Master Thesis
Computer Science
Thesis no: MCS-2008:38
December 2008

Modeling and simulation of intrusion detection system in mobile ad-hoc networks

Piotr Jarmal

School of Engineering
Blekinge Institute of Technology
Box 520
SE – 372 25 Ronneby
Sweden

This thesis is submitted to the Department of Interaction and System Design, School of Engineering at Blekinge Institute of Technology in partial fulfillment of the requirements for the degree of Master of Science in Computer Science. The thesis is equivalent to XXX weeks of full time studies.

Contact Information:
Author:
Piotr Jarmal
Address: ul. Benedyktyńska 30/10 50-350 Wrocław, Poland
E-mail: piotrjarmal@gmail.com

University advisors:
Mia Persson
School of Engineering
Blekinge Institute of Technology, Sweden

Adam Grzech
Institute of Information Science and Engineering
Wrocław University of Technology, Poland

School of Engineering
Blekinge Institute of Technology
Box 520
SE – 372 25 Ronneby
Sweden

Politechnika Wrocławska

Wydział Informatyki i Zarządzania
kierunek studiów: Informatyka
specjalność: Studium Podstawowych Problemów Informatyki

Master Thesis

Modeling and simulation of intrusion detection system in mobile ad-hoc networks

Piotr Jarmal

Keywords:
mobile ad-hoc networks
intrusion detection systems
mobility models

short abstract:
The thesis investigates the process of modeling and simulation of the mobile ad-hoc networks. It provides a overview of the actual state of art together with a literature survey. Basic ideas of both security issues in mobile ad-hoc networks as well as intrusion detection systems are presented. Additionally some new ideas for improvements - like the AGM mobility model - are proposed, and tested during the simulation proces. As an addition a set of applications designer for automating the simulation processes were created.

Supervisor:	Adam Grzech	.....	.....
	name	*grade*	*signature*

Wrocław 2008

Contents

Mobilne sieci ad-hoc są stosunkowo młodą, lecz szybko rozwijającą dziedziną. Powodowane jest to szerokimi możliwościami wykorzystania zarówno w zastosowaniach militarnych (manewry wojsk), sytuacjach nadzwyczajnych (akcje ratunkowe) jak również cywilnych (konferencje naukowe). Wraz z ciągłym rozwojem pojawiają się różnorodne problemy, z których bezpieczeństwo wydaje się być jednym z najistotniejszych. W odróżnieniu od typowych sieci przewodowych nie istnieje bowiem możliwość zastosowania metod scentralizowanego nadzoru.

Bezprzewodowy kanał transmisji niesie ze sobą wiele niespotykanych wcześniej słabości, takich jak podatność na zagłuszanie, podsłuchiwanie jak również inne typy naruszeń bezpieczeństwa. Cechy te nakładają wysokie wymagania dotyczące procesu autoryzacji węzłów w sieci jak również zapotrzebowanie na skuteczny system detekcji (i/lub prewencji) naruszeń bezpieczeństwa zarówno z zewnątrz, jak i z wewnątrz sieci.

Uwzględniając powyższe wymagania oczywistą staje się konieczność posiadania metod służących do testowania różnorodnych pomysłów bez konieczności tworzenia kosztownej i czasochłonnej rzeczywistej sieci ad-hoc. Niniejsza praca, na podstawie przeglądu literatury jak również przemyśleń autora stara się zaprezentować ogólną metodologię procesu analizy mobilnych sieci ad-hoc, jak również wskazuje obszary wymagające dalszych badań.

Ponadto praca prezentuje analizę wyników otrzymanych poprzez przeprowadzenie symulacji, jak również dostarcza zestawu aplikacji mających na celu standaryzacje procesu tworzenia symulacji.

Mobile ad-hoc networks are a relatively new and rapidly evolving area of interests, which has been quickly adapted in many different fields such as military operations, emergency situations as well as civilian ad-hoc situations like conference and classroom. With the development of wireless communication security becomes one of the major problems. Unlike in the wired equivalents we cannot use any centralized approach which leaves nodes without any external supervision.

Wireless transmission is susceptible to signal jamming, eavesdropping or other similar attacks. The fact that nodes are mobile brings further complications. These features require advanced authentication procedures and intelligent intrusion detection system (IDS). Considering all of the possible problems, it is important to have the possibility of testing and evaluate new ideas without having to establish a real working test network.

Based on a survey of literature as well as some own work, the author presents the overall methodology for analyzing mobile ad-hoc networks as well as some of possible upgrades to the process. Some basic simulations with different scenarios were carried out and the results discussed. Also a set of applications for automation of the simulation process were created as a part of this work.

INTRODUCTION

Background

Ad-hoc networks are a relatively new and rapidly evolving area of interests, which has been quickly adapted in many different fields such as military operations, emergency situations as well as civilian ad-hoc situations like conference and classroom. With the development of wireless communication security becomes one of the major problems. Unlike in the wired equivalents we cannot use any centralized approach which leaves nodes without any external supervision.

New problems arise on the medium access control layer. Wireless transmission is susceptible to signal jamming, eavesdropping or other similar attacks. Moreover attacks against the ad-hoc routing infrastructure may be made from external or internal nodes. The fact that nodes are mobile brings further complications . These features require advanced authentication procedures and intelligent intrusion detection system (IDS). The latter is a system that tries to detect and alert on attempted intrusions into a system or network, where an intrusion is considered to be any unauthorized or unwanted activity on that system or network. The main role of intrusion detection systems is to monitor audit data, look for intrusions to the system, and initiate a proper response.

Research Aims and Objectives 'guidelines'

The thesis investigates if it is possible to successfully apply intrusion detection systems into mobile ad-hoc networks. MANETs compared to other network suffer from some serious drawbacks. The lack of any fixed infrastructure available is one of the more important ones. Cause of this ad-hoc networks are best suited in situations where the infrastructure is either unavailable or not trusted for example military scenarios. Cause of this the security aspect becomes one of the most important and nontrivial aspects.

Thus, the aim of the thesis is to create a methodology for examining mobile ad-hoc networks, both hostile and non-hostile environment. Also there is a strong focus on simplifying the simulating process in other to allow gathering signatures for distributed IDS scheme.

Following objectives are realized to attain the goal:

- Review of ad-hoc networks modeling methods
- Presentation of most common security issues in MANET
- Review of IDS system proposed for MANET
- Analyze of the simulation process and present possible improvements
- Perform experiments in order to evaluate existing methods
- Evaluation of gathered information

Research Questions

The thesis addresses the following research questions:

Research Question 1: What are the important elements when modeling mobile ad-hoc networks.

Research Question 2: What improvements can be proposed for the process of modeling mobile ad-hoc networks.

Research Question 3: What are the most common security issues (attacks) in mobile ad-hoc networks.

Research Question 4: Do the existing intrusion detection systems work in MANET and is it possible to improve their performance.

Thesis Outline

This thesis is organized as follows:

Introduction

Chapter 1 introduces mobile ad-hoc networks.

Chapter 2 provides basic information about routing protocols and mobility models used while modeling ad-hoc networks. It also introduces the authors idea for the new mobility model AGM.

Chapter 3 discusses the possible issues of mobile ad-hoc networks security.

Chapter 4 provides basic information about intrusion detection systems, possible classifications as well as some of the more interesting research areas.

Chapter 5 presents the simulation process, shows some of the results and discusses the outcomes.

Chapter 6 present the discussion of the topic as well as some future work recommendations. There answers to the research questions can also be found.

1 MOBILE AD-HOC NETWORKS

1.1 History

First ad-hoc networks emerged in 1972 as Packet Radio Network (PRNET) and it was used for providing different networking capabilities in combat environment. They evolved into a part of Survivable Adaptive Radio Networks (SURAN) program in the early 1980s. The goal of the program was to provide a packed-switched network to mobile battlefield in an hostile (soldiers, tanks etc.) environment without any infrastructure. SURAN successfully improved the radios' performance, making them smaller, cheaper and more resilient to electronic attacks [Fre01].

In 1990s open software and also notebook computers become popular. Together with development of communications equipment, it caused a new phase in ad-hoc networks development. Idea of a collection of mobile nodes in an environment without any given infrastructure was presented at several research conferences. IEEE 802.11 adopted the term "ad-hoc networks" and the concept of non-military become more popular in the research community. Meanwhile Department of Defense continued their research in projects like Global Mobile Information Systems (GloMo) or Near-term Digital Radio (NTDR).

Constantly growing interest with ad-hoc networking encourage creation of a Mobile Ad Hoc Networking (MANET) working group within the Internet Engineering Task Force (IETF). The main goal of the group was to standardize routing protocols for ad hoc networks. As a result of their work reactive and proactive routing has evolved. After that IEEE 802.11 subcommittee standardized a medium access protocol, based on collision avoidance that tolerated hidden terminals, making it usable for building mobile ad-hoc networks prototypes out of notebooks and 802.11 PCMCIA cards.

Today, MANET research mainly concentrates on two issues. First scalability, mainly in networks designed for military use, and second – security. Due to the shared radio medium and potentially insecure environment ad-hoc networks are more sensitive to attacks then their wired equivalent.

1.2 Overview

A mobile ad hoc network (MANET) is a kind of wireless ad-hoc network. It is defined as a collection of mobile nodes (routers and associated hosts) connected by a wireless link in such a manner that the interconnections between them can change on a continual manner. The union of the nodes forms an arbitrary, dynamical topology. The nodes are able to move independently from each other or organize themselves on any level (single nodes, groups etc.) causing the topology to change in rapidly and unpredictably ways. Those networks can operate in a standalone fashion as well as they can be connected to a larger network (some nodes can provide internet connection for the network).

They can be used in military operations, emergency situations as well as civilian ad-hoc situations like conference and classroom. And so many different protocols were (and still are) developed to meet arising issues.

There are even some specific subtypes of MANET that are dedicated to some of the problems like vehicular ad-hoc networks (VANETs). VANETs are mainly used for communication among vehicles and between vehicles and roadside equipment generally (in most research) within a city environment.

A good example of mobile ad-hoc networks usage and challenges in military environment is shown on figure 1.1.

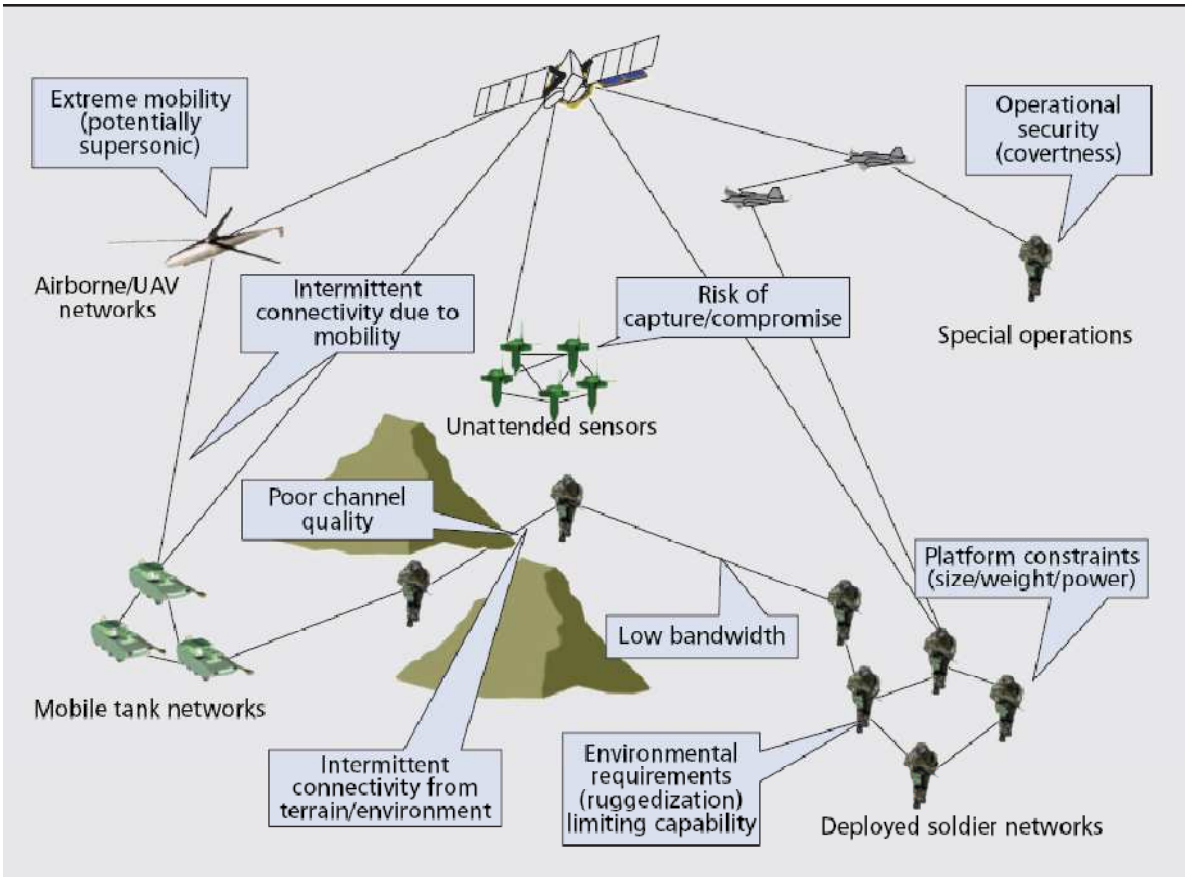


Figure 1.1 Areas of MANET usage and possible difficulties in military environment (from [Bur06])

2 MODELING AD-HOC NETWORKS

Whenever working with ad-hoc networks it is high likely that simulations will be needed. The main goal of any simulations is to imitate the real life scenarios as well as possible. To do this an appropriate models should be created and two of the most important parts of an ad-hoc network model that can have significant affect on the network performance is the mobility model and routing protocol.

In this chapter some of the basic mobility models for mobile ad-hoc networks (MANETs), as well as the most popular routing protocols. Additionally author own idea in that field of mobility models will be described.

2.1 Mobility models

2.1.1 Introduction

Mobility of users in an ad-hoc network has many advantages, but providing the necessary quality and performance of such a network is one of the key problems. In almost every work related with mobile ad-hoc networks, simulations play a very important role. To obtain more reliable results we need to create as realistic simulations as possible. And one of the most important issues that can have a significant impact on the simulation results is the mobility model. Many work has been done in this area and there exists many mobility models like the random direction model, the Brownian walk or the random Gauss-Markov model [Ben79]. Nevertheless still new models are created in order to make scenarios more realistic [Sou05].

2.1.2 Basic random models

In this section few mobility models commonly used in MANETs simulations will be presented. All of them focus their interest on the way a single object (mobile node) behaves within the simulation area. The final simulation is created by adding a set number of mobile nods each independent from the rest. Random models are easy to implement, test and don't require many computing resources. There are also a good reference point for most of newly created models and ideas. Many different versions of those models were created during the last few years. Some of the basic versions are presented below.

2.1.2.1 Random Walker Mobility Model

Random walker mobility model This is one of the most basic and easiest to understand mobility model that is still widely used in simulations and research. The basic idea of this model is an assumption that most of the studied objects move in unpredictable way. The random walker model was developed to imitate those movements. There is many work concentrated on this model and it's mathematical characteristics [KH01]. It was proven that in one or two-dimensional version of this

model the ‘walker’ returns to its starting point (with probability of 1) [Wei98]. The model is quite easy to comprehend.

We start with a mobile node, chose a direction from [0, 2π] and a velocity from [velmin, velmax] where velmin and velmax are parameters representing the minimal and maximal possible velocity that the node can have. The chosen values should be fully random, for example uniform distribution. Then the node moves according to the chosen values for a set period of time t (there is also a version of this model that the limit is not time but a distance d traveled by the node) after which a new direction and velocity is chosen in the same manner. If at any time the node reaches the borders of the simulation area it ‘bounces off’ changing its direction according to the angle it ‘hit’ the border. En example of random walker is shown on figure 2.1.

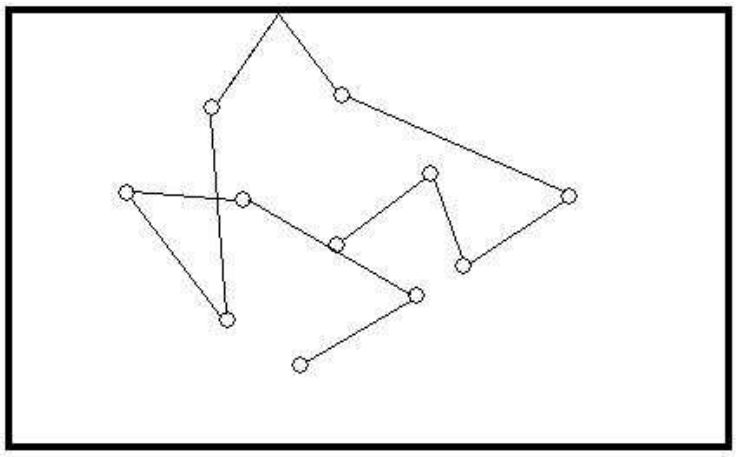


Figure 2.1 Example movement path in random walker

2.1.2.2 Random Waypoint Mobility Model

Random waypoint mobility model was first used to test Dynamic Source Routing [Joh96] and still it’s commonly used in MANET simulations. In this model, the mobile node travels from its starting position on the simulation area to another point chosen randomly (uniform distribution) with a constant velocity (independent from its actual positions) chosen randomly from [velmin, velmax].

Random waypoint model includes delays between changes of direction and velocity. It means that the mobile node after reaching its destination will wait for a set period of time. En example of random waypoint is shown on figure 2.2.

It is often suggested to ignore the initial phase of the simulation using random waypoint model (about 1000 first seconds, depending on the movement speed and t). It is also said that using this model an occurrence of a density wave can be expected – a situation when more nodes are gathering in one part of the simulation area (in this case the middle). The reason of this is a high probability that either the next destination point is in the middle or the path to the point from the current location leads trough the center.

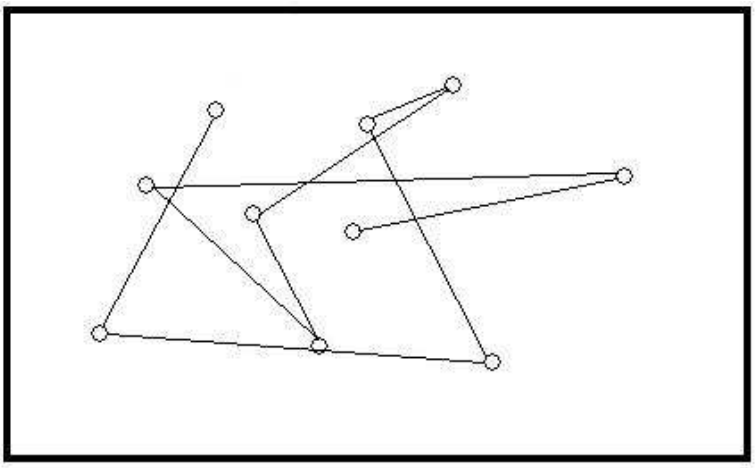


Figure 2.2 Example movement path in random waypoint

2.1.2.3 Random Direction Mobility Model

Random direction is another widely used mobility model. It is very close related to the random walker model. As the previous it chose a random direction and velocity at the start of the simulation, then it moves not for a set period of time but until it reaches any of the simulation borders. At this point it waits for a set period of time t after which new direction and velocity are chosen (respectively from [0, π] and [velmin, velmax]). En example of random direction is shown on figure 2.3.

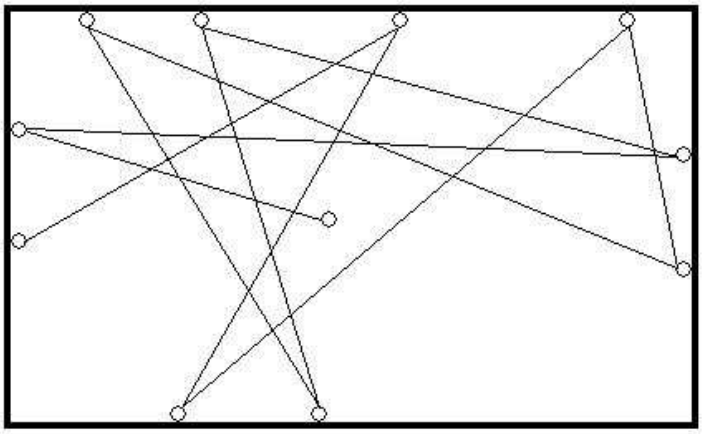


Figure 2.3 Example movement path in random direction

Although random direction may seem very similar to random walker model (or others), it is shown [Roy01][Di06] that it has different characteristics when used with MANET. For example using this model can solve the density wave problem that arises in random waypoint model.

