The Path Detection (PD) protocol used to select the shortest path and DNAP (Dynamic Non-linear authentication Protocol algorithm) protocol used to find the alternate path to transfer a message to the destination.
To improve the detection accuracy to exploit the correlations between lost packets. Furthermore, to ensure truthful calculation of these correlations, develop a Dynamic Path Routing protocol (DPR) mechanism based dynamic routing privacy preserving protocol architecture that allows the detector to verify the truthfulness of the packet loss information reported by nodes. This architecture is privacy preserving, collusion proof, and incurs low communication and storage overheads. Through extensive simulations and verification proposed mechanism achieves significantly better detection accuracy than conventional methods such as a maximum-likelihood based detection.



Jayaprakash Ramasamy
Radha Balasubramanian

# Path Detection Protocol in Wireless AdHoc Networks

## An Optimal Dynamic Non-Linear Authentication Algorithm

Mr. R. Jayaprakash has done his M.Phil in Comp. Science from Bharathiar University. He did Master of Computer Applications Degree in 2010-2013 and B.Sc. Degree in 2007-2010. He is currently working as an Asst. Professor in the Department of Computer Technology, NGM College, Pollachi.

978-620-0-50521-7

LAP LAMBERT Academic Publishing

**Jayaprakash Ramasamy**
**Radha Balasubramanian**

**Path Detection Protocol in Wireless AdHoc Networks**

Jayaprakash Ramasamy
Radha Balasubramanian

# Path Detection Protocol in Wireless AdHoc Networks

## An Optimal Dynamic Non-Linear Authentication Algorithm



LAP LAMBERT Academic Publishing

**Imprint**


Cover image: www.ingimage.com

Publisher:
LAP LAMBERT Academic Publishing
is a trademark of
International Book Market Service Ltd., member of OmniScriptum Publishing Group
17 Meldrum Street, Beau Bassin 71504, Mauritius

Printed at: see last page
**ISBN: 978-620-0-50521-7**

# INTRODUCTION

## 1.1 OVERVIEW

In a multi-hop wireless network, nodes cooperate in relaying/routing traffic. An adversary can exploit this cooperative nature to launch attacks. For example, the adversary may first pretend to be a cooperative node in the route discovery process. Once being included in a route, the adversary starts dropping packets. In the most severe form, the malicious node simply stops forwarding every packet received from upstream nodes, completely disrupting the path between the source and the destination. Eventually, such a severe denial-of-service (DoS) attack can paralyze the network by partitioning its topology. Even though persistent packet dropping can effectively degrade the performance of the network, from the attacker's standpoint such an "always-on" attack has its disadvantages. First, the continuous presence of extremely high packet loss rate at the malicious nodes makes this type of attack easy to be detected. Second, once being detected, these attacks are easy to mitigate. For example, in case theattack is detected but the malicious nodes are not identified, one can use the randomized multi-path routing algorithms to circumvent the black holes generated by the attack, probabilistically eliminating the attacker's threat. If the malicious nodes are also identified, their threats can be completely eliminated by simply deleting these nodes from the network's routing table.



Hybrid wireless networks are network in which any mobile node in a wireless network may have connectivity, either directly or via gateway node, to an infrastructure network. This latter network may be an IP network as the Internet, a 3G wide area wireless network, or an 802.11 local area wireless network. Actually, any other network technology may be considered. In this context, the notion of Intra technology and Inter technology appears. If a mobile node communicates with another network of similar technology, this can be seen as Intra technology hybrid wireless network. As for example, the case of a mobile node in an ad hoc 802.11 network communicating with an 802.11 Access Point (AP) in an infrastructure network. On the other hand, if a mobile node communicates with another network of different technology, this can be seen as Inter technology hybrid wireless network. For example, the case of a mobile node in an 802.11 network communicating

with a 3G network Moreover, hybrid wireless networks may integrate both Intra and Inter technology cases and the mobile node itself may support heterogeneous technologies switching between them in an on-demand fashion.

There are several motivations for considering such hybrid networks design. Firstly, the required hardware already exists, where wireless access points are becoming ubiquitous and all laptops and many PDAs sold today are pre-installed with Wi-Fi. Also, some cell phone manufacturers started offering smart phones that integrate Inter wireless technologies, with a focus on GSM and Wi-Fi. These smart phones can be used both to offer high bandwidth Internet access when an access point is available, as well as to carry voice conversations over organizations internal network or even home networks using VoIP techniques, thereby reducing operating costs. Secondly, orchestrating this available hardware to work as a hybrid wireless network can provide architectures deployments that allow users to achieve higher throughput and switch between different types of networks, having seamless access to integrated or distributed services. Consequently, several advantages to both users and service providers/network operators are expected. For example, through offering integrated services between a 3G network and an 802.11 network, 3G operators and Wireless Internet Service Providers (WISP) could attract a wider user base and ultimately facilitates the ubiquitous introduction of high speed wireless data. Such a combined service allows enhanced performance and low overall cost. In addition, through exploiting the ad hoc network potential, to extend the coverage zone of an infrastructure network, mobile users in a Wi-Fi hotspot region can achieve service access in a seamless manner independent of their existence in a WLAN communication range and allowing the new business opportunities.



In fact, there is an increasing attention from the industry and the research community on such issue. As a result of, some hybrid wireless network architectures have emerged combining multi-hop radio relaying and infrastructure support, aiming to provide high capacity wireless networks. Also, an emerging challenge, in this context, lies in introducing the computation grid concept in such hybrid wireless networks environment. One promising trend is to harvest the widespread resources of wireless mobile devices, such as PDAs and laptops, to be beneficially useful within one or more mobile grid clusters. On the other hand, mobile nodes could benefit from the large resources in the fixed grid clusters.

## 1.2 HYBRID WIRELESS NETWORKS

A hybrid wireless network (HWN) is a blend to an infrastructure based network, where a Mobile Device (MD) may connect to a base station (BS) using wireless channels via other MDs or sensor nodes (SNs) over Internet. A large number of SNs capable of sensing, communicating, and actuating are the key elements of WSN. These special nodes are having limited processing and communication capability due to limited energy. While MANET is a collection of wireless MDs, all of which may be mobile, that dynamically create a wireless network amongst them without using any infrastructure.

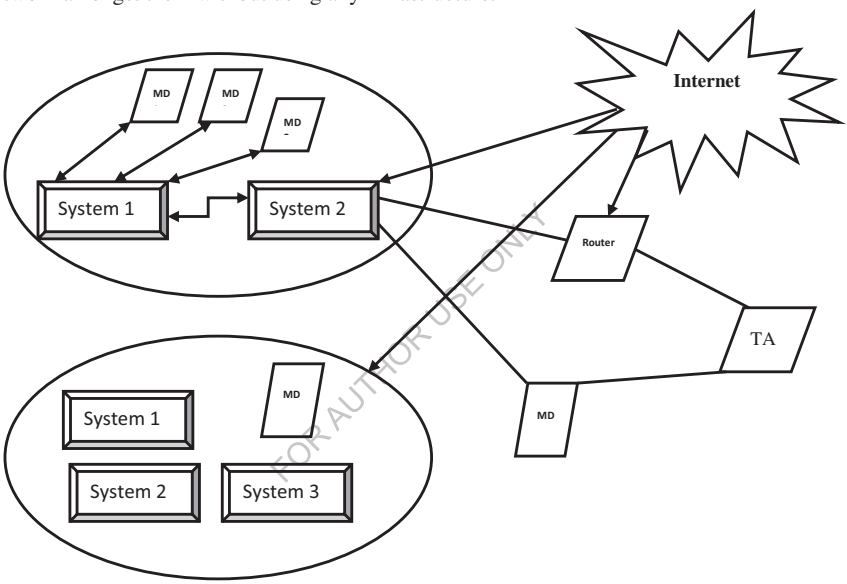





**Fig. 1.1** Hybrid wireless Network

MDs are battery operated, and battery power is restricted. Similarly autonomous, Sensor Nodes (SNs) congregate information and detect events to forward processed data in power restricted manner.

## 1.3 WIRELESS SENSOR NETWORKS

A wireless ad-hoc network is a collection of mobile/semi-mobile nodes with no pre-established infrastructure, forming a temporary network. Each of the nodes has a wireless interface and communicates with each other over either radio or infrared. Laptop computers and personal digital assistants that communicate directly with each other are some examples

of nodes in an ad-hoc network. Nodes in the adhoc network are often mobile, but can also consist of stationary nodes, such as access points to the Internet.

Semi mobile nodes can be used to deploy relay points in areas where relay points might be needed temporarily. Figure 1.2 shows a simple ad-hoc network with three nodes. The outermost nodes are not within transmitter range of each other. However the middle node can be used to forward packets between the outermost nodes. The middle node is acting as a router and the three nodes have formed an ad-hoc network.

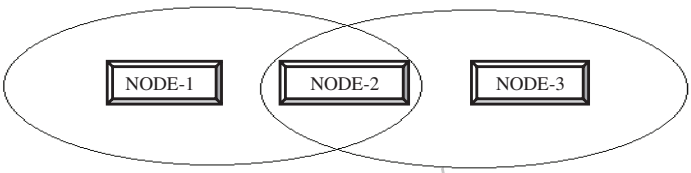





**Fig 1.2** Ad-hoc network with three participating nodes.

An ad-hoc network uses no centralized administration. This is to be sure that the network wont collapse just because one of the mobile nodes moves out of transmitter range of the others. Nodes should be able to enter/leave the network as they wish. Because of the limited transmitter range of the nodes, multiple hops may be needed to reach other nodes. Every node is wishing to participate in an ad-hoc network must be willing to forward packets for other nodes. Thus every node acts both as a host and as a router. A node can be viewed as an abstract entity consisting of a router and a set of affiliated mobile hosts. A router is an entity, which, among other things runs a routing protocol. A mobile host is simply an IP-addressable host/entity in the traditional sense.

Ad-hoc networks are also capable of handling topology changes and malfunctions in nodes. It is fixed through network reconfiguration. For instance, if a node leaves the network and causes link breakages, affected nodes can easily request new routes and the problem will be solved. This will slightly increase the delay, but the network will still be operational. Wireless ad-hoc networks take advantage of the nature of the wireless communication medium. In other words, in a wired network the physical cabling is done a priori restricting the connection topology of the nodes. This restriction is not present in the wireless domain

and, provided that two nodes are within transmitter range of each other, an instantaneous link between them may form.

## 1.4 WIRELESS ROUTING PROTOCOLS

Because of the fact that it may be necessary to hop several hops (multi-hop) before a packet reaches the destination, a routing protocol is needed. The routing protocol has two main functions, selection of routes for various source-destination pairs and the delivery of messages to their correct destination. The second function is conceptually straightforward, using a variety of protocols and data structures (routing tables). This report is focused on selecting and finding routes.

### 1.4.1 Conventional protocols

If a routing protocol is needed, why not use a conventional routing protocol like a link state or a distance vector? They are well tested and most computer communications people are familiar with them. The main problem with link-state and distance vector is, that they are designed for a static topology, which means that they would have problems to converge to a steady state in an ad-hoc network with a very frequently changing topology.



Link state and distance vector would probably work very well in an ad-hoc network with low mobility, i.e. a network where the topology is not changing very often. The problem is still remains, that link-state and distance-vector are highly dependent on periodic control messages. As the number of network nodes can be large, the potential number of destinations is also large. This requires large and frequent exchange of data among the network nodes. This is in contradiction with the fact that all updates in a wireless interconnected ad hoc network are transmitted over the air and thus are costly in resources such as bandwidth, battery power and CPU. Because both link-state and distance vector tries to maintain routes to all reachable destinations, it is necessary to maintain these routes and this also wastes resources for the same reason as above. Another characteristic for conventional protocols are that they assume bi-directional links, e.g. that the transmission between two hosts works equally well in both directions. In the wireless radio environment this is not always the case. Because many of the proposed ad-hoc routing protocols have a traditional routing protocol as underlying algorithm, it is necessary to understand the basic operation for conventional protocols like distance vector, link state and source routing.

### 1.4.2 Link State

In link-state routing, each node maintains a view of the complete topology with a cost for each link. To keep these costs consistent; each node periodically broadcasts the link costs of its outgoing links to all other nodes using flooding. As each node receives this information, it updates its view of the network and applies a shortest path algorithm to choose the next-hop for each destination. Some link costs in a node view can be incorrect because of long propagation delays, partitioned networks, etc. Such inconsistent network topology views can lead to formation of routing-loops. These loops are however short-lived, because they disappear in the time it takes a message to traverse the diameter of the network.

### 1.4.3 Distance Vector

In distance vector each node only monitors the cost of its outgoing links, but instead of broadcasting this information to all nodes; it periodically broadcasts to each of its neighbors an estimate of the shortest distance to every other node in the network. The receiving nodes then use this information to recalculate the routing tables, by using a shortest path algorithm. Compared to link-state, distance vector is more computation efficient, easier to implement and requires much less storage space. However, it is well known that distance vector can cause the formation of both short-lived and long-lived routing loops. The primary cause for this is that the nodes choose their next-hops in a completely distributed manner based on information that can be stale.



### 1.4.4 Source Routing

Source routing means that each packet must carry the complete path that the packet should take through the network. The routing decision is therefore made at the source. The advantage with this approach is that it is very easy to avoid routing loops. The disadvantage is that each packet requires a slight overhead.

