Metric of Trust for Mobile Ad hoc Networks Using Source Routing Algorithms

by

Denise Umuhoza

A thesis submitted in fulfillment of the requirements
for the degree of Magister Scientiae
in the Department of Computer Science,
University of the Western Cape

Supervisor: Prof. Christian W.P. Omlin

May 2006

packets. We have developed a model that detects anomalies in network traffic and calculates probability of anomalies being caused by link failure or by attack.

The metric uses traffic analysis tool to collect traffic patterns and it uses statistical analysis to calculate probability of occurrence of attack on a path based on anomalous behaviour detected in traffic. The metric is useful in circumstances where nodes are unable to link the activities and identity of their neighbors, thus not possible to monitor activities of a certain node in the network. In that case, sender and receiver can only monitor status of traffic patterns at their ends and make a conclusion on what is happening on the path in use.

We have designed and implemented the metric on an experimental MANET we set up. We imitated different scenarios that change the status of traffic status. As results showed, in some cases it is not possible to distinguish the exact cause of the change of the status of network traffic. The effectiveness achieved by the metric was up to the highest of 90% and the lowest of 50%.

All the same, the metric developed in this thesis does a reasonable rating of the trustworthiness of the path as it decreases trust when an anomaly is detected, and it decreased trust again when anomaly appear in regular patterns.

The metric also has been evaluated in terms of giving indication that attack happened while they have not. Those false positives are caused by the fact that in some cases the results of link failure and attacks affect the status of a network in the same manner. The results showed that in some cases the metric can go up to 30% of false positives. That is a high percentage, yet when we look at the purpose of the metric; it is still acceptable because trust of path reduces whenever probability of anomalies increases whether caused by attack or link fault.

Performance of the metric in terms of bandwidth consumption and processing delays was beyond the scope of our goal, but they will be addressed in our future work. As an extension to our work we will modify the routing protocol to respond to information given by the metric. We will test our model on a bigger network and compare the effectiveness and performance in different conditions.

KEYWORDS

Wireless networks

Mobile Networks

Ad hoc networks

Routing

Trust

Metric

ABSTRACT

Metric of Trust for Mobile Ad hoc Networks Using Source Routing Algorithms
D Umuhoza
MSc Thesis, Department of Computer Science, University of the Western Cape

This thesis proposes and presents technical details of new probabilistic metrics of trust in the links wireless Ad Hoc networks for unobservable (covert) communications. In covert communication networks, only the end nodes are aware of the communication characteristics of the overall path. We overview the most widely used protocols of ad hoc networks. We review also the routing protocols of ad hoc networks with trust considerations and select Destination Sequence Routing (DSR), a protocol that can be used in distributed ad hoc network settings for path discovery. It establishes a path through which all packets sent by a source must pass to the destination. The end nodes are responsible for examining the statistics of the received packets and deriving inferences on path feature variations which are used for computing new trust metrics. When a path is judged not trustworthy based on the metrics, the DSR is informed to undertake a new trusted path discovery between the end points. The thesis adds a new feature based on the quality of service parameters of the path to create trust in the links in recognition of attacks. The new metrics of trust uses delay, congestion, inserted packets, packet losses, variation in packet transit time between source and destination and replayed packets to derive probabilistic metrics. We modify and recompile DSR suitable for application under the Linux Debian Operating System on PC and Linux Familiar on PDA platforms for communications. The modified DSR is uploaded into the nodes (PDAs and PC). We validate and evaluate the performance of the metrics using a practical ad hoc network consisting of four nodes, one a PC with a wireless LAN (WLAN) card and the other three nodes are wireless enabled PDAs. After implementing the ad hoc network, we undertake communication in a laboratory environment. We also simulate on the network attacks by injecting probe packets, packet drop outs and link failures as could occur in a network under attack. It is shown that the new metrics of trust recognize such attacks for more than 90% of the time and in the least case about 70% of the time. The thesis is concluded by detailing further research on trust metrics for intermediate nodes between the two end points.

DECLARATION

I declare that *A Metric of Trust for Mobile Ad-hoc Networks Using Source Routing Algorithms* is my own work, that it has not been submitted before for any degree or examination in any other university, and that all the sources I have used or quoted have been indicated and acknowledged by complete references.

Full name: Denise Umuhoza Date: May 2006.

Signed

Dedication

To my son; Yannick Bertrand Mbabazi, for being a good boy while I was away.

Acknowledgements

I first of all thank God for keeping me alive and healthy till today. I would like to express gratitude to my supervisor; Prof Christian Omlin for giving me admission, for guiding me and for caring about my welfare during my studies. I am grateful to the government of Rwanda through Human Resource Development Program for the sponsorship that made my studies possible. Kigali Institute of Science and Technology is worthy my thanks for the assistance in all administration formalities that were needed for my sponsorship and for the financial support to attend conferences while I was doing my research.

Particularly I would like to thank Ralf Staudemeyer for his great ideas and his technical help and his comments that helped me to improve the thesis. It is he who coached me in networking and in physical implementation of my work. His encouragements helped me get to the end of my research. I would like to show my gratitude to Prof Johnson Agbinya for his valuable advices and for his care for my welfare. I would like to thank the CoE and the Department of Computer Science for providing me with conducive working environment and financial support. My thanks to all faculty staff specially Mrs Verna Connan, Prof IM Venter, Mr Michael Norman and Mrs Rene Abbot for their administrative assistance.

I would like to show my appreciation to Prof. Melvin Ayogu for his countless support and encouragements. My love and my thanks are due to my family, specially my mother and my sister Olga Muhimakazi who cared and supported me all through. Lastly but not least, I would like to thank all my colleagues and all my friends who in one way or another contributed to the success of my thesis.

Table Of Contents

Figures

Tables

Chapter 1

Introduction

1 About this chapter

In the introductory chapter of this thesis, the typical characteristics, the importance of mobile ad hoc networks and security challenges in those types of networks are elaborated broadly. Thereafter, the premises, on which this research is based, are described together with its limitations. Problem statement and research hypotheses follow respectively. Next come technical objectives and methodology adopted in order to achieve those objectives. Expected contributions of this thesis to the knowledge of the field are mentioned subsequently and finally the organization of the thesis is outlined.

1.1 Motivation of the research

The motivation of this research is divided into three sections for the sake of clarity. First of all the thesis gives a general view of characteristics of mobile ad hoc networks, especially those distinguishing them from the traditional networks and standard wireless networks. The thesis carries on by highlighting circumstances in which mobile ad hoc networks can be useful. Lastly, typical challenges that are related to the nature of mobile ad hoc networks are briefly elucidated.

1.2 Characteristics of wireless Mobile ad hoc networks

Wireless networks can be classified as infrastructure based and ad hoc networks. Wireless fixed networks operate with the help of various networks supporting equipment such as base stations and access points and the whole network is managed through this equipment. Wireless ad hoc networks are unlike wireless infrastructure networks on that aspect.

A mobile ad-hoc network is a collection of wireless mobile nodes self-organized to create a temporally connection between them. Neither pre-defined network infrastructure nor centralized network administration exists to assist in communication in mobile ad hoc networks. Nodes communicate with one another via direct shared wireless radio links. Each mobile node has limited transmission range. Nodes wishing to communicate with other nodes out of their transmission range employ a multi-hop strategy.

In a multi-hop environment, nodes forward packets for each other; therefore each node simultaneously acts as a router and as a host. An ad hoc network has a dynamic topology. Nodes change location within the network as people carrying them move around. Nodes also join and leave the network at any time as they are switched off when they want to save power, and switched on when they need to communicate again. These types of networks are beneficial because they are very easy to deploy; by the fact that they operate in the absence of any existing infrastructure. Mobile ad hoc networks are a cooperative way of exchanging peer-to-peer information among various mobile devices, and there are many cases in which these types of networks can be very practical.

1.3 Importance of mobile ad hoc networks

Mobile ad hoc networks are advantageous in situations where there are no network infrastructures available when there is a need for people to communicate using mobile devices. There are few cases of such situations that are given as examples in this section. Think of a scenario where a natural disaster like earthquake devastates an area and the existing network infrastructures get destroyed. The rescue team will need to communicate in order to perform its task of saving people in peril. In that case of emergency, the ad hoc network will be the only option. Mobile Ad hoc network can also be useful in situations where people find themselves in a place like a conference room and need to share some information or facilities like printing while there is no wireless infrastructure.
Mobile ad hoc networks may be employed in everyday life in more other cases where people carry devices equipped with sensors and they send information to each other as people move around. Even though mobile ad hoc networks are very important and will

become quite popular, there are a number of challenges in this area that make communication tricky .

Absence of network infrastructures renders the existing communication techniques in infrastructure based networks unsuitable for mobile ad hoc networks. It is indispensable to allude to these challenges in order to obtain appropriate communication techniques for mobile ad hoc networks.

1.4 Security Challenges in mobile ad hoc networks

Some issues in networking are specific to wireless networks and in particular to ad hoc networks because of their characteristics. Security is one of the issues in mobile ad hoc network, among battery power of mobile devices that is easily exhaustible, bandwidth is usually scarce in wireless networks because of costs and the limited transmission capacity of devices. This work is mainly focused on some of the security issues.

Consider however a non-ad hoc network, the traditional wired network where the attacker of the network must have a physical access to the network in order to perform an attack [1]. Wireless links however are more susceptible to security attacks as wireless links are accessible by both legitimate users and attackers with malicious intent. The degree of susceptibility to attacks gets even higher in mobile ad hoc networks because of the lack of centralised trusted management of the network and the movement and the availability of nodes in the network.

Depending on the circumstances in which the ad hoc network is used, the information sent across the network may be very sensitive; hence a secure communication must be guaranteed to the users of the network. A lot of research has been conducted on security issues in ad hoc networks, as security is a concern of users of networks, be they wired or wireless. Many researches have been conducted and solutions have been proposed, but none has been a complete solution of security in ad hoc networks. There is still significant research in this area on unsolved problems and for optimization of found solutions.

Seeking to play a part in optimization of solutions to security issues, this thesis mainly contemplates the security matters in routing procedures in mobile ad hoc networks.

As is the case for infrastructure based networks, the basic problem of routing is to find the lowest cost path between any two communicating nodes. The solution to that problem is to run routing protocols among a subset of intermediate nodes; dedicated routers. Classical routing protocols such as Routing Information Protocol (RIP) and Open Shortest Path First (OSPF) [2], used in traditional wired networks run on dedicated routers to maintain and keep routing information. Since there is no central administration and each node acts as a router, these protocols are not suitable for mobile ad hoc networks. Thus, special routing protocols have been developed to adapt to characteristics of mobile ad hoc networks [3,4]. Routing protocols run at each node, hence each node has access to the routing information. The challenge is that nodes participating in the network might have malicious intent. While nodes access the routing information they might use it to perform attacks on the network.

Consequently a secure routing mechanism is the basis of the security in mobile ad hoc networks. Seeing that nodes have to share the routing information in order for each node to find the route to the destination, and that ad hoc network is an open setting where every one can participate, trust is a key concept in secure routing mechanisms.

Given the nature of mobile ad hoc networks, trust is a tricky concept to establish among nodes in the network. Trust is not granted, it must be gained with time based on nodes' behavior. In some circumstances nodes might not stay in the network for a long time for their behavior to be observed or nodes might be having a change behavior over the time.

It is the intention of this work to propose a metric of trustworthiness of a communication path in mobile ad hoc networks. Two nodes engaged in a communication will actively measure the trustworthiness of a communication path that they use in a particular communication by analyzing the quality of service (QoS) behaviour of the data traffic. Users and nodes will update the trustworthiness of the communication path as the communication goes on. A communication path with trustworthiness with a given threshold will be said to be untrustworthy. That communication path will be avoided and another communication path will be used.

1.5 Assumptions and Premises

The proposed metric of trust in this work is based on a number of assumptions about the characteristics of a mobile ad hoc network path that are not trivial but significantly realistic. These are

1. Encryption and authentication algorithms are implemented for secure data transmission. Cryptography techniques are there to protect the content of transmitted data from being tempered with. Only the intended receiver is able to read and therefore to change the content of transmitted packets. Authentication mechanisms let the communicators verify if their partners are truly what they claim to be.
2. Although mobile devices have limited battery life time, it is assumed that they have enough memory and power to keep and maintain the routing tables and information about traffic patterns.
3. Nodes in a network may move without prior notice, but the movement will be moderate, since mobile devices are usually carried by humans.
4. Proper synchronization of the system time between communicating nodes; this is essential for reliable record keeping about packet transmission.
5. Eavesdroppers cannot derive valuable information from the network.

Mobile ad hoc networks can have a wide range of properties depending on the number of nodes in the network, the distance between nodes, the devices used and the movement of the node in the network. Our work at this stage is limited to specific cases where parameters of the environment can be predictable, for example the use of mobile devices in a conference room, in an office or other places where we can predict movement and obstacles between devices. In such environments, it is possible to define the essential wireless link parameters necessary for creating the metric of trust of a communication path which is composed of a set of wireless links.

1.6 Problem statement

In any type of computer network, reliable delivery of the information to the intended destination is of major interest to users sending information across that network. The information on the network might not be delivered to the destination as it is designed by the system because of many reasons. These reasons can be grouped into two categories: Network faults and security attacks. The main problem is to detect these abnormal changes in the network and categorize them. If these anomalies can be detected, the other problem is to prevent them.

In mobile ad hoc networks, there is no central administration to take care of detection and prevention of anomalies. Therefore nodes have to cooperate for the integrity of the operation of the network.

However, nodes may refuse to cooperate by not forwarding packets for others for selfish reasons and/or not wanting to exhaust their resources. On the other hand nodes may refuse to comply with the routing protocols with a malicious intent. Those nodes will be considered as attackers to the networks and will have to be detected and avoided.

Progressively routing protocols are being designed with a strong anonymity; mainly location and identity. In that way there will be no information that can be used by an observer to the network to identify a particular node (location and Identity). In that situation, it will be a challenge to detect a misbehaving node.

1.7 Research hypotheses

Several hypothesis were considered as part of this thesis and the following are of particular interest:

- Are there anomalies in relaying packets mechanism that can be detected by the sender or the receiver?
- If anomalies can be detected, can they be prevented?

- Is it possible to know if the anomaly is caused by link faults or by the misbehavior of some nodes in the network?
- Is it possible for the sender and the receiver to monitor the behavior of the communication path that they are using?
- Can the sender and the receiver decide to use an alternative route for their communication whenever they judge it necessary to do so to avoid an existing attack on the path?

1.8 Technical objectives

For our hypothesis, the following main technical objectives of the thesis are specified to be:

- Study of wireless link parameters and possible wireless link faults in mobile ad hoc networks.
- Study of attacker models; we want to know possible attacks on wireless links.
- Identification of the attacks that can be detected and that can be prevented.
- Identification of the attacks that can be detected but cannot be prevented.
- Study of the normal behavior of a mobile ad hoc network.
- Study of the behavior of a network in case there is link fault.
- Study of the behavior of a network in case there is attack.
- Propose a metric that observes the behavior of the communication. Then measure the trustworthiness of a communication path whenever an anomaly is detected.

1.9 Methodology

We adopted several methods for this research. Like every other research, this study includes review of relevant literature, metrics of trust modeling in which we propose new models for measuring trust in attack prone network paths and finally we implement an ad hoc network system using hand held devices to enable us demonstrate the metrics. We overview these three aspects of our methods below.

• Literature review:

We consulted books published in this area as one of the first places to look for certain vital information, which are helpful during the investigation. We also established contacts with other researchers doing related work in this area, thereby getting some literature which was be very useful. Magazines and news letters for example, IEEE monthly magazines, etc, were very useful in keeping pace with the technological developments. Subscription to mailing and discussion groups was undertaken for exchange of materials which may help to understand certain literature and hardware designs. Lastly, we looked at the white papers produced by the industry players for example, Microsoft, HP, IBM, etc.

• Design of a metric of trust:

The new metric is based on Dynamic Source Routing (DSR) algorithm. The metric has three main divisions as follows:

Traffic pattern collection: Packet Identification and time stamp (departure and arrival) of each packet are recorded by sender and receiver.

Anomaly detection: The behavior of a communication path is observed using the traffic pattern collected compared to the expected behavior of that communication path.

Trust update: According to the behavior observed, the trustworthiness of the communication path is judged and when necessary adjusted by sender and receiver. A mathematical model is used for the trust update.

• Implementation:

Simulation and physical experimentation are both acceptable methods for evaluating ad hoc network routing protocols. In general simulation is easier than full physical implementation and can permit repeatable experiments. On the other hand it can be rather difficult to model some aspects of the ad hoc network like realistic node mobility and data traffic. In that case simulation may not succeed to depict the exact behavior of the network.

Although much research ends at the simulation point using software alone, we undertook a further step and for the reason stated above we chose to do a full physical implementation of an ad hoc network. We constructed a network with four nodes: 3 iPAQs running linux-familiar and one notebook running Debian as linux distribution.
Nodes were carried around with a distance between them that allow forming a path with 3 hops depending on the transmit power level of the wireless interface of each node.

1.10 Contributions

The thesis brings up the awareness of behavior of data traffic in the real mobile ad hoc networks. It summarizes the attacks on wireless links and the possibility of preventing some of the attacks. The thesis makes clear that some attacks cannot be prevented or not even be detected.

The thesis builds up a simple metric of trust that will allow users of the network to monitor the behavior of their communication path. Especially, packet oriented attacks are considered while designing this metric of trust.

This metric of trust developed in this thesis will be very useful in a network where the privacy of identity and location will be fully implemented and yet users will be able to monitor the traffic of their communication without violating the privacy rules.

1.11 Organization of the thesis

This thesis consists of six chapters. In chapter 1, motivation and premises of the research are given. Motivation and the problem statement are described afterwards and right after the research hypotheses and technical objectives are constructed. Later, methods used in this research are explained briefly. That contribution that the thesis adds to knowledge is summarized and lastly comes the outline of the thesis. Chapter 2 gives the context in which this thesis is undertaken. It contains a detailed review of the routing mechanisms in mobile ad hoc networks, wireless link parameters, security attacks and prevention of those attacks in mobile ad hoc networks; it finally gives the trust definition and its essence. In chapter 3, the review of the literature is done based on the context we give in chapter 2; the thesis reviews the trust models and trust metrics for wireless networks. Chapter 4 gives the detailed steps of design of the metric of trust. In chapter 5, the physical implementation of the metric is done. The hardware set up is described. Chapter 5 goes on to describe the real world network implementation. Finally the implementation of the new metric is done and the results are analyzed. Chapter 6 concludes the whole thesis and proposes directions for future research.

Chapter 2

Routing, Security and Trust in MANET

2 About this chapter

In this chapter, an introduction and background of wireless network and challenges that arise in wireless internetworking are given. Afterwards some routing protocols in MANET are reviewed and challenges that arise in routing process are mentioned. Security issue in routing is discussed next and security attacks at network layer are recapitulated. Later trust is discussed as one of the solution to security problem in routing and data forwarding; its definition and measurements are discussed. Lastly our concept of trust is explained.

2.1 Introduction to Wireless Networks

2.1.1 Basic concepts in wireless networks

There are few basic network concepts that should be explained since they will be used often in this thesis. Many of the concepts are derived from Figure 1. The figure depicts a wireless network composed of nodes labeled A, B, C and D. Nodes represent mobile devices such as Personal Digital Assistants (PDA), portable computers, etc. Edges of the graph represent wireless links connecting devices to each other and each edge is associated with a cost (a penalty for using the edges). The big circle around each node represents the transmission range of the respective node, where the circle intersects shows where the transmission ranges over lap.

