

IMPLEMENTATION AND COMPARISON OF MOBILITY MODELS IN NS-2

A THESIS SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF

**Bachelor of Technology
In
Computer Science and Engineering**

By

Sukumar Panda

Rahul Mohapatra



**Department of Computer Science and Engineering
National Institute of Technology
Rourkela**

2009

May 2009

IMPLEMENTATION AND COMPARISON OF MOBILITY MODELS IN NS-2

A THESIS SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF

**Bachelor of Technology
In
Computer Science and Engineering**

By:

**Sukumar Panda
Roll No.: 10506009**

**Rahul Mohapatra
Roll No.: 10506051**

Under The Guidance of:

**Prof Santanu Kumar Rath
Prof Suchismita Chinara**



Department of Computer Science and Engineering

National Institute of Technology, Rourkela
May, 2009



**National Institute of Technology
Rourkela**

CERTIFICATE

This is to certify that the thesis entitled, “IMPLEMENTATION AND COMPARISON OF MOBILITY MODELS IN NS-2” submitted by Sukumar Panda and Rahul Mohapatra in partial fulfillment of the requirements for the award of Bachelor of Technology Degree in Computer Science and Engineering at the National Institute of Technology, Rourkela (Deemed University) is an authentic work carried out by them under my supervision and guidance.

To the best of my knowledge, the matter embodied in the thesis has not been submitted to any other university / institute for the award of any Degree or Diploma.

Date:

Prof S.K.Rath
Dept. of Computer Science and Engineering
National Institute of Technology, Rourkela
Rourkela - 769008

Contents

	Page No.
Abstract	5
List of Figures	6
Chapter 1. Introduction	7
Chapter 2. Mobility Models	9
2.1 Random Walk Mobility Model	10
2.1.1 Random Walk with Wrapping	11
2.1.2 Random Walk with Reflection	12
2.1.3 2 D Random Walk Mobility Model	13
2.1.4 Model for square cells	15
2.1.5 Advantages of RW Mobility Model	21
2.1.6 Disadvantages of RW Mobility Model	21
2.2 Random Waypoint Mobility Model	
2.2.1 A General Random Waypoint Model	22
2.2.2 Advantages of RWP Mobility Model	23
2.2.3 Disadvantages of RWP Mobility Model	23
2.3 Random Direction Mobility Model	
2.3.1 A General Random Direction Mobility Model	24
2.3.2 Advantages of RD Mobility Model	25
2.3.3 Disadvantages of RD Mobility Model	25
2.4 Limitation of Mobility Models	26
Chapter 3 Methodology	
3.1 About NS 2	29
3.2 Parameters	30
Chapter 4 Results	33
Chapter 5 Conclusion	38
References	40

ABSTRACT

In the performance evaluation of a protocol for an ad hoc network, the protocol should be tested under realistic conditions including, but not limited to, a sensible transmission range, limited buffer space for the storage of messages, representative data traffic models, and realistic movements of the mobile users (i.e., a mobility model). Simulation is universally considered the most effective method of designing and evaluating new network protocols. When developing protocols for mobile networking, the chosen mobility model is one of the key determinants in the success of an accurate simulation. The main role of a mobility model is to mimic the movement behaviors of actual users. Given the critical role of the mobility model in supporting realistic and accurate protocol simulations, its correct design and selection is essential. We have described 3 mobility models that represent mobile nodes whose movements are independent of each other (i.e., entity mobility models). Lastly, we present simulation results that illustrate the importance of choosing a mobility model in the simulation of an ad hoc network protocol. Specifically, we illustrate how the performance results of an ad hoc network protocol drastically change as a result of changing the mobility model simulated.

List of Figures

Figure No.	Figure Name	Page no.
Fig 1	Traveling pattern of an MN using the 2-D Random Walk Mobility Model	10
Fig 2	Random Walk with wrapping	12
Fig 3	Random Walk with reflection	13
Fig 4	Analysis of Square Model	20
Fig 5	Analysis of Location Area	20
Fig 6	State transition diagram for a 5 * 5 square cell model	21
Fig 7	Traveling pattern of an MN using the 2-D Random Waypoint Mobility Model	23
Fig 8	Traveling pattern of an MN using the Random Direction Mobility Model	24
Fig 9	average number of hops vs. speed	34
Fig 10	average number of hops vs. no. of nodes	35
Fig 11	throughput ratio vs. speed	36

Chapter 1

INTRODUCTION

INTRODUCTION

In recent years, the deployment and importance of wireless networks has grown rapidly. Mobility plays a key role in this regard, and has driven the development of many new services, such as VoIP and interactive multimedia. In support of these developments, new mobile and wireless protocols are proposed by researchers to support novel applications and to improve the overall performance of wireless networking amongst mobile nodes. Obviously, developers must evaluate the impact of their protocols on network characteristics and analyze the behaviors of the network under the proposed conditions. Simulations of wireless networks employ several components critical to the accuracy of the simulations, one of the most important being the choice of mobility model, which creates the movement patterns of mobile nodes that forms the varying topology of the network. A typical mobility model first places the mobile nodes in their initial locations and defines the way that the nodes move within the network. Indeed, the mobility model emulates the real user movement through the inclusion of the critical movement factors such as direction, speed, destination, and the movement histories of the same, or similar, users. [2][3]

The most important characteristic of a mobility model is the degree of realism with respect to the movement of users in real life. More realistic models enable more accurate simulation and evaluation of network parameters. In addition, because there exists no single, comprehensive, mobility model the incorrect selection of an inappropriate model, not mimicking the movement patterns expected in the real life environment under consideration, leads to incorrect observations and results. [1]

The rest of the thesis is organized as follows. Chapter 2 briefly reviews the mobility models. Chapter 3 describes the simulation environment and the methodologies used. Chapter 4 illustrates the simulation results and chapter 5 presents the conclusion.

Chapter 2

MOBILITY MODELS

2.1 RANDOM WALK MOBILITY MODEL

The random walk mobility model is the simplest mobility model, generating completely random movement patterns. It was designed for simulations in which the movement patterns of mobile nodes are completely unpredictable. Since many entities in nature move in extremely unpredictable ways, the Random Walk Mobility Model was developed to mimic this erratic movement. In this mobility model, an MN moves from its current location to a new location by randomly choosing a direction and speed in which to travel. In this model a mobile node is initially placed in a random location in the simulation area, and then moved in a randomly chosen direction between $[0, 2\pi]$ at a random speed between $[SpeedMin, SpeedMax]$. [3]

