The Pennsylvania State University

The Graduate School

Department of Electrical Engineering

METRIC DRIVEN MOBILITY MODELING IN TACTICAL NETWORKS

A Thesis in

Electrical Engineering

by

Sucharita Ray

Submitted in Partial Fulfillment
of the Requirements
for the Degree of

Master of Science

May 2011

The thesis of Sucharita Ray was reviewed and approved* by the following:

Thomas La Porta
Distinguished Professor of Computer Science and Engineering
Director of Networking and Security Research Center
Thesis Advisor

John J. Metzner
Professor of Electrical Engineering

Kenneth Jenkins
Professor of Electrical Engineering
Head of the Department of Electrical Engineering

*Signatures are on file in the Graduate School

ABSTRACT

Mobility management is a key aspect of designing and evaluating protocols for Mobile Ad Hoc Networks (MANETs). The high mobility of nodes in a MANET constantly causes the network topology to change. Mobility patterns of nodes have a direct effect on network parameters like path length, neighborhood size, link stability and so on. Consequently the network performance is strongly affected by the nature of mobility pattern. Hence, while evaluating protocols for a specific MANET application, it becomes imperative to use a mobility model that is able to capture the movement of nodes in a considerably accurate manner.

The objective of this work is to produce mobility models that are able to describe tactical mobility in military applications of MANETs. We begin with a quick survey of the common mobility models that have been used in the past and go on to show that these models prove to be insufficient when modeling tactical mobility in military networks. We then describe the design of four novel models that capture common military scenarios. In doing so we present a trace based analysis, a motion model as well as a mathematical model. The errors between these three approaches have been calculated for specific parameters. In the last part of this work we evaluate different data replication algorithms for the four mobility scenarios and suggest the algorithm that performs best in each case.

TABLE OF CONTENTS

LIST OF FIGURES

ACKNOWLEDGEMENTS

I would like to thank my research advisor, Dr. Thomas La Porta for his invaluable guidance and support throughout the course of my research. I will always be indebted to him for giving me the opportunity to join his research team at the Networking and Security Research Center and making me a part of the prestigious Networking Science Collaborative Technology Alliance (NS CTA). He has always been a source of inspiration and encouragement to me. His inputs, ideas and challenging discussions were very helpful towards the completion of this work.

I would also like to thank the members of my thesis committee, Dr. John J. Metzner and Dr. Kenneth Jenkins for their efforts in reviewing and evaluating my research.

I thank Prithwish Basu and Alberto Medina for providing me with the simulation tool that was necessary for my research. The completion of this work would not have been possible without their constant support and valuable guidance.

I would like to acknowledge my fellow department mates Aayush Gupta and Hayawardh for their timely help and assistance during the course of my graduate studies. Finally, I thank my dear parents, family and friends for their support and encouragement.

This research was sponsored by the Army Research Laboratory and was accomplished under Cooperative Agreement Number W911NF-09-2-0053. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the Army Research Laboratory or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation here on.

Chapter 1

Introduction

Modern day warfare has evolved into an amalgamation of complex military tactics and highly advanced information technology. In addition to developing powerful strategies, the United States Department of Defense has pioneered a new military doctrine called Network-centric Warfare, discussed exhaustively in [1]. This novel theory aims at forming robustly networked forces on the battlefield, with enhanced information sharing, higher collaboration and increased speed of command, resulting in a dramatic increase in mission effectiveness. Enabling Network-centric Warfare calls for the deployment of a complex communication network, comprising of military equipment, personnel and communication protocols to relay information among forces. Moreover, a means of accurately analyzing the network must be devised so that the best performing communication protocols and algorithms can be implemented.

Communication in military scenarios is often carried out using tactical mobile ad hoc networks (MANETs), which are a collection of wireless mobile nodes forming self-configuring networks without using any existing infrastructure. A key characteristic of these networks is high mobility of nodes, which results in a dynamic and fast changing network topology. Hence, while evaluating protocol performance in MANETs, it is essential to be able to capture the mobility of nodes accurately as well as to quantify the effect of mobility on various network parameters. An effective means of doing this is through the development of mobility models.

Mobility models are designed to describe the movement pattern of mobile users and capture how their location, velocity and acceleration change over time. Since mobility patterns play a significant role in determining protocol performance in a MANET, it is desirable for mobility models to emulate the movement pattern of the targeted real life scenario to a reasonable

extent. Otherwise, the observations made and the conclusions drawn from the simulation studies may be misleading. Thus, when evaluating MANET protocols, it is necessary to choose the most appropriate underlying mobility model.

In this work we aim at developing tactical mobility models that are accurately able to capture the movement pattern of nodes in military scenarios. Several research efforts have been made in the past towards modeling mobility in MANETs, but in this work we find conventional mobility models to substantially deviate from the real life movement patterns of military scenarios. Hence, we develop tactical mobility models with the motivation of using them to effectively evaluate the performance of communication protocols for military applications, so that more informed decisions can be made towards Network-centric Warfare. The following is an outline of this thesis.

In the second chapter, we begin by summarizing some of the related research work. We present an overview of the well known mobility models that have been developed and used in the past for evaluating various MANET applications. We then take a look at some common mobility modeling techniques and finally discuss a specific approach in detail, which has been adopted in this work for modeling military mobility.

The third chapter presents four tactical scenarios that are modeled in this thesis, each representing a well known military tactic. The mobility in each scenario is modeled using several approaches, including scenario simulations for generating synthetic mobility traces, design of a motion model using fundamental laws of physics and finally mathematical modeling using regression analysis. Further, for each scenario we discuss several key properties of the movement pattern, which significantly impact protocol evaluation for that tactic.

In the fourth chapter we evaluate the errors between the three approaches taken to model tactical mobility, namely the trace based analysis, the motion model and the mathematical model. For the purpose of this work, we assume the mobility traces to represent the ground truth. In this

chapter we show that the motion model and the mathematical model are able to produce results which are within a bounded error margin of the results generated by the traces. This demonstrates the usability of the motion and mathematical models, which can be used for protocol evaluation when synthetic traces of a military scenario are not available.

In the fifth chapter, we describe how the tactical mobility models hence developed can be used for protocol evaluation, as well as how the decisions of performance evaluation change if conventional models are used in place of tactical models. To illustrate this we take the example of four data replication algorithms, and evaluate their performance with conventional mobility models as well as with tactical models. We find a significant deviation between the two results, strongly indicating that conventional mobility models fail to capture the structured nature of tactical scenarios. Moreover, they are misleading and result in wrong conclusions being drawn when applied to protocol evaluation for military scenarios.

Chapter 2

Related Work

The need to characterize the mobility of nodes in wireless ad hoc networks has drawn many researchers to the topic of mobility modeling. In this chapter we take a look at some well known mobility models, describe important mobility modeling techniques and closely study the specific tool that has been used to model tactical mobility in this thesis.

2.1 Mobility Models – An Overview

Figure 2-1 below is a schematic view showing the categorization of mobility models based on specific properties that they show.

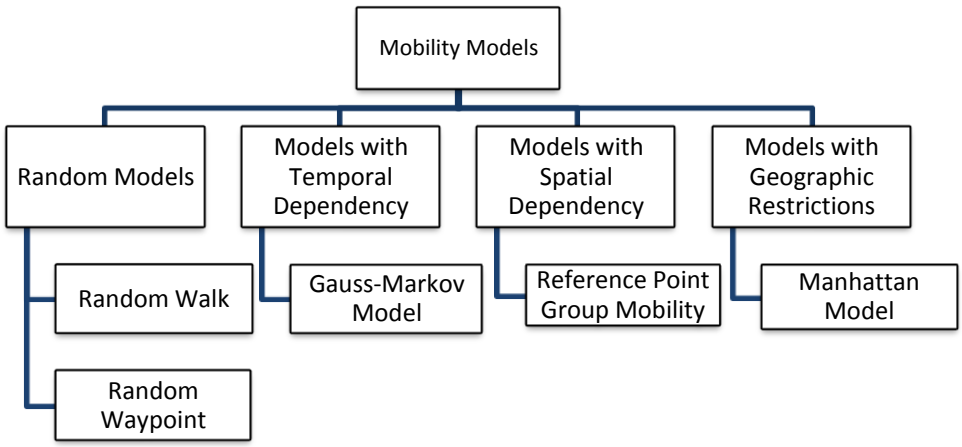


Figure 2-1. Categorization of mobility models

The following subsections summarize the characteristics of each category and discuss some well known mobility models along with their merits and demerits.

2.1.1 Random Models

In random-based mobility models, mobile nodes move freely without restrictions. Their destination, speed and direction are chosen randomly and independent of other nodes. This kind of model has been used in many simulation studies. The following section describes two models showing random patterns in mobility.

A. Random Walk Model

The Random Walk model, described by Zonoozi and Dassanayake in [2], was originally proposed to emulate the unpredictable movement of particles in physics. It is also referred to as the Brownian Motion. In scenarios where some mobile nodes are believed to move in an unexpected way, the Random Walk mobility model is proposed to mimic their movement behavior. In this model a mobile node goes from its current position to its next position randomly. For every new interval t, the speed and direction are picked randomly from $[V_{max}, V_{min}]$ and $(0,2\pi]$. Therefore, during time interval t, the node moves with the velocity vector $(v(t)\cos\theta(t), v(t)\sin\theta(t))$. The Random Walk model is a memoryless model where the information about the previous status is not used for future decision. The current velocity is independent of its previous velocity. However, this is often not the case with real life applications, as we will see in further sections.

B. Random Waypoint Model

The Random Waypoint Model was first proposed by Johnson and Maltz [3-4]. It has since been widely used as a 'benchmark' mobility model to evaluate the MANET protocols as it is easily available and simple to analyze.

In the Random Waypoint model, each mobile node randomly selects one target in the simulation field as the destination. It then travels towards this destination with constant velocity chosen uniformly and randomly from $[0,V_{max}]$, where the parameter V_{max} is the maximum allowable velocity for every mobile node. The velocity and direction of a node are chosen independent of other nodes. Upon reaching the destination, the node stops for a duration defined by the 'pause time' parameter, T_{pause} . If T_{pause}=0, continuous mobility is observed. After this duration, it again chooses another random destination in the simulation field and moves towards it. This process continues till the end of the duration of the real time application.

Though easy to use and understand, random models over-simplify the mobility pattern of nodes in real MANET applications. They are specially unsuited for use in tactical networks where nodes have predefined target sets, velocity ranges, regular behavior and so on.

2.1.2 Models with Temporal Dependency

The mobility models described in the previous section are memoryless, in that the parameters (such as velocity, pause time and so on) of a node in one epoch are independent of its parameters in other epochs. But in real life applications, a node's mobility is often limited by physical laws of force, velocity, acceleration and hence its mobility parameters in a given epoch may well be correlated to its previous epoch. This is often called 'temporal dependency' of parameters. A prominent model showing such dependency is the Gauss-Markov Mobility Model [5-6].

2.1.3 Models with Spatial Dependency

In models like Random Waypoint and Random Walk, a node's movement, its selection of target destination and velocity are independent of other nodes. This does not accurately capture real life scenarios where nodes move in collaboration with each other. Such applications are common in military missions and vehicular mobility where the mobility of a node is affected by its neighboring nodes. In such cases mobility parameters are said to show 'spatial dependency'. A common mobility model showing spatial dependency is Reference Point Group Mobility (RPGM).

In RPGM, described in [7-8], nodes move in groups consisting of a leader node and other member nodes. The motion vector of the group leader $V_{group}(t)$ is randomly chosen in each epoch, as described in Random Walk. This vector also describes the general motion trend of the group. Each member node deviates from this vector by some degree. A motion vector $RM_i(t)$ is defined, which denotes the deviation of the i^{th} member from the leader. The motion vector of the i^{th} member is then defined by

$$V_i(t) = V_{group}(t) + RM_i(t) \qquad [2\text{-}1]$$

The length of $RM_i(t)$ is uniformly distributed in the interval $[0, r_{max}]$ and its direction is uniformly distributed in $[0, 2\pi)$. The following is a schematic view of the same.

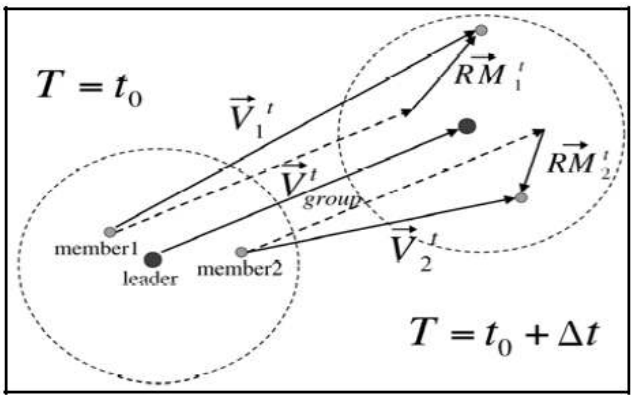


Figure 2-2. Schematic view of Reference Point Group Mobility Model (courtesy [8])

2.1.4 Models with Geographic Restrictions

In the Random Waypoint model, described previously, nodes are allowed to move freely throughout the simulation region. But in real life scenarios we often find that node mobility gets restricted by environmental factors. A common example of this is the vehicle movement that is restricted to the freeways and streets of a region. Such mobility patterns are said to show geographic restrictions. A well known mobility model belonging to this category is the Manhattan Model [9], where the simulation region is divided into a grid like region and nodes are allowed to move only along the edges of the grid (which depict the roads of a city). Each node chooses a destination randomly and moves towards this destination through the shortest path along the edges. Upon arrival it pauses for a period of time and again chooses a new destination.

In comparison to the models described above, military mobility is much more complex and structured. This is so because soldiers in a battlefield always adhere to a predefined mission and code of conduct, which leaves less room for randomization in their movement. Military missions often show group mobility, but using the RPGM model would not suffice since the movement of these groups is not random. Similarly, soldiers in a battlefield are also restricted to specific paths but unlike the Manhattan Model, their choice of target and direction is well defined and not random. In other words, military mobility shows a combination of temporal, spatial and geographic dependencies. It is also prone to dynamic events, which could cause a change in route or strategy of nodes. Hence, there arises a need to study military mobility separately and develop specialised models which capture the nuances of mobility in tactical networks.

2.2 Mobility Modeling Techniques

Among the research efforts made towards modeling mobility in wireless ad hoc networks, two broad approaches are found. On the one hand, performance analysis of protocols in MANETs requires models that can accurately capture the movement patterns of nodes. A natural approach is to establish a one-to-one correspondence between the application scenario and the mobility model. This calls for a new mobility model to be designed for every emerging MANET application. On the other hand, several efforts have been made to develop 'general' models which try to capture certain properties of applications [10-11]. Several of these models have been described in the previous section, 2.1. These models are often over simplified and show considerable departure from the real life mobility patterns of MANET applications.

A more recent approach attempts at encapsulating the common features of all mobility models into a universal mobility modeling framework [12], which can be parameterized to generate application specific models. This removes the need to develop a framework for every new application and reduces the problem to defining a set of parameters that describe the scenario within the universal framework. This approach of modeling has been adopted in this work due to its ease of use, modularity and adaptability. In chapter 3, we show how the universal mobility framework can be parameterized to generate the tactical mobility models developed in this thesis. In the future, this framework can easily be used to generate other instances of tactical mobility. We look into the implementation of the universal mobility modeling framework more closely in section 2.3.