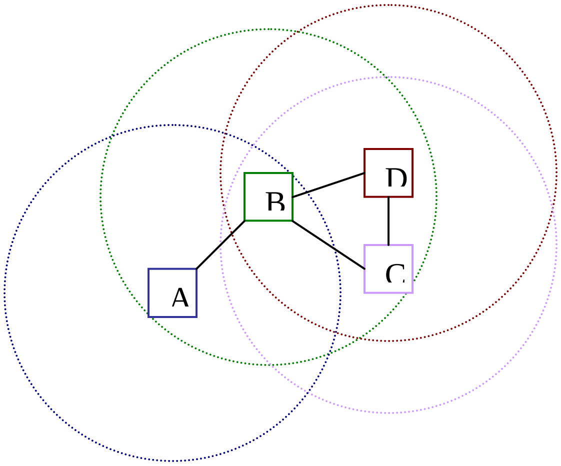

Figure 1: Wireless Ad hoc Network

Wireless link: a link in networking terms is the physical medium between two nodes and is used to propagate signals. In wireless networks, space is used as the physical medium to propagate radio waves, microwaves and infrared beams that transport signals. The signals are electromagnetic waves traveling at the speed of light [6].

Link metric: a standard of measurement of value of any specific characteristic of the wireless link. Costs are associated with links in this thesis. The *link cost* can be measured based on different parameters [6]. The link cost can be the number of hops between two distant communicators. Delay can also be a cost of the link and the link with small latency will be considered less costly, a link with high capacity measured in bits per second is considered less costly because it is easier to send data through such links and the delays are also significantly smaller. Similarly, current load carried by a link can be measured by considering the queue load and the link with a long queue is more costly.

Link faults: Natural incidents happen on the physical medium and they cause the link to fail to propagate signals correctly and accurately. These incidents result in link faults.

2.2 Wireless network

Shortly after notebook computers emerged, people started to have ideas of getting connected to their personal computers in the office and to the Internet via their notebooks. Such connections became practically possible only when both notebook and personal computers were equipped with short-range radio transmitters and receivers to permit them to communicate.

For notebooks and personal computers from different manufacturers to communicate, a wireless local area network (WLAN) standard has been designed. That standard's name is the IEEE 802.11, but it is commonly referred to as WiFi. The IEEE 802.11x suite of standards is aware of mobility of mobile devices and it is compatible with the Ethernet above the data link layer [6]. In the wireless LAN, IP packets are sent in the same way a wired network send IP packets over Ethernet.

The standard works in two modes:
1. *Managed mode*: In this case the base stations also called access points are in the middle of communication of all devices in the network. All communications are first sent to the access point before they are forwarded to the intended destination.

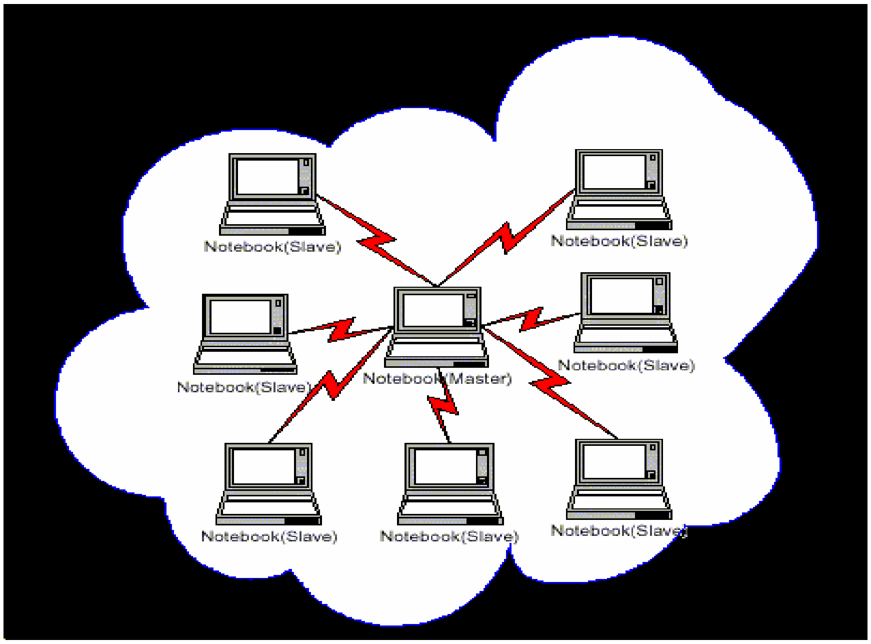

Figure 2: Managed Wireless Network [39]

2. Pure ad-hoc mode: In this mode mobile devices send data to one another directly. For example two people in a place where there is no wireless infrastructure would send information to one another directly via their mobile devices if they are in the same transmission range. That is one-hop connectivity and it is achieved via the data link layer by the use of wireless Medium Access Control (MAC) sublayer.

If two people in different transmission range want to communicate, one-hop connectivity is extended to multi-hop connectivity. Multi-hop connectivity is achieved via network layer using network layer routing and data forwarding protocol [7].

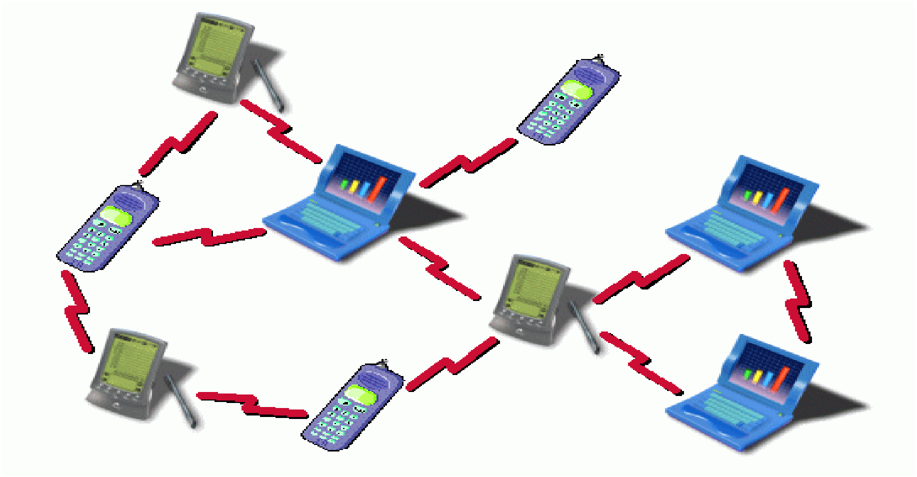

Figure 3: Mobile Ad-hoc Network [40]

2.2.1 Main challenges in wireless networks and ad hoc internetworking

In this section four main challenges in wireless and ad hoc networks are discussed. These are collision of frames, delay in frame transmission, interference in wireless links and violation of security goals.

2.2.1.1 Collision of frames

The MAC sub-layer allows one hop connection by using protocols based on Carrier Sense Multiple Access (CSMA) technique. A node using CSMA listens for other transmissions and only transmits when the channel is idle in order to avoid collision of frames. However the CSMA technique does not deal properly with problems like hidden node problem. The hidden node problem occurs when the receiving node is in the middle of the other two nodes. These two nodes are not in the same transmission range. When one of these two nodes senses the medium it cannot know that another node is busy transmitting to the receiver and this node also starts to transmit. The frames sent by these two nodes will then collide at the receiving node.

2.2.1.2 Delay in frame transmission

There is delay in transmission of frames at the MAC layer caused by the exposed node problem. That problem occurs when a node senses the medium and hears another node transmitting but in different direction as that one sensing the medium. Since the node sensing the medium can only know that its neighbor is transmitting, it waits to transmit while in reality it could start to transmit frames immediately without creating any conflict.

2.2.1.3 Interference on wireless links

In MANET wireless network interface of mobile devices may operate in promiscuous mode to allow connection to other devices within the same transmission range. Frames are sent using radio waves. Radio waves are not sent in one direction, instead they are sent in many directions and all devices in the directions of the radio waves can perceive them. If there are other sources of radio waves near the network, they might interfere with the radio waves transporting frames in MANET.

These interferences cause inconsistency in data transmission. Wireless links are also prone to breakage without any prior notice because of the physical objects that may be between communicating devices. The breakage of the wireless link can also be due to topology changes. Every time a link in use breaks, data are lost at some point and they have to be retransmitted.

2.2.1.4 Violation of security goals

Challenges of security in wireless and ad hoc networks occur on wireless links and on mobile devices (nodes). As is often the case, a secure computer network or system must provide services with the following security attributes as mentioned often in [8] and [9]:
Confidentiality: ensures that information is not disclosed to unauthorized users. Some information is very sensitive and can be used by adversaries for malicious actions.

Integrity: ensures that the information sent is the same as the information received without being corrupt on the way.

Availability: ensures that services of the network are always available. Nodes in the network should be available for relaying data for each other. Wireless links should not be jammed either by congestion or by intentional action of adversaries.

Authentication: enables nodes to verify if their peers are what they have claimed to be.

Non-repudiation: ensures that a sender cannot deny to have transmitted a message.

Anonymity: is also seen as a security attribute and is defined as state of not being identifiable in a set of entities. Anonymity can be defined in terms of *Unlinkability* which means a message in a communication cannot be linked to a particular user. *Relationship anonymity* can also be defined here as a not being able to trace who is communicating with whom [5].

In current wireless networks and mobile ad hoc networks, each one of the security attributes is not guaranteed all the time, and there is still improvement to be made in order to obtain a close to secure mobile ad hoc network. Any of the above security attributes can be lost at any time or it is not at all implemented in wireless ad hoc networks.

In wireless networks, the whole network relies on centralized trusted Certificate Authority (CA) for the management of public key certificates. If the CA is compromised, the whole network is at risk, the attacker is able to read messages of users on the network. In that case confidentiality is tempered with.

A secure communication among nodes is necessary to allow the integrity of the delivered packets. Nodes must be able to identify themselves to each other. A node must give their identification and associated credentials to another node to allow authentication. This information sent across the network must be well protected to ensure the integrity of delivered information. Each node must be able to validate the information received so as to verify if the sender is the one it claimed to be.

If the identity of anode is revealed to an attacker, the attacker can use it to impersonate a legitimate node or he can launch a denial of service attacks by keeping a node busy by sending a lot of dumb messages to that node. The legitimate node will be unavailable to other nodes that would need to use it for packets relay.

In MANET, security solutions must be decentralized and be integrated in routing protocols that run at each node in the network.

2.3 Introduction to routing in MANET

2.3.1 Routing in MANET

Routing is the process of exchanging packets of information between nodes in network. Packets are sent via the communication channels from source to the destination. That connection between source and destination is called route or path. The route is composed of at least two nodes; the *source* which is the node that initiates the communication and the *destination* which is the target to receive the communication. Some times the source and the destination are not in close proximity to each other to allow direct communication. In that case they bring into play *intermediate nodes* so that they can help in relaying packets and then a route will be composed of more than two nodes. The methods that nodes use to connect to each other and to forward packets for each other are handled by routing protocols.

A type of network is determined by many aspects, including: number of nodes participating in the network, equipment used as node, area on which the network is implemented, the purpose of the network, movement of nodes in the network, the life time of the network, users.

Networks have to be managed differently depending on their types. Different routing protocols are then necessary for different types of networks.

Wireless networks also have their particular routing protocols. These wireless routing protocols are different; depending on weather the network is managed or is ad-hoc and also depending on routing strategies used. The next section explains these concepts further.

2.3.2 Routing strategies in MANET

Factors like performance of the network, security implementation, scalability of the network, resources utilization etc are the basis of the choice of the strategy to use in designing the routing protocols in MANET.

2.3.3 Timing of route Discovery

Timing of route discovery is considered in routing protocol design when there is a concern of bandwidth consumption against performance of the network. Proactive or reactive techniques are used.

Proactive: proactive protocols are also referred to as table driven protocols because they use routing tables. They discover routes for each node to any destination in the network. All routes discovered are maintained (and the routing tables updated) regularly. If a route breaks, another route has to be discovered immediately. This technique has an advantage that nodes have routes to their destination all the time and data packets can be sent immediately as soon as the need arises without the delay to wait for the route to be discovered.

However, the technique has a drawback because of frequently changing topology of MANET, route discovery and route maintenance activities are also frequent. Bandwidth which is generally limited in wireless networks is then continuously consumed in discovering and maintaining routes that are likely to break before they are used.

Reactive: Reactive protocols are initiated on demand. i.e. routes are discovered only when the need to send data packets rises. Routes are only maintained if they are actively in use.

As opposed to proactive protocols, reactive protocols do not consume unnecessary resources for discovering routes that may not be used before they break. The drawback of this technique is that when a node needs to send data packets, it has to discover the route first; and this causes some delay in communication.

2.4 Organization of nodes in the network

Scalability of a routing protocol might be an issue when the network is distributed on a wide area and is densely populated. In addressing the scalability issue, performance of the network must also be thought of. Flat or hierarchical techniques might be used depending on what is the critical issue in the network.

Flat: flat routing protocols are for a network where all nodes can communicate at the same level. In these protocols each node can connect to every other node in its transmission range and each node can discover a route to any destination using the broadcast or multicast techniques.

Hierarchical: hierarchical protocols organize nodes into small groups called clusters.
Each cluster has a cluster head through which all nodes within the same cluster must connect before they reach other nodes outside their cluster. Nodes inside the same cluster do not have to communicate through the cluster head.

Hierarchical routing protocols may scale better than flat protocols in case a network is composed of a big number of nodes.

The overhead carried by the cluster head might be big since it has always to adjust to the mobility of nodes joining and leaving the cluster. A lot of information has to be shared between all nodes in the same cluster each time they leave the former cluster head and the existence of a new cluster. Other cluster heads in the network also have to be informed of the change of any cluster head. Nodes leaving a cluster use the cluster head as a handover node all the time, so nodes in the cluster need only interrogate the cluster head periodically to learn of nodes leaving.

2.5 Route Discovery and Maintenance

Routing protocols differ also depending on how routes are discovered and maintained as described below.

Link-state routing: a node broadcasts all routing information to all nodes in the network. Each node, however, sends only the entry of the routing table that describes the state of its own links. In link-state protocols, each node knows about the picture of the whole network and chooses the shortest path to use based on its view of the network.

Distance vector routing: during the route discovery process a node broadcasts a part of its routing table to its neighbours only. In distance vector protocol, nodes only know their neighbors and the length of the route to destination in hop counts.

Link state routing protocols are more scalable than distance vector routing, but the former requires more computation power and memory.

Dynamic source routing: in dynamic source protocols, a node discovers a route by sending broadcasting request messages to its neighbors. A node keeps the discovered routes in its cache. Each node keeps the sequence of all nodes to destination for each route.

The metric of trust we develop in this thesis is based on protocols that share the property of source routing for the reason that in source routing, data packets follow the same route. Hence packets are most likely to get to the final destination in order even in cases where the link breakage occurs, because if a link fails, data packets are sent via a different route from the point of failure or from the source of the packets. This is essential for data reconstruction and playback in the destination. It is also possible to estimate delay on all packets since they all follow the same route to the destination. Therefore for real-time applications, the average delay on each packet is about equal making it easier to transport and recompose real-time data. We also care about bandwidth consumption and hence we consider flat and reactive protocols.

2.6 Routing Protocols in Ad Hoc Networks

In this section we review four of the most popular routing protocols in MANET. There are many more routing potocols in current literature but are not of significant interest to us in this research.

2.6.1 AODV Overview

Ad hoc On-demand Distance Vector (AODV) protocol was designed to improve performance characteristics of DSDV (Dynamic Destination-Sequence- Vector) by minimizing broadcasts and transmission latency when new routes are discovered [10]. As the name suggests, AODV operate purely on demand. Each node does not discover the route till there is a need to send a packet or a node has to provide its service as an intermediate node in the route of other two communicating nodes.

AODV has its own distinguishing characteristics. AODV broadcasts discovery packets only when necessary. It can distinguish local connectivity; i.e. one hop away nodes for each node and the general topology. AODV broadcasts information about change in local connectivity to one hop away nodes that need to know about that information.

This protocol has four main functions; path discovery, route table management, path maintenance and local connectivity management.

Path discovery: that process is initiated whenever a node need to send packets to another node for which it has no route to follow. A local broadcast technique is used to discover the route by an initiator of a communication; broadcast route request (RREQ) packets are sent to one hop away nodes. Each node keeps in its routing table “a node sequence number” and a “broadcast id”. Nodes receiving the RREQ send the request reply message (RREP) if they are the destination otherwise they locally broadcast the RREQ received. By the broadcasting technique, it is possible for one node to receive the same RREQ more than once. If a node receives a RREQ it checks if it has already received a RREQ with the same “a node sequence number” and a “broadcast id”; if it finds out that it has already received the same RREQ, then the last one is dropped. Each node that is not the last destination keeps track of the following information in order to be able to transmit the RREP: destination IP address, source IP address, broadcast_id, expiration time for reverse path route entry and the source node sequence number.

As the RREQ travels from node to node, the information is saved and it automatically sets up the reverse path from the last destination to the initiator of the path discovery process.

When the final destination receives the RREQ it then checks if it has the route to the previous sender in its table by checking the source sequence number. If that final destination finds an equal or greater sequence number for the previous sender, then the former sends the RREP to the later. Otherwise the final destination broadcasts the RREP to all one hop away nodes and the process goes on till the RREP reaches the initiator of the path discovery. As the RREP travels the network each node sets the forward pointer to the node from which the RREP came from. Nodes that have the RREQ entry but do not receive the RREP, delete the reverse pointer after a time out period. The same for the RREQ, a node only forwards the RREP if it has done it before or if the sequence number is greater that what is contained in its table. The initiator of the path discovery process finally receives the RREP and it can now start to communicate with the intended destination.

Route table management: there is a so called “soft state” associated with the route table entries from the path discovery process. There is “route expiration timer” whose role is to eliminate the reverse routing entries for the nodes that are not part for the route from a source to a destination. “Route caching time out” indicates the period of time a route is supposed to be valid. Nodes are part of the route if they are only active, otherwise after a period of time without forwarding any packet, they are said to be inactive and in that case they can not be used in relying packets. The node becomes inactive after the “active time_out” expires. All routes in the routing table are tagged with destination sequence number that ensures that no cyclic loops are formed.

When a node receives a new route to the same destination, it only uses that route, if it has a greater sequence number or if it has the same sequence number but fewer hops to the destination.

Path maintenance: periodic “hello” messages are used to detect link failure [10]. When a link fails, a RREP is sent from the point of failure. That RREP must have greater sequence number than the failed route and must have (infinite sign) as hop count. This will allow all nodes receiving this RREP to delete that route from the route table. Any node upon receipt of the RREP because of the failed link, another route from the table is chosen if it exists or a route discovery process is started again if there are still packets to be sent.

Local connectivity management: a node knows its one hop away neighbors by getting broadcast messages when routes are being discovered. Each time a node gets broadcast messages, it checks the source sequence number and updates the entry of its neighbors in the routing table if necessary. A node can also learn about its neighbors by receiving and sending periodic hello messages.

2.6.2 DSDV Overview

Dynamic Destination-Sequence-Vector (DSDV) routing operates in many similar ways as AODV described above. In DSDV packets are sent between nodes in the network based on the information stored in routing table of each node [11]. The route discovery process is done by broadcasting the request to the network as it is done in AODV. However there is a difference in the route table entry. In DSDV the route table at each node contains the list of all available nodes and the number of hops to each. Each route table entry has a sequence number that is used to verify the freshness of the route. Periodic update messages are sent to neighboring nodes to ensure that the links to those neighboring nodes are still working and that nodes are still active. These periodic updates are used to update the entry of route table whenever there is relevant new information available. When there is topology change, routing information is advertised by broadcasting messages that are sent periodically. In order to avoid unnecessary and bandwidth consuming fluctuations of route tables, the difference in time between the arrival of the first route and the best route is recorded. Based on this record, a decision to delay the advertisement of the route that might change soon can be made to allow other routes with the same sequence number and that might be more stable to be advertised instead of the unstable one.

In DSDV only bidirectional links are considered. A node does not add entry in its route table from the messages coming from a neighbor, unless the neighbor is also able to receive packets from that node.

2.6.3 DSR Overview

DSR is a routing protocol that is designed for use in a multi-hop environment like wireless mobile ad hoc networks. It allows mobile nodes to organize and configure themselves to form connections between them without any aid of an existing infrastructure or administration. DSR routing protocol reacts to the change of topology of the mobile ad hoc network caused by mobility of mobile nodes in the network or by interferences on the wireless communication links [12].

DSR allows a pair of nodes (source and destination) to communicate even if they are not in the same transmission range by using intermediate nodes “hops”. Each source knows the route(hops through which data packet has to pass) to any destination in the network.

Each data packet sent from source carries in its header the ordered addresses of all nodes composing the route; source node, intermediate nodes and the destination.

In DSR like in most routing protocols, two mechanisms are implemented together for the functionality of the protocol: *route discovery* and *route maintenance.*

Route discovery: if the source node wants to communicate to a destination node to which it does not know the route to, the source node has to use the route discovery techniques to find the route to the intended destination.

Route request (RREQ) message is broadcasted by an initiator of the route discovery process to all nodes within its transmission range. If a node receive the RREQ and it is not the intended destination or the "target" it also broadcasts the RREQ further and the process goes on till the target receives the RREQ. The two important information in the RREQ are "route record" and "request id". Each RREQ message contains "route record"; the sequence number of all nodes through which the RREQ was sent during the route discovery process. Each route contains a unique "request id" set by the initiator of the route discovery process. Each node keeps a list with two entries (initiator address, request id) to be able to detect duplicate routes.

Upon receipt of a RREQ message, a node checks first its list. In the first case, if a route with the same information as one of the entries in the list, then the route is not processed further and it is discarded. Discarding a route when it is already in the list ensures that one single RREQ does not get propagated endlessly, thereby forming loops. In the second case, if a node receives a RREQ and its address is contained in the "source record", that route is also discarded and it is not processed further, this avoids redundant routes for a single node. In the third case, if the node's address corresponds to the target address in the "route record" then the RREQ has reached its destination and the route reply (RREP) message is sent with a copy of the route to the initiator of the route discovery process. If the target receives the RREQ and if it already has a route to the initiator of the RREQ, it may use that route to send the RREP. Otherwise the target will reverse the route used to send the RREQ and sends the RREP via that route but this is only possible if the links on all intermediate nodes are bidirectional. If the target does not know the route to the initiator of the route request and if links are not bidirectional, then the piggybacking approach is used. The RREP messages are piggybacked on a RREQ message targeted at the initiator of the route discovery to which it is replying.

If none of the three cases are true, the node then adds its own address to the "route record" and rebroadcasts the RREQ message.

Route maintenance: if a route fails at any hop in the path during a communication between source and destination, the node encountering the error sends an error message to the originator of the route. The failing route is detected when a node fails to forward a packet by using periodic broadcast messages. When a route error is received, the node experiencing the error is removed from the cache of the node receiving the error message. All routes containing the node in error must be shortened at that point. If links do not work equally well in both directions then end-to-end acknowledgment is used to detect the failing link.

If the source knows any other available route, it uses it. Otherwise a route discovery mechanism is started.

2.6.4 TORA Overview

Temporally-Ordered Routing Algorithm (TORA) is a distributed routing protocol for mobile, multihop, wireless networks [13]. TORA has the capability to minimize reactions to topological changes in MANET. The protocol is composed of three main functions that are route creation, route maintenance and route deletion.

Route creation: A sequence of links is created from source to destination. During this process directions are assigned to links from source to destination in the network, and in that way a Directed Acyclic Graph (DAG) is created. Source node and intermediate nodes have downstream links and the DAG is said to be destination-oriented. Control packets; query (QRY) packet and update (UPD) packet are used in creation of route. Source node and intermediate nodes in the route keep the route-required flag (RR) which is initially unset. Those nodes also maintain the time each UPD packet was broadcasted and the time at which a link was active. A node that does not have a direct link or RR flag broadcasts QRY packet to its neighbors and sets the RR flag. If a node receives a QRY and has no directed link, it rebroadcasts the QRY packet and set its RR flag. If a node receives a QRY packet but has no directed link but has RR flag set, it means that it had already received that QRY packet and it does not rebroadcast it, instead it discards it. If the receiving node of QRY packet has a directed link and its height in NULL, it means that it has not yet received that QRY packet; it sets its height as described in [13]. If a receiving node of QRY packet has the directed link and the height is not NULL then, the node have to check if the QRY packet is the most recent one by comparing the time the UPD packet was broadcast and the last time the link by which the QRY packet was sent was active. If the UPD packet was sent since the link was active, it means there are no updates necessary and the QRY packet is discarded, otherwise the UPD packet is broadcasted. A node that sets the RR flag after re-establishment of a new route, broadcasts a QRY packet.

When a node receives a UPD packet, it adjusts the necessary information that composes the height of the route; that includes setting the RR flag and link-state array.

Route maintenance: Routes are maintained as a reaction to topological changes. When topological changes affect a route, a route to destination is re-established by creating a destination-oriented DAG within a finite time. Different cases lead to route maintenance when an existing link fails and a node is left without any link to neighbors for a certain route. TORA uses reference levels to categorize neighbors that form a portion of a network. Different portions of the same network use different reference levels and the reference level is used to detect a network partition at some points. Reaction to topological changes is only done by nodes in the affected area or level.

Route deletion: TORA can detect network partitions and delete broken routes by removing direction that are previously assigned to links that have become invalid. A clear (CLR) packet is broadcasted by a node that experiences a failing link. All nodes that receive the CLR packet first checks if the CLR is a recent one and deletes the failing link as detailed in [13].

2.7 Problems and challenges in ad-hoc routing

2.7.1 Dynamic topology of MANET

Frequently changing topology of network due to mobility of nodes in the network or failure of wireless links makes routing a difficult task in MANET. As nodes move around in the network, its connections with other nodes break because the area of transmission changes and new connections to other nodes are established.

Nodes are switched off when they want to save batteries, and switched on again when they want to communicate, that also contributes to the changing topology of MANET. As network topology changes, error messages have to be sent to the network and new routes have to be discovered when old ones break. Routing algorithms for MANET do not only have to be adaptive to the changing topology but they also have to be considerate of the scarce resources like bandwidth and processing capability of mobile devices.

2.7.2 Cooperation of nodes in the network

In MANET, nodes have to cooperate in order to allow connectivity and proper functionality of the network. Most routing protocols in MANET are based on the assumption that nodes will always cooperate to forward packets for each other. In practice that assumption does not hold all the time. Nodes participating in the routing mechanisms in the network may not comply with routing protocols for various reasons. On one hand a node may be selfish by refusing to participate in the forwarding process of packets for other nodes in order not to exhaust its battery. In that case a selfish node simply drops all the packets that are not destined to it. On the other hand a node may participate in the routing process but perform malicious actions during the process. When nodes do not comply with routing protocols, it results in disruption of the routing process itself or in improper delivery of data packets.

2.8 Security Issues in routing

Researches have been done to explore security problems in MANET and various solutions to security issues at different network layers have been proposed [14]-[20]. Our work is only limited to exploration of security issues in the network layer. Two main issues here are detection and prevention of security attacks on routing and forwarding protocols. When designing solutions to security problems, other problems mentioned earlier must also be taken into consideration and thus becomes a tricky equation to solve. In [7] security problems have been summarized on different layers of the wireless ad hoc network as indicated in Table 1.

Layer	Security Issues
Application layer	Detecting and preventing viruses, worms, malicious codes, and application abuses
Transport layer	Authenticating and securing end-to-end communications through data encryption
Network layer	Protecting the ad hoc routing and forwarding protocols
Link layer	Protecting the wireless MAC protocol and providing link-layer security support
Physical layer	Preventing signal jamming denial-of-service attacks

Table 1: Security Issues in Ad Hoc Networks [7]

A secure solution will be the one that will be able to detect malicious behaviours in the routing and forwarding mechanisms and react to those misbehaviours.

Initially ad hoc routing protocols like DSR [12], AODV [10] and DSDV [11] assume that all nodes in the work cooperate in relaying packets to each other so that connectivity and integrity of the forwarded data can be maintained. In practice, MANET environment is hostile because of the openness of peer-to-peer architecture. When nodes do not comply with the routing protocols, the risk of security threats on routing mechanisms or on data traffic is high.

2.8.1 Security attacks on routing mechanisms

Security attacks may be put into categories because of the properties they share and because of how they affect the routing process.

Modification: In *source routing* protocols, for example in DSR [12], the attacker modifies the information in the header of the route discovery packets (RREQ or RREP). The attacker may change the list of nodes in the packet header by deleting a node, interchanging the order of nodes, or inserting a new node [21].

In case of *distance-vector routing* protocols, for example in AODV [10], the attacker may advertise a false route. The attacker can advertise a route with a smaller distance metric than its real distance to the destination in order to attract the traffic to himself. The attacker can also advertise false routing updates with a large sequence number and invalidate all the routing updates from other nodes [22].

Masquerading: An attacker can forge a packet with identification and impersonate a legitimate node [23]. This kind of attack will subvert the authentication. This attack is easily possible in almost all traditional ad hoc routing protocols because they do not verify the information and routing messages (RREQ, RREP, RERR) are trusted by default. The attacker impersonates a node and sends a RREQ with the address of the legitimate node as the source address. For example in AODV, because the sequence number of a node can be set, the attacker can put a much bigger sequence number so that the lifetime of the RREQ is long and this will create loops in the network.

The attacker may also render neighbouring nodes inaccessible by reporting non existent broken link by forging the route error messages (RERR). This can happen in case of on-demand ad hoc routing protocols, when the attacker targets the route maintenance process and advertise that an operational link is broken [21].

Tunneling: Attackers may collaborate to launch a denial of service attack by preventing a source node from finding any route to the destination. In the worse case, the attacker will cause partition of the network [7].

Colluding adversaries may create a wormhole attack [24] and shortcut the normal flows between each other. The attackers must have a cryptographic material. One attacker listens to the message on one part of the network, and the colluding attacker helps to replay the message on the other part of the network [25].

Rush attack: In all on demand protocols, an attacker can render all nodes incapable of finding routes longer than two hops. That is one intermediate node between sender and receiver [26]. In on demand protocols, a node wishing to find a route to a destination sends RREQ to neighboring nodes. Each neighboring node also forwards messages to its neighbors and so on. Using that technique, one node can receive same RREQ many times because one node is a neighbor to more than one node at the same time, but each node only forwards RREQ that arrives first and others are discarded. In order to avoid collision, normally there are compulsory delays between when a packet arrives at the interface for transmission at the link layer, and when the packet is actually transmitted. If the attacker does not respect that delay, its RREQ will arrive first and therefore will be forwarded. All other RREQ of the same route discovery arriving at the same neighbors will be discarded. This will result in a denial-of-service attack. The source node will be unable to discover any other usable route except the one that includes the attacker.

2.8.2 Security attacks on data traffic

In addition to routing attacks, the attacker may disrupt the packet forwarding operations in a way that data traffic will be noticeably affected. Attacks in this category aim to identify a particular communication between a sender and a receiver. The attacker introduces a particular pattern in the traffic that he can follow and that can allow him to trace a communication that would be otherwise very difficult to identify. These kinds of attacks cause delivery of data packets to be purposely inconsistent with the routing states. These packet oriented attacks affect timing or quantity of transmitted packets. These attacks are a threat to privacy of nodes engaged in a communication in MANET.

Delete attack: the attacker consistently deletes *n* packets and he follows a particular pattern to delete packets. For example the attacker may delete a packet, every certain number of packets or every period of time.

Delay attack: this attack is done in the same way as in delete attack but instead of delete, *n packets* are delayed regularly.

Insertion attack: the attacker inserts *n* packets in the traffic. These packets might be replayed packets or new packets.

The above mentioned attacks change the status of the traffic in the same way a natural fault occurrence might change the status of the traffic. It is hard to conclude that an attack happened on a link by just observing the behavior of traffic and that is why till now there are no solutions to prevent such attacks.

2.9 Introduction of Trust into routing

Difficulties with the solutions to the attacks mentioned above have led to a change in strategy. Therefore, in MANET trust has been the center of solutions to security threats in routing. Trust is used in different contexts and it is regarded as very important in any interaction between two different entities.

2.9.1 Definition of Trust

Trust is a concept we encounter in everyday life. It is a fundamental aspect on which we base our interactions, transactions and communications in our daily life. Everything we do in human life is the risk we take based on trust we have for each other, or trust we have for systems or equipment we use. In different fields trust is defined in different ways depending on the context in which it is used.

Trust in the Cambridge international Dictionary of English is defined as follows [27]:
“to have belief or confidence in the honesty, goodness, skill or safety of (a person, organization or thing)”

In Psychology trust was defined in [28] as confidence, which is confidence that one will get what is wished from another, rather than what one is afraid of.

In sociology, the author in [29] defines trust as a means of reducing complexity of society that becomes more and more complex. When one faces a decision making situation, one has to make some assumptions taking into account a particular situation and a particular environment and then make some trusting choices [30].

In mathematical terms, trust has been defined in [31] as follows:
“Trust (or, symmetrically, distrust) is a particular level of the subjective probability with which an agent assesses that another agent or group of agents will perform a particular action, both before he can monitor such action (or independently or his capacity ever to be able to monitor it) and in a context in which it affects his own action.”

The introduction of probability in the definition of trust makes trust more concrete than abstract-like as it has been defined in psychology and in sociology.

Thus trust can now be measured with a mathematical model. Trust can be placed on or be represented with a probabilistic distribution with different values of expectations. This definition recognizes that trust is applicable where there is probability of distrust, betrayal, exit or defection [30]. Probabilistic distribution of trust can have a range of values from the lowest value representing distrust to the highest value representing trust where the middle point represents uncertainty [31]. Our theoretical probabilistic model of trust detailed in chapter four is developed based on that.

2.9.2 Estimation of trust

There is yet no unit in which trust can be measured, but it is still possible to estimate its values. To be able to estimate the trust level, in many cases trust level has been associated with a trust relationship [32]. Different metrics of trust in networks have been designed for public key authentication [33], [34]-[36] and in peer-to-peer networks in [49] and [50]. In some of these metrics trust has been estimated using linguistic descriptions of trust relationships [37]-[41] as mentioned in [31].

For example in [35] trust values are assigned to three aspects of each key and two of the three aspects are *Owner trust* and *Signature Trust*. Owner trust and Signature trust are measured as undefined, unknown user, usually not trusted, usually trusted, always trusted and ultimate. The third aspect is *Key legitimacy* which is measured as trusted, not trusted, marginally trusted and completely trusted.

In other metrics, trust has been estimated using numerical values [34], [35], [39] and discrete or continuous numerical values are allocated to the level of trust [42]. For example in [35] trustworthiness is measured as a triplet as belief, disbelief and uncertainty {b,d,u} that belongs to the range $[0,1]^3$. In [35] trust is represented from –1 to +1 signifying a continuous range from complete distrust to complete trust.

2.10 Our concept of trust

As it is detailed in the following chapter, most of the proposed trust models in MANET [56, 57, 58] are based on the approach of nodes watching their neighbors and reporting to the network members the suspicious behavior detected. These trust models measure trust of nodes towards each other in the network. They can detect a malicious node and isolate it.

Trust is based not only on knowing an entity and having evidence of behavior for that entity in the past in order to take risk for future interaction, but also every human being has right to privacy . Consequently it is important to include privacy in any type of network.

If the goal of “anyone anywhere any time” must be enhanced in MANET, then privacy should be preserved so that users can feel free to participate in any network. When privacy is preserved then real users’ identity should not be revealed nor linked with their location or with their activities. Research is already focusing on the anonymity issues. In [44] route anonymity and location privacy are preserved. In [45] a distributed path control protocol for anonymizing communication in ad hoc networks is developed.
However users may decide to trade privacy against trust depending on what they value most. In that case at least pseudonym should be used in collecting evidence instead of real identity like it is done in [46].

The argument in this thesis is that when it is not possible to identify a particular node in the network, at least communicating partners (source and destination) in the network should have freedom to locally analyze their communication traffic patterns and derive significant conclusions out of the patterns.

In this thesis we conceive a communication link as the whole path or route from source to destination, because if anonymity is implemented, then a node will only know what is happening on its side and will not know who are on the other nodes nor in the intermediate nodes and what they are doing.

Our work measures trust of communicating partners towards the communication link they use based on status of traffic patterns of the communication. Traffic patterns cannot be conclusive, since link faults are likely to happen in MANET and their results on traffic behavior is the same as some security attacks. However, by observing the change of traffic patterns for a period of time, a statistical analysis can be used to detect any mischievous actions and measure the trustworthiness of a communication link.

2.11 Conclusions

In this chapter, we have reviewed four of the most popular routing protocols in MANET and highlighted their strengths and weaknesses. The choice of DSR as the protocol to use for our work is shown to be based on its properties of permitting a complete path to be discovered between the source and destinations before communication is initiated between them. Since it supports multi-hop communications, it also permits new paths to be discovered in times of attacks on paths or data routes.

The author also overviewed the security problems in MANET and showed that the key to creating confidence in ad hoc networks is introduction of trust constructs to security of the paths and the networks as whole. Our work focuses mainly on anonymous communications and the security of the paths and with the use of traffic patterns as seen by the source and destination. This is significantly different and novel compared to techniques which rely on observable communication between the source and destination whereby the intermediate nodes are able to interpret the traffic patterns. In our case the intermediate nodes are completely oblivious of the traffic patterns they forward or they are incapable of interpreting them.

Chapter 3

Trust Models in MANET's Routing Protocols

3 About this chapter

Chapter 3 reviews routing protocols that have incorporated trust in their functions in order to enhance security of MANET. Next, it outlines functions of trust models in each protocol and gives a summary of the protocols in two categories based on how trust is established. Finally, it briefly shows how trust models in discussed protocols are different from the metric of trust designed in this thesis.

3.1 Trust models based on distributed recommendation

The following sections overviews protocols that have elements of trust constructs in them. They are mostly modifications of the MANET routing protocols discussed earlier in this thesis.

3.1.1 Trust Model for TAODV

Trusted AODV adds a trust model to an existing AODV protocol to secure routing process in ad hoc networks. Trust among nodes in TAODV is represented by opinions derived from the subjective logic [51].

In TAODV opinion of node A about node B is defined as $w_B^A = \left(b_B^A, d_B^A, u_B^A\right)$ where this equation represents any node A's opinion about any node B's trustworthiness in a MANET, the first, second and third components correspond to belief, disbelief and uncertainty respectively. These three components are related as in the following equation:

$b_B^A + d_B^A + u_B^A = 1$ **(Equation 1)**

Here, belief (**b**) or disbelief (**d**) is the probability that node B can be trusted or not be trusted by node A respectively. When there is neither belief nor disbelief then node A is uncertain (**u**) about the behavior of node B. The sum of the three components is equal to 1.

Based on positive and negative evidences about other node's trustworthiness that a node is able to collect and record, the opinion value can be obtained as defined by mapping opinions as follows.

Let w_B^A be node's A ***opinion*** about node's B trustworthiness in a MANET, and let p and n respectively be the positive and negative evidences collected by node A about node B's trustworthiness, then w_B^A can be expressed as a function of p and n as the following system of equations:

$$\begin{cases} b_B^A = \dfrac{p}{p+n+2} \\ d_B^A = \dfrac{n}{p+n+2} \\ u_B^A = \dfrac{2}{p+n+2} \end{cases} \text{, where } u_B^A \neq 0 \qquad \textbf{(Equation 2)}$$

Four procedures are at the core of trust related operations in TAODV and explaining them gives a clear understanding of TAODV.

Trust recommendation

A node issues a trust request message to its neighbors when that node wants to get information about another node's trustworthiness. Nodes that receive the trust request message return trust reply message to the node that issued trust request message. Throughout a communication, when a node believes that another node is acting maliciously, it broadcasts a trust warning message.

Trust combination

A node uses collected neighbors' opinion about another node of interest, and combines those opinions in order to be able to make a relatively objective judgment about that node of interest. Trust combination is done in two main processes; discounting combination and consensus combination.

"Discounting combination" consists of a recommendation a node gets for another node from a third party. For example, let's say A has opinion about B and B recommends C to A, then A combines its opinion about B and the B's opinion about C in order to get its opinion about C.

"Consensus combination" is used for a node to combine opinions from different nodes about a particular node.

Trust judging

There are trusts judging rules set that are used for a node in order to make a decision to update trust. Node A and node B are taken as an example.

1. In node A's opinion if the belief value b_B^A is greater than 0.5, then node A trusts node B, and node A do the routing related to node B.
2. If in node A's opinion the disbelief value d_B^A is greater than 0.5, then node A does not trust node B, and it stops all routing related to node B. Then the routing entry of node B is deleted from the routing table of node A after an expiration time.
3. If in node A's opinion the value of uncertainty u_B^A is greater than 0.5, then node A will request digital signature before having any interaction with node B.
4. If in node A's opinion the three values of belief, disbelief and uncertainty are less than 0.5, then node A requests digital signature before any interaction with node B.

Trust updating

As nodes communicate, some forwarding actions succeed and others fail. The opinion among nodes varies depending on success or failure of forwarding actions performed by a particular node. The following policies are used to update trust opinion whenever there is a success or a failure in the forwarding process. The same example of node A and node B is used.

1. Whenever node A performs a successful routing operation with node B. Node A increments the B’s successful events by 1.
2. Whenever node A performs a unsuccessful routing operation with node B. Node A increments the B’s failed events by 1.
3. The value of opinion is recalculated whenever there is change of successful event or failed event values.
4. If there is no current route of node B in node A’s routing table, then the opinion value w_B^A is set to the initial value; (0, 0, 1).

3.1.2 Cooperation Of Nodes: Fairness In Dynamic Ad-hoc NeTworks (CONFIDANT)

CONFIDANT [52] is an extension to the DSR protocol and it aims at detecting and isolating misbehaving nodes in order to discourage the uncooperative behavior of nodes during the routing process. CONFDANT is composed of four main components that are:

The monitor

With the aim to detect a non-compliant participant in the network, nodes watch their neighbors when the later are forwarding packets. When the behavior observed is different from the behavior expected, a node failing to behave as expected is reported by sending the ALARM message to warn other nodes.

The trust manager

It is the responsibility of the trust manager to send the ALARM message and to receive them. ALARM messages are sent by nodes that experience or observe misbehavior, and a node that gets an ALARM also broadcasts it to other nodes. Each node has a list of friends from which ALARM messages can be accepted. Before any reaction is taken after receiving an ALARM message, the trustworthiness of a friend sending the message is verified.

The reputation system

Each node has a local list of friends and their rating and a black list containing nodes with bad rating; these lists are shared among friends so that rating can be decentralized. The reputation system manages a table that contains a list of nodes and their rating. Rating of a node is only changed if there are enough evidences about the malicious behavior suspected. i.e. if the malicious behavior happened more than times that are acceptable as number of accidents. The rate is changed according to the way the behavior has been detected. The greatest weight is assigned to own experience, a small rate for the observation in the neighborhood and a smaller rate for the reported behavior.

The path manager

The path manager ranks the routes according to reputation of the nodes in the path and it deletes routes that contain malicious nodes. The path manager also takes action when a malicious node requests a route. Usually a malicious node is ignored when it requests a route.

3.1.3 CORE: A Collaborative Reputation Mechanism for node cooperation in Ad hoc Networks

CORE adopts the approach of reputation as a foundation of security mechanism to solve problems caused by misbehaving nodes in the network [53]. As it is done in TAODV and CONFIDANT, in CORE also nodes observe their neighbors or get information about the behavior of other nodes in the network from network members. Reputation is formed by combining information a node gets from its own experience and information that node gets from other nodes about a particular node of interest. A final reputation of a node is formed by a combination of three types of reputation; subjective reputation, indirect reputation and functional reputation.

Subjective reputation is got from direct current observations and past experience with a node and its neighbors. Past experience is given more weight for irregular misbehavior not influenced by the final reputation value. Subjective reputation is calculated at a particular time by a particular node. In calculation of subjective reputation, a past values is more relevant. The past value is gotten from a number of observations, and a scale from -1 for an unexpected behavior observed to + 1 for an expected behavior observed is used. When the observed behaviors are conclusive enough during a certain period of time, then the value 0 is used for neutral experience.

Indirect reputation is calculated with the aid of information that a node gets from other members of the network. Negative information from members of the network are not considered, thus indirect reputation can only take positive values. The cause of consideration of positive values is to avoid denial of service attacks by providing negative reports about well behaved nodes.

Functional reputation is the combination of direct and indirect reputation with respect to different tasks like packet forwarding or routing that nodes are usually supposed to perform for each other.

3.2 Trust models based on group security

3.2.1 A Security-Aware Routing (SAR)

Security-Aware Routing (SAR) introduces a method that incorporates security levels into routing mechanisms. SAR categorizes nodes and explicitly defines trust values for each category. Trust values are used and to make routing decisions [54]. During communication, routing information concerning trust value is secured so that it is not possible to change route behavior of a secure route discovered. Authors in [54] argue that quality of protection and security attributes to route metrics must be specified because some applications require routes not only to be the shortest but also to be secure.

SAR protocol is based on any on-demand ad hoc routing protocol like DSR or AODV and it has two main goals. Discovering routes with security levels and protecting information on transit so that security levels cannot be altered.

Discovering routes with security levels

SAR extends route discovery process of the existing on-demand routing protocol to include security levels into routing. Security metric is imbedded into RREQ message itself, and the forwarding behavior is changed with respect to RREQs. When a node receives a RREQ message during the route discovery process, only nodes with required security level can process that RREQ message. A RREP message is sent back with appropriate modifications after an end-to-end connection.

Trust levels are incorporated in routing based on the hierarchy organization of nodes in the network. A number is then associated with each privilege level. For example in an organization, people on different levels have different privileges and these levels can be used in particular organizations to incorporate trust level in routing of a communication network.

Protecting information on transit

In SAR, level of protection is developed and it is associated with security of information in transit in routing protocol. User and identity are bound and their associated trust level in order to avoid nodes that can impersonate other nodes and use their privileges if they have higher trust level. Cryptographic techniques are suggested as the method to use for authentication in order to combat the impersonation. SAR supposes that a node cannot interrupt with routing that is not in its security level, because a node process routing if it is for its own security level, otherwise it has to drop packets. SAR proposes the use of digital signature and encryption in order to guarantee the integrity of routing protocol packets.

3.2.2 A Secure Distributed Anonymous Routing Protocol (SDAR)

A novel secure distributed path construction protocol is presented in [55]. The protocol is aimed at providing anonymous communication in wireless ad hoc networks by using multicast and layered encryption techniques. The protocol introduces trust level during route discovery process. A node broadcasts a route request message to its neighbors with a trust level required. Only neighbors satisfying trust level required rebroadcast the route request message if they are not the destination. Before neighbors rebroadcast any route request message, they add their encrypted ID and a session key. This process goes on until the message reaches an intended destination. Upon receipt of a route request message, the destination sends back a route request reply with the information about intermediate node encapsulated in layers. The route reply message follows the reverse path that the route request message followed. Each intermediate node removes one encrypted layer before forwarding it to its neighbor towards the source node.

A trust management approach used to set trust level for a route is based on the past behavior of a node. Trust is computed by direct neighbors according to past experience with a particular node or according to current observed behavior of a node. When a node behaves as it is expected trust increases, otherwise trust decreases.

A node knows about its active neighbors by sending periodic HELLO messages. That node and its neighbors form a community. Neighbors are classified into categories according to their trust levels. There are three categories according to trust level values empirically determined; low, medium and high.

A central node in the community generates one key for the high level category and another key for the medium category. All nodes in the high level community will have the key for their level and the key for the medium level. All nodes in the medium level community share the same key of their level only. Nodes in the low level community do not share any community key at all.

When a node joins a neighborhood, a central node assigns an initial trust to it and trust value can increase or decrease based on interaction with other nodes in the neighborhood. For protection of nodes in communities, keys are renewed when a node leaves a community either because a trust level is decreased or because a node leaves a neighborhood.

3.3 Review

Each one of the above protocols has provided a solution to some of the security issues but yet, none of them is a complete solution. [50] and [54] enhance security of routes by incorporating trust in route discovery process and furthermore they have no considerations for covert or unobservable communications between two end points. Route discovery process is resistant to attacks that modify or fabricate routing information with the aim of denial of service attacks. Once routes are discovered in compliance to those protocols, they are considered to be secure for any communication. To implement surety at the stage of route discovery is vital. It is as well as necessary to consider the changing behavior of a node that might comply with route discovery process but does the opposite for packet forwarding.

CONFIDANT and TAODV consider the changing behavior of nodes in the network at anytime. They adjust trust of nodes towards each other at every interaction. However, these two protocols have not considered nor distinguished between malicious node behavior and problems caused by traffic congestion or benign link failures which are the most likely causes of routing failures in mobile ad-hoc networks. Even if a failing node might be regarded as useless as malicious node, but at least when a failing node recover from error it should not be regarded as harmful to communication of other nodes in the network. We recognize that distinguishing a malicious node and a failing node because of error is still an open problem.

Group security provides an avenue for new attacks and security risks. In cases where users of mobile devices are roaming in insecure environment, nodes can be captured while already being in use. For example in SAR and in SDAR if one or more users of the same trust level group are compromised, it exposes all users of the same group to security attacks.

As nodes will watch their neighbors forwarding packets and report to other nodes in the network. Nodes giving false report about their neighbors in CONFIDANT can force other nodes to be excluded from the network.

As negative reports are not considered in CORE, a node can have rapidly increasing positive credits if there is a node reporting false positive information about that node. That positive credit that a node gets will not reduce in the same ratio as it has increased since only positive reports are considered. Nodes may collaborate to give false positive reports for each other and they may be able to remain in the network even if they are behaving maliciously.

3.4 Summary

There are Common basic functions of the reviewed trust models as they are summarized in Figure 4 and Figure 5. In each model, these functions are executed differently but here we give a general view. Figure 4 illustrates discovery of secure route based on the security level explicitly assigned at different users in the network. A node can participate in the route discovery process only if it has the required security level.

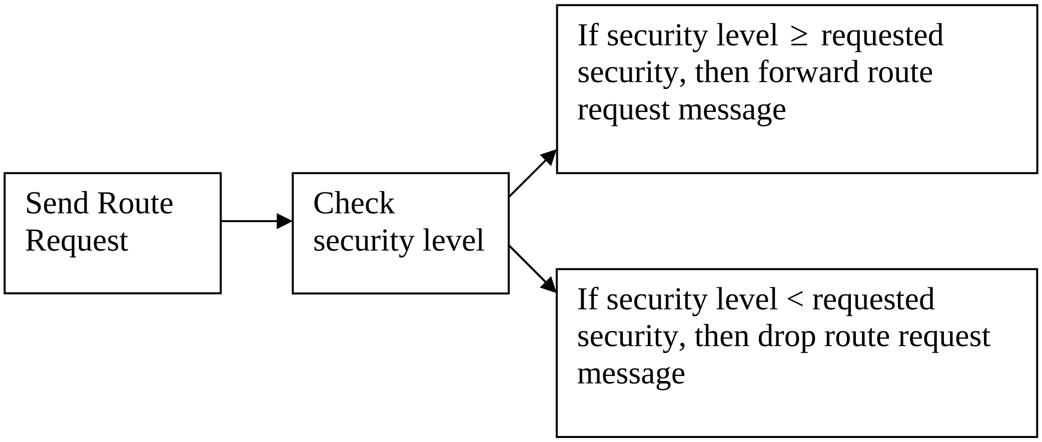
Send Route Request
Check security level
If security level ≥ requested security, then forward route request message
If security level < requested security, then drop route request message

Figure 4: Trust models based on group security

Figure 5 points out the most common functions of the models based on distributed recommendation. Nodes watch their neighbors during a communication and a report is sent to the members of the network. Each node updates trust of its neighbors by combining the report about neighbors and that node’s experience with that neighbor. If a node’s experience is less that a certain threshold, then that node is excluded from the network.

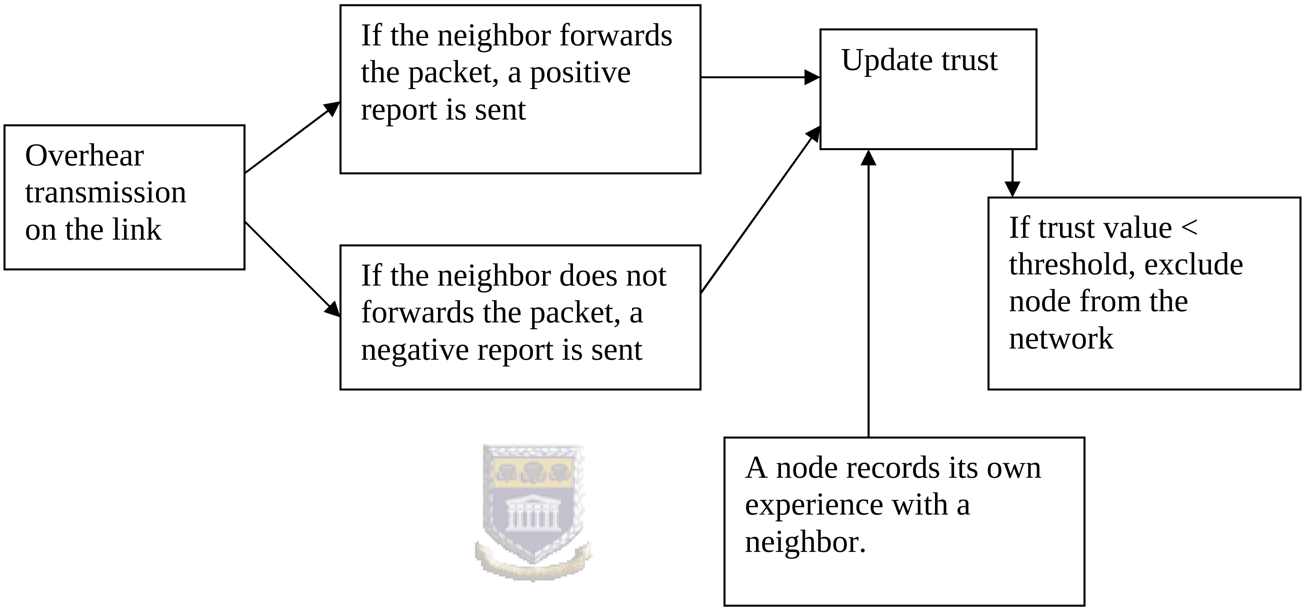
Overhear transmission on the link
If the neighbor forwards the packet, a positive report is sent
If the neighbor does not forwards the packet, a negative report is sent
Update trust
If trust value < threshold, exclude node from the network
A node records its own experience with a neighbor.

Figure 5: Trust models based on distributed recommendation

Our metric of trust differs from the previous trust models in the sense that it considers an environment where only two end nodes collect evidences on the communication. Therefore only two end nodes in a communication are involved in the process of updating their opinion on the trustworthiness on the communication path. This metric does not consider unchangeable trust among members of a certain group in a network since each node can be individually compromised. Our method therefore permits unobservable communications as in military establishments and also reduces the overhead associated with determining trust at the intermediate nodes.

Chapter 4

New Metrics of Trust

4 Introduction

This chapter starts by giving an overview of a metric of trust. It explains how traffic patterns are collected by end users on a communication path. Afterwards, the chapter describes how specific anomalies are detected based on traffic patterns collected. Later, a probability model for each anomaly is developed and finally functions to update trust of a good path are defined.

4.1 Overview

In MANET, multi-hop technique is employed in order to transport packets from a source node to a destination node as mentioned in chapter two. Nodes composing a route or a path have to, or at least are supposed to cooperate to ensure a proper functionality of a route. Ideally, in favour of all nodes and the performance of the network as a whole, if a node misbehaves it should be identified and isolated so that it should have less chance to misbehave again and therefore inconvenience other nodes or put them at security risks.

However, it might be impossible to completely and consistently link activities of a node and its identity in a network where privacy is a concern and anonymity is implemented. An attacker who wants to identify a particular communication can then be forced to perform some actions that can allow him to trace the communication of the peers. In those circumstances where users of the network are concerned about privacy they should be able to look into their communication to verify if their expected privacy is not tampered with.

In this thesis, we present a method that helps end users to monitor the behavior of traffic patterns of their communication by using traffic analysis tools, and to let them judge whether the communication path is trustworthy or not [76].

On the given ongoing communication, end users may have details of how normal traffic flows from one intermediate node to another, by attaching appropriate additional information to original traffic or in a manner of setting up separate packets. However, in our case we suppose that there are anonymity techniques that can wipe details of how traffic flow through intermediate nodes so that activities at intermediate nodes cannot be linked with their identity. Sender and receiver are then only able to collect patterns of traffic at both ends of the communication.

With traffic patterns collected by sender and receiver, we can find out alteration of status of the communication. Depending on how the status of the communication varies, users may know to a certain extent how a third party or an attacker modifies or monitors the communication. Based on observed behavior of traffic patterns, end users will make decisions on whether they should continue to use the same path or a different one. Following sections present the methods used to get to the decision of whether it is safe to use the same path or it is necessary to find an alternative that might be considered to be trustworthy. Key anomalies are detected using traffic patterns collected. Trust is computed based on probabilities for anomalies to happen due to benign link faults or security attacks.

4.2 Anomaly Detection

Collected by a sender and a receiver, when combined, Traffic patterns serve to detect anomalies in a communication. Quantity of packets and timing of packets have been identified in this work as measured elements that contribute to the change of traffic behavior. The change of those elements causes anomalies.

Anomaly in network traffic is defined as any behavior of traffic different from what is expected or not satisfactory to the users of a particular communication path. Therefore we assume that the source and destination have what is an acceptable or “satisfactory” communication through the paths. For example, in source routing, the receiver expects to receive packets in the same order as they were sent. Another example is that, in a network where users insist on anonymous communication, a user prevents non-peer users from knowing with whom he/she is communicating at a particular time.

To see how the behavior of network traffic changes, sender and receiver share the information collected on network traffic, put them together, and analyze them. We focus on both regular and frequent patterns because they can be meaningful.

We reason that an attacker might introduce regular patterns in traffic with the aim of mapping out a particular communication to link a sender with a receiver. We argue that when anomalies occur on a path frequently, there might be a high chance for those anomalies to occur due to misbehavior of certain intermediate nodes on the path.

After a sender and a receiver start to exchange data packets, they build tables to keep traffic patterns. There is one table built by the sender and another one built by the receiver. The two tables have the same structure. Each table is composed of two fields: Packet identification number and time of action. Each time a packet is sent, the sender records the packet ID and the time. Each time a packet is received a receiver records the packet ID and the time.

Every five (or t) seconds, the receiver sends the sender a table. Upon receipt of the table from the receiver, the sender merges it with his own table into an anomaly detection table. The anomaly detection table contains packet identification, sending timestamp and receiving timestamp for each packet. Obviously, the sender gets the table refreshed every 5 seconds. Using this information the sender can calculate the various values that will be mentioned in the following subsections and keep them in respective variables.

We are aware that exchanging of tables containing traffic patterns between sender and receiver will cause traffic increases and therefore consume more bandwidth. However we did not intend to address the issue of bandwidth consumption in this thesis. However we believe that security must come with a cost. The investigation on whether that cost is worth it or not remains an open problem. We now discuss the traffic pattern parameters that are recorded by the source and destination for use in computing trust and updating trust.

4.2.1 Trip Time Variation (ΔT_t)

Trip time T_t of each packet is the time a packet spends on the way, starting when it is transmitted, ending when it is received. That time is calculated using the sender's time stamp when a packet was sent and recipient's time stamps when a packet was received.

$$T_t = T_r - T_s \qquad (1)$$

where T_s is the time a packet is sent and the time it is received is T_r. In order to measure how long it would normally take a packet to travel from source to destination, reference time variable (T_R) calculated in equation (2) is used. The difference of the time the route request message is received and the time the route request is sent, gives the duration of the route request message on the way from sender to receiver, including all transmission and propagation delays. The difference for the route reply message, obtained in the same way, gives the duration of the route reply message on the way from receiver to sender.

The estimated reference time (T_R) can be given by the sum of the duration of the time of the route request message and the duration of the route reply message divided by two.

$$T_R = \frac{(RQr - RQs) + (R\Pr - RPs)}{2} \qquad (2)$$

where *RQr= Time the route request message is received*

RQs= Time route request message is sent

RPr= Time route reply message is received

RPs= Time route reply message is sent

Having the value calculated in equation (2), we can calculate the variation of trip time of each packet.

The variation of trip time is measured with respect to the average of these values as experienced in the network and the reference time (T_R) is calculated during route discovery process by the following expression.

$$\Delta T_t = T_R - T_t \qquad (3)$$

The trip time of one packet alone is not meaningful but observing the variances and size of trip times can give useful information for detection of abnormities caused by traffic congestion or attack. Observing trip time variations over a period of time will allow the computation of probability of a packet to be delayed. Comparing trip time variation of many packets helps noticing and examining regular delays that are most likely to be caused by attacks.

4.2.2 Change of packets frequency; ΔPf

The sender compares both the frequency at which packets were sent and the frequency at which packets were received, measured in packets per second. By comparing the two frequencies, delays of packets can be noticed. Delays generally are useful to detect when dealing with Quality of Service of a route because delays are most likely to be caused by congestion at intermediates nodes.

4.2.3 Lost packets; d

By looking at the anomaly detection table, it is easy to see packets that have been sent but not received. When there is a packet identification entry with sent time, but without received time, it shows that a packet was sent but not received. That packet could have been dropped after a time out on the queue, or it could have been dropped intentionally by an attacker.

4.2.4 Inserted packets; i

To identify inserted packets, there are more than one options. For one, comparing the actual contents of sent and received packets as an attacker is likely going to use a 'probe packets' that could be different from the actual data packets. But to differentiate probe packets from normal packets, we may have to set up a mechanism to build up another pattern. For simplicity, in our case, after the sender merges its table of traffic patterns with that one from the receiver, packets with received time entries and without corresponding sent time entries, are considered to have been forged by intermediate nodes along the route.

4.2.5 Multiplied packets; t

In the anomaly detection table, if for a packet with one sent time entry and more than one entry for received time, that means packets were copied and resent along the route by intermediate nodes. If for a packet there are more than one entry for sent time, that means the packets were copied and resent along the route by intermediate nodes. For example, after receiving a packet with an identification of AAA, an intermediate node makes more than two copies of the packet, kills the original one, and forwards the copies.

4.2.6 Disordered packets; r

As explained in Chapter one, this work is based on source routing protocols where all packets in one communication are expected to take the same route from sender to receiver. If the route breaks, another one is used and the remaining packets continue to flow trough that new route. In normal circumstances, all packets are expected to arrive at the destination in the same order they were sent. If a packet fails along the route, the sender is notified by the passive acknowledgment so that failing packet is resent. In the anomaly detection table, the number of packets identified should increase with time otherwise the order of packets is not normal.

We define anomaly as a function of the following variables:

$$a = f(\Delta Tt, \Delta Pf, lp, ip, mp, dp) \qquad (4)$$

Where ΔT_t is trip time variation of packets, ΔP_f is change of packet frequency, *lp* is lost packets, *ip* is inserted packets, *mp* is multiplied packets and *dp* is disordered packets.

4.3 Trust Modeling and Update Using Communication Anomalies

A trust metric is developed from a statistical analysis of network traffic patterns. A sender initiates a route discovery process using broadcasting technique to send a route request message, and the same sender records the time when the route request message departs. The message is forwarded to all one-hop neighbours. When a node receives the message and it is not the intended recipient, it forwards the message to its one-hop neighbours. The process goes on until the route request message reaches the intended recipient. Upon receipt of the route request message, the intended recipient records the time that message is received. The recipient makes up and sends back in the same manner a route reply message with the attachment of the received time of the request message and the sent time of the reply message. This process establishes a path between the sender and the receiver. For each route discovered, the initiator of the route discovery process has an estimation of how long a massage takes to travel from sender to receiver. The initiator of the route makes the average of the time taken by the route request message and the time spent by the route reply message as calculated in equation (2). That average time is used as a reference of how long a packet would take to go through the same path.

After the path is established, for subsequent communications along the path, both the sender and the receiver build and keep tables as mentioned in Section 4.2. Furthermore, the sender builds and refreshes the Anomaly detection table regularly. The purpose of the anomaly detection table is for building the patterns for individual data packets. To do so, a few variables for anomaly detection have been introduced for equation (1); sent time of a packet is (T_s) and received time is (T_r). In addition to those two variables we define packet identification as P_{id} . We define traffic pattern (F_p) as a function of timing of each packet and its corresponding identification.

$$F_p = f(T_s, T_r) + P_{id} \qquad \textbf{(5)}$$

The values of those variables are used to estimate the time each packet spends on the path.

An anomalous behavior of a communication path is likely to be caused by a malicious security attack or by a link failure.

Behaviours or patterns of traffic are strongly connected with probability for an attack to happen. As communication goes on, the behavior of traffic can be analyzed. Consequently, the probability of undertaken security attack can be computed. As a result, trust initialized at first place will be updated based on the distribution and the variation of the probabilities. Usually if a behavior observed is beyond the normal traffic behavior, it is reasonable to suppose an attack happened on the path. If an anomalous traffic behavior is observed regularly, the probability an attack occurred are higher.

We assign an initial value to the variable of trust related to a path. Updating of the trust is based on how the sequence of the distributions of the probability has been shaped. In other words, the trust value is to be updated over time based on observed behaviour. As the probability of anomalies increases, trust decreases.

When the trust of the path reaches a given threshold, we consider the path untrustworthy. Furthermore, in this case the path will not be used anymore and the initiator of the communication turns to use an alternative path if he/she has one from its alternative pool. Otherwise the initiator enters its routing discovery process for an alternative to use.

We introduce a new model of computing and updating trust with respect to selected certain specific cases where given parameters of the environment can be predictable. For example use of mobile devices in a conference room, in an office or other place where we can predict movement and obstacle between the devices. Therefore we can calculate the probability of link failure, transit time variation and delays given the parameters. We design various events that result in selected anomalies. We associate different events with different probabilities. The stable probabilities are used as measures of trust. Unusual variations in the probabilities away from their averages are considered as abnormalities which should lead to distrust of the path.

4.3.1 Trust Computation Using the Probability of Transit Time Variation

With the aid of equation (1) and (2), we define the probability that a packet is not delayed in the path (network) by an external influence with equation (6) as follows:

$$p_{tt} = \frac{T_t}{T_R + b\sigma} \qquad (6)$$

where, p_{tt} is the trip time probability of the packet being received within the expected time that it should arrive (under normal circumstances); T_t is trip time of each packet calculated in equation (1); T_R is reference time or average trip time calculated in equation (2); b is a constant which we assume lies in the range $2 \le b \le 5$ and σ is standard deviation in trip time of the packet. The choice of b is empirical and may be evaluated by experimentation.

In a nutshell, we have assumed that most packets arrive at their destinations within at most the sum of the average time and not more than five times the standard deviation. If the trip time of a packet is bigger than the denominator of equation (6) we assume that an unusual event has occurred delaying the packet from arriving on time and hence the path should be less trusted and changed.

4.3.1.1 Trust Update

Trust update η_u is defined as a function of two probabilities as follows:

$$\eta_u = f(p_1, \Delta p_1) \qquad (7)$$

where p_1 is the initial trust of the entity at time t_1 and Δp_1 is the variation of the trust over the time between t_1 and t_2. Therefore, based on the equation (6), we define the trust update probability to be:

$$\Delta p_{tt} = \frac{\Delta T_t}{T_R + b\sigma} \qquad (8)$$

Therefore our trust update equation becomes:

$$\eta_u = p_{tt} + \Delta p_{tt} = \frac{T_t}{T_R + b\sigma} + \frac{\Delta T_t}{T_R + b\sigma} = \frac{T_t + \Delta T_t}{T_R + b\sigma} \qquad (9)$$

Equation (9) is very significant as it provides the point at which we should start to distrust a path. Therefore, for the path to remain trusted, the trust update probability must lie within the range:

$$0 \le \Delta p_{tt} \le \frac{b\sigma}{T_R + b\sigma}$$

Equation (9) includes four variables that influence the trust value of a path (in terms of the trip time of a packet). These are the trip time, the variation in trip time, the average trip time and the standard deviation of the trip time. These are measurable quantities (by experiment). We can therefore define the trust changes with time (the rate of change of trust) as:

$$\eta(t) = \eta t + \frac{d\eta}{dt} t = (\eta + \Delta\eta)t \qquad (10)$$

where $\Delta\eta = \Delta p_{tt} = \frac{\Delta T_t}{T_R + b\sigma}$.

4.3.2 Trust Computation Using the Probability of Link or Path Failure

Probability of link failure in normal circumstances can be defined as:

$$p_l = \frac{total\ link\ \ failure\ time}{Total\ observation\ time} \qquad (11)$$

$p_{l(k)}$ is link failure at time instant k and $p_{l(k+1)}$ at instance (k+1) respectively. Consider the wireless link between two network elements (two nodes for example). If the node has L links from it to L nodes, and if N of the L links fail or are down for a recorded length of time, we use the recorded link failure time as measure of probability of trust. Link failure probabilities under normal circumstances are compared with link failure probabilities under abnormal circumstances. For the probabilities, we use a set of practical data provided on link failures by a cellular communication company. The data set covers 12 months, exactly 365 days. In Table 1, the time link k failed over the time is given as $T_l(k)$. The total time all the links on BSC(j) failed is given by $T_{total} = \sum_{k=1}^{24} T_l(k)$, put in the last row of Table 1.

$T_l(k)$	BSC1	BSC2	BSC3	BSC4	BSC5	BSC6	BSC7	BSC8	BSC9
$T_l(1)$	0	0	0	0	4260	13440	0	0	2580
$T_l(2)$	0	1020	0	0	0	0	0	0	2580
$T_l(3)$	0	1020	0	25380	0	0	0	0	16320
$T_l(4)$	0	2880	0	162660	0	0	4800	0	21120
$T_l(5)$	0	0	0	163860	0	0	0	0	21120
$T_l(6)$	0	0	0	0	0	0	600	0	1140
$T_l(7)$	0	0	0	0	0	0	0	0	0

$T_l(8)$	0	0	0	5820	0	0	0	12120	0
$T_l(9)$	0	0	0	0	28860	0	0	0	0
$T_l(10)$	0	2820	0	0	28860	0	0	0	1140
$T_l(11)$	0	0	13740	0	0	0	32040	0	0
$T_l(12)$	0	0	0	2400	0	0	-	0	0
$T_l(13)$	0	1020	0	21300	0	0	-	0	0
$T_l(14)$	38100	23520	1980	0	16500	0	-	0	0
$T_l(15)$	0	600	8340	5400	0	0	-	17880	0
$T_l(16)$	0	600	2640	398280	23280	0	-	0	-
$T_l(17)$	0	4560	0	0	0	14160	-	0	-
$T_l(18)$	0	0	0	0	0	87600	-	0	-
$T_l(19)$	0	-	-	0	0	13440	-	0	-
$T_l(20)$	0	-	-	0	98580	1860	-	0	-
$T_l(21)$	0	-	-	0	98580	1860	-	0	-
$T_l(22)$	0	-	-	0	0	1860	-	0	-
$T_l(23)$	0	-	-	0	-	0	-	0	-
$T_l(24)$	0	-	-	0	-	-	-	0	-
Total Failure (sec/yr)	**38100**	**37980**	**26700**	**785100**	**298920**	**134220**	**37440**	**30000**	**82560**

Table 2: Link failure time over 365 days in seconds

In Table 2, the probability (and hence trust) that a link failed or would fail over the given time is given in the columns by equation (11).

$T_l(k)$	BSC1	BSC2	BSC3	BSC4	BSC5	BSC6	BSC7	BSC8	BSC9
$P_l(1)$	0	0	0	0	0.0001	0.0004	0	0	0.0000
$P_l(2)$	0	0.0000	0	0	0	0	0	0	0.0000
$P_l(3)$	0	0.0000	0	0.0008	0	0	0	0	0.0005

$P_l(4)$	0	0.0000	0	0.0051	0	0	0.0001	0	0.0006
$P_l(5)$	0	0	0	0.0051	0	0	0	0	0.0006
$P_l(6)$	0	0	0	0	0	0	0.0000	0	0.0000
$P_l(7)$	0	0	0	0	0	0	0	0	0
$P_l(8)$	0	0	0	0.0001	0	0	0	0.0003	0
$P_l(9)$	0	0	0	0	0.0009	0	0	0	0
$P_l(10)$	0	0.0000	0	0	0.0009	0	0	0	0.0000
$P_l(11)$	0	0	0.0004	0	0	0	0.0010	0	0
$P_l(12)$	0	0	0	0.0000	0	0	-	0	0
$P_l(13)$	0	0.0000	0	0.0004	0	0	-	0	0
$P_l(14)$	0.0012	0.0007	0.0000	0	0.0005	0	-	0	0
$P_l(15)$	0	0.0000	0.0000	0.0000	0	0	-	0.0005	0
$P_l(16)$	0	0.0000	0.0000	0.0126	0.0007	0	-	0	-
$P_l(17)$	0	0.0001	0	0	0	0.0004	-	0	-
$P_l(18)$	0	0	0	0	0	0.0027	-	0	-
$P_l(19)$	0	-	-	0	0	0.0004	-	0	-
$P_l(20)$	0	-	-	0	0.0031	0.0000	-	0	-
$P_l(21)$	0	-	-	0	0.0031	0.0000	-	0	-
$P_l(22)$	0	-	-	0	0	0.0000	-	0	-
$P_l(23)$	0	-	-	0	-	0	-	0	-
$P_l(24)$	0	-	-	0	-	-	-	0	-
Total Probability Of Link Failure	**0.0012**	**0.0008**	**0.0004**	**0.0341**	**0.0093**	**0.0039**	**0.0011**	**0.0008**	**0.0017**

Table 3: Probabilities of Link Failures

The probabilities in Table 2 can be used to estimate the trust to a link on these BSCs. We consider the trust as:

$$\eta = \left(1 - p_l\left(k\right)\right) \qquad \left(12\right)$$

This means that the trust values are based on the probability for the link not to fail. If the link fails often, it means the trust assigned to the link will decrease as for such links the probabilities of failures would be high.

The trust to a path consists of the aggregate of the trust assigned to all links forming the path from the source to the destination. Based on our approach for estimating trust, it is unwise to just add up the trusts for all the links in the path since they could add to more than unity. Instead, we use a different method to fuse the data. One approach is to weight the trust for links in the path provided we have a means of weighting them. We can give weights to each of the links if we have enough historical data and/or recommendations from other sources on the trusts to be assigned to the links. In our case, such recommendations are unavailable. Therefore at this stage of approximation we assume that the trust for a path is given by the expression:

$$\eta_p = \coprod_{k=1}^{L} \eta_k \qquad \left(13\right)$$

Another approach to estimate the trust of a path is to use the average trust of all the links.

$$\eta_p = \frac{1}{L}\sum_{k=1}^{L} g_k \eta_l\left(k\right) \qquad \left(14\right)$$

Where g_k is a weight assigned to trust $\eta_l\left(k\right)$. These estimates assume that the trust of the intermediate links in the paths can be estimated and that intermediate-link trusts can be reported to the end points by separated messages or by being attached to normal packets. The weights in (14) are assigned based on historical evidence of how often a link is expected to fail within for example one year. We are not interested yet in how intermediate trust are measured or reported. For the sake of completeness, agent-based trust computing and reporting can be used at the intermediate nodes to monitor the states of the links and to report on when they are up or down.

We regard the trust for the intermediate nodes together as the estimate of the trust for the paths. Such aggregate trust for nodes can be obtained using the methods as described in [62,63].

4.3.3 Trust Computation with Intermediate Node Congestion Probability & End-Node Delay

Under normal circumstances, a node receives p packets per second, the so-called packet arrival rate, which is approximated to be a Poisson process [70]. When congestion occurs at intermediate nodes, packet arrival rate at the destination decreases.

The probability of measuring congestion is therefore given as:

$$p_c = \frac{\lambda_c}{\lambda_n} \qquad (15)$$

where λ_n is packet arrival rate at receiver under normal conditions and λ_c is arrival rate under congestion conditions at the same receiver with the assumption that $\lambda_c \leq \lambda_n$ and $\lambda_n = \lambda_m + y\sigma$. Here λ_m is the mean arrival rate, y is a constant set to not more than 5 and σ is the standard deviation of the packet arrival rate at the destination.

Trust can therefore be computed for each path using a computation function similar to equation (12). In this case trust is equal to the probability that a path will not be congested. A path with congested links is less reliable and hence less trusted. Therefore,

$$\eta = (1 - p_c(k)) \qquad (16)$$

Trust can therefore be updated for each path at time (*k+1*) using the trust update function:

$$T_u = f\left(p_{c(k)}, p_{c(k+1)}\right) \quad (17)$$

Therefore, based on the equation (15), we define the trust update probability to be:

$$\Delta p_c = \frac{\Delta\lambda_c}{\lambda_m + y\sigma} \qquad (18)$$

Therefore our trust update equation becomes:

$$\eta_u = p_c + \Delta p_c = \frac{\lambda_c}{\lambda_m + y\sigma} + \frac{\Delta\lambda_c}{\lambda_m + y\sigma} = \frac{\lambda_c + \Delta\lambda_c}{\lambda_m + y\sigma} \qquad (19)$$

Equation (19) is very significant as it provides the point at which we should start to distrust a path. Therefore, for the path to remain trusted, the trust update probability must lie within the range:

$$0 \le \Delta\lambda_c \le \frac{y\sigma}{\lambda_m + y\sigma}$$

Equation (19) includes four variables that influence the trust value of a path (in terms of the congestion a packet experiences in a path). These are the congestion-time arrival rate, the variation in congestion-time arrival rate, the average path congestion-time rate and the standard deviation of the congestion time arrival rate. These are measurable quantities (by experiment). We can therefore define the trust changes with time (the rate of change of trust) using equation (10) as well.

