Real-time Intrusion Detection for Ad hoc Networks

Ioanna Stamouli

A dissertation submitted to the University of Dublin, in partial fulfilment of the requirements for the degree of Master of Science in Computer Science

September 12, 2003

Declaration

I declare that the work described in this dissertation is, except where otherwise stated, entirely my own work and has not been submitted as an exercise for a degree at this or any other university.

Signed: _____

Ioanna Stamouli

September 12, 2003

Permission to lend and/or copy

I agree that Trinity College Library may lend or copy this dissertation upon request.

Signed: _____
Ioanna Stamouli
September 12, 2003

Acknowledgements

I would like to thank my supervisor, Mr. Hitesh Tewari, for all his guidance and assistance throughout the duration of this project. I would also like to thank my family who put me where I am today. Thanks and appreciation goes to my classmates for their continuous support throughout the year and for making me feel like home. Finally, I would like to thank Patroklos Argyroudis for his enlightening critique in many aspects of the project and for the essential comments concerning this document.

Abstract

In the recent years, wireless technology has enjoyed a tremendous rise in popularity and usage opening new fields of applications in the domain of networking. One such field concerns mobile ad hoc networks (MANETs) where the participating nodes do not rely on any existing network infrastructure. By definition the nature of ad hoc networks is dynamically changing and they have a fully decentralised topology. Hence security is hard to achieve due to the dynamic nature of the relationships between the participating nodes as well as the vulnerabilities and limitations of the wireless transmissions medium.

The RIDAN system is a novel architecture that uses knowledge-based intrusion detection techniques to detect active attacks that an adversary can perform against the routing fabric of mobile ad hoc networks. Moreover, the system is designed to take countermeasures to minimise the effectiveness of an attack and keep the performance of the network within acceptable limits. The novelty of the system lies in the usage of timed finite state machines that enable the real-time detection of active attacks.

The RIDAN system does not introduce any changes to the underlying routing protocol and operates as an intermediate component between the network traffic and the routing protocol. The system was developed and tested to operate in AODV-enabled networks using the network simulator (ns-2). The simulator parameters that were used in the scenarios developed to evaluate the RIDAN system consider both the accuracy and the efficiency of the simulation. The system was evaluated using as main the metric the delivery ratio. Thus when the system is under the sequence number attack the delivery ratio drops to 38.3% while the RIDAN-enabled AODV increases its performance by 16.6%. When the network is under the resource consumption attack the delivery ratio of AODV drops to 42.6% and the RIDAN system improves it by 31.6%. The final implemented attack is the dropping routing packets attack and when it is performed the delivery ration decreases to 23% while the RIDAN-enabled AODV manages to keep the network performance 13.8 % higher.

Table of Contents

List of Figures

List of Tables

Chapter 1
Introduction

1.1 Background

In the recent years, wireless technology has enjoyed a tremendous rise in popularity and usage, thus opening new fields of applications in the domain of networking. One of the most important of these fields concerns mobile ad hoc networks (MANETs), where the participating nodes do not rely on any existing network infrastructure. A mobile ad hoc network is a collection of wireless nodes that can be rapidly deployed as a multi-hop packet radio network without the aid of any existing network infrastructure or centralized administration [CE89]. Therefore, the interconnections between nodes are capable of changing on continual and arbitrary basis. Nodes within each other's radio range communicate directly via wireless links, while those that are further apart use other nodes as relays.

Ad hoc networks have a wide array of military and commercial applications. They are ideal in situations where installing an infrastructure network is not possible or when the purpose of the network is too transient or even for the reason that the previous infrastructure network was destroyed.

Security in mobile ad hoc networks is a hard to achieve due to dynamically changing and fully decentralized topology as well as the vulnerabilities and limitations of wireless data transmissions. Existing solutions that are applied in wired networks can be used to obtain a certain level of security. Nonetheless, these solutions are not always be suitable to wireless networks. Therefore ad hoc networks have their own vulnerabilities that cannot be always tackled by these wired network security solutions [ACP+02].

One of the very distinct characteristics of MANETs is that all participating nodes have to be involved in the routing process. Traditional routing protocols designed for infrastructure

networks cannot be applied in ad hoc networks, thus ad hoc routing protocols were designed to satisfy the needs of infrastructureless networks. Due to the different characteristics of wired and wireless media the task of providing seamless environments for wired and wireless networks is very complicated. One of the major factors is that the wireless medium is inherently less secure than their wired counterpart. Most traditional applications do not provide user level security schemes based on the fact that physical network wiring provides some level of security [Bha94]. The routing protocol sets the upper limit to security in any packet network. If routing can be misdirected, the entire network can be paralyzed [WLB03]. This problem is enlarged in ad hoc networks since routing usually needs to rely on the trustworthiness of all nodes that are participating in the routing process. An additional difficulty is that it is hard to distinguish compromised nodes from nodes that are suffering from broken links.

1.2 Proposed Goals

The main goal of this M.Sc. thesis is to design and implement an intrusion detection component that will operate in ad hoc networks and more specifically in networks that utilise the Ad hoc On-demand Distance Vector (AODV) routing protocol [PR03]. The intrusion detection component will be able to keep the network resources operating within normal parameters while malicious nodes attempt to paralyze the network by performing active routing attacks. To achieve the main goal the following have to be achieved:

- In depth understanding of how the AODV protocol operates in route discovery and maintenance.
- Understanding of ns-2, the network simulator, where the implementation of the intrusion detection component will take place.
- Formal definition of the active routing attacks using finite state machines (FSMs).
- Implementation of selected active routing attacks and development of the intrusion detection component that identifies them.

1.3 Document Overview

The rest of this document is organised as follows. Chapter 2 gives an introduction to mobile ad hoc networks in general and describes how they differ from conventional communication networks. Problems specific to wireless communication in mobile ad hoc networks highlighted out and a detailed description of the AODV routing protocol is presented.

Chapter 3 outlines the security goals and challenges in ad hoc environments. The active attacks that an adversary can perform against the routing fabric are presented in detail. The security schemes that are currently utilised in ad hoc networks are also studied.

Chapter 4 presents the state of the art intrusion detection techniques in the field of wired networking. Moreover, in the same chapter the intrusion detection research related to ad hoc networking is outlined.

Chapter 5 the design of the RIDAN system is analysed. An important element of this study was the design and the implementation of the active attacks that an adversary can perform against the routing protocol. Thus in this chapter the design of the active attacks along with the design of the timed finite state machines is examined.

Chapter 6 the implementation details of both the attacks and the detection mechanism is discussed. In order to give a more clear view of the implementation details involved part of the code are presented as pseudocode.

Chapter 7 in this final chapter the evaluation of the RIDAN system takes place. The metrics that were used to measure the performance of the system are presented along with diagrams that illustrate the performance measurements. In this chapter some future extensions for the RIDAN system are also proposed.

Chapter 2
Ad hoc Networks

2.1 Introduction

In the recent years wireless networks have witnessed a tremendous increase of popularity in both research and industry. There are currently two variations of mobile networks. The first is widely known as *infrastructure networks* since the gateways that connect them to other networks (like the Internet) are fixed and wired. The bridges in these networks are also known as *base stations*. In an environment like this, a node is able to roam freely and establish a connection link with the nearest base station that is within its communication range. As the mobile node moves out of the range of the base station that it was connected with, it falls into the range of another and a *hand off* occurs between the old base station and the current one, enabling the mobile unit to continue communication seamlessly through the network [RT99]. These types of networks are most widely applied in office areas and include the wireless local area networks (WLANs).

The second type of wireless networks is the *infrastructureless mobile network* that is also known as an *ad hoc network*. Infrastructureless mobile networks have no fixed routers and base stations and the participating nodes are capable of movement. Due to the limited transmission range, multiple hops may be required for nodes to communicate across the ad hoc network. Routing functionality is incorporated into each host, thus ad hoc networks can be characterized as having dynamic, multi-hop, and constantly changing topologies [PHO02]. Example scenarios for the application of ad hoc networks include search and rescue operations, meetings or conventions in which persons wish to quickly share information and data acquisition operations in inhospitable environments.

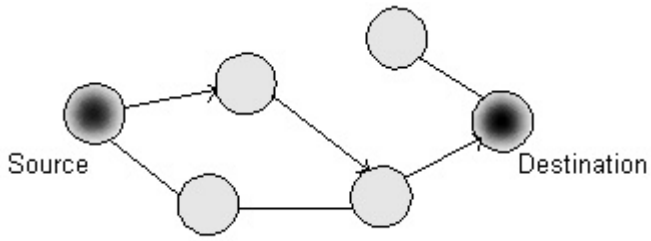

Figure 2.1: Ad hoc network example.

Figure 2.1 illustrates an example ad hoc network. The participating nodes act both as end hosts and routers forwarding traffic from the source to the destination host. In the following section the properties and differences of ad hoc networks are discussed in more detail.

2.2 Properties of Ad hoc Networks

As already mentioned, ad hoc networks do not rely on pre-existing infrastructure and this may be their most distinguishing attribute. Instead ad hoc networks are formed by individual nodes when they come to close proximity and need to communicate with each other. This implies that there is no need for stationary components such as routers, bridges and cables and of course central administration is not required.

Due to the lack of stationary infrastructure, the participating nodes in the ad hoc network have to forward traffic on behalf of other nodes that are not in close proximity to the destination node. If they deny participating in the routing process, the connectivity between nodes may be lost and the network could be segmented. Therefore, the functionality of an ad hoc network heavily depends on the forwarding behaviour of the participating nodes.

Ad hoc networks can be characterized as autonomous in the sense that most commonly they offer connectivity between the participating nodes and not connectivity to external LANs or internets. However in theory it is possible that some of the ad hoc nodes are multi-homed with connections in both the ad hoc network and one or more external networks. Nothing prevents these nodes from acting as gateways between these networks but is not a common element.

Another very important property of ad hoc networks is their dynamic topology. Since the topology arbitrarily changes due to node mobility and changes of the surrounding environment, the utilised routing protocols have to be able to adapt to the dynamic topology. Traditional wired routing protocols like OSPF [Moy98] do not incorporate in their normal operation support for frequent network topology changes. Thus, the routing protocols that are currently utilised in ad hoc environments have specifically been designed to handle node mobility and rapidly changing topologies.

The devices that are usually employed in the ad hoc networks have their own limitations. Since, the only hardware component that is required to connect a device in an ad hoc network is a wireless interface, PDAs and mobile telephones can be utilised. Furthermore, differences in the radio transmission ranges and reception equipment sensitivities may lead to unidirectional links which could complicate routing in the ad hoc networks. Apart from the communication differences between the nodes, ad hoc networks suffer from limited hardware resources like limited battery, constrained CPUs and small memory capacity.

2.3 Comparison with Wired Networks

Ad hoc networks when compared to traditional wired networks have significant differences as far as the hardware infrastructure, addressing, naming and most importantly routing. These differences are presented in more detail in the following sections in order to give a more clearer view of the limitations and strengths of the ad hoc networks.

2.3.1 Infrastructure

A conventional network most often consists of a fixed infrastructure that is built up of computer nodes, routers, switches, bridges, base stations, gateways and other network devices that are all connected with wires. One of the main characteristics of these networks is that their topology is fixed. When there is a need to reconfigure the network or add more network devices there has to be physical manual intervention. During the reconfiguration of the network most often there is loss of services in either some nodes or the entire network while the changes are carried out. Moreover, traditional wired networks usually have centralised administration, since many of the nodes rely on central servers for storage, access and processing of data.

Wireless networks and more specifically ad hoc networks follow a different paradigm that resolves some of these issues. Due to the use of IEEE 802.11b as a transmission medium [Iee97] that is utilised in wireless networks, problems of cabling reconfiguration and possible service unavailability caused by topology changes are avoided. Ad hoc networks add to this ability by allowing the formation of networks on-the-fly without the need of any existing infrastructure. The results is an on-demand network that has all the advantages of the wireless network combined with a virtually easy to setup system, in contrast with the conventional wired networks where their establishment requires tedious administrative tasks. Furthermore, since ad hoc networks are formed without any external aid but from the participating nodes themselves, and the topology of the network may change arbitrarily, centralised solutions in administration and in data sharing are not frequently used as in conventional wired networks.

2.3.2 Addressing

In traditional networks, the address distribution is performed either manually, meaning that the network administrator handles the assignment of the addresses himself, or this process is carried out by using special protocols, such as DHCP (Dynamic Host Configuration Protocol) [Dro97]. The DHCP servers as well as other address allocation entities use the hardware address of the network interface to assign addresses and guarantee their uniqueness.

However, in ad hoc networks due to the fact that there is no central authority responsible for assigning IP addresses this issue becomes more complicated. Therefore, there is no guarantee that the address that was taken or somehow assigned to the node reflects either the nodes geographical location due to mobility or that it is unique.

2.3.3 Routing

The operation of routing in traditional wired networks is performed by special dedicated equipment that can either be hardware-based devices specialised for this task or computer nodes equipped with several network interfaces and adequate software to perform the actual routing. In ad hoc networks a node does not need to be equipped with several network interfaces, since all communication is usually done through a single wireless channel which is broadcast in nature. Additionally, in contrast to wired networks, in ad hoc networks all the participating nodes have to contribute in the process of routing. All the nodes in the ad hoc network should be capable of forwarding network traffic on behalf of other nodes. The fact that all nodes participate in the routing process enables the network traffic to flow through the ad hoc network over multiple hops.

An important property of ad hoc routing is that a flat addressing scheme is used. In ad hoc routing no subnetting assumptions are made, thus the routing tables of the participating nodes may end up consisting of separate IP addresses with no correlation to each other. On the other hand, in traditional networks routing tables most often contain network prefixes along with the appropriate network interface identifiers and not specific addresses.

2.4 Ad hoc Routing Protocols

In ad hoc networking environments an application packet from a specific node may have to travel several hops in order to reach its destination. The main function of a routing protocol is to form and maintain a routing table with information relevant to which the next hop for this packet

should be in order to reach its ultimate destination. All the nodes have their own routing tables that they consult to forward the routing traffic that it is not destined for them.

Although the problem of routing is not a new one in computer networks, routing in ad hoc networks due to its unique requirements cannot be successfully handled by utilising existing routing schemes such as traditional link-state and distance vector routing protocols. One of the reasons that for example OSPF [Moy98] and RIP [Mal98] cannot be used in ad hoc networks is that these protocols where originally designed to operate in environments with relatively static topology. However, the nature of the ad hoc networks allows the participating nodes to move freely in and out of the network. Another issue that contributes to the fact that the available routing protocols cannot operate in ad hoc mode is that they were designed with the assumption that all the links are bidirectional. In mobile ad hoc networks this is not always the case. The differences of the wireless networking hardware of the nodes or the radio signal fluctuations may result in some links becoming unidirectional. Finally, both OSPF and RIP attempt to maintain routes to all the reachable destinations, but in ad hoc networks with high density this may lead in having very large numbers of routing entries imposing performance overhead. Therefore, there is a need for special routing protocols that will be able to cope with the unique attributes and limitations of mobile wireless ad hoc networks.

2.4.1 Properties of Ad hoc Routing Protocols

As it is clear from the previous analysis, there is a special need for routing protocols specifically designed to address the requirements of ad hoc networking. Some of the properties that ad hoc routing protocols should posses are suggested in [CM99] and are analysed below:

- *Distributed operation*: One of the most essential properties due to the decentralised nature of ad hoc networks.
- Loop-freedom: Although it is not strictly implied that a protocol has to provide loop-freedom it is generally a desirable attribute as it usually leads to better overall performance.
- *On-demand operation*: The routing protocol instead of maintaining routing table entries for all the possible destinations it should rather find routes as they are needed in order to conserve both energy and bandwidth.
- *Proactive operation*: It is the opposite of the “on-demand” operation. When the reactive, on-demand behaviour produces unacceptable overhead in searching for routes a

proactive operation is desirable. The proactive and the on-demand operations are analysed in depth in a following section.

- *Security*: It is fundamental that the routing protocol must provide security features that prohibit the disruption or modification of network traffic. However, as discussed in a following section the security that is provided from the currently employed protocols is not adequate.
- *Sleep*: Due to the energy constants of the participating devices of the ad hoc network it is required that the nodes have a sleep period resulting in energy conservation. The routing protocol should be able to accommodate such sleep periods without overly adverse consequences.
- *Unidirectional link support*: In ad hoc networks unidirectional links can occur. The routing protocol should be able to use separate unidirectional links in both direction to replace a bidirectional link.

2.4.2 Table-driven Ad hoc Routing Protocols

The table-driven operation, also known as proactive, requires that a node maintains a routing table that contains routing information regarding the connectivity to all other nodes that participate in the ad hoc network. Any changes in the topology are periodically propagated by means of updates throughout to the entire network to ensure that all nodes share a consistent view of the network [RT99]. However, proactive behaviour suffers from the disadvantage of additional control traffic that is required to continually update old route entries. Due to the mobility of the nodes the routes are likely to be broken frequently, thus two mobile nodes that had established a link between them will no longer be able to support that link and subsequently any other routes that were depended on that link [Per01]. If the broken route has to be repaired even though no applications are using it the effort can be considered wasted. As an example of two protocols that follow the table-driven design approach we briefly present the Destination-Sequence Distance-Vector (DSDV) [PB94] and the Optimised Link State Routing (OLSR) protocol [CJL+01].

2.4.2.1 Destination-Sequenced Distance-Vector (DSDV)

DSDV is a proactive protocol based on the Bellman-Ford algorithm [Per00]. The problem count-to-infinity of the Bellman-Ford algorithm is avoided by the use of sequence numbers that enables the nodes to distinguish between stale and fresh routes and ensure loop freedom.

Since DSDV is a table-driven protocol, each node maintains a routing table with all the destinations listed accompanied with information about the next hope and the number of required hops to reach each destination. In order to update the routing tables DSDV uses two different types of update packets, namely the full dump and the incremental update packets. The full dump update packet contains all routing information available at a node. These packets are transmitted infrequently and only if the node experiences occasional movement since they produce considerable routing overhead. The incremental update packet contains only the information that has changed since the last full dump packet transmission. Thus, incremental packets are more frequent and consume only a small fraction of the network resources [RT99]. Therefore, the DSDV protocol is probably desirable in ad hoc networks where the node mobility is low to moderate.

2.4.2.2 Optimised Link State Routing (OLSR)

The OLSR protocol is a link-state proactive protocol that is based on the Open Shortest Path First (OSPF) protocol. OLSR manages to disseminate routing information through an efficient flooding technique. The key concept in this protocol is the multipoint relays (MPRs) [CJL+01]. A node's multipoint relay is a subset of its neighbours whose combined radio range covers all nodes two hops away. In order for a node to determine its minimum multipoint relay set based on its two-hop topology, periodic broadcasts are required.

Similar to conventional link-state protocols the link information updates are propagated throughout the network. However, in OLSR when a node has to forward a link update it only forwards it to its MPR set of nodes. This behaviour is inspired from the Zone Routing Protocol (ZRP) and its property of border-casting with one hop radius [Has97]. Finally, the distribution of topological information is realised with the use of periodic topology control messages and has as a result each node knowing a partial graph of the topology of the network that is further used to calculate the optimal routes [CJL+01]. OLSR is mostly preferred when the ad hoc network consists of a large number of nodes and has a high density. One of the main advantages of the OLSR protocol is that it does not make any assumptions concerning the underlying link layer, allowing it to be used in a variety of configurations.

2.4.3 On-demand Ad hoc Routing Protocols

An alternative approach to the one proposed by table-driven protocols is the on-demand approach, also known as a reactive behaviour. These types of routing protocols create routes only

when they are required by the source node. Therefore, when a route is desired a route discovery process is initiated by the source node within the network [RT99]. The application packets transmitted while the route discovery process is in progress are buffered and sent after the route has been established. Once the route path has been established, the route maintenance process is not triggered until either the destination becomes inaccessible or until the route is no longer required. The Ad hoc On-demand Distance Vector (AODV) protocol [PB03] and the Dynamic Source Routing (DSR) protocol [JMHJ02] are presented as examples of the on-demand design behaviour.

2.4.3.1 Ad hoc On-demand Distance Vector (AODV)

The AODV routing protocol builds on top of the DSDV protocol that was previously described. AODV is an improvement of DSDV as it minimises the number of required broadcasts since it creates routes in an on-demand basis, in contrast to DSDV which maintains a complete set of routes [RT99]. It utilises destination sequence numbers to ensure loop-freedom at all times and to avoid the count-to-infinity problem associated with classical distance-vector protocols.

