Spring 2009

Communication models for monitoring and mobility verification in mission critical wireless networks

Maheswaran Thiagarajan

Follow this and additional works at: http://scholarsmine.mst.edu/masters_theses

Part of the Systems Engineering Commons

Recommended Citation

COMMUNICATION MODELS FOR MONITORING AND MOBILITY VERIFICATION IN MISSION CRITICAL WIRELESS NETWORKS

by

MAHESWARAN THIAGARAJAN

A THESIS

Presented to the Faculty of the Graduate School of the MISSOURI UNIVERSITY OF SCIENCE AND TECHNOLOGY

In Partial Fulfillment of the Requirements for the Degree MASTER OF SCIENCE IN SYSTEMS ENGINEERING

2009

Approved by

Ann Miller, Advisor
Sriram Chellappan
Cihan H. Dagli
ABSTRACT

Recent technological advances have seen wireless sensor networks emerge as an interesting research topic because of its ability to realize mission critical applications like in military or wildfire detection.

The first part of the thesis focuses on the development of a novel communication scheme referred here as a distributed wireless critical information-aware maintenance network (DWCIMN), which is presented for preventive maintenance of network-centric dynamic systems. The proposed communication scheme addresses quality of service (QoS) issues by using a combination of a head-of-the-line queuing scheme, efficient bandwidth allocation, weight-based backoff mechanism, and a distributed power control scheme. A thorough analysis of a head-of-the-line priority queuing scheme is given for a single-server, finite queue with a batch arrival option and user priorities. The scheme is implemented in the Network Simulator (NS-2), and the results demonstrate reduced queuing delays and efficient bandwidth allocation for time-critical data over non time-critical data.

In the second part, we introduce a unique mobility verification problem in wireless sensor networks wherein the objective is to verify the ‘claimed’ mobility path of a node in a co-operating mission critical operation between two allies. We address this problem by developing an efficient power-control based mobility verification model. The simulation framework is implemented in Matlab and the results indicate successful detection of altered claimed paths within a certain error bound.
ACKNOWLEDGMENTS

I would like to thank my parents for their constant support and sacrifices without which I would not be where I am today.

I would like to acknowledge the help from my advisor, Dr. Ann Miller who made this work possible and the rest of my committee for reviewing my work.

Heartfelt thanks to Dr. Sriram Chellappan for his extensive advice and patient hours of discussion.

I would like to thank Dr. Cihan H. Dagli and the Systems Engineering Department for their support of this work through funding.

I would like to express my gratitude to Dr. Jagannathan Sarangapani for his guidance and motivation. Special thanks to Dr. Maciej Zawodniok and Anil Ramachandran for their ideas, efforts and help in writing the Communication Model paper.

I would like to thank Harini Kannan for being a constant source of motivation, support.

Finally I would like to thank my friends for supporting me through my thesis.
# TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>ABSTRACT</td>
<td>iii</td>
</tr>
<tr>
<td>ACKNOWLEDGMENTS</td>
<td>iv</td>
</tr>
<tr>
<td>LIST OF ILLUSTRATIONS</td>
<td>viii</td>
</tr>
<tr>
<td>LIST OF TABLES</td>
<td>x</td>
</tr>
<tr>
<td>1. INTRODUCTION</td>
<td>1</td>
</tr>
<tr>
<td>1.1 COMMUNICATION SCHEME FOR DISTRIBUTED WIRELESS SYSTEMS</td>
<td>2</td>
</tr>
<tr>
<td>1.2 WIRELESS SENSOR NETWORKS</td>
<td>2</td>
</tr>
<tr>
<td>1.3 MOBILITY VERIFICATION PROBLEM</td>
<td>4</td>
</tr>
<tr>
<td>2. COMMUNICATION SCHEME FOR A DISTRIBUTED WIRELESS CRITICAL INFORMATION-AWARE MAINTENANCE NETWORK</td>
<td>7</td>
</tr>
<tr>
<td>2.1 BACKGROUND</td>
<td>7</td>
</tr>
<tr>
<td>2.1.1 Time critical and Non time-critical Systems</td>
<td>7</td>
</tr>
<tr>
<td>2.1.2 QoS Parameters</td>
<td>7</td>
</tr>
<tr>
<td>2.1.3 Back-off Mechanism</td>
<td>8</td>
</tr>
<tr>
<td>2.1.4 Queuing</td>
<td>9</td>
</tr>
<tr>
<td>2.1.5 Distributed Power Control Algorithm</td>
<td>10</td>
</tr>
<tr>
<td>2.2 COMMUNICATION SCHEME FOR A DISTRIBUTED WIRELESS CRITICAL-INFORMATION-AWARE MAINTENANCE NETWORK</td>
<td>11</td>
</tr>
<tr>
<td>2.2.1 Weight-based Back-off Scheme</td>
<td>12</td>
</tr>
<tr>
<td>2.2.2 Head-of-the-line Priority Queuing Scheme</td>
<td>13</td>
</tr>
</tbody>
</table>
2.2.3. Distributed Power Control. ....................................................... 21

2.3 SIMULATIONS .................................................................................. 21

2.3.1. Simulation Setup. ...................................................................... 21

2.3.2. Simulation Parameters. .............................................................. 23

2.3.3. Performance Metrics. .............................................................. 23

2.4 RESULTS AND ANALYSIS ............................................................. 24

2.5 CONCLUSIONS ................................................................................. 33

3. MOBILITY VERIFICATION OF WIRELESS SENSOR NETWORKS ...... 35

3.1 BACKGROUND ................................................................................. 35

3.1.1. Problem Solution. ...................................................................... 37

3.1.2. Mean number of intersections. .................................................. 38

3.1.3. Sectorization algorithm. ............................................................ 38

3.2 DERIVING A STATISTICAL BOUND ON $\Delta L$ ............................... 38

3.3 ANALYSING THE REQUIREMENTS OF BASE STATION’S TX POWER ................................................................. 43

3.4 MOBILITY VERIFICATION OF A CO-OPERATING MOBILE NODE (MN) IN A WIRELESS SENSOR NETWORK (WSN) ...... 46

3.4.1. Setup Phase................................................................................ 47

3.4.2. Verification Phase. .................................................................... 51

3.5 SIMULATIONS .................................................................................. 55

3.5.1. Simulation Framework. ............................................................. 55

3.5.2. Simulation Parameters. .............................................................. 56

3.6 RESULTS ............................................................................................ 58

3.6.1. Basic test cases. ......................................................................... 59
3.6.2. Performance evaluation of the mobility verification model..... 66

4. CONCLUSION AND FUTURE WORK ....................................................... 73

BIBLIOGRAPHY ....................................................................................... 74

VITA ......................................................................................................... 78
# LIST OF ILLUSTRATIONS

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.1</td>
<td>Wireless Sensor Networks</td>
<td>4</td>
</tr>
<tr>
<td>1.2</td>
<td>Claimed Path vs Actual Path</td>
<td>5</td>
</tr>
<tr>
<td>2.1</td>
<td>Architecture of a wireless network with traffic patterns for prognostic applications</td>
<td>9</td>
</tr>
<tr>
<td>2.2</td>
<td>A node with two contending flows</td>
<td>10</td>
</tr>
<tr>
<td>2.3</td>
<td>DPC Feedback Loop</td>
<td>11</td>
</tr>
<tr>
<td>2.4</td>
<td>Simulation Setup</td>
<td>22</td>
</tr>
<tr>
<td>2.5</td>
<td>End-end delay for intra-cluster communication using 802.11</td>
<td>24</td>
</tr>
<tr>
<td>2.6</td>
<td>End-end delay for intra-cluster communication using DWCIMN</td>
<td>25</td>
</tr>
<tr>
<td>2.7</td>
<td>End-end delay for inter-cluster communication using 802.11</td>
<td>26</td>
</tr>
<tr>
<td>2.8</td>
<td>End-end delay for inter-cluster communication using DWCIMN</td>
<td>27</td>
</tr>
<tr>
<td>2.9</td>
<td>End-end delay for inter-cluster communication using 802.11</td>
<td>28</td>
</tr>
<tr>
<td>2.10</td>
<td>End-end delay for inter-cluster communication using DWCIMN</td>
<td>28</td>
</tr>
<tr>
<td>2.11</td>
<td>Frequency distribution of end-end delay for the non time-critical data</td>
<td>29</td>
</tr>
<tr>
<td>2.12</td>
<td>Frequency distribution of end-end delay for the time-critical data</td>
<td>30</td>
</tr>
<tr>
<td>2.13</td>
<td>Throughput comparison between the two types of traffic</td>
<td>31</td>
</tr>
<tr>
<td>2.14</td>
<td>Power consumed using DPC</td>
<td>32</td>
</tr>
<tr>
<td>2.15</td>
<td>Power consumed using 802.11</td>
<td>33</td>
</tr>
<tr>
<td>3.1</td>
<td>Impact of change in Length L on the number of intersections</td>
<td>39</td>
</tr>
<tr>
<td>3.2</td>
<td>Impact of change in Length ‘L’ on number of intersections (Theoretical)</td>
<td>41</td>
</tr>
<tr>
<td>3.3</td>
<td>A zoomed-in version of the previous graph which reveals the value of ΔL</td>
<td>42</td>
</tr>
</tbody>
</table>
3.4 : Comparison between the theoretical results vs. simulation results ......................... 43
3.5 : Calculation of 'D' based on the side of the sample space ................................. 44
3.6 : BS Transmission distance values for any given Length .................................... 46
3.7 : Network setup with two Base stations ................................................................. 47
3.8 : Simulation of a Uniform Random Deployment of a WSN ....................... 55
3.9 : Different types of straight line mobility pattern in the sample space ............ 57
3.10 : Horizontal mobility pattern .............................................................................. 60
3.11 : Slanting mobility pattern ................................................................................... 60
3.12 : Mobility Verification model - Claimed Path rejection – I ................................. 61
3.13 : Mobility Verification Model - Claimed Path Rejection – II ............................ 63
3.14 : Mobility Verification Model - Offset of claimed path ...................................... 64
3.15 : Mobility Verification Model - Opposite directions ............................................ 65
3.16 : Best case deployment scenario ......................................................................... 72
# LIST OF TABLES

<table>
<thead>
<tr>
<th>Table</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.1: Terms and Explanations</td>
<td>48</td>
</tr>
<tr>
<td>3.2: Algorithm for Power-control based Sectorization (Set-up)</td>
<td>50</td>
</tr>
<tr>
<td>3.3: Parameters and Values for simulation</td>
<td>57</td>
</tr>
<tr>
<td>3.4: Reason and result codes</td>
<td>58</td>
</tr>
<tr>
<td>3.5: Verification model results and codes when varying the offset value</td>
<td>67</td>
</tr>
<tr>
<td>3.6: Verification model results and codes when varying the offset value</td>
<td>68</td>
</tr>
<tr>
<td>3.7: Simulation (1) results when varying the <em>grid distance</em></td>
<td>70</td>
</tr>
<tr>
<td>3.8: Simulation (2) results when varying the <em>grid distance</em></td>
<td>71</td>
</tr>
</tbody>
</table>
1. INTRODUCTION

Wireless networks have been a commercial success because of their low-cost, accessibility, increased productivity, mobility and are designed in various modalities, ranging from wireless mesh to sensor networks. However, the requirements for a wireless network in network centric logistics and intelligent monitoring and maintenance applications are different from typical computer wireless ad hoc networking applications.