4.3.4 Trust Computation Using Probability of Delays at Intermediate Nodes

Under normal circumstances there are delays that happen in a path, including transmission delays, processing delays and propagation delays. Consider a situation in which an attacker captures packets and re-processes them and thereby incurring processing delays. In this thesis we consider the result of all these delays together. The equation (2) can be used to calculate the normal average time a packet should take from sender to receiver including various delays under normal circumstances.

When a packet takes longer time than expected to travel from its source to its destination, there is a probability that extra delay is caused by congestion or by an attack on the path. We have in the previous section dealt with trust related to congestion at the intermediate nodes. Suppose the delay is caused by other effects such as reprocessing, inserting of new start time stamp and re-packaging? The probability of delay can be computed as follows:

$$p_d = \frac{\tau_d}{\tau_m + z\sigma} \qquad (20)$$

Where τ_d is the total time of delays, τ_m is the expected average delay time for a path and z as before is a constant not more than 5. Trust is computed by equation (21).

$$\eta = \left(1 - p_d(k)\right) \qquad (21)$$

When delays are encountered on a path for sometime, probability of delays to happen is increased and trust for that path decreases. Following the pattern of analysis in the previous sections, it can be shown that the trust update function is:

$$\eta_u = p_d + \Delta p_d = \frac{\tau_d + \Delta\tau_d}{\tau_m + z\sigma} \qquad (22)$$

and $0 \le \Delta\tau_d \le \dfrac{z\sigma}{\tau_m + z\sigma}$

Regular delays are expected in normal paths and circumstances whereas sporadic delays could be evidence of attacks. If the regular delays are highly structured, they are more meaningful and suspected to be caused by attack. Because an attacker is more likely to introduce and use them as a regular pattern that she can follow to link the sender and the receiver in a communication path supposed to be anonymous.

4.3.5 Trust Computation Using the Probability of Lost, Inserted and Multiplied Packets

Over a period of time, according to section 4.2, total number of sent packets can be compared to total number of received packets. The difference in number of sent packets and received packets can be noticed easily. That difference might be caused by loss of packets, inserted packets or multiplied packets. The probability of packets being lost, inserted and multiplied can be computed by the following equation:

$$p_n = \frac{\pi_{dn}}{\pi_{ns}} \qquad (23)$$

Where π_{dn} is obtained by $\left|numberofsentpackets - numberofreceivedpackets\right|$ and π_{ns} is the number of sent packets. As in earlier analysis, trust can be computed as the probability of the difference in number of sent and received packets not occurring at time *K*.

$$\eta = \left(1 - p_n(k)\right) \qquad (24)$$

Probability of packets to be lost is computed as

$$p_{loss} = \frac{\pi_{nl}}{\pi_{ns}} \qquad (25)$$

Where π_{nl} is the number of lost packets between time *k-1* and time *k*. At time *k* trust is computed as it is done previously based on probability of packets not being lost.

$$\eta = \left(1 - p_{loss}(k)\right) \qquad (26)$$

In general, we can apply a similar consideration to packet variations. Thus at time *k* probability of packets to be inserted π_{isn} and the probability of packet to be multiplied (π_{mul}) are computed using number of inserted packet (π_{nisn}) and number of multiplied packets (π_{nmul}) respectively in the same way it is done in equation (26). Trust is updated based on probability of packets to be inserted and on probability of packets to be multiplied respectively as it is done in sections 4.3.1.1.

4.3.6 Trust Computation Using the Probability of Normal Traffic

Some research has shown that interference in the links results in increasingly bit error rates, mainly due to normal traffic transmission problems [71]. We therefore define the probability of normal traffic transmission as the bit error rate of the path as:

$$p_{\varepsilon} = bit\ error\ rate\ of\ link\ or\ route\ at\ normal\ behaviour\ time \qquad (27)$$

This means that, although bit error rate varies over time, congestion, abnormality and bit error rate vary abnormality over the time a link or a path when subject to attack.

Here again, we use $\varepsilon_{t(k)}$ as bit error rate of a link or a path at time instance k and $\varepsilon_{t(k+1)}$ at instance (k+1) respectively. If the bit error rate of a path increases, trust should decrease. Bit error rates are usually very small and increases in them is a cause for concern. The trust value based on bit error rate is:

$$\eta = \left(1 - p_{\varepsilon}(k)\right) \qquad (28)$$

Trust update functions in this case are also given by similar expressions as in the equations (13) and (14):

All computed trusts are combined in order to calculate overall trust of a path. Many probabilities for anomalies to occur are related. For example, when probability of transit time variation increases, probability of delay also increases. When probability of congestion increases, probability of delay also increases. When probability of regular delay increases, also probability of delay also increases. When probability of the difference in sent and received number of packets increase, also probabilities of inserted, lost or multiplied packets increase. Because of these inter-relationship in increase or decrease of probabilities of anomaly to occur, we combine all trust derived from these probabilities so that we can have the overall trust value. We have used 11 expressions to compute trust. We reason that since at least two trusts are related and increase or decrease together, it will be sensible to sum all trusts in order to get the overall trust. The overall trust will not decrease slowly and take a long time to reach the threshold while anomalies are continuously being detected on the path.

We defined threshold value of overall trust as the average of all possible maximum values of the trust expressions.

4.4 Summary

We have proposed new expressions for new metrics of trust based on QoS parameters. They are based on transit time variation, link failures, dropped packets, packet losses, congestion and delays in the path. We also show through a practical set of network statistics the manner and range of expected link failure rates using a MANET as an example. We have also shown how to update trust based on QoS variables chosen. In chapter 5, these metrics are evaluated.

Chapter 5

Experiments and Results

5 Scenarios in Experiments

We have described in the introductory chapters of this thesis how ad –hoc networks can be used in different situations for different purposes. We believe that it is very hard to design one solution that can suit all situations at the same time. In this chapter we design a solution considering a situation where a mobile ad hoc network is used for a mini conference for sharing files. The solution designed in this chapter can also suit similar situations as a situation where students are in an area at school where there is no network infrastructure and they need to share data or information.

Authors in [56] list quantitative metrics used for evaluation considerations in MANET. These metrics include end-to-end data throughput, delays, out of order packets delivery and efficiency in terms of data and control bits transmitted and data and control bits delivered.

Evaluations of performance of ad hoc networks can be done by simulations methods based on software like GloMoSim[57], OPNET[58] and NS-2 [59]. Simulations are easy to use in terms of logistics; equipment needed and man power. Simulations permit repeatable experiments and many measurements can be done by just altering values of some parameters. However simulations may fail to capture the feasibility of real world operation of wireless ad hoc network [60].

The other method for evaluation of performance of a network is using real operation environment. Testing MANET in real world has limitations of size of network that can be constructed because affordability of mobile devices and available people who can carry the devices around to implement mobility. Despite those limitations there exist test beds for real world implementation for MANET.

In [64] the testbed consists of 5 moving nodes. It is for DSR protocol and it is implemented on FreeBSD operating system. The experiments are done outdoors using laptops and GPS. The testbed is for ABR routing protocol. In the DAWN at BBN a testbed that consisted of up 10 laptops was implemented using special hardware based on Nokia Wireless routers [64]. The testbed is for NLS routing protocol running on DAWN or FreeBSD. The APE simulation in [60] is not bound to any routing protocol. It was implemented indoor using 37 laptops running Linux operating system.

5.1 Environmental Setup

To appreciate the experiments and results reported in this chapter, a description of the equipment involved is given in Table 4. The equipment were sourced and purchased locally.

5.1.1 Hardware and software Components

In our experiments we used 1 PC equipped with a wireless LAN card for main development and 3 handheld devices as targets systems. The choice of the devices used is based on their availability on the market and their affordable prices. They are suitable for the embedded OS we intended to use. They are also popular devices and it is easy to find documentation or learn from experiences of other developers using the same devices.

No	Device	Description
items	Model	
2	iPAQ h5550	400 MHz Intel PXA250 xscale ,RAM 128MB ::SDRAM
1	iPAQ h3870	206 MHz Intel StrongARM SA-1110 RAM 32MB ::
1	PC	Intel (R) Pentium (R) 4 CPU 1.80 GHZ 512 RAM

Table 4: Specifications of Devices Used for Experiments

The iPAQs h5550 have integrated 802.11b radio. The iPAQ h3870 is equipped with Pretec PocketPC 802.11b compactWLAN. Capacity storage of these iPAQs were enhanced[1] MMC cards; two 64 MB and one 1 GB. The PC is equipped with Netgear Wireless PC card and a 32-bit CardBus WG511. The PC runs Linux; the Debian sarge with 2.4 Kernel.

All the iPAQs run Linux FamiliarV0.8.2 with 2.4 kernel. The h5550 iPAQs we used were brand new and they came with windows operating system on them. New bootloader and embedded OS were installed on them. The h3870 iPAQ had been used previously but we had to update the bootloader and install FamliarV0.8.2. The Linux Familiar installed in the iPAQs was downloaded from the handheld webpage[1]. Familiar-Linux was our choice as embedded OS because of its flexibility of installation and the available software updates in packages freely available. In addition to that there are documents on how to install familiar and how to configure the iPAQs available at handheld webpage. The Debian distribution of Linux was also chosen for the reason it is free of charge, is availabile in packages that allows easy software update and customization. Device drivers and updated firmware are available for Linux OS and imbedded OS. All software in Linux platform can be modified according to GNU General public License.

C and C++ languages were used for development because we had to make changes to the base Linux-kernel that is in C. We also had to implement loadable modules for the base Linux kernel. Scratchbox[1] tool was used for cross-compiling to save compilation time of the code on iPAQs with their low capacity. The Scratchbox tool was already designed for PC hosts running Debian and iPAQs for targets running Familiar. It has built-in compilation toolchains for targets with Xscale/ARM based processor so there was no need for extra time and effort to develop and compile new toolchains.

DSR has been selected as the underlying routing protocol for its on-demand and source routing characteristics as we explained in the introductory chapter 2. We implemented DSR as in [72]. The make files have been modified to correspond to the architecture of the device used and the path to correspond to current directory hierarchy.

[1] http://handhelds.org/download/familiar/releases/v0.8.2/

One of the most important aspects of the hardware in our experiments is the clock synchronization as we intend to use local host timing for collecting traffic patterns.

5.2 Device's Clock Synchronisation

As seen in the previous chapter, data needed for the metric of trust is *time* and *packet ID* collected by end users of a communication path. Some of the measurements done by the trust model developed in this thesis are transit time variation and delay. Clock synchronization is necessary for data fusion in order to estimate delays with accuracy.

Device clock Synchronization has been a well known problem in wireless networks and many techniques have been proposed to solve that issue [65, 66, 67, 68]. In this work we have chosen to use Reference-Broadcast Synchronization (RBS) scheme for its proven performance and precision [68]. Another motivation for us to use RBS is that it has already been implemented on devices similar to what we used (iPAQ with StrongARM-based processor) and running the same operating system (Linux-Familiar) and the same kernel version (kernel 2.4) and that saved a lot development time for us. The source code is publicly available[1]. RBS can do local time scale synchronization and global time scale synchronization for external references.

The author have implemented local time scale synchronization only as we considered a small network on a small area like a conference hall, so the author supposes that all nodes will be within the same domain. For time synchronization, a reference packet is broadcasted by a transmitter. Then, each receiver of that broadcast records the receipt time according to its local clock. Finally, the receivers exchange their records. Each receiver computes its phase offset to any other receiver as the average of the phase offsets implied by each pulse received by both nodes that shared records. Timestamping of packets is done while the packet is still at NIC and the LBNL packet capture library is used to access the metadata. We have installed the kernel and have changed the make file for the correct entry of platforms. The kernel patch in [74] was modified according to the wireless card drivers used and directory structures. The code was cross-compiled using arm-gcc toolchain in Scratchbox. In our experimental network a precision of the range of

$6\mu \sec$ was achieved. That is different from network to network depending on the number of users and the devices used. That precision is acceptable since delays measured by our model is up to millisecond precision and the path is up to the maximum of three hops.

5.3 The Logic of Our Experiments

Our metric proposed in chapter 4 lies between DSR and network traffic as it is recapitulated in Figure 6. We see DSR as an agent that dwells in a node, parts of such agents communicate with each other to build routing tables by exchanging DSR routing packets. There is another agent, called traffic generator agent (TGA) in each node to create data packets. Data packets are forwarded according to entries in routing tables created by DSR. As it will be explained in the following sections, we produce various patterns of data traffic by controlling TGA. We set up various attacks onto the network to modify the previous traffic patterns. As mentioned in chapter 4, depending on the probability model based on variation of traffic patterns, trust value for a path is computed. When a path is rated untrustworthy, a feedback is given to DSR. DSR refreshes the routing table and removes the route just rated untrustworthy. If there is no alternative route in the cache of a node then a route discovery process is started by DSR. Our metric has been implemented to the level of rating the path because that was enough to measure the effectiveness of our metric. To delete a route, refresh the routing table and set the routing discovery process if necessary was beyond the scope of our experiments.

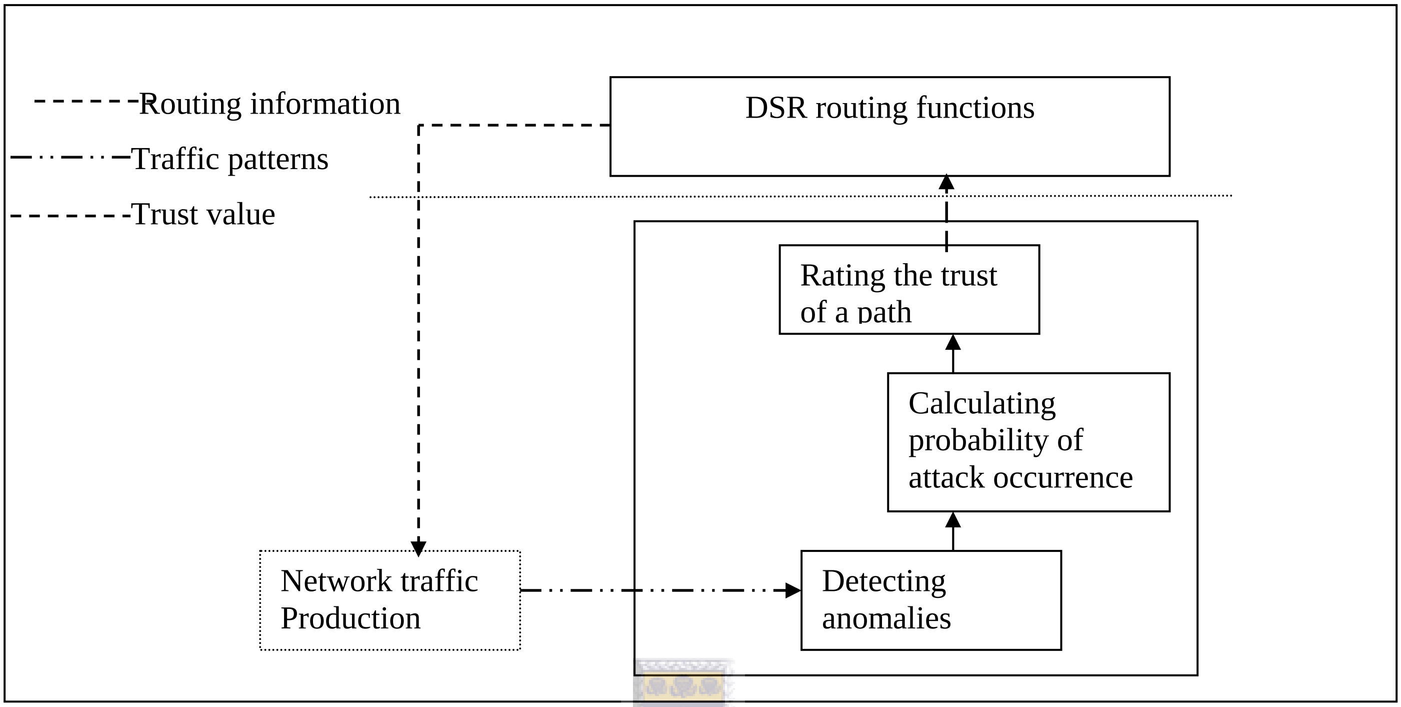
Routing information
Traffic patterns
Trust value
DSR routing functions
Rating the trust of a path
Calculating probability of attack occurrence
Network traffic Production
Detecting anomalies

Figure 6: System diagram Including Trust Update and DSR Integration

In summary, we have taken the following steps:

(1) Building the TGA agents to generate data traffic; (2) setting up various attacks; (3) identifying the changes of traffic patterns after launching attacks; (4) inferring probability or trust values based on the trust model.

We have also implemented an ad-hoc network and observed traffic patterns in different conditions which we created in order to validate our trust model. We have taken various actions to influence traffic patterns.

5.3.1 Network description

All experiments were done indoors in our lab on a small area that allowed a maximum of 6m between two devices. The Debian box was using the wireless interface for the private network to connect to iPAQs and the NIC interface to connect to the university LAN. The wireless interface was configured to forward and to receive packets from outside the wireless private network through the NIC interface. iPAQs used the wireless interface of the Debian box as a gateway to access the Internet for most software download. In all the experiments we explain below we used four nodes labeled N1, N2, N3 and N4. For simplicity we always used N1 as source node and N4 as destination node and other two nodes were always used as intermediate nodes. N1 and N4 are the two h5550 iPAQs, N2 is the h3870 iPAQ and N3 is the PC with a wireless LAN card. Unless stated otherwise, in all the experiments movement was created by a random walk by individuals carrying the iPAQs and the movement is estimated at a speed of about 2m/s. To be able to implement a multihop routing on a small area of our lab, traffic from chosen nodes was blocked using the feature of packet filtering rule set of iptables tool that resides in the netfilter framework [73]. Commands were put in the initialization script so that the configuration is not lost at reboot of the machines. That forced traffic generated at N1 to pass through N2 then N3 before it reaches N4. Otherwise there would be no opportunity of N1 and N4 to use intermediate nodes since they would be in the same transmission range.

5.3.2 Traffic generation and traffic pattern collection

Distributed Internet Traffic Generator (D-ITG) was implemented to generate traffic [75]. D-ITG is composed of set of tools to generate different kinds of traffic at different layers of networking protocol stack and to analyze results. We have chosen to use the D-ITG for its property to generate distributed traffic with flexible inter-departure times and packet sizes. We implemented D-ITG without interest on its delay meter since we defined how to measure delay using time stamping and packet identification as described in the previous chapter. In our implementation, the daemon at N1 captures time and packet ID at the NIC and saves the information locally in binary log file *sent_pat* as packets are sent out. At N4 the daemon does the same as packets are received but *receive_pat* file is used. Every 10 seconds, the daemon at N4 makes a backup of the *receive_pat* file, and that file is sent to N4. The 10 seconds are calculated starting when the first packet is received. At N4 a patch traf_pat has many sections that merge *sent_pat* and *receive_pat*, put the results in a new file, and sorts the new file by packet ID. A function calls Trustd daemon that detects anomalies as described in the previous chapter, computes probabilities according to formulas given in the previous chapter as well, rates the path and returns values. The backup of log file at N4 is kept till another data packet arrives from N1. All files *receive_pat*, *sent_pat* and the *sent_receive_pat* are refreshed with recent entries according to local system time.

Each session of experiment lasted for 10 seconds and it was repeated 20 times. All experiments can be collapsed in two general scenarios. Different traffic conditions were experimented using different traffic load as low traffic load medium traffic load and maximum traffic load that are 1 Mbps, 4 Mbps and 5 Mbps respectively. Both protocols; TCP and UDP have been used.