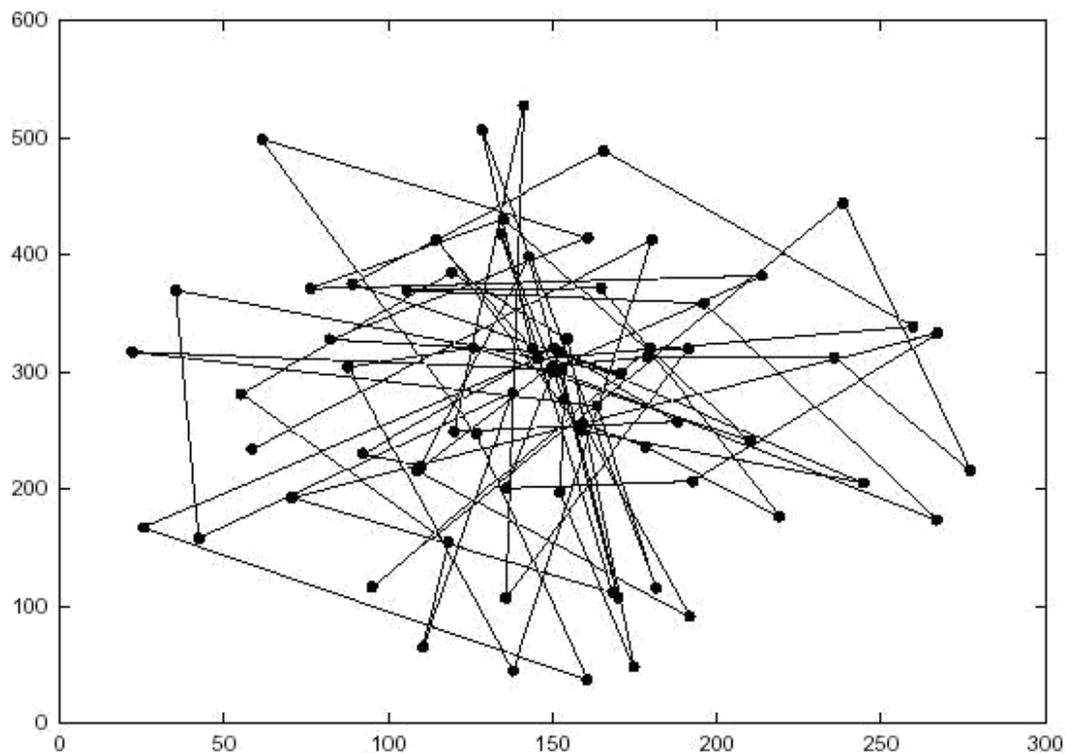


Figure 1: Traveling pattern of an MN using the 2-D Random Walk Mobility Model

Each movement in the Random Walk Mobility Model occurs in either a constant time interval t or a constant distance traveled d , at the end of which a new direction and speed are calculated and this process is repeated a predetermined number of times. Figure 1 shows the result of a single node executing the random walk mobility model with a constant travel time. Many derivatives of the Random Walk Mobility Model have been developed including the 1-D, 2-D, 3-D, and d-D walks. In 1921, Polya proved that a random walk on a one or two-dimensional surface returns to the origin with complete certainty, i.e., a probability of 1.0 [5]. This characteristic ensures that the random walk represents a mobility model that tests the movements of entities around their starting points, without worry of the entities wandering away never to return.[3][1]

Two variations of the random walk mobility model were proposed by Nain et al., [6] to address the problem experienced when mobile nodes reach the boundary of their simulation area namely:

1. Random Walk with wrapping
2. Random Walk with reflection

2.1.1 Random Walk with wrapping

In the random walk with wrapping approach [2], when a mobile node reaches an edge, it wraps to the opposite edge and continues its movement with the same direction and speed. This is illustrated in Fig 2

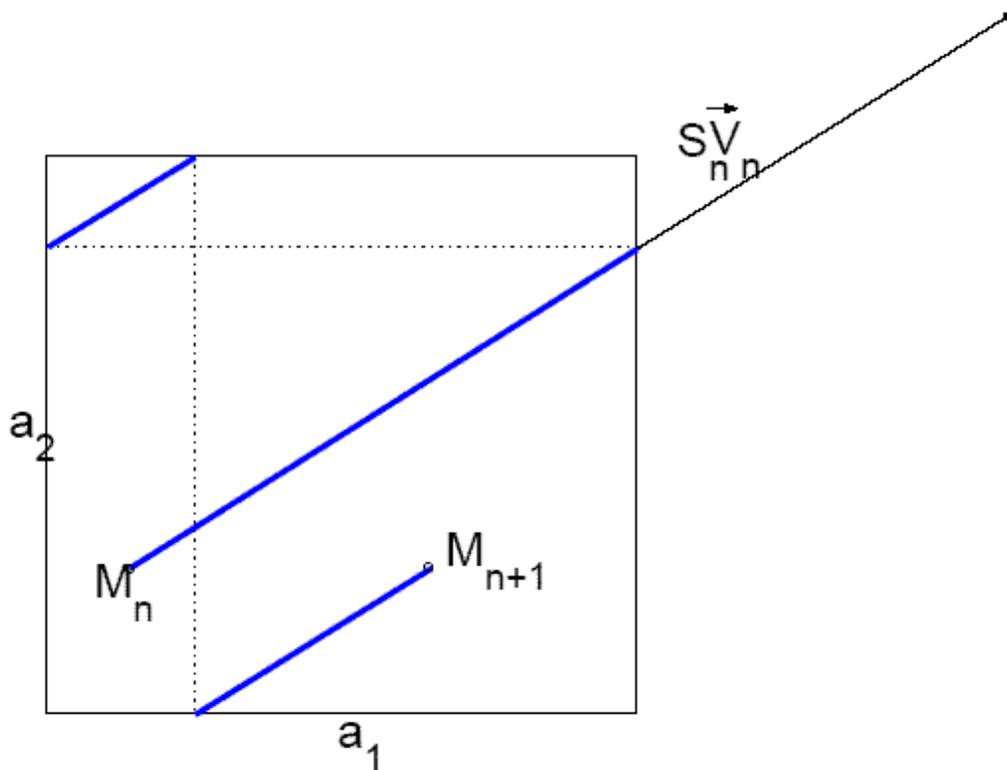


Figure 2 : Random Walk with wrapping

2.1.2 Random Walk with reflection

In a further approach, random walk with reflection [2], when a mobile node reaches any edge of the simulation area, the node changes its angle of movement and its velocity remains constant. The approach employing reflection clearly generates more accurate movement patterns, simply because real life mobile nodes are more likely to reflect their movement when reaching an obstacle. This is illustrated by Fig 3

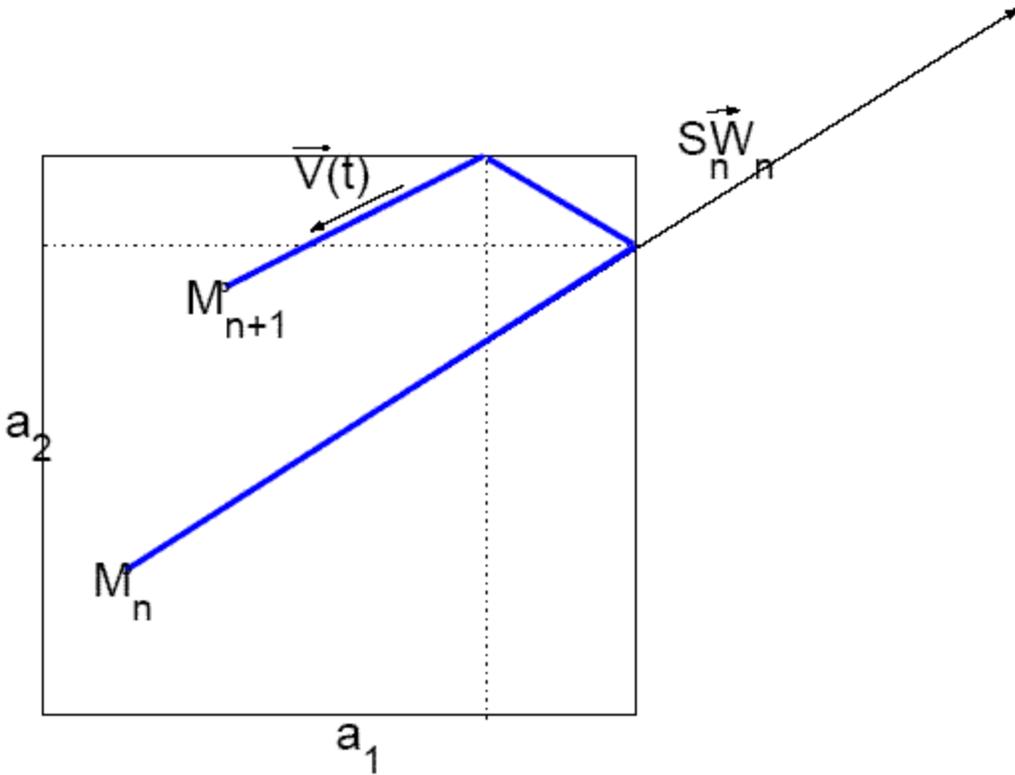


Figure 3 : Random Walk with reflection

2.1.3 A 2D Random Walk Mobility Model

Mobility of users is a major advantage of wireless over fixed telecommunications systems. The signaling traffic and database processing to support mobility of users are always the key concerns in the design and performance of wireless networks. Mobility models play a key role in studying different mobility management features such as registration, paging, handoff, and database approaches. A mobility model with minimum assumptions and which is simple to analyze will be very useful under such circumstances. In most wireless network performance studies the cell is assumed to be either hexagonal or square in shape, though in real life cell shapes may be highly irregular. [1][4]

The highlights of the model are its simplicity, minimal assumptions, and adaptability to conduct both location crossing rate and dwell time studies using the same model with slight modifications for both square and hexagonal cells. The model uses a set of aggregate states to trace user movement within one location area and then uses a set of special (asterisk) states when crossing the boundary of the location area. Due to the wrap-around technique, the movement of the mobile is modeled to again enter the original states from the special states, once the mobile starts moving within the new location area. Average number of location area crossing rates or updates made by the user is obtained by solving for the regular Markov chain. [4]

A slightly modified model with absorbing states was used to derive the dwell time. This is the first model of its kind that can be used for studying area-crossing rates and dwell times using a simple model. Overlap location area strategies are commonly used concepts in cellular networks. The fluid flow model and random walk model are the two main types of mobility model that have been applied in location management studies. The fluid flow model can derive the average rate of boundary crossings per unit time out of a given area, but it is difficult to apply the model to the modern per-user based location area strategies or to study overlapped area strategy performance. Most random walk models are designed for dynamic location area strategy or for deriving the dwell time. [2]

The simple 2-D random walk model is based on the properties of the regular and absorbing Markov chain. The regular model is used for computing the location update rates. The same model with absorbing states is used to derive the dwell time in an area. The number of computational states in the model is very less, making its analysis is easy. A 'states wrap-around' feature has been introduced in the model, which facilitates estimating location

update rate using simple equations. The model can be adapted to both square cell and hexagonal cell structures. Because of the simplicity in calculations, the model can be extended to study overlapped location areas strategies, which has hitherto been difficult to handle. These are features, which make the model ideally suited to mobility management studies in cellular networks, though with limitation that the cells will be either square or hexagonal. [4]

Users in cells belonging to a Location Area (LA) have identical movement pattern within and across LAs. Such cells can be assigned to a single state in the Markov chain using the lumped process property. This property has been applied here to achieve reduced computational states in the model. A state which lumps a number of cells with identical movement pattern is called an 'aggregate state'.

2.1.4 Model for square cells

Fig. 1A shows an LA configuration for $n \times n$ square cells where $n = 5$. The bold lines indicate the boundary of the LA. Cells in the adjacent LAs are shown outside the thick boundary lines. There are 25 cells in the LA and they have been numbered, identifying each cell with its row and column position. It is assumed that the mobile user can make movement in 4 directions only with an equal probability of $\frac{1}{4}$ in each direction as shown in Fig. 4B. In Fig 4A arrows show the possible user movement across cells within one LA and across LAs. In some of the cells, with reference to the LA, the mobile user will have identical movement patterns to other cells within the same LA and to cells in adjacent

LAs. For e.g. from cells numbered 11, 15, 55 and 51 the mobile user movement pattern and transition probabilities will be identical but rotated by 90 degrees. The rotation of the movement pattern will not affect the studies to be conducted using Markov chains. Hence these four cells can be grouped into one aggregate state in the Markov chain model. This property of lumping the states was carried out to reduce the total number of computational states in the mathematical analysis. An algorithm to perform state aggregation is given below (Figs 5A and 5B are provided for the purpose of explaining the aggregation process) [4]

Algorithm

1. Number the cells belonging to an LA as 'rc' where r is the row number and c is the column number as shown in Fig 4A. The numbering scheme has been applied to the cells in the adjacent LAs also i.e. cells just outside the bold lines.

2. As shown as in Fig. 5A assign the cells to different loops. In this case there are 3 loops. Let l = number of loops. Then l=1 is the outer loop, l=2 is the next loop and l=3 is the innermost loop. Cell 33 belongs to loop 3. Cells numbered 22, 23, 24, 34, 44, 43, 42, 32 belong to loop 2. The rest of the cells in the LA belong to loop 1.

3. Start with n =5, l=1

Until (n =l)

Repeat { x= l; y = n;

Until(x > y)

x = {Src U Scr}

Repeat { set $S_l^x = \{Src \cup Scr\}$

Where $r = l$ for $c = x, y$ and $r = n$ for $c = x, y$

$x = x + 1; y = y - 1;$

}

$n = n - 1, l = l + 1;$

}

S_l^x - is the aggregate state obtained as a set of the cells Src and Scr where, Src is the cell numbered 'rc' and Scr is the cell numbered 'cr'.

After the aggregation process the following state aggregation is obtained

$$\begin{aligned}
 S_1^1 &= \{S_{11} \ S_{15} \ S_{51} \ S_{55} \}, \\
 S_2^1 &= \{S_{21} \ S_{25} \ S_{41} \ S_{45} \ S_{12} \ S_{52} \ S_{14} \ S_{54} \}, \\
 S_3^1 &= \{S_{31} \ S_{35} \ S_{13} \ S_{53} \}, \\
 S_1^2 &= \{S_{22} \ S_{24} \ S_{42} \ S_{44} \} , \\
 S_2^2 &= \{S_{32} \ S_{34} \ S_{23} \ S_{43} \} , \\
 S_1^3 &= \{S_{33} \} .
 \end{aligned}$$

For ease-of-use, the aggregate states were assigned numbers as shown below.

$$S_1^1 \rightarrow 1, S_2^1 \rightarrow 2, S_3^1 \rightarrow 3, S_1^2 \rightarrow 4, S_2^2 \rightarrow 5, S_1^3 \rightarrow 6,$$

We now have 6 aggregate states. The LA is redrawn in Fig. 5B with the cells numbered with their newly assigned aggregate state numbers. In Fig. 5B aggregate states 1, 2 and 3 are in the

boundary of the LA and are called the 'boundary' states. We further define 'asterisk boundary' states (or 'star states') i.e. 1^* , 2^* and 3^* , which are the boundary states in the adjacent LAs. Let LA0 be the LA under consideration then in Fig 5B, LA1, LA2, LA3 and LA4 are the adjacent LAs. From LA0 a user can move only into any one of these adjacent LAs. [4]

Using the direction and movement probabilities given in Fig. 4B, the state transition diagram for the regular Markov chain based on the aggregate states can be derived as shown in Fig 6. In this approach the entire movement of the mobile user across different cells and across LAs is modeled. The 'asterisk boundary' states and the wrap around mechanisms, achieve this freedom-of-movement feature. Movement into an 'asterisk' boundary state indicates a location area crossing and can be used to study the location area crossing rates or location updates.

As long as the user moves within cells in a location area, he is in one of the main aggregate states i.e. 1, 2, 3, 4, 5 or 6. His movement is accordingly traced by the transitions shown. Transition probabilities are marked beside the transition. The dotted transitions to the asterisk states indicate the mobile user's transition to a boundary state in an adjacent location area and the probabilities of such transitions are given besides these dotted transitions. These transitions accordingly model the mobile user's movement probability from the main location area cells to the adjacent location area cells. While in an asterisk states the user can move across to other asterisk states. For e.g. (in Fig 5B) the user move

from 1* in LA3 to 1* in LA7 or from 2* in the adjacent LA3 back to 2 in LA0 (2 in LA0 now would be 2* from the mobile user perspective as it is in an adjacent LA). [4]

In Fig. 6, this movement is modeled by the dotted transitions to itself in the asterisk states. Again, looking at the user movement from state 1* in LA3 (Fig 5B), the user has a probability of $\frac{1}{4}$ to move into state 1* in LA7, and probability of a $\frac{1}{4}$ to move into state 1 in LA0 (which is now state 1* in the adjacent LA from the user perspective) – hence a transition probability of $\frac{1}{2}$ to itself in state 1* is shown in the Fig. 6. [4]

When the user starts moving in the new LA, for e.g. when he moves from state 1* in LA3 to 2* in LA3 or from 2* to 1* in LA3 the model wraps back to the normal states – indicating that the user has now started moving in cells belonging to the new LA. This is shown by the dotted transitions from the asterisk states to the normal states in Fig 6. This novel wrap-around feature has subsequently led to the possibility of studying the location area crossings using simple equations. This model can be applied at various levels of the network hierarchy. It can be used to study the LA crossing and dwell time within one VLR (Visitor Location Area) and the traffic due to VLR crossing at the next higher level in the hierarchy. [4]

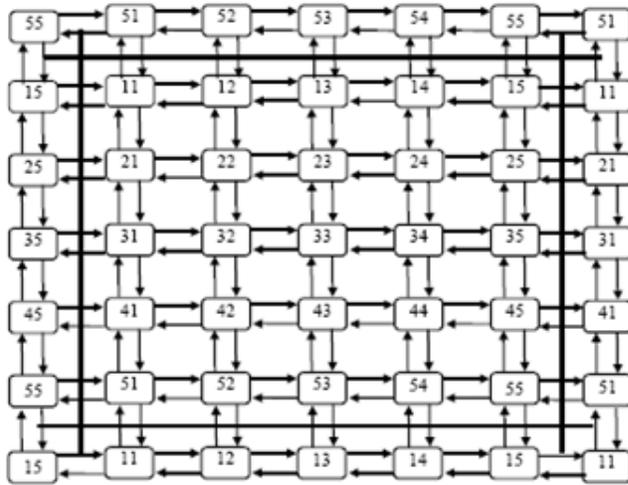


Fig 4 A Micro-cells LA Boundary

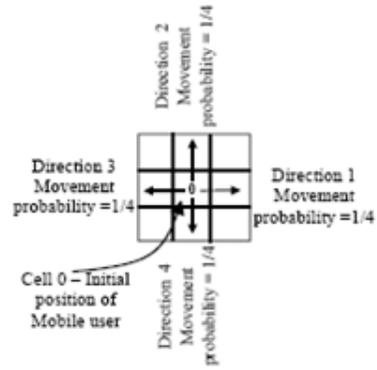
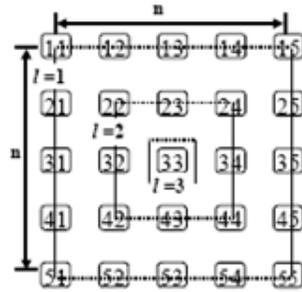


Fig 4B



n = size of LA = 5, l = loop number

Fig 5 A

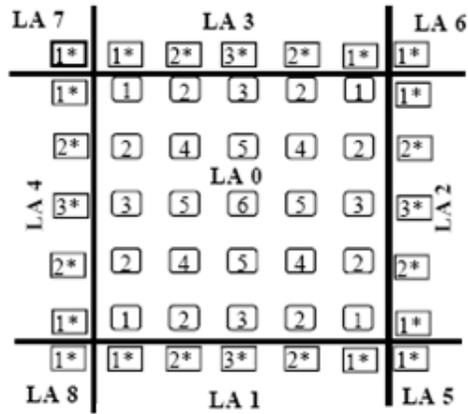


Fig 5 B

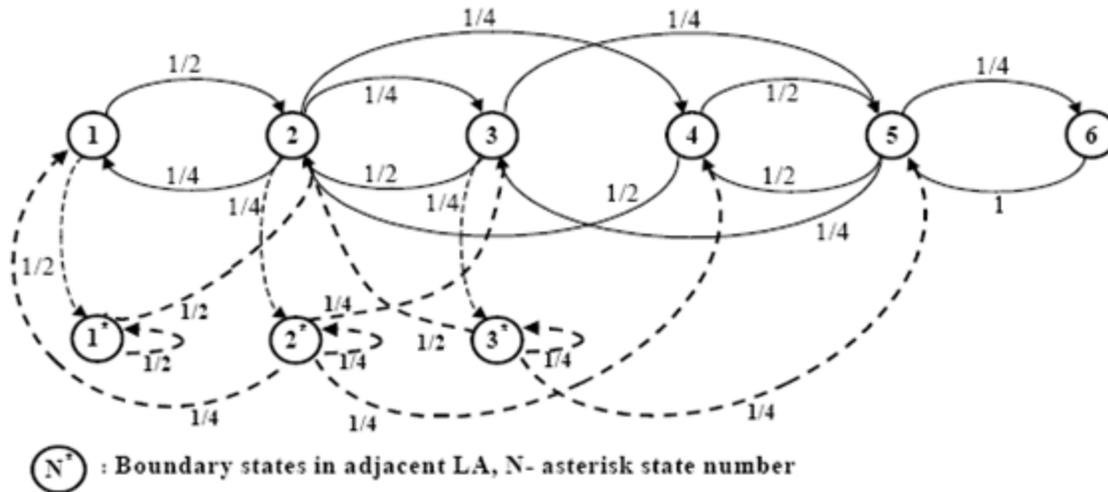


Figure 6 : State transition diagram for a 5 * 5 square cell model

2.1.5 Advantages of RW Mobility Model

- The simplest model to implement.
- Generates unpredictable movements, enabling a long-running simulation to consider all locations and node interactions. [2]

