SicAri – A security architecture and its tools for ubiquitous Internet usage

Deliverable PE6

Secure Routing Mechanisms

Version 2.0, March 26, 2007

Peter Ebinger, Fraunhofer IGD
Matthias Hollick, TUD-KOM
André König, TUD-KOM
Jan Peters, Fraunhofer IGD

Contents

1 Introduction

> Trees have roots, men and women have legs, with which to traverse the barbed-wire idiocy of frontiers, with which to visit, to dwell among mankind as guests.
>
> *George Steiner*

1.1 Motivation

Within this document, we analyze how communication paths are set up in *Mobile Ad hoc Networks (MANETs)* and especially on how this procedure can be improved with security. We assume that no one disagrees, when we claim that communication is needed to be secure in general; so what is left, is to motivate the examination of MANETs and of challenges for routing and security in this context. This shall be done for each of the terms mobile and ad hoc which classify the networking paradigm of MANETs.

Mobile Mobility has become an important feature of communication during the last decade. Besides the enormous growth of cellular networks, as used for cell phones, mobility also established in the field of computer networks.

What does this mean for a communication network? There are in fact many consequences and even more publications addressing these. Here, we set the focus on two of those aspects, out of which the first one is routing. Since mobility generates dynamic environments, the conventional routing algorithms used in and optimized for (more or less) static networks can not be deployed in MANETs. We therefore discuss new routing paradigms and take a closer look on possible attack mechanisms related to these.

The second aspect of mobility is that information is transmitted using a shared medium which is freely accessible by anyone, since mobile devices have to make use of wireless communication. This clearly provides a challenge with respect to security. As with any other radio signals, everyone can tune in the respective frequency and just listen along. Avoiding this is hardly possible, but we present an approach that copes with this challenge.

Ad hoc Current networks depend on infrastructures. Well known examples are networks for cell phones or the upcoming wireless LAN hotspots. In contrast, ad hoc networks omit any infrastructural components. In order to set up a communication between two devices, every other device in the MANET is in duty to forward messages. Do you remember how you sent a letter to your classmate five seats away from yourself in primary school? MANETs follow the same principle.

Again, we pose the question, what this means for a communication network. And again there are many consequences, out of which we concentrate on routing and security. Like mobility, ad hoc communication states a challenge for routing, since there are no central instances which are in duty of this task. Several approaches for routing algorithms in MANETs have been proposed. The most common are presented in this document.

In terms of security, building a network in an ad hoc way exhibits new possibilities for attacks. Since every device forwards messages of other devices, every (potentially malicious) device is able to take influence on this process and therefore may tamper the functionality of a large part of the network. Since every device forwards messages of other devices, every (potentially malicious) device is able to listen in without having the need to be in direct radio range of sender or receiver.

Now that we have pointed out challenges concerning routing and security that arise in mobile ad hoc networks, one could ask why we want to deploy such networks and not make life easier and use infrastructure based networks. We want to provide three answers.

The first answer is: There are scenarios, in which the establishment of a communication infrastructure is not possible. Catastrophes or battlefields are often mentioned in this context. Communication is needed to coordinate actions which can not wait for an extensive infrastructure to be established.

The second answer is: MANETs are cheaper. Setting up and maintaining a communication infrastructure produces high costs (as one can see every month on the telephone bill).

The third answer is: For convenience. Think of (future) networked homes. A storage unit which provides audio and video files may be situated in the basement, an entertainment system which needs to access this files could be on the second floor. Perhaps the radio range of the storage unit is not sufficient to reach the entertainment system but it will reach a device on the first floor which can forward the requested information. The average user does not have the knowledge to set up an infrastructure and does not want to invest the time that would be necessary to learn it. Self configuring ad hoc networks for such an environment already start to get available.

1.2 MANET Properties

We now want to take a short, but closer and more technical look on the properties of MANETs.

We introduced effects of mobility on routing and security in the previous section. At this point, we want to give a definition of mobile communication in order to make a distinction to mobile devices. We do not consider the scenario of a notebook (a mobile device), that may well be carried around, but is plugged into a wired network or uses a modem for communication, as mobile communication. In MANETs, mobile communication means the exchange of information while the device is in motion. Thereby, the movement of the device is not restricted in space or time. So necessarily, communication in a MANET means wireless communication.

With respect to the ad hoc structure of a MANET in contrast to infrastructure based networks, we distinguish between end-systems and infrastructural components. In general, infrastructural components are used to set up and route all communication of the end-systems in the respective, infrastructure based network. A MANET consists only of end-systems, which we refer to as nodes in the following. In view of the routing process, nodes in an ad hoc network are treated equally. Every node has to route data packets that belong to the communication of other nodes. Since a MANET is a dynamic environment, we can not deploy routing mechanisms that have shown to perform in existing wired (and more or less static) networks. We give an overview of routing mechanisms that are able to cope with the respective challenges in MANETs. These mechanisms can (on the first level of abstraction) be categorized into two types. Pro-active routing algorithms, as the first one, find routes in an ad hoc network before (and whether or not) they are actually needed. In general, this is done in three steps. First, a node has to find out, which other nodes are in its direct radio range. In the second step, this information is sent to all other nodes in the network. Using this information, the current topology of the network (and therefore the routes between nodes) can be deduced in step three. The second type of routing algorithms is referred to as reactive mechanisms. Here, a route is not searched, until it is needed and requested. In general, the initiator of the communication sends the request throughout the network. The desired receiver sends back an answer upon reception of the request. The route is then learned from the trace of the path which the request and the answer traveled.

In view of the routing process, nodes in MANETs are treated equally, but in fact, a MANET is a highly heterogeneous environment. The range of the involved devices may reach from sensors with very specific functionality and restricted resources over cell phones to notebooks with high bandwidth and computing power. The speed of the devices may vary from zero to the speed of cars or trains. Heterogeneity also occurs with respect to transmission power of devices. This may lead to unidirectional edges in a routing graph, meaning that transmission of Device A may reach Device B, but not vice versa. Thus, the flow of the data packets through a MANET can be different, when a sender and a receiver change their roles.

For our MANET scenario, we additionally assume, that the position of (at least some) of the nodes can be determined. This can either be done by the nodes themselves (e. g. by using GPS), or by an intrusion detection system (e. g. by triangulation of the received signal strength).

1.3 Structure of the Report

This document continues with an overview of three common routing protocols for MANETs in Section 2. We present two reactive mechanisms, namely AODV routing which works in a distributed way, and DSR as a source based mechanism. From the field of pro-active protocols, we introduce the operating mode of OLSR.

Section 3 cover our application scenario. This is first described in detail, before we provide an analysis of security relevant issues. General security objectives, that is, which security related criteria a communication should meet (in our scenario), are described in Section 5.

In Section 5, we present several attack mechanisms that become possible in MANETs due to their aforementioned properties. A distinction is made between active and passive attack mechanisms. How those attacks can be identified and what can be done for prevention, is presented in Section 6. Sections 7 and 8 addresses the respective simulation setups, the used simulation tools, and the obtained results.

We conclude this document in Section 9 with an outlook concerning open topics of research and our future work in the area of MANETs. We further describe shortly, how we will proceed in the subsequent SicAri work package PE 7.

2 Routing Protocols

Due to the special characteristics of MANETs well-known routing mechanisms for computer networks are not directly applicable. The following two properties are the major goal for routing in mobile ad hoc networks:

- High responsiveness and fast reaction with rising mobility/dynamics
- Small (if possible in number of transmissions) overhead data arising for co-ordination

Routing protocols for MANETs can mainly be distinguished by their overall behavior They are partitioned into pro-active and reactive routing protocols.

Pro-active Routing Protocols Nodes, which use pro-active routing, always keep a table, which is the basis for their forwarding decisions. Therefore they are also called “table-based” (or “table driven”) routing protocols.

This table is checked in periodic time intervals for correctness and corrected if necessary with up-to-date values. The check of individual paths and the update produce additional data traffic for the coordination data between the different nodes. A periodic alignment of this data means at the same time that in the network constantly data communication takes place, without actual (user/application) data to be transferred.

Reactive Routing Protocols In contrary to the pro-active routing, reactive routing protocols maintain no table actively. They rather keep a cache, which was built up from past transmissions, which represents a table as well on which future forwarding decisions are based.

Routes that connect a source node with a destination are always only explored if they are actually needed for a data transmission. For this reason these routing protocols are also called “passive” or “on demand”.

Since route information is only conditionally exchanged when the need arises, reactive routing protocols produce substantially less coordination data than pro-active. Thus they are better suitable for less dynamic scenarios, in which changes of the network architecture do not appear as often and therefore fast reaction times are not so important.

The following sections introduce three common routing protocols for MANETs which are further investigated in the next sections with a focus on possible attacks and security countermeasures. Two reactive mechanisms, AODV routing and DSR are described as well as the pro-active routing protocols: OLSR.

These three protocols are the most important representatives of the two routing protocol classes. They are actively developed and maintained by the IETF and have been releases as RFC (OLSR [13, 29]) or Internet draft (DSR). Geographic-based routing is investigated in the following sections as well. However the geographic positions are not used to reduce the routing overhead but to increase the security of the underlying protocols.

2.1 Ad hoc On-demand Distance Vector Routing (AODV)

The following introduces the basics of the *Ad hoc On-demand Distance Vector Routing Protocol (AODV)* [29, 30, 11]. AODV is based (as the name suggests) on the distance-vector principle. Each node manages vectors for all kinds of targets with each vector containing information including the distance to the target as well as which node is required for reaching the target.

Like most reactive routing protocols AODV is based on a broadcast mechanism for route finding. AODV comprises three phases:

- Route Discovery
- Route Maintenance
- Route Deletion

Each phase is described in more detail in the following sections.

2.1.1 Route Discovery

Route discovery becomes active when a node (source S) wants to establish a connection to another node (target D) but does not yet know a route to this target. A *Route Request (RREQ)* is sent via broadcast to the network containing a `BroadcastID` and two sequence numbers (one newly generated and if available the last known number of the target).

To enable the construction of both backward and forward paths later in the transmission, addresses of sender and target, the `BroadcastID` and the sequence numbers are all stored. Figure 1 shows a simplified representation of the process of route discovery. It was assumed that only directly adjacent nodes are within transmission range of a node.

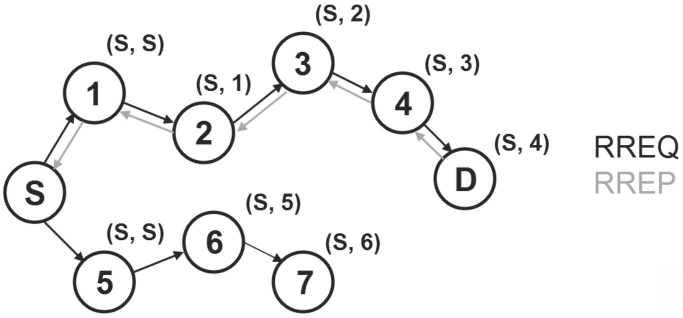


Figure 1: Route Discovery in AODV

Notice that node S first transfers the route request to nodes 1 and 5 within reach. These nodes know S as a direct neighbor and transfer the RREQ packet to their other neighbors 2 and 6. Node 2 memorizes that node S can be reached via node 1 as shown by the values in brackets. Other nodes act accordingly and the process is continued until the route request finally reaches its target.

At this point a *Route Reply (RREP)* can be sent back to source S via the backward path. Once sent S now knows how D can be reached. At the same time the other nodes involved in the route store the relevant routing information with the sequence number of the target. This sequence number can serve to shorten a route request possibly initiated by another node if it already knows a current route to the target. It then directly sends a route reply back to the source.

2.1.2 Route Maintenance

AODV now enters the second phase, maintaining a valid route after a successful route has been established as a result of the first phase. AODV uses two timers or time-out mechanisms during this phase. The first keeps the backward route open for a set period of timed inactivity after which the entry for this routing table is canceled from the corresponding table. This time-out must be long enough for a RREP packet to reach source S via the backward path held in the routing tables. The second timer is used to cancel forward paths that are no longer needed from the routing tables. If a particular entry of a routing table is not used for a certain period defined by a time-out this entry is removed.

2.1.3 Route Cancellation

The third phase of AODV (the so-called Route Error) serves to recognize faulty routes and facilitate their elimination. If a node S can no longer reach his target D using a certain route, a *Route Error (RERR)* message is sent to the source.

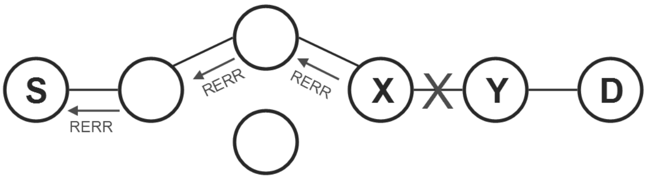


Figure 2: Route Error in AODV

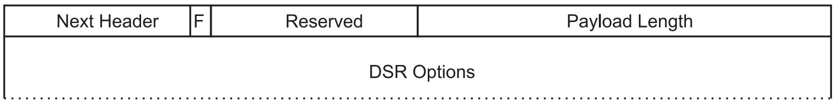
Figure 2 shows what happens if the connection between the nodes X and Y breaks down. If node X recognizes the fault it initiates the return of a RERR to source S. The route that has become void can thus be canceled from the routing tables of the nodes on the path to S.

The sequence number for target D was raised before node X has sent the RERR message. Node S can then, after receiving the RERR packet, initiate a new route discovery for D with the new sequence number.

2.2 Dynamic Source Routing (DSR)

In this section, we outline the operating mode of DSR [23], as the second reactive routing protocol for MANETs which we present in this document. While reading this section, one will recognize, that the basic mechanisms of DSR are similar to those of AODV. The main difference between the two protocols is that DSR is a source based mechanism while AODV works in a distributed way. In an AODV based MANET, every node only knows the next hop for the transmission of a message on its way from sender to receiver. When using DSR, the whole path is determined by the sender and appended to each message.

In the following, we describe the the phases of route discovery, data transfer, and route maintenance for the DSR protocol. Each of these phases relies on different DSR options, that are appended to a fixed portion of the DSR header, which is shown in Figure 3.



Next Header	F	Reserved	Payload Length
DSR Options			

Figure 3: Static part of the DSR header

The fields of this part of the DSR header are to be interpreted as follows:

Next Header Identifies the header that follows the DSR header, such as for example TCP or UDP.

F Used in combination with DSR flow state operation.

Reserved Not used in current versions. Must be set to zeros.

Payload Length Length of the DSR Options which are appended to the fixed part of the header.

Options DSR Options such as for example route request option, route reply option, or source route option. To be further described in detail.

The DSR deader including the DSR options directly follows the header of the network layer as shown in Figure 4.

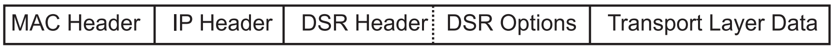


Figure 4: Position of DSR header and options

2.2.1 Route Discovery

The route discovery phase of DSR is initiated each time a packet for which the route to the desired destination is unknown arrives at the network layer of the originating node. A route request option as shown in Figure 5 is appended to the DSR header. The resulting packet (a route request) is sent via IP broadcast throughout the network.

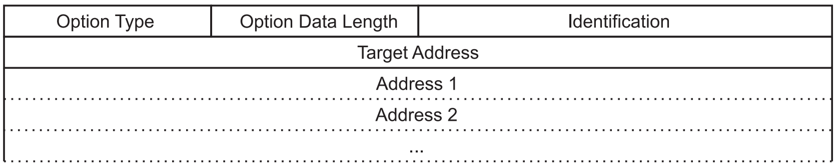


Figure 5: DSR route request option

The fields of the route request option are used as follows:

Option Type Identifies the option. For the route request option, this field is set to 1.

Option Data Length The length of the route request option (without option type and option data length fields).

Identification A sequence number, that uniquely identifies this route request.

Target Address Network layer address of the receiver, for which a route is to be discovered.

Address n Network layer address of the n^{th} node, that receives and forwards this route request.

To discover the route, each node that receives a route request appends its own address to the list of addresses in the route request option. Exemplarily, this should be shown for the network topology depicted in Figure 6, where node A acts as sender and node D as receiver.

If the route to node D is unknown to A, in the first step node A generates a route request message. A appends its own network layer address (`0.0.0.1` in our example) to the address list and sets D's address (`0.0.0.4`) as the target address. The resulting route request option is shown in Figure 7.

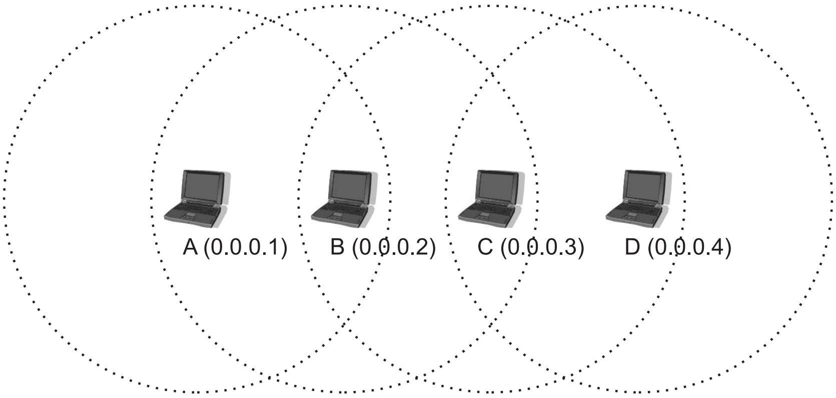


Figure 6: Example topology for DSR route request

00000001	00001010	0101010101010101
00000000 00000000 00000000 00000100		
00000000 00000000 00000000 00000001		

Figure 7: Route request as sent by node A to discover D

00000001	00001010	0101010101010101
00000000 00000000 00000000 00000100		
00000000 00000000 00000000 00000001		
00000000 00000000 00000000 00000010		
00000000 00000000 00000000 00000011		

Figure 8: Route request as received by node D

This route request message is forwarded by nodes B and C to the target node D. B and C append their network layer addresses (`0.0.0.2` and `0.0.0.3`) to the address list of the route request option. The route request option as it arrives at node D is given in Figure 8.

When node D receives the route request from A, there are several possibilities how to react. If the underlying MAC protocol depends on bidirectional radio links (like 802.11), D sends a route reply message to A by reversing the route that is contained in the route request. If bidirectional links can not be assumed, D has to start a route discovery for A. The route reply is piggybacked to D's route request. In this case, the route reply (and following messages from D to A) may take a different path through the network than the route request (and messages from A to D).

The structure of the DSR route request option is shown in Figure 9. Figure 10 depicts the route reply option for our example scenario (assuming bidirectional links). The fields of the route reply option are used as follows:

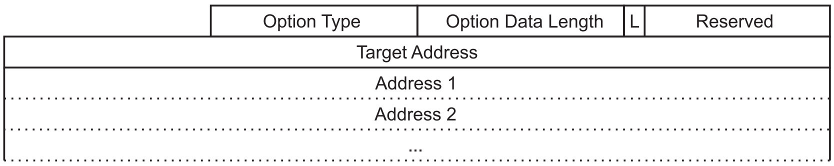
Option Type Identifies the option. For the route reply option, this field is set to 2.

Option Data Length The length of the route reply option (without option type and option data length fields).

Last Hop External (L) Denotes whether this route continues in another routing domain (e. g. AODV).

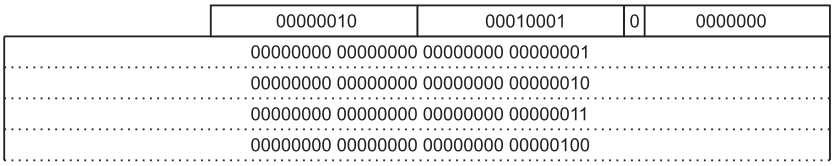
Reserved Not used in current versions. Must be set to zeros.

Address n Network layer address of the n^{th} node in the discovered route from sender (1) to receiver (n).



	Option Type	Option Data Length	L	Reserved
Target Address				
Address 1				
Address 2				
...				