2.3 Universal Mobility Modeling Framework

In [12], Medina et al. develop the Universal Mobility Modeling Framework (UMMF), a tool that enables a general model to be parameterized in order to generate application specific mobility models. The UMMF strives to decouple specific application semantics from mobility features and is able to capture all mobility scenarios in terms of three fundamental factors – Target Selection, Steering Behavior and Locomotion. Nodes in UMMF are modeled as autonomous agents capable of certain amount of intelligence, such as adjusting their paths to avoid obstacles and reacting to dynamic events. The behavior of these autonomous agents is described in three parts:

1. Target Selection – This part of the behavior decides the goal of the agent, which can be a geographic target or a strategic aim.
2. Steering Behavior – The steering behavior of an agent is used to calculate the route that it will take to arrive at its target. This is done by defining a set of steering forces, obstacles and using navigation graphs to compute the best route to the target.
3. Locomotion – This part defines the mechanical behavior of the agent by converting the resultant steering force vector into acceleration, velocity and position vectors at each step of the simulation.

The UMMF tool uses XML files as input, where configuration parameters for the scenario being modeled are defined. Some example parameters are target coordinates, target sets, number of nodes, number of groups and group size, steering behaviors and obstacle coordinates. This input file is parsed by an event driven simulation engine, which is responsible for node placement, topology evolution, calculation of network statistics and so on. The output is viewed in two forms. One is in text format, containing

node coordinates as well as computed statistics of network parameters at every time instant. The second is a graphical output, viewed using a visualization tool, which shows snapshots of how the topology evolves at every step of the simulation and also gives traces for network parameters such as node degree, average path length, neighborhood size and so on.

In the following chapter we describe how the UMMF can be parameterized to simulate tactical mobility and generate synthetic traces, which are assumed to reflect the ground reality for the purpose of this work.

Chapter 3

Tactical Mobility Models

This chapter describes the four new mobility models that have been developed as a part of this thesis. These four scenarios belong to a wide variety of commonly used military tactics that have been studied and documented in [13-15]. Each of these tactics is well defined and has its own unique mobility characteristics.

The modeling of tactical mobility has been approached in three stages. The first stage of modeling involves simulating the scenario using the UMMF tool, described in section 2.3, to generate synthetic traces of several network parameters as a function of time. Specifically, the network parameter of node degree has been studied in detail, as an example. Its dependency on other quantities and their implications has been discussed in depth throughout this chapter. In the second stage a motion model has been developed, that provides a graphical representation of the mobility as well as mathematical equations for mobility parameters of interest. Finally, in the third stage, we perform time series analysis of node degree, using the traces generated by UMMF, and present tractable mathematical equations using the technique of curve fitting. We then quantify the error between the mathematical equations and the traces.

3.1 Repeated Traversal

In the following sub-sections, we illustrate the model for the Repeated Traversal military tactic.

3.1.1 Scenario Description and Traces

The objective of Repeated Traversal, described in detail in [13-14], is to maximize the security of troops moving through an unknown, highly hostile terrain. When troops navigate through an unexamined terrain that is suspected to contain ambushes and traps, a common practice is to traverse through the area along the same path one squad after another. The 'following' squads depart along the same path, only when the 'frontal' squad has arrived at a target of confirmed security and has communicated information about the terrain to the other squads. Frontal squads may also have to rest and wait for the following squads to provide supply and back up. If an 'unfriendly event' is observed by the leading squad (such as a trap or bomb explosion), it communicates the same to the following squad so that the following squad avoids that route.

The movement in Repeated Traversal is very structured and regular. It shows group mobility with an emphasis on inter-group communication. It is important for the leading group to be able to communicate information of the newly traversed terrain to the following group, so that they can plan their movement accordingly. For example if the leading group finds an ambush on its way, it would notify the following group of the same. The following group would then modify their route to ensure maximum safety.

The scenario can be described as an epoch-based model. The simplest case contains two groups in each epoch. It is necessary for these two groups to be able to communicate at the

boundaries of each epoch i.e., once the groups have reached their destinations (for that epoch) they should be able to establish contact. The following figure shows two groups doing Repeated Traversal, from target T_1 to target T_6.

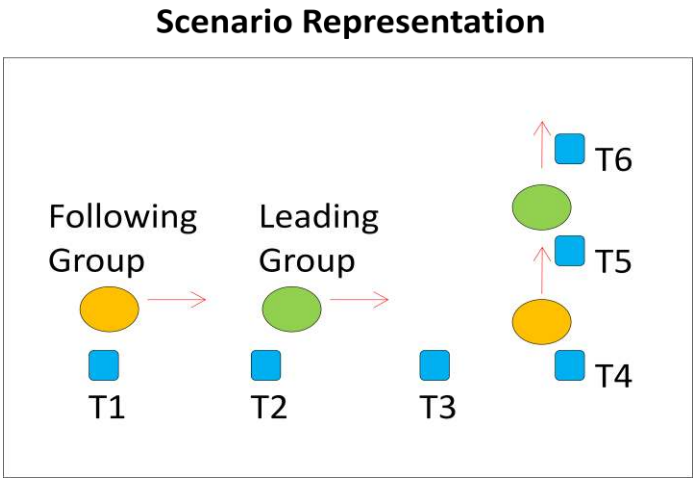


Figure 3-1. Schematic view of Repeated Traversal.

To simulate this scenario in UMMF, a number of mobility features were incorporated in the tool:

1. Nodes move in groups
2. Inter-group communication is necessary
3. Path of 'following' groups is dependent on 'leading group'
4. Static targets positioned at predefined locations
5. Steering Behaviors used: Group leaders – 'Arrive', Group members – 'Pursuit'
6. Occurrence of an unfriendly dynamic event causes that part of the navigation graph to get invalidated.

Traces were generated for several values of input parameters such as velocity, inter-target distance, group size and a large variety of seed values. Figure 3-2 shows two groups of ten nodes

each traversing through a region with four target destinations. In each group the node in red is the leader.

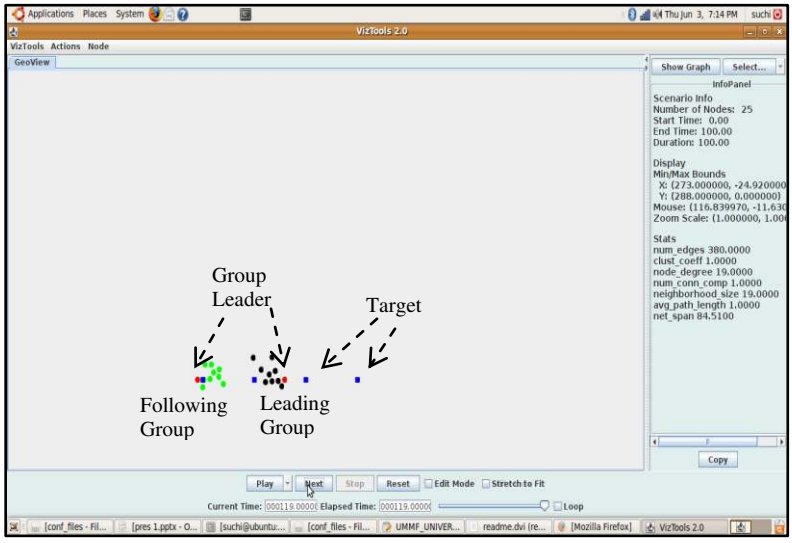


Figure 3-2. UMMF Snapshot of Repeated Traversal

Figures 3-3 and 3-4 show the traces versus time of node degree (graph in red), average path length (graph in blue) and number of connected components (graph in green) as a function of time. The first trace is for a simple scenario with no 'hostile' events whereas the second trace is for a scenario with dynamic hostile events.

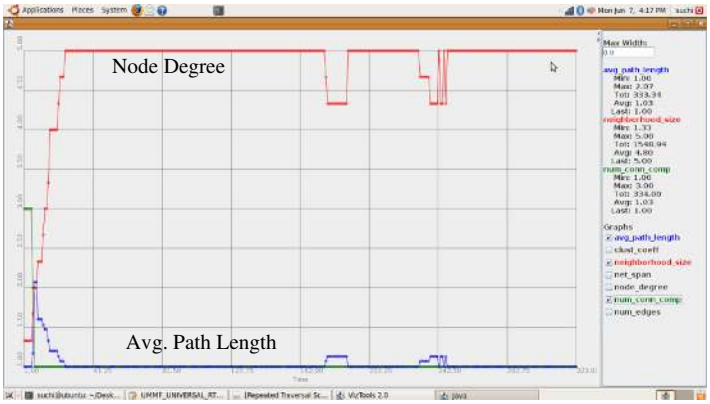


Figure 3-3. Mobility Traces for Repeated Traversal – simple mobility

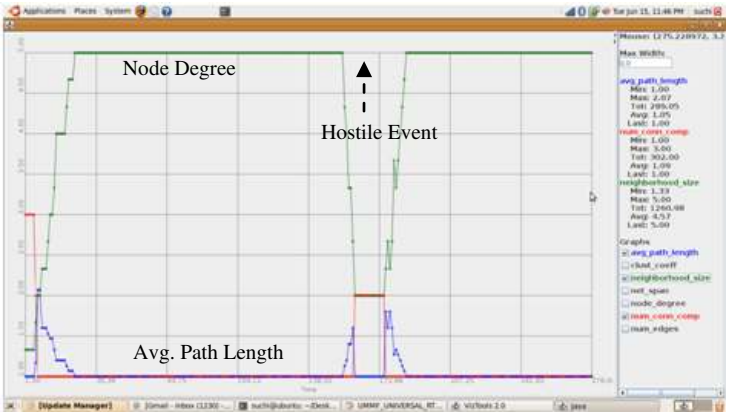


Figure 3-4. Mobility Traces for Repeated Traversal – mobility with hostile events

Analyzing the UMMF traces, the following inferences can be made intuitively. When the leading group moves significantly slower than the following group, the groups remain in communication range for most of the epoch. Similarly, if the following group begins its movement before the leading group and the velocity of the leading group is not considerably higher, the groups remain in contact. In all other cases, the groups are most likely to move out of communication range initially, remain isolated and then come back in contact towards the end of the epoch. Hence, the curve for node degree versus time is typically a downward parabola. Change in group size only shifts the graph along y-axis but does not alter shape. On the other hand, change in relative velocity and pause time does affect the shape of the graph, and in different ways. Behavior of the graph remains consistent for variations in group sizes as well as for simple and complex (involving dynamic events) mobility scenarios.

3.1.2 Motion Model

The motion model of a tactical scenario provides a graphic representation of the movement pattern of nodes. At this stage we create a new layer of abstraction, that of mobility

parameters. These are variables of interest which can be directly calculated from the motion model using fundamental concepts of physics. Some examples are inter-group distance and outage time. We formulate tractable mathematical equations for such parameters and describe their behavior over time. The graph of these mobility parameters can further be used to derive the behavior of other network parameters such as the performance of routing protocols and algorithms in the tactical MANET.

As was inferred in section 3.1.1, Repeated Traversal is an epoch based motion. Figure 3-5 shows the motion model for the base case that consists of two groups moving for one epoch.

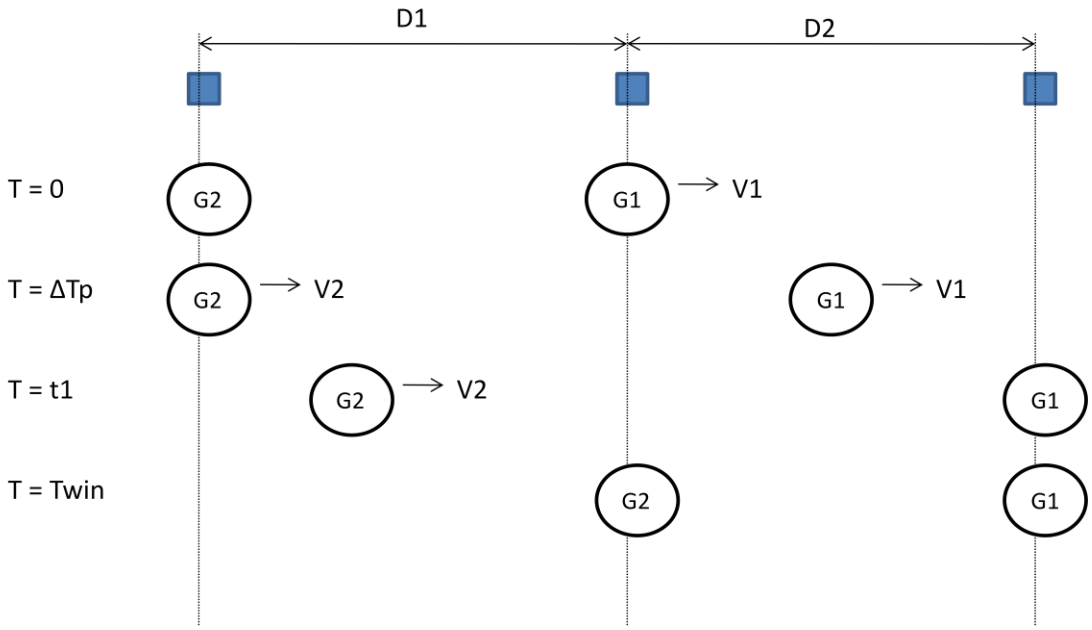


Figure 3-5 Motion Model of Repeated Traversal

The following are the definitions of the notations used in the model. Each epoch has an associated :

- V_1 : velocity of G_1 (leading group)
- V_2 : velocity of G_2 (following group)
- D_1 : distance to next target for G_1

- D_2 : distance to next target for G_2
- ΔT_p : difference in pause times of G_1 & G_2
- T_{win} : size of epoch
- t_1 : time taken by G_1 to reach its target destination; $t_1 = D_1/V_1$
- t_2 : time taken by G_2 to reach its target destination; $t_2 = D_2/V_2$

The epoch begins at time t=0 when G_1 begins moving towards its target destination, which is a distance D_1 away. G_2 waits at its position. ΔT_p time later G_2 begins to move with velocity V_2 as G_1 continues to advance. At time=t_1, G_1 arrives at its target while G_2 is still moving. The epoch finishes at time=T_{win}, when G_2 reaches its destination. G_1 waits at its position till G_2 has finished its movement.

Now let us see how this motion model can be used to analyze the network parameter Node Degree, the UMMF traces of which were discussed in the previous section. To do this, we identify an underlying mobility parameter which can be computed from the mobility model and the behavior of which is linked to that of node degree. The parameter in question is inter-group distance, which we denote as D_g. Further, we denote inter-group distance as a function of time by $D_g(t)$. In the following text we explain the formulation of the equation for $D_g(t)$ and its dependence on the variables V_1, V_2, D_1, D_2 and ΔT_p.

Initially, the distance between the two groups is D_2. For the first part of the epoch ($0<t<\Delta T_p$) D_g increases due to G_1 moving away from G_2. Hence,

$$D_g(t) = D_2 + V_1 \cdot t \qquad [3\text{-}1]$$

For the second part of the epoch ($\Delta T_p < t < t_1$), the D_g changes in accordance to the relative velocity between G_1 and G_2. Hence,

$$D_g(t) = D_2 + V_1 \cdot \Delta T_p + (V_1 - V_2)(t - \Delta T_p) \quad [3\text{-}2]$$

In the last part of the epoch, D_g decreases as G2 comes closer to G1. Hence,

$$D_g(t) = D_2 + V1 \cdot \Delta T_p + (V_1 - V_2)(D_1/V_1 - \Delta T_p) - V_2(t - D_1/V_1) \quad [3\text{-}3]$$

Now consolidating the three cases using unit step functions we get:

$$\boldsymbol{D_g(t) = [D_2 + V_1 \cdot t] \cdot u(\Delta T_p - t) + [D_2 + V_1 \cdot \Delta T_p + (V_1 - V_2) \cdot (t - \Delta T_p)] \cdot u(t - \Delta T_p) \cdot u(D_1/V_1 - t) +}$$

$$\boldsymbol{[D_2 + V_1 \cdot \Delta T_p + (V_1 - V_2) \cdot (D_1/V_1 - \Delta T_p) - V_2 \cdot (t - D_1/V_1)] \cdot u(t - D_1/V_1) \cdot u(T_{win} - t)} \quad [3\text{-}4]$$

Figure 3-6 illustrates the typical behavior of D_g when V_1 is considerably greater than V_2. This is the case we are most interested in since there is the highest likelihood of loss of communication between the groups. As can be seen, $D_g(t)$ begins at an initial value (here D_2) and increases till a maximum value of $D_{g,max}$ (point where the groups are farthest apart). Then it decreases and settles at the value D_1. Now, let us assume that once inter-group distance increases beyond a value of $D_{threshold}$, the two groups lose contact. So given the communication model (and hence $D_{threshold}$) we can tell not only when we can expect a network partition (a point where node degree would drop to the value of the group size) but also for how long we can expect the partition to remain. This is shown in this figure by the parameter T_{out}, which denotes outage time.

