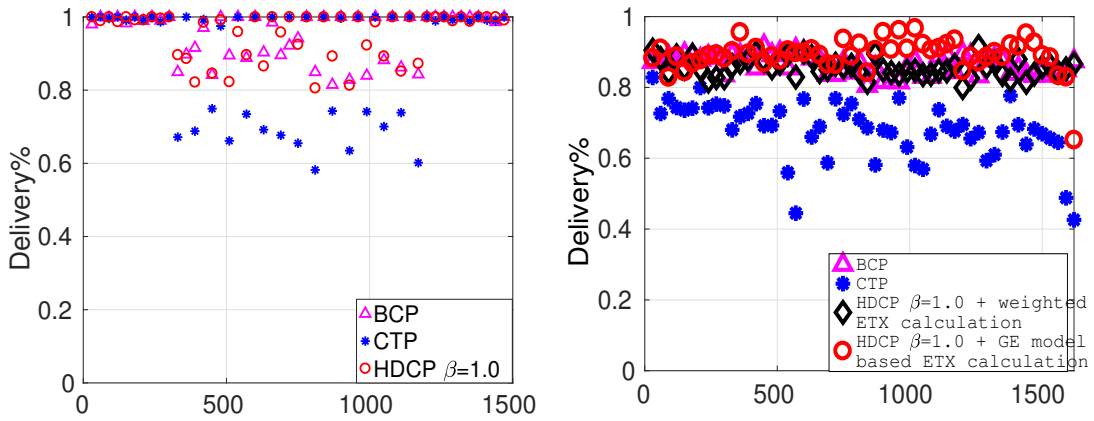
In order to verify the performance of HDCP on a low power communication stack, we performed a set of experiments with 35 sources and a sink (first 36 nodes of the testbed). For these experiments, we used CX-MAC protocol, a version of X-MAC [161] that is provided in Contiki, with a duty cycle of 5% for HDCP, BCP, and CTP. However, the choice of CX-MAC protocol over the other protocols is just a matter of the availability of Contiki implementation. Furthermore, since we are using a duty cycle, we also need to cut-back our source rates to a very low rate. For the presented set of experiments, we used a packet generation rate of 1 packet per 60 seconds (i.e., 1/60 PPS). We present the results in Fig. 6.6. Fig. 6.6 shows that the HDCP protocol with $\beta = 1$ performs well in a low power communication stack, at a very low duty cycle setting where even CTP shows some deterioration in the fairness of goodput. However, in this setting the performance of the baseline with $\beta = 0$ is much worse, leading us to conclude that it is a very poor setting indeed. Now, in order to estimate the actual energy consumptions, we record the different energy consumption components using the Contiki PowerTrace tool (in terms of the percentage of time spent in different radio phases: Transmit, Listen/receive). Based on our traces, in HDCP with 5% duty cycle, the radio of each node is on for 5.92% of the total execution time, out of which the node is transmitting and receiving approximately 0.65% and 5.27% of the total execution time, respectively. Now, to get the actual energy consumption, one can use the current and voltage ratings from the specifications of the devices used. For example, in Tmote-sky the

rated voltage of operation is approx $3.3V$ and the average current consumptions are $17.4mA$ and $19.7mA$ for radio transmission and radio reception, respectively. This results in approximately $113.78mJ$ energy consumption in each Tmote-Sky for the experiment period of 30 minutes.

6.4.6 External Interference

In this section, we evaluate the performance of the HDCP protocol with the optimized $\beta = 1$ in the presence of external interference and compare it with both BCP and CTP. This is necessary because the 802.15.4 radios share the frequency band with WiFi, Bluetooth, and other Zigbee radios and as a result of their performance often suffers from severe interference. To emulate such scenarios, we performed a set of experiments with forty sources and a single sink (Node 1) while four nodes are used as interference sources on channel 26. The interfering nodes are inactive for the first five minutes of the experiment, periodically transmit for next fifteen minutes, and become inactive again for the last five minutes of the experiment. During the on period, each of the interfering nodes transmits 110 Byte packets at a rate of 100PPS for 15 seconds and then does not transmit anything for the next 15 seconds, and so on. Furthermore, we reduced the power level of all 41 nodes from level 31 to level 15 whereas the interfering nodes were kept at level 31, in order to intensify the effect of interference. The outcome of this set of experiments is presented in Fig. 6.7a that plots the delivery percentage of the packets over a series

of 30 seconds time window for HDCP with $\beta = 1$, BCP and CTP. It demonstrates that while CTP performance significantly suffers from the interference, the HDCP protocol maintains its good packet delivery ratio, similar to BCP.



(a) (b)

Figure 6.7: Thirty Second Windowed Average Sourced Packet Delivery Ratio for: (a) Synthetically Generated Interfering 802.15.4 Channel 26 Traffic (b) Real Interference Scenario on 802.15.4 Channel 13

The above-mentioned settings are used to stay consistent with the interference settings presented in the original BCP paper[102]. However, it is well known that the simple Gilbert-Eliot model used for ETX estimation might work perfectly with some specific synthetic interference models and might fail in realistic interference scenarios. In order to explore the performance of the HDCP algorithm, in presence of real interference, we performed a set of experiments with 44 source nodes and 1 sink node, running on channel 13 of the 802.15.4 standard which is known to be one of the most interfered channels. We also compare the performance of HDCP based on the Gilbert-Eliot (GE) ETX model with the performance of BCP with the GE

model as well as with HDCP based on the ETX model used in the original BCP paper [102]. The results presented in Fig. 6.7b clearly demonstrate that even in the presence of constant real interference, the HDCP algorithm with Gilbert-Eliot ETX model performs comparable to the BCP algorithm and the HDCP algorithm with the basic ETX model presented in [102], while outperforms the CTP algorithm. Furthermore, both Figs. 6.7a and 6.7b show that the BCP and the HDCP algorithms can achieve approx 85% delivery ratio in presence of interference while the CTP achieves approx 70%.

6.4.7 Node Failures

In this section, we evaluate the performance of the HDCP protocol with the optimized $\beta = 1$ in the presence of node failures/joins and compare it with both BCP and CTP. This is necessary because node failures and node joins are very common events in an RWN. To emulate such scenarios, we performed a set of experiments with twenty sources, single sink, and twenty-five forwarding nodes, i.e., total forty-six nodes in the network. All nodes were set to transmit at the maximum power level, i.e., level 31 and on channel 26. In our experiments, we randomly turned off four of the forwarding nodes (i.e., $\approx$ 10% nodes) after five minutes from the beginning and then turn them back on after ten minutes from the beginning. Each

source node was set to transmit at 1/2 PPS. A sample outcome of this set of experiments is presented in Fig. 6.8 which plots the delivery percentage of the packets over a series of 30 seconds time windows for HDCP with $\beta = 1$, BCP, and CTP.

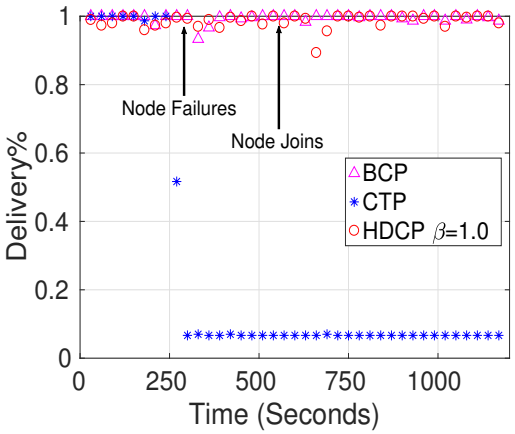


Figure 6.8: Thirty Second Windowed Average Sourced Packet Delivery Ratio for 10% Node Failures

Figure 6.8 demonstrates that CTP performance significantly suffers after the node failures and could not recover from that due to very high packet generation rate and the reliance on a single predetermined path from each source to the sink. On the other hand, the performance of both the HDCP protocol and the BCP protocol are unaffected by the node failure/join events. This pertains to the fact that both HDCP and BCP do not rely on a single path and, thus, able to take advantage of the alternate paths in the network, after the node failures. This validates that queue-aware routing algorithms such as BCP and HDCP perform well in presence of high node dynamics while the predetermined route based algorithms such as CTP suffer after node failures. The similarity in performance between BCP

and HDCP is again due to the similarity in the neighbor rankings in terms of the weights (explained in Section 6.5).

6.5 Similarity Analysis Between HDCP and BCP

In this section, we analyze the BCP and the HDCP algorithms to identify the reasons behind the similarity in their performance. The performance of both the BCP and the HDCP depend on the rankings of the neighbors (based on the link weighing functions) of a node, which in turn translates to the selection of routing paths to sink. From a theoretical standpoint, the performance of HDCP and BCP will be different in a network if their respective rankings of the neighbors under same queue size conditions are different. *Conversely, we hypothesize that the similar rankings of neighbors for both the HDCP and the BCP protocol will result in similar performance.*

6.5.1 Theoretical Analysis

In order to analyze the scenarios that will result in different or similar rankings of neighbors for HDCP and BCP, we compare the simplified weighing functions of BCP and HDCP with $\beta = 1$, which can be written as follows:

$$\begin{aligned} w_{ij}^{bcp}(n) &= q_{ij}(n) - 2.\overline{ETX_{ij}}(n) \\ w_{ij}^{hdcp}(n) &= \frac{q_{ij}(n) - \overline{ETX_{ij}}(n)}{\overline{ETX_{ij}}(n)} \end{aligned} \tag{6.14}$$

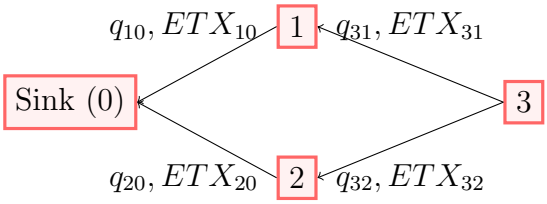


Figure 6.9: A Simple Topology For Ranking Similarity Analysis Between HDCP and BCP

provided that $V = 2$ and $\frac{q_{ij}(n)}{2\overline{ETX}_{ij}(n)} \geq 1$, i.e., the links have non zero weights according to both BCP and HDCP weighing schemes. First of all, we try to identify the range of the possible network configurations that will result in different rankings for HDCP and BCP. For this purpose, we analyze a toy topology illustrated in Fig. 6.9. Assume that the ETX_{31} and ETX_{32} are 1 and $e \geq 1$, respectively. Now, the weights of the respective links according to BCP will be:

$$w_{31}^{BCP} = q_{31} - 2 \qquad \text{and} \qquad w_{32}^{BCP} = q_{32} - 2e \tag{6.15}$$

Similarly, the weights for the links according to the HDCP rule for $\beta = 1$ will be (provided that $q_{31} \geq 2$ and $q_{32} \geq 2e$):

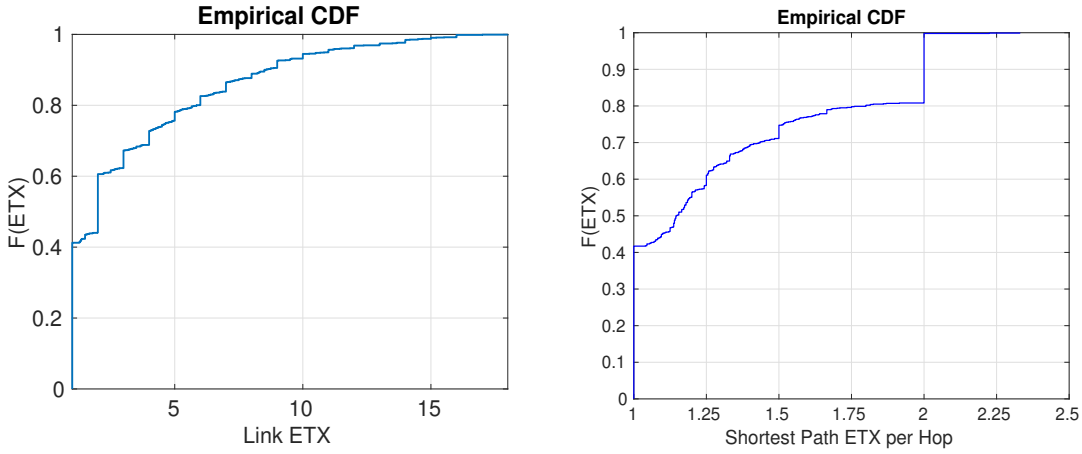
$$w_{31}^{HDCP} = q_{31} - 1 \qquad \text{and} \qquad w_{32}^{HDCP} = \frac{q_{32}}{e} - 1 \tag{6.16}$$

Now,

$$\begin{aligned} w_{31}^{BCP} > w_{32}^{BCP} \quad if \quad e > (q_{32} - q_{31})/2 + 1 \\ w_{31}^{HDCP} > w_{32}^{HDCP} \quad if \quad e > \frac{q_{32}}{q_{31}} \end{aligned} \tag{6.17}$$

Thus, if $\frac{q_{32}}{q_{31}} < e < (q_{32} - q_{31})/2 + 1$, the rankings of the outgoing links of node 3 are different, while the rankings are the same for all other values of e. As an example, say, $q_{31} = 4$ and $q_{32} = 6$, then only for $3/2 < e < 2$, the rankings are different. However, according to Eqn. (6.14) as well as Eqn. (6.17), for $ETX = 1$ both schemes will put similar weights on the links but with different negative offsets (2 for BCP and 1 for HDCP). Thus, the steady state performance will be same for both but with slightly lesser queue sizes in HDCP, which is also verified by our experiments. To verify whether the presence of too many perfect links is one of the reasons behind the similar performance of HDCP and BCP, we plot the CDF of the ETX traces collected from all the nodes during a real collection experiment, in Fig. 6.10a. In Fig. 6.10b, we plot the CDF of the average link costs (average ETX per link) of the shortest paths between every possible pair of nodes in the testbed. Figure 6.10a illustrates that a significant number ($\approx 40\%$) of links are perfect links ($ETX \approx 1$) while Fig. 6.10b implies that approximate 40% of the shortest paths consists of only perfect links ($ETX \approx 1$). Furthermore, approximate 60% of the shortest paths in the network between any possible node pair consists of links with average ETX of 1.25, as shown in Fig. 6.10b. All these statistics suggest similarity in the rankings of neighbors as well as the similarity in performance for the BCP and the HDCP algorithms. In summary, since we do not observe much of a difference in the performance of the HDCP and the BCP algorithms, we conjecture that in our

experimental setup, the probabilities for different rankings of the neighbors (more specifically, top 2 neighbors) are very low.