### 1.4.5 Flooding

Many routing protocols uses broadcast to distribute control information, that is, send the control information from an origin node to all other nodes. A widely used form of broadcasting is flooding and operates as follows. The origin node sends its information to its neighbors (in the wireless case, this means all nodes that are within transmitter range). The

neighbors relay it to their neighbors and so on, until the packet has reached all nodes in the network. A node will only relay a packet once and to ensure this some sort of sequence number can be used. This sequence number is increased for each new packet a node sends.

### 1.4.6 Protocol Classification

Routing protocols can be classified into different categories depending on their properties.

- Centralized vs Distributed
- Static vs Adaptive
- Reactive vs Proactive

One way to categorize the routing protocols is to divide them into centralized and distributed algorithms. In centralized algorithms, all route choices are made at a central node, while in distributed algorithms, the computation of routes is shared among the network nodes. Another classification of routing protocols relate to whether they change routes in response to the traffic input patterns. In static algorithms, the route used by source-destination pairs is fixed regardless of traffic conditions. It can only change in response to a node or link failure. This type of algorithm cannot achieve high throughput under a broad variety of traffic input patterns. Most major packet networks use some form of adaptive routing where the routes used to route between source-destination pairs may change in response to congestion A third classification that is more related to ad-hoc networks is to classify the routing algorithms as either proactive or reactive. Proactive protocols attempt to continuously evaluate the routes within the network, so that when a packet needs to be forwarded, the route is already known and can be immediately used. The family of Distance-Vector protocols is an example of a proactive scheme. Reactive protocols, on the other hand, invoke a route determination procedure on demand only. Thus, when a route is needed, some sort of global search procedure is employed. The family of classical flooding algorithms belongs to the reactive group. Proactive schemes have the advantage that when a route is needed, the delay before actual packets can be sent is very small. On the other side proactive schemes needs time to converge to a steady state. This can cause problems if the topology is changing frequently.

## 1.5 APPLICATION AREAS OF WIRELESS SENSOR NETWORKS

WSNs have opened the eye of new generation scientists to observe never before phenomenon, paving the way for designing of numerous applications. These applications of WSNs can be classified into categories.

Monitoring the interactions of things with each other and the encompassing space monitoring includes environmental and habitat monitoring, precision agriculture, indoor climate control, surveillance, treaty verification and intelligent alarms. Whereas monitoring things includes structural monitoring, eco-physiology, condition-based equipment maintenance, medical diagnostics and urban terrain mapping. Further, the most dramatic applications of WSN involve monitoring complex interactions, including wildlife habitats, disaster management, emergency response, ubiquitous computing environments, asset tracking, manufacturing process flow and healthcare.

Some of the major applications



**a) Habitat monitoring**

Researchers in the life sciences are becoming increasingly concerned about potential impacts of the human presence in monitoring plants and animals in field conditions. For example the seabird colonies are notorious for their sensitivity to human disturbance. Therefore the WSNs can be used to gather information on the habitat of a plant/animal without disturbing them. The gathered data can be analyzed later on to learn optimal environmental conditions favorable for the flora/fauna's growth.

**b) Military**

The use of WSN can provide real time information of the enemy activities to commando teams thus making coordination and planning more effective. The sensing, monitoring and decision-making should be integrated seamlessly, for designing effective military applications. The accurate and timely gathering of visual surveillance and intelligence data can play a central role in attaining objectives as well as minimizing loss of human lives.

**c) Home Automation**

Networking various home appliances, such as vacuum cleaners, micro-wave ovens, and refrigerators, with wireless medium, has been dreamt for many years. Embedded sensors inside such appliances can interact with each other, and with the external network via the internet or satellites. They allow users to manage home devices locally and remotely more easily.

**d) Precision Agriculture**

The WSNs monitors environmental conditions in which farming is done to make it more profitable and sustainable. The WSNs are proving useful for controlling in economical way climate, irrigation and nutrient supply to produce best crop condition, increase in production efficiency while decreasing cost. They are also helping in strategic planning and counter measures to increase yield of the crop.



**e) Healthcare**

Sensors are used in biomedical applications for healthcare. Sensors are implanted in the human body for monitoring medical problems such as cancer and help patients to maintain their health.

**f) Building monitoring**

Sensors can be used in buildings for detection of fire and smoke. In case of fire a network of sensors deployed in a huge building can track the source and direction in which fire is expanding. In addition, sensors can be used to monitor vibration that could damage the structure of a building.

**g) Environmental observation**

WSNs can be used to monitor environment such as forest fire detection, flood detection, air pollution detection, rainfall observation in agriculture etc. Sensor nodes can be used for detection of toxic waste, illegally dumped into the lake by a factory located nearby and relaying the exact origin of a pollutant to a centralized authority, which then can take appropriate measures, to limit spread of the pollution. Without the WSN, it would be difficult

to get the data without the nearby factory's knowledge, in which case the factory would prevent the data gathering process.

**h) Disaster Management**

The reliable early warning system based on WSN can be deployed in areas with high risk of disasters. The use of WSN promise to provide real time information of the disaster area to rescue teams making coordination and planning more effective. Location information of victims, rescuers and objects in the disaster is vital for the rescue operations. It has been known that, for an operationally effective disaster management: sensing, monitoring and decision-making should be integrated seamlessly. Timely and updated disaster information is extremely important for efficient response and effective actions, it will help disaster managers make better decisions and take actions in time.

## 1.6 PACKET DROPPING IN WIRELESS AD HOC NETWORKS



Like in any other network, packet loss is expected in ad hoc networks at least to an acceptable percentage. Not all packets lost should be viewed as malicious. In this section, we discuss some of the packet loss scenarios in wireless ad hoc networks.

### 1.6.1 Legitimate Packet Dropping

Packet dropping can be experienced in wireless ad hoc networks where no compromised nodes are present. This packet loss is mainly associated with the following events;

**a) Network Congestion**

Network congestion in wireless ad hoc networks is something unavoidable. These networks are mainly scalable due to in and out movements of nodes. As a result, congestion is more likely to happen which can lead to loss of packets.

**b) Channel Conditions**

In wireless networking the channel condition cannot be neglected since it changes drastically. Free path loss, interference, presence of noise on the channel and fading of the transmitted wireless signals are among the channel conditions that can lead to packet loss or

bit errors in the transmitted signal. In the presence of these factors, some packets can get dropped.

**c) Resource Constraints**

Nodes in wireless ad hoc networks have limited energy resource. Intermediate nodes in these networks may behave selfishly and fail to forward the received packets in order to conserve their limited resources battery power. These packets in turn get dropped.

### 1.6.2 Malicious Packet Dropping

Mostly, the first step in launching a packet dropping attack is for a malicious node to get involved during route formation. This is better done by exploiting the vulnerabilities of the underlying well known routing protocols used in wireless ad hoc networks which are designed basing on the assumption of trustworthiness between nodes in a network.

Once in the route, the malicious node can do anything including maliciously dropping packets. This Packet dropping at a malicious intermediate node can lead to suspension of communication or generation of wrong information between the source and destination which is an undesirable situation. For better understanding, following are the illustrations of malicious packet dropping scenarios in wireless ad hoc networks under the two commonly used routing protocols AODV (Ad hoc On Demand Distance Vector) and OLSR (Optimized Link State Routing).



### 1.6.3 Packet Dropping in AODV

The route discovery process between source (S) and destination (D) under AODV routing protocol is as illustrated in Figure 1.3. The source broadcasts a RREQ (Route Request) message with unique identifier to all its one hop neighbors. Each receiver rebroadcasts this message to its one hop neighbors until it reaches the destination. The destination on receiving the message updates the sequence number of the source and sends a RREP (Route Reply) message back to its neighbor which relayed the RREQ. On the other hand, an intermediate node that has a route to the destination with destination sequence number equal to the one in RREQ can send back a RREP packet to the source node without relaying to the destination. For a node to launch packet dropping attack, it must be involved in at least one routing paths in the network. This is illustrated in Figure 1.4, C is a malicious node intending to drop

packets from S to D. To discover a path from S to D, S first broadcasts RREQ packet to its neighbors. Each neighboring node continues to rebroadcast this message as explained earlier until it reaches D.

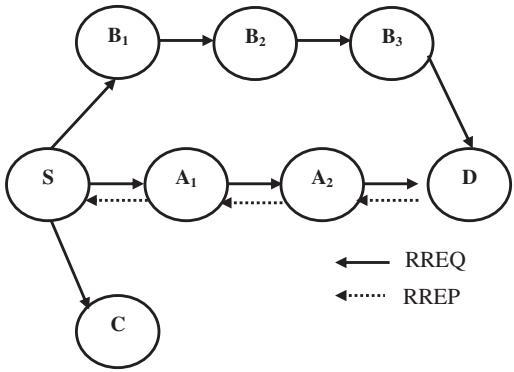





**Figure 1.3: Route Discovery in AODV**

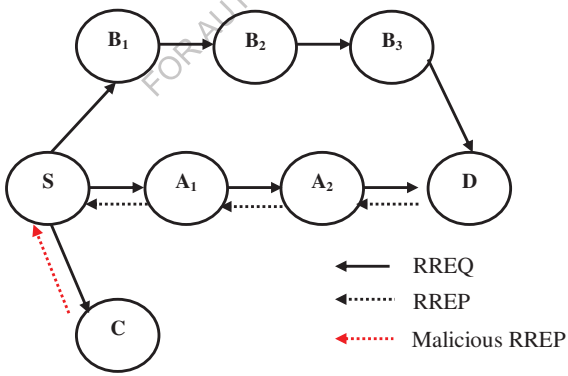



**Figure 1.4: Packet Dropping Attack in AODV**

The malicious node C disobeys this rule and lies to S claiming it has the shortest path to D and sends a RREP packet to S. As a result, S assumes that the shortest route to D is through C and starts to send data packets to D through C which are in turn dropped.

### 1.6.4 Packet Dropping in OLSR

OLSR uses Multipoint Relays (MPRs) which are set of neighboring nodes that are responsible for spreading the local link state information to the whole network for optimization. The link state is broadcasted periodically through Topology Control (TP) messages. Each node in OLSR selects its MPR set from its one hop neighbors such that it can easily reach all its two hop neighbors with minimum number of retransmissions. Selection of the MPR depends on the number of two hop neighbors reachable through the candidate node and it's "Willingness" value obtained from "Hello" message which indicates the readiness of a node to forward packets of its neighbors.

Through periodic exchange of link state, each node senses its neighbors and disseminates the network topology. Each node constructs a partial topology graph of the network from broadcasted TC messages which allows it to establish routes to non-neighboring nodes. For a packet dropping attack, a malicious node may send a TC message claiming to be a MPR of nodes although it may not. As the network depends on the MPRs for routing services, the malicious node may decide to drop packets passing through it.



## 1.7 INTRODUCTION ABOUT THE PROPOSED SYSTEM

The thesis is to design to a secure controller based forwarding misbehavior packet drop detection system that uses a privacy preserving based detection delay sampling algorithm and a verification mechanism to detect the malicious packet drops introduced by Wireless Ad Hoc Networks.

The proposed mechanism is based on detecting the correlations between the lost packets over each hop of the path. The basic idea is to model the packet loss process of a hop as a random process alternating between 0 (loss) and 1 (no loss). Specifically, consider that a sequence of $M$ packets that are transmitted consecutively over a wireless channel. By observing whether the transmissions are successful or not, the receiver of the hop obtains a bitmap ($a_1, \ldots, a_M$), where $a_j \in \{0, 1\}$ for packets $j = 1, \ldots, M$. The correlation of the lost packet is calculated as the auto-correlation function of this bitmap. Under different packet

dropping conditions, i.e., link-error versus malicious dropping, the instantiations of the packet-loss random process should present distinct dropping patterns (represented by the correlation of the instance). This is true even when the packet loss rate is similar in each instantiation.

## 1.8 OBJECTIVE AND MOTIVATION OF THE RESEARCH

Wireless ad hoc networks have gained lots of attention due to their ease and low cost of deployment. This has made ad hoc networks of great importance in numerous military and civilian applications. But, the lack of centralized management of these networks makes them vulnerable to a number of security attacks. One of the attacks is packet drop attack, where a compromised node drops packet maliciously. Several techniques have been proposed to detect the packet drop attack in wireless ad hoc networks. Therefore, the proposed system review some of the packet drop attack detection techniques and comparatively analyze them basing on; their ability to detect the attack under different attack strategies (partial and or cooperate attacks), environments and the computational and communication overheads caused in the process of detection.



The proposed the monitoring agent technique. The technique is based on capturing packets sent by neighboring nodes within a transmission range. All the nodes in a network collect information about their one hop neighbors within a certain period of time. The collected information include; the total number of packets transmitted from a particular node, the average number of transmitted packets from all its one hop neighbors, the packet drop rate of a particular one hop neighbor, and the average packet dropping rate by all its one hop neighbors which are used for identifying a malicious node.

In a multi-hop wireless network, nodes cooperate in relaying/routing traffic. An adversary can exploit this cooperative nature to launch denial-of-service (DoS) attacks. For example, the adversary may first pretend to be a cooperative node in the route discovery process. Once being included in a route, the adversary may start maliciously dropping packets.

In the most straightforward form of this attack, the malicious node simply stops forwarding packets received from upstream nodes, completely disrupting the traffic delivery between the source and the destination. Eventually, such severe DoS attacks can paralyze the network by partitioning its topology. Even though persistent packet dropping can effectively

degrade the performance of the network, from the attacker's standpoint performing such an "always-on" attack has its disadvantages in terms of the ease of detection. A malicious node that is part of the route can actually exploit its knowledge of the network protocols and the communication context to launch an insider's attack, aiming at achieving the same attack effect but at a much lower risk of being detected. Specifically, the malicious node can identify the importance of various packets and drop a small number of packets that are deemed highly critical to the performance of the network.