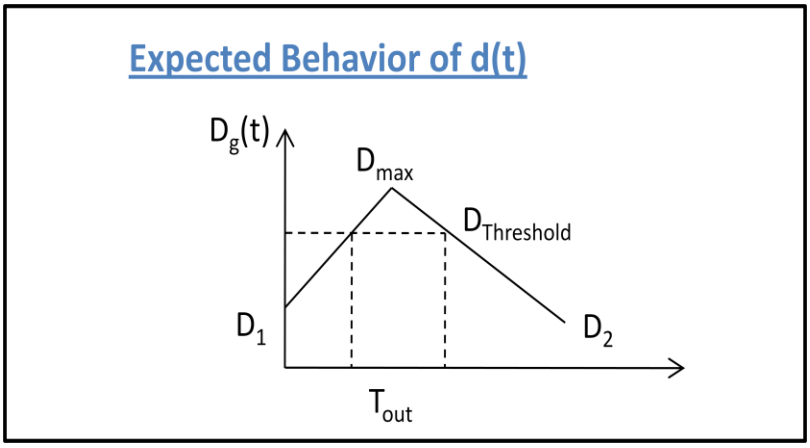


Figure 3-6 Behavior of inter-group distance over time for Repeated Traversal

Also, using the signal propagation model for a specific military application undergoing repeated traversal, a clear mathematical relationship can be established between the signal strength and inter-group distance. This will easily enable the computation of node degree at any given instant of time during the mission.

3.1.3 Curve Fitting

In the third stage of modeling Repeated Traversal, we try to find the 'best fit' equations to the curves generated for network parameters by UMMF in section 3.1.3. Here again we take the specific example of node degree. We have seen in section 3.1.3 the effects input parameters like group velocity and group size have on node degree. In this section we try capture these 'effects' mathematically. The objective is to construct tractable mathematical equations for node degree as a function of time ($\lambda(t)$) for a given epoch, with the aim of performing regression analysis. Regression analysis is a technique that allows one to understand how the value of a dependent variable (here node degree) changes when any one of the independent variables is varied while all other independent variables are kept fixed. Similar work has been done in [16-17]. Since the data points used here are those generated by the UMMF traces, the regression analysis is in essence curve fitting done for the mobility traces.

The problem further complicates as the dependent variable, node degree, being modeled here is in itself a function of time, denoted by $\lambda(t)$. Hence, the problem is approached in two steps. In the first step we fit a generic curve for $\lambda(t)$, which captures the broad behavior of node degree as a function of time, for any given set of input parameters. Figure 3-7 shows the true nature of the curve $\lambda(t)$ and the approximate curve that we use to describe it.

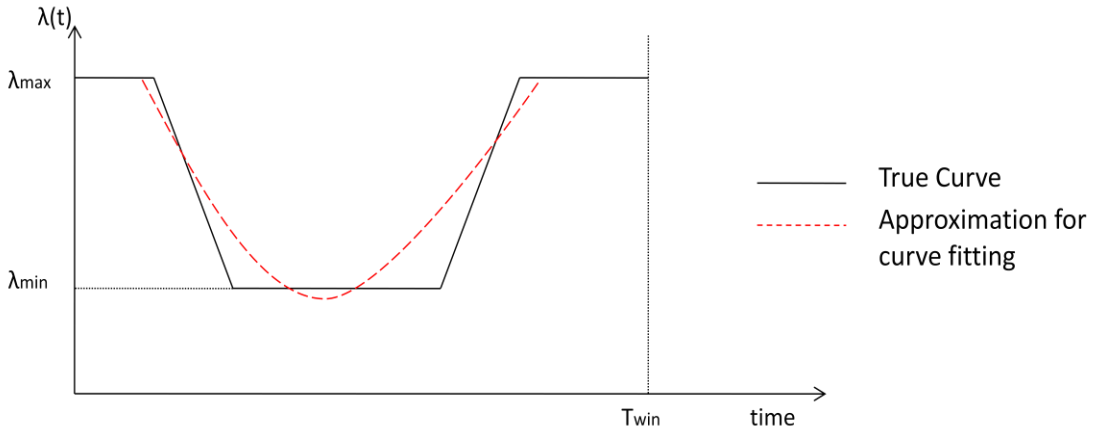


Figure 3-7 Curve Fitting for λ(t) in Repeated Traversal.

N_{tot}: total number of nodes in mission, N_{grp}: Number of nodes in a group, $\lambda_{max} = N_{tot} - 1$, $\lambda_{min} \in (\lambda_{max}, N_{grp} - 1)$

The true curve (shown in black in the figure) is the actual UMMF trace of λ(t), for a given set of input parameters and a particular seed value. To perform curve fitting, we run simulations with twenty different seed values for the same set of input parameters and averaged out the results. Hence, we obtain the approximate curve (shown in red in the figure), which captures the behavior of node degree within tolerable error margins. We have discussed these errors in detail in chapter 4. The general equation of Node Degree (λ) as a function of time (t) is of the form:

$$\lambda(t) = \alpha t^2 - \beta t + \gamma \qquad [3\text{-}5]$$

where, $t \in [0, T_{win})$ and coefficients α, β, γ are functions of the independent parameters.

Now we go on to identify the independent input parameters which affect the shape of the curve $\lambda(t)$, and hence affect coefficients α, β, γ. From the inferences of the trace based analysis, in section 3.1.3, we find group velocity and pause time to be two such parameters. Other variables

such as group size and inter-target distance were found to only shift the curve along the vertical or horizontal axis (respectively) by an offset. Hence, we keep such quantities out of the curve fitting model and perform time series analysis of λ with respect to group velocity (V) and pause time (T_p).

In this work, we use an online curve fitting tool [18] which uses the least squares approach to curve fitting, i.e. it tries to fit a given set of data points to a standard set of 2D or 3D functions with the aim of minimizing the sum of squared residuals. Similar technique has been studied and used in [19]. On performing curve fitting the following equations were formulated, which define the dependence of coefficients α, β, γ on V and T_p.

$$\boldsymbol{\alpha = a_1+b_1T_p+c_1V+d_1VT_p} \quad [3\text{-}6]$$

a= -7.77e-3 RMSE= 9.40e-4

b= 3.89e-3 SSQABS= 8.84e-06

c=-2.54e-4

d= -8.46e-5

$$\boldsymbol{\beta = a_2+b_2ln(V)+c_2ln(T_p)} \quad [3\text{-}7]$$

a=1.32e-1 RMSE= 1.16e-3

b=-1.64e-1 SSQABS= 6.71e-06

c=-5.43e-2

$$\boldsymbol{\gamma=a_3+b_3V+c_3T_p} \quad [3\text{-}8]$$

a=5.32 RMSE= 3.53e-3

b=2.52e-2 SSQABS= 6.24e-05

c=-1.85e-4

In the above equations, RMSE is the root mean squared error, which is the difference between the values predicted by the model and the values actually observed from simulations. The sum of squares absolute error (SSQABS) is the square of RMSE. The results of curve fitting comply with

the requirements of regression analysis [16-17], wherein the error is a normally distributed random variable with zero mean and constant variance.

In summary, the dependence of $\lambda(t)$ on V and T_p is described by the following equation:

$$\lambda(t) = [a_1+b_1T_p+c_1V+d_1VT_p]\, t^2 - [a_2+b_2ln(V)+c_2ln(T_p)]\, t + a_3+b_3V+c_3T_p \quad [3\text{-}9]$$

Given the velocity and pause time of two military troops performing Repeated Traversal in a specific epoch, the above equation calculates node degree at any time instant in the epoch, within the error margins.

3.2 Bounding Overwatch

In the following sub-sections, we describe the model for the Bounding Overwatch military tactic.

3.2.1 Scenario Description and Traces

The Bounding Overwatch tactic [14] was designed to covertly capture the enemy (usually an enemy post or a small enemy troop) while keeping them engaged in other activity. Two or more troops work in collaboration such that one (or more) troop(s) covers by firing at the enemy and keeps them engaged while the other troop(s) advances to a new position, closer to the enemy. The firing troop(s) always halts at a position from where it is able to fire at the enemy and is behind an obstacle covering it. Once the advancing troop reaches its new target, it begins to fire from there while one of the troops that were firing previously advances to a new position. This pattern continues till one of the troops is able to reach the enemy post.

The movement pattern in Bounding Overwatch is found to be regular. The scenario shows group mobility and the groups typically form reliable neighborhoods. Hence, a set of groups (forming a neighborhood) regularly go from ‘firing’ mode (where they remain static) to ‘advancing’ mode (where they move to a common destination, trying to maintain the neighborhood). The following figure shows two groups executing Bounding Overwatch.

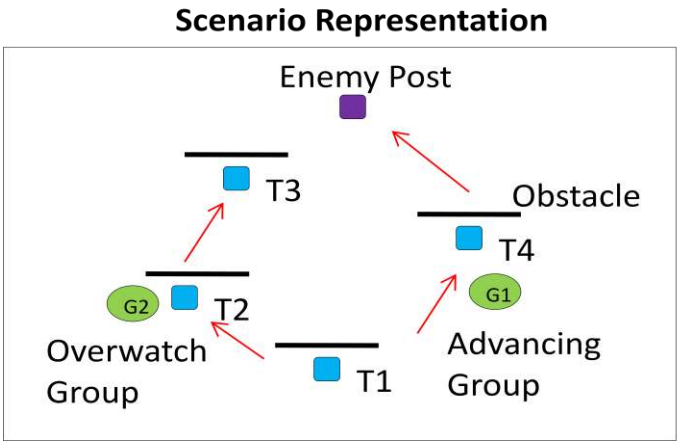


Figure 3-8. Schematic view of Bounding Overwatch.

This scenario was simulated in UMMF as an epoch-based model with a base case of two neighborhoods in one epoch, and one group in each neighborhood. The following features of the tool were used.

1. Nodes move in groups
2. Inter-group communication
3. Static targets placed at geographical locations behind obstacles
4. Steering Behaviors used: Group leaders – 'Arrive', Group members – 'Pursuit'
5. Nodes do 'Obstacle Avoidance' while advancing from one target to the other.

Figure 3-9 is a snapshot of the simulation, showing two groups executing the Bounding Overwatch tactic. The enemy post is assumed to be behind the top most obstacle. At the instant of time shown below, the black group is firing at the enemy and covering for the green group, as they advance towards the enemy.

Figure 3-9. UMMF Snapshot of Bounding Overwatch

Figures 3-10 and 3-11 show the traces for mobility parameters like average path length, neighborhood size and number of connected components as a function of time, for the simple case (with no hostile events) and the case including the occurrence of dynamic hostile events.

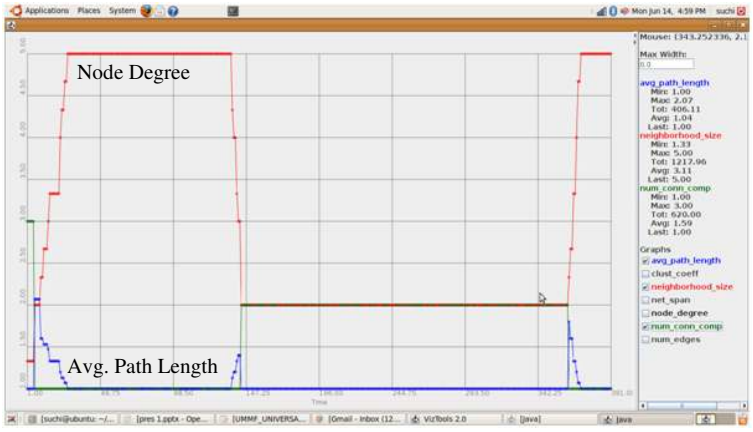


Figure 3-10. Mobility Traces of Bounding Overwatch – simple mobility

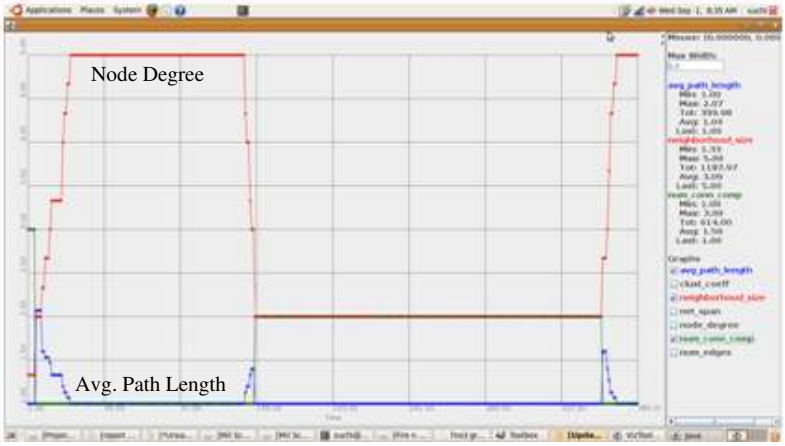


Figure 3-11. Mobility Traces of Bounding Overwatch – with hostile events

Here again the UMMF traces give some insight into the general behavior of node degree that can be expected in the Bounding Overwatch scenario. The periodicity is clearly seen in $\lambda(t)$ as its value rises to a maximal value, when communication is possible across neighborhoods, and then falls to a minimal value, when only nodes within a neighborhood are in contact. Again we find that change in group size only shifts the graph along y-axis but does not alter shape, while change in group velocity alters the shape of the graph. Behavior of the graph remains consistent for variations in group sizes as well as for simple and complex (involving dynamic events) mobility scenarios.

3.2.2 Motion Model

Figure 3-12 is a schematic representation of the motion model for Bounding Overwatch.

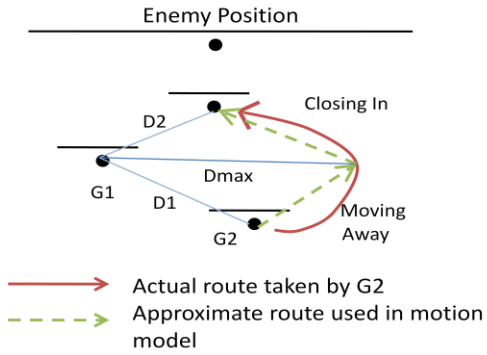


Figure 3-12. Motion model for Bounding Overwatch

The following are the definitions of the notations used in the model. Each epoch has an associated:

- V : velocity of moving group (G_2)
- D_1 : initial inter-group distance
- D_2 : final inter-group distance
- D_{max}: farthest that G_2 can go from G_1
- T_{win} : size of epoch
- Θ_1: initial angle of velocity vector
- Θ_2: final angle of velocity vector

One group (G_1 in above figure) remains stationary and fires at enemy while the other group (G_2 in the figure) advances to its next target. The approximate route taken by G_2 consists of two parts:

1. The 'Moving Away' phase, where inter-group distance increases
2. The 'Closing In' phase where inter-group distance decreases.