2.1.2.4 Gaus-Markow Mobility Model

All previous mobility models are memoryless, it means that the next steps movement (direction, velocity) doesn't depend in any way on previous decisions. And so many unexpected changes in both direction and speed will occur. In real life

situations such a behavior is very rare. A random mobility model without this drawback was presented in [Lia99]. In this model the mobile node moves according to calculated direction and velocity for a period of time after which new direction (velocity) are calculated using the values from the previous step and a random values. It is done according to the fallowing equations:

$$v_n = \alpha v_{n-1} + (1-\alpha)\bar{v} + \sqrt{(1-\alpha^2)}v_{x_{n-1}}$$

$$d_n = \alpha d_{n-1} + (1-\alpha)\bar{d} + \sqrt{(1-\alpha^2)}d_{x_{n-1}}$$

Where vn and dn are velocity and direction in the n-th time period (step), vn-1 and dn-1 are velocity and direction in the n-1 step, $\bar{v}$ and $\bar{d}$ are mean values of velocity and direction when $n \to \infty$ and α is the tuning parameter. Changing α we can change the randomness level of the model (for example α=0 will provide a linear motion when α=1 will result in fully random model – random walker). Sometimes an additional rule is applied to ensure that the mobile node will not remain near the border of the area for a long time. En example of Gaus - Markow is shown on figure 2.4.

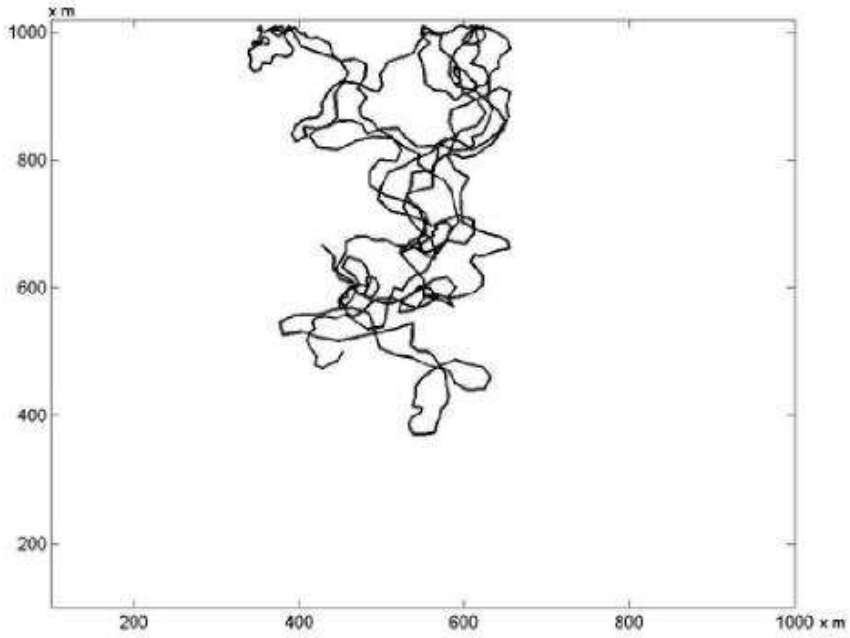


Figure 2.4 Example movement path in Gaus-Markow (from [Ari06])

2.1.3 Group mobility models

Previously all the mobile nodes ware independent objects that moved without any care about other objects within the simulation area. That approach often doesn't represent the reality in a sufficient manner. If it is important to mimic the real situations more accurate some other models should be considered. The group mobility models concentrate on the behavior of an entire group of objects not single instances. That could be important when trying to simulate some of the real life situations like rescue operations, pursues or military operations where we must assume that all the objects movements are in some way coordinated. The first from the presented below models could easily represent a group of soldiers marching together towards an enemy or after some changes a group of people fallowing each other. The second a pursuit situation. The third one will present an interesting idea

for a mobility model that could imitate many different situations depending on the chosen parameters.

2.1.3.1 Column Mobility Model

Using this model we try to simulate a group of mobile nodes forming a line that is moving in a specified direction. Each node is also granted some limited freedom. In this model each node has it's reference point that moves according the advance vector (new_reference_point = old_reference_point + advance_vector). The 'freedom' is provided by allowing the mobile node to move randomly around it's reference point (any of the random mobility models can be used for this). En example of column mobility is shown on figure 2.5, the black dots represent the reference points and the circles are mobile nodes.

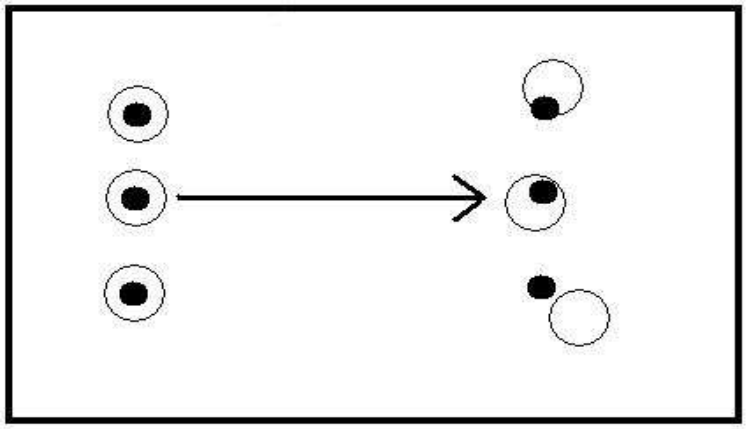


Figure 2.5 Example movement column mobility model

2.1.3.2 Pursuit Mobility Model

This mobility model (accordingly to its name) is supposed to imitate a group of mobile nodes pursuing an escaping target. The new nodes positions are calculated using their old positions, an acceleration function, and a random vector according to: new_position = old_position + acceleration(target_old_position) + random_vector The role of the acceleration brings the information about the target movement and according to [San08] it is supposed to allow only a limited maximum speed during the pursuit. The random vector can be obtained using any random mobility models (for example random walker). En example of column mobility is shown on figure 2.6, where the black dot is the target and the circles are the mobile nodes.

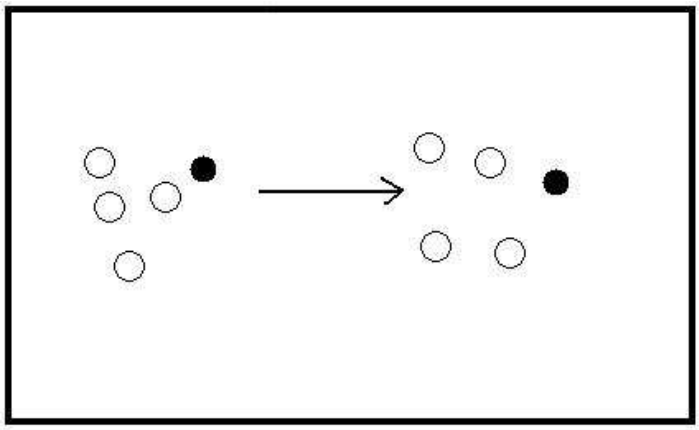


Figure 2.6 Example movement pursuit mobility model

2.1.3.3 Reference Point Group Mobility Model (RPGM)

Many of the group mobility models movement patterns can be quite easy obtained using reference point group mobility model proposed in [Hon99]. In this model each group of mobile nodes has a logical 'center' and the motion of that center determines the behavior of the group nodes (direction, velocity). It is possible for individual nodes to move freely (using any mobility models) around their individual reference points. The group motion is represented by a vector called a group motion vector $\overline{GM}$. In the model $\overline{GM}$ is used to calculate the new coordinates of the references points. It can be defined earlier or chosen randomly using another (or the same) model as the individual nodes.

En example two group model is shown on figure 2.7.

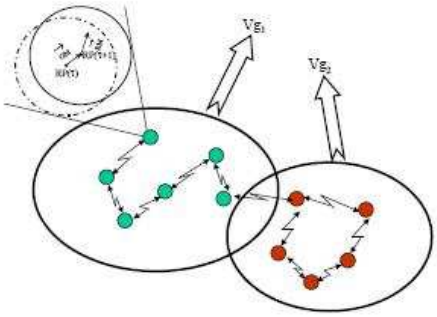


Figure 2.7 Example of RPGM (from [Hon99])

2.1.4 AGM mobility model

AGM stand for anti-gravity movement. It was first proposed as a movement strategy for robocode fights. It was supposed to keep the bots away from the center of the field and from larger groups of other bots (or other dangerous places). It was also difficult for any targeting algorithms to predict the next step of a 'bot' using this movement pattern. There were few different versions of AGM depending on the user. More information about robocode and ATM used there can be found in [Jar06].

The model proposed below is a AGM version adopted for MANETs environment. It assumes that there are some specific points (or moving objects) on

the simulation area that can affect the mobile node movement. In real life situations such as rescue missions or battlefield situations such points could represent some dangerous points (high slopes, rivers, enemy soldiers) or after slight changes an attraction points (observation point, cover, friendly soldiers). The AGM mobility model is supposed to mimic movement of a node that chooses its movement path according to such points and adding some level of freedom in decision making.

2.1.4.1 AGM Basics

The movement is carried out on a rectangular area. It is also possible to use unlimited simulation area but it could cause the mobile nodes no move fully randomly after moving away from other nodes (objects). The lower left corner will have coordinates [0, 0] and the upper right [xmax, ymax]. Every object on the board is represented by two coordinates $[x_o, y_o]$ where $x_o \in [0, x_{\max}]$ and $y_o \in [0, y_{\max}]$.

To each object will be assigned a 'repulsive' value h. The mobile nodes can have this value but it isn't obligatory (it can be set to 0). If h would be a negative value it's meaning would change from repulsive to attracting force.

$d_{o_1 o_2}$ will be the distance between objects calculated according to the Euclidean metric that is:

$d_{o_1 o_2} = \sqrt{(x_{o_1} - x_{o_2})^2 + (y_{o_1} - y_{o_2})^2}$ where: $x_{o_1}, x_{o_2}, y_{o_1}, y_{o_2}$ are the coordinates of objects o_1 and o_2.

r_o - the range of an object. It should be applied only to mobile nodes and represents the distance that other objects should be within to affect the node. Setting this value too high would result with a situation that many needles calculations will be carried out.

v - velocity of the mobile node.

$\overline{mv_c}$ - a movement vector in step c. Specifies in which direction from the current position the mobile node will move.

g - a distance factor. Determines how fast (or slow) the influence of other objects fade with increasing distance.

t - time period for one step.

2.1.4.2 Movement

The mobile node will move according to its movement vector and velocity in every step for a fix period of time t. After this period new movement vector (and velocity) will be calculated according to the equation:

$$\overline{mv_c} = [\sum_{o \in o_{n_r}} (x_n - x_o) * (h / d_{n,o} * g),$$

$$\sum_{o \in o_{n_r}} (y_n - y_o) * (h / d_{n,o} * g)] + \overline{R}$$

where:

n - mobile node

o_{n_r} - set of objects within r range from node n

$\overline{R}$ - random vector ($[x_r, y_r]$)
The movement vector determines the direction in which mobile node will move. It also affects the velocity of the node that can be calculated according to:
$v = \sqrt{x_{mv}^2 + y_{mv}^2} * s$ where s is randomly chosen from [velmin, velmax].

According to the above it would be possible (in some specific situations) for the node to move with almost infinite velocity. To solve this problem it's necessary to put some additional limitations on the v such as:
$v' = \min(\ vel\ \max,\ \max(\ vel\ \min,\ v))$
The above assure us that the final velocity v' will be between velmin and velmax values.

2.1.4.3 Time Steps

The movement of the mobile nodes will represent a series of straight line moves in a selected direction. The movement time is regulated by the t parameter. At the beginning of each of the moves movement vectors and velocities have to be calculated for every mobile node.

Different t values can have significant influence on the final movement paths. The more often all calculations will be made the more 'smoothly' the movement paths will look, but this can require more computational power. Also when setting a low value of t it is expected that the random vector will have much more effect on the node movement. In this case it is suggested to apply an additional time period that will represent the time that once generated random vector is valid. It would be a good idea that the new time period (t_R) to be a multiple value of t (for example $10t$).

Similar issue can be observed regarding to the velocity value. I is highly recommended that the s value is also generated after time t_R. This will assure that the mobile nodes won't have many significant changes and obtaining more realistic movement paths.

2.1.5 Vehicular ad-hoc networks

Vehicular ad-hoc networks (VANETs) are a subset of MANET. They are attracting increasing attention in recent years. Their main concern is to simulate vehicles and pedestrians flows within city areas. While creating such a model there are many important issues that have to be addressed. First all the movement is limited by the city topology meaning roads, intersections buildings (they can block the transmission path). Another problem is to realistically present the vehicle movement that will have to include accelerations and deceleration, concerning about other vehicles, traffic jams etc. Another aspect of the city scenario is the fact that many observed patterns (for example commonly used roads, gathering points) can change in time. There may be a lot of vehicle traffic in the city center during the day but in the evening the pedestrians will be more common. All this forced the researchers to work with specific and more complicated mobility models for

VANET. A good survey in this area along with a proposition of a taxonomy can be found in [Har06].

2.2 Routing protocols

2.2.1 Introduction

Many wireless networks like WLANs depend on some provided infrastructure. All mobile nodes connect to so called base stations that are supposed to act as routers. When a node moves further away from the base station another one will 'intercept' the connection maintaining the communication within the network. However ad-hoc networks do not have this advantage and cannot depend on any provided infrastructure. Moreover constantly changing topology of the network due to the mobility of the nodes can cause some additional problems.

There are some routing protocols designed for ad-hoc networks that tray to overcome those difficulties. Generally they are divided into three groups, namely pure proactive protocols, pure reactive protocols and hybrid of the previous two [Tao03].

2.2.2 Proactive routing

Pure proactive protocols are also called table-driven. In those protocols each node trays to maintain an up-to-date information about routes to each other node by regularly exchanging packets to determine the actual network topology. In this way every single node has a complete (and actual) picture of the network. Using those protocols ensures minimum delay in determining the route between two nodes. However higher mobility of the nodes can cause some rapid changes that would require the routing information to be updated more often resulting in generation of a high overhead of routing maintenance packets. Some of the most popular proactive routing protocols are: Destination-Sequenced Distance-Vector Routing protocol (DSDV), Clusterhead Gateway Switch Routing (CGSR) or Wireless Routing Protocol (WRP).

2.2.2.1 Destination-Sequenced Distance-Vector Routing

Destination-Sequenced Distance-Vector Routing is a table-driven routing scheme for ad hoc mobile networks based on the Bellman-Ford algorithm. It was developed by C.Perkins and P.Bhagwat in 1994 [Per94]. One of the important improvements of DSSV to Bellman-Ford algorithm is that it solves the routing loop problem.

In DSDV each mobile node maintains a routing table with the number of hops to all possible destinations within the network. Each entry is also provided with a sequence number that helps the nodes to distinguish new updates from old ones. Routing information is distributed between nodes by sending full dumps – that carry all available routing information – Infrequently and smaller incremental updates – that contains only the information that changed since the last full dump –more frequently.

DSDV is quite suitable for creating ad hoc networks with small number of nodes. Since no formal specification of this algorithm is present there is no commercial implementation of this algorithm. Many improved forms of this algorithm have been suggested however there are still two major drawbacks. First DSDV requires a regular update of its routing tables, which uses up battery power and a small amount of bandwidth even when the network is idle. Second one concerns the necessity of a new sequence number after any changes of the network topology, thus DSDV is suggested for non dynamic networks.

2.2.2.2 Clusterhead Gateway Switch Routing

The Clusterhead Gateway Switch Routing [Chi97] uses DSDV as an underlying protocol.

The mobile nodes within the network are aggregated into clusters and in each cluster a cluster-head is selected. All nodes that are in the communication range of the cluster-head belong to its cluster. If a node is in a range of two different cluster-heads it becomes a gateway node and will participate in data exchange of clusters. Using CGSR with dynamic networks can cause performance degradation due to frequent cluster-head elections, so CGSR uses a Least Cluster Change (LCC) algorithm. In LCC, cluster-head change occurs only if a change in network causes two cluster-heads to come into one cluster or one of the nodes moves out of the range of all the cluster-heads. The general routing algorithm for CGSR is as follows:

First the packet is sent to the cluster-head of the current cluster. Then the cluster-head checks the cluster-head of the destination node and sends the packet to the gateway node that connects this cluster and the next cluster along the route to the destination. The gateway sends it to that cluster-head and so on till the destination cluster-head is reached in this way. The destination cluster-head then transmits the packet to the destination. En example of CGSR is shown on figure 2.8.

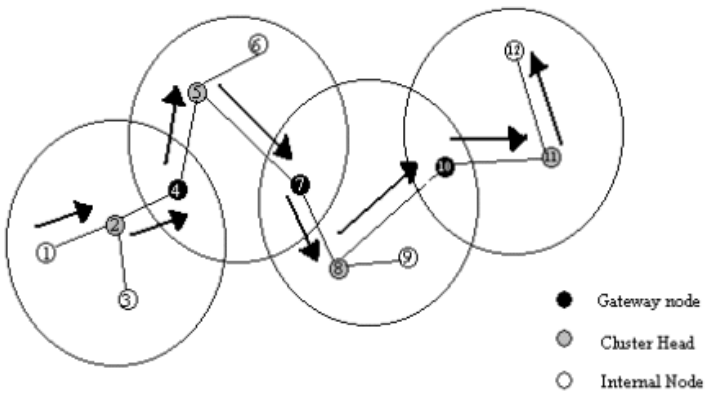


Figure 2.8 CGRS example (from wiki.uni.lu/wiki-wiki 11.03.2008)

2.2.2.3 Wireless Routing Protocol

The wireless routing protocol [Mur96] (similar to DSDV), is based on the distributed Bellman-Ford algorithm. It uses a unique method of maintaining information regarding the shortest distance to every destination node in the network and the penultimate hop node on the path to every destination. WRP maintains an up-to-date view of the network so every node has a readily available route to every destination node in the network. For maintaining that knowledge each node uses a set

of tables: distance table (DT) contains the network view of the neighbors of a node. It contains a matrix where each element contains the distance and the penultimate node reported by a neighbor for a particular destination; routing table (RT) contains the up-to-date view of the network for all known destinations. It keeps the shortest distance, the predecessor node (penultimate node), the successor node (the next node to reach the destination), and a flag indicating the status of the path (correct/error/null); link cost table (LCT) contains the cost (the number of hops to the destination) of relaying messages through each link. The cost of a broken link is set as infinity. It also contains the number of update periods (intervals between two successive periodic updates) passed since the last successful update was received from that link; message retransmission list (MRL) contains an entry for every update message that is to be retransmitted and maintains a counter for each entry. This counter is decremented after every retransmission of an update message.

Each update message contains a list of updates. A node also marks each node in the RT that has to acknowledge the update message it transmitted. Once the counter reaches zero, the entries in the update message for which no acknowledgments have been received are to be retransmitted and the update message is deleted. Thus, a node detects a link break by the number of update periods missed since the last successful transmission. After receiving an update message, a node not only updates the distance for transmission neighbors but also checks the other neighbors' distance.

The complexity of maintenance of multiple tables puts a demand on memory and processing power of nodes. At high mobility, the control overhead involved in updating table entries is similar to DSDV and hence is not suitable for highly dynamic and also for a very large ad hoc wireless network.