5.3.2.1 Standard traffic

Standard traffic is referred to as traffic that is not influenced by users' actions in packets forwarding. That includes normal traffic where packets travel from N1 to N4 at uniformly distributed packet rate and uniformly distributed packet payload size.

Standard traffic also includes traffic where there is link failure. The link failure is implemented between N2 and N3. In the first case N2 is moved away from N3 at walking speed till the link breaks. N2 is again moved back closer to N3 at the same speed pattern. Since the area of implementation is small, we borrowed the idea of shortening the coverage transmission of a node by wrapping the antenna on the wireless LAN card with an aluminium foil paper from [72]. On N2 and N3 the antenna on the wireless LAN cards were tightly wrapped with foil paper. The link between N2 and N3 was breaking while these two nodes were only 5 meters apart. In the second case, link failure was implemented by restarting N3 while traffic is traveling from N1 to N4.

Congestion also has been implemented as part of standard traffic. Packets sending rate and receiving rate were different at N1 and N2. N1 was sending packets at a rate that is slightly higher than the receiving rate of N2 in order to make packets to be queued at N2, and start to be drooped when the queue is more than its capacity.

Delay deviation has been measured under different traffic loads with different packet sizes. Three different conditions are considered as normal traffic, link failure and congestion as they have been explained in this section.

Traffic load	Protocol	Packet size in Byte	Delay standard deviation
Low load	TCP	512	0.9332098
		1024	1.1678909
	UDP	512	0.9000230
		1024	1.0310357
Medium load	TCP	512	1.0900423
		1024	1.7963210
	UDP	512	1.0400001
		1024	1.5499997
Heavy load	TCP	512	2.1919995
		1024	2.9499200
	UDP	512	1.8970983
		1024	1.9090042

Table 5: Standard delay in case of link failure

Traffic load	Protocol	Packet size in Byte	Delay standard deviation
Low load	TCP	512	0.9456909
		1024	1.9576315
	UDP	512	0.9213456
		1024	1.2093410
Medium load	TCP	512	1.7928643
		1024	1.9098590
	UDP	512	1.5037897
		1024	1.7053200
Heavy load	TCP	512	2.1911833
		1024	2.9985403
	UDP	512	1.7865213
		1024	1.7932109

Table 6: Standard delay in case of congestion

Table 5 and 6 represents the delay standard deviation. The first row depicts behavior of traffic in TCP whereas the second row depicts behavior of traffic in UDP. It is observed that delay is very low and it increases slightly when the packet size increases. As expected, it is noticed that under the low traffic load the delay is minimum and delay increase with the traffic load. Delays in case of congestion and link failure increase dramatically in high traffic load and they are much higher than delays in normal traffic. It was interesting to notice that in TCP protocol, link failure and congestion are closely related while in UDP protocol there is a clear difference in behavior of network under congestion or link failure. It is our conclusion that that close behavior of two cases in TCP protocol is due to the fact that packets loss are considered to be caused by congestion and the TCP reduces the transmission rate whenever a loss is encountered.

5.3.2.2 Change in packet forwarding

Packets are manipulated at the IP layer of the TCP/IP protocol stack using the Netfilter framework that is implemented in Linux kernel version 2.4.x and 2.6.x for IPv4 [73]. DSR is implemented at the IP layer. Hooks composing netfilter allow kernel modules to register callback functions with the network stack Layers of protocol stack are connected to each other via hooks and packets travel from one protocol to the other through those hooks. When a packet arrives at the hook, the Netfilter is called to check if there is any request of application about that packet. If there is no request for that packet, the packet is forwarded to the next hop. Otherwise a function is called. At that time, one or many packets can be manipulated in many ways; for example it can be deleted, delayed, dropped, copied, modified, replaced. The LOCAL_OUT hook is called at the node that created packets. Upon receipt of a packet, a PRE_ROUTE hook is called if the checksum was verified. If the host receiving that packet is the destination, then the packet must be passed to upper layers of TCP/IP stack. Before passing the packet to upper layers, LOCAL_IN hook is called. If the receiver of the packet is not the destination, then the packet must be forwarded to the next hop. Before the packet is passed to the next hop, the FORWARD hook is called. The packet goes through the FORWARD hook, and then the POST_ROUTE hook is called before the packet is sent further to the next hop.

All changes implemented in packet forwarding are on data packets only, and they do not interfere with routing packets.

Regular delays: On N3 before a packet is forwarded, the POST_ROUTE hook is called with a delay. The function that delays the hook is called after every period of time. The objective is to introduce a regular pattern in the traffic at intermediate node and observe the result at the receiver. The aim is to check if the model defined in the previous chapter is capable of detecting the regular pattern. Figure 7 represents the capability of the model to compute probability of regular delay. Looking at the standard delay deviation, it is clear that the delay can be picked out. Opposite to our expectation, the highest regular delay probability is below 0.7. Looking at Figure 7, it is noticeable that the probability of regular delay decreases as the delay is decreased.

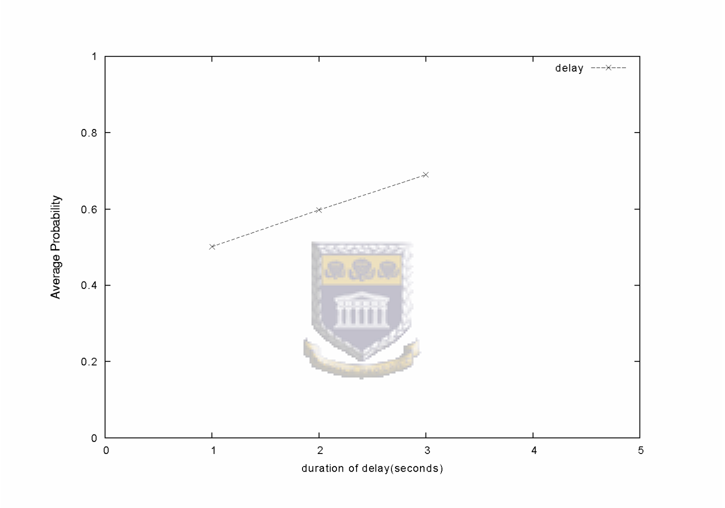

Figure 7: Probability of delays attack in the network

Inserted packets: packets is generated at N3 and the function *ins_pack* inserts the generated packets in the traffic randomly. N4 gets packets from N3 and from N1.

Multiplied packets: at N3 multpacd daemon captures a packet form the traffic and makes a copy of it. The packet is then forwarded normally . The copy of the packet is introduced in the traffic also 1 second after it has been copied. The daemon is called randomly.

5.4 Results and analysis

In this section we present the results of our experiments. We assessed the effectiveness of the metric for detecting the anomalies in the network traffic and for detecting attacks that change the behavior of traffic patterns. We measured the aptitude of our trust metric to react to the change in the network traffic. We also measured the level of failure of the metric and we called them false positives.

Effectiveness in Detecting Regular Packets' Drops: At N3, one packet is dropped after every 500 packets are forwarded to N4. Figure 4 represents the capability of the model to pick up regular loss of packets. The regular loss pattern is picked up with high probability.

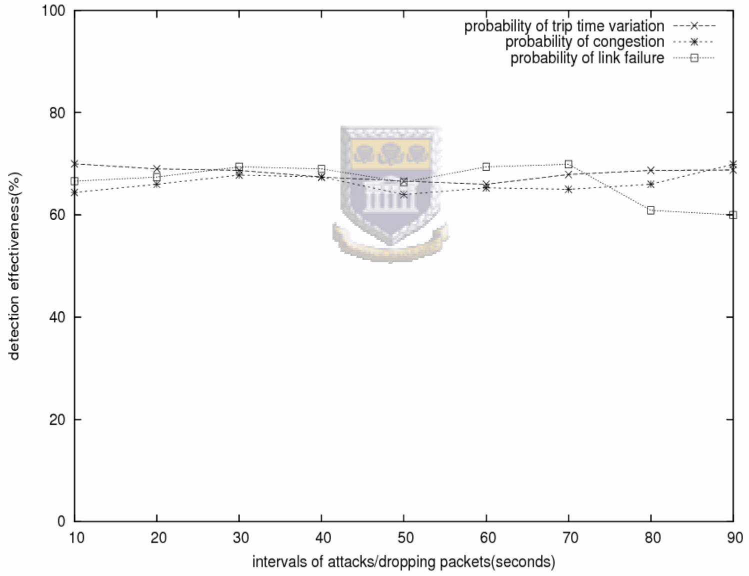

Figure 8: Comparison of Performance of Metrics for dropping packets

Figure 8 represents the comparison of the performance of the metrics on dropping packets with the expectation on the path. The vertical axis describes the effectiveness of the metric that is used to model attacks in the link. The attacks are instigated at regular intervals.

False positives are caused in the system as shown in Figure 9. In instances when there are dropped packets but not really caused by an attack. The system tends to think that at such times attacks have happened and we call them false positives.

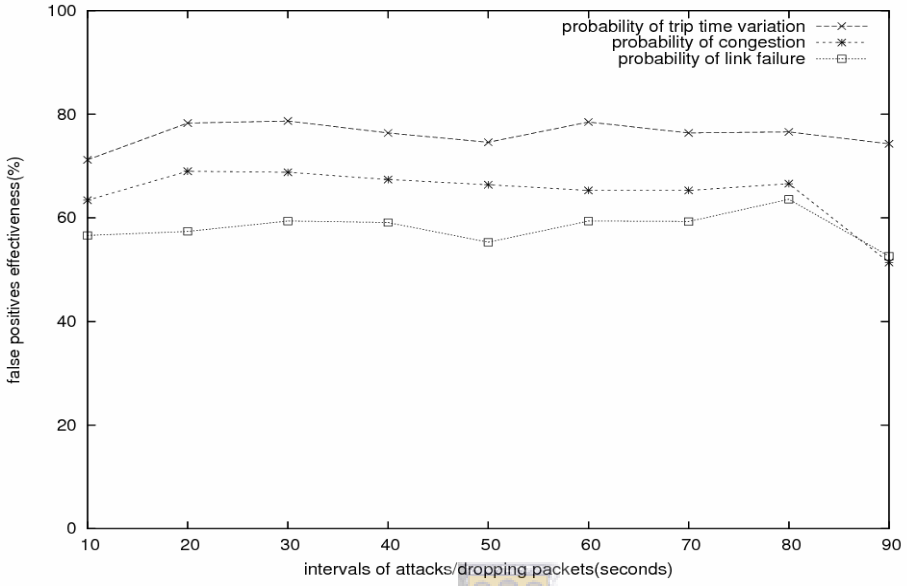

Figure 9: False Positives for Dropped Packets

Effectiveness in Delay Detection: the performance of the metric for detecting delay in the path was demonstrated through experiments and results are shown in Figure 10. The Figure shows delays due to attacks as compared with delays when there are no attacks.

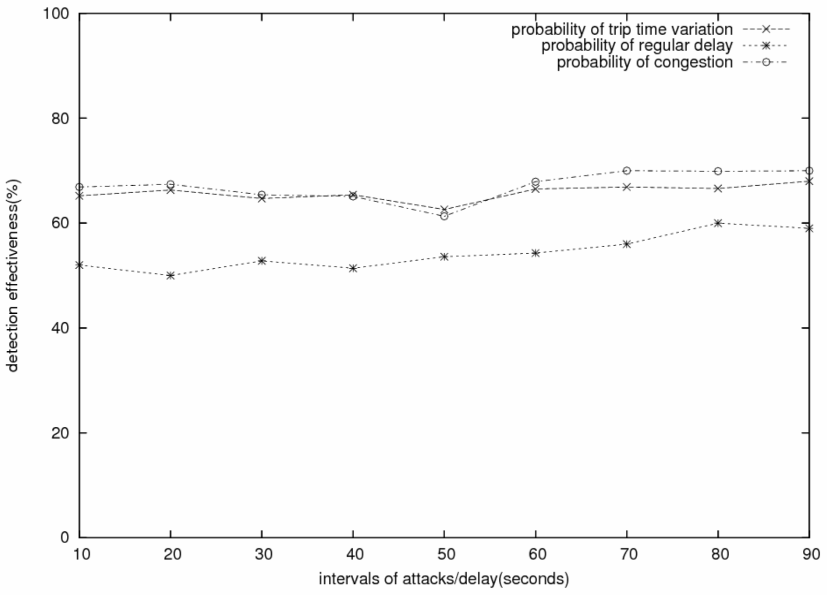

Figure 10: Effectiveness in Detecting Delays Due to Attacks

The graphs show that the metrics are able to detect delay attacks about 65% of the time. The lowest of the three graphs shows the system in this case has not performed better than about 60% of the time in detecting delay attacks.

False positives for delay attacks are insignificant for this metric and lie mostly below 20% of the time. In fact they are on average not more than 15% as shown in Figure 11.

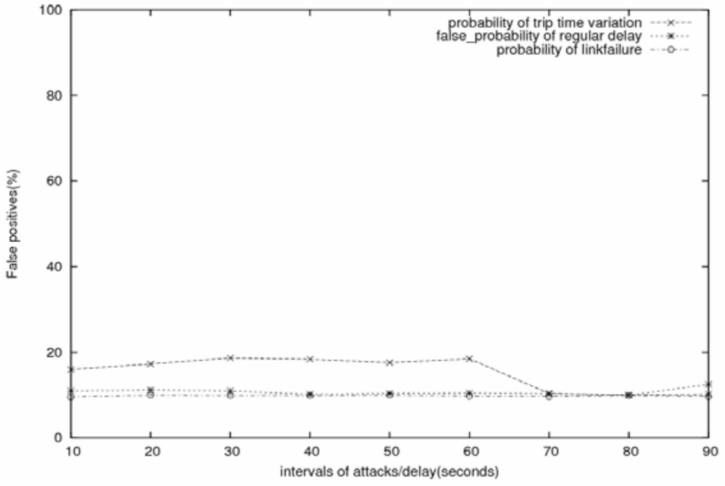

Figure 11: False Positives in Delay Attacks in Paths

Effectiveness in Detecting Multiplied Packets Attack: an attacker can insert multiplied packets into the path. Our metric is also able to detect such instances of attacks as shown in Figure 12. The performance of the metrics in this case is very impressive as it is able to detect multiplied packets for over 90% of the time.

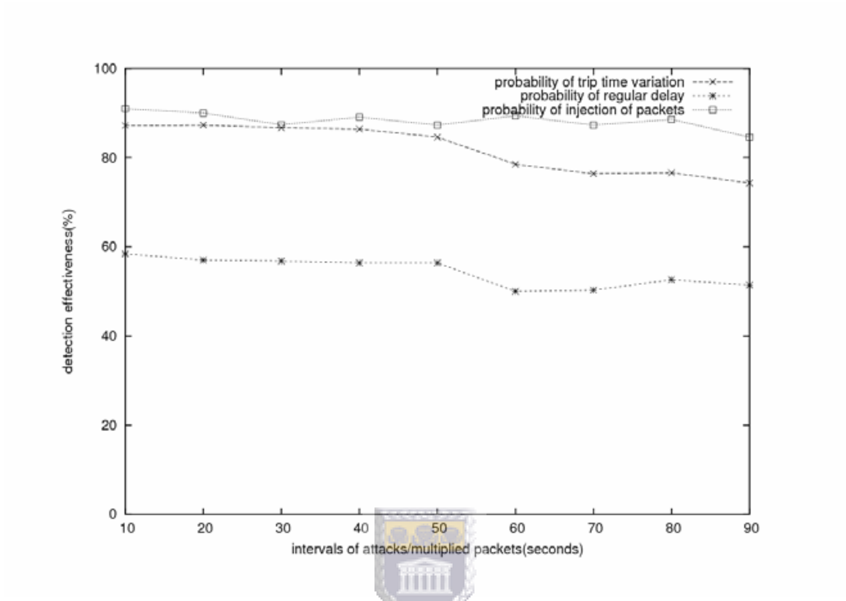

Figure 12: Effectiveness in Detecting Multiplied Packets Attack

Figure 13 shows the false positives for the case of multiplied packets attack.

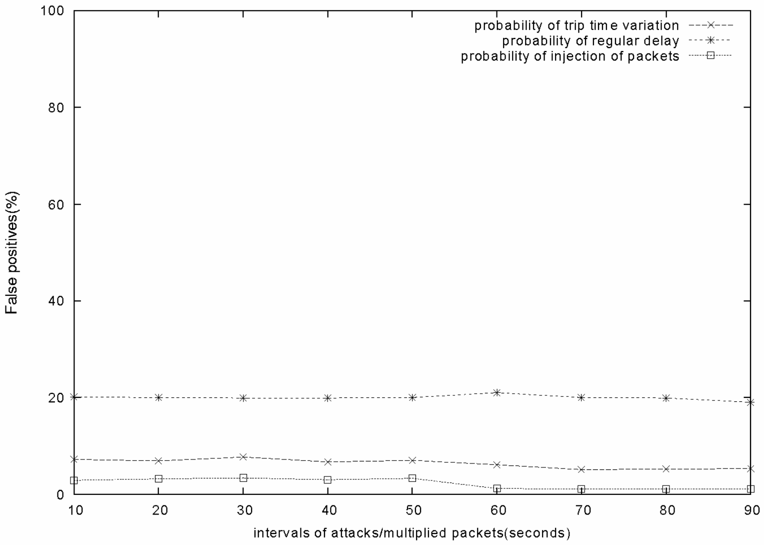

Figure 13: False positives in multiplied packets in a path

5.5 Summary

This chapter is an evaluation of the metrics as to how effective they are as a ratio of normal traffic pattern probabilities. These effectiveness values are represented in the previous figures as well as the false positives. The results show that for most of the metrics defined, we were able to determine when attacks occur in some cases more than 90% of the time and in worst cases around 70%.

Chapter 6

Conclusion and Future Work

The main objective of the thesis was to design a model that detects regular patterns in network traffic of a communication between end users path in a MANET. It has been explained that the openness of MANET exposes users at security risks mainly caused by malicious participants in the network. The absence of central administration makes prevention and defense against attacks a difficult task. The thesis illustrated main routing challenges related to characteristics of MANET. Different attacks have been discussed and the thesis pointed out challenges in preventing security attacks in MANET; especially attacks caused by participants in the network.

Cooperation of participants in the network has been highlighted as key to proper functionality of a multi-hop network such as MANET. Definitions of trust in different fields and contexts have been given. It has been explained that without some form of trust, cooperation of participants cannot be achieved. Trust among users of a network between each other has been the main concern and that concern has been addressed in many ways as indicated in this thesis. Trust models in MANET proposed in the literature compute trust between participants in the network based on the opinion of nodes about each other. Our model computes trust of sender and receiver towards a path they use based on variation of status of traffic patterns. That is in case there is no information available about individual intermediate nodes' behaviour in the path.

Some attacks are possible to detect, especially those that change the status of the network traffic. However, that is not a strait forward issue as the status of the network traffic might also change due to link faults. We have assumed that an attacker to the network traffic introduces a distinguishable pattern that can reveal valuable information to him/her. We thought of regular patterns possible in number of packets in timing of

Appendix I

References

[1] D. Umuhoza, R.C. Staudemeyer, C.W. Omlin, “*A metric of trust in Mobile Ad hoc Networks using Direct Source Routing Algorithms,*” Proceedings of the Southern African Telecommunication Networks and Applications Conference (SATNAC). 2005.

[2] L. Peterson and B. Davie, “*Computer Networks, A System Approach,*” Second Edition,p284-292. 2000.

[3] D. B. Johnson, D. A. Maltz, and J. Broch, “*DSR The Dynamic Source Routing Protocol for Multihop Wireless Ad Hoc Networks*,” In Ad Hoc Networking, edited by Charles E. Perkins, chapter 5, pages 139–172. Addison-Wesley. 2001.

[4] C. Perkins and E. Royer, “*Ad Hoc On-Demand Distance Vector Routing*,” In Proceedings 2nd IEEE Workshop on Mobile Computing Systems and Applications (WMCSA’99). 1999.

[5] A. Pfitzmann, “*Anon Terminology Paper*,**”** available at http://dud.inf.tu-dresden.de/Anon_Terminology.shtml Accessed March 2006

[6] A. S. Tanenaum, “*Computer Networks*,” fourth edition P 69. 2003.

[7] H. Yang, H. Y. Luo, F. Ye, S. W. Lu, and L. Zhang, “*Security in mobile ad hoc networks: Challenges and solutions*,” IEEE Wireless Communications, pp. 38-47. 2004.

[8] L. Zhou and Z. Haas, "*Securing ad hoc networks*," IEEE Network Magazine Special Issue on Network Security, vol. 13, no.6, Nov./Dec. 1999

[9] X. Li, “*Trust Model Based Self-Organized Routing Protocol for Secure Ad Hoc Networks,”* citeseer.ist.psu.edu/628444.html. Accessed March 2006.

[10] C. E. Perkins and E. M. Royer, “*Ad hoc On-Demand Distance Vector Routing*,” Proceedings of the 2nd IEEE Workshop on Mobile Computing Systems and Applications, pp. 90-100, New Orleans, LA, February 1999.

[11] C. Perkins and P. Bhagwat, “*Highly Dynamic Destination-Sequenced Distance-Vector Routing ({DSDV}) for Mobile Computers ,*” {ACM} {SIGCOMM}'94 Conference on Communications Architectures, Protocols and Applications, pp. 234—244. 1994

[12] D. Johnson and D. Maltz, “*Dynamic Source Routing in Ad Hoc Wireless Networks,*” Mobile Computing, T. Imielinski and H. Korth, Ed., Kluwer, 1996.

[13] V.Park and S.Corson. "*Temporally-Ordered Routing Algorithm (TORA) Version 1*", IETF Internet Draft, July 2001

[14] K. Sanzgiri et al, “ *A Secure Routing Protocol for Ad Hoc Networks,”* In Proceedings of 2002 IEEE International Conference on Network Protocols (ICNP), November 2002.

[15] P. Papadimitrats and Z. J. Haas,*” Secure Routing Mobile Ad hoc Networks,”* In Proceedings of the SCS Communication Network and Distributed Systems Modeling and Simulation Conference(CNDS 2002), January 2002.

[16] A. Khalili, J. Katz, and W.A. Arbaugh, “*Toward Secure Key Distribution in Truly Ad-Hoc Networks,*” 2003 Symp. Applications and the Internet Workshops (SAINT 03 Workshops), pp. 342-346. IEEE CS Press, 2003,

[17] B. Awerbuch et al., *“An On-Demand Secure Routing Protocol Resilent to Byzantine Failures,*” Proceedings ACM Workshop Wireless Security, ACM Press, pp. 21-30. 2002.

[18] A. Patwardhan et al, "*Secure Routing and Intrusion Detection in Ad Hoc Networks,*" Third IEEE International Conference on Pervasive Computing and Communications, Kauaii Island, Hawaii, March 8-12, 2005.

[19] S. Yi, P. Naldurg and R. Kravets, “*A Security-Aware Routing Protocol for Wireless Ad Hoc Networks,”* ACM Symposium on Mobile Ad Hoc Networking & Computing (Mobihoc ’01). October, 2001.

[20] K. Sanzgiri et al, *“A Secure Routing Protocol for Ad Hoc Networks,”* In Proceedings of 2002 IEEE International Conference on Network Protocols (ICNP), November 2002

[21] Y. Hu, A. Perrig, and D. Johnson, “*Ariadne: A Secure On-demand Routing Protocol for Ad Hoc Networks,*” *ACM MOBICOM*, 2002.

[22] M. Zapata, and N. Asokan, "*Securing Ad Hoc Routing Protocols,*" ACM WiSe, 2002.

[23] B. Dahill et al., "*A Secure Protocol for Ad Hoc Networks,*" IEEE ICNP, 2002.

[24] Y. Hu, A. Perrig, and D. Johnson, "*Packet Leashes: A Defense against Wormhole Attacks in Wireless Networks,*" IEEE INFOCOM, 2002.

[25] A. Patwardhan et al, "*Secure Routing and Intrusion Detection in Ad Hoc Networks,*" Proceedings of the 3rd International Conference on Pervasive Computing and Communications, Kauai Island, Hawaii, 2005.

[26] Y. Hu, A. Perrig, and D. Johnson, "*Rushing attacks and defense in wireless ad hoc network routing protocols,*" Proc. of 2nd ACM Wireless Security (WiSe'03), pp. 30--40, 2003.

[27] Cambridge International Dictionary of English, Cambridge University Press, 1995. ISBN: 0521 588359.

[28] M. Deutsch, "*The Resolution of Conflict: Constructive and Destructive Process,*" New Haven, CT: Yale University, 1972.

[29] S. Marsh, "*Formalizing Trust as a Computational Concept,*" PhD thesis, University of Stirling, Uk, 1994.

[30] P.Lamsal, "*Understanding trust and security,*" 2001 http://www.cs.Helsinki.FI/u/lamsal/papers/*UnderstandingTrustAndSecurity*.pdf.

[31] D. Gambetta' " *Can we trust trust?,"* In D. Gambetta, editor, *Trust: Making and Breaking Cooperative Relations*, chapter 13, pages 213.237. Department of Sociology, University of Oxford, electronic edition, 2000.

[32] F.L. Mayer, "*A brief comparison of two different environmental guidelines for determining "levels of trust","* In Sixth Annual Computer Security Applications Conference, 1990.

[33] M. K. Reiter and S. G. Stubblebine, "*Resilient authentication using path independence,*" IEEE Transactions on Computers, vol. 47, no. 12, pp. 1351–1362, December 1998.

[34] U. Maurer, "*Modelling a public-key infrastructure,*" in Proceedings 1996 European Symposium on Research in Computer Security(ESORICS' 96), volume 1146 of Lecture Notes in Computer Science, pp. 325–350, 1996.

[35] A. Jsang, “*An algebra for assessing trust in certification chains,”* in Proceedings of the Network and Distributed Systems Security (NDSS’99) Symposium, 1999.

[36] R. Levien and A. Aiken, “*Attack-resistant trust metrics for public key certification,”* in Proceedings of the 7th USENIX Security Symposium, January 1998, pp. 229–242.

[37] P. R. Zimmermann, “*The Official PGP User’s Guide,”* MIT Press, 1995.

[38] M. Blaze, J. Feigenbaum, and J. Lacy, “*Decentralized trust management,”* in Proceedings of the 1996 IEEE Symposium on Security and Privacy, pp. 164–173, May 1996.

[39] A. Abdul-Rahman and S. Hailes, “*A distributed trust model,”* in Proceedings of 1997 New Security Paradigms Workshop, ACM Press, pp. 48–60, 1998.

[40] A. Herzberg, Y. Mass, J. Michaeli, D. Naor, and Y. Ravid, “*Access control meets public key infrastructure or: Assigning roles to strangers,”* in Proceedings of the 2000 IEEE Symposium on Security and Privacy, pp. 2–14. May 2000.

[41] D. Clarke et al, “*Certificate chain discovery in spki/sdsi,”* Journal of Computer Security, vol. 9, no. 4, pp. 285–322, 2001.

[42] Y. L. Sun, W. Yu, Z. Han and K. J. R. Liu, “*Trust Modeling and Evaluation for Ad Hoc Networks,* “ MuRI Report No. 20041017-21, October 2004.

[43] A. A. Pirzada and C. McDonald, “Establishing trust in pure ad-hoc networks ,” Proceedings of the 27th conference on Australasian computer science - Volume 26,pp. 47 – 54, Dunedin, New Zealand. 2004.

[44] P. Papadimitratos and Z. Haas, "*Secure routing for mobile ad hoc networks,*" in SCS Communication Networks and Distributed Systems Modeling and Simulation Conference, pp. 27--31, January 2002**.**

[45] K. El-Khatib, L. Korba, R. Song and G. Yee, *"Secure Dynamic Distributed Routing Algorithm for Ad Hoc Wireless Networks"* icppw , p. 359, 2003.

[46] J. Seigneur, A. Gray and C. D. Jensen, ” Trust Transfer: Encouraging Self-recommendations Without Sybil Attack,” In proceedings Trust Management: Third International Conference, iTrust 2005, Paris, France, May 23-26, 2005.

[47] http://www.palowireless.com/infotooth/images/tutorial_images/pan_adhoc_network.gif, Accessed March 2006

[48] http://www.acorn.net.au/report/adhocnetworks/adhocnet.gif, Accessed March 2006

[49] S. D. Kamvar, M. T. Schlosser, and H. Garcia-Molina, “*The eigentrust algorithm for reputation management in p2p networks,”* in Proceedings of 12th International World Wide Web Conferences, May 2003.

[50] R. Guha, R. Kumar, P. Raghavan, and A.T. Propagation, *“Propagation of trust and distrust,”* in Proceedings of International World Wide Web Conference, 2004.

[51] V. D. Park , *“A Highly Adaptive Distributed Routing Algorithm for Mobile Wireless Networks,”* IEEE Conference on Computer Communications, INFOCOM'97, pp. 1405-1413 IEEE . 7.-11. Kobe, Japan Volume 3, April 1997.

[51] A.A. Pirzada and C . McDonald,*”Secure Routing with the AODV Protocol,* Proceedings 2005 Asia-Pacific Conference on Communications, October 03-05 2005, pp. 57 – 61

[52] S. Buchegger and J. Le Boudec, *“Performance Analysis of the CONFIDANT Protocol: Cooperation Of Nodes Fairness In Dynamic Ad-hoc NeTworks,* “ Proceedings of 3rd IEEE/ACM Symposium on Mobile Ad Hoc Networking and Computing (MobiHOC), 2002, pp. 226 -236

[53] P. Michiardi and R. Molva, “CORE: A collaborative reputation mechanism to enforce node cooperation in mobile ad hoc networks, ” Sixth IFIP conference on security communications, and multimedia (CMS 2002), Portoroz, Slovenia, September 26 - 27, 2002, pp. 107 – 121

[54] G.S Yovanof and K. Erikci, “*Performance evaluation of security-aware routing protocols for clustered mobile ad hoc networks,* “ Proceedings 2004 International Workshop on Wireless Ad-Hoc Networks, 31 May-3 June 2004, pp. 286 - 290

[55] A. Boukerche et al , *“SDAR: a secure distributed anonymous routing protocol for wireless and mobile ad hoc networks,* “ Proceedings 29th Annual IEEE International Conference on Local Computer Networks, 2004. 16-18 Nov. 2004, pp. 618 - 624

[56] S. Corson and J. Macker, *“RFC 2501: Mobile Ad Hoc Networking (MANET),”* Routing Protocol Performance Issues and Evaluation Considerations, January 1999.

[57] GloMoSim, http://pcl.cs.ucla.edu/projects/glomosim Accessed March 2006

[58] OPNET documentation V.8.0.B, *OPNET Technologies Inc*., Washington DC. http://www.opnet.com/products/library/des_model_library.html Accessed March 2006

[59] NS-2, ftp://ftp.isi.edu/nsnam/ Accessed in March 2006

[60] C. Tschudin, “*Lessons from experimental MANET research* , ” http://um.netmedia.gist.ac.kr/seminar_file/20050513_hank_lessons%20from%20experimental%20manet%20research.pdf accessed January 2006

[61] D. Cavin, Y. Sasson and A. Schiper,”*On the accuracy of MENET Simulators,”* In proceedings of the Workshop on Principle of Mobile Computing (POMC’02),2002.

[62] M. Momani, J. Agbinya, G. P. Navarrete, and M. Akache, *"A New Algorithm of Trust Formation in Wireless Sensor Networks,"* in *AusWireless’06*. Sydney, Australia, March 13-16, 2006.

[63] M. Momani et al *"A New Framework of Establishing Trust in Wireless Sensor Networks,"* presented at International Conference on Computer & Communication Engineering, (ICCCE '06), Kuala Lumpur, Malaysia, 2006.

[64] R. Ramanathan and R. Hain, *”An ad hoc wireless testbed for scalable, adaptive QoS support,”* In Proceedings of the IEEE Wireless Communications and Networking Conference (WCNC), pages 998 – 1002, September 2000.

[65] F. Cristian, *“Probabilistic clock synchronization. Distributed Computing,“* pp.146–158, 1989.

[66] D. L. Mills, *“Internet Time Synchronization: The Network Time Protocol,*” In Zhonghua Yang and T. Anthony Marsland, editors, *Global States and Time in Distributed Systems*. IEEE Computer Society Press, 1994.

[67] R. Gusell and S. Zatti, *“The accuracy of clock synchronization achieved by TEMPO in Berkeley UNIX 4.3 BSD,”* IEEE Transactions on Software Engineering, pp.847–853, 1989.

[68] J. Elson, L. Girod, and D. Estrin,” *Fine Grained Network Time Synchronization using Reference Broadcasts*," In Proceedings of the Fifth Symposium on Operating Systems Design and Implementation (OSDI), pp. 147-163, Boston, MA, December 2002.

[69] D. A. Maltz, J. Broch, and D. B. Johnson, *“Experiences Designing and Building a Multi-Hop Wireless Ad Hoc Network Testbed*,” CMU School of Computer Science Technical Report CMU-CS-99-116, March 1999.

[70] W. Fischer and K. Meier-Hellstern, "*The markov-modulated poisson process (MMPP) cookbook*," Performance Evaluation, vol. 18, no. 2, pp. 149--171, 1993.

[71] A. Muqattash and K. Marwan, *“CDMA-based MAC protocol for wireless ad hoc networks,”* In (MOBIHOC '03) ,pp153-164, June 2003.

[72] A. Song, *“Thesis: picoNet II a wireless ad hoc network for mobile handheld devices,”* University of Queensland. 2001.

[73] http://www.netfilter.org/, Accessed March 2006.

[74] http://cvs.cens.ucla.edu/viewcvs/viewcvs.cgi/emstar/timesync/patches/hostap-sync-patch.diff?rev=1.1&content-type=text/vnd.viewcvs-markup, Accessed March 2006.

[75] http://www.grid.unina.it/software/ITG/download.php, Accessed March 2006.

[76] D. Umuhoza, M. Momani, J. Agbinya and C. Omlin “*On trust Metric and Trust Modelling in Mobile Ad Hoc Networks” In proceedings ICTe Africa 2006. Kenya 2006

Appendix II

List of Acronyms

AODV: Ad hoc On-demand Distance Vector

BBN: Bolt, Beranek and Newman; the last names of the three founders of BBN Technologies.

CA: Certificate Authority is trusted third party that issues digital certificates for use for other parties.

CONFIDNAT: Cooperation Of Nodes: Fairness In Dynamic Ad-hoc NeTworks

CORE: Collaborative Reputation Mechanism for node cooperation in Ad hoc Networks

CSMA: Carrier Sense Multiple Access is a MAC protocol in which before a node transmits, it verifies the availability of the physical medium.

DAG: Directed Acyclic Graph

DAWN: Density- and Asymmetry-adaptive Wireless Network

D-ITG: Distributed Internet Traffic Generator

DSDV: Dynamic Destination-Sequence- Vector

DSR: Dynamic Source Routing

GB: gigabyte is a unit storage in computer, one GB is equal to one billion byte

GloMoSim: Global Mobile system Simulator

GNU: “GNU’s Not Unix” is a free software operating system

GPS: global positioning system is a satellite navigation system

HP: Hewlett-Packard is an information technology corporations.

IBM: International Business Machines Corporation

ID: Identification

IEEE: Institute of Electrical and Electronics Engineers is an international non-profit, professional organization for the advancement of technology related to electricity.

iPAQ: is a type of Pocket Personal Computer

IP: Internet Protocol

LAN: Local area Network

LBNL: is a packet capture library in the Linux Operating System

MAC: Medium Access Control is sublayer is the part of the OSI network model data link layer

MANET: Mobile Ad hoc Network

Mbps: Megabit per second

MMC: Multimedia card is a flash memory card standard

MB: megabyte is a storage unit in computer, one megabyte is equal to one million byte

NIC: network interface card

NS-2: is a discrete event simulator developed for network simulations

OPNET: is a MANET discrete event simulator

OS: Operating System

OSPF: Open Shortest Path First

PC: Personal Computer

PDA: Personal Digital Assistants

QoS: quality of service

RBS: Reference-Broadcast Synchronization

RIP: Routing Information Protocol

RREP: Route Reply

RREQ : Route Request

SAR: Security-Aware Routing

SDAR: Secure Distributed Anonymous Routing

TAODV: Trusted Ad hoc On demand Distance Vector

TCP: Transmission Control protocol

TGA: Traffic Generator Agent

TORA: Temporally-Ordered Routing Algorithm

Wi-Fi: Wireless Fidelity is an underlying technology for WLAN based on IEEE802.11 specifications.

WLAN: wireless Local Area Network

UDP: User Datagram protocol

Appendix III

Pictures of the main Hardware

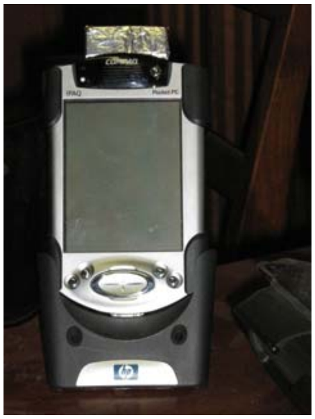

Figure 14: iPAQ h3870 with network card wrapped into an aluminum foil paper

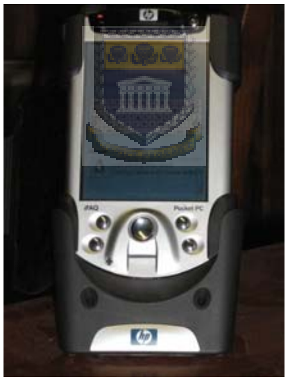

Figure 15: iPAQ h5550 in the cradle

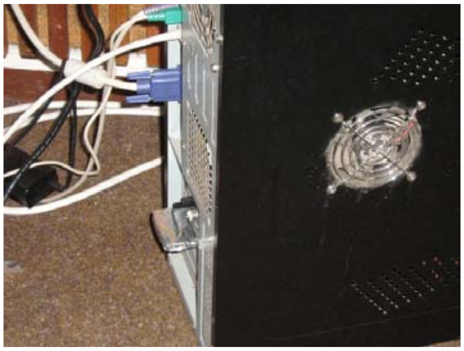

Figure 16: The PC with the wireless card

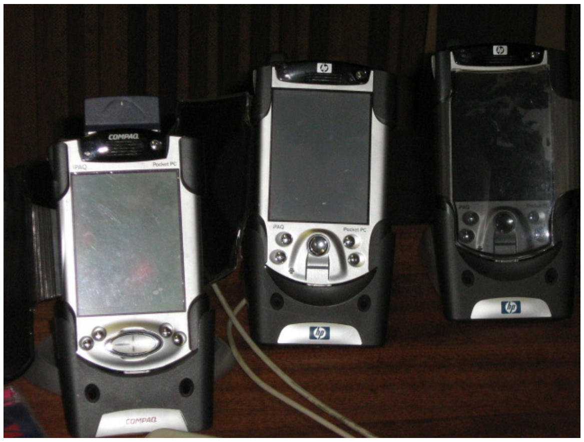

Figure 17: The three iPAQs used

Figure 18: The Netgear wireless PC card used in the PC

Figure 19: The Pretec pocket PC CompactWLAN used in the iPAQ h3870