We know take a closer look at the motion of G_2 in the Figure 3-13.

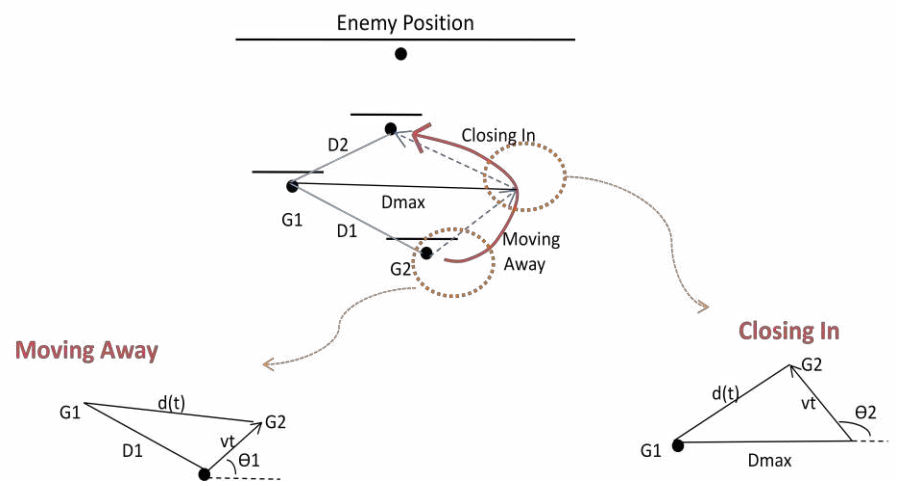


Figure 3-13. Motion Model for Bounding Overwatch.

Here, G_2 moves away from G_1 at an angle of θ_1, till it reaches a maximal distance of D_{max}. It then begins closing in towards its target at an angle of θ_2. Using the law of cosines, we formulate the following equations that describe the inter-group distance (between G_1 and G_2) as a function of time (denoted here as $d(t)$).

For 'Moving In' phase:

$$d^2(t) = D_1^2 + (vt)^2 - 2(D_1)(vt)\cos(\prod - \Theta_1) \quad [3\text{-}10]$$

For 'Closing In' phase:

$$d^2(t) = D_{max}^2 + (vt)^2 - 2(D_{max})(vt)\cos(\prod - \Theta_2) \quad [3\text{-}11]$$

The following graph shows the expected behavior of $d(t)$ for the epoch shown in figure 3-14.

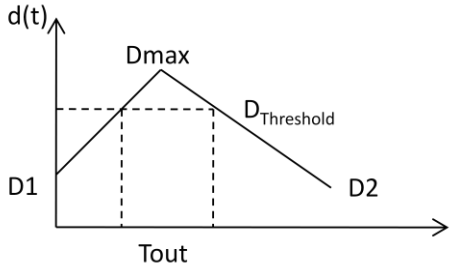


Figure 3-14. Expected behavior of inter-group distance for Bounding Overwatch.

Given a threshold distance, beyond which groups loose contact, the outage time can easily be calculated, as shown in the figure above. Also, given the signal propagation model, the motion model equations can be used to calculate node degree at any instant of time within the epoch.

3.2.3 Curve Fitting

In the third stage of modeling Bounding Overwatch, we perform curve fitting using the tool and approach described in section 3.1.3. Figure 3-15 shows the approximate curve for which we aim to find the best fit equation.

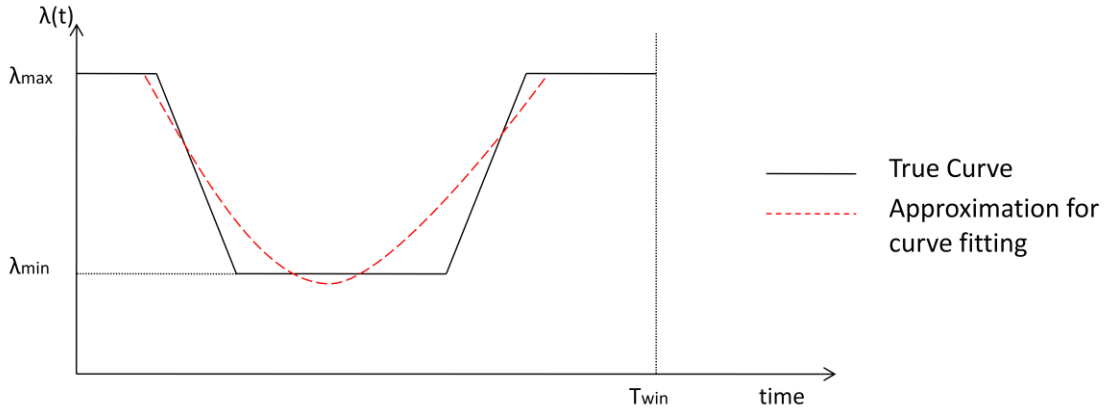


Figure 3-15 Curve Fitting for λ(t) in Bounding Overwatch.

N_{tot}: total number of nodes in mission, N_{grp}: Number of nodes in a group, $\lambda_{max} = N_{tot} - 1$, $\lambda_{min} \in (\lambda_{max}, N_{grp} - 1)$

The general equation for this curve is given by:

$$\lambda(t) = \alpha t^2 - \beta t + \gamma \qquad [3\text{-}12]$$

where, $t \in [0, T_{win})$ and coefficients α, β, γ are functions of the independent input parameters.

Here again, we identify the input parameters that affect the shape of $\lambda(t)$. From the UMMF traces that were analyzed in section 3.2.1, group velocity *(V)* was found to be the input parameter that had a significant impact on the behavior of $\lambda(t)$. All other variables, such as number of groups, group size as well as inter-target distance were found to merely shift the parabola of $\lambda(t)$ by an offset along the *y* and *x* axis (respectively). Hence, we perform curve fitting of $\lambda(t)$ with respect to group velocity, *V*. As a result, the following equations were formulated, which describe the dependence of coefficients α, β, γ on *V*.

$$\alpha = 1/aV(1+b\, e^{cV}) \quad [3\text{-}13]$$

a=2.1997e1 RMSE= 1.82e-3

b=1.864e4 SSQABS= 1.66e-5

c=-2.679

$$\beta = a/(1+b\, e^{-c\,V}) + Offset \quad [3\text{-}14]$$

a=-3.04 RMSE= 1.91e-3

b=5.69 SSQABS= 1.81e-05

c=5.96

Offset=-5.62

$$\gamma = aV+b \quad [3\text{-}15]$$

a=0.3099 RMSE= 1.3e-1

b=4.479 SSQABS= 8.51e-2

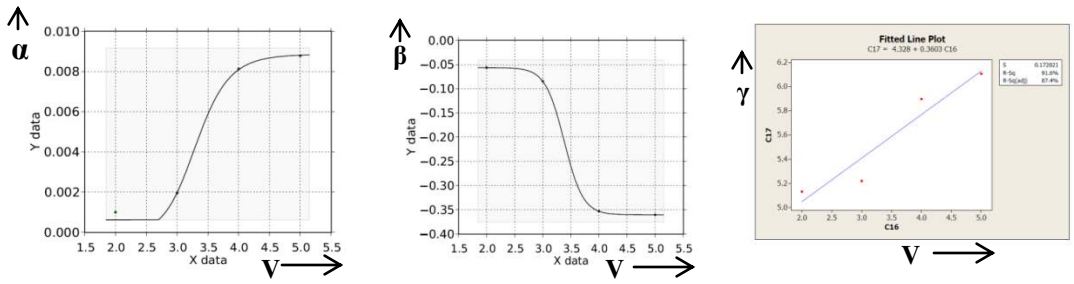


Figure 3-16 Coefficients of λ (t) and their dependence on group velocity, in Bounding Overwatch

In summary, the equation for $\lambda\ (t)$ as a function of time and group velocity is:

$$\boldsymbol{\lambda\ (t) = [1/a_1 V(1+b_1\, e^{c_1 V})]t^2 - [a_2/(1+b_2\, e^{-c_2 V}) + Offset]t + a_3 V + b_3} \qquad [3\text{-}16]$$

Figure 3-16 gives a schematic view of how coefficients α, β, γ (left to right) depend on V. The curve fitting equations coupled with the UMMF traces provide the following insights into this scenario. As group velocity increases, the amount of time for which groups in different neighborhoods remain out of contact reduces. But, with increasing group velocity, link stability between groups decreases. Nodes typically try to maintain reliable neighborhoods and do not move between neighborhoods. Communication between neighborhoods is usually multi-hop whereas within a neighborhood it can be both one hop or multi-hop.

3.3 Pincer Movement

In the following sub-sections, we describe the model for the military tactic called Pincer Movement.

3.3.1 Scenario Description and Traces

The third military tactic studied in this thesis is known as Pincer Movement, Pincer attack or Double Envelopment. Described in detail in [13-14], the objective here is to surround the enemy troop from two or more directions, making it very easy to destroy them. This movement is a flanking attack from both sides of the enemy troops. A flanking attack is an attack on the sides of an opposing force. More units of the attacking force can move in from the front and rear and result in encircling the enemy. To avoid being a victim of this a group of nodes can ensure that it is flanked by difficult terrain on both sides (eg, river, marshy land etc.). The key mobility features of this tactic are fast movement, small reaction time, small groups. The following figure is a schematic representation of this scenario.

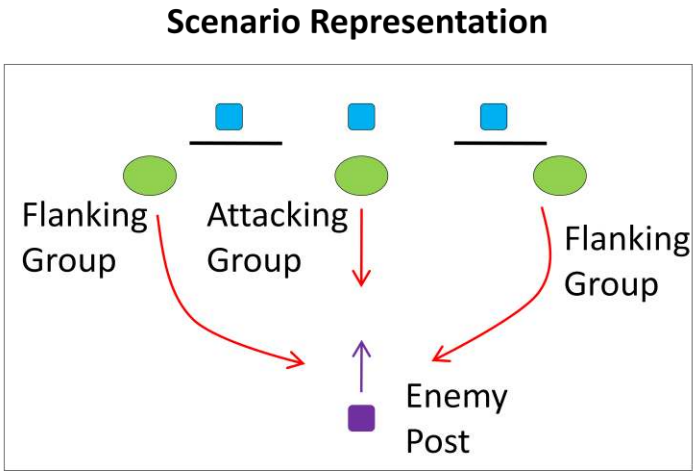


Figure 3-17. Schematic view of Pincer Movement

In contrast to the earlier two models, this scenario is not periodic or epoch based. The motion is comparatively less structured and defined. The specifications that can be defined for this scenario are the initial position of nodes (nodes are usually placed far from the enemy, behind obstacles), final position of nodes (final targets are strategically placed so that nodes encircle the enemy) and the velocity with which nodes advance. Here we simulate Pincer Movement with groups rather than individual nodes, since it is more common for small groups of soldiers to perform this task rather than individual soldiers scattering around the region.

The following features of the UMMF tool were used while simulating this scenario:

1. Nodes move in groups
2. Specification of initial node positions, behind obstacles.
3. Strategic targets placed to bring out the motion of 'encircling'
4. Steering Behaviors used: Group leaders – 'Arrive', Group members – 'Pursuit'
5. Nodes do 'Obstacle Avoidance' while advancing from their initial positions.

Figure 3-18 is a snapshot of an example Pincer scenario with three groups (in green) trying to encircle the enemy troop (in black). The group leaders are shown by the red nodes. One group attacks the enemy head on (centre group), whereas the other two flank the enemy from either side.

Figure 3-18. UMMF Scenario snapshot of Pincer Movement

Figures 3-19 and 3-20 show the traces for node degree, average path length and number of connected components as a function of time, for the simple case (with no hostile events) and the case including the occurrence of dynamic hostile events, respectively.

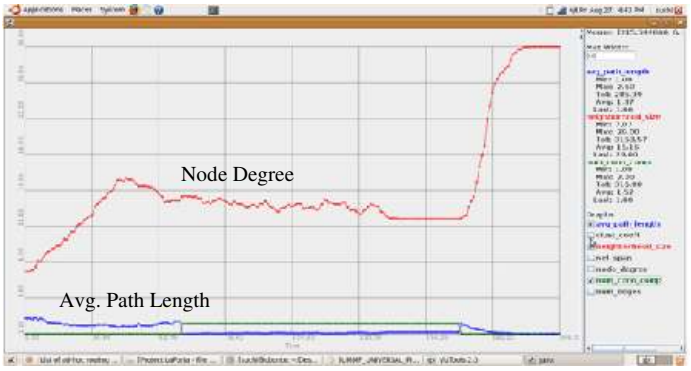


Figure 3-19 Mobility Traces for Pincer Movement (simple mobility)

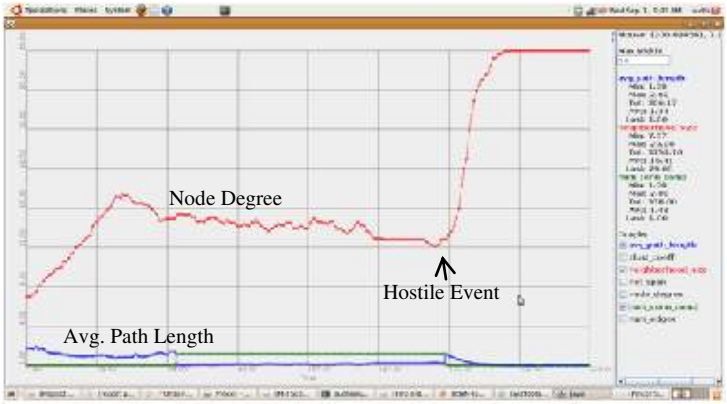


Figure 3-20 Mobility Traces for Pincer Movement (mobility with hostile events)

Running several simulations, for different values of group velocity, initial node positions and group sizes, we find that the behavior of node degree shows a decline followed by a steep rise. The value of node degree falls in the first part of the movement as the groups move away from each other while advancing towards the enemy. In the second part, the groups come closer to one another as they try to encircle the enemy and finally place themselves such in a close knit

circle where most of the nodes are in contact. The simulations also show group velocity and initial inter-group distance (i.e. the distance between neighboring groups just before the beginning of the movement) to be two input parameters which affect the curve of node degree substantially. Other variables such as group size and number of nodes only shift the curve along the y-axis by an offset.

Since the Pincer movement does not follow any specific pattern, we do not attempt to design a motion model for it and instead go on to the third stage of finding the best fit curve to the UMMF traces and performing regression analysis.

3.3.2 Curve Fitting

In the third stage of modeling Pincer Movement, we perform curve fitting using the tool and approach described in section 3.1.3. Figure 3-21 shows the approximate curve for which we aim to find the best fit equation.