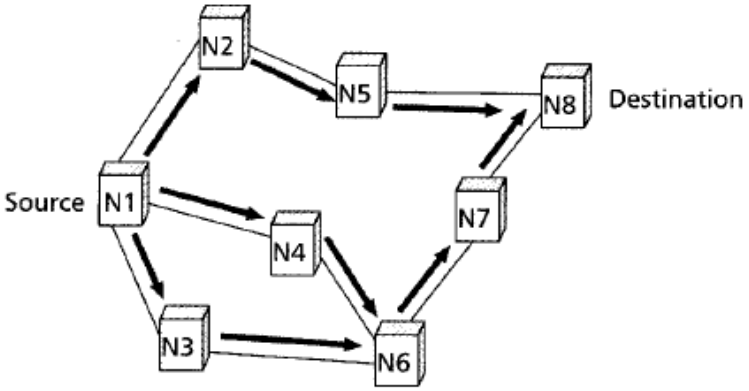
2.2.3 Reactive routing

Pure reactive protocols also called on-demand routing protocols are based on a different approach. In those protocols information about a route between two nodes is acquired only when it is desired by the source node. The node initiates a route discovery process within the network by sending a route request then waiting for a response. Once the route is established it is maintained until it is no longer valid or no longer desired [Roy99]. Some popular reactive routing protocols are: Ad Hoc On-Demand Distance Vector (AODV), Dynamic Source Routing (DSR) or Temporally Ordered Routing Algorithm (TORA).

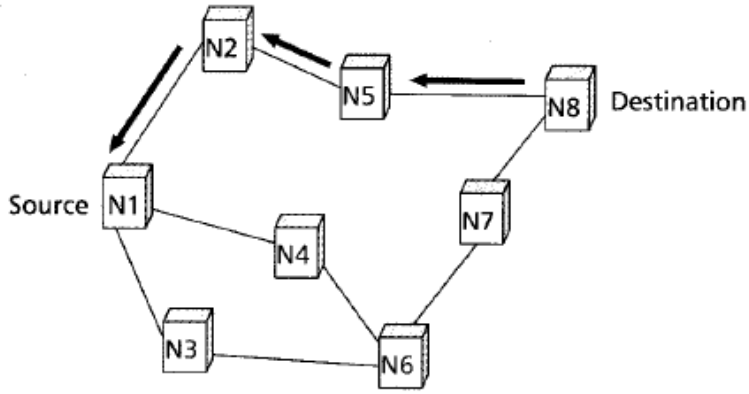
2.2.3.1 Ad-hoc On-demand Distance Vector Routing Protocol

Ad-hoc On-demand Distance Vector routing protocol [Per03] it is jointly developed in Nokia Research Center of University of California, Santa Barbara and University of Cincinnati by C. Perkins and S. Das. It is based on the DSDV algorithm discussed earlier. AODV main improvement is minimizing the number of required broadcasts by creating routes on demand basis. When a source node wants to send a message to another node to which it does not know a valid route it initializes a rout discovery process by broadcasting a route request packet (RREQ) to its neighbors. The neighbors in turn broadcast the packet to their neighbors till it reaches the destination node or an intermediate node with valid route information about the destination. Each node discards RREQs that it has already seen. That can

be done by using sequence number and broadcast ID maintained on each node that the source node includes into RREQ. During the forwarding RREQ packages the intermediate nodes also records in their route tables the source (from which they got the message) thereby establishing a reverse path. After reaching the destination node (or intermediate node with 'fresh enough' routing information) a route reply packet (RREP) is created and sent back along the reverse path. Each node along the way updates its routing table according from which node the RREP came. En example of rout discovery process is shown on figure 2.9.



(a) Propagation of the RREQ



(b) Path of the RREP to the source

Figure 2.9 AODV rout discovery (from [Roy99])

If the source moves then it can reinitiate route discovery to the destination. If a node along the route moves, its upstream neighbor notices the move and send a link failure notification message to

each of its active upstream. These nodes in turn propagate the link failure notification to their upstream neighbors, and so on until the source node is reached. The source node may then choose to reinitiate route discovery for that destination if a route is still desired.

2.2.3.2 Dynamic Source Routing

Dynamic Source Routing (DSR) [Joh07] is similar to AODV in that it forms a route on-demand when a transmitting computer requests one. However, it uses source routing instead of relying on the routing table at each intermediate node.

Instead each node is required to maintain a route cache containing routes the node is aware of.

The protocol consists of two major phases: route discovery and route maintenance. When a source node wants to send a packet to some destination, it first checks its route cache for a route to the destination. If it has an unexpired route, it will use it to send the packet. But, if the node does

not have such a route, it initiates route discovery process by broadcasting a route request packet. This route request contains the address of the source and the destination, and a unique identification number. Each intermediate node after receiving the packet checks whether it knows of a route to the

destination. If it does not, it adds its own address to the route record and then sends the packet to its neighbors. Intermediate nodes forward the route request only if the request has not yet been seen by the node and if it's address does not already appear in the route record. A route reply is generated when either the destination or an intermediate node with current information

about the destination receives the route request packet. The route reply packet is generated using the route record of the route request packet and – if not generated by the destination node – route cash of the generating node. The route reply is sent back according to the route record. All nodes along the reply packet update their route cash.

Route maintenance is ensured through the use of route error packets and acknowledgments. On an encounter of a fatal transmission problem (on the data ling layer) route error packet is created. When a node receives an error packet it removes the hop in error from its route cache and truncate at that point all routes containing the hope. Acknowledgments are used to verify existing links. An example of DSR is shown on figure 2.10.

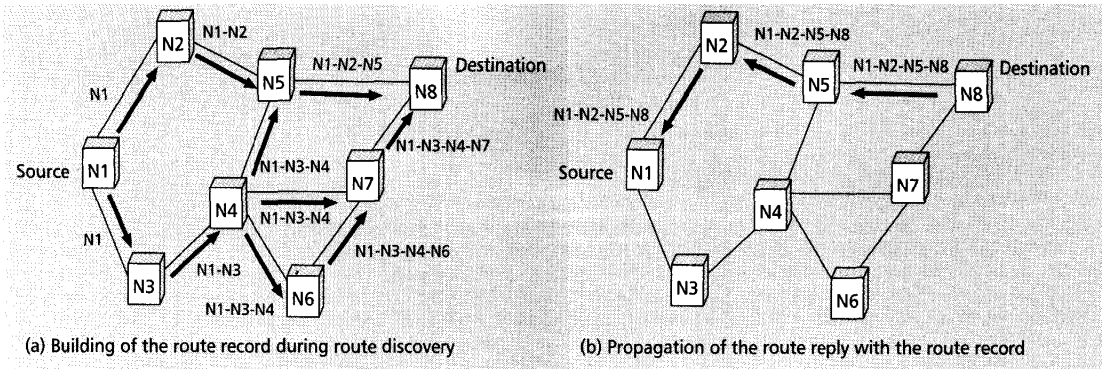


Figure 2.10 Creation of a route record in DSR (from [Roy99])

2.2.3.3 Temporally-Ordered Routing Algorithm

The Temporally-Ordered Routing Algorithm [Par97] was developed by Vincent Park at the University of Maryland and the Naval Research Laboratory. Park has patented his work, and it was licensed by Nova Engineering, who are marketing a wireless router product based on Parks algorithm.

TORA is proposed for highly dynamic mobile, multihop wireless networks. It is a source-initiated on-demand routing protocol. It finds multiple routes from a source node to a destination node. Its main feature is that the control messages are localized to a very small set of nodes near the occurrence of a topological change. To achieve this, the nodes maintain routing information about adjacent nodes. The

protocol has three basic functions: Route creation, Route maintenance, and Route erasure.

TORA does not use a shortest path solution, an approach which is unusual for routing algorithms of this type. It builds and maintains a directed acyclic graph rooted at a destination. No three nodes may have the same height. Information may flow from nodes with higher heights to nodes with lower heights. Information can therefore be thought of as a fluid that may only flow downhill. By maintaining a set of totally-ordered heights at all times, TORA achieves loop-free multipath routing, as information cannot 'flow uphill' and so cross back on itself. En example of the directed acyclic graph is shown on figure 2.11.

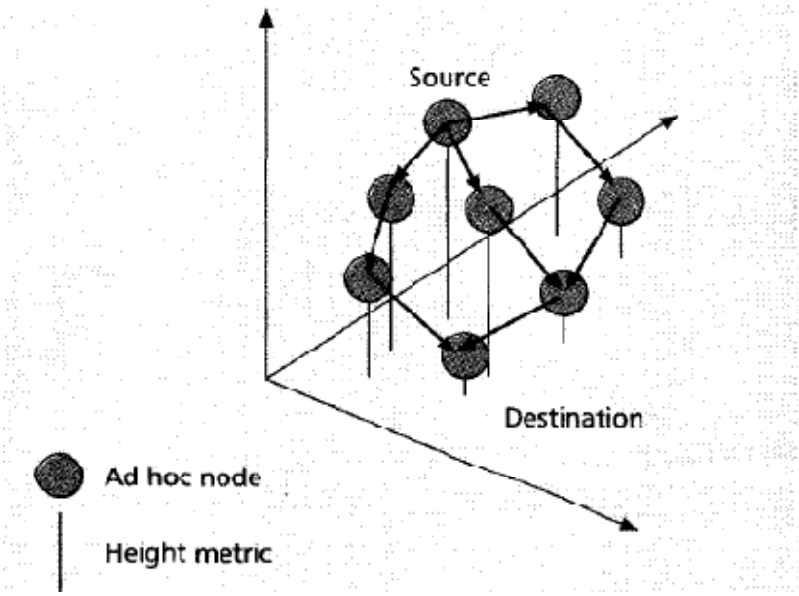


Figure 2.11 TORAs directed acyclic graph (from [Roy99])

2.2.4 Hybrid routing

Hybrid routing protocols try to combine all the advantages of reactive and proactive protocols. The basic idea is to use both protocols but in different areas of the network. The proactive routing could be restricted to a small domain ‘around’ the node while the reactive routing would be used for the rest of the network. En example of hybrid routing protocol is: Cornell's Zone Routing Protocol (ZRP).

3 SECURITY ISSUES IN AD-HOC NETWORKS

3.1 Introduction

Security is an essential for both wired and wireless networks and are well addressed. However the specific characteristics of mobile ad-hoc networks can cause the security issues to be much more complicated. MANET is a collection of mobile nodes that form a network without any fixed infrastructure that could be responsible for some centralized administration. The nodes have to rely on each other making any attack from 'inside' the network much easier than in other networks. Also the wireless is much less reliable in the sense of security then its wired equivalent.

3.2 Classifications

There are many ways in which an ad-hoc network can be attacked. The basic classification of attacks against MANET (and also other networks) divides attacks into two groups: passive and active. Passive attacks are mostly targeted against confidentiality, malicious nodes can obtain information about the data exchanged within the network. Due to the wireless connection simple eavesdropping is impossible to detect therefore encryption algorithms are used to hide the information (there still exists a possibility to gather information for traffic analysis by the malicious node). Active attacks on the other hand can violate the availability, integrity or authentication by deleting, modifying, or injecting erroneous messages. Some example security threats are shown on figure 3.1.

Another classification is made based on the 'source' of the attack. Namely the attacker can be either attacking from outside the network – external attack, or from inside the network – internal attack. The latter ones are much more severe because the malicious node is a trusted member of the network and can posses some privileged access rights.

We can also make an attack classification based on the network protocol stack [Bin06] as shown on figure 3.2.

In further work we will concentrate on active, internal attacks on network layer.

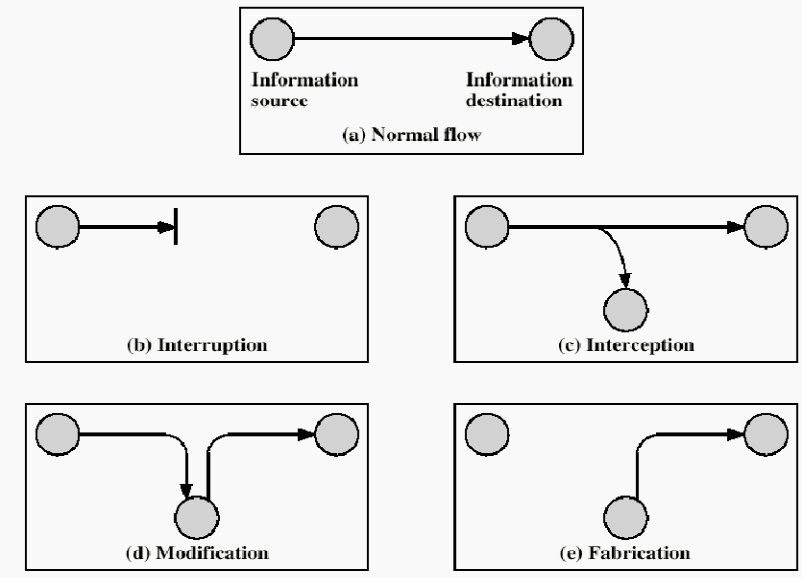


Figure 3.1 Example security threats (from [Erl07])

Layer	Attacks
Application layer	Repudiation, data corruption
Transport layer	Session hijacking, SYN flooding
Network layer	Wormhole, blackhole, Byzantine, flooding, resource consumption, location disclosure attacks
Data link layer	Traffic analysis, monitoring, disruption MAC (802.11), WEP weakness
Physical layer	Jamming, interceptions, eavesdropping
Multi-layer attacks	DoS, impersonation, replay, man-in-the-middle

Figure 3.2 Security threats according to network protocol stack (from [Bin06])

3.3 Attacks on the network layer

There are many different attacks on the network layer depending on the used routing algorithm. Most of the basic attacks can be divided into three groups [Bin06], namely:

Attacks at the routing discovery phase in which the malicious node will launch the attack by not following the routing protocol requirements. Example attacks are: routing table overflow attack or routing cache poisoning attack.

Attacks at the routing maintenance phase which can be launched against a routing protocol that implements maintenance procedures by sending false control messages.

Attacks at data forwarding phase. In this case malicious nodes cooperate correctly during the routing discovery and maintenance phases but during the data forwarding phase they either drop the packet or forward it incorrectly.

There are also some more complicated attacks like wormhole attack, blackhole attack or DoS attack that will be described further on.

3.3.1 Wormhole Attack

In physics, a wormhole is a hypothetical topological feature of space/time that is basically a 'shortcut' through space and time. A wormhole has at least two mouths which are connected to a single throat or tube. If the wormhole is traversable, matter can 'travel' from one mouth to the other by passing through the throat.

In MANET wormhole is a term adopted to describe an attack against the routing protocol in which two cooperating malicious nodes create a ‘tunnel’ between two points of the network [Hu03]. The attack is possible even if none hosts were compromised and the attacked network introduced strong authentication and encryption algorithms. The wormhole is created using covert communication mechanism, it can be established using a long range, directional wireless connection or through a wired link. When using this method the attacker can ‘convince’ two distant nodes that they are only one hop away.

The wormhole can transmit any packets it ‘hears’ regardless to whom they were addressed. In this way it cheats the routing protocol and probably will attract traffic from distant parts of the network, creating a bottleneck fully controlled by the malicious nodes. That gives the attacker a strong position to manipulate the traffic in the future. He can selectively drop packages, modify or fabricate them jeopardizing the network.

The wormhole also disrupts the network's perception of its topology that can be an important aspect in tactical or sensor networks. An example of wormhole is shown on figure 3.3.

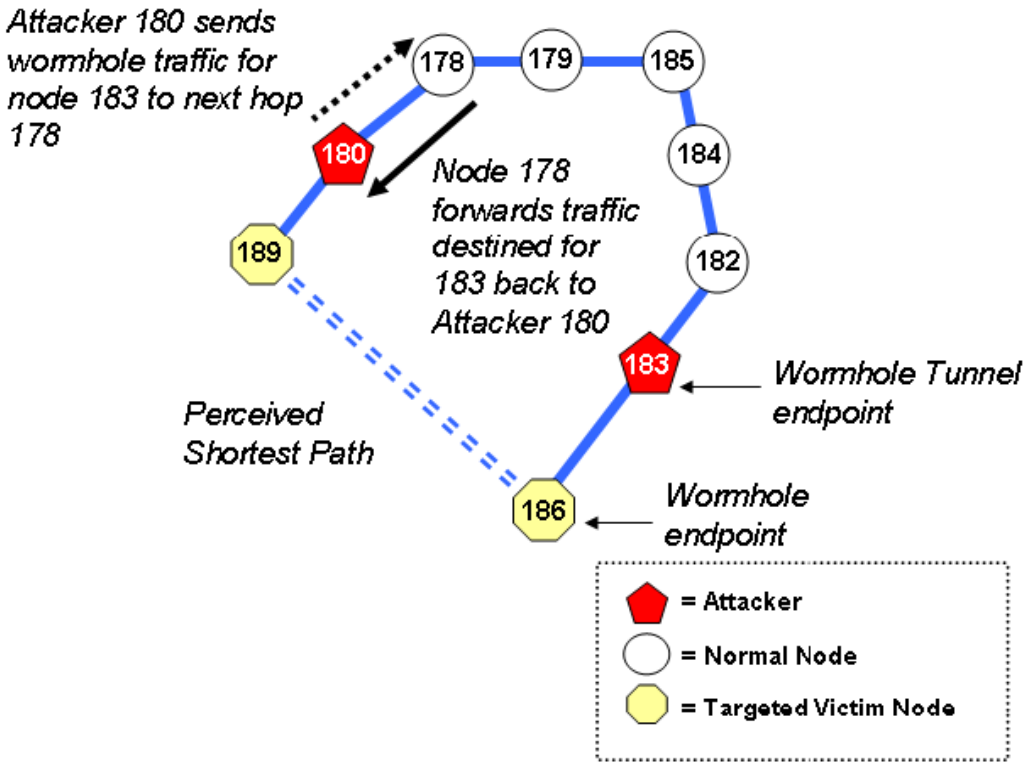


Figure 3.3 Wormhole attack (from [Bar07])

3.3.2 Blackhole Attack

Using astronomic terms a black hole is a region of space in which the gravitational field is so powerful that nothing, not even light, can escape its pull.

In MANET a blackhole (or sinkhole) is an attack against the routing protocol [Tam07]. It is a denial of service attack in which the malicious node trays to attract as many packets as possible and then simply drops them.

Unlike the wormhole attack, blackhole attack requires only one malicious node that is authenticated within the attacked network.

During the route discovery phase malicious node advertises itself as having a valid, direct (or short in term of hops) path to the destination. That causes the sending nodes to 'believe' in false routing paths and sending most (if not all) of the packages through the malicious node. The attacker then simply drops all the packets without any forwarding.

In some cases it is also possible for the malicious node to selectively forward packets making the detection of the attack harder. An example of blackhole attack is shown on figure 3.4.

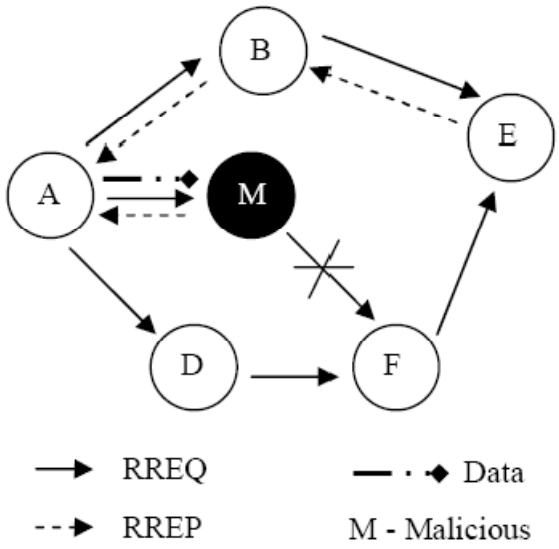


Figure 3.4 Blackhole attack in AODV (from [Tam07])

3.3.3 Flooding Attack

As described in [Pin05] the ad hoc flooding attack is a denial of service attack targeted against networks using reactive routing protocols. Unlike two previous mentioned attacks id does not target the routing protocol directly but the network resources making it incapable of correct behavior.

We can distinguish two kinds of flooding attack. First is the RREQ flooding attack. It ignores the network limitations for sending RREQ messages and sends a large number of RREQ packages with a maximum TTL value addressing nodes that do not exist in the network (or random addresses if the malicious node do not have enough knowledge about the network). In that way the network will be full with invalid RREQ messages disallowing the nodes to handle the proper ones.