Figure 9: DSR route reply option



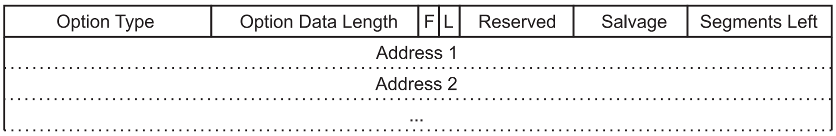
	00000010	00010001	0	0000000
00000000 00000000 00000000 00000001				
00000000 00000000 00000000 00000010				
00000000 00000000 00000000 00000011				
00000000 00000000 00000000 00000100				

Figure 10: Route reply as sent by node D

In the current draft, three possible extensions for the route discovery phase are proposed, but not mandatory. The first one is a cache, in which routing information, that bypasses a node could be temporarily stored for future use. Secondly, nodes could reply to route requests using the cached information on routes. In our example, this means that if node B already knows a route to node D, B could for reasons of performance directly answer the route request of A. The third possible extension is to limit the propagation of route request messages by using the IP TTL field. By increasing this value on unsuccessful route request, an expanding ring search can be implemented.

2.2.2 Data Transfer

Once a route from the source node to the desired destination has been found, DSR enters the Data transfer phase. Every packet that is to be sent from source to destination carries the route from sender to receiver. For this, the source route option of DSR is appended to the DSR header. Figure 11 shows the structure of this option.



Option Type	Option Data Length	F	L	Reserved	Salvage	Segments Left
Address 1						
Address 2						
...						

Figure 11: DSR source route option

The fields of the source route option are used as follows:

Option Type Identifies the option. For the source route option, this field is set to 96.

Option Data Length The length of the route reply option (without option type and option data length fields).

First Hop External (F) Denotes whether this route originates in another routing domain (e. g. AODV).

Last Hop External (L) Denotes whether this route continues in another routing domain (e. g. AODV).

Reserved Not used in current versions. Must be set to zeros.

Salvage Counts how often this packet was salvaged. To be described in Section 2.2.3.

Segments Left Denotes how many nodes are left in the source route until the packet reaches its destination.

Figure 12 depicts the source route option as sent by node A from our example out of the previous section, where data should be sent from node A to node D.

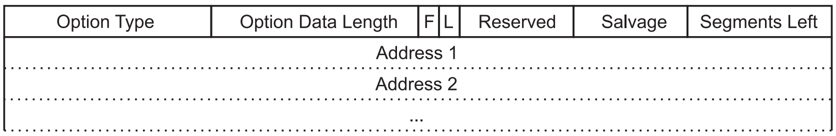


Figure 12: Source route option for data transfer from A to D

2.2.3 Route Maintenance

Since a mobile as hoc network is a highly dynamic environment where the topology may change rapidly, it is necessary, to (periodically) check whether a discovered route is still valid. DSR performs this check by requesting acknowledgments for transmitted data. These acknowledgments may either be provided by the underlying MAC protocol such as 802.11 or if not provided by lower layers, through DSR itself. Since our simulations are based on an 802.11 MAC layer, the DSR acknowledgment option should not further be described here. It is important to notice, that acknowledgments are requested hop by hop. In our example out of Section 2.2.1 this means that nodes A, B, and C expect acknowledgments from nodes B, C, and D, respectively.

If no acknowledgment is received after a packet is forwarded, the link is identified as broken. In this case, a route error message is sent to the originator of the packet. The structure of the route error option is shown in Figure 13. The fields of the route error option are used as follows:

Option Type Identifies the option. For the route error option, this field is set to 3.

Option Data Length The length of the route reply option (without option type and option data length fields).

Error Type Denotes the type of error that was detected, as for example an unreachable node (1).

Reserved Not used in current versions. Must be set to zeros.

Salvage Counts how often this packet was salvaged. To be described later in this section.

Error Source Address Network layer address of the node which detected a broken link.

Error Destination Address Network layer address of the which originated the data packet that should have been transmitted.

Error Specific Information Further information on the error that caused this message, as for example the address of a node that was detected to be unreachable.

Option Type	Option Data Length	Error Type	Reserved	Salvage
Error Source Address				
Error Destination Address				
Error Specific Information				

Figure 13: DSR route error option

Let us assume, that in our example the link between nodes B and C is broken. The resulting route error message, that will be sent from B to A is shown in Figure 14.

00000011	00000101	00000001	0000	0000
00000000 00000000 00000000 00000010				
00000000 00000000 00000000 00000001				
00000000 00000000 00000000 00000011				

Figure 14: Route error option as sent from B to A

As it holds for the route discovery phase, the current draft of DSR proposes several extensions for the route maintenance out of which two shall shortly be explained here. The first extension is the so called packet salvaging mechanism. Here, if a node recognizes a broken link but has knowledge of another route to the destination of the respective packet, it reroutes the packet by changing its source route option. The salvage counter is used to prevent loops when a packet is salvaged more than once. The second extension to the basic route maintenance mechanism is automatic route shortening. If in our example, node A can hear node C forwarding A's packets, it knows that its transmission could also reach C. In this case, node B is not needed anymore for forwarding A's packets and can be removed from the source route.

2.3 Optimized Link State Routing (OLSR)

The routing protocol *Optimized Link State Routing (OLSR)* [13, 11] is a pro-active protocol basing on the link-state principle. Compared to variants for wired networks it is adapted in some parts to the different requirements in MANETs.

Each node autonomously gathers the necessary information for the route calculation within the network. So each node is able at any time to build a complete graph of the network and, using it, to find a route from source to destination. This directly results in the important advantage that well-known problems like emerging loops or the so-called count-to-infinity problem can be avoided. Transmission decisions are, however, made locally, each node deciding autonomously to which neighbor a packet shall be transmitted on its way to the destination. So each data packet does not carry the complete route but just a small table containing information about the destination and the maximum number of hops allowed on the path to the destination.

Since OLSR is a (pro-)active routing protocol and therefore routing information is permanently exchanged a classification into stages as with reactive routing protocols such as AODV is difficult. The identification of valid routes is mainly based on the steps mentioned in the following two paragraphs.

2.3.1 HELLO Messages – Locate Neighbors

OLSR specifies that so-called HELLO messages are sent at regular time intervals. These messages serve to detect connections, to identify neighbors, and to signal the *Multi-point Relay (MPR)* (see below) and contain the following information:

`HTime` Time interval between which the node sends HELLO messages.

`Willingness` Willingness of a node to forward data for other participants of the network. This is an integer value in the range of 0 to 7 which mirrors the willingness of the node.

Furthermore several blocks which appear for each neighbor node of the originator of the HELLO message and bear the following information:

`Link Code` Type of connection to the neighbor node (Is there a symmetric or asymmetric connection? Is the neighbor also MPR of the sender? Is the connection canceled?)

`Neighbor Interface Address` Address of an interface of the particular neighbor. Each node can be connected to the OLSR network with several interfaces with different addresses.

This way each recipient of such messages is informed about the neighbors in reach. By the transmission time identified for each packet a node can calculate the time distance to each neighbor and store it in a local table.

2.3.2 TC Messages – Distribute Neighbor Information

If a node observes a change in his direct neighborhood, i. e. if a new neighbor is added or an existing one departs, this information about the changed network topology is forwarded to all neighbors in reach, i. e. a node that is aware of changes distributes a packet with the corresponding information. In particular, this so-called *Topology Control (TC)* message contains the following data:

`Advertised Neighbor Sequence Number (ANSN)` Unique sequence number generated by the sender. With this number recipients can decide if a TC message received is up to date or possibly out-dated.

`Advertised Neighbor Main Address` Besides the sequence number the TC message contains an entry with the address of each neighbor of the sender of the message. In this way each node of the network first gets to know not only its own neighbors but also the adjacent nodes of these neighbors – the so-called 2-hop neighborhood (in the following called $N2$ in short).

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1

+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|          Reserved             |     Htime     |  Willingness  |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|   Link Code   |   Reserved    |       Link Message Size       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                  Neighbor Interface Address                   |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                  Neighbor Interface Address                   |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
:                             .  .  .                           :
:                                                               :
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|   Link Code   |   Reserved    |       Link Message Size       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                  Neighbor Interface Address                   |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                  Neighbor Interface Address                   |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
:                                                               :
```

Figure 15: Format of the HELLO message in OLSR [13]

The recipients of this message will then forward this new information by flooding.

The so-called flooding works in a way that all arriving packets which have not yet been forwarded are sent to all neighbors within reach. This means that the unique sequence number contained in each packet is checked to avoid that packets are endlessly passed on and on and block the network. Then each network node can store a network graph represented as a routing (or forwarding) table based on the information received.

Networks with a high mobility and low bandwidth can expose problems using OLSR. An enormous amount of data has to be constantly sent due to the changes in the neighborhood of nodes. This data amount can soon overload the whole network. This is why OLSR implicates –compared to the underlying link-state routing – a significant improvement of the flooding mechanism.

2.3.3 Optimized Flooding using Multipoint Relays

In order to prevent an excessive flooding of the network the set of all direct neighbors of each node is subdivided into two subsets. On the one hand, there are the MPR nodes, on the other hand, just those nodes which do not belong to the set of MPRs. All direct neighbors receive and process the messages of the sender, but only the selected MPRs forward them. The basic rule is:

For each neighbor n at a distance of 2 hops at least one MPR m must exist such that n can be reached via m. The number of selected MPRs shall be minimal.

Figure 17 illustrates this.

It is important to note that the selection of the MPRs is done automatically during the exchange of the HELLO messages, so that there is no additional overhead. Each node stores the list of

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|              ANSN             |           Reserved            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|               Advertised Neighbor Main Address                |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|               Advertised Neighbor Main Address                |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                              ...                              |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 16: Format of the TC message in OLSR [13]

neighbors that have selected it as MPR in a table, the so-called “MPR selector set”. The list of the node’s own MPRs is the so-called “MPR set”.

2.3.4 Algorithm for MPR Selection

The algorithm for the MPR selection proposed in the OLSR RFC [13] is as follows:

1. Start with an MPR set which includes all nodes of the direct neighborhood (in short N) with `N_willingness = WILL_ALWAYS`. These are the direct neighbors generally willing to forward data.

2. Calculate $D(y)$, y being member of N, for all nodes of N. $D(y)$ is the number of direct (symmetric) neighbors of y, without the neighbors contained in N and without the calculating node itself.

3. Add the nodes from N to the MPR set which are the only nodes allowing the reachability of a node from $N2$ (2-hop neighborhood, see above). Example: Node b from $N2$ can only be reached by a (symmetric) connection to node a from N, so a must be added to the MPR set. Then remove all nodes from $N2$ which are already reachable via a node of the MPR set.

4. As long as there exist nodes in $N2$ that are not reachable by at least one node contained in the MPR set:

   (a) For each node in N calculate the reachability, i. e. the number of nodes in $N2$ that cannot be reached by at least one MPR node and which can be reached by this 1-hop neighbor.

   (b) Choose that node as MPR which has the highest `N_willingness` among all nodes from N with a reachability greater than 0. If several nodes are available select the node that can reach the maximum number of nodes from $N2$. If there are still several nodes available choose the node as MPR with the greater value $D(y)$. Remove the nodes from $N2$ which can now be reached by a node of the MPR set.

5. The MPR set of a node is created by the combination of the MPR sets of all its interfaces. As an optimization, each node y of the MPR set is processed in ascending order

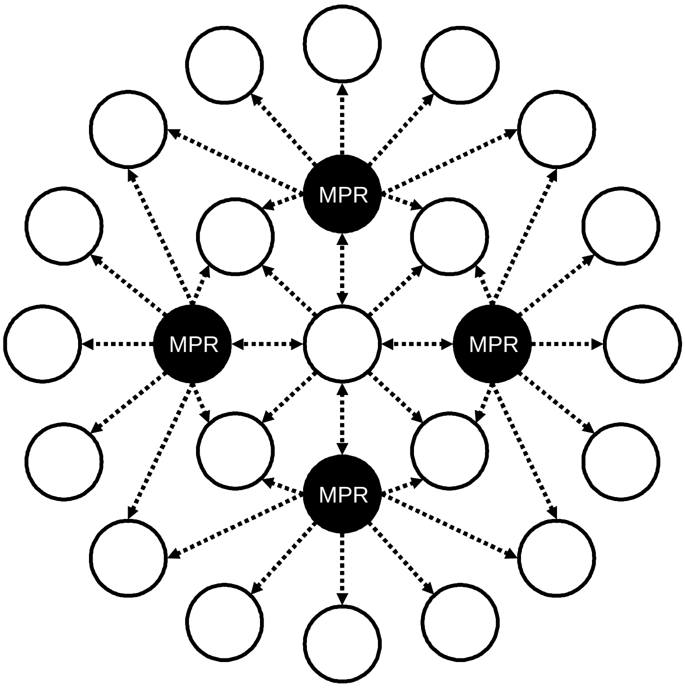


Figure 17: Optimized flooding by Multipoint Relays in OLSR

of N_willingness. If all nodes from $N2$ are still reachable by the MPR set if node y is removed and the N_willingness of node y is smaller than WILL_ALWAYS, then node y CAN be removed from the MPR set.

The specification of OLSR allows the use of other algorithms as well as extensions of the algorithm described above.

2.3.5 MID and HNA Messages

Besides the messages described above OLSR provides MID messages (Multiple Interface Declaration). These messages are used by a node to inform its neighbors about its addresses in the network. The normal case is that a node possesses exactly one address. In this case the node must not send an MID message. If a node possesses several network addresses the connection between the addresses of the OLSR interfaces and the main address is distributed by means of MID messages. A node with several addresses has to periodically send MID messages with the related information about its interface configuration. The time interval is determined by the parameter MID_INTERVAL. Like other control messages these are distributed via the MPR mechanism, i. e. via the MPRs of the nodes in the network.

Each node in the network stores the information about the interfaces of the other nodes of the network so that this information can be included in the calculation of routes. The format of an MID message is as follows:

Finally OLSR has another message type: *Host and Network Association (HNA).* These messages represent an optional subset of the OLSR functionality which is only applied if an OLSR

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                    OLSR Interface Address                     |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                    OLSR Interface Address                     |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                              ...                              |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 18: Format of the MID messages in OLSR [13]

network is connected to another network not working with OLSR. Therefore these messages are not relevant for this report.

2.3.6 Tables Used

To enable the routing of data packets, each node manages a total of four tables which are described in the following:

1. Duplicate set: In this table the information about received data packets is stored. This consists of `D_addr` (sender address), `D_seq` (sequence number), and `D_time` (time until which the table entry is stored). With this information a node can decide if a particular message has been received and processed before and can, therefore, be discarded upon receipt or if it must still be processed.

2. Neighbor table: This table contains information about the direct neighbors of a node and contains for each entry the fields `N_addr` (neighbor address) and `N_status`. The status of the connection with the neighbor can take the values *symmetric*, *asymmetric*, and *MPR*.