Network centric logistics typically gather information, which is used to make decisions in order to fulfill their mission. For a particular application, the information can be categorized based on its importance and delivery deadline. A robust, networked environment improves information sharing, collaboration, and event-awareness, and it results in an increased level of sustainability, speed, and system performance in intelligent maintenance applications.

With the deployment of wireless networks almost everywhere, this industry has had an unprecedented growth in the last decade. Though it had problems achieving the performance of a wired network initially, the continuous advent of new technologies and the IEEE 802.11 standards has helped break the performance barriers and meet the increased demand in speed and bandwidth requirements. The first part of this thesis focuses on designing a new wireless communication scheme for preventive maintenance scheduling in distributed mobile systems that have communication and memory constraint. The second part of the thesis, we introduce a unique ‘mobility verification’ problem and later proper a power-control based sectorization model for solving it.
1.1 COMMUNICATION SCHEME FOR DISTRIBUTED WIRELESS SYSTEMS

Monitoring and prognostics of time-critical and non time-critical information is vital for the correct operation of industrial and military applications. For such application environments, it is necessary to have a novel communication scheme that addresses QoS parameters in real-time. Providing differing QoS for wireless networks is not trivial in the case of a dynamic and shared wireless channel. Moreover, prioritization of the two different data classes is necessary to significantly increase the performance of the system. The contributions of the proposed work are as follows: (1) analytical derivation of the probability distribution functions of lost packets for a discrete-time queuing system with Markovian input, constant service time, finite buffer size (D-BMAP/D/1/K), and with priorities in a wireless network; and (2) development of a communication scheme that provides adequate QoS to time-critical data by using three key techniques: (a) head-of-the-line priority queuing for a batch arrival process; (b) a weight-based back-off mechanism for channel access, which renders minimal overhead; the initial weights are assigned by the user based on the critical nature of the flows; and (c) distributed power control to offer energy-efficient data communication and increased network throughput for wireless network-centric monitoring, diagnostic, and prognostic (M/D/P) application environments. The net result is the development and implementation of the new communication scheme for a distributed wireless critical information-aware maintenance network (DWCIMN).

1.2 WIRELESS SENSOR NETWORKS

With the deployment of wireless networks almost everywhere, this industry has had an unprecedented growth in the last decade. Though it had problems achieving the
performance of a wired network initially, the continuous advent of new technologies has helped break the performance barriers and meet the increased demand in performance and power-constraint requirements. Therefore, they are becoming more affordable, and consequently are designed in various modalities, ranging from wireless mesh to sensor networks.

It is imperative that any new technology with significant benefits has a number of frequently overlooked challenges. Mobility in networks is usually perceived as an advantage for portable computing and it has generated significant interest over the last few years. Mobility in WLAN can be viewed as an advantage where real-time information is readily available even through remote-access though the performance degrades slightly. Designing protocols that support mobility has always been challenging, however, mobility provides information regarding the mobile path that a wireless node has taken in a network. This additional information could be used along with intelligent algorithms to provide a host of applications (localization, source-tracking) in wireless ad-hoc and sensor networks. The potential opportunities available in mobile-centric networks motivate us to develop efficient ‘verification’ models for wireless networks.

As shown in Figure 1.1, Wireless sensor networks involve the deployment of a large number of small low-cost wireless nodes with the additional capability of sensing environmental changes. Wireless sensor networks promise a collaborative, distributed mobile computing environment with wireless connectivity in a number of hostile environments or in last-mile applications scenarios. A few applications include disaster recovery (fire, flood, earthquake), and mission-critical communications.
Wireless Sensor Networks has been used by the military for a few years now, where they are used for battlefield surveillance, source-tracking of mobile agents or track the enemy in areas of our interest. Mobility is an interesting concept in the WSN modality since it offers many potential application scenarios like target-tracking.

![Diagram of Wireless Sensor Networks]

Figure 1.1 : Wireless Sensor Networks

1.3 MOBILITY VERIFICATION PROBLEM

In the second part of this thesis, we intend to solve a new problem called ‘mobility verification’. The objective of the verification model is to either accept or reject a claimed path movement made by a mobile node in the area of our interest (E.g. Military
applications). The excessive execution time of the existing source tracking methods does not provide swift verification of the claimed mobility path by a co-operating ally in a wireless sensor network. **Figure 1.2** shows the claimed and actual path in a sample space.

![Figure 1.2 : Claimed Path vs Actual Path](image)

The mobility verification scheme suggested in this work has been modeled using co-ordinate geometric techniques along with computationally less-intensive algorithms to reduce the execution time. We design the verification model for a specific case wherein we do not have the position information of the sensors deployed in the network. The main contribution for the mobility verification work is the design of a two step algorithm which will validate the claimed path against the actual path.

Without position information of the WSN, two consecutive steps need to be followed. (1) A standard ΔL value for the sample space is calculated and this is compared
to the length traveled for the actual (L1) and claimed path (L2). The results of this step are used to calculate the mean number of intersections and are used to make a first level verification of the claimed vs actual path. (2) If the claimed path and actual path are deemed similar at the end of the first step, a two phase Neighbor selection and Sectorization technique is employed as a second step to further narrow down the results.
2. COMMUNICATION SCHEME FOR A DISTRIBUTED WIRELESS CRITICAL INFORMATION-AWARE MAINTENANCE NETWORK

2.1 BACKGROUND

This section presents an overview of a multi-hop wireless network for M/D/P environments. Also, the problems to be addressed are discussed.

2.1.1. Time critical and Non time-critical Systems. The presented work addresses the need for industrial monitoring, diagnostic, and prognostic (M/D/P) applications. Consider a distributed sensing application in a large industrial manufacturing plant. In such application environments, the measured data can be categorized into time-critical and non time-critical. The time-critical data includes information that is vital for correct operation of the systems and that needs to be delivered within a narrow time-window. For instance, the system level alarms generated during emergency situations are typical examples of time-critical data. However, such information is not transmitted often in the network, and generally it is viewed as a batch arrival process. In contrast, the non time-critical information provides supporting data to be used offline, such as in daily reports, and it is transmitted periodically. In either case, the collected data have to be transmitted to a base station or central controller to provide observability. For such application environments, it is necessary to have a novel communication scheme that addresses QoS parameters in real-time.

2.1.2. QoS Parameters. Providing differing QoS for wireless networks is not trivial in the case of a dynamic and shared wireless channel. Moreover, prioritization of the two different data classes is necessary to significantly increase the performance of the
system. The desired QoS can be achieved through differential service by using priority-aware queuing or by bandwidth control for certain flows as has been addressed by a number of papers in the literature [1-3]. The latter method has certain disadvantages; for example, it results in inaccurate allocation of resources when the network bandwidth is inefficiently used. Past literature has proposed queuing schemes including head-of-the-line priority queuing and its performance analysis [4-8]. Head-of-the-line queuing is the most drastic type of priority scheduling where the time-critical traffic has absolute priority over the non time-critical traffic. However, these schemes are not designed to address the issues with time-critical traffic with end-end delay constraints in the presence of other traffic types and that have batch arrival process [4-8].

In [9], the authors have derived the probability distribution function of lost packets for a discrete-time batch Markovian arrival process (D-BMAP) with finite queue length. However, a FIFO queue is modeled, while in our paper we derive the probability distribution functions of lost packets for a head-of-the-line (HOL) queue for the data arrival with priority assignment, which is representative of a network typically used for prognostic applications. In [15], the authors have modeled a weight-based adaptive fair scheduling scheme; however, this scheme ensures fairness based on the weights, and it might be unsuitable for the application environments under consideration.

2.1.3. Back-off Mechanism. Figure 2.1 shows a wireless with several nodes trying to access a shared channel network used for prognostic applications. Other medium access schemes such as CSMA/CD and CSMA/CA can be used to determine which nodes access the shared channel. A random back-off interval time is calculated in these schemes when a collision occurs. However, these schemes do not support service
differentiation based on the class in which the packet belongs because nodes will get proportional access to the channel regardless of the data type. Therefore, the proposed scheme uses the weight-based back-off interval scheme to satisfy the required QoS criterion.

![Architecture of a wireless network with traffic patterns for prognostic applications](image)

**Figure 2.1**: Architecture of a wireless network with traffic patterns for prognostic applications

**2.1.4. Queuing.** For prognostic applications, the timely delivery of packets is critical to achieve a target performance. Hence, limiting the end-to-end delay is of great concern. Otherwise, the overly delayed data may lose their informative value and render the overall system useless. In a generic wireless ad hoc network, fair scheduling schemes implement a distributed algorithm to achieve service for each flow in a network. However, in a network used for prognostic applications, desired QoS will vary between
flows based on the importance of the information. In addition, lifetime and fairness are not as important as delivering the time-critical data.

**Figure 2.2** illustrates two different classes of data contending for buffer storage on the output link. In generic wireless networks, the packets are stored in an FIFO queue, which may be unfair to the time-critical data flow. However, in this paper the head-of-the-line priority queuing scheme is extended to the case of multiple flows to provide QoS based on flow importance, where the time-critical information experiences a smaller service time and reduced queuing delay.

![Figure 2.2: A node with two contending flows](image)

**2.1.5. Distributed Power Control Algorithm.** Besides the queuing scheme, the transmitter power control is exerted in order to save battery power. The objectives of transmission power control include minimizing power consumption while increasing the network capacity and prolonging the battery life by managing mutual interference effects in the network.
Optimal selection of the transmission power level will improve energy efficiency of the communication and throughput through increased spatial reuse. A feedback loop as shown in Figure 2.3 is used between the transmitter and receiver in order to successfully implement the DPC [10]. Node 1 transmits data to node 2, and a feedback loop is established between the transmitter and the receiver in order to successfully implement the DPC algorithm [10].