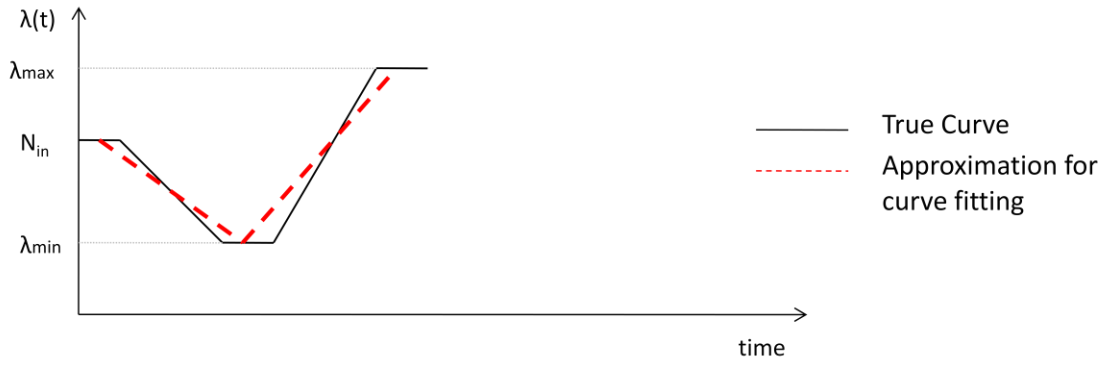


Figure 3-21. Curve Fitting in Pincer Movement

N_{tot}: total number of nodes in mission; N_{grp}: Number of nodes in a group; N_{in}: number of nodes in contact initially; $\lambda_{max} = N_{tot} - 1$; $\lambda_{min} \in (\lambda_{max}, N_{grp} - 1)$

As can be seen in figure 3-21, the trace of node degree is approximated to a curve consisting of two line segments describing the two parts of Pincer movement – the 'Advancing' phase (the downslope) and the 'Surrounding' phase (the upslope). The general equation that describes node degree as a function of time for both these is phases in given below.

$$\boldsymbol{\lambda(t) = mt + c} \quad [3\text{-}17]$$

where, $\boldsymbol{m}$ (slope of line) and $\boldsymbol{c}$ (y-axis intercept) are both functions of the independent input parameters.

From the trace based analysis in section 3.3.1, the input parameters which were found to have a significant impact on the trace of $\lambda(t)$ are the initial inter-group distance (denoted here by D_i) and group velocity (V). Using 3D curve fitting we formulate the following equations which quantify the effect of D_i and V on the slope of the curve.

For downslope: $\boldsymbol{m = a + b*ln\,(D_i) + c*ln(V)}$ [3-18]

a = 2.34E-01

b = -2.51E-01

c = 4.24E-01

sum of squared absolute error = 4.55E-02

For upslope: $\boldsymbol{m = a + b*ln\,(D_i) + c*ln(V)}$ [3-19]

a = -1.32E-01

b = 2.96E-02

c = 1.79E-01

sum of squared absolute error = 1.78E-03

Hence, summing these results, we have

$$\boldsymbol{\lambda\,(t) = [a + b*ln\,(D_i) + c*ln(V)]t + offset} \quad [3\text{-}20]$$

As can be seen in Figure 3-21, the offset can easily be calculated for each phase, given the values of N_{in}, λ_{max} and λ_{min}.

3.4 Patrolling

In the following sub-sections, we describe the model for the Patrolling military tactic.

3.4.1 Scenario Description and Traces

The final military scenario studied in this thesis is known as Patrolling, described further in [13-14]. During Patrolling, small groups or individual units are deployed from a larger formation to achieve a specific objective and then return to base. Different patrols have different objectives, as stated below:

1. **Reconnaissance Patrol** aims at covertly gathering information about enemy positions
2. **Clearing Patrol** aims at securing a newly occupied defensive position
3. **Fighting Patrol** is organized to raid or ambush the enemy.

From a mobility perspective, in this scenario nodes visit a set of targets in a manner that best serves the objective of the Patrol. It is common for nodes to have targets which are more 'important' than others where the nodes would like to spend more time. For example, soldiers in a reconnaissance patrol would be more interested in spying over larger enemy camps. Hence, in this model we introduce the idea of targets having different weights. Depending on the objective of the patrol, these weights indicate the 'importance' level of a target, a lower weight implying a higher desirability of visiting that target. Hence, we can say that nodes try to take the shortest/minimum weight path to cover all targets along its patrol.

The following features were incorporated in the UMMF tool while simulating this scenario.

- Nodes move individually

- Targets are geographic and known beforehand
- Route calculation is strategic
- Choice of speed may be random but choice of direction is as per strategy

Figure 3-22 is a snapshot of an example Patrolling scenario with sixteen nodes (in black) patrolling through 16 targets (in blue). The targets are placed in a 4x4 grid, with decreasing weight (and hence decreasing importance) from top to bottom. Each node is given a set of 3-4 targets from the grid. Hence, two nodes could have a few targets in common between them.

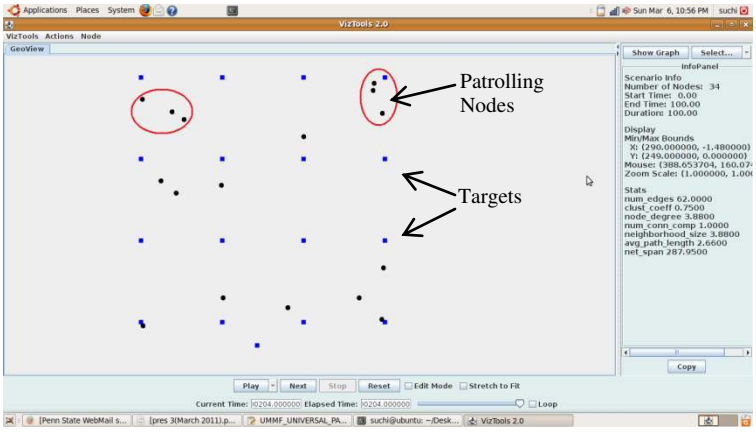


Figure 3-22. UMMF Snapshot of Patrolling

The simulations clearly show nodes aggregating occasionally near targets of higher importance. This is shown in the above figure by red circles. This is also clear from the spikes in the trace of node degree, shown below in Figure 3-23.

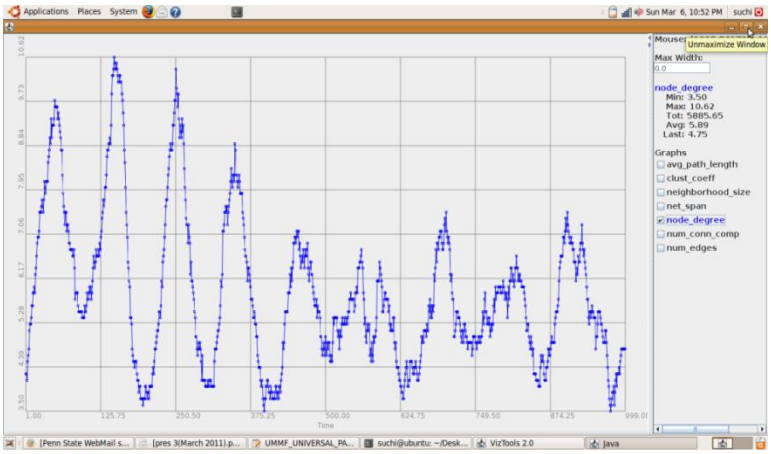


Figure 3-23. Mobility trace for Patrolling

The UMMF traces of Patrolling show an inherent characteristic of this model, which is that node degree increases for short durations around targets of higher importance. This directly implies that every now and then a set of nodes come in contact, which can be leveraged for timely and efficient sharing of information. It can also be seen that if all targets in this scenario are of equal weight the Patrolling model degenerates to the Random Waypoint Model, described in section 2.1. But as we have stressed before, in real life military applications, the movement of soldiers would always be defined by an objective, rather than being random. We hence develop a motion model for Patrolling, in the following section, that is able to capture the real life behavior more accurately.

3.4.2 Motion Model

In this section we devise an algorithm that each node doing Patrolling must follow. The objective of each node performing a patrol is to incur maximum returns before returning to base. A natural approach to modeling this problem is to attach weights to the edges between each pair of targets that a node is to visit. The problem then is to find the minimum weight route that visits

each target exactly once. But, this problem is an instance of the famous Travelling Salesman Problem [20], which is known to be NP hard.

Hence, in this thesis we devise an alternate way of modeling the mobility which simplifies the problem and captures an approximate behavior. Following are the specifications of the model:

- Each target in the scenario is given an associated 'weight' (w_i), denoting its level of importance
- Each node is assigned a set of targets, $\{X_1,\ldots,X_n\}$ that it has to visit during its patrol. Total weight of targets is w_{tot}
- Each node is given a total time t_{Tot} within which it has to complete the mission and return to base.
- Nodes try to divide their time between targets, in proportion to the target weights.

We solve the problem using a greedy approach that makes the locally optimal choice at each step, hoping to find the global optimum [21]. Each node executes the following:

do{

calculate min{weight} from targets yet to be visited;

select target X_i as next target such that $w_i = min\{weight\};$

spend time = $t_{Tot} *(w_i/w_{tot})$ at target X_i;

remove X_i from target list $\{X_1,\ldots,X_n\}$;

} continue until all targets have been visited;

The UMMF traces of Patrolling reveal that node degree spikes occasionally, but there is no periodic behavior or dependence on input parameters that the trace shows. Hence, we do not attempt to do curve fitting for this scenario.

Having discussed the UMMF traces, motion models and mathematical equations for the four military models, we discuss the errors between these three approaches in the following chapter.

Chapter 4

Error Analysis

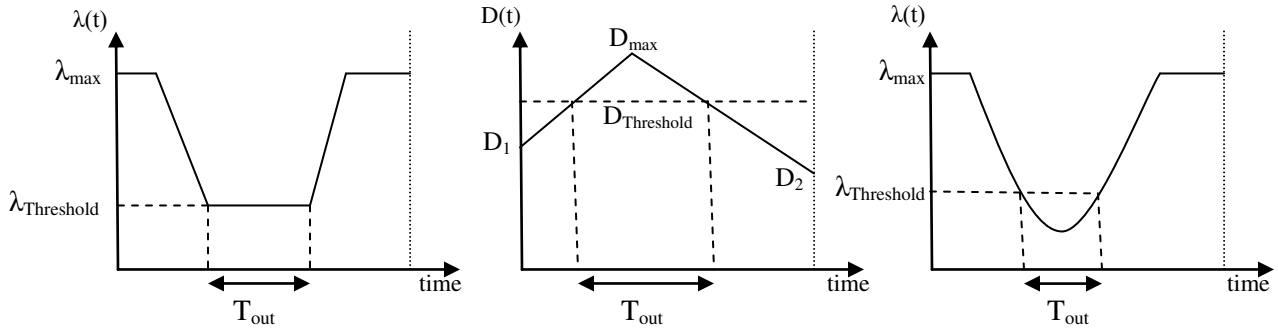
In the previous chapter we illustrate the following three approaches taken to model tactical mobility:

1. The UMMF Model – Mobility traces generated by the UMMF tool.
2. The Motion Model – Equations that characterize the physical movement pattern
3. The Curve Fitting Model – Mathematical equations formulated through regression analysis.

As mentioned earlier, we consider the traces generated by the UMMF model to represent the ground truth for the purpose of this work. In this chapter, we aim to capture the error margin by which the Motion Model and Curve Fitting Model deviate from the UMMF traces. This error analysis is done for Repeated Traversal, Bounding Overwatch and Pincer Movement scenarios. In the case of Patrolling, since the motion model is more qualitative and described by way of an algorithm, it is not feasible to quantify its error in the manner described in this chapter.

To perform the error analysis, for each scenario we calculate the value of an example output parameter, namely outage time (T_{out}), using the three models and compare the results, for several sets of input parameters. We define T_{out} broadly as the period of time for which inter-group communication is impaired. Scenario specific definitions are included in the following sections.

Figure 4-1 below is a schematic representation of the calculation of T_{out} from the graph of node degree versus time ($\lambda(t)$) or inter-group distance versus time ($d(t)$) as generated by each of the three models mentioned above.



(a) UMMF Model (b) Motion Model (c) Curve Fitting Model

Figure 4-1. Calculation of T_{out} in tactical mobility models

N_{tot}: total number of nodes in mission; N_{grp}: Number of nodes in a group; $\lambda_{max} = N_{tot} - 1$; $\lambda_{Threshold} = N_{grp} - 1$; D_1: initial inter-group distance; D_2: Final inter-group distance; $D_{Threshold}$ = communication range.

The relation between $\lambda_{Threshold}$ and $D_{Threshold}$ is defined by the communication model used in the UMMF simulator. UMMF being a mobility simulator uses simple Line of Sight, whereby communication between two nodes/groups is lost once they move apart by a certain distance. Hence, we define $D_{Threshold}$ as the maximal inter-group distance at which communication between groups is impaired, resulting in node degree reducing to $\lambda_{Threshold}$.

4.1 Error Analysis for Repeated Traversal

For the purpose of error analysis we consider the simplest case of Repeated Traversal, consisting of two groups undergoing this tactic in a single epoch. The UMMF traces, motion model and curve fitting equations for this case are discussed extensively in section 3.1, where we found group velocity (V) and pause time (T_p) to be the main two input parameters that impacted the behavior of node degree.

Here, T_{out} is specifically defined as the time for which the 'leading' and 'following' groups go out of communication range within the epoch. We calculate T_{out} from the UMMF model, the Motion model and the Curve Fitting model for the following three sets of input parameters:

i. Case 1:
   - Low group velocity (V)
   - $V_1 = V_2$; velocity of leading and following groups are equal
   - Small pause time (T_p)

ii. Case 2:
   - High group velocity (V)
   - $V_1 = V_2$; velocity of leading and following groups are equal
   - Large pause time (T_p)

iii. Case 3:
   - Low group velocity (V)
   - $V_1 > V_2$; velocity of leading is greater than following group
   - Small pause time (T_p)

For the Motion Model, the outage time is calculated using equation 3-4, restated below:

$$D_g(t) = [D_2 + V_1 \cdot t] \cdot u(\Delta T_p - t) + [D_2 + V_1 \cdot \Delta T_p + (V_1 - V_2) \cdot (t - \Delta T_p)] \cdot u(t - \Delta T_p) \cdot u(D_1/V_1 - t) +$$

$$[D_2 + V_1 \cdot \Delta T_p + (V_1 - V_2) \cdot (D_1/V_1 - \Delta T_p) - V_2 \cdot (t - D_1/V_1)] \cdot u(t - D_1/V_1) \cdot u(T_{win} - t)$$

For the Curve Fitting model, outage time is calculated using equation 3-9, restated below:

$$\lambda(t) = [a_1 + b_1 T_p + c_1 V + d_1 V T_p]\, t^2 - [a_2 + b_2 \ln(V) + c_2 \ln(T_p)]\, t + a_3 + b_3 V + c_3 T_p$$

The following figure is a bar chart that schematically represents the values of outage time obtained from each model in all three cases.

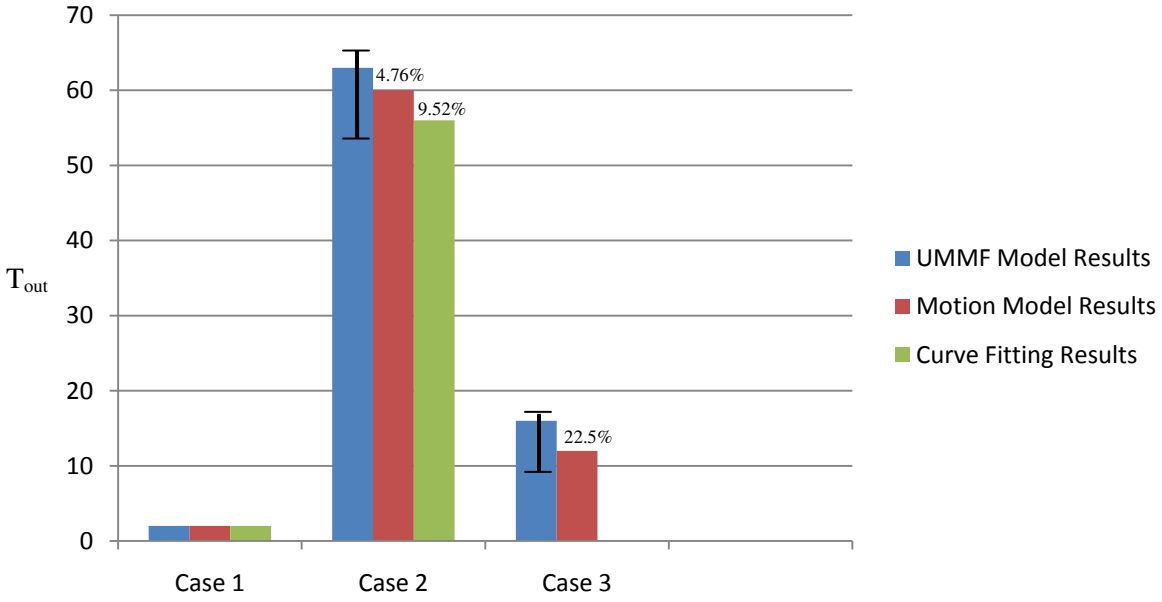


Figure 4-2. Bar chart representing error analysis results for Repeated Traversal.