The second is called DATA flooding attack. Here the malicious node firstly sets up paths to all nodes in the networks and then sends large volumes of useless

DATA packets to all nodes along these paths. Depleting in this way the available network bandwidth. Both of the attacks consumes the available network disallowing other nodes to communicate correctly. An example of RREQ flooding attack is shown on figure 3.5.

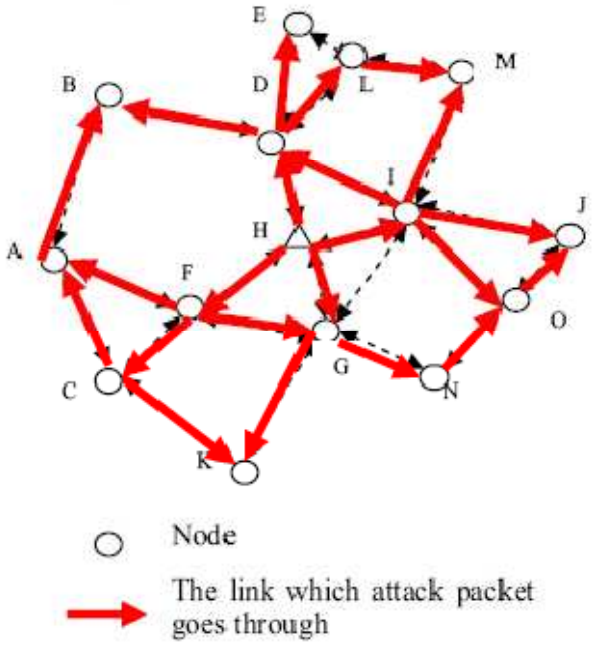


Figure 3.5 RREQ flooding attack (from [Pin05])

4 INTRUSION DETECTION SYSTEMS

4.1 History

Concept of intrusion detection systems has been around for nearly twenty years but only recently has it seen a dramatic rise in popularity. I started in 1980, with James Anderson's paper, *Computer Security Threat Monitoring and Surveillance*, where the notion of intrusion detection was created. The paper, was written for a government organization. It introduced the notion that audit trails contained vital information that could be valuable in tracking misuse and understanding user behavior. With the release of this paper, the concept of "detecting" specific user events emerged. Anderson's work also provided the foundation for future intrusion detection system design and development. His work was the start of host-based intrusion detection and IDS in general.

Next key date was 1983 when Dr. Dorothy Denning, began working on a government project that launched a new effort into intrusion detection development. Their goal was to analyze audit trails from government mainframe computers and create profiles of users based upon their activities. One year later, Dr. Denning helped to develop the first model for intrusion detection, the Intrusion Detection Expert System (IDES), which provided the foundation for the IDS technology development that was soon to follow. Using her research and development work at SRI, Dr. Denning published the decisive work, *An Intrusion Detection Model*, which revealed the necessary information for commercial intrusion detection system development. Her paper is the basis for most of the work in IDS that followed.

Since then interest within the field of intrusion detection system constantly grew. What started as a government research was mostly developed by commercial companies like Haystack Labs, SAIC or Cisco. Some of the more important dates for IDS systems are shown on figure 4.1.

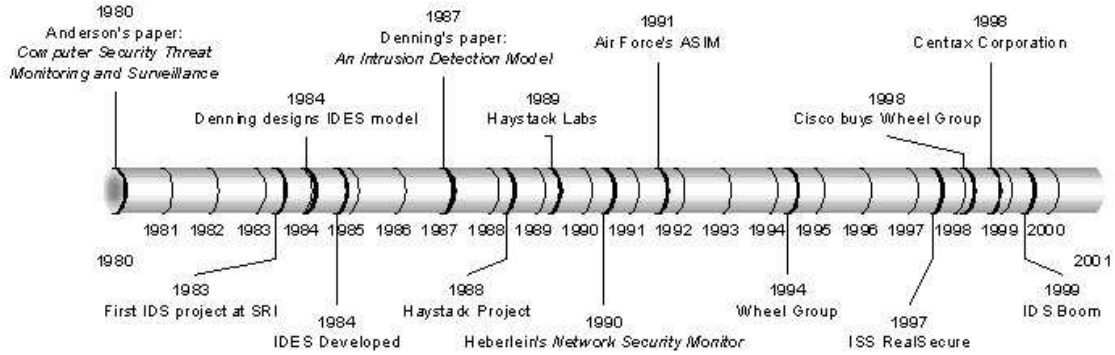


Figure 4.1 IDS history (from www.securityfocus.com, 19.01.2008)

4.2 Overview

Intrusion Detection System is defined as a software or hardware system that tries to detect and then log (or notify the system operator) inappropriate, incorrect, or anomalous activity within a given domain. The domain can be either a single host or an entire network. In other words, intrusion detection is a process of identifying malicious activity targeted at computing and networking resources.

It is also possible for IDS tools to distinguishing between attacks started from inside the network and external ones. Unlike some other systems (firewalls) IDSs act only if an intrusion has occurred and a node or network has been compromised. Because of this IDSs are usually called the second line of defense.

According to [Sca07] key functions of an IDS are:

- monitoring
- analyzing events
- recording information related to observed events
- notifying administrators of important observed events
- producing reports

Also worth mentioning are Intrusion Prevention Systems (IPS) often combined with IDS forming Intrusion Detection and Prevention Systems (IDPS). Unlike its predecessor IPS can also respond to a detected threat.

Depending on the detection mechanism, intrusion detection systems can be classified [Mis04] into three categories: signature based detection systems, anomaly based detection systems and specification based detection systems.

4.2.1 Signature based detection systems

In signature based detection system (also called misuse based) the detection process is based on comparing monitored events to known definitions (models) of hostile activities. Examples of the activities can be:

- sending a TCP SYN to different ports
- trying to logon with root privileges

It is a similar approach as in most virus scanners. This kind of detection can be very efficient against known attacks and will never (or rarely) trigger a false alarm. However it is dependent on receiving regular updates for the pattern base and will be unable to detect unknown earlier threats or new versions of known ones.

4.2.2 Anomaly based detection system

In anomaly based detection system the detection process is based on classifying network (system) activity either as normal or anomalous. The classification is based on heuristics or rules, rather than patterns or signatures, and will detect any type of misuse that falls out of normal system operation.

To perform this the system first needs to be learned about the normal (safe) behavior of the network. It is called the calibration phase in which the system gathers information which can be used to build predefined (safe) profiles for the monitored activities. In some cases it is also possible for the system to learn ‘on the way’ by adjusting the profiles during the system runtime.

Unlike the signature based systems it can detect previous unknown threats but will probably rise more false positive – anomalous activities that are not intrusive, or false negatives – intrusive activities that do not cause the network to 'act' differently.

4.2.3 Specification based detection system

In specification based detection systems each program or protocol that is activated on the system is described by a set of constraints that describe its correct operations. The definitions (models) are created basing on vendor documentation or process/protocol specification.

The system is responsible for monitoring the processes and matching the actual data with the program (protocol) model and alerting in case of any anomalous behavior.

Similar to anomaly based detection proper behavior models have to be maintained and updated whenever any changes to any of the monitored programs are made. It will also be able to detect earlier unknown attacks and should have less number of false positives then the anomaly based approach.

Specification based detection is also the most resource consuming among mentioned system. Because of that it is also hard in implementing with MANET.

4.3 Intrusion Detection Systems for MANET

When writing about intrusion detection systems for wired networks it is possible to refer to some existing systems like SNORT NIDS, Untangle or Bro NIDS. On the other hand, when concentrating on wireless ad-hoc network most work is still to be done. There are some ideas for adjusting some existing ideas (most of them ineffective), but much more work is done in defining and solving new problems caused by the 'new' networking environment. Some of them tested with different types of simulators, but yet there are very few systems that were sufficiently documented, tested and published. That's why this chapter will present only some of the more promising and interesting IDS ideas and architectures.

4.3.1 New vulnerabilities

First and probably most important is the usage of wireless link. Mobile nodes sending data 'into the open' are much more susceptible to eavesdropping or interfering. Unlike in the wired networks the attacker does not need to get access to the network bypassing gateways or firewalls the wireless can be attacked by anyone within the communication range.

Secondly, many MANET protocols do not have any equivalent in wired networks that can be a potential week point. A good example are routing protocols used with ad-hoc networks.

Third, internal attacks. Each node is an independent unit which reacts with others. In a network it is possible for a node to be captured by the attacker who can then become a threat from within the network. Because of this nodes within the network cannot fully 'trust' each other.

Fourth, cooperation. Mobile ad-hoc networks are self organizing structures. They do not (and cannot) depend on any centralized authority. This decentralization

forces them to cooperate while carrying out their tasks which creates a serious vulnerability, especially regarding the previous point.

Fifth, others. There are also some specific requirements and parameters in mobile networks that do not concern (or are much less concern) the wired ones. Such as limited bandwidth, battery power or slower links. They may not directly affect the security issues but can cause some restrictions for the proposed solutions.

4.3.2 Intrusion Detection System - Architecture

Architecture presented below was proposed by Zhang et al. in [Zha04].

The main idea for the architecture is that the Intrusion Detection System should be both distributed and cooperative. Every node within the network can participate in detecting any malicious behavior by locally monitoring data for any signs of intrusion. It is also possible for neighboring nodes to collaboratively investigate specific situations. The overall architecture is shown on figure 4.2.

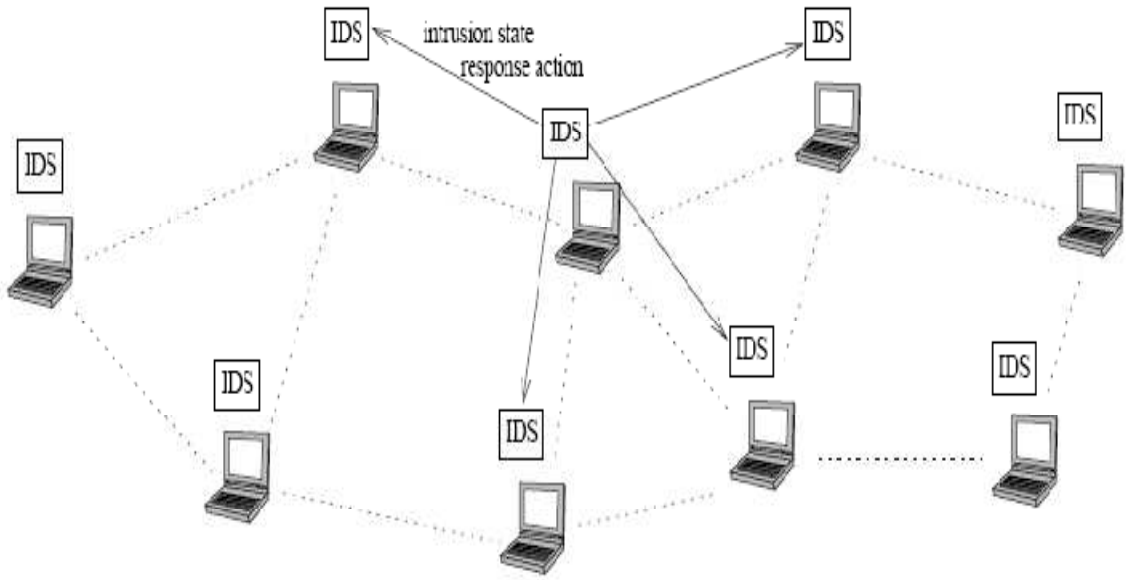


Figure 4.2 The IDS Architecture for Wireless Ad-Hoc Networks (from [Zha04])

Each of the IDS agents placed within the mobile node works independently from others. It is able to monitor user and system activities as well as any communication within radio range, starting a global intrusion action (for example re-authentication) or starting a cooperative intrusion detection with neighbor nodes. Conceptually an agent consists of six elements shown on figure 4.3.

Intrusion Detection Using Autonomous Agents

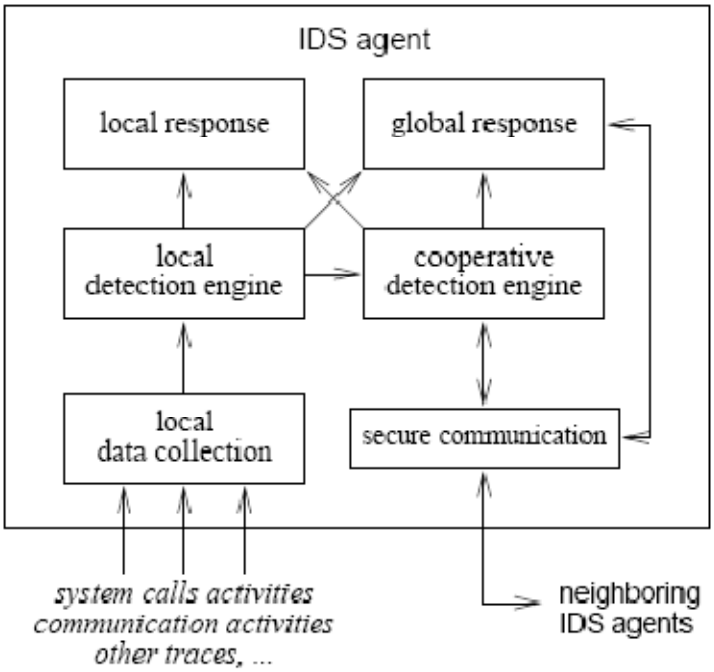


Figure 4.3 IDS Agent model (from [Zha04])

Local data collection module is responsible for gathering streams of real-time information from different sources. The nature of the data collected depends on the intrusion detection algorithm and as mentioned earlier can include system and user activities, communication within radio range and others. It is possible for multiple data collection modules within one agent (i.e. for multi-layer intrusion detection).

Local detection engine is responsible for detecting intrusions based on the local data. It can be any kind of anomaly based or signature based detection algorithm, but because of possible limitations of memory and increasing number of new attacks types a anomaly based system would be favored.

Local response and global response modules are responsible for generating actions in case of detecting any malicious activity. Generated response will depend on the malicious activity type as well as on network protocols used.

Cooperative detection engine is responsible for actions that will allow to confirm (or deny) existence of malicious activity that was detected by one of the nodes but with week confidence. The distributed algorithm could be for example a simple majority-based voting algorithm in with each of the nodes propagate locally evaluated likelihood of detected anomaly to its neighbors.

Secure communication is required for the intrusion detection agents to exchange information about possible threats in highly-confident communication channel.

4.3.3 Watchdog

Mobile ad-hoc networks require all nodes to cooperate witch each other to provide satisfying quality of services. In military applications this can be forced as all nodes share the same goal and are subordinated to some authority. However in civilian applications (like conference or network of cars), nodes can have independent goals. In such networks, forwarding packets for other nodes that will

consume node resources (like power) is not in their direct interest. It may cause a node to misbehave and act selfishly. A node my act like this because of different reasons. It is possible for the node to be overloaded, in such case it will have not enough CPU cycles, available network bandwidth or other resources needed to forward packages. It can also be configured to act selfish (for example when the battery level is low) and will not willingly 'share' it's resources still requiring other nodes to do so. A broken nod will also fail to cooperate properly. Finally a node can try to sabotage the network by some malicious activities like dropping packages when applying black hole attack.

Independently of the reasons, such nodes should be detected and excluded from the network. One of the possible techniques is called Watchdog [Dje05]. The idea behind this technique is quite simple. Whenever a node sends (or forwards) a packet, watchdog connected to that node will check if the next node in the packet path also forwards it. It can do this by listening the next node transmissions.

To be more exact: whenever node A forwards packet to node B and requires it to then forward it to node C, the watchdog connected to node A adds this packet to its buffer. Then it monitors all traffic within nodes A radio range and compare each overheard packet with those in its buffer. If there is a match, packet is assumed to be forwarded and is removed from the buffer. On the other hand, if a packet stays in the buffer for more than some given time it is assumed to be dropped and node B to misbehave and proper register is updated. If the register concerning node B reaches a certain threshold then it is considered as malicious [Dje05].

The drawbacks of watchdog are that it will not detect malicious activities in case when the attacker drops packages in a partial manner so the threshold will not be reached. It also cannot be used in networks that use power aware routing protocol because of the possibility of rising many false alarms in situation when node B uses less power to transmit the packet to node C then is needed for node A to receive it as shown on figure 4.4.

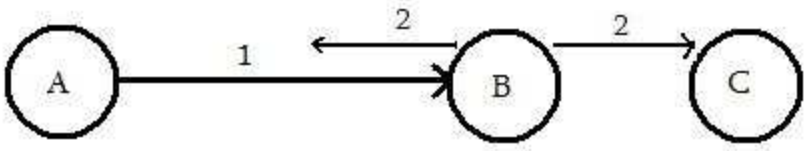


Figure 4.4 Forwarding with power aware routing protocol.

4.3.4 Anomaly detection

This intrusion detection technique was proposed together with the secure architecture in [Zha04] and is based on the method proposed by Yoshinori Okazaki at el. in [Oka02]. The basic assumption of this approach is that there are some observable characteristics of normal behaving network that are distinct from those in the network with misbehaving (malicious) nodes. Those characteristics can be used to build a basic model of the network and then used together with some classification methods detect any abnormalities.

The overall procedure of the method would be:

a) select network characteristics

b) produce sets of those characteristics for normal network behaviour
c) perform appropriate data transformations
d) create a basic profile based on the data
e) chose a way to calculate differences between the basic profile and sample data
f) apply to test data

If, as in [Zha04], we would want to use classifiers then steps d and e would be changed to first chose the classifier and second learn the classifier using training data. If not, it is possible to apply any distance function and define the limit above which the network will be considered as attacked. Function proposed in [Oka02] is one of the definitions used in speech recognition and is as follows:

Let $A = a_1, a_2, ..., a_i, ..., a_I$ be the definition of the basic profile, and $B = b_1, b_2, ..., b_j, ..., b_J$ will define a test sample then distance between A and B D(A,B), is calculated according to:

$$D(A,B) = \begin{cases} \sqrt{\sum_{k=1}^{I}(a_k - b_k)^2 + \sum_{k=I+1}^{J}(b_k)^2} \; for(I < J) \\ \sqrt{\sum_{k=1}^{I}(a_k - b_k)^2} \; for(I = J) \\ \sqrt{\sum_{k=1}^{J}(a_k - b_k)^2 + \sum_{k=J+1}^{I}(b_k)^2} \; for(I > J) \end{cases}$$

5 SIMULATIONS

5.1 Introduction

Every time a new idea concerning mobile ad-hoc networks arises there is also the need to test its influence on other MANET elements. A good way to do this is through simulations. They may not be as exact as real life situations but allows some initial verification.

Simulations presented below in the most part concentrate on differences between different mobility scenarios as well as a comparison of AGM scenario build with different parameters. Presented results should show possible similarities or differences between them and should help in further associating them with some real scenarios.

Additionally, the simulations allow to test the applications, that are also a part of this work, and allow the author to make any necessary improvements.

5.2 Simulator

There are many different network simulators available. Some of the more popular are: OMNET++, NetSim, QualNet or NS-2. The last one, NS-2 (NS stands for Network Simulator) is one of the most commonly used mainly because of its extensibility (open source) and very good support over the internet – online documentation, discussion groups, forums.

NS-2 is a discrete event simulator targeted at networking research. It first appeared in 1989 as a variant of the REAL network simulator. In 1995 Defense Advanced Research Projects Agency (DARPA) has supported NS development within the VINT project.

NS is a simulator based (like many others) on two parts/languages. Namely an object oriented simulator written in C++ and a OTcl interpreter used to execute command scripts. Using the OTcl, user can define network topology, chose protocols to be used or the output form of the simulators result. C++ is used to create protocols implementation.

All simulations in this chapter are created using NS-2, supporting programs added to NS (in all-in-one version like *calcdest*) and applications created by the author for the needs of script generation and interpreting output files.

5.3 Mobility

In this and all other simulations there are few assumptions. First the number of nodes simulated is 36. Using to many nodes can make the scenario files (processed by the *calcdest)* size 35MB and bigger. Processing such files can be very time consuming. Next thing is the map, it is assumed to be a square area sized 1000x1000 units (meters). Simulation length is set to 500 sec, the velocity of a node is a random variable (uniform distribution) with minimum value of 5 m/s and maximum value of 30 m/s.

5.3.1 Mobility scenarios

For the simulations six different mobility scenarios were created. They are created with *MovementsPatternsPro* application (described later) using different mobility models.

All the visualizations were created using *nam.* Nam is provided by the all-in-one version of *ns-2* and is used to visualize the simulation results. However when simulating wireless network it can only show the nodes movement.

5.3.1.1 AGMNoGroups

This scenario was generated by setting the *Groups* parameter to 0 in the AGM generation tab. Rest of the parameters are set as follows:

- Step length = 10
- h (repulsive factor) = 1
- g (distance factor) = 1
- r (range) = 250

In this scenario all nodes are treated as individuals that try to avoid others. Thanks to this they are evenly spread through the simulation area.

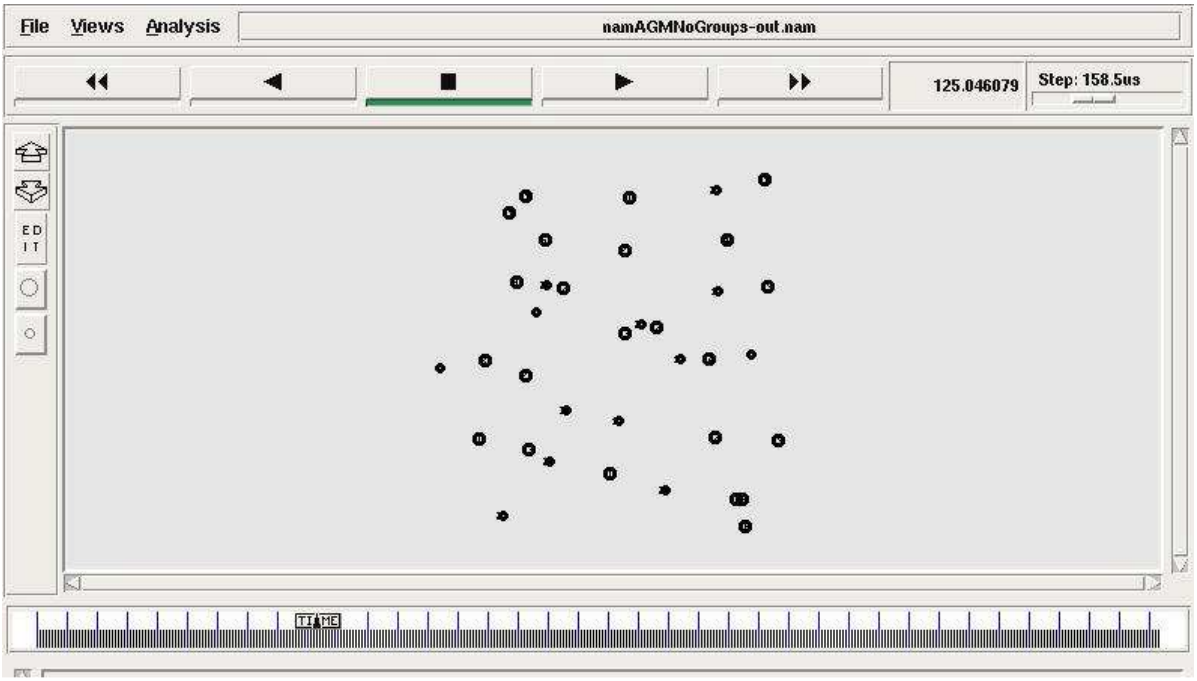


Figure 5.1 AGMNoGroups visualization in *nam*

5.3.1.2 AGM1Group

This scenario was generated by setting the *Groups* parameter to 1 in the AGM generation tab. Rest of the parameters are set as follows:

- Step length = 10
- h (repulsive factor) = 1

- g (distance factor) = 1
- r (range) = 250

In this scenario all nodes are members of one group. Because of this the repulsive factor become an attraction factor. Nodes instead of avoiding each other try to stay as close as possible. After a short time a center point around which nodes gather is created. Due to the possible maximum velocity and the step values it is possible for the nodes to 'move away' but within the next step they go back. An interesting thing is that the center was formed out of one nodes range leaving it 'alone' at the beginning of the simulation. Thanks to the random factor in AGM it moves around some part of the simulation area and finally joins the rest.

In the second half of the simulation movements of the center can be clearly observed.

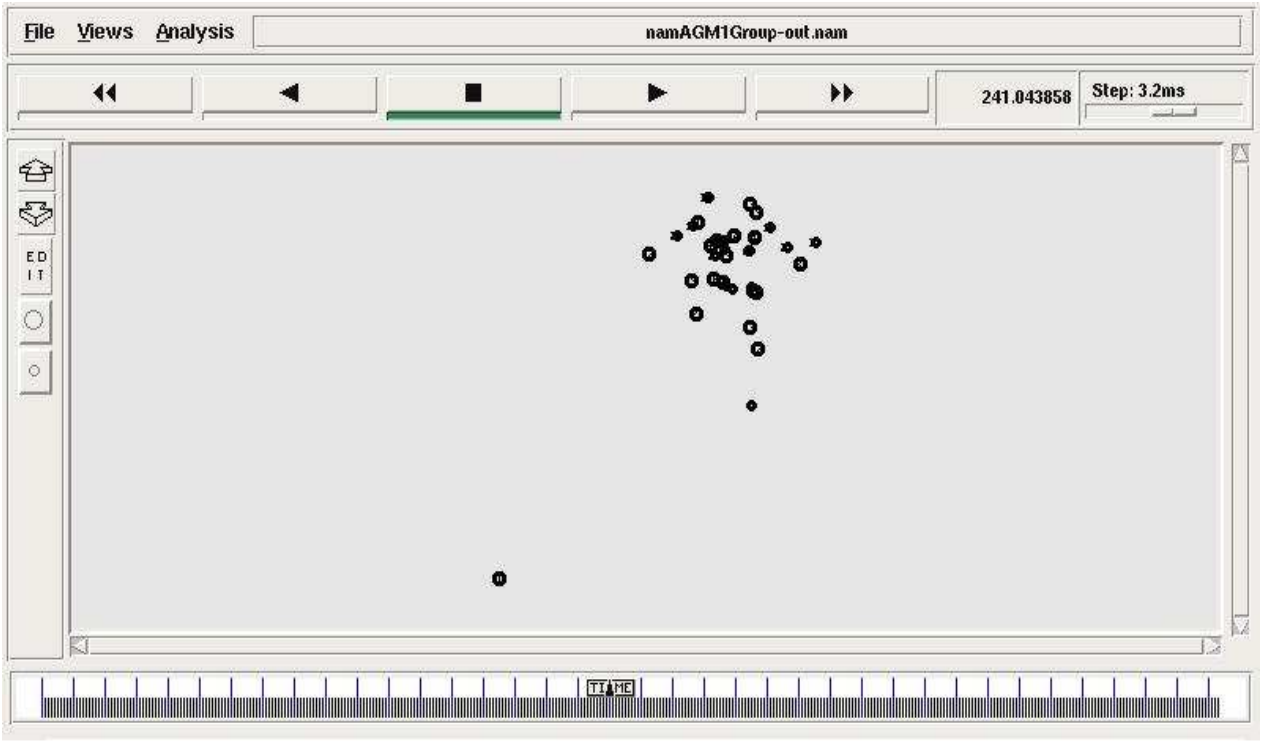


Figure 5.2 AGM1Group visualization in *nam*

5.3.1.3 AGM3Groups

This scenario was generated by setting the *Groups* parameter to 3 in the AGM generation tab. Rest of the parameters are set as follows:

- Step length = 10
- h (repulsive factor) = 1
- g (distance factor) = 1
- r (range) = 250

In this scenario all nodes are evenly divided into three groups. Each node will be attracted by other nodes from the same group but it will also try to avoid nodes from other groups. The repulsive force of nodes from other groups is two times stronger than the attractive force from nodes of the same group.

During this simulation nodes often group themselves for short periods of time. It is impossible to distinguish three independent groups, during the whole simulation

time there are always some not grouped nodes that soon break apart the existing groups. Worth mentioning is that when using only two groups nodes created two independent clusters at about half of the simulation time.

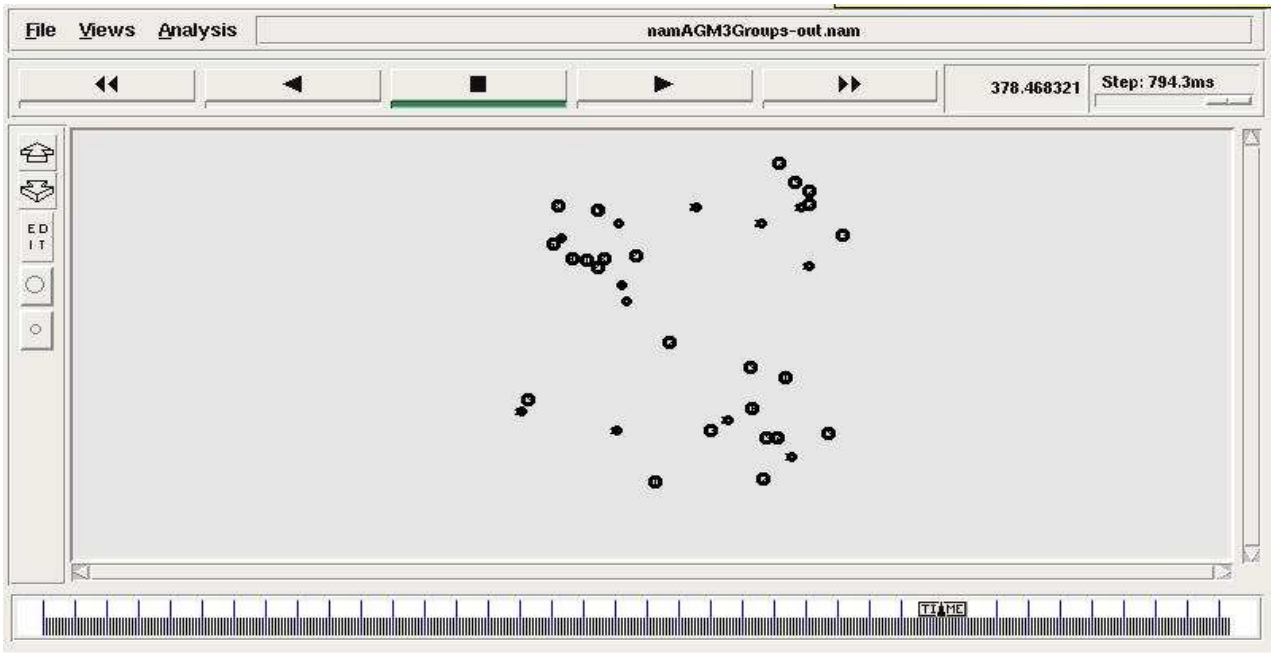


Figure 5.3 AGM3Group visualization in *nam*

5.3.1.4 Random

RandomDirection was generated using by with the *Sleep time* parameter set to 0 in the *Random direction* tab. RandomWaypoint (*Random waypoint* tab) *Sleep time* parameter also was set to 0, and RandomWalker (*Random walker* tab) *Step* parameter was set to 10 (like in the AGM simulations).

Nodes in all of the three scenarios behaved as expected. Interesting thing is that in the RandomWaypoint the nodes often concentrate more on the center part of the simulation area and in the RandomDirection they tend to 'stay close' to the borders.

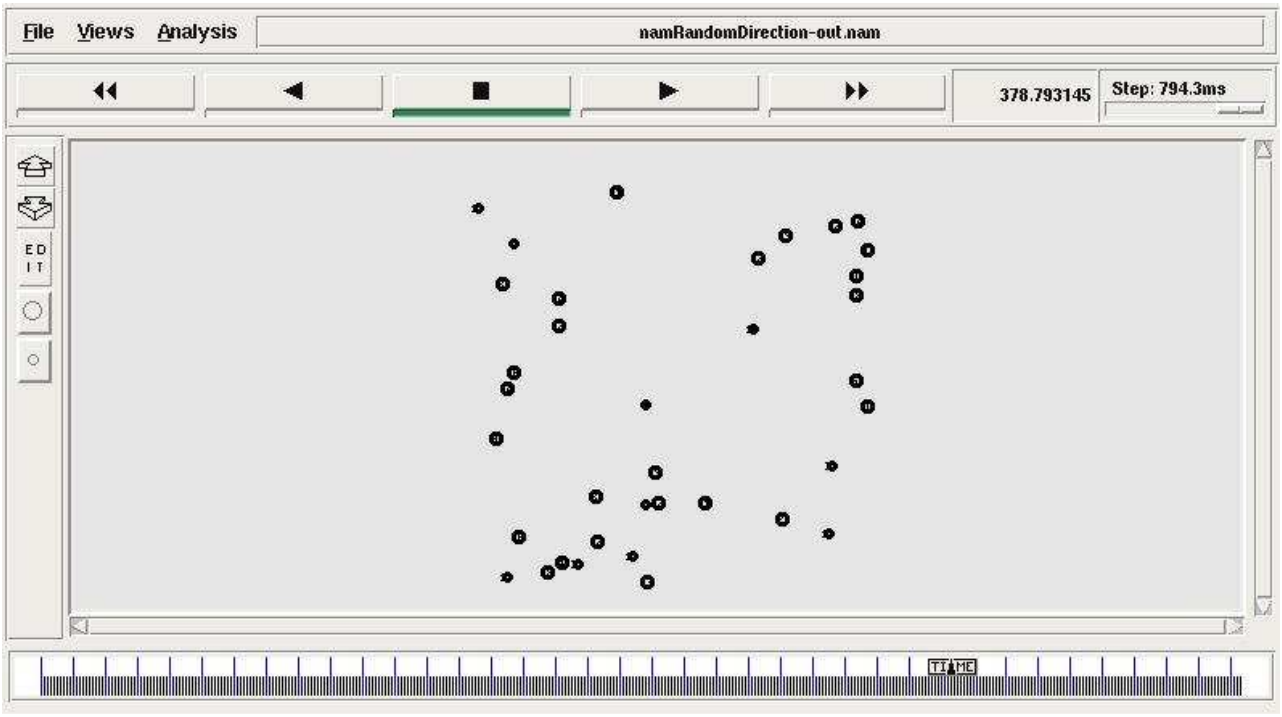


Figure 5.4 RandomDirection visualization in *nam*

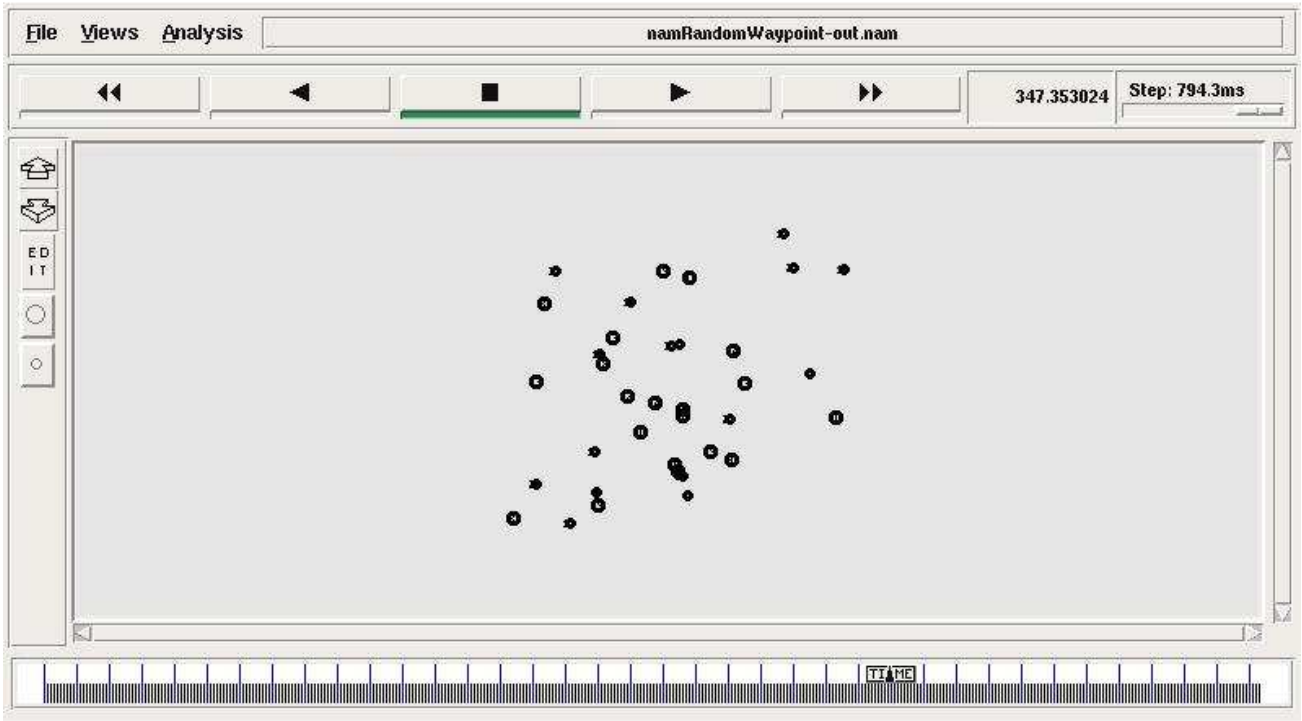


Figure 5.5 RandomWaypoint visualization in *nam*

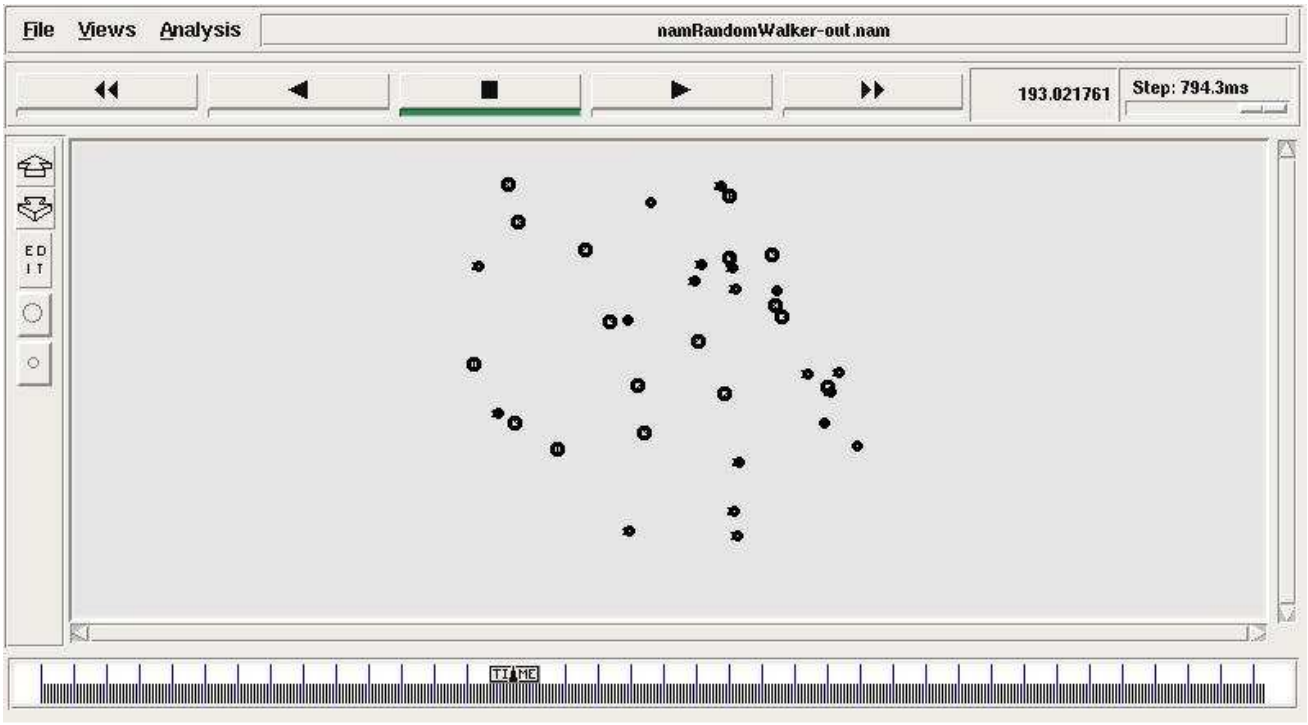


Figure 5.6 RandomWalker visualization in *nam*

5.4 Calc file analysis

First data after generation of movement files can be gathered from analyzing the calc files. They can be created out of the movement files (generated by *MovementsPatternsPro*) using the calcDesc program provided by the all-in-one version of *ns-2*. The calc files are then analyzed by the *Calc File Analyzer (CFAPro)* application. The CFAPro output files are written in Comma Separated Values (csv) format, and can be easily processed using excel.

The overall statistics are presented in table 5.1. All presented values are the average per second data for one node during the simulation time. Neighbors count stands for the number of nodes that are within one hop of the specific node. Partitioning is defined as a proportion of node pairs between no paths exist to the number off all node pairs. Standard deviation values of corresponding elements from table 5.1 are presented in table 5.2

Mobility scenario	velocity	neighbors count	number of hops	number of unconnected node pairs	partitioning
AGM1Group	22,3982	25,6796	1,1767	24,1480	0,0383
AGM3Groups	20,8902	5,9902	3,0476	161,4980	0,2563
AGMNoGroups	21,3947	5,7244	3,0306	44,0900	0,0700
RandomWalker	12,4317	6,5483	2,8056	67,0120	0,1064
RandomDirection	11,5423	4,9177	3,6869	98,5140	0,1564
RandomWaypoint	12,4829	7,7676	2,5563	43,7500	0,0694

Table 5.1 Average values of simulation parameters

Mobility scenario	velocity	neighbors count	number of hops	number of unconnected node pairs
AGM1Group	3,2789	6,0127	0,4125	43,1960
AGM3Groups	3,1148	1,2518	0,4005	85,3061
AGMNoGroups	3,1653	0,6057	0,2578	29,4200
RandomWalker	2,0573	1,2015	0,3815	38,1514
RandomDirection	2,4864	0,6707	0,5336	86,3417
RandomWaypoint	2,1746	1,3240	0,2721	35,0078

Table 5.2 Standard deviation values of simulation parameters

While reading tables 5.1 and 5.2 two interesting things can be observed. First is the fact that all AGM type scenarios have much higher average velocity then the rest of the scenarios. This is probably caused by the way velocity is obtained in AGM. Namely that it depends on the length of movement vector and as so can more often reach the maximum value.

Second thing is the high standard deviation value of the unconnected nodes pairs in AGM1Group scenario regarding its average value. Additionally through most of the simulation there is only one node 'outside' the group and during the remaining part all are groped so it would be expected for the standard deviation to have the value at about 17. This situation can be only explained when looking at some more detailed data.

At the beginning of the AGM1Group simulation some nodes are placed randomly over the simulation area. During the first seconds some of the nodes are groped away from the main group and so causing the number of unconnected pairs to be high (up to 302) and in the overview causing the standard deviation to be higher than expected.

5.4.1 Detailed analysis

Additionally to the overview data the CFAPro application can generate some more detailed information. It divides the simulation into parts of ten seconds length. For each of this parts average values are calculated.

5.4.1.1 Velocity

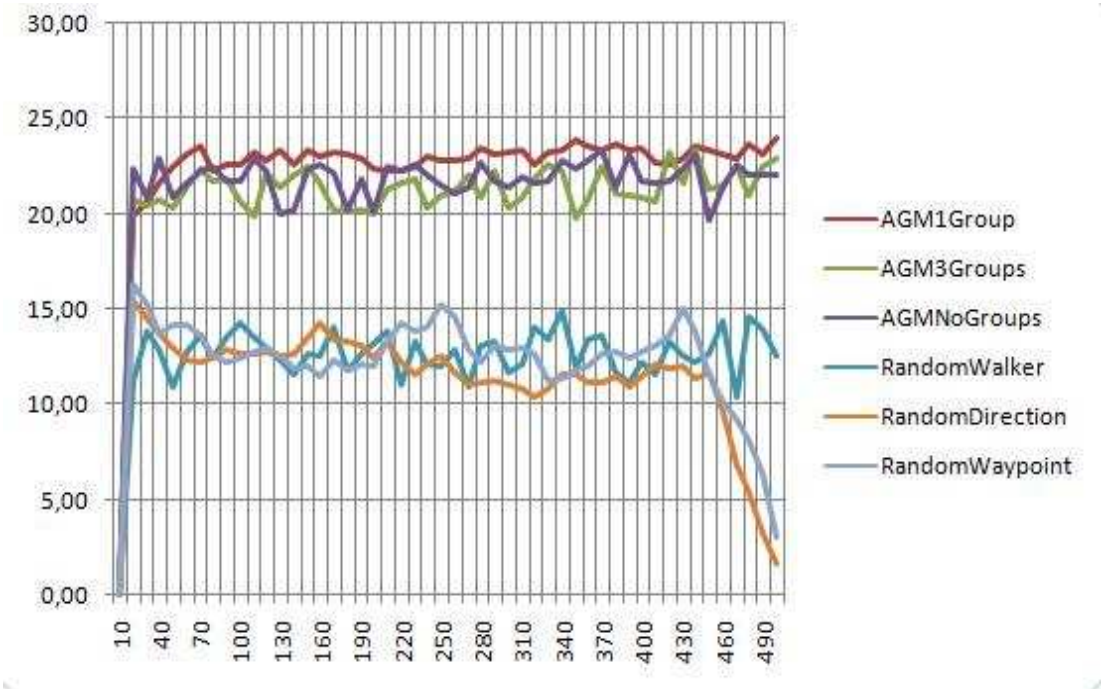


Figure 5.7 Average velocity

At the beginning of the simulation all nodes start from 0 velocity. As was earlier observed the AGM scenarios have a higher value of velocity. Also it can be observed that the AGM1Group and RandomDirection have much smoother directions then the others. This may be an effect of the gathering in AGM and the value of movement vector. In RandomDirection nodes move from one map edge to another with constant velocity and change it after reaching their destination. Because of this they change it less often than in the other scenarios.

In RandomDirection and RandomWaypoint scenarios, nodes seem to slow down at the end of the simulation. This may be an effect caused by differences in selecting new destinations and velocities in those patterns (they do not use steps). It is also possible for the applications to read the data from the last ten seconds incorrectly.

5.4.1.2 Number of neighbors

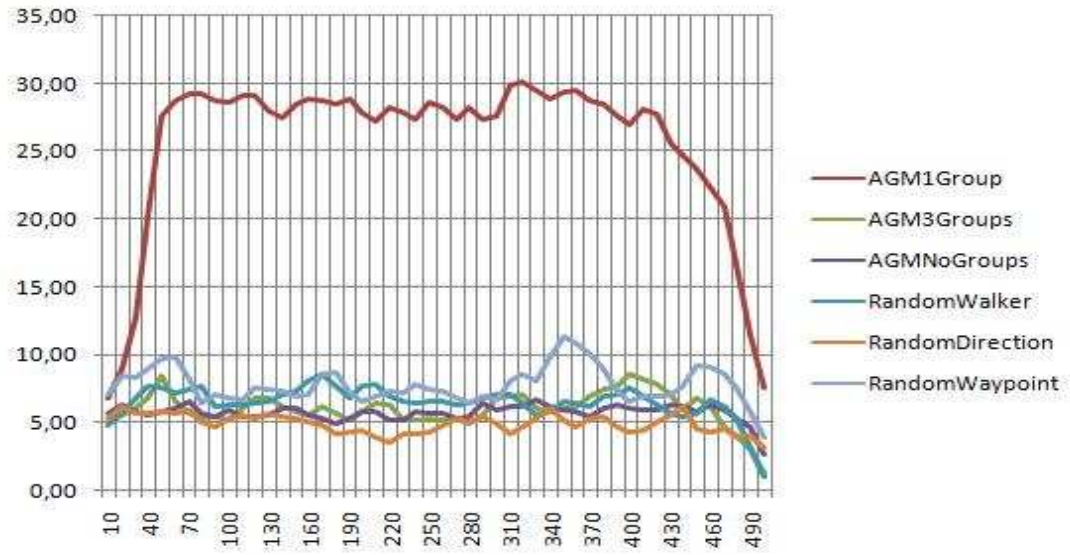


Figure 5.8 Average number of neighbors

AGM1Group as suspected has a high value of average number of neighbors. In RandomWaypoint there can be observed moments of temporary 'higher' values, those may be the density waves when more nodes are moving near the simulation area center.

On the other hand, RandomDirection has the lowest average. It can be easily connected with the fact that the nodes arrange themselves around the simulation area borders.

The lower values at the end of simulation are probably connected to the application problems with parsing the last ten seconds of the simulation.

5.4.1.3 Number of hops

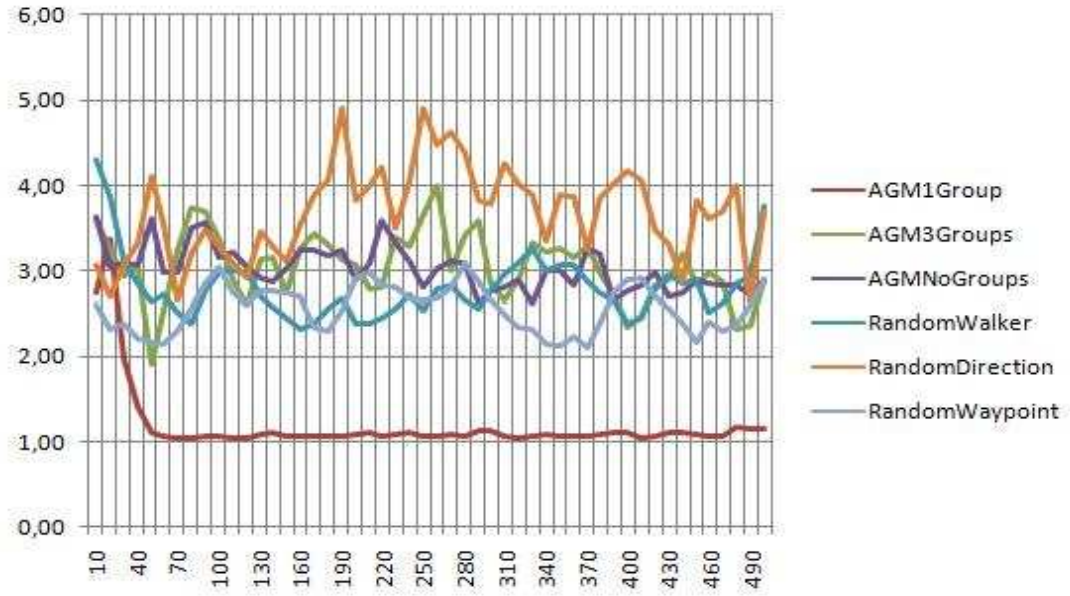


Figure 5.9 Average number of hops

When, in AGM1Group, a group is created at about 70s of simulation time, most nodes have a directional connection to each other.

In RandomWaypoint some ‘lower’ values can be observed at about the same times that the density waves took place.

AGM3Groups and RandomDirection seems to change more rapidly than the other scenarios. In RandomDirection this may be caused by the nodes traversing the center of the simulation area and by doing so creating shorter paths between nodes close to different edges. However the lifetime of those paths is limited to the time nodes spend near the middle of the simulation area.

AGMNoGroups and RandomWalker seem to have the most ‘constant’ values. This may be caused by their tendency to fill evenly the whole simulation area.

5.4.1.4 Unconnected pairs

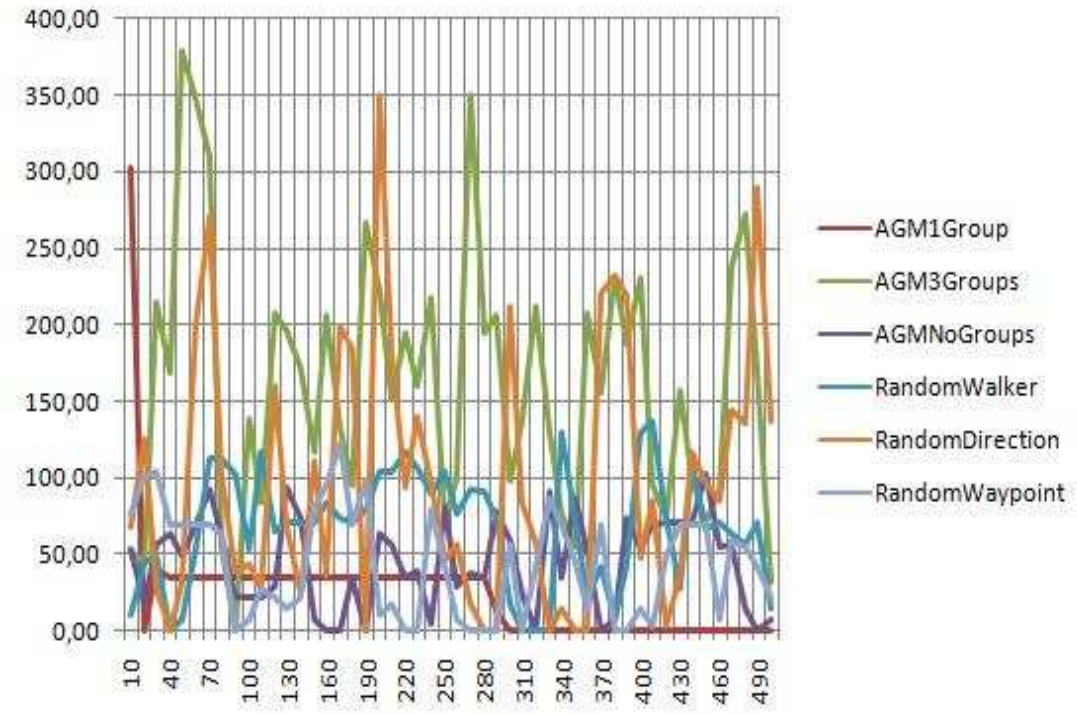


Figure 10 Average number of not connected nodes

AGM1Group scenario grouping can be well observed, as well as the moment of the one ‘missing’ node ‘joining’ the rest. Also the high value at the beginning that can help understand the high value of standard deviation.

Here we can again see some similarities between AGM3Groups and RandomDirection scenarios. In both of them rapid changes and high values can be observed. In RandomDirection, as mentioned earlier, this may be caused by the nodes traversing the center of simulation area. In AGM3Groups it can be an effect of temporary grouping and then breaking of the groups.

Rest of the scenarios have more ‘consistent’ charts. It suggest that no more than two node or three nodes were ‘out of the reach’ at the same time.

5.4.1.5 Partitioning

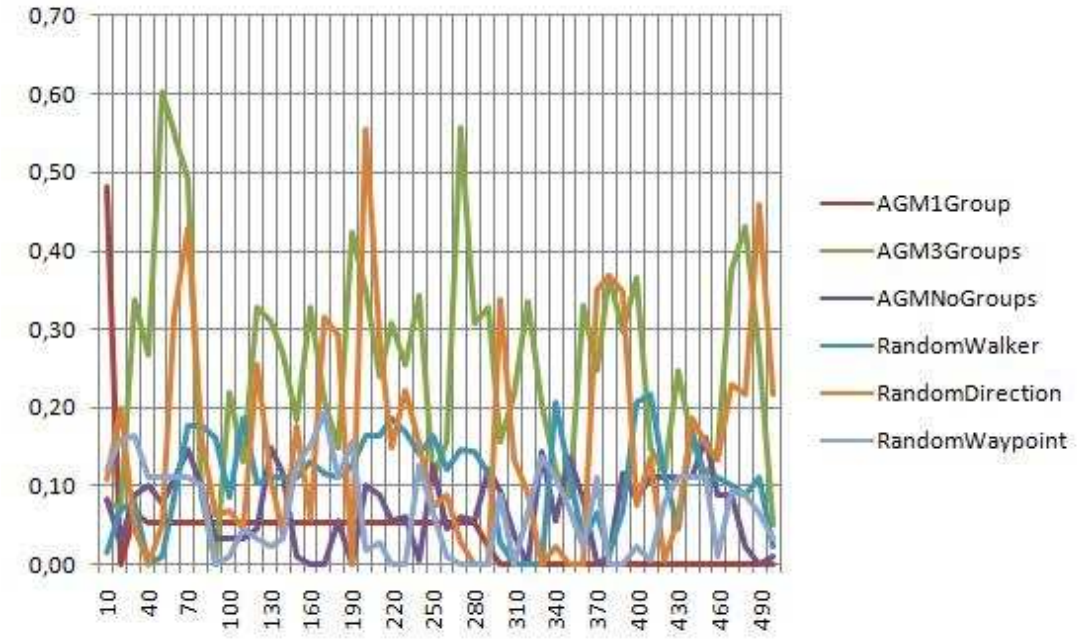


Figure 11 Average partitioning

As the partitioning value is calculated using the number of not connected node pairs and the number of all pairs (which is constant within the simulation), this chart differences from the previous one only in scale, and all comments made earlier are still true.

5.5 Scenario files

Scenario files define the network traffic within the simulation. It provides the information about nodes sending data, destination nodes and the traffic generator objects. At this time *ns-2* supports generators with exponential distribution, Pareto distribution and constant bit rate (CBR). It is also possible to attach application type generators (like FTP).

The scenario file used is the same for all of the movement scenarios. It is generated by the TrafficGeneratorPro application and It is presumed that half (randomly chosen) of the nodes will generate traffic with exponential distribution with both burst and idle times set to fifteen.

5.6 Simulations with scenario file

For the simulations with scenario file AODV protocol was used. There also was a series of simulations with DSDV protocol but all of them that used larger (more than 1MB) calc files failed with ‘segmentation fault’ result. The author thinks it is because of some memory issues within the c++ implementation of the DSDV but was unable to find and fix the problem.

The *ns-2* simulations output files – trace files were then parsed and analyzed using Trace File Analyzer (TFAPro) application. Output files of TFAPro are as in CFAPro saved in csv standard.

CFAPro can search trace files for different kinds of packets (exp, CBR, RTS etc.), below only exp (packet from exponential source), ADOV and ACK packets are analyzed. And same as earlier the values are average per second.

The overall statistics are presented in table 5.3 and the corresponding standard deviation in table 5.4.