3. Topology table: This table contains information about the topology of the network and serves as basis for the calculation of the routing table. For each MPR of other nodes of the network a node records data in his topology table. Each entry consists of the field `T_dest`, `T_last`, `T_seq` und `T_time`. Each node named in `T_last` is an MPR selected in the MPR set by `T_dest` with the sequence number `T_seq`. Node n `T_dest` can therefore be reached via node `T_last`. Each entry is removed from the table after expiration of the validity time `T_time`.

4. Routing table: Based on the topology and neighbor table each node builds a routing table. This table contains fields `R_dest`, `R_next`, and `R_dist`. Such an entry signifies that the node `R_dest` is at an approximate distance of `R_dist` hops and can be reached via the direct neighbor `R_next`.

In the whole, in spite of the optimization of the flooding using MPRs, OLSR is one of the routing protocols which generates a rather bit routing overhead, but it is suited for the use in MANETs and is more and more advancing to become a standard [13].

In this context the protocol *Topology Dissemination Based on Reverse-Path Forwarding (TBRPF)* [27] should be mentioned which in some parts resembles OLSR but has some advantages.

3 Basic Scenario

The first part of this section gives a short overview about the pilot scenario defined by the SicAri consortium and its relation to the basic scenarios underlying this report with respect to MANET applications. The following two parts describe technical details and security aspects of a simple scenario and corresponding passive attacks to be analyzed as well as an extended scenario including the detection of active attacks as depicted in the following sections of this report.

3.1 SicAri Pilot Scenario

The SicAri project aims to provide both a security architecture and a toolkit for ubiquitous Internet usage. Besides the scenarios that have been proposed by the various partners, the consortium agreed to develop an overall pilot-scenario that includes as many aspects as possible of the SicAri project. The purpose of the scenario is to demonstrate a set typical platform functions within a single scenario.

SicAri Pilot Scenario: Mobile Work Today, there are mobile workers in many fields of work such as production, maintenance or working out of the office. Mobile workers usually make use of different mobile devices, e. g. PDAs, laptops, talking assistants [4] which may be owned by the workers themselves or by their companies. Companies usually specify and enforce a security policy for all security related issues containing rules that must be met. For this, both employees and companies' services and resources may be grouped into different levels of trust.

The generic pilot-scenario covers tasks and workflows of mobile worker, such as personalization of devices, secure access to services and resources, exchange of documents, and delivery of work results.

Further characteristics of the pilot scenario are:

- The scenario deals with a coherent group of experienced and knowledgeable workers.
- Workers are familiar with mobile devices and security tokens, such as smart-cards.
- Workers may require additional information in order to perform their tasks.
- Workers may produce written work results (such as reports and checklists).
- The company provides cryptographic key pairs and public key certificates for all workers, issued by a PKI.

The pilot scenario is divided into a generic scenario "Mobile Work" and various instances of this scenario. The following list summarizes the policy-relevant parameters of the pilot scenario:

- Actors:
  - Mobile workers (coherent group of workers)
  - Company's security officer / administrator

- Objects / Resources:
  - All kind of resources such as applications, services, network resources, information, documents, checklists
  - Certificates, cryptographic key-pairs
- Derived Requirements:
  - Personalization of devices, use of digital identities
  - Authentication of workers against their work environment
  - Controlled access to all resources and services provided
  - Generation of digital signatures for integrity protection of work results (e. g., documents, checklists)
  - Confidentiality: Secure transfer of data via networks (transport and document security)
  - Controlled search of document in peer-to-peer networks
  - Controlled exchange of documents via peer-to-peer networks

3.2 Generalized Scenario with Different, Geographical Security Areas

Within our generalized scenario we consider an ad hoc network with trustworthy and non-trustworthy mobile communication devices. Non-trustworthy devices are assumed to be restricted to a designated area within the network. For trustworthy devices no restrictions with respect to movement are made.

From the technical point of view, trustworthy devices are further divided into devices whose functionality can be changed with small effort (like notebooks or PDAs running open source operating systems) and devices where this is not (easily) possible (like network printers with proprietary operating systems).

Our approach for the prevention of passive attacks is based on an extension of the functionality of nodes. Nodes, whose functionality we extend, are further called *extended nodes*. Nodes with unchanged functionality are referred to as *standard nodes*.

The exchanged information is classified into confidential and non-confidential data flows. To setup the required routes, the deployed routing protocol is DSR as described in Section 2.2.

From the described scenario, we can extract four possible communication cases, which are shown in Figure 19:

- Trustworthy nodes exchanging confidential information
- Trustworthy nodes exchanging non-confidential information
- Trustworthy node and non-trustworthy node exchanging non-confidential information
- Non-trustworthy nodes exchanging non-confidential information

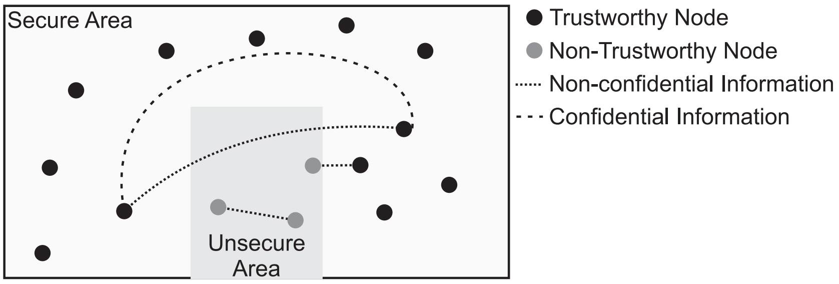


Figure 19: Scenario with schematic end-to-end communication relationships

3.2.1 Security Aspects

Various attacks that are inherently possible in mobile ad hoc networks due to their infrastructureless nature have been identified so far [39] and will be described in this document. These can be classified into active and passive attacks. In general, the intention of an active attack is to change the data flow in a mobile ad hoc network. For this, active attacks require changes in the behavior of the deployed routing protocols to achieve the desired effects. Thus, nodes that perform active attacks, can be detected (and located) by an intrusion detection system, as described later in this document. In contrast to this, passive attacks like traffic analysis or eavesdropping of specific communications do not require the attacker to change the routing protocol and have no direct effect on the behavior of the mobile ad hoc network as a whole. In fact, passive attacks do not require the node to transmit any information, what makes it (nearly) impossible to detect passive attackers. Active attack mechanisms may (but do not have to) be used in combination with eavesdropping in order to make the result even worse (or better from the perspective of the attacker).

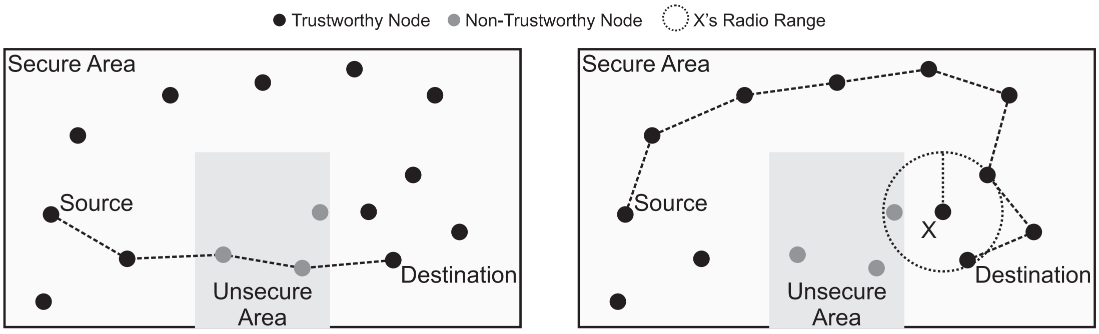
Today, data encryption is the method of choice, to prevent that information collected during an eavesdropping attack can be exploited by the attacker. In view of long-term security, state of the art encryption mechanisms could fail in a few years, giving the potentially malicious visitor in our scenario the chance to reveal the secrets once collected. If we further consider public key infrastructures [18], where a private/public key pair is usually used for a long period of time, it might be possible for an adversary to compute the private key from collected data and to later still misuse this knowledge. From this perspective, a feasible way to keep information confidential in the future with today's techniques is to keep it away from unauthorized persons.

For our given scenario, we expect all attacks, whether they are of active or passive nature, to be restricted to a designated area with a low (physical) security level (as for example a visitor area within an airport). The consequence is that confidential information should not enter this area, as shown in Figure 19 where end-to-end communication relationships are sketched.

3.2.2 Resulting Requirements

In the cases of non-confidential communication, non-trustworthy nodes should be used for hop by hop information forwarding in order to provide the expected connectivity. A respective route in our scenario is shown in Figure 20(a). This route will most likely be chosen by DSR since it is the shortest with respect to number of hops.

In the case of a confidential communication between trustworthy nodes, a non-trustworthy should (for the reasons described in the previous section) not be part of a path between trustwor-



(a) Non-confidential route

(b) Confidential route

Figure 20: Confidential versus non-confidential route

thy nodes. So at the first stage, we need the ability of an explicit user interaction or an implicit policy mechanism within the utilized application to inform the routing process whether or not the information to transmit is confidential.

For confidential communications, we have to establish routes which bypass the insecure area. Figure 20(b) shows a route in our scenario that meets this restriction.

When we take a look at node X in Figure 20(b), we see that due to its proximity to the visitor area, its transmission would also reach unauthorized nodes. Even though X is a trustworthy node, it should at its current position not be used for forwarding confidential information.

For the decision whether a trustworthy node may be contained in a route that is used for confidential communication, information about its position and its radio range has to be available.

With respect to the subclassifications of trustworthy devices, we have to assume that connectivity decreases if we restrict routes to trustworthy nodes whose routing functionality we extended to distinguish between confidential and non-confidential communication and to handle position information (extended nodes). To overcome this, we allow a route to contain a certain number of non-extended trustworthy nodes (standard nodes) between any two adjacent extended nodes. The endpoints of a route, that is sender and receiver, are expected to be extended nodes.

The properties of our scenario and the requirements can be summarized as follows:

- Devices are classified into trustworthy and non-trustworthy
- Non-trustworthy devices are restricted to the insecure area
- Trustworthy devices are further classified into extended nodes and standard nodes
- End points of communications are extended nodes
- An information exchange between application and routing process is needed
- Knowledge of the position and the radio range of extended nodes has to be available

4 Security Objectives

ISO 7498-2 [21] distinguishes five security objectives for communication systems. These following security objectives should also be met in our scenario described in the previous section.

Confidentiality (also known as Privacy) implies "keeping information secret from all but those who are authorized to see it" [25]. The information is viewed only by authorized entities and some encryption mechanisms ensure that all other entities do not have access to the protected information. This requires that the authorized entities are clearly identified, listed and provided with cryptographic primitives (keys) to access the protected information. With clearly identified we mean that the identification information is forgeable, unique and reliable in the system. Mechanisms must be provided to verify these identities allowing the separation of authorized and unauthorized entities in two disjoint sets.

Data Integrity is the property of ensuring that the "information has not been altered by authorized or unknown mean" [25]. The information is contained in the data and the data is transmitted. To avoid a loss or an alteration of the information the data must be received exactly in the form it has been sent. Digital signatures allow the detection of any modification of the data. To achieve this, the sender signs the data prior to sending. The receiver then verifies the signature of the sender. The process therefore implies that the signature is clearly bound to the sender of the data.

Access Control is the act of "restricting access to resources to privileged entities" [25]. Also for this security objective privileged entities must be clearly identified to be provided with credentials. The credential is presented to the resource provider and only a valid credential grant the usage of the resource.

Non-repudiation means the ability of "preventing the denial of previous commitments or actions" [25]. In a system providing this security objective an entity can not deny the actions it has performed. Every action has to be digitally signed by its initiator.

Authentication (data origin and entity) requires the "corroborating of the identity of an entity" [25] in case of the entity authentication and the "corroborating of the source of information" [25] in case of the message authentication. Authentication is in other terms the identification of an entity or a message and the possibility to verify the claimed identity. The authentication instance is submitted a proof (password or smart-card) to verify the identity of the entity or of the source of an information.

5 Attack Scenarios

Attack scenarios especially with respect to MANETs can be subdivided into passive and active attacks scenarios. In this section passive attacks as there are eavesdropping and freeriding are described, first. Afterwards, active attacks are presented comprising black hole, worm hole, rushing, sybil, and other attacks.

5.1 Passive Attacks

In this section, we describe attack mechanisms that are passive with respect to two aspects. First, they do not have a direct influence on the communication of other nodes or on the infrastructure of the MANET. Second they do not require to change the used routing protocol in order to achieve the desired effects.

5.1.1 Eavesdropping

Eavesdropping is in fact a very simple attack that is inherently possible in MANETs due to their infrastructural properties. Since every node that is used to build up an ad hoc network may be contained in a routing path between any communicating nodes, every node is able to read along the communication. Caused by the wireless data transmission, it is further on fully sufficient if an attacker is in the radio-range of the respective route as shown in Figure 21. One could say that this is no matter if we just make use of encryption mechanisms to protect the transmitted information. But what (with an eye on long term security) if this information is to be still confidential in 10 years? The past has shown, that encryption algorithms may fail with growth in computing power, which makes brute force attacks on encryption keys possible in an affordable amount of time. The DES algorithm as one example, once considered as secure can not anymore be described with this adjective today [17]. The future may show, that quantum computers will make this situation even worse [10].

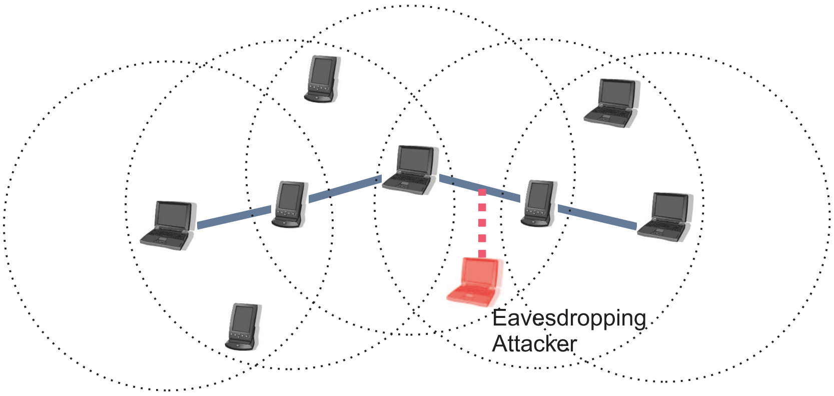
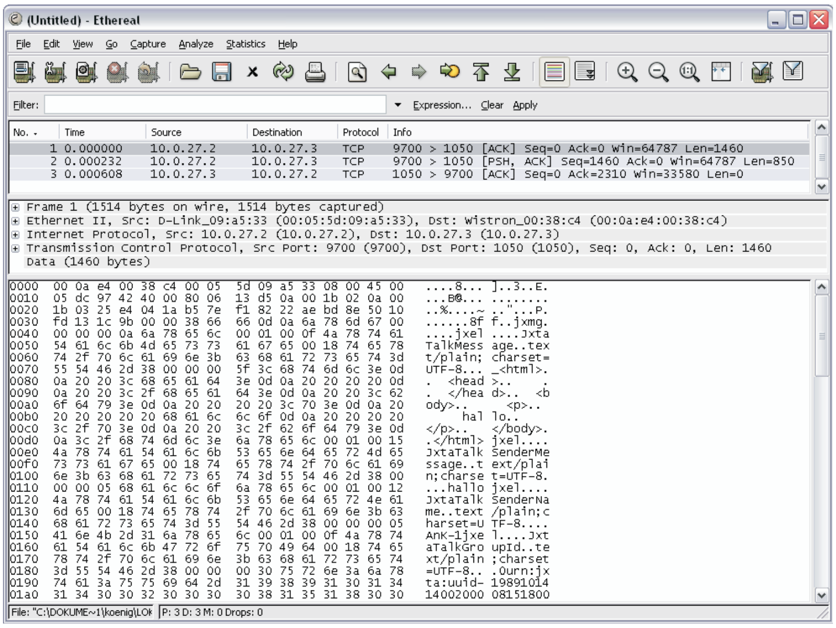


Figure 21: Schematic Eavesdropping Attack

What follows, shall be pointed out with two short examples, out of which the first one is the transmission of health related data which is an up to date topic when we look at the introduction of the German "Gesundheitskarte" [12]. With this, it becomes possible for everyone to access health-information (like the report of the last check up at the family doctor) from public terminals or (in the future) from private devices, which may be connected via an ad hoc network. Who is interested in reading along this information and collecting the history of your diseases? Well, the provider of your life insurance could be for the next calculation of your contribution.

The second example provides a more general and technical point of view, that also holds for our scenario as described in 3.2.1. For this, we take a look at public key infrastructures, where certificates as the binding of an identity to a public / private key pair can be valid for a long period of time. A malicious node which performs an eavesdropping attack will in the future possibly be able to calculate private keys from collected data. If the assigned certificates are then still valid in the respective (ad hoc) network, the attacker can come back and decrypt and listen in all communication that is routed via his node in real time. In our scenario, this opens doors for industrial espionage.

Compared to active attacks, there is no need for exhaustive preparation of eavesdropping. The described routing protocols do not (necessarily) have to be changed to obtain the desired result. One can simply capture bypassing data packets. Tools for this, like Ethereal [2], as shown in Figure 22 are freely available for download. Active attack mechanisms as described in the previous sections may be used in combination with eavesdropping in order to make the result even worse (or better from the perspective of the attacker) but this is not necessary at all.


(Untitled) - Ethereal
File Edit View Go Capture Analyze Statistics Help
Filter:
Expression... Clear Apply