(a) (b)

Figure 6.10: (a) Empirical CDF of the Link ETX Values for Our Testbed (b) Empirical CDF of the Average ETX per Link for the Shortest Paths Between Any Pair of Nodes

Next, in order to verify whether similarity in neighbors' ranking will result in similar performance, we perform a theoretical analysis of the steady-state queue gradients for both HDCP and BCP. The steady-state queue sizes depend on the smallest cost path to the sink. Say, for a node i, there exists $k \in \{1, 2, \cdots, K\}$ possible paths and each path consists of one or more links l_k. Then the steady state queue size for node i in BCP will be $\overline{w}_i^{bcp} = \min_{k\in\{1,2,\cdots,K\}} \left[\sum_{j=1}^{l_k} 2 * ETX_{k,j}\right]$, while in HDCP the steady state queue size will be $\overline{w}_i^{hdcp} = \min_{k\in\{1,2,\cdots,K\}} \left[\sum_{j=1}^{l_k} ETX_{k,j}\right]$, where $ETX_{k,j}$ represents the ETX of the j^{th} link of the k^{th} path from node i to the sink. Thus, we can say that the steady-state path to sink for each node in HDCP is same as the BCP. Now, if the packet generation rate is low, every packet will

always follow the steady state gradient and, thereby, follow the same path leading to similar performance. Now, if the packet generation rate is high, in the worst case we will have a batch arrival of packets at some node, say i. Let us assume that when node i disseminate the batch arrival packets, all the neighboring nodes of i are unchanged, i.e., no packet arrival (except the node i) or departure takes place. In this situation, node i will keep on transmitting to the neighbor that is part of the best path to sink, until a point when the weight for the respective link becomes worse than the 2nd best link. In the following, we analyze at what point, i.e., after how many packet transmissions, node i will switch to the second best link.

- **BCP:** Node i prefers a neighbor node m over another neighbor node n, iff:

$$q_i - q_m - 2 \times etx_{im} > q_i - q_n - 2 \times etx_{in} \tag{6.18}$$

where q_i, q_m, q_n represent the queue sizes at node i, m, and n, respectively and etx_{im}, etx_{in} represents the etx of the links im and in, respectively. Now, say after x number of transmissions, the 2nd link is considered:

$$\begin{aligned} &\implies (q_i - x) - (q_m + x) - 2 \times etx_{im} = (q_i - x) - q_n - 2 \times etx_{in} \\ &\implies x = q_n - q_m + 2 \times (etx_{in} - etx_{im}) \end{aligned} \tag{6.19}$$

Now, WLOG assume that the node m is part of the best path to the sink from node i, while node n is part of the second best path. Then, $etx_{im} \approx etx_{in} \implies$

$x \approx q_n - q_m$. If $etx_{im} < etx_{in}$, $x = q_n - q_m + 2 \times (etx_{in} - etx_{im}) > q_n - q_m$. On the other hand if $etx_{im} > etx_{in}$ implies $(q_m < q_n)$ for feasibility of the rankings in focus, which implies $x = q_n - q_m - 2 \times (etx_{im} - etx_{in}) \leq (q_n - q_m)$.

- **HDCP:** Node i prefers a neighbor node m over another neighbor node n, iff:

$$\frac{q_i - q_m}{etx_{im}} > \frac{q_i - q_n}{etx_{in}} \tag{6.20}$$

where q_i, q_m, q_n represent the queue sizes at node i, m, and n, respectively, and etx_{im}, etx_{in} represent the etx of the links im and in, respectively. Now, say after x number of transmissions, the 2nd link is considered:

$$\begin{aligned}
&\implies \frac{(q_i - x) - (q_m + x)}{etx_{im}} = \frac{(q_i - x) - q_n}{etx_{in}} \\
&\implies x(2 - \frac{etx_{im}}{etx_{in}}) = q_i \times (1 - \frac{etx_{im}}{etx_{in}}) - q_m + q_n \times \frac{etx_{im}}{etx_{in}} \\
&\implies x = q_i \times \frac{(1 - \frac{etx_{im}}{etx_{in}})}{(2 - \frac{etx_{im}}{etx_{in}})} - q_m \times \frac{(1 - \frac{etx_{im}}{etx_{in}})}{(2 - \frac{etx_{im}}{etx_{in}})} - q_m \times \frac{(\frac{etx_{im}}{etx_{in}})}{(2 - \frac{etx_{im}}{etx_{in}})} \\
&\quad + q_n \times \frac{\frac{etx_{im}}{etx_{in}}}{(2 - \frac{etx_{im}}{etx_{in}})} \\
&\implies x = \frac{1}{2} \times (1 - \mathbf{z}) \times (q_i - q_m) + \mathbf{z} \times (q_n - q_m) \text{ where } \mathbf{z} = \frac{\frac{etx_{im}}{etx_{in}}}{(2 - \frac{etx_{im}}{etx_{in}})}
\end{aligned} \tag{6.21}$$

Now, WLOG assume that the node m is part of the best path to the sink while node n is the second best path. Similar to BCP, $etx_{im} \approx etx_{in} \implies x \approx q_n - q_m$. It also suggest that in HDCP, the number of transmissions before

switching depends on a weighted sum of the queue differential of the best link $(q_i - q_m)$ and the queue differential of the 2nd best and the best neighbor $(q_n - q_m)$, where the weights depend on the ratio $(\frac{etx_{im}}{etx_{in}})$. If $etx_{im} < etx_{in}$, $0.5 \leq \mathbf{z} \leq 1$ that implies more weight on $(q_n - q_m)$ thereby increasing the chances of switching as $q_i \geq \max\{q_m, q_n\}$ which also implies $x \geq \frac{1+\mathbf{z}}{2}(q_n - q_m) \geq (q_n - q_m)$. On the other hand, $etx_{im} > etx_{in} \implies (q_m < q_n)$ for feasibility of the rankings in focus and $\mathbf{z} > 1$ that suggests $x = \mathbf{z} \times (q_n - q_m) - \frac{\mathbf{z}-1}{2} \times (q_i - q_m) \leq \frac{1+\mathbf{z}}{2}(q_n - q_m)$.

The above analysis suggests that if the outgoing best and 2nd best link of a node have similar ETX, both BCP and HDCP will switch after exactly same number of transmissions under same queue conditions. Even in other cases for any particular network, the switching patterns are similar and just switches after a slightly different number of transmissions, which is a function of $(q_n - q_m)$. Therefore, the observed performance of BCP and HDCP will be similar. However, this analysis is pertinent to the fact that both HDCP and BCP have same rankings of the neighbors (at least best two neighbors) which validates our hypothesis.

Based on the theoretical analysis, we conjecture that the similarity of performance between HDCP and BCP in our testbed experiments is due to the similarity of rankings of the neighbors in most of the nodes. To verify this conjecture, we perform a Kendall Tau test of the ranking data collected from our real experiment setup, as follows.

6.5.2 Kendall's Tau Test

In the previous sections, we observed that the optimized versions of BCP and HDCP with $\beta = 1$ are very similar to each other in performance. We hypothesized that this may be due to the similarity in the neighbor rankings for the two protocols. In order to verify our hypothesis, we collected a set of routing table snapshots from three representative nodes, located at one hop, two and three hop distance from the sink, respectively, during a real collection experiment. These snapshots contain the information about their neighbors such as backpressure and ETX information from the real experiment. Based on those snapshot values, we calculated the Kendall's Tau distance between the neighbor rankings generated by the weight calculation in BCP on one hand, and the neighbor rankings generated by the weight calculation in HDCP for different values of β on the other, for all neighbors that have a positive weight in at least one of the two protocols under comparison. Kendall's Tau distance between two rankings indicates the fraction of pairs that are ordered the same in the two rankings. If it is 0, then the two rankings are identical. Higher values indicate more different rankings.

We present the results in Fig. 6.11. It clearly shows that while there is a lack of correlation between lower values of β, for $\beta \rightarrow 1$ there is a strong correlation between HDCP and BCP. This verifies our hypothesis and justifies the results shown in this study.

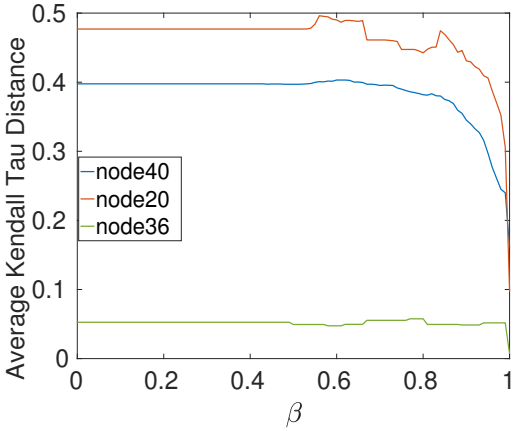


Figure 6.11: Variation of Kendall's Tau Distance between HDCP and BCP Neighbor Rankings for Different Values of β

Another noticeable fact is that for $\beta \in [0, 0.6]$ the Kendall Tau distance with respect to BCP remains almost the same. We performed additional Kendall's Tau correlation analysis between neighbor rankings of HDCP for every possible pair of $\beta \in \{0, 0.2, 0.4, 0.6\}$ and the average distance for each case was found to be less than 0.1. This is the reason behind the similarity in performance of HDCP with $\beta \in \{0, 0.2, 0.4, 0.6\}$.

6.6 Discussion

We have proposed and implemented a new data collection and routing protocol for robotic wireless networks and wireless sensor networks called HDCP that is the first practical realization of a theoretical algorithm called the Heat Diffusion algorithm which is inspired by Thermodynamics. We have evaluated HDCP on a real wireless network testbed. We have compared the performance of HDCP with

two well-known protocols on this testbed: CTP and BCP. Based on the results, we can conclude that HDCP with an optimized parameter setting of $\beta = 1$ performs as well as BCP and outperforms CTP with respect to throughput performance, interference resilience, and low power operation, while all three generally offer about the same end-to-end delay on average in the full throughput region.

The equivalent performance of HDCP to the previously published BCP is a somewhat surprising finding of this study. From a mathematical perspective, this is not obvious as they employ quite different equations for the weight calculations and indeed in our prior theoretical works, Heat Diffusion has been found to perform better than Backpressure scheduling in some respects. But as we have shown, nevertheless, the two protocol implementations provide very similar neighbor rankings in a real network. We believe our finding also lends some support to the notion that it may not be possible to get any higher performance in practice with a dynamic routing protocol that takes into account both queue states and link quality.

In summary, HDCP is a well-designed routing protocol with a smaller end-to-end delay that does not rely on precomputed routing paths yet provide better or comparable performance to the existing alternatives. The delay and network dynamics adaptability of the protocol makes it suitable for an RWN where the control packets need to be fast and efficiently routed between the participating devices to achieve the deployment objectives.

Chapter 7

Unified Communication Protocol for Control and Sensing in RWN

In this chapter, we present our fourth study on a low-power application layer communication protocol with low bandwidth consumption for an RWN that abstracts the notion of control and data transfer under a unified overlay networking.[1] The typical application contexts of an RWN imposes a heavy demand on the underlying communication between the robotic agents in terms of efficiency, reliability, and scalability. The robots inherently work under heavy power constraints which in turn imposes energy consumption constraint on the communications hardware and protocols as well. Another major factor related to the scalability of the network is the limited communication bandwidth which leaves the supported number of simultaneous wireless communications up to the design of the communication protocols and packet formats. Thus, the communication protocols in a dense network

[1]The material in this chapter is based in part on the work in [162].

of wireless robots must be lightweight with low bandwidth requirements. However, we have noticed a lack of such application layer communication protocols engineered specifically for efficient control and data collection in a network of heterogeneous robots.

In this study, we present the Robotic Overlay coMmunicAtioN prOtocol (ROMANO), a lightweight, application layer overlay communication protocol for a unified sensing and control abstraction of a network of heterogeneous robots mainly consisting of low power, low-compute-capable robots. ROMANO is built to work in conjunction with the well-known Message Queuing Telemetry Transport for Sensor Nodes (MQTT-SN) protocol, a lightweight publish-subscribe communication protocol for the Internet of Things and makes use its concept of "topics" to designate the addition and deletion of communication endpoints by changing the subscriptions of topics at each device. We also develop a portable implementation of ROMANO for low power IEEE 802.15.4 radios and deployed it on a small testbed of commercially available, low-power, and low-compute-capable robots called Pololu 3pi robots. Based on a thorough analysis of the protocol on the real testbed, as a measure of throughput, we demonstrate that ROMANO can guarantee more than a 99.5% message delivery ratio for a message generation rate up to 200 messages per second. The single hop delays in ROMANO are as low as 20ms with linear dependency on the number of robots connected. Lastly, as proof of concepts, we

implement four different applications of ROMANO in a network of robots: (1) simplified control of the robots, (2) seamless sharing of sensor data, (3) control of any functions of a robot such as peer-to-peer radio transmission, and (4) communication and control between multiple networks of robots over internet.

7.1 The Proposed ROMANO Protocol

We have discussed the basic concepts of a MQTT-SN communication in Section 2.4. In this section, we discuss the details of ROMANO protocol as well as how ROMANO build on top of MQTT-SN.

From a pool of different types of MQTT-SN messages, the ROMANO utilizes mainly the MQTT-SN Publish Messages. The typical format of an MQTT-SN Publish message is presented in Table 2.1. ROMANO utilizes the variable length "Data" field of a MQTT-SN Publish Message for sending ROMANO messages between nodes and the TopicId field (containing the topic id value or short topic name) for identifying communication endpoints.

7.1.1 Requirements:

At a minimum, ROMANO requires each end device/robot to run a MQTT-SN client on a multi-threaded OS. Each of the devices needs to be connected to a MQTT-SN broker/forwarder while the broker nodes are bridged together either directly or over the internet. Furthermore, the broker device also runs a ROMANO

server program. Each end device (robot or sensor node) needs to follow a standard connection *establishment/initialization phase* to initiate ROMANO as follows.

- Set up a MQTT-SN connection to a MQTT-SN broker where the device's IPV6 address is used as the device identifier.
- Subscribe to a topic named after the last 8 characters of the device's IPv6 address which we refer to as the ROMANO ID. For example, a device with address $fe80 :: 212 : 4b00 : abcd : 1234$ subscribes to the topic "$abcd1234$".[2] The ROMANO ID can be used to communicate to a specific device.
- Publish the ROMANO ID on a predefined topic "init-info" and wait for a fixed amount of time (2 seconds in our implementation) for an acknowledgment to be published by the ROMANO server on the respective ROMANO ID topic. If no acknowledgment is received on time, the node retries indefinitely.
- Subscribe to the topic "common" which ROMANO uses for broadcast communication.

7.1.2 The ROMANO Protocol

Our proposed ROMANO protocol can be described as follows.

- According to the five-layered Internet model of networks, the ROMANO protocol falls under the application layer alongside the MQTT-SN protocol. More

[2]One can use the whole IPv6 address as the topic. However, ROMANO uses the last 8 characters to keep ROMANO ID small while accommodating up to 2^{32} devices.