Mobility scenario	exp sent/received/dropped/ forwarded	AODV sent/received/dropped/ forwarded	ACK sent/received/dropped/ forwarded
AGM1Group	0,0563/ 0,0615/ 0,0011/ 0,0056	1,0377/ 17,3538/ 5,3531/ 0,0021/	0,1181/ 0,1215/ 0,0002/ 0,0036/
AGM3Groups	0,1487/ 0,2048/ 0,0211/ 0,0652	1,2054/ 7,7229/ 0,4833/ 0,0381	0,3477/ 0,3952/ 0,0020/ 0,0491
AGMNoGroups	0,1081/ 0,1461/ 0,0099/ 0,0435	1,0216/ 5,5982/ 0,4130/ 0,0361	0,2917/ 0,3286/ 0,0005/ 0,0374
RandomWalker	0,1134/ 0,1551/ 0,0107/ 0,0476	1,0665/ 6,6849/ 0,4866/ 0,0388	0,2933/ 0,3301/ 0,0008/ 0,0373
RandomDirection	0,1492/ 0,2109/ 0,0198/ 0,0701	1,3359/ 6,3651/ 0,3097/ 0,0482	0,3811/ 0,4379/ 0,0016/ 0,0582
RandomWaypoint	0,0923/ 0,1209/ 0,0080/ 0,0329	1,0562/ 7,5575/ 0,6172/ 0,0289	0,2397/ 0,2674/ 0,0004/ 0,0281

Table 5.3 Average values of simulation events

Mobility scenario	exp sent/received/dropped/ forwarded	AODV sent/received/dropped/ forwarded	ACK sent/received/dropped/ forwarded
AGM1Group	0,0449/ 0,0554/ 0,0036/ 0,0152	0,4216/ 7,3370/ 2,3182/ 0,0067	0,1004/ 0,1102/ 0,0012/ 0,0123
AGM3Groups	0,0848/ 0,1286/ 0,0190/ 0,0533	0,5613/ 3,9288/ 0,3316/ 0,0273	0,2127/ 0,2499/ 0,0063/ 0,0434
AGMNoGroups	0,0651/ 0,0870/ 0,0224/ 0,0410	0,4100/ 2,2793/ 0,1923/ 0,0305	0,1590/ 0,1796/ 0,0113/ 0,0363
RandomWalker	0,0654/	0,4578/	0,1429/

	0,0893/ 0,0266/ 0,0424	2,9441/ 0,2178/ 0,0328	0,1615/ 0,0177/ 0,0361
RandomDirection	0,0801/ 0,1165/ 0,0285/ 0,0521	0,6671/ 3,4450/ 0,2301/ 0,0338	0,1925/ 0,2257/ 0,0155/ 0,0451
RandomWaypoint	0,0574/ 0,0742/ 0,0273/ 0,0364	0,4338/ 3,2804/ 0,3245/ 0,0303	0,1223/ 0,1385/ 0,0189/ 0,0320

Table 5.4 Standard deviation values of simulation events

5.6.1 Exp packets

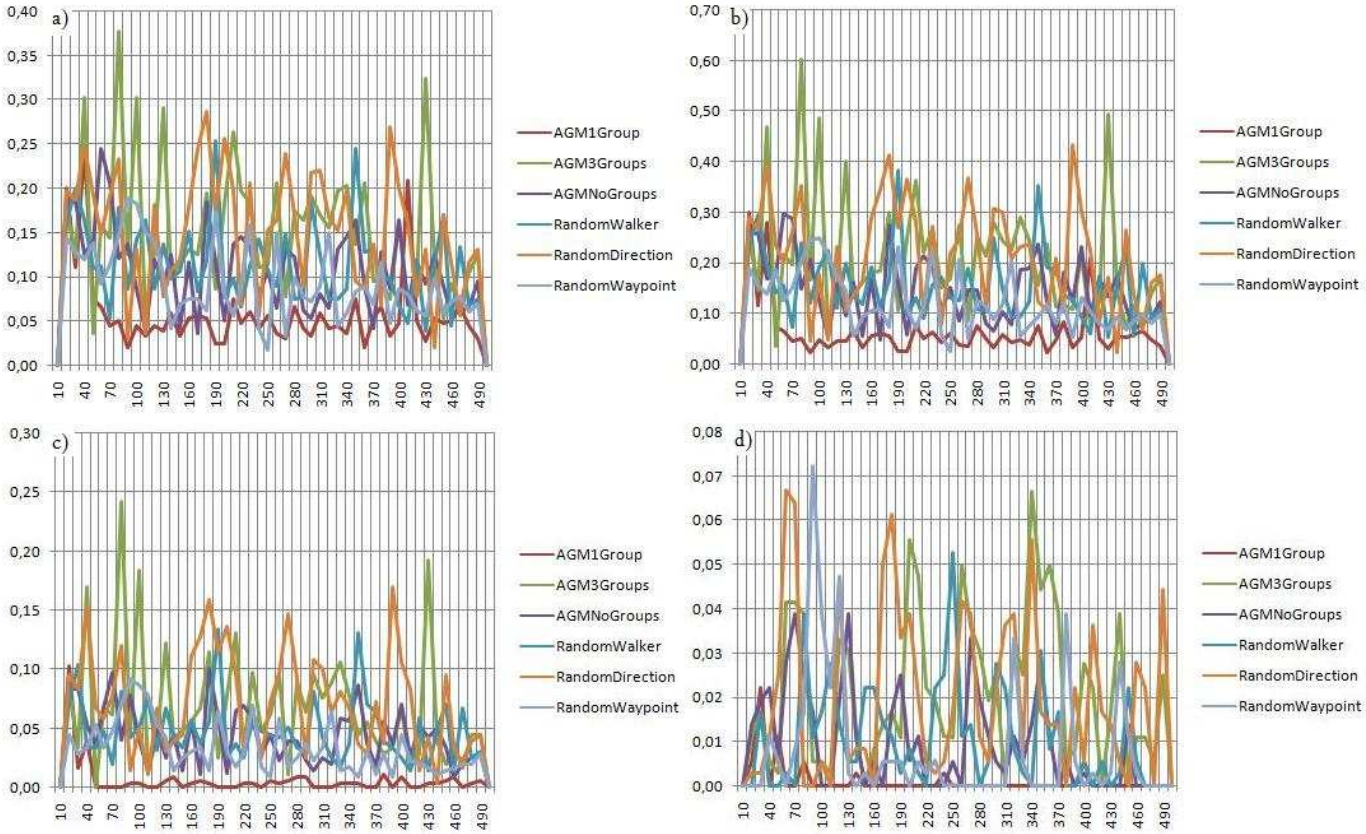


Figure 12 Packets from exponential source. a)sent b)received c)forwarded d)dropped (average per second for one node)

RandomDirection and AGM3Groups values are changing rapidly during the simulation with the tendency for high values. This can be connected to their partitioning chart (especially at times 70, 190 and 290). It could be an effect of rapidly changing topology and the creation of smaller unconnected groups.

The AGM1Group scenario characteristics are, as expected, quite low. With high number of neighbors and low value of average hops number most of the packets reach their destination in no more than two hops. An intriguing thing is the high

value of dropping packets in the RandomWaypoint scenario (times 80, 310 or 370) which is hard to connect with any of the others presented earlier characteristics.

5.6.2 AODV packets

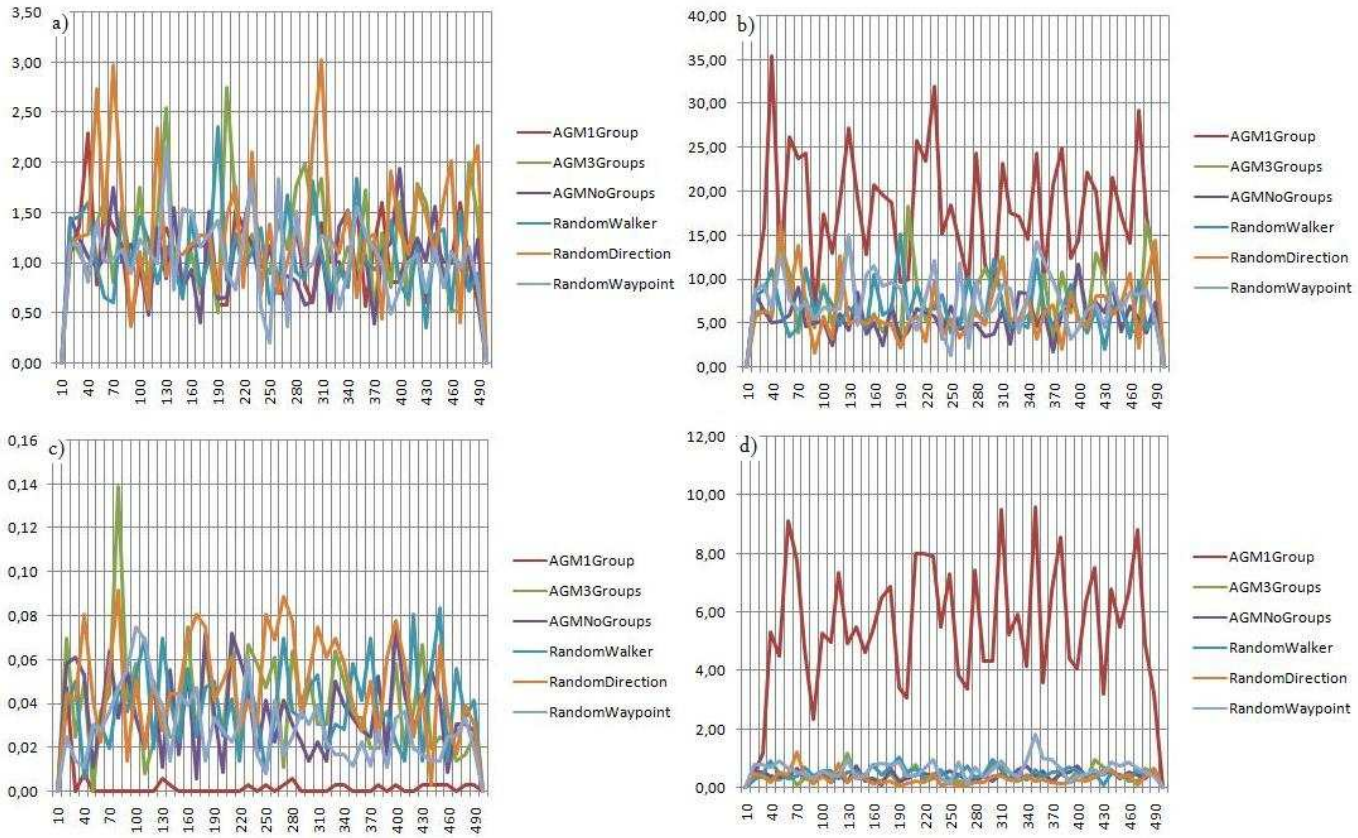


Figure 13 AODV packets. a)sent b)received c)forwarded d)dropped (average per second for one node)

In most cases ADOV packets are sent, received, forwarded and dropped in a similar fashion. It seems interesting that for the fires time AGM1Group scenario has much higher average values. While the sending number of ADOV packets is similar to other scenarios then the received and dropped values are much higher. That can be explained in the same way that the low values of exp packet characteristics. While all nodes are close there is much higher chance of hearing more ADOV packets which then will be dropped more often.

AGM3Groups and RandomDirection have a higher number of packets sent, what probably is connected to their changing topology.

Rest of the scenarios are quite stable with some differences in RandomWaypoint (higher dropping rate at time 350).

5.6.3 ACK packets

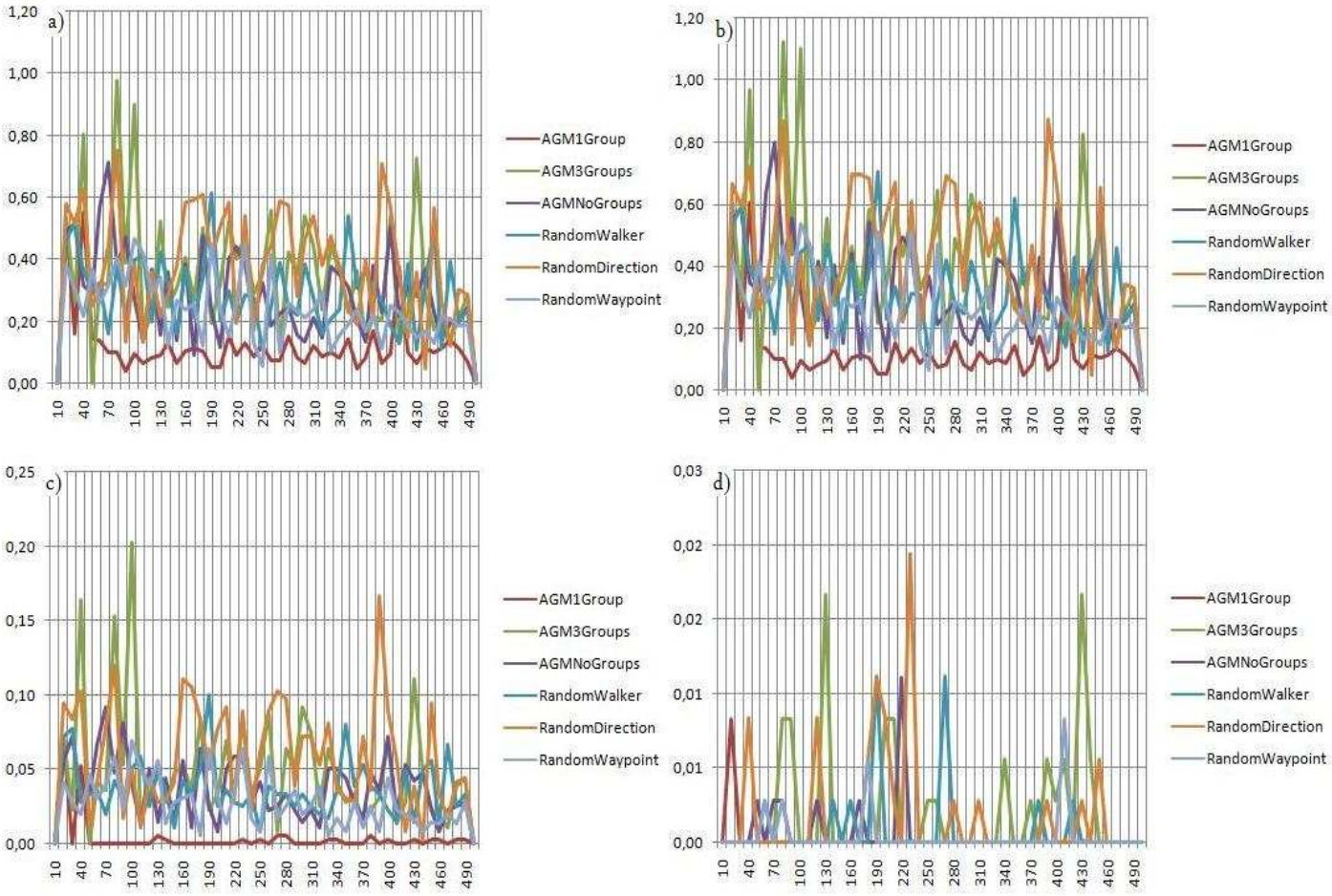


Figure 14 ACK packets. a)sent b)received c)forwarded d)dropped (average per second for one node)

The protocol used during the simulation was TCP so ACK packets could be observed. Their characteristics (sending and receiving) are very similar to the characteristics of the exp packets. Which is as expected. They are also dropped less often dropped then the other packet types.

5.7 Summary

An important thing that was shown by the simulations is the differences in the behavior of the network (in respect to different characteristics) when using different mobility scenarios.

While trying to create any kind of node behavior model the most important thing would be to chose what kind of network will be used. Another answer would be to use an adaptive algorithm that would adapt the model to the changing characteristics of the network.

Another thing that should be considered is to omit the first and the last ten or twenty seconds of simulation and focus on the remaining part of the simulation.

It also seams interesting how AGM scenarios can ‘mimic’ other movement patterns (AGMNoGroups and RandomWalker or AGM3Groups and RandomDirection) in respect to different characteristics and in other cases (AGM1Group) behave quite differently.

6 CONCLUSIONS

6.1 Research Question 1: What are the important elements when modeling mobile ad-hoc networks.

The most important elements that should be considered while modeling mobile ad-hoc networks are:

- mobility
- routing

The choice of appropriate routing protocol is one of the most important things. There are two main ideas for routing, namely proactive and reactive. The choice between them should be well considered according to the network requirements and purpose. Hybrid protocols based on both mentioned earlier may still require some more research but seem to be quite promising.

Mobility model is at least as important as the routing protocol. It should reflect the reality as truly as possible. It is almost impossible to overestimate the possible influence of the mobility model on the network characteristics. This was clearly shown during the simulations within which six different mobility scenarios were used.

Another important element can be the number of nodes within the network. While for many cases it could be treated as a parameter sometimes it may have some interesting influences. A good example of this could be a large number of nodes within a small area (a situation similar to the AGM1Group scenario).

6.2 Research Question 2: What improvements can be proposed for the process of modeling mobile ad-hoc networks.

The situation with routing protocols is that much have been already done. Many ideas and standards were presented, tested and well documented. Most of the work now is with changing them to provide security.

With the mobility models however, there is still much that can be improved. While they should imitate the reality as well as possible most of them still are just random models. Some of them are even aware of their previous states (Gaus-Markow mobility model) but still lack of some basic intentionality. On the other hand, there is some complex work done including monitoring city traffic to create almost identical with the reality movement traces (VANETs).

The problem is the absence of any ‘middle ground’ between those two approaches. And this work with the Anti Gravity Movement Mobility Model addresses this problem. It is possible to generate many different scenarios with AGM, and while the basic idea behind it is quite simple and easy to implement the possible results are almost limitless. This was shown during the simulations in which three different AGM scenarios were created on some very simple assumptions (without any external ‘repulsive force’ sources). By changing only one parameter (number of groups) scenarios witch highly differ in their characteristics were

achieved. It is an authors assumption that any of the random mobility models (or group mobility models) could be generated using different AGM settings.

6.3 **Research Question 3:** What are the most common security issues (attacks) in mobile ad-hoc networks.

In chapter 3 some of the basic elements of security in mobile ad-hoc networks was presented. There are few security issues that are common for this kind of networks such as:

- wireless channel of communication
- nodes cooperation requirement
- compromised nodes (attacks from within the network)
- distributed nature of MANET

Each of the above could be exploited by the attacker. Most of the attacks analyzed in the literature are targeted against the routing protocol. They try to either use routing to affect other messages flow (black hole attack) or deny nodes the possibility of its proper usage (flooding attack).

One of the most interesting (and one of the hardest to detect) is the wormhole attack. It do not concentrate on any of the routing protocols, it do not require of compromising any node or even having any information about the attacked network. It simply exploits the wireless channel and nodes limited range. In the initial phase it could cause the network to perform better than expected (sometimes close to the situation as in AGM1Group scenario) only to have the possibility of causing more damage in the future.

6.4 **Research Question 4:** Do the existing intrusion detection systems work in MANET and is it possible to improve their performance.

As mentioned in chapter 4 intrusion detection (as well as intrusion prevention) systems for mobile ad-hoc networks are mostly in the development and testing phase. While there can be found some different (even open source) IDSs for wired networks like SNORT NIDIS, Untangle or Bro NIDIS, it is almost impossible to apply them in the wireless environment.

The amount of work however is impressive. Some of the ideas focuses on the overall architecture of the systems for wireless environment while others concentrate on creating simple, but detecting some of the basic attacks, systems (like watchdog). There are also big projects of systems that actually work, but the documentation and implementation information is still limited.

Many improvements are still possible to achieve in those systems. The number of problems that need to be addressed is almost as high as the number of ideas that still needs to be tested.

6.5 Future work

The future work for this thesis can go in two directions. First, it is possible to expand the idea of AGM mobility model. Some more implementations that try to mimic different real world situations could be developed and then tested. A god starting point for this could be the set of applications created as a part of this thesis.

The second possibility would be to focus on intrusion detection systems. It would require of creating actual attack simulations, specify and implement some of the IDS ideas. In this case applications provided could be used to improve the simulation process and analysis of results.

SOURCE CODE AND EXECUTABLE FILES

The source code and executable files are attached to the thesis on a CD disk (hard copy of the thesis) or they are packed to the same zip file with the word document (electronic version).