Detecting malicious selective packet dropping is extremely challenging in a highly dynamic wireless environment. The difficulty stems from the requirement that we need not only detect the location (or hop) where the packet drop took place, but also identify whether the drop is intentional or not. Specifically, because of the open nature of the wireless medium, the quality of the channel typically fluctuates due to fading, shadowing, interference, and background noise. As a result, a packet drop in the route could be caused by harsh channel conditions (a.k.a., link errors) or by malicious behavior. In some cases, e.g., a highly mobile environment, link errors are quite significant. So, a malicious node can camouflage its attack under the background of harsh channel conditions by selectively dropping a small number of highly important packets. In this case, observing the packet loss rate is not enough to accurately identify the exact cause of a packet loss, because the packet drop rate by the malicious node is comparable to that of wireless link errors. Clearly, deciding whether a packet drop is intentional or unintentional in such an ambiguous setup is a challenging problem.



The features of the proposed system are as follows

- All nodes may work in promiscuous mode.
- Misbehaving nodes do not drop acknowledgement packets.
- Misbehaving nodes do not work in groups.
- Misbehaving nodes do not send or forward false acknowledgement packet.

# LITERATURE REVIEW

## 2.1 802.11 MARKOV CHANNEL MODELING [1]

**Authors:** *J. N. Arauz* (***2004***)

In order to understand the behavior of upper layer protocols and to design or fine tune their parameters over wireless networks, it is common to assume that the underlying channel is a flat Rayleigh fading channel. Such channels are commonly modeled as finite state Markov chains. Recently, hidden Markov models have also been employed to characterize these channels. Although Markov models have been widely used to study the performance of communications protocols at the link and transport layers, no validation of their accuracy has been performed against experimental data. These models are not applicable to frequency selective fading channels. Moreover, there are no good models to consider the effects of path loss (average received SNR), the packet size, and transmission rate variations which are significant in IEEE 802.11 wireless local area networks. This research performs validation of Markov models with experimental data and discusses the limitations of the process. In this dissertation, we present different models that have been proposed along with their validity analysis. We use the experimental data with stochastic modeling approaches to characterize the frame losses in IEEE 802.11 wireless LANs. We also characterize the important factor of current wireless LAN technology, the transmission rate variations. New guidelines for the construction of Markov and hidden Markov models for wireless LAN channels are developed and presented along the necessary data to implement them in performance studies. Furthermore we also evaluate the validity of using Markovian models to understand the effects on upper layer protocols such as TCP.



## 2.2 PROVABLE DATA POSSESSION AT UNTRUSTED STORES [2]

**Authors:** *C. Ateniese, et.al* (***2007***)

We introduce a model for provable data possession (PDP) that allows a client that has stored data at an untrusted server to verify that the server possesses the original data without retrieving it. The model generates probabilistic proofs of possession by sampling random sets of blocks from the server, which drastically reduces I/O costs. The client maintains a constant amount of metadata to verify the proof. The challenge/response protocol transmits a small,

constant amount of data, which minimizes network communication. Thus, the PDP model for remote data checking supports large data sets in widely-distributed storage system.

The authors presented two provably-secure PDP schemes that are more efficient than previous solutions, even when compared with schemes that achieve weaker guarantees. In particular, the overhead at the server is low (or even constant), as opposed to linear in the size of the data. Experiments using our implementation verify the practicality of PDP and reveal that the performance of PDP is bounded by disk I/O and not by cryptographic computation.

The efficient PDP scheme is the fundamental construct underlying an archival introspection system that we are developing for the long-term preservation of Astronomy data. We are taking possession of multi-terabyte Astronomy databases at a University library in order to preserve the information long after the research projects and instruments used to collect the data are gone. The database willbe replicated at multiple sites. Sites include resource-sharing partners that exchange storage capacity to achieve reliability and scale. As such, the system is subject to freeloading in which partners attempt to use storage resources and contribute none of their own. The location and physical implementation of these replicas are managed independently by each partner and will evolve over time. Partners may even outsource storage to third-party storage server providers. Efficient PDP schemes will ensure that the computational requirements of remote data checking do not unduly burden the remote storage sites.



## 2.3 PROOFS OF STORAGE FROM HOMOMORPHIC IDENTIFICATION PROTOCOLS [3]

**Author:** *G. Ateniese, S. Kamara, and J. Katz* (***2009***)

Proofs of storage (PoS) are interactive protocols allowing a client to verify that a server faithfully stores a file. Previous work has shown that proofs of storage can be constructed from any homomorphic linear authenticator (HLA). The latter, roughly speaking, are signature/message authentication schemes where `tags' on multiple messages can be homomorphically combined to yield a `tag' on any linear combination of these messages. We provide a framework for building public-key HLAs from any identification protocol satisfying certain homomorphic properties. We then show how to turn any public-key HLA into publicly-verifiable PoS with communication complexity independent of the file length and supporting an unbounded number of verifications. We illustrate the use of our

transformations by applying them to a variant of an identification protocol by Shoup, thus obtaining the first unbounded-use PoS based on factoring (in the random oracle model).

Advances in networking technology and the rapid accumulation of information have fueled a trend toward outsourcing data management to external service providers (“servers"). By doing so, organizations can concentrate on their core tasks rather than incurring the substantial hardware, software and personnel costs involved in maintaining data “in house".

Outsourcing storage prompts a number of interesting challenges. One problem is to verify that the server continually and faithfully stores the entire file ~f entrusted to it by the client. The server is untrusted in terms of both security and reliability: it might maliciously or accidentally erase the data or place it onto temporarily unavailable storage media. This could occur for numerous reasons including cost-savings or external pressures (e.g., government censure). The server might also accidentally erase some data and choose not to notify the client. Exacerbating the problem (and precluding na□ve approaches) are factors such as limited bandwidth between the client and server, as well as the client's limited resources.



## 2.4 ODSBR: AN ON-DEMAND SECURE BYZANTINE RESILIENT ROUTING PROTOCOL FOR WIRELESS AD HOC NETWORKS [4]

**Authors:** *B. Awerbuch* (***2008***)

Ad hoc networks offer increased coverage by using multi-hop communication. This architecture makes services more vulnerable to internal attacks coming from compromised nodes that behave arbitrarily to disrupt the network, also referred to as Byzantine attacks. In this work we examine the impact of several Byzantine attacks performed by individual or colluding attackers. The authors proposed ODSBR, the first on-demand routing protocol for adhoc wireless networks that provides resilience to Byzantine attacks caused by individual or colluding nodes. The protocol uses an adaptive probing technique that detects a malicious link after log n faults have occurred, where n is the length of the path. Problematic links are avoided by using a route discovery mechanism that relies on a new metric that captures adversarial behavior. Our protocol never partitions the network and bounds the amount of damage caused by attackers. We demonstrate through simulations ODSBR’s effectiveness in mitigating by zantine attacks. Our analysis of the impact of these attacks versus the adversary’s effort gives insights into their relative strengths, their interaction and their importance when designing multi-hop wireless routing protocols

The goal of this work is to provide routing survivability under an adversarial model where any intermediate node or group of colluding nodes perform Byzantine attacks. While some existing work provides protection against specific attacks that may be conducted by a single Byzantine node against different routing components, no other existing work provides an ad hoc wireless routing protocol for coping with a large set of attacks available to a set of colluding Byzantine attackers and targeting both route discovery and data forwarding.

The circumvent of this obstacle by avoiding the assignment of "guilt" to individual nodes. Instead, whenever the endpoints of a link disagree, we deduce that at least one of them is faulty; therefore the link is considered faulty and should be avoided. Our method ensures that as long as a fault-free path exists between two nodes, they can communicate reliably even if an overwhelming majority of the network acts in a Byzantine manner. We focus on attacks at the network layer and do not consider at tacks against the MAC or physical layers.

### 2.5 TWOACK: PREVENTING SELFISHNESS IN MOBILE ADHOC NETWORKS[6]



**Authors:** *K. Balakrishnan, J. Deng, and P. K. Varshney* (***2005***)

Mobile Ad hoc Networks (MANETs) operate on the basic underlying assumption that all participating nodes fully collaborate in self-organizing functions. However, performing network functions consumes energy and other resources. Therefore, some network nodes may decide against cooperating with others. Providing these selfish nodes, also termed misbehaving nodes, with an incentive to cooperate has been an active research area recently. In this paper, we propose two network-layer acknowledgment-based schemes, termed the TWOACK and the S-TWOACK schemes, which can be simply added-on to any source routing protocol. The TWOACK scheme detects such misbehaving nodes, and then seeks to alleviate the problem by notifying the routing protocol to avoid them in future routes. Details of the two schemes and our evaluation results based on simulations are presented in this paper. We have found that, in a network where up to 40% of the nodes may be misbehaving, the TWOACK scheme results in 20% improvement in packet delivery ratio, with a reasonable additional routing overhead.

The TWOACK scheme can be implemented on top of any source routing protocol such as DSR. This follows from the fact that a TWOACK packet derives its route from the source route established for the corresponding data packet. The TWOACK scheme uses a special

type of acknowledgment packets called TWOACK packets, which are assigned a fixed route of two hops (or three nodes) in the direction opposite to that of data packets.

The values assigned to thresh and timeout play an important role in determining the effectiveness of the TWOACK scheme. These parameters should be large enough so that intermittent failures or excessive transmission delays (due to collisions) of TWOACK packets are not interpreted as misbehavior. On the other hand, they should not be so large that a significant number of data packets are lost before a misbehaving node (link) is detected.

### 2.6 SHORT SIGNATURES FROM THE WEIL PAIRING [7]

**Authors:** *D. Boneh, B. Lynn, and H. Shacham* (***2004***)

The authors introduced a short signature scheme based on the Computational Diffie–Hellman assumption on certain elliptic and hyper elliptic curves. For standard security parameters, the signature length is about half that of a DSA signature with a similar level of security. Our short signature scheme is designed for systems where signatures are typed in by a human or are sent over a low-bandwidth channel. We survey a number of properties of our signature scheme such as signature aggregation and batch verification.



Short digital signatures are needed in environments where a human is asked to manually key in the signature. For example, product registration systems often ask users to key in a signature provided on a CD label. More generally, short signatures are needed in low-bandwidth communication environments. For example, short signatures are needed when printing a signature on apostage stamp. Currently, the two most frequently used signatures schemes, RSA and DSA, provide relatively long signatures compared to the security they provide. For example, when one uses a 1024-bit modulus, RSA signatures are 1024 bits long. Similarly, when one uses a 1024-bit modulus, standard DSA signatures are 320 bits long.

### 2.7 PERFORMANCE ANALYSIS OF THE CONFIDANT PROTOCOL (COOPERATION OF NODES: FAIRNESS IN DYNAMIC ADHOC NETWORKS), [8]

**Authors:** *S. Buchegger and J. Y. L. Boudec,* (***2002***)

Mobile ad-hoc networking works properly only if the participating nodes cooperate in routing and forwarding. However, it may be advantageous for individual nodes not to cooperate. We propose a protocol, called CONFIDANT, for making misbehavior

unattractive; it is based on selective altruism and utilitarianism. It aims at detecting and isolating misbehaving nodes, thus making it unattractive to deny cooperation. Trust relationships and routing decisions are based on experienced, observed, or reported routing and forwarding behavior of other nodes. The detailed implementation of CONFIDANT in this paper assumes that the network layer is based on the Dynamic Source Routing (DSR) protocol. We present a performance analysis of DSR fortified by CONFIDANT and compare it to regular defenseless DSR. It shows that a network with CONFIDANT and up to 60% of misbehaving nodes behaves almost as well as a benign network, in sharp contrast to a defenseless network. All simulations have been implemented and performed in GloMoSim.

## 2.8 STIMULATING COOPERATION IN SELFORGANIZING MOBILE AD HOC NETWORKS [9]

**Authors:** *L. Buttyan and J. P. Hubaux* (***2003***)

In military and rescue applications of mobile ad hoc networks, all the nodes belong to the same authority; therefore, they are motivated to cooperate in order to support the basic functions of the network. In this paper, we consider the case when each node is its own authority and tries to maximize the benefits it gets from the network. More precisely, we assume that the nodes are not willing to forward packets for the benefit of other nodes. This problem may arise in civilian applications of mobile ad hoc networks. In order to stimulate the nodes for packet forwarding, we propose a simple mechanism based on a counter in each node. We study the behavior of the proposed mechanism analytically and by means of simulations, and detail the way in which it could be protected against misuse.



The authors addressed the problem of stimulating cooperation in self-organizing, mobile ad hoc networks for civilian applications. We assume that each node belongs to a different authority, its user, which has full control over the node. In particular, the user can tamper with the software and the hardware of the node, and modify its behavior in order to better adapt it to her own goals (e.g., to save battery power). We understand that regular users usually do not have the required level of knowledge and skills to modify their nodes. Nevertheless, our assumption is still reasonable, because criminal organizations can have enough interest and resources to reverse engineer a node and sell tampered nodes with modified behavior on a large scale. The experience of cellular network shows that as soon as

the nodes are under the control of the end-users, there is a strong temptation to alter their behavior in one way or another.