2.1.6 Disadvantages of RW Mobility Model

- Unrealistic movement patterns
- Sharp and sudden turns.
- Wrapping not observed in real applications. [2]

2.2 RANDOM WAYPOINT MOBILITY MODEL

The random waypoint mobility model introduces specific pause times between movements i.e changes in direction and speed. The random waypoint model is the most popular mobility model employed in contemporary research, and can be considered a foundation for building other mobility models. [1][2]

An MN begins by staying in one location for a certain period of time (i.e., a pause time). Once this time expires, the MN chooses a random destination as well as a speed that is uniformly distributed between $[0, \text{MAXSPEED}]$. It then travels towards the newly chosen destination at the selected speed. Upon arrival, the MN takes another break before starting the process again. [1]

We note that the movement pattern of an MN using the Random Waypoint Mobility Model is similar to the Random Walk Mobility Model if pause time is zero and $[0, \text{MAXSPEED}] = [\text{speedmin}, \text{speedmax}]$. This is a simple mobility model and is hence adopted by many authors in their simulation studies. [1][3]

Fig 7 shows the travelling pattern of an MN using the Random Waypoint Mobility Model.

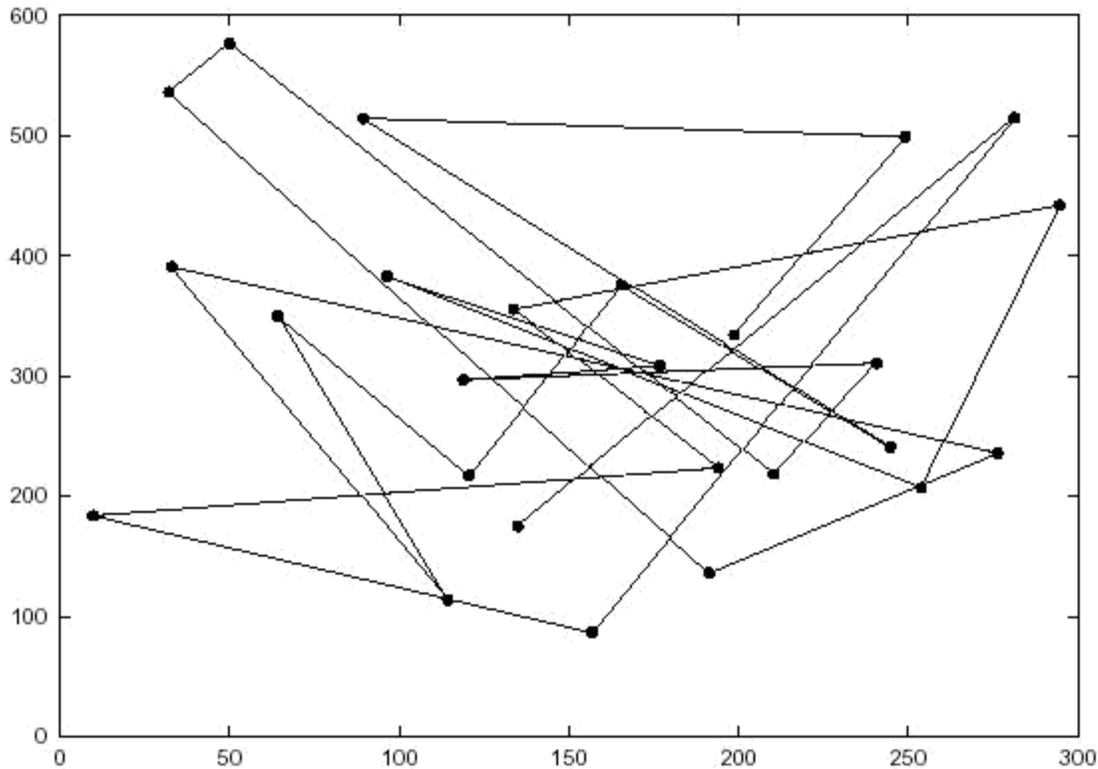


Figure 7 : Traveling pattern of an MN using the 2-D Random WayPoint Mobility Model

2.2.1 Advantages of RWP Mobility Model

- The most common use mobility model, because of its simplicity.
- A building block for developing a variety of mobility models. [2]

2.2.2 Disadvantages of RWP Mobility Model

- Lack of regular movement modeling.
- Exhibits speed decay.
- Exhibits density wave.
- Memory-less movement behaviors (a common problem for all random waypoint variations). [2]

2.3 RANDOM DIRECTION MOBILITY MODEL

The Random Direction Mobility Model (Royer et al., Submitted) was created in order to overcome a flaw discovered in the Random Waypoint Mobility Model. MNs using the Random Waypoint Mobility Model often choose new destinations, and the probability of choosing a new destination that is located in the center of the simulation area, or requires travel through the middle of the simulation area, is high. This trend is illustrated in Figure 2.4. The MNs moving with the Random Waypoint Mobility Model appear to converge, disperse, converge again, etc. In order to alleviate this type of behavior and promote a semi-constant number of neighbors, the Random Direction Mobility Model was developed. [1][3]

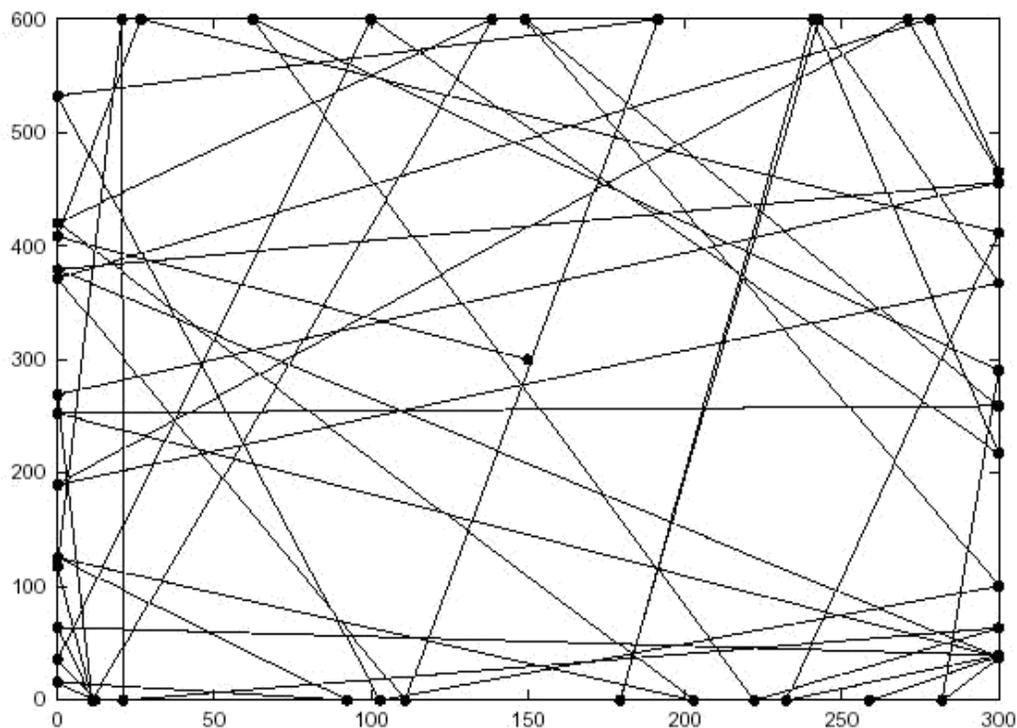


Figure 8 : Traveling pattern of an MN using the Random Direction Mobility Model

In this model, MNs choose a random direction in which to travel instead of a random destination. After choosing a random direction, an MN travels to the border of the simulation area in that direction. As soon as the boundary is reached the MN stops for a certain period of time, chooses another angular direction (between 0 and 180 degrees) and continues the process. Figure 8 shows an example path of an MN, which begins at the center of the simulation area using the Random Direction Mobility Model. [1][2]