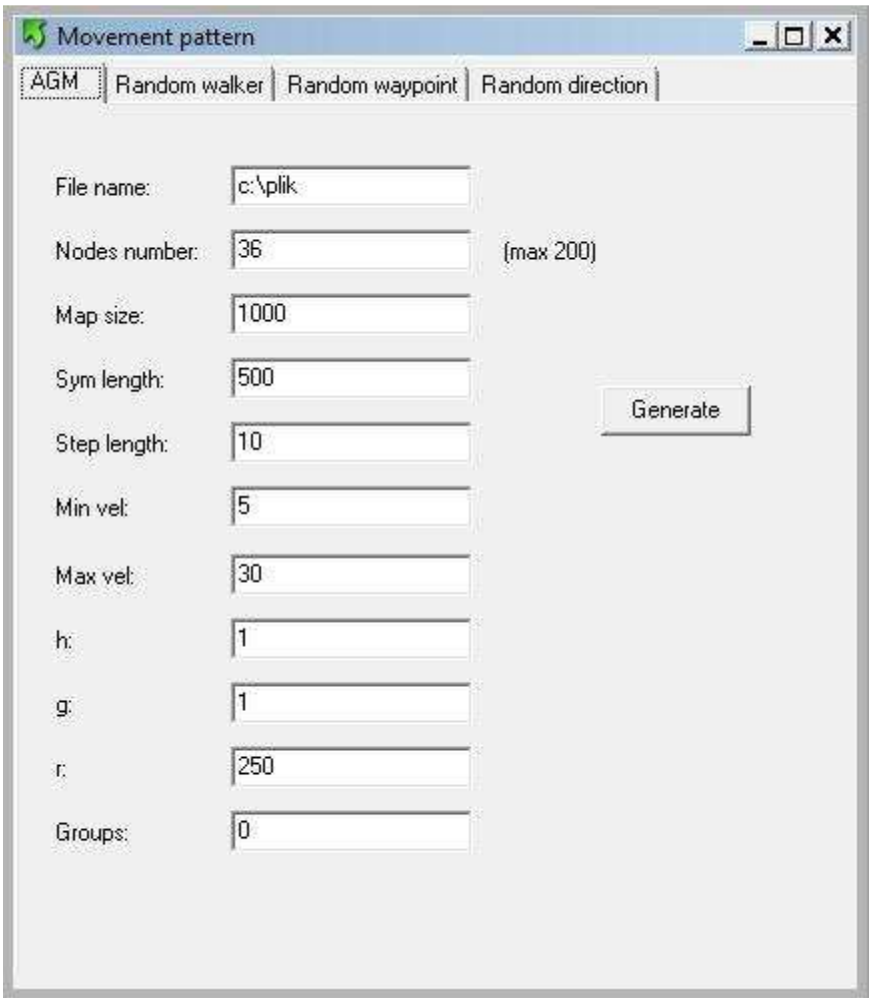
The source code of the applications is compressed to a zip file and placed in folder ‘source’, the executables (also compressed to a zip file) can be found in ‘applications’ folder.

APPLICATIONS MANUAL

1. MovementPatternsPro

MovementPatternsPro is an application designed for creating movement pattern files in *ns-2* standard. It can create patterns according to AGM, Random walker, Random destination and Random direction mobility models.

Example for AGM generation can look like this:



All fields are required to be filled with appropriate data. Application do not check if the input is correct so any mistakes may cause an error. The '*File name*' parameter defines the output file where the movement information will be stored, if it exists it will be rewritten. Other parameters depend on the mobility model. After clicking the '*Generate*' button, the output file will be created.

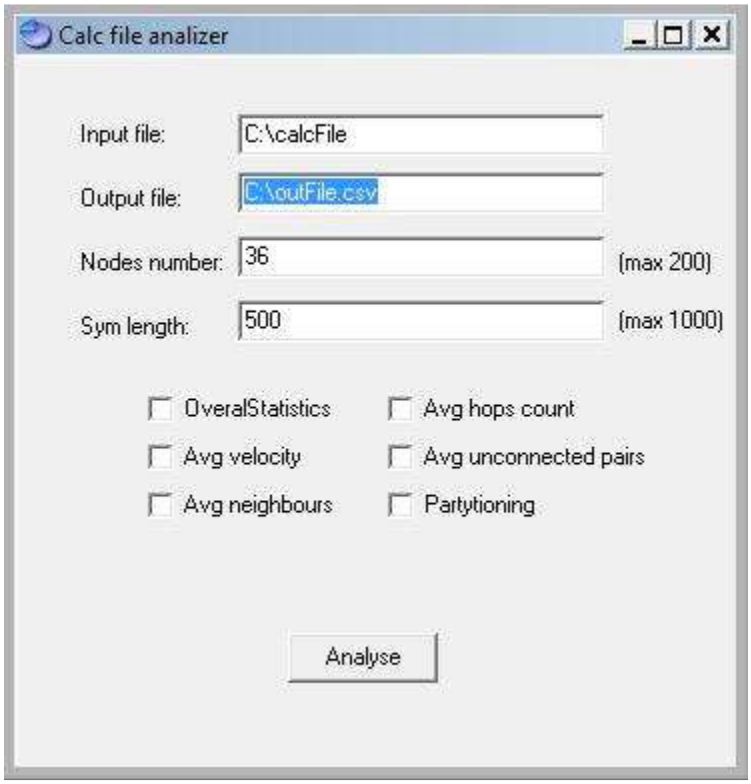
It is important to change any ',' in the output file to '.', this is caused by different floating point values formats in object pascal (in which the application is written) and the otcl language.

For Windows Vista it may be required to start the application using compatibility with Windows XP mode and with administrators privileges.

2. CFAPro

CFAPro is an application designed to analyze calc files that are outputs from *calcdest* program from *ns-2*. As an output it creates a coma separated file that can be then easily processed by excel or similar applications.

Example of CFAPro instance can look like this:



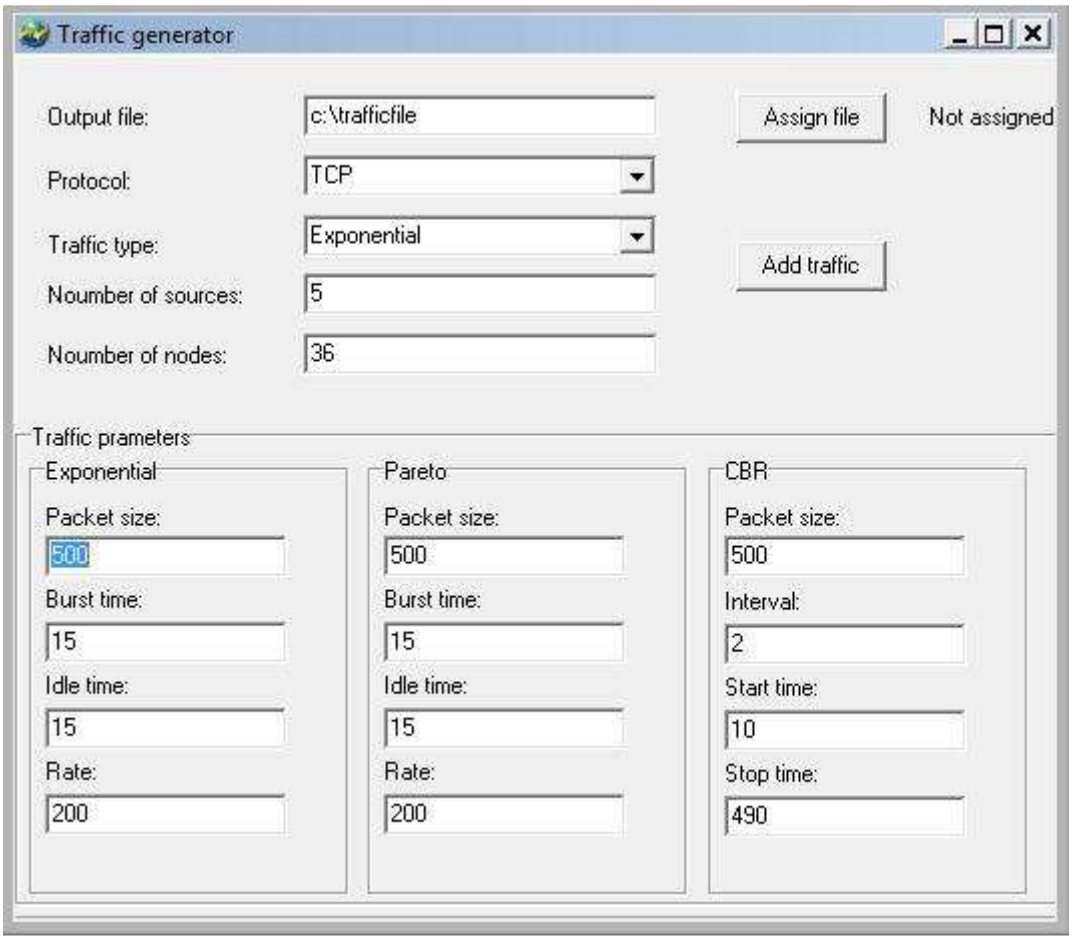
'Input file' defines the calc file to analyze. '*Output file*' defines the csv file in which the generated data will be saved. If the file already exists it will be rewritten. The application also requires additional information about the number of nodes used during the simulation as well as simulation time. The actual simulation length may be higher than the input parameter. In such a case all statistics will be calculated only till the time defined by the parameter. The checkboxes define the statistics that will be calculated

Application do not check if the input is correct so any mistakes may cause an error. For Windows Vista it may be required to start the application using compatibility with Windows XP mode and with administrators privileges.

3. TrafficGeneratorPro

TrafficGeneratorPro is an application designed for creating scenario files for *ns-2* simulations that meet required standard. It can generate traffic sources that uses different protocols (TCP/UDP) and using one of the supported by *ns-2* distributions (exponential, Pareto, CBR).

Example of TrafficGeneratorPro instance can look like this:



'Output file' defines the file where the scenario will be written. It is required to first assign the file (by pressing the '*Assign file*' button). If a file already exists it will be rewritten. The application also requires additional information about the number of nodes used during the simulation. The source and destination nodes are chosen randomly from all the nodes (with uniform distribution), it is impossible for a node to send data to itself.

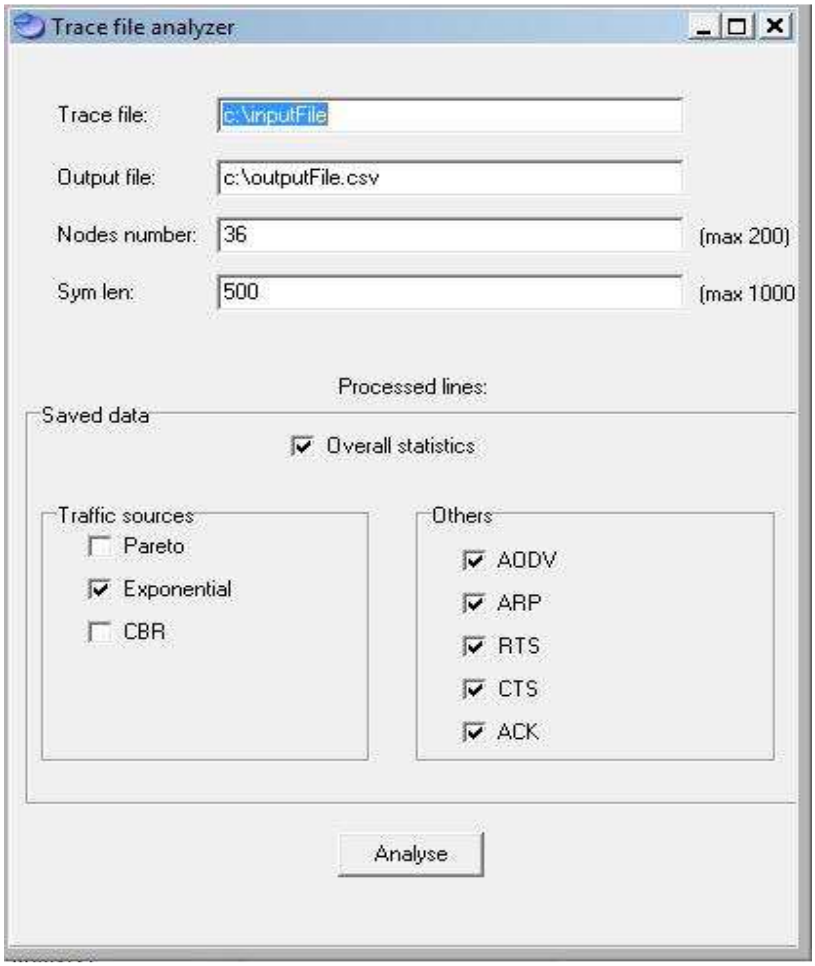
Clicking the '*Add traffic*' button will append information about new sources to the output file according to the set parameters, so it is possible to create a file with many different sources.

Application do not check if the input is correct so any mistakes may cause an error. For Windows Vista it may be required to start the application using compatibility with Windows XP mode and with administrators privileges.

4. TFAPro

TFAPro is an application designed to analyze trace files that are outputs from *ns-2* simulations. As an output it creates a coma separated file that can be then easily processed by excel or similar applications.

Example of TFAPro instance can look like this:



'Trace file' defines the trace file to analyze. '*Output file*' defines the csv file in which the generated data will be saved. If the file already exists it will be rewritten. The application also requires additional information about the number of nodes used during the simulation as well as simulation time. The actual simulation length may be higher than the input parameter. In such a case all statistics will be calculated only till the time defined by the parameter. The checkboxes define the types of packet types that will be processed.

Because it is possible for the trace files to be quite large (even 50MB and more) a number of processed lines is used and updated after reading every hundred lines.

Application do not check if the input is correct so any mistakes may cause an error. For Windows Vista it may be required to start the application using compatibility with Windows XP mode and with administrators privileges.

BIBLIOGRAPHY

[Ari06] Ariyakhajorn J., Wannawilai P., Sathitwiriyawong C., *A Comparative Study of Random Waypoint and Gauss-Markov Mobility Models in the Performance Evaluation of MANET*, International Symposium on Communications and Information Technologies, 2006.

[Bar07] Baras John S., Radosavac Svetlana, Theodorakopoulos George, Sterne Dan, Budulas Peter, Gopaul Richard, *Intrusion Detection System Resiliency to Byzantine Attacks: The Case Study of Wormholes in OLSR,* Military Communications Conference, 2007. MILCOM 2007. 29-31 Oct. 2007 Page(s):1 - 7

[Ben79] Bentley J.L., Ottmann T.A., *Algorithms for Reporting and Counting Geometric Intersections,* Transactions on Computers Volume C-28, Issue 9 page(s): 643-647 Publication Date: Sept. 1979.

[Bin06] Bing W., Jianmin C., Jie W., Mihaela C., *A Survey on Attacks and Countermeasures in Mobile Ad Hoc Networks*, WIRELESS/MOBILE NETWORK SECURITY Chapter 12, 2006 Springer.

[Bur06] Burbank J.L., Chimento P.F., Haberman B.K., Kasch W.T., *Key Challenges of Military Tactical Networking and the Elusive Promise of MANET Technology*, Communications Magazine, IEEE, Volume 44, Issue 11, November 2006 Page(s):39 – 45.

[Chi97] Chiang Ching-Chuan, Wu Hsiao-Kuang, Liu Winston, Gerla Mario, *Routing in Clustered Multihop, Mobile Wireless Networks with Fading Channel,* IEEE Singapore International Conference on Networks, SICON'97 pp. 197-211 Singapore 16.-17. April 1997.

[Di06] Di W., Xiaofeng Z., Xin W., *Analysis of 3-D Random Direction Mobility Model for Ad Hoc Network* 6th International Conference on ITS Telecommunications Proceedings.

[Dje05] Djenouri D., Badache N., *New approach for selfish nodes detection in mobile ad hoc networks*, Workshop of the 1st International Conference on Security and Privacy for Emerging Areas in Communication Networks, Page(s):288 – 294, 5-9 Sept. 2005.

[Erl07] Erlandsson Fredrik, *lecture notes on ET1318 – Network Security*, available on www.itslearning.com, 2007

[Fre01] Freebersyser J., Leiner B., *A DoD Perspective on Mobile Ad Hoc Networks*, Ad Hoc Networking, ed. C. 2001, pp. 29-51

[Hon99] Hong X., Gerla M., Pei G., Chiang C., *A group mobility model for ad hoc wireless networks,* In Proceedings of the ACM International Workshop on Modeling and Simulation of Wireless and Mobile Systems (MSWiM), August 1999.

[Har06] Harri J., Filali F., Bonnet C., *Mobility Models for Vehicular Ad Hoc Networks: A Survey and Taxonomy,* March 5th, 2006 Research Report RR-06-168.

[Hu03] Hu Y.-C., Perrig A., Johnson D.B., Packet leashes: a defense against wormhole attacks in wireless networks, Twenty-Second Annual Joint Conference of the IEEE Computer and Communications Societies. Volume 3, 30 March-3 April 2003 Page(s):1976 - 1986 vol.3

[Jar06] Jarmal P., Sawko R., Stelmach P., Szwarc J., Juszczyszyn K., *Ontology Based Approach to Situation Awersness Framework in Robocode Environement,* Information Systems Architecture and Technology, ISAT 2006.

[Joh96] Johnson D., Maltz D., Dynamic Source Routing in Ad Hoc Wireless Network, In T. Imielinski and H. Korth, Editors, Mobile Computing, Kluwer Academic Publishers, pp. 153-181, 1996.

[Joh07] Johnson D., *RFC 4728 - The Dynamic Source Routing Protocol (DSR) for Mobile Ad Hoc Networks for IPv4,* The IETF Trust (2007).

[KH01] Kuo-Hsing Chiang, Shenoy N., *A random walk mobility model for location management in wireless networks*, 12th IEEE International Symposium on Personal, Indoor and Mobile Radio Communications, Volume 2, 30 Sept.-3 Oct. 2001.

[Lia99] Liang B., Haas Z., *Predictive Distance-based Mobility Management for PCS Networks*, Proceedings of the IEEE INFOCOM 1999, Vol. 3, New York, NY, USA, pp. 1377-1384, March 1999.

[Mis04] Mishra, A., Nadkarni, K., Patcha, A., Intrusion detection in wireless ad hoc networks, Wireless Communications, IEEE Volume 11, Issue 1, Feb 2004 Page(s):48 - 60

[Mur96] Murthy S, Garcia-Luna-Aceves J.J., *An Efficient Routing Protocol for Wireless Networks*, ACM Mobile Networks and App. J., Special Issue on Routing in Mobile Communication Networks, Oct. 1996, pp. 183-97.

[Oka02] Okazaki Y., Sato I., Goto S., *A New Intrusion Detection Method based on Process Profiling*, Proc SAINT 2002., Jan. 28–Feb. 1, 2002, Pages: 82–90.

[Par97] Park V. D., Corson M. S., *A Highly Adaptive Distributed Routing Algorithm for Mobile Wireless Networks*, Proc. lNFOCOM '97, Apr. 1997.

[Per94] Perkins and C. E., Bhagwat P., *Highly Dynamic Destination-Sequenced Distance-Vector Routing (DSDV) for Mobile Computers*, Comp. *Com*mun. Rev., Oct. 1994, pp. 234-44.

[Per03] Perkins C. E., *RFC 3561 - Ad hoc On-Demand Distance Vector (AODV) Routing,* The Internet Society (2003).

[Pin06] Ping Yi, Zhoulin Dai, Yiping Zhong, Shiyong Zhang, *Resisting flooding attacks in ad hoc networks*, International Conference on Information Technology: Coding and Computing, 2005. ITCC 2005. Volume 2, 4-6 April 2005 Page(s):657 - 662 Vol. 2

[Roy99] Royer E.M., Chai-Keong Toh, *A review of current routing protocols for ad hoc mobile wireless networks.* Personal Communications, IEEE Volume 6, Issue 2, April 1999 Page(s):46 – 55.

[Roy01] Royer E., Melliar-Smith P.M., Moser L., *An analysis of the optimum node density for ad hoc mobile networks*. In Proceedings of the IEEE International Conference on Communications (ICC), 2001.

[San08] Sanchez M., *Mobilitymodels*, http://www.disca.upv.es/misan/mobmodel.htm accessed on 03.05.2008.

[Sca07] Scarfone K., Mell P., *Guide to Intrusion Detection and Prevention Systems (IDPS)*, Recommendations of the National Institute of Standards and Technology, February 2007

[Sou05] Souley A.-K.H., Cherkaoui S., Advanced mobility models for ad hoc network simulations, Systems Communications, 2005. Proceedings 14- 17 Aug. 2005 Page(s):50 – 55.

[Tam07] Tamilselvan L., Sankaranarayanan V., *Prevention of Blackhole Attack in MANET*, The 2nd International Conference on Wireless Broadband and Ultra Wideband Communications, 2007. 27-30 Aug. 2007 Page(s):21 - 21

[Tao03] Tao Lin, Midkiff S.F., Park J.S. *A framework for wireless ad hoc routing protocols.* Wireless Communications and Networking, 2003. WCNC 2003. 2003 IEEE Volume 2, 16-20 March 2003 Page(s):1162 - 1167 vol.2.

[Wei98] Weisstein E. W., *The CRC Concise Encyclopedia of Mathematics*, CRC Press, 1998.

[Zha03] Zhang Y., Lee W., Huang Y., *Intrusion Detection Techniques for Mobile Wireless Networks*, Wireless Networks archive, Volume 9 Issue 5 Pages:545-556, September 2003.