## 2.9 MODELLING INCENTIVES FOR COLLABORATION IN MOBILE AD HOC NETWORKS [10]

**Authors:** *J. Crowcroft, R. Gibbens, F. Kelly, and S. Ostring* (***2003***)

This paper explores a model for the operation of an ad hoc mobile network. The model incorporates incentives for users to act as transit nodes on multi-hop paths and to be rewarded with their own ability to send traffic. The paper explores consequences of the model by means of uid-level simulations of a network and illustrates the way in which network resources are allocated to users according to their geographical position.

The approach to route selection and allocation closely follows the theoretical formulation given. The formulation develops a mechanism to allow nodes to make decentralized decisions concerning the choice of the allows on potential routes. The nodes make these decisions based on congestion prices announced by relevant nodes. In this way, nodes with a given willingness to-pay for congestion costs can adjust their resource usage accordingly. The approach in this paper builds on by the incorporation of power as well as bandwidth prices to redirect additional constraints that arise in wireless networks.



## 2.10 ROUTING AMID COLLUDING ATTACKERS [11]

**Authors:** *J. Eriksson, M. Faloutsos, and S. Krishnamurthy* (***2007***)

The authors proposed a practical solution to the long-standing problem of secure wireless routing in the presence of multiple colluding attackers. Our secure routing protocol, Sprout 1, probabilistically generates a multiplicity of routes from a source to any given destination. Routing is done in two stages, route generation and route selection. In the route generation stage, a large number of routes is generated without concern for performance. Instead, the objective is to generate a highly diverse set of routes. This makes Sprout extremely resilient to attack, compared to deterministic routing algorithms. It the route selection stage, to avoid compromised routes, and to ensure good overall performance, the source continuously evaluates the quality of each active route, by means of signed end-to-end acknowledgments. The amount of traffic sent on each route is adjusted accordingly. Sprout

effectively mitigates the vast majority of known routing layer attacks, even when under assault from a large number of colluding attackers. Extensive experimentation on our 31node testbed demonstrates the real-world performance of Sprout in terms of packet delivery ratio, round-trip times and TCP throughput. We perform a security analysis of our protocol, and demonstrate through simulation that Sprout is able to quickly find working paths in networks of up to 200 good nodes and up to 64 colluding attackers. Overall, Sprout consistently delivers high, reliable performance in benign as well as hostile environments.

### 2.11 CASTOR: SCALABLE SECURE ROUTING FOR AD HOC NETWORKS [12]

**Authors:** *W. Galuba, et.al* (***2010***)

Wireless ad hoc networks are inherently vulnerable, as any node can disrupt the communication of potentially any other node in the network. Many solutions to this problem have been proposed. In this paper, we take a fresh and comprehensive approach that addresses simultaneously three aspects: security, scalability and adaptability to change the network conditions. Our communication protocol, Castor, occupies a unique point in the design space: it does not use any control messages except simple packet acknowledgments, and each node makes routing decisions locally and independently without exchanging any routing state with other nodes. Its novel design makes Castor resilient to a wide range of attacks and allows the protocol to scale to large network sizes and to remain efficient under high mobility. We compare Castor against four representative protocols from the literature. Our protocol achieves up to two times higher packet delivery rates, particularly in large and highly volatile networks, while incurring no or only limited additional overhead. At the same time, Castor is able to survive more severe attacks and recovers from them faster.



### 2.12 DETECTING MALICIOUS PACKET DROPPING IN THE PRESENCE OF COLLISIONS AND CHANNEL ERRORS IN WIRELESS AD HOC NETWORKS [13]

**Authors:** *T. Hayajneh, P. Krishnamurthy, D. Tipper, and T. Kim,* (***2009***)

Detecting malicious packet dropping is important in ad hoc networks to combat a variety of security attacks such as black hole, grey hole, and wormhole attacks. We consider the detection of malicious packet drops in the presence of collisions and channel errors and describe a method to distinguish between these types. We present a simple analytical model

for packet loss that helps a monitoring node to detect malicious packet dropping attacks. The model is analyzed and evaluated using simulations. The results show that it is possible to detect malicious packet drops in the presence of collisions and channel errors.

## 2.13 SORI: A SECURE AND OBJECTIVE REPUTATION-BASED INCENTIVE SCHEME FOR AD HOC NETWORKS [14]

**Authors:** *Q. He, D. Wu, and P. Khosla* (***2004***)

In an ad-hoc network, intermediate nodes on a communication path are expected to forward packets of other nodes so that the mobile nodes can communicate beyond their wireless transmission range. However, because wireless mobile nodes are usually constrained by limited power and computation resources, a selfish node may be unwilling to spend its resources in forwarding packets which are not of its direct interest, even though it expects other nodes to forward its packets to the destination. It has been shown that the presence of such selfish nodes degrades the overall performance of a non-cooperative ad hoc network. To address this problem, we propose a Secure and Objective Reputation-based Incentive (SORI) scheme to encourage packet forwarding and discipline selfish behavior. Different from existing schemes, under our approach, the reputation of a node is quantified by objective measures, and the propagation of reputation is efficiently secured by a one-way-hash-chain-based authentication scheme. Armed with the reputation-based mechanism, we design a punishment scheme to penalize selfish nodes. The experimental results show that the proposed scheme can successfully identify selfish nodes and punish them accordingly.



In this paper, we make the following assumptions.

1) The nodes in an ad hoc network under our consideration are non-cooperative in packet forwarding, that is, a node is not willing to forward packets of other nodes unless it can benefit from the packet forwarding. If nodes are cooperative, e.g., in military ad hoc networks, there is no need to use incentive mechanisms.

2) There is no conspiracy among nodes.

3) Broadcast transmission: A packet can be received by all the neighbors of the transmitting node (within its transmission range) because of the broadcast nature of the wireless medium.

4) Desire to communicate: All the participating nodes have the desire to communicate with some others.

5) Invariant identity: No node changes its identity during its life time.

6) Selfish but not malicious: A node may be selfish in terms of conservation of power and computing resources, but not malicious, which means that it will not try something that could be more expensive in consuming energy and computing resources than cooperating in packet forwarding.

7) Promiscuous Mode: Each node operates in a promiscuous mode, i.e., each node listens to every packet transmitted by its neighbors even if the packet is not intended for the node; and each node is able to determine who transmits the packet.

## 2.14 DSR: THE DYNAMIC SOURCE ROUTING PROTOCOL FOR MULTI-HOP WIRELESS AD HOC NETWORKS [15]

**Authors:** *D. B. Johnson, D. A. Maltz, and J. Broch* (***2001***)



The Dynamic Source Routing protocol (DSR) is a simple and efficient routing protocol designed specifically for use in multi-hop wireless ad hoc networks of mobile nodes. DSR allows the network to be completely self-organizing and self-configuring, without the need for any existing network infrastructure or administration. The protocol is composed of the two mechanisms of Route Discovery and Route Maintenance, which work together to allow nodes to discover and maintain source routes to arbitrary destinations in the ad hoc network. The use of source routing allows packet routing to be trivially loop-free, avoids the need for up-to-date routing information in the intermediate nodes through which packets are forwarded, and allows nodes forwarding or overhearing packets to cache the routing information in them for their own future use. All aspects of the protocol operate entirely on-demand, allowing the routing packet overhead of DSR to scale automatically to only that needed to react to changes in the routes currently in use. We have evaluated the operation of DSR through detailed simulation on a variety of movement and communication patterns, and through implementation and significant experimentation in a physical outdoor ad hoc networking testbed we have constructed in Pittsburgh, and have demonstrated the excellent performance of the protocol. In this chapter, we describe the design of DSR and provide a summary of some of our simulation and testbed implementation results for the protocol.

We assume that all the nodes wish to communicate with other nodes within the ad hoc network are willing to participate fully in the protocols of the network. In particular, each node participating in the network should also be willing to forward packets for other nodes in the network. We refer to the minimum number of hops necessary for a packet to reach from any node located at one extreme edge of the ad hoc network to another node located at the opposite extreme, as the diameter of the ad hoc network. We assume that the diameter of an ad hoc network will often be small (e.g., perhaps 5 or 10 hops), but may often be greater than 1. Packets may be lost or corrupted in transmission on the wireless network. A node receiving a corrupted packet can detect the error and discard the packet. Nodes within the ad hoc network may move at any time without notice, and may even move continuously, but we assume that the speed with which nodes move is moderate with respect to the packet transmission latency and wireless transmission range of the particular underlying network hardware in use. In particular

## 2.15 DEALING WITH LIARS: MISBEHAVIOR IDENTIFICATION VIA RENYI-ULAM GAMES [16]



**Authors:** *W. Kozma Jr. and L. Lazos* (***2009***)

The authors discussed the problem of identifying misbehaving nodes that refuse to forward packets in wireless multi-hop networks. We map the process of locating the misbehaving nodes to the classic Renyi-Ulam game of 20 questions. Compared to previous methods, our mapping allows the evaluation of node behavior on a per-packet basis, without the need for energy-expensive overhearing techniques or intensive acknowledgment schemes. Furthermore, it copes with colluding adversaries that coordinate their behavioral patterns to avoid identification and frame honest nodes. We show via simulations that our algorithms reduce the communication overhead for identifying misbehaving nodes by at least one order of magnitude compared to other methods, while increasing the identification delay logarithmically with the path size.

## 2.16 REACT: RESOURCE-EFFICIENT ACCOUNTABILITY FOR NODE MISBEHAVIOR IN ADHOC NETWORKS BASED ON RANDOM AUDITS [17]

**Authors:** *W. Kozma Jr., and L. Lazos* (***2009***)

Wireless ad hoc networks rely on multi-hop routes to transport data from source to destination. The routing function is implemented in a collaborative manner, with each node respossible for relaying traffic to the destination. However, an increasingly sophisticated pool of users with easy access to commercial wireless devices, combined with the poor physical and software security of the devices, can lead to node misconfiguration or misbehavior. A misbehaving node may refuse to forward packets in order to conserve its energy (selfishness), or simply degrade network performance (maliciousness). In this paper, we investigate the problem of uniquely identifying the set of misbehaving nodes that refuse to forward packets. We propose a novel misbehavior identification scheme called REACT that provides resource-efficient account-ability for node misbehavior. REACT identifies misbehaving nodes based on a series of random audits triggered upon a performance drop. We show that a source-destination pair using REACT can identify any number of independently misbehaving nodes based on behavioral proofs provided by nodes. Proofs are constructed using Bloom filters which are storage-efficient membership structures, thus significantly reducing the communication overhead for misbehavior detection.



## 2.17 AN ACKNOWLEDGEMENT-BASED APPROACH FOR THE DETECTION OF ROUTING MISBEHAVIOR IN MANETS [18]

**Authors:** *A. Nasipuri and S. Das* (***2000***)

The authors presents the results of studies under taken routing misbehavior in MANETs (Mobile Ad Hoc Networks). The node misbehaviors may be introduced, due to the open structure and scarcely available battery-based energy, and such routing misbehavior is caused by the selfish nodes that when processor participate in the route discovery and maintenance refuses to forward the data packets. In the present studies proposed a novel scheme named 2ACK which provides an add-on technique for routing schemes that detects the routing misbehavior and to overcomes their adverse effect. The main feature of 2ACK is to send two-hop acknowledgment packets in the opposite direction of the routing path and to reduce additional routing overhead. The performances of the proposed scheme were analyzed and simulated and 95% packet delivery ratios were achieved when 40% misbehaving nodes were present in the MANETs.

In order to demonstrate the adverse effect of routing misbehavior, it is proposed to estimate the probability of misbehaving routes. A route which misbehaves when there is at least one router along the same route is termed as misbehaving route. The analysis was carried out with the following assumptions.

- The network nodes are randomly distributed over the entire network area. Each node's location is independent of other. There are N nodes in the network area of size $X * Y$ ;
- The source and the destination of each transmissions were chosen randomly among other nodes;
- Nodes (other than the source and the destination) are chosen as misbehaving nodes and are independent with probability denoted as pm.

# SYSTEM ANALYSIS

## 3.1 EXISTING SYSTEM

Depending on how much weight a detection algorithm gives to link errors relative to malicious packet drops, the related work can be classified into the following two categories. The first category aims at high malicious dropping rates, where most (or all) lost packets are caused by malicious dropping. In this case, the impact of link errors is ignored. Most related work falls into this category. Based on the methodology used to identify the attacking nodes, these works can be further classified into four sub-categories. The first sub-category is based on credit systems [9], [10].

- A credit system provides an incentive for cooperation. A node receives credit by relaying packets for others, and uses its credit to send its own packets.
- A reputation system relies on neighbors to monitor and identify misbehaving nodes.
- The shape of traffic at the MAC layer of the source node according to a certain statistical distribution, so that intermediate nodes are able to estimate the rate of received traffic by sampling the packet arrival times.
- The Bloom filters to construct proofs for the forwarding packets at each node. By examining the relayed packets at successive hops along a route, one can identify suspicious hops that exhibit high packet loss rates.



## 3.2 DRAWBACKS IN EXISTING SYTEM

- As a credit system result, a maliciously node that continuous to drop packets will eventually deplete its credit, and will not be able to send its own traffic.
- A reputation node with a high packet dropping rate is given a bad reputation by its neighbors. This reputation information is propagated periodically throughout the network and is used as an important metric in selecting routes. Consequently, a malicious node will be excluded from any route.
- All methods mentioned above do not perform well when malicious packet dropping is highly selective. More specifically, for the credit-system-based method, a malicious node may still receive enough credits by forwarding most of the packets receive from upstream nodes.