A slight modification to the Random Direction Mobility Model is the Modified Random Direction Mobility Model. In this modified version, MNs continue to choose random directions but they are no longer forced to travel to the simulation boundary before stopping to change direction. Instead, an MN chooses a random direction and selects a destination anywhere along that direction of travel. [1]

2.3.1 Advantages of RD Mobility Model

- A variation of the random waypoint without drawback of density wave.
- Uniform distribution of chosen routes [2]

2.3.2 Disadvantages of RD Mobility Model

- Unrealistic movement pattern
- Average distances between mobile nodes are much higher than other models, leading to incorrect results for routing protocols evaluation [2]

2.4 Limitations of the Random Waypoint Model and other Random models

The Random Waypoint model and its variants are designed to mimic the movement of mobile nodes in a simplified way. Because of its simplicity of implementation and analysis, they are widely accepted. However, they may not adequately capture certain mobility characteristics of some realistic scenarios, including temporal dependency, spatial dependency and geographic restriction: [3]

- **Temporal Dependency of Velocity:** In Random Waypoint and other random models, the velocity of mobile node is a memoryless random process, i.e., the velocity at current epoch is independent of the previous epoch. Thus, some extreme mobility behavior, such as sudden stop, sudden acceleration and sharp turn, may frequently occur in the trace generated by the Random Waypoint model. However, in many real life scenarios, the speed of vehicles and pedestrians will accelerate incrementally. In addition, the direction change is also smooth. [3]
- **Spatial Dependency of Velocity:** In Random Waypoint and other random models, the mobile node is considered as an entity that moves independently of other nodes. This kind of mobility model is classified as entity mobility model. However, in some scenarios including battlefield communication and museum touring, the movement pattern of a mobile node may be influenced by certain specific 'leader' node in its neighborhood. Hence, the mobility of various nodes is indeed correlated. [3]

- **Geographic Restrictions of Movement:** In Random Waypoint and other random models, the mobile nodes can move freely within simulation field without any restrictions. However, in many realistic cases, especially for the applications used in urban areas, the movement of a mobile node may be bounded by obstacles, buildings, streets or freeways. [3]

Due to these shortcomings, other realistic mobility models were developed to examine the network characteristics of a scenario employing mobile nodes. Gauss-Markov Mobility model and Smooth-Random Mobility model are examples of mobility models with temporal dependency. Reference Point Group Mobility model(RPGM), Column Mobility model, Pursue Mobility model etc are examples of mobility models with spatial dependency. Pathway mobility model and Obstacle Mobility model are examples of mobility models with geographic restriction.

Chapter 3

METHODOLOGY

Many network simulators like NS2, Globosim, OPNET, QualNet, NetSim etc are available which can be used to simulate the mobility models and determine the network characteristics. We have implemented the mobility models in NS-2 and made a comparative study between them. [8]

3.1 About NS-2

ns is an object oriented simulator, written in C++, with an OTcl interpreter as a frontend. *ns* uses two languages because simulator has two different kinds of things it needs to do. On one hand, detailed simulations of protocols requires a systems programming language which can efficiently manipulate bytes, packet headers, and implement algorithms that run over large data sets. For these tasks run-time speed is important and turn-aroundtime (run simulation, find bug, fix bug, recompile, re-run) is less important. [9]

On the other hand, a large part of network research involves slightly varying parameters or configurations, or quickly exploring a number of scenarios. In these cases, iteration time (change the model and re-run) is more important. Since configuration runs once (at the beginning of the simulation), run-time of this part of the task is less important.

ns meets both of these needs with two languages, C++ and OTcl. C++ is fast to run but slower to change, making it suitable for detailed protocol implementation. OTcl runs much slower but can be changed very quickly (and interactively), making it ideal for simulation configuration. [9]

In NS-2, the frontend of the program is written in TCL(Tool Command Language). The backend of NS-2 simulator is written in C++ and when the tcl program is compiled, a tracefile and namfile are created which define the movement pattern of the nodes and keeps track of the number of packets sent, number of hops between 2 nodes, connection type etc at each instance of time. In addition to these, a scenario file defining the destination of mobile nodes along with their speeds and a connection pattern file(CBR file) defining the connection pattern, topology and packet type are also used to create the trace files and nam files which are then used by the simulator to simulate the network.

Also the network parameters can be explicitly mentioned during the creation of the scenario and connection-pattern files using the library functions of the simulator.

3.2 Parameters

To create the connection pattern scenario file, the syntax is

```
ns cbrgen.tcl -type val1 -nn val2 -seed val3 -mc val4 > file1
```

where:

type is used to define packet type tcp/cbr

nn is the number of mobile nodes

mc is the max number of connections

file1 is the destination where the generated machine level code will be saved so that it can be used by tracefiles to simulate the network [10]

To create the movement pattern scenario file, the syntax is

```
./setdest -nn val1 -p val2 -M val3 -t val4 -x val5 -y val6 > file2
```

where:

nn is the number of mobile nodes

p is pausetime between 2 movements

M is max speed at which a node can move

t is simulation time for which code will be generated

x is the x-coordinate of topography

y is the y-coordinate of topography

file2 is the destination where the generated machine level code will be saved so that it can be used by tracefiles to simulate the network [10]

The files file1 and file2 are also inputted in the tcl program and after the compilation of the program, tracefiles and namfiles are created which define the characteristics of network and operation of the simulator.

The choice of a mobility model can have a significant effect on the performance investigation of an ad hoc network protocol and hence to compare different mobility models, we have to analyse the trace files which define the characteristics of that mobility model for a particular protocol.

The trace file usually consists of thousands of lines of code like

```
s 10.43122623 _3_ AGT --- 21 tcp 40 [0 0 0 0] ----- [3:0 0:0 32 0] [0 0] 0 1  
r 10.43246612 _3_ RTR --- 20 ack 60 [13a 3 0 800] ----- [0:0 3:0 32 3] [0 0] 1 1
```

These can be analysed to define the network characteristics of a network simulating some mobility model and protocol. [7]

The different fields are:

```
event(send/receive) time node level --- pktid type pktsize [MAC info] ....[sending  
node: port number receiving node: port number TTL hops] additional info [7]
```

For example, by using the command

```
grep " ^s.* -NL AGT.* -lt tcp.*" tracefile-name | wc -L >> sent
```

we can get the number of tcp files sent in the network.

Similarly , we can get the no. of files received, end-to-end delay and other parameters and understand the network characteristics by analyzing the NS-2 files.

So, the trace file can be analysed to evaluate network performance parameters like number of packets sent, number of packets received, average number of hops, end-to-end delay etc. and hence to make a comparative study between the different mobility models.