The mark on the UMMF model bar shows the range of T_{out} results obtained from UMMF, for ten runs of the simulation, with a variety of seed values and group sizes. The figure clearly shows that the T_{out} values calculated from the Motion Model and Curve Fitting Model always remain within the bounds of the UMMF results. Further, the figure also shows the error margin by which each model deviates from the UMMF traces. In Case 3 we do not consider the Curve Fitting model since this model assumes equal velocity for both groups and hence is not able to capture a scenario where V_1 is greater than V_2. The above figure not only captures the level of accuracy of the Motion and Curve Fitting models but also justifies our claim that group size does not affect the shape of the curve of node degree in Repeated Traversal (described further in section 3.1.1).

4.2 Error Analysis for Bounding Overwatch

In our second tactical scenario, Bounding Overwatch, we again consider the simplest case wherein two groups undergo this mission, with one group being stationary and the second group advancing in each epoch. The corresponding UMMF traces, motion model and curve fitting equations are discussed extensively in section 3.2. Here the Curve Fitting model has the input parameter of group velocity (V) and the Motion Model uses the direction of group velocity (θ) as an input, in addition to V.

For this scenario, T_{out} is specifically defined as the time for which the advancing group goes out of contact from the stationary group within the epoch. Extending this to the general case where several groups forming separate neighborhoods perform Bounding Overwatch, T_{out} is the period of time for which communication between all neighborhoods is impaired. We calculate T_{out} separately using the UMMF model, the Motion model and the Curve Fitting model for the following three sets of input parameters:

i. Case 1:
   - Low group velocity (V)
   - D_1 (initial inter-group distance) and D_2 (final inter-group distance) are comparable
   - Medium angles of Moving Away and Closing In phases
     - Specifically: $\theta_1 = \pi/4$; $\theta_2 = 3\pi/4$

ii. Case 2:
   - High group velocity (V)
   - $D_1 >> D_2$
   - $D_{max} = D_1$; Inter-group distance starts at D_1 and decreases to D_2
   - Movement consists of only the Closing In phase and no Moving Away phase

- Specifically: $\theta_2 = 3\pi/4$

iii.Case 3:

- Low group velocity (V)
- $D_1 << D_2$
- $D_{max} = D_2$; Inter-group distance starts at D_1 and increases to D_2
- Movement consists of only the Moving Away phase and no Closing In phase
  - Specifically: $\theta_1 = \pi/4$

For the Motion Model, the outage time is calculated using equations 3-10 and 3-11, restated below:

$$d^2(t) = D_1^2 + (vt)^2 - 2(D_1)(vt)\cos(\prod - \Theta_1),$$ for the 'Moving Away' phase

$$d^2(t) = D_{max}^2 + (vt)^2 - 2(D_{max})(vt)\cos(\prod - \Theta_2),$$ for the 'Closing In' phase

For the Curve Fitting model, outage time is calculated using equation 3-16, restated below:

$$\lambda(t) = [1/a_1 V(1+b_1 e^{c_1 V})]t^2 - [a_2/(1+b_2 e^{-c_2 V}) + Offset]t + a_3 V + b_3$$

The following bar chart schematically represents the values of outage time obtained from each model in all three cases

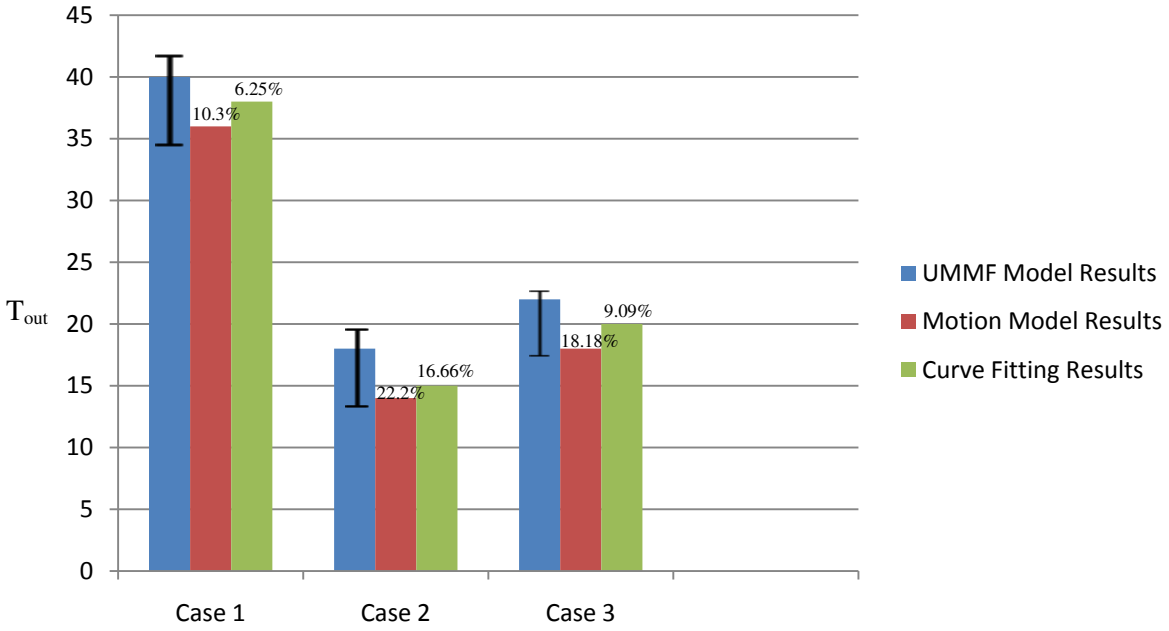


Figure 4-3. Bar chart representing error analysis results for Bounding Overwatch

Here again, the mark on the UMMF model bar shows the range of T_{out} results obtained from the tool for ten runs of simulations with a variety of seed values. The figure clearly shows that the T_{out} values calculated from the Motion Model and Curve Fitting Model always remain within the bounds of the UMMF results. The specific error margin for each model has been specified above the corresponding bar, for each case. It is also interesting to note that, across cases 2 and 3 we vary inter-target distance and find that the Curve Fitting equations are able to produce good results, although inter-target distance is not considered as an independent input parameter. This proves our claim that parameters other than group velocity, such as inter-target distance, do not alter the shape of the $\lambda(t)$ curve in a major way.

4.3 Error Analysis for Pincer Movement

Error analysis for the third military tactic, Pincer Movement, is done for the example scenario where three groups attempt to encircle the enemy. The corresponding UMMF traces and Curve Fitting equations are discussed extensively in section 3.3. As we had seen in this section, the two important input parameters which impact the behavior of node degree are initial inter-group distance (D_i) and group velocity (V). Since this scenario does not have a Motion Model, the error analysis is only done between the Curve Fitting Model and the UMMF model.

For the Pincer Movement scenario, T_{out} is specifically defined as the time for which the communication between the attacking groups is impaired. We do not consider the case where groups are able to communicate via two or more hops as a network outage. T_{out} is separately calculated using the UMMF model and the Curve Fitting model for the following three sets of input parameters:

i. Case 1:
   - Low group velocity (V)
   - Small initial inter-group distance (D_i)
   - Small group size

ii. Case 2:
   - High group velocity (V)
   - Small initial inter-group distance (D_i)
   - Large group size

iii. Case 3:
   - High group velocity (V)
   - Large initial inter-group distance (D_i)
     - Small group size

For the Curve Fitting model, outage time is calculated using equation 3-20, restated below:

$$\lambda(t) = [a + b*\ln(D_i) + c*\ln(V)]t + offset$$

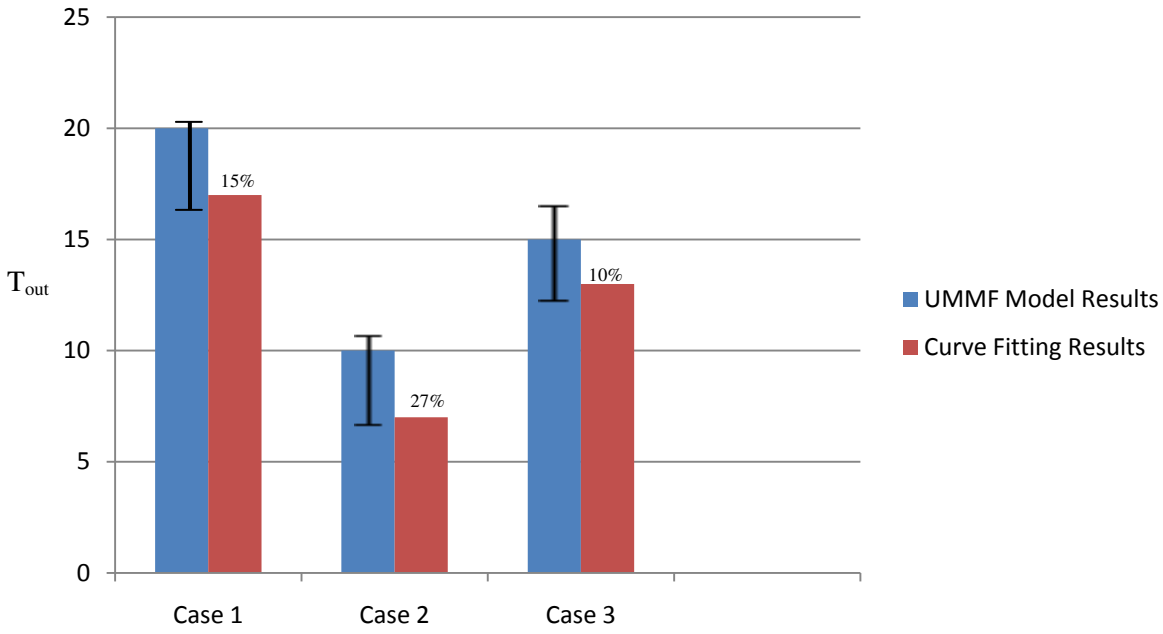


Figure 4-4. Bar chart representing error analysis results for Pincer Movement

The figure clearly shows that the Curve Fitting model is able to produce results of outage time which are within the range of values produced by UMMF, generated for ten runs of each simulation with a variety of seed values. The error percentage by which the Curve Fitting model deviates in each case is also represented in the figure. Across the three cases group size has been varied, which we had claimed, in section 3.3.2, to be a parameter that does not impact the shape of $\lambda(t)$. The fact that varying this parameter does not reduce the accuracy of the Curve Fitting model is proof of our claim.

Hence in this chapter we find that the Motion Model and Curve Fitting Model are effectively able to capture a wide variety of input parameters in group-based tactical scenarios and are able to produce results within a bounded error margin. Therefore, both these approaches can prove to be very powerful when evaluating protocols for military scenarios where synthetic mobility traces are unavailable.

Chapter 5

Data Replication in Tactical Networks

This chapter illustrates the application of tactical mobility models, developed in this thesis, in evaluating the performance of data replication algorithms in military networks. We study the performance of different data replication schemes under different tactical mobility scenarios. We highlight the deviation of results when regular mobility models, such as Random Waypoint and Random Point Group Mobility (RPGM), are used in place of tactical models.

In [22], Zhang et al. have looked at the performance of four prominent data replication schemes in well known mobility scenarios: Random Walk, Random Waypoint, Manhattan Model and RPGM. In [23], this work is extended to evaluate the performance of the same replication algorithms with three tactical mobility models that have been developed in this thesis, namely Repeated Traversal, Bounding Overwatch and Pincer Movement. Both technical reports study the same replication algorithms using the same evaluation metrics but at different granularity. In the first work, [22], replication is considered to be between individual nodes whereas in the second work, [23], replication occurs between groups.

In the following sections we begin with the motivation for this work, briefly explain the replication algorithms and evaluation metrics used. We then go on to our prime objective of characterizing the effects of tactical mobility on data replication schemes, comparing the output with results obtained with conventional mobility models and suggesting the best data replication scheme for each of the three military scenarios.

5.1 Motivation for Data Replication in Military Networks

The motivation for data replication in military networks is three fold. First, the topology of wireless ad hoc networks in military scenarios is highly dynamic. Links are made and broken often. Nodes which are within one hop range at a given time instant may well move out of contact in the near future. Therefore, data must be replicated efficiently to ensure data availability. Secondly, military scenarios have a highly hostile environment, wherein nodes are prone to getting destroyed or scattered. Hence links are unstable, network partitions are common and neighborhood sizes can change. If data is not replicated, information held by a node that gets destroyed would be lost forever. The last and most important factor is the criticality of information in tactical networks. Information carried by each node must be communicated for the mission to be successful. Often the data collected by a single node could be important enough to change the course of the mission or even abort it. Hence high data availability is of prime importance, followed closely by low data access delay.

In the following section we summarize the data replication algorithms that have been evaluated, along with the metrics of evaluation. Further details of the same can be found in [22].

5.2 Data Replication – Algorithms and Metrics

Data replication aims at increasing data availability and reducing access delay. This comes at the cost of storage space, since nodes in MANETs have limited storage and power capacities. It is thus important to choose data replication mechanisms that make the most judicious decision of which data items to replicate at which nodes, so as to best utilize the storage capacity of the network. In this work we assume that there is a common set of data items that all nodes in the network try to access, albeit with different access probabilities. Hence by the term

replication we mean that nodes try to store copies of different data items in their own local memory as well as within their reliable neighborhoods, so that there is higher data availability even when network partitions occur.

Since group mobility is common in tactical networks, we break the problem into the following two parts:

1. Intra-group Data Replication – This algorithm defines the best way to replicate data within a group.
2. Inter-group Data Replication – These techniques look at replicating data across groups.

5.2.1 Intra-Group Data Replication

In [23], Yang et al. present the novel Best Location Intra-Group Data Replication algorithm, which optimizes data access within a group by finding the 'best' node to store each data replica such that cost of data access as well as degree of data redundancy within the group is minimized. This is done by computing the following quantities:

a. A_i: Average access probability of a data item d_i

b. v_{kj}: expected access cost of data d_j at node n_k

v_{kj} is calculated as :

$$v_{kj} = \sum_{i=1}^{l} a_{ij} \times c_{ijk} \quad [5\text{-}1]$$

In the above equation a_{ij} is access probability of data d_j at node n_i and c_{ijk} is the access cost of data d_j between nodes n_i and n_k in a group. The algorithm then finds the best replication location within the group, for each data item starting with the one with highest average access probability. Hence for each data item d_j, a node n_i is found within the group such that access cost, v_{ij}, is minimum. At each step, if the memory of the best possible location n_i is full, the replica of d_j is

stored in the next best possible location (i.e. the node corresponding to the second lowest v_{ij}). This algorithm ensures the largest number of different data items in a group.