When a node needs a route to a destination it broadcasts a Route Request (RREQ) message. The RREQ message is spread throughout the network and as soon as the message reaches a node with a fresh enough route to the specific destination or the destination node itself, a Route Reply (RREP) message is unicasted back to the requesting node [PR03]. Generally AODV offers low overhead, quick adaptation to dynamic link conditions and low processing and memory overhead. Since the AODV routing protocol is the one that it used in this research and in the development of the Real-Time Intrusion Detection system it is presented in great detail in a following section.

2.4.3.2 Dynamic Source Routing (DSR)

DSR is a reactive ad hoc routing protocol based on a method known as source routing [JMHJ02]. Source routing means that each packet contains in its header an ordered list of addresses through which the packets should pass on their way to the destination. The source route is initially created by the node that originated the route discovery packet. The utilisation of source routing enables a trivial loop-free routing, and it also avoids the need of keeping up-to-date routing information like sequence numbers in intermediate nodes. DSR operates in *promiscuous* mode, meaning that all participating nodes can overhear packets containing source routes to cache this information for their own future use.

Apart from the route discovery process, DSR uses a route maintenance mechanism that employs the confirmations that nodes generate when they can verify that the next node has

successfully received a packet. If a node fails to verify the successful reception of a packet it tries to retransmit it. DSR in general is an attractive protocol for many network configurations due to its route caching features. However, the source routing approach comes at a cost of increased overhead since each packet must carry the complete path to its destination in its packet header.

2.4.5 AODV Operational Details

In this section the operational details of the AODV protocol are presented. We believe that this section is essential since the proposed research utilises AODV-enabled ad hoc networks. AODV is designed specifically to address the routing problems in ad hoc wireless networks and provides communication between mobile nodes with minimal control overhead and minimal route acquisition latency [Per01]. AODV being a reactive protocol does not require the maintenance of routes to destinations that are not in active communication; instead it allows the mobile nodes to obtain routes quickly to new destinations. Moreover, AODV enables mobile nodes to respond to link breakages and changes in the network topology in a timely manner [PR03]. As was highlighted earlier loop-freedom is a desirable property in ad hoc routing protocols. The operation of AODV is loop-free; it also avoids the Bellman-Ford count-to-infinity problem, and provides quick convergence when the network topology changes. In the following sections properties of AODV are presented along with the operational details of its most fundamental functionalities, namely the route discovery and the route maintenance processes.

2.4.5.1 Properties

As it was mentioned earlier AODV provides loop-freedom that is accomplished through the use of sequence numbers. Every node maintains its own sequence number that it increases monotonically each time it learns of a change in the topology of its neighbourhood. This sequence number ensures that the most recent route is selected whenever a route discovery process is executed. In addition, in multicast-enabled AODV each multicast group has its own sequence number, which is maintained by the multicast group leader [Per01].

Furthermore, AODV is able to provide unicast, multicast, and broadcast communication ability. This capability of having all three communication forms in a single protocol offers numerous advantages. When searching by using the multicast route discovery it increases the unicast routing knowledge and vice versa. In mobile environments any reduction in control overheads has a significant advantage. Additionally, having all three communication forms in a single protocol simplifies the implementation process of the protocol.

Route tables are used in AODV to store applicable routing information. AODV utilises both a route table for unicast routes and a multicast route table for multicast routes. The unicast route table includes information about the destination, the next-hop IP address and its sequence number. For each destination a node maintains a list of precursor nodes, which route through it in order to reach the destination [Per01]. This list is maintained for the purpose of route maintenance in case of a link breakage. Additionally, a lifetime is associated with each route table entry which is updated whenever the route is successfully used. When an entry's lifetime attribute expires because it was not frequently used it is removed form the routing table and if there is a need for this route again it is reacquired through a route discovery process.

AODV is able to maintain both unicast and multicast routes even for nodes with mobility. Also it provides a quick detection mechanism of invalid routes through the use of route errors (RERR) messages. The protocol is able to respond to topological changes that affect the active routes in a quick and timely manner. Finally, because it does not use source routing it does not introduce additional overhead since it requires only the next-hop routing information.

2.4.5.2 Route Discovery

When a node desires to communicate with some destination node, it checks if the route to this destination is available and valid in its routing table. In the case that the route is available and valid the communication is feasible right away, but if the route is either unavailable or it has expired a route discovery process has to be initiated. In order to initiate a route discovery process the source node has to send a RREQ packet. The fields of the route request packet are illustrated in figure 2.2. After creating the RREQ packet the node sets a timer and waits for a route reply (RREP) message [PR03].

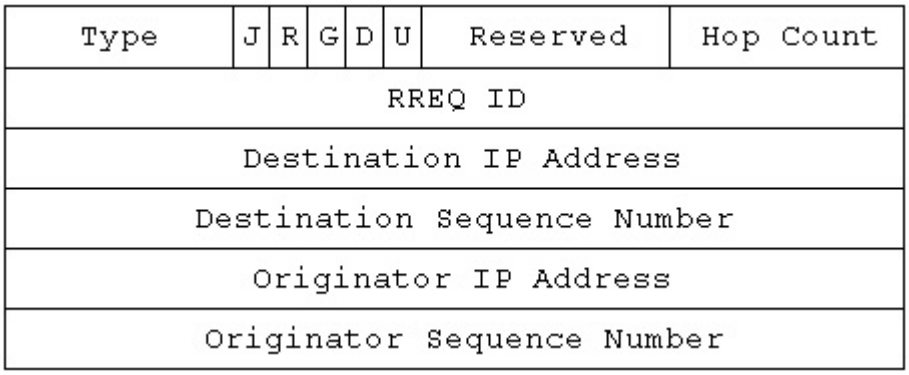

Figure 2.2: The format of Route Request packet.

An intermediate node upon the reception of a RREQ packet checks whether it has seen it before by examining the originator's IP address and the RREQ broadcast ID pair. Each node maintains a list of the originator IP and RREQ broadcast ID pair for each route request that it receives. This information remains in this list for a finite period of time and it is used to avoid flooding attacks or anomalous node behaviour. If the intermediate node has already seen this RREQ it silently discards the packet.

If it has not seen this RREQ within this finite period of time it starts processing it. The first step is to set up the reverse route in its routing table. The reverse route contains the originator IP address, the sequence number, the hops required to reach the source node and the neighbour from which it has received the packet. This process is essential since it is used to forward back the RREP. Figure 3.2 indicates the propagation process of a RREQ along with the formation of the relevant reverse routes.

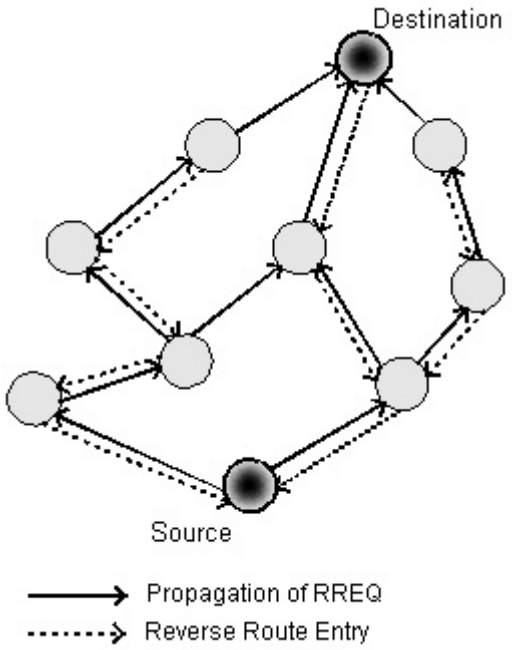

Figure 2.3: Propagation of an AODV RREQ and establishment of the reverse routes.

In order for an intermediate node to reply to a RREQ it has to have an unexpired entry for the destination in its routing table. Additionally, the sequence number associated with that destination must be greater or equal to the one indicated in the RREQ packet. If the entry satisfies these two conditions then it unicasts a RREP back to the source of the RREQ by incrementing the hop count by one. The structure of the RREP and the fields it contains are presented in figure 2.4 [PR03].

Type	R	A	Reserved	Prefix Sz	Hop Count
Destination IP Address					
Destination Sequence Number					
Originator IP Address					
Lifetime					

Figure 2.4: Format of a Route Reply (RREP) packet.

If none of the intermediate nodes is able to reply, the RREQ eventually reaches the destination node. When the destination node sends the RREP it places its current sequence number in the packet, initialises the hop count to zero and places the length of time this route is valid in the RREP's *Lifetime* field [PR03]. If this is the first time the source node communicates with this node the sequence number will not be available and therefore it will not be included in the packet. When an intermediate node receives the RREP it uses the reverse route established for the RREQ to forward the packet to each destination, but before doing so it increments the hop count by one. Figure 2.5 indicates the path of a RREP from the destination to the source node.

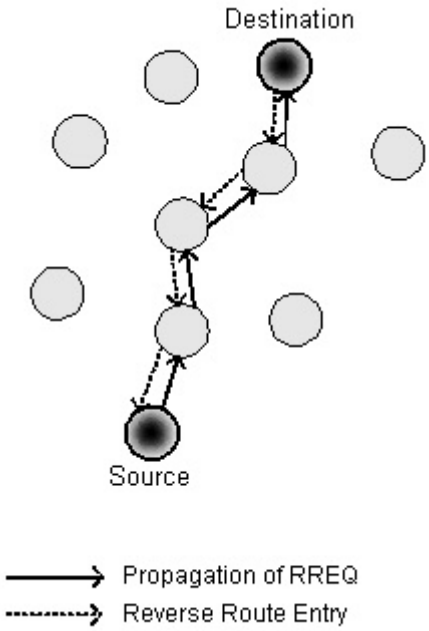

Figure 2.5: Propagation of a RREP message from the destination to the source node.

It is possible that the destination node will receive more than one RREP from its neighbours. In this case it uses the first RREP that it receives and upon the reception of another reply it checks if the later packet contains a greater destination sequence number or if it has a smaller hop count,

meaning that it provides a fresher or sorter route. In this case it updates the route entry with the new values; otherwise the reply packet is discarded.

2.4.5.3 Route Maintenance

Once the route between the source and the destination nodes is established it is maintained for the source node as long as it remains active. If the source node moves during an active session, it can simply reinitiate a route discovery process and establish a new route to the destination and continue communication. However, if either the destination or an intermediate node moves a RERR packet is sent to the source affected nodes. The RERR packet header fields are illustrated in figure 2.6 [PR03].

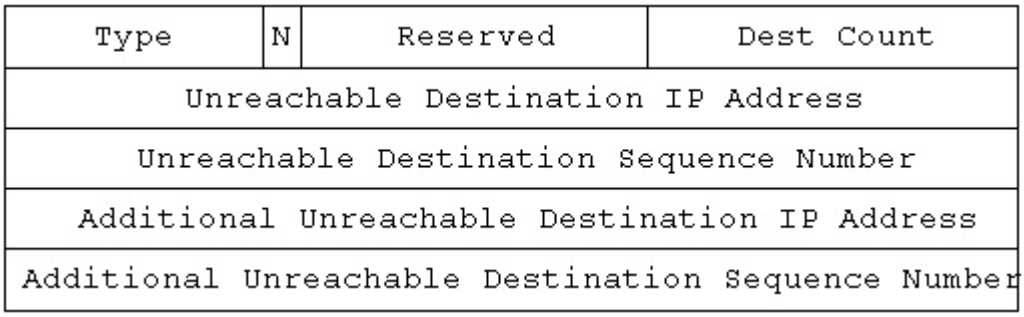

Figure 2.6: The format of the Route Error (RERR) message.

The RERR message is initiated by the node upstream of the link failure which is closer to the source. If the node upstream of the break has listed more that one nodes as a precursor node for the destination, it broadcasts the REER to these neighbours. When the neighbour nodes receive the RERR packet they mark the route to the destination as invalid by setting the distance to this destination node to infinity, and if they have any precursor list of their own they propagate this message forward to their precursor nodes. When the RERR reaches the source node it can reinitiate a route discovery if the route is still needed.

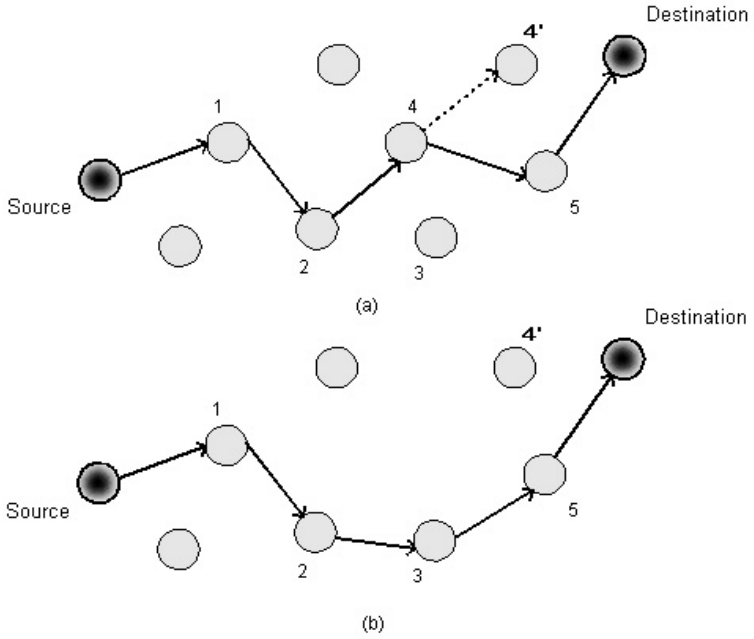

Figure 2.7: Route maintenance.

In figure 2.7 the route maintenance procedure is illustrated. In figure 2.7(a) the route from source to destination contains the nodes 1, 2, 4, and 5. When node 4 decides to move to position 4` breaks the connectivity in node 2. Node 3 being the closest upstream neighbour to the link loss sends a RERR to node 1. Node 1 upon reception of the REER packet marks the route as invalid and then forwards the RERR to the source node that reinitiates a route discovery process since it still requires communication with the destination node. The new route that was created is presented in figure 2.7(b) where node 4 was replaced by node 3.

RERRs are also sent when a node receives data packets for a destination that is not listed in its routing table [PR03]. In this way the node without the route that is receiving the data packets can inform its upstream neighbour that it should stop sending them, thus they are not constantly discarded.

2.5 Summary

The mobile ad hoc networking paradigm poses great challenges in the general field of networking. From the example of routing protocols that where presented in this chapter it is now evident that the selection of a routing protocol for use in ad hoc networks requires careful thought. Parameters such as network size, mobility and traffic load have a great impact on the suitability of each protocol. Finally the detailed presentation of the Ad hoc On-demand Distance Vector (AODV) protocol will be essential in presenting the functionality of the RIDAN system.

Chapter 3
Security in Ad hoc Networks

3.1 Introduction

As the approach of ad hoc networking varies from traditional networking approaches, the security aspects that are valid in the conventional wired networks are not fully applicable in the context of ad hoc networks. While the basic security requirements such as confidentiality and authenticity remain, the ad hoc networking approach restricts the set of applicable security mechanisms to be used since the level of security and the performance are related to each other and must be carefully balanced.

In this chapter the security goals and challenges that the field of ad hoc networking faces are explored in more detail. Since this research in mostly concerned with active attacks performed against the routing fabric an overview of the most important active attacks is included. Additionally, some of the most important security schemes are presented in order to illustrate common approaches that are currently followed to ensure network security in infrastructureless networks.

3.2 Security Goals

Security is an important issue for ad hoc networks especially for the more security-sensitive applications used in military and critical networks. An ad hoc network can be considered secure if it holds the following attributes [ZH99, Kar00]:

- *Availability*: Ensures that the network manages to provide all services despite denial of service attacks. A denial of service attack can be launched at any layer of an ad hoc network. On the physical and media access control layer a malicious user can employ jamming in order to interfere with signals in the physical layer. On the network layer, a malicious user can disrupt the normal operation of the routing table in various ways that are presented in a following section. Lastly, on the higher layer, a malicious user can bring down high-level services such as the key management service.
- *Confidentiality*: Ensures that certain information is never disclosed to unauthorised users. This attribute is mostly desired when transmitting sensitive information such as military and tactical data. Routing information must also be confidential in some cases when the user's location must be kept secret.
- *Integrity*: Guarantees that the message that is transmitted reaches its destination without being changed or corrupted in any way. Message corruption can be caused by either a malicious attack on the network or because of radio propagation failure.
- *Authentication*: Enables a node to be sure of the identity of the peer with which it communicates. When there is no authentication scheme a node can masquerade as some other node and gain unauthorised access to resources or sensitive information.
- *Non-repudiation*: Ensures that the originator of a message cannot refuse sending this message. This attribute is useful when trying to detect isolated compromised nodes.
- *Access and usage control*: Access control ensures that access to information is controlled by the ad hoc network. Usage control ensures that the information resource is used correctly by the authorised node having the corresponding rights.

3.2 Security Challenges

The prominent features of ad hoc networks pose both challenges and opportunities in achieving the proposed security goals. The main security challenges that ad hoc networks face have been thoroughly analysed in the literature [ZH99, Kar00, Sta02].

One of the main challenges that ad hoc networking faces is related to the use of wireless links. Due to the use of wireless medium an ad hoc network is vulnerable to link attacks ranking from passive eavesdropping to active impersonation, message replay and message corruption. An adversary can easily eavesdrop network traffic by placing a wireless enabled device within the range of the ad hoc network and capture routing and application packets. By eavesdropping the malicious node can gain access to secret information and violate the confidentiality requirement. Passive attacks like eavesdropping are very hard to detect since they do not present any

significant pattern or impact in the performance of the network. Active attacks may allow a malicious node to delete or inject to the network traffic erroneous messages, modify messages and impersonate as another node, hence violating availability, integrity, authentication and non-repudiation. As opposed to passive attacks, active attacks can be detected and limited with the utilisation of various schemes.

Moreover, nodes that roam in hostile environments with relatively poor physical protection face a greater probability of being compromised. Therefore, attacks against the ad hoc network can be launched from within the network by compromised or malicious nodes. In order to be able to claim high availability in such an environment, an ad hoc network should have a distributed protection architecture with no central entities. The introduction of any central entity into a security solution could lead to a significant vulnerability since the possibility of the centralised component of the security scheme becoming compromised cannot be eliminated.

Due to the dynamic nature of an ad hoc network both its topology and membership can change arbitrarily. This fact prevents the establishment of long-lived trust relationships among the participating nodes. Unlike other wireless mobile networks, like mobile IP [IDG91], nodes in ad hoc networks may dynamically become affiliated with different administrative domains. Thus, any security solution with static configuration will not be sufficient. It is desirable for a security mechanism to adapt on the fly to these changes.

Finally, an ad hoc network is not limited to a specific number of participating nodes. Even though it has not been practically attempted, ad hoc networks theoretically can be composed of hundred or even thousands of nodes. Therefore a security mechanism in order to be able to sufficiently accomplish its tasks has to be scalable and able to handle arbitrarily large networks.

3.3 Active Routing Attacks

Unlike the passive attacks, active attacks can be detected and eventually avoided by the legitimate nodes that participate in an ad hoc network. A malicious node may perform an active attack in order to disable a service or in order to conserve energy. An active attack may either being directed to disrupt the normal operation of a specific node or target the performance of the ad hoc network as a whole. In this section the most important active attacks are presented that can be easily be performed by an internal node against the utilised ad hoc routing protocol [DBRS01, Lun00, Kar00].

- *Black Hole*: In this attack, a malicious node uses the routing protocol to advertise itself as having the shortest path to the destination node of the packet that was intercepted. This attack can be easily implemented in AODV during the routing discovery process. Upon

reception of a route request the malicious node can guarantee that its reply will be preferable from the source node by either increasing significantly the destination sequence number or by advertising a considerably shorter path. Once the forged route has been established the malicious node is able to become a member of the active route and intercept the communication packets. The outcomes of this attack can vary. The malicious node can either stop after inserting the false route information in the network and aim in creating instability and unnecessary network traffic or drop all incoming application packet for the specific destination and perform a denial-of-service attack. This attack can also be used by the malicious node as the first step to a man-in-the-middle attack.