Chapter 4

RESULTS

RESULTS

We have done the simulation on a network topology of 700 x 700 ms. and with varying speeds of the mobile nodes. the speed of MNs varies between sets of simulations to include 0,10, 20, 30, 40 and 50 m/s. These speeds represent maximum values. Thus, a simulation run with a maximum speed of 20 m/s has an average speed of (approximately) 10 m/s. By examining statistics including average number of hops, average end-to-end delay, and throughput (i.e., reliability) we are able to compare the effects of a mobility model on an ad hoc network protocol.

Fig 9 plots the average number of hops vs. speed

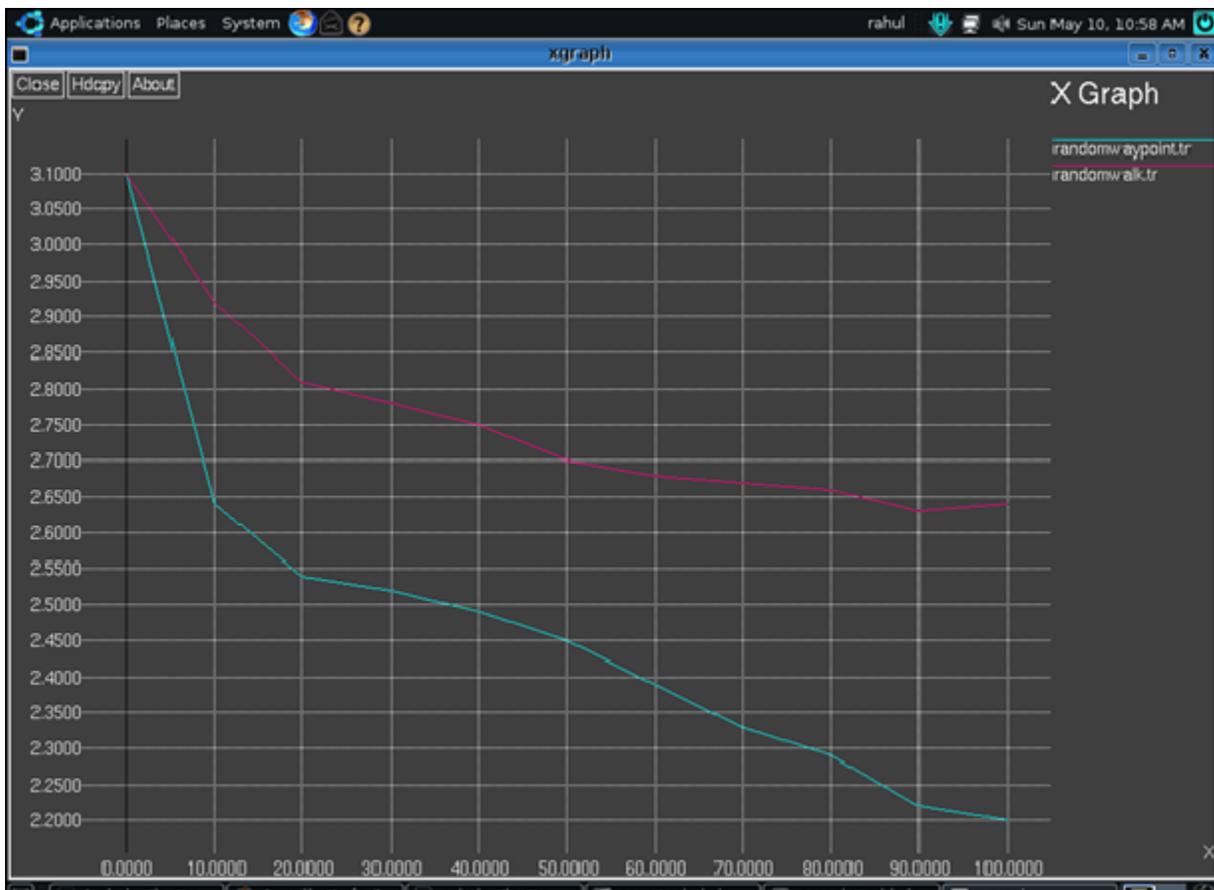


Figure 9 : average number of hops vs. speed

At speed = 0, the average number of hops is the same for both mobility models because all three mobility models begin with the same initial configuration and zero speed equates to a static network. As speed increases to 10 m/s, the mean number of hops decreases for both Random Walk and Random Waypoint. The MNs traveling with Random Waypoint often travel in the middle of the simulation area, which decreases the number of hops between any 2 nodes.

Fig 10 plots the average number of hops vs. the number of nodes in the network.