5.2.2 Inter-Group Data Replication

In [23], the following three data replication algorithms have been studied, at the granularity of groups.

a. **Greedy Data Replication:** The greedy replication algorithm is non-cooperative; wherein a group always replicates its most frequently accessed data item.

b. **Pairwise Cooperation Data Replication:** In the Pairing algorithm, each group coordinates with one reliable neighbor to decide which data to replicate. The decision is taken based on the access frequency of data items for both groups as well as the probability of group partition.

c. **Reliable Neighboring Data replication:** In this algorithm groups share data with multiple reliable neighboring groups. Replication decisions are made based on the access frequencies of all cooperating groups within a neighborhood and the stability of links between them.

5.2.3 Evaluation Metrics

The following are the metrics we use in [23] to evaluate the performance of the above mentioned replication algorithms.

Metrics of Data Access delay

a. Average Access delay (D)

This metric denotes the average number of hops from the query node to the nearest node containing the requested data.

$$D = \frac{\sum_{i=1}^{m} \sum_{j=1}^{R(i)} t_{ij}}{\sum_{i=1}^{m} R(i)} \quad [5\text{-}2]$$

where, t_{ij} is the access delay of the j^{th} request of node i; $R(i)$ is a function that returns the number of requests initiated by node i;

b. Temporal Distribution of Access Delay (D_k)

The following equation is used to define the access delay in the k^{th} time interval.

$$D_k = \frac{\sum_{i=1}^{m} \sum_{j=1}^{R_k(i)} t_{ij}}{\sum_{i=1}^{m} R_k(i)} \quad [5\text{-}3]$$

where, t_{ij} is the access delay of the j^{th} request of node i; $R_k(i)$ is a function that returns the number of requests initiated by node i in the k^{th} interval

c. Spatial Distribution of Access Delay ($D_{(hx,hy)}$)

The geographical distribution of access delay in the sub area (h_x,h_y) is defined by

$$D_{(h_x,h_y)} = \sum_{i=1}^{m} \sum_{j=1}^{R(i)} \mathcal{L}(t_{ij}\ , (h_x, h_y)) \quad [5\text{-}4]$$

Where, t_{ij} is the access delay of the j^{th} request of node i; $R(i)$ is a function that returns the number of requests initiated by node i; $\mathcal{L}(tij\ , (hx, hy))$ is a function that returns if the request takes place in the subarea (hx, hy).

Metrics of Data Availability

a. Average Data Availability (A)

The average availability is defined as the average probability that an in initiated query is served successfully. The following equation quantifies this metric.

$$A = \frac{\sum_{i=1}^{m} \sum_{j=1}^{R(i)} s_{ij}}{\sum_{i=1}^{m} R(i)} \quad [5\text{-}5]$$

Where, s_{ij} denotes whether the j^{th} request of node n_i is served (s_{ij}=1) or not (s_{ij}=0) and $R(i)$ is a function that returns the number of requests initiated by node i.

b. Temporal Distribution of Data Availability (A_k)

Similar to D_k, A_k represents the temporal distribution of data availability in the k_{th} time interval, defined specifically by the following equation.

$$A_k = \frac{\sum_{i=1}^{m} \sum_{j=1}^{R_k(i)} s_{ij}}{\sum_{i=1}^{m} R_k(i)} \quad [5\text{-}6]$$

Where, s_{ij} denotes whether the j^{th} request of node n_i is served (s_{ij}=1) or not (s_{ij}=0); $R_k(i)$ is a function that returns the number of requests initiated by node i in the k^{th} interval

c. Spatial Distribution of Data Availability ($A_{(h_x,h_y)}$)

The spatial distribution of data availability is given by

$$A_{(h_x,h_y)} = \sum_{i=1}^{m} \sum_{j=1}^{R(i)} \mathcal{L}(s_{ij}\ , (h_x, h_y)) \quad [5\text{-}7]$$

Where, s_{ij} denotes whether the j^{th} request of node n_i is served (s_{ij}=1) or not (s_{ij}=0); $R(i)$ is a function that returns the number of requests initiated by node i; $\mathcal{L}(sij\ , (hx, hy))$ is a function that returns if the request takes place in the subarea (hx, hy).

5.3 Results and Discussion

This section presents an in depth analysis of the evaluation results for the inter-group replication algorithms under group based tactical mobility models. We further compare the results obtained with our military scenarios, described in [23] to those obtained when conventional mobility models are used with the same replication schemes, presented in [22].

5.3.1 Average Delay and Average Availability

The results for average data access delay and average data availability are presented in Table 5-1 and 5-2 below.

Table 5-1. Average Data Access Delay

	RT-Mobility	FO-Mobility	PI-Mobility
Greedy	1.822	1.963	1.695
Pairing	1.826	2.019	1.704
Neighboring	1.467	1.974	1.727

Table 5-2. Average Data Availability

	RT-Mobility	FO-Mobility	PI-Mobility
Greedy	0.604	0.617	0.632
Pairing	0.826	0.823	0.517
Neighboring	0.603	0.893	0.463

Here we note that access delay is computed only for the queries that are served successfully and can hence be a misleading metric if looked at independently. Scenarios can often show low access delay with certain replication schemes because most of the requests are actually unsuccessful. Hence, access delay and data availability must always be studied together. Before analyzing these results it is important to note that, due to the criticality of information in military networks, data availability is a more important parameter than access delay for deciding the best replication algorithm for a given mobility pattern. We now discuss the results for each mobility scenario in detail.

Repeated Traversal

The pairing algorithm in this scenario gives the highest average data availability, although its average delay is comparable to that of the other schemes. Intuitively, this result is consistent with our understanding of Repeated Traversal, where there is close collaboration between the 'leading' and 'following' groups. Hence, it is mutually beneficial for each pair of leading and following groups to share information in order to ensure maximal data availability.

Bounding Overwatch

In Bounding Overwatch, groups try to maintain reliable neighborhoods during the mission, as was explained in section 3.2.1. Hence in this tactic, nodes belonging to a neighborhood should effectively be able to share information, such that maximal data items are stored within the neighborhood. The results in the above tables are consistent with our understanding, whereby the Reliable Neighboring algorithm gives the highest data availability with considerably low access delay. The Pairing algorithm emerges as the next best choice, but since neighborhoods in Bounding Overwatch can have more than two groups (unlike the case of Repeated Traversal), the Reliable Neighboring algorithm still exceeds its performance.

Pincer Movement

The results for Pincer Movement clearly show the Greedy replication algorithm to be the best choice as it gives the highest data availability with the lowest access delay. This is so because in this scenario each group has its own strategic target and moves independently to achieve it as effectively as possible. Groups do not collaborate with one another during their movement. Hence, it is best for each group to act greedily and replicate the data items which are of maximum importance to it. In this case, cooperative algorithms like Pairing and Neighboring replication work against the group's interests as they try to share data with other neighboring groups. When these groups move out of communication range, the replicated data items are lost.

Comparison with Conventional Mobility Models

We now compare the results in tables 5-1 and 5-2 with those presented in [22] for average access delay and average availability. In this report, the Greedy, Pairing and Neighborhood replication schemes work between individual nodes whereas in our work these algorithms replicate data between groups. Further the conclusions drawn from the Random Point Group Mobility (RPGM) model, which suggest that the Group-based replication scheme should be used between nodes showing group mobility, has already been incorporated in our work through the Intra-Group Replication algorithm (described in section 5.2.1). Hence, we study the performance of the Greedy, Pairing and Neighborhood replication schemes in [22], assuming groups in place of nodes, and compare the results with those derived from our tactical models.

We find that none of the 'general' models in [22] are able to yield results that can be applied to all military scenarios. For example, the conclusions drawn from Random Walk model, where the Pairing algorithm performs best, cannot be applied to the Bounding Overwatch and Pincer scenarios. Moreover, although the Pairing algorithm also performs the best in Repeated Traversal, Random Walk cannot be used to depict this scenario because the reasoning in these

two cases is drastically different. Pairing algorithm works well in Random Walk because nodes only vibrate in their positions in this model and are hence able to maintain a stable link with a neighbor. On the other hand, in Repeated Traversal, two groups work in collaboration while moving through a terrain and hence the Pairing algorithm works well here. Similarly, neither of Random Waypoint or Manhattan Model are able to yield conclusions, along with accurate reasoning, that can be applied to tactical scenarios.

5.3.2 Temporal Access Delay and Availability

Figure 5-1 and 5-2 schematically represent the temporal distribution of access delay and data availability for Greedy, Pairing and Neighboring algorithms in each military scenario.

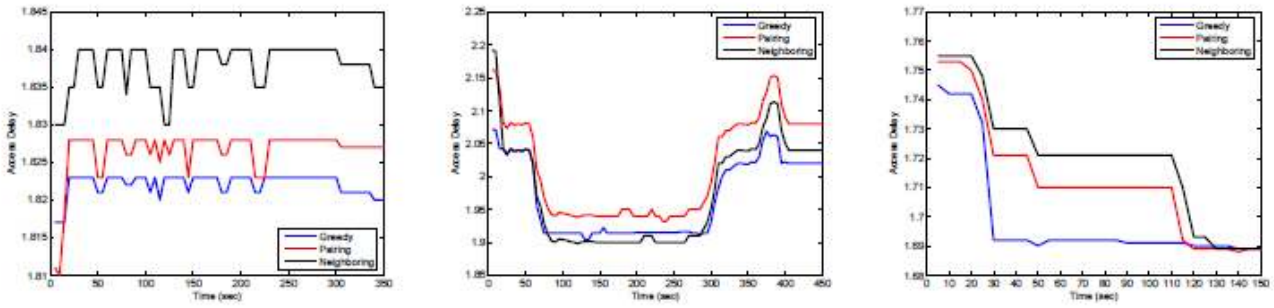
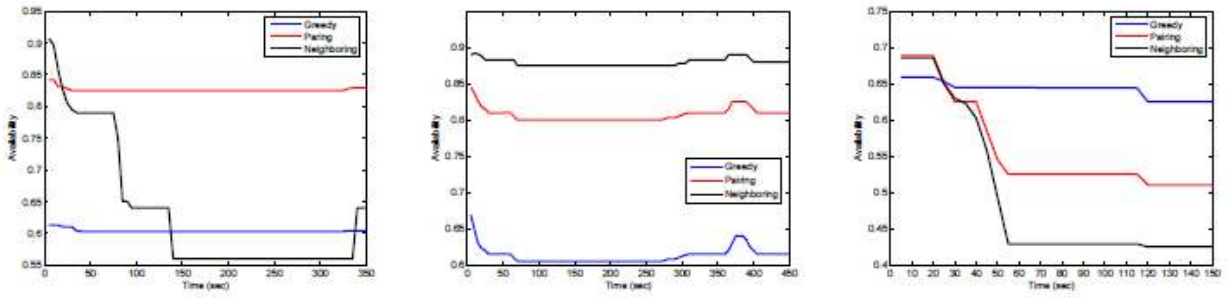


Figure 5-1. Temporal dependency of Data Access Delay (*y-axis*: Access Delay, *x-axis*: Time (sec))



(a)RT-Mobility (b)FO-Mobility (c)PI-Mobility

Figure 5-2. Temporal dependency of Data Availability (*y-axis*: Availability, *x-axis*: Time (sec))

Figures 5-1(a) and 5-2(a) clearly show that the Pairing algorithm outperforms the other two in Repeated Traversal, with a largely greater data availability than the other two schemes. The lower access delay curve for Neighboring algorithm is misleading in this case, because while calculating access delay only the successfully served queries are considered. The low data availability curve for the Neighboring algorithm shows that most requests are not successful with this scheme.

From 5-1(b) and 5-2(b) the Neighboring algorithm can be seen to have the highest curve for data availability for Bounding Overwatch along with a low access delay curve. The access delay curve for all three algorithms falls to a trough in the middle as groups are successfully able to replicate data within their reliable neighborhoods. Towards the end of the mission access delay rises again as groups come closer together while trying to capture the enemy. This results in more requests being successfully served, although number of hops needed to do so increases.

In figures 5-1(c) and 5-2(c) the Greedy algorithm clearly emerges as the best choice in Pincer Movement, with the highest curve for data availability and the lowest curve for access delay.

For all three tactical scenarios, not only do the temporal dependency curves help in making a more accurate decision regarding data replication, they can be analyzed more closely to reveal the instances when network partitions and link failures can be expected. Hence, we argue that the closer the underlying mobility model is to the real life tactical movement pattern, the more informed the evaluation of protocols can be.

5.3.3 Spatial Access Delay and Availability

Figures 5-3 and 5-4 schematically represent the spatial distribution of access delay and data availability for Greedy, Pairing and Neighboring algorithms in the Repeated Traversal scenario.

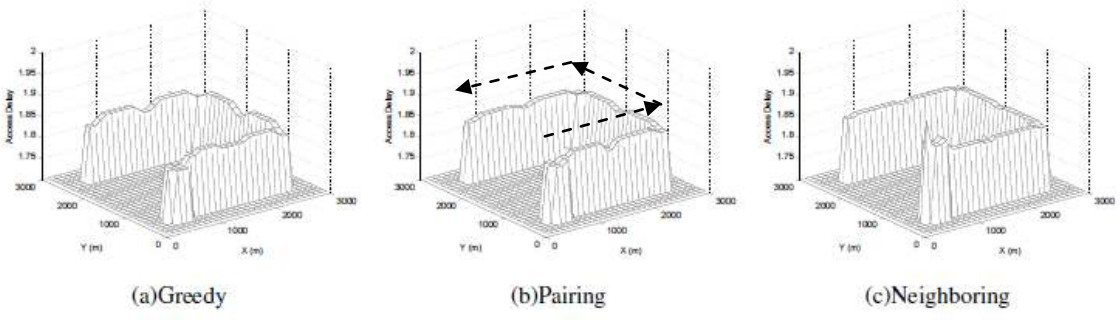


Figure 5-3. Spatial distribution of data access delay in Repeated Traversal

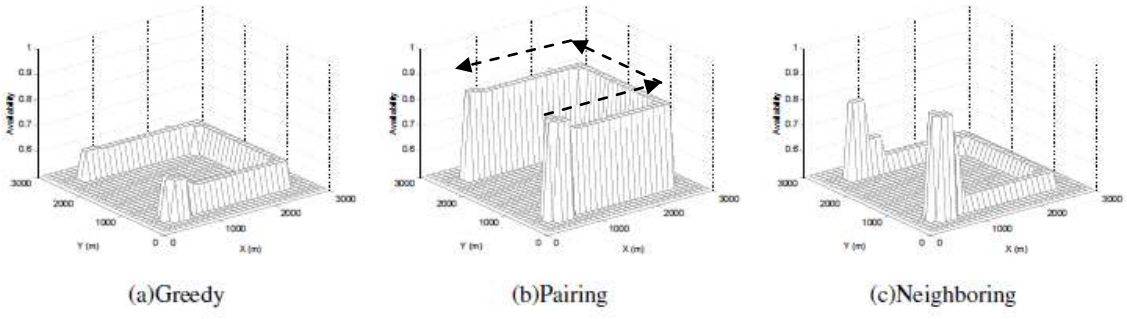


Figure 5-4. Spatial distribution of data availability in Repeated Traversal

The arrows on graphs 5-3(b) and 5-4(b) show the path taken by the nodes in this scenario. From the above figures we find that access delay is comparable for the three schemes but using the Pairing algorithm, information is shared such that there is high data availability throughout the path being traversed. The neighboring algorithm does well in the beginning, when all groups are together at the start point, but its performance degrades subsequently as pairs of groups start to advance and network partitions occur. The greedy algorithm does moderately well, but is clearly out performed by the Pairing scheme.