![Figure 2.3 : DPC Feedback Loop](image)

**2.2 COMMUNICATION SCHEME FOR A DISTRIBUTED WIRELESS CRITICAL INFORMATION AWARE MAINTENANCE NETWORK**

In this section, the flow-assigned priorities for the packets (or flows) are used during both queuing and channel access. The priority will dictate the order in which packets from the contending traffic categories are queued. Also, the priorities are used to vary the back-off interval during the channel access procedure. On the other hand, the
transmission power control will increase the network capacity. These are described in detail in this section.

In the proposed work, the HOL queuing technique with two priority categories, time-critical and non time-critical, is used at the scheduling level. Implementation of the classifier at every relay node ensures prioritized treatment of the time-critical information throughout the network. The time-critical data that are received at a node are scheduled separately to ensure reduced delays.

Subsequently, the probability distribution function of lost packets is derived. This function guarantees that the desired QoS is achieved by the scheme. The presented analysis is based on the work done by Moltchanov et al, which derives the PDF of lost packets for a D-BMAP process for different flows without priorities [9]. In contrast, this paper considers the prioritization of the flows with the arrival rate as a D-BMAP process with a finite queue size. The packet delay in the time-critical flows is unaffected by the non time-critical flows because the high priority packets are expedited to the front of the queue.

2.2.1. **Weight-based Back-off Scheme.** The proposed communication scheme varies the back-off interval based on the flow priority. When multiple nodes of a wireless network compete to access the shared channel, the back-off interval determines which node is granted access to the channel. In order to achieve differential QoS for the time-critical packets, the nodes must access the channel in a prioritized manner.

The proposed scheme is implemented at the MAC layer by adjusting the back-off intervals based on the weight assigned initially to indicate the flow priority. The back-off interval, $\xi_i$, for the $i^{th}$ flow is defined as
\[ \zeta_i = \left[ \Lambda^* \Phi_i^* \varphi_i \right] \]  

where \( \Phi_i \) is a priority-assigned weight set by the user, \( \varphi_i \) is the fair back-off interval, \( \Lambda \) is a random variable with a mean of 1, and \( i \) is either 1 or 2 based on the flow for the current example. Next, the analytical performance of the head-of-the-line queuing scheme is introduced.

### 2.2.2. Head-of-the-line Priority Queuing Scheme.

The basic definition of the discrete-time batch Markovian arrival process that takes into consideration the structure and behavior of autocorrelation is presented in [9]. The input rate vector of the D-MAP process, \( \vec{I} = (I_1, I_2, \ldots, I_M) \), whose elements are defined as [9]

\[
I = \sum_{j=1}^{\infty} \sum_{k=1}^{\infty} k d_j(k), i \in \{1, 2, \ldots, M\}
\]  

(2)

The input rate at the time slot \( n \) can be defined by a random variable \( I \) that takes the values of the vector \( \vec{I} = (I_1, I_2, \ldots, I_M) \) with respective probabilities \( \vec{\pi} = (\pi_1, \pi_2, \ldots, \pi_M) \). The input rate process of a D-BMAP \( \{W(n), n = 0,1,\ldots\} \) is defined by a Markov modulated process \( \{I(n), n = 0,1,\ldots\} \) with \( I(n) = I_i, i \in \{1,2,\ldots,M\} \), while the Markov chain is state \( i \) at time slot \( n \). The conditional probability distribution function (CDPF) of the D-BMAP process is given by \( d_j(k) \), where \( k = 0,1,\ldots \) and \( i, j \in \{1,2,\ldots,M\} \). The autocorrelation function, \( K^{[I]} \) for the Markov modulated process, \( \{I(n), n = 0,1,\ldots\} \), is given by [11]
\[ K^{[1]} = \sum_{\forall l, l \neq 1} A_l^T \ell l, \]  

(3)

Where

\[ \ell l = \pi \left( \sum_{k=1}^{\infty} kD(k) \right) \tilde{g}_l \tilde{h}_l \left( \sum_{k=1}^{\infty} kD(k) \right) \bar{e}, \]  

(4)

where \( D(k) \) is the transition matrix of the Markov chain; \( \lambda_l \) is the \( l^{th} \) eigenvalue of \( D(k) \); \( \tilde{g}_l \) and \( \tilde{h}_l \) are the \( l^{th} \) right column and left row eigenvectors of \( D(k) \) respectively; and \( \bar{e} \) is an appropriate vector of ones.

2.2.2.1 Queuing System. The batch arrival process for the queuing system is assumed to follow the system time diagram shown in [12]. Then, the dynamic state equations of the queuing system for the two different-priority flows can be expressed as

\[ S^{[\text{QP}]}(n+1) = \max[0, S^{[\text{QP}]}(n)+ \]  
\[ + \min[(W^{[\text{QP}]}(n+1), K - S^{[\text{QP}]}(n)] \]  

(5)
\[ s^{[\{o\}}(n + 1) = \max\{0, \max\{0, s^{[\{o\}}(n)\} + \]
\[ + \min\{0, K - s^{[\{o\}}(n) - s^{[\{o\}}(n) - w^{[\{a\}}(n + 1)\}]\] (6)

where the state of the system (number of packets) at the time of packet departures for the two flows is given by \( s^{[\{o\}}(n + 1) \) and \( s^{[\{o\}}(n + 1) \in \{0, 1, \ldots, K\} \) and \( w^{[\{a\}}(n + 1) \) denote the number of arrivals of the time-critical and non time-critical traffic, respectively, in the \((n + 1)^{th}\) time slot.

2.2.2 Probability Distribution Function of Lost Packets from Time-Critical Sources. The distribution function for the time-critical traffic can be modeled as if the queue were FIFO because it is given higher priority over the non time-critical packets in the queue. Additionally, when the number of arriving time-critical packets exceeds the available buffer space, the non time-critical packets are dropped from the queue. Hence, the distribution function can be derived directly from the probability function of lost packets for the superposed process from Moltchanov et al. [9].

Let us consider the event when \( v^{[\{o\}} \) packets from the arrival process are lost, the probability distribution function, \( p^{[\{o\}]}(v^{[\{o\}}) \), can be given by

\[ p^{[\{o\}]}(v^{[\{o\}}) = \Pr\{o^{[\{o\}} = v^{[\{o\}}\}, v^{[\{o\}} = 1, 2, \ldots \] (7)

where \( o^{[\{o\}} \) is the random variable corresponding to the number of time-critical packets lost during time slot \( n \).
At a given time, $\kappa_1$ time-critical packets are in the system. Hence, $K - \kappa_1$ packets can be accommodated in the queue. In the case of dropped packets, there will be $K - \kappa_1 + v^{[p_1]}$ time-critical packet arrivals.

Now, all the possible transitions for the Markov chain have been considered, and the distribution function is given as

$$p_o^{[p_1]}(v^{[p_1]}) = \sum_{k=0}^{K} \bar{x}_k D(K - \kappa_1 + v^{[p_1]}) \bar{e}, v^{[p_1]} = 1, 2,...$$

(8)

$$p_o^{[p_1]}(v^{[p_1]}) = 1 - \sum_{i=1}^{\infty} \sum_{k=0}^{K} \bar{x}_k D(K - \kappa_1 + v^{[p_1]}) \bar{e}, v^{[p_1]} = 0$$

where $\bar{x}_k$ is the row array that contains stationary probabilities of the two-dimensional Markov chain.

**2.2.2.3 Probability Distribution Function of Lost Packets from Non Time-Critical Sources.** In the case of non time-critical flows, the probability function of lost packets has to include packets dropped due to both limited buffer space and preemption by time-critical packets.

Let $v^{[p_2]}$ denote the packets generated from the arrival process that are lost. Then, the probability distribution function is expressed as

$$p_o^{[p_2]}(v^{[p_2]}) = Pr\{d^{[p_2]} = v^{[p_2]}\}, v^{[p_2]} = 1, 2,...$$

(9)
where \( o_i^{[p_2]} \) is the random variable corresponding to the number of non time-critical packets lost during time slot \( n \).

The packets are dropped when at least one of the following conditions are true:

There are \((\Gamma - 1)\) packets in the system, where \(\Gamma = (\kappa_1 + \kappa_2)\) and \(\kappa_1, \kappa_2\) are the existing time-critical and non time-critical packets, respectively. Consequently, the queue can accommodate \((K - \Gamma)\) packets.

There are \(k^{[p_1]}\) non time-critical packets arriving, where \(k^{[p_1]} \geq v^{[p_2]}\) and \(k^{[p_1]} \leq (K - \Gamma + v^{[p_2]}\).

There are \(k^{[p_1]}\) time-critical packets arriving to the system.

There are \(v^{[p_1]}\) lost packets between \((K - \kappa_1 + 1)\) and \(\eta = k^{[p_1]} + v^{[p_2]}\) in the arrival batch. Let \(v^{[p_2]} = v^{[p_1]} + v^{[p_2]}\) where \(v^{[p_2]}\) corresponds to the number of packets dropped from the arrival process, and \(v^{[p_2]}\) is the number of packets dropped from the existing queue.

Now the distribution function (9) can be rewritten as

\[
p_{o}^{[p_1]}(v^{[p_2]}) = \sum_{i=0}^{K} \sum_{\substack{\eta = 0 \leq v^{[p_2]} \leq \eta \leq K}} \sum_{\substack{k^{[p_1]} = 0 \leq k^{[p_1]} \leq k^{[p_1]}} \eta} 1 \times \\
\times \chi(k^{[p_1]}, k^{[p_1]}, K - \Gamma) \\
\times \xi^{\eta} \eta (v^{[p_2]})
\]  

(10)
where $\chi(k^{[p_1]}, k^{[p_1]}, K - \Gamma)$ is the probability that $k^{[p_1]}, k^{[p_1]}$ packets are arriving for the two different priorities. This term can be expressed as

$$\chi(k^{[p_1]}, k^{[p_1]}, K - \Gamma) = \sum_{l=1}^{M^{[p_1]}} \sum_{h=1}^{M^{[p_2]}} \psi_l(k^{[p_1]}) \times \psi_h(k^{[p_1]}) \times x((K - \Gamma), (l, h))$$

(11)

where $x((K - \Gamma), (l, h)), l \in (1, 2, ..., M^{[p_1]}), h \in (1, 2, ..., M^{[p_2]})$ is a probability that there are $(K - \Gamma)$ cells are in the system and is given by the stationary distribution of the two-dimensional Markov chain. The terms $\psi_l(k^{[p_1]})$ and $\psi_h(k^{[p_1]})$ are the probabilities that the time-critical and non time-critical flows change their states. They are given by [9] as

$$\psi_l(k^{[p_1]}) = \sum_{j=1}^{M^{[p_1]}} d_{lj}^{[p_1]} k^{[p_1]}, l \in (1, 2, ..., M^{[p_1]})$$

$$\psi_h(k^{[p_1]}) = \sum_{j=1}^{M^{[p_2]}} d_{hj}^{[p_1]} k^{[p_1]}, h \in (1, 2, ..., M^{[p_2]})$$

(12)

The second term in (10) is the probability that there are $\eta^{[p_1]}$ packets of the non time-critical traffic are between the $(\kappa_1 + 1)^{th}$ and $\eta$ packet of the batch arrival.

The arrival batch includes $\eta = k^{[p_1]} + k^{[p_2]}$ packets. However, the buffer space available in the finite queue is $(K - \Gamma)$ packet and hence only these can be accommodated.
by the system in a given time slot. Non time-critical packets from \( k^{[p_1]} \) will be accommodated only if there is buffer space after accommodating all the arriving time-critical packets \( k^{[p_1]} \). Additionally, if the number of time-critical packets is more than the available buffer space, then the non time-critical packets are dropped from the queue to accommodate them. Therefore, the probability that exactly \( v^{[p_2]} \) non time-critical packets are within the \( \eta=(K-\Gamma) \) packets that cannot be accommodated has to be derived.

There can be \( Z^{k^{[p_2]}}_\eta = \frac{\eta!}{\eta-(\eta-k^{[p_2]})!} \) combinations of \( k^{[p_2]} \) packets from the non time-critical sources in an arrival batch of size \( \eta \). Additionally, there can be

\[
Z^{[p_{21}]}_k = \frac{k^{[p_2]}!}{v^{[p_2]}!(\eta-k^{[p_2]}-v^{[p_{21}]})!} \text{ combinations of } v^{[p_2]} \text{ lost packets from the non time-critical sources.}
\]

The overall number of time-critical packets is \( (n-k^{[p_1]}) \). Next, there are

\[
Z^{(K-k^{[p_1]},v^{[p_2]})}_{(n-k^{[p_1]})} = \frac{(n-k^{[p_1]});}{(K-k^{[p_1]}+v^{[p_2]});(\eta-k^{[p_2]}-v^{[p_{21}]})!} \text{ combinations of } (K-k^{[p_1]}+v^{[p_{21}]}); \text{ packets in } (n-k^{[p_2]}); \text{ packets.}
\]

Let us define, \( v^{[p_{22}]} \) as the number of non time-critical packets that will be dropped from the existing queue. There are \( Z^{K_2}_{\Gamma} = \frac{\Gamma!}{\kappa_2!(\Gamma-\kappa_2)!} \) combinations of \( \kappa_2 \) non time-critical packets in the system at any given time slot \( n \). Hence, there can be

\[
Z^{v^{[p_{22}]} }_{\kappa_2} = \frac{\kappa_2!}{v^{[p_{22}]!}!(\kappa_2-v^{[p_{22}]})!} \text{ combinations of } v^{[p_{22}]} \text{ lost packets from the existing non time-}
\]
critical packets. The remaining time-critical packets will have

\[ Z_{(K - \kappa_2 - v^{[p_2]})}^{(K - \kappa_2, -v^{[p_2]})} = \frac{(\Gamma - \kappa_2)!}{(K - \kappa_2 - v^{[p_2]})!(\Gamma - \kappa_2) - (K - \kappa_2 - v^{[p_2]}))!} \] combinations.

Consequently, the probability \( \xi_{\kappa_1 + v^{[p_2]}}^{p_2} \) that there are \( v^{[p_2]} \) packets of the non
time-critical traffic in the batch arrival can be written as

\[ \xi_{\kappa_1 + v^{[p_2]}}^{p_2} = \frac{Z_{K - K_2 - v^{[p_2]}}^{(K - K_2, -v^{[p_2]})}Z_{(K - K_2 - v^{[p_2]})}^{(K - K_2, -v^{[p_2]})}}{Z_{K_2}^{K_2}Z_{(K - K_2 - v^{[p_2]})}^{(K - K_2, -v^{[p_2]})}} \] (13)

Substituting equations, (13), (10), and (11) into (9), the probability distribution
function of the non time-critical packets can be obtained as

\[ p_o^{[p_2]}(v^{[p_2]}) = \sum_{i=0}^{K - M^{[p_1]}} \sum_{k=1}^{M^{[p_1]}} \sum_{l=1}^{M^{[p_2]}} \sum_{h=1}^{M^{[p_2]}} 1 \times \]

\[ \times \left( \sum_{i=1}^{M^{[p_1]}} d_i^{[p_1]} \right) \times \left( \sum_{h=1}^{M^{[p_2]}} d_h^{[p_2]} \right) \times \]

\[ \times \frac{Z_{K - K_2 - v^{[p_2]}}^{(K - K_2, -v^{[p_2]})}Z_{(K - K_2 - v^{[p_2]})}^{(K - K_2, -v^{[p_2]})}}{Z_{K_2}^{K_2}Z_{(K - K_2 - v^{[p_2]})}^{(K - K_2, -v^{[p_2]})}} \] (14)

where \( v^{[p_2]} = 1, 2, \ldots n \). The case where \( v^{[p_2]} = 0 \) can be derived as in (8).
2.2.3. **Distributed Power Control.** Unlike wired channels that are stationary and predictable, radio channels involve many uncertain factors that are difficult to analyze. This paper focuses on these main channel uncertainties, such as path loss, Shadowing, and Rayleigh Fading. Hence, the DPC scheme in the presence of channel uncertainties is implemented as part of the new communication scheme. Zawodniok et al. discussed in detail the development and implementation of the DPC [10].

2.3 **SIMULATIONS**

2.3.1. **Simulation Setup.** The networks implemented in prognostic applications often use multi-hop communication. A communication scheme with a three-level hierarchical topology is simulated. It consists of the following:

*Clusters* – a number of co-located, network-enabled agents that form a cluster, each with a cluster head and a number of communicating sensor nodes.

*Network-enabled agents* – stations with equipment and supplies which have information to transmit. The agents inside a cluster can communicate with each other.

*Cluster heads* – the cluster heads are responsible for the routing and they relay information from/to nodes and the base station. Each of the cluster heads can communicate to agents in its cluster, other cluster heads, and the base station.

*Base station* – the kernel node that handles the high-level monitoring and system level event recording. This station has a central database.

Communication messages can include updates about supply levels and required components (e.g., for maintenance) or operation-related information and control, which are time-critical information.
Communication in the network includes agents randomly communicating with the following:

- Other agents within a cluster,
- Agents in other clusters, and
- The base station

Agents randomly generate two different classes of data: time-critical and non-time-critical messages. Additionally, a number of network-management packets are generated, for example during route establishment. Random packet generation time was assumed to ensure a realistic traffic pattern.
2.3.2. **Simulation Parameters.** The performance of the proposed scheme was evaluated using the Ns2 simulator by modifying the Ns2 in order to implement the new communication scheme [13]. Changes were made in (1) the MAC protocol to adjust the back-off interval, and (2) the queuing scheme (IFQ) to implement packet scheduling. The following values were used for all the simulations, unless otherwise specified: the channel bandwidth is taken as 2 Mbps, a “two-ray ground” propagation model with a “path-loss exponent” of 4.0 is used with $\alpha = 0.75$ and $\beta = 0.25$, and $\rho$ is a random variable. The AODV protocol was used for routing (however, the scheduling algorithm is independent of routing protocol). The Pareto traffic was setup for sources, the queue limit at each node was taken as 50, and the packets size was taken as 1040 bytes unless otherwise stated. The flows with the same traffic priority are assigned the same back-off weight. The weights are selected such that the overall sum of weights of all the flows is equal to unity.

2.3.3. **Performance Metrics.** In this paper, the end-end delay, frequency distribution of end-end delays, total power consumed for transmission, throughput, and fairness index are selected for performance evaluation of the proposed scheme. The end-end delay comparison is accomplished for the three types of communication possible in the network. The total power consumed by all nodes is used to evaluate energy efficiency. The simulations were repeated 10 times, and the results were averaged to ensure accuracy of the outcome.
2.4 RESULTS AND ANALYSIS

The end-to-end delay with simulation time is presented in Figure 2.5 and Figure 2.6. The results for the time-critical and non time-critical data using the 802.11 standard and the proposed scheme are presented and compared. Intra- and inter-cluster communication is considered.

![Figure 2.5 : End-end delay for intra-cluster communication using 802.11](image-url)
Figure 2.5 shows the end-end delay in intra-cluster communication using 802.11. The difference between the flows is due to random calculation of back-off intervals and burstiness of traffic from the Pareto sources. Figure 2.6 presents the end-end delay comparison for the two flows using the new communication scheme. The differential QoS provided by the new communication scheme using the weight-based back-off resulted in lower delay for the time-critical traffic. Additionally, the delay jitter was reduced because variability was reduced in back-off interval and queuing.

Figure 2.7 shows the end-end delays using 802.11 in inter-cluster communication. The FIFO queue and CSMA/CA MAC protocol used in these simulations do not differentiate between time-critical and non time-critical flows. As a
result, the delay for both types of data experiences similar delay levels and variation. The results for the proposed differential communication scheme are shown in Figure 2.8. The end-to-end delay for time-critical data is reduced because the proposed scheme renders preferential treatment to the priority packets. The same performance differentiation is observed as in the inter-cluster communication scenario because the proposed queuing and back-off techniques are used at each node.

![Figure 2.7: End-end delay for inter-cluster communication using 802.11](image-url)
Figure 2.8: End-end delay for inter-cluster communication using DWCIMN.

**Figure 2.9** and **Figure 2.10** depict the end-end delay using 802.11 and the new communication scheme during transmissions from the agents to the base station, respectively. Analogous to results in **Figure 2.7, Figure 2.8, Figure 2.9, Figure 2.10**, the proposed scheme decreases the end-to-end delay for the time-critical data because the packets are placed at the front of the queue, and applied back-off intervals are smaller than for non time-critical packets.
Figure 2.9: End-end delay for inter-cluster communication using 802.11

Figure 2.10: End-end delay for inter-cluster communication using DWCIMN
In summary, the end-to-end delay for time-critical data is significantly lower in all of the cases discussed above using the new communication scheme because all the time-critical packets are given high priority and are efficiently de-queued. In the case of channel access, the node with time-critical data is given priority over the other nodes, which in turn ensures differential QoS.

Figure 2.11 and Figure 2.12 show the frequency distribution of the end-end delay observed for the two different flows. The proposed communication scheme has a consistently smaller delay. Additionally, the end-to-end delay for the time-critical packets is lower than in the case of non-time-critical packets because the vital data packets are sent faster at the cost of increased delay experienced by the other packets.

Figure 2.11 : Frequency distribution of end-end delay for the non time-critical data
The flow fairness is evaluated using a fairness index \( (FI) \) [14], which is defined as

\[
FI = \frac{\left( \sum_f \frac{T_f}{\phi_f} \right)^2}{\eta \cdot \sum_f \left( \frac{T_f}{\phi_f} \right)^2}
\]  

\( (15) \)

where \( T_f \) is the throughput of flow \( f \), and \( \eta \) is the number of flows. The fairness indices were calculated for the topology using the two communication schemes. It was found to be 0.99 using DWCIMN and 0.94 for 802.15. The channel capacity was not
exceeded, and thus all data packets were transmitted. Hence, to evaluate performance in the case of the congested channel, the number of sources was altered.

**Figure 2.13** shows a comparison of the throughput for the nodes transmitting time-critical information using the new communication scheme and 802.11. The distributed power control scheme reduces the mutual interferences in the network and efficient scheduling, and the back-off scheme offers the time-critical nodes more bandwidth in the channel when in need. In the case of the 802.11 protocol, the throughput decreased with the number of active sources because the time-critical traffic has to share the bandwidth and buffers with non time-critical packets. Consequently, the time-critical flows experienced a higher number of lost packets.

![Figure 2.13: Throughput comparison between the two types of traffic](image)
Figure 2.14 and Figure 2.15 show the total power consumed for the transmissions through the simulation time when using the proposed scheme and 802.11, respectively. When compared with 802.11, the energy consumption is significantly lower for the proposed scheme consumption because the usage of DPC improved the spatial reuse. Additionally, the usage of DPC improved the throughput and QoS. Consequently, the proposed scheme increases the node and network lifetime.

![Figure 2.14 : Power consumed using DPC](image)
2.5 CONCLUSIONS

In this section, the distributed wireless critical information-aware network for maintenance applications is presented. The probability distribution function of lost packets for a D-BMAP process with priorities has been derived providing an analytical guarantee of better QoS for the time-critical traffic. The simulation results show that the differential QoS is provided, thus meeting the requirements of an M/D/P-based maintenance network. The results show the effectiveness of the implemented scheme against 802.11. Overall, the performance of the system is increased in terms of throughput via increased spatial reuse.

The performance of the proposed scheme is verified using various metrics: end-to-end delay, throughput, and energy consumption. The results show that the proposed
protocol fulfills the promise of reduced end-end delays. Hence, the scheme can be effectively applied to a maintenance network to both enhance the performance and improve the handling of critical information.
3. MOBILITY VERIFICATION OF WIRELESS SENSOR NETWORKS

3.1 BACKGROUND

The advantages provided by the wireless sensor networks have seen the development of many interesting applications based on location information of the sensors deployed in the network. A very common application is the localization of sensors in the network and target tracking of mobile nodes in the deployed network. In this section, we intend to solve a relatively new problem in the area of wireless sensor networks – “Mobility verification modeling.” The core objective of our approach is to provide a lightweight solution for the mobility verification problem.

Consider the sample deployment of wireless nodes as shown in Figure 1.2. In addition, let there be two allies (A & B) that are involved in a mission of monitoring the space as shown in Figure 1.2. Ally A is the dominant one and it often issues orders to Ally B. These orders demand B to physically travel through the area under surveillance to carry out operations. Though A and B are co-operating allies, A would still want to check if B took the path that it claims to have taken.

There are a number of solutions proposed in existing literature in the area of real-time localization and source tracking applications using wireless sensor networks since they provide an attractive approach for spatial monitoring. Though WSN makes these applications relatively easy to achieve, it places a heavy demand on their energy consumption levels. Therefore, a tradeoff has to be made between the total energy consumption, resources, the accuracy of the results and the time taken to achieve the results.
Verification models in wireless sensor networks have been developed for many applications. Location verification is one such concept, wherein the location claim made by sensors in the network is verified by various algorithms. In [19] propose two lightweight location verification algorithms to detect any abnormal location claims made by the sensor nodes and withstand attacks. In [20] have proposed a location verification scheme which can defend itself against attacks based on distance, RSSI and RTT. This work concentrates more on the establishment of a secure communication channel which is used later by the location verification schemes.

However, these solutions will not be applicable to our problem for three reasons:
(1) Our intention is to verify the path taken by the mobile node in the network after it completes its movement
(2) The mobile node is not a part of the wireless sensor network
(3) We do not intend to use RSSI or RTT as a part of our scheme since algorithms involving RSSI are computationally intensive.

Tracking targets via a wireless sensor network is a very challenging, multifaceted problem and several research groups have tackled various aspects of it [21-26]. In [27] the authors have proposed decentralized source localization and tracking in wireless sensor networks. However, there is a significant difference between the problems suggested above and the one that we are intending to solve. Our objective is to use the deployment of the wireless sensor networks and simply ‘verify’ the path taken by the Ally B in Ally A’s wireless sensor network by using the least amount of resources.

The source tracking problem is an interesting area of research that is often confused with the problem we are trying to solve, because in both the overarching aim is to monitor the movement of a mobile node. Our problem differs from the source-tracking
question because our aim is to simply assert whether the claim is ‘right’ or ‘wrong’. It does not involve the tedious calculation of the probable path taken by the mobile node or to predict the various paths that the mobile node would take in the future.

To solve this, we have designed a two-step power-control based approach for mobility verification of a co-operating mobile node in a wireless sensor network. The two steps include the verification using (1) Mean number of intersections and (2) Sector-based verification algorithm (SEBA).

3.1.1. Problem Solution. The Mobility verification model is divided into two main phases (1) the setup phase and (2) the verification phase. The setup phase involves all the operations that are performed once initially when the network is setup or when the sample space to be monitored is chosen. The verification phase includes the two steps involving the mean number of intersections and the sectorization algorithm.

The knowledge of sensor nodes’ position or network deployment information would immensely reduce the resources needed by the algorithm. Hence, a good approach would be to develop localization or use an existing scheme. But, we need to evaluate the option of developing or using an existing localization scheme against the advantages that it can provide. Because, the previously discussed localization schemes will exhaust a considerable amount of resources in the setup phase alone. And our intention is to develop a method that uses minimum resources but achieves a very high level of accuracy in differentiating the claimed path from the actual path. Therefore, a power control based grid approach has been developed which would not only reduce the usage of resources but in addition give very accurate results making it the most optimum solution for the problem.
3.1.2. **Mean number of intersections.** We intend to use the mean number of intersections property from [28] to perform a first-level check on the claimed path. The mean and variance of number of intersections for the sample space is calculated by with respect to change in length is calculated for the sample space using theorem 4.3. Using the setup information, we estimate the grids through which the node has actually travelled. With this information the distance travelled at the center of these grids is calculated which gives us an approximate estimate of the actual length travelled. After estimating the actual and claimed lengths, the number of intersections is calculated using [28]. The mean and variance of the number of intersections in a given sample space are then used to estimate the accuracy of the claimed profile.

3.1.3. **Sectorization algorithm.** This method is employed to further validate the claimed path against the actual path travelled by the sensor node after the first level verification using the mean number of intersections is completed. The basic principle behind this scheme is to divide the sample space into sector matrices and allow the two base stations to broadcast their information. By comparing the claimed profile between the base stations, a comprehensive verification can be performed. Section 3.3 gives a complete picture of how sectorization works.

### 3.2 DERIVING A STATISTICAL BOUND ON AL

In our analysis, we model the deployment of the sensors as a spatial point process as illustrated in [28], where the point process governs the location of events(sensors) in a bounded area $X$, which is a subset of $\mathbb{R}^2$.

The expected number of intersections of a straight Line $I$ in $\mathbb{R}^2$ with length $L$ is derived as theorem 1 in [28]. We leverage that concept to estimate the change in the
number of intersections which will help us differentiate the “Claimed” from the “Actual” path in our research problem when there is a change in the length L in a scenario like the one shown below in Figure 3.1

Figure 3.1 : Impact of change in Length L on the number of intersections

Let \( \alpha \) denote the total number of intersections of line \( I \) with parts of boundaries of covering discs. A disc is nothing but a sensor located at a point \( \xi_i \) with a sensing range \( r_i \). From [28], the expected number of intersections is expressed as

\[
\alpha^* = 4\lambda \cdot L \cdot r_{\text{mean}}
\]  

(16)

where,

\( \lambda \) = density of the driving point process
L = length of the line

$r_{\text{mean}} = \text{mean of all the sensing ranges of the sensors}$

Let the length of the “Actual” Path be $L_1$ and the length of the “Claimed” path be $L_2$. Now, the mean number of intersections can be expressed as

\begin{equation}
\alpha_1 = 4\lambda . L_1 . r_{\text{mean}}
\end{equation}

\begin{equation}
\alpha_2 = 4\lambda . L_2 . r_{\text{mean}}
\end{equation}

To derive a statistical bound on $\Delta L$, lets subtract (2) and (3)

\begin{equation}
(\alpha_1 - \alpha_2) = (4\lambda . L_1 . r_{\text{mean}}) - (4\lambda . L_2 . r_{\text{mean}})
\end{equation}

Our intention here is to derive the minimum change in the length which will have an impact on the mean number of intersections. Hence, lets equate $(\alpha_1 - \alpha_2) = 1$,

\begin{equation}
(4\lambda . L_1 . r_{\text{mean}}) - (4\lambda . L_2 . r_{\text{mean}}) = 1
\end{equation}

\begin{equation}
\Delta L = (L_1 - L_2) = 1/(4\lambda . r_{\text{mean}})
\end{equation}

To verify the bound, a few simulations were performed and the results of one specific case are presented in Figure 3.2 and Figure 3.3. The length of the line was varied from 0 to Maximum_value (Diagonal length). The mathematical calculation and the graphs from the simulations are shown below. From the results, it is evident that equation (5) is valid.
Sample space dimensions = [Length Breadth] = [400 300] ; \( r_{\text{mean}} = \text{sensitivity\_range} = 50; \) Total number of sensors = 100 ; \( \lambda = \text{total\_sensors} / \text{Area\_sample\_space} = 100/120000 = 1/1200 \)

\[
\Delta L = 1/ (4*(1/1200)*50) = 6
\]  

(21)

Figures 3.2 : Impact of change in Length ‘L’ on number of intersections (Theoretical)

From Figure 3.2, we can see that the impact of change in Length ‘L’ is linear and the number of intersections changes as the length changes. A zoomed in version of the same graph is shown in the following figure to illustrate that simulations relate to the calculated theoretical values.
Now that the statistical bound for delta L has been derived, we performed six simulations with different sensor distribution patterns to compare the theoretical results with the simulation values. In this setup, the length of the path taken inside the sample space was varied similar to the theoretical setup and the actual number of intersections from the simulation was calculated. The average of the six different simulations was taken and is compared against the theoretical values as shown in the Figure 3.4. We observe that the practical values follow the theoretical setup very closely indicating that the ‘Delta L’ property is stable and can be used for decision making. By performing some more simulations, we have a good chance of matching the practical with the theoretical curve.
3.3 ANALYSING THE REQUIREMENTS OF BASE STATION’S TX POWER

Our sectorization algorithm model requires the Base Station (BS) to be position at a distance D away from the center of the sides of the sample space under surveillance. This setup is quite critical for the performance of the algorithm. Hence, in this section we evaluate the possibility of this arrangement and theoretically calculate the distances D at which the BS has to position itself from the sides of the sample space.

Figure 3.5 shows a simple representation of how the distance D is calculated for a given sample space. The calculation of D is very critical for the BS to be able to sectorize the sample space into grids. We know that wireless sensor networks are deployed in mission-critical systems and hence increasing the power to a very high level should not be a problem. Let’s assume the BS station has sufficient power to adjust the distance D and broadcast from a long distance.
Figure 3.5: Calculation of 'D' based on the side of the sample space

The main aim of this approach is to calculate D for any given length 'C' [the distance between CT and PT in Figure 3.5] such that the line CT-PT is very close to the arc CT-PT. Now, let us draw two lines to the points CT and PT and also drop a perpendicular from the center of the circle to the line CT-PT. If the angle created between the perpendicular and the line 'r' be $\Theta$, then by geometry, a tangent passing through the circle at points CT or PT will be at an angle $\Theta$ with the line CT-PT. Therefore, we can say that
\[ C = 2r \sin \Theta \quad (22) \]

Now, we need to fix a value for \( \Theta \) such that we can approximate \( D = r \). By virtue of simulations, this value has been identified as 22.5 degrees which is shown in the Figure 3.5. Hence, for any given sample space with side of length ‘C’, the value of D is calculated using the formula,

\[ D = \frac{C}{2 \sin (22.5^\circ)} \quad (23) \]

The Figure 3.6 shows the probable values for the BS Transmission distance for a given length of the side of the sample space. When deciding on the specifications of the Base station for a given sample space, a reference to this graph shown below will help in deciding on the transmission power to ensure that the sectorization algorithm works fine. If the sample space is large, then a very high BS Tx power could be of a concern. However, if we are deploying these applications in a military scenario, then this is definitely feasible. Also, the process of mapping the sample space is done only once after the sensor network is deployed assuming that the sensor nodes do not change the position.
3.4 MOBILITY VERIFICATION OF A CO-OPERATING MOBILE NODE (MN) IN A WIRELESS SENSOR NETWORK (WSN)

In this section, we will discuss the design of the mobility verification model which verifies the correctness of a claimed path given by a co-operating mobile node. We assume that we do not have any position information regarding the deployment of the WSN. The mobility verification model is divided into two phases: a) Setup phase, b) Verification Phase. These phases are explained in detail below.
3.4.1. Setup Phase. In this phase, we intend to finalize the neighbor selection for the sensor nodes and develop a power-control based method for dividing the sample space into $G$ sectors for better performance.

1) Neighbor selection: There are many instances of neighbor selection algorithms in the literature and we intend to use one such algorithm for our model. Once the WSN is setup, all the nodes use this algorithm to get a list of their neighbors which will be used later in the model for flooding control information to the Base Station.

2) ‘Sectorization’: In our mobile verification technique, we propose to use two Base Stations as a part of the Network setup to aid us in sectorizing the sample space which is monitored. For instance, Figure 3.7 presents the location of the base stations with respect to the sample space.

Figure 3.7 : Network setup with two Base stations
We assume that BS I and II will be able to locate themselves at a distance $d_1$ and $d_2$ respectively away from the sides of the sample space. In addition, the Base stations have enough power to reach all the nodes inside the space. The base stations also synchronize the actions between themselves. The algorithm described below is independently executed by both the base stations.

Described below in Table 3.1 are the terms that are used in the algorithm.

Table 3.1: Terms and Explanations

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Power_level$_i$</td>
<td>The value of the power level at a given instance for the $BS_i$. This value is increased in uniform steps till the $BS_i$ gets the id’s of all the sensor nodes in the sample space.</td>
</tr>
<tr>
<td>start_transmit_power_level</td>
<td>Initial power level with which the BS begins the transmission.</td>
</tr>
<tr>
<td>transmit_distance</td>
<td>The maximum distance that the $BS_i$ with Power_level$_i$</td>
</tr>
<tr>
<td>$d_i$</td>
<td>The perpendicular distance between the $BS_i$ and a side of the sample space</td>
</tr>
<tr>
<td>$w_i$</td>
<td>The width of the sample space with respect to a $BS_i$</td>
</tr>
<tr>
<td>respond_with_id</td>
<td>A message with specific code which instructs the sensor nodes in the sample space to send their id’s to the requesting $BS_i$.</td>
</tr>
<tr>
<td>sector$_i$</td>
<td>A matrix that holds the id’s of the sensors that can be reached with increasing order of power levels.</td>
</tr>
<tr>
<td>$r_{mean}$</td>
<td>Mean sensing range of the nodes in the sample space</td>
</tr>
</tbody>
</table>
The BS’s negotiate between themselves and finalize which BS should start the Sectorization algorithm. Once, a BS is selected for executing the algorithm, it begins by calculating the initial transmit_distance which is \( d_i + r_{\text{mean}} \). Then, it sets the power level for its transmissions based on the transmit_distance. Further, the base station sends out a respond_with_id request and it receives the id’s of all the nodes that have heard this message. The BS stores all such id’s. It increments the transmit_distance once again by \( r_{\text{mean}} \). The BS then recalculates the transmit power level and sends the respond_with_id request. The respond_with_id request gets the id’s of all sensor nodes that hear message the now and then the BS stores that information. The BS keeps repeating this process till it completely covers the width \((w_i)\) of the sample space with respect to its own location. The Sectorization algorithm is executed for all the base stations included in the network model before the verification begins.

After the execution of the algorithms, the base stations exchange their sector matrices with each other. If sector_1 has \( M \) rows and sector_2 has \( N \) rows, then there are \( M \times N \) grids in the sample space. The values in these two matrices could be compared to get the nodes that will lie in each of these \( M \times N \) grids. Let’s call this new matrix which contains the list of nodes that lies in each of these grids as the Grid Matrix.
for each $BS_i$ in the network model do

    negotiate and select a $BS_i$ that will execute the algorithm

    calculate $transmit\_distance$

    set $Power\_level_i$ based on $transmit\_distance$

    set $j = 0$

    while $transmit\_distance$ less than $d_i + w_i$ do

        transmit a $respond\_with\_id$ message with $Power\_level_i$

        for each received message do

            while $sensor\_id \in sector_{i}[j-1]$

                add $sensor\_id$ to $sector_{i}[j]$

            end while

        end for

        incr $j$

        incr $transmit\_distance$ by $r_{mean}$

        reset $Power\_level_i$ based on new $transmit\_distance$

    end while

end for
3.4.2. Verification Phase.

Mobility verification Algorithm Step I

1) Query Actual profile

Once the mobility verification algorithm is initiated, the BS decides to send out a `respond_with_id_heard_nons` request to all the nodes in the sample space. The sensor nodes that have heard the `nons` respond back with their unique id and the exact value of the `nons` that they heard. This information is stored by the base stations in the `actual_nons_nodes` matrix.

2) Claimed profile’s sector-matrix formation

In this section, the base station estimates the sectors through which the Mobile Node should have travelled. Since the base station knows the co-ordinates of the sample space and also the information on how the grids are formed, it plots the ‘claimed’ [x y] profile in space and creates a `claimed-sector-matrix` which contains all the sectors through which the Mobile node should have travelled. In addition, the BS compares the `Grid` matrix with the `claimed-sector-matrix` to list down a probable set of nodes (`claimed_nons_probable_nodes` matrix) that could have heard the `nons` broadcast from the Mobile Node.

Mobility verification Algorithm Step II

3.4.2.1 Mean number of intersections. In this verification step, we will use the mean number of intersections property to perform a first-level check on the path claimed by the mobile node. This verification method involves the following steps

- Calculation of the mean and variance
We intend to use the mean number of intersections property from [28] to perform a first-level check on the claimed path. Hence, we calculate the mean and variance for the number of intersections for varying length in the sample space. The mean can be derived by integrating the theorem derived in 4.3 for the varying lengths of L in the sample space. The variance can be given by the formula

\[ \sigma^2 = \frac{N}{N-1} s^2 \]  

(24)

Where

\[ s^2 = \frac{1}{N} \sum_{i=1}^{N} (x_i - \bar{x})^2 \]  

(25)

The Standard deviation could be calculated as well.

- **Estimation of the actual and claimed paths**

Once the node shares the claimed profile with the Base Station, we can use the profile points to calculate the length of the claimed path. Consequently, to calculate the length of the actual path, we make an assumption that the entry and exit points of the mobile node through the sample space happen through a sensor node. This helps us estimate the length of the actual path in a better manner. In spite of this assumption, it is not straightforward to calculate the length of the actual path since we do not have the exact position information of the nodes that have received the broadcasts. Hence, we will use the sectorization information estimated during the setup phase along with the ‘actual’ profile information to calculate the ‘actual’ length travelled by the mobile node.
Using the information about the sensors that have received the sensor information setup information, we estimate the grids through which the node has actually travelled. With this information the distance travelled at the center of these grids is calculated which gives us an approximate estimate of the actual length travelled.

- **Verification method**

A standard delta L value for the sample space is calculated using the Theorem 1

$$\Delta L = \frac{1}{4} \lambda \cdot r$$

Then we should estimate the length travelled for the actual (L1) and claimed path (L2). If (L1-L2) is greater than Delta L, then we should use the SD property. Else, directly go to Sectorization because, the mean no. of intersections will change at least by one if and only if (L1-L2) is greater than Delta L.

After estimating the actual and claimed lengths, the number of intersections is calculated for these lengths are calculated using theorem 4.3. Let the mean number of intersections calculated using the claimed and actual length be $\text{claimed\_int}$ and $\text{actual\_int}$ respectively. The mean and variance of the number of intersections in a given sample space are used to estimate the accuracy of the claimed profile.