```
No.  Time      Source      Destination  Protocol  Info
   1 0.000000  10.0.27.2   10.0.27.3    TCP       9700 > 1050 [ACK] Seq=0 Ack=0 Win=64787 Len=1460
   2 0.000232  10.0.27.2   10.0.27.3    TCP       9700 > 1050 [PSH, ACK] Seq=1460 Ack=0 Win=64787 Len=850
   3 0.000608  10.0.27.3   10.0.27.2    TCP       1050 > 9700 [ACK] Seq=0 Ack=2310 Win=33580 Len=0
```

```
⊞ Frame 1 (1514 bytes on wire, 1514 bytes captured)
⊞ Ethernet II, Src: D-Link_09:a5:33 (00:05:5d:09:a5:33), Dst: Wistron_00:38:c4 (00:0a:e4:00:38:c4)
⊞ Internet Protocol, Src: 10.0.27.2 (10.0.27.2), Dst: 10.0.27.3 (10.0.27.3)
⊞ Transmission Control Protocol, Src Port: 9700 (9700), Dst Port: 1050 (1050), Seq: 0, Ack: 0, Len: 1460
  Data (1460 bytes)
```

```
0000  00 0a e4 00 38 c4 00 05  5d 09 a5 33 08 00 45 00   ....8... ]..3..E.
0010  05 dc 97 42 40 00 80 06  13 d5 0a 00 1b 02 0a 00   ...B@... ........
0020  1b 03 25 e4 04 1a b5 7e  f1 82 22 ae bd 8e 50 10   ..%....~ .."...P.
0030  fd 13 1c 9b 00 00 38 66  66 0d 0a 6a 78 6d 67 00   ......8f f..jxmg.
0040  00 00 00 0a 6a 78 65 6c  00 01 00 0f 4a 78 74 61   ....jxel ....Jxta
0050  54 61 6c 6b 4d 65 73 73  61 67 65 00 18 74 65 78   TalkMess age..tex
0060  74 2f 70 6c 61 69 6e 3b  63 68 61 72 73 65 74 3d   t/plain; charset=
0070  55 54 46 2d 38 00 00 00  5f 3c 68 74 6d 6c 3e 0d   UTF-8... _<html>.
0080  0a 20 20 3c 68 65 61 64  3e 0d 0a 20 20 20 20 0d   .  <head >..    .
0090  0a 20 20 3c 2f 68 65 61  64 3e 0d 0a 20 20 3c 62   .  </hea d>..  <b
00a0  6f 64 79 3e 0d 0a 20 20  20 20 3c 70 3e 0d 0a 20   ody>..     <p>..
00b0  20 20 20 20 20 68 61 6c  6c 6f 0d 0a 20 20 20 20        hal lo..
00c0  3c 2f 70 3e 0d 0a 20 20  3c 2f 62 6f 64 79 3e 0d   </p>..   </body>.
00d0  0a 3c 2f 68 74 6d 6c 3e  6a 78 65 6c 00 01 00 15   .</html> jxel....
00e0  4a 78 74 61 54 61 6c 6b  53 65 6e 64 65 72 4d 65   JxtaTalk SenderMe
00f0  73 73 61 67 65 00 18 74  65 78 74 2f 70 6c 61 69   ssage..t ext/plai
0100  6e 3b 63 68 61 72 73 65  74 3d 55 54 46 2d 38 00   n;charse t=UTF-8.
0110  00 00 05 68 61 6c 6c 6f  6a 78 65 6c 00 01 00 12   ...hallo jxel....
0120  4a 78 74 61 54 61 6c 6b  53 65 6e 64 65 72 4e 61   JxtaTalk SenderNa
0130  6d 65 00 18 74 65 78 74  2f 70 6c 61 69 6e 3b 63   me..text /plain;c
0140  68 61 72 73 65 74 3d 55  54 46 2d 38 00 00 00 05   harset=U TF-8....
0150  41 6e 4b 2d 31 6a 78 65  6c 00 01 00 0f 4a 78 74   AnK-1jxe l....Jxt
0160  61 54 61 6c 6b 47 72 6f  75 70 49 64 00 18 74 65   aTalkGro upId..te
0170  78 74 2f 70 6c 61 69 6e  3b 63 68 61 72 73 65 74   xt/plain ;charset
0180  3d 55 54 46 2d 38 00 00  00 30 75 72 6e 3a 6a 78   =UTF-8.. .0urn:jx
0190  74 61 3a 75 75 69 64 2d  31 39 38 39 31 30 31 34   ta:uuid- 19891014
01a0  31 34 30 30 32 30 30 30  30 38 31 35 31 38 30 30   14002000 08151800
```

File: "C:\DOKUME~1\koenig\LOk   P: 3 D: 3 M: 0 Drops: 0

Figure 22: Eavesdropped Peer-to-Peer Chat Session with Ethereal

Eavesdropping is independent with respect to the deployed routing protocol. Reading along the communication of other nodes is possible as soon as the attacker is part of or at least in radio range of the discovered routing path between sender and receiver. If the attacker is interested in the communication of two specific nodes, the knowledge of the protocol (in combination with the knowledge of the node's locations) will help do deduce the route that will be discovered. But also in this case, regardless of the routing protocol, the discovered path will most likely be the shortest one with respect to number of contained nodes and geographical proximity.

5.1.2 Freeriders

The concept of freeriders is mostly known from the area of peer-to-peer systems [35]. Here it describes the behavior of a peer which consumes resources of other peers, but does not provide any by itself. In a file sharing peer-to-peer system, a freerider would be a peer who just downloads files from other peers without offering files to be downloaded by others. Regarding a distributed computing peer-to-peer system, a freerider uses computing power of other participating peers but is not willing to perform foreign tasks.

This concept also applies for MANETs. Here, nodes which take part in an ad hoc network may behave selfish, that is they make use of the network for their communication but they reject to route data that belongs to other nodes. The aim is to achieve a saving in own bandwidth or battery power which both are usually limited for mobile nodes like cell phones or PDAs.

Unlike eavesdropping and against our definition of passive attacks, freeriding requires to change the used routing protocols in order to drop data packets that do not belong to the node's own communication. Nevertheless, we classify this attack to be a passive one, since the necessary changes are only of a minor nature. Also, there is no influence on the communication of other nodes besides the additional usage of bandwidth that is needed to route the data of the freerider. One could say, that obviously the number of routes that can be discovered, will get lower when a node changes its behavior from protocol conform to freerider. This is due to the fact, that this node could be a single point of failure in the route from sender to receiver, as shown in Figure 23. Since this also happens, when the node is switched off, what is common for nodes in MANETs when we consider the limited available energy, we will neglect this effect of freeriders in the scope of this document.

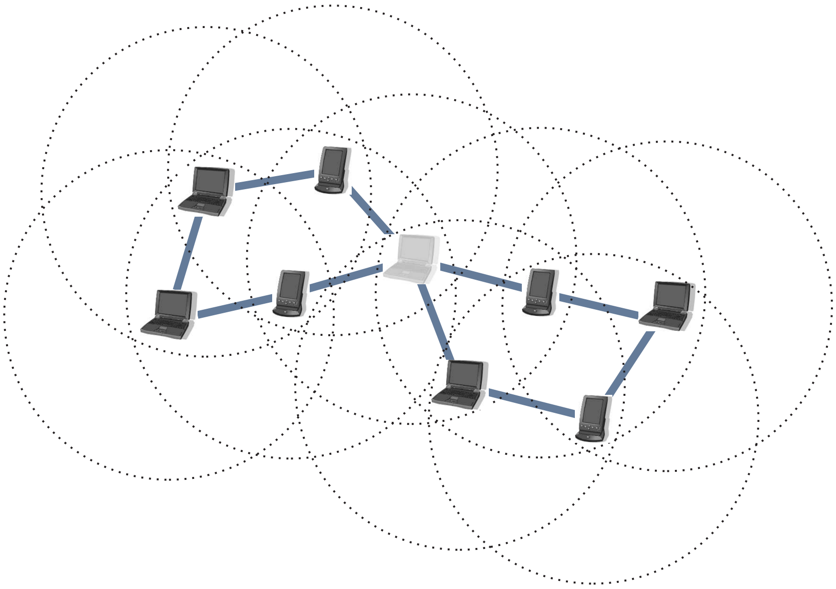


Figure 23: Freerider as single point of failure

5.2 Active Attacks

The following presents the basic concepts of some important active attacks to MANET routing. Each attack is first presented in a general way and then analyzed in details for the routing protocols AODV and OLSR.

5.2.1 Blackhole

The black hole attack [3, 5] uses the idea of purposefully generating incorrect routes so that packets are no longer forwarded to the proper recipient D but instead get lost or sent to an attacker. Thus derives the name as something similar to a black hole is created in order to "swallow" the data packets. Fig. 24 shows an example of normal data traffic transferred via adjacent nodes to node D on the left and the effects of a successful attack on the right. Messages intended for node D do not reach their actual target but are intercepted by the attacker.

The attacker may also distribute fake routing information in order to become included in as many valid routes of the network as possible. This type of attack is always used during route finding or routing information update phases of the process.

A black hole attack can also serve as a precursor for the execution of further attacks.

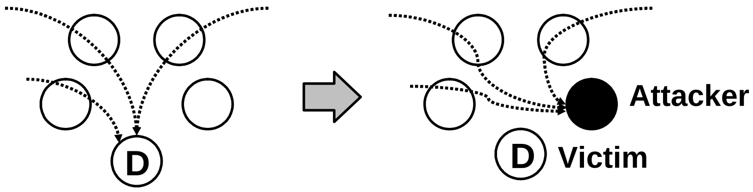


Figure 24: Data flow to target D before and during a black hole attack

AODV One problem with route finding in AODV is that not only the destination node can send a RREP message, it is also possible that a node in the middle knows a valid route and can send an RREP message back to the sender (see section 2.1.1). An attacker X (see fig. 25) who receives an RREQ message can take advantage of this by sending an RREP packet back to the sender, pretending that the destination node is only one or few hops away from the attacking node. The attacker will then be masked as the shortest path and be included within the transmission route.

Instead of pretending to have a shorter route the attacker can fake a higher sequence number in his RREP message. This way the new route overwrites routes transferred by other nodes. A combination of both tactics is also possible.

Each approach is based on the fact that attackers must await a route discovery process initiated by another node. Attackers can artificially initiate a route finding process by sending a fake RERR packet which pretends that a section of the route between attacker and recipient node is no longer available. This intentionally generated route fault can lead to the initiation of a RREQ message that the attacker can later misuse for his aims.

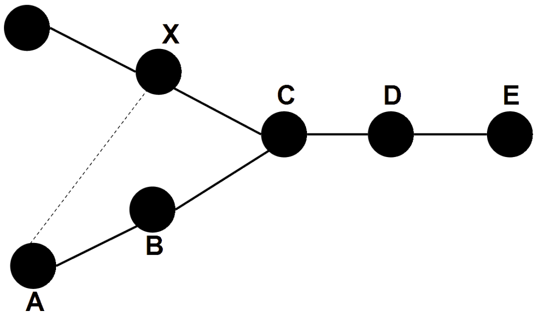


Figure 25: Node X pretends to be connected to A using HELLO messages

This procedure creates a "black hole" that collects and discards all arriving data. Possible goals of this attack are the following:

- to selectively delete data (*gray hole*)
- to isolate a node (DoS)

OLSR To create a black hole in an OLSR based network, an attacker has the possibility to just not forward any TC messages which has as the result that nodes are possibly no longer reachable [31]. Especially in a network which does not provide redundancies, i. e. several paths are created and stored in parallel (multi-path routing), this inevitably leads to the interruption of some connections. If no MID and HNA messages are forwarded additional information via the interfaces of a node or via connections to external networks get lost which can have the same effect.

The real black hole attack, however, is an attack which attracts and retains not only control messages but also data packets exchanged between nodes. In general a black hole attack serves to exclude a node from the network. In principle also other scenarios are thinkable so that the goals of a black hole attack may be roughly classified as follows:

- to remove data selectively (*gray hole*)
- to isolate nodes (DoS)

It is also important that a gray hole attack can only work if the attacker is able to forward data to the victim as not discarded messages can otherwise not be delivered. All goals have in common that in a first step the data traffic of the network must be attracted. To achieve this, an attacker must first trap as many messages meant for his victim as possible. The victim may be a single node or part of the network. In OLSR the attacker has two options to spread false information about his neighborhood by such manipulation. More precisely a node pretends that certain other nodes belong to his neighborhood although this is not the case.

- Generation of false HELLO messages

  As displayed in figure 26, an attacker X could e. g. make such a manipulation using appropriate HELLO messages to pretend that node A is his neighbor. It would follow from this that node C and all other neighbors of X would store a forged 2-hop neighborhood and therefore also a wrong MPR set. Presumedly node C would mark nodes X and D as MPR and not, as it should be, nodes X, B, and D because the first set is smaller. Routing messages the routing of which is influenced by the MPR mechanism can no longer reach node A and would instead be led to X. Furthermore, the attacker has the possibility to signal a higher readiness to forward messages by indicating a high `Willingness`.

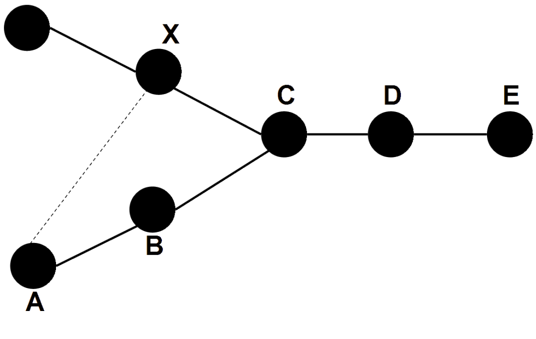


Figure 26: Node X pretends to have a connection to A using HELLO messages

- Generation of false TC messages

  TC messages with a manipulated sender address lead to false neighborhood information which is then distributed on the network. If e. g. node X forwards a TC message in the name of node C in which he pretends that A is his neighbor (see figure 27), node D, upon receipt of this message, will wrongly assume that nodes C and A are neighbors. Such an attack can only work if the TC message has an ANSN which is bigger than the highest ANSN associated with C and stored in the topology table of D. Otherwise D will discard this message according to the protocol so that the attack fails. This allows the short-time spoofing of a false identity, but differs, however, from the Sybil attack explained later in which an attacker tries to permanently adopt the identity of another node.

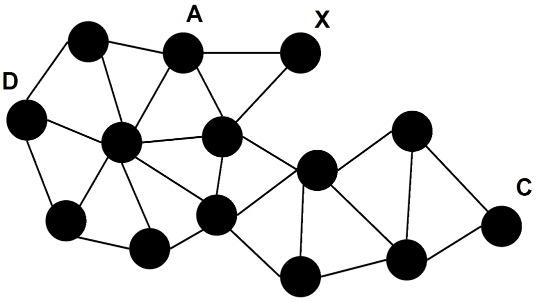


Figure 27: Node X pretends by TC messages to be C

5.2.2 Wormhole

A wormhole attack [38] uses two directly connected nodes of a network to re-route data traffic. In order for this to be successful, the two nodes must "ally" themselves and establish an additional channel outside normal network communications which serves as a tunnel. This shortcut is named after a wormhole as it mimics this hypothetical physical phenomenon.

In this type of attack the two nodes mask that they are not directly adjacent nodes and pretend to be neighbors (therefore disposing a fast connection to each other and their neighbors). As these paths are used for sending data that is not part of the proper network, wormholes are very difficult to detect.

Wormholes themselves are not necessarily only negative for a network as such a shortcut can have positive benefits such as relief for the network or shorter transfer times for packets on the routes containing the wormhole. Attackers use wormholes in the network to make their nodes appear more attractive (with perceived faster transfer times) so that more data is routed through their nodes. Similar to the black hole attack, the wormhole attack can also be used as a basis for further attacks.

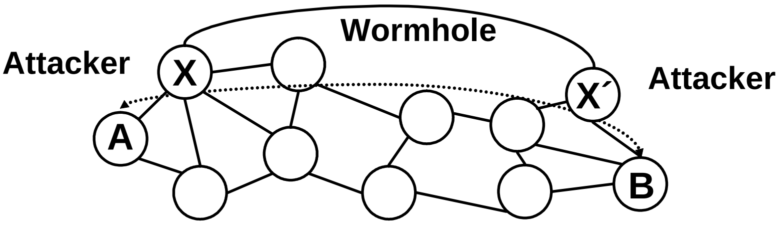


Figure 28: Data flow during a wormhole attack of X and X'

AODV During a wormhole attack two attackers X and X' work together (as described above) to create an additional channel or out-of-band connection (see Abb. 28). The actions taken by the wormhole attacker first resembles those of a black hole attacker. Upon receipt of a RREQ message both send an RREP message back to the sender which bears fake information intended to attract all data traffic originally intended for another recipient. The difference lies in that a pair of attackers act at two different places of the network, e. g. to control the data traffic between two nodes A and B in a wormhole attack.

As with black hole attacks, wormhole attackers have the possibility to artificially raise the sequence numbers of this RREP message to overwrite route information sent by other nodes with his RREPs.

Possible aims of a wormhole attack are:

- to eavesdrop messages
- to selectively delete data
- to manipulate data
- to isolate nodes (DoS)

OLSR A direct connection between A and B, lying outside of the actual network, can be artificially created by an intruder X, to exchange messages between A and B via X (see figure 29). A further possibility is the creation of a longer wormhole by two collaborating nodes X and X' (see figure 30). This is the more common approach, which will be examined here in more detail. The procedure differs anyway only in few details, two co-operating aggressors can however accomplish a substantially more effective attack, since a genuine out-of-band connection provides an actually faster connection to the network.

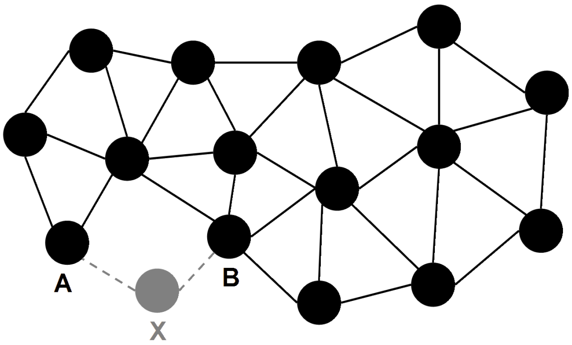


Figure 29: Node X generates a wormhole

In [31] still another possibility is presented where the aggressors X and X do not appear to the other network nodes all. For node A it has thus the appearance, as if it was directly communicating with node B. In principle also such a variant is conceivable, it concerns however a wormhole that works at a lower network level than the switching layer, which is responsible for the routing. This variant can be implemented by the aggressors very easily, since they must only forward the HELLO messages of their victims to the other end of the wormhole tunnels. This way the victims A and B consider themselves to be direct neighbors. If a wormhole is created

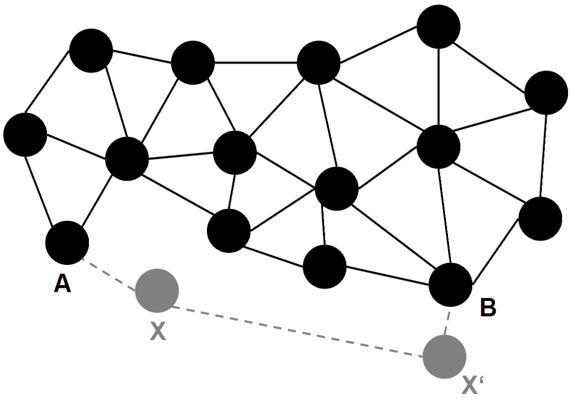


Figure 30: Two nodes X and X' jointly create a wormhole

in this way, then it is the task of the aggressor nodes, to forward the data stream between the nodes through the wormhole tunnel.

In order to be able to successfully produce a wormhole, the aggressors X and X' basically just have to distribute the information about the available connection between them to their respective neighbors. This means at first very little costs. Such a wormhole does however in most cases probably not yet serve its purpose, since as much additional data packets as possible should be routed through the new tunnel. An aggressor can accomplish this with additional steps, similar as for a black hole attack.

Also for these attacks it makes sense to divide them according to their goals. Possible goals are:

- to eavesdrop messages
- to selectively delete messages
- to manipulate messages
- to isolate a node (DoS)

All goals have in common that the first step is to create an additional direct connection. In addition the attacker nodes must attract packets, as with the black hole attack. The difference is that this happens at the same time for two victims at two different places in the network. The victim can be again on both sides of the wormhole an individual node or also a complete part of the network. Once this connection is established, the aggressor is in control of the connection between node A and B and if maybe further connections that use the created wormhole.

5.2.3 Rushing

This attack is based on the idea of transferring messages as soon as possible so that they forestall other messages in finding a route. In this way an attacker can exert a considerably greater influence on route generation (ensuring the inclusion of the attackers node on the route). This works especially well as many routing protocols dispose of security mechanisms against copies with the result that only the data packet that arrives first is evaluated while all others are discarded.

To be able to successfully execute a rushing attack an attacker can also make use of the possibilities the lower network layers offer. So he can, e. g., ignore certain rules that normally force him to wait a certain time before sending the message.