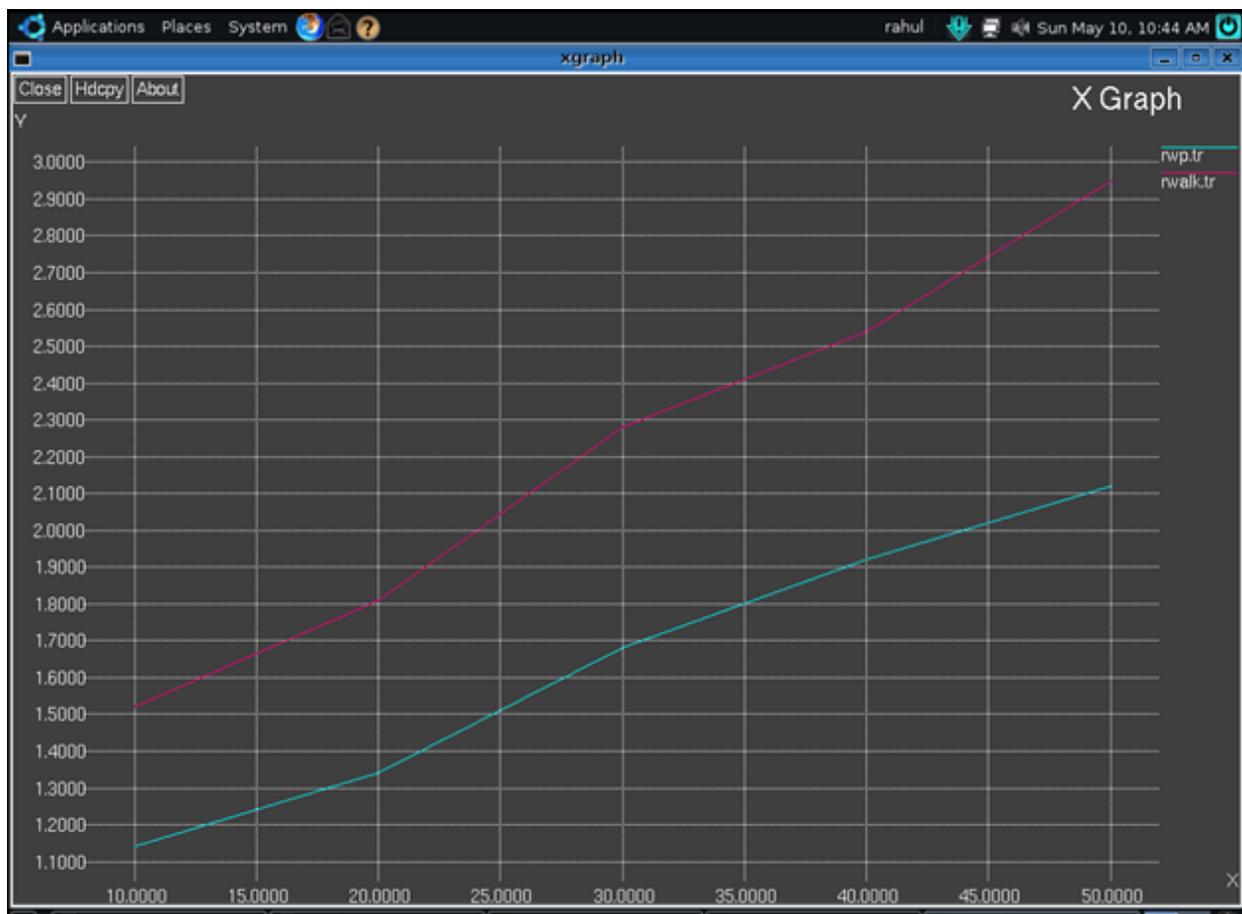


Figure 10 : average number of hops vs. number of nodes

As can be seen from figure 10, with increase in the number of mobile nodes in the network , the average number of hops also increases for both the models as nodes tend to be concentrated in the centre and hence the hop count also increases.

Fig 11 plots the throughput ratio vs. speed.

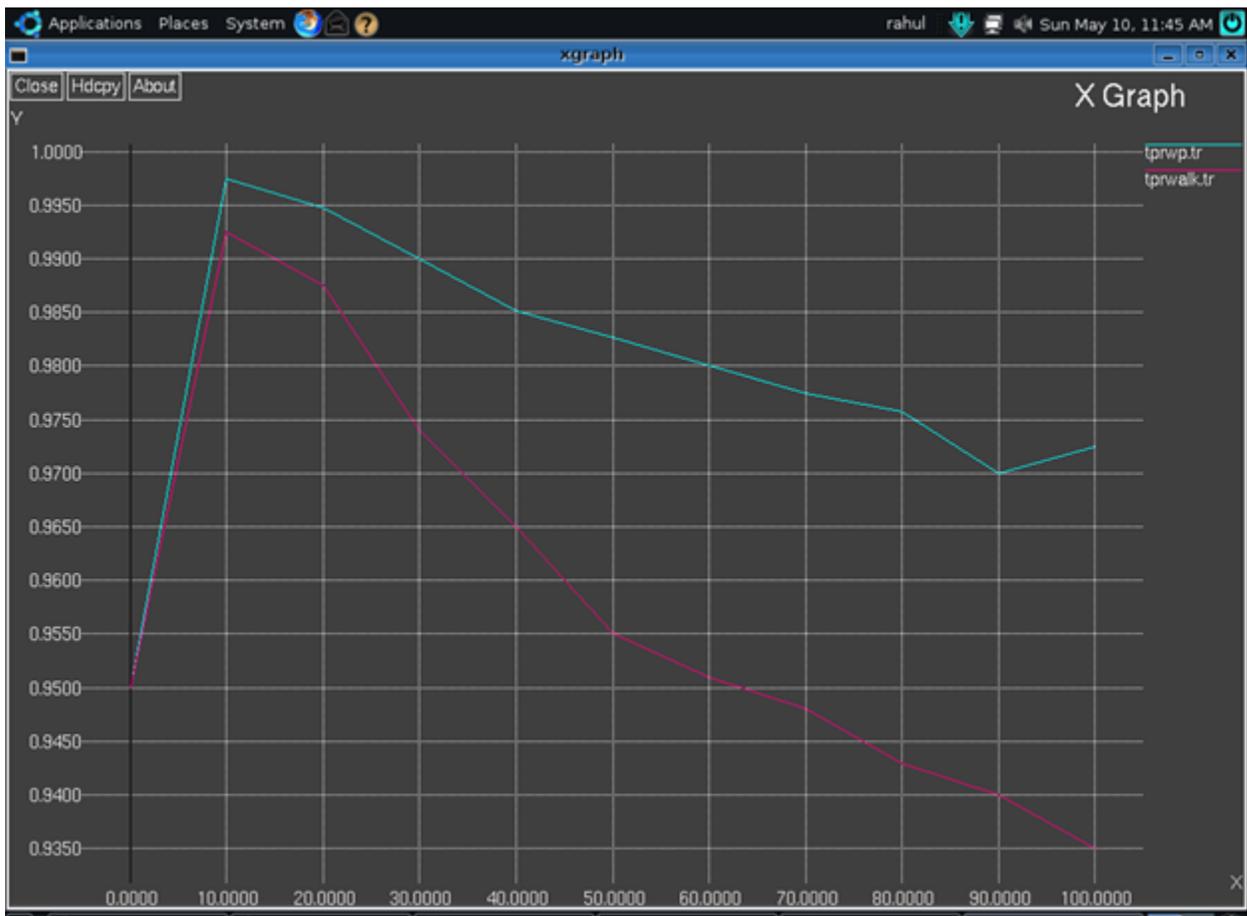


Figure 11 : throughput ratio vs. speed

The throughput ratio is a measure of reliability; it is the number of bytes received by the members divided by the number of bytes transmitted by the source. At speed = 0, the identical initial static networks produce the same results for all three mobility models. As we increase speed to 10 m/s, the mean values for both models experience an increase in throughput. As we continue to increase speed the throughput ratio for both models becomes steady. Also as the average no. of hops is less in Random Waypoint model, its throughput should be more as depicted in fig 11.

Chapter 5

CONCLUSION

CONCLUSION

The results in chapter 4 illustrate that an educated analysis of mobility models must occur before the performance evaluation of an ad hoc network. The performance of an ad hoc network protocol can vary significantly with different mobility models. The performance of an ad hoc network protocol can also vary significantly when the same mobility model is used with different parameters. The performance of an ad hoc network protocol should be evaluated with the mobility model that most closely matches the expected real-world scenario. In fact, since a single mobility model cannot accurately depict the behavior of MNs in all scenarios, researchers should consider evaluating their protocol with multiple mobility models. Different mobility models have different characteristics and serve different purposes. Therefore, instead of defaulting to a fixed Mobility Model for every simulation, or implementing a model that fails to model accurate MN behavior, the researchers should conduct a thorough analysis of appropriate mobility models before beginning their simulations.

REFERENCES

1. A survey of mobility models for Ad Hoc Network research by Tracy Camp, Jeff Boleng and Vanessa Davies
2. IEEE paper on “A Critique of Mobility Models for Wireless Network Simulation” by Babak Pazand and Chris McDonald
3. IEEE paper on “Impact of Realistic Mobility Models on Wireless Networks Performance” by Preetha Prabhakaran and Ravi Sankar
4. IEEE paper on “A Random Walk Mobility Model for Location Management in Wireless Networks” by Kuo Hsing Chiang and Nirmala Shenoy.
5. E. W. Weisstein. *The CRC Concise Encyclopedia of Mathematics*. CRC Press, 1998
6. P. Nain, D. Towsley, B. Liu, and Z. Liu, “Properties of Random Direction Models,” *INRIA technical report RR-5284, July 2004*
7. *Ns2 tutorial running simulations by Matthias Transier*
8. www.wikipedia.org
9. Mahmood Hasanlou “Introduction to mobility and network simulator 2”
10. www.isi.edu/nsnam/ns/tutorial Marc Greis tutorial on ns2