Peer Reviewed

Title:
Context-aware computing for wireless networks

Author:
Ergüt, Salih

Acceptance Date:
2010

Series:
UC San Diego Electronic Theses and Dissertations

Degree:
Ph. D., UC San Diego

Permalink:
http://escholarship.org/uc/item/4v95d28k

Local Identifier:
b6714595

Abstract:
Context-aware computing has been the center of attention in computer science research for many years. Context-aware systems gather contextual data from their sensors, other cooperative nodes or persistent databases and adapt to this information without requiring explicit user intervention. In this thesis we first address the benefits of certain contextual data (such as network connectivity, communication bandwidth, cost of operation, user's location, as well as nearby people and objects) applied to wireless networks. Other important contextual data include social surrounding, environment related conditions, and time context (time of day, month, season, or year). As a result of advancements in technology, the accessing, storing, and incorporating of such massive amounts of data has become a mainstream service. We then develop active and passive localization algorithms for wireless networks. This thesis emphasizes context-aware models for the network layer of wireless and cellular networks rather than classical application layer context-awareness. We first propose a mobile client based active queue management technique which remotely controls the dedicated base station queue size significantly reducing the experienced packet latency. Here, mobile makes use of its knowledge of the underlying cellular technology to enhance the end user experience. We then introduce a packet size aware path setup mechanism for wireless mesh networks where routers benefit from packet size information in computing the optimal route. Both techniques are implemented on real hardware; implementation details and practical considerations are also provided. The second part of this thesis focuses on locating a target in wireless networks. First, we propose a localization algorithm that uses the multipath profile of a mobile device in a cellular network. This algorithm is implemented and evaluated using real data from a commercial cellular network. Finally, we provided linear least squares and neural network techniques for active and passive localization algorithms in a sensor network. The term active localization indicates that the target is active in the localization process, while passive localization refers to locating an uncooperative target.
UNIVERSITY OF CALIFORNIA, SAN DIEGO

Context-aware Computing
for Wireless Networks

A dissertation submitted in partial satisfaction of the
requirements for the degree
Doctor of Philosophy

in

Electrical Engineering
(Communication Theory and Systems)

by

Salih Ergüt

Committee in charge:

Professor Ramesh R. Rao, Chair
Professor Rene L. Cruz
Professor Sujit Dey
Professor Bill Hodgkiss
Professor Geoffrey M. Voelker

2010
The dissertation of Salih Ergüt is approved, and it is acceptable in quality and form for publication on microfilm:

Chair

University of California, San Diego

2010
# TABLE OF CONTENTS

Signature Page .......................................................... iii  
Table of Contents ......................................................... iv  
List of Figures .............................................................. vii  
List of Tables ............................................................... xii  
Acknowledgements ........................................................ xiv  
Vita and Publications ...................................................... xvi  
Abstract of the Dissertation ............................................ xvii  

1 Introduction ............................................................. 1  
   1.1 Context-aware Systems ......................................... 2  
   1.2 Overview of Chapters ........................................... 3  

Bibliography .............................................................. 5  

Part I  Client and Network Based Algorithms to reduce  
Packet Delay in Wireless Networks .................................... 7  
   2 Client Side Active Queue Management for 3G Cellular Networks . 8  
      2.1 Motivation ......................................................... 10  
           2.1.1 Benefits of Using an AQM Scheme at a 3G Link ........ 10  
           2.1.2 TCP Basics .................................................. 11  
           2.1.3 Rate Dependent BS Queue Size .......................... 13  
      2.2 Analytical Model ................................................ 13  
           2.2.1 Single TCP Flow ........................................... 16  
           2.2.2 Multiple TCP Connections ............................... 22  
      2.3 Experimental Results ......................................... 26  
      2.4 Remarks .......................................................... 30  

Bibliography .............................................................. 35  

3 Packet size aware path setup for 802.11 based mesh networks . . . 37  
   3.1 Network Model ..................................................... 41  
      3.1.1 Delay/Throughput Model .................................. 41  
   3.2 Path Setup Optimization ........................................ 46  
      3.2.1 A Sample Implementation .................................. 51
<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.2.2</td>
<td>General Solution</td>
<td>55</td>
</tr>
<tr>
<td>3.3</td>
<td>Remarks</td>
<td>58</td>
</tr>
<tr>
<td></td>
<td>Bibliography</td>
<td>61</td>
</tr>
<tr>
<td>Part II</td>
<td>Localization for Cellular and Sensor Networks</td>
<td>64</td>
</tr>
<tr>
<td>4</td>
<td>Wireless Positioning Overview</td>
<td>65</td>
</tr>
<tr>
<td>4.1</td>
<td>Active positioning</td>
<td>65</td>
</tr>
<tr>
<td>4.1.1</td>
<td>Direct positioning</td>
<td>66</td>
</tr>
<tr>
<td>4.1.2</td>
<td>Two-step positioning</td>
<td>66</td>
</tr>
<tr>
<td>4.2</td>
<td>Passive positioning</td>
<td>67</td>
</tr>
<tr>
<td></td>
<td>Bibliography</td>
<td>68</td>
</tr>
<tr>
<td>5</td>
<td>Localization via Multipath Strengths in a CDMA2000 Cellular Network</td>
<td>69</td>
</tr>
<tr>
<td>5.1</td>
<td>Cellular Network Overview</td>
<td>71</td>
</tr>
<tr>
<td>5.1.1</td>
<td>Propagation Model</td>
<td>72</td>
</tr>
<tr>
<td>5.2</td>
<td>Neural Network Model</td>
<td>72</td>
</tr>
<tr>
<td>5.3</td>
<td>Methodology</td>
<td>73</td>
</tr>
<tr>
<td>5.4</td>
<td>Experimental Results</td>
<td>74</td>
</tr>
<tr>
<td>5.4.1</td>
<td>Setup</td>
<td>74</td>
</tr>
<tr>
<td>5.4.2</td>
<td>Results</td>
<td>75</td>
</tr>
<tr>
<td>5.5</td>
<td>Remarks</td>
<td>77</td>
</tr>
<tr>
<td></td>
<td>Bibliography</td>
<td>80</td>
</tr>
<tr>
<td>6</td>
<td>Linear Least Squares Techniques for Active and Passive Localization</td>
<td>82</td>
</tr>
<tr>
<td>6.1</td>
<td>Active Localization</td>
<td>82</td>
</tr>
<tr>
<td>6.1.1</td>
<td>System Model</td>
<td>83</td>
</tr>
<tr>
<td>6.1.2</td>
<td>Nonlinear LS Estimation</td>
<td>85</td>
</tr>
<tr>
<td>6.1.3</td>
<td>Linear LS Estimation</td>
<td>87</td>
</tr>
<tr>
<td>6.1.4</td>
<td>Simulation Results</td>
<td>101</td>
</tr>
<tr>
<td>6.1.5</td>
<td>Complexity Analysis</td>
<td>104</td>
</tr>
<tr>
<td>6.2</td>
<td>Passive Localization</td>
<td>106</td>
</tr>
<tr>
<td>6.2.1</td>
<td>System Model</td>
<td>108</td>
</tr>
<tr>
<td>6.2.2</td>
<td>Nonlinear LS Estimation</td>
<td>108</td>
</tr>
<tr>
<td>6.2.3</td>
<td>Linear LS Estimation for independent noise</td>
<td>111</td>
</tr>
<tr>
<td>6.2.4</td>
<td>Linear LS Estimation for correlated noise</td>
<td>117</td>
</tr>
<tr>
<td>6.3</td>
<td>Remarks</td>
<td>121</td>
</tr>
<tr>
<td></td>
<td>Bibliography</td>
<td>123</td>
</tr>
</tbody>
</table>
Localization via TDOA in a UWB sensor network using Neural Networks

### 7.1 Related Work

A framework for detecting passive targets

### 7.3 Simulation Setup

### 7.4 Neural Network Model

### 7.5 Single Object Detection

#### 7.5.1 Simulation Results

#### 7.5.2 Evaluating the performance of NN algorithm

### 7.6 Multiple Object Detection

#### 7.6.1 An example: Detecting two objects

### 7.7 Remarks

---

**Part III Concluding Remarks**

### 8 Conclusion

---

Bibliography
LIST OF FIGURES

Figure 2.1: Slow start and congestion avoidance phases of a TCP flow. 11

Figure 2.2: Implications of using a fixed vs. rate dependent buffer size at the base station. At any time for a TCP flow, maximum number of packets in transit is upper bounded by the sum of the delay bandwidth product of the bottleneck wireless link and the queue size at the base station. In the steady state, window size fluctuates between this maximum value and half of it. $C_H$ and $C_L$ stand for the high and low rate links, respectively. 14

Figure 2.2: (Continued) Darker red filling corresponds to minimum window size level at any time and lighter filling to the average window size. When minimum window size drops below the bandwidth delay product (or the blue dashed line), link is underutilized because the link can accommodate more packets than available in transit. The length of the green arrow is proportional to the average queuing delay, $\bar{q}$, where $\bar{q}$ is the average queue length, and $C_x$ is the link rate. Propagation delay is not compared since it is same for all cases. The rate dependent buffer size in part (c) reaps benefits of small and large fixed buffer sizes in parts (a) and (b), respectively. 15

Figure 2.3: At steady state, window size $W(t)$ of TCP server fluctuates between $A$ and $2A$ packets with a period of $T = t_{A+1} - t_0$. $\Delta_i$ for $i$ integer is the interval between two successive change of window size in terms of packets instances. 17

Figure 2.4: At steady state, aggregate window size in packets, $W(t)$, of $N$ synchronized TCP flows fluctuates between $A$ and $2A$ packets with a period of $T = t_{A/N+1} - t_0$. $\Delta_i$ for $i$ integer is the interval between two successive change of window size in terms of packets instances. $t_i$, $i$ integer, is the time instance when $W(t)$ is incremented Aggregate window size, $W_N(t)$, for $N$ synchronized simultaneous flows. All flows increment and reduce their window size at the same time. $W_N(t)$ repeats itself with a period of $T' = t_{A/N+1} - t_0$. 22

Figure 2.5: Instead of dropping packets from all flows at the same time, packet drops are distributed within the period of TCP window. Aggregate window size (shown in red) fluctuates between $w_{max}'$ and $w_{max}' - A$. Shift duration $\phi$ is equal to $T/N$. 23

Figure 2.6: Experimental setup. Commercial CDMA2000 1xRTT cellular network was used. Fader and noise generator provided emulating similar channel conditions when comparing R-AQM with mere drop-tail queues. 27
Figure 2.7: Some of the components of the CDMA2000 1xRTT cellular network that was used in the experiments: BSC, RBS, PDSN, and MSC. PDSN is the gateway that connects the core network to the Internet. Telos switch was used as the Mobile Switching Center (MSC).

Figure 2.8: Comparing BSC queue size and RTT for drop-tail and RAQM algorithms when forward link rate is 86kbps. Note that R-AQM scheme achieves a mean RTT of 0.9 sec with a standard deviation of 0.2 sec, while drop-tail has a RTT of 2.0 sec on the average with a standard deviation of 0.4 sec. Both links achieve high utilization since BSC queue almost never drains completely.

Figure 2.9: Comparison of average number of TCP flows vs. BSC queue sizes for drop-tail and R-AQM algorithms for different forward link rates. Larger queue sizes cause larger packet delays.

Figure 2.10: Comparison of average number of TCP flows vs. RTT for drop-tail and R-AQM algorithms for different forward link rates. R-AQM only performs slightly worse than drop-tail for forward link rate of 163 kbps. Because BSC queue size is optimized for this highest rate for the drop-tail algorithm.

Figure 2.11: Comparison of average number of TCP flows vs. utilization for drop-tail and R-AQM algorithms for different forward link rates. Both algorithms utilize the link better when there are two or more flows.

Figure 2.12: Comparison of RAQM and drop-tail only schemes. A large file was being downloaded while the channel conditions changed. a) Top subfigure plots downlink data rate as a function of time. Since experimental, it was not possible to perfect match the channel conditions. b) Middle subplot plots the queue lengths at BSC. c) Bottom subfigure plots end-to-end delays.

Figure 3.1: Model vs. estimated throughputs for single-hop UDP flows for various packet sizes. Transmission rates 1, 2, 5.5, and 11 Mb/s are allowed physical transmission rates for 802.11b standard. Data rates indicate estimated average UDP throughputs from measurements and model values. Mismatch between the model and the measurements increases for higher rates and smaller packet sizes. Data is from Table 3.1.

Figure 3.2: Delay comparison of a 1 Mb/s single-hop link with three-hop alternatives.

Figure 3.3: Delay comparison of 2 Mb/s and three-hop alternatives.

Figure 3.4: Delay comparison of 2 Mb/s and two-hop alternatives.

Figure 3.5: Delay comparison of 1 Mb/s and multi-hop alternatives.
Figure 3.6: Hardware for implementation. A Soekris 4521 [10] boards with Prism 2.5 chipset based 802.11b wireless cards are seen in the figure. A trimmed version of the Debian Linux with Host-AP driver was used in the implementation.

Figure 3.7: Experiments were conducted to measure average throughputs by explicitly setting various transmission rates for the above configurations. **PSA module** resides on the source and destination routers.

Figure 5.1: Feed forward back propagation neural network architecture.

Figure 5.2: System Architecture.

Figure 5.3: Performance of the algorithm: NN estimates on the left and averaged output on the right.

Figure 5.4: CDF of estimation error for different regions.

Figure 5.5: Effect of averaging the estimated location. Overall CDF of estimation errors from all regions are plotted. **Averaged Output** corresponds to averaging the position estimation over 15 samples.

Figure 5.6: Drive test was conducted in San Diego, CA. Whole area is divided into 3 regions.

Figure 6.1: **TOA based localization.** Nodes $s_1$, $s_2$, and $s_3$ communicate with target $t$ and measure the distances $r_1$, $r_2$, and $r_3$ which allows them locate the target on a circle. However due to noise these circles intersect at multiple locations. Light color bands around these circles indicate the uncertainty due to noise.

Figure 6.2: *Simulation setup.* Sensor nodes $s_1$, $s_2$, $s_3$, and $s_4$ are placed at the corners of a $100m \times 100m$ rectangular area. Location estimations from TOA measurements are performed at 10000 points on the grid inside this region and error is recorded. At each point simulation is repeated 1000 times to obtain average error.

Figure 6.3: Overall average RMSE (in m) of TOA based localization techniques and CRLB vs. standard deviation of TOA measurement, $\sigma_1$, over the simulation region given in Figure 6.2. Standard deviations of TOA measurements for different nodes are modeled as i.i.d. L-LS-M and L-LS-A both performs closer to CRLB and better than L-LS-1 and L-LS-3.

Figure 6.4: RMSE of TOA based localization techniques and CRLB over the simulation region given in Figure 6.2 when standard deviation of TOA measurement, $\sigma_1$, is 6m. Standard deviations of TOA measurements for different nodes are modeled as i.i.d.
Figure 6.5: Multipath distance estimation. Sensor node $s_m$ transmits a signal. Node $s_n$ receives two replicas of the same signal, one from direct path and one reflected from the target $t$. $d_{m,n}$ is known since the locations of the sensor nodes are known apriori. Multipath distance, $d_{m,t} + d_{n,t}$, is to be estimated.

Figure 6.6: TDOA based localization. Each sensor node pair can locate a target on an ellipse using the multipath distance. Uncertainty in the multipath distances due to measurement noise is displayed by a band.

Figure 6.7: Simulation setup. Nodes $s_1$, $s_2$, $s_3$, and $s_4$ are placed at the corners of a 100m x 100m rectangular area. These nodes communicate with each other and estimate the location of a target from the reception of delayed signals reflected from the target. Location estimations using this TDOA information are done at 10000 points inside the region and error is recorded. At each point experiment is repeated 1000 times to obtain average error.

Figure 6.8: Variance coefficient $K$ vs. number of sensor nodes $N$ when all nodes communicate with each other.

Figure 6.9: RMSE (in m) of TDOA based passive localization for various LS techniques and CRLB over the simulation region. Multipath distance estimation error is considered i.i.d. with a standard deviation of 6m.

Figure 6.10: RMSE (in m) vs. standard deviation of distance estimation error of TDOA measurements for various LS techniques and CRLB.

Figure 6.11: Comparison of RMSE of the target position estimation error for correlated and independent error models using L-LS-A algorithm. $x$-axis indicates the standard deviation of overall multipath distance estimation error.

Figure 7.1: Effects of external objects on the multipath profile of UWB signals. a) Multipath profile when there is not any object in the medium, b) Multipath profile is modified due to reflecting signals coming from the external object.

Figure 7.2: Using the multipath distance target object is located on an ellipse for each sensor pair. Target object is positioned at the intersection of all these ellipses.

Figure 7.3: Feed forward back propagation neural network architecture.

Figure 7.4: CDF of error with increasing number of sensors.

Figure 7.5: Comparing NN with Cramer-Rao and least squares estimates.
Two phase detector. In the first phase a NNBA or any linear LS technique can be used to estimate target location with an associated cost for each estimation. In the second step the decision box estimates $L$ target locations by grouping smartly and using the cost of the estimations.

Location estimator. A set of multipath distance measurements are fed to NNBA. NNBA computes an estimate $(x_i, y_i)$ for the target location. New multipath distance estimates corresponding to this target estimate $(x_i, y_i)$ and known sensor node locations $(s_i, i = 1, \ldots, N)$ are computed. Square of the difference between new multipath distance estimates and input multipath distance measurements are summed to generate a cost metric $C_i$.

Performance of detecting two objects is comparable to the performance of detecting single object.
LIST OF TABLES

2.1 Test results for R-AQM algorithm. 'BSC Q' is the queue length at the BSC, 'RTT' is the average round-trip-delay, 'Utilization' is the percent of the time BSC buffers are non-empty when the wireless link is active, and $N$ is the number of active connections for the same user. ........................................... 28

2.2 Test results for drop-tail algorithm. 'BSC Q' is the queue length at the BSC, 'RTT' is the average RTT, 'Utilization' is the percent of the time BSC buffers are non-empty when the wireless link is active, and $N$ is the number of active connections for the same user. ........ 28

3.1 Throughput estimations from measurements using IPERF and model values for single-hop links. Deviations are with respect to UDP estimations. ........................................... 43

3.2 Throughput measurements and estimations for 2-hop links. Deviations are with respect to UDP measurements. Packet sizes are greater than or equal to 500 bytes. ......................... 47

3.3 Throughput measurements and estimations for 2-hop links. Deviations are with respect to UDP measurements. Packet sizes are less than or equal to 500 bytes. ......................... 48

3.4 Throughput measurements and estimations for three-hop links. Deviations are with respect to UDP measurements. ......................... 49

3.5 Throughput for 1-hop and 3-hop alternatives for various packet sizes and performance improvement when the module is used 1-hop link is $2 \text{ Mb/s}$ and all the links on the three hop path is $11 \text{ Mb/s}$. .................. 55

3.6 Average throughputs when only 1-hop link is used, only 3-hop link is used, or either link is adaptively selected by the packet-size aware (PSA) module. 1-hop link is $1 \text{ Mb/s}$ and transmission rates on 3-hop link are 2, 11, $11 \text{ Mb/s}$. Results for UDP flows with various packet sizes and TCP flows with an MTU of 1500 bytes are listed. PSA module selects the link with highest throughput for UDP traffic, therefore selects the best of the alternatives. As for TCP traffic, however, PSA module routes data and acknowledgements into different links achieving overall data throughput greater than any of the links. ......................... 56
3.7 Average throughputs when only 1-hop link is used, only 2-hop link is used, or either link is adaptively selected by the packet-size aware (PSA) module. 1-hop link is 2 $Mb/s$ and transmission rates on 2-hop link are both 5.5 $Mb/s$. Results for UDP flows with various packet sizes and TCP flows with an MTU of 1500 bytes are listed. *PSA module* selects the link with highest throughput for UDP traffic, therefore selects the best of the alternatives. As for TCP traffic, however, *PSA module* routes data and acknowledgements into different links achieving overall data throughput greater than any of the links. .......................................................... 57

5.1 Localization accuracy of NN and smoother .................................. 77

6.1 LS-matrix $T_B$ and variance coefficient $K$ for various number of sensor nodes ($N = 3, 4,$ and 5). .................................................. 116

7.1 The basis preamble symbol set .................................................. 129
ACKNOWLEDGEMENTS

I would like to express my deepest gratitude to my committee chair, Professor Ramesh R. Rao, whose continuous guidance, encouragement, and support enabled me developing this thesis.

I would like to thank to my committee members, Professor Rene Cruz, Professor Sujit Dey, Professor Bill Hodgkiss, and Professor Geoffrey M. Voelker for their invaluable feedback.

I would like to show my gratitude to my wife, my parents, and my sister for their love and constant support. I am indebted to my close friends and my cousin for being there for me whenever I needed emotional and material support.

I am thankful for the support of Ericsson and California Institute For Telecommunications and Information Technology throughout the development of my thesis.

Chapter 2, in full, is a reprint of the material as it appears in proceedings of IEEE Consumer Communications & Networking Conference 2006. Akin, Ozdemir; Ergüüt, Salih; Rao, Ramesh R. The dissertation author and Mr. Akin were primary investigators and authors of this paper with equal contributions.

Chapter 3, in full, is a reprint of the material as it appears in proceedings of IEEE Consumer Communications & Networking Conference 2008. Arisoylu, Mustafa; Ergüüt, Salih; Cruz, Rene L.; Rao, Ramesh R. The dissertation author and Mr. Arisoylu were primary investigators and authors of this paper with equal contributions.

Chapter 5, in full, is a reprint of the material as it appears in proceedings of IEEE World Conference on Computational Intelligence 2008. Ergüüt, Salih; Rao, Ramesh R.; Dural, Özgüür. The dissertation author was the primary investigator and author of this paper.

Chapter 6, in part, has been submitted for publication of the material as it may appear in proceedings of IEEE Personal, Indoor and Mobile Radio 2010. Ergüüt, Salih; Rao, Ramesh R. The dissertation author was the primary investigator and author of this paper.

Chapter 7, in full, is a reprint of the material as it appears in proceedings
of IEEE International Conference on Communications 2008. Ergüt, Salih; Rao, Ramesh R.; Dural, Ö zgür; Sahinoglu, Zafer. The dissertation author was the primary investigator and author of this paper.
VITA


1994–1998  B.S., Electrical and Electronics Engineering, Bilkent University, Ankara, TURKEY

1998–2000  M.S., Electrical and Computer Engineering, Northeastern University, Boston, MA

2000–2001  DSP Engineer, Aware, Inc., Boston, MA

2001–2006  Senior Engineer, Ericsson, Inc., San Diego, CA

2007/2008  Summer Intern, MAC Systems, Nexwave Broadband, Inc., San Diego, CA

2002–2010  Ph.D., Electrical and Computer Engineering, University of California, San Diego

PUBLICATIONS


Salih Ergüt, Ramesh R. Rao and Özgür Dural *Locating mobile users via multipath pilot strengths using Neural Networks*, IEEE World Conference on Computational Intelligence (WCCI 2008), Hong Kong, China, 2008.


Naciye Hardalac, Nevhiz Ercan, Firat Hardalac and Salih Ergüt *Classification of educational backgrounds of students using musical intelligence and perception with the help of artificial neural networks*, Frontiers in Education Conference (FIE 2006), San Diego, CA, 2006
ABSTRACT OF THE DISSERTATION

Context-aware Computing for Wireless Networks

by
Salih Ergüüt
Doctor of Philosophy in Electrical Engineering
(Communication Theory and Systems)
University of California San Diego, 2010
Professor Ramesh R. Rao, Chair

Context-aware computing has been the center of attention in computer science research for many years. Context-aware systems gather contextual data from their sensors, other cooperative nodes or persistent databases and adapt to this information without requiring explicit user intervention. In this thesis we first address the benefits of certain contextual data (such as network connectivity, communication bandwidth, cost of operation, user’s location, as well as nearby people and objects) applied to wireless networks. Other important contextual data include social surrounding, environment related conditions, and time context (time of day, month, season, or year). As a result of advancements in technology, the accessing, storing, and incorporating of such massive amounts of data has become a mainstream service. We then develop active and passive localization algorithms for wireless networks.

This thesis emphasizes context-aware models for the network layer of wireless and cellular networks rather than classical application layer context-awareness. We first propose a mobile client based active queue management technique which remotely controls the dedicated base station queue size significantly reducing the experienced packet latency. Here, mobile makes use of its knowledge of the underlying cellular technology to enhance the end user experience. We then introduce a
packet size aware path setup mechanism for wireless mesh networks where routers benefit from packet size information in computing the optimal route. Both techniques are implemented on real hardware; implementation details and practical considerations are also provided.

The second part of this thesis focuses on locating a target in wireless networks. First, we propose a localization algorithm that uses the multipath profile of a mobile device in a cellular network. This algorithm is implemented and evaluated using real data from a commercial cellular network. Finally, we provided linear least squares and neural network techniques for active and passive localization algorithms in a sensor network. The term active localization indicates that the target is active in the localization process, while passive localization refers to locating an uncooperative target.
Introduction

In 1981, Saltzer et. al. introduced the *end-to-end system design principle* \[12\] and argued that implementing functions on the low level (i.e. network level,) would be costly and possibly redundant as these functionalities may be implemented at the end terminals. The end-to-end principle has been advocated by many and has promoted smart terminals and dumb networks. As Gilder stated “In a world of dumb terminals and telephones, networks had to be smart. But in a world of smart terminals, networks have to be dumb.” in 1992 \[9\].

Internet protocols were designed with the wired network topology in mind. Applying these protocols to wireless devices without any modification resulted in inefficient use of network resources and overall performance degradation. This issue became significant with the rapid growth of wireless and cellular networks.

TCP, a reliable transport protocol, is an example of poor performance of Internet protocols under wireless networks. TCP perceives any packet loss as an indication of congestion in the network, consequently reducing its data rate. For wireless networks, however, packet loss is very common, and such unnecessary rate reductions lead to waste of system resources. Recently the end-to-end design paradigm has been called into question as it failed to completely address the needs of modern networks. While some server based solutions continued to follow the end-to-end principle \[11\], others proposed splitting the connection and applying remedies inside the network \[4, 3\].

Supporting the diversion from the end-to-end principle, Blumenthal et. al.
argued that some of the application layer requirements in a network may not be implemented in the end-points but only fulfilled inside the network [6].

With advances in technology, smaller yet more powerful mobile devices are becoming affordable and highly available in the market. Today, worldwide mobile phone subscriptions exceeded 4.6 billion [10]. Infrastructures are also following a similar trend: high processing power at a reduced cost. Powerful network components and end terminals facilitated the popularity and widespread deployments of context-aware systems.

1.1 Context-aware Systems

Context-aware systems aim to adapt seamlessly to the available contextual data without requiring explicit user intervention. First introduced in [13], context-aware devices suggest acquiring contextual data using sensors or through some persistent databases to modify their behavior. Schilit et. al. considered contextual data to include parameters such as network connectivity, cost of operation, communication bandwidth, nearby people, objects and resources, user’s location, social surrounding, and environment related conditions [13]. [7] extends the scope of contextual data by introducing time context (e.g. time of a day, month, year) and classifies context-awareness into active and passive awareness. The former implies automatic adaptation to the acquired context by triggering some actions. The latter involves storing or transferring context to interested parties, usually to be retrieved at a later time. Due to increased interest in accessing location and activity patterns of mobile users, context-awareness has been largely studied in location estimation and activity recognition related research fields.

[7] provides a survey of some context-aware applications. For example, Active Badge location system in [15] routes telephone calls to the nearest phone by using user’s location and teleporting system [5] allows users desktop computing environment to follow him to the nearest workstation. [14] introduced Mobisaic which extends the web by providing contextual information such as current location to the documents. Web browsers in today’s smart phones already use location
information in returning better search results. [2] suggested using active tags to locate shoppers in a store with the intention to identify their shopping preferences. Thus they can better market its products and assist shoppers in finding goods and deals across the store. Location is also used in many tourist guide applications [1, 8]. For cellular networks, Ortiva Wireless markets a product that adapts media streaming quality to changes in user’s link quality to provide a better user experience while reducing the load on the service providers network [16]. All impose context-awareness at the application layer, except for [16], which contains elements of context-awareness in the network layer.

1.2 Overview of Chapters

Thesis is divided into two parts.

Part I focuses on using contextual data in the network layer of cellular and 802.11 based mesh networks to reduce packet delays. Chapter 2 advocates smart end-terminals enhancing user experience without changing the network infrastructure, while Chapter 3 promotes smart network components to achieve better network throughput without requiring modifications to the end-terminals.

Chapter 2 proposes a mobile client based remote active queue management (AQM) scheme that reduces large TCP delays on a 3G cellular network. The algorithm is implemented and evaluated on a commercial CDMA2000 cellular network. Instead of treating the wireless link as a mere data pipe, the mobile client takes the lead to reduce delays for sensitive applications by controlling the buffering in the network.

Chapter 3 suggests using packet size in routing decisions for wireless mesh networks. In this setup, the “dumb network” is replaced by a smarter one that takes packet size into account. We show that system throughput is increased as a result of reduced transmission delays. We implemented the proposed algorithm and evaluated its performance for various scenarios.

Localization is the most widely applied contextual information for the context-aware systems. Part II of the thesis provides an overview and discusses localization
algorithms for cellular and sensor networks.

Chapter 5 studies localization of a mobile from its multipath signal strengths using neural networks. The proposed localization algorithm is implemented in a commercial CDMA2000 cellular network.

Chapter 6 introduces two novel linear least squares (LS) techniques for wireless localization that use the time of arrival (TOA) and the time difference of arrival (TDOA) measurements. This chapter also provides comparative performance analysis and simulation results for these algorithms.

Finally, Chapter 7 discusses TDOA based passive localization of targets in an ultra-wideband (UWB) sensor network to detect nearby people or objects. We introduced a novel multiple target detection algorithm and evaluated its performance using simulations.
Bibliography


Part I

Client and Network Based Algorithms to reduce Packet Delay in Wireless Networks
In recent years increasing demand on wireless packet networks augmented the interest of both industry and academia in wired-cum-wireless networks. A typical example of a wired-cum-wireless network is where the end-user is connected to the wired packet network via a cellular or a Wi-Fi link.

Although wired and wireless links can be used together in a network, it is important to note that these links have very different characteristics. As opposed to wired links, wireless links generally have lower capacity, experience much higher bit error rates (BER) and longer delays due to fading, shadowing, power limitations, handoffs, etc. In [3, 5] it has been shown that this variability in wireless link characteristics may adversely affect the performance of protocols such as TCP which are designed for wired links in mind with the assumption of low error rate and static link capacity.

Many solutions are suggested for improving the performance of TCP over wireless links. A comparison of these techniques can be found in [2]. [9] is a more recent survey about the problems and solutions of TCP over wireless links.

In this chapter we focus on improving the performance of TCP flows on a 3G cellular networks, such as CDMA2000 or WCDMA, by using a new active queue
management (AQM) technique. We implement our AQM scheme at the mobile host (MH) to regulate the queue length at the base station (BS), rather than implementing it at the BS. We call this new technique as Remote-AQM (R-AQM) as we regulate the per user queue length at the BS remotely from the MH. We use BS in a loose sense referring as the node between MH and the wired network.

[7, 6, 10] propose using AQM schemes implemented at BS in a 3G environment to address extended packet delays. Therefore, the proposed solution is viable only if access to BS is feasible. In [7] an AQM scheme is presented for a single TCP connection where packets are dropped deterministically. Via simulations it is shown that although this new algorithm outperforms Random Early Drop (RED) [4] and drop-tail schemes in a single TCP connection scenario, its performance degrades when the number of TCP connections increases. In [6] an adaptive variation of the same scheme that modifies its parameters according to the link state is proposed. In [10] a similar approach to [6] is used but instead of dropping packets in a deterministic manner, packets are marked using ECN in a probabilistic fashion. This scheme also considers a single TCP connection.

For low bandwidth links, [8] proposes a receiver based active queue management algorithm which uses the advertised window size of a TCP flow to regulate its rate. Their algorithm assumes a constant link speed such as connections over dial-up modems and setting of certain parameters requires user intervention.

In the intermediate nodes of a network, such as routers and switches, AQM schemes are used to prevent congestion and regulate the queue length by dropping packets in a proactive manner. This would eventually cause protocols such as TCP to decrease their sending rate [4, 1].

We propose a new AQM scheme in a 3G environment to regulate the queue length at the BS. The objective is to maintain the BS buffer occupation at a steady small value to achieve low per-packet delay and jitter, while maintaining high link utilization. Throughput remains same as a result of high link utilization.

*Link utilization* is defined as percent of the time BS queue is non-empty while radio resources are allocated to the mobile. Lack of packets in the BS queue due to transport layer protocols back-off causes wasting of radio resources. For
example, TCP may interpret packet losses or higher RTT delays as congested link hence reduce its window size. This will reduce packets in transmission. Radio resource allocation may not be synchronized with upper layer protocols and hence introduce inefficiency.

Our contribution is three fold. First, we implemented our algorithm at the client node (i.e., MH) like [8] as opposed to BS or the server. This property makes our approach easy to deploy since it does not require any modifications to the service provider equipment or the TCP protocol stack. Second, our scheme can handle multiple simultaneous TCP flows as opposed to [7, 6, 10]. Finally, we used a commercial CDMA2000 1xRTT network to evaluate the performance of our algorithm rather than a simulator.

Section 2.1 discusses the benefits of using an AQM scheme for 3G links and motivates an active management of BS queue, Section 2.2 establishes analytical models for single and multiple TCP connections and proposes the remote active queue management (R-AQM) algorithm, Section 2.3 presents experimental results of the R-AQM implementation, and Section 2.4 concludes the chapter.

2.1 Motivation

2.1.1 Benefits of Using an AQM Scheme at a 3G Link

Benefits of having a small average queue length at the BS are discussed in [7]. We will briefly mention them here as it is the motivation behind using our AQM algorithm.

1. Performance of delay sensitive applications such as telnet, streaming audio and video, VoIP is adversely affected by large round-trip times due to long queue lengths. Also with high queuing delays the retransmission time-out timer of TCP increases, thus the time required from recovering from a packet loss increases.

2. Long queue length causes new flows to get an unfairly low bandwidth share and SYN packets to time-out.
Figure 2.1: Slow start and congestion avoidance phases of a TCP flow.

3. During web browsing, users may cancel their existing webpage request before the page is fully downloaded and proceed with a new page. If the queue length is high, there may be many packets in the queue from the initial request that would still be transmitted over the air unnecessarily, hence waste radio resources, and delay the viewing of the second webpage.

4. Short queue length benefits the mobile host (MH) during handoff since state of connection at the BS, including the packets in the queue, must be forwarded to the new BS. As a result, time to complete handoff will be increased.

In a 3G network, BS allocates a dedicated buffer for each MH. Therefore MH can employ an AQM scheme to regulate BS queue length remotely since its actions can be directly effective. Details of the remote AQM (R-AQM) is elaborated in the following section.

2.1.2 TCP Basics

TCP uses a sliding window protocol that allows multiple simultaneous unacknowledged packets to be in transmission. Window size is set to be the smallest of two parameters: advertised window size of the client ($awnd$) and congestion window of the server ($cwnd$). With the advertised window field in the TCP header,
client reports maximum number of bytes that it is willing to receive. Server, on the other hand, keeps track of congestion window, updated with acknowledgements and packet losses, and only allows outstanding number of bytes equal to the minimum of the received advertised window and local congestion window without receiving an acknowledgement.

TCP server maintains another variable, slow start threshold size \((ssthresh)\), to avoid congestion. \(ssthresh\) is initialized to a large value at the beginning and it is set to half of window size with the detection of packet loss. TCP is considered in slow-start for \(cwnd \leq ssthresh\), and congestion avoidance otherwise.

During slow start congestion window grows exponentially, incremented by one segment every time and ACK received. In the congestion avoidance phase window size is incremented by one segment with the reception of number of ACKs equal to the window size. More details can be found in RFC 2001\(^1\), TCP Slow Start, Congestion Avoidance, Fast Retransmit, and Fast Recovery Algorithms. In the steady state TCP exhibits a well-known saw-tooth behaviour.

For a TCP connection we assume that the server’s congestion window is always smaller than the receiver’s advertised window, and therefore window size is equal to congestion window. Advertised window size would be small if the client is altering it to remotely control the server’s rate or the local buffers are insufficient to handle the incoming traffic. Since our client is not manipulating the advertised window size and has a large buffer, this assumption is valid in this case.

Traditionally, routers notify network congestion by dropping packets. RFC 3168\(^2\), The Addition of Explicit Congestion Notification (ECN) to IP, however, extended TCP and IP protocol by providing mechanisms to signal network congestion without resorting to packet drops. We used ECN to reduce the incoming TCP rate remotely.

2.1.3 Rate Dependent BS Queue Size

Figure 2.2 explains the motivation for a rate dependent queue size for a simplified single TCP flow. TCP is considered to operate in the congestion avoidance phase where packet drops due to buffer overflow cause the window size to drop to half. Link is underutilized when TCP window size drops below the bandwidth delay product (displayed with blue dashed line) because it indicates that the channel can handle more packets than that are in transit.

Figure 2.2a shows the TCP window size under a large fixed buffer. Both high and low data rates fully utilize the link. Low data rate channels experience higher packet delays since it takes longer to drain the average queue length ($\bar{q}$). This type of queue sizes, optimized for the higher rate, allows channel to achieve its capacity and are, therefore, preferred in real deployments.

Figure 2.2b shows how the higher rate suffers from small fixed buffer. Both data rates experience a small queuing delay, however, high data rate underutilizes the link. Note that TCP periods are also reduced since maximum window size is smaller compared to Figure 2.2a.

Figure 2.2c displays the benefits of using a variable buffer whose size depends on the channel data rate; i.e. larger queue size for higher rate and smaller queue size for lower rate. Both data rates achieve small queueing delays while achieving full utilization. A variable queue size can be implemented either directly at BS or remotely at MH. In this study, we promote a client based remote management of BS queue at MH.

2.2 Analytical Model

We first start with a deterministic model for a single TCP connection similar to the one used in [7, 6] and then generalize this model for multiple TCP connections.

In our analysis we assume a network topology where the client is connected to the wired network through a 3G BS and the wireless link is the bottleneck. The capacity of the forward wireless links is denoted by $C$ (packets/seconds).
(a) *Large buffer for all data rates.* The high rate link (on the left) experiences small queuing delay and the low rate link (on the right) large queueing delay. Both cases fully utilize the channel.

(b) *Small buffer for all data rates.* Both links experience small queueing delays. However, the high rate link (on the left) underutilizes the channel.

Figure 2.2: *Implications of using a fixed vs. rate dependent buffer size at the base station.* At any time for a TCP flow, maximum number of packets in transit is upper bounded by the sum of the delay bandwidth product of the bottleneck wireless link and the queue size at the base station. In the steady state, window size fluctuates between this maximum value and half of it. $C_H$ and $C_L$ stand for the high and low rate links, respectively.
(c) Rate dependent buffer size. Both links experience small queueing delays while fully utilizing the link.

Figure 2.2: (Continued) Darker red filling corresponds to minimum window size level at any time and lighter filling to the average window size. When minimum window size drops below the bandwidth delay product (or the blue dashed line), link is underutilized because the link can accommodate more packets than available in transit. The length of the green arrow is proportional to the average queueing delay, $\frac{\bar{q}}{C_L}$, where $\bar{q}$ is the average queue length, and $C_L$ is the link rate. Propagation delay is not compared since it is same for all cases. The rate dependent buffer size in part (c) reaps benefits of small and large fixed buffer sizes in parts (a) and (b), respectively.
The round-trip propagation delays of wireless and wire-line (or fixed network) are respectively denoted by \(d_w\) and \(d_f\), where the total round-trip propagation delay is given by

\[d_0 = d_w + d_f\]

We also assume that \(d_f\) is small compared to \(d_w\), therefore \(d_0 \approx d_w\). The round trip delay \(r\) of a packet in the network is given by

\[r = d_0 + q/C\] (2.1)

where \(q\) is the queue length in packets.

Link layer retransmissions at BS keep the packet loss at the wireless link very low hence we neglect the packet drops both at wired and wireless links. The only packet loss is assumed to be due to buffer overflow at the BS.

In the following sections, for multiple TCP flows we will compute the optimal BS queue size as a function of rate. We will also discuss estimating the corresponding round trip time at MH using (2.1).

### 2.2.1 Single TCP Flow

In [7] large delays due to excessive queueing is addressed at the BS and they propose dropping a single packet at BS when its queue size reaches a limit value \((q_{max})\) to slow down the rate of the TCP connection.

Multiple packet losses from a single sending window is prevented since there is a delay in TCP server’s respond to packet loss and also TCP does not recover well from successive packet losses. To achieve this, packet dropping is not allowed for a specific period, in terms of specified number of packets or bytes, following a recent drop.

The resulting steady state window size behavior is similar to the one shown in Figure 2.3.

**Proposition 1.** When multiple synchronized TCP flows traverses a link with a bottleneck last hop, to maintain full utilization of the last hop maximum queue
Figure 2.3: At steady state, window size \( W(t) \) of TCP server fluctuates between \( A \) and \( 2A \) packets with a period of \( T = t_{A+1} - t_0 \). \( \Delta_i \) for \( i \) integer is the interval between two successive change of window size in terms of packets instances.
size, \( q_{\text{max}} \), must be equal to or greater than the bandwidth-delay product or \( C \times d \), where \( C \) is the bandwidth and \( d \) is the delay experienced by packets. [7]

**Proof.** This can easily be proven by equating the number of packets sent in interval \([t_0, t_{A+1}]\) to \( C(t_{A+1} - t_0) \).

In Figure 2.4, let \( \Delta_i = t_{i+1} - t_i \) for some integer \( i \). Note that these intervals are not necessarily equal to one another but \( \Delta_i = \Delta_{i+A+1} \), since the window size is periodic with a period of \( T = t_{A+1} - t_0 \). At \( t_{A+1} \) the window size is halved and therefore during interval \( \Delta_{A+1} \) no packets are sent by the server as there are already \( 2NA \) unacknowledged packets in transit. Server has to receive acknowledgments for \( NA \) packets before it can send any new packets.

Let congestion window, \( W(t_{k^*}) \), be equal to the bandwidth delay product at interval \( \Delta_{k^*} \), where \( t_0 \leq t_{k^*} \leq t_A \), namely the queue will be empty for all intervals up to \( t_{k^*} \). Therefore congestion window will be equal to

\[
W(t_{k^*}) = N (A + k^*) = Cd < 2NA \tag{2.2}
\]

and length of intervals up to \( t_{k^*+1} \) will simply be equal to propagation delay, that is

\[
\Delta_i = d, \quad i = 1, \ldots, k^* \tag{2.3}
\]

and after \( t_{k^*+1} \), queueing delay becomes effective as number of outstanding packets exceeds bandwidth delay product. Thus

\[
\Delta_i = d + \frac{N}{C} (i - k^*), \quad i = k^* + 1, \ldots, A \tag{2.4}
\]

Both \( \Delta_A \) and \( \Delta_{A+1} \) are special cases as the acknowledgements of \( 2NA \) packets sent during \( \Delta_A \) are received during both \( \Delta_A \) and \( \Delta_{A+1} \). Due to periodicity \( \Delta_{A+1} = \Delta_0 \). Therefore

\[
\Delta_0 + \Delta_A = d + \frac{NA}{C}
\]

As we are trying to fully utilize the link, the number of packets sent in period \([t_0, t_{A+1}]\) must be equal to \( C \cdot T \), where \( C \) is the transmission rate in packets/sec and \( T \) is the period, \( T = \sum_{i=0}^{A} \Delta_i \).
From Figure 2.3, number of packets, $P$, transmitted can be computed as

$$P = (NA + N) + (NA + 2N) + \ldots + (2NA - N) + (2NA)$$

$$= \sum_{j=1}^{A} N(A + j)$$

$$= NA^2 + \frac{NA(A + 1)}{2}$$

(2.5)

since number of transmitted packets during $\Delta_0$ and $\Delta_A$ is equal to $2A$.

Using (2.3) and (2.4), channel capacity, $R$, in number of packets can be computed as

$$R = CT = C \sum_{i=0}^{A} \Delta_i$$

$$= C \sum_{i=0}^{k^*} d + C \sum_{j=k^*+1}^{A} \left( d + \frac{N}{C} (j - k^*) \right)$$

$$= Ck^*d + C(A - k^*)d + N \sum_{j=1}^{A-k^*} j$$

$$= CdA + \frac{N (A - k^*) (A - k^* + 1)}{2}$$

(2.6)

Letting $P = R$, it is clear that

$$NA = Cd$$

(2.7)

$$k^* = 0$$

(2.8)

is a solution. In this case window size fluctuates between $NA$ and $2NA$ and (2.7) suggests that it should never be less than the bandwidth-delay product to ensure full utilization.

Using (2.7) we can find the queue length at any interval $\Delta_i$ by calculating the difference between $W(t)$ and the bandwidth delay product, which can be written as

$$q(t) = W(t) - Cd$$

Then by substituting $2NA$ as the maximum window size, $w_{\text{max}}$, the maximum queue size in period $[t_0, t_{A+1}]$ becomes,

$$q_{\text{max}} = w_{\text{max}} - Cd = 2NA - Cd = Cd$$

(2.9)
Implementing this algorithm at BS is advantageous as the connection state, such as queue length and rate, is available. Otherwise, they need to be estimated. Accessing BS, however, is not always feasible.

We propose a client-based approach which does not require any modification to the infrastructure or the TCP servers. This client-based algorithm remotely controls the base station queue size from the mobile, hence called remote AQM or R-AQM. We first laid the foundations of R-AQM for a single TCP flow, then extend it to handle multiple simultaneous TCP flows.

In order to implement this algorithm at the MH we need to compute the ideal queue size $q_{max}$ for a given rate, then estimate the queue length at the BS and compare it to $q_{max}$ to trigger congestion. Instead of the queue sizes we compute a maximum delay which is a function of $q_{max}$ and compare it with current round trip delay. The relationship between round-trip delay, $r$, experienced by the packets, and queue length, $q$, at the BS can be written as,

$$r = d + \frac{q}{C} \quad (2.10)$$

We can write a similar relationship for the maximum allowed delay, $r_{max}$, and $q_{max}$,

$$r_{max} = d + \frac{q_{max}}{C} \quad (2.11)$$

Using (2.10) and (2.11), the deterministic packet drop criteria ($q > q_{max}$) is equivalent to $r > r_{max}$. According to Proposition 1, substituting $q_{max} = C \times d$ into (2.11) we obtain $r_{max} = 2d$. Therefore we drop a packet if $r > r_{max}$ or

$$r > 2d \quad (2.12)$$

Dropping successfully received packets at MH results in unnecessary wasting of the radio resources. ECN provides a more efficient scheme to signal server to reduce its window size without dropping packets. Therefore any packet dropping suggested by R-AQM in this chapter is substituted with ECN whenever the feature is supported by the remote server.
Algorithm 1 R-AQM for a single TCP flow

for each received packet

    if \( r > 2d \) and \( now > p_{\text{drop\_timer}} \)

        drop packet from a flow

        \( p_{\text{drop\_timer}} = now + r \)

    else

        do nothing

    end

end

To implement (2.12) at the client we need to estimate propagation delay, \( d \), and queueing delay experienced by the TCP packets. Propagation delay can be estimated either from the three-way handshake of TCP or by pinging the server when no data exchange is done to ensure empty dedicated buffer at B. In Section 2.3, we used a static value for \( d \) averaged from various measurements as the delay is mostly dominated by the delay of the wireless link.

The round-trip delays are estimated by continuously pinging the server. In order to minimize the extra load and to get a better estimate of the queue length we use the minimum packet size for ping. Another approach can be to ping the BS instead of pinging the server and use a fixed delay for \( d_f \).

In our algorithm we use (2.12) to make the packet dropping decisions, i.e., whenever the observed packet delay exceeds \( 2d \) a packet is dropped. The effect of this packet drop will only be observed after a round-trip time, \( r \), thus after dropping a packet we do not drop any other packet for a duration of \( r \) even if (2.12) is satisfied.

A simplified pseudo code of our algorithm is given in Algorithm 1.
Figure 2.4: At steady state, aggregate window size in packets, $W(t)$, of $N$ synchronized TCP flows fluctuates between $A$ and $2A$ packets with a period of $T = \frac{t_{A/N+1}}{t_0} - t_0$. $\Delta_i$ for $i$ integer is the interval between two successive change of window size in terms of packets instances. $t_i$, $i$ integer, is the time instance when $W(t)$ is incremented. Aggregate window size, $W_N(t)$, for $N$ synchronized simultaneous flows. All flows increment and reduce their window size at the same time. $W_N(t)$ repeats itself with a period of $T' = \frac{t_A}{N+1} - t_0$.

### 2.2.2 Multiple TCP Connections

Section 2.2.1 assumes a single TCP connection, in this section we will generalize the algorithm for multiple flows.

Worst case scenario is $N$ synchronized TCP flows that changes their window size simultaneously, causing large fluctuations. For example, aggregate window size of $N$ flows is incremented by $N$ packets every round-trip time and is reduced to half when packets are dropped from all flows. A sample of the window size behavior of this model is shown in Figure 2.4.

In Proposition 1, it is shown that $NA$ must be equal to $Cd$ in order to fully
utilize the link with minimum $q_{\text{max}} = Cd$ for $N$ synchronized flows. Therefore Algorithm 1 for the single flow case can also be used for any number of synchronized flows. Enforcing all flows change their window size at the same time is not possible to achieve on a real network, nor is it preferred due to large fluctuations. Therefore, we will consider the case where $N$ flows are uniformly distributed within one period. In order to prevent synchronization of all connections, we drop packets uniformly during interval $[t_0, t_{A/N+1}]$ or over a period, i.e. wait $\phi = T/N$ between successive packet drops, where $T = t_{A/N+1} - t_0$. In Proposition 1, (2.5) states that the total number of packets transmitted within a period is given by

$$P = NA^2 + \frac{NA(A+1)}{2} = (3A+1) \frac{NA}{2}$$
substituting $NA = Cd$ using (2.7) we obtain

$$P = \left( \frac{3Cd}{N} + 1 \right) \frac{Cd}{2}$$

Since $P$, the total number of packets transmitted in $T$, must be at least equal to bandwidth times period for full utilization, namely $P = CT$, shift duration $\phi$ becomes

$$\phi = \frac{T}{N} = \frac{P}{NC} = \left( \frac{3Cd}{N} + 1 \right) \frac{d}{2N}$$

where $C$ is in packets per second.

We drop at most one packet in a round-trip time. It is important to note that the flows whose packets are not yet dropped would continue increasing their window sizes. As a result the maximum queue length may become more than the $q_{max}$ value calculated in the model.

When each flow is shifted by $\phi$, aggregate window size exhibits a behavior similar to Figure 2.5, reaching a maximum size of $w'_{max}$, given by,

$$w'_{max} = 2A + \sum_{i=1}^{N-1} \left( A + \frac{iA}{N} \right)$$

$$= 2A + \frac{3A}{2}(N - 1)$$

$$= \frac{(3N + 1)}{2} A$$

$$= \frac{(3N + 1)}{2} \frac{Cd}{N}$$

Then the new maximum queue length is computed as

$$q'_{max} = w'_{max} - Cd$$

$$= \frac{(3N + 1)}{2} \frac{Cd}{N} - Cd$$

$$= \left( \frac{N + 1}{2N} \right) Cd$$
As a result the new dropping criteria becomes

\[ r > d + \frac{q_{\text{max}}}{C} \]

\[ = d + \left( \frac{N + 1}{2N} \right) d \]

\[ = \left( \frac{3N + 1}{2N} \right) d \]

\[ = \left( 1.5 + \frac{1}{2N} \right) d \]  \hspace{1cm} (2.13)

where \( r \) is the round-trip-delay. (2.13) is equivalent to (2.12) when \( N = 1 \). For \( N > 1 \), we drop packets earlier since flows are unsynchronized and hence fluctuate less.

In our model we assume that a new packet will arrive to the BS as soon as the queue is drained. However, in a real environment there can be many variations in the timing that would result in under utilization. Thus, in our experiments we targeted a minimum queue length of 2 packets. Dropping criteria in (2.13) is updated as

\[ r > \left( 1.5 + \frac{1}{2N} \right) d + \frac{2}{C} \]  \hspace{1cm} (2.14)

We estimate the bandwidth by averaging the received packets over a period of 3 seconds. In our algorithm (2.14) is used for making the packet drop decisions and \( \phi \) is for preventing synchronization of all flows. In the analytical model these two values are supposed to be reached simultaneously, but in a real environment there are many uncontrollable factors that would prevent this. During testing to prevent \( \phi \) from being the limiting factor in packet drop decisions we scaled it with some constant, \( \beta < 1 \). We evaluated different values for \( \beta \) and observed very little variations in the results. During our experiments we fixed \( \beta \) to 0.8.

The final pseudo code for multiple TCP flows is shown in Algorithm 2.
Algorithm 2 R-AQM for multiple TCP flows

for each received packet

    if $r > (1.5 + \frac{1}{2N}) d + \frac{2}{C} \text{ and } now > p_{\text{drop_timer}}$
        drop packet from a flow
        $p_{\text{drop_timer}} = now + \phi \times \beta$
    else
        do nothing
    end
end

2.3 Experimental Results

We used the setup shown in Figure 2.6 consisting of a commercial CDMA2000 1xRTT network including Radio Base Station (RBS), Base Station Controller (BSC) and Packet Data Serving Node (PDSN). We used a wireless channel emulator (Spirent TAS4500 fader) and a noise generator (Spirent TAS4600A) in the forward link from BS to mobile client to maintain similar channel conditions when comparing different algorithms. Figure 2.6 shows some of the components of the cellular network that was used in experiments. As our focus in this study is the forward link, the reverse link is assumed to be perfect hence was directly connected to the BS with small attenuation.

We compared our new scheme, R-AQM, with regular drop-tail (no AQM employed) in a single user scenario where the user downloaded four files simultaneously from a remote server creating four different flows. The file sizes were 300, 800, 1700, and 2400 kB, respectively. Due to different life span of each flow, we were able to observe the performance of R-AQM under different number of flows. The server were running the Reno version of TCP with the maximum packet size fixed to 1000 bytes.

In R-AQM algorithm, to signaling rate reduction by dropping a packet already successfully transmitted to the client over the air wastes scarce radio re-
Figure 2.6: Experimental setup. Commercial CDMA2000 1xRTT cellular network was used. Fader and noise generator provided emulating similar channel conditions when comparing R-AQM with mere drop-tail queues.

sources and hence is not efficient. Therefore, in our experiments we marked the ECN bit in the outgoing ACK packets instead of packet drops. The response of the TCP server is the same in both cases, simply reducing the window size.

We also conducted some experiments with packet dropping to validate our R-AQM and observed that even though BS queues were highly utilized, file transfers took longer due to retransmission of the discarded packets.

We ran three sets of experiments both for drop-tail and R-AQM using different links speeds, 48, 86, and 163kbps. The effective rates that we observed at the transport layer, however, were close to 38, 75, and 150 kbps, respectively. The reduction in the rates are due to link layer overhead and retransmissions.

The experimental results for R-AQM and drop-tail are summarized in Table 2.1 and 2.2. In Figure 2.8 we also present a sample plot for 86 kbps case. From the RTT subplot, it can be seen that R-AQM outperforms drop-tail without compromising utilization.

It is important to note that R-AQM has an overhead as it is continuously
Table 2.1: Test results for R-AQM algorithm. 'BSC Q' is the queue length at the BSC, 'RTT' is the average round-trip-delay, 'Utilization' is the percent of the time BSC buffers are non-empty when the wireless link is active, and $N$ is the number of active connections for the same user.

<table>
<thead>
<tr>
<th>Rate</th>
<th>BSC Q (KB)</th>
<th>RTT (sec)</th>
<th>Utilization (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>48 kbps</td>
<td>86 kbps</td>
<td>163 kbps</td>
</tr>
<tr>
<td></td>
<td>48 kbps</td>
<td>86 kbps</td>
<td>163 kbps</td>
</tr>
<tr>
<td></td>
<td>48 kbps</td>
<td>86 kbps</td>
<td>163 kbps</td>
</tr>
<tr>
<td>$N$</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>3.5</td>
<td>3.7</td>
<td>3.6</td>
</tr>
<tr>
<td></td>
<td>1.1</td>
<td>0.8</td>
<td>0.6</td>
</tr>
<tr>
<td></td>
<td>99.2</td>
<td>97.4</td>
<td>94.8</td>
</tr>
<tr>
<td>2</td>
<td>4.2</td>
<td>4.3</td>
<td>4.7</td>
</tr>
<tr>
<td></td>
<td>1.3</td>
<td>0.9</td>
<td>0.7</td>
</tr>
<tr>
<td></td>
<td>98.6</td>
<td>99.2</td>
<td>99.6</td>
</tr>
<tr>
<td>3</td>
<td>5.0</td>
<td>5.3</td>
<td>5.3</td>
</tr>
<tr>
<td></td>
<td>1.4</td>
<td>1.0</td>
<td>0.7</td>
</tr>
<tr>
<td></td>
<td>99.8</td>
<td>99.9</td>
<td>98.5</td>
</tr>
<tr>
<td>4</td>
<td>5.3</td>
<td>5.9</td>
<td>6.5</td>
</tr>
<tr>
<td></td>
<td>1.5</td>
<td>1.0</td>
<td>0.7</td>
</tr>
<tr>
<td></td>
<td>99.9</td>
<td>99.8</td>
<td>99.5</td>
</tr>
</tbody>
</table>

Table 2.2: Test results for drop-tail algorithm. 'BSC Q' is the queue length at the BSC, 'RTT' is the average RTT, 'Utilization' is the percent of the time BSC buffers are non-empty when the wireless link is active, and $N$ is the number of active connections for the same user.

<table>
<thead>
<tr>
<th>Rate</th>
<th>BSC Q (KB)</th>
<th>RTT (sec)</th>
<th>Utilization (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>48 kbps</td>
<td>86 kbps</td>
<td>163 kbps</td>
</tr>
<tr>
<td></td>
<td>48 kbps</td>
<td>86 kbps</td>
<td>163 kbps</td>
</tr>
<tr>
<td></td>
<td>48 kbps</td>
<td>86 kbps</td>
<td>163 kbps</td>
</tr>
<tr>
<td>$N$</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>13.8</td>
<td>11.1</td>
<td>7.2</td>
</tr>
<tr>
<td></td>
<td>3.21</td>
<td>1.64</td>
<td>1.21</td>
</tr>
<tr>
<td></td>
<td>100</td>
<td>99.9</td>
<td>98.1</td>
</tr>
<tr>
<td>2</td>
<td>15.7</td>
<td>15.0</td>
<td>13.8</td>
</tr>
<tr>
<td></td>
<td>3.62</td>
<td>2.03</td>
<td>1.15</td>
</tr>
<tr>
<td></td>
<td>100</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td>3</td>
<td>16.1</td>
<td>15.3</td>
<td>14.1</td>
</tr>
<tr>
<td></td>
<td>3.71</td>
<td>2.02</td>
<td>1.13</td>
</tr>
<tr>
<td></td>
<td>100</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td>4</td>
<td>16.0</td>
<td>15.9</td>
<td>15.2</td>
</tr>
<tr>
<td></td>
<td>3.73</td>
<td>2.03</td>
<td>0.79</td>
</tr>
<tr>
<td></td>
<td>100</td>
<td>100</td>
<td>100</td>
</tr>
</tbody>
</table>
Figure 2.7: Some of the components of the CDMA2000 1xRTT cellular network that was used in the experiments: BSC, RBS, PDSN, and MSC. PDSN is the gateway that connects the core network to the Internet. Telos switch was used as the Mobile Switching Center (MSC).

pinging the server to estimate the queue size at the BS. As a percentage of the link capacity, these overheads are 3%, 1.5%, and 0.7% for link capacities 48, 86, and 163 kbps respectively.

We also experimented with changing link rates over time for multiple TCP flows. Top subfigure in Figure 2.12 plots link rates as a function of time. Note that since this was an experimental study, it was impossible to match the exactly same channel conditions when comparing R-AQM with drop-tail algorithm. The channel for R-AQM, however, faces slightly deeper fade between 345 and 395 seconds and has a lower rate. BSC queue sizes in the middle subfigure suggests that R-AQM has a high utilization since queue size almost never drops to zero. Bottom subfigure compares the RTT values. Drop-tail observes as high as 4 second delays when the
link rate is very slow. R-AQM, on the hand, maintains a low packet delay even for very slow links.

BS maintains a large buffer size to accommodate the highest link rate when a drop-tail scheme without an AQM is employed. Otherwise packet drops due to buffer overflow would cause TCP to unnecessarily slow down and never attain the provided high link rate. For our case BS queue size was set to 20 kB. Benefits of R-AQM vanishes for the highest link rate since the queue size is already at the optimal level and becomes very significant for slower link rates. For slow link rates BSC takes much longer to drain the queue and as a result higher per-packet delays are experienced. R-AQM provides lower packet delays and jitter without compromising the utilization by targeting a small but non-zero BS queue size.

2.4 Remarks

In this chapter we first proposed an analytical model for a single TCP connection and then extended this model to multiple TCP flows. Using this model we developed an active queue management (AQM) algorithm that can be implemented at the client side instead of a network node.

We also provided methods for estimating the required network parameters, such as round trip time and link capacity at the client with minimal overhead. We keep the BS queue length at a sufficiently small value to maintain high link utilization.

We compared our algorithm with drop-tail scheme using a CDMA2000 1xRTT network. The results show that R-AQM outperforms drop-tail in terms of per-packet delay and jitter without compromising link utilization.

This chapter, in full, is a reprint of the material as it appears in proceedings of IEEE Consumer Communications & Networking Conference 2006. Akin, Ozdemir; Ergüt, Salih; Rao, Ramesh R. The dissertation author and Mr. Akin were primary investigators and authors of this paper with equal contributions.
Figure 2.8: Comparing BSC queue size and RTT for drop-tail and RAQM algorithms when forward link rate is 86kbps. Note that R-AQM scheme achieves a mean RTT of 0.9 sec with a standard deviation of 0.2 sec, while drop-tail has a RTT of 2.0 sec on the average with a standard deviation of 0.4 sec. Both links achieve high utilization since BSC queue almost never drains completely.
Figure 2.9: Comparison of average number of TCP flows vs. BSC queue sizes for drop-tail and R-AQM algorithms for different forward link rates. Larger queue sizes cause larger packet delays.

Figure 2.10: Comparison of average number of TCP flows vs. RTT for drop-tail and R-AQM algorithms for different forward link rates. R-AQM only performs slightly worse than drop-tail for forward link rate of 163 kbps. Because BSC queue size is optimized for this highest rate for the drop-tail algorithm.
Figure 2.11: Comparison of average number of TCP flows vs. utilization for drop-tail and R-AQM algorithms for different forward link rates. Both algorithms utilize the link better when there are two or more flows.
Figure 2.12: Comparison of RAQM and drop-tail only schemes. A large file was being downloaded while the channel conditions changed. a) Top subfigure plots downlink data rate as a function of time. Since experimental, it was not possible to perfect match the channel conditions. b) Middle subplot plots the queue lengths at BSC. c) Bottom subfigure plots end-to-end delays.
Bibliography


Packet size aware path setup for 802.11 based mesh networks

802.11 based wireless local area networks are commonly used in residential/business broadband deployments as a last mile Internet access solution. Generally it is operated in Independent Basic Service Set (IBSS) mode where there is a single access point (AP) and multiple client devices associated with this AP. Wireless multi-hop access networks have recently been studied extensively in academia and by the industry [25, 20, 26, 19, 12, 24, 1].

These wireless access networks serve various types of applications ranging from file download and web browsing to Voice over IP (VoIP). Each of these applications has different delay and throughput requirements and generate packets with relatively different sizes. For example, file download applications optimize the throughput rather than delays and therefore generally use 1500 byte packets, which is equal to the size of the Maximum Transmission Unit (MTU). VoIP type of applications, on the other hand, generate as small packets as 200 bytes to satisfy stringent delay requirements. Previous research on packet size distribution from real traffic traces suggest diversity in the packet size [28, 18, 13, 6, 8].

802.11 supports multiple data rates to adapt their transmission rates on the channel conditions. Therefore, in a real deployment, it is very likely to observe different transmission rates on various wireless links.

Furthermore, as the link speeds are getting faster, the MTU sizes tend
to get larger; for example, jumbo frames (around 9000 bytes) are proposed for gigabit ethernet networks. This suggests that the asymmetry among the packet sizes generated by different applications may significantly effect the performance of the multi-hop wireless access networks.

Supporting multiple transmission rates, 802.11 is known to assign long term fair rates among the contending links such that each link gets a fair chance to access the channel. When there are multiple links with different data rates in the same contention region, the overall effective throughput of the network may considerably degraded.

The links with the lower data rates require more time to send a packet of certain size than the ones with the higher rates do. Not only the lower rate link suffers from low throughput but also the overall network throughput is significantly degraded. There are many novel research papers in the literature that identifies [14] and studies the problem [17, 16, 9, 7].

In [17], authors discuss the advantages of multi-hop extensions of 802.11 based WLAN networks and propose a novel cooperative MAC-layer solution that targets to solve this inefficient rate assignment of 802.11 WLAN networks. The idea is that each mobile client with a low data rate to AP identifies a helper (i.e. relay) node that has a higher data rate and communicates with the AP over two hop links using this helper node if the overall transmission delays are smaller when compared to the direct link. MAC-layer overheads are ignored in their system model. In [16], authors implemented the cooperative MAC-protocol proposed in [17] and demonstrated the benefits of such a scheme.

Alternative mechanisms, however, can be employed to achieve the same objective based on this delay model. It can be argued that instead of a modified MAC layer solution as in [17], simple off-the-shelf technologies such as Wireless Distribution System (WDS) [15] and the Spanning Tree Protocol (STP) [23] can be used to set up a cooperative multi-hop environment in an 802.11 WLAN network. For instance, each cooperating node in the network enables WDS feature and runs STP and the AP is set as the root of the spanning tree. In addition, the cost metric of link $i$, $C_i$, is set inversely proportional to the corresponding bit rate.
$R_i$ (i.e $C_i = A/R_i$, where $A$ is constant). Since STP minimizes the path costs from each node to the root, the transmission delays from each client node to the AP is minimized. Furthermore, such a mechanism does not require all the nodes to be cooperative, namely to have WDS and run optimized STP. Therefore, any standard 802.11 client node can network with the cooperating nodes.

[9] discusses an extensive transmission delay model for 802.11 based wireless networks that includes the MAC-layer overhead, which becomes more significant as the size of the packet gets smaller. The transmission delay to send a packet of size $P$ over an 802.11 link $i$ is modeled as $D_i(P) = \theta_i + P/b_i$ where $\theta_i$ and $b_i$ are the MAC-layer overhead time and the transmission rate respectively. Using this model, they proposed setting lower MTU values to the lower data rate clients to ensure rate proportional fairness and overcome the fairness problem of 802.11.

As modeled in [9], MAC-layer overhead becomes more significant as the size of the packet gets smaller. In order to tackle this problem, [22] proposes aggregation of VoIP packets as a traffic shaping mechanism in order to decrease the MAC-layer overhead per transmitted byte in the system. Aggregation leads to a lower number of packets in the system with higher payloads, eventually decreasing the effective MAC overhead for a single packet.

[3, 4] proposes using a Medium Time Metric (MTM) which aims to minimize the end-to-end transmission delay of a packet. As in [9], the delay model here includes the overhead time. Another design consideration for the authors is the reliability metric for each link. They propose using MTM for a certain tuned packet size, which is 1500 bytes. Path setup is optimized with respect to this packet size such that link weights are set proportional to the estimated transmission delay of a 1500-byte packet. Therefore, in their scheme a single path is assigned for all the packets that share the same source and destination. Since the reliability is the main concern, MTM optimized paths are generally found to avoid the lower rate links.

In this chapter, we propose path setup algorithms to optimize the overall network delay and throughput by taking into consideration of different packet sizes and transmission rates. We show that rather than being oblivious to the routed
packet contents, network nodes greatly benefit from using a contextual data such as packet size.

First we conducted experiments to measure the transmission delay experienced by packets with different sizes. Based on these measurements, the transmission delay on an 802.11 link is modeled as a function of the MAC overhead time, packet size and the effective transmission rate as in [9].

Using this model, the total transmission delays of alternative paths between a source and destination nodes are compared. It is seen that the optimum path for a packet that maximizes throughput indeed depends on the size of the packet. Note that minimizing the overall transmission delay is equivalent to maximizing the end-to-end throughput [3, 4].

Next, a sample packet size aware routing mechanism is implemented as a Linux kernel module and the corresponding performance is evaluated through various experiments.

Finally, we discuss general solutions for packet-size aware path setup for wireless networks considering the packet size statistics.

Our work differs from [17] in many aspects. First unlike [17], we do not require any change in the existing MAC-layer protocol (e.g. 802.11). In addition to that, our proposed path setup mechanisms are not limited to two-hop communication, instead three or more hop paths can be setup from source node (e.g. client node) to the destination node (e.g. AP). Also, our solution takes into account overhead times such that the path setup decision depends not only on the bit rate of the channel but also on the size of the packet to be transmitted.

Our results show that optimizing the path only for a single packet size (usually 1500 bytes) as in MTM may significantly reduce the network performance. Considering the packet size statistics [13, 11] and the newly emerging inelastic applications such as VoIP, it is likely to observe significant amount of small packets. We show that VoIP packets, which can be as small as 200 bytes, should be treated differently than the web traffic packets, which can be as large as 1500 bytes. Our experiments and estimations support that the packet size aware path setup schemes are able to increase the overall performance up to 50% compared to MTM, that
tunes path setup for 1500 bytes packets in an 802.11b based mesh network. For certain scenarios, we observed 13-17% throughput increase for TCP flows since data packets (1500 bytes) and ACK packets (around 40 bytes) traversed different paths.

One of the important results of this chapter is that as the number of hops increases, the MAC overhead time for each packet accumulates and it may not be optimum to send a packet with certain size over a multi-hop higher data rate alternative path. Instead, sending the packet over a direct link or a path with smaller number of hops with lower data rate may minimize the delay and hence improve the overall network throughput.

The rest of this chapter is as follows: Section 3.1 discusses the network model. In Section 3.2, the path cost optimization problem is examined and algorithms and implementations are presented.

3.1 Network Model

We consider static multi-hop 802.11 networks where all the network nodes are in the same contention neighborhood. This type of networks are being used as the last mile access solution in residential or business areas, and emergency response applications [2].

3.1.1 Delay/Throughput Model

[9] estimates the delay and throughput performance of multi-rate 802.11b WLAN networks experimentally. In their model, transmission delay of a packet with size $P$ is approximated by

$$D_i(P) = (\theta_i + P/b_i)$$

(3.1)

where $\theta_i$ is the MAC layer overhead time and $b_i$ is the effective transmission rate for link i. The details on the MAC layer overhead time can be found in [9, 27].
Note that this delay does not include any queuing delays on the link. Basically, this model captures the time elapses to transmit a packet from the transmitter to the receiver without any queueing. Therefore we call this delay as Transmission Delay.

Transmission Delay can be used to estimate the throughput, $T_i(P)$, of a flow on a single 802.11 link as follows (assuming all the packets of the connection have the same size, $P$)

$$T_i(P) = \frac{P}{D_i(P)} \quad (3.2)$$

In addition to that, 802.11 assigns long term fair rates among the contending wireless links on the same channel such that each link gets a fair chance to access the channel. Therefore, the throughput of a flow traversing multiple 802.11 links can be modelled as

$$R = \frac{P}{\sum_i \theta_i + P/b_i} \quad (3.3)$$

It is still assumed that the sizes of all the packets belonging to the same flow are fixed and equal to $P$.

**Validation of the model**

In order to validate this model, we have performed various experiments to estimate the parameters such as the overhead time ($\theta_i$) and effective bit rate ($b_i$) for each transmission rate.

Our methodology is first to measure the throughput of a UDP traffic (with infinite demand) over a single hop wireless link. The size of the packets belonging to this traffic is kept constant throughout each experiment.

Table 3.1 shows the measured throughput values of UDP and TCP flows over a single 802.11b link with various transmission rate modes. In the same table, the throughput values for different packet sizes (1500, 1000, 500, 300 bytes) are available.
Table 3.1: Throughput estimations from measurements using IPERF and model values for single-hop links. Deviations are with respect to UDP estimations.

<table>
<thead>
<tr>
<th>MTU/PS (bytes)</th>
<th>Rate (Mb/s)</th>
<th>Est. TCP (kb/s)</th>
<th>Est. UDP (kb/s)</th>
<th>Model (kb/s)</th>
<th>Deviation (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1500/1450</td>
<td>1</td>
<td>678</td>
<td>758</td>
<td>759</td>
<td>-0.1319%</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>1270</td>
<td>1443</td>
<td>1441</td>
<td>0.1386%</td>
</tr>
<tr>
<td></td>
<td>5.5</td>
<td>2777</td>
<td>3390</td>
<td>3398</td>
<td>-0.2360%</td>
</tr>
<tr>
<td></td>
<td>11</td>
<td>4060</td>
<td>5088</td>
<td>5150</td>
<td>-1.2186%</td>
</tr>
<tr>
<td>1000/950</td>
<td>1</td>
<td>626</td>
<td>721</td>
<td>717</td>
<td>0.5548%</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>1130</td>
<td>1323</td>
<td>1331</td>
<td>-0.6047%</td>
</tr>
<tr>
<td></td>
<td>5.5</td>
<td>2317</td>
<td>2880</td>
<td>2929</td>
<td>-1.7014%</td>
</tr>
<tr>
<td></td>
<td>11</td>
<td>3123</td>
<td>4055</td>
<td>4126</td>
<td>-1.7509%</td>
</tr>
<tr>
<td>500/450</td>
<td>1</td>
<td>504</td>
<td>601</td>
<td>609</td>
<td>-1.3311%</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>855</td>
<td>1083</td>
<td>1069</td>
<td>1.2927%</td>
</tr>
<tr>
<td></td>
<td>5.5</td>
<td>1533</td>
<td>2135</td>
<td>2027</td>
<td>5.0585%</td>
</tr>
<tr>
<td></td>
<td>11</td>
<td>1817</td>
<td>2735</td>
<td>2515</td>
<td>8.0439%</td>
</tr>
<tr>
<td>300/250</td>
<td>1</td>
<td>381</td>
<td>499</td>
<td>495</td>
<td>0.8016%</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>615</td>
<td>825</td>
<td>822</td>
<td>0.3636%</td>
</tr>
<tr>
<td></td>
<td>5.5</td>
<td>977</td>
<td>1450</td>
<td>1381</td>
<td>4.7586%</td>
</tr>
<tr>
<td></td>
<td>11</td>
<td>1053</td>
<td>1714</td>
<td>1579</td>
<td>7.8763%</td>
</tr>
</tbody>
</table>
Figure 3.1: Model vs. estimated throughputs for single-hop UDP flows for various packet sizes. Transmission rates 1, 2, 5.5, and 11 $Mb/s$ are allowed physical transmission rates for 802.11b standard. Data rates indicate estimated average UDP throughputs from measurements and model values. Mismatch between the model and the measurements increases for higher rates and smaller packet sizes. Data is from Table 3.1.
As described in (3.2), throughput is a function of the transmission delay. By reverse engineering this simple equation, we compute the transmission delay from the measured throughput of UDP traffic over a single link with fixed packet size such that \( D_i(P) = P/T_i(P) \). We repeated the same experiment for various transmission rates and packet sizes.

Using least square line fitting method, for each transmission rate mode (e.g. 1, 2, 5.5, and 11 Mb/s), corresponding overhead time variable \( (\theta_i) \) and effective rate \( (b_i) \) are estimated from obtained delay values.

Based on these estimations, the delay, \( D_x \) (in milliseconds), to send a packet with size \( P \) (in bytes) for an 802.11 link with transmission rates 1, 2, 5.5, and 11 Mb/s are as follows

\[
D_{1Mb/s} = 1.69msec + P/851.1kb/s \\
D_{2Mb/s} = 1.26msec + P/1702.1kb/s \\
D_{5.5Mb/s} = 1.04msec + P/5000.0kb/s \\
D_{11Mb/s} = 1.06msec + P/10000.0kb/s
\]

In order to see the accuracy of the estimated parameters using (3.2), the estimated throughput values are computed and compared at Table 3.1. As can be seen in Table I, the deviation of the model from the measured throughput is at most 8%.

We computed the estimated throughput for a single link according to (3.2) using the estimated delays and for multiple hops according to (3.1). As can be seen in Tables 3.1, 3.2, 3.3, and 3.4, small deviations of the estimated delay model from the measurements validate our model for single hop, two-hop and three-hop paths.

In addition to the single hop throughput measurements, we performed multi-hop (i.e. 2-hop and 3-hop) throughput measurements with various transmission rate combinations on the wireless links. Tables 3.2, 3.3, and 3.4 shows the measured UDP and TCP throughput values for 2-hop and 3-hop scenarios, respectively.

Multi-hop throughput is estimated using (3.3) and estimated single-hop
channel parameters, $\theta_i$ and $b_i$, and compared with the measured data. Tables 3.2, 3.3, and 3.4 show that the deviation of the model from the measured values are at most 14%.

Iperf [21], a network bandwidth estimation tool, is used to measure the TCP and UDP throughputs for single and multi-hop path scenarios. As for the hardware, Soekris 4521 [10] communication computers with Prism2.5 chipset based 802.11b wireless cards are used (see Figure 3.6). A trimmed version of the Debian Linux was used as the operating system and Host-AP driver was used for the wireless card.

\section*{3.2 Path Setup Optimization}

In the previous section, delay and throughput performance of single, 2-hop, and 3-hop paths with various transmission rates are examined experimentally. In this section, relative delay performance as a function of the packet size of a single hop path and multi-rate multi-hop paths is examined. We found that it is likely to encounter certain scenarios such that for the same source destination node pair, packets with different sizes are to traverse different paths in order to minimize the end-to-end transmission delay, or equivalently maximize the throughput. This section also discusses the implementation details, evaluates the performance of a sample packet size aware path selection mechanism for sample network topologies, and offer potential general solutions.

Consider network topologies as shown in Figure 3.7; between source and destination pairs there are a direct link and multi-hop alternatives. Figures 3.2, 3.3, and 3.4 compare the estimated mean packet transmission delay for various combinations of link rates and packet sizes over these topologies. We observed that as the hop count increases on the alternative path it is more likely to see a crossover point corresponding to a packet size over which the selected path is switched. For example, in Figure 3.2, total transmission delay on 1 $Mb/s$ direct link is lower than that of (2,5.5, 5.5 $Mb/s$) and (2, 11, 11 $Mb/s$) 3-hop paths when the packet size is smaller than about 1200 and 600 bytes, respectively. In
Table 3.2: Throughput measurements and estimations for 2-hop links. Deviations are with respect to UDP measurements. Packet sizes are greater than or equal to 500 bytes.

<table>
<thead>
<tr>
<th>MTU/PS (bytes)</th>
<th>Rate (Mb/s)</th>
<th>Est. TCP (kb/s)</th>
<th>Est. UDP (kb/s)</th>
<th>Model (kb/s)</th>
<th>Deviation (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1500/1450</td>
<td>1 - 1</td>
<td>323</td>
<td>363</td>
<td>378</td>
<td>-4.1322%</td>
</tr>
<tr>
<td></td>
<td>1 - 2</td>
<td>407</td>
<td>467</td>
<td>497</td>
<td>-6.4240%</td>
</tr>
<tr>
<td></td>
<td>1 - 5.5</td>
<td>510</td>
<td>588</td>
<td>620</td>
<td>-5.4422%</td>
</tr>
<tr>
<td></td>
<td>1 - 11</td>
<td>540</td>
<td>639</td>
<td>662</td>
<td>-3.5994%</td>
</tr>
<tr>
<td></td>
<td>2 - 2</td>
<td>616</td>
<td>715</td>
<td>721</td>
<td>-0.8392%</td>
</tr>
<tr>
<td></td>
<td>2 - 5.5</td>
<td>831</td>
<td>967</td>
<td>1012</td>
<td>-4.6536%</td>
</tr>
<tr>
<td></td>
<td>2 - 11</td>
<td>901</td>
<td>1080</td>
<td>1126</td>
<td>-4.2593%</td>
</tr>
<tr>
<td></td>
<td>5.5 - 5.5</td>
<td>1353</td>
<td>1723</td>
<td>1699</td>
<td>1.3929%</td>
</tr>
<tr>
<td></td>
<td>5.5 - 11</td>
<td>1653</td>
<td>2043</td>
<td>2047</td>
<td>-0.1958%</td>
</tr>
<tr>
<td></td>
<td>11 - 11</td>
<td>2123</td>
<td>2723</td>
<td>2575</td>
<td>5.4352%</td>
</tr>
<tr>
<td>1000/950</td>
<td>1 - 1</td>
<td>298</td>
<td>353</td>
<td>359</td>
<td>-1.6997%</td>
</tr>
<tr>
<td></td>
<td>1 - 2</td>
<td>382</td>
<td>455</td>
<td>466</td>
<td>-2.4176%</td>
</tr>
<tr>
<td></td>
<td>1 - 5.5</td>
<td>474</td>
<td>571</td>
<td>576</td>
<td>-0.8757%</td>
</tr>
<tr>
<td></td>
<td>1 - 11</td>
<td>506</td>
<td>622</td>
<td>611</td>
<td>1.7586%</td>
</tr>
<tr>
<td></td>
<td>2 - 2</td>
<td>565</td>
<td>671</td>
<td>666</td>
<td>0.7452%</td>
</tr>
<tr>
<td></td>
<td>2 - 5.5</td>
<td>742</td>
<td>897</td>
<td>915</td>
<td>-2.0067%</td>
</tr>
<tr>
<td></td>
<td>2 - 11</td>
<td>808</td>
<td>971</td>
<td>1006</td>
<td>-3.6045%</td>
</tr>
<tr>
<td></td>
<td>5.5 - 5.5</td>
<td>1200</td>
<td>1507</td>
<td>1465</td>
<td>2.7870%</td>
</tr>
<tr>
<td></td>
<td>5.5 - 11</td>
<td>1367</td>
<td>1730</td>
<td>1713</td>
<td>0.9827%</td>
</tr>
<tr>
<td></td>
<td>11 - 11</td>
<td>1700</td>
<td>2243</td>
<td>2063</td>
<td>8.0250%</td>
</tr>
</tbody>
</table>
Table 3.3: Throughput measurements and estimations for 2-hop links. Deviations are with respect to UDP measurements. Packet sizes are less than or equal to 500 bytes.

<table>
<thead>
<tr>
<th>MTU/PS (bytes)</th>
<th>Rate (Mb/s)</th>
<th>Est. TCP (kb/s)</th>
<th>Est. UDP (kb/s)</th>
<th>Model (kb/s)</th>
<th>Deviation (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>500/450</td>
<td>1 - 1</td>
<td>242</td>
<td>298</td>
<td>305</td>
<td>-2.3490%</td>
</tr>
<tr>
<td></td>
<td>1 - 2</td>
<td>310</td>
<td>388</td>
<td>388</td>
<td>0.0000%</td>
</tr>
<tr>
<td></td>
<td>1 - 5.5</td>
<td>367</td>
<td>475</td>
<td>468</td>
<td>1.4737%</td>
</tr>
<tr>
<td></td>
<td>1 - 11</td>
<td>396</td>
<td>506</td>
<td>490</td>
<td>3.1621%</td>
</tr>
<tr>
<td></td>
<td>2 - 2</td>
<td>432</td>
<td>553</td>
<td>535</td>
<td>3.2550%</td>
</tr>
<tr>
<td></td>
<td>5.5 - 2</td>
<td>547</td>
<td>706</td>
<td>700</td>
<td>0.8499%</td>
</tr>
<tr>
<td></td>
<td>5.5 - 5.5</td>
<td>811</td>
<td>1097</td>
<td>1014</td>
<td>7.5661%</td>
</tr>
<tr>
<td></td>
<td>11 - 2</td>
<td>584</td>
<td>755</td>
<td>750</td>
<td>0.6623%</td>
</tr>
<tr>
<td></td>
<td>11 - 5.5</td>
<td>880</td>
<td>1183</td>
<td>1122</td>
<td>5.1564%</td>
</tr>
<tr>
<td></td>
<td>11 - 11</td>
<td>1040</td>
<td>1440</td>
<td>1258</td>
<td>12.6389%</td>
</tr>
<tr>
<td>300/250</td>
<td>1 - 1</td>
<td>186</td>
<td>253</td>
<td>248</td>
<td>1.9763%</td>
</tr>
<tr>
<td></td>
<td>1 - 2</td>
<td>227</td>
<td>319</td>
<td>309</td>
<td>3.1348%</td>
</tr>
<tr>
<td></td>
<td>1 - 5.5</td>
<td>278</td>
<td>383</td>
<td>364</td>
<td>4.9608%</td>
</tr>
<tr>
<td></td>
<td>1 - 11</td>
<td>293</td>
<td>404</td>
<td>377</td>
<td>6.6832%</td>
</tr>
<tr>
<td></td>
<td>2 - 2</td>
<td>316</td>
<td>436</td>
<td>411</td>
<td>5.7339%</td>
</tr>
<tr>
<td></td>
<td>5.5 - 2</td>
<td>387</td>
<td>535</td>
<td>515</td>
<td>3.7383%</td>
</tr>
<tr>
<td></td>
<td>5.5 - 5.5</td>
<td>539</td>
<td>750</td>
<td>691</td>
<td>7.8667%</td>
</tr>
<tr>
<td></td>
<td>11 - 2</td>
<td>404</td>
<td>553</td>
<td>541</td>
<td>2.1700%</td>
</tr>
<tr>
<td></td>
<td>11 - 5.5</td>
<td>567</td>
<td>791</td>
<td>737</td>
<td>6.8268%</td>
</tr>
<tr>
<td></td>
<td>11 - 11</td>
<td>640</td>
<td>925</td>
<td>790</td>
<td>14.5946%</td>
</tr>
</tbody>
</table>
Table 3.4: Throughput measurements and estimations for three-hop links. Deviations are with respect to UDP measurements.

<table>
<thead>
<tr>
<th>MTU/PS (bytes)</th>
<th>Rate (Mb/s)</th>
<th>Est. TCP (kb/s)</th>
<th>Est. UDP (kb/s)</th>
<th>Model (kb/s)</th>
<th>Deviation (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1500/1450</td>
<td>2 - 5.5 - 5.5</td>
<td>584</td>
<td>756</td>
<td>780</td>
<td>-3.1746%</td>
</tr>
<tr>
<td></td>
<td>11 - 5.5 - 5.5</td>
<td>913</td>
<td>1.22</td>
<td>1278</td>
<td>-4.7541%</td>
</tr>
<tr>
<td></td>
<td>11 - 11 - 5.5</td>
<td>1.06</td>
<td>1.45</td>
<td>1465</td>
<td>-1.0345%</td>
</tr>
<tr>
<td></td>
<td>11 - 11 - 2</td>
<td>680</td>
<td>854</td>
<td>924</td>
<td>-8.1967%</td>
</tr>
<tr>
<td></td>
<td>5.5 - 5.5 - 5.5</td>
<td>840</td>
<td>1090</td>
<td>1133</td>
<td>-3.9450%</td>
</tr>
<tr>
<td></td>
<td>11 - 11 - 11</td>
<td>1290</td>
<td>1790</td>
<td>1717</td>
<td>4.0782%</td>
</tr>
<tr>
<td>1000/950</td>
<td>2 - 5.5 - 5.5</td>
<td>538</td>
<td>695</td>
<td>697</td>
<td>-0.2878%</td>
</tr>
<tr>
<td></td>
<td>11 - 5.5 - 5.5</td>
<td>801</td>
<td>1083</td>
<td>1081</td>
<td>0.1847%</td>
</tr>
<tr>
<td></td>
<td>11 - 11 - 5.5</td>
<td>881</td>
<td>1223</td>
<td>1210</td>
<td>1.0630%</td>
</tr>
<tr>
<td></td>
<td>11 - 11 - 2</td>
<td>606</td>
<td>778</td>
<td>809</td>
<td>-3.9846%</td>
</tr>
<tr>
<td></td>
<td>5.5 - 5.5 - 5.5</td>
<td>724</td>
<td>984</td>
<td>976</td>
<td>0.8130%</td>
</tr>
<tr>
<td></td>
<td>11 - 11 - 11</td>
<td>1047</td>
<td>1493</td>
<td>1375</td>
<td>7.9035%</td>
</tr>
<tr>
<td>500/450</td>
<td>2 - 5.5 - 5.5</td>
<td>395</td>
<td>534</td>
<td>520</td>
<td>2.6217%</td>
</tr>
<tr>
<td></td>
<td>11 - 5.5 - 5.5</td>
<td>536</td>
<td>768</td>
<td>722</td>
<td>5.9896%</td>
</tr>
<tr>
<td></td>
<td>11 - 11 - 5.5</td>
<td>593</td>
<td>867</td>
<td>776</td>
<td>10.4960%</td>
</tr>
<tr>
<td></td>
<td>11 - 11 - 2</td>
<td>427</td>
<td>595</td>
<td>578</td>
<td>2.8571%</td>
</tr>
<tr>
<td></td>
<td>5.5 - 5.5 - 5.5</td>
<td>497</td>
<td>723</td>
<td>676</td>
<td>6.5007%</td>
</tr>
<tr>
<td></td>
<td>11 - 11 - 11</td>
<td>672</td>
<td>971</td>
<td>838</td>
<td>13.6972%</td>
</tr>
<tr>
<td>300/250</td>
<td>2 - 5.5 - 5.5</td>
<td>274</td>
<td>411</td>
<td>375</td>
<td>8.7591%</td>
</tr>
<tr>
<td></td>
<td>11 - 5.5 - 5.5</td>
<td>351</td>
<td>538</td>
<td>480</td>
<td>10.7807%</td>
</tr>
<tr>
<td></td>
<td>11 - 11 - 5.5</td>
<td>384</td>
<td>578</td>
<td>502</td>
<td>13.1488%</td>
</tr>
<tr>
<td></td>
<td>11 - 11 - 2</td>
<td>297</td>
<td>435</td>
<td>403</td>
<td>7.3563%</td>
</tr>
</tbody>
</table>
Figure 3.3, for a packet size of 900 bytes or larger, the delay on $(11, 11, 11 \text{ Mb/s})$ 3-hop path is lower than that of $2 \text{ Mb/s}$ direct link.

![Graph showing delay comparison](image)

**Figure 3.2:** Delay comparison of a $1 \text{ Mb/s}$ single-hop link with three-hop alternatives.

Similarly the delay performance comparison of a $2 \text{ Mb/s}$ single hop path with two hop alternatives are shown in Figure 3.4. For instance, 600 byte is the crossover packet size between the $2 \text{ Mb/s}$ link and the $5.5, 5.5 \text{ Mb/s}$ 2-hop path.

Furthermore, as the number of hops of the alternative paths increase, it is possible to come across more complicated path setup problems. In Figure 4 the delay performance of a direct link with $1 \text{ Mb/s}$ bitrate and various alternative multi-hop paths are demonstrated. Depending on the number of alternative paths from source to destination, the problem of optimum path selection may become significantly complicated.

As can be seen in the previous examples, it is likely to come across the scenarios such that forwarding packets with different sizes to different paths may improve the delay/throughput performance of the overall network significantly. In
the next section, we discuss a sample implementation of a packet size aware path setup mechanism and examine the performance improvement.

3.2.1 A Sample Implementation

We considered two different network structures as in Figure 3.7. In the first setup there are two alternative paths between source and destination nodes: one with single hop and the other with two hops. In the second setup single and 3-hop alternatives are considered.

We implemented a Linux (2.6) kernel module called *PSA module* which is capable of forwarding packets according to their sizes. Basically, this kernel module registers with the netfilter prerouting hook in Linux. All the incoming packets pass this hook before any routing decision is made. *PSA module* leverages the advanced policy routing feature of Linux. For each next hop candidate a new routing table is created and an appropriate policy rule is associated with that routing table. Optimal next route of a packet with respect to its size is computed as it arrives to the *PSA module*, and the packet is marked to be forwarded according to the

Figure 3.3: Delay comparison of 2 Mb/s and three-hop alternatives
Figure 3.4: Delay comparison of 2 Mb/s and two-hop alternatives

appropriate routing table. This ensures the delivery of the packet in question to a
node on the minimum delay path from the source to the destination.

In our prototype implementation, PSA module is passed a threshold packet
size, $P_{\text{thresh}}$, used in next hop selection. Two static advanced routing tables are
set for two next hops to be used by the kernel module to route the packets based
on their sizes and $P_{\text{thresh}}$.

We performed experiments explicitly changing the transmission rates on
each link. Loaded only on the source and the destination nodes, PSA module
forwarded packets dependent upon their sizes via the direct single hop link or the
multihop alternative path which had a faster data rate.

We measured UDP and TCP throughputs on a number of scenarios, and
confirmed that the throughput results were consistent with Tables 3.2, 3.3, and
3.4.

We first considered a scenario as in Figure 3.7-b, consisting of a 2 Mb/s
single-hop link and alternative 3-hop path with each link having a rate of 11 Mb/s.
For this scenario $P_{\text{thresh}}$ is equal to 900 bytes as shown in Figure 3.3. PSA modules
resided both at the source and destination and selected minimum delay path for each packet.

Table 3.5 shows that PSA module forwards the packets belonging a UDP flow, which has size of 150, 200, or 250 bytes, to single hop path. Respective end-to-end throughputs for these packets sizes are 628 $kb/s$, 742 $kb/s$, and 853 $kb/s$. This is 50% greater than the throughput of the multihop alternative path. On the other hand, PSA module forwards 1450 byte UDP flow packets through the 3-hop path achieving a throughput close to 1713 $kb/s$, which is 24% more than the corresponding throughput for the single hop link for this packet size.

50% throughput gain for the UDP traffic with a packet size of 200 bytes is significant because that packet size is similar to a VoIP traffic. Using a packet size aware routing, network can support 50% more VoIP calls without negatively effecting the performance of larger size packets. Larger size packets may belong to applications such as web browsing, and file downloading.

PSA module also increases the throughput of a TCP flow by 15%. With the
Figure 3.6: Hardware for implementation. A Soekris 4521 [10] boards with Prism 2.5 chipset based 802.11b wireless cards are seen in the figure. A trimmed version of the Debian Linux with Host-AP driver was used in the implementation.

help of TCPDUMP\(^1\), we confirmed that 40-byte TCP ACK packets traversed the single hop link (2 \(Mb/s\)), whereas 1450-byte data packets traversed the alternative 3-hop path.

There are two potential explanations for the TCP throughput increase:

1. TCP throughput is inversely proportional to RTT and *PSA module* reduces RTT by transmitting ACK packets through a shorter delay path.

2. ACKs are using less channel time, as a result more data packets are allocated to the residual time.

\(^1\)A command line packet analyzer, http://www.tcpdump.org/
Table 3.5: Throughput for 1-hop and 3-hop alternatives for various packet sizes and performance improvement when the module is used 1-hop link is 2 \(\text{Mb/s}\) and all the links on the three hop path is 11 \(\text{Mb/s}\).

<table>
<thead>
<tr>
<th>Routing</th>
<th>TCP</th>
<th>UDP 150 bytes</th>
<th>UDP 200 bytes</th>
<th>UDP 250 bytes</th>
<th>UDP 1250 bytes</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 hop</td>
<td>1275 kb/s</td>
<td>622 kb/s</td>
<td>737 kb/s</td>
<td>841 kb/s</td>
<td>1378 kb/s</td>
</tr>
<tr>
<td>3 hops</td>
<td>1318 kb/s</td>
<td>401 kb/s</td>
<td>487 kb/s</td>
<td>591 kb/s</td>
<td>1730 kb/s</td>
</tr>
<tr>
<td><strong>PSA Module</strong></td>
<td>1490 kb/s</td>
<td>628 kb/s</td>
<td>742 kb/s</td>
<td>853 kb/s</td>
<td>1713 kb/s</td>
</tr>
<tr>
<td>(1 hop)</td>
<td>(3 hops)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Gain | 13% | 57% | 52% | 44% | 24% |

In addition, Tables 3.6 and 3.7 present the results for the same test setup that has different link rates. Similar performance improvements are also observed. We only conducted experiments at most for 3-hop paths.

### 3.2.2 General Solution

As we have showed in the previous sections, packet size is an important parameter for path setup in multi-hop wireless mesh networks. An optimum packet size aware path setup solution may contain multiple paths for the same source-destination node pair.

When the delay/throughput model in (3.1) is considered, cost metric of a link \(l\), \(f_l(r_l, P)\), becomes a function of the size of the packet, \(P\), and the link data rate, \(r_l\). Packet size is bounded by \(L_P \leq P \leq H_P\), where \(L_P\) and \(H_P\) are the lowest and the highest possible packet sizes, respectively. For example \(H_P\) is 1500 bytes for ethernet.

Let \((s, d)\) be the source and destination node pair in a network, and \(P_t(s, d)\) be the set of all loop free paths between node \(s\) to node \(d\). Let \(p\) be a path in \(P_t(s, d)\), i.e. \(p \in P_t(s, d)\), and \(l_p\) be the set of all links belonging to \(p\). \(C_p\) denotes
Table 3.6: Average throughputs when only 1-hop link is used, only 3-hop link is used, or either link is adaptively selected by the packet-size aware (PSA) module. 1-hop link is 1 \( Mb/s \) and transmission rates on 3-hop link are 2, 11, 11 \( Mb/s \). Results for UDP flows with various packet sizes and TCP flows with an MTU of 1500 bytes are listed. *PSA module* selects the link with highest throughput for UDP traffic, therefore selects the best of the alternatives. As for TCP traffic, however, *PSA module* routes data and acknowledgements into different links achieving overall data throughput greater than any of the links.

<table>
<thead>
<tr>
<th>Routing</th>
<th>TCP</th>
<th>UDP 150 bytes</th>
<th>UDP 200 bytes</th>
<th>UDP 250 bytes</th>
<th>UDP 1250 bytes</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-hop link</td>
<td>621 ( kb/s )</td>
<td>391 ( kb/s )</td>
<td>453 ( kb/s )</td>
<td>487 ( kb/s )</td>
<td>672 ( kb/s )</td>
</tr>
<tr>
<td>3-hop link</td>
<td>697 ( kb/s )</td>
<td>303 ( kb/s )</td>
<td>371 ( kb/s )</td>
<td>422 ( kb/s )</td>
<td>833 ( kb/s )</td>
</tr>
<tr>
<td><em>PSA module</em></td>
<td>740 ( kb/s )</td>
<td>388 ( kb/s )</td>
<td>449 ( kb/s )</td>
<td>490 ( kb/s )</td>
<td>857 ( kb/s )</td>
</tr>
<tr>
<td>Gain</td>
<td>6%</td>
<td>28%</td>
<td>21%</td>
<td>16%</td>
<td>28%</td>
</tr>
</tbody>
</table>

the cost of path \( p \) and is given by

\[
C_p = \sum_{l \in l_p} (f_l(r_l, P)) \tag{3.5}
\]

Thus, the general path-setup problem that selects the minimum cost path \( \hat{p} \) between node \( s \) and \( d \) can be formulated as follows:

\[
\hat{p} = \arg\min_{p \in P_{s,d}} C_p \tag{3.6}
\]

The solution to this optimization is complex and is not addressed in this thesis. Instead we will simplify (3.6) using experimental packet statistics, and provide solutions to the simplified version by modifying the classical path setup schemes.

There are a number of studies on the distribution of packet sizes in real networks [13, 6, 11]. All these studies observe similar cumulative distribution
Table 3.7: Average throughputs when only 1-hop link is used, only 2-hop link is used, or either link is adaptively selected by the packet-size aware (PSA) module. 1-hop link is 2 $Mb/s$ and transmission rates on 2-hop link are both 5.5 $Mb/s$. Results for UDP flows with various packet sizes and TCP flows with an MTU of 1500 bytes are listed. *PSA module* selects the link with highest throughput for UDP traffic, therefore selects the best of the alternatives. As for TCP traffic, however, *PSA module* routes data and acknowledgements into different links achieving overall data throughput greater than any of the links.

<table>
<thead>
<tr>
<th>Routing</th>
<th>TCP</th>
<th>UDP 150 bytes</th>
<th>UDP 200 bytes</th>
<th>UDP 250 bytes</th>
<th>UDP 1250 bytes</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-hop link</td>
<td>1250 kb/s</td>
<td>635 kb/s</td>
<td>762 kb/s</td>
<td>853 kb/s</td>
<td>1730 kb/s</td>
</tr>
<tr>
<td>2-hop link</td>
<td>1420 kb/s</td>
<td>541 kb/s</td>
<td>662 kb/s</td>
<td>773 kb/s</td>
<td>1720 kb/s</td>
</tr>
<tr>
<td><em>PSA module</em></td>
<td>1460 kb/s</td>
<td>638 kb/s</td>
<td>755 kb/s</td>
<td>849 kb/s</td>
<td>1730 kb/s</td>
</tr>
<tr>
<td>Gain</td>
<td>3%</td>
<td>18%</td>
<td>14%</td>
<td>10%</td>
<td>22%</td>
</tr>
</tbody>
</table>

functions for packet size statistics: 10-25% are between 1400 and 1500 bytes, 10-20% are close to 500 bytes, and the remainder is less than 500 bytes (majority is close to 40 bytes).

Furthermore, we collected packet statistics using Skype\(^2\) by making video and voice calls. We observed that 51.5% of the packets are between 80 and 159 bytes and 44.4% of the packets have packet sizes between 150 and 359 bytes. Growing interest into similar applications will likely change the overall packet size statistics in the Internet.

In the light of the statistical data from the previous research, we grouped packets into three bins as follows

1. Between 1000 and 1500 bytes
2. Between 500 and 1000 bytes
3. Less than 500 bytes

\(^2\)A software application to make VoIP calls, http://skype.com
Statistical average of packets sizes are computed for each bin. Using the distributions in [13, 6, 11], we calculated the average packet sizes for each bin as 1400, 600, and 200 bytes.

Classical shortest path algorithms [5] can be extended to have a vector cost metric instead of a generic scalar cost metric. For example, a link state routing protocol can utilize a vector cost metric, with each entry set proportional to transmission delay of the link given by (3.1) corresponding to average packet sizes for each bin.

An off the shelf path setup engine can be used to obtain link cost vector such that for each bin a possibly different next hop can be assigned. This can also be considered as the vector metric extension of MTM [4].

Spanning tree protocols [23] are commonly used for single gateway multihop wireless access networks [24]. A packet size aware spanning tree can be implemented in the form of a multiple spanning tree formation such that the link costs of each spanning tree are set proportional to the link transmission delay for a particular packet size (e.g. 1400, 600 or 200 bytes). When a node receives a packet, it classifies it into a bin based on its packet size and forwards it to the corresponding spanning trees.

### 3.3 Remarks

We see a broad range of packet sizes inside access networks due to variety of applications that has different delay and throughput requirements. For example, VoIP type of applications utilize smaller packets (as smalls as 200 bytes) due to stringent end-to-end delay requirements and are becoming increasingly popular. File downloading and web browsing type of applications, on the hand, utilize larger packets (usually close to 1500 bytes) to maximize their throughput.

In this chapter we showed the benefits of a packet size aware routing in a multihop wireless network. These types of networks contain a combination of realtime and non-realtime applications. Considering the packet size distributions, we claimed that packet size aware routing will potentially enhance the overall
network performance. It is usually inefficient to optimize routing within a network solely with respect to the highest packet size (e.g. 1500 bytes), which represents less than 25% of the overall traffic.

This chapter, in full, is a reprint of the material as it appears in proceedings of IEEE Consumer Communications & Networking Conference 2008. Arisoylu, Mustafa; Ergür, Salih; Cruz, Rene L.; Rao, Ramesh R. The dissertation author and Mr. Arisoylu were primary investigators and authors of this paper with equal contributions.
Figure 3.7: Experiments were conducted to measure average throughputs by explicitly setting various transmission rates for the above configurations. *PSA module* resides on the source and destination routers.
Bibliography


Part II

Localization for Cellular and Sensor Networks
4

Wireless Positioning Overview

Wireless positioning concerns with the position estimation of a target node with the help of reference nodes.

Positioning approaches can be classified as active vs. passive; target to be located actively cooperates with the reference nodes in the active approach, and reference nodes infer the location of a target node from the disturbances in the exchanged signals between each other caused by the target in the environment in the passive approach [3]. Following sections briefly summarizes the classification of positioning algorithms as discussed in [2].

4.1 Active positioning

Active approaches are further classified as client-based (self-positioning) vs. network-based (remote positioning). In client-based approaches, target itself deduces its location from the received signals from the reference nodes. In network-based approaches, on the hand, a central data unit processes the received information from the reference nodes and estimates the location of a target [2].

Two different positioning schemes are used in position estimation: direct and two-step positioning.
4.1.1 Direct positioning

Direct positioning schemes directly use the received signal in position estimation, therefore requires the received signal to be available at the central unit for network-based algorithms [1]. Direct schemes provide a better position estimation accuracy compared to two-step positioning schemes at the expense of increased complexity.

4.1.2 Two-step positioning

In two-step positioning schemes, certain signal parameter estimation, such as time of arrival (TOA), angle of arrival (AOA), and received signal strength (RSS), is followed by position estimation using these parameters; hence estimated parameters in the first step are transmitted to the central unit. [1]. Details of various two-step positioning schemes can be found in [2], which is summarized here.

First-step: Parameter estimation

- **Received Signal Strength (RSS):** Received power, or received signal strength (RSS), of a signal at the receiver contains information related to the distance from the transmitter. Taking into account of path-loss and shadowing models for the environment, RSS can be used to determine the target location.

- **Time of Arrival (TOA):** Flight time from a transmitter to a receiver is closely related with the distance between them. A single TOA measurement locates a target on a circle.

- **Angle of Arrival (AOA):** The angle between the transmitter and the receiver locates a target on a line. Antenna arrays or directional antennas are used in determining the angle.

- **Time Difference of Arrival (TDOA):** The difference between the arrival times of two signals travelling between the target node and two reference nodes is used in
Second-step: Position estimation

Fingerprinting techniques correlate a database of estimated parameters for known locations to the estimated parameters from a target.

Geometric techniques and Statistical techniques exploit the geometrical and statistical relationships between the target and sensors.

4.2 Passive positioning

Passive positioning algorithms deal with locating an uncooperative target from the disturbances in the multipath profile of a received signal. Time difference of arrival of the direct path and the reflection path reduces the ambiguity of target to an ellipse.
Bibliography


Localization via Multipath Strengths in a CDMA2000 Cellular Network Using Neural Networks

Locating a mobile phone has become very popular recently. As the number of cell phone users increases tremendously year by year, the percent of emergency calls made from them also increases. As opposed to landline phones, these phones do not have a fixed location. U.S. Federal Communications Commission (FCC) mandates that all operators provide precise location for cell phone users that make emergency calls.

Emergency is not the only factor for the interest in localization. A lot of new applications using location based services (LBS) such as fleet tracking, navigational assistance, finding local businesses, road side assistance, and highway traffic monitoring are emerging in the market.

Location is one of the essential elements of context-aware computing. Devices can react to the changes in their environments.

Network operators can also benefit from position information of the mobile user. For example, mobiles can assist handoffs by reporting their location and
network planning and optimization can be enhanced based on popular locations of users.

Even though GPS provides an accurate localization for outdoors, not all the phones are equipped with a GPS receiver and its performance severely degrades in urban settings. This leads to new type of services: Google [6] introduced map service for mobile phones that estimates their locations in the absence of GPS. Navizon [13] is another service that makes it possible to locate mobile phones that are not equipped with GPS.

The fundamental methods for position location using wireless networks, which are Time of Arrival (TOA), Time Difference of Arrival (TDOA), Angle of Arrival (AOA), are explained in [14]. The same methods are presented in a GSM network in [11]. A recent survey on position location techniques for indoor and outdoor wireless networks is presented in [7], where the applications of the fundamental methods for GSM and CDMA networks are summarized. This survey also includes hybrid methods, GSM assisted algorithms, and fingerprinting for cellular networks.

A fingerprinting algorithm in [9] determines a point in the area of interest by fitting the measured signal strength of certain base stations to the pre-recorded average signal strength in this area according to a least squares criterion. Another fingerprinting algorithm and simulation results for this algorithm are presented in [1] for Universal Mobile Telecommunications Systems (UMTS). This algorithm uses multipath profiles for pattern matching.

A theoretical basis is developed for localization by fingerprinting of indoor positioning systems in [10]. This basis is used to determine the accuracy of any positioning system given a set of system parameters such as number of access points, grid spacing, path loss component, and standard deviation of received signal strength and radio propagation characteristics. A passive radar approach using neural networks (NN) in an UWB sensor network for localization, which is most suitable for indoors, is proposed in [3].

Another algorithm for position location with training and Kalman-Bucy filter smoothing is proposed in [5].
The results of a practical work, which compares several position location algorithms applied to collected real data in GSM networks, are presented in [2].

In this chapter we propose a position location algorithm for cellular phones that uses multipath strengths with NN. We evaluate our algorithm against real data collected in a commercial CDMA2000 1xRTT network operated by Verizon Wireless [16]. The NN is trained by strengths of pilots and multipath arrivals and then independent test data are run through the NN. We use a simple averaging window as a smoother at the output of the NN to reduce fluctuations and enhance the performance.

The chapter is organized as follows: Section 5.1 gives an overview of CDMA air-interface that is relevant to our research. Section 5.2 describes the NN that we use in our work. Section 5.3 explains the methodology that we follow. Section 5.4 presents the experimental setup and discusses our results. Section 5.5 concludes the chapter.

5.1 Cellular Network Overview

Cellular networks are composed of cells, and each cell is divided into sectors. In CDMA2000 networks, sectors transmit pilots that are differentiated from each other by PN sequences.

CDMA2000 networks employ direct sequence spread spectrum (DSSS) over 1.222 MHz bandwidth. Due to its wideband nature, multipath arrivals reflected from different objects that are not very close and small in size can be distinguished. Rake receivers are used to exploit the time diversity by resolving the delayed multipath arrivals and adding them using maximal ratio combiner. As opposed to narrowband systems such as GSM, multipath does not hurt but rather help to improve the performance.

In a smaller scale, multipaths tend to change quite frequently. However by averaging the received samples over an interval, fast fading effects are smoothed and scatters from stationary objects such as the buildings nearby become more significant.
5.1.1 Propagation Model

In [15] received signal strength for cellular communication system as a function of distance is given as

\[ P_r = K \frac{P_t}{d^n} \]  \hspace{1cm} (5.1)

where \( P_t \) and \( P_r \) are transmit and receive powers respectively, \( d \) is the distance, \( n \) is the power factor, and \( K \) is a constant to factor out attenuation due to antenna gains and heights.

In the dB scale (5.1) becomes linear as follows:

\[ 10 \log P_r = 10 \log K + 10 \log P_t - 10n \log d \]  \hspace{1cm} (5.2)

5.2 Neural Network Model

Neural networks (NN) are non-algorithmic methods, which use parallel computing technique. They imitate functioning of the brain. Even though inter-neuron communication speed is quite slow for the brain, parallel processing allows it to analyze very complicated data in a short period of time. Neural networks learn directly from current examples rather than programming [8].

Feed forward neural networks with multiple hidden layers have been widely used and showed to operate successfully (see Figure 5.1). Multi Layer Perceptron (MLP) learning algorithm is used in the training of the network. MLP is a back propagation algorithm and it computes the error at the output of the network and sets weights of neurons iteratively. This operation is spread out on all layers and the error in the output is reduced. Deviations between the real and the predicted values are computed to evaluate the learning success of the network.

Even though the performance of NN generally increases with the addition of more neurons to the hidden layer, using more than necessary hidden units have certain side effects. Hence starting with small number and gradually incrementing the number of hidden units until a satisfactory performance is achieved is a good practice.
A common problem with NN training is overfitting. Usually it arises from using too many data points during the training phase. NN learns the noise specific to the training set, therefore performs poorly when applied to a different data set. One way to overcome this issue is keeping two sets of data; one for training and one for testing. Once the error of the test data starts rising training should be stopped.

Mean square error (MSE) is used to determine the compliance between the predicted output and computed network output. The exit criterion for the supervised learning is set on the value of MSE (e.g. when MSE is below 0.001).

After successful termination of the learning process, the classification performance is determined by applying test data to the neural network. If the performance values meet the desired criteria at the end of the test, the structure of the neural network is completed and it is ready to classify any external data.

5.3 Methodology

Each location has a unique multipath profile because of its surroundings. In this study, we test drive in a region and collect the pilot strengths at equal time intervals and extract the multipath profile for each interval. We may receive pilots from multiple sectors and their delayed versions at each interval. The strengths of pilots and their delayed arrivals are first averaged to reduce the fast fading effect.
and then used for training a neural network. The same network is then applied to an independent test data for performance evaluation.

The complexity of the neural network grows non-linearly with each distinct pilot. As the size of the drive test region increases, the number of pilots observed in that region also increases. Therefore we need to limit the size of the region.

Finally, the estimated locations at the NN output is passed through a smoother to reduce the fluctuations and enhance the overall performance. We use a simple averaging window approach. More complicated algorithms such as Kalman filters can also be used.

## 5.4 Experimental Results

### 5.4.1 Setup

Our experimental setup consists of a laptop with a Verizon Wireless PC 5220 CDMA2000 data card that is connected through PCMCIA slot. A Garmin GPS receiver is connected to the serial port. We used Ericsson TEMS CDMA Investigator software to log cellular air-interface data from the control port of the PC card over the Verizon Wireless CDMA2000 1xRTT network and location information from the GPS receiver. [4]

We conducted drive tests along the same area consecutively for a week. Six days of data is used for training and the other remaining one is used to test our algorithm. Drive test area (see Figure 5.6) is a combination of interstate highway and city streets.
We observed 25 distinct pilots in the whole area. In order to reduce the complexity of NN, we grouped pilots that are seen in close neighborhoods and created 3 regions (see Figure 5.6 and Figure 5.3). There are 10, 6, and 11 pilots at region 1, 2, and 3, respectively. Note that one pilot was shared between region 1 and 2, and another between 2 and 3.

Received pilot strengths are averaged over half a second interval to minimize the fast fading effects. NN output is averaged over 15 samples to smooth the location estimation.

MATLAB Neural Network Toolbox [12] is used to implement the neural network. Increasing the number of inputs causes the complexity of the network to increase too. Data set revealed that at any time observing four multipath arrivals of the same pilot is very rare (less than 2

Location estimations are compared to the GPS recorded actual positions to evaluate the accuracy of the system.

5.4.2 Results

Figure 5.4 shows the cumulative distribution (CDF) of the error for each region. As it can be seen from the figure, the first and the third regions exhibit similar behavior since they both have similar number of pilots. The second region, on the hand, performs worse on the average since it has only 6 pilots that covers a larger area. Having more pilots per unit area provides better accuracy.

Note that the regions 1 and 3 both resemble to each other: they both contain an L-shaped route, some portion in the highway while the other part in the street.

In Figure 5.3, both neural network output and averaged output are displayed on top of the original drive test route. We observe that the simple averaging improves the performance on the average more than 10% (see Figure 5.5 and Table 5.1).
Figure 5.3: Performance of the algorithm: NN estimates on the left and averaged output on the right.
Table 5.1: Localization accuracy of NN and smoother

<table>
<thead>
<tr>
<th>Percentile</th>
<th>67%</th>
<th>90%</th>
</tr>
</thead>
<tbody>
<tr>
<td>NN</td>
<td>217.93m</td>
<td>378.82m</td>
</tr>
<tr>
<td>Smoother</td>
<td>193.52m</td>
<td>325.94m</td>
</tr>
<tr>
<td>Improvement</td>
<td>11.20%</td>
<td>13.96%</td>
</tr>
</tbody>
</table>

Figure 5.4: CDF of estimation error for different regions

5.5 Remarks

We proposed a neural network based algorithm that exploits the multipath strengths in CDMA2000 cellular network for locating a mobile user. We collected air-interface logs from a commercial network by doing drive tests in a preselected area, which was composed of an interstate highway and city streets. We trained our neural network using logged multipath strengths. We evaluated the performance of our algorithm with independent set of data.

Our results indicated that we can locate the mobile within 217.93m 67% of the time, and within 378.82m 90% of the time. Using a simple averaging window
Figure 5.5: Effect of averaging the estimated location. Overall CDF of estimation errors from all regions are plotted. *Averaged Output* corresponds to averaging the position estimation over 15 samples.
Figure 5.6: Drive test was conducted in San Diego, CA. Whole area is divided into 3 regions.

smoother we improve the accuracy to 193.52m and 325.94m for 67-th and 90-th percentile, respectively.

This chapter, in full, is a reprint of the material as it appears in proceedings of IEEE World Conference on Computational Intelligence 2008. Ergüt, Salih; Rao, Ramesh R.; Dural, Özgür. The dissertation author was the primary investigator and author of this paper.
Bibliography


6

Linear Least Squares Techniques for Active and Passive Localization

6.1 Active Localization

Time of arrival (TOA) measurements between a mobile device and multiple stationary sensor nodes is commonly used in device localization. Each TOA measurement with a sensor locates the device on a circle and intersection of three or more such circles gives the device location. Due to imperfections in TOA measurements, all circles do not intersect at a single location; linear and nonlinear least-squares (LS) techniques, therefore, are used to estimate the device position under such ambiguity. Even though nonlinear methods provide more accurate results compared with linear methods, linear methods are usually preferred due to their reduced complexity. In this chapter we propose two novel linear LS techniques and compare their performances to previous linear LS techniques and Cramer-Rao lower bounds (CRLB). We also provide simulations to evaluate the performances.

Location based services are gaining increased popularity due to potential applications for cellular networks such as location sensitive billing, enhanced 911 service to report location for emergency calls from a cellphone, fraud detection,
cellular system design and resource management [1]; and for short range networks such as interactive toys, smart badges, home automation, intruder detection, fleet management, intelligent transport systems, and patient monitoring [5, 3]. A crude location information is sufficient for some of these applications, while others require more accurate position estimation.

Trilateration with TOA measurements, travel time of a radio signal between two nodes, is a commonly used technique for position estimation. Non-linear approaches to solve the position problem are complex and costly, hence reduced complexity linear least squares (LS) techniques that provide comparable accuracy to their non-linear counterparts are usually preferred. A detailed survey of signal processing techniques for wireless localization can be found in [9, 11, 3].

[2] showed that linear LS techniques requires the number of equations to be at least one more than the number of unknowns and either an arbitrary equation or the average of all equations can be selected as the reference. [8] studied CRLB for a sensor network consisting of nodes with known and unknown locations. Unknown sensor locations are estimated using TOA or received signal strength (RSS) from known sensors. [6] extended the CRLB for TOA-based localization algorithms where the noise variance is considered as a function of range estimate and physical layer.

[4] investigated the performance of three linear least-squares (L-LS) approaches: L-LS-1, L-LS-2, L-LS-3 (see Section 6.1.3). In this chapter we propose two novel L-LS approaches that takes into account the target location in reference node selection for linear LS equations and show that they perform better than the ones discussed here.

6.1.1 System Model

Consider a wireless network same as [4] consisting of $N$ sensor nodes, located at $s_i = [x_i, y_i]^T$, $i = 1, \ldots, N$. Position of a target with an active tag located at $t = [x, y]^T$ using time of arrival (TOA) measurements from $N$ sensor nodes to the target is to be estimated. Distance estimate, $r_i$, from the $i$-th sensor node to the
Figure 6.1: TOA based localization. Nodes $s_1$, $s_2$, and $s_3$ communicates with target $t$ and measure the distances $r_1$, $r_2$, and $r_3$ which allows them locate the target on a circle. However due to noise these circles intersect at multiple locations. Light color bands around these circles indicate the uncertainty due to noise.

target, is equal to

$$r_i = c\tau_i = d_i(x, y) + n_i$$  \hspace{1cm} (6.1)$$

where $c$ is the speed of light, $\tau_i$ denotes TOA estimate, $n_i$ is the noise, and $d_i(x, y)$ is the true distance between the target ($t$) and $i$-th sensor node ($s_i$), given by

$$d_i(x, y) = \sqrt{(x - x_i)^2 + (y - y_i)^2}$$  \hspace{1cm} (6.2)$$

In this chapter, we will assume Gaussian noise, which suits well for line-of-sight (LOS) conditions. Noise distribution is given by

$$n_i \sim \mathcal{N}(0, \sigma_i^2)$$  \hspace{1cm} (6.3)$$
6.1.2 Nonlinear LS Estimation

Given a set of TOA measurements, nonlinear LS estimator calculates the target position as follows:

$$\hat{t} = \underset{(x,y)}{\arg\min} \sum_{i=1}^{N} \beta_i (r_i - d_i(x,y))^2$$

(6.4)

where $\beta_i$ is the weighting coefficient as an indicator of reliability of TOA measurement from $i$-th sensor node [4]. Gradient descent and linearization via Taylor series expansion are commonly used to implement nonlinear LS estimation given by (6.4).

Cramer-Rao Lower Bound

Cramer-Rao lower bound (CRLB) states the inverse of the Fisher Information Matrix (FIM) as the lower bound for the variance of any unbiased estimator. A solution that achieves CRLB has the lowest mean square error (MSE) among all unbiased estimators.

Let $\theta = [\theta_1, \theta_2, \ldots, \theta_d]^T$ be set of parameters to be estimated with conditional probability density function (pdf) $f(x|\theta)$. Then the $i$-th and $j$-th element of $d \times d$ FIM is given by[12]

$$I_{ij} = E \left[ \frac{\partial}{\partial \theta_i} \ln f(x|\theta) \frac{\partial}{\partial \theta_j} \ln f(x|\theta) \right]$$

Mean squared error (MSE) of any unbiased estimator is given by [10]

$$MSE = E \left\{ \| \hat{t} - t \| \right\} \geq \text{trace} \left\{ I^{-1} \right\}$$

Therefore for two-variate distributions

$$MSE \geq \text{trace} \left\{ I^{-1} \right\} = \frac{I_{11} + I_{22}}{I_{11}I_{22} - I_{12}^2}$$

Theorem 2. Fisher information matrix (FIM) for an unbiased nonlinear estimator for the system modeled as (6.1) is given by

$$I = \begin{bmatrix}
\sum_{i=1}^{N} \frac{(x-x_i)^2}{\sigma_i^2 d_i^2(x,y)} & \sum_{i=1}^{N} \frac{(x-x_i)(y-y_i)}{\sigma_i^2 d_i^2(x,y)} \\
\sum_{i=1}^{N} \frac{(x-x_i)(y-y_i)}{\sigma_i^2 d_i^2(x,y)} & \sum_{i=1}^{N} \frac{(y-y_i)^2}{\sigma_i^2 d_i^2(x,y)}
\end{bmatrix}$$
where the noise term \( n_i \) in (6.1) is modeled as Gaussian with zero mean and variance \( \sigma_i^2 \), i.e. \( n_i \sim \mathcal{N}(0, \sigma_i^2) \).

**Proof.** Conditional pdf of \( \mathbf{r} \) given \( \mathbf{t} = [x, y]^T \) is

\[
\mathbf{r} \mid \mathbf{t} \sim \mathcal{N}(d_i(x, y), \sigma_i^2)
\]

with

\[
p(\mathbf{r} | \mathbf{t}) = \prod_{i=1}^{N} \frac{1}{\sqrt{2\pi\sigma_i}} e^{- \frac{(r_i - d_i(x, y))^2}{2\sigma_i^2}}
\]

Taking the logarithm, we obtain

\[
\ln p(\mathbf{r} | \mathbf{t}) = - \sum_{i=1}^{N} \frac{(r_i - d_i(x, y))^2}{2\sigma_i^2} + K
\]

where \( K = \frac{N}{2} \ln(2\pi) + \sum_{i=1}^{N} \ln(\sigma_i) \). Partial derivatives of log conditional pdf with respect to \( x \) and \( y \) is computed as

\[
\frac{\partial}{\partial x} \ln p(\mathbf{r} | \mathbf{t}) = - \sum_{i=1}^{N} \frac{(r_i - d_i(x, y))}{\sigma_i^2} \frac{\partial d_i(x, y)}{\partial x}
\]

\[
\frac{\partial^2}{\partial x^2} \ln p(\mathbf{r} | \mathbf{t}) = - \sum_{i=1}^{N} \frac{1}{\sigma_i^2} \left( - \left( \frac{\partial d_i(x, y)}{\partial x} \right)^2 + (r_i - d_i(x, y)) \frac{\partial^2 d_i(x, y)}{\partial x^2} \right)
\]

\[
\frac{\partial^2}{\partial x \partial y} \ln p(\mathbf{r} | \mathbf{t}) = - \sum_{i=1}^{N} \frac{1}{\sigma_i^2} \left( - \frac{\partial d_i(x, y)}{\partial y} \frac{\partial d_i(x, y)}{\partial x} (r_i - d_i(x, y)) \frac{\partial^2 d_i(x, y)}{\partial x \partial y} \right)
\]

and similarly

\[
\frac{\partial^2}{\partial y^2} \ln p(\mathbf{r} | \mathbf{t}) = - \sum_{i=1}^{N} \frac{1}{\sigma_i^2} \left( - \left( \frac{\partial d_i(x, y)}{\partial y} \right)^2 + (r_i - d_i(x, y)) \frac{\partial^2 d_i(x, y)}{\partial y^2} \right)
\]

where

\[
\frac{\partial d_i(x, y)}{\partial x} = \frac{x - x_i}{d_i(x, y)}
\]

\[
\frac{\partial d_i(x, y)}{\partial y} = \frac{y - y_i}{d_i(x, y)}
\]
Since $E[r_i] = d_i(x, y)$, FIM becomes

\[
I = \begin{bmatrix}
E\left[\frac{\partial^2}{\partial x^2} \ln p(r|t)\right] & E\left[\frac{\partial^2}{\partial x \partial y} \ln p(r|t)\right] \\
E\left[\frac{\partial^2}{\partial y \partial x} \ln p(r|t)\right] & E\left[\frac{\partial^2}{\partial y^2} \ln p(r|t)\right]
\end{bmatrix}
\]

\[
= \begin{bmatrix}
\sum_{i=1}^{N} \frac{1}{\sigma_i^2} d_i(x, y) & \sum_{i=1}^{N} \frac{1}{\sigma_i^2} \frac{\partial}{\partial y} d_i(x, y) \\
\sum_{i=1}^{N} \frac{1}{\sigma_i^2} \frac{\partial}{\partial x} d_i(x, y) & \sum_{i=1}^{N} \frac{1}{\sigma_i^2} \left(\frac{\partial}{\partial y} d_i(x, y)\right)^2
\end{bmatrix}
\]

\[
= \begin{bmatrix}
\sum_{i=1}^{N} \frac{(x-x_i)^2}{\sigma_i^2 d_i^2(x, y)} & \sum_{i=1}^{N} \frac{(x-x_i)(y-y_i)}{\sigma_i^2 d_i^2(x, y)} \\
\sum_{i=1}^{N} \frac{(x-x_i)(y-y_i)}{\sigma_i^2 d_i^2(x, y)} & \sum_{i=1}^{N} \frac{(y-y_i)^2}{\sigma_i^2 d_i^2(x, y)}
\end{bmatrix}
\]

\[
6.1.3 \text{ Linear LS Estimation}
\]

Linear equations are achieved by subtracting the square of the range measurements from reference equations. Square of distance estimate from TOA measurement from node $i$ is given by

\[
r_i^2 = d_i^2(x, y) + 2d_i(x, y)n_i + n_i^2
\]

\[
= (x - x_i)^2 + (y - y_i)^2 + 2(r_i - n_i)n_i + n_i^2
\]

\[
= (x - x_i)^2 + (y - y_i)^2 + 2r_i n_i - n_i^2
\]

\[
= x^2 + y^2 - 2x_i x - 2y_i y + k_i^2 + 2r_i n_i - n_i^2 \tag{6.5}
\]

where $k_i^2 = x_i^2 + y_i^2$ and second step substitutes $d_i(x, y)$ using (6.1). A reference equation is selected by using either a single distance estimate or a linear combination of multiple distance estimates from TOA measurements. Therefore each technique to estimate target location, $t$, subtracts the square of a reference equation of the form

\[
r_f^2 = x^2 + y^2 - 2x_f x - 2y_f y + 2r_f n_f - n_f^2 \tag{6.6}
\]

from remaining set of equations to obtain new linear set of equations as follows

\[
r_i^2 - r_f^2 = 2(x_f - x_i)x + 2(y_f - y_i)y + k_i^2 - k_f^2
\]

\[
+ 2(r_i n_i - r_f n_f) - n_i^2 + n_f^2
\]
or

\[ 2(x_f - x_i)x + 2(y_f - y_i)y + \tilde{n}_i = r_i^2 - r_f^2 - k_i^2 + k_f^2 \]  \hspace{1cm} (6.7)

where

\[ \tilde{n}_i = 2(r_i n_i - r_f n_f) + n_i^2 - n_f^2 \]  \hspace{1cm} (6.8)

For small noise variances \( n_i^2 - n_f^2 \) can be modeled as Gaussian [4]. (6.7) can be rewritten in the matrix form as:

\[ A\mathbf{t} + \tilde{\mathbf{n}} = \mathbf{p} \]  \hspace{1cm} (6.9)

with \( \mathbf{t} = [x, y]^T \) and \( i \)-th row of \( A, \mathbf{p}, \) and \( \tilde{\mathbf{n}} \) are given by

\[
\begin{align*}
\alpha_i &= 2[(x_f - x_i), (y_f - y_i)] \\
p_i &= r_i^2 - r_f^2 - k_i^2 + k_f^2 \\
\tilde{n}_i &= (6.10)
\end{align*}
\]

for \( i = 1, \ldots, N \), respectively. Then the LS estimate of the location becomes

\[ \hat{\mathbf{t}} = T_A \mathbf{p} \]  \hspace{1cm} (6.11)

where

\[ T_A = (A^T A)^{-1} A^T \]  \hspace{1cm} (6.12)

**Theorem 3.** If \( X \) is a zero mean Gaussian random variable with variance \( \sigma_X^2 \) and \( Y = X^2 \), then characteristic function of \( Y \) is \( \Phi_Y(\omega) = (1 - 2j\sigma_X^2\omega)^{-1/2} \)

**Proof.** Let \( f_X(x) \) be the probability density function (pdf) of some random variable \( X \), then pdf of \( Y \) becomes [7]

\[ f_Y(y) = \frac{1}{2\sqrt{y}} (f_X(\sqrt{y}) + f_X(-\sqrt{y}) \]  \hspace{1cm} (6.13)

Since \( X \) is Gaussian with \( X \sim \mathcal{N}(0, \sigma_X^2) \),

\[ f_X(x) = \frac{1}{\sqrt{2\pi\sigma_X^2}} e^{-x^2/2\sigma_X^2} \]  \hspace{1cm} (6.14)
Substituting (6.14) in (6.13), pdf of $Y$ becomes

$$f_Y(y) = \frac{1}{\sqrt{2\pi y\sigma_x}} e^{-y/2\sigma^2_y}, \quad y \geq 0$$

Characteristic function for $Y$ is

$$\Phi_Y(\omega) = \int_0^\infty f_Y(y)e^{-j\omega y}dy$$

$$= \int_0^\infty \frac{1}{\sqrt{2\pi y\sigma_x}} e^{-y/2\sigma^2_y} e^{-j\omega y}dy$$

$$= \frac{1}{\sqrt{2\pi \sigma_x}} \int_0^\infty \frac{1}{\sqrt{y}} e^{-y\left(\frac{1}{2\sigma^2_x} - j\omega\right)}dy$$

Let $y' = \sqrt{y}$ and $2\sigma^2 = \left(\frac{1}{2\sigma^2_x} - j\omega\right)^{-1}$. Then $dy' = \frac{1}{2\sqrt{y}}dy$, $\sigma' = \frac{1}{\sqrt{2}} \left(\frac{1}{2\sigma^2_x} - j\omega\right)^{-1/2}$, and

$$\Phi_Y(\omega) = \frac{2}{\sqrt{2\pi \sigma_x}} \int_0^\infty e^{-\frac{y'^2}{2\sigma'^2}} dy'$$

$$= \frac{2\sigma'}{\sigma_x} \left[ \frac{1}{\sqrt{2\pi \sigma'}} e^{-\frac{y'^2}{2\sigma'^2}} \right]_{-1/2}$$

$$= \frac{\sigma'}{\sigma_x}$$

$$= \frac{1}{\sqrt{2\sigma_x}} \left[ \frac{1}{\left(\frac{1}{2\sigma^2_x} - j\omega\right)^{1/2}} \right]$$

$$= \left(1 - 2j\sigma^2_x\omega\right)^{-1/2}$$

\[\square\]

**Corollary 1.** If $X$ is a zero mean Gaussian random variable with variance $\sigma^2_x$ and $Y = X^2$, then mean and variance of $Y$ is $\mu_y = \sigma^2_x$ and $\sigma^2_y = \sigma^2_x (3\sigma^2_x - 1)$

**Proof.** Using the characteristic function in Theorem 3, first moment (or the mean) of $Y$ can be computed as

$$E[Y] = \frac{1}{j} \frac{d\Phi(\omega)}{d\omega} \bigg|_{\omega=0}$$

$$= \sigma^2_x \left(1 - 2j\sigma^2_x\omega\right)^{-3/2} \bigg|_{\omega=0}$$

$$= \sigma^2_x$$
Similarly second moment becomes
\[
E[Y^2] = \frac{1}{j^2} \left. \frac{d^2 \Phi(\omega)}{d\omega^2} \right|_{\omega=0} \\
= 3\sigma_x^4 (1 - 2j\sigma_x^2 \omega)^{-5/2} \bigg|_{\omega=0} \\
= 3\sigma_x^4
\]
hence variance of \(Y\) is
\[
\sigma_Y^2 = E[Y^2] - E^2[Y] = \sigma_x^2 (3\sigma_x^2 - 1)
\]
\[\square\]

**Theorem 4.** If \(X\) is a zero mean Gaussian random variable with variance \(\sigma_x^2\), i.e. \(X \sim \mathcal{N}(0, \sigma_x^2)\), then \(E[X^3] = 0\).

**Proof.**
\[
E[X^3] = \int_{-\infty}^{\infty} x^3 \frac{1}{\sqrt{2\pi}\sigma_x} e^{-\frac{x^2}{2\sigma_x^2}} dx \\
= \int_{-\infty}^{0} x^3 \frac{1}{\sqrt{2\pi}\sigma_x} e^{-\frac{x^2}{2\sigma_x^2}} dx + \int_{0}^{\infty} x^3 \frac{1}{\sqrt{2\pi}\sigma_x} e^{-\frac{x^2}{2\sigma_x^2}} dx
\]
Substituting \(x' = -x\) in the first integral, with \(dx' = -dx\), expectation becomes
\[
E[X^3] = -\int_{0}^{\infty} x^3 \frac{1}{\sqrt{2\pi}\sigma_x} e^{-\frac{x'^2}{2\sigma_x^2}} dx' + \int_{0}^{\infty} x^3 \frac{1}{\sqrt{2\pi}\sigma_x} e^{-\frac{x^2}{2\sigma_x^2}} dx
\]
\[\square\]

**Corollary 2.** If \(X\) is a zero mean Gaussian random variable with variance \(\sigma_x^2\) and \(Y = X^2\), then \(X\) and \(Y\) are uncorrelated.

**Proof.** The covariance of \(X\) and \(Y\) is
\[
\text{Cov}(X,Y) = E[(X - \mu_x) (Y - \mu_y)] \\
= E[X (Y - \mu_y)] \\
= E[XY] - \mu_y E[X] \\
= E[X^3] \\
= 0
\]
since $E[X] = 0$ and $E[X^3] = 0$ (Theorem 4). Therefore $X$ and $Y$ are uncorrelated.

**Corollary 3.** If $X$ is a zero mean Gaussian random variable with variance $\sigma^2_x$, $Y = X^2$ with variance $\sigma^2_y$, and $Z = aX + bY$ with variance $\sigma^2_z$ for some $a, b \in \mathbb{R}$, then $\sigma^2_z = a^2\sigma^2_x + b^2\sigma^2_y$.

**Proof.**

\[
\sigma^2_z = a^2\sigma^2_x + b^2\sigma^2_y + \text{Cov}(aX, bY) \\
= a^2\sigma^2_x + b^2\sigma^2_y + ab\text{Cov}(X, Y) \\
= a^2\sigma^2_x + b^2\sigma^2_y
\]

since $X$ and $Y$ are uncorrelated (Corollary 2) and their covariance is zero.

**Linear LS Techniques**

In this section we will summarize the three linear LS approaches (L-LS-1, L-LS-2, L-LS-3) discussed in [4], and introduce two new linear LS approaches, L-LS-A and L-LS-M to estimate target location.

Following is the list of reference equations similar to (6.6) used for linearization by each of these approaches.

- **L-LS-1:** Reference equation is randomly selected from one of the distance estimate from TOA measurement and subtracted from the rest.
  \[
r^2_{f,(1)} = r^2_k, \quad k = 1, \ldots, N
\]

- **L-LS-2:** $\binom{N}{2}$ pairs of equations, one of which is used as reference, are subtracted from each other to constitute linear equations, i.e.
  \[
r^2_{r,(2)} = r^2_k, \quad k = 1, \ldots, N
\]

- **L-LS-3:** Average of the square of all measurements is used as a reference and subtracted from all measurements.
  \[
r^2_{f,(3)} = \frac{1}{N} \sum_{j=1}^{N} r^2_j
\]
• L-LS-A: A weighted average of square of all measurements, where weights are optimized to have minimum linear equation (6.7) noise variance, is used as a reference and subtracted from the all distance estimates. Therefore the reference equation for the \( i \)-th measurement, \( i = 1, \ldots, N \), is

\[
\hat{r}_{f,(A),i}^2 = \sum_{j=1}^{N} w_{ij} r_j^2
\]

where each weight is \( 0 \leq w_{ij} \leq 1 \) and \( \sum_{j=1}^{N} w_{ij} = 1 \).

• L-LS-M: Minimum distance estimate from TOA measurement from say node \( m \) is selected as the reference.

\[
\hat{r}_{f,(M)}^2 = r_m^2, \quad \text{where } r_m \leq r_i, \quad i = 1, 2, \ldots, N, \quad i \neq m
\]

Since L-LS-2 and L-LS-3 are shown to perform similar and both better than L-LS-1 in [4], we will compare our approaches only with L-LS-3.

**L-LS-3**

L-LS-3 averages squares of distance estimations to obtain the reference equation. Using (6.5), reference equation becomes

\[
\hat{r}_{f,(3)}^2 = \frac{1}{N} \sum_j r_j^2
\]

\[
= x^2 + y^2 - \left( \frac{2}{N} \sum_j x_j \right) x - \left( \frac{2}{N} \sum_j y_j \right) y + \frac{1}{N} \sum_j \left( k_j^2 + 2r_j n_j - n_j^2 \right)
\]

where \( k_i^2 = x_i^2 + y_i^2, \quad i = 1, \ldots, N \). Note that, in the rest of the chapter, summation limits are from 1 to \( N \) whenever omitted. Then the matrix form as in
(6.9) for linear LS solution will have the following parameters:

\[ \alpha_{i,(3)} = 2 \left[ \left( \frac{1}{N} \sum_j x_j - x_i \right), \left( \frac{1}{N} \sum_j y_j - y_i \right) \right] \]

\[ p_{i,(3)} = r_i^2 - \frac{1}{N} \sum_j r_j^2 - k_i^2 + \frac{1}{N} \sum_j k_j^2 \]

\[ \tilde{n}_{i,(3)} = 2r_in_i - n_i^2 - \frac{1}{N} \sum_j (2r_jn_j + n_j^2) \]

\[ = 2r_in_i \left( 1 - \frac{1}{N} \right) - n_i^2 \left( 1 - \frac{1}{N} \right) - \frac{1}{N} \sum_{j \neq i} (2r_jn_j + n_j^2) \]

When \( n_i \) is Gaussian (\( n_i \sim \mathcal{N}(0, \sigma_i^2) \)), according to Corollary 1, mean and variance of \( n_i^2 \) are \( E[n_i^2] = \sigma_i^2 \) and \( Var(n_i^2) = \beta_i^2 = \sigma_i^2(3\sigma_i^2 - 1) \), respectively. Therefore using Corollary 3, mean and variance of \( \tilde{n}_{i,(3)} \) becomes

\[ E[\tilde{n}_{i,(3)}] = -\sigma_i^2 + \frac{1}{N} \sum_j \sigma_j^2 \]

\[ Var(\tilde{n}_{i,(3)}) = 4r_i^2\sigma_i^2 \left( 1 - \frac{1}{N} \right)^2 + \beta_i^2 \left( 1 - \frac{1}{N} \right)^2 + \frac{1}{N^2} \sum_{j \neq i} (4r_j^2\sigma_j^2 + \beta_j^2) \]

For i.i.d. TOA estimation error, i.e. \( n_i = n_1 \) for \( i = 2, 3, \ldots, N \), mean and variance simplifies to

\[ E[\tilde{n}_{i,(3)}] = 0 \]  \hfill (6.16)

\[ Var(\tilde{n}_{i,(3)}) = 4 \left( r_i^2 \left( 1 - \frac{1}{N} \right)^2 + \frac{1}{N^2} \sum_{j \neq i} r_j^2 \right) \sigma_i^2 + \left( 1 - \frac{1}{N} \right)^2 + \frac{1}{(N-1)^2} \beta_i^2 \]

\[ = 4 \left( r_i^2 \left( 1 - \frac{1}{N} \right)^2 + \frac{1}{N^2} \sum_{j \neq i} r_j^2 \right) \sigma_i^2 \]

\[ + \left( 1 - \frac{1}{N} \right)^2 + \frac{1}{(N-1)^2} \sigma_i^2 (3\sigma_i^2 - 1) \]  \hfill (6.17)

where the variance is a quadratic equation in \( \sigma_i^2 \).

**L-LS-M**

(6.5) shows that noise for \( i \)-th measurement is amplified by the estimated distance, suggesting that smallest measurement, \( r_m \) from \( m \)-th sensor node, will
introduce the smallest error among other nodes. Therefore L-LS-M reduces the error introduced by the reference equation for the linear equations, and in effect reduces the overall error.

Matrix form rows as in (6.9) for L-LS-M becomes

\[
\alpha_{i,(M)} = 2 [(x_m - x_i), (y_m - y_i)] \quad (6.18)
\]

\[
p_{i,(M)} = r_i^2 - r_m^2 - k_i^2 + k_m^2 \quad (6.19)
\]

\[
n_{i,(M)} = 2r_in_i - n_i^2 - 2rn_m + n_m^2 \quad (6.20)
\]

for \(i = 1, \ldots, N\) and \(i \neq m\). Mean and variance of \(\tilde{n}_{i,(M)}\) are

\[
E[\tilde{n}_{i,(M)}] = 0
\]

\[
Var(\tilde{n}_{i,(M)}) = 4(r_i^2 \sigma_i^2 + r_m^2 \sigma_m^2) + \beta_i^2 + \beta_m^2
\]

where \(\beta_i^2 = Var(n_i^2)\).

For i.i.d. TOA based distance estimation error, \(\sigma_i^2 = \sigma_j^2\) for \(i = 2, 3, \ldots, N\), variance simplifies to another quadratic function in \(\sigma_1^2\):

\[
Var(\tilde{n}_{i,(M)}) = 4(r_i^2 + r_m^2)\sigma_1^2 + 2\sigma_1^2 (3\sigma_1^2 - 1) \quad (6.21)
\]

**L-LS-A**

To obtain linear equations as in the form of (6.7), distance estimate from a single sensor node from TOA measurement or a linear combination of distance estimates from all or a subset of the sensor nodes is subtracted from the \(i\)-th measurement. This subtraction effectively introduces additional uncertainty and hence increase the noise variance of (6.8).

L-LS-1, L-LS-2, and L-LS-M uses a single measurement, where L-LS-3 takes the average of all measurements. In this section we will introduce another technique, L-LS-A, which takes a weighted sum of the measurements excluding the measurement to be linearized. Weights are optimized to minimize the variance of the noise given by (6.8).

Let the reference equation for the \(i\)-th measurement be \(r_{j,i}^2\) and set of \(N\) weights be \(\{w_{ij}\}\) with \(i, j = 1, \ldots, N\) and \(w_{ij} = 0\) for \(i = j\). \(w_{ii}\) is set to zero to
prevent the use of same equation as reference. If not minimum noise variance is achieved when \( w_{ij} \) is 1 for \( i = j \) and 0 otherwise.

The reference equation can be written as

\[
\begin{align*}
\hat{r}_{f,i}^2 &= \sum_j w_{ij} r_j^2 \\
&= x^2 + y^2 - 2x \sum_j w_{ij} x_j - 2y \sum_j w_{ij} y_j + \sum_j w_{ij} k_j^2 \\
&\quad + \sum_j w_{ij} (2r_j n_j - n_j^2)
\end{align*}
\]

(6.22)

Note that to cancel the \( x^2 \) and \( y^2 \) terms after subtracting from the \( i \)-th measurement, weights must add to unity, i.e. \( \sum_j w_{ij} = 1 \). Using the reference equation (6.22) we get the following linear relations for the \( i \)-th measurement:

\[
\begin{align*}
\alpha_{i,(A)} &= 2 [(x_{f,i} - x_i), (y_{f,i} - y_i)] \\
p_{i,(A)} &= r_i^2 - r_{f,i}^2 - k_i^2 + k_{f,i}^2 \\
\hat{n}_{i,(A)} &= 2r_i n_i - n_i^2 - 2 \sum_j w_{ij} r_j n_j + \sum_j w_{ij} n_j^2
\end{align*}
\]

(6.23)

(6.24)

(6.25)

where

\[
\begin{align*}
x_{f,i} &= \sum_j w_{ij} x_j \\
y_{f,i} &= \sum_j w_{ij} y_j \\
k_{f,i}^2 &= x_{f,i}^2 + y_{f,i}^2 \\
k_i^2 &= x_i^2 + y_i^2
\end{align*}
\]

(6.26)

(6.27)

(6.28)

(6.29)

Mean and variance for the noise term becomes

\[
\begin{align*}
E[\hat{n}_{i,(A)}] &= -\sigma_i^2 + \sum_j w_{ij} \sigma_j^2 \\
Var(\hat{n}_{i,(A)}) &= 4r_i^2 \sigma_i^2 + \sigma_i^2 (3\sigma_i^2 - 1) + \sum_j w_{ij}^2 \sigma_j^2 (4r_j^2 + 3\sigma_j^2 - 1)
\end{align*}
\]

(6.30)

For the \( i \)-th measurement, we need to choose set of weights, \( \{w_{ij}\} \), where \( i, j = 1, \ldots, N \) and \( w_{ii} = 0 \), that minimizes (6.30):

\[
\hat{w}_{ij} = \arg\min_{w_{ij}} \Var(\hat{n}_{i,(A)})
\]

(6.31)
subject to $\sum_j w_{ij} = 1$. Using Lagrange multipliers method, define a new function

$$g_i = Var(\hat{n}_{i,(A)}) + \lambda \left( \sum_j w_{ij} - 1 \right)$$

$$= 4r_i^2 \sigma_i^2 + \sigma_i^2 (3\sigma_i^2 - 1) + \sum_j \sigma_j^2 (4r_j^2 + 3\sigma_j^2 - 1) + \lambda \left( \sum_j w_{ij} - 1 \right)$$

Then for $j \neq i$

$$\frac{\partial g_i}{\partial w_{ij}} = 2w_{ij}\sigma_j^2 (4r_j^2 + 3\sigma_j^2 - 1) + \lambda = 0$$

and hence

$$w_{ij} = \begin{cases} -\lambda \theta_j & j \neq i \\ 0 & j = i \end{cases} \quad (6.32)$$

where $\theta_j = \left(2\sigma_j^2 (4r_j^2 + 3\sigma_j^2 - 1)\right)^{-1}$ and $w_{ii} = 0$. Since $\sum_j w_{ij} = 1$, using (6.32), we obtain

$$\sum_j w_{ij} = -\lambda \sum_{j \neq i} \theta_j = 1$$

and therefore

$$\lambda = -\frac{1}{\sum_{j \neq i} \theta_j} = -\frac{1}{\sum_{j} \theta_j - \theta_i} \quad (6.33)$$

Substituting (6.33) into (6.32), optimal weights become

$$w_{ij} = \begin{cases} \frac{\theta_j}{\sum \theta_k - \theta_i} & j \neq i \\ 0 & j = i \end{cases} \quad (6.34)$$

For i.i.d TOA estimation errors, i.e. $\sigma_i^2 = \sigma_1^2$ for $i = 2, \ldots, N$, define $\theta_j'$ as

$$\theta_j' = 2\sigma_i^2 \theta_j$$

$$= (4r_j^2 + 3\sigma_1^2 - 1)^{-1}$$
and coefficients becomes

\[
   w_{ij} = \begin{cases} 
      \frac{\theta_j}{\sum_k \theta_k - \theta_i} & j \neq i \\
      0 & j = i 
   \end{cases}
\]

Therefore using these set of weights, linear equation noise variance becomes

\[
   \text{Var} \left( \tilde{n}_{i,(A)} \right) = \sigma_1^2 \left( 4r_i^2 + 3\sigma_1^2 - 1 \right) + \sigma_1^2 \sum_{j \neq i} w_{ij}^2 (4r_j^2 + 3\sigma_1^2 - 1) \quad (6.35)
\]

**Hybrid techniques**

For a given noise variance of individual linear equations \( \sigma_x^2 \) and \( \sigma_y^2 \), variances of overall estimation error in \( x \) and \( y \) dimensions, respectively, can be computed using \( T_A \) defined in (6.12)

\[
   \sigma_x^2 = \sum_j T_{1j}^2 \text{Var} \left( \tilde{n}_j \right)
\]

\[
   \sigma_y^2 = \sum_j T_{2j}^2 \text{Var} \left( \tilde{n}_j \right) \quad (6.36)
\]

where \( T_{ij} \) is the \( i \)-th row and \( j \)-th column of \( T_A \) and \( \text{Var} \left( \tilde{n}_j \right) \) is the noise variance of \( j \)-th linear equation. For example, \( \text{Var} \left( \tilde{n}_j \right) \) for L-LS-3, L-LS-M, and L-LS-A are given in (6.17), (6.21), and (6.35), respectively for i.i.d. TOA estimation errors.

A hybrid technique first computes \( \text{Var} \left( \tilde{n}_j \right) \) for a set of measurements and then forms \( T_A \) to compute the average error in \( x \) and \( y \) dimensions given by (6.36). The technique that has the smallest overall noise variance is selected in location estimation.

**Global optimization of error on \( x \) and \( y \) dimensions**

In Section 6.1.3 we minimized the introduced error for each equation due to linearization. Overall error in \( x \) and \( y \) dimensions, however, are linear combination of individual errors from these equations. In this section, we will drive the analytic expression for the overall error and show that estimation of coefficients to minimize error becomes another non-linear optimization problem.
We will optimize the the coefficients \{\omega_i, i = 1, \ldots, N\} of a reference equation in the form \(r^2_j = \sum_j \omega_j r^2_j\), to achieve the minimum overall target location estimation error. Let \(\omega^T = [\omega_1, \ldots, \omega_N]\) denote the coefficient vector.

Note that in Section 6.1.3, for the \(i\)-th sensor node, unique set of coefficients \(\{w_{ij}, j = 1, \ldots, N\}\) were used to obtain the reference equation, \(r^2_{j,i} = \sum_j w_{ij} r^2_j\).

In this section we propose using the same set of coefficients for all sensor nodes despite performance loss, due to complexity.

Let \(A\) be the \(N \times 2\) coefficient matrix as in (6.10) and \(A = [a_x, a_y]\), \(a_x\) and \(a_y\) column vectors. Estimation error, \(\mathbf{e}\), in (6.11) is

\[
\hat{t} = T_A \mathbf{p} = T_A \mathbf{a} + T_A \mathbf{n} = t + \mathbf{e}
\]

where \(T_A = (A^T A)^{-1} A^T\) and \(\mathbf{e} = T_A \mathbf{n}\). Error can be computed as follows

\[
\mathbf{e} = T_A \mathbf{n} = \left( \begin{bmatrix} a_x^T \\ a_y^T \end{bmatrix} \begin{bmatrix} a_x & a_y \end{bmatrix} \right)^{-1} \begin{bmatrix} a_x^T \\ a_y^T \end{bmatrix} \mathbf{n} = \left[ \begin{array}{c} a_x^T a_x \\ a_y^T a_y \end{array} \right]^{-1} \begin{bmatrix} a_x^T \mathbf{n} \\ a_y^T \mathbf{n} \end{bmatrix} = \frac{1}{\text{det}A} \begin{bmatrix} a_y^T a_y & -a_x^T a_y \\ -a_y^T a_x & a_x^T a_x \end{bmatrix} \begin{bmatrix} a_x^T \mathbf{n} \\ a_y^T \mathbf{n} \end{bmatrix} = \frac{1}{\text{det}A} \begin{bmatrix} a_y^T a_y a_x^T \mathbf{n} - a_x^T a_y a_y^T \mathbf{n} \\ -a_y^T a_x a_x^T \mathbf{n} + a_x^T a_y a_y^T \mathbf{n} \end{bmatrix} = \frac{1}{\text{det}A} \begin{bmatrix} m_x \\ m_y \end{bmatrix}
\]

where

\[
\text{det}A = a_x^T a_x a_y^T a_y - a_y^T a_x a_x^T a_y = \sum_i \sum_j a_{x,i}^2 a_{y,j}^2 - \sum_i \sum_j a_{y,i} a_{x,i} a_{x,j} a_{y,j}
\]
with \(a_{x,k}\) and \(a_{y,k}\) being the \(k\)-th element of column vectors \(a_x\) and \(a_y\), respectively

\[
m_x = a_y^T a_y a_x^T \tilde{n} - a_x^T a_y a_y^T \tilde{n}
\]

\[
m_y = -a_y^T a_x a_x^T \tilde{n} + a_x^T a_x a_y^T \tilde{n}
\]

Since

\[
a_{x,i} = 2 \left( \sum_k \omega_k x_k - x_i \right) = 2 \left( \omega^T x - x_i \right)
\]

\[
a_{y,i} = 2 \left( \sum_k \omega_k y_k - y_i \right) = 2 \left( \omega^T y - y_i \right)
\]

\[
a_{y,i}^2 = 4 \left( y_i^2 - 2 \omega^T y y_i + (\omega^T y)^2 \right)
\]

\[
a_{x,i}a_{y,i} = 4 \left( \omega^T x \omega^T y - \omega^T y x_i - \omega^T y x_i + x_i y_i \right)
\]

with \(\omega^T = [\omega_1, \omega_2, \ldots, \omega_N]\), \(x^T = [x_1, x_2, \ldots, x_N]\), and \(y^T = [y_1, y_2, \ldots, y_N]\). Then

\[
m_x = a_y^T a_y a_x^T \tilde{n} - a_x^T a_y a_y^T \tilde{n}
\]

\[
= \sum_i a_{y,i}^2 \sum_j a_{x,j} \tilde{n}_j - \sum_i a_{x,i} a_{y,i} \sum_j a_{y,j} \tilde{n}_j
\]

\[
= 8 \left\{ \sum_i \left( y_i^2 - 2 \omega^T y y_i + (\omega^T y)^2 \right) \right\} \sum_j (\omega^T x - x_j) \tilde{n}_j
\]

\[
-8 \left\{ \sum_i (\omega^T x \omega^T y - \omega^T y x_i - \omega^T y x_i + x_i y_i) \right\} \sum_j (\omega^T y - y_j) \tilde{n}_j
\]

\[
= 8 \left( y^T y - 2 \omega^T y \sum_j y_j + N (\omega^T y)^2 \right) \sum_j (\omega^T x - x_j) \tilde{n}_j
\]

\[
-8 \left( N \omega^T x \omega^T y - \omega^T y \sum_i x_i - \omega^T x \sum_i y_i + x^T y \right) \sum_j (\omega^T y - y_j) \tilde{n}_j
\]

\[
= \gamma_1(\omega, y) \sum_j (\omega^T x - x_j) \tilde{n}_j - \gamma_2(\omega, x, y) \sum_j (\omega^T y - y_j) \tilde{n}_j
\]

\[
= \sum_j \tilde{n}_j \left\{ \gamma_1(\omega, y) (\omega^T x - x_j) - \gamma_2(\omega, x, y) (\omega^T y - y_j) \right\}
\]  

(6.37)
where
\[
\gamma_1(\omega, y) = 8 \left( y^T y - 2\omega^T y \sum_j y_j + N (\omega^T y)^2 \right) \\
= 8 \left( y^T y - 2\omega^T y y^T 1_N + N (\omega^T y)^2 \right)
\]
\[
\gamma_2(\omega, x, y) = 8 \left( N\omega^T x\omega^T y - \omega^T y \sum_i x_i - \omega^T x \sum_i y_i + x^T y \right) \\
= 8 \left( N\omega^T x\omega^T y - \omega^T y x^T 1_N - \omega^T x y^T 1_N + x^T y \right)
\]
\[
\tilde{n}_j = 2r_j n_j - n_j^2
\]
with \(1_N\) being the \(N \times 1\) all 1’s column vector.

Using Corollary 3, variance of \(\tilde{n}_j\) becomes
\[
\tilde{\sigma}_j^2 = 4r_j^2 \sigma_j^2 + \sigma_j^2 (3\sigma_j^2 - 1)
\]
and variance of \(m_x\) given in (6.37) for i.i.d. estimation error, \(\sigma_j^2 = \sigma_1^2\) for \(j = 2, \ldots, N\), becomes
\[
\text{Var} (m_x) = \sum_j \tilde{\sigma}_j^2 \left\{ \gamma_1(\omega, y) (\omega^T x - x_j) - \gamma_2(\omega, x, y) (\omega^T y - y_j) \right\}^2 \\
= \gamma_1^2(\omega, y) \sum_j \tilde{\sigma}_j^2 (\omega^T x - x_j)^2 - 2\gamma(\omega, y) \gamma_2(\omega, x, y) \sum_j \tilde{\sigma}_j^2 (\omega^T x - x_j) (\omega^T y - y_j) + \gamma_2^2(\omega, x, y) \sum_j \tilde{\sigma}_j^2 (\omega^T y - y_j)^2 \\
= \gamma_1^2(\omega, y) \left( (\omega^T x)^2 \sum_j \tilde{\sigma}_j^2 - 2\omega^T x \sum_j \tilde{\sigma}_j^2 x_j + \sum_j \tilde{\sigma}_j^2 x_j^2 \right) \\
- 2\gamma_1(\omega, y) \gamma_2(\omega, x, y) \left( \omega^T x \omega^T y \sum_j \tilde{\sigma}_j^2 - \omega^T x \sum_j \tilde{\sigma}_j^2 y_j - \omega^T y \sum_j \tilde{\sigma}_j^2 x_j + \sum_j \tilde{\sigma}_j^2 x_j y_j \right) + \gamma_2^2(\omega, x, y) \left( (\omega^T y)^2 \sum_j \tilde{\sigma}_j^2 - 2\omega^T y \sum_j \tilde{\sigma}_j^2 y_j + \sum_j \tilde{\sigma}_j^2 y_j^2 \right) (6.38)
\]
Our objective is to find the coefficient vector \(\omega\) that minimizes the variance given in (6.38).
\[ \hat{\omega} = \arg\min_{\omega} \text{Var}(m_x) \]

subject to \( \sum_i \omega_i = 1 \) or \( \omega^T 1_N \). Using Lagrange multipliers define

\[ h = \text{Var}(m_x) + \lambda \omega^T 1_N \]

then

\[ \frac{\partial h}{\partial \omega} = \frac{\partial \text{Var}(m_x)}{\partial \omega} + \lambda 1_N \]

Let’s first calculate

\[ \frac{\partial}{\partial \omega} \gamma_1(\omega, y) = 8 \left( -2y \sum_j y_j + 2N y \omega^T y \right) \]

\[ = 16y \left( -\sum_j y_j + N \omega^T y \right) \]

(6.39)

\[ \frac{\partial}{\partial \omega} \gamma_2(\omega, x, y) = 8 \left( N x \omega^T y + N y \omega^T x - y \sum_j x_j - x \sum_j y_i + y \right) \]

(6.40)

\[ = 8 \left( N (xy^T + yx^T) \omega - xy^T 1_N - xy^T 1_N + y \right) \]

\[ = 8 \left( (xy^T + yx^T) (N\omega - 1_N) + y \right) \]

We can compute \( \partial h/\partial \omega \) using (6.39) and (6.40) and set it to zero to compute \( \omega \). This is, however, another non-linear optimization problem, and will not be addressed in this thesis.

### 6.1.4 Simulation Results

Consider a network setup with four sensor nodes, \( s_i, i = 1, \ldots, 4 \), placed at the corners of a 100m \( \times \) 100m rectangular area where a target resides anywhere
inside this region as shown in Figure 6.2. We evaluated performance of each algorithm at 10000 points inside the rectangular region and averaged the error over 1000 repetitions for each point. TOA measurements are assumed to be i.i.d. for all sensor nodes.

Figure 6.3 plots root-mean-square of error (RMSE) for various standard deviation values, $\sigma_i$ given in (6.3), for the TOA estimation error. i.i.d. estimation error is assumed, i.e. $\sigma_i^2 = \sigma_1^2$ for $i = 2, \ldots, N$. As it is seen in the figure, L-LS-1 performs the worst and L-LS-A achieves closest estimation error to CRLB. Performance of L-LS-M is slightly worse than L-LS-A but significantly better than L-LS-3. Difference in the performance between linear LS algorithms and CRLB vanishes for small estimation error variances.

Figure 6.4 plots error distribution for each technique and CRLB over the region when standard deviation is 6m. Some observations from the figure are as follows:

- L-LS-1 has smaller error when the target is closer to the reference node, which is on the upper left corner in this case, and higher error for the distant
Figure 6.3: Overall average RMSE (in m) of TOA based localization techniques and CRLB vs. standard deviation of TOA measurement, $\sigma_1$, over the simulation region given in Figure 6.2. Standard deviations of TOA measurements for different nodes are modeled as i.i.d. L-LS-M and L-LS-A both perform closer to CRLB and better than L-LS-1 and L-LS-3.
locations.

- L-LS-3, on the other hand, performs better at the center where all TOA measurements are close to each other and worsens away from the center. This is because averaging TOA measurements that are close to each other reduces the estimation error variance.

- L-LS-M replicates the performance of L-LS-1 on the upper left quadrant to all other three quadrants. However the performance slightly deteriorates along the horizontal and vertical lines in the middle where averaging two or more sensor nodes introduces smaller noise than any single node.

- L-LS-A has the most uniform performance all over the region except for the corners where performance suffers slightly.

### 6.1.5 Complexity Analysis

In this section we will evaluate the complexity of the algorithms given in Section 6.1.3.

**L-LS-1 and L-LS-3**

\( \alpha_{i,(1)} \) and \( \alpha_{i,(1)} \), \( i \)-th row of LS matrix \( A \) in (6.12) for L-LS-1 and L-LS-3, respectively is given as follows

\[
\alpha_{i,(1)} = 2 \left[ (x_k - x_i), (y_k - y_i) \right] \quad (6.42)
\]

\[
\alpha_{i,(3)} = 2 \left[ \left( \frac{1}{N} \sum_j x_j - x_i \right), \left( \frac{1}{N} \sum_j y_j - y_i \right) \right] \quad (6.43)
\]

where \( k \)-th sensor node is selected as the reference node for L-LS-1. Therefore \( A \) for L-LS-1 and L-LS-3 is independent from distance estimations from TOA measurements, \( \{ r_i : i = 1, \ldots, N \} \), and \( T_A \) remains same for each estimation throughout the region. Hence it can be precomputed and stored to be used in target location estimation.
Figure 6.4: RMSE of TOA based localization techniques and CRLB over the simulation region given in Figure 6.2 when standard deviation of TOA measurement, $\sigma_1$, is 6m. Standard deviations of TOA measurements for different nodes are modeled as i.i.d.
L-LS-M

\( \mathbf{\alpha}_{i,(1)} \) for L-LS-M is given by (6.18) and repeated here for convenience

\[
\mathbf{\alpha}_{i,(M)} = 2 \left[ (x_m - x_i), (y_m - y_i) \right]
\]

where \( m \)-th sensor node is the one with minimum distance estimation. Therefore \( \mathbf{A} \), and hence \( \mathbf{T}_A \) is also independent from TOA measurements in this case; it differs, however, as the choice of reference nodes depends on target location. \( \mathbf{T}_{A,k}, k = 1, \ldots, N \), for the \( N \) sensor nodes can be precomputed and stored making the complexity of L-LS-M similar to L-LS-1 and L-LS-M.

L-LS-A

For L-LS-A, on the other hand, \( \mathbf{A} \) is a function of distance estimation from TOA measurement since \( \mathbf{\alpha}_{i,(A)} \) as given in (6.23)

\[
\mathbf{\alpha}_{i,(A)} = 2 \left[ (x_{f,i} - x_i), (y_{f,i} - y_i) \right]
\]

where \( x_{f,i} \) (6.26) and \( y_{f,i} \) (6.27) both depend on \( w_{ij} \) given by (6.32), which is a function of TOA based distance estimate, \( r_i, i = 1, \ldots, N \). As a result \( \mathbf{T}_A \) is also dependent on TOA measurement, and hence needs to be computed online for each target location estimation. Computation of \( \mathbf{T}_A \) requires \( N \times 2 \times 2 \times N \) matrix multiplication, \( 2 \times 2 \) matrix inversion and \( 2 \times 2 \times 2 \times N \) matrix multiplication. Despite its increased complexity, however, L-LS-A’s performance gain with respect to L-LS-M is not significant; L-LS-M provides a lower complexity and high performance alternative.

6.2 Passive Localization

In Section 6.1, we introduced new LS techniques to estimate target location and compared their performances to several existing techniques. In this section we will show that these techniques can also be utilized in time difference of arrival (TDOA) based passive localization problems.
Consider two sensor nodes with known locations, $s_m$ and $s_n$, as in Figure 6.5. When one of the sensors ($s_m$) transmits a signal, the other node ($s_n$) receives two copies of the same signal, one from direct path and one reflected from the target \( t \). Let $\tau_d$ and $\tau_r$ denote the TOA of the direct and reflection paths, respectively, and $\Delta \tau = \tau_r - \tau_d$ be the time difference of arrival.

\[
d_{m,t} + d_{n,t} = c\tau_r = c(\tau_d + \Delta \tau) = d_{m,n} + c\Delta \tau
\]  

(6.44)

where $c$ is the speed of light. $\Delta \tau$ is measured and $d_{m,n}$ is known. Set of points whose sum of the distances to two anchor points is a constant constitutes an ellipse. Therefore, using (6.44) target can be located on an ellipse. In the absence of estimation errors, three or more such ellipses intersect at a single point where the target resides. However, in the presence of noise in measurements ambiguity in the target location arises. Linear and non-linear techniques are used to estimate target’s location in such cases.

Ultra-wideband (UWB) signals are preferred in these types of scenarios due to their ability of fine time resolution. When the signal bandwidth is sufficiently large, it is possible to distinguish multipaths from nearby objects.
6.2.1 System Model

Consider a wireless network consisting of $N$ sensor nodes, which are located at $s_i = [x_i, y_i]^T$, $i = 1, \ldots, N$. The aim is to estimate the position of a target located at $t = [x, y]^T$ using multipath distance estimations from the communications between $\binom{N}{2}$ sensor node pairs and reflections from the target node. As opposed to Section 6.1.1, target is not an active tag that communicates with the sensor nodes; therefore this type of localization is sometimes called as passive localization.

Let

$$S = \{(s_k, s_n) : k, n \in (1, \ldots, N), k \neq n\} \quad (6.45)$$

denote the set of sensor pairs that communicate with each other and $M$ denote the cardinality of $S$. Note that communication between all pairs may not exist and hence $M \leq \binom{N}{2}$.

Let $z_i$ denote the multipath distance estimation between sensor nodes in the $i$-th pair in $S$ and the target node. It is modeled as

$$z_i = ct_i = l_i(x, y) + n_i, \quad i = 1, \ldots, M \quad (6.46)$$

where $c$ is the speed of light, $\tau_i$ is the time elapsed to traverse multipath distance, $n_i$ is the estimation error and $l_i(x, y)$ is the true multipath distance, given by

$$l_i(x, y) = \sqrt{(x - x_{tx_i})^2 + (y - y_{tx_i})^2} + \sqrt{(x - x_{rx_i})^2 + (y - y_{rx_i})^2}$$

with $tx_i$ and $rx_i$ being the index of the transmitting and receiving sensor nodes in the $i$-th pair in $S$, respectively.

6.2.2 Nonlinear LS Estimation

Given a set of multipath distance estimates, nonlinear LS estimator calculates the target position as follows:

$$\hat{t} = \arg\min_{(x, y)} \sum_{i=1}^{M} \beta_i (z_i - l_i(x, y))^2$$

where $\beta_i$ is the weighting coefficient as an indicator of reliability of multipath distance measurement for the path from $i$-th sensor node to $j$-th one.
Figure 6.6: TDOA based localization. Each sensor node pair can locate a target on an ellipse using the multipath distance. Uncertainty in the multipath distances due to measurement noise is displayed by a band.
Cramer-Rao Lower Bound

Theorem 5. Fisher information matrix (FIM) for an unbiased nonlinear estimator for the system modeled as (6.46) is given by

\[
I = \begin{bmatrix}
\sum_{i=1}^{M} \frac{1}{\sigma_i^2} \left( \frac{\partial}{\partial x} l_i(x, y) \right)^2 & \sum_{i=1}^{M} \frac{1}{\sigma_i^2} \frac{\partial}{\partial x} l_i(x, y) \frac{\partial}{\partial y} l_i(x, y) \\
\sum_{i=1}^{M} \frac{1}{\sigma_i^2} \frac{\partial}{\partial x} l_i(x, y) \frac{\partial}{\partial y} l_i(x, y) & \sum_{i=1}^{M} \frac{1}{\sigma_i^2} \left( \frac{\partial}{\partial y} l_i(x, y) \right)^2
\end{bmatrix}
\]

where the noise term \( n_i \) is Gaussian with zero mean and variance \( \sigma_i^2 \), i.e. \( n_i \sim \mathcal{N}(0, \sigma_i^2) \).

Proof. Conditional pdf of \( z \) given \( t = [x, y]^T \) is

\[
z|t \sim \mathcal{N}(l_i(x, y), \sigma_i^2)
\]

with

\[
p(z|t) = \prod_{i=1}^{M} \frac{1}{\sqrt{2\pi\sigma_i}} e^{-\frac{(z_i - l_i(x, y))^2}{2\sigma_i^2}}
\]

Then

\[
\ln p(z|t) = -\sum_{i=1}^{N} \frac{(z_i - l_i(x, y))^2}{2\sigma_i^2} + K
\]

where \( K = \frac{N}{2} \ln(2\pi) + \sum_{i=1}^{N} \ln(\sigma_i) \). Partial derivatives of log conditional pdfs with respect to \( x \) and \( y \) are computed as

\[
\frac{\partial}{\partial x} \ln p(z|t) = -\sum_{i=1}^{M} \frac{(z_i - l_i(x, y))}{\sigma_i^2} \frac{\partial}{\partial x} l_i(x, y)
\]

\[
\frac{\partial^2}{\partial x^2} \ln p(z|t) = -\sum_{i=1}^{M} \frac{1}{\sigma_i^2} \left( -\left( \frac{\partial}{\partial x} l_i(x, y) \right)^2 + (z_i - l_i(x, y)) \frac{\partial^2}{\partial x^2} l_i(x, y) \right)
\]

\[
\frac{\partial}{\partial y} \ln p(z|t) = -\sum_{i=1}^{M} \frac{(z_i - l_i(x, y))}{\sigma_i^2} \frac{\partial}{\partial y} l_i(x, y)
\]

\[
\frac{\partial^2}{\partial y^2} \ln p(z|t) = -\sum_{i=1}^{M} \frac{1}{\sigma_i^2} \left( -\left( \frac{\partial}{\partial y} l_i(x, y) \right)^2 + (z_i - l_i(x, y)) \frac{\partial^2}{\partial y^2} l_i(x, y) \right)
\]

\[
\frac{\partial^2}{\partial x \partial y} \ln p(z|t) = -\sum_{i=1}^{M} \frac{1}{\sigma_i^2} \left( -\frac{\partial}{\partial x} l_i(x, y) \frac{\partial}{\partial y} l_i(x, y) + (z_i - l_i(x, y)) \frac{\partial^2}{\partial x \partial y} l_i(x, y) \right)
\]
where

\[
\begin{align*}
\frac{\partial}{\partial x} l_i(x, y) &= \frac{x - x_{tx_i}}{\sqrt{(x - x_{tx_i})^2 + (y - y_{tx_i})^2}} + \frac{x - x_{rx_i}}{\sqrt{(x - x_{rx_i})^2 + (y - y_{rx_i})^2}} \\
\frac{\partial}{\partial y} l_i(x, y) &= \frac{y - y_{tx_i}}{\sqrt{(x - x_{tx_i})^2 + (y - y_{tx_i})^2}} + \frac{y - y_{rx_i}}{\sqrt{(x - x_{rx_i})^2 + (y - y_{rx_i})^2}}
\end{align*}
\]

with \(tx_i\) and \(rx_i\) being the index of the sensor nodes in the \(i\)-th pair in \(S\) (or the \(i\)-th multipath).

Since \(E[z_i] = l_i(x, y)\), FIM becomes

\[
I = \begin{bmatrix}
E \left[ \frac{\partial^2}{\partial x^2} \ln p(z|t) \right] & E \left[ \frac{\partial^2}{\partial x \partial y} \ln p(z|t) \right] \\
E \left[ \frac{\partial^2}{\partial x \partial y} \ln p(z|t) \right] & E \left[ \frac{\partial^2}{\partial y^2} \ln p(z|t) \right]
\end{bmatrix}
= \begin{bmatrix}
\sum_{i=1}^{M} \frac{1}{\sigma_i^2} \left( \frac{\partial}{\partial x} l_i(x, y) \right)^2 & \sum_{i=1}^{M} \frac{1}{\sigma_i^2} \frac{\partial}{\partial x} l_i(x, y) \frac{\partial}{\partial y} l_i(x, y) \\
\sum_{i=1}^{M} \frac{1}{\sigma_i^2} \frac{\partial}{\partial x} l_i(x, y) \frac{\partial}{\partial y} l_i(x, y) & \sum_{i=1}^{M} \frac{1}{\sigma_i^2} \left( \frac{\partial}{\partial y} l_i(x, y) \right)^2
\end{bmatrix}
\]

Cramer-Rao lower bound (CRLB) provides a lower bound on the mean squared error (MSE) using FIM as follows

\[
MSE = E \left\{ \| \hat{t} - t \| \right\} \geq \text{trace} \left\{ I^{-1} \right\} = \frac{I_{11} + I_{22}}{I_{11}I_{22} - I_{12}^2}
\]

with \(I_{ij}\) representing the \(i\)-th row and \(j\)-th column [4].

### 6.2.3 Linear LS Estimation for independent noise

Each measurement in (6.46) locates the target over an ellipse. In the absence of noise, target is located at the intersection of three or more these ellipses. Multiple intersection points may exist when noise is present as shown in Figure 6.6. In this section we consider independent multipath distance estimation errors. Section 6.2.4 explores the case when the noise on the multipath distance estimation is modeled to break into two components: one from transmitter to the target and one from the target to the receiver. This model suggests that noise on different multipaths that share a common path are correlated.

We will investigate a two-step linearization process. First step estimates the distances between the target and each sensor node and in the second step, target will be located using trilateration from these estimated distances.
First Step: Estimating distances between target and the sensor nodes

Assume that $N$ sensor nodes that are capable of transmitting and receiving with each other. Then noisy multipath distance estimates can be related to true distances between the target and the nodes as follows:

$$z = Bd + n \quad (6.47)$$

where $z$ is an $M \times 1$ vector of multipath estimates from TDOA measurements, $d^T = [d_1, \ldots, d_N]$ is an $N \times 1$ vector of true distances between sensor nodes and targets, $B$ is an $M \times N$ matrix consisting of 1’s and 0’s that maps true distances to multipath distances, $n$ is an $M \times 1$ noise vector, and $M \leq \binom{N}{2}$ is the number of pairs that can communicate with each other, i.e. $M$ is the cardinality of $S$ as defined in (6.45). Each row of $B$ has exactly two 1’s corresponding to sensors involved in the multipath distance estimation.

Overall effect of multiple transmissions between two sensors to estimate multipath distance is reducing estimation error, regardless of direction of the transmission. Therefore these types of transmissions are treated as a single transmission with accordingly scaled error variance.

Using LS technique, we can estimate true distances from multipath estimates in (6.47) as

$$\hat{d} = T_B z \quad (6.48)$$

where

$$T_B = (B^T B)^{-1} B^T \quad (6.49)$$

Example 1. A Sample Scenario with Four Sensors

Consider a network setup where four sensor nodes, $\{s_i, i = 1, \ldots, 4\}$, that are placed at the corner of $100m \times 100m$ rectangular area where a target resides anywhere inside as shown in Figure 6.2.

Suppose each sensor node transmits one after another and others receive the transmission and measure the multipath distance, $r_i$, from the source to target.
Figure 6.7: Simulation setup. Nodes $s_1$, $s_2$, $s_3$, and $s_4$ are placed at the corners of a $100m \times 100m$ rectangular area. These nodes communicate with each other and estimate the location of a target from the reception of delayed signals reflected from the target. Location estimations using this TDOA information are done at 10000 points inside the region and error is recorded. At each point experiment is repeated 1000 times to obtain average error.

To itself as defined in (6.46). Following measurements are recorded:

\[
\begin{align*}
  z_1 &= d_1 + d_2 + n_1 \\
  z_2 &= d_1 + d_3 + n_2 \\
  z_3 &= d_1 + d_4 + n_3 \\
  z_4 &= d_1 + d_2 + n_4 \\
  z_5 &= d_2 + d_4 + n_5 \\
  z_6 &= d_3 + d_4 + n_6
\end{align*}
\]
which can be rewritten in matrix form as

\[
\begin{bmatrix}
  z_1 \\
  z_2 \\
  z_3 \\
  z_4 \\
  z_5 \\
  z_6 \\
\end{bmatrix} = 
\begin{bmatrix}
  1 & 1 & 0 & 0 \\
  1 & 0 & 1 & 0 \\
  1 & 0 & 0 & 1 \\
  0 & 1 & 1 & 0 \\
  0 & 1 & 0 & 1 \\
  0 & 0 & 1 & 1 \\
\end{bmatrix}
\begin{bmatrix}
  d_1 \\
  d_2 \\
  d_3 \\
  d_4 \\
  d \\
\end{bmatrix} + 
\begin{bmatrix}
  n_1 \\
  n_2 \\
  n_3 \\
  n_4 \\
  n_5 \\
  n_6 \\
\end{bmatrix}
\]

or

\[ z = Bd + n \]

Then LS estimate becomes

\[ \hat{d} = T_B z = (B^T B)^{-1} B^T z \]

\[
\begin{bmatrix}
  2 & 2 & 2 & -1 & -1 & -1 \\
  2 & -1 & -1 & 2 & 2 & -1 \\
  -1 & 2 & -1 & 2 & -1 & 2 \\
  -1 & -1 & 2 & -1 & 2 & 2 \\
\end{bmatrix}
\]

Error analysis  We can obtain estimates to distances between sensor nodes and the target using (6.48) as follows

\[ \hat{d} = T_B z = d + T_B n \]

where \( z \) is the multipath distance estimates and \( T_B \) is defined in (6.48). Therefore the \( i \)-th distance estimate can be expressed as

\[ \hat{d}_i = d_i + \sum_{j=1}^{M} T_{B,ij} n_j \]

where \( T_{B,ij} \) is the \( i \)-th row and \( j \)-th column of \( T_B \), and \( M \) is number of multipath measurements. This is an unbiased estimator since mean is equal to

\[ E[\hat{d}_i] = d_i \]
The noise in multipath distance estimation is assumed to be independent. With \( \sigma_j^2 \) being the noise variance of \( j \)-th multipath measurement, variance of distance estimates becomes

\[
Var (\hat{d}_i) = \sum_{j=1}^{M} T_{B,ij}^2 \sigma_j^2
\]

For the i.i.d case, \( \sigma_j^2 = \sigma_1^2, j = 2, \ldots, M \), variance simplifies to

\[
Var (\hat{d}_i) = \sigma_1^2 \sum_{j=1}^{M} T_{ij}^2
\]

which suggests that the \( i \)-th measurement noise variance is scaled by a coefficient \( K_i \) given by

\[
K_i = \sum_{j=1}^{M} T_{ij}^2
\]

When all \( N \) nodes can communicate with each other, namely there are \( M = \binom{N}{2} \) multipath measurements between distinct pairs, all the variance coefficients become equal, i.e. \( K_i = K \) for \( i = 1, \ldots, M \). Table 6.1 shows LS matrix \( T_B \) and corresponding variance coefficient \( K \) for various number of sensor nodes, \( N \). Figure 6.8 shows how \( K \) scales down with increasing \( N \).

**Second Step: Estimating target location**

First step estimates the distances, \( \hat{d} \), between the target and sensor nodes from noisy measurements \( z \) and transforms the problem similar to the localization from TOA measurements as defined in Section 6.1.1. Therefore we can use the techniques in that section. As opposed to Section 6.1.1, noise is no longer white and L-LS-A in Section 6.1.3 is not optimized for the colored noise. Simulation results indicate that we still achieve performances close to CRLB as in the TOA case.

**Simulation Results**

Consider a network setup where four sensor nodes, \( s_i, i = 1, \ldots, 4 \), are placed at the corner of a 100m \( \times \) 100m rectangular area where a target resides...
Table 6.1: LS-matrix $T_B$ and variance coefficient $K$ for various number of sensor nodes ($N = 3, 4, $ and 5).

<table>
<thead>
<tr>
<th>$N$</th>
<th>$T_B$</th>
<th>$K$</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>$\frac{1}{2} \begin{bmatrix} 1 &amp; 1 &amp; -1 \ 1 &amp; -1 &amp; 1 \ -1 &amp; 1 &amp; 1 \end{bmatrix}$</td>
<td>0.7500</td>
</tr>
<tr>
<td>4</td>
<td>$\frac{1}{6} \begin{bmatrix} 2 &amp; 2 &amp; 2 &amp; -1 &amp; -1 &amp; -1 \ 2 &amp; -1 &amp; -1 &amp; 2 &amp; 2 &amp; -1 \ -1 &amp; 2 &amp; -1 &amp; 2 &amp; -1 &amp; 2 \ -1 &amp; -1 &amp; 2 &amp; -1 &amp; 2 &amp; 2 \end{bmatrix}$</td>
<td>0.4167</td>
</tr>
<tr>
<td>5</td>
<td>$\frac{1}{12} \begin{bmatrix} 3 &amp; 3 &amp; 3 &amp; 3 &amp; -1 &amp; -1 &amp; -1 &amp; -1 &amp; -1 &amp; -1 \ 3 &amp; -1 &amp; -1 &amp; -1 &amp; 3 &amp; 3 &amp; 3 &amp; -1 &amp; -1 &amp; -1 \ -1 &amp; 3 &amp; -1 &amp; -1 &amp; 3 &amp; -1 &amp; -1 &amp; 3 &amp; 3 &amp; -1 \ -1 &amp; -1 &amp; 3 &amp; -1 &amp; -1 &amp; 3 &amp; -1 &amp; 3 &amp; -1 &amp; 3 \ -1 &amp; -1 &amp; -1 &amp; 3 &amp; -1 &amp; -1 &amp; 3 &amp; -1 &amp; 3 &amp; 3 \end{bmatrix}$</td>
<td>0.2917</td>
</tr>
</tbody>
</table>

Figure 6.8: Variance coefficient $K$ vs. number of sensor nodes $N$ when all nodes communicate with each other.

(a) For small $N$  
(b) For large $N$, logarithmic scales
anywhere inside this region as shown in Figure 6.7. We evaluated performance of each algorithm at 10000 points inside the rectangular region and averaged the error over 1000 repetitions for each point. TDOA measurements are assumed to be i.i.d. for all sensor nodes.

Figure 6.10 plots error distribution over the simulation region when multipath distance estimation error standard deviation is 6m. Figure 6.9 plots standard deviation vs. RMSE for different algorithms. L-LS-M and L-LS-A performs similarly and both better than L-LS-3 and L-LS-1 as in the TOA case (see Figure 6.4 and Figure 6.3).

### 6.2.4 Linear LS Estimation for correlated noise

In this section we explore a system model where the errors in multipath distance estimations are correlated as opposed to the previous section, where estimation errors belonging to different multipath distances are considered to be independent.

Consider the system in Figure 6.5. Multipath distance estimation is modeled below similar to (6.46) except for the characteristics of the error

\[ z_i = l_i(x, y) + n_i \]  

(6.52)

Here we assume that the estimation error \( n_i \) can be broken into two components that constitutes the \( i \)-th multipath: one from transmitter to the target \((\tilde{n}_{t,tx_i})\) and one from target to the transmitter \((\tilde{n}_{t,rx_i})\), i.e.

\[ n_i = \tilde{n}_{t,tx_i} + \tilde{n}_{t,rx_i} \]  

(6.53)

This type of error model would be justified in multicast scenarios where a node transmits a signal and all the remaining nodes receive this transmission and estimate corresponding multipath distances.

Using this model the \( i \)-th multipath distance can be written as

\[ z_i = d_{tx_i} + d_{rx_i} + n_i = d_{tx_i} + d_{rx_i} + \tilde{n}_{t,tx_i} + \tilde{n}_{t,rx_i} \]  

(6.54)

where \( d_k \) is the true distance between the target and the \( k \)-th sensor node, \( tx_i \) and \( rx_i \) are the indices of transmitter and receiver sensor node, respectively. Using
L-LS-1 RMSE - mean 5.07m

L-LS-M RMSE - mean 4.42m

L-LS-A RMSE - mean 4.37m

$\sqrt{\text{CRLB}}$ - mean 4.05m

Figure 6.9: RMSE (in m) of TDOA based passive localization for various LS techniques and CRLB over the simulation region. Multipath distance estimation error is considered i.i.d. with a standard deviation of 6m.
Figure 6.10: RMSE (in m) vs. standard deviation of distance estimation error of TDOA measurements for various LS techniques and CRLB.
(6.33) and (6.34), noise term in (6.47) can be written in matrix form as

\[ z = Bd + B\tilde{n}_t \]  \hspace{1cm} (6.55)

where \( \tilde{n}_t \) is the \( N \times 1 \) noise vector for the paths between the target and the sensor nodes. Each element of \( \tilde{n}_t, \tilde{n}_{t,i}, i = 1, \ldots, N \) is i.i.d.

The linear LS approximation of true distances using \( T_B \) in (6.49) becomes

\[ \hat{d} = T_B z = d + n_t \]  \hspace{1cm} (6.56)

which is in the same form as (6.1) with i.i.d. noise components. Therefore we can apply the same linear LS techniques as in Section 6.1.3 to this estimation problem.

**Error Analysis**

Each row of \( B \) has only two 1’s and the rest is all 0’s. See (6.50) for an example. Using (6.53), variance of error in the \( i \)-th multipath distance estimation is computed as

\[ \sigma_i^2 = Var(\tilde{n}_{t,tx_i}) + Var(\tilde{n}_{t,rx_i}) \]

\[ = 2Var(\tilde{n}_{t,1}) \]  \hspace{1cm} (6.57)

when \( \tilde{n}_{t,j} \) is i.i.d., i.e. \( \tilde{n}_{t,j} = \tilde{n}_{t,1}, j = 2, \ldots, N \). Note that i.i.d \( \tilde{n}_{t,j} \) indicates that \( n_i \) is also i.i.d, namely \( \sigma_i^2 = \sigma_1^2 \) for \( i = 1, \ldots, M \). (6.57) shows that the overall multipath estimation error variance is twice as much as the variance of the error on each paths of the multipath route, that is

\[ Var(\hat{d}_k) = Var(\tilde{n}_{t,k}) = \frac{1}{2} \sigma_1^2, \quad k = 1, \ldots, N \]  \hspace{1cm} (6.58)

The relation between the variance of the estimated true distance between \( k \)-th sensor node and the target is similar to (6.51) where noise scale coefficient here is \( K = \frac{1}{2} \). For the same simulation scenario in Figure 6.2, when the multipath estimation error is i.i.d, \( K \) is equal to 0.4167 (See Table 6.1). A higher scaling coefficient when noise is correlated is due to reduced number of independent measurements. In the i.i.d. noise case there are \( M > N \) independent measurements whereas correlated case only has \( N \) independent measurements. Smaller coefficient is desirable as it indicates a smaller noise variance for the true distance estimation in the first step.
Simulation Results

We used the same simulation in Figure 6.2 where 4 sensor nodes located a target in a $100m \times 100m$ region. Figure 6.11 compares the RMSE of the target position estimation error for two different error models as a function of standard deviation of overall multipath distance estimation. L-LS-A is used as the linear LS technique. As expected i.i.d. noise scenario locates the target with lower estimation errors because it has a lower noise scaling coefficient ($K$).

![L-LS-A](image)

Figure 6.11: Comparison of RMSE of the target position estimation error for correlated and independent error models using L-LS-A algorithm. $x$-axis is indicates the standard deviation of overall multipath distance estimation error.

6.3 Remarks

In this chapter, we introduced two new LS techniques to be used in a two-step positioning algorithm to locate active targets from TOA measurements. We then computed the closed form of error variance for each linear equation; these can be used together with LS matrix to compute overall position estimation error variance. Finally we compared the performances of each LS algorithm to CRLB and showed that both L-LS-A and L-LS-M outperformed L-LS-3 and L-LS-M provided a comparable accuracy to L-LS-A at a lower complexity.
We first introduced two new LS techniques to locate active targets from TOA measurements, analyzed their performances and compared with the existing techniques. Then we showed that passive localization from TDOA multipath estimates are transferred to a form equivalent to TOA case and applied same techniques in a two step linearization process.

This chapter, in part, has been submitted for publication of the material as it may appear in proceedings of IEEE Personal, Indoor and Mobile Radio 2010. Ergüt, Salih; Rao, Ramesh R. The dissertation author was the primary investigator and author of this paper.
Bibliography


Localization via TDOA in a UWB
csensor network using Neural
Networks

Localization and tracking have been the focus of both the industry applications and academic research. There are two main approaches: active (e.g. [1, 5, 8, 7]) and passive ranging (e.g. [4, 3, 2, 9, 10]).

In the active approach, tags are attached to objects to be tracked. These tags communicate with the nodes in the sensor network. Sensor nodes thus estimate the distances between the objects and the nodes which are used to locate the objects via triangularization.

In the passive approach, objects do not wear tags and hence they are not collaborating with the positioning process. When the nodes communicate with each other, the presence of the object causes disturbances in the received signals. By analyzing these disturbances the location of the object can be estimated.

Active tags are used in a new range of applications, including logistics (package tracking), security applications (localizing authorized persons in high-security areas), medical applications (monitoring of patients), family communications, supervision of children, search and rescue (communications with fire fighters, or avalanche/earthquake victims), control of home appliances, and military applications [5]
The systems built based on passive approach, on the other hand, have great potential for perimeter security and intrusion detection and they can be deployed around buildings or at the borders between countries.

In this chapter, we are considering the passive approach and proposing a neural network based algorithm to locate objects in an UWB sensor network.

UWB is preferred in passive approach applications since it provides high resolution in time domain. UWB signals are perfect fit for wireless position location since they are able to resolve multipath components which provide accurate location estimates without the need for complex estimation algorithms.

UWB sensor network provides a structure where low to medium rate communication and position location can be performed simultaneously. UWB technology not only facilitate centimeter accuracy in ranging but also make low power and low cost implementation of communication systems possible [5].

IEEE introduced a new standardization group 802.15.4a for low data rate communications combined with positioning capabilities which employs UWB technology as its physical layer.

Our contributions in this study are: We first define a framework for passive localization in 802.15.4a sensor networks. Then we introduce a neural network based algorithm (NNBA) to locate a single object with the known sensor node positions. The main obstacle in locating multiple objects is to identify multipaths between different sensor nodes that correspond to different objects. We devise a two-step algorithm which uses NNBA as building blocks to overcome this problem. Finally, we present performance results for locating two objects in a 3-node sensor network using this algorithm.

We only consider the cases where the objects are relatively closer to the nodes, which enables us to work in a high SNR regime. Although we focus on two-dimensional sensor network, it is straightforward to extend the algorithms to three-dimensional space.
7.1 Related Work

[4, 3, 2] studied the Cramer-Rao bounds of passive localizations in an UWB sensor network for the asymptotic case by increasing the number of sensors. They considered two types of scenarios: 1) the locations of all sensor nodes are known apriori, 2) a subset of the sensor nodes with known locations are used as anchors to locate the remaining sensor nodes. They proposed a semi-linear algorithm that uses least squares estimator for single target detection and compared the performance of their algorithm to Cramer-Rao bounds. For multiple target detection they proposed a heuristic centralized algorithm since they claim exhaustive search requires \((L!)^{NM-1}\) iterations, where \(L\) is the number of objects, \(N\) and \(M\) are the number of transmitters and receivers, respectively. However we show that there are only \(L^N(N-1)/2\) different combinations to choose from, which is much smaller than the above figure when \(M = N\). No error performance for the multiple target detection algorithm is provided and therefore we could not compare our algorithm with this research.

[9, 10] experimentally compared the performance of active and passive detection algorithms and discussed the pros and cons of both techniques. Pulse positions are estimated by means of a high-resolution maximum likelihood estimator.

7.2 A framework for detecting passive targets

The IEEE 802.15.4a packet consists of a synchronization header (SHR) preamble, a physical layer header (PHR) and a data field. The SHR preamble is composed of the ranging preamble and the start of frame delimiter (SFD).

The ranging preamble can consist of \(\{16,64,1024,4096\}\) symbols. The longer lengths \(\{1024, 4096\}\) are preferred for non-coherent receivers to help them improve the signal to noise ratio (SNR) via processing gain. Hence, they can have a better time-of-arrival estimate. The underlying symbol of the ranging preamble uses one of the length-31 ternary sequences, \(S_i\), in Table 7.1. Each \(S_i\) of length \(L = 31\) contains 15 zeros and 16 non-zero codes, and has the much desired property of
Figure 7.1: Effects of external objects on the multipath profile of UWB signals. a) Multipath profile when there is not any object in the medium, b) Multipath profile is modified due to reflecting signals coming from the external object.

Figure 7.2: Using the multipath distance target object is located on an ellipse for each sensor pair. Target object is positioned at the intersection of all these ellipses.
perfect periodic autocorrelation. In other words, the side-lobes at the periodic correlation output become zero; and what is observed at the receiver between two consecutive correlation peaks is only the power delay profile of the channel. Thus, the channel profile estimation does not get deteriorated by any side-lobe.

Table 7.1: The basis preamble symbol set

<table>
<thead>
<tr>
<th>Index</th>
<th>Symbol</th>
</tr>
</thead>
<tbody>
<tr>
<td>$S_1$</td>
<td>-1000010-1011101-10001-111100-110-100</td>
</tr>
<tr>
<td>$S_2$</td>
<td>0101-10101000-1110-11-1-1-1-10010011000</td>
</tr>
<tr>
<td>$S_3$</td>
<td>-11011000-11-11100110100-10000-1010-1</td>
</tr>
<tr>
<td>$S_4$</td>
<td>00001-100-100-1111101-1100010-10110-1</td>
</tr>
<tr>
<td>$S_5$</td>
<td>-101-100111-11000-1101110-1010000-00</td>
</tr>
<tr>
<td>$S_6$</td>
<td>1100100-1-1-11-1011-10001010-11010000</td>
</tr>
<tr>
<td>$S_7$</td>
<td>100001-101010010010111-1-1-10-1100-11</td>
</tr>
<tr>
<td>$S_8$</td>
<td>0100-10-10110000-1-1100-1101111-11010000</td>
</tr>
</tbody>
</table>

Assume that $\omega$ is the transmitted UWB pulse waveform with unit energy, $T_{sym}$ denotes the symbol duration, $N_{sym}$ is the number of symbol repetition within the preamble, $T_{pri}$ is the pulse repetition interval, $N_s$ is the total number of pulses per symbol and $E_s$ denotes the symbol energy. Then, using any basis symbol $S_i$, the preamble symbol waveform $w_i(t)$ and the preamble waveform $P_i(t)$ can be written as

$$w_i(t) = \sqrt{\frac{E_s}{N_s}} \sum_{j=0}^{L-1} S_i[j] \omega(t - jT_{pri})$$  (7.1)

$$P_i(t) = \sum_{n=0}^{N_{sym}-1} N[n] w_i(t - nT_{sym})$$  (7.2)

where $N = [11...1]_{1 \times N_{sym}}$.

A coherent receiver correlates the received waveform $Y_i(t) = P_i(t) \otimes h(t)$ with a template matched to $w_i(t)$. Then, assuming an AWGN channel the corre-
lator output $C_i(k)$ is

$$C_i(k) = \sum_{k=0}^{\infty} \int_{kT_s}^{(k+1)T_s} (Y_i(t) + n(t))dt$$

(7.3)

where $n(t)$ is the AWGN noise. Differences in $C_i$ between two observations are indicative of changes in channel profile.

As seen in Figure 7.1-a, the multipath profile is recorded when two sensor nodes are communicating with each other in the absence of any external object. When an object arrives, multipath profile alters due to receiving reflections from the object (see Figure 7.1-b). By estimating the time difference of arrival (TDOA), $\Delta t$, between the direct path and reflecting path, the multipath distance, $d$, can be computed at the sensor node $S_2$ as:

$$d = |S_1 - S_2| + c \times \Delta t$$

where $c$ is the speed of light and $S_1$ and $S_2$ are the locations of the sensors. Since $d$ gives the sum of the distances from two sensor nodes whose locations are fixed, $S_2$ can locate the object on an ellipse (Figure 7.1-b). We need at least three sensor pairs since the intersection of three or more ellipses uniquely identify the object location as it can be see in Figure 7.2.

Note that we do not need to transmit special signals between these sensor node pairs during the recording of the multipath profile, the preamble can simply be used for this purpose while these nodes are communicating with each other. This way there is no need for a secondary channel to transmit the recorded multipath profile to a data processing center, the same network can be used for this purpose.

7.3 Simulation Setup

In all simulations $1 \times 1$ unit grid is considered. Sensors are placed on a circle uniformly. For the sake of simplicity the shape of the objects and sensors are ignored and modeled as a point on the grid. Location of the objects are randomly generated.

In this study, we only consider high SNR regimes where estimation errors can be modeled as white Gaussian [4, 3]. White Gaussian assumption holds when
the errors are assumed to be due to thermal noise only. However, in reality there are other sources of errors, such as clock drifting, processor latencies, and interferences which may violate the white Gaussian assumption. We ignore all those types of errors in this chapter.

7.4 Neural Network Model

Neural networks (NN) are a non-algorithmic methods, which use parallel computing technique. They imitate functioning of the brain. Even though inter-neuron communication speed is quite slow for the brain, parallel processing allows it to analyze very complicated data in a short period of time. Neural networks learn directly from current examples rather than programming [6].

Feed forward neural networks with multiple hidden layers have been widely used and showed to operate successfully (see Figure 7.3). Multi Layer Perceptron (MLP) learning algorithm is used in the training of the network. MLP is a back propagation algorithm and it computes the error at the output of the network and sets weights of neurons iteratively. This operation is spread out on all layers and the error in the output is reduced. Deviations between the real and the predicted values are computed to evaluate the learning success of the network.

Mean square error (MSE) is used to determine the compliance between the predicted output and computed network output. The exit criterion for the
supervised learning is set on the value of MSE (e.g. when MSE is below 0.001).

After successful termination of the learning process, the classification performance is determined by applying test data to the neural network. If the performance values meet the desired criteria at the end of the test, the structure of the neural network is completed and it is ready to classify any external data.

7.5 Single Object Detection

Using time difference of arrival (TDOA) between the direct path and a reflecting path, the distance traversed by the multipath can be estimated. The multipath distance and the locations of the 2 sensors constitute an ellipse. We need at least three sensor pairs to figure out the location of the target object at the intersection of these ellipses. Assume that there are \( N \) nodes and a single object in the sensor network, and let \((x_{ci}, y_{ci})\) denote the midpoint between the \(i\)-th sensor pair, then

\[
\frac{(x - x_{c1})^2}{a_1^2} + \frac{(y - y_{c1})^2}{b_1^2} = 1
\]

\[
\frac{(x - x_{c2})^2}{a_2^2} + \frac{(y - y_{c2})^2}{b_2^2} = 1
\]

\[
\vdots
\]

\[
\frac{(x - x_{cN})^2}{a_N^2} + \frac{(y - y_{cN})^2}{b_N^2} = 1
\]

where \(a_i\) and \(b_i\) are the major and minor axes of the ellipse. There are different techniques to solve this set of non-linear equations. Linear least squares estimators as discussed in Chapter 6 provide reduced complexity solutions. In this chapter we propose to use artificial neural networks.

We assume that the locations of the sensors are known apriori. In the training phase, a set of random points on the grid are generated. Total distance from a transmitter sensor node to the target object and from the target object to a receiver node is computed as:

\[
d_i = \sqrt{(x - x_{i1})^2 + (y - y_{i1})^2} + \sqrt{(x - x_{i2})^2 + (y - y_{i2})^2} + \epsilon_i
\]
where $d_i$ is the multipath distance between the $i$-th pair, $(x, y)$ is the location of the object, $(x_{ik}, y_{ik})$, $k = 1, 2$, is the coordinate of the $k$-th sensor node in the $i$-th pair, and $\epsilon_i \ N(0, \sigma^2)$ is the white Gaussian error.

The multipath distances computed as above are then fed into the NN. The locations of the objects, i.e. $(x, y)$, are used as the output to be matched by the NN as it is trained. During the verification phase, another set of random points are used in a similar fashion to evaluate the performance of the network.

### 7.5.1 Simulation Results

Figure 7.4 shows the performance of the NN with increasing number of sensors when the error variance is fixed, namely $\sigma^2 = 0.01$. The error between the actual location, $(x, y)$ and the estimated location, $(\hat{x}, \hat{y})$ of the object is defined as

$$\varepsilon = \sqrt{(x - \hat{x})^2 + (y - \hat{y})^2}$$

Cumulative distribution function (CDF) shifts to the right, and hence mean squared error (MSE) gets smaller, as the number of sensors are increased. Note
that the transmit power of each sensor is limited, which is regulated by Federal Communications Commission (FCC) in the US. However, the total power transmitted by the sensor network is not limited. Therefore, one can benefit using more sensors to increase the accuracy of the estimates. Also the more sensors are there in the network, the more robust the network will become by tolerating individual sensor failures.

7.5.2 Evaluating the performance of NN algorithm

In this section we will compare the performance of our NN based algorithm with Cramer-Rao bound and a least squares based algorithm introduced in [2, 4]. Cramer-Rao bound gives a lower bound on the standard deviation of the estimation error, which can be used as a benchmark.

Cramer-Rao Bound

In [2], Cramer-Rao bound on the position estimation from multipaths is shown to be

\[ V(x) = V(y) \sim \frac{\sigma^2}{N^2} \]

where \( V(x) \) and \( V(y) \) are the bounds on the estimations of \( x \) and \( y \) coordinates, respectively and \( N \) is the number of transceivers, which are capable of both transmitting and receiving. Then the total variance becomes:

\[ V(x) + V(y) \sim \frac{2\sigma^2}{N^2} \]

Least squares estimator

In [2], a two-step least squares estimator is proposed. First, using the multipath distances, piece-wise distances between the sensors and the object are estimated via least squares technique. Then using these estimates, the target object is located via triangulation. They showed that the variance of this technique is:

\[ \sigma_{LS}^2 = \frac{28\sigma^2}{3N^2} \]
Figure 7.5 compares the MSE of the NN estimator with the Cramer-Rao bound and the least squares estimator as described above when the number of sensor networks are 3, 4, 5, and 6. The NN performance is comparable with the least squares estimator. As more sensor nodes are used the performance of NN and linear LS technique approaches to the Cramer-Rao bound.

When the underlying geometry is well understood, linear LS techniques perform better because they utilize this information. NN solutions tries to extract this geometrical relationship. In real implementations, however, geometrical relationships may not always be available; target may block some of the existing reflections and create new multipaths. We believe NNs would be better in estimating the underlying relationship with the multipath profile and the target locations.

### 7.6 Multiple Object Detection

Tracking multiple objects in a sensor network becomes difficult since for each sensor pair it is hard to distinguish which multipath distance belongs to which object. In order to locate objects, one of the multipath distances from each sensor pairs are grouped into a set. Let $N$ denote the number of sensors in the
network and \( L \) denote the number of objects to be detected. Then each set will contain \( \left( \frac{N}{2} \right) = \frac{N(N-1)}{2} \) elements and therefore there will be \( M = L \frac{N(N-1)}{2} \) different combination of sets to choose from.

Furthermore, these sets can be grouped such that all multipath distance measurements are used. Each such group uses a distinct measurement from each of the \( \frac{N(N-1)}{2} \) sensor pairs and hence each group contains exactly \( L \) sets since there are \( L \) objects. Therefore there are \( \frac{L \frac{N(N-1)}{2}}{L} = L \frac{N(N-1)}{2} - 1 \) such groups. Only one of these groups corresponds to the right group of sets.

As it can be seen in Figure 7.6, the overall detection algorithm is composed of two steps. In the first step, a set of multipath measurements are fed as the input and a possible target location is estimated with a cost associated with it. Apriori known sensor locations are internally used in the cost computation.

In the second step, the cost metrics for each set in the group are added together to form the group metric and the group with the lowest cost is selected.

The block used in the first step (see Figure 7.7) uses the NNBA that is trained for estimating the single object location given multipath distances from each sensor pairs as described in Section 7.5. The estimation, \( \hat{\bar{P}}_i = (\tilde{x}_i, \tilde{y}_i) \), in conjunction with the apriori known sensor locations are used to estimate the multipath distances:
\[
\hat{d}_{k,\alpha_k} = |S_{k1} - \hat{\bar{P}}_i| + |\hat{\bar{P}}_i - S_{k2}|
\]

where \( \hat{d}_{k,\alpha_k} \) is the estimated multipath distance and \( S_{kj} \) is the location of the \( j \)-th sensor node, \( j = 1, 2 \), of the \( k \)-th sensor pair. Here \( \alpha_k \in (1, 2, ..., L) \) indicates one of the \( L \) multipath distance measurements for this sensor pair.

Then, the difference between the estimated and measured multipath distances are squared and added to compute the cost metric, \( C_i \).
\[
C_i = \sum_{k=1}^{N} \left( d_{k,\alpha_k} - \hat{d}_{k,\alpha_k} \right)^2
\]
Figure 7.6: Two phase detector. In the first phase a NNBA or any linear LS technique can be used to estimate target location with an associated cost for each estimation. In the second step the decision box estimates $L$ target locations by grouping smartly and using the cost of the estimations.

Group metric is then computed by adding the cost of individual sets in that group

$$GC_j = \sum_{k=1}^{L} C_{g_{j,k}}, \text{ where } j = 1, 2, \ldots, L^{N-1}$$

where $g_{i,k}$ is the $k$-th cost index that belongs to the group $i$.

### 7.6.1 An example: Detecting two objects

In this section, as an example for multiple target detection, we will consider the case when there are two objects in a sensor network with three nodes, i.e. $N = 3$, $L = 2$. At the end of this section we will discuss the simulation results.
Figure 7.7: Location estimator. A set of multipath distance measurements are fed to NNBA. NNBA computes an estimate \((x_i, y_i)\) for the target location. New multipath distance estimates corresponding to this target estimate \((x_i, y_i)\) and known sensor node locations \((s_i, i = 1, \ldots, N)\) are computed. Square of the difference between new multipath distance estimates and input multipath distance measurements are summed to generate a cost metric \(C_i\).

All possible input combination sets are:

\[
\begin{align*}
S_1 &= \{d_{11}, d_{21}, d_{31}\} \\
S_2 &= \{d_{11}, d_{21}, d_{32}\} \\
S_3 &= \{d_{11}, d_{22}, d_{31}\} \\
S_4 &= \{d_{11}, d_{22}, d_{32}\} \\
S_5 &= \{d_{12}, d_{21}, d_{31}\} \\
S_6 &= \{d_{12}, d_{21}, d_{32}\} \\
S_7 &= \{d_{12}, d_{22}, d_{31}\} \\
S_8 &= \{d_{12}, d_{22}, d_{32}\}
\end{align*}
\]

where \(d_{i,j}\) is the \(j\)-th multipath distance measured by the \(i\)-th sensor pair. Then
the groups with complimentary sets becomes:

\[ G_1 = \{S_1, S_8\} \]
\[ G_2 = \{S_2, S_7\} \]
\[ G_3 = \{S_3, S_6\} \]
\[ G_4 = \{S_4, S_5\} \]

Finally, the group with the minimum cost is selected.

The CDF of the error between each the actual and the estimated location of the objects is plotted in Figure 7.8. As it can be seen from CDF in the same figure single target detection slightly performs better than two target detection system as expected. This is mainly due to false selection of the final group, i.e. the group with minimum cost differs from the actual one. In the simulations this error was around 3.52%. Note that even when the wrong group was chosen, the estimated locations are still close to the actual targets, therefore the estimation error is not adversely affected and hence is still comparable to single target case.
7.7 Remarks

We discussed a passive localization framework to detect objects in an UWB sensor network. First we addressed single object problem and used neural networks for localization from TDOA measurements. We compared the performance of our algorithm with Cramer-Rao bound and least squares estimator. Then we proposed a two step algorithm to detect multiple objects.

In our simulations we considered an UWB network with three sensor nodes in a $1 \times 1$ unit region. Two objects randomly are placed in this region and noisy multipaths distance measurements are generated. Scenarios that involve multiple objects is more difficult to analyze than a single object case because it is not possible to identify which multipath distance measurement belongs to which object. Inspite of this difficulty, the two step algorithm introduced in Section 7.6 achieved an accuracy as good as the single object case.

Linear least squares techniques discussed in Chapter 6 can also be used instead of neural network algorithm. When the underlying geometry is well understood such as the localization problems described in this chapter, linear least squares techniques perform better because they utilize this geometric information. Neural network solutions, on the other hand, tries to extract this geometrical relationship by sampling the multipath profile during training.

In real implementations geometrical relationships may not always be available due to complex nature of reflections from objects and obstacles nearby. Target objects may block some of the existing multipaths while creating new ones. We believe neural networks would perform better in such scenarios where underlying geometry is ambiguous.

This chapter, in full, is a reprint of the material as it appears in proceedings of IEEE International Conference on Communications 2008. Ergüt, Salih; Rao, Ramesh R.; Dural, Özgür; Sahinoglu, Zafer. The dissertation author was the primary investigator and author of this paper.
Bibliography


Part III

Concluding Remarks
Conclusion

Network protocols were initially designed for wired networks. Saltzer et. al. [2] proposed *end-to-end system design principle* in 1981 and suggested that most of the intelligence to be concentrated on the end terminals rather than inside the network. Proponents of this camp argued that violation of the end-to-end principle may cause redundant implementation of certain functionality at the network and the end-points.

Complications arose with the advent of wireless communication systems because networking protocols were not initially designed with wireless in mind. As a result, when they were employed in wireless networks without adaptation they experienced performance degradation in throughput, packet latency and jitter. Server based protocols were proposed in accordance with the end-to-end principle, while network based solutions in opposition to the end-to-end principle were also proposed.

To satisfy the needs of today’s modern applications, additional contextual information is utilized by the network components and the end-terminals. *Context aware systems* monitor their environment and seamlessly adapt to the changes without requiring user intervention. Some examples of contextual data are as follows:

- network connectivity
- cost of operation
- communication bandwidth
- user’s location
- social surrounding
- nearby people, objects and resources
- environment related condition
- time of the day, month, season, or year, i.e. time context

In this thesis, we addressed the benefits of using contextual data related to network connectivity, communication bandwidth, user’s location, as well as nearby people and objects for certain wireless networks. We focused on applying contextual information on the network layer, as opposed to popular emphasis on the application layer.

The context-aware computing paradigm is becoming increasingly popular; particularly since high capability devices are now produced at a lower cost due to advancements in technology. These devices have replaced the older models that had limited functionalities. For example, many smart phones currently available in the market are as powerful as older personal computers. Today a forty dollar wireless router offers similar services to those of the expensive high-end routers of the past. Such services include providing application dependent quality of service, supporting virtual private networks (VPN), and implementing context filtering.

In Chapter 2, we showed how a mobile client in a 3G cellular network proactively reduced the high packet delays. Such delays are typically caused by abrupt rate changes due to highly dynamic nature of the wireless channel. Rather than accepting the cellular channel as a mere data pipe, the mobile exploits the knowledge of underlying technology and the awareness of its shortcomings to enhance the end user experience with little compromise from the throughput. The proposed remote active queue management (R-AQM) algorithm is a client-based solution that manipulates the base station backlog by marking the congestion flag in the TCP header, based on its estimates on the round-trip-delay (RTT) and the channel rate. R-AQM is easy to deploy as it does not require any changes to existing network
infrastructure or end-servers. We implemented our algorithm on a Linux laptop that connected to a commercial CDMA2000 1xRTT cellular network over the air using a compatible cell phone as a modem. Our results showed R-AQM achieved a greater reduction in packet delay especially among lower rates (e.g. delays were down to 1-1.5 sec from 3-4 secs for a 48 kbps link rate). We observed only two to three percent throughput loss. These results indicate that R-AQM efficiently utilizes the channel while reducing the delays.

Chapter 3 proposed a packet size aware path setup for 802.11 based mesh networks. Wireless mesh networks are increasingly used in a variety of fields such as adhoc data service for emergency responders or large scale data access for university communities. Packet routing in mesh networks has been extensively investigated in academia and by the industry, some of the research has translated into commercial products. In classical path setup protocols, cost metric on each link is widely considered to be independent of the packet size and is generally designed to be a function of some networks parameters (such as hop count, bandwidth, delay, packet loss, and traffic load) [1]. Wireless mesh networks contain varying link rates dependent upon the channel conditions. In addition, the size of the packets transmitted across the network varies due to a broad range of applications from file downloading to voice over IP (VoIP) to streaming audio and video. For each transmitted packet there is a MAC overhead which accumulates over multiple hops. Therefore a smaller number of hops may be preferable for small packets even though an alternative path with a larger number of hops might offer a higher data rate. We showed that optimal route depends on packet size and proposed a packet-size aware path setup for wireless mesh networks.

We experimentally validated our channel models and evaluated the performance of our algorithm on Linux based Soekris communication computers formed into a 802.11b mesh network. In certain scenarios we observed throughput gains as high as 50% for UDP traffic and 13% for TCP traffic. Benefit of packet-size aware routing for TCP is due to disproportionate packet sizes of the acknowledgement and the data. In a multihop network, when packet size is used as a routing criteria, data packets (up to 1500 bytes) and ACK packets (around 40 bytes) may traverse
different paths to reduce experienced packet delay.

Part II of the thesis studies active and passive localization algorithms for cellular and wireless sensor networks. In active localization, targets actively collaborate with the reference nodes (e.g. base stations or sensor nodes). Whereas passive localization tries to locate an object not actively participating in the positioning framework. Location is the most commonly used contextual data. Emerging location based services have different accuracy requirements. For some applications crude location information is sufficient (e.g. yellow book services), while others rely on highly accurate positioning systems (e.g. inventory tracking).

In Chapter 5, we investigated active localization of a mobile in cellular networks. Mobile localization has drawn much attention due in part to Federal Communications Commission’s (FCC) mandate for service providers to provide location information in emergency calls with certain accuracy. We implemented a neural network based algorithm that uses multipath arrivals at the mobile from base stations to estimate the location of a mobile. Similar localization algorithms for GSM networks solely rely on signal strengths and do not use such multipath information due to narrowband nature of GSM radio signals.

Using Qualcomm CAIT tool we collected extensive data from commercial cellular networks to train a neural network and to validate the performance of our algorithm. We noticed that position accuracy degrades severely as the radius of the data collection region grows. We, therefore, split the overall region into smaller subregions over which a dedicated neural network was trained. Our results indicate that 67-th and 90-th percentile of mobile position estimation error using this algorithm are 217.93m and 378.82m, respectively.

In Chapter 6, we gave a brief overview of existing linear least squares (LS) techniques (L-LS-1, L-LS-2, and L-LS-3) for wireless localization and introduced two novel LS techniques (L-LS-M and L-LS-A). We compared each of these techniques and showed that the newly proposed techniques achieved a better performance than pre-existing techniques described above. The performances of the proposed techniques were close to the Cramer Rao Lower Bound (CRLB) of a non-linear unbiased position estimator. CRLB provides a lower bound on the variance
of an unbiased estimator. The root mean average of position estimation errors for L-LS-1, L-LS-3, L-LS-M, and L-LS-A were 18.4%, 12.4%, 4.3%, and 3.2% greater than CRLB, respectively. Among the newly proposed techniques, L-LS-M performs very close to L-LS-A, and is less complex.

In Chapter 7, we addressed the passive localization in an ultra-wide band (UWB) sensor network. UWB is preferred due to its fine time resolution and ability to distinguish multipaths from nearby objects. We defined a framework for 802.15.4a based sensor networks that use packet preamble for estimating multipaths reflected from target objects. Using this information, multipath distance between the source, target, and destination were computed. A multipath distance locates the target on an ellipse. The target can be located at the intersection of three or more such ellipses. We used neural network classifiers to solve this non-linear localization problem. Linear LS techniques, however, achieve a better performance in terms of position accuracy since they can explicitly use the underlying geometric relationship. Obtaining an analytic expression is not always possible in real world scenarios. We believe neural networks would be better at estimating the underlying continuous functional relationship between multipath distances and target location in such cases.

Multiple target detection for passive localization is more complex since it is not possible to identify which multipath is caused by which target. To solve this problem, we devised a two-step algorithm that uses single target detection as its building blocks and groups alternatives to reduce number of possible outcomes. We proposed a cost metric to be assigned for each group and selected the one with the smallest cost. We provided simulation results for locating two objects in a 3-node sensor network, and showed that performance of the two object detection algorithm is similar to single object detection.
Bibliography