If the following condition is satisfied, then we might have to move onto the Sectorization algorithm, else the claimed path is wrong.

$$\text{Claimed\_int} - \sigma \leq \text{actual\_int} \leq \text{claimed\_int} + \sigma$$

### 3.4.2.2 Sector-based verification algorithm (SEBA)

The verification phase is triggered when a CP Mobile Node completes a movement inside the sample space and shares the ‘claimed’ profile with either of the Base Stations. The base stations share this information between each other. And the mobility verification algorithm is triggered.
1) *Compare number of nons messages (Check 1)*

The first check that we do is to compare the unique id of the *nons* received by the sensor nodes with the total number of broadcasts made by the mobile node. The total number of broadcasts is the length of the ‘claimed’ profile matrix shared by the mobile node.

*Claimed path is wrong:* If the maximum value of the unique id from the *nons* received is greater than the total number of broadcasts, then the claimed path is definitely wrong. It goes to show that the mobile node has stayed longer than its claim inside the sample space.

However, if the maximum value of the unique id from the *nons* received is lesser than the total number of broadcasts, the case is indecisive. Hence, the next section of the algorithm is executed to continue the verification process.

2) *Checking out-of-sector message delivery*

In this section, we will compare every row of the *claimed_nons_probable_nodes* matrix with the *actual_nons_nodes* matrix to see if there are any ids of nodes that are present in *actual_nons_nodes* but missing in *claimed_nons_probable_nodes*. If yes, then it means that the mobile node has taken a different path and a node from a different sector has heard the *nons* indicating that the claim is wrong.

3) *Verify the order-of-nons reception*

In addition to the comparison that’s done in 4), the order-of-nons reception is also checked to ensure that the entry and exit points are the same as well
3.5 SIMULATIONS

3.5.1. Simulation Framework. A robust simulation model is necessary for (1) The simulation of the wireless sensor network deployment, (2) Path planning for the actual and claimed paths of the mobile node and (3) Implementation of the complete mobility verification model. Therefore, a wireless sensor network framework is developed in MATLAB which can simulate the random deployment of any number of sensors in the network. Figure 3.8 shows the output of one such simulation run. The simulation begins with a random deployment of the wireless sensor nodes in the network, followed by selecting a specific area as the sample space under surveillance. The framework is also capable of randomly planning the paths for both the claimed and the actual profile. The mobility verification model is built on top of this MATLAB based simulator framework.

![Simulation of a Uniform Random Deployment of a WSN](image)

Figure 3.8 : Simulation of a Uniform Random Deployment of a WSN
3.5.2. **Simulation Parameters.** To analyze the performance of the proposed verification model, we have made the following assumptions.

- During any given movement from an entry point to the exit point in the sample space, the mobile node does not tread back in the path that it has already taken.
- The Mobile Node keeps broadcasting nons messages with a time interval $T$.
- The wireless sensor nodes will have a uniform random deployment within the sample space.
- Every sensor node in the WSN will have a unique id associated with it.
- After completing its movement inside the sample space, the Mobile node shares its ‘Claimed’ path with the Base station. The ‘Claimed’ path consists of a set of $[x, y]$ co-ordinates in which the Mobile Node claims to have travelled through within the sample space.
- For generality, if the sensing range of the ‘n’ sensor nodes is given as $r_1, r_2... r_n$ : it is taken that the mean sensing range be $r_{\text{mean}}$
- The Base station has the $[x \ y]$ coordinates profile for the sample space.

Figure 3.9 illustrates the probable paths that the Mobile Node can take and Table 3.3 shows the parameters and the values used for the simulations.
Table 3.3: Parameters and Values for simulation

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value (units)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Length of the Simulation area</td>
<td>1000</td>
</tr>
<tr>
<td>Breadth of the simulation area</td>
<td>1000</td>
</tr>
<tr>
<td>Number of sensors deployed</td>
<td>100</td>
</tr>
<tr>
<td>Reception range of sensors</td>
<td>50</td>
</tr>
<tr>
<td>Grid distance for sectorization algorithm</td>
<td>100</td>
</tr>
<tr>
<td>Broadcast interval</td>
<td>20</td>
</tr>
<tr>
<td>Sample space selection</td>
<td>Random</td>
</tr>
</tbody>
</table>

Once the simulation framework is ready, the actual and claimed profile are generated based on the offset distance provided initially. Once the execution of the
verification model is complete it gives a result code which indicates whether the claimed path is correct or not. In addition, it returns a specific error codes for all possible failures. **Table 3.4** shows the reason and the result codes that are returned.

### Table 3.4: Reason and result codes

<table>
<thead>
<tr>
<th>Case Number</th>
<th>Result Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Case 1</td>
<td>0</td>
<td>The Claimed path is deemed right</td>
</tr>
<tr>
<td>Case 2</td>
<td>-1</td>
<td>More messages are received than the claimed path</td>
</tr>
<tr>
<td>Case 3</td>
<td>-2</td>
<td>The sectorization algorithm detects an anomaly in the path</td>
</tr>
<tr>
<td>Case 4</td>
<td>-3</td>
<td>Number of intersections property failed</td>
</tr>
<tr>
<td>Case 5</td>
<td>-4</td>
<td>The order of message broadcast was wrong</td>
</tr>
</tbody>
</table>

### 3.6 RESULTS

The results for the proposed mobility verification model are presented below. This section is broadly divided into two categories (1) Evaluation of the mobility verification model against basic test cases (2) Analyzing the performance of the algorithm and evaluating the parameter selection.
3.6.1. Basic test cases. In this section, we evaluate four basic test cases to evaluate the correctness of the mobility verification model. The parameters varied in this section are (1) The length of the actual and the claimed path (2) The offset distance between the claimed and actual path (3) The direction in which the claimed and actual path navigate inside the sample space.

3.6.1.1 Lengths: \( L_1 = L_2 \), Offset distance = 0, Direction: Same. The aim of this experiment is to test the functioning of the mobility verification algorithm and also to spot any potential error case scenario. In this simulation, the claimed path and the actual path are set to be the same. And the experiment was repeated for all possible arrangements of the actual and the claimed path and the algorithm successfully cleared the claimed path with the result code ‘0’ indicating that the claimed path was correct. In addition, we carried out simulations varying the arrangements of the nodes inside the sample space. The results once again were positive, indicating that the algorithm is independent of the deployment knowledge of the sensors. Out of all the simulations, we have shown the horizontal and a slanting mobility pattern in Figure 3.10 and Figure 3.11

Another important aspect of this simulation run was the direction of travel between the claimed and actual path. For this setup, they were set to be in the same direction. However we have performed additional simulations in the following sections changing the direction of mobility between the claimed and actual path.
Figure 3.10: Horizontal mobility pattern

Figure 3.11: Slanting mobility pattern
3.6.1.2 Lengths: L1 ≠ L2, Offset distance > 0, Direction: Same. In this experiment, an offset distance of 30 units was introduced between the claimed path and the actual path when the mobility pattern is parallel to the diagonal of the sample space. Hence, the lengths of the claimed and actual path inside the sample space would not be equal and the directions in which they travel were the same.

The mobility verification model rejected the claimed path shown in Figure 3.12 returned the code ‘-1’ indicating that the actual number of messages heard were more than the claimed broadcasts indicating that the mobile node was present in the sample space longer than the claimed time. Multiple simulation runs were performed to ensure the correctness of the result.

![Figure 3.12: Mobility Verification model - Claimed Path rejection – I](image)
3.6.1.3 Lengths: L1 = L2, Offset distance > 0, Direction: Same. This experiment is very similar to the previous, except that the lengths of the two paths were retained to be the same. Therefore, the mobility pattern was set to be parallel to one of the sides of the sample space. The intention is to check if the algorithm is able to detect the claimed path offset though the number of broadcasts is similar. Hence, the lengths of the claimed and actual path inside the sample space would not be equal and the directions in which they travel were also the same.

The mobility verification model rejected the claimed path shown in Figure 3.13, and the model returned the code ‘0’ indicating that the claimed path was correct. This is because, the chosen offset distance (30) was lesser than ‘Delta L’ (50). Hence, the though there is an offset, the paths might still travel through the same grids because of which the claimed path was deemed correct.

The simulation shows that value of ‘Delta L’ does play a critical role as explained previously. Therefore, it is imperative that we perform a study on the performance of the algorithm when the offset value is changed beyond ‘Delta L’. In such a uniform deployment model, the choice of \( r_{\text{mean}} \) and ‘Delta L’ would impact the performance of the mobility verification algorithm.
However, repeating the same experiment again revealed a result code ‘-2’ indicating that the claimed profile had some broadcasts other than the ones that actually heard the message. Upon repeating the experiment multiple times, the results were mixed because of the above mentioned reason.

3.6.1.4 Lengths: $L_1 \neq L_2$, Offset distance > 0, Direction: Opposite. In sections 4.1.4 and 5, we examine two special mobility cases where in the direction of the claimed path is exactly opposite to that of the actual path. The intention here is to check if the mobility verification model is able to detect the change in direction though the nodes or grids though it might pass through be the same.
Figure 3.14 shows the claimed mobility path is offset from the actual path by a certain distance. And the verification model deemed the claimed path was wrong with a result code ‘-2’ indicating that an actual non-message receive node did not exist in the probable list of claimed nodes. And another interesting aspect was that the result was achieved only when the third message comparison was made. This is because the other grids did not have nodes that could solve this case emphasizing the fact that with more
broadcast messages, we have a higher chance of detecting any anomaly in the claimed path.

3.6.1.5 Lengths: $L_1 \neq L_2$, Offset distance = 0, Direction: Opposite. This experiment is exactly similar to the last case, except that the offset distance between the two paths is set to zero. The mobility verification model was once again able to determine the change in direction and reported that claimed path is wrong all the simulation runs. It returned with the code ‘-2’ indicating that an actual receive node did not exist in the probable list of nodes. This result was achieved even as the first set of entrance grid sensors was compared and that's because nodes heard the first broadcast.

Figure 3.15 : Mobility Verification Model - Opposite directions
3.6.2. **Performance evaluation of the mobility verification model.** In this section, we intend to test the performance of the algorithm by varying three important parameters in our approach. They are (1) The reception range of the sensors (2) The offset distance between the claimed and actual path (3) The grid distance chosen for the setup phase of the sectorization algorithm. Table 3.4 contains the legend of all the codes that’s returned by the algorithm and the reason for the failure.

3.6.2.1 **Offset value between the claimed and actual path.** In this section, we test the performance of the algorithm against the change in offset distance between the claimed and the actual path. The offset value is a significant factor in determining the performance of the verification model. From Table 3.5, we can see that the accuracy of the algorithm remained stable almost throughout even for small changes in the offset value. The successful performance can be attributed to the following reasons:

1. A relatively large number of nodes received the mobile node’s actual message broadcasts. It is quite obvious that lesser the dropped messages, higher the accuracy in detecting any anomaly in the claim.

2. The offset distance will induce a change in the length of the mobility path inside the sample space. If the mobile node’s actual path is parallel to the diagonal of the sample space, then a change in offset value will mostly introduce a change in the actual path. Therefore, we would have heard nons messages greater than the mobile node claims.

3. If any of the actual nons message had been broadcasted in the intersection of two grids. Then, the performance of the algorithm increased further. Since, the algorithm expects the claimed profile point to lie within the overlapping region between the sensors.
To check the accuracy of the results, the results were noted for 10 simulation runs with different sensor distribution patterns. The accuracy for these simulations was between 80% – 100%. This shows that the algorithm is stable.

Table 3.5: Verification model results and codes when varying the offset value

<table>
<thead>
<tr>
<th>Code</th>
<th>Offset Value</th>
<th>Claimed Path</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>10</td>
<td>Right</td>
</tr>
<tr>
<td>-1</td>
<td>20</td>
<td>Wrong</td>
</tr>
<tr>
<td>-1</td>
<td>30</td>
<td>Wrong</td>
</tr>
<tr>
<td>-1</td>
<td>40</td>
<td>Wrong</td>
</tr>
<tr>
<td>-2</td>
<td>50</td>
<td>Wrong</td>
</tr>
<tr>
<td>-2</td>
<td>60</td>
<td>Wrong</td>
</tr>
<tr>
<td>-2</td>
<td>70</td>
<td>Wrong</td>
</tr>
<tr>
<td>-2</td>
<td>80</td>
<td>Wrong</td>
</tr>
<tr>
<td>-2</td>
<td>90</td>
<td>Wrong</td>
</tr>
<tr>
<td>-2</td>
<td>100</td>
<td>Wrong</td>
</tr>
</tbody>
</table>

3.6.2.2 Reception range of the sensors. In this section, the reception range of the sensors is varied to evaluate the performance of the algorithm. The choice of reception range for the wireless sensor nodes is important because a higher reception range might severely reduce the life-time of the network. On the other hand, a lower reception range, might reduce the effectiveness of the mobility verification model. Hence, by varying the reception range of the wireless sensor nodes, we intend to effectively tradeoff and choose
an optimum value for the reception range. Table 3.6 shows the results of the mobility verification model when the reception range ($r_{\text{mean}}$) is increased to 100. The mobility verification model gives approximately 100% accuracy when $r_{\text{mean}}$ is increased because of the following reasons

The total number of overlapping sensors has increased. This increases the change that the actual nons messages are heard in these overlapping locations, thereby increasing the performance.

The total coverage of the sample space has increased. Therefore, most of the nons messages are heard. As we mentioned already, the accuracy of the algorithm increases with an increase in the number of nons message received.

Table 3.6: Verification model results and codes when varying the offset value

<table>
<thead>
<tr>
<th>Return Code</th>
<th>Offset Value</th>
<th>Claimed Path</th>
</tr>
</thead>
<tbody>
<tr>
<td>-2</td>
<td>10</td>
<td>Wrong</td>
</tr>
<tr>
<td>-1</td>
<td>20</td>
<td>Wrong</td>
</tr>
<tr>
<td>-2</td>
<td>30</td>
<td>Wrong</td>
</tr>
<tr>
<td>-1</td>
<td>40</td>
<td>Wrong</td>
</tr>
<tr>
<td>-1</td>
<td>50</td>
<td>Wrong</td>
</tr>
<tr>
<td>-1</td>
<td>60</td>
<td>Wrong</td>
</tr>
<tr>
<td>-2</td>
<td>70</td>
<td>Wrong</td>
</tr>
<tr>
<td>-1</td>
<td>80</td>
<td>Wrong</td>
</tr>
<tr>
<td>-2</td>
<td>90</td>
<td>Wrong</td>
</tr>
<tr>
<td>-1</td>
<td>100</td>
<td>Wrong</td>
</tr>
</tbody>
</table>
To check the accuracy of the results, the results were noted for 10 simulation runs with different sensor distribution patterns. The accuracy for these simulations was between 90% - 100%. This shows that the algorithm is consistent.

It is justified from the above simulations that increasing the reception range of the sensors has increased the accuracy of the algorithm.

3.6.2.3 Grid distance in the sectorization algorithm. In this experiment, the grid distance chosen for the setup phase of the sectorization algorithm is varied. We intend to identify the impact of grid distance on the performance of the algorithm. Two specific setups are identified for the simulation and they are discussed below.

The actual and claimed paths were chosen parallel to the sides of the sample space. In other words, the lengths of the paths remained the same through the simulation even as the offset values were varied.

This simulation setup was repeated 10 times to verify the accuracy of the results with different sensor distribution patterns. The results for step (1) varied from 30% - 70% indicating that the results were heavily dependent on the distribution of sensors in the sample space. The algorithm hardly detects an anomaly only when an actual broadcast happens right at the intersection of two grids. In practical applications, the deployment of the sensors is highly random and hence our choice for the grid with 2rmean * 2rmean as the area proves to be the best solution.
Table 3.7: Simulation (1) results when varying the grid distance

<table>
<thead>
<tr>
<th>Return Code</th>
<th>Offset Value</th>
<th>Claimed Path</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>10</td>
<td>Right</td>
</tr>
<tr>
<td>0</td>
<td>20</td>
<td>Right</td>
</tr>
<tr>
<td>0</td>
<td>30</td>
<td>Right</td>
</tr>
<tr>
<td>0</td>
<td>40</td>
<td>Right</td>
</tr>
<tr>
<td>-2</td>
<td>50</td>
<td>Wrong</td>
</tr>
<tr>
<td>0</td>
<td>60</td>
<td>Right</td>
</tr>
<tr>
<td>-2</td>
<td>70</td>
<td>Wrong</td>
</tr>
<tr>
<td>-2</td>
<td>80</td>
<td>Wrong</td>
</tr>
<tr>
<td>0</td>
<td>90</td>
<td>Right</td>
</tr>
<tr>
<td>0</td>
<td>100</td>
<td>Right</td>
</tr>
</tbody>
</table>

The actual and claimed paths were chosen parallel to the diagonal of the sample space. Hence, the lengths of the actual and claimed paths might change as we vary the offset distance between them.

This simulation setup described in (2) was repeated 10 times to verify the accuracy of the results with different sensor distribution patterns. The results varied from 50% - 70% indicating that the results were heavily dependent on the distribution of sensors in the sample space like in the previous case. Also, since the lengths of the mobility path vary, the algorithm identifies ‘Type 2’ failure for some.
Table 3.8: Simulation (2) results when varying the *grid distance*

<table>
<thead>
<tr>
<th>Code</th>
<th>Offset Value</th>
<th>Claimed Path</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>20</td>
<td>Right</td>
</tr>
<tr>
<td>0</td>
<td>40</td>
<td>Right</td>
</tr>
<tr>
<td>0</td>
<td>60</td>
<td>Right</td>
</tr>
<tr>
<td>-1</td>
<td>80</td>
<td>Wrong</td>
</tr>
<tr>
<td>-1</td>
<td>100</td>
<td>Wrong</td>
</tr>
<tr>
<td>-1</td>
<td>120</td>
<td>Wrong</td>
</tr>
<tr>
<td>-2</td>
<td>140</td>
<td>Wrong</td>
</tr>
<tr>
<td>-1</td>
<td>160</td>
<td>Wrong</td>
</tr>
<tr>
<td>-1</td>
<td>180</td>
<td>Wrong</td>
</tr>
<tr>
<td>-1</td>
<td>200</td>
<td>Wrong</td>
</tr>
</tbody>
</table>

To explain the results better, we would like to discuss the problems that a best case deployment scenario would solve. We do realize that a best case deployment scenario is not achievable practically but is explained here only to elaborate on the limiting factors affecting the performance. Figure 3.16 shows the best case scenario it’s quite clear that the selection of $2*r_{\text{mean}}$ as the side of the grid is justified. Because, a grid with $2*r_{\text{mean}}$ as its sides would completely cover a sensor.
In addition, the reception range of a covered sensor overlaps with the other sensors, which ensures that there is no vacant region in the grid. Therefore, the two factors that would ensure a perfect mobility verification model are (1) The sample space should be completely covered (2) Any given point in the sample space should be overlapped by at least two sensors. From the above explanation, we can say that the closer the sample space deployment is to the best case scenario, the better our verification model results would be.

Figure 3.16: Best case deployment scenario
4. CONCLUSION AND FUTURE WORK

In this section, we have introduced a relatively new problem in the area of wireless sensor networks - 'Mobility verification’. In addition, the differences between mobility verification and traditional source tracking problems are listed. To solve the problem, we propose a power-control based Mobility verification model which uses (1) statistical bounds and (2) Sector based verification algorithm. The power-control based sectorization approach significantly reduces the algorithmic intensity and power usage. The performance of the proposed scheme is verified by implementing a simulation framework in Matlab. The results show that the algorithm is effective in handling the different mobility patterns with varying offset values. The performance of the algorithm was further analyzed by varying the following parameters - (1) Grid distance and (2) Reception range of the sensors. And the results demonstrate that the model's ability to perform in most scenarios. Therefore, this verification model can be successfully used in mission-critical applications.

As a part of the future work, we need to address the establishment of secure communication channels and study its defense mechanisms against existing attacks. Because, the models discussed here do not discuss the security aspect of information exchange. Also, in real-world, the wireless environment will have a number of obstacles such as channel error and dropped packets. Hence, we need to develop error models and embed it with the mobility verification scheme.
BIBLIOGRAPHY


S. Jagannathan, “Quality of Service Control in Ad hoc wireless and Sensor Networks,” Taylor & Francis, USA.

S. William, “High-Speed Networks and Internets, performance and Quality of service,” Prentice Hall, USA.


38 Ekici E, McNair J, Al-Abri D, “A probabilistic approach to location verification in wireless sensor networks,” ACM workshop on security of ad hoc and sensor networks (SASN05), November 2005
VITA

Maheswaran Thiagarajan was born in Palani, India on March 18, 1984. In May 2005, he received his Bachelor of Engineering [B.E. (Hons.)] degree from the Birla Institute of Technology & Science (BITS), Pilani, India. In that same year, he was employed as an engineer with Honeywell Technology Solutions Lab, Bangalore, India. In May 2009, he received his M.S. in Systems Engineering from the Missouri University of Science & Technology (Missouri S&T), Rolla, Missouri, USA.