AODV The aim of rushing attacks is to become part of as many routes as possible in order to eavesdrop, manipulate data or support other attacks. Node A sends a RREQ packet (see fig. 31) to search for a route to C (or to a node behind C). The attacker X receives the RREQ message sent via flooding and will then try to forward this message as soon as possible (possibly violating the rules of lower network layers). If successful node C receives the route request of attacker X before getting the corresponding message from node B. Node C processes this message and subsequently discards the message of B received afterwards since according to AODV specification it must only consider the first route request that bears a certain ID. Node C then returns an RREP message to the sender via X so that X is henceforth on the route between A and C instead of node B that would also have offered this connection.

The attack is rather simple since the attacker conforms to protocol with the exception of using "hurrying":

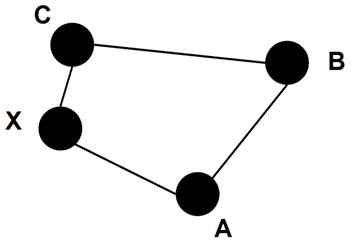


Figure 31: Node X performs a rushing attack

OLSR In OLSR there is the rule that a node that receives an MPR flooding message checks if the sending node is contained in its MPR selector set. If this is the case, the received message is forwarded. If the sender is not an MPR of the node, it will not forward the message but discard it. Under performance aspects the rule makes sense, in the same time it is a vulnerability of the protocol. This behavior can be used to undermine or prevent the correct forwarding of control messages.

The corresponding attack known as rushing attack, virtually originating in the connection of reactive protocols such as AODV, is also called MPR attack in OLSR. It is only one of different possible rushing attacks, however, the most important one. In the scenario presented in figure 32 node A sends a message to his neighbors B and X, where B is an MPR of A, X is no MPR and node C is MPR of B. The attacker X selects his MPRset not correctly and forwards the sent message although he is not obliged and authorized. Node C receives this message which is also forwarded by node B to C. The important thing is that node C will not forward the message, although he is MPR, because it has already received the message by the attacker X. To be able to perform a rushing attack, an attacker can moreover make use of the possibilities offered by lower networks layers. He can e. g. ignore certain rules which would normally force him not to send a message immediately but wait until the network channel is free.

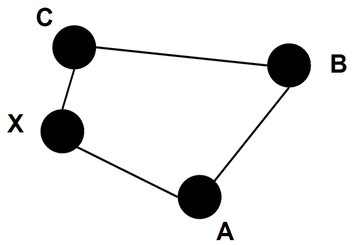


Figure 32: Node X performs a rushing respectively MPR attack

The rushing attack aims at bringing himself into a strong position in the network, i. e. becoming part of as many routes as possible. Such an attack can e. g. support the execution of a black hole attack or be carried out to eavesdrop or manipulate data. In principle the attacker only forwards the messages immediately and ignores the MPR rules.

5.2.4 Sybil

A Sybil attack[1] [26, 16] occurs when a node in the network tries to masquerade as several identities. This can be achieved in two ways, by feigning the existence of an additional node or by stealing the identity of an existing node. The advantage of controlling several identities to an attacker is that he can extensively conceal their activities within the network (e. g. a black hole attack).

AODV Eavesdropping to discover and existing identity is required in order to execute a Sybil attack.[2]

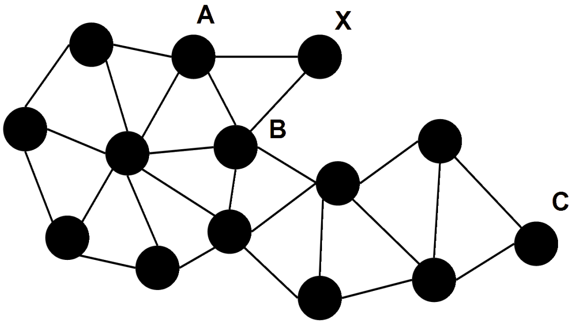


Figure 33: Node X pretends to be C

The stealing of an identity can be a rather simple as an attacker must only answer a RREQ message as if it were the destination node (sending a route reply with the identity of another node). Preventative measures include sending a RREP message so early that attacking messages

[1] Sybil is the name of a book [33] with the authentic report of the first psychoanalysis of a multiple split personality.

[2] In principle it is also conceivable that an attacker creates a completely new identity, however this is a less interesting attack to this report and the procedure only differs in few details.

are ignored. As with black hole and wormhole attacks, Sybil attacks can benefit from sending RERR messages (pseudo route error) to activate new route requests.[3]

OLSR A Sybil attack is normally performed with the aim to bring oneself into a strong position in the network. If an attacker knows the structure of a network, she can, e. g., adopt particular identities to become part of as many different routes as possible. In this position the attack can be the basis of many other attacks.

An attacker X can send HELLO messages which contain a faked sender address, i. e. in the presented example that of node C (see figure 34). From this it follows that nodes A and B send HELLO and TC messages that claim that they can reach node C. Furthermore, node X selects MPRs from his neighbors and distributes this information by his TC messages further under the fake address of node C. As a result of this, the selected MPRs will distribute the information that they are direct neighbors of C in their TC messages. This leads to conflicts in the routes to node C which can lead to connection breaks.

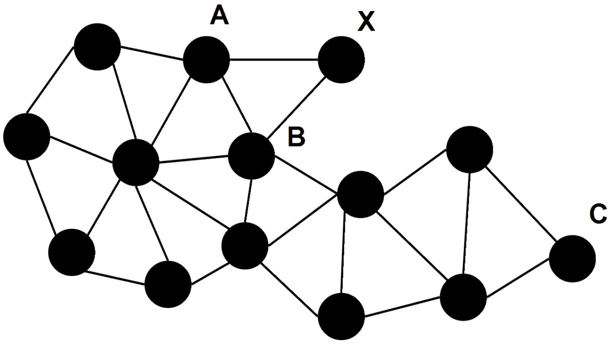


Figure 34: Node X pretends to be C using HELLO messages

An attacker X has additionally the possibility to impropriate interfaces which do not belong to him by generating false MID messages (or HNA messages). This is of relevance if a node has several interfaces with different network addresses and the network is informing over this correlation by the transfer of MID messages. Another possibility is the faking of the sender's address of the MID message and not only those addresses assigned to the interfaces. In both cases nodes will have difficulties to reach the correct owners of the nodes. HNA messages are an extension of OLSR and serve to connect to other networks which are not using OLSR for the routing. These two types of messages are, however, of very small relevance in most networks.

The following attack tree resembles a lot that of a black hole attack, but the sent messages differ from each other as at a Sybil attack e. g. also in HELLO messages a false identity is given, but no false neighborhood is pretended with the right identity as with the black hole attack.

[3] HELLO messages offer another possibility to execute a Sybil. An attacker X could send HELLO messages containing a fake sender address, as shown in fig. 33) for example to node C to spread incorrect neighborhood information. As HELLO messages are an optional extension of AODV and are usually not used this possibility is not addressed in more detail.

5.2.5 Other Active Attacks

AODV Another possible attack on AODV is the so-called *Vicious Query Flooding*. With this classic *Denial-of-Service (DoS)* attack the attacker sends big amounts of RREQ messages into the network. Here it is important that with these route requests a route is searched to a not existing destination. The resulting delays can lead to a collapse of the performance of the whole network.

OLSR A possible attack in OLSR is the so-called *ANSN attack*, which is based on a mere manipulation of sequence numbers. The principle is simple: An attacker forwards on behalf of another node packets which have very high ANSNs. Messages that really originate from this victim are discarded in the network, since these true messages have lower ANSNs and are therefore interpreted as deprecated. Such an attack is therefore a classic DoS attack with the aim to incapacitate the victim.

6 Solutions

Within this section, we present an approach to prevent passive attacks in mobile ad hoc networks and methods to identify active attacks. A promising possibility to prevent passive attacks with today's available mechanisms is to keep confidential data away from unauthorized persons. Here, we propose an approach for transmitting sensible information in the scenario described in Section 3.2.1 over routes which are restricted to appropriate nodes. To stay compatible with the existing Internet protocol architecture, we provide an extension based on cross-layer techniques.

6.1 Prevention of Passive Attacks

To achieve the desired functionality as stated in Section 3.2.1, we have to take influence on the vertical control flow within one node (from application to routing process) as well as on the horizontal data flow between two nodes. In the following, we describe the required extensions.

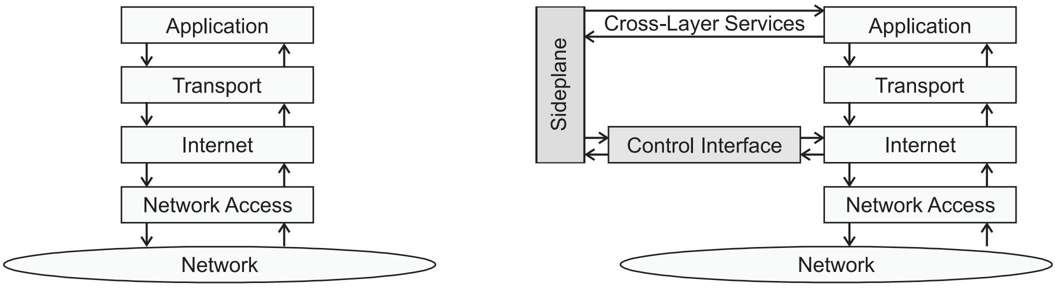
6.1.1 Cross-Layer Architecture

For reasons of compatibility, we base our approach on the well established Internet model with its strictly separated layer architecture, as shown in Figure 35(a). To exchange the necessary information between the application and the routing process, we add a cross-layer extension similar to the design proposed in [14] in a two step process.

In the first step an additional control interface which will be described in detail in the next section is attached to the network layer. This way, the desired influence on the routing process becomes possible. A similar approach with the aim to rewrite routing tables in order to optimize Gnutella networks is presented in [15].

Step two adds an orthogonal side-plane, which offers the service primitives for cross-layer communication. A draft of the resulting architecture is given in Figure 35(b).

The side-plane is organized as a lightweight data structure containing *(name, value)* tuples. Services are offered to add and change tuples, as well as to register a process to be informed



(a) Internet layer model (b) Cross-layer design

Figure 35: Extension of the Internet layer model

about changes in a specific tuple. With respect to our scenario, an application adds and changes tuples as for example the required position information of the node *("NodePosition", GPS coordinates)* and the insecure area *("UnsecureArea", Polygon)*. The control interface of the network layer registers for changes in both tuples. The information is then used to influence the routing process respectively.

6.1.2 Routing Control Interface

To stay compatible with nodes that run a standard DSR protocol, we leave the DSR header unchanged. The necessary information is contained in an additional header that follows the DSR header. We assign the header number *253* that is reserved by IANA for experimentation and testing. Thus the value of the *next header* field of the DSR header (which itself has not been assigned a fixed number yet) is *253* and points to our additional header. The resulting MAC frame is shown in Figure 36.

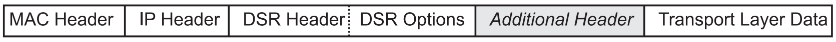


Figure 36: MAC frame with additional header

Nodes that run a standard DSR protocol simply ignore the additional header, whereas nodes that are equipped with our extension can read and evaluate the contained information. We define three header formats for the three phases of a communication which are route request, route reply and data transfer:

Route Request To provide security at the earliest possible point in time, we already demand the route request not to reach the insecure area. This way, we prevent non-trustworthy devices to perform traffic analysis or (if able to handle our extension) to pretend a position outside the insecure area with the aim to be included in a confidential route.

For the route discovery phase, the additional header contains the following information:

Next Header This is used to determine the transport layer protocol as for example TCP or UDP.

TTL The time to live for this route request. This field is decreased at each extended node by the number of hops that where traversed since the previous extended node. If we allow a route to only consist of extended nodes, TTL will be decreased by one at each (extended) node. If standard nodes are allowed to be situated between extended nodes, TTL is decreased respectively.

Max. intermediate standard nodes This field specifies the maximum amount of standard nodes that may be situated between two adjacent extended nodes. The TTL of the IP header is set to this value at every extended node. This way, the broadcast of a route request that will be done by standard DSR nodes is restricted to the desired amount of standard nodes between adjacent extended nodes.

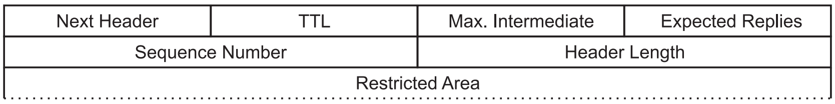
Expected Replies The amount of expected route replies which will be of relevance for our multi-path approach as a part of our future research.

Sequence Number This field is reserved for our multi-path approach.

Header Length The overall length of the additional header. The length is not fixed, since the following field contains a flexible description of the position and the shape of the visitor area.

Restricted Area A polygonal model of the insecure area.

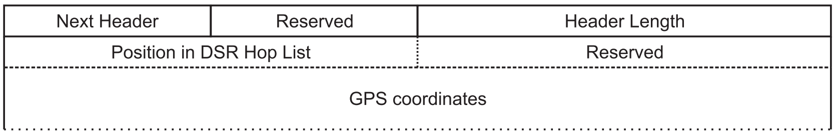
Figure 37 depicts the resulting structure of the additional header for the route request phase.



Next Header	TTL	Max. Intermediate	Expected Replies
Sequence Number		Header Length	
Restricted Area			

Figure 37: Header format for route request

Route Reply During the route reply phase, each extended node appends its current geographical position and its position in the recorded list of hops in the DSR options to the extended header. The resulting header format is shown in Figure 38.



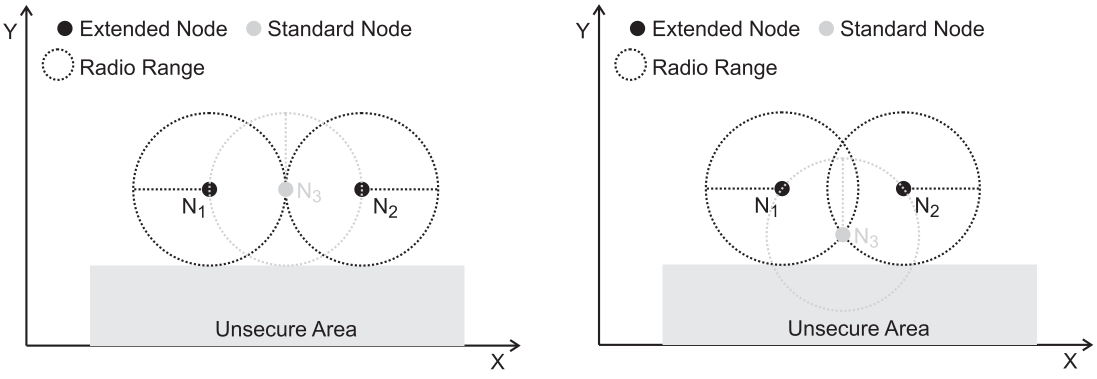
Next Header	Reserved	Header Length	
Position in DSR Hop List		Reserved	
GPS coordinates			

Figure 38: Header format for route reply

Based on this information, the source evaluates the degree of security, a route can offer. As an example, we assume that one standard node is situated between two extended nodes. For a worst case scenario, we further assume each extended node to be as close to the insecure area as it is allowed by its radio range.

If the distance between the extended nodes then converges to the sum of their radio ranges, the transmission of the intermediate standard node can not reach the insecure area, as shown in Figure 39(a).

If, on the other hand, the distance between the extended nodes is smaller than the sum of their radio ranges, the transmission of the intermediate standard node may well reach the insecure



(a) Secure route

(b) (Possibly) insecure route

Figure 39: Route with one intermediate standard node

area. Regarding this, a quantitative assertion about the security of a route becomes possible. The possibility that the transmission of an intermediate node reaches the insecure area can be calculated from the knowledge of the position of the two neighboring extended nodes. Figure 39(b) depicts the worst case of a (possibly) insecure situation. We have to notice, that the intermediate standard node could be situated anywhere within the intersecting plane of the radio ranges of the extended nodes.

Data Transfer During the phases of route request and route reply, we established a route, that meets our security requirements. Since we are confronted with mobile devices, the route has to be maintained with respect to security during the data transfer phase.

To obtain the required flexibility for our further work on more unrestricted scenarios, we define a header for the data transfer phase, that may be used optionally when the scenario (and thus the restrictions with respect to security) changes. The header therefore contains for each extended node in the DSR hop list a description of an area, where the node is allowed to detain. If a node leaves this area, it has to stop forwarding messages for the respective communication.

Figure 40 shows the additional header that is used during the data transfer phase.

<table>
<tr><td>Next Header</td><td>Reserved</td><td>Header Length</td></tr>
<tr><td colspan="2">Position in DSR Hop List</td><td>Reserved</td></tr>
<tr><td colspan="3">Permitted Area</td></tr>
</table>

Figure 40: Header format for data transfer

6.2 Identification of Active Attacks

After presenting the most important attacks in MANETs based on OLSR or AODV in the previous chapter, we will now describe possibilities to identify such attacks using *Intrusion Detection Systems (IDS)*. This description does not claim to be complete which is not possible due to the complexity of computer networks, but delivers some starting points which can be helpful for the implementation of an IDS for OLSR or AODV networks.

Intrusion Detection Systems rely on the facts that user and program activities are observable, and what is more important, normal and intrusion activities have distinct behavior. Intrusion Detection therefore involves capturing audit data and reasoning about the evidence in the data to determinate whether the system is under attack.

Although many IDSs have been developed for wired networks, the big number of differences in MANETs demand the design of new intrusion detection architectures and algorithms.

IBM labs in Zurich defined the following IDS features [6]:

1. Audit source location: The data to be analyzed may be obtained on a host, in application or system log files by *Host based IDS (HIDS)*, or network packets can be captured and examined by a *Network based IDS (NIDS)*.
2. Methodology of detection: Two approaches are used for the detection of intrusion, misuse detection and anomaly detection. With anomaly detection the system knows the user's standard profile and detects deviations from this habitual pattern. This model is well suited to detect unknown or previously not encountered attacks. On the other hand misuse detection monitors networks and hosts for known attacks. This class of IDS is useful in networks with highly dynamic behavioral patterns of the users. Besides that, they are more efficient as are less time and power consuming, and is a choice of many commercial IDS products. However, a frequently updated (and large) database of known attack signatures should be managed.