## 3.3 SYSTEM REQUIREMENTS

### 3.3.1 Hardware Requirements

- Processor : Pentium IV
- Speed : 2.5 GHz
- RAM : 1 GB RAM
- Hard Disk Drives : 40 GB
- Monitor : 15” Color Monitor

### 3.3.2 Software Requirements

- Operating System : Fedora 8
- Simulator : NS 2.34

## 3.4 SOFTWARE DESCRIPTIONS



Network simulator 2 is used as the simulation tool in this project. NS was chosen as the simulator partly because of the range of features it provides and partly because it has an open source code that can be modified and extended. There are different versions of NS and the latest version is ns-2.1b9a while ns-2.1b10 is under development Network simulator (NS) is an object–oriented, discrete event simulator for networking research. NS provides substantial support for simulation of TCP, routing and multicast protocols over wired and wireless networks. The simulator is a result of an ongoing effort of research and developed. Even though there is a considerable confidence in NS, it is not a polished product yet and bugs are being discovered and corrected continuously.

NS is written in C++, with an OTcl1 interpreter as a command and configuration interface. The C++ part, which is fast to run but slower to change, is used for detailed protocol implementation. The OTcl part, on the other hand, which runs much slower but can be changed very fast quickly, is used for simulation configuration. One of the advantages of this split-language program approach is that it allows for fast generation of large scenarios. To simply use the simulator, it is sufficient to know OTcl. On the other hand, one disadvantage is that modifying and extending the simulator requires programming and debugging in both languages.

NS can simulate the following:

1. **Topology:** Wired, wireless
2. **Scheduling Algorithms:** RED, Drop Tail,
3. **Transport Protocols:** TCP, UDP
4. **Routing:** Static and dynamic routing
5. **Application:** FTP, HTTP, Telnet, Traffic generators

### 3.4.1 User's View of Ns-2

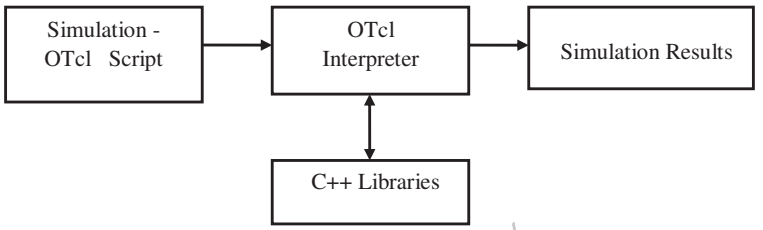



**Fig. 3.1** Architecture of NS-2



### 3.4.2 Network Components

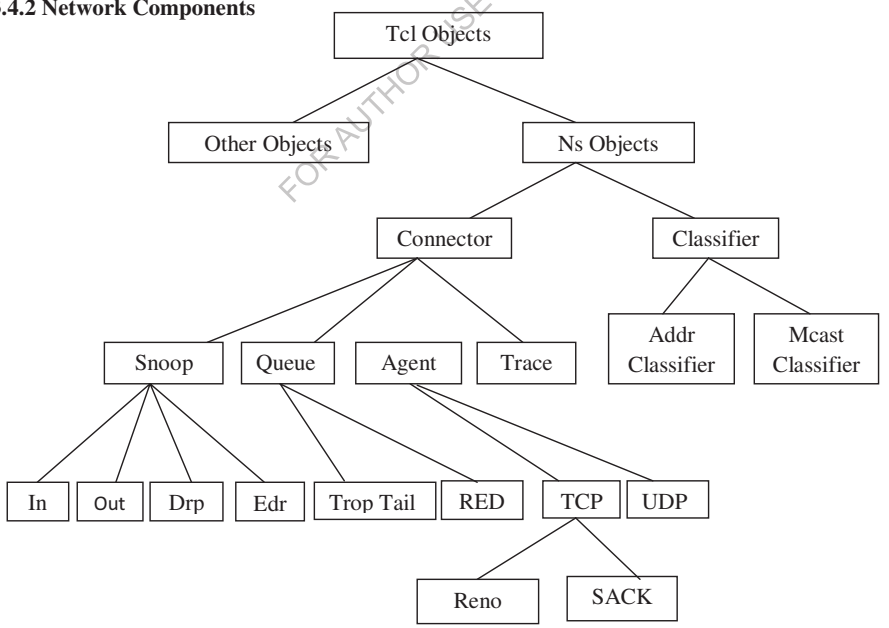



**Fig 3.2** OTcl Class Hierarchies

This section talks about the NS components, mostly compound network components. Figure 3.2 shows a partial OTcl class hierarchy of NS, which will help understanding the basic network components.

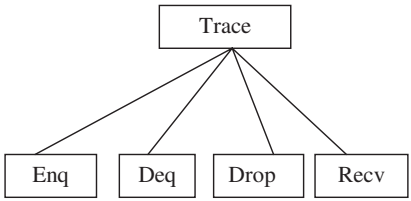



**Fig 3.3** Trace Hierarchies

The root of the hierarchy is the TclObject class that is the super class of all OTcl library objects (scheduler, network components, timers and the other objects including NAM related ones). As an ancestor class of TclObject, NsObject class is the super class of all basic network component objects that handle packets, which may compose compound network objects such as nodes and links. The basic network components are further divided into two subclasses, Connector and Classifier, based on the number of the possible output DATA paths. The basic network and objects that have only one output DATA path are under the Connector class, and switching objects that have possible multiple output DATA paths are under the Classifier class.



### 3.4.3 Class Tcl

The class Tcl encapsulates the actual instance of the OTcl interpreter and provides the methods to access and communicate with that interpreter, code. The class provides methods for the following operations:

1. Obtain a reference to the Tel instance
2. Invoke OTcl procedures through the interpreter
3. Retrieve, or pass back results to the interpreter
4. Report error situations and exit in an uniform manner
5. Store and lookup "Tcl Objects"

6. Acquire direct access to the interpreter.

- Obtain a Reference to the class Tcl instance

  A single instance of the class is declared in -tclcl/Tcl.cc as a static member variable.

The statement required to access this instance is Tel& tel = Tcl::instance();

- Invoking OTcl Procedures

There are four different methods to invoke an OTcl command through the instance, tcl. They differ essentially in their calling arguments. Each function passes a string to the interpreter that then evaluates the string in a global context. These methods will return to the caller if the interpreter returns TCL_OK. On the other hand, if the interpreter returns TCL_ERROR, the methods will call tkerror{}. The user can overload this procedure to selectively disregard certain types of errors.

1. **Passing Results to/from the Interpreter :** When the interpreter invokes a C++ method, it expects the result back in the private member variable, tcl-> result.

2. **Error Reporting and Exit:** This method provides a uniform way to report errors in the compiled code.



### 3.4.4 Command Methods: Definition and Invocation

For every Tcl Object that is created, ns establishes the instance procedure, cmd{}, as a hook to executing methods through the compiled shadow object. The procedure cmd{} invokes the method command() of the shadow object automatically, passing the arguments to cmd{} as an argument vector to the command() method. The user can invoke the cmd {} method in one of two ways, by explicitly invoking the procedure, specifying the desired operation as the first argument, or implicitly, as if there were an instance procedure of the same name as the desired operation. Most simulation scripts will use the latter form. Consider the distance computation in SRM is done by the compiled object. It is often used by the interpreted object. It is usually invoked as $srmObject distance? (agentAddress)If there is no instance procedure called distance? the interpreter will invoke the instance procedure unknown{}, defined in the base class TclObject. The unknown procedure then invokes $srmObject cmd distance? (agentAddress) to execute the operation through the compiled

object's command() procedure. The user could explicitly invoke the operation directly. One reason for this might be to overload the operation by using an instance procedure of the same name.

For example,

```
Agent/SRM/Adaptive instproc distance? addr {
$self instvar distanceCache_($addr)
if![info exists distanceCache_($addr)] {
set distanceCache_($addr) [$self cmd distance? $addr]
}
set distanceCache_($addr)
}
```

The following shows how the command() method using SRMAgent::command()

```
int ASRMAgent::command(int argc, const char*const*argv) {
Tcl& tcl = Tcl::instance();
if (argc == 3) {
if (strcmp(argv[1], "distance?") == 0) {
int sender = atoi(argv[2]);
SRMinfo* sp = get_state(sender);
tcl.tesultf("%f", sp->distance_);
return TCL_OK;    '
}
}
```

return (SRMAgent::command(argc, argv));

The following observations are made from this piece of code:

The function is called with two arguments. The first argument (argc) indicates the number of arguments specified in the command line to the interpreter. The command line arguments vector (argv) consists of argv[0] contains the name of the method, "cmd" and argv[1] specifies the desired operation. If the user specified any arguments, then they are placed in argv[2...(argc - 1)]. The arguments are passed as strings. They must be converted to the appropriate data type.If the operation is successfully matched, the match should return the result of the operation, command () itself must return either TCL_OK or TCL_ERROR to indicate success or failure as its return code. If matched in this method, it must invoke its parent's command method, and return the corresponding result. This permits the user to conceive of operations as having the same inheritance properties as instance procedures or compiled methods. In the event that this command method is defined for a class with multiple inheritance, one of two implementations can be chosen Either they can invoke one of the parent's command method, and return the result of that invocation. They can each of the parent's command methods in some sequence, and return the result of the first invocation that is successful. If none of them are successful, then they should return an error.



### 3.4.5 Mobile Networking In Ns

The wireless model essentially consists of the Mobile Node at the core with additional supporting features that allows simulations of multi-hop ad-hoc networks, wireless LANs etc. The Mobile Node object is a split object. The C++ class Mobile Node is derived from parent class Node. A Mobile Node thus is the basic Node object with added functionalities of a wireless and mobile node like ability to move within a given topology, ability to receive and transmit signals to and from a wireless channel etc. A major difference between them is that a mobile Node is not connected by means of Links to other nodes or mobile nodes.

Mobile Node is the basic nsNode object with added functionalities like movement, ability to transmit and receive on a channel that allows it to be used to create mobile, wireless simulation environments. The class Mobile Node is derived from the base class Node. The four ad-hoc routing protocols that are currently supported are, Dynamic Source Routing (DSR), Temporally ordered Routing Algorithm (TORA) and Adhoc On-demand Distance Vector (AODV).

The general structure for defining a mobile node in ns2 is described as follows:

$ns node-config -adhocRouting $opt (adhocRouting)

-IIType $opt (II)

-macType $opt (mac)

-ifqType $opt (ifq) -ifqLen $opt (ifqlen)

-antType $opt (ant)

-proplnstance [new $opt (prop) -phyType $opt (netif)

-channel [new $opt (chan)]

-topolnstance $topo

-wiredRouting OFF

-agent Trace ON

-router Trace OFF

-macTrace OFF



The above API configures for a mobile node with all the given values of ad hoc-routing protocol, network stack, channel, topography, propagation model, with wired routing turned on or off (required for wired-cum-wireless scenarios) and tracing turned on or off at different levels (router, mac, agent).

# RESEARCH METHODOLOGY

## 4.1 METHODOLOGY ANALYSIS

The proposed architecture accepts the network parameters as input which contains the NS2 simulator where the dynamic routing privacy preserving algorithm is applied to the wireless ad hoc network. This architecture in figure 4.1 follows a path from the start to end state. The users initialize the number of mobile nodes as network parameters in which the routing process is to be evaluated.

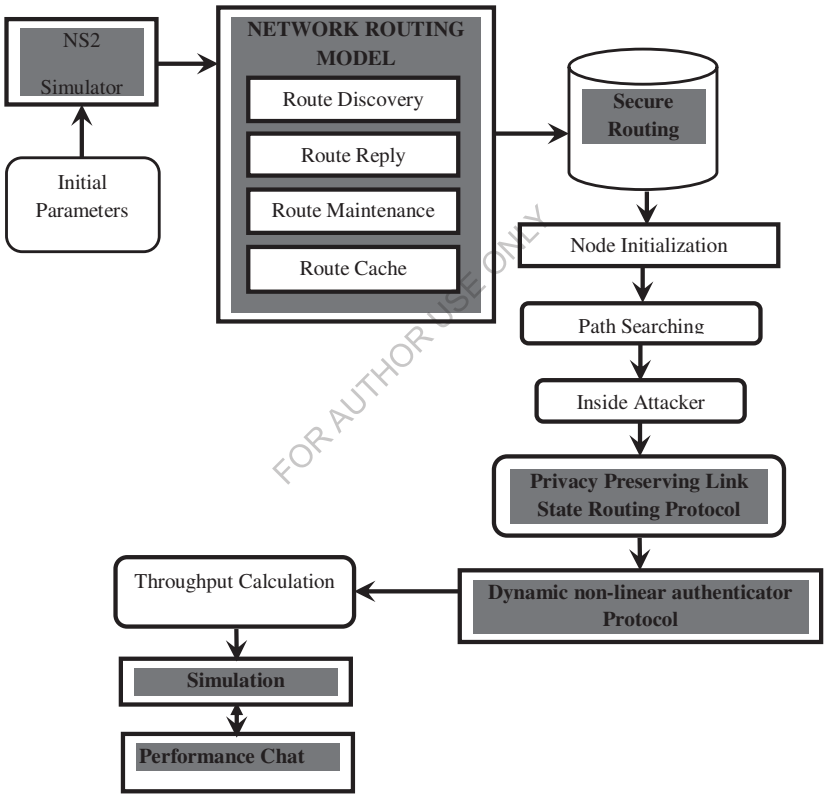



**Figure 4.1 Architecture of Proposed System**

In a multi-hop wireless ad hoc network, packet losses are attributed to harsh channel conditions and intentional packet discard by malicious nodes. In this paper, while observing a sequence of packet losses, we are interested in determining whether losses are due to link errors only, or due to the combined effect of link errors and malicious drop. We are especially interested in insider's attacks, whereby a malicious node that is part of the route exploits its knowledge of the communication context to selectively drop a small number of packets that are critical to network performance. Because the packet dropping rate in this case is comparable to the channel error rate, conventional algorithms that are based on detecting the packet loss rate cannot achieve satisfactory detection accuracy. To improve the detection accuracy, we propose to exploit the correlations between lost packets. Furthermore, to ensure truthful calculation of these correlations, we develop a link state routing protocol (LSRP) based dynamic routing privacy preserving protocol architecture that allows the detector to verify the truthfulness of the packet loss information reported by nodes. This architecture is privacy preserving, collusion proof, and incurs low communication and storage overheads. Through extensive simulations, we verify that the proposed mechanism achieves significantly better detection accuracy than conventional methods such as a maximum-likelihood based detection



Malicious nodes can take advantage of this covert channel to hide their misbehavior and reduce the chance of being detected. For example, an upstream malicious node may drop a packet on Path of source to destination ($P_{SD}$), but may secretely send this packet to a downstream malicious node via the covert channel. When being investigated, the downstream malicious node can provide a proof of the successful reception of the packet. This makes the auditor believe that the packet was successfully forwarded to the downstream nodes, and not know that the packet was actually dropped by an upstream attacker.