- *Routing Table Overflow*: In a routing table overflow attack the attacker attempts to create routes to non-existing nodes. The goal is to create enough routes to prevent new routes from being created or to overwhelm the protocol implementation. Proactive routing protocols are more vulnerable to this attack, since they attempt to create and maintain routes to all possible destinations. A malicious node to implement this attack can simply send excessive route advertisements to the network. To implement this attack in order to target a reactive protocol like AODV is slightly more complicated since two nodes are required. The first node should make a legitimate request for a route and the malicious node should reply with a forged address.
- *Resource Consumption*: This attack aims in flooding the network with routing traffic in order to consume battery life from the nodes and available bandwidth from the ad hoc network. The malicious node continually requests for either existing or non-existing destinations forcing the neighbouring nodes to process and forward these packets and therefore consume batteries and network bandwidth hindering the normal operation of the network.
- *Dropping Routing Traffic*: It is essential in the ad hoc network that all nodes participate in the routing process. However, a node may act selfishly and process only routing information that are related to itself in order to conserve energy. This behaviour/attack can create network instability or even segment the network.
- Location disclosure: A location disclosure attack can reveal information related to the location of a node or the topology and structure of the network. The information gained might reveal which other nodes are adjacent to the target or the physical location of a participating node. The attack can be implemented by using a command similar to *traceroute* that exists in Unix-like systems or with the use of the *time-to-live* attribute of the routing packet and the addresses of the devices by sending ICMP error messages. In

the end, the attacker knows which nodes are situated on the route to the target node. If the locations of some of the intermediary nodes are known, one can gain information about the location of the destination node as well.

There are several other similar active attacks presented in the literature [ACP+02, HHB03] but they exploit more or less the same routing protocol vulnerabilities to achieve their goals.

3.4 Security Schemes

There are two main approaches in securing ad hoc environments currently utilised. The first is the intrusion detection approach that aims in enabling the participating nodes to detect and avoid malicious behaviour in the network without changing the underlined routing protocol or the underling infrastructure. Although the intrusion detection field and its applications are widely researched in infrastructure networks it is rather new and faces greater difficulties in the context of ad hoc networks. The second approach is secure routing that aims in designing and implementing routing protocols that have been designed from scratch to include security features. Mainly the secure protocols that have been proposed are based on existing ad hoc routing protocols like AODV and DSR but redesigned to include security features. In the following sections we briefly present the two approaches in realising security schemes that can be employed in ad hoc networking environments.

3.4.1 Intrusion Detection

Intrusion is defined as “any set of actions that attempt to compromise the integrity, confidentiality, or availability of a resource” [HLMS90]. Intrusion protection techniques works as the first line of defence. However, intrusion protection alone is not sufficient since there is no perfect security in any system, especially in the field of ad hoc networking due to its fundamental vulnerabilities. Therefore, intrusion detection can work as the second line of protection to capture audit data and perform traffic analysis to detect whether the network or a specific node is under attack [ZL00]. Once an intrusion has been detected in an early stage, measures can be taken to minimise the damages or even gather evidence to inform other legitimate nodes for the intruder and maybe launch a countermeasures to minimise the effect of the active attacks.

An intrusion detection system (IDS) can be classified as network-based or host-based according to the audit data that is used. Generally, a network-based IDS runs on a gateway of a network and captures and examines the network traffic that flows through it. Obviously this approach is not suitable for ad hoc networks since there is no central point that allows monitoring

of the whole network. A host-based IDS relies on capturing local network traffic to the specific host. This data is analysed and processed locally to the host and is used either to secure the activities of this host, or to notify another participating node for the malicious action of the node that performs the attack.

The intrusion detection techniques can be categorised into *misuse detection* and *anomaly detection* [ZL00]. The misuse detection uses patterns of well-known attacks to match and identify known intrusions. This technique can accurately and effectively detect instances of known attacks. However this technique is unable to detect newly invented attacks. In ad hoc networking due to its dynamic nature it is difficult, but not impossible, to define traffic patterns that indicate an attack. The anomaly detection technique observes activities and network traffic that significantly deviates from the established normal usage and identifies intrusions. Thus, after the normal behaviour of the network traffic has been established this technique does not require any prior knowledge of the attack, and for that reason it can detect newly invented attacks. However, this technique produces a greater percentage of false alarms since the definition of normal routing operation is difficult to de defined, especially in an ad hoc network. There are some intrusion detection systems that have been proposed for ad hoc environments [MGLB00, BA01, PW02] and are presented in more detail in the following chapter.

3.4.2 Secure Routing

This approach attempts to design secure routing protocols for ad hoc networks. These protocols are either completely new stand-alone protocols, or in some cases incorporations of security mechanisms into existing protocols like AODV and DSR. Generally the existing secure routing protocols that have been proposed can be broadly classified into two categories, those that use hash chains, and those that in order to operate require predefined trust relationships.

The Secure Efficient Ad hoc Distance vector routing protocol (SEAD) [HJP02] employs the use of hash chains to authenticate hop counts and sequence numbers. SEAD is based on the design of the proactive ad hoc routing protocol DSDV. The SEAD protocol has as minimum requirement the utilisation of a clock synchronisation mechanism or the establishment of a shared secret between each pair of nodes. It provides loop freedom and protects the nodes from impersonation and several other attacks. Another secure routing protocol is Ariadne [HPJ02]. Unlike SEAD, Ariadne is based on a reactive protocol, namely DSR, and it follows an end-to-end approach for building a security mechanism. Ariadne assumes the existence of a shared secret key between two nodes and uses a message authentication code (MAC) in order to authenticate point-to-point messages between nodes [HPJ02]. An additional routing protocol that utilises hash

chains to provide security features is the Secure Ad hoc On-demand Distance Vector (SAODV) [ZA02]. SAODV proposes a set of extensions that secure the AODV routing packets. For authenticating the non-mutable fields it uses cryptographic signatures, while one-way hash chains are used for securing every different route discovery process. In order to carry out the asymmetric cryptography it requires the existence of a key management mechanism.

The Authenticated Routing for Ad hoc Networks (ARAN) protocol [DLRS01], falls into the second category of protocols that require predefined trust relationships. ARAN is a stand-alone protocol that utilises cryptographic public-key certificates in order to achieve the security goals of authentication and non-repudiation. The protocol assumes that each node knows *a priori* the public key of the certification authority that will be used to authenticate the other participating nodes. Another protocol is the Security-aware Ad hoc Routing (SAR) [YNK01] that extends on-demand ad hoc routing protocols like AODV and DSR. The main aspect of SAR is that it introduces a new security metric in the route discovery and maintenance process, treating secure routing as a quality of service (QoS) issue. SAR uses security attributes such as trust values and trust relationships in order to define this metric. Its operation is applicable in situations where a route that satisfies certain security requirements is more important and therefore preferable than any other route that satisfies other requirements (i.e. shortest path). The final secure routing protocol to be presented is the Secure Routing Protocol (SRP) [PH02]. SRP is a set of security extensions that can be used in any protocol that uses broadcasting and route queuing methods although the authors suggest that DSR is a particularly appropriate choice. The operation of SRP requires the existence of a security association between the source node that engages the route discovery process and the destination node. Upon the establishment of the security association the nodes share a secret key that is further used by the protocol.

3.5 Summary

In this chapter we presented the security goals and challenges that the field of ad hoc networking faces. The attacks that an adversary can perform against the fabric of the routing protocol are many and relatively easy to implement. The application of intrusion detection techniques is promising but there are not many applications utilising these techniques. Some of the secure protocols that were briefly discussed in this chapter require some kind of trust relationship to be pre-established between the participating nodes. However, this requirement in some cases is unrealistic since the membership in ad hoc networks arbitrarily changes and most often one cannot explicitly know before-hand which nodes are going to participate. Additionally, in hostile ad hoc environments, like battle-fields, mobile nodes are extremely vulnerable to capture and key compromise. Furthermore, in security schemes where hash chains are utilised the computational

overhead in some cases is unbearable and often these solutions suffer from scalability problems. Thus, the selection of a routing protocol or a security scheme that is to be used in an ad hoc environment should be carefully and thoroughly considered.

Chapter 4
Intrusion Detection

4.1 Introduction

Intrusion detection systems (IDSs) are mainly used to detect and call attention to odd and suspicious behaviour. The first intrusion detection model was developed in 1987 in which Denning proposed a model based on the hypothesis that security violations can be detected by monitoring a system's audit records for abnormal patterns of system usage [Den87]. Hence, intrusion detection is a relatively young technology, as a non-cryptographic approach to computer security in general. However this research has produced a wide range of proposed solutions and strategies for accomplishing intrusion detection goals. Since Denning's initial work, many IDS prototypes have been created and several surveys have been published [MHL94, Fra94, CH96]. A good taxonomy of intrusion detection systems can be found in [DDW98].

Current approaches to intrusion detection can be broadly classified into two trends, *anomaly-detection*, also known as *behaviour-based* intrusion detection, and *misuse-detection*, also called *knowledge-based* intrusion detection. Behaviour-based intrusion detection systems monitor and build a reference profile of normal behaviour for the information system by using statistical methods and try to detect activity that deviates from the normal behaviour profile [Den87, AFV95]. Anything that does not correspond to a previously learned behaviour is considered anomalous and suggests an intrusion attempt. The main advantage of this method is that it can detect attempts to exploit new and unforeseen vulnerabilities without an *a priori* knowledge of explicit security flaws. Thus it can automatically discover new potential attacks. However, this technique suffers from a high volume of false positives, since the entire scope of the system behaviour may not be covered during the learning phase and of course legitimate behaviour may change over time [CWJ01]]. Another weakness of this technique is that it requires

a training period and the assumption that the system in question is free of anomaly during the training period. Of course this cannot always be ensured. Thus in the case that during the training period the network was under attack suggest that the behaviour profile may contain intrusive events.

Knowledge-based IDSs accumulate knowledge about attacks, examine traffic and try to identify patterns indicating that a suspicious activity is occurring. This approach can be applied against known attack patterns only, and needs to update the knowledge base frequently. Virus checkers and scanners follow the knowledge-based paradigm. Generally, knowledge-based systems are attractive in commercial products due to their low false alarm rates and high accuracy. Several techniques have been proposed for knowledge based IDSs and some of those are discussed in following sections. Along with IDSs and techniques that were proposed for infrastructure networks the state of the art research of intrusion detection in ad hoc networking is also presented in this chapter.

4.2 Intrusion Detection in Infrastructure Networks

A wide variety of research papers that present intrusion detection systems and techniques are available in the context of infrastructure networks. Some of the most up-to-date systems were reviewed in the following sections. Along with the other research papers, the “Real-time protocol analysis for detecting link-state routing protocol attacks” approach is presented which constructs the research basis for the RIDAN system.

4.2.1 Specification-based Anomaly Detection

This research study presents a new approach for detecting network intrusions. The new approach is called *specification-based anomaly detection* and it is a hybrid combination of anomaly-detection and knowledge-based intrusion detection techniques [SGF+02]. The authors suggest that the new approach mitigates the weaknesses of the two approaches while magnifying their strengths. To realise their approach they have developed state machine specifications of network protocols, and then they augment these state machines with information about the statistics that need to be maintained to detect anomalies. Furthermore, a specification language was specifically developed in which all of the required information can be captured in a concise manner. The protocol specifications that it are utilised simplify the feature selection process that is required from the anomaly-detection component [SGF+02]. Thus, the machine learning component is claimed to be robust enough to operate without human supervision. The

experiments that were performed in this study indicate that the developed system has low rate of false alarms and that it is able to identify unseen stealthy email viruses in intranet environments.

4.2.2 Statistical Process Control for Computer Intrusion Detection

In this study an interesting architecture of a distributed, host-based IDS is proposed. The system is developed based on statistical process control and employees both of the intrusion detections techniques mentioned earlier. By utilising each technique it determines an intrusion warning level based on the audit data events [YELC01]. The intrusion warning levels are then fused to produce a combined intrusion level. The composite intrusion warning level can have values of 0 for normal to 1 for intrusive, any value that is in between signifies a level of intrusiveness.

4.2.3 A New Intrusion Method based on Process Profiling

This proposed system utilises the anomaly intrusion detection technique in order to identify newly and unseen attacks. The authors suggest that this system requires updated data describing the users' behaviour and the statistics in normal use [OS02]. They call this information *profiles*. Since the profiles updates are usually large it requires extensive use of system resources like CPU time, memory and disk space. They manage to solve these problems by recording system calls from daemon processes. Obviously, this system operates only on Unix-like environment. Thus, they actually protect the system only from attackers that desire to gain root privileges and this is how they manage to reduce the size of the required profiles.

4.2.4. Real-Time Protocol Analysis for Detecting Link-State Routing Protocol Attacks

In this study a real-time knowledge-based network intrusion detection model for detecting link-state routing protocol attacks was developed specifically for the OSPF protocol [CWJ01]. The model is composed of three main layers; a data process layer, an event abstractor layer and an extended finite state machine layer. The data process layer is used to parse packets and dispatch data, while the event abstractor is used to abstract predefined real-time events for the link-state protocol. The extended timed finite state machine layer, which is the most important, is used to express the real-time behaviour of the protocol engine and to detect intrusions by using pattern matching. The timed FSM is called JiNao Finite State Machine (JFSM) and it extends the conventional FSM model with timed states, multiple times, and time constraints on the state

transition. The JFSM is implemented as a generator that can create any FSM by constructing the configuration file only. The results of this research show that this IDS is very effective in identifying real-time intrusions and especially known attacks.

The RIDAN system uses this work as a basis and applies the developed concepts in the field of ad hoc networking environments and more specifically to the AODV routing protocol.

4.3 Intrusion Detection in Ad hoc Networks

Due to the different nature of ad hoc networks, the requirements of an intrusion detection component designed to operate in ad hoc mode should fulfil the following [ACP+02]:

- It should not introduce a new weakness for the system. Ideally it should ensure its own integrity.
- It should require minimum resources to run and it should not degrade the system performance by introducing additional overhead.
- It should run continuously and remain transparent to the system and the users.

In the following sections some of the major intrusion detection works in the field of ad hoc networking (at the time of the writing) are presented.

4.3.1 Watchdog and Pathrater

The watchdog and pathrater scheme consists of two extensions to the DSR routing protocol that attempt to detect and mitigate the effects of nodes that do not forward packets although they have agreed to do so [MGLB00]. The watchdog extension is responsible for monitoring that the next node in the path forwards data packets by listening in promiscuous mode. It identifies as misbehaving nodes the ones that fail to do so. The pathrater assesses the results of the watchdog and selects the most reliable path for packet delivery. The main assumption of this scheme is that malicious nodes do not collude in order to circumvent it and perform sophisticated attacks against the routing protocol. When a node transmits a packet to the next node in the path, it tries to promiscuously listen if the next node will also transmit it. Furthermore, if there is no link encryption utilised in the network, the listening node can also verify that the next node did not modified the packet before transmitting it [MGLB00]. The watchdog of a node maintains copies of recently forwarded packets and compares them with the packet transmissions overheard by the neighbouring nodes. If a node that was supposed to forward a packet fails to do so within a certain timeout period, the watchdog component of an overhearing node increments a failure

rating for the specific node. This effectively means that every node in the ad hoc network maintains a rating assessing the reliability of every other node that it can overhear packet transmissions from. A node is identified as misbehaving when the failure rating exceeds a certain threshold [MGLB00]. The source node of the route that contains the offending node is notified by a message sent by the identifying watchdog. As the authors of the scheme have identified, the main problem with this approach is its vulnerability to blackmail attacks.

4.3.2 Security Enhancements in AODV

In this study the authors propose a solution to attacks that are caused from a node internal to the ad hoc network where the underlying routing protocol is AODV. The intrusion detection system is composed of the Intrusion Detection Model (IDM) and the Intrusion Response Model (IRM) [BA01]. The intrusion detection model claims to capture the following attacks:

- Distributed false route requests.
- Denial of service.
- Destination is compromised.
- Impersonation.
- Routing Information disclosure.

The intrusion response model is a counter that is incremented wherever a malicious act is encountered. When the value reaches a predefined threshold the malicious node is isolated. Although the authors provide some diagrams depicting the accuracy of the model they provide minimal implementation details regarding the model. Thus, even the idea and the model seems feasible the study is not thoroughly documented.

4.3.3 Context Aware Detection of Selfish Nodes in DSR

This system utilises hash chains in the route discovery phase of DSR and destination keyed hash chains and promiscuous mode of the link-layer to observe malicious acts of neighbourhood nodes [PW02]. The observers of the malicious node independently communicate their acquisition to the source node. The source node executes an interference scheme based on the majority voting to rate an accused node. After the source node has reached a decision it advertises this rating along with adequate proofs to trusted nodes. The trusted nodes upon reception of these ratings decide not provide any service to the malicious node. This approach introduces a

fear-based awareness in the malicious nodes that their actions are being watched and rated, which in turn helps in reducing mischief in the system [PW02]. The research does not present any performance measurements but it provides with thorough mathematic proofs their model of operation. A potential problem of this system could be the node mobility. Since the malicious node can go out of range and again come in the network and have a different IP address, it can still take advantage of the network. Although this system cannon be classified as a pure intrusion detection system for the reason that it uses cryptographic mechanisms to detect the malicious attacks, it holds many properties like network auditing to decide whether a node is malicious.

4.4 Summary

The intrusion detection approach in providing security is particularly attractive since it does not require any change in the underlying routing protocol and most of the times it does not require to allocate any valuable network resources. The intrusion detection techniques that have been proposed for ad hoc networks are not many and the field has not been researched thoroughly. We believe that the proposed RIDAN system that is analysed in depth in the following chapters will have a positive impact in the field of wireless intrusion detection.

Chapter 5
Design

5.1 Introduction

In this chapter the design of the RIDAN intrusion detection component is described in more detail. As it was mentioned in a previous chapter, the RIDAN system is based on the research presented in the “Real-time protocol analysis for detecting link-state routing protocol attacks” [CWJ01]. However, that system was developed to operate for OSPF, a routing protocol that is widely utilised in wired networks, therefore their solution could not be directly applied in AODV-enabled networks. The main components that are included in the RIDAN system and were inspired by the above mentioned research are the timed finite state machines, the traffic interception module and the prevention module that deals with countering malicious activity.

A finite state machine can be defined as an abstract machine consisting of a set of states (including the initial state), a set of input events, a set of output events, and a state transition function [Nis03]. The function takes the current state and an input event and returns the new set of output events and the next state. The state machine can also be viewed as a function which maps an ordered sequence of input events into a corresponding sequence of (sets of) output events. *Timed finite state machines* are an extension to normal finite state machines that are used when real-time behaviour is required. Thus, timed finite state machines are state-transition graphs with timing constraints using finitely many real-valued clocks [HM96]. Hence, a time related event may trigger the machine to move forward to a new state.

In the following sections the operation of the RIDAN system is overviewed and the main objectives and assumptions required for its successful operation are illustrated. Moreover, the routing attacks that can be performed against the AODV protocol that the system attempts to detect are also presented. Finally, the modelling of the intrusion detection component along with

the timed finite state machines is described along with the countermeasures that minimise the effects of the active attacks.

5.2 System Overview

The RIDAN system can be characterised as an architecture model for intrusion detection in wireless ad hoc networks, while its implementation targets specifically the AODV routing protocol. The reason why it can be classified as an architecture model is that it does not perform any changes in the underlying routing protocol but it merely intercepts routing and application traffic. Thus, the security component operates in a different layer without interfering with the normal operation of the routing protocol. Since the RIDAN system does not utilise any cryptographic mechanism to ensure protection from malicious activities, it does not introduce any additional computation overhead to the routing process. Furthermore, it does not require the sending of additional packets, thus it does not consume the available bandwidth.

The underlying protocol used for the implementation of the RIDAN intrusion detection component is the AODV routing protocol that recently became an Internet standard [PR03], coming one step closer towards being established as the main routing protocol to be used in ad hoc networking environments. Although some of the attacks that the RIDAN system is designed to detect are specific to the AODV protocol, like the sequence number attack that is presented in a following section, the process of detecting the attacks and the overall architecture can be extended to operate with other protocols, like DSR and OLSR.