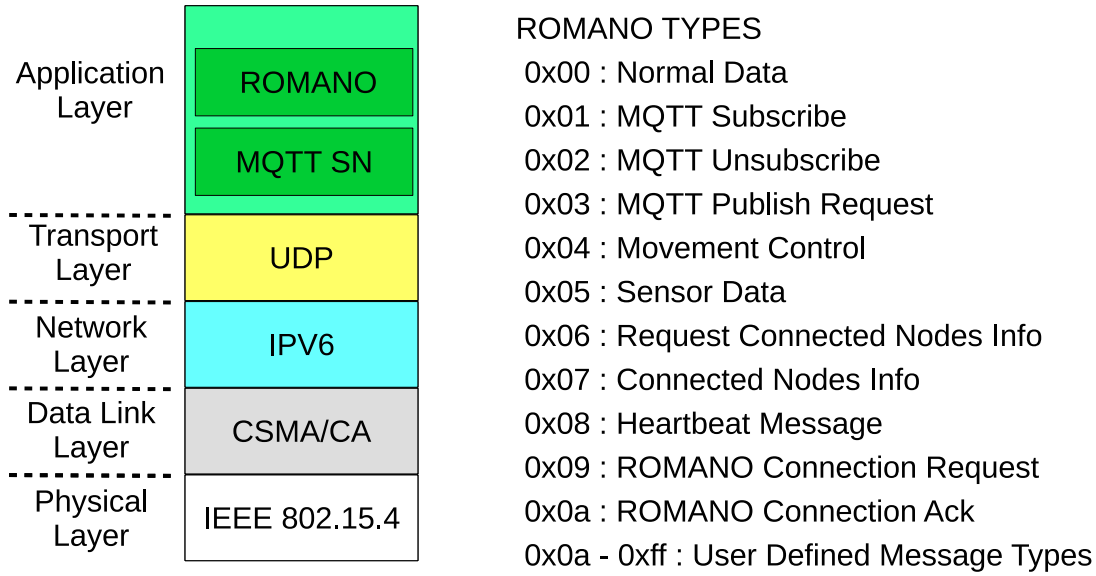


Figure 7.1: (Left) The ROMANO Network Stack, (Right) ROMANO Data Types

specifically, the ROMANO and the MQTT-SN protocols form a nested, layered structure inside the application layer as presented in Fig. 7.1.

- ROMANO uses the MQTT-SN topics to define the communication endpoints. Any publisher to a certain topic is the transmitter node while all the subscribers of that topic are the receivers. Any node of a ROMANO network can be a transmitter at any instance of time for any topic while the receiver nodes need to subscribe first with the broker and remain connected. Thus, ROMANO allows all types of communication: *one-to-one (unicast), one-to-many (multicast), one-to-all (broadcast), many-to-one, and many-to-many.*

- ROMANO uses the MQTT-SN data section for the overlay communication where a complete ROMANO message is embedded in the MQTT-SN data section.

- ROMANO has the feature of controlling the subscriptions of a node. One of the ROMANO message types, MQTT Subscribe can instruct the receivers to subscribe to a particular topic (say, 'test-topic') so that they can register themselves as receivers of that topic ('test-topic').

- ROMANO also has the feature of instructing the receivers to publish certain types of data (e.g. telemetry data) to certain topics (e.g. 'telemetry'). This feature can be used for active polling of sensor/control data for a specific robot such as the leader.

- The ROMANO overlay protocol allows any node (e.g. robot or sensor) in the network to control the movements of a robot via the same abstraction regardless of whether they are either connected directly, connected via an ad hoc network, or connected over the internet.

- ROMANO has an optional periodic 'heartbeat' messaging feature to notify its presence to all the connected nodes, which can be used for neighbor discovery or end-to-end reliable message transfer.

7.1.3 Message Formats:

The communication in ROMANO follows certain message structures as described in this section. The base messaging format in ROMANO is presented in Table 7.1. The ROMANO data type field dictates the communication type. The default types are presented in Fig. 7.1 (Right).

Table 7.1: ROMANO Message Format

ROMANO Data Type (1 octet)	ROMANO MSG Length (1 octet)	ROMANO Data (1-253 octets)

The ROMANO MSG Length section denotes the length *(say, k + 1 octets)* of the whole ROMANO message, and the ROMANO Data Section contains the data of variable length. *Each of the ROMANO message types (except type ROMANO Connection Ack which only uses the Data Type and the Data length section) have their own formats, summarized in Table 7.2.* Most of the message formats are self-explanatory except the ROMANO MQTT Publish Request message and Movement Control message. In MQTT Publish Request messages, the MQTT Topic Length field *(say, m octets)* marks the end of the Topic ID field (to publish data to) from the beginning of the ROMANO message where the Topic ID field starts at the 3rd octet. The 'data to publish' section in the Publish Request message defines the custom type of data to publish. The ROMANO movement control message can be used to control different types of movements. We have also defined some basic movement control message type listed in Table 7.3. One can define up to 2^{16} different movement control functions (using the allocated 2 octets) with custom arguments.

In principle, entire sequences of useful movements could be encoded into a single movement control message; for example, a semi-circular motion clockwise around an obstacle, specified by a radius parameter. ROMANO outputs each movement command to the built-in ROMANO control data mailbox queue (a structure made available for both C and C++) which is serviced by the thread running a robot's movement controller. The implementation of a movement controller is still independent of ROMANO, but controllers are required to retrieve commands from the ROMANO control data mailbox.

7.1.4 ROMANO for Bootstrapping of Robots:

One key feature of ROMANO is that it is designed with the aim of easier integration of robots in a network as well as bootstrapping new devices with coherent configurations. In a network of robots, due to many reasons such as the network dynamics and failures, robots/devices might come and go. In such cases, whenever a new device joins the network, it needs to be configured for consistency. In this regard, ROMANO allows a plug and play type system where the robots joining the network just need to run a ROMANO client with proper access to its resources such as movement controller logic and state information. Now any robot or a human connected to the network can use ROMANO to dynamically configure the robot based on the network and application demands on the fly. This reduces the amount of manual configuration required in a robot before its deployment. This feature has

Table 7.2: ROMANO Message Formats

Message Format for Request Connected Nodes Info, Heartbeat Message, and Connection Request

ROMANO Data Type (octet 0)	MSG Length (1)	ROMANO ID (2 - 9)

Normal Data / Connected Nodes Info message Format

ROMANO Data Type (octet 0)	MSG Length (1)	Data (2 - k)

MQTT SUB/UNSUB Control Message Format

ROMANO Data Type (octet 0)	MSG Length (1)	Topic to subscribe to or unsubscribe from (2 - k)

MQTT PUB Request Message Format

ROMANO Data Type (octet 0)	MSG Length (1)	MQTT Topic Length, (m) (2)	Topic ID (3 - m)	Data to Publish (m+1 - k)

Movement Control Message Format

ROMANO Data Type (octet 0)	MSG Length (1)	Movement Control Type (2 - 3)	Movement Control Data (4 - k)

Sensor Data Message Format

ROMANO Data Type (octet 0)	MSG Length (1)	Sensor Type (2 - 3)	Sensor Data (4 - k)

Table 7.3: Movement Control Types

Movement Control Type		Movement Control Data
Type	Value	
Move Front	0x0000	Distance
Move Back	0x0001	Distance
Move Left	0x0002	Distance
Move Right	0x0003	Distance
Rotate Left	0x0004	Angle
Rotate Right	0x0005	Angle

been incorporated into our in-house wireless robotic network testbed called the IRIS testbed for quick and easy bootstrapping of the entire testbed without the need to manually reconfigure every time.

7.2 Real Implementation and Experimentation:

7.2.1 Core Implementation:

In this section, we present our real implementation of the proposed ROMANO protocol in a testbed of five cheap, low power, and commercially available robots called Pololu 3pi [148]. A 3pi, illustrated in Fig. 7.2, also comes with an expansion board that can accommodate an XBee form factor device for IEEE 802.15.4 communication and a mbed board. For communication, we use a commercially available product for IoT called the OpenMote [139], and for the mbed device, we use the LPC1768 model of mbed [140]. We choose this particular set of hardware due to the following reasons. (1) These devices are compatible with each other and have very low computation power, very small communication energy consumption, small form factor, and also comparably low cost. (2) These devices form the base of our in-house scalable, portable, cheap, open-source wireless robotic IoT testbed called the IRIS testbed that is used for research on low power robotics with a major focus on communication and networking.

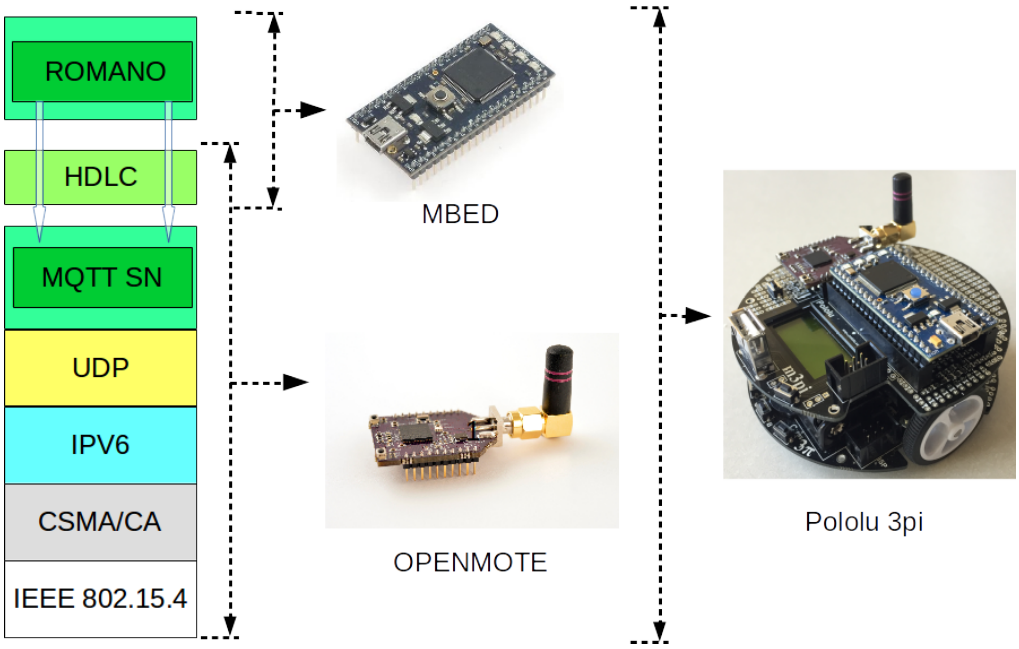


Figure 7.2: ROMANO Implementation Stack on Pololu 3pi

In this testbed, we implemented the ROMANO in a modular distributed manner over a mbed and an OpenMote. The mbed device does not have a radio, but it has more processing power and memory than the OpenMote. The OpenMote comes with a radio but does not have enough General Purpose Input Output (GPIO) pins and memory to act as the robot controller. Thus, we implemented ROMANO across both devices with controllers on the mbed and the communication software stack on the OpenMote with UART bridging data between the two. For reliable UART communication, we have implemented a low power version of a reliable data transfer protocol called the High-level Data Link Control (HDLC) [143]. Our stack implementation is illustrated in Fig. 7.2. For software development, we use a well-known open source OS for IoT called RIOT-OS on the OpenMote and the open-source real-time operating system MBED-OS 5 on the mbed. Note that, for implementing

the ROMANO protocol, one device with both a radio and microcontroller is also sufficient.

In our current implementation, the MQTT-SN broker is running on a Raspberry Pi running Raspbian with an OpenMote connected via USB to act as the 802.15.4/6LoWPAN gateway device. All the 3pis are connected to the broker as well as the internet via either a direct link or a multihop link (illustrated in Fig. 7.3). For routing in the multihop network we use a well-known routing protocol for 802.15.4 networks called RPL [109]. We use IPv6 for addressing instead of IPv4 due to its wide applicability in IoT systems as well as its compatibility with existing IPv4 systems. The IEEE 802.15.4 standard operates on the same spectrum as WiFi but with different communication channels and modulations. In the USA, the IEEE 802.15.4 standard offers 16 different channels numbered from 11-26. We performed the experiments using channel 26 of the IEEE 802.15.4 standard to avoid external interference from WiFi networks.

7.2.2 Performance Analysis

With the experimental setup detailed in Section 7.2.1, we performed a set of experiments to analyze the performance of the proposed ROMANO protocol. In this section, we focus on three important communication/networking aspects in a robotic network: scalability, end-to-end delay of communication, and throughput.

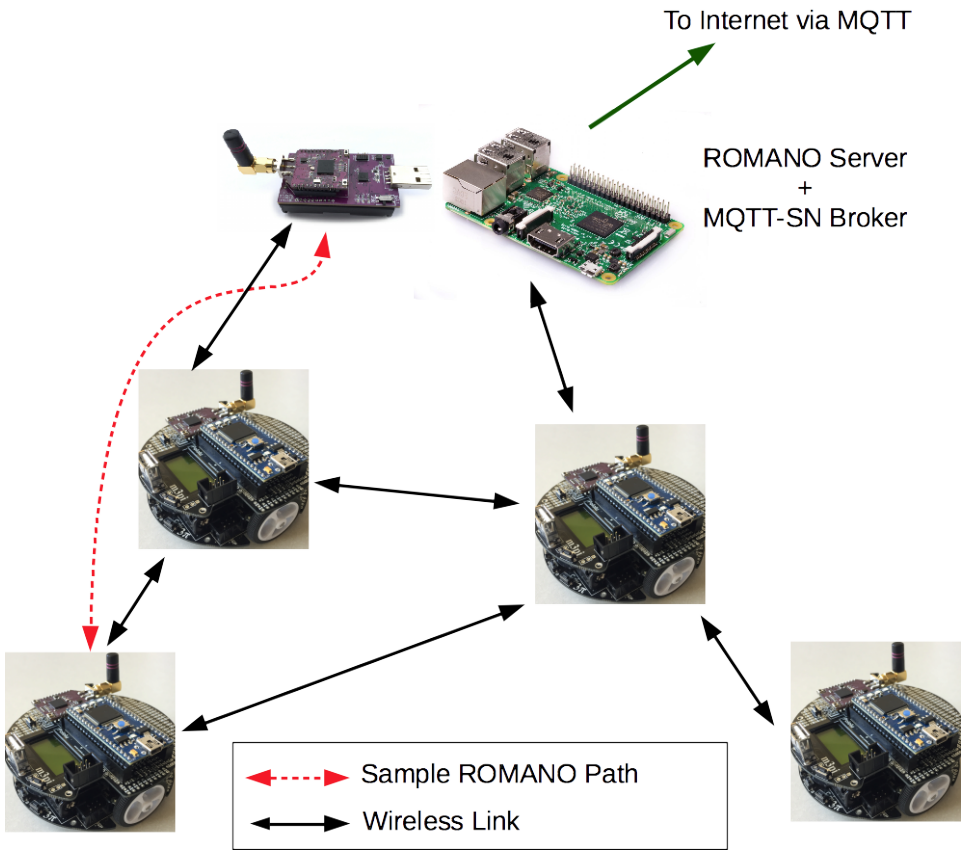


Figure 7.3: Our Testbed Architecture for the ROMANO Experimentation

We performed a series of stress tests with the experimental setup to find the performance boundaries of ROMANO. To test message delivery ratio, we ran a set of experiments where the ROMANO server script publishes messages to the connected robots via ROMANO for message generation rates of 1, 10, 20, 50, 75, 100, 200, 300, 400, and 500 Message(s) Per Second (MPS). For each rate, we ran 10 experiments, each with 5000 messages. We find that the message delivery percentage is $\geq 99.5\%$ for a messaging rate of *200 MPS* or less. For higher message generation rates the testbed system fails after a while due to the radio buffer overflow (summarized in Table 7.4). After careful investigation, we find that this buffer overflow is due to

the radio hardware limitations and not due to the limitations of the ROMANO protocol. This is further justified by the fact that, until the radio buffer overflows, the message delivery ratio is $\approx 99.5\%$.

Table 7.4: Message Delivery Ratio for Different Message Generation Rates

MPS	Message Delivery Ratio		Comments
	Minimum	Maximum	
100	99.9%	100%	No Radio Buffer Failure
200	99.5%	100%	No Radio Buffer Failure
300	44%	100%	Radio Buffer Overflow after roughly 2200 message
400	15%	100%	Radio Buffer Overflow after roughly 1300 Messages
500	13%	96%	Radio Buffer Overflow after roughly 600 Messages

To study the scalability as well as the delay, we performed a set of experiments where the server published message at a rate of 20 MPS to the topic "common" while we varied the number of robots. The results, presented in Fig. 7.4, demonstrates that typically the minimum delay is $\approx$ 20ms which is justifiable as typical packet transfer time in an IEEE 802.15.4 network is $\approx 10 - 20$ms. Figure 7.4 also demonstrates that the individual delay experienced by the robots are different. This difference in delay is deterministic ($\approx$ 8 ms) and due to the operation principle of MQTT-SN (not ROMANO) where the broker dispatches broadcast messages via sequential unicast messages to one subscriber node at a time. This suggests that there is a linear relationship between the maximum delay over ROMANO with the number of subscribed robots on that topic.

In terms of reliability, ROMANO can provide a device to broker reliability by using reliability feature of the MQTT-SN with different QoS modes. In the current version, ROMANO does not have any end-to-end reliability. But one can easily add some level of reliability by using the feature of heartbeat message where each device periodically sends a 'heartbeat' message to the 'common' topic to inform all nodes about its presence and adding a logic to send messages to a device only if a heartbeat message was received. If a heartbeat message has not been received, the sender can queue it until the destination device rejoins the network.

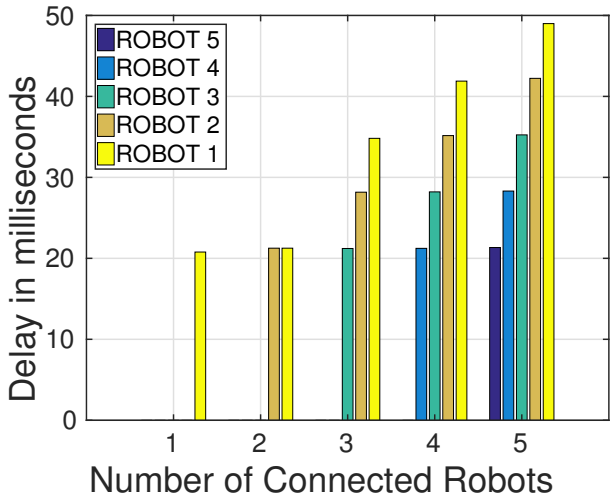


Figure 7.4: Scalability Analysis of the ROMANO Protocol

Note that in this study we do not compare this performance with other protocols as we were unable to find another overlay protocol for robotic networks with a focus on lightweight, low power communications. Of course, ROS architecture gives similar features as our proposed ROMANO protocol. Nonetheless, to our knowledge, there exists no generic lightweight low power version of ROS. In this

study, we do not focus on higher power robots with fully functional computers running Linux with WiFi.

7.2.3 Application Implementation Experiments

To test and analyze how our protocol works, we have implemented and tested four different applications detailed as follows. The videos from the experiments are available at `https://anrg.usc.edu/www/research/robotic-networks/romano/`.

7.2.3.1 ROMANO for control of a group of robots

In this implementation, we mainly use the ROMANO Movement control message format. We have implemented a movement control thread in each robot, which wakes up upon receiving a movement control message via the ROMANO protocol and executes the movement instructions (e.g. move left or right by a fixed amount). The movement control messages can be published to any topics. Therefore, to control a subgroup of robots in a swarm, nodes can either publish to all target ROMANO IDs or publish to a special topic subscribed by only the target subgroup. To form a group, we can simply use the MQTT Sub/Unsub Control Message to individually instruct the member robots to subscribe/unsubscribe to the respective group topic.

7.2.3.2 ROMANO Path Copy

In this application, we implement a very simple system where one 3pi robot runs line-follower code that uses the five onboard reflective light sensors to follow a black trail on the ground while it shares its telemetry information via ROMANO to a certain topic. All the other robots listen for the telemetry and use the telemetry to replicate the path. This illustrates how easily sensor data or any kind of data can be shared between a group of low power low capability robots, and this also shows how ROMANO can be used to control one robot from another.

7.2.3.3 ROMANO to Control Peer-to-Peer UDP Communication

In this implementation, two robots use ROMANO to control the UDP packet transmissions of one another and disperse while both robots maintain a certain radio communication link quality level. We add two different custom packet types for this purpose: "UDP-SEND-REQ" (0x11) and "UDP-SEND-GO" (0x12). To illustrate the application, we describe a sample sequence of events between a robot A and robot B (presented in Fig. 7.5) as follows. First, robot A publishes a UDP-SEND-REQ message to robot B and receives a UDP-SEND-GO reply. Upon receiving UDP-SEND-GO, Robot A transmits a broadcast UDP packet. Upon receiving the UDP packet, Robot B record the Radio Signal Strength Information (RSSI) of that packet. If the RSSI is above a user defined threshold $RSSI_{th}$, the controller of Robot B moves it away from A by a fixed step size d_s. Similarly, if the value is less

than $RSSI_{th}$, Robot B will move closer to Robot A to improve the link quality. After the movement step, robot A becomes ready to reply to a UDP-SEND-REQ message from robot B (B is sending UDP-SEND-REQ messages at a regular interval until A replies). This process continues indefinitely, and B will follow the same procedure as A. The movement is restricted to forward and backward movement along a black line to leverage the 3pi's reflective sensors for simplicity. The whole process is randomly initiated by one of the robots given the two are within range of their radios.

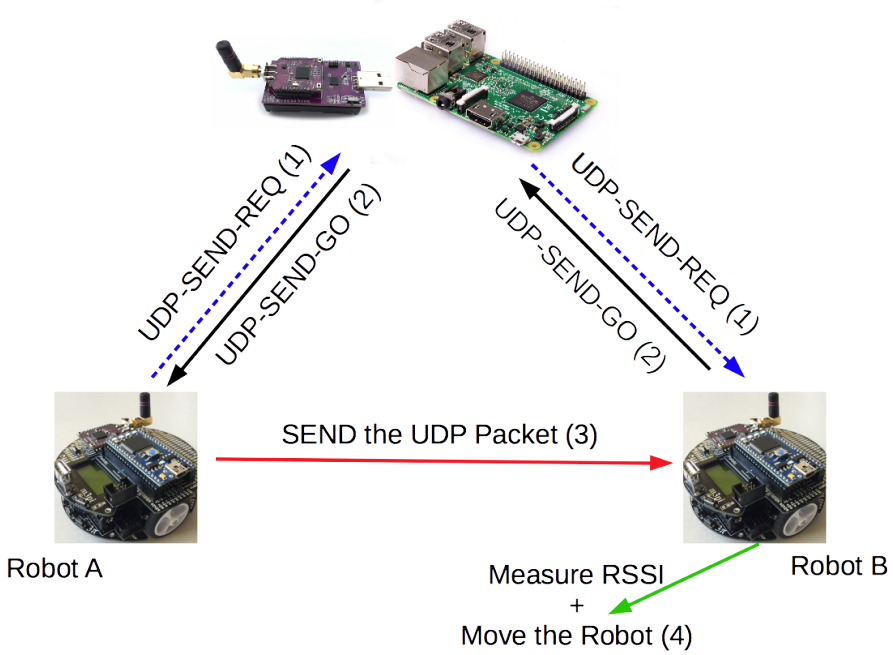


Figure 7.5: Illustration of using ROMANO to control peer-to-peer UDP communication

7.2.3.4 ROMANO over Internet

Until now, we discussed employing ROMANO over a single 802.15.4 network. In this application, we show how ROMANO can be used between two networks of robots bridged via the internet. The 802.15.4 networks use IPv6 address and MQTT-SN protocol for communication while over the internet communication use IPv4 address and MQTT protocol, illustrated in Fig. 7.6. It shows that ROMANO is compatible with both IPv6 and IPv4 as well as both MQTT and MQTT-SN. In this set of experiments, we continually send movement control commands to a robot which in turn relays that information to another robot in a different network bridged via the internet.

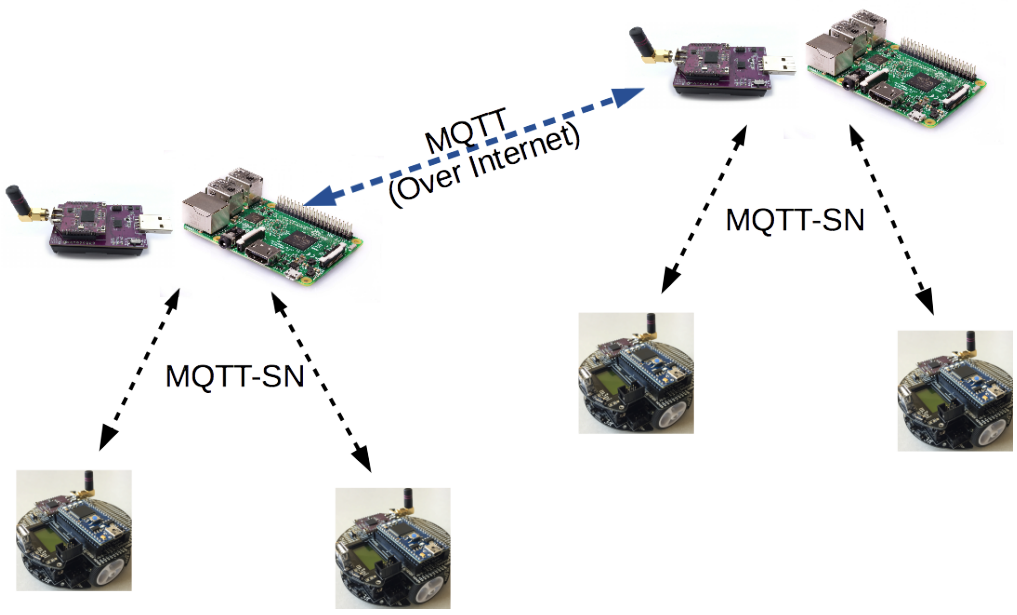


Figure 7.6: Application of ROMANO for communication between two robotic networks connected over internet.

7.2.4 Outcomes of the Application Specific Experiments

Now, we briefly discuss some of our interesting findings from the application specific implementations and experiments. In the first set of experiments i.e., control of a group of robots via ROMANO, the system performed as expected with 100% delivery of control messages. We were able to seamlessly switch between controlling different groups of robots as well as individual robots. For the second set of experiments, i.e., sensor data sharing between robots, we observed that while the messages the delivered in order with around 99.5% message delivery ratio, the delays between consecutive messages are not uniform. This has to do with the wireless channel access time, variable delay in the software stack, and the queueing of messages at the MQTT broker. Therefore, in the streaming sensor data type application, one needs to be careful about the control logic implementation. To illustrate, if each of the shared sensor data has a major impact on the operation of the robots, extra precaution needs to be taken for late or missing messages. The results of the third experiment i.e., controlling UDP transmission with ROMANO, concur with our expected outcomes. It is expected that the robots will disperse until they reach each other's communication boundary and will keep oscillating to-and-fro in the boundary region. The videos available at `https://anrg.usc.edu/www/research/robotic-networks/romano/` clearly shows that the experiment outcomes align with the expected outcomes. The outcomes of the last and final experiment i.e., ROMANO over the internet, also

follow our expected outcomes. One interesting observation, in this case, is that both robots follows same control instructions but with a substantial time offset. This delay is mainly caused by the propagation time over the internet as well as the relaying delay.

7.2.5 Code Complexity Analysis

In this section, we compare the code complexity of the ROMANO based implementations. One key feature of the ROMANO protocol is that it simplifies the job of an application developer for an RWN. Due to complexity and inter-dependencies, an RWN application developer, ideally, needs to have knowledge about a range of diverse topics such as communication stack, wireless communication, and control. This hinders the progress of the RWN domain as typical application developers lack deep knowledge about embedded communication stacks and networking. In that context, ROMANO offers a simple abstraction that does not require deep understanding of the underlying communication stack. Moreover, it reduces the amount of effort required from application developers to implement a new functionality. We characterize the effort required from a developer in terms of the line of codes (LOC) to be written for each application. To this end, we compared the LOC for implementing the functionalities required for the above mentioned four applications with and without ROMANO in our in-house RWN testbed. To calculate the LOC without ROMANO, we count both the application specific number of lines that is

implemented on top of ROMANO, say x, and the lines of codes it uses (through function calls etc.) from ROMANO, say y. Then the LOC for writing that application with ROMANO is x and the LOC for writing it without ROMANO is $x + y$. The results are presented in Table 7.5.

Table 7.5: Code Complexity Analysis in Terms of Lines of Codes

Functionality	With ROMANO	Without ROMANO
Subscribe/Publish	7	889
Movement Control	107	962
Path Copy	209	1027
UDP Control	257	1295

In the first application, discussed in Section 7.2.3.1, we mainly require two functionalities: (1) dynamic update of the MQTT-SN subscriptions on the RWN nodes without any need of manual configuration and (2) movement control of the robots over MQTT-SN. To implement the first functionality, ROMANO reduces the LOC by more than 100 fold, as shown in Table 7.5. The LOC for MQTT-SN based movement control is also reduced by approximately 800 lines by the usage of ROMANO. Moreover, in our testbed implementation, ROMANO simplifies the coding to only one embedded platform i.e., MBED, instead of the originally required multi-platform code. The second application, i.e., ROMANO based path copy (discussed in Section 7.2.3.2), is also implemented with approx. 800 less LOC than implementation without ROMANO. The third application, i.e., ROMANO

based UDP transmission control application (discussed in Section 7.2.3.3), requires approx. 1000 less LOC due to usage of ROMANO. This is the most complex implementation among the four application. Lastly, the ROMANO over internet, discussed in Section 7.2.3.4, requires only the Movement Control functionality which is already discussed in the context of the first application. Also note that the LOC with ROMANO for all cases except the Subscribe/Publish mainly includes the application specific robotic control code. In all cases, the application specific LOC does not include any 802.15.4 standard protocol stack coding. In summary, ROMANO significantly reduces the LOC required to implement complex applications in an RWN as well as simplifies the process.

7.3 Discussion

In this study, we presented a novel overlay protocol called ROMANO that works on top of MQTT-SN to provide a light-weight, scalable and low power communication abstraction for sensing and control in a wireless network of robots. We also developed a real system on a robotic testbed consisting of five Pololu 3pi robots and performed a set of evaluations for the proposed ROMANO protocol. Through a set of four different application implementations, we demonstrate how the ROMANO protocol can be used in different application contexts of a robotic wireless network. Morever, we imagine an RWN to include a group of comparably high power robots such as the Trackbot from Study 1, which will act as a distributed set of ROMANO

brokers and MQTT Servers. In such an RWN, ROMANO can be easily used for all sorts of communication towards achieving certain end goals.

Chapter 8

Passive RF Sensing in RWN

Over last decade, Radio Frequency (RF) signal based sensing has become very popular among researchers due to its ubiquitous nature and a wide range of applicability.[1] One obvious application context is the mapping of unknown areas with a network of robots [63, 8]. In this study, we present a proof-of-concept low-power IEEE 802.15.4 standard based bistatic radar [62] system for localizing unknown radio-wave reflecting objects in an unknown environment. Unlike prior multi-antenna based approaches, we employ a single standard low power omnidirectional transmitter with known transmission parameters and a single rotating directional receiver antenna to collect a set of directional RSSI samples and, thereafter, exploit the directionality information of the samples to determine the locations of the reflecting objects. To this end, we employ the well-known Maximum Likelihood Estimator (MLE) to extract the required information from the collected RSSI samples. Note that a bistatic radar [62] is a well-known RF system that is composed of

[1]The material in this chapter is based in part on the work in [163].

a transmitter and a receiver separated by a large distance compared to the distance to an object. A bistatic-radar employs the difference in the path lengths traveled by a direct path signal and a multipath signal reflected by an object to passively localize the object.

One motivation for our investigation into building this system was our interesting findings in the course of our first study. In the ARREST experimentation during our first study, we observed that the received angular RSSI pattern at the TrackBot includes prominent hints of multi-path components which in turn indicates the presence of reflective surfaces. This motivated us to look into the possibility of employing the TrackBot system for mapping purpose instead of localization and tracking. Through a set of simulation and real experiments with the Trackbot, we demonstrate the potential of the proposed concept. To our knowledge, this is the first bistatic radar system demonstrated with low-cost low power off-the-shelf 802.15.4 radios.

8.1 Problem Formulation

In this section, we explain the problem formulation and our proposed system design in details. Let there exist a set of N reflecting *point objects*[2] in a 2D unknown environment with known dimensions, say $D_G \times D_G$ square region. Denote the location

[2]For simplicity, we ignore the dimensions of the reflecting objects and represent them by the points of reflection on the objects.

of the point reflectors as $\mathbf{X}_{\mathcal{R}} = \{X^i_{\mathcal{R}} = (x^i_{\mathcal{R}}, y^i_{\mathcal{R}}) | i = 1, \cdots N\}$. Our objective is to estimate the unknown N and also the unknown location set $\mathbf{X}_{\mathcal{R}}$.

Our proposed system consists of a single directional antenna based receiver (R_x) with known gain pattern, $\mathbf{g_a} = \{g_{(-180+i\cdot\delta)} : i \in \mathbb{Z} \text{ and } i \in [0, 360/\delta)\}$ where δ is the angular scanning granularity and $g_{(\phi)}$ refers to the antenna gain along ϕ direction, and a single omnidirectional antenna based transmitter (T_x). Now, let the environment be modeled as a 2D discrete grid space with origin at the receiver location and positive y-axis direction of the 2D space along the 0° orientation of the receiver. The length of a side of each grid (d_G) is a parameter to control the granularity of the estimation. The number of grid point is n^2_G with the set of locations denoted as $\mathbf{X}_G = \{X^i_G = (x^i_G, y^i_G) | i = 1, \cdots n^2_G\}$. Let us assume that the reflector locations and the transmitter location are restricted to the set of the grid points, $\mathbf{X}_G$. Let us also denote the locations of the transmitter and the receiver as X_{T_x} and X_{R_x}, respectively, and the distance between the T_x-R_x pair as d. The directional rotating receiving antenna collects a set of RSSI samples[3] for different angular orientation of the antenna, starting from $-180°$ (orientation toward negative y axis of the 2D reference frame) to 180° in steps of $\delta°$, to generate a RSSI vector, $\mathbf{r_o} = \{r_{(-180+i\cdot\delta)} : i \in \mathbb{Z} \text{ and } i \in [0, 360/\delta)\}$. This RSSI vector is used by the proposed MLE based reflector localization algorithm, detailed in Section 8.2.

[3]We do not employ the RF phase information as it is not readily available (unlike RSSI) in most of the cheap, low power off-the-shelf radios.

For the wireless channel modelling, we use standard log-normal fading model [18]. For directional antenna, the model can be described as follows.

$$P_{R_x}(\theta) = \mathcal{C}.g_{(\theta)}.P_{T_x}.10^{\Psi/10} \cdot d_{(\theta)}^{-\gamma} \tag{8.1}$$

where $P_{R_x}(\theta)$ is the received signal power along direction θ with respect to the antenna orientation, $g_{(\theta)}$ is the directional gain of receiving antenna, $\mathcal{C}$ is a constant, P_{T_x} is the transmitter power, $d_{(\theta)}$ is the distance travelled by signal incident along angle θ, γ is the path loss exponent, and $\Psi \sim \mathcal{N}(0, \sigma^2)$ is the log normal fading noise with variance σ^2. If the signal is a reflected signal, there will be some attenuation by the reflecting object which we denote by $\mathcal{A}(\kappa)$, where κ is the angle of incidence on the reflecting object. However, for simplicity, we take a constant value $\mathcal{A}$ as the attenuation constant. Also, note that the coefficient of reflection is just 1 for the direct path component. Now, for each orientation of the receiver antenna, say, θ_o with respect to positive y-axis, the measured RSSI is actually a sum of different multipath components and can be represented as follows.

$$r'_{\theta_o} = \sum_{\theta \in [-180°, 180°)} \mathcal{C}.\mathcal{A}.g_{(\theta-\theta_o)}.P_{T_x}.10^{\Psi/10} \cdot d_{(\theta)}^{-\gamma} \tag{8.2}$$

where θ actually signifies the angle of the multipath component with respect to the positive y axis (i.e, the rotating antenna's base orientation). Now using (8.2) for each possible set of reflector locations, $\mathbf{X}_{\mathcal{R}}$, and transmitter location, X_{T_x}, we

can mathematically estimate the resulting RSSI vector as $\mathbf{r_v} = \{r'_{(-180+i\cdot\delta)} : i \in \mathbb{Z} \text{ and } i \in [0, 360/\delta)\}$. Now the objective can be summarized as follows: *find the most likely location set* $\mathbf{X}_\mathcal{R}$ *such that, for a known (or unknown) location of the transmitter* X_{T_x}*, the probability* $\mathbb{P}(\mathbf{r_v} = \mathbf{r_o})$ *is the highest, where* $\mathbf{r_o}$ *is the RSSI observation vector and the cardinality of the set* $\mathbf{X}_\mathcal{R}$ *i.e.,* N*, is also unknown.*

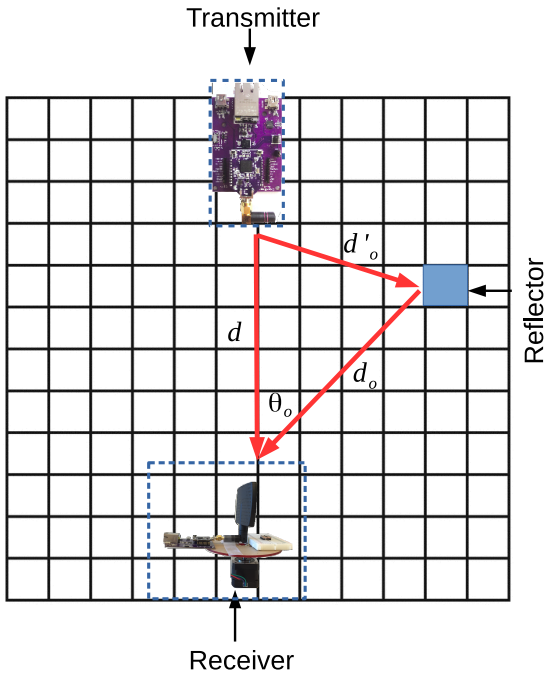


Figure 8.1: Illustration of our bistatic radar equivalent system

8.2 Maximum Likelihood Estimation

In this section, we describe our maximum likelihood estimation [164] based approach for localizing a set of reflector objects. First, let us simplify the problem by restricting our focus to a single reflector scenario with known transmitter location,

X_{T_x}. Next, we calculate the log likelihood of each of the grid point X_G^i to be the reflector location, $X_{\mathcal{R}}$, as follows:

$$\log \mathcal{L}(\mathbf{r_o}|X_{\mathcal{R}} = X_G^i, X_{T_x}) = \sum_{\theta_o \in [-180°, 180°)} \log \mathbb{P}(r'_{\theta_o} = r_{\theta_o}|X_{\mathcal{R}} = X_G^i, X_{T_x}) \tag{8.3}$$

where we use Eqn (8.2) to estimate the probability $\mathbb{P}(r'_{\theta_o} = r_{\theta_o}|X_{\mathcal{R}} = X_G^i, X_{T_x})$. Then we choose the grid position with the maximum value of the likelihood function as the estimated reflector location.

$$X_{\mathcal{R}} = \arg\max_{X_G^i} \log \mathcal{L}(\mathbf{r_o}|X_{\mathcal{R}} = X_G^i, X_{T_x}) \tag{8.4}$$

This requires $\mathbb{O}(n_G^2)$ numbers of likelihood estimations where the number of grid points, n_G^2, depends on the search granularity, d_G, and the size of the environment, D_G. Now, if the location of the transmitter X_{T_x} is also unknown, the MLE formulation can be written as follows.

$$\begin{aligned} \{X, X_{T_x}\} &= \arg\max_{\{X_G^i, X_G^j\}} \log \mathcal{L}(\mathbf{r_o}|X_{\mathcal{R}} = X_G^i, X_{T_x} = X_G^j) \\ &= \arg\max_{\{X_G^i, X_G^j\}} \sum_{\theta_o \in [-180°, 180°)} \log \mathbb{P}(r'_{\theta_o} = r_{\theta_o}|X_{\mathcal{R}} = X_G^i, X_{T_x} = X_G^j) \end{aligned} \tag{8.5}$$

where $i, j \in \{1, n_G^2\}$. This requires $\mathbb{O}(n_G^4)$ numbers of likelihood estimation. Generally speaking, *for a known number of reflectors (N)*, the MLE based approach iterates through different subsets of the grid points, X_G, as the possible set of the

transmitter and reflector(s) locations and calculates the likelihood probability of the observation vector $\mathbf{r_v}$. Through comparing these likelihoods of the measured RSSI vector, we obtain the most likely location set for the reflectors and the transmitter. Now, *if N is also unknown*, we need to perform likelihood for a range of values of N as well. In such contexts, the MLE formulation can be expressed as follows.

$$\begin{aligned}\{X, X_{T_x}, N\} &= \underset{\{\mathbf{X}_\mathcal{R}, X_G^j, k\}}{\arg\max} \log \mathcal{L}(\mathbf{r_o}|\mathbf{X}_\mathcal{R}, X_{T_x} = X_G^j, N = k) \\ &= \underset{\{\mathbf{X}_\mathcal{R}, X_G^j, k\}}{\arg\max} \sum_{\theta_o \in [-180^\circ, 180^\circ)} \log \mathbb{P}(r'_{\theta_o} = r_{\theta_o}|\mathbf{X}_\mathcal{R}, X_{T_x} = X_G^j, N = k)\end{aligned} \tag{8.6}$$

where $\mathbf{X}_\mathcal{R} \subset \mathbf{X}_G^k$ is a k dimensional vector with n_G^{2k} possible values for $N = k$ and $j \in \{1, n_G^2\}$.

8.3 Performance Evaluations

8.3.1 Simulation Experiment Results

To verify the localization performance of the MLE algorithm, we performed a set of simulation experiments. The simulated RSSI data were generated based on the standard path loss model [18] with log-normal fading noise Ψ with a maximum standard deviation of $\sigma^2 = 5$. The path loss exponent, γ, was set to be 1.856 in order to match our real experiments, detailed later. The transmitter power P_{T_X}

is set to be $7dBm$ to match the maximum transmission power of Openmotes [139] used in real experiments.

To simulate the resultant RSSI for an environment with reflecting objects ($\mathbf{X}_{\mathcal{R}}$), the simulated RSSI values for the transmitter was superposed with simulated multi-path RSSI values contributed by each of the reflective surfaces. We use a granularity of $\delta = 1.8°$ to match the granularity of our real system implementation. The grid granularity, d_G, is set to be 1m since the standard RSSI based localization errors are in the order of meters. The distance traveled by multipath components are calculated using cosine rule as follows: $d_r = \sqrt{d^2 + d_o^2 - 2.d.d_o.cos(\theta_o)} + d_o$ where d is the distance between the transmitter and the receiver, d_o is the distance between the reflector and the receiver, and θ_o is the angle formed by the transmitter and the reflecting object, at the receiver (illustrated in Fig. 8.1).

We performed the simulation experiment for a single reflector with a known location of the transmitter as well as with unknown location of the transmitter. In Fig. 8.2, we present the probability heat map for a single reflector with known transmitter location. In this instance, the transmitter was placed at $(0, 3)$ while the receiver was located at $(0, 0)$. The reflector was placed at $(3, 3)$. Figure 8.2 shows that the grid locations close to the actual position of the reflector have higher probability according to MLE than any other grid locations. To further analyze the performance, we vary the position of the reflector as well as the transmitter. In Fig. 8.3, we present the error performance statistics for a fixed location of the

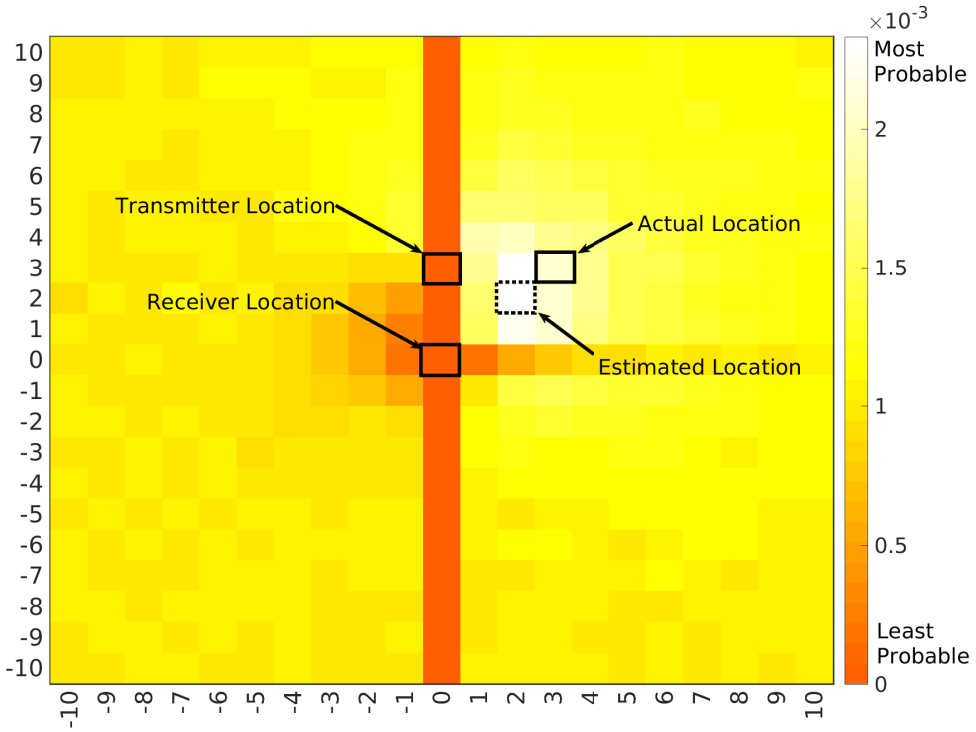


Figure 8.2: Probability heat map of different possible positions of the reflector for known position of transmitter $(0, 3)$ and receiver $(0, 0)$.

reflector at $(3, 3)$ while the distance to the transmitter, d, is varied from 1m to 10m. Figure 8.3 illustrates that the performance of the MLE is worse when the transmitter is much closer to the receiver than the reflector i.e., $d << d_o$. In this case, the power of the reflected signal is much lower than the direct path power from a nearby transmitter. As the distance between the transmitter and the receiver (d) increases, the error decreases. Based on this observation, we hypothesize that MLE detection performance improves when direct path power and the multipath power are comparable, and deteriorates when the multipath power is much smaller compared to the direct path power. To further verify this hypothesis, we fixed the

location of the transmitter at $(0, 3)$ and varied the distance between the receiver and the reflector (d_o) from 1m to 10m. The experiment outcomes, illustrated in Fig. 8.4, shows that with increasing distance to the reflector, the performance prominently deteriorate. This validates our hypothesis.

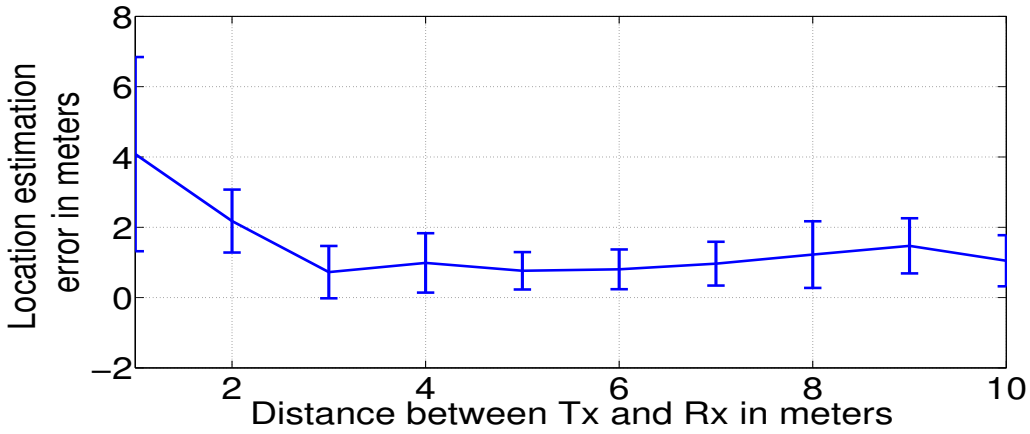


Figure 8.3: Error Statistics for varying distance between T_x and R_x while the reflector is kept fixed at $(3, 3)$

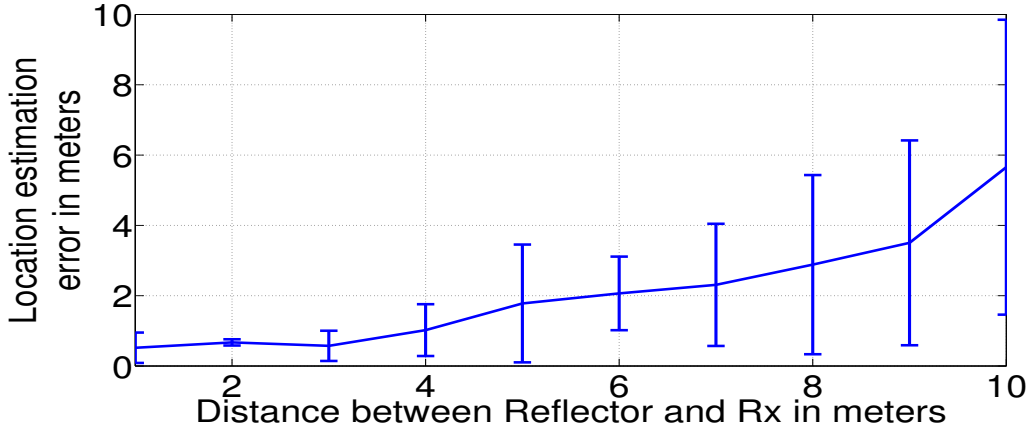


Figure 8.4: Error statistics for varying distance to the reflector from the receiver while the T_x is kept fixed at $(0, 3)$

We also performed a small set of experiments with unknown locations of the transmitter. In this set of experiments, the distance between T_x and R_x was varied

while the reflector is kept static at $(2, 2)$. The results are summarized in Table 8.1. These results concur with the earlier results i.e., when the distance d is comparable or higher than the distance d_o, the performance of the MLE based estimation is better. It also shows that the error in the detection of the transmitter location is very small ($< 1m$) in all three cases.

Table 8.1: Simulation experiment based error statistics (in meters) for unknown transmitter and unknown reflector

$T_x - R_x$ Distance (d)	(Mean, Std) of Errors in T_x location	(Mean, Std) of Errors in reflector location
1m	$(0, 0)$	$(3.7708, 3.9555)$
2m	$(0.4, 0.2667)$	$(2.2494, 0.5166)$
3m	$(0.7, 0.4556)$	$(2.5891, 1.0524)$

8.3.2 Real Experiment Results

To test the practicality of the concept, we use the TrackBot system presented in Fig. 4.1 where we mainly employ the rotating platform with the directional antenna as illustrated in Fig. 8.5. For completeness, we provide a brief overview of the portion of TrackBot system used for the experimentation.

The rotating receiver module of the TrackBot system, which we use for this study, consists of a stepper motor, a stepper motor driver, and an embedded sensor node (Openmote [139]) with an external directional antenna. Openmote is an open hardware platform for implementing open source IoT standard protocols. It consists of a TI 32-bit CC2538 @ 32 MHz with 512KB Flash memory, 32KB RAM, and 2.4GHz IEEE 802.15.4-based Transceiver connected via SMA plug. The directional

antenna used is Rosewill Model RNX-AD7D that works for both WiFi bands, i.e., 2.4MHz and 5GHz, with maximum Gain of 5dBi and 7dBi, respectively. The Half Power Beam Widths (HPBW) of the antenna are 70° and 50° for 2.4GHz and 5GHz, respectively.

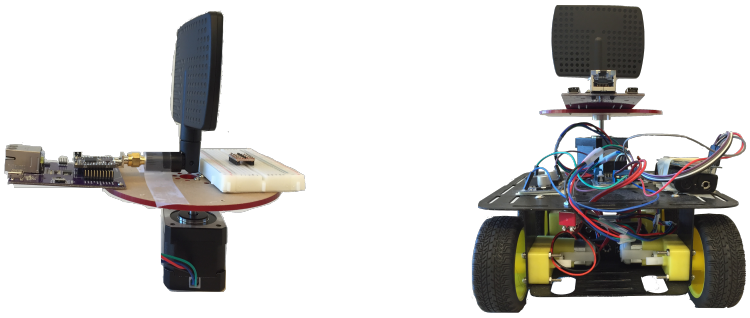


Figure 8.5: Real System for Experimentation: We only employ the rotating platform with the directional antenna (left) of the TrackBot (right) developed in the course of Study 1.

The whole idea of using an antenna with such a wide directional HPBW is to demonstrate that the system can be built with cheap, off-the-shelf antennas instead of costly, custom solutions. In this system, the transceiver of the Openmote is switched with the directional antenna. The Openmote with the directional antenna is mounted on a stepper motor using a mounting plate. The stepper motor used for this purpose is a Nema 17, 4-wire bipolar motor of dimension $1.65'' \times 1.65'' \times 1.57''$ with step size of 1.8° (200 steps/rev). The Rated current is 2A and the rated resistance is 1.1 Ohms. The motor driver used for this purpose is an EasyDriver - Stepper Motor Driver. We use an mbed NXP LPC1768 for precise control of the motor via the Easydriver. The Openmote sends interrupts to the mbed whenever

it collects an RSSI measurement and sends the data through serial communication, whereas the mbed generates proper output to turn the motor periodically by one step, i.e., 1.8°. In our implementation, the rotation directions are alternated in consecutive cycles in order to avoid wire twisting issues. The transmitter is this case is a standalone Openmote. For programming of the Openmotes, we used the open source RIOT OS [144] for low power IoT Devices. The rotating antenna platform completes a full scanning in $2s$ to generate the output RSSI vector, $\mathbf{r_o}$. This RSSI data is then fed to a computer via USB for processing. The size of the entire robot is roughly $8'' \times 8'' \times 8''$.

With this system, we collected a set of real measurement data via a range of experiments in a controlled anechoic chamber environment with precise control over the reflector locations. The anechoic chamber prevents uncontrollable reflection of radio waves from any surrounding surface not accounted for. The transmitter was placed at $\approx 0°$ with respect to the receiver antenna assembly. We performed the experiment for three sets of configurations with a single reflector. We used a metallic plate as a reflector with the center located at $\approx (1, 1)$ in the 2D space illustrated in the problem formulation. An illustration of the experiment is presented in Fig. 8.1. For each configuration, we collected 100 sets of RSSI vector. We fed this vectors to a MATLAB code to process the data and estimate the location of the reflector. We varied the distance d between the transmitter and the receiver to be 1m, 2m, and 3m, respectively. The granularity of the search grid, d_G, is set to be 0.5m

in this set of experiments. For this set of experiments, we calculated the path loss exponent ($\gamma = 1.856$) and the constants in Eqn (8.2) based on some initial RSSI measurements. We use a transmission power of $P_{T_x} = 7dBm$ which is the maximum transmission power of the Openmote. Note that, in these experiments, some extra unaccountable errors are still introduced by the fact that the reflector is not a point source. The error statistics for this set of experiments is presented in Table 8.2. The results suggest that the proposed system performs well with an error in the order of meters. The similar performance in all three cases is justified by the fact that d and d_o are comparable for all three cases. With this system, one can potentially achieve reasonable mapping performance up to $\approx 6m$ separations among the transmitter, the receiver, and the reflectors.

Table 8.2: Error statistics for real-world experiments in meters

$T_x - R_x$ Distance (d)	Mean of Errors	Std of Errors
1m	0.9m	0.1m
2m	1.4m	0.1m
3m	0.7m	0.2m

8.4 Discussion

In this study, we have presented a proof of concept method of employing a single RF transmitter and a single RF receiver toward passively localizing reflecting objects which can be potentially extended to a full-fledged mapping of the environment. We use an MLE based estimation method for this purpose. Based on some base

simulation and real-world experiments, we demonstrated that the proposed system works with reasonable performance for a single unknown reflector with both known and unknown location of the transmitter. This system can be easily integrated with the ARREST system presented in Chapter 4 to build an integrated system for relative localization, tracking, and mapping in an RWN deployment context. Nonetheless, the MLE based approach is definitely not scalable as the search space increases exponentially with the number of unknowns. Thus, the research direction for a more scalable and efficient approach with similar performance guarantees is still open.

Chapter 9

Conclusion

With robots being heavily integrated into our daily life, there is an increasing focus towards employing a network of robots to collaboratively perform a set of tasks such as exploration of unknown terrains or providing temporary communication backbone. This has led researchers to invest significant attention towards the cutting-edge field of Robotic Wireless Networks with main focus on the integrated communication and control systems. The controllability of the robotic nodes has opened up a whole new dimension in the design of tradition solutions such as communication protocols and localization techniques for RWN. Moreover, the networking between controllable moving agents has imposed additional demands on the traditional communication protocols such as light-weight communication, low-bandwidth communication, and timeliness.

In this thesis, we addressed key problems pertaining to five key areas of research in RWN:

- Localization and Relative Positioning
- Communication-Aware Robot Positioning
- Network Layer Protocol: Routing Protocols
- Application Layer Protocol: Unified Protocol for Control and Sensing
- RSSI Models, Measurements, and RF mapping

All these work combined leads towards a Robotic Wireless Network system where the developed miniRadar system is used for modeling communication characteristics as well as mapping of an unknown environment, the ARREST system is used for relative localization and maintaining certain distance that is required for connectivity and good communication links, the HDCP routing protocol is used as the underlying routing protocol, ROMANO is the overlay application layer communication and control protocol, and finally the calculated bound is used to choose the number of robots to deploy to support certain communication demands. Therefore, this thesis is not just meant to provide solutions to some individual problems. Rather, it is meant to provide starting points and guidelines for building a complete robotic wireless network system with the main focus on communication oriented application such as using the RWN to support a temporary wireless communication backbone. All the studies are presented with sufficient details and pointers to open-source material for any researcher to replicate the systems for further research.

Reference List

[1] Kalman-Filtering, "`https://en.wikipedia.org/wiki/Kalman_filter`."

[2] R. Banirazi, E. Jonckheere, and B. Krishnamachari, "Heat-diffusion: Pareto optimal dynamic routing for time-varying wireless networks," in *Proceedings of the IEEE International Conference on Computer Communications (INFOCOM)*, 2014.

[3] J. Penders, L. Alboul, U. Witkowski, A. Naghsh, J. Saez-Pons, S. Herbrechtsmeier, and M. El-Habbal, "A robot swarm assisting a human firefighter," *Advanced Robotics*, vol. 25, no. 1-2, pp. 93–117, 2011.

[4] R. R. Murphy, "Trial by fire [rescue robots]," *IEEE Robotics & Automation Magazine*, vol. 11, no. 3, pp. 50–61, 2004.

[5] V. Gazi and K. M. Passino, *Swarm Stability and Optimization*. Springer, 2011.

[6] E. Aahin and A. Winfield, "Special issue on swarm robotics," *Swarm Intelligence*, vol. 2, no. 2-4, pp. 69–72, 2008.

[7] R. K. Williams, A. Gasparri, and B. Krishnamachari, "Route swarm: Wireless network optimization through mobility," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2014.

[8] Y. Mostofi, "Cooperative wireless-based obstacle/object mapping and see-through capabilities in robotic networks," *IEEE Transactions on Mobile Computing*, vol. 12, no. 5, pp. 817–829, 2013.

[9] Y. Yan and Y. Mostofi, "Robotic router formation in realistic communication environments," *IEEE Transactions on Robotics*, vol. 28, no. 4, pp. 810–827, 2012.

[10] S. Gil, D. Feldman, and D. Rus, "Communication coverage for independently moving robots," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2012.

[11] A. Ollero, J. Alcázar, F. Cuesta, F. López-Pichaco, and C. Nogales, "Helicopter teleoperation for aerial monitoring in the comets multi-uav system," in *Proceedings of the 3rd IARP Workshop on Service, Assistive and Personal Robots*, 2003.

[12] S. Thrun, S. Thayer, W. Whittaker, C. Baker, W. Burgard, D. Ferguson, D. Hahnel, D. Montemerlo, A. Morris, and Z. Omohundro, "Autonomous exploration and mapping of abandoned mines," *IEEE Robotics & Automation Magazine*, vol. 11, no. 4, pp. 79–91, 2004.

[13] M. McClure, D. R. Corbett, and D. W. Gage, "The darpa landroids program," in *Proceedings of the SPIE Defense, Security, and Sensing*, 2009.

[14] R. R. Murphy, J. Kravitz, S. L. Stover, and R. Shoureshi, "Mobile robots in mine rescue and recovery," *IEEE Robotics & Automation Magazine*, vol. 16, no. 2, pp. 91–103, 2009.

[15] M. D. Weiss, J. Peak, and T. Schwengler, "A statistical radio range model for a robot manet in a subterranean mine," *IEEE Transactions on Vehicular Technology*, vol. 57, no. 5, pp. 2658–2666, 2008.

[16] S.-H. Baeg, J.-H. Park, J. Koh, K.-W. Park, and M.-H. Baeg, "Building a smart home environment for service robots based on rfid and sensor networks," in *Proceedings of the IEEE International Conference on Control, Automation and Systems (ICCAS)*, 2007.

[17] IEEE Networked Robots Report, "`http://www-users.cs.umn.edu/~isler/tc/`."

[18] T. S. Rappaport, *Wireless communications: principles and practice*, vol. 2. Prentice Hall PTR New Jersey, 1996.

[19] G. A. Hollinger, S. Choudhary, P. Qarabaqi, C. Murphy, U. Mitra, G. Sukhatme, M. Stojanovic, H. Singh, and F. Hover, "Communication protocols for underwater data collection using a robotic sensor network," in *Proceedings of the IEEE GLOBECOM Workshops (GC Wkshps)*, 2011.

[20] H. Pham and S. Jha, "An adaptive mobility-aware mac protocol for sensor networks (ms-mac)," in *Proceedings of the IEEE International Conference on Mobile Adhoc and Sensor Systems (MASS)*, 2004.

[21] M. Ali, T. Suleman, and Z. A. Uzmi, "Mmac: A mobility-adaptive, collision-free mac protocol for wireless sensor networks," in *Proceedings of the IEEE International Performance, Computing, and Communications Conference (IPCCC)*, 2005.

[22] A. Jhumka and S. Kulkarni, "On the design of mobility-tolerant tdma-based media access control (mac) protocol for mobile sensor networks," in *Distributed Computing and Internet Technology*, pp. 42–53, Springer, 2007.

[23] S. R. Das, E. M. Belding-Royer, and C. E. Perkins, "Ad hoc on-demand distance vector (aodv) routing," 2003.

[24] I. D. Chakeres and E. M. Belding-Royer, "Aodv routing protocol implementation design," in *Proceedings of the IEEE International Conference on Distributed Computing Systems Workshops*, 2004.

[25] S. E. Deering, "Internet protocol, version 6 (ipv6) specification," 1998.

[26] D. Johnson, Y.-c. Hu, and D. Maltz, "The dynamic source routing protocol (dsr) for mobile ad hoc networks for ipv4," tech. rep., RFC 4728, 2007.

[27] T. Clausen and P. Jacquet, "Optimized link state routing protocol (olsr)," tech. rep., RFC 3626, 2003.

[28] D. Johnson, N. Ntlatlapa, and C. Aichele, "Simple pragmatic approach to mesh routing using batman," 2008.

[29] S. Wang, A. Gasparri, and B. Krishnamachari, "Robotic message ferrying for wireless networks using coarse-grained backpressure control," *IEEE Transactions on Mobile Computing*, vol. PP, no. 99, pp. 1–1, 2016.

[30] D. W. Gage, "Network protocols for mobile robot systems," in *Proceedings of the Intelligent Systems & Advanced Manufacturing*, pp. 107–118, International Society for Optics and Photonics, 1998.

[31] Robot Operating System, "http://www.ros.org/."

[32] R. Olfati-Saber and R. M. Murray, "Consensus problems in networks of agents with switching topology and time-delays," *IEEE Transactions on Automatic Control*, vol. 49, pp. 1520–1533, Sept 2004.

[33] L. Xiao, S. Boyd, and S. Lall, "A scheme for robust distributed sensor fusion based on average consensus," in *Proceedings of the ACM/IEEE International Conference on Information Processing in Sensor Networks*, 2005.

[34] M. Franceschelli and A. Gasparri, "Gossip-based centroid and common reference frame estimation in multiagent systems," *IEEE Transactions on Robotics*, vol. 30, pp. 524–531, April 2014.

[35] Z. Lin, B. Francis, and M. Maggiore, "Necessary and sufficient graphical conditions for formation control of unicycles," *IEEE Transactions on Automatic Control*, vol. 50, pp. 121–127, Jan 2005.

[36] P. Yang, R. A. Freeman, and K. M. Lynch, "Multi-agent coordination by decentralized estimation and control," *IEEE Transactions on Automatic Control*, vol. 53, pp. 2480–2496, Dec 2008.

[37] M. Guo, M. M. Zavlanos, and D. V. Dimarogonas, "Controlling the relative agent motion in multi-agent formation stabilization," *IEEE Transactions on Automatic Control*, vol. 59, pp. 820–826, March 2014.

[38] D. V. Dimarogonas and K. J. Kyriakopoulos, "Connectedness Preserving Distributed Swarm Aggregation for Multiple Kinematic Robots," *IEEE Transactions on Robotics*, vol. 24, no. 5, pp. 1213–1223, 2008.

[39] D. P. Spanos and R. M. Murray, "Robust connectivity of networked vehicles," in *Proceedings of the IEEE Conference on Decision and Control (CDC)*, 2004.

[40] Z. Yao and K. Gupta, "Backbone-based connectivity control for mobile networks," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2009.

[41] D. V. Dimarogonas and K. H. Johansson, "Bounded control of network connectivity in multi-agent systems," *IET control theory & applications*, vol. 4, no. 8, pp. 1330–1338, 2010.

[42] F. Knorn, R. Stanojevic, M. Corless, and R. Shorten, "A framework for decentralised feedback connectivity control with application to sensor networks," *International Journal of Control*, vol. 82, no. 11, pp. 2095–2114, 2009.

[43] M. Schuresko and J. Cortés, "Distributed motion constraints for algebraic connectivity of robotic networks," *Journal of Intelligent and Robotic Systems*, vol. 56, no. 1-2, pp. 99–126, 2009.

[44] R. K. Williams, A. Gasparri, G. S. Sukhatme, and G. Ulivi, "Global connectivity control for spatially interacting multi-robot systems with unicycle kinematics," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2015.

[45] S. Gil, S. Kumar, D. Katabi, and D. Rus, "Adaptive communication in multi-robot systems using directionality of signal strength," *The International Journal of Robotics Research*, vol. 34, no. 7, pp. 946–968, 2015.

[46] G. Tuna, B. Nefzi, and G. Conte, "Unmanned aerial vehicle-aided communications system for disaster recovery," *Journal of Network and Computer Applications*, vol. 41, pp. 27–36, 2014.

[47] P. Ghosh, R. Pal, and B. Krishnamachari, "Towards controllability of wireless network quality using mobile robotic routers," *CoRR*, vol. abs/1607.07848, 2016.

[48] H. Liu, H. Darabi, P. Banerjee, and J. Liu, "Survey of wireless indoor positioning techniques and systems," *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, vol. 37, no. 6, pp. 1067–1080, 2007.

[49] G. Sun, J. Chen, W. Guo, and K. R. Liu, "Signal processing techniques in network-aided positioning: a survey of state-of-the-art positioning designs," *IEEE Signal Processing Magazine*, vol. 22, no. 4, pp. 12–23, 2005.

[50] I. Guvenc and C.-C. Chong, "A survey on toa based wireless localization and nlos mitigation techniques," *IEEE Communications Surveys & Tutorials*, vol. 11, no. 3, pp. 107–124, 2009.

[51] I. Amundson and X. D. Koutsoukos, "A survey on localization for mobile wireless sensor networks," in *Mobile Entity Localization and Tracking in GPS-less Environnments*, pp. 235–254, Springer, 2009.

[52] Y. Mao, C. You, J. Zhang, K. Huang, and K. B. Letaief, "A survey on mobile edge computing: The communication perspective," *IEEE Communications Surveys & Tutorials*, vol. 19, no. 4, pp. 2322–2358, 2017.

[53] M. Narang, S. Xiang, W. Liu, J. Gutierrez, L. Chiaraviglio, A. Sathiaseelan, and A. Merwaday, "Uav-assisted edge infrastructure for challenged networks," in *Proceedings of the IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, 2017.

[54] X. Wang, A. Chowdhery, and M. Chiang, "Networked drone cameras for sports streaming," in *Proceedings of IEEE International Conference on Distributed Computing Systems (ICDCS)*, 2017.

[55] N. P. Papanikolopoulos, P. K. Khosla, and T. Kanade, "Visual tracking of a moving target by a camera mounted on a robot: A combination of control and vision," *IEEE Transactions on Robotics and Automation*, vol. 9, no. 1, pp. 14–35, 1993.

[56] B. Jung and G. S. Sukhatme, "Detecting moving objects using a single camera on a mobile robot in an outdoor environment," in *Proceedings of the International Conference on Intelligent Autonomous Systems (IAS)*, 2004.

[57] M. Lindström and J. Eklundh, "Detecting and tracking moving objects from a mobile platform using a laser range scanner," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2001.

[58] G. Han, H. Xu, T. Q. Duong, J. Jiang, and T. Hara, "Localization algorithms of wireless sensor networks: a survey," *Telecommunication Systems*, vol. 52, no. 4, pp. 2419–2436, 2013.

[59] Y. Yan and Y. Mostofi, "Robotic router formation-a bit error rate approach," in *Proceedings of the Military Communications Conference (MILCOM)*, 2010.

[60] A. Dunkels, O. Schmidt, N. Finne, J. Eriksson, F. Österlind, N. Tsiftes, and M. Durvy, "The contiki os: The operating system for the internet of things," 2011.

[61] MQTT-SN, "http://mqtt.org/."

[62] N. J. Willis, *Bistatic radar*, vol. 2. SciTech Publishing, 2005.

[63] Y. Mostofi, "Compressive cooperative sensing and mapping in mobile networks," *IEEE Transactions on Mobile Computing*, vol. 10, no. 12, pp. 1769–1784, 2011.

[64] M. S. Grewal, "Kalman filtering," in *International Encyclopedia of Statistical Science*, pp. 705–708, Springer, 2011.

[65] M. Athans, "The role and use of the stochastic linear-quadratic-gaussian problem in control system design," *IEEE Transactions on Automatic Control*, vol. 16, no. 6, pp. 529–552, 1971.

[66] D. P. Bertsekas, *Dynamic programming and optimal control*, vol. 1 & 2. Athena Scientific Belmont, MA, 1995.

[67] P. Lancaster and L. Rodman, *Algebraic riccati equations*. Clarendon press, 1995.

[68] L. Tassiulas and A. Ephremides, "Stability properties of constrained queueing systems and scheduling policies for maximum throughput in multihop radio networks," *IEEE Transactions on Automatic Control*, vol. 37, no. 12, pp. 1936–1948, 1992.

[69] L. Georgiadis, M. J. Neely, and L. Tassiulas, *Resource allocation and cross-layer control in wireless networks*. Now Publishers Inc, 2006.

[70] M. J. Neely, "Stochastic network optimization with application to communication and queueing systems," *Synthesis Lectures on Communication Networks*, vol. 3, no. 1, pp. 1–211, 2010.

[71] D. Schulz, W. Burgard, D. Fox, and A. B. Cremers, "Tracking multiple moving targets with a mobile robot using particle filters and statistical data association," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2001.

[72] I. Marković, F. Chaumette, and I. Petrović, "Moving object detection, tracking and following using an omnidirectional camera on a mobile robot," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2014.

[73] E. Prassler, J. Scholz, and A. Elfes, "Tracking multiple moving objects for real-time robot navigation," *Autonomous Robots*, vol. 8, no. 2, pp. 105–116, 2000.

[74] M. Kleinehagenbrock, S. Lang, J. Fritsch, F. Lomker, G. A. Fink, and G. Sagerer, "Person tracking with a mobile robot based on multi-modal anchoring," in *Proceedings of the IEEE International Symposium on Robot and Human interactive Communication (RO-MAN)*, 2002.

[75] J. Graefenstein, A. Albert, P. Biber, and A. Schilling, "Wireless node localization based on rssi using a rotating antenna on a mobile robot," in *Proceedings of the IEEE Workshop on Positioning, Navigation and Communication (WPNC)*, 2009.

[76] P. Tokekar, J. Vander Hook, and V. Isler, "Active target localization for bearing based robotic telemetry," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2011.

[77] J. N. Twigg, J. R. Fink, L. Y. Paul, and B. M. Sadler, "Rss gradient-assisted frontier exploration and radio source localization," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2012.

[78] P. Tokekar, E. Branson, J. Vander Hook, and V. Isler, "Tracking aquatic invaders: Autonomous robots for monitoring invasive fish," *IEEE Robotics & Automation Magazine*, vol. 20, no. 3, pp. 33–41, 2013.

[79] P. Tokekar, V. Isler, and A. Franchi, "Multi-target visual tracking with aerial robots," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2014.

[80] S. Zickler and M. Veloso, "Rss-based relative localization and tethering for moving robots in unknown environments," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2010.

[81] L. Oliveira, H. Li, L. Almeida, and T. E. Abrudan, "Rssi-based relative localisation for mobile robots," *Ad Hoc Networks*, vol. 13, pp. 321–335, 2014.

[82] J. Pugh, X. Raemy, C. Favre, R. Falconi, and A. Martinoli, "A fast onboard relative positioning module for multirobot systems," *IEEE/ASME Transactions on Mechatronics*, vol. 14, no. 2, pp. 151–162, 2009.

[83] F. Rivard, J. Bisson, F. Michaud, and D. Létourneau, "Ultrasonic relative positioning for multi-robot systems," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2008.

[84] D. Vasisht, S. Kumar, and D. Katabi, "Decimeter-level localization with a single wifi access point," in *Proceedings of the USENIX Symposium on Networked Systems Design and Implementation (NSDI)*, 2016.

[85] J. Elson, "Simulation of constant-distance robot following using radio proximity beacons." `http://www.circlemud.org/jelson/writings/constant-following/constant-following.html`, 1999.

[86] B.-C. Min and E. T. Matson, "Robotic follower system using bearing-only tracking with directional antennas," in *Robot Intelligence Technology and Applications 2*, pp. 37–58, Springer, 2014.

[87] J. Van Den Berg, P. Abbeel, and K. Goldberg, "Lqg-mp: Optimized path planning for robots with motion uncertainty and imperfect state information," *The International Journal of Robotics Research*, vol. 30, no. 7, pp. 895–913, 2011.

[88] J. Van Den Berg, D. Wilkie, S. J. Guy, M. Niethammer, and D. Manocha, "Lqg-obstacles: Feedback control with collision avoidance for mobile robots with motion and sensing uncertainty," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2012.

[89] J. Tornero, R. Piza, P. Albertos, and J. Salt, "Multirate lqg controller applied to self-location and path-tracking in mobile robots," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2001.