3. Computing location: Most IDS use a centralized architecture to gather and audit data. Others, as IDS in MANETs, must use a distributed and collaborative model, as it will be explained in the State of the Art section.
4. Usage frequency: An IDS can collect and analyze data at regular intervals or provide a continuous intrusion detection service. The latter is needed by MANETs as intrusions should be detected "on the fly".
5. Response to intrusions: When an intrusion is detected the system may react in different ways. Most systems generate an alarm informing the administrator, who decides of the reaction to have. A more sophisticated response consists in a corrective action (a new rule in the firewall, disconnection of suspicious connection, ...) to prevent an identical future attack.

6.2.1 MANET Intrusion Detection Architecture

The structural and behavioral differences between wired and wireless mobile networks render existing IDS designs not suitable for wireless networks. Wireless networks do not have a fixed, well-protected communication medium, therefore network monitoring should be performed at every node.

The idea here is to provide a distributed and cooperative IDS. This means that every node in the wireless ad hoc network does intrusion detection locally and independently, but neighbor nodes may help, as they investigate over a broader range [41, 7, 40].

Each agent is completely independent from the others, and monitors user and system level activities in addition to the communication activities which are in the radio range. When an anomaly is detected, if the evidence is clear enough, the node can initiate a response. On the other hand, if the evidence is inconclusive, it can start a collaborative investigation.

From the conceptual point of view six modules are defined as shown in the figure 41 according to [41, 7, 40].

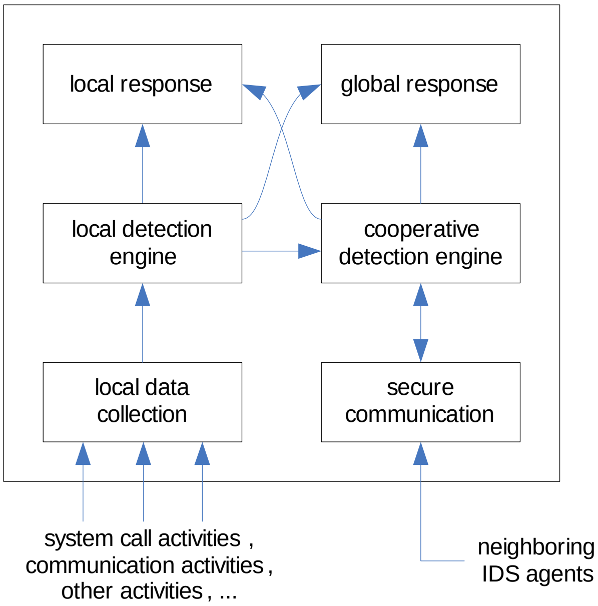


Figure 41: Modules in each IDS node [41, 7, 40].

Data Collection This module collects the real-time data from various sensors. These sensors can gather data from user and system applications; and from network packets, including those observable within the radio range of the monitoring node.

Local Detection This engine processes the data collected, looking for intrusion detection. At this step, not only misuse detection techniques should be used but anomaly detection too, as it is very probable that new attack types will be developed.

Cooperative Detection When a node detects an inconclusive intrusion, it starts a cooperative intrusion detection process. This consists of broadcasting the information about the potential intrusion to the rest of the nodes, and if other nodes find enough evidence, it starts a response.

The difference between local detection and cooperative detection is that, while in the first case the information analyzed is from the local node, in the second one an IDS agent relies on the data from other agents.

Local and Global Response The response that must be taken differs depending on the type of intrusion, the network protocols, etc. For example, a response can be to re-initialize communication channels (force re-key); or identifying the compromised nodes and re-organizing the network to exclude them.

6.2.2 Rule Based vs. Anomaly Based Intrusion Detection

Intrusion Detection Systems can be categorized into two groups: rule based and anomaly based. Rule based, or also known as signature based Intrusion Detection Systems identify intrusions by watching for patterns of traffic or application data supposed to be malicious. These type of systems are presumed to be able to detect only known attacks. However, depending on their rule set, signature based IDSs can sometimes detect new attacks which share characteristics with old attacks.

The rule based IDS analyses the information it gathers and compares it to large databases of attack signatures. Essentially, the IDS looks for a specific attack that has already been documented. Like a virus detection system, misuse detection software is only as good as the database of attack signatures that it uses to compare packets against.

MANET is still not a very widespread technology and not many MANET-specific attacks have been developed and specified as a rule set. That is one of the main reasons why, in addition to the rules based detection, an intrusion detection system based on anomaly detection methods should be also used. This approach is based on the assumption that there is a significant difference between normal user behavior and an attack which can be automatically detected. Anomaly based IDS provide a mechanism against attacks, where an intense analysis is made on a big number of features, and then AI based techniques are applied to classify the input as normal or abnormal (attack).

6.2.3 Detection of Attacks on AODV

For AODV there are a number of possibilities to recognize attacks as well. Some of them of course resemble the approaches shown for OLSR, especially those that are based on the storage and the comparison of control messages.

The identification of black hole attack is examined in [36], where the following two features for the identification of black hole attacks are developed which can, however, also indicate a wormhole attack:

- Deviations of sequence numbers: Each node stores a list of the last eavesdropped sequence numbers of the neighbors. If there are gaps, i. e. is a new sequence number is considerably larger than one overheard by this node, this might indicate an attack. An attacker would have to eavesdrop the sequence numbers himself to be successful and to avoid gaps in the sequence numbers.

- Frequency of routes: Also a strong indicator for an attack – each node stores for each other node a counter for the number of routes containing this node. If a certain node appears with above-average frequency there might be an attack.

Also suited to recognize wormholes are the aforementioned packet leashes [19]. These can be used both in (pro-)active protocols like OLSR and reactive methods like AODV, because they do not directly influence the routing but are based upon an extension of the data messages.

The recognition of a rushing attack in a MANET [20] is very difficult since the attacker behaves in compliance with the protocol and only tries to forestall others. Precautionary measures are, however, possible. A simple solution is to forward not the first control message received but to randomly select the first `n` packets instead. This increases the security, but has a negative impact on the performance of the network.

In addition, the extension of RREQ messages by a node list makes sense. The result is that a node knows its neighbors, comparable to the HELLO messages in OLSR, and can identify suspicious packets. So also packets can be identified which an attacker forwards outside his own reach with a very high transmitting power.

[9] proposes an asymmetric encoding of control messages. Furthermore, the following three indicators for attack recognition are specified:

- Distribution of wrong routes: An attacker regularly generates unnecessary route requests. Therefore an upper limit is suggested. If a node initiates a number of RREQ messages within a certain period which lies above this threshold value so this node is suspected to be an attacker.
- Denial of Service: An attacker performs a DoS attack by sending false control or data messages to paralyze the network or part of it, i. e. to take away resources from other nodes. Again the identification can be achieved by counting the control messages and comparing them to an upper limit for a certain time interval.
- Impairment of the destination: A destination cannot answer because

  1. it is not within reach of the network,
  2. it is presently overloaded,
  3. it has not received the route request for some other reason, or
  4. it acts viciously.

  The latter is assumed first if the sender gets no answer from the desired destination within a certain time. Furthermore, HELLO messages can be sent to get information about the neighborhood. If a node is recognized as belonging to the network but does not answer to route requests so it is identified as the attacker.

6.2.4 Detection of Attacks on OLSR

For the recognition of attacks in MANETs, in which OLSR is used, there are some approaches that are based on the storage of certain information that has been distributed in control messages for later evaluation and comparison with new messages.

Such an approach is introduced in [32]. The idea is that the recipient of a HELLO message will not only evaluate it but also store it. If it receives a HELLO message at time `t` and another one at a later time `t+1` this node can compare the new information with the stored one. In this way changes can be detected and the plausibility can be checked, e. g. according to the specification of OLSR a symmetric connection can never be created directly but only by a mutual exchange of HELLO messages. A respective attack in which a symmetric connection is faked by a node without a previous exchange of E HELLO messages could thus be identified by simple means.

A similar approach is pursued in [37], the evaluation based on rules refers, however, both to HELLO as well as TC messages and is much more complex. By storing the data included in control messages transferred to a node and a subsequent comparison and evaluation breaches of the rules can be detected. So, different attacks can be identified in this way. Mainly the following four rules are mentioned which must be always valid in an OLSR network:

1. Comparison of a TC message with a previous HELLO message: The nodes listed in a TC message must always be a subset of the nodes mentioned in a HELLO message according to the OLSR specification. If node P sends the messages $Hello_P = \{A, B, C, D\}$ and $TC_P = \{C, D\}$, then the relation $TC_P \subseteq Hello_P$ must apply.

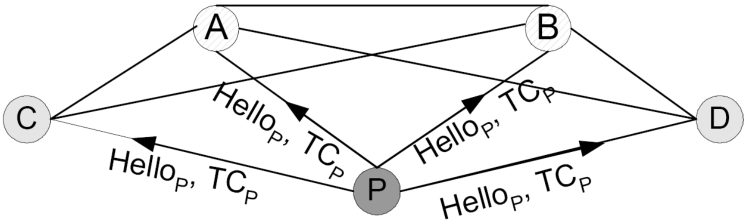


Figure 42: Rule 1 for attack identification in OLSR [37]

2. If a node receives a TC message in which it is mentioned as MPR selector the sender of the TC message must of course be a neighbor of the node, i. e. he must have been included in the last HELLO message. So if node C receives a message TC_P and notes that $C \in \{TC_P = \{C, D\}\}$ applies, then node P must be a neighbor of C. That is, $P \in \{Hello_C = \{A, B, P\}\}$.

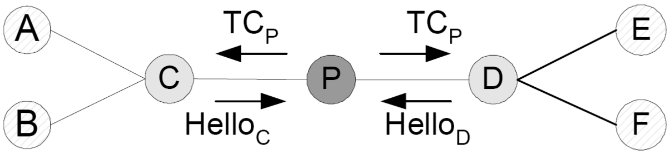


Figure 43: Rule 2 for attack identification in OLSR [37]

3. In addition: If a node receives a TC message in which it is listed as MPR selector this node must have previously selected the sender of the TC message as MPR in a HELLO message. So if $TC_P = \{C, D\}$, node P must have been selected by nodes C and D as MPR. This in turn means that $P \in MPR_C$ und $P \in MPR_D$ must apply.

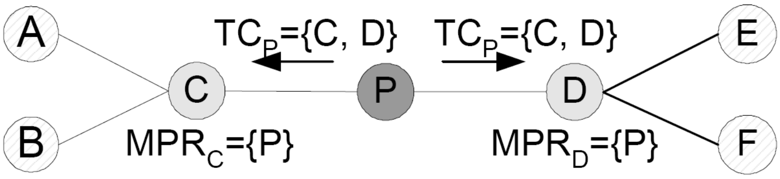


Figure 44: Rule 3 for attack identification in OLSR [37]

4. The sender of a TC message eavesdrops all related TC messages forwarded by his MPR. The MPRs only changes the sender address in the header but do not modify the content of the TC message. Node P sends $TC_P = \{C, D\}$ and nodes C and D forward TC_P for P. If the forwarded message TC_P is eavesdropped as $TC_P\,[C]$ or $TC_P\,[D]$ respectively, $TC_P\,[C] = TC_P\,[D] = TC_P$ must always apply.

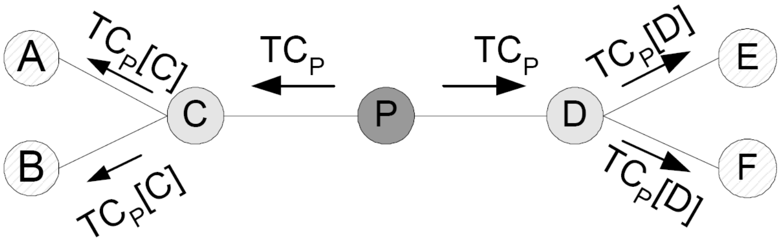


Figure 45: Rule 4 for the attack identification in OLSR [37]

Another possibility of attack recognition is a check of noticeable discrepancies of the sequence numbers (ANSN). If there are suddenly gaps in the sequence of ANSNs, i. e. the ANSN of a node is not followed by an ANSN with the next possible value, then this node is possibly the victim of an attack, e. g. an ANSN attack, where an attacker is creating and distributing unduly high ANSNs.

Another approach is pursued by *packet leashes* introduced in [19]. The notion leash means additional information is appended to the data messages. We are differentiating between two sorts of leashes.

Geographical leashes serve to limit the distance between two nodes. The sender attaches a time stamp and his location to a packet, the clock time of the nodes must be approximately synchronized. The recipient can compare his time and location information after the reception and compare it with that of the packet and decide whether these values are in a plausible proportion to each other.

Temporal leashes serve to limit the network propagation time of a packet. To all packets the sender attaches an encoded and very precise time stamp, this requires very exactly synchronized clocks on the network nodes. So the recipient can decide again whether a packet had taken an unrealistically short time to get to him. The sender can in addition define for each packet a validity time.

Another possibility, especially for recognizing wormhole attack, is the use of multiple path routing [34], in connection with statistical evaluation of all existing routes. Such an analysis deals mainly with the following two values:

- Relative frequency of each connection between two nodes
- Difference between the most frequent and second most frequent connection between two nodes

In the statistical evaluation following the collection of these values anomalies can be recognized and be further examined so that steps for the prevention of an attack can be induced.

Also [38] deals with the recognition of wormholes. His approach is the recognition of wormholes by means of graphs. As this requires location information about all nodes such an attack recognition approach is rather reasonable and possible in sensor networks than in MANETs.

7 Simulation Tools

Within this section we want to present a comparison of three simulation tools for mobile ad hoc networks. We present ns-2 and JiST/SWANS as two open source simulators and OPNET as a commercial tool. Our goal is to show similarities and differences and to point out diverse application domains.

7.1 ns-2

Network simulator 2 (ns-2) [1] is a popular and efficient simulation environment. It is a discrete event-oriented simulator for network topologies. ns-2 was meant at the beginning for wired networks; afterwards it was extended for wireless networks, including wireless LANs, mobile ad hoc networks (MANETs) and sensor networks. ns-2 is open source and freely available and supports the simulation of transport protocols such as UDP, TCP and their extensions, as well as routing and multicast protocols.

The design of ns-2 is package-event-oriented. The smallest units regarded in ns-2 are packages. Thus for example data is not regarded as constant bit stream, but in each case only complete packages are considered. An event-oriented view is made possible by the discrete view of packages. An event is defined as a package, a time stamp and the object which can be worked on. As a substantial component of this architecture the event planner steers the succession of these events and thus the simulation process.

History As a variant of the REAL network simulator, the development of ns began 1989 at the University of California in Berkeley. The original task was the simulation of dynamic aspects in package-oriented nets such as load analysis and congestion control. The first version ns-1, with which one could examine the scalability and the interaction between protocols, appeared in the year 1995 within the VINT project, with the support of several institutes: Lawrence Berkeley National Laboratory (LBNL), Xerox Palo Alto Research Center (PARC), Information Science Institute at University of Southern California (USC/ISI) and University of California Berkeley (UCB). 1996/97 appeared the second version of ns-2 with further simulation capabilities such as scheduling algorithms and support for mobile hosts.

Structure and Functionality ns-2 is a C++ based object oriented simulator, with an OTcl interpreter as front-end. This simulator supports a class hierarchy in C++ (so to say the compiled hierarchy) and a similar class hierarchy in the OTcl interpreter (thus the interpreted hierarchy).

There is a one-to-one relationship from the user perspective between each class in the interpreted hierarchy and in the compiled hierarchy (cf. figure 46).

- *Tool Command Language (Tcl)* is an interpreted script language, in which the control scripts for the simulator are written. Tcl can be installed on any Linux operating system.
- *Object Tcl (OTcl)* is an extension of the Tcl language by object-oriented capabilities; which allows the user to produce several instances of objects.
- *TclCL* is the glue between C++ and OTcl. It connects the objects that are called by the script to the objects available in C++.

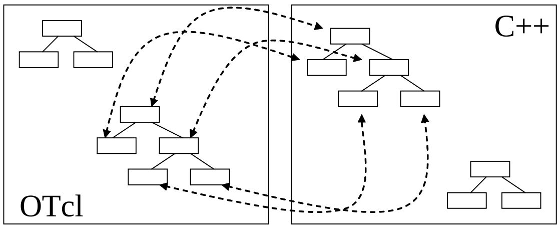


Figure 46: Structure of ns-2

The execution of a simulation is an internal computation that results in a trace file, which one can read by means of visualization tools such as NAM or iNSpect.

Visualization In order to understand better the data traffic during a network simulation a visualization tool is required. For these reasons the Network Animator (NAM) [1] was designed that provides a graphical user interface for the representation of wired network topologies. However, NAM is not suitable for wireless network simulations despite many efforts into this direction; it can only illustrate the position of nodes as well as their movements in the network.

With in NAM communications are only visualized as transferred energy pulses of one node to another when a packet is sent, besides that it neither provides packet flows nor accounting capabilities.