The system utilises timed finite state machines to formally define attacks against the routing process. Therefore, it follows the knowledge-based technique to detect network intrusions. The fact that it uses timed finite state machines (FSMs) enables the system to detect malicious activity in real-time rather than using statistical analysis of previously captured traffic. The intrusion detection component operates locally in every participating node and thus depends on the network traffic a node observes. Based on the observed packets more that one FSMs that are part of the intrusion detection component may be triggered. The FSMs were constructed after researching the internal operations of the AODV routing protocol. In order to recognise the traffic patterns occurring when a malicious attack is performed against the routing fabric the traffic of the protocol was analysed in both its normal operation and when an active attack is in progress. The timers that control the transition between the states of the FSMs where derived form theoretical research and practical experimentation using the network simulator (ns-2).

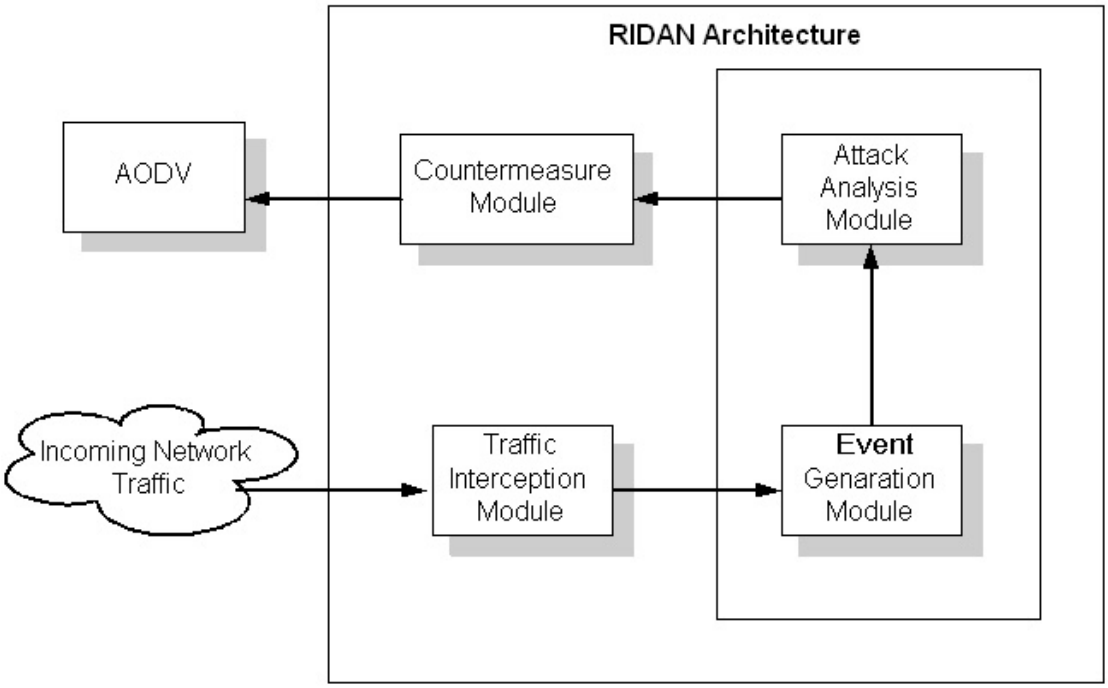

Figure 5.8: High-level architecture of the RIDAN logical components

In figure 5.8 the high-level architecture of the RIDAN system logical components is shown. The traffic interception module captures the incoming traffic from the network and selects which of these packets should be further processed. The event generation module is responsible for abstracting the essential information required for the attack analysis module to determine if there is malicious activity in the network. The event generation module and the attack analysis module are realised by the use of the timed finite state machines. The final component of the architecture is the countermeasure module that is responsible for taking the appropriate measures to keep the network performance within acceptable performance measures. Therefore, the RIDAN intrusion detection component operates between the network traffic and the routing protocol requiring no modifications to the routing protocol that is utilised in the network.

The RIDAN intrusion detection system runs locally in every participating node and it makes decisions upon the partial view of the traffic that it observes. Thus, RIDAN is a host-based network intrusion detection system. The operation of the system could be terminated when a malicious node is detected, however in order to provide a more complete solution the nodes upon an alarm take countermeasures to deal with the isolation of the detected misbehaving node and to keep the performance of the network within acceptable limits. As it is presented in the following chapter, the RIDAN intrusion detection component has high rates of accuracy in detecting the malicious nodes and the countermeasures taken by the nodes individually enable the network to

withstand aggressive attacks and keep the network operating within acceptable performance limits.

5.2.1 Objectives

An enumeration of the objectives of the system will assist in the evaluation process of the RIDAN intrusion detection component. Hence, the objectives of the RIDAN system can be summarised in the following points:

- Create an intrusion detection model for wireless ad hoc networks that can be further extended to operate for many reactive or proactive routing protocols.
- Select some of the active attacks that a malicious node can perform against the AODV routing protocol and implement them.
- Formally describe the detection of the attacks with the use of timed finite state machines and fine tune them to achieve the maximum accuracy.
- According to the observed routing traffic more that one FSM may be triggered simultaneously, however the system as a whole should not reach contradicting decisions.
- Upon detection of malicious activity the detecting node should be able to take countermeasures to hold the network performance in acceptable performance measures.
- The detected malicious nodes will be penalised for a finite period of time rather than being isolated for ever in order to avoid the impact of possible false positive alarms.

5.2.2 Assumptions

In order for the RIDAN system to work some factors have to be true. The assumptions that were made during the implementation of this system are not farfetched or unrealistic and can be easily realised in an ad hoc networking environment. Specifically, our assumptions are the following:

- Every link between the participating nodes is bidirectional.
- Nodes operate in promiscuous mode, meaning that they can listen to their neighbours' transmissions.
- All the participating nodes have the RIDAN intrusion detection component activated apart from the malicious nodes.

5.3 AODV Routing Attacks

In order to test the RIDAN intrusion detection component we selected three attacks that have significant impact on network performance degradation when they are actively performed. Only one of these attacks is specific to the AODV routing protocol, while the other two can be applied to any routing protocol. The three attacks are described in detail in the following sections.

5.3.1 Sequence Number Attack

Protocols such as AODV and DSDV create and maintain routes by assigning monotonically increasing sequence numbers to routes towards specific destinations. Since the freshness of a route is determined from the destination's sequence number and of course fresher routes are preferable, a malicious node can redirect and inject false routing information to the network. The malicious node can perform the black hole attack by inserting itself into the active route.

In figure 5.9 the source node initiates a route discovery process directed to the destination node by sending a RREQ packet. When the malicious node receives the RREQ even if it does not have a fresh enough route in its routing table it creates a RREP with forged information about the sequence number and the next hop. In order for the false information to be favoured the malicious node puts a high sequence number to the destination sequence number field. If the RREP from the malicious node is received before the one from the legitimate source node then it manages to put itself in the route and it can intercept the routing packets or perform a black hole attack. Even if the malicious RREP does not reach the source node first it will eventually reach it and because the destination sequence number will be greater the original route will be replaced by the forged route.

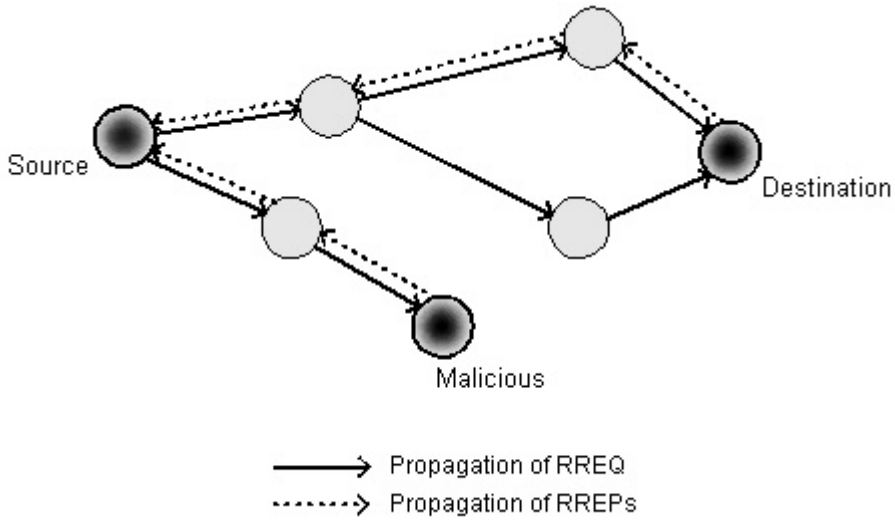

Figure 5.9: Example of the sequence number attack

The strength of this attack is that the forged route will be propagated from the legitimate nodes as well since they can reply to future RREQs with the false entries that exist in their routing tables. Thus, the false routing information will propagate to other nodes without the intervention of the malicious node.

5.3.2 Dropping Routing Traffic Attack

Mobile nodes due to limited battery life and limited processing capabilities may decide not to participate in the routing process in order to conserve energy. Thus, a malicious node upon receiving a routing packet that is not destined for itself or it was not initiated by it deliberately drops it. The node by acting selfishly conserves energy but it may also cause network segmentation. If the some of the participating nodes are only connected with the malicious node then they become unreachable and isolated from the rest of the network.

5.3.3 Resource Consumption Attack

In this attack the malicious node attempts to consume both the network and node resources by generating and sending frequent unnecessary routing traffic. This routing traffic can only be RREQ and RERR packets since all false RREP are automatically discarded by the specification of the AODV protocol. The goal of this attack is to flood the network with false routing packets to consume all the available network bandwidth with irrelevant traffic and to consume energy and processing power from the nodes.

5.4 Modelling of the RIDAN Intrusion Detection Component

The design of the timed FSMs is crucial in the RIDAN system since it is the component that decides if a node should trust another node or go to an alarm state and take countermeasures against it. However, because in ad hoc networks it is difficult to define a normal traffic flow and the FSMs do not have complete accuracy in always identifying correctly the malicious node, the countermeasures that a node takes does not isolate or penalise the node for ever. The following sections present the timed FSMs that were designed specifically to detect the above mentioned attacks.

5.4.1 Sequence Number Attack Detection

In order for the RIDAN intrusion detection component to identify the sequence number attack three different FSMs are required.

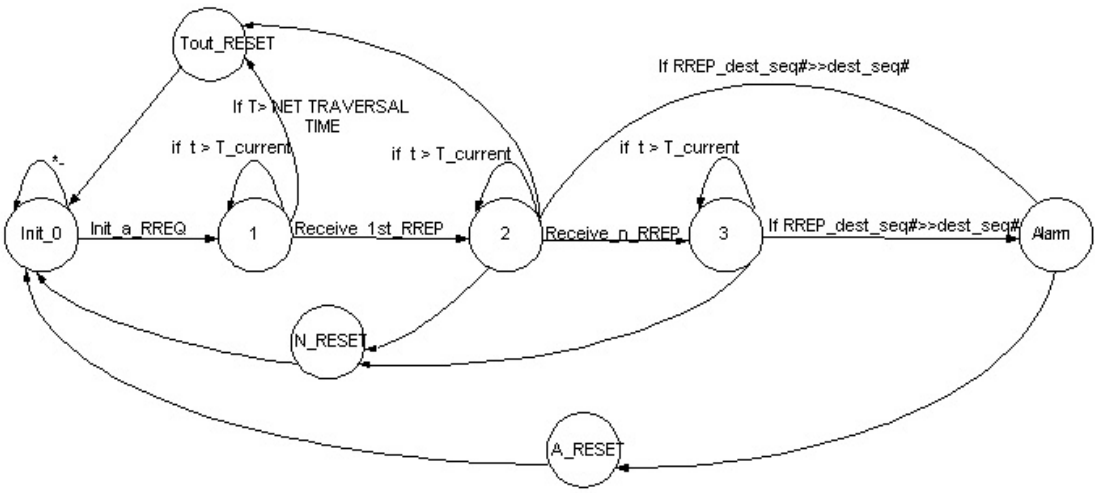

Figure 5.10: First sequence number attack detection FSM.

In figure 5.10 the first FSM is graphically illustrated. This FSM is triggered whenever a node initiates a route discovery process. If a RREP message does not arrive within the *NET_TRAVERSAL_TIME* defined internally in the AODV implementation the FSM goes to Time-out RESET (*Tout_RESET*) and resets to its initial state (*init_0*). Upon the reception of the first RREP it checks if the RREP_destination_sequence_number (*RREP_dest_seq#*) is much higher than the original_sequence_number (*orig_dest_seq#*) included in the RREQ. If it is

suspiciously higher it goes directly in the alarm state (*Alarm*). If it is not, it waits in the same state for time *t*. If the timer expires without receiving another RREP it moves to Normal_Reset (*N_RESET*). If within the timer it receives another RREP(s) it checks for the validity of the destination sequence number and similarly decides whether to move to an alarm state. When an alarm occurs the source node knows that the information in the RREP is forged and that it must not update the routing table with the invalid routing information. From the state of the *Alarm* the FSM resets with an Alarm Reset state (*A_RESET*) to its initial state (*init_0*).

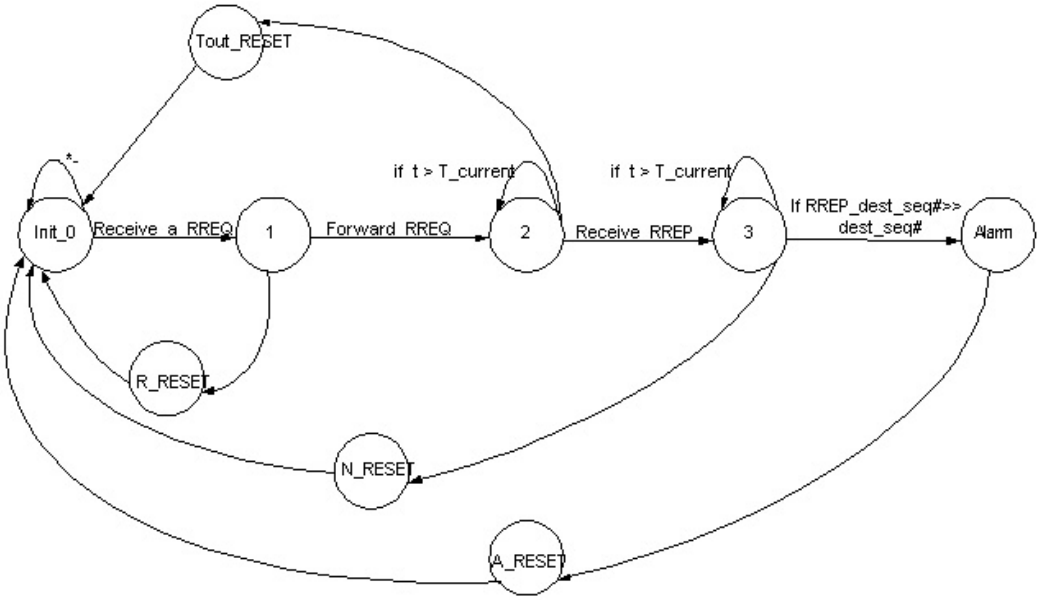

Figure 5.11: Second sequence number attack detection FSM.

The second FSM (figure 5.11) for this attack protects the intermediate nodes that receive the RREQ initiated for the source node. Thus when an intermediate node receives a RREQ and has a fresh enough route to the destination it replies and the FSM moves to REPLY_RESET (*R_RESET*). In case the intermediate node does not have the necessary information to reply to this RREQ it forwards the packet down stream and moves to state *two*. The FSM remains to this state for time t. If the timer expires it moves to a *Tout_RESET* and back to the initial state *init_0*. If within the time limits it receives a RREP it moves to state 3 and checks for the validity of the destination sequence number as in the previous FSM. If the sequence number is within the acceptable time limits it goes to the *N_RESET* state and normally resets the FSM, otherwise it goes to an alarm state and it does not add the forged route to its routing table. From the alarm

state it resets with *A_RESET*. It should be noted at this point that the intermediate node does not drop the RREP even if it determines that it is forged. Alternatively, it unicasts the RREP back to the source node that will determine by itself that the packet contained invalid information.

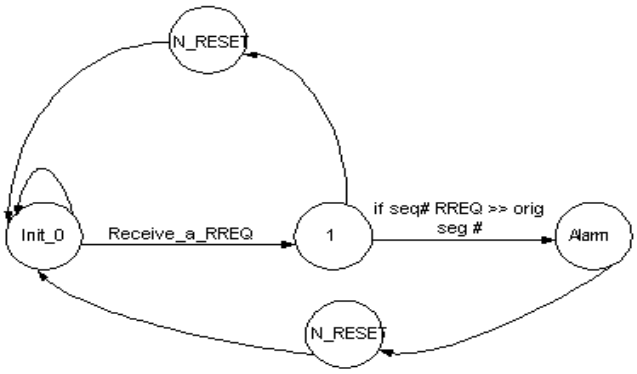

Figure 5.12: Third sequence number attack FSM.

The conditions that are required for the third FSM to be triggered are unlikely to hold, however it is included in order for the system to be complete. This FSM presented in figure 5.12 is triggered when the destination node receives a RREQ for a route to it. If the destination sequence number included in the RREQ is equal to its own it resets normally with *N_RESET*. However, if the *RREP_dets_seq#* is much greater than the one that the destination holds it goes to the *Alarm* state and it does not update its sequence number. If this FSM is triggered it means that the node that initiated the route discovery process had forged route information in its routing table that originated from a malicious node.

5.4.2 Dropping Routing Packets Attack Detection

Since all the nodes participating in the network are in promiscuous mode the neighbouring nodes can detect whether a malicious node has forwarded a routing packet. However, the node in question may have not forwarded the routing packet due to traffic overload. The intrusion detection component in order to prevent false alarms caused by traffic overload, moves first to a pre-alarm state and in this state it unicasts the routing packet to the offending node again.

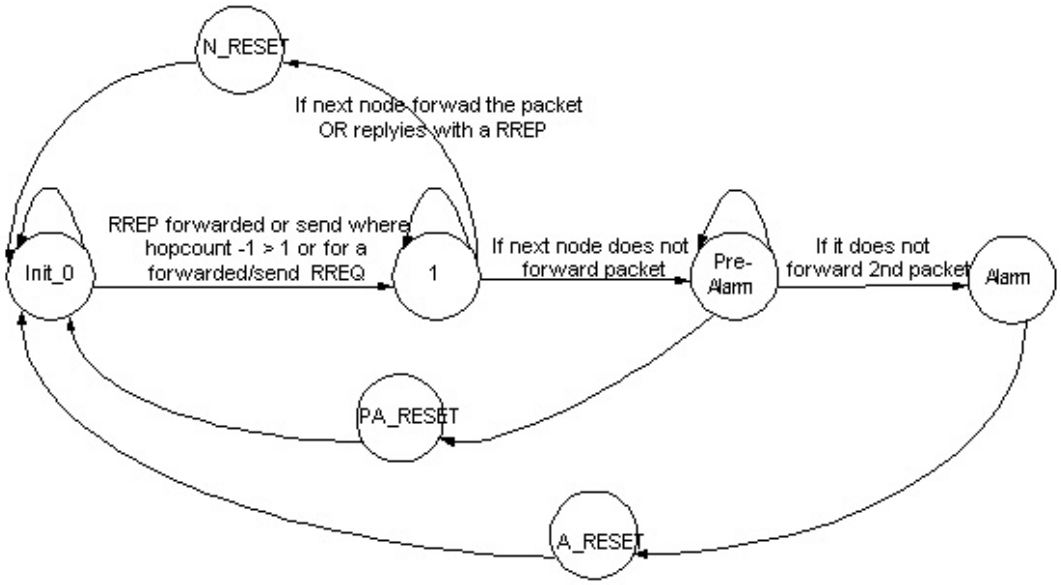

Figure 5.13: Dropping routing packets attack detection FSM.

In figure 5.13 the presented FSM is triggered whenever a node sends or forwards a RREQ or a RREP packet. It remains to state 2 for time t waiting for the node to forward/reply to the routing packet. If the node replies or forwards the packet it normally resets the FSM with *N_RESET*. If the node fails to appropriately respond to the forwarder routing traffic the FSM moves to a *Pre-Alarm* state and remains there for time t. If the node manages to respond appropriately by forwarding the routing traffic or by replying to a RREQ it is removed from the suspected nodes list and the FSM normally resets. Otherwise, the FSM goes to an alarm state and the observing node marks this node as malicious, thus it does not forward any kind of traffic through this node and it also sends a RRER packet to the upstream neighbours in order to prevent them from sending traffic through the malicious node.