Figures 5-5 and 5-6 schematically represent the spatial distribution of access delay and data availability for Greedy, Pairing and Neighboring algorithms in the Bounding Overwatch scenario.

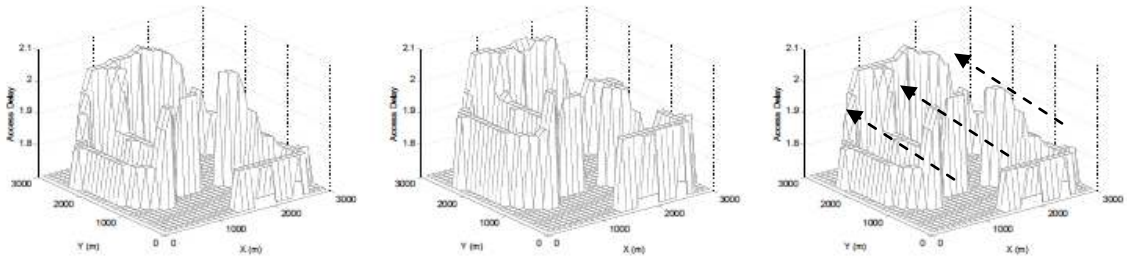


Figure 5-5. Spatial distribution of data access delay in Bounding Overwatch

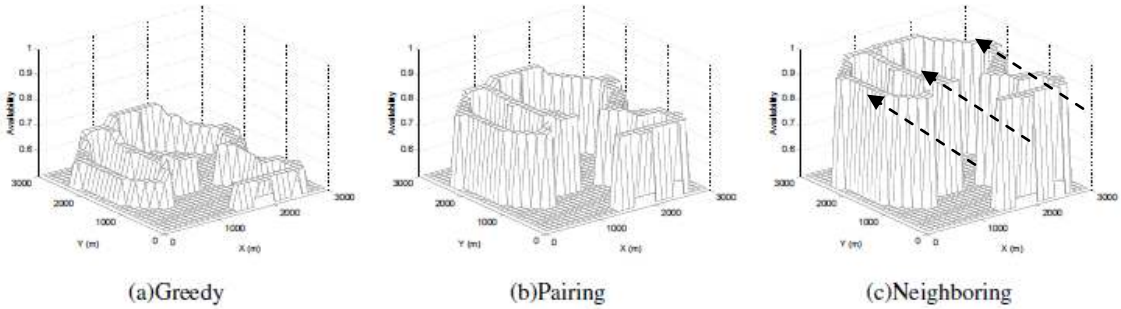


Figure 5-6. Spatial distribution of data availability in Bounding Overwatch

The arrows on graphs 5-5(c) and 5-6(c) show the path taken by nodes in this scenario. In the above figures, Greedy and Neighboring algorithms show comparably low access delays but figures 5-5(c) and 5-6(c) show that the Neighboring scheme results in high data availability throughout the route taken by nodes in this scenario. As expected the access delay is lower in the beginning when all nodes are together at the initial target and increases as the mission progresses. This increase is visible in figure 5-5(c) along the increasing y-axis. The pairing algorithm performs the next best followed by the greedy mechanism, as we had also concluded from the temporal distribution graphs.

Figures 5-7 and 5-8 schematically represent the spatial distribution of access delay and data availability for the three replication algorithms in the Pincer Movement scenario.

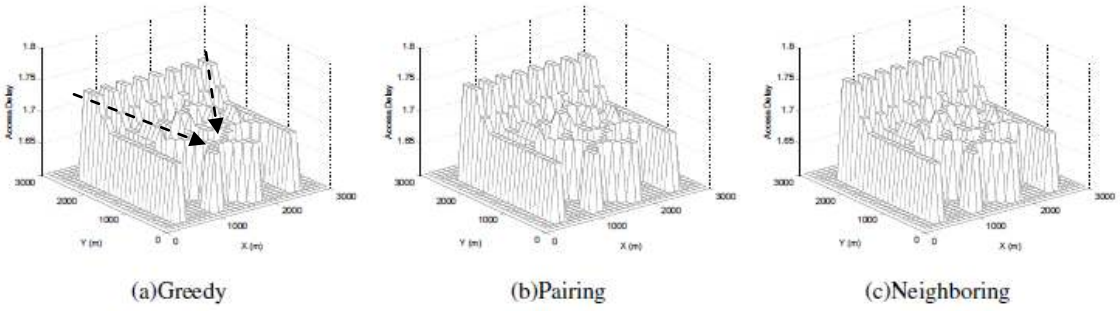


Figure 5-7. Spatial distribution of data access delay in Pincer Movement.

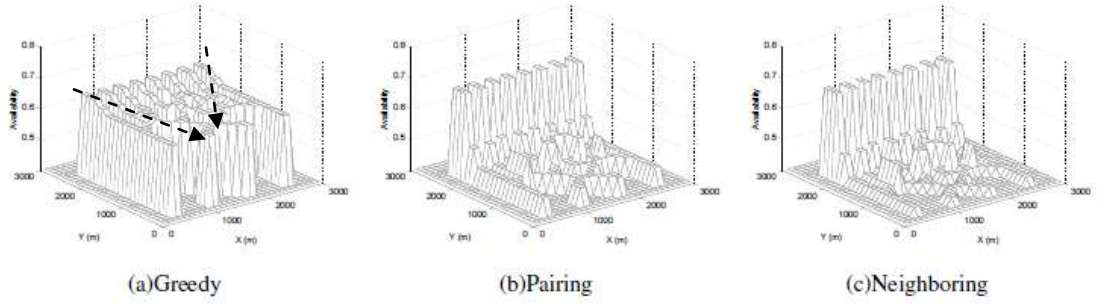


Figure 5-8. Spatial distribution of data availability in Pincer Movement.

From the above figures, the greedy algorithm ensures much higher data availability than the other replication schemes, throughout the simulation area. For Pincer also, the spatial access delay is lower in the beginning of the mission, when groups are closer to one another, and increases as groups move apart as they try to encircle the enemy. The attempt to cooperatively share information in the Pairing and Neighboring algorithm degrades the performance of these schemes, resulting in very low data availability for the most part of the mission.

Comparison with Conventional Mobility Models

We now compare the above results for spatial distribution of access delay and availability with those obtained with conventional models for the same metrics, in [22]. In the Random Walk model access delay and availability are independent of location. There is a uniform distribution of both metrics throughout the simulation area. In the Random Waypoint model there is higher data availability and consequently lower access delay at the centre of the region suggesting that nodes tend to aggregate at the center. But this is not the case in military scenarios, as we have seen from the above figures. With RPGM, the spatial distribution graphs are similar to those of Random Walk and hence although tactical mission show group mobility, RPGM is not a good indicator of the mobility pattern. These findings corroborate the fact that conventional models like Random Walk and RWP deviate largely from the true mobility patterns of tactical networks and therefore are not a good choice for representing military mobility.

In this chapter we find that the results of performance evaluation obtained with conventional mobility models are incorrect and misleading when applied to military scenarios.

None of the conventional mobility models studied in this thesis are able to succinctly capture the nuances of tactical mobility. This clearly demonstrates the importance of developing and using tactical mobility models while evaluating the performance of applications for military missions.

Chapter 6

Conclusion

In this thesis we construct tactical mobility models which closely capture the nuances of common military movement. These models can hence be used effectively to evaluate the performance of network protocols for military applications.

The thesis begins with a study of some common mobility models that have been developed in the past, such as the Random Walk, Random Waypoint and Reference Point Group Mobility models. Although these models have often been used to represent mobility for several MANET applications due to their ease of use, a deeper look into their characteristics reveals that they cannot be applied to military scenarios. Broadly stated, the movement patterns in military scenarios are governed by predefined tactics, objectives and code of conduct. Hence, military mobility is found to be very structured and often epoch-based, whereas the mobility depicted by most conventional models is more randomized and simplified. Hence, this study demonstrated the need for the development of a separate set of models that are able to capture tactical mobility.

The modeling of tactical mobility was done in three stages. First, we identified four common military tactics performed in modern day warfare, namely Repeated Traversal, Bounding Overwatch, Pincer Movement and Patrolling. Each scenario was simulated using the Universal Mobility Modeling Framework (UMMF) to generate traces for network parameters, like node degree, as a function of time, for a wide variety of input parameters and seed values. Analyzing these traces allowed us to identify the input parameters which significantly affected the behavior of node degree in each scenario. Further, the traces exhibited some key mobility properties of each tactic. The Repeated Traversal and Bounding Overwatch tactics were found to be epoch-based with node degree showing a regular behavior in each epoch. Moreover, in

Repeated Traversal high link stability was found between a pair of 'leading' and 'following' groups, whereas in Bounding Overwatch, groups tried to maintain reliable neighborhoods throughout the tactic. Pincer Movement was found to be less cooperative, with groups mainly aiming at fulfilling their individual objectives. Lastly, Patrolling was found to differ from the other three tactics since it did not require group mobility. Rather in Patrolling individual nodes were found to gather around targets of higher importance.

In the second stage of modeling, we developed a Motion Model that graphically explains the movement and provides equations for computing mobility parameters. The Motion Model for Repeated Traversal and Bounding Overwatch scenarios uses fundamental laws of physics to calculate inter-group distance as a function of time as well as other input parameters such as group velocity. The Motion Model for Patrolling comprises of an algorithm that each node in the scenario implements, which dictates its movement pattern. Since the mobility in Pincer Movement is more strategic rather than geographic, we have not attempted to describe it using the motion modeling approach.

The third stage of modeling was done using the technique of curve fitting. We performed time series analysis of node degree, with respect to the input parameters of interest which had been identified from the UMMF simulations. As a result, we were able to formulate tractable mathematical equations that describe the dependence of node degree on input parameters such as group velocity, pause time and inter-group distance. For Repeated Traversal and Bounding Overwatch these equations apply to a specific epoch. In the case of Patrolling, the mobility is determined by the objective of the patrol. Hence, the movement has been captured more as an optimization problem, which is to be solved using heuristics rather than definite equations. Therefore, we have not attempted to do curve fitting for this tactic.

Having generated the UMMF traces, Motion Model and Curve Fitting equations for tactical scenarios the next step was to analyze the error between these three approaches. This was

done by identifying an output parameter, namely outage time, which could easily be computed using all three approaches. Comparing the values computed from each approach we found that both the Motion Model and the Curve Fitting Model yielded results which were within a bounded error margin of the results derived from UMMF. This leads to the conclusion that the Motion Model and the Curve Fitting Model can be used to capture mobility parameters for tactical scenarios when mobility traces are not available.

In the final part of our work, we attempt to demonstrate the necessity and usability of tactical mobility models in protocol evaluation for military applications. Since effective sharing of information is a prime objective of Network-centric warfare, implementation of efficient data replication schemes is of high importance in military communication networks. Hence, in this work we evaluate the performance of common data replication schemes using both conventional mobility models as well as the tactical models developed in this thesis. Comparing the results for metrics such as access delay and data availability, it was found that the conclusions drawn from conventional models are not applicable to military scenarios and are often misleading. This is so because the conventional models over-simplify mobility and are not able to capture key properties of tactical scenarios, such as link stability between reliable neighbors, higher node density around specific targets, and so on. Therefore, through this thesis we are clearly able to establish the need to model tactical mobility separately and use these models to capture the mobility for military applications.

References

[1] D. S. Alberts, J. J. Garstka, F. P. Stein, NETWORK CENTRIC WARFARE:Developing and Leveraging Information Superiority, in DoD C4ISR Cooperative Research Program, 2nd Edition (Revised), August 1999.

[2] M. M. Zonoozi and P. Dassanayake, User mobility modeling and characterization of mobility patterns, IEEE Journal on Selected Areas in Communication, Vol. 15, No. 7, pp. 1239-1252, September 1997.

[3] David B. Johnson, David A. Maltz, "Dynamic Source Routing in Adhoc Networks", In Mobile Computing, Vol, 353, pp. 153-181, 1996.

[4] J. Broch, D. A. Maltz, D. B. Johnson, Y.-C. Hu, and J. Jetcheva, A performance comparison of multi-hop wireless ad hoc network routing protocols, in Proceedings of the Fourth Annual ACM/IEEE International Conference on Mobile Computing and Networking(Mobicom98), ACM, October 1998.

[5] B. Liang, Z. J. Haas, Predictive Distance-Based Mobility Management for PCS Networks, in Proceedings of IEEE Information Communications Conference (INFOCOM 1999), Apr. 1999.

[6] Y.-C. Hu and D. B. Johnson. Caching Strategies in On-Demand Routing Protocols for Wireless Ad Hoc Networks, in Proceedings of the Sixth Annual International Conference on Mobile Computing and Networking (MobiCom 2000), ACM, Boston, MA, August 2000.

[7] X. Hong, M. Gerla, G. Pei, and C.-C. Chiang, A group mobility model for ad hoc wireless networks, in Proc. ACM Intern. Workshop on Modeling, Analysis, and Simulation of Wireless and Mobile Systems (MSWiM), August 1999.

[8] F. Bai, A. Helmy, A Survey of Mobility Models in Wireless Ad Hoc Networks, University of Southern California, USA.

[9] F. Bai, N. Sadagopan, and A. Helmy, Important: a framework to systematically analyze the impact of mobility on performance of routing protocols for ad hoc networks, in Proceedings of IEEE Information Communications Conference (INFOCOM 2003), San Francisco, Apr. 2003.

[10] T. Camp, J. Boleng, and V. Davies, A Survey of Mobility Models for Ad Hoc Network Research, in Wireless Communication and Mobile Computing (WCMC): Special issue on Mobile Ad Hoc Networking Research, Trends and Applications, vol. 2, no. 5, pp. 483- 502, 2002.

[11] C. Bettstetter. Mobility Modeling in Wireless Networks: Categorization, Smooth Movement, and Border Effects, in ACM Mobile Computing and Communications Review, vol. 5, no. 3, pp. 55-67, July 2001.

[12] A. Medina, G. Gursun, P. Basu, I. Matta, On the Universal Generation of Mobility Models, in Proc. IEEE/ACM MASCOTS 2010, Miami Beach, FL, August 2010.

[13] Army Field Manuals: http://armypubs.army.mil/doctrine/active_fm.html

[14]Field Manual 3-21.8, Chapter 3, Tactical Movement, https://rdl.train.army.mil/soldierPortal/atia/adlsc/view/public/23583-1/FM/3-21.8/chap3.htm

[15] R. Johnson and M. Whitby, J. France, How to Win on the Battlefield: The 25 Key Tactics of All Time, Thames and Hudson, 2010

[16] R.J. Freund, W.J. Wilson, and P. Sa, Regression Analysis: Statistical Modeling of a Response Variable, second ed. Academic Press, 2006

[17] A. Sen, M. Srivastava, Regression Analysis: Theory, Methods, and Applications, Springer, New York, 1990.

[18] Online Curve Fitting and Surface Fitting Web Site, www. zunzun.com

[19] Å. Björck Numerical Methods for Least Squares Problems, ISBN 978-0-898713-60-2, 1996

[20] Applegate, D. L.; Bixby, R. M.; Chvátal, V.; Cook, W. J. , The Traveling Salesman Problem, ISBN 0691129932, 2006

[21] Cormen, Leiserson, Rivest, and Stein, Introduction to Algorithms, Chapter 16 "Greedy Algorithms", 2001.

[22] Y. Zhang, G. Cao, T. La Porta, On Quantifying the Effects of Mobility on Data Replication in Mobile Ad Hoc Networks, Percom2011 (*under submission)*

[23] Y. Zhang, S. Ray, G. Cao, T. La Porta, Data Replication in Mobile Tactical Networks, Milcom 2011 (*under submission)*