**The Proposed System**

The methodology right from the network parameters as input until the link state routing protocol based on Dynamic routing algorithm measurement is explained as a step by step process below.

In wireless ad hoc networks, nodes communicate with each other using multi hop wireless links. Data to out of range nodes can be routed through intermediate nodes. That node in

wireless ad hoc networks can act as both hosts and routers. The following methodology are listed below

1. Network Model
2. Network Routing Model
3. Privacy Preserving Link State Routing Protocol
4. Dynamic Non-linear Authenticator Protocol

### 4.1.1 Network Model

The simulations are evaluated in networks of 49 nodes. As the number of nodes in the ad hoc network is increased, the size of the simulation area is also increased so that a consistent node density is maintained. The simulation areas are 330m x 330m, 670m x670m and 1000m x1000m, respectively. All mobile nodes move according to the random waypoint mobility model. Node speeds are randomly distributed between zero and some maximum, where the maximum speed varies between 0 and 20 m/s. The pause time is consistently 10 seconds. Each data point represents an average of 10 runs with the same traffic models, but different randomly generated mobility scenarios. The second set of simulations examine the performance of two routing schemes with different percentages of Internet (wired) traffic. All traffic is CBR traffic with 512 byte data packets at the sending rate of 10 packets per second. All the sources are within the ad hoc network; the correspondent nodes are either within the ad hoc network or reachable through the wired network.



### 4.1.2 Network Routing Model

The network model process as follows Route Discovery, Route Reply, Route Maintenance and Route cache.

**Route Discovery**

This mechanism is launched whenever a node wishes to send or contact a destination node which isn't in its transmission range therefore it must obtain a route to that node by launching the Route discovery mechanism. The Route discovery mechanism. Normally the sender must first search this route in its route cache if there is no route it precedes as follow:

- It creates a route request packets containing its address and the address of the destination node then it broadcast this packet to all its neighbors using flooding.

- Each neighbor when receiving this request consults its cache to find an eventual route to this destination to be returned back to the sender otherwise it rebroadcast the same route request to all its neighbors after adding its address to the header of the route request and learns from this request information to be added to its cache. If the node has already treated this route request it ignores the new received request by verifying its sequence number since each route request is identified by a unique sequence number.

- The same procedure is executed by each neighboring node until the route request arrives to destination which add its address at the end of the header and sends a route reply.

**Route Reply**

The Route reply mechanism is executed by a node after receiving a route request destined to him thus this node executes the following actions:

- Adds this new route to its cache for future use.

- Add its address at the end of the path contained in the header of packets.

- Replies to this request using unicast along the path contained in the header.



**Route Maintenance**

When forwarding a packet each intermediate node is responsible for confirming that the packet is correctly received by the next node, however due to the dynamic topology and the constraints of the wireless medium it may occur some situation where a node doesn't receive the acknowledgement of reception from link layer of a given packet, therefore it resends the same packet it until reaches a predefined value of attempts. Whenever this number of attempt was reached this node consider this link as broken than it deletes each route containing this link from its cache than it generates a route error packet to inform the source node and all intermediate nodes about this link failure in the same way each intermediate node deletes all routes containing this route until the route error packet arrives to its destination which chooses to launch a new route request or to find a new route in its route cache.

**Route Cache**

The route cache is used to maintain frequently used routes in order to avoid new route discovery mechanism which consumes lot of network resources in the way that each new discovered route is saved in the route cache of the corresponding node for future use, a node can also learns from route request to adds new routes to its cache it also learns from route error packets to update its cache.

### 4.1.3 Privacy Preserving Link State Routing Protocol

Privacy Preserving Link State Routing Protocol based on the shortest path is usually energy saving optimized. So different metrics are considered and weight is assigned to each link. Between two end-to-end nodes, usually exists more than one route. In the potential relay node set, there will be relatively energy-optimal routes that can achieve the least cost based on the nodes' battery capacity and propagation loss of the links. The research work have a simple multi-hop Hetro-network, with the relay node set $\Re$ between the source and destination, and the immediate neighbor set $\Re^*$ for each node, there exists an energy efficient route. For example, the route with relay nodes A, B, and C. Links with less propagation power loss and nodes with higher residual battery capacity are preferred. So the problem is simplified to minimize the power consumed during transmission and maximize the battery capacity of the next node to be used that is to minimize:



$$\frac{p(i)}{g(i)} \quad i \in \Re^* \qquad (1)$$

for local (the immediate next hop) optimization

$$\sum_{i \in \Re} \frac{p(i)}{g(i)} \qquad i \in \Re \qquad (2)$$

for global (all end-to-end hops) optimization where *g(i)* is the residual battery capacity of the *i*th node, and *p(i)* is the power cost per packet from node *i*-1 to node *I* (that it, Joules per second per packet). A detailed study of the Lithium-Ion battery discharging property is presented. The voltage decrease and the battery capacity are non-linear functions of discharging time: the lower the capacity remains, the faster the battery voltage drops. The residual battery capacity can be evaluated as the amount of energy remains in the battery, that is, the time duration for the battery to discharge when the transmitter is consuming power.

The residual battery capacity is reduced for the amount of energy consumed by the transmitter. If we define *f(i) = I/g(i)* and expand *p(i)*, then (1) for local optimization will be,

$$\frac{p(i)}{g(i)} = [P_{loss}(i-1,i) + P_{rx}(i) + P_c(i)] \cdot f(i) \quad (3)$$

where the power cost per packet *p(i)* from node *i*-1 to node *i* can be expanded to the sum of the power loss of this link (from node *i*-1 to *i*), the power cost to receive the packet at the *i*th node , and the power cost for routing messages to maintain this connection. The algorithm favors a link with less power loss and hence reduces the amount of energy consumed by potential retransmission and link error. Usually the minimum threshold of receiving power of the receiver is constant (for instance, -80 dBm for current IEEE 802.11b cards) for all receivers (i.e, independent of the node index *i*). So the minimum value of prx(i) can be set as a constant prx. Since the routing messages for route discovery and maintenance are the same for all nodes for on-demand routing protocols, we can consider pc(i) a constant value pc too. Hence, both control and data packets are considered to consume energy according to their packet sizes. Note also that, when more link error occurs, more routing maintenance is needed and more energy is consumed.



$$\sum_{i \in \Re} \frac{p(i)}{g(i)} = \sum_{i \in \Re} f(i).[P_{loss}(i-1,i) + P_{rx}(i) + P_c(i)].f(i) \quad (4)$$

This algorithm can either optimize locally for each hop or globally for the end-to-end route between a source-destination for the global optimization, the data source will get to know the summation of the cost for all possible routes and decide which route to choose, based on the global cost function. While for local optimization, each intermediate node will choose locally a different next hop to forward data for energy efficiency from the local cost function. Global optimization tends to prefer routes with fewer hops (because cost function is a summation and is an implicit function of hop count) and hence can achieve less delay.

### 4.1.4 Dynamic Non-linear authenticator Protocol

Dynamic Non-linear authenticator Protocol selects best routes based on the current state information for the network. The state information can be predicted or measured but the

route will change depending on the available state information at the time of the traffic request. The privacy network can cope now with the dynamics of traffic and react to real-time network traffic accordingly, by introducing real-time behavior and state dependency in order to avoid congestion and to achieve optimal performance.

Dynamic routing protocol is distinguished by two factors:

- The computational model that the routing service is using
- The state information nature

There are two computational models used in Dynamic Routing the centralized and the distributive.

The basic operation of privacy preserving is allowing a source to specify a destination area and simultaneously discover multiple nodes in it. However, to keep the description simple, we assume that only one node exists within each destination area.



An alternate path through two links, A and B, with PP parameters (Privacy Preserving parameter on a link if that link is to be used as an alternative path) rA and rB, is considered least-loaded if it has the lowest value loadA, B where

$$load_{A,B} = \min\{(r_A - load_A), (r_B - load_B)\}$$

This quantity is often referred to as the PP permissibility of a path since it is a reflection of the bandwidth available in addition to the reservation parameters. A negative PP permissibility would indicate that an alternate route is not available while a large PP permissibility indicates an underutilized path. This is a computationally intensive routing logical decision to find the best route when any available route can carry the call but its shown to have better performance than either Dynamic Non-Hierarchical Routing or homogenous routing protocol. The Dynamic routing protocol algorithm may prove extensible to multiple classes of service the network is currently operating in this mode.

**Algorithm for Dynamic Non-linear authenticator Protocol**

**Step 1** If the Source node *S* wants to send data to the destination node *D*, it will first send *REQ* message to all its neighbor nodes.

**Step 2** When neighbor nodes receive *REQ* message they will check their broadcast, if this packet's *ID* is already in their Cache then packet will be discarded.

**Step 3** Otherwise, node will calculate its energy by using: *Enew* = *Etx* - *Er* + *Eth* + *Em* + *Eover* and send this value as a reply to source node.

**Step 4** Source node will calculate the mean value of all the values of Enew of all the nodes and send a *RREQ* message to the node whose *Enew* value is nearest to the mean value.

**Step 6** Assign the Attacker node depending on the routing environment.

**Step 7** When the node receives a *RREQ* message it will send privacy preserving message to its own neighbors and this process will be continued till the destination node reaches.

**Step 8** When destination node will receive the *RREQ* message it will send the *RREP* message back with the same route.

# IMPLEMENTATION

The implementation is developing the dynamic privacy preserving measurement uses NS2.34 simulator. The protocol will provide a link state routing search in the network and prevent the packet drop in distributed manner.

## 5.1 NEIGHBOR DISCOVERY

In this step, each node establishes a two-hop neighbor information base. At the network initialization phase, this information can be established by exchanging 'Hello' messages. In each 'Hello' message, a node will advertise its ID, location, energy, and number of neighbors. In order to make sure that each node is able to receive a 'Hello' message from all neighbors, we implement a timer. As long as the timer does not expire, nodes will process any 'Hello' message received. In processing 'Hello' messages, the neighbor field will be examined and the node will update its database. This implementing timer also ensures that nodes are allowed sufficient time to complete the discovery process.



## 5.2 SHORTEST PATH SELECTION

In this step, the information from is used to compute the minimum tour length. The problem is modeled using the well-known searching method. The operation of the graph representation is mapped onto that of each node and link states are mapped onto each nodes. The link state routing will start its tour at the sink, visit each node in sequence according to the computed path, and then ultimately return to the sink.

Link state routing (LSR) search assumes that when the numbers of cities are given, and each should be visited by each nodes, the challenge is to find the shortest tour that involves the shortest distances at the lowest cost. The protocol is considered an NP-Complete problem, where it is not guaranteed that the optimum solution will be reached. LSR has been used in a great deal of research in different fields and its success has been demonstrated. Replacing the links with nodes and the packets with the mobile collector, the problem has been used in many WSN applications and protocols to optimize the tour for gathering data in WSNs.

Generally, LSR are categorized into two types: symmetrical and asymmetrical. The symmetrical LSRs are the cases where the distance (or cost) from starting point to end point is equal to the distance (cost). In this type of LSR, when number of nodes is given, there are always visible solutions. This task is to find which one of these solutions is the shortest in distance unit or least in cost unit. On the other hand, with the second type of LSR, the asymmetrical ones, the distance (cost) between two nodes differs by the direction. Thus, the distance (cost) from to is *not equal* to the distance. In this case, when number of nodes is given, there will be solutions. LSR usually assumes that the nodes can move freely from any directions, no matter which node the starting point is.

### 5.3 SIMULATION TIME CALCULATION

At this point, the simulation time can calculate its time by taking the last two points into account. Each tour time can be represented as

$$T_t = \left(\min_{dis} T\,/MA\_v\right) + (data/DR)$$

Where min_*dis* is the minimum distance of the LSR, *MA_v* is the node velocity by the distance unit over the time unit, *data* is the collected data for each Routing Path (RP), and the *DR* is the data rate for each node.