5.4.3 Resource Consumption Attack Detection

The resource consumption detection FSM is triggered for every different node that sends a routing packet. The observing node keeps a list with all the nodes from which it has recently received routing traffic along with a counter that signifies the number of packets that the specific node sent and a timer.

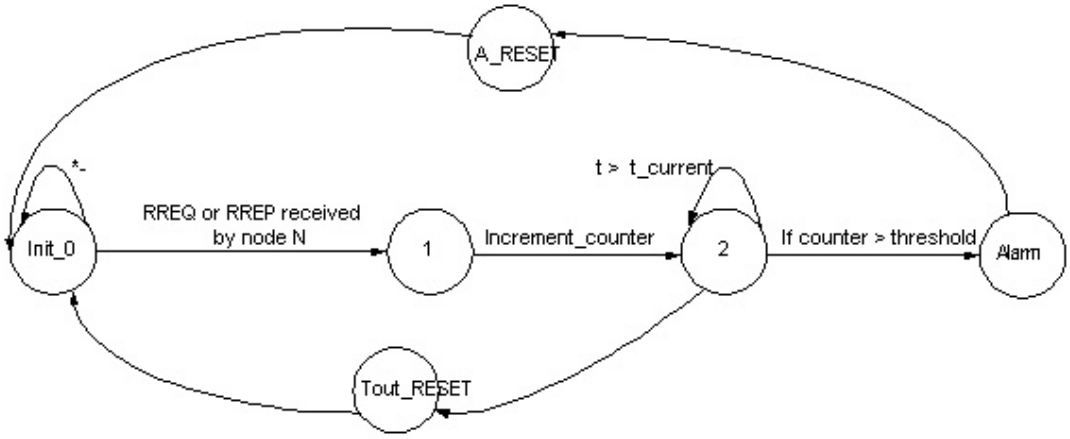

Figure 5.14: Resource consumption attack detection FSM.

In figure 5.14 the FSM increments the counter for every new routing packet received from this node. It remains in the state 2 for time *t*. If the counter reaches the threshold value it means that it has detected abnormal traffic generation and it moves to the *Alarm* state. Upon an alarm the node drops all the incoming routing traffic from the offending node for finite time interval so that it does not consume network and node resources.

5.5 Summary

The main design of the RIDAN system was presented in this chapter. The design of the timed finite state machines is a very important element in detecting the active attacks described. The traffic patterns that denote the occurrence of an attack were derived from the theoretical analysis of the behaviour of the protocol as well as from practical experiments in the network simulator (ns-2). The implementation details of the attacks and the timed finite state machines are demonstrated in depth in the following chapter.

Chapter 6
Implementation

6.1 Introduction

The RIDAN system was implemented in the network simulator (ns-2). The utilised version of AODV was the default one included in ns-2. The three active attacks and the RIDAN intrusion detection component were developed as new routing agents based on the implementation of AODV included in ns-2. In order to realise the appropriate behaviour several methods of the default AODV routing agent were extended. Every routing agent inherits the methods and attributes of the normal AODV implementation and modifies only the methods required to achieve either one of the attacks or the timed finite state machines of the intrusion detection component as illustrated in figure 6.15. The main operation of AODV is implemented in the header file named `aodv.h` and in the C++ file named `aodv.cc`. The packet reception routines of AODV are rather simple. For every packet that is received the `recv` method analyses the header and checks whether the received packet is an application or a routing packet. If it is an application packet destined for the current node the processing continues, else it is forwarded towards its destination. In the case of a routing packet arrival the `recv` method calls the `recvAODV` method that analyses further the packet header to determine if the packet is a route request, a route reply or a route error and calls the appropriate method to handle it. Due to the clear processing distinction between the application and the routing packets, the RIDAN traffic interception module can be easily placed in the `recv` method. The event generation and the attack analysis modules, which are together implemented by the timed finite state machines, along with the countermeasure module are distributed in the methods of the RIDAN intrusion detection agent as presented in table 6.1.

Method name	RIDAN module	Used in the detection of
`recv`	Traffic Interception Module	All the attacks.
`recvReply`	Countermeasure Module Attack Analysis Module	Sequence number attack
	Countermeasure Module	Resource consumption attack
`sendRequest`	Event Generation Module	Sequence number attack
`tap`	Attack Analysis Module Countermeasure Module	Dropping routing traffic attack
`recvRequest`	Event Generation Module Attack Analysis Module	Resource consumption attack
	Event Generation Module	Dropping routing traffic attack

Table 6.1: Distribution of the logical modules of the RIDAN system in the methods of the RIDAN-enabled AODV agent.

It should be noted at this point that AODV does not operate normally in promiscuous mode. However, for detecting the dropping routing packets attack it is required for the AODV to be promiscuously enabled. The tap agent that handles the traffic that the promiscuously enabled node overhears was used only to facilitate the intrusion detection component while the normal operation of AODV was not affected by it.

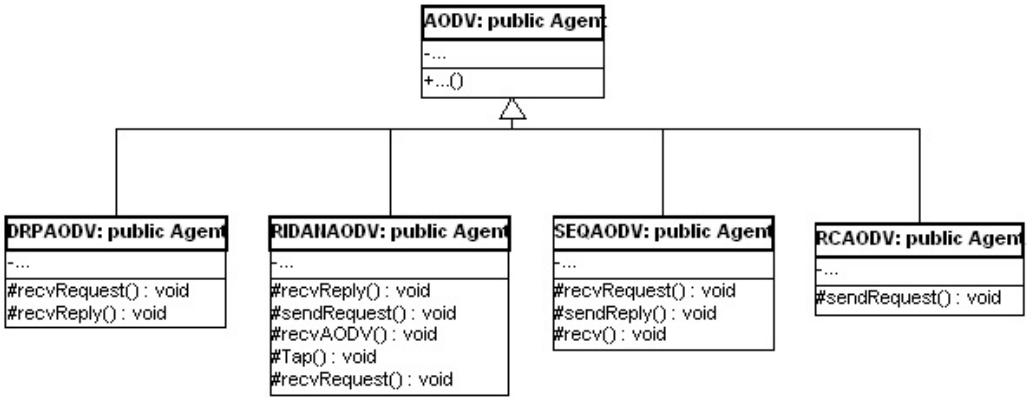

Figure 6.15: The class diagram of the system. The methods and attributes of the AODV public Agent are omitted for readability reasons.

In the following sections the implementation details of both the attacks and the detection mechanisms are presented in depth. Additionally, the results of the experiments are graphed to

illustrate diagrammatically the network performance improvements of the RIDAN-enabled AODV protocol under active attacks.

6.2 The ns-2 Network Simulator

The network simulator *ns-2* [NS03] is an object–oriented, discrete event-driven network simulator developed at the OC Berkley and ISC ISI as part of the VINT project [VIN03]. It is a very useful tool for conducting network simulations involving local and wide area networks. In the recent years its functionality has grown to include wireless and ad hoc networks as well.

The ns-2 network simulator has gained an enormous popularity among participants of the research community, mainly because of its simplicity and modularity. The network simulation allows *simulation scripts*, also called *simulation scenarios*, to be easy written in a script-like programming language TCL. More complex functionality relies on C++ code that either comes with ns-2 or is supplied by the user. The utilisation of the two programming languages increases the flexibility of ns-2. C++ is mainly used for event handling and per-packet processing tasks for which TCL would become too slow. TCL is most commonly used for simpler routing protocols, general ns-2 code and simulation scenario scripts. The usage of TCL for simulation scenario scripts allows the user to change parameters of a simulation without having to recompile any source code.

Simulations in ns-2 can be logged to *trace files*, which include detailed information about packets in the simulation and allows for post-run processing with some analysis tool [KV02]. It is also possible to let ns-2 generate a special trace file that can be used by *NAM* (Network Animator), a visualisation tool that is part of the simulator distribution.

6.3 Implementation of the Sequence Number Attack

The malicious node in the sequence number attack replies to all RREQ packets it receives regardless of whether it is has the route to the specific destination or not. Of course the RREQ packets for which the malicious node is the destination are replied normally. In order to add the malicious routing agent that performs the sequence number attack in the network simulator (ns-2) some internal TCL files had to be modified. In table 6.2 the TCL files where the new malicious routing agent was added are presented. In these files the malicious agent, called SEQAODV, was added similarly to the normal AODV routing agent. The SEQAODV agent was declared in the `aodv.h` header file and it was implemented in the `aodv.cc` file. The methods that had to be changed to achieve the goal of the attack are the `recvRequest`, `sendReply` and `recv` as presented in figure 6.15.

File location	File name	Code added

../ns-2.1b9a/tcl/lib	ns-agent.tcl	Agent/SEQAODV instproc init args {
		$self next $args
		}
		Agent/SEQAODV set sport_ 0
		Agent/SEQAODV set dport_ 0

../ns-2.1b9a/tcl/lib	ns-lib.tcl	Simulator instproc
		create-seqaodv-agent { node } {
		set ragent [new Agent/SEQAODV [$node id]]
		$self at 0.0 "$ragent start"
		$node set ragent_ $ragent
		return $ragent
		
		and
		
		SEQAODV {
		set ragent [$self
		create-seqaodv-agent $node]
		}

../ns-2.1b9a/tcl/lib	ns-packet.tcl	In foreach prot SEQAODV
../ns-2.1b9a/tcl/lib	ns-mobilenode.tcl	set seqaodvonly [string first "SEQAODV" [$agent info class]] if {$seqaodvonly != -1 } { $agent if-queue [$self set ifq_(0)]}

Table 6.2: The TCL files that were modified to add the new SEQAODV routing agent.

The method that handles the reception of a request (`recvRequest`) is obviously required in the implementation of the attack. The following table presents the pseudocode of the `recvRequest` method as used in the normal AODV protocol and the modified `recvRequest` as used in SEQAODV.

AODV	SEQAODV

void recvRequest(Packet)
{
extract the header of the packet;
 	
drop the packet if
{
I am the source;
or if I have recently heard of
this request;
}
 	
cache the broadcast ID;
add the reverse route;
 	
if I am the destination of this
	void recvRequest(Packet)
	{
	extract the header of the packet;
	
	drop the packet if
	{
	I am the source;
	or if I have recently heard of this
	request;
	}
	
	cache the broadcast ID;
	add the reverse route;
	
	if I am the destination of this

```
 route discovery
 {
  increment sequence number;
  send reply with hop count one;
 }

 if I have a fresh enough route to
 the destination
 {
  reply with the appropriate
  information from my routing
  table;
 }
 else
 {
  forward the route request;
 }
}
```

```
 route discovery
 {
  increment sequence number;
  send reply with hop count one;
 }
 else
 {
  add to the sequence number a
  random number;

  fill the rest fields of the packet
  as if I am the destination of this
  route request;

  send reply;
 }
}
```

Table 6.3: recvRequest pseudocode.

The malicious node replies to all route requests with false routing information. In order to guarantee that its reply will be preferred over the legitimate one it generates a (peudo) random number between 5 and 200 and adds it to the sequence number that is included in the header of the route request packet. The reason that a randomly generated sequence number is used instead of a static one is to make the attack more realistic.

The method `sendReply` was modified so as not to produce any errors when the forged RREP packet is sent. The recv method is the first method that is used upon a packet reception. When legitimate nodes use the forged route to send packets the malicious node has to decide what to do with these application packets. Hence, the malicious node drops all the packets that come from nodes that use the forged route. To realise this behaviour the attacking node uses a linked list were it adds all the source nodes that were added in their routing table the forged information included in the RREP that it previously sent. Thus, every time the misbehaving host sends an invalid RREP it adds the source node that initiated the route discovery process in this list along with the destination for which the RREQ was originally destined to. When an application packet arrives the node checks if the source of the packet was one of the nodes that use the forged route. If this is true then the malicious node drops that application packet. If it is not part of the list it forwards the packet to its real destination.

We believe that this is the most appropriate behaviour that the malicious node should have after becoming part of the route. However, in research presented in [WLB03] the dropping rates illustrated in the diagrams of the study can be only achieved if the malicious host drops all application packets that go through a route that it is part of. In this implementation lower dropping rates are observed however the malicious node operates in a more robust and realistic way.

6.3.1 Implementation of the Sequence Number Attack Detection

In the detection of the sequence number attack three different timed finite state machines are utilised. The timed finite state machines are realised in the *event generation* and *attack analysis* modules. The RIDAN intrusion detection component was implemented as a new routing agent similarly to the way that the sequence number attack routing agent was realised. The internal TCL files of the network simulator that were modified in order to add the new routing agent, called RIDANAODV, are presented in table 6.4.

File location	File name	Code added

../ns-2.1b9a/tcl/lib	ns-agent.tcl	`Agent/RIDANAODV instproc init args {`
		`$self next $args`
		`}`
		`Agent/RIDANAODV set sport_ 0`
		`Agent/RIDANAODV set dport_ 0`

../ns-2.1b9a/tcl/lib	ns-lib.tcl	`Simulator instproc`
		`create-ridanaodv-agent { node } {`
		`set ragent [new Agent/RIDANAODV`
		`[$node id]]`
		`$self at 0.0 "$ragent start"`
		`$node set ragent_ $ragent`
		`return $ragent`
		
		and
		
		`RIDANAODV {`
		`set ragent [$self`
		`create-ridanaodv-agent $node]`
		`}`

../ns-2.1b9a/tcl/lib	ns-packet.tcl	In `foreach prot` `RIDANAODV`

../ns-2.1b9a/tcl/lib	ns-mobilenode.tcl	`set ridanaodvonly [string first`
		`"RIDANAODV" [$agent info class]]`
		`if {$ridanaodvonly != -1 } {`
		`$agent if-queue [$self set ifq_(0)]}`

Table 6.4: The TCL files that were modified to add the new RIDANAODV routing agent.

The timed finite state machines used to detect the sequence number attack were presented in the previous chapter. The first finite state machine is triggered whenever a route discovery process is initiated. If the *NET_TRAVERSAL_TIMER* which is 0.15 seconds expires the FSM resets due to time out (T_OUT). If within this time interval the source node receives a RREQ in regard to the previously sent RREQ, the FSM transits to state 2 where it checks if the sequence number included in the RREP is higher than the threshold value 5. The threshold value was

determined through experiments to be the most efficient in successfully detecting the sequence number attack. If the sequence number of the reply is higher than the threshold value the Alarm state becomes true and the route included in the RREP is not added to the routing table. If the sequence number of the RREP is lower than the threshold value then the FSM remains to state 2 for the time interval of 3 seconds in order to wait for any other RREP packets that respond to the RREQ. Upon reception of another RREP the FSM performs again the same check for the sequence number. This FSM was implemented with boolean states that are used for achieving state transition. The pseudocode of the implementation of the detection is presented in table 6.5.

RIDANAODV

```
void sendRequest(nsaddr_t dst)
{
 check if I have the route for the destination dst;

 if the route exists in the routing table and it has not expired
 {
  return;
 }

 else
 {
  send RREQ for destination dst;
  sent_rreq_state = true; // The FSM transits from the initial state to
                          // the first state of the FSM.
 }
 // The method continues normally as in the normal AODV
}
```

RIDANAODV

```
void recvReply(Packet p*)
{
 extract the IP_header of the packet p;
 extract the routing information included in the packet p;

 if (sent_rreq_state == true and the RREP is in response to the RREQ sent
 and the timer of sent_rreq_state has not expired)
 {
  recv_rrep_state = true;
 }

 else if timer has expired
 {
  reset(seqFSM 1);
 }

 if (recv_rrep_state == true)
 {
  if (RREQ_dest_seq_no + threshold_value < RREP_dest_seq_no)
  {
   Alarm = true;
   drop RREP packet;
   return;
```

```
   reset(seqFSM 1);
  }
 }
 else
 {
  // proceed with the normal operation of the method
 }
}
```

Table 6.5: Pseudocode of the implementation of the RIDAN detection component for the first FSM used to detect the sequence number attack.

The second FSM is triggered whenever an intermediate node receives a RREQ. If the intermediate node has a fresh enough route for the destination specified in the route request then the FSM resets normally. If not it moves to state 2 where it remains for 15 seconds. If within this time period it receives a route reply in response to the RREQ then it moves to state 3 and checks for the validity of the sequence number included in the reply against the threshold value 5 as previously. If a RREP does not arrive within the time limit the FSM resets due to time out. If the sequence number is determined to be invalid then the FSM transits to the Alarm state where it does not add the routing information to its own routing table and forwards the packet towards its destination. When the destination sequence number is normal then the FSM resets normally. The pseudocode of the implementation of the detection mechanism follows.

```
RIDANAODV

void recvRequest(Packet p*)
{
 extract the IP_header of the packet p;
 extract the routing information included in the packet p;
 drop it if I am the source of the packet of if I have recently heard of
 this request;

 recv_rreq_state = true; // Upon reception of a RREQ trigger FSM

 check if I have a fresh enough route to the destination;

 if I have and I can reply
 {
  send RREP;
  reset(seqFSM 2);
 }
 else
 {
  forward RREQ packet to neighbouring nodes;
 }

 // The method continues as in the normal AODV
}
RIDANAODV
```

```
void recvReply(Packet p*)
{
 extract the IP_header of the packet p;
 extract the routing information included in the packet p;

 if (recv_rreq_state == true and the RREP is in response to the RREQ sent
 and the timer of sent_rreq_state has not expired)
 {
  recv_rrep_state = true;
 }
 else if (timer has expired)
 {
  reset(seqFSM 2);
 }

 if (recv_rrep_state == true and timer has not expired)
 {
  if (RREQ_dest_seq_no + threshold_value < RREP_dest_seq_no)
  {
   Alarm = true;
   discard any updates done;
   forward packet;
   reset(seqFSM 2);
  }
 }
 else
 {
  reset(seqFSM 2);
  // proceed with the normal operation of the method
 }
}
```

Table 6.6: Pseudocode of the implementation of the RIDAN detection component for the second FSM used to detect the sequence number attack.

The third FSM that is required for the detection of the sequence number attack is more of a supplementary nature. It is triggered whenever a node receives a RREQ for which it is the destination. If the sequence number included in the RREQ is not the one that it has for itself then it transits to an Alarm state and it does not update its own sequence number to be equal to the one included in the RREQ. The only case where this may happen is if the source node of the RREQ has at some point being cheated by a false RREP originating from a malicious node. In the case that everything is normal the FSM resets normally. The pseudocode of the implementation of the detection mechanism is the following.

RIDANAODV

```
void recvRequest(Packet p*)
{
 extract the IP_header of the packet p;
 extract the routing information included in the packet p;

 drop if I am the source of the packet of if I have recently heard of
 this request;

 if I am the destination of the RREQ
 {
  recv_rreq_state = true; // Upon reception of a RREQ trigger FSM

  if(RREQ_dest_seq_no > my_dest_seq_no)
  {
   Alarm = true;
   increment sequence number by 1; // Like in normal operation
   send RREP packet;
   return;
  }
 }
}
```

Table 6.7: Pseudocode of the implementation of the RIDAN detection component for the third FSM used to detect the sequence number attack.

6.4 Implementation of the Dropping Routing Packets Attack

The rogue routing agent that was implemented to carry out the malicious behaviour of the dropping routing packets attack was added similarly to the SEQAODV and RIDANAODV routing agents. It was required to modify some of the TCL internal files of ns-2 in order to add the new agent called DRPAODV routing agent. The internal files of the simulator that were modified along with the modifications made to add the DRPAODV routing agent are presented in table 6.8.

File location	File name	Code added

../ns-2.1b9a/tcl/lib	ns-agent.tcl	`Agent/DRPAODV instproc init args {`
		`$self next $args`
		`}`
		`Agent/DRPAODV set sport_ 0`
		`Agent/DRPAODV set dport_ 0`

../ns-2.1b9a/tcl/lib	ns-lib.tcl	`Simulator instproc`
		`create-drpaodv-agent { node } {`
		`set ragent [new Agent/DRPAODV [$node id]]`
		`$self at 0.0 "$ragent start"`
		`$node set ragent_ $ragent`
		`return $ragent`

		and
		
		DRPAODV {
		set ragent [$self
		create-drpaodv-agent $node]
		}

../ns-2.1b9a/tcl/lib	ns-packet.tcl	In foreach prot DRPAODV
../ns-2.1b9a/tcl/lib	ns-mobilenode.tcl	set drpaodvonly [string first "DRPAODV" [$agent info class]] if {$drpaodvonly != -1 } { $agent if-queue [$self set ifq_(0)]}

Table 6.8: The TCL files that were modified to add the new DRPAODV routing agent.