[90] J. Van Den Berg, S. Patil, R. Alterovitz, P. Abbeel, and K. Y. Goldberg, "Lqg-based planning, sensing, and control of steerable needles," in *Algorithmic Foundations of Robotics IX*, pp. 373–389, Springer, 2010.

[91] M. Haenggi, "Mean interference in hard-core wireless networks," *IEEE Communications Letters*, vol. 15, no. 8, pp. 792–794, 2011.

[92] R. K. Ganti and M. Haenggi, "Interference and outage in clustered wireless ad hoc networks," *IEEE Transactions on Information Theory*, vol. 55, no. 9, pp. 4067–4086, 2009.

[93] A. Busson and G. Chelius, "Capacity and interference modeling of csma/ca networks using ssi point processes," *Telecommunication Systems*, vol. 57, no. 1, pp. 25–39, 2014.

[94] H. ElSawy and E. Hossain, "Modeling random csma wireless networks in general fading environments," in *Proceedings of the IEEE International Conference on Communications Workshops (ICC)*, 2012.

[95] A. Busson and G. Chelius, "Point processes for interference modeling in csma/ca ad-hoc networks," in *Proceedings of the 6th ACM symposium on Performance evaluation of wireless ad hoc, sensor, and ubiquitous networks*, 2009.

[96] P. Cardieri, "Modeling interference in wireless ad hoc networks," *IEEE Communications Surveys & Tutorials*, vol. 12, no. 4, pp. 551–572, 2010.

[97] R. Hekmat and P. Van Mieghem, "Interference in wireless multi-hop ad-hoc networks and its effect on network capacity," *Wireless Networks*, vol. 10, no. 4, pp. 389–399, 2004.

[98] G. Bianchi, "Performance analysis of the ieee 802.11 distributed coordination function," *IEEE Journal on Selected Areas in Communications*, vol. 18, no. 3, pp. 535–547, 2000.

[99] J. Dai and W. Lin, "Asymptotic optimality of maximum pressure policies in stochastic processing networks," *The Annals of Applied Probability*, vol. 18, no. 6, pp. 2239–2299, 2008.

[100] D. Shah and D. Wischik, "Optimal scheduling algorithms for input-queued switches," in *Proceedings of the IEEE International Conference on Computer Communications (INFOCOM)*, 2006.

[101] M. Naghshvar, H. Zhuang, and T. Javidi, "A general class of throughput optimal routing policies in multi-hop wireless networks," *IEEE Transactions on Information Theory*, vol. 58, no. 4, pp. 2175–2193, 2012.

[102] S. Moeller, A. Sridharan, B. Krishnamachari, and O. Gnawali, "Routing without routes: The backpressure collection protocol," in *Proceedings of the ACM/IEEE International Conference on Information Processing in Sensor Networks*, 2010.

[103] J. Nunez-Martinez, J. Mangues-Bafalluy, and M. Portoles-Comeras, "Studying practical any-to-any backpressure routing in wi-fi mesh networks from a lyapunov optimization perspective," in *Proceedings of the IEEE International Conference on Mobile Adhoc and Sensor Systems (MASS)*, 2011.

[104] M. Alresaini, M. Sathiamoorthy, B. Krishnamachari, and M. J. Neely, "Backpressure with adaptive redundancy (bwar)," in *Proceedings of the IEEE International Conference on Computer Communications (INFOCOM)*, 2012.

[105] O. Gnawali, R. Fonseca, K. Jamieson, D. Moss, and P. Levis, "Collection tree protocol," in *Proceedings of the ACM Conference on Embedded Networked Sensor Systems*, 2009.

[106] F. Ferrari, M. Zimmerling, L. Thiele, and O. Saukh, "Efficient network flooding and time synchronization with glossy," in *Proceedings of the ACM/IEEE International Conference on Information Processing in Sensor Networks*, 2011.

[107] N. Burri, P. Von Rickenbach, and R. Wattenhofer, "Dozer: ultra-low power data gathering in sensor networks," in *Proceedings of the ACM/IEEE International Conference on Information Processing in Sensor Networks*, 2007.

[108] F. Ferrari, M. Zimmerling, L. Mottola, and L. Thiele, "Low-power wireless bus," in *Proceedings of the ACM Conference on Embedded Networked Sensor Systems*, 2012.

[109] T. Winter, "Rpl: Ipv6 routing protocol for low-power and lossy networks," March 2012.

[110] S. Nori, S. Deora, and B. Krishnamachari, "Backip: Backpressure routing in ipv6-based wireless sensor networks," *USC CENG Technical Report CENG-2014-01*.

[111] R. Banirazi, E. Jonckheere, and B. Krishnamachari, "Dirichlet's principle on multiclass multihop wireless networks: minimum cost routing subject to stability," in *Proceedings of the ACM International Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems*, 2014.

[112] C. Bormann, A. P. Castellani, and Z. Shelby, "Coap: An application protocol for billions of tiny internet nodes," *IEEE Internet Computing*, vol. 16, no. 2, pp. 62–67, 2012.

[113] M. H. Amaran, N. A. M. Noh, M. S. Rohmad, and H. Hashim, "A comparison of lightweight communication protocols in robotic applications," *Procedia Computer Science*, vol. 76, pp. 400–405, 2015.

[114] W. Kang, K. Kapitanova, and S. H. Son, "Rdds: A real-time data distribution service for cyber-physical systems," *IEEE Transactions on Industrial Informatics*, vol. 8, no. 2, pp. 393–405, 2012.

[115] Y. Chen and T. Kunz, "Performance evaluation of iot protocols under a constrained wireless access network," in *Proceedings of the International Conference on Selected Topics in Mobile & Wireless Networking (MoWNeT)*, pp. 1–7, IEEE, 2016.

[116] A. S. Huang, E. Olson, and D. C. Moore, "Lcm: Lightweight communications and marshalling," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2010.

[117] N. Aroon, "Study of using mqtt cloud platform for remotely control robot and gps tracking," in *Proceedings of the IEEE International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON)*, 2016.

[118] R. Kazala, A. Taneva, M. Petrov, and S. Penkov, "Wireless network for mobile robot applications," *IFAC-PapersOnLine*, vol. 48, no. 24, 2015.

[119] M. A. Rahman, M. S. Miah, W. Gueaieb, and A. E. Saddik, "Senora: A p2p service-oriented framework for collaborative multirobot sensor networks," *IEEE Sensors Journal*, vol. 7, pp. 658–666, May 2007.

[120] J. Claro, B. Dias, B. Rodrigues, J. P. Pimentão, P. Sousa, and S. Onofre, "Autonomous robot integration in surveillance system: Architecture and communication protocol for systems cooperation," in *Proceedings of the IEEE International Conference on Power Electronics and Motion Control*, 2014.

[121] P. Sauer, T. Hausten, and P. Hofstedt, "Using internet of things technology to create a really platform independent robotics framework," in *Proceedings of the IEEE International Symposium on Systems Engineering (ISSE)*, 2016.

[122] B. Yan, D. Shi, J. Wei, and C. Pan, "Hibot: A generic ros-based robot-remote-control framework," in *Proceedings of the Asia-Pacific Conference on Intelligent Robot Systems (ACIRS)*, 2017.

[123] C. R. Karanam and Y. Mostofi, "3d through-wall imaging with unmanned aerial vehicles using wifi," in *Proceedings of the ACM/IEEE International Conference on Information Processing in Sensor Networks*, 2017.

[124] N. Patwari and J. Wilson, "Rf sensor networks for device-free localization: Measurements, models, and algorithms," *Proceedings of the IEEE*, vol. 98, no. 11, pp. 1961–1973, 2010.

[125] S. Aditya, A. F. Molisch, and H. Behairy, "Bayesian multi-target localization using blocking statistics in multipath environments," in *Proceedings of the IEEE International Conference on Communication Workshop (ICCW)*, 2015.

[126] B. Gulmezoglu, M. B. Guldogan, and S. Gezici, "Multiperson tracking with a network of ultrawideband radar sensors based on gaussian mixture phd filters," *IEEE Sensors Journal*, vol. 15, no. 4, pp. 2227–2237, 2015.

[127] J. Wang and D. Katabi, "Dude, where's my card?: Rfid positioning that works with multipath and non-line of sight," *ACM SIGCOMM Computer Communication Review*, vol. 43, no. 4, pp. 51–62, 2013.

[128] B. Tan, K. Chetty, and K. Jamieson, "Thrumapper: Through-wall building tomography with a single mapping robot," in *Proceedings of the 18th International Workshop on Mobile Computing Systems and Applications*, 2017.

[129] F. Adib, C.-Y. Hsu, H. Mao, D. Katabi, and F. Durand, "Capturing the human figure through a wall," *ACM Transactions on Graphics (TOG)*, vol. 34, no. 6, p. 219, 2015.

[130] F. Adib, Z. Kabelac, D. Katabi, and R. C. Miller, "3d tracking via body radio reflections," in *Proceedings of the USENIX Symposium on Networked Systems Design and Implementation (NSDI)*, 2014.

[131] K. Chetty, G. E. Smith, and K. Woodbridge, "Through-the-wall sensing of personnel using passive bistatic wifi radar at standoff distances," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 50, no. 4, pp. 1218–1226, 2012.

[132] S. Kumar, E. Hamed, D. Katabi, and L. Erran Li, "Lte radio analytics made easy and accessible," *ACM SIGCOMM Computer Communication Review*, vol. 44, no. 4, pp. 211–222, 2014.

[133] P. Ghosh, J. A. Tran, and B. Krishnamachari, "ARREST: A RSSI based approach for mobile sensing and tracking of a moving object," in *Proceedings of the IEEE International Workshop on Wireless Networking and Control for Unmanned Autonomous Vehicles (WiUAV), Globecom*, 2017.

[134] P. Ghosh, J. A. Tran, and B. Krishnamachari, "ARREST: A RSSI based approach for mobile sensing and tracking of a moving object," *IEEE Transactions on Mobile Computing*, vol. Under Revision, 2017.

[135] P. Ghosh, J. A. Tran, and B. Krishnamachari, "ARREST: A rssi based approach for mobile sensing and tracking of a moving object," *arXiv preprint arXiv:1707.05493*, 2017.

[136] V. Kumar, D. Rus, and S. Singh, "Robot and sensor networks for first responders," *IEEE Pervasive Computing*, vol. 3, no. 4, pp. 24–33, 2004.

[137] J. Xiong and K. Jamieson, "Arraytrack: a fine-grained indoor location system," in *Proceedings of the USENIX Symposium on Networked Systems Design and Implementation (NSDI)*, 2013.

[138] J.-R. Jiang, C.-M. Lin, F.-Y. Lin, and S.-T. Huang, "Alrd: Aoa localization with rssi differences of directional antennas for wireless sensor networks," *International Journal of Distributed Sensor Networks*, vol. 2013, 2013.

[139] Openmote, "http://www.openmote.com/."

[140] mbed LPC1768, "https://developer.mbed.org/platforms/."

[141] HC-SRO4, "https://cdn.sparkfun.com/datasheets/Sensors/Proximity/HCSR04.pdf."

[142] RTOS, "http://www.ni.com/white-paper/3938/en/."

[143] E. Gelenbe, J. Labetoulle, and G. Pujolle, "Performance evaluation of the hdlc protocol," *Computer Networks*, vol. 2, no. 4-5, pp. 409–415, 1978.

[144] RIOT OS, "https://riot-os.org/."

[145] E. Baccelli, O. Hahm, M. Gunes, M. Wahlisch, and T. C. Schmidt, "Riot os: Towards an os for the internet of things," in *Proceedings of the IEEE International Conference on Computer Communications (INFOCOM) Workshops*, 2013.

[146] VICON, "https://www.vicon.com/."

[147] A. Savvides, C.-C. Han, and M. B. Strivastava, "Dynamic fine-grained localization in ad-hoc networks of sensors," in *Proceedings of the ACM International conference on Mobile computing and networking (MobiCom)*, 2001.

[148] Pololu-3pi, "https://www.pololu.com/product/975."

[149] MB1300-XL-MaxSonar-AE0, "http://www.maxbotix.com/Ultrasonic_Sensors/MB1300.htm."

[150] P. Ghosh and B. Krishnamachari, "Interference power bound analysis of a network of wireless robots," in *Proceedings of the International Conference on Communication Systems and Networks*, pp. 7–23, Springer, 2017.

[151] Y. Zeng, P. H. Pathak, and P. Mohapatra, "A first look at 802.11 ac in action: energy efficiency and interference characterization," in *Proceedings of IFIP Networking Conference*, 2014.

[152] A. Safak, "Statistical analysis of the power sum of multiple correlated log-normal components," *IEEE Transactions on Vehicular Technology*, vol. 42, no. 1, pp. 58–61, 1993.

[153] M. Hifi and R. M'hallah, "A literature review on circle and sphere packing problems: models and methodologies," *Advances in Operations Research*, vol. 2009, 2009.

[154] P. Ghosh, H. Ren, R. Banirazi, B. Krishnamachari, and E. A. Jonckheere, "Empirical evaluation of the heat-diffusion collection protocol for wireless sensor networks," *Computer Networks*, vol. 127, pp. 217–232, 2017.

[155] M. Mauve, J. Widmer, and H. Hartenstein, "A survey on position-based routing in mobile ad hoc networks," *IEEE Network*, vol. 15, no. 6, pp. 30–39, 2001.

[156] S. Taneja and A. Kush, "A survey of routing protocols in mobile ad hoc networks," *International Journal of Innovation, Management and Technology*, vol. 1, no. 3, p. 279, 2010.

[157] R. Fonseca, O. Gnawali, K. Jamieson, S. Kim, P. Levis, and A. Woo, "The collection tree protocol (ctp)," *TinyOS TEP*, vol. 123, no. 2, 2006.

[158] P. Levis, N. Patel, D. Culler, and S. Shenker, "Trickle: A self-regulating algorithm for code propagation and maintenance in wireless sensor networks," in *Proceedings of the USENIX Symposium on Networked Systems Design and Implementation (NSDI)*, 2004.

[159] S. Farahani, *ZigBee wireless networks and transceivers*. Newnes, 2011.

[160] Tutornet, "`http://anrg.usc.edu/www/tutornet/`."

[161] M. Buettner, G. V. Yee, E. Anderson, and R. Han, "X-mac: a short preamble mac protocol for duty-cycled wireless sensor networks," in *Proceedings of the ACM Conference on Embedded Networked Sensor Systems*, 2006.

[162] P. Ghosh, J. A. Tran, D. Dsouza, N. Ayanian, and B. Krishnamachari, "Romano: A novel overlay lightweight communication protocol for unified control and sensing of a network of robots," *arXiv preprint arXiv:1709.07555*, 2017.

[163] P. Ghosh, J. Xie, and B. Krishnamachari, "miniradar: A low power ieee 802.15. 4 transceiver based implementation of bistatic radar," in *Proceedings of the 4th ACM Workshop on Hot Topics in Wireless (HotWireless), MobiCom*, 2017.

[164] J. Aldrich, "Ra fisher and the making of maximum likelihood 1912-1922," *Statistical Science*, vol. 12, no. 3, pp. 162–176, 1997.