## EXPERIMENTAL RESULTS

In the Proposed System nearly 49 nodes are taken for simulation process. The nodes are arranged based on the network model and route request process mechanism. The fig 6.1 shows how the nodes are arranged and finds the exact place of an each node to transfer the data.

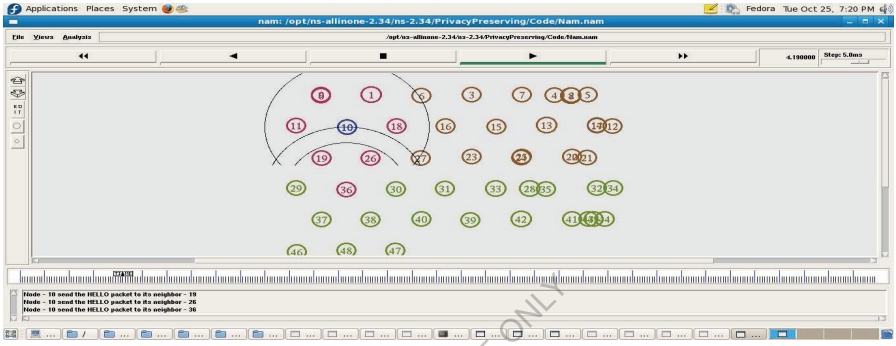



**Figure 6.1 Network Model and Route Request process**



Identifying the nodes for discovering the route is useful for the broadcasting purpose. Making an advertisement to each interface which contains list of all routers address through that broadcasting method can be done easily which is shown in fig 6.2.

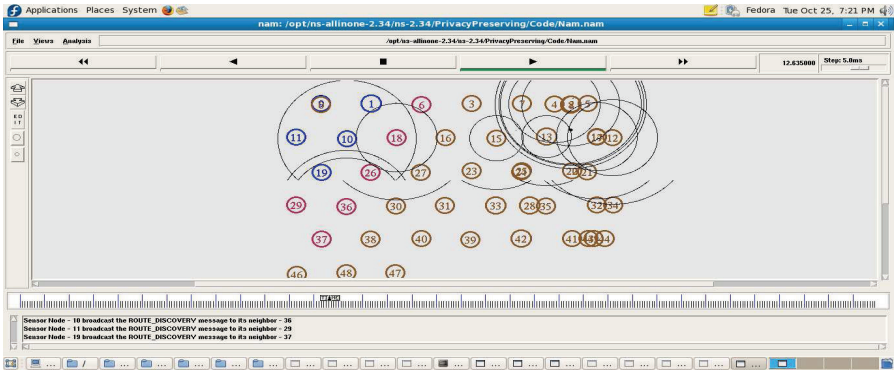



**Figure 6.2 Route Discovery Broadcasting process**

In fig 6.3, Choosing the source and destination nodes to transfer the data based on that the interface choose the intermediate path for the broadcast.

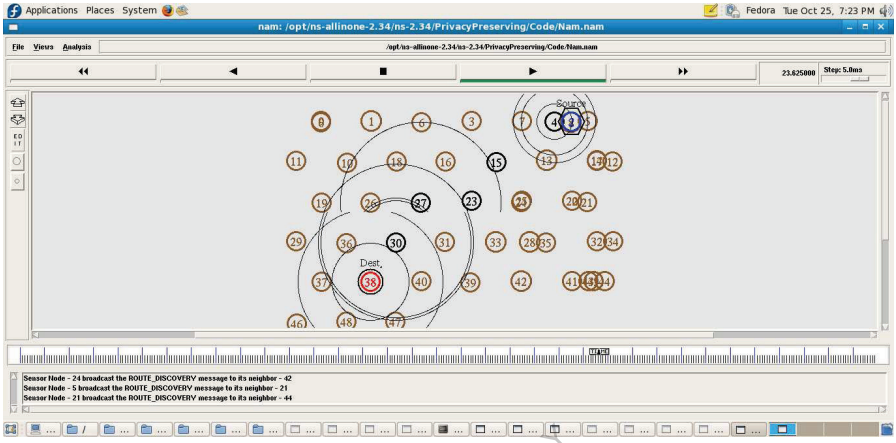



**Figure 6.3 Packet Sending source to destination**



Here we select the source node and destination node which shown in red and blue color for identification respectively and intermediate nodes are referenced as black color, now we send a packet from source to destination through the intermediate interface that shown in a below screen shot.

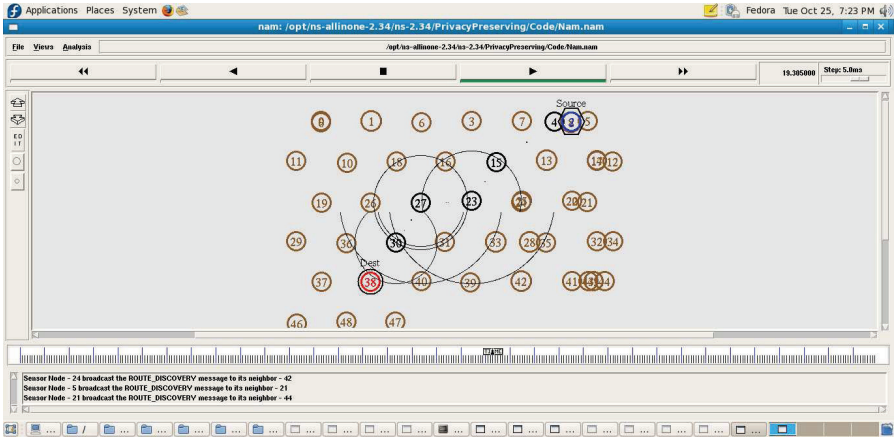



**Figure 6.4 Packet sending Source Node to Destination Node**

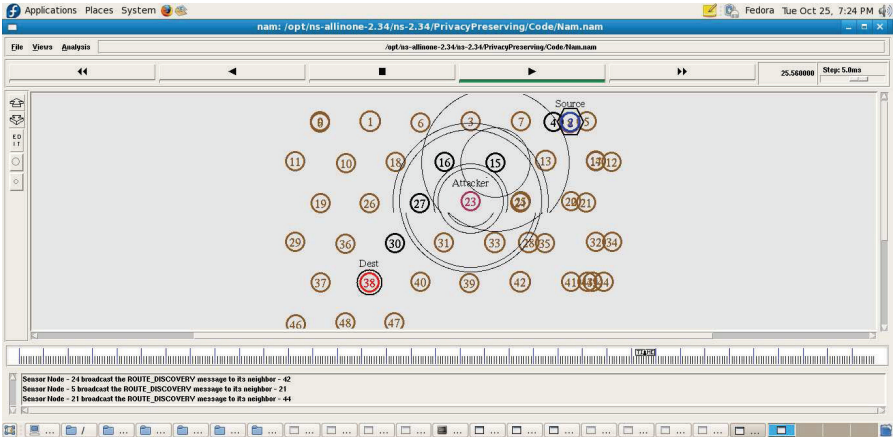



**Figure 6.5 Attacker Generation process**

While sending the packet through the suggested path we initiate the attacker node that attacks the packet which does not makes to reach the packet to the desired location which shown in the above.

After finding the attacker in the path, the interfaces choose for an alternative path to transfer the packet from one location to another. Attacker node represented as different color to differentiate from other nodes which is in the below figure.



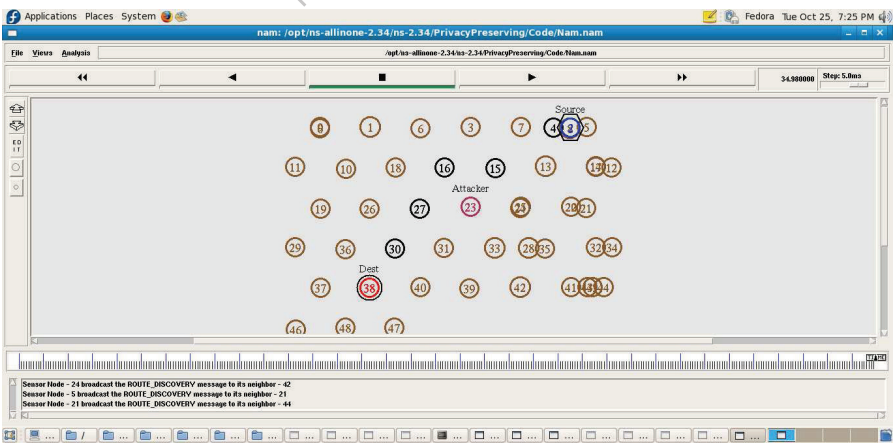



**Figure 6.6 Alternative path selection Process**

Here we use the dynamic network topology development where the nodes are arranged in the dissimilar fashion and it also identify the neighbor nodes to transfer the packet from source root node to a destination node

**Figure 6.7 Dynamic Network Topology Developments**



Packets transfer from source node to destination node in dynamic non-linear allocation method is shown in the below fig 6.8.

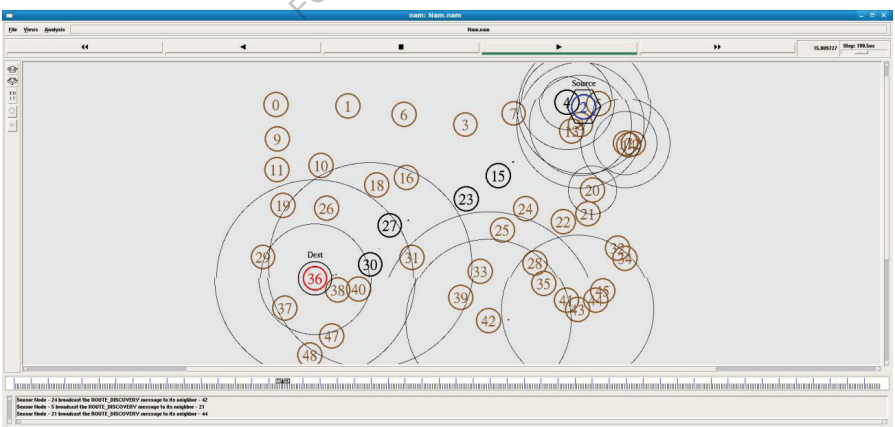



**Figure 6.8 Packet Sending Source to Destination**

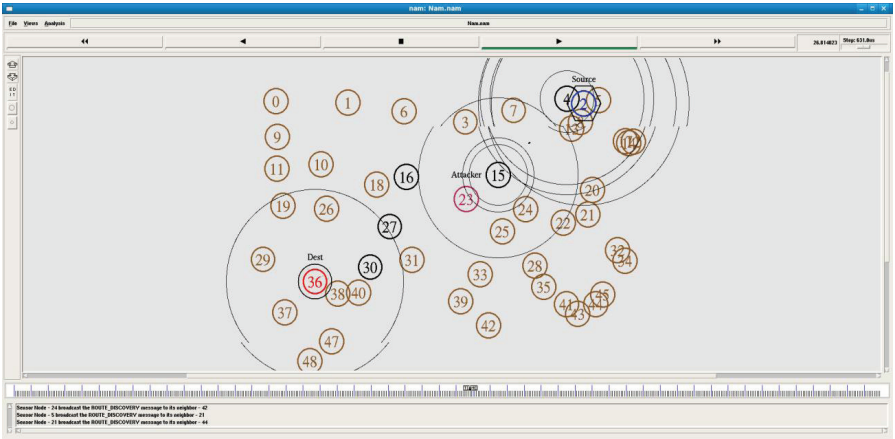



**Figure 6.9 Alternate Path selection using Dynamic Non-linear authenticator Protocol**

In the above figure 6.9, alternate path selection using dynamic non-linear authenticator protocol fools the attacker in intermediate path through that we can send a packet by choosing the different path without knowledge of the attacker.

## PERFORMANCE EVALUATION

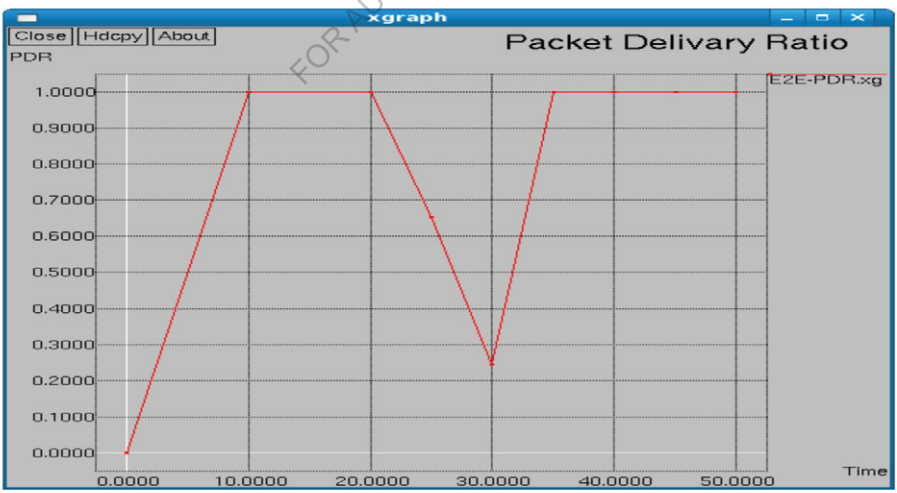
In the Proposed System nearly 49 nodes are taken for simulation process.

**Packet delivery Ratio (PDR)**: the ratio of the data packets delivered to the destinations to those generated by the Constant Bit Rate (CBR) sources. The PDR shows how successful a protocol performs delivering packets from source to destination. The higher for the value give use the better results. This metric characterizes both the completeness and correctness of the routing protocol also reliability of routing protocol by giving its effectiveness.