ad-hockey is a visualization tool for wireless simulations of ns-2. The last development and software update of this tool was performed by the developers in 1999. Therefore ad- hockey is not any more compatible with the currently used ns-2 Tcl version.

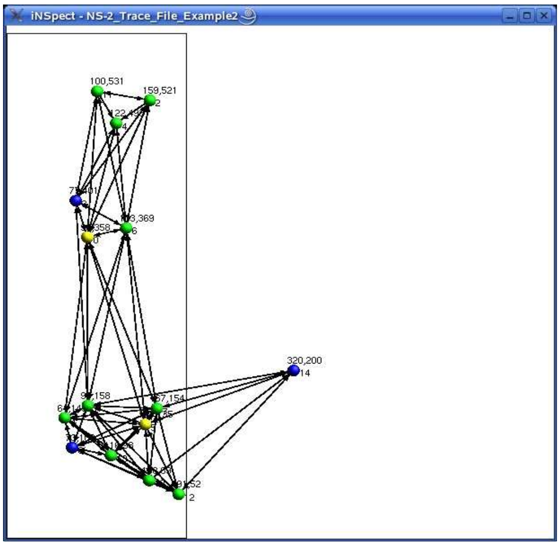


Figure 47: Visualization of Simulation Results using iNSpect

The interactive ns-2 visualization and validation environment iNSpect [24] is written in C++ and based on OpenGL. It is a visualization tool to analyze wireless networks simulated by ns-2 (see figure 47). iNSpect is platform independent and can be run under Linux, Windows, MacOS X and Cygwin.

7.2 OPNET Modeler

OPNET Modeler [28] is one of the leading commercial environments for network modeling and simulation. It is used by large technology companies for the analysis of communication networks (normal data exchange as well as routing information) and allows to model network topologies based on specific desires and requirements. OPNET Modeler supports all kinds of networks protocols, applications and technologies. With its object-oriented modeling and the graphical editors it is possible to simulate network structures and their components in such a way that they are reflected exactly by the model.

In OPNET Modeler all nodes and protocols are modeled as classes, whereby protocols are realized as finite state automata. Thus there are no events, which are activated at a fixed time; instead there is a logical succession of states. OPNET Modeler is implemented in C/C++.

Before the production of new network technologies or the implementation of new network architectures, OPNET modeler can be used to test network products in realistic scenarios to increase the product quality. OPNET Modeler can be used as well to analyze the end-to-end behavior of existing network to improve their network performance.

OPNET Editors OPNET Modeler consists of multiple hierarchical editors, which deal with networks, their equipment and protocols (cf. figure 48).

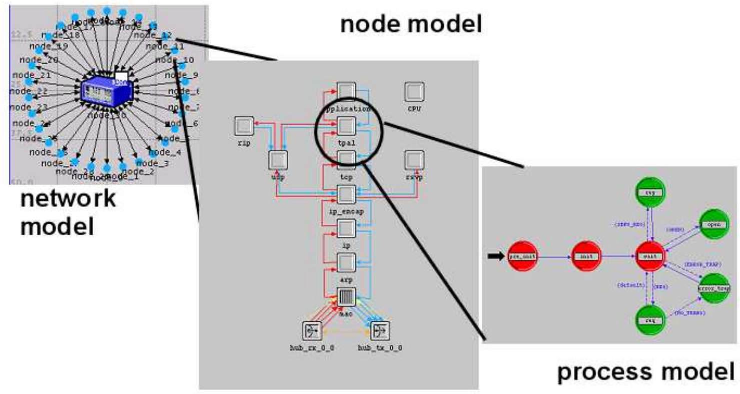


Figure 48: OPNET Modeler Hierarchy (OPNET Tutorial [28])

- *Project Editor* The Project Editor graphically represents the topology of network communications. Networks consist of node and link objects which can be configured using dialog windows. A network can be simply built by creating nodes and links using drag&drop in the editor object panel, by creating nodes and links from existing objects of the OPNETs library or by importing them from external sources.

- *Node Editor* The task of the Node Editor is to specify the architecture of the network or system by examining the data flow between the different modules. Modules are applications, protocol layers, algorithms or physical resources such as buffers or ports. Each module can generate packets, send or receive packets from other modules to enhance the node functionality.

- *Process Editor* The Process editor is used to describe the progress of processes that depend on events. Each state of a processing model is implemented in C/C++ and supported by a methodology library. Finite state machines (FSM) model protocols as well as other processes; they are dynamic and can be created by other FSMs during the simulation depending upon other events. These dynamic FSMs simplify the specification of protocols such as TCP or ATM, which have to manage several resources and sessions. New processing models can be developed and sketched with the Process Editor.

OPNET Modeler was originally developed at MIT and introduced in 1987 as the first commercial network simulator. It scales well and allows fast and efficient simulations, since it uses sophisticated acceleration techniques for wireless as well as wired networks. It offers an exact accounting of delays, availability and bit errors of network packets. Networks of thousands of nodes with dynamic applications can be simulated faster than real time on standard workstations. OPNET provides several analysis tools to better understand the simulation results, for example probability functions and parametric curves. OPNET includes tools for the animation of the simulated model during or after the simulation.

The OPNET wireless module permits the modeling of movements in mobile networks whether they are terrestrial or satellite systems. The modeling of node movements is done as three dimensional positions, which can change during the simulation.

Visualization The 3D Network Visualization Tool (3DNV), extends OPNET with the possibility to visualize mobile network performance, behavior and operation. 3DNV permits the three dimensional visualization of network simulations, network topology, links and statistics of the network performance in a realistic environment. This module receives the necessary information during simulation run time from other OPNET modules, to allow the user the maximization of some specific parameters during a mission, as well as the analysis of the network performance, e.g. ad hoc routing. 3DNV uses the industry standard data base OpenFlight for the production of three dimensional synthetic environments, in order to plot earth surfaces and network groups. The OpenFlight data base illustrates urban areas, weather as well as mobile platforms (e.g. airplanes, ships...).

7.3 JiST/SWANS

JiST/SWANS is an open source available simulation tool for mobile ad hoc networks. Completely written in Java, it was developed in 2004 in the Ph. D. thesis [8] of Rimon Barr at the School of Electrical and Computer Engineering of Cornell University. While the basic functionalities and protocol implementations are similar to those of the simulation tools described so far, JiST/SWANS does not provide the wide range of features known from NS2 or OPNET. Nevertheless, we used JiST/SWANS for part of our simulation studies, since it has shown to be easy extendable and to provide good scalability and performance. Also for our future work in the scope of SicAri as described in Section refsec:outlook, JiST/SWANS offers required features that are not (at least not with comparable effort) available in NS2 or OPNET. In the following we shortly explain the two main components JiST and SWANS.

7.3.1 JiST

JiST (Java in Simulation Time) builds the fundament of JiST/SWANS. Here, the execution of native Java code in a discrete, event based simulation time is enabled. The necessary steps to bring simulation time into Java programs are shown in Figure 49. In a first step, native Java code is tagged using the JiST API, to control the simulation time flow. The resulting code is compiled using the standard Java compiler. During the execution of the Java code, the bytecode that was generated in the first step is rewritten by the modified, rewriting JiST classloader. In this second step, the encapsulation of objects into entities with independent simulation time flows is done. The rewritten bytecode is executed in step three by the JiST simulation kernel which is running in a standard Java Virtual Machine.

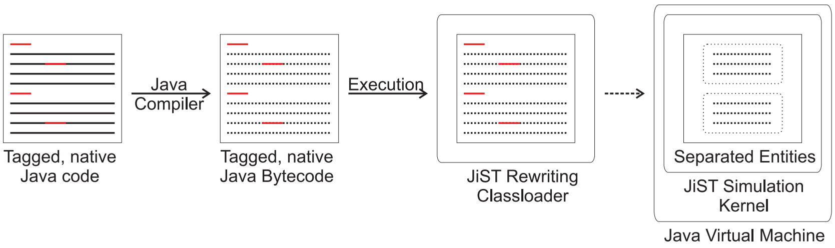


Figure 49: Compilation and execution of Java code with simulation time

The main concept of JiST for the execution of Java programs in simulation time is the encapsulation of objects in JiST-entities. Within one Entity, the code is executed linearly as it holds for any Java program. Simulation time of entities is synchronized on method calls between entities. This means, that a method call on an entity is scheduled until the respective entity has reached the same simulation time as the entity that originated the call. Figure 50 gives an example that shows the communication between two entities. Note that the simulation time of entities passes independently from each other, until a method invocation occurs.

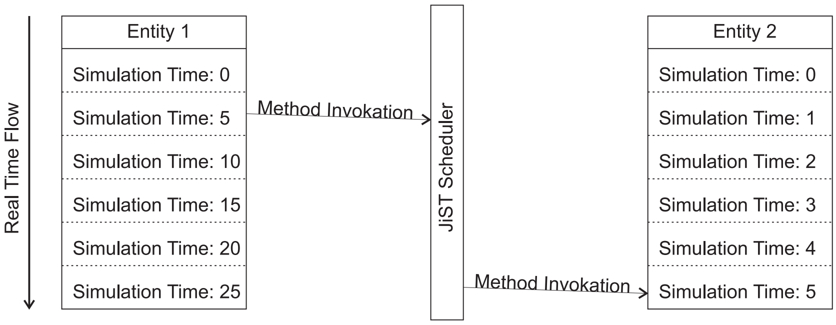


Figure 50: Compilation and execution of Java code with simulation time

7.3.2 SWANS

SWANS (Scalable Wireless Ad Hoc Network Simulator) is built on top of JiST, using the offered primitives for the invocation of simulation time. As it holds for the TCP/IP layer model, SWANS offers a layered architecture for the assembling of devices that form the simulated network. A sketch of this architecture in comparison to the TCP/IP model is shown in Figure 51. The main entities, that correspond to the layers of the TCP/IP model are shortly described in the following.

Field The SWANS field entity is responsible for the simulation of the physical properties of the ad hoc network as a whole. Placement and mobility of nodes as well as signal propagation and fading as properties of the wireless transmission channel are handled here.

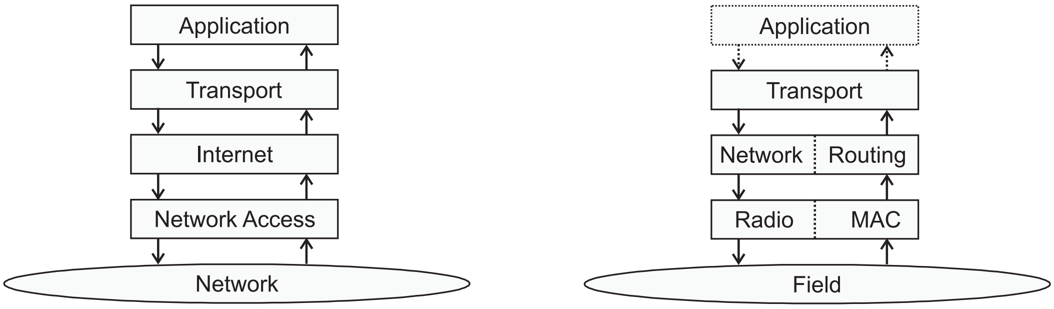
Radio Within the radio entity, radio interfaces of nodes with associated attributes like interference and error models, transmission frequency and power, bandwidth, and antenna sensitivity are modeled. The radio entity corresponds to the network access layer of the TCP/IP model.

MAC The MAC entity provides an implementation of the 802.11 wireless LAN MAC protocol. Together with the radio entity, the mac entity corresponds to the network access layer of the TCP/IP model.

Network The network entity offers an implementation of IPv4 as a protocol of the Internet layer of the TCP/IP model.

Routing Together with the network entity, the routing entity belongs to the Internet layer of the TCP/IP model. The offered routing protocols are Zone Routing (ZRP), Dynamic Source Routing (DSR), and Ad Hoc On-demand Distance Vector (AODV) routing.

Transport The transport entity of SWANS corresponds to the transport layer of the TCP/IP model. Implemented protocols are TCP and UDP.



(a) TCP/IP model

(b) SWANS entities

Figure 51: TCP/IP model and SWANS implementation

7.4 Comparison of the Simulation Tools

Within this section we present the results of a comparison of the two introduced open source simulation tools: ns-2 and JiST/SWANS. For this, we first design a common evaluation scenario and select the values that should be compared for the different tools.

7.4.1 Configuration of the Comparison Scenario

Routing Protocol AODV

Number of Nodes and Simulation Field Size The size of the field is selected in such a way that meaningful data for the average number of neighbors per nodes and thus the probability of success (route found) result. Each node has on the average seven to eight neighbors.

Number of Nodes	Field Size
50	1000 m x 1000 m
100	1500 m x 1500 m
150	1850 m x 1850 m
200	2150 m x 2150 m
250	2400 m x 2400 m
300	2700 m x 2700 m
350	2900 m x 2900 m
400	3100 m x 3100 m
450	3300 m x 3300 m
500	3500 m x 3500 m

The number of nodes is increased in steps of 50 from 50 nodes up to 500 nodes.

Mobility Model Random Way-point with the following parameters is selected as mobility model:

- no pause time
- 1 meter steps
- 1 m/s minimum speed
- 15 m/s maximum speed

Simulation Time Every simulation shall last for 300 seconds (simulated time).

Traffic Pattern As data traffic a *Constant Bit Rate (CBR)* traffic with 30 UDP packets per minute and transmitter is selected. Each UDP packet contains 1400 bytes of payload data.

- five communicating pairs of nodes (transmitter/receivers) (independently of number of nodes)
- number of pairs of transmitter/receivers = 10 percent of the number of nodes, i. e. 5 pairs for 50 nodes, 10 pairs for 100 nodes...

Transmission Range As radio transmission range 250 meters are to be used with spherical signal dissemination.

Measured variables For each simulation (at least) the following values are to be determined:

- duration of the simulation
- percentage of routing packets in respect to the total number of transmitted packets
- average number of hops per transferred packet
- percentage of successfully transferred packets
- percentage of successfully found routes
- number of RERR messages (number of broken routes)

7.4.2 Comparison Results

We now present the results of our comparison study. While we expect the simulation time to show significant differences between the three introduced simulation tools, all other compared values reflect the behavior of AODV and therefore should be of the same magnitude for the different implementations.

ns-2 Table 1 shows the results of the comparison scenario with low network load (5 communicating pairs of nodes) for ns-2.

The simulation time increases exponentially, it more than doubles from scenario to scenario, i.e. for each increase of the number of nodes by 50. Since the probability that a route breaks gets higher, the more (mobile) nodes are contained in this route, the number of route error messages increases together with the average number of hops. Each time a route breaks, a new one has to be discovered for the respective communicating pair of nodes, so along with the number of route error messages the number of route requests and with this the fraction of routing packets (routing overhead) increases. As described in Section 7.4.1 the size of the simulation field is adapted to the number of nodes to obtain a comparable connectivity for all setups. Therefore, the fraction of successful route requests is (more or less) stable around a mean value of 90%.

The results for the scenario with high network load (10% communicating pairs of nodes) is shown in Table 2. Like in the previous scenario, the simulation time increases exponentially, it more than doubles from scenario to scenario, i.e. for each increase of the number of nodes by 50. The other statistics behave in a similar way as in the scenario described above. The increase is, however, a lot steeper since there are more communicating nodes and therefore the network is more congested. This causes a higher number of broken routes and therefore a higher number of RREQs.

Number of Nodes	50	100	150	200	250
Communicating Pairs	5	5	5	5	5
Messages to Transfer	750	750	750	750	750
Fraction of Transfered Messages	86.67%	81.07%	80.53%	79.60%	74.93%
Total Amount of Packets	975	1261	1293	1293	1712
Fraction of Routing Packets	23.08%	40.52%	42.00%	42.00%	56,19%
Number of RREQs	32	95	108	101	187
Successful RREQs	28	83	98	80	160
Number of RERRs	165	333	337	362	615
Average Hops	4,05	5,10	4,95	4.89	6.20
Simulation Time	14	69	347	1380	2899

Table 1: ns-2 Results for Low Network Load

Number of Nodes	50	100	150	200	250
Communicating Pairs	5	10	15	20	25
Messages to Transfer	750	1500	2250	3000	3750
Fraction of Transfered Messages	86.67%	78.33%	79.29%	68.43%	71.73%
Total Amount of Packets	975	2591	4192	7062	8895
Fraction of Routing Packets	23.08%	42.11%	46.33%	57.52%	57.84%
Number of RREQs	32	235	411	879	986
Successful RREQs	28	198	343	682	778
Number of RERRs	165	658	1188	2501	3381
Average Hops	4,05	4,89	5,54	6,00	6.02
Simulation Time	13	118	553	2050	4590

Table 2: ns-2 Results for High Network Load

JiST/SWANS Table 3 shows the results of the comparison scenario with low network load (5 communicating pairs of nodes) for JiST/SWANS. The duration of the simulation increases (nearly) linearly from 117 seconds for the scenario which consists of 50 nodes to 1687 seconds for 500 nodes. Since the probability that a route breaks gets higher, the more (mobile) nodes are contained in this route, the number of route error messages increases together with the average number of hops. Each time a route breaks, a new one has to be discovered for the respective communicating pair of nodes, so along with the number of route error messages the number of route requests and with this the fraction of routing packets (routing overhead) increases. As described in Section 7.4.1 the size of the simulation field is adapted to the number of nodes to obtain a comparable connectivity for all setups. Therefore, the fraction of successful route requests is (more or less) stable around a mean value of 90%.

The results for the scenario with high network load (10% communicating pairs of nodes) is shown in Table 4. Like in the previous scenario, a (nearly) linear increase of the simulation time from 117 seconds for 50 nodes to 2816 seconds for 400 nodes can be observed. The rapid increase beginning at 400 nodes can be explained by a congested network which causes packet loss due to the limited capacity of routing queues at the nodes. The congestion is therefore also reflected in the fraction of successfully transferred messages, that decreases (nearly) linearly from 72% (50 nodes) to 45% (400 nodes) and then rapidly below 10%.

Number of Nodes	50	100	150	200	250
Communicating Pairs	5	5	5	5	5
Messages to Transfer	750	750	750	750	750
Fraction of Transfered Messages	72,40%	66,67%	68,27%	59,73%	62,00%
Total Amount of Packets	971	1069	1126	1162	1163
Fraction of Routing Packets	22,76%	29,84%	33,39%	35,46%	35,51%
Number of RREQs	76	108	128	145	140
Successful RREQs	74	108	125	127	138
Number of RERRs	71	103	123	140	135
Average Hops	3,20	4,90	6,00	6,50	7,70
Simulation Time / s	117	279	364	467	555

Number of Nodes	300	350	400	450	500
Communicating Pairs	5	5	5	5	5
Messages to Transfer	750	750	750	750	750
Fraction of Transfered Messages	55,73%	57,87%	52,93%	49,07%	51,87%
Total Amount of Packets	1187	1202	1229	1223	1249
Fraction of Routing Packets	36,82%	37,60%	38,97%	38,68%	39,95%
Number of RREQs	153	158	169	168	171
Successful RREQs	136	141	146	142	162
Number of RERRs	148	153	164	163	166
Average Hops	8,70	9,30	9,80	10,50	11,10
Simulation Time / s	907	966	1181	1390	1687

Table 3: JiST/SWANS Results for Low Network Load

Number of Nodes	50	100	150	200	250	300	350	400	450	500
Communicating Pairs	5	10	15	20	25	30	35	40	45	50
Messages to Transfer	750	1500	2250	3000	3750	4500	5250	6000	6750	7500
Fraction of Transfered Messages	72,40%	63,73%	61,82%	53,77%	52,99%	43,78%	37,64%	45,07%	9,32%	4,75%
Total Amount of Packets	971	2242	3215	4548	5770	6955	8115	9361	9849	11168
Fraction of Routing Packets	22,76%	33,10%	30,02%	34,04%	35,01%	35,30%	35,30%	35,90%	31,47%	32,84%
Number of RREQs	76	257	295	537	687	852	1005	1178	1345	1720
Successful RREQs	74	238	390	494	671	781	890	1045	454	278
Number of RERRs	71	247	280	517	662	822	970	1138	1300	1670
Average Hops	3,20	4,90	6,00	6,50	7,70	8,70	9,30	9,80	10,50	11,10
Simulation Time / s	117	601	773	1133	1406	2561	3143	2816	12870	17316

Table 4: JiST/SWANS Results for High Network Load

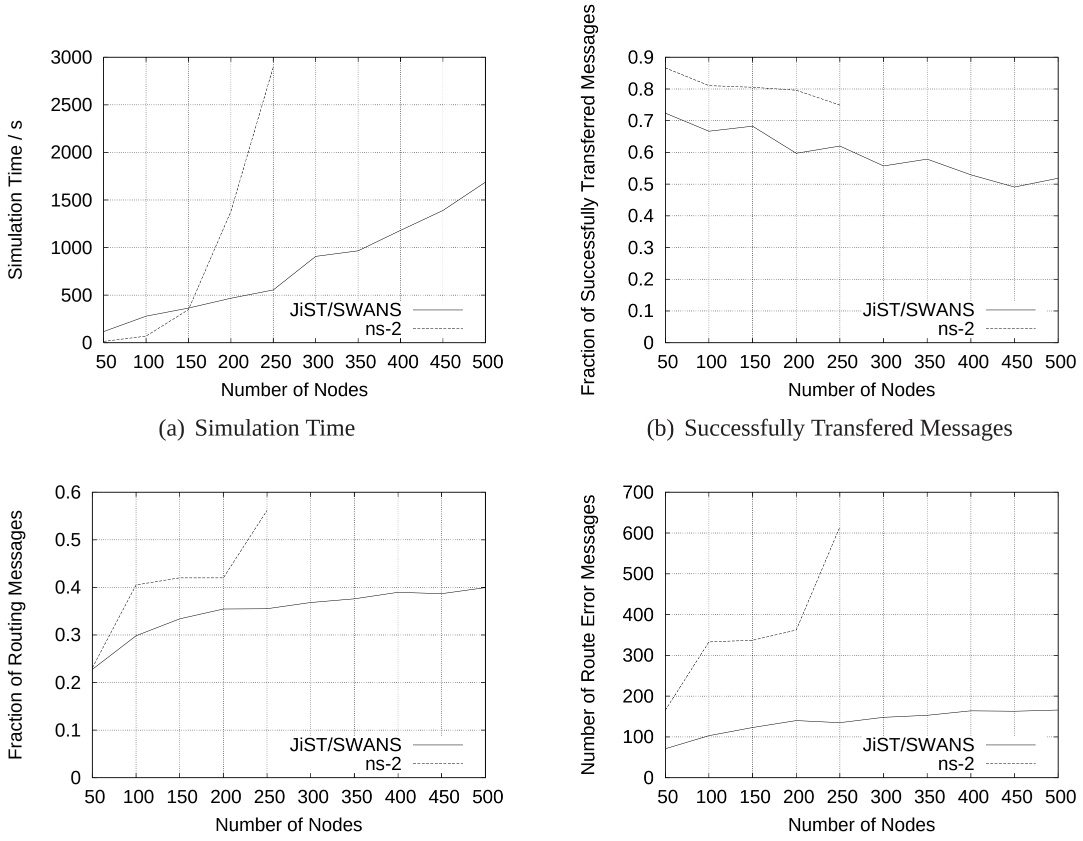
Summary A graphical summary of our comparison study is shown in figures 52 and 53 for low and high network load. As we expected, the simulation time as depicted in figures 52(a) and 53(a) shows significant differences for the individual simulation tools.

The most obvious difference can be seen in figures 52(a) and 53(a). While the simulation time increases linear with the number of nodes for JiST/SWANS it shows a more or less exponential behavior for ns-2.

Figures 52(b) and 53(b) show the fraction of successful transferred messages which behave in a similar way, although the values for ns-2 are 15 to 20 percent higher than for JiST/SWANS.

In Figures 52(c) and 53(c) the routing overhead as the fraction of routing messages out of the total amount of transferred messages is depicted. Here it can be seen that in the simulation with ns-2 the routing overhead increases a lot faster than for JiST/SWANS. This behavior is also reflected in figures 52(d) and 53(d) which show the number of originated route error messages.

The most obvious difference and for practical reasons maybe the most important one is the big difference in the increase of the simulation time with the number of nodes. Therefore the simulation tool JiST/SWANS has a big advantage, at least for MANET simulations with a big number of network nodes.



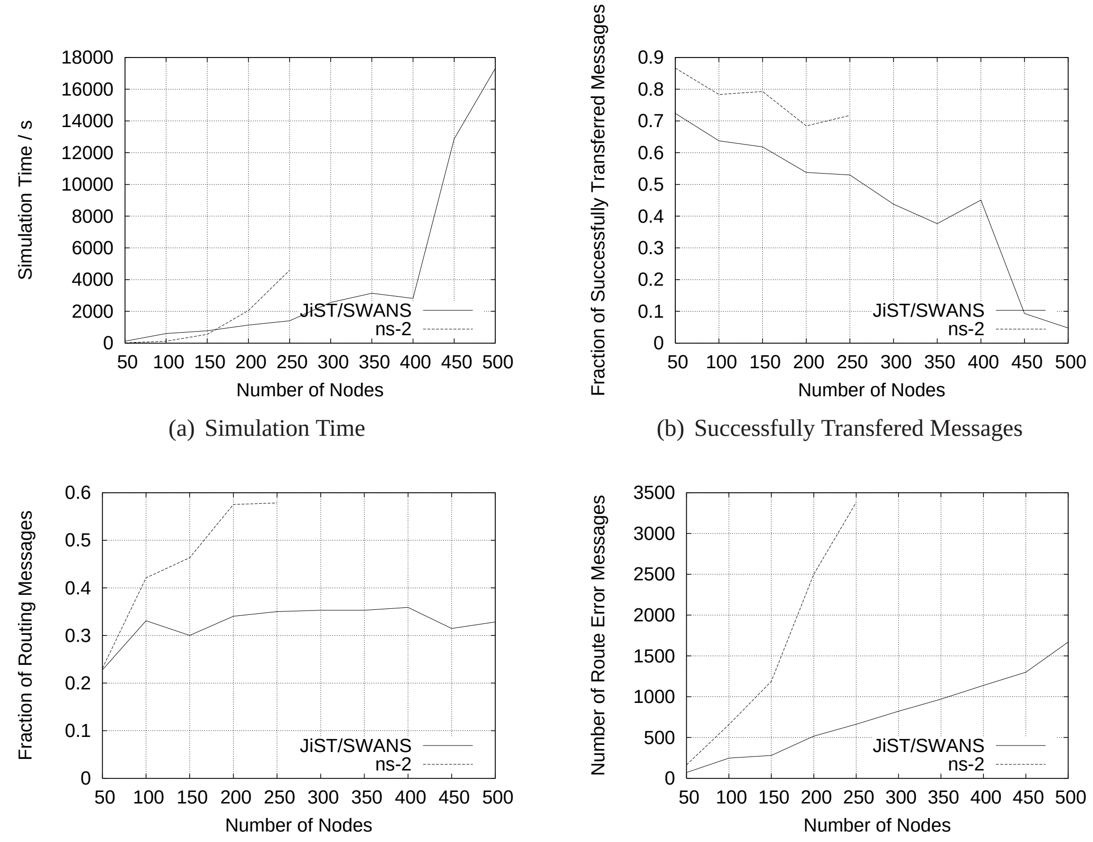
(a) Simulation Time

(b) Successfully Transfered Messages

(c) Fraction of Routing Messages (Routing Overhead)

(d) Number of Broken Routes

Figure 52: Comparison Results for 5 Communicating Pairs of Nodes



(a) Simulation Time

(b) Successfully Transfered Messages

(c) Fraction of Routing Messages (Routing Overhead)

(d) Number of Broken Routes

Figure 53: Comparison Results for 10% Communicating Pairs of Nodes

8 Simulation and Evaluation

8.1 Overview

We apply the methodology proposed by Jain [22] for the experimental analysis of our approach and adopt the individual steps to our scenario described in Section 3.2.1. The methodological steps can be summarized as follows:

- Definition of the system, goals, and services
- Selection of the metrics
- Definition of the parameters to study
- Selection of the factors/elements of the parameter set
- Choice of the evaluation technique
- Selection of the workload
- Design of the individual experiments
- Analysis and interpretation of the obtained data
- Presentation of the results

8.2 Simulation Settings

A detailed description of the simulation setup regarding the dimensions is shown in Figure 54. We consider a mobile ad hoc network of 3000 meters width and 2000 meters height. The insecure area is situated at the center of the bottom line with a width of 1000 meters and a height of 500 meters. The radio range of 250 meters is equal for each node within our scenario. No packets are lost during transmission.

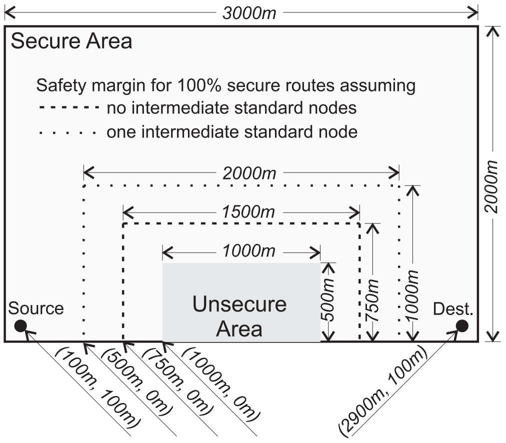


Figure 54: Dimensions of our scenario

To obtain a worst-case scenario for our approach, we place the source and the destination in the lower left and the lower right corner of the mobile ad hoc network. If successful, our approach discovers a route which bypasses the insecure area, whereas we expect standard DSR to discover a route straight through the insecure area, as we already drafted in Figure 20.

In a first step, we compare the connectivity of standard DSR to the connectivity of our approach. As a metric for this we use the fraction of successful route requests out of the number of total route requests. A statistical mean value is determined during 1000 simulation runs with random node placement and one route request each. Sender and receiver are placed at the fixed positions as shown above. Each extended node with a position outside the inner safety margin shown in Figure 54 (so at least the sending range of 250 meters away from the insecure area) is allowed to be contained in a route.

For this evaluation, the parameters for each 1000 runs are

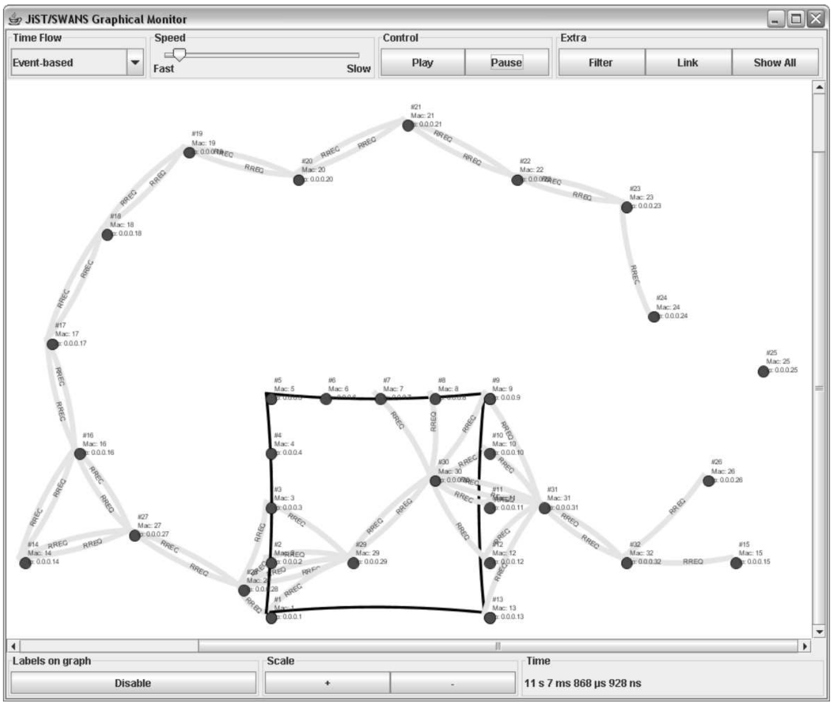
- the total number of nodes in the scenario,
- the fraction of extended nodes, and
- the number of intermediate standard nodes between two adjacent extended nodes.

In the second step, we quantitatively evaluate the degree of security that is reached, if we allow one intermediate standard node to be situated between two adjacent extended nodes. For this, we choose a fixed parameter set, that showed to achieve 100% connectivity in the first step of evaluation described above. We then introduce a second safety margin, as shown in Figure 54. No extended node that is situated within this area is allowed to forward messages. It is obvious that if this second safety margin then has the size of 250 meters (measured from the inner margin), no messages reach the insecure area, since the sending range of an extended node and a following standard node can at most be 500 meters.

For the second step, the parameter that is considered in simulation is the width of the outer safety margin. Again, we perform 1000 simulation runs with random node placement and one route request each. Sender and receiver stay fixed at the positions shown in Figure 54. To check whether a discovered route is in reception range of the insecure area, we modeled listening nodes for JiST/SWANS. These are placed with a distance of 100 meters along the boundaries of the insecure area. The listening nodes do not forward any messages and thus have no effect on the route discovery process. The metric to measure the security of a route is the fraction of route request messages that are received by the listening nodes around the insecure area out of the total amount of route request messages that are sent during the route request phase.

8.3 Evaluation Results

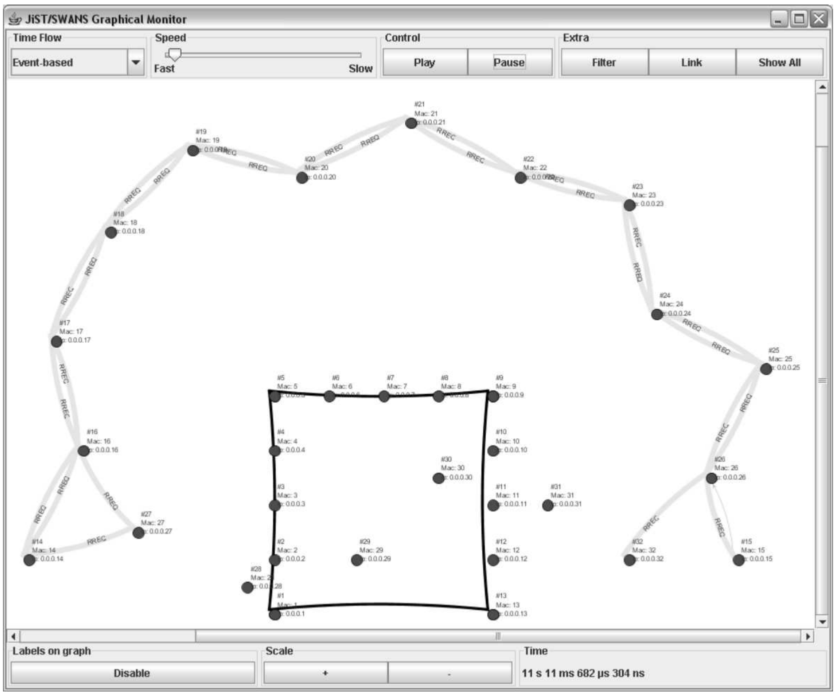
To graphically show that our approach works in general, Figures 55 and 56 provide screenshots of the route request phases of our approach and DSR. The visualization is done with a graphical monitoring tool for the JiST/SWANS simulator which has been developed as a part of the SicAri project. We model a static simulation setup, that is similar to the scenario as depicted in Figure 20.



Time Flow: Event-based

Time: 11 s 7 ms 868 µs 928 ns

Figure 55: Standard DSR route request



Time Flow: Event-based

Time: 11 s 11 ms 682 µs 304 ns

Figure 56: Restricted route request

In Figures 57 and 58, the results of the first step of the evaluation of our approach are shown. Figure 57 depicts the evaluation of the connectivity of our approach and of DSR. For this evaluation we used homogeneous setups, which consist either of standard DSR nodes or of extended nodes.

As we expected, the price for a secure route is a decrease in connectivity of our approach compared to standard DSR. The mean decrease in connectivity for our scenario shows to be approximately 20%. The elaboration of the reason for this will be part of our future work.

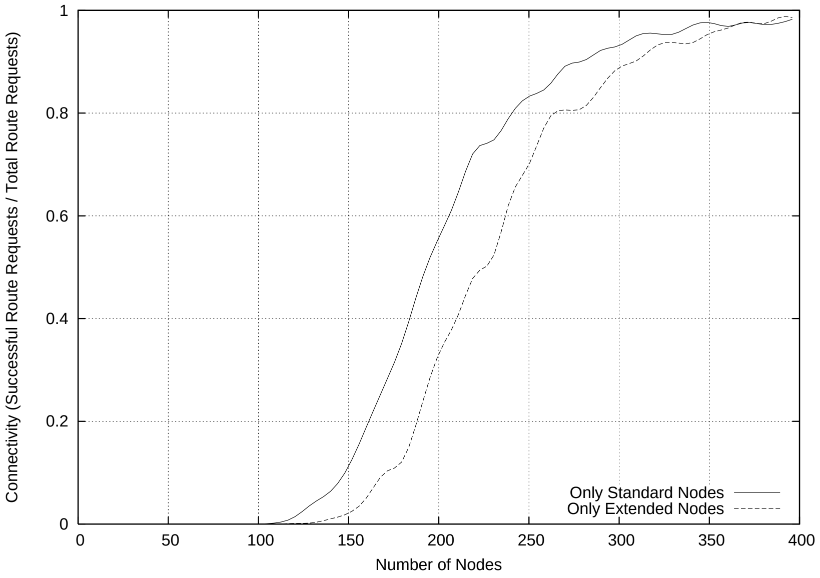


Figure 57: Connectivity of DSR and our approach

For the evaluation shown in Figure 58, a heterogeneous setup of standard and extended nodes is considered. Depicted is the connectivity of routes which only consist of extended nodes and of routes which may contain one intermediate standard node between adjacent extended nodes. We simulate a random distribution of 400 nodes and stepwise increase the fraction of extended nodes.

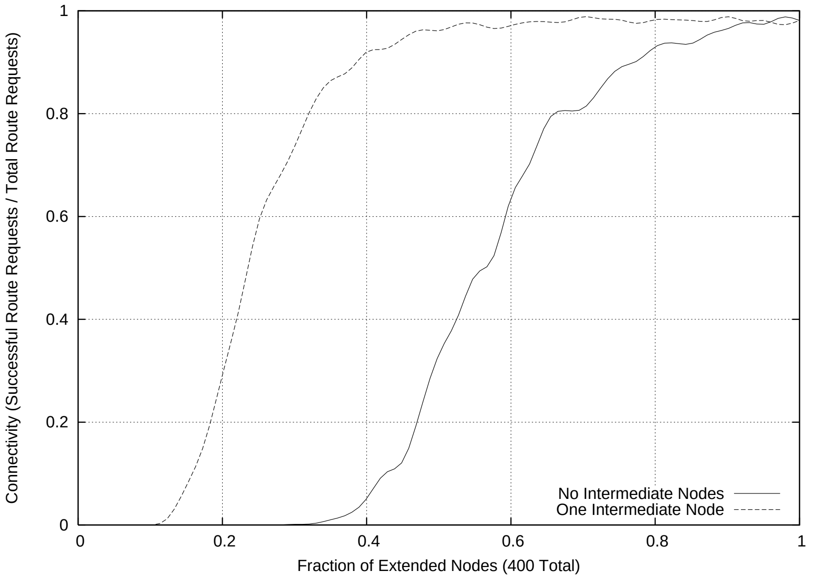


Figure 58: Connectivity for routes with none and one intermediate standard node

In Figure 59 the results of the second step of our evaluation are presented. We simulate a setup with 200 standard nodes and 200 extended nodes. One intermediate standard node is allowed to be situated between two adjacent extended nodes. The size of the outer safety margin is increased stepwise.

Like presumed, the degree of security of a route decreases if we allow standard nodes to be contained in a route and along with this reduce the size of the outer safety margin. The explanation for this observation is that like DSR, our approach will most likely find the shortest route with respect to number of hops. This route is most likely the one with the shortest distance to the insecure area. Therefore, as shown in Figure 39, the transmission of an intermediate standard node can reach the insecure area, if the distance between the neighboring extended nodes is smaller than the sum of their radio ranges. In this evaluation, our approach always performs better than standard DSR. For our scenario, DSR shows to have a constant rate of 50% intercepted route request messages out of the total amount of transmitted route request messages.

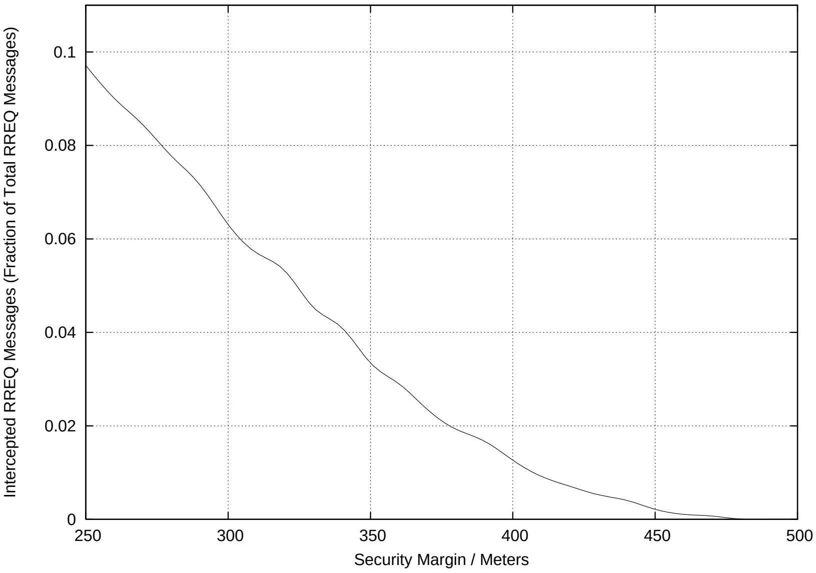


Figure 59: Quantitative security of our approach

9 Outlook

Routing with regard to the geographical position of nodes with extended functionality and number of intermediate nodes with unchanged functionality as two degrees of freedom have been our concerns for until now the introduced mechanism to prevent attacks. Our scenario quickly reaches a high complexity when we take more than two levels of trustworthiness into account. Also a dynamic change of these levels may be necessary, when a trustworthy node enters the insecure area. Furthermore, non-trustworthy nodes may be allowed to move relatively unrestricted throughout the research site. These more realistic and thus more complex scenarios will be the focus of our future research.

Within these scenarios, also temporal aspects will be part of our research. As an example we plan to delay sending on transport layer in order to prevent a high security level node from transmitting while in proximity to a low security level node. We furthermore will consider the adaptation of the transmission power of nodes to increase the number of possible routing devices and with this further improve connectivity.

References

[1] Network Simulator ns-2. http://www.isi.edu/nsnam/ns/.

[2] Ethereal - the world's most popular network protocol analyzer. *www.ethereal.com*, 2006.

[3] Imad Aad, Jean-Pierre Hubaux, and Edward W. Knightly. Denial of Service Resilience in Ad Hoc Networks. In *Proceedings of the 10th Annual International Conference on Mobile Computing and Networking*, pages 202–215, Philadelphia, PA, USA, September 2004. ACM Press.

[4] E. Aitenbichler and M. Mühlhäuser. Audiobasierte Endgeräte für Ubiquitous Computing und geeignete Infrastrukturen. *Praxis der Wirtschaftsinformatik: Ubiquitous Computing*, 229:68–80, 2002.

[5] Mohammad Al-Shurman, Seong-Moo Yoo, and Seungjin Park. Black Hole Attack in Mobile Ad Hoc Networks. In *Proceedings of the 42nd Annual ACM Southeast Regional Conference*, pages 96–97, Huntsville, AL, USA, April 2004. ACM Press.

[6] Patrick Albers, Olivier Camp, Jean-Marc Percher, Bernard Jouga, Ludovic Mé, and Ricardo Staciarini Puttini. Security in Ad Hoc Networks: a General Intrusion Detection Architecture Enhancing Trust Based Approaches. In *Wireless Information Systems*, pages 1–12, 2002.

[7] Yi an Huang and Wenke Lee. A Cooperative Intrusion Detection System for Ad Hoc Networks. In *Proceedings of the 1st ACM workshop on Security of Ad Hoc and Sensor Networks*, pages 135–147, Fairfax, VA, USA, October 2003. ACM Press.

[8] Rimon Barr. *JiST - An efficient, unifying approach to simulation using virtual machines*. PhD thesis, Cornell University, 2004.

[9] Sonali Bhargava and Dharma P. Agrawal. Security Enhancements in AODV protocol for Wireless Ad Hoc Networks. In *Vehicular Technology Conference*, Center for Distributed and Mobile Computing Department of ECECS, University of Cincinnati, 2001.

[10] Gilles Brassard. Quantum computing: the end of classical cryptography? *SIGACT News*, 25(4):15–21, 1994.

[11] Tobias Bucher. Modellierung und Analyse von Angriffen auf Routingverfahren in mobilen Ad-hoc-Netzen, Diploma Thesis, December 2005.

[12] Bundesministerium für Gesundheit. Die gesundheitskarte. *www.die-gesundheitskarte.de*, 2006.

[13] T. Clausen and P. Jacquet. OLSR - Request For Comments, RFC3626, October 2003. http://ietf.org/rfc/rfc3626.txt.

[14] Marco Conti, John Crowcroft, Gaia Maselli, and Giovanni Turi. *Handbook on Theoretical and Algorithmic Aspects of Sensor, Ad Hoc Wireless, and Peer-to-Peer Networks*, chapter A Modular Cross Layer Architecture for Ad Hoc Networks. CRC Press, 2005.

[15] Marco Conti, Enrico Gregori, and Giovanni Turi. A cross-layer optimization of gnutella for mobile ad hoc networks. In *MobiHoc ’05: Proceedings of the 6th ACM international symposium on Mobile ad hoc networking and computing*, pages 343–354, New York, NY, USA, 2005. ACM Press.

[16] J. Douceur. The Sybil Attack. In *Proceedings of the IPTPS02 Workshop*, Cambridge, MA (USA), March 2002.

[17] Electronic Frontier Foundation. *Cracking DES - Secrets of Encryption Research, Wiretap Politics & Chip Design*. O’Reilly, 1998.

[18] David Henry. Who’s got the key? In *SIGUCCS ’99: Proceedings of the 27th annual ACM SIGUCCS conference on User services*, pages 106–110, New York, NY, USA, 1999. ACM Press.

[19] Yih-Chun Hu, Adrian Perrig, and David B. Johnson. Packet Leashes: A Defense against Wormhole Attacks in Wireless Networks. In *INFOCOM*, 2003.

[20] Yih-Chun Hu, Adrian Perrig, and David B. Johnson. Rushing Attacks and Defense in Wireless Ad Hoc Network Routing Protocols. In *Proceedings of the 2003 ACM Workshop on Wireless Security*, pages 30–40, San Diego, CA, USA, September 2003. ACM Press.

[21] ISO - International Organization for Standardization. Iso 7498-2: Information processing systems – open systems interconnection – basic reference model – part 2: Security architecture. 2000.

[22] Raj Jain. *The Art of Computer Systems Performance Analysis: Techniques for Experimental Design, Measurement, Simulation, and Modeling*. Wiley, 1991.

[23] David B. Johnson, David A. Maltz, and Yih-Chun Hu. The dynamic source routing protocol for mobile ad hoc networks. *IETF MANET Working Group INTERNET-DRAFT*, draft-ietf-manet-dsr-10.txt, 2004.

[24] S. Kurkowski, T. Camp, N. Mushell, and M. Colagrosso. A visualization and analysis tool for NS-2 wireless simulations: iNSpect. In *Proceedings of the IEEE International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems (MASCOTS)*, pages 503–506, Atlanta, Georgia, 2005.

[25] Alfred J. Menezes, Scott A. Vanstone, and Paul C. Van Oorschot. *Handbook of Applied Cryptography*. CRC Press, Inc., Boca Raton, FL, USA, 1996.

[26] James Newsome, Elaine Shi, Dawn Song, and Adrian Perrig. The Sybil Attack in Sensor Networks: Analysis & Defenses. In *Proceedings of the Third International Symposium on Information Processing in Sensor Networks*, pages 259–268, Berkeley, CA, USA, April 2004. ACM Press.

[27] R. Ogier, F. Templin, and M. Lewis. TBRPF - Topology Dissemination Based on Reverse-Path Forwarding, RFC3684, February 2004. http://ietf.org/rfc/rfc3684.txt.

[28] OPNET. OPNet Modeler. http://www.opnet.com/products/modeler/home.html.

[29] C. Perkins, E. Belding-Royer, and S. Das. AODV - Request For Comments, RFC3561, July 2003. http://ietf.org/rfc/rfc3561.txt.

[30] C. E. Perkins and E. M. Royer. Ad-hoc On-Demand Distance Vector Routing. In *Proceedings, Second IEEE Workshop on Mobile Computing Systems and Applications (WMCSA '99)*, pages 90–100, New Orleans, LA, USA, February 1999. IEEE Press.

[31] Daniele Raffo. *Security Schemes for the OLSR Protocol for Ad Hoc Networks*. PhD thesis, Université Paris 6, September 2005. http://perso.crans.org/ raffo/papers/raffo-phdthesis.pdf.

[32] Daniele Raffo, Cédric Adjih, Thomas Clausen, and Paul Mühlethaler. An Advanced Signature System for OLSR. In *Proceedings of the 2nd ACM Workshop on the Security of Ad Hoc and Sensor Networks*, pages 10–16, Washington D.C., USA, October 2004. ACM Press.

[33] Flora Rheta Schreiber. *Sybil: The True Story of a Woman Possessed by 16 Separate Personalities*. Henry Regnery Co., Chicago, IL, USA, 1st edition, 1973.

[34] Ning Song, Lijun Qian, and Xiangfang Li. Wormhole Attacks Detection in Wireless Ad Hoc Networks: A Statistical Analysis Approach. In *Proceedings of the 19th IEEE International Parallel and Distributed Processing Symposium (IPDPS'05)*, 2005.

[35] Ralf Steinmetz and Klaus Wehrle, editors. *Peer-to-Peer Systems and Applications*. Springer, 2005.

[36] H. Chris Tseng and B. Jack Culpepper. Sinkhole intrusion in mobile ad hoc networks: The problem and some detection indicators. *Computers & Security*, 24:561–570, 2005.

[37] M. Wang, L. Lamont, P. Mason, and M. Gorlatova. An Effective Intrusion Detection Approach for OLSR MANET Protocol. In *First Workshop on Secure Network Protocols (NPSec)*, Boston, Massachusetts, USA, November 2005. First Workshop on Secure Network Protocols (NPSec).

[38] Weichao Wang and Bharat Bhargava. Visualization of Wormholes in Sensor Networks. In *Proceedings of the 2004 ACM Workshop on Wireless Security*, pages 51–60, Philadelphia, PA, USA, October 2004. ACM Press.

[39] Bing Wu, Jianmin Chen, Jie Wu, and Mihaela Cardei. *A Survey on Attacks and Countermeasures in Mobile Ad Hoc Networks*, chapter 12. Springer, 2006.

[40] Yongguang Zhang and Wenke Lee. *Security in Mobile Ad-hoc Networks*, pages 249–268. Springer, 2005.

[41] Yongguang Zhang, Wenke Lee, and Yi-An Huang. Intrusion Detection Techniques for Mobile Wireless Networks. *Wireless Networks*, 9(5):545–556, September 2003.