In this attack the malicious node acts selfishly and drops all routing traffic that it is not destined for itself. Thus, upon of a RREQ packet it checks if the destination of the route discovery is itself and if this holds then it further processes the packet and sends a RREP. When it receives a RREP packet it checks if it has sent the original request for this route and if this holds it adds the new route to its routing table. The RERR packets are processed normally in all cases. In any other cases it drops the packets without further processing them. The attack is implemented in the methods `recvRequest` and `recvReply` and the pseudocode of the implementation follows.

```
DRPAODV

void recvRequest(Packet p*)
{
 extract the IP_header of the packet p;
 extract the routing information included in the packet p;
 drop if I am the source of the packet of if I have recently heard of this
 request;

 check if I am the destination of this RREQ packet;

 if I am the destination
 {
  send RREP;
 }

 else
 {
  drop the packet p;
 }
}
```

```
DRPAODV

void recvReply(Packet p*)
{
 extract the IP_header of the packet p;
 extract the routing information included in the packet p;
```

```
 check if I am the destination of this RREP packet;

 if I am the destination
 {
  add the new route to the routing table;
  further process the packet normally;
 }
 else
 {
  drop the packet;
 }
}
```

Table 6. 9: Pseudocode of the implementation of the dropping routing packets attack.

6.4.1 Implementation of the Dropping Routing Packets Attack Detection

The timed FSM developed to detect this attack was incorporated into the existing RIDANAODV routing agent that was previously described. It is essential for the detection of this attack to place the participating nodes in promiscuous mode, hence becoming able to overhear the forwarded traffic. Since AODV does not operate in promiscuous mode by default, some modifications had to be performed in the internal files of ns-2 along with some modifications in the AODV protocol implementation to add this functionality. The modifications that were performed are presented in the table 6.10. The fact that promiscuous mode was enabled in AODV had no impact in the overall performance of AODV and the `tap` method that handles the overheard packets is only utilised in the detection of the dropping routing packets attack.

File location	File name	Code added

../ns-2.1b9a/tcl/lib	ns-mobilenode.tcl	`# Special processing for AODV`
		`set aodvonly [string first "AODV"`
		`[$agent info class]] if {`
		`$aodvonly != -1 }`
		`{$agent if-queue`
		`[$self set ifq_(0)]`
		`$agent install-tap $mac_(0)`

../ns-2.1b9a/aodv	aodv.h	`class AODV: public Agent, public Tap`

../ns-2.1b9a/aodv	aodv.cc	In `method command it was added:`
		`else if(strcmp(argv[1],`
		`"install-tap") == 0)`
		`{`
		` Mac *mac = (Mac*)`
		` TclObject::lookup(argv[2]);`
		
		` if(mac == 0)`
		` return TCL_ERROR;`
		` mac->installTap(this);`

		return TCL_OK;
		}
		
		and
		
		void AODV::tap(const Packet *p)
		
		// this method implements the RIDAN
		// intrusion detection component for
		// the dropping routing packets
		// attack

Table 6.10: Changes required to enable AODV in promiscuous mode.

The FSM developed to detect this attack is triggered whenever a node forwards routing traffic to its neighbouring nodes. A structure called `DRP_Node` was developed to hold information necessary to monitor the neighbouring nodes that are suspected for malicious behaviour. The `DRP_Node` data structure holds the following information:

- *node_id*: the IP address of the node to which the routing traffic was forwarded.
- *send_reply*: a boolean value that becomes true whenever the offending node replies to a RREQ packet that was forwarded to it.
- *pre_alarm*: a boolean value that becomes true if the node does not respond as expected to the forwarded traffic.
- *alarm*: a boolean value that becomes true whenever we decide that the offending node performs the dropping routing packets attack.
- *time*: a double variable that keeps the time where the offending node was added in the data structure.

Hence, whenever a node forwards routing traffic for which a neighbouring node is not the destination it adds each neighbouring node to the data structure and waits to observe their behaviour. Then in the `tap` method if it overhears that a neighbouring node has replied to the forwarded RREQ, it means that it has acted appropriately and it can be removed from the monitoring list. If this is not the case and the packet was a RREP then the offending node has to forward the packet. If it fails to do so within the pre_alarm_time_threshold time period, which was determined by experiments to be 0.01 seconds, the pre_alarm state becomes true. The FSM remains in the pre_alarm state for 0.45 seconds which is the alarm_threshold time period. If the offending node fails to forward the routing packet within this time limit the FSM moves to the Alarm state. In case of an alarm the legitimate node marks this node as malicious and stops forwarding traffic to it for 2 seconds and it also sends a RERR message to all its upstream

neighbours to inform them that all the routes that include this node are not valid any more. The pseudocode of the `tap` method that realises the attack analysis and the countermeasure modules is illustrated in the following table 6.11.

AODV

```
void AODV::tap(const Packet *p)
{
 extract the header of the packet p;

 if it is a RREQ packet
 {
  if I am listening to my own RREQ packet
  {
   return;
  }
  if the source of the packet exists in the monitoring list
  {
   remove(source); // since it has forwarded the traffic
  }
 }
 else if it is a RREP packet
 {
  if I am listening to my own RREP packet
  {
   return;
  }

  if the source of the packet exists in the monitoring list
  {
   remove(source); // since it has forwarded the traffic
  }

  if the node exists in the monitoring list
  {
   if the pre_alarm state is false
   {
    check the pre_alarm threshold against the time that this node was
    added in the list;

    if the time threshold has not expired
    {
     move the FSM to the pre_alarm state by setting the pre_alarm value
     to true;
    }
    else // has expired, so remove it
    {
     remove the node form the monitoring list;
    }
   }
   else // the pre_alarm state is true
   {
    check the alarm threshold time against the time that this node was
    in the alarm state;

    if the time threshold has not expired
```

```
    {
     set alarm value to true;
     send RRER packet to upstream neighbours;
     start timer for 2 seconds that the offending node will be penalised;
    }
    else
    {
     remove the node from the monitoring list;
    }
   }
  }
 }
}
```

Table 6.11: Pseudocode of the implementation of the RIDAN detection component for the FSM used to detect the dropping routing packets attack.

6.5 Implementation of the Resource Consumption Attack

In order to implement the resource consumption attack a new rogue routing agent that would realise this behaviour had to be created. Similarly with the other routing agents, some internal TCL files in ns-2 were modified. The new routing agent is called RCAODV and the files that were modified are presented in the table 6.12.

File location	File name	Code added

../ns-2.1b9a/tcl/lib	ns-agent.tcl	`Agent/RCAODV instproc init args {`
		`$self next $args`
		`}`
		`Agent/RCAODV set sport_ 0`
		`Agent/RCAODV set dport_ 0`

../ns-2.1b9a/tcl/lib	ns-lib.tcl	`Simulator instproc`
		`create-rcaodv-agent { node } {`
		`set ragent [new Agent/RCAODV [$node id]]`
		`$self at 0.0 "$ragent start"`
		`$node set ragent_ $ragent`
		`return $ragent`
		
		and
		
		`RCAODV {`
		`set ragent [$self create-rcaodv-agent $node]`
		`}`

../ns-2.1b9a/tcl/lib	ns-packet.tcl	In `foreach prot` `RCAODV`

../ns-2.1b9a/tcl/lib	ns-mobilenode.tcl	`set rcaodvonly [string first "RCAODV" [$agent info class]]`
		`if {$rcaodvonly != -1 } {`
		`$agent if-queue [$self set ifq_(0)]}`

Table 6.12: The TCL files that were modified to add the new RCAODV routing agent.

The implementation of this attack was rather simple. The only modification that had to be made was a loop that sends frequent unnecessary routing traffic. One of the methods that is steadily and frequently used by the AODV routing protocol implementation is the `sendRequest` method. Thus, we decided that this loop that sends the unnecessary routing traffic should be implemented there. The destinations that are used in the unnecessary RREQ and RERR packets are nodes that do not exist in the network. In order for these packets not to be discarded automatically by the protocol implementation the destination nodes should be different each time. Hence, a function that returns pseudo random addresses was developed to realise this task. The number of the additional routing packets that are sent each time the malicious node initiates a route discovery process is randomly chosen between 2 and 10.

6.5.1 Implementation of the Resource Consumption Attack Detection

The timed FSM developed to detect the resource consumption attack was incorporated into the existing RIDANAODV routing agent that was presented in the sequence number attack detection section. To identify this attack two different data structures were developed. The first one is called `RC_Node` and holds the address of a node, a counter that denotes the number of routing packets that were received from the node and a time value that signifies the period in which the traffic was received. For every node that a legitimate node receives traffic from it keeps an associated counter and a timer. If within the time threshold of 7 seconds the counter of a node reaches the threshold value, which is 10, then it is removed from this list and it is added to the Alarm list. The Alarm list is realised by the second data structure called `Alarm_Node` that stores the time that the node was added in the list and the node's address. For the time threshold of 5 seconds all the traffic that origins from nodes that exist in the Alarm list is dropped without being further processed. The pseudocode that illustrates the implementation of the timed FSM is presented in the following table.

RIDANAODV

```
void recvRequest(Packet p*)
{

 // the normal operation of the method remains unchanged

 check if the node's IP address is included in the alarm list
 {

  if the timer has expired
  {
   it is safe to reset by removing the node from the alarm list;
  }
```

```
  else
  {
   drop the packet;
   return;
  }
 }

 check if the node's IP address is included in the list
 {
  if the timer has expired
  {
   remove node the node from the list;
  }
  else
  {
   if the counter is lower than the threshold value
   {
    increment the counter;
   }
   else if the counter has exceeded the threshold value
   {
    add the node to the alarm list;
    remove it from the original list;
   }


  else it means that we do not have a entry for this node
  {
   so we add the node to the original list;
  }
 }

 // The method continues normally
}
```

Table 6.13: Pseudocode of the implementation of the RIDAN detection component for the FSM used to detect the resource consumption attack.

6.6 Summary

In this chapter the implementation details of the attacks and the RIDAN system was presented. One of the most challenging parts in the development of the timed finite states machines was to fine tune them to achieve maximum accuracy. Both the timers and the threshold values that are included the FSMs had to be tested individually as well as combined all together in order to reach satisfactory results. In the following chapter were the evaluation and metrics that were used to determine the performance of the RIDAN system are analysed, the importance of the values presented here will become clearer.

Chapter 7

Evaluation and Conclusions

7.1 Introduction

In this chapter the evaluation of the RIDAN system is presented. Since the system was developed to operate in the environment of the network simulator, the evaluation was carried out with the same tool by developing several experiments that illustrate the performance of the system. The simulation parameters that were used in this evaluation are similar to other research projects like [MGLB00, PW02, ZL00], that used ns-2 as the basic tool for development and evaluation. Additionally, the measurements that were utilised to evaluate performance of both the attacks and the RIDAN system were also based in similar research projects. The reason we decided to use as a basis of the evaluation experiments similar measurements and scenario parameters was to provide meaningful results that could easily be compared to other solutions in the same filed.

Moreover, in this chapter the conclusions that were drawn from this research are discussed. The RIDAN system does not provide for the moment a complete security solution for ad hoc networks. Some proposals for possible future extensions of the RIDAN system are investigated towards this goal.

7.2 Experiments and Measurements

The experiments were carried out using the network simulator (ns-2). The scenarios developed to carry out the tests use as parameters the mobility of the nodes and the number of active connections in the network. The different routing agents that were presented previously were utilised in the experiments. The choices of the simulator parameters that are presented in table 6.8 consider both the accuracy and the efficiency of the simulation.

Simulator	ns-2
Simulation duration	1000 seconds
Simulation area	1000*1000 m

Number of mobile hosts	30
Transmission range	250 m
Movement model	Random waypoint
Maximum speed	5 – 20 m / sec
Traffic type	CBR (UDP)
Data payload	512 bytes
Packet rate	2 pkt / sec
Number of malicious nodes	1
Host pause time	10 seconds

Table 6.14: Simulation parameters.

The host moving speed converses a range from human jogging to vehicle riding in a country field [WB03]. Faster speeds are not considered because the frequency of the route changes will be too high and the resulted performance degradation will not be entirely the effect of the active attacks. The packet rate of connections is chosen to avoid packet dropping caused by congestion even when there are multiple connections converting at the same host.

The following metrics were chosen to evaluate the impact of the sequence number attack: (1) packet delivery ratio, (2) false routing packets sent by the attacker, (3) additional routing overhead introduced by the attacker, and (4) routing packet dropped ratio. The first metric is selected to evaluate the percentage of delivered packets that are affected by the attack and the improvement that it is achieved through the use of the RIDAN intrusion detection component. The other three metrics are more specific to the routing attacks and were used to measure the severity of the attack and the improvement that the RIDAN system manages to achieve over these performance measurements. Apart from the first metric that is used in all attacks, the other three are specifically used to evaluate the impact of the attacks on the overall performance network. Every point in the produced graphs is an average value of data collected from repeating the same experiment five times in order to achieve more realistic measurements.

7.3 Evaluation of the Sequence Number Attack Detection

The two metrics that were used in the evaluation of the sequence number attack detection mechanism are the delivery ratio and the number of false routing packets sent by the attacker. The first metric which is the delivery ratio was plotted against the number of active connections and against the node mobility (figures 6.16 and 6.17 respectively). The following two diagrams present the AODV without any modifications, the normal AODV with one malicious node present performing the sequence number attack, and the AODV with the RIDAN component enabled having in the network a malicious node performing the sequence number attack.

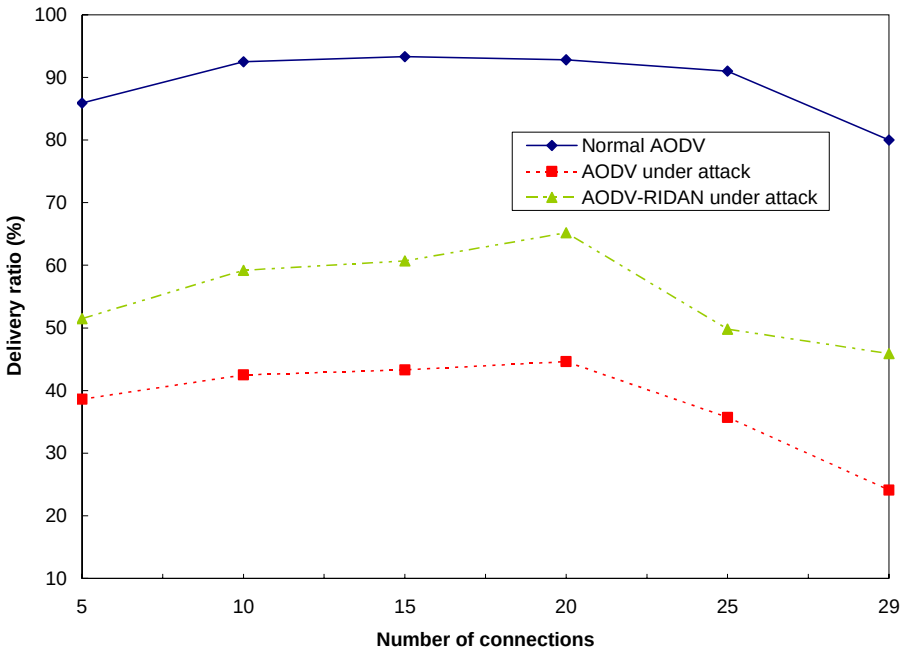

Figure 7.16: Delivery ratio versus number of connection in the sequence number attack.

In this diagram the number of participating nodes in 30 and the maximum number of connections can be 29. A general observation shows that AODV achieves maximum delivery ratio between 10 to 25 active connections and when the number of active connections is increased to 29 there is a slight decrease to 80%. It should be noted that the delivery ratio metric is calculated by dividing the number of total application packets sent by the number of application packets received. The sequence number attack performed against the normal AODV has a very big impact in the delivery ratio decreasing it to lower than the half compared to the normal AODV. The RIDAN intrusion detection component manages to keep the delivery ratio higher in around 60% having a significant improvement.

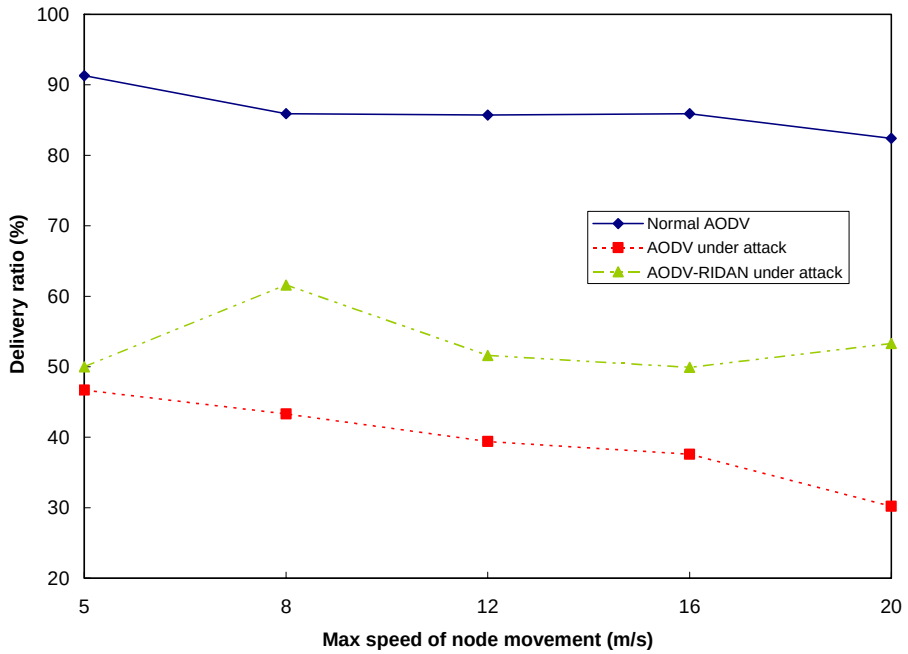

Figure 7.17: Delivery ratio versus node mobility in the sequence number attack.

In the graph 7.17 were the delivery ratio is plotted against the node mobility we observe that AODV performs better in low node mobility rate while as the mobility rate increases the delivery ratio slightly drops. The performance of the network is also significantly reduced when AODV is under the sequence number attack and the node mobility increases. However this behaviour is normal because as the node mobility increases the network topology changes making them more frequently, and therefore the malicious node has the opportunity to send more false RREP packets to the increased route requests that are sent to cope with the route changes. The average values from the two diagrams show the sequence umber attack decreases the delivery ratio from 87.7% to 38.3%. The RIDAN system increases the delivery ratio to 54.3% having an average improvement of 16.6%.

The second metric that was used in the evaluation of this attack is the number of false packets sent by the attacking node versus the number of active connections and the node mobility. This metric is used to examine the overhead of the sequence number attack and we consider only the extra cost on communication imposed by the attack. In the following two figures we can observe that the average number of false RREQ packets sent by the malicious node was 2056.25 and the average number of nodes that inserted the false route into their routing table was 21.7 out of 30. As the mobility increases the number of false RREP packets that are sent by the malicious node also increasing.

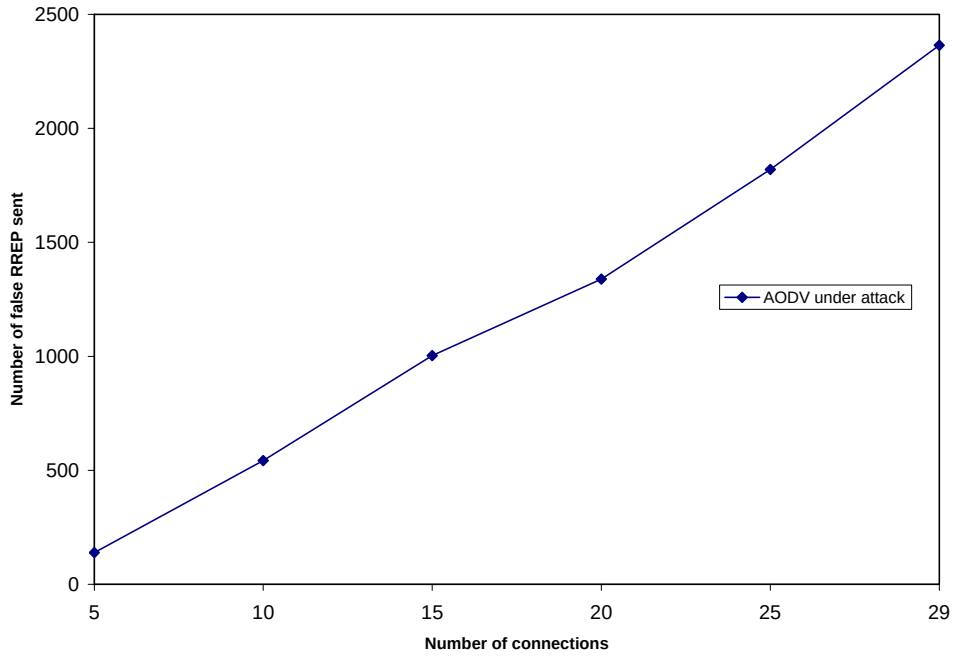

Figure 7.18: Number of false replies sent by the malicious node versus the number of connections.

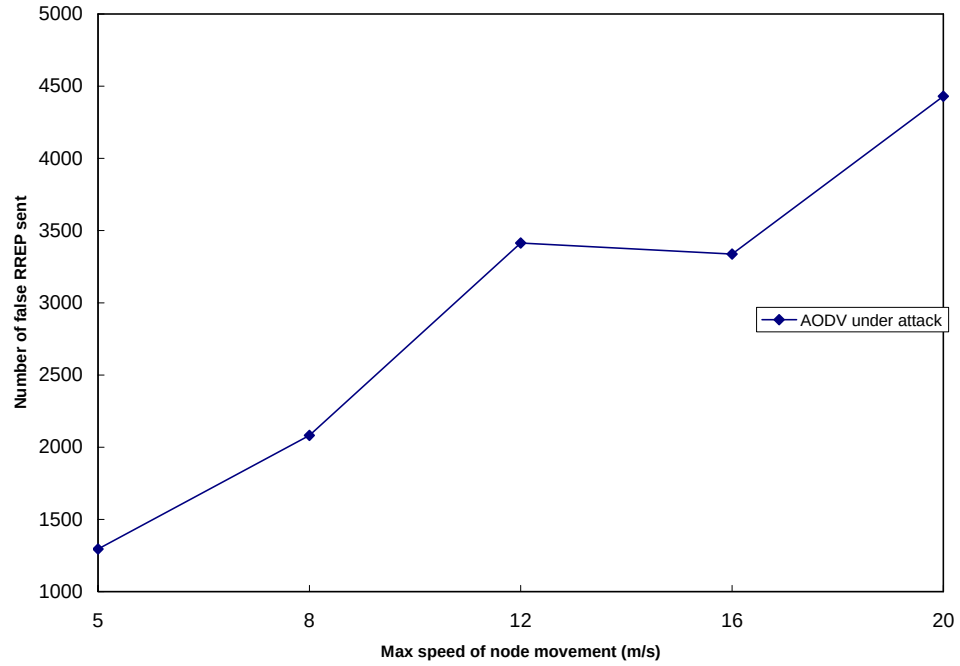

Figure 7.19: Number of false replies sent by the malicious node versus node mobility.

7.4 Evaluation of the Dropping Routing Packets Attack Detection

To evaluate the dropping routing packets attack detection mechanism the metrics of delivery ratio and routing overhead ratio are utilised. The delivery ratio is plotted against the number of active connections and against the node mobility in figures 7.20 and 7.21.

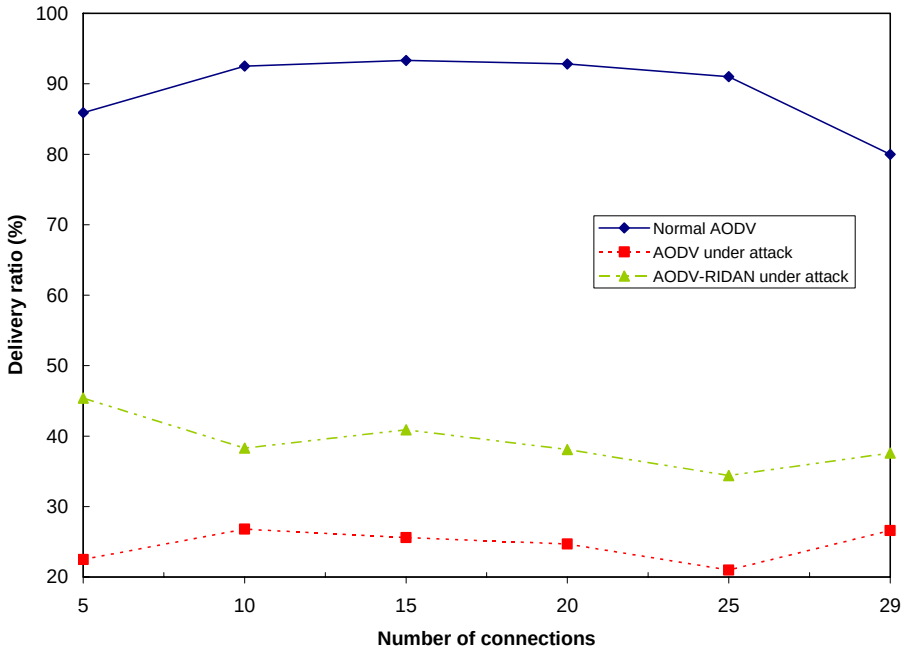

Figure 7.20: Delivery ratio versus number of connection in the dropping routing packets attack.

The dropping routing packets attack has a major impact in network connectivity and this is obvious by the very low delivery ratio that is achieved when the normal AODV is under attack. Even though the RIDAN system improves the delivery ratio by informing all the other nodes for the maliciously behaving node, still the improvement is not very high since the system cannot force the malicious node to forward routing traffic.

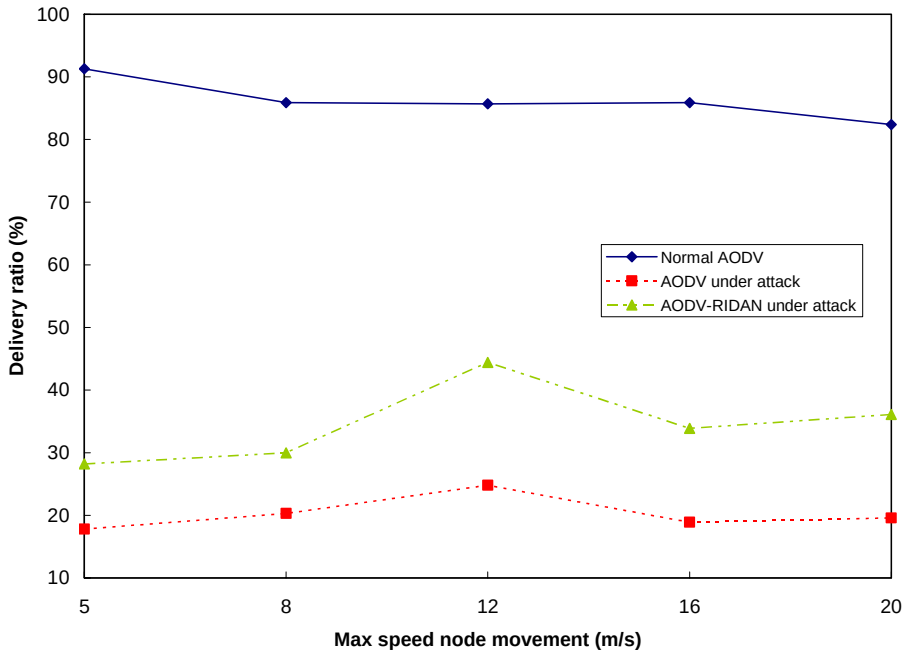

Figure 7.21: Delivery ration versus node mobility in the dropping routing packets attack.

In this diagram (7.21) where the delivery ratio versus the node mobility is graphed, we observe an even lower delivery ratio when normal AODV is under the dropping routing packets attack. This happens because as the node mobility increases, the participating nodes initiate route discovery processes more frequently and the malicious node can drop more routing packets disturbing network connectivity more effectively. The average values from the two diagrams show that the attack decreases the delivery ratio from 87.7% to 23% in average, while the AODV with the RIDAN system improves the delivery ratio to 35.8% having an average improvement of 13.8%.

The second metric that is used in the evaluation of the detection mechanism of this attack is the routing overhead ratio. There are two diagrams plotted; one with the routing overhead ratio against the active connections in the network and one with the routing overhead ratio graphed against the node mobility in figures 7.22 and 7.23 respectively. This metric is used to present the severity of the attack in the additional routing overhead that it introduces and to show how the routing overhead is improved by the RIDAN system.

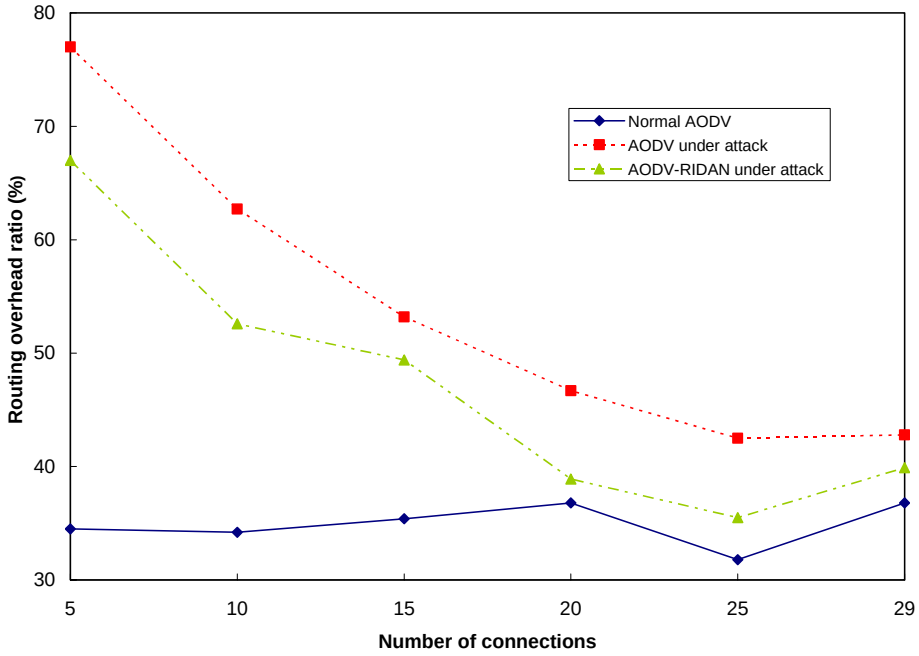

Figure 7.22: Routing overhead ratio versus number of active connections in the dropping routing packets attack.

The additional routing overhead introduced by the attacking node reaches 78% when the network size is small and decreases as the number of connections increases. This behaviour is normal since when there are only five active connections in the network the routes to a destination node are limited, and therefore it is essential that all nodes participate in the routing process. The results drawn from the two diagrams show that the routing overhead created by the attack reaches 50.3% while normal AODV has only 38.7% average of routing overhead. The AODV with the RIDAN system decreases the delivery ratio to 45.1% having an average improvement of 6.1%. As we pointed out before, the improvement is not very high because the RIDAN system does not introduce any mechanism that could force the misbehaving node to participate in the routing process. However, as it is presented in figure 7.23 the RIDAN system manages to reduce the routing overhead ratio to the level that normal AODV demonstrates.

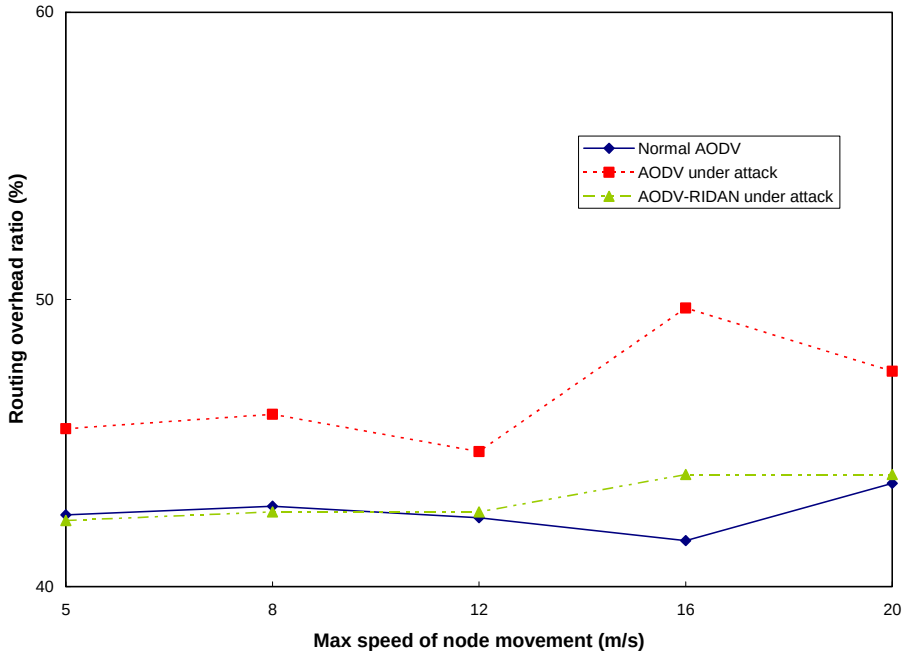

Figure 7.23: Routing overhead ratio versus node mobility in the dropping routing packets attack.

7.5 Evaluation of the Resource Consumption Attack Detection

To evaluate the resource consumption attack detection the delivery ratio and the dropping rate of routing packets ratio are used as metrics. Both metrics are plotted against the number of connections and the node mobility in figures 7.25 and 7.26 respectively.

The impact of network resource consumption attack in the network performance mainly depends on the additional traffic that the malicious node sends every time it initiates a route discovery process. The diagram presented in figure 7.24 shows that for every two additional packets sent by the attacker the additional routing overhead increases approximately 11.8%. The experiments presented in this figure were carried out in a medium-sized network of 15 nodes with medium mobility of 8 meters per second. The number of additional routing packets that is generated by the attacker is 5 for every legitimate packet that he sends in the figures 7.25 and 7.26 that present the resource consumption attack.

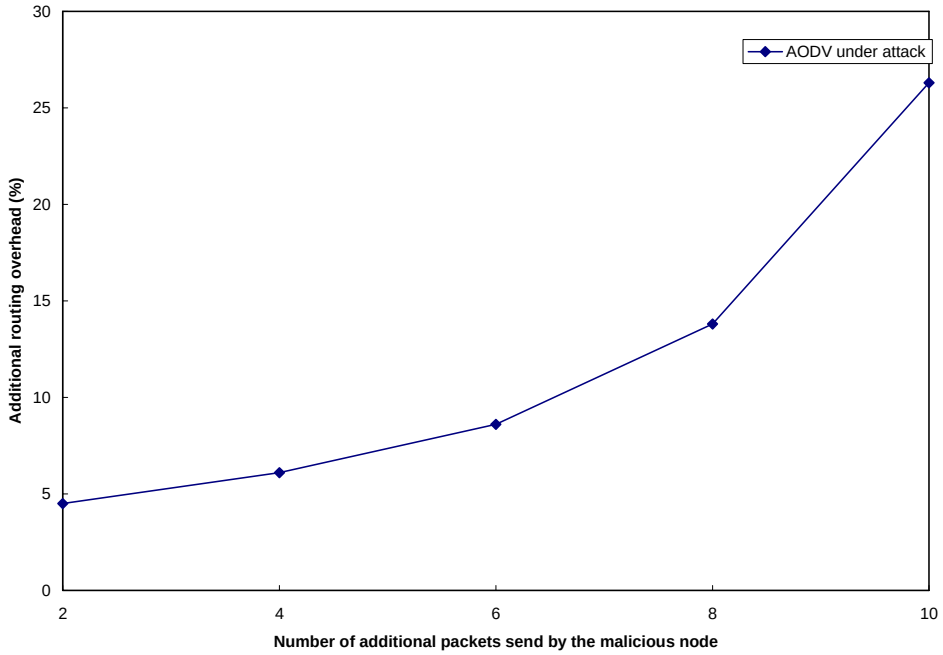

Figure 7. 24: The percentage of additional routing traffic introduced when the number of additional packets sent by the malicious node increases.

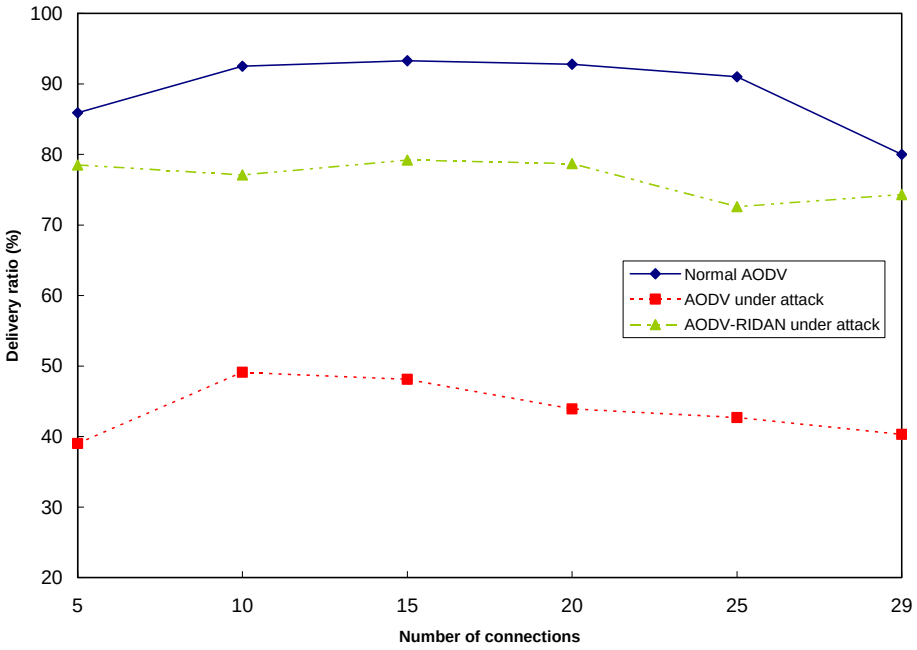

Figure 7.25: Delivery ratio versus number of connection in the resource consumption attack.

The delivery ratio is visibly reduced when AODV is under the resource consumption attack. However, when the RIDAN system is enabled the delivery ratio stays within acceptable performance limits almost only 10% lower than it should normally be. The intrusion detection mechanism of the resource consumption attack is very effective since the countermeasure module enables the observing node to drop the irrelevant traffic preventing it from flooding the network. The average results from the two diagrams (figures 7.25 and 7.26) show that the attack decreases the delivery ratio from 87.7% to 46.2%. The AODV with the RIDAN system improves the delivery ratio to 76.8% having an average improvement of 31.6%.

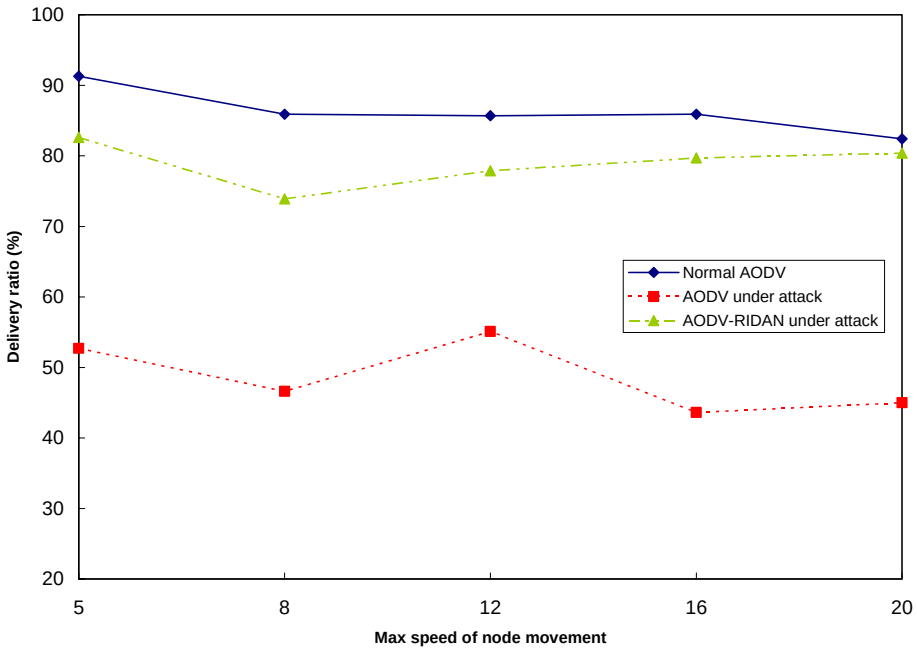

Figure 7.26: Delivery ration versus node mobility in the resource consumption attack.

The second metric that is used in the evaluation of the RIDAN system is the routing packets dropped ratio. The reason that this metric was used is to show that the RIDAN system increases the dropping routing packet ratio to 34% while in normal AODV and in AODV under the resource consumption attack is approximately 1.5%. AODV by default drops very few routing packets; however the resource consumption attack takes advantages of this behavior to flood the network with unnecessary routing traffic. Thus, as it is illustrated in the following two figures, the fact that the routing dropping rate is high has a positive impact in the overall performance of the network when a flooding attack is in process.

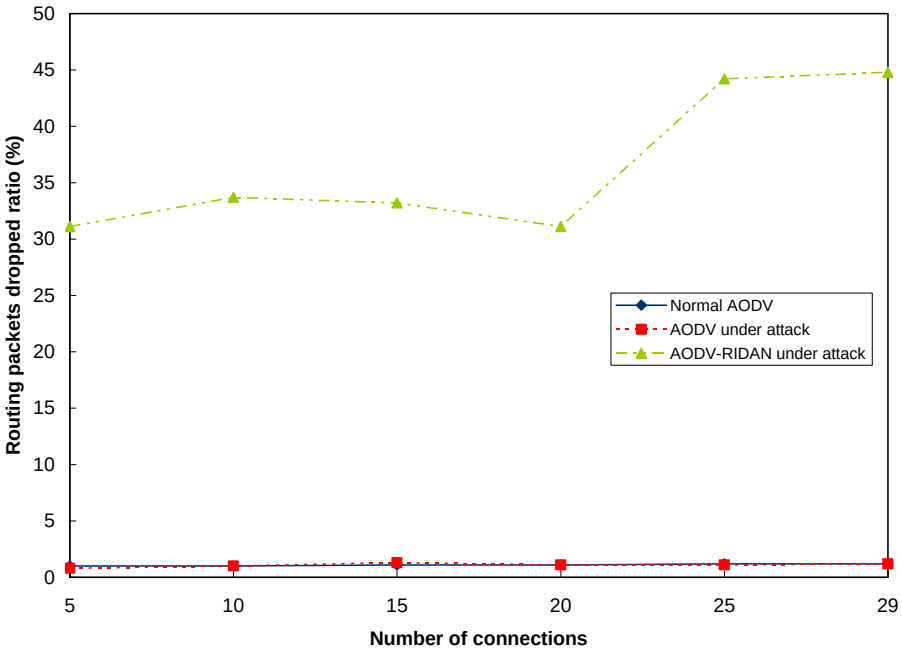

Figure 7.27: Routing packets dropped ratio versus number of connections.

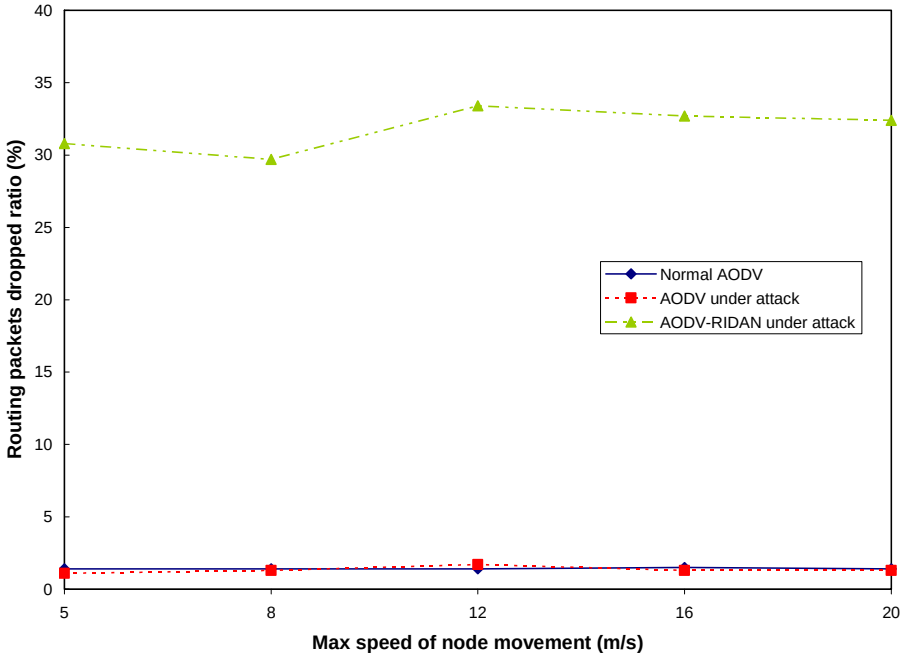

Figure 7.28: Routing packets dropped ratio versus node mobility.

7.6 Accuracy of the RIDAN system

All intrusion detection systems suffer from false alarms that occur whenever the system incorrectly concludes in an alarm but there is no malicious behaviour present in the network. The knowledge-based intrusion technique that the RIDAN system utilises to detect malicious activity is less error prone than other intrusion techniques, like for example the behaviour-based intrusion detection approach. However, the traffic patterns that denote that an active attack is performed against the routing protocol can be realised when the AODV operates normally due to high application traffic and high node mobility. The RIDAN system was tested in terms of detection accuracy and the percentages of detection accuracy for the three attacks are the following:

- Sequence number attack detection accuracy: 81.2%.
- Dropping routing packets attack detection accuracy: 71.5%.
- Resource consumption attack detection accuracy: 74.8%.

The detection accuracy of the system in all the three attacks can be considered high compared to results of other similar projects [YELC01, OS02, CWJ01].

7.7 Conclusions and Further Work

When we decided to go through with this project there was not any solid base that could guarantee that the system would have positive results, or even that any meaningful results could be drawn from this research in general. Even though the field of intrusion detection in wired networks has been thoroughly researched [MHL94, Fra94, CH96], the area of intrusion detection in wireless network has very little to show in research and related studies [PW02, MGLB00, BA01]. The RIDAN system is one of the few pure intrusion detection components developed for wireless ad hoc networks and the only one that provides real-time behaviour with the use of timed finite state machines. Although the idea of intrusion detection with the use of finite state machines is not novel by itself, the RIDAN system gains its novelty from being the first intrusion detection system that utilises timed finite state machines in wireless mobile ad hoc networks.

The challenges that were faced during the design and the development of the RIDAN system were many. Despite that fact that there are many research papers that claim to have implemented similar active attacks using the network simulator, the information that is available in the papers and on the Internet is minimal raising suspicion on whether or not they have actually correctly implemented these attacks. Additionally, the network simulator and the CMU extensions for ad hoc networks [NS02] that include the AODV were not flexible in usage and in modification. For

that reason the malicious behaviours and the RIDAN-enabled AODV (RIDAN-AODV) were implemented with rogue routing agents and many modifications in the internal files of the network simulator had to be made. Furthermore, the patterns that denote a specific malicious behaviour had to be first proven theoretically to decide whether it is feasible to design traffic patterns that can uniquely identify a specific malicious behaviour. This proved to be harder than it first appeared to be since it was proven that some attacks that involve node impersonation (a node uses another node's address to perform an attack) could not be identified without the use of a cryptographic mechanism.

As it was presented in the evaluation of the RIDAN system, the developed intrusion detection mechanism manages to detect the active attacks that were specified with high accuracy and keeps the network performance within acceptable limits. The RIDAN system operates currently for the AODV routing protocol, however it does not alter any of its fundamental operational functions. It also provides a lightweight mechanism to ensure protection from malicious activities performed against the routing fabric. However, the RIDAN system is not a complete security solution since for the moment it only provides protection from the three specific attacks analysed in the previous sections. The possible directions for future work on the RIDAN system are summarised below:

- The RIDAN system can be further extended to provide security from more active attacks that a malicious node can perform against the routing protocol.
- The dropping routing packets attack, the resource consumption attack and the RIDAN detection mechanisms for these attacks can theoretically work as they are for DSR and OLSR. However, since this was not investigated some changes in the detection patterns may be required.
- The RIDAN system could be also be extended to operate for proactive routing protocols like DSDV. The main architecture and the high level components of the RIDAN system will remain the same and the only things that should change are the patterns that signify the attacks and of course the threshold values should be modified to mach the implementation of the underlying protocol.
- The RIDAN system could be also extended to include some cryptographic mechanism like a certification authority that would prevent nodes from impersonating other nodes. This would provide a more complete security solution however it would introduce additional overhead.
- Another thing that could be considered future work is to implement and test the RIDAN system in a real ad hoc network environment. To re-implement the RIDAN system for

real AODV it is not expected to be very difficult since the main components will remain the same and the traffic patterns that signify the attacks will also be the same. The interesting part of this work would be to observe how the real network behaves when it is under attack and of course to measure the performance of the network when the RIDAN system is in operation.

7.8 Summary

In this final chapter of this study the evaluation of the RIDAN system was presented. The RIDAN system is a novel lightweight system that detects and takes countermeasures against active attacks that can be performed against the AODV routing protocol in mobile ad hoc networks. Although it does not provide protection from all possible active attacks, the RIDAN system can be further extended to protect the ad hoc network from more active attacks. In this chapter there were proposed some further extensions that could be implemented and make the RIDAN system a complete security component that could be employed for securing ad hoc networks. The system as it operates does not introduce any changes in the underlying protocol and therefore it can be applied unmodified to other reactive protocols like DSR and OLSR.

Bibliography

[ACP+02] P. Albers, O. Camp, J. M. Parcher, B. Jouga, L. Me, R. Puttini, "Security in Ad Hoc Networks: a General Intrusion Detection Architecture Enhancing Trust Based Approaches", The 1st International workshop on Wireless Information Systems" (WIS 2002), in the 4rth International Conference on Enterprise Information Systems, 2002.

[AFV95] D. Anderson, T. Frivold, A. Valde, “Hext Generation Intrusion Detection Expert System (NIDES): A summary”, Technical Report, Computer Science Laboratory, SRI International, 1995.

[BA01] S. Bhargava, D. P. Agrawal, “Security Enhancements in AODV protocol for Wireless Ad hoc Networks”, in IEEE Semi-annual Proceedings of Vehicular Technology Conference (VCT’01), 2001.

[Bac00] R. G. Bece, “Intrusion Detection”, Macmillan Technical Publishing, 2000.

[Bha94] Vaduvur Bharghavan, "Secure Wireless LANs", in Proceedings of ACM conference on Computer and Communications Security, November 1994, pp 10-17.

[CE89] M. Corson and A. Ephremides, "A distributed routing algorithm for mobile radio networks", in Proceedings of Military Communication Conference, 1989.

[CH96] J. Cannady, J. Harrell, “A Comparative Analysis of Current Intrusion Detection Technologies”, in Proceedings of the 4th Conference on Technology for Information Security, (ISC’96), 1996.

[CJL+01] T. Clausen, P. Jaquet, A. Laouti, P. Minet, P. Muhlethaler, A. Quyyum, L. Viennot, “Optimized Link State Routing Protocol”, Internet Draft, draft-ietf-manet-olsr-06.txt, work in progress, September 2001.

[CM99] S. Corson, J. Macker, “Mobile Ad hoc Networking (MANET): Routing Protocol Performance Issues and Evaluation Considerations”, Request For Comments (RFC) 2501, January 1999.

[CWJ01] H. –Y. Chang, S. F. Wu, Y. F. Jou, “Real-Time Protocol Analysis for Detecting Link-State Routing Protocol Attacks”, ACM transactions on Information and System Security, vol. 4, no.1, pp. 1-36, February 2001.

[DBRS01] B. Dahill, B. N. Levine, E. Royer, C. Shields, “A Secure Routing Protocol for Ad hoc Networks”, Technical report, UM-CS-2001-037, University of Massachusetts, August 2001.

[DDW98] H. Debar, M. Dacier, A. Wespi, “Towards the Taxonomy of Intrusion-Detection Systems” Technical Report, IBM Zurich Laboratory, 1998.

[Den87] D. E. Denning, “An Intrusion Detection Model”, IEEE Transactions in Software Engineering, vol. 13, no2, February 1987.

[DLRS01] B. Dahill, B. N. Levine, E. M. Royer, C. Shields, “A Secure Routing Protocol for Ad hoc Networks”, Technical Report, UM-CS-2001-037, University of Massachusetts, August 2001.

[Dro97] R. Droms, “Dynamic Host Configuration Protocol”, Request For Comments (RFC) 2131, March 1997.

[Fra94] J. Frank, “Artificial Intelligent and Intrusion Detection: Current and Future Directions”, in Proceedings of the 17th National Conference on Computer Security, October 1994.

[Has97] Z. J. Hass, “The Zone Routing Protocol (ZRP) for Ad hoc Networks”, IETF Internet Draft, draft-zone-routing-protocol-00.tct, work in progress, November 1997.

[HHB03] A. Habib, M. H. Hafeeda, B. Bhargava, “Detecting Service Violation and DoS Attacks”, in Proceedings of Network and Distributed System Security Symposium (NDSS), 2003.

[HJP02] Y. –C. Hu, D. B. Johnson, A. Perrig, “SEAD: Secure Efficient Distance Vector Routing for Wireless Ad hoc Networks”, in Proceedings of the 4th IEEE Workshop on mobile Computing Systems and Applications (WMCSA’02), pp. 3-13, June 2002.

[HLMS90] R. Heady, G. Luger, A. Maccade, M. Servilla, “The architecture of a Network Level Intrusion Detection System”, Technical report, Computer science Department, University of New Mexico, August 1990.

[HPJ02] Y. –C. Hu, A. Perrig, D. B. Johnson, “Ariadne: A Secure On-demand Routing Protocol for Ad hoc Networks”, in Proceedings of the 8th ACM International Conference on Mobile Computing and Networking (MobiCom’02), pp. 12-23, September 2002.

[IDG91] J. Ioannidis, D. Duchamp, J. M. Gerald, “IP Based Protocols for Mobile Internetworking”, ACM SIGCOMM Computer Communication Review (SIGCOMM’91), pp. 234-245, September 1991.

[Iee97] IEEE Computer Society LAN/MAN Standars Committee, “Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications”, IEEE std. 80211-1997. IEEE, New York, NY 1997.

[JMHJ02] D. B. Johnson, D. A. Maltz, Y-C Hu, J. G. Jetcheva, “The dynamic Source Routing Protocol for Mobile Ad hoc Networks (DSR), Internet Draft, draft-ietf-manet-dsr-07.txt, work in progress, February 2002.

[Kar00] V. Karpijoki, “Security in Ad hoc Networks”, In Proceedings of the Helsinki University of Technology, Seminars on Network Security, Helsinki, Finland 2000.

[KV02] K. Fall, K. Varadhan, “The ns Manual (formerly ns Notes and Documentation), April 2002, http://www.isi.edu/nsnam/ns-ducumentation.html.

[Lun00] J. Lundberg, “Routing Security in Ad hoc Networks”, http://citeseer.nj.nec.com/400961.html.

[Mal98] G. Malkin, “Routing Information Protocol (RIP) Version 2”, Request For Comments (RFC) 2453, November 1998.

[MGLB00] S. Marti, T. J. Giuli, K. Lai, M. Baker, “Mitigating Routing Misbehaviour in Mobile Ad hoc Networks”, in Proceedings of the 6th Annual ACM/IEEE international Conference on Mobile Computing and Networking, pp. 255-265, 2000.

[MH96] C. Heitmeyer, D. Mandrioli, “Formal Methods for Real-Time Computing”, John Wiley & son, 1996.

[MHL94] B. Mukherjee, L. T. Heberlein, K. N. Levitt, “Network Intrusion Detection”, IEEE Network, vol. 8, no. 1, January 1994.

[Moy98] J. Moy, "OSPF Version 2", Request For Comments (RFC) 2328, April 1998.

[Nis03] National Institute of Standards and technology, “Definition of finite state machines”, http://www.nist.gov/dads/HTML/finiteStateMachine.html 2003.

[NS02] CMU extensions for ns-2, “http://www.isi.edu/nsnam/ns/”, September 2002.

[OS02] Y. Okazaki, I. Sato, “A New Intrusion Detection Method based on Process Profiling”, in Proceedings Symposium on Applications and the Internet, (SAINT‘02), pp. 82-90, 2002.

[PB94] C. E. Perkins, P. Bhagwat, “Highly Dynamic Destination-Sequenced Distance-Vector Routing (DSDV) for Mobile Computers”, in Proceedings of the SIGCOMM ’94 Conference on Communications Architectures, Protocols and Applications, August 1994.

[Per00] R. Perlman, “Interconnections: Bridges, Routers, Switches and Internetworking Protocols”, 2nd Edition, Addison-Wesley, 2000.

[Per01] C. E. Perkins, “Ad hoc Networking”, Addison-Wesley, 2001.

[PH02] P. Papadimitratos, Z. J. Haas, “Secure Routing for Mobile Ad hoc Networks”, in Proceedings of the SCS Communication Networks and Distributed Systems, Modelling and Simulating Conference (CNDS’02), pp. 27-31, January 2002.

[PH02] P. Papadimitratos, Z. J. Hass, “Securing the internet Routing Infrastructure” IEEE Communications, Vol. 40, No, 10, October 2002.

[PHO02] D. D. Perkins, H. D. Hughes, C. B. Owen, "Factors Affecting the Performance of Ad Hoc Networks", in Proceedings of the IEEE International Symposium on Performance Evaluation of Computer and Telecommunication System, San Diego, July 2002.

[PR03] C. Perkins, E Belding-Royer, “Ad hoc On-demand Distance Vector (AODV)” Request For Comments (RFC) 3561, July 2003.

[PW02] K. Paul, D. Westhoff, “Context Aware Detection of Selfish Nodes in DSR based Ad hoc Networks”, in IEEE Semi-annual Proceedings of Vehicular Technology Conference (VCT’02), 2002.

[RT99] E. Royer, C-K Toh, "A Review of Current Routing Protocols for ad-hoc Mobile Wireless Networks", IEEE personal communications, Vol. 6, no 2, pp.46-55 April 1999.

[SGF+02] R. Sekar, A. Gupta, J. Frullo, T. Shanbhag, A. Tiwary, H. Yang, S. Zhou, "Specification-based Anomaly Detection: A new approach for Detecting Network Intrusions", in Proceedings of the 9th ACM Conference on Computer and Communication Security, pp. 265-274, 2002.

[Sta02] F. Stajano, "Security for Ubiquitous Computing", Wiley, 2002.

[VIN03] The VINT (Virtual InterNetwork Testbed) Project homepage, http://www.isi.edu/nsnam/vint/index.html

[WLB03] W. Wang, Y. Lu, B. K. Bhargava, "On Vulnerability and Protection of Ad Hoc On-demand Distance Vector Protocol", to appear in International Conference on Telecommunications (ICT'2003), France, 2003.

[YELC01] N. Ye, S. M. Emran, X. Li, Q. Chen, "Statistical Process fro Computer Intrusion Detection", in Proceeding in DARPA Information Survivability Conference and Explosion (DISCEX'01), pp 3-14, 2001.

[YNK01] S. Yi, P. Naldurg, R. Kravets, "Security-aware Ad hoc Routing for Wireless Networks", in Proceedings of the 2nd ACM Symposium on Mobile Ad hoc Networking and Computing (MobiHoc'01), pp. 299-302, October 2001.

[ZA02] M. G. Zapata, N. Asokan, "Secure Ad hoc On-demand Distance Vector Routing" ACM Mobile Computing and Communications Review, Vol 6, no 3, July 2002.

[ZH99] L. Zhou, Z.J. Haas, "Securing Ad hoc Networks", IEEE, Networks Magazine, Vol. 13, no 6, November/December, 1999.

[ZL00] Y. Zhang, W. Lee, “Intrusion Detection on Wireless Ad hoc Networks”, in Proceedings 6th Annual International Conference on Mobile Computing and Networking (MobiCom’00), August 2000.