**PDR** is the ratio of the number of data packets received by the destination node to the number of data packets sent by the source mobile node. It can be evaluated in terms of percentage (%). This parameter is also called "success rate of the protocols", and is described as follows:



$$PDR = \frac{number\ of\ Packet\ sent}{number\ of\ packet\ Receive} \times 100$$



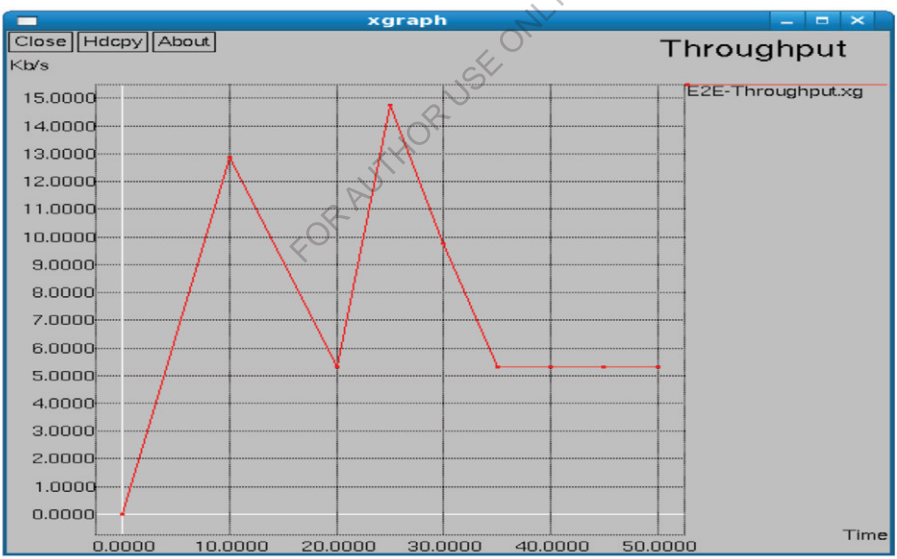

**Packet Delivery Ratio**

**Throughput**: The ratio of the total amount of data that reaches a receiver from a sender to the time it takes for the receiver to get the last packet is referred to as throughput. It is expressed in bits per second or packets per second. Factors that affect throughput include frequent topology changes, unreliable communication, limited bandwidth and limited energy. A high throughput network is desirable.

**Throughput** is the average rate of successful message delivery over a communication channel. This data may be delivered over a physical or logical link, or pass through a certain network node.

$$X = \frac{C}{T}$$

Where X is the throughput, C is the number of requests that are accomplished by the system, and T denotes the total time of system observation.

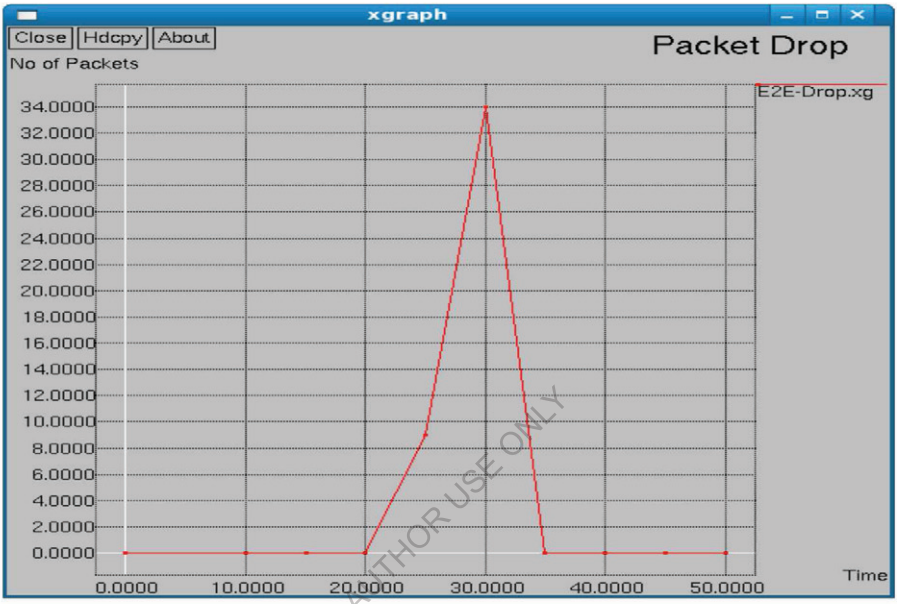




**Throughput**

**Data Packet Loss (Packet Loss):** Mobility-related packet loss may occur at both the network layer and the MAC layer. Here packet loss concentrates for network layer. When a packet arrives at the network layer. The routing protocol forwards the packet if a valid route

to the destination is known. Otherwise, the packet is buffered until a route is available. A packet is dropped in two cases: the buffer is full when the packet needs to be buffered and the time that the packet has been buffered exceeds the limit.





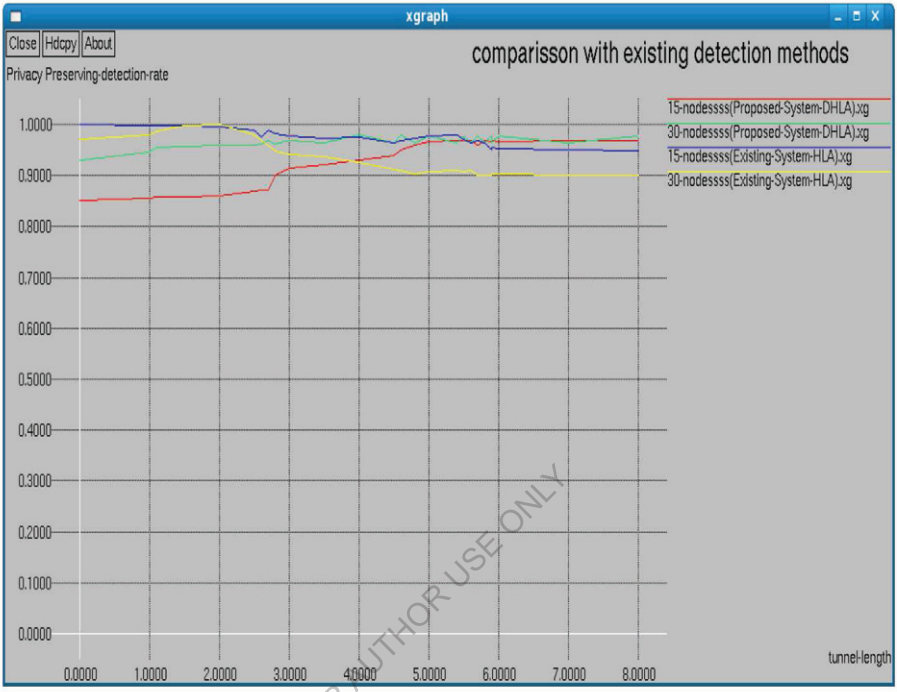

**Packet Drop**

**Comparison Ratio** Average end-to-end delay comparisons of existing methods signify how long it will take a packet to travel from source to destination node. It includes delays due to route discovery, queuing, propagation delay and transfer time.

$$D_{end-end} = N(d_{trans} + d_{prop} + d_{proc} + d_{queue})$$

Where $D_{end\text{-}end}$= end-to-end delay, $d_{trans}$= transmission delay, $d_{prop}$= propagation delay, $d_{proc}$ = processing delay, $d_{queue}$= Queuing delay and $N$= number of links.

This metric is useful in understanding the delay caused while discovering path from source to destination.






**Comparison with Existing and Proposed System**

# CONCLUSION

The research presents a new approach which combines techniques from various fields and adapts to solve the problem of packet sending, path selection and privacy preserving. The results show that the proposals generated the extremely relevant. It has been observed that as the packet delivery ratio, packet drop size and Throughput prevents the quality of proposed system.

The goal of this research is to show that compared with conventional detection algorithms that utilize only the distribution of the number of lost packets, exploiting the correlation between lost packets significantly improves the accuracy in detecting malicious packet drops. Such improvement is especially visible when the number of maliciously dropped packets is comparable with those caused by path errors. To correctly calculate the correlation between lost packets, it is critical to acquire truthful packet-loss information at individual nodes. The research work developed a Dynamic Non-linear authenticator Protocol algorithm (DNAP) based routing privacy preserving protocol auditing architecture that ensures truthful packet-loss reporting by individual nodes. This architecture is collusion proof, requires relatively high computational capacity at the source node, but incurs low communication and storage overheads over the route. To reduce the computation overhead of the baseline construction, a packet-block-based mechanism was also proposed, which allows one to trade detection accuracy for lower computation complexity.



Some of future woks can include:

- A malicious packet dropping detection technique that effectively detects the packet dropping attack in any environment while keeping the generated overheads minimal will be our focus.
- To evaluate the different topological changes in the network. The impact of dynamic topology remains an issue to be evaluated.

## REFERENCES

[1] J. N. Arauz, “802.11 Markov channel modeling,” Ph.D. dissertation, School Inform. Sci., Univ. Pittsburgh, Pittsburgh, PA, USA, 2004.

[2] C. Ateniese, R. Burns, R. Curtmola, J. Herring, L. Kissner, Z. Peterson, and D. Song, “Provable data possession at untrusted stores,” in Proc. ACM Conf. Comput. and Commun. Secur., Oct. 2007, pp. 598–610.

[3] G. Ateniese, S. Kamara, and J. Katz, “Proofs of storage from homomorphic identification protocols,” in Proc. Int. Conf. Theory Appl. Cryptol. Inf. Security, 2009, pp. 319–333.

[4] B. Awerbuch, R. Curtmola, D. Holmer, C. Nita-Rotaru, and H. Rubens, “ODSBR: An on-demand secure byzantine resilient routing protocol for wireless ad hoc networks,” ACM Trans. Inform. Syst. Security, vol. 10, no. 4, pp. 1–35, 2008.



[5] B. Awerbuch, R. Curtmola, D. Holmer, C. Nita-Rotaru, and H. Rubens, “ODSBR: An on-demand secure byzantine resilient routing protocol for wireless ad hoc networks,” ACM Trans. Inf. Syst. Secur., vol. 10, no. 4, pp. 11–35, 2008.

[6] K. Balakrishnan, J. Deng, and P. K. Varshney, “TWOACK: Preventing selfishness in mobile ad hoc networks,” in Proc. IEEE Wireless Commun. Netw. Conf., 2005, pp. 2137–2142.

[7] D. Boneh, B. Lynn, and H. Shacham, “Short signatures from the weil pairing,” J. Cryptol., vol. 17, no. 4, pp. 297–319, Sep. 2004.

[8] S. Buchegger and J. Y. L. Boudec, "Performance analysis of the confidant protocol (cooperation of nodes: Fairness in dynamic adhoc networks)," in Proc. 3rd ACM Int. Symp. Mobile Ad Hoc Netw. Comput. Conf., 2002, pp. 226–236.

[9] L. Buttyan and J. P. Hubaux, "Stimulating cooperation in selforganizing mobile ad hoc networks," ACM/Kluwer Mobile Netw.Appl., vol. 8, no. 5, pp. 579–592, Oct. 2003.

[10] J. Crowcroft, R. Gibbens, F. Kelly, and S. Ostring, "Modelling incentives for collaboration in mobile ad hoc networks," presented at the First Workshop Modeling Optimization Mobile, Ad Hoc Wireless Netw., Sophia Antipolis, France, 2003.

[11] J. Eriksson, M. Faloutsos, and S. Krishnamurthy, "Routing amidcolluding attackers," in Proc. IEEE Int. Conf. Netw. Protocols, 2007, pp. 184–193.

[12] W. Galuba, P. Papadimitratos, M. Poturalski, K. Aberer, Z.Despotovic, and W. Kellerer, "Castor: Scalable secure routing for ad hoc networks," in Proc. IEEE INFOCOM, Mar. 2010, pp. 1 –9.

[13] T. Hayajneh, P. Krishnamurthy, D. Tipper, and T. Kim, "Detecting malicious packet dropping in the presence of collisions and channel errors in wireless ad hoc networks," in Proc. IEEE Int. Conf. Commun., 2009, pp. 1062–1067.

[14] Q. He, D. Wu, and P. Khosla, "Sori: A secure and objective reputation-based incentive scheme for ad hoc networks," in Proc. IEEE Wireless Commun. Netw. Conf., 2004, pp. 825–830.

[15] D. B. Johnson, D. A. Maltz, and J. Broch, "DSR: The dynamic source routing protocol for multi-hop wireless ad hoc networks," in Ad Hoc Networking. Reading, MA, USA: Addison-Wesley, 2001, ch. 5, pp. 139–172.

[16] W. Kozma Jr. and L. Lazos, “Dealing with liars: Misbehavior identification via Renyi-Ulam games,” presented at the Int. ICST Conf. Security Privacy in Commun. Networks, Athens, Greece, 2009.

[17] W. Kozma Jr., and L. Lazos, “REAct: Resource-efficientaccountability for node misbehavior in ad hoc networks based on random audits,” in Proc. ACM Conf. Wireless Netw. Secur., 2009, pp. 103–110.

[18] K. Liu, J. Deng, P. Varshney, and K. Balakrishnan, “An acknowledgement-based approach for the detection of routing misbehavior in MANETs,” IEEE Trans. Mobile Comput., vol. 6, no. 5, pp. 536–550, May 2006.

Content: