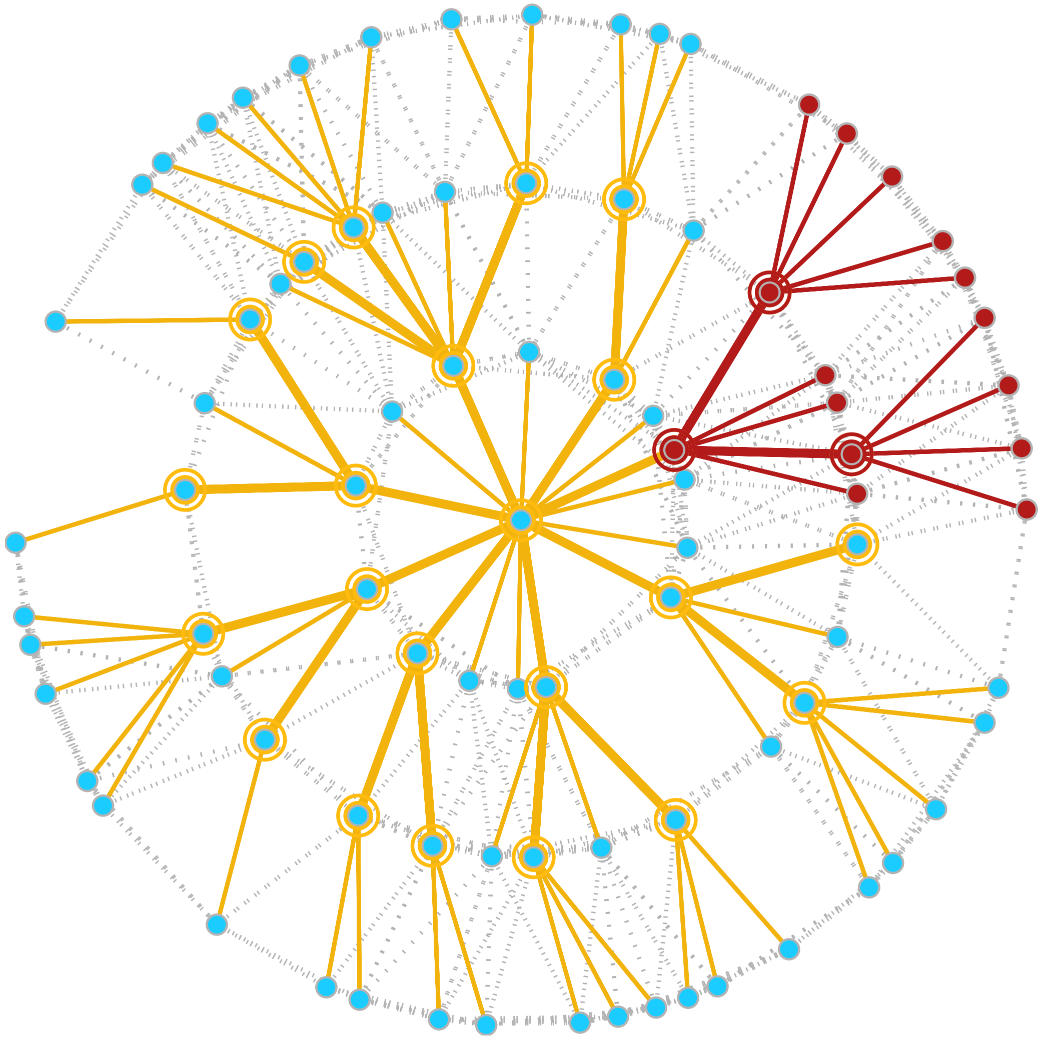
FOI
Swedish Defence Research Agency
Command and Control Systems
P.O. Box 1165
SE-581 11 LINKÖPING

Phone: +46 13 37 80 00
Fax: +46 13 37 81 00

www.foi.se

OLSR Broadcast Security in Mobile Ad hoc Networks

JIMMI GRÖNKVIST, ANDERS HANSSON, MATTIAS SKÖLD



FOI-R--2323--SE
ISSN 1650-1942

Scientific Report
September 2007

Command and Control Systems

OLSR Broadcast Security in Mobile Ad hoc Networks

FOI-R--2323--SE
ISSN 1650-1942

Scientific Report
September 2007

Command and Control Systems

Issuing organization
Swedish Defence Research Agency
Command and Control Systems
P.O. Box 1165
SE-581 11 LINKÖPING
SWEDEN

Report number, ISRN
FOI-R--2323--SE

Report type
Scientific Report

Research area code
7. C^{4}I and Human Factors

Month year
September 2007

Project No.
E7108

Sub area code
71. Command, Control, Communications, Computers, Intelligence (C^{4}I)

Sub area code 2

Author/s
Jimmi Grönkvist, Anders Hansson, Mattias Sköld

Project manager
Mattias Sköld

Approved by
Sören Eriksson

Sponsoring agency
Swedish Armed Forces

Scientifically and technically responsible
Jan Nilsson

Report title
OLSR Broadcast Security in Mobile Ad hoc Networks

Abstract

A mobile ad hoc network consists of wireless nodes that build a robust radio network without any pre-existing infrastructure or centralized servers. The risk of malicious attacks makes security an important but difficult issue in this type of network.

In military tactical scenarios, multicast and broadcast traffic is generally considered very important. However, designing ad hoc networks for multicast traffic is more difficult than the corresponding design for unicast traffic.

Advanced signatures is one method that can be used to improve security for the Optimised Link State Routing (OLSR) protocol. In this report we show that the overhead cost of this solution is very high and further work will be needed to tune the method. Local information is simpler to secure, which can be used for MPR flooding.

We have also studied the efficieny of broadcasting with different amounts of information. Our results suggest that Multi-Point Relay (MPR) flooding seems to be a good compromise between efficiency, security and robustness in a mobile ad hoc network.

Keywords
Ad hoc networks, OLSR, secure routing, broadcast routing

Further bibliographic information

Language English

ISSN 1650-1942

Pages 44 p.

Price acc. to pricelist

Utgivare

Totalförsvarets Forskningsinstitut
Ledningssystem
Box 1165
SE-581 11 LINKÖPING

Rapportnummer, ISRN
FOI-R--2323--SE

Klassificering
Vetenskaplig rapport

Forskningsområde
7. Ledning med MSI

Månad, år
September 2007

Projektnummer
E7108

Delområde
71. Ledning

Delområde 2

Författare
Jimmi Grönkvist, Anders Hansson, Mattias Sköld

Projektledare
Mattias Sköld

Godkänd av
Sören Eriksson

Uppdragsgivare/kundbeteckning
Försvarsmakten

Teknisk och/eller vetenskapligt ansvarig
Jan Nilsson

Rapportens titel

Säkerhetsanalys av broadcast med OLSR i mobila ad hoc-nät

Sammanfattning

Ett mobilt trådlöst ad hoc-nät består av ett antal noder, som bildar ett robust radionät utan fast infrastruktur och centraliserade funktioner. I dessa sammanhang är sårbarhet för attacker och säkerhetsfrågor viktiga problem att lösa.

I militära scenarier brukar multicast-trafik och broadcast-trafik anses viktiga. Ad hoc-nät som kan hantera sådan trafik är dock mycket svårare att realisera än motsvarande nät som bara hanterar unicast-trafik.

”Advanced signatures” är en metod att höja säkerheten i OLSR. I denna rapport visar vi att andelen administrativ trafik blir mycket hög och detta kräver vidare utveckling. Lokalt informationsutbyte är enklare att säkra med signaturer, vilket kan utnyttjas för MPR-flooding.

Vi har också jämfört olika broadcast-tekniker som utnyttjar varierande mängd information och studerat deras effektivitet. Resultaten visar att MPR-flooding kan ge en god balans mellan kapacitet, säkerhet och robusthet i mobila ad hoc-nät.

Nyckelord
ad hoc-nät, OLSR, säker routing, broadcast-routing

Övriga bibliografiska uppgifter

Språk Engelska

ISSN 1650-1942

Antal sidor: 44 s.

Distribution enligt missiv

Pris: Enligt prislista

Contents

1 Introduction 9

1.1 Background . . . 9

1.1.1 Main Focus . . . 9

1.2 Broadcast Traffic in ad hoc networks . . . 10

1.3 Security in ad hoc networks . . . 11

1.3.1 Overview of the Report . . . 12

2 Optimal Link State Routing 13

2.1 Overview of OLSR . . . 13

2.1.1 MPR selection . . . 14

2.1.2 Packet Formats, Forwarding, and Processing Messages . 15

2.1.3 OLSR Control Traffic . . . 15

2.2 Weaknesses . . . 16

2.2.1 Incorrect traffic generation . . . 16

2.2.2 Incorrect traffic relaying . . . 17

3 Overhead Aspects of Securing OLSR 19

3.1 Advanced Signatures . . . 19

3.2 Comments and Limitations of the Solution . . . 20

3.3 Overhead Evaluation . . . 21

3.3.1 HELLO message overhead . . . 23

3.3.2 TC message overhead . . . 26

3.4 Signature Overhead Results . . . 27

4 Broadcast Evaluation **33**
4.1 Tree Generation 34
4.2 Simulation Results 34
4.3 Concluding remarks 40

5 Conclusions **41**
5.1 Future Work 42

Chapter 1

Introduction

1.1 Background

An ad hoc network is a collection of wireless mobile nodes that dynamically form a temporary network without the need for any pre-existing network infrastructure or centralized administration. Due to the limited transmission range of radio interfaces, multiple "hops" may be needed for one node to exchange data across the network with another node. An ad hoc network is both self-forming and self-healing and can thus be deployed with minimal or no network pre-planning. However, one drawback of this is that the network will not always be connected. A tactical network may be partitioned or fragmented into parts, e.g. due to movements or terrain obstacles. It is therfore necessary that parts of the network can function autonomously, which requires a distributed network control.

1.1.1 Main Focus

We study the consequences for communication when adding security components to broadcast routing protocols in tactical ad hoc networks. Adding security mechanisms is necessary to reach a sufficient security level, but will decrease the capacity available for the user during normal operation. There are many such extensions to existing protocols, see for example [1, 2], but little work has been done on evaluating this negative consequence [3].

The Optimised Link State Protocol (OLSR) [4] will be the protocol on which we base our investigations, as this protocol have many properties that make it useful for our military scenarios and for broadcast traffic. Furthermore, some previous work on securing this protocol also exists, which is also an additional advantage. In this report we evaluate the overhead that the OLSR traffic amount to, both for standard OLSR and for a secure extension of the protocol. Another problem we address, is how efficiently different broadcast methods reach all nodes in the network.

1.2 Broadcast Traffic in ad hoc networks

In military tactical scenarios, multicast (one-to-many) and broadcast (one-to-all) traffic is generally considered important, and to a much higher degree than what most civilian applications and services would require. However, designing ad hoc networks for broadcast traffic is more difficult than the corresponding design for unicast traffic. For this reason most ad hoc network research have been conducted for unicast traffic, and most existing solutions for broadcast (and multicast) routing are additions to the unicast algorithms. This may have more or less an impact on how well they perform as compared to algorithms originally designed for broadcast traffic.

In some cases the same solution can be used for broadcast as for unicast without any loss in efficiency. However, in other cases, the solutions may be considerably different. One example is broadcast with variable data rates, which is much more difficult problem than the corresponding unicast problem.

Most solutions to the broadcast routing problem are of one of two types. In the first, all (or most) information describing the network topology is assumed to be known by all nodes and an appropriate transmission tree is generated which can be used for sending the broadcast traffic. In such a tree, the root is the source node, and all nodes that are not leafs will retransmit the traffic.

The other solution is to use some form of flooding (possibly limited so not all nodes need to resend the traffic). The first solution is more efficient, but the cost for upholding the tree during mobility is high. Flooding techniques are less efficient for the data traffic but usually need less overhead traffic (if any).

1.3 Security in ad hoc networks

The specific properties of ad hoc networks makes them a good choice for tactical military scenarios, which can be unpredictable and where the loss of any node is possible. However, the basic properties of ad hoc networks also makes them difficult to secure. Unlike traditional networks, there are no central points that can be used to control access to the network and its resources. Furthermore, all nodes are mobile and are sensitive to hijacking (at least to some degree) and the use of radio means that a hostile node can attempt to access any of the network nodes, which makes it difficult to separate the network into secured and unsecured parts. For example, there is no single place for a firewall or an intrusion detection system which can protect the network.

The traditional method of protecting radio networks has been the use of cryptographic mechanisms, such as encryption and message authentication. However, such mechanisms can only protect against attacks by external nodes, not compromised nodes that are already part of the network, and therefore already have many of the keys [5].

With the development of Software-Defined Radio and Network Centric Warfare, many security issues known from the Internet may be a reality also in military networks. It is difficult to design and implement software systems without introducing design and programming errors that an adversary can exploit. If an adversary has adequate resources and tries hard enough, there is always a risk that the adversary succeeds in infiltrating the system.

History has taught us that no matter how many security mechanisms (e.g. encryption, authentication and firewalls) that are inserted in the network, there are always weak points that adversaries can exploit.

Hence, to obtain an acceptable level of security in military contexts, traditional security solutions should be coupled with intrusion detection systems (IDS) that continuously monitor the network and determine whether the system (the network or any node of the network) is under attack. Once an intrusion is detected, e.g. in the early stage of a denial of service attack, a response can be put into place to minimize the damage.

In the past few years much has been published on security in ad hoc networks. One problem, however, is that most of the research into security has been forced to deal with completed algorithms that were not originally designed for security. The solution therefore, is often patches to existing algorithms, rather than something included in the original design.

Assumptions

In this report we will make the following assumptions: First, the network is protected against direct attacks by lower layer encryption and mechanisms such as frequency hopping. This will protect the system against most simple attacks and jamming, and we will not study these further. However, the nodes themselves are suseptible to hijacking, possibly with passwords and keys intact. From such a node, a well informed enemy can get information out of the system and can introduce for example malicious code, for example, that can be used to attack different protocols inside the system.

In order to protect against these kinds of attacks on the rest of the network, we assume that all nodes have had one or more signed certificates from a trusted source installed in each node of the network. These certificates can then be used to set up sessions and common keys. The certificates must then be manually installed prior to network setup. Such information can subsequently be used to prove the origin of messages and ensure that only specified nodes can read information.

Furthermore, in our simulations we will also assume that the ad hoc network is always connected. This assumption will affect the values shown in the latter chapters, but it is not a limitation of the methods (routing protocol and security additions) we are investigating.

1.3.1 Overview of the Report

In Chapter 2 we give an overview of OLSR and describe some of the weaknesses of this protocol. In Chapter 3 we describe a method that can be used to secure OLSR (advanced signatures) and continue by evaluating what consequences this will have for the overhead of OLSR. We also study the overhead of the basic OLSR algorithm. We continue in Chapter 4 by studying the efficency of different methods of broadcast routing and discuss the security aspects of using different amounts of information. Chapter 5 concludes the report.

Chapter 2

Optimal Link State Routing

Optimised Link State Routing, OLSR [4], is one of the few ad hoc routing protocols that have reached the status of RFC, and besides the Ad-hoc On-demand Distance Vector protocol (AODV) [6], it is the routing protocol that has aroused the most interest from the research community. Unlike AODV, which only finds routes when they are needed, OLSR is a proactive protocol. This means that the routing protocol will attempt to build routes between all nodes regardless of whether they are needed or not. The advantage of this is that a path already exists when it is needed and no route search needs to be made before user traffic can be sent. The disadvantage is more overhead traffic in some scenarios. OLSR is most beneficial if many nodes often want to communicate in the network so that information about most paths will be needed (thereby giving little unnecessary overhead). This is often the case in military networks because multicast to all other nodes is a common type of traffic.

In this chapter we will first give a short description of OLSR (for more details we refer to RFC 3626 [4]), and then we will discuss some of its weaknesses.

2.1 Overview of OLSR

OLSR is based on the classic link state routing protocol but with some changes that make it more useful for mobile ad hoc networks with low link capacity. In LSR, each node sends information about all links to the entire network, thereby making it possible for each node to calculate the route with the least cost for each destination. However, this protocol generates very high overhead traffic, which makes it impractical for mobile networks, in which changes in topology is common.

In OLSR the overhead information is decreased by letting each node choose a subset of neighbors called multipoint relays (MPR) which are the only nodes that will retransmit a message. These MPRs are chosen so that all two-hop neighbors will be reached if all MPRs retransmit the control messages. This reduces the number of required retransmission of the link state messages (especially for a dense network). In addition, all links to these neighbors need to be symmetric, i.e. communication in both directions must be possible.

The second method of decreasing routing overhead compared with LSR is that OLSR only sends partial link state information rather than sending information about all links. The minimum information is that all nodes chosen to be MPR nodes send information about the links to those nodes that selected them as MPRs, although more information than this may be sent for greater robustness.

As the control messages used are sent periodically, a reasonable message loss can be accepted by the protocol without significant degradation.

2.1.1 MPR selection

Each node must select a subset of its one hop neighbors that will be MPRs. However, in the OLSR specification only a suggested heuristic algorithm is proposed, and there is no hard requirement that it must be used. In short, the algorithm starts by looking for two-hop neighbors that can only be reached through a single neighbor, then setting these neighbors to be MPRs.

One important feature is the “willingness” of a node to become a MPR. This is set by each node and goes from “will never” to “will always”. Nodes advertising “will always” must be chosen as MPRs. In the next step the algorithm looks at those neighbors with the highest “willingness” and assigns those in the order of the highest number of two-hop neighbors they can reach. If all two-hop neighbors are not reached when this group has been added, groups with lower “willingness” will be added until all can be reached.

2.1.2 Packet Formats, Forwarding, and Processing Messages

OLSR communicates using a unified packet format for all data related to the protocol.

On receiving a basic packet, a node examines each of the message headers. Based on the value of the Message Type field, the node can determine how to handle each message. A node can receive the same message several times. Thus, to avoid re-processing of messages which were already received and processed, each node maintains a record about the most recently received messages, called the Duplicate Set. This is used to avoid duplicate processing of a message.

2.1.3 OLSR Control Traffic

Control traffic in OLSR is mainly exchanged through two different types of messages: HELLO and TC (Topology Control) messages.

HELLO messages are exchanged periodically among neighbor nodes, in order to detect links to neighbors, to detect the identity of neighbors and to signal MPR selection. On receiving a HELLO message, a node examines the lists of addresses. If its own address is included, it receives confirmation that bi-directional communication is possible between the originator and the recipient of the HELLO message. When a link is confirmed as bi-directional, this is advertised periodically by a node with a corresponding link status of "symmetric". In addition to giving information about neighbor nodes, periodic exchange of HELLO messages also allows each node to maintain information describing the links between neighbor nodes and nodes two hops away. This information is recorded in a node's 2-hop neighbor set and is explicitly utilized for the MPR optimization.

TC messages are periodically flooded to the entire network in order to spread link state (topological) information to all nodes. A TC message contains a set of bi-directional links between a node and a subset of its neighbors. The topological information is used in the MPR optimization. Only nodes that have been selected as an MPR generate (and relay) TC messages. The TC message contains a field with the Advertised Neighbor Sequence Number (ANSN). This number is associated with the node's advertised neighbor set and is incremented each time the node detects a change in this set.

There are two more types of control messages in OLSR: Multiple Interface

Declaration (MID) and Host and Network Association (HNA). MID messages are only generated by nodes with multiple OLSR interfaces in order to announce information about its interface configuration to the network. HNA messages are only generated by nodes with multiple non-OLSR interfaces and have the purpose of providing connectivity from an OLSR network to a non-OLSR network.

2.2 Weaknesses

In this section, we will very shortly describe some possible attacks on OLSR. For a more detailed description and more attacks, see [7], [8] and [3].

In order for OLSR to update a node's routing table it has two different responsibilities. Firstly, each node must correctly generate routing protocol control traffic according to the protocol specification. Secondly, each node must forward control traffic generated in other nodes in the network. Hence, incorrect behavior of a node can result from a node generating incorrect control messages and/or from incorrect relaying of control traffic from other nodes.

2.2.1 Incorrect traffic generation

A node can misbehave by generating false HELLO, TC or MID/HNA messages. This can be done in two different ways: by generating control traffic, pretending to be another node or by transmitting incorrect information in control messages.

To exemplify incorrect traffic generation, we look at HELLO messages. A misbehaving node, E, may send HELLO messages pretending to be another node, C (see Figure 2.1). This will result in nodes A and B announcing that C is a one-hop neighbor in their HELLO and TC messages. Conflicting routes to node C with possible loops or connectivity loss may result from this.

A misbehaving node may also send HELLO messages containing incorrect information about its set of neighbors. This can be done in two ways: sending out an incomplete set of neighbors or stating non-neighbors are neighbors. In the first case the network may be without connectivity to the ignored neighbors. In the second case nodes may select an incorrect set of neighbors as MPRs, with the result that some nodes may not be reachable in the network.

2.2.2 Incorrect traffic relaying

If a node does not properly relay control messages network malfunctions are possible. For example, if a node does not relay TC messages, the network may experience connectivity problems. In networks where no redundant path exists, connectivity loss will be the result, but other topologies may provide redundant connectivity and routes can still be found.

If MID and HNA messages are not properly relayed, information about multiple nodes interfaces and connection to other networks may be lost.

Another attack consists of replaying old control messages. This causes nodes to record out-of-date topology information. However, a control message cannot be replayed as it is because nodes that have already received it will ignore the replayed message because of the Message Sequence Number (MSN) (and ANSN for TC messages). The attacker needs to increase MSN (ANSN for TC) for the messages to be accepted. This may cause connectivity problems and possible loss of data messages and that correct routing packets wont be accepted due to already used values of MSN and ANSN.

Furthermore, a misbehaving node may also choose not to forward data packets. Hence, data packets transmitted along routes containing the misbehaving node will not reach their destination.

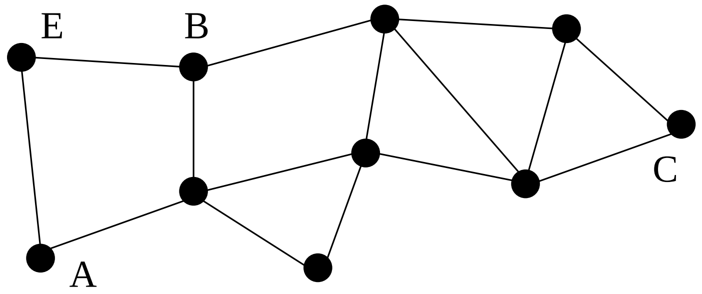


Figure 2.1: Node E sends HELLO messages pretending to be node C.

Chapter 3

Overhead Aspects of Securing OLSR

In this chapter we evaluate the overhead that the OLSR traffic amount to, both for standard OLSR and for a secure extension of the protocol. Much work has been done on how to secure routing algorithms for mobile ad hoc networks, see e.g. [1, 2]. However, once trusted nodes becomes compromised the problem is significantly more difficult. Some suggestions for overcoming this problem aim at identifing and blacklisting such nodes, for example [9], but their efficency is limited, especially since they often generate false alarms.

3.1 Advanced Signatures

In [10] a promising technique is described that introduces advanced signature techniques for the OLSR protocol. The protocol relies on creating and sending additional OLSR messages in conjunction with the regular routing messages. These messages contain additional signatures from several nodes and are used to prove that the information sent in the HELLO and TC message is correct. This solution does not require any change in the original OLSR protocol as additional message types can be added to the protocol.

However, all nodes need the public key of all other nodes, and time synchronization between all nodes is needed.

The main idea is that each node stores information about itself that it has received from other nodes (in their HELLO messages) and uses this information as proof by including it in the control messages that it sends out (both HELLO and TC messages).

The functionality of this protocol is based on the HELLO messages. In short, the purpose of a HELLO message is to give nodes the 2-hop information needed for selecting the MPRs. To do this, each node sends out information about which neighbors it has. From this nodes can calculate the 2-hop neighbor set. A malicious node can interfere with this process in several ways: it can send false HELLO messages claiming to be another node or it can add or remove neighbors from a HELLO message.

Usually, a malicious node would try to add more neighbors as it will then more likely be selected as an MPR. To prevent this each node does three things. Firstly, all HELLO messages are signed so that they cannot be faked. Secondly, each neighbor which from a node receives a HELLO message (correctly signed) is included in the HELLO message (as usual) and are seperately signed. Thirdly, for all links, signatures from the neighbors are included (second part of the neighbors HELLO messages), which then prove that the neighbor can hear the node and consider it a symmetric neighbor.

For a TC message only Steps 1 and 3 is necessary.

3.2 Comments and Limitations of the Solution

The above solution gives the nodes the ability to prove that they have the neighbors they claim in their HELLO and TC messages. However, a malicious node can still do several things. For example, this method does not prevent a node from sending too many TC messages that will subsequently be flooded through the network. Furthermore, several cooperating nodes can also create HELLO and TC messages with correctly signed links that do not exist which makes wormholes possible. Unlike reactive protocols where such set-ups are difficult, here they are simple because the exchange can be done at an earlier time.

However, we will defend against most of the attacks against OLSR, although a specification-based IDS for both TC and HELLO messages should probably be added to further remove threats. We however consider this future work.

A problem with this solution is the large additions of overhead due to the added signatures. The signature size needs to be much larger than the address size which at the moment is 32 bits (for IPv4). In the next section we will study the added requirements for the overhead traffic that this method incurs.

3.3 Overhead Evaluation

In this section we provide an overhead evaluation of HELLO and TC message traffic for OLSR with and without signatures.

We want to investigate whether signatures dramatically increase the amount of overhead traffic in OLSR. Most of the overhead traffic is generated by transmitted HELLO and TC messages in OLSR. According to the default message rates, given in [4], the HELLO messages are exchanged periodically every 2 seconds and the TC messages every 5 seconds. In the secure case, described in [10], the OLSR standard is extended with signature messages, a new message type that is sent with each TC or HELLO message.

One or more OLSR messages can be sent in one OLSR packet (see Figure 3.1). Consider an OLSR packet containing OLSR messages of size $M_i, i = 1, \cdots$. Each OLSR packet is sent over IP and UDP. Assuming IPv4, we calculate the total packet size as

$$256 + \sum_i (96 + M_i) \text{ bits}, \tag{3.1}$$

where each OLSR message in the packet follows a message header of 96 bits, and all packet headers sum up to 256 bits:

IP header size:	160 bits
UDP header size:	64 bits
OLSR packet header size:	32 bits
Total packet header size:	256 bits.

As a worst case, we assume that only one OLSR message (together with its signature message) is sent in each OLSR packet. The size of the OLSR messages depends on the neighbors of the node sending the message. We derive the message sizes in more detail in the following subsections.

```
    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
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |         Packet Length         |    Packet Sequence Number     |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |  Message Type |     Vtime     |         Message Size          |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                      Originator Address                       |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |  Time To Live |   Hop Count   |    Message Sequence Number    |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                                                               |
   :                            MESSAGE                            :
   |                                                               |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |  Message Type |     Vtime     |         Message Size          |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                      Originator Address                       |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |  Time To Live |   Hop Count   |    Message Sequence Number    |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                                                               |
   :                            MESSAGE                            :
   |                                                               |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   :                                                               :
```

Figure 3.1: The basic layout of an OLSR packet (omitting IP and UDP headers) [4].

3.3.1 HELLO message overhead

Each node in the network periodically transmits a HELLO message to its neighbors. HELLO messages are not forwarded to other nodes. As we can see in Figure 3.2, the HELLO message consist of a 32-bit header followed by a number of link messages. Each link message, in turn, starts with a 32-bit link message header specifying the link and neighbor type of the succeeding neighbor interface addresses. The possible link and neighbor types are specified in section 6.1.1 in [4]:

- Link types
  - UNSPEC_LINK – indicating that no specific information about the links is given.
  - ASYM_LINK – indicating that the links are asymmetric (i.e., the neighbor interface is "heard").
  - SYM_LINK – indicating that the links are symmetric with the interface.
  - LOST_LINK – indicating that the links have been lost.
- Neighbor types
  - SYM_NEIGH – indicating that the neighbors have at least one symmetrical link with this node.
  - MPR_NEIGH – indicating that the neighbors have at least one symmetrical link AND have been selected as an MPR by the sender.
  - NOT_NEIGH – indicating that the nodes are either no longer or have not yet become symmetric neighbors.

To simplify the calculations, we consider only stationary networks with symmetric links. We can also view this as a number of stationary snapshots of mobile networks. In those snapshots, all mobility changes have been communicated (link sensing, neighbor detection and MPR selection). This means that no combinations of links of type: UNSPEC_LINK, ASYM_LINK or LOST_LINK, and nodes of type NOT_NEIGH will occur in the transmitted HELLO messages. Thus, each HELLO message can contain only two possible link messages: a list

```
    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 3.2: HELLO message format [4].

of selected MPR neighbors and/or a list of non-MPR neighbors. A node that has a total of n neighbors transmits HELLO messages of size

$$M_{\text{HELLO}} = \begin{cases} 32(n+2) & \text{bits (one enclosed link message)} \\ 32(n+3) & \text{bits (two enclosed link messages).} \end{cases} \tag{3.2}$$

As a motivation for this simplification we can argue that if the refreshing period for the HELLO transmissions is well adjusted (with some margin) to the mobility of the network, these two kinds of HELLO messages will be the most common form of the transmitted HELLO messages in the mobile network.

In the secure case, each HELLO message is accompanied by a signature message (see Figure 3.3) in the same OLSR packet. Assuming 32-bit timestamps and 128-bit signatures and proofs for the authentication mechanism, we get a signature message header size of $32+32+128 = 192$ bits. Each advertised neighbor in the HELLO message also generates a signature, a timestamp and a proof ($128 + 32 + 128 = 288$ bits) in the signature message. The total size of a signature message accompanying a HELLO message advertising n neighbors is:

$$M_{\text{HELLOsig}} = 192 + 288n \text{ bits.} \tag{3.3}$$

```
 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Sign. Method  |   Reserved    |         MSN Referrer          |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
:                      Global Timestamp                         :
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
:                      Global Signature                         :
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
:           Signature of Certificate #1 (HELLOs only)           :
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
:           Signature of Certificate #2 (HELLOs only)           :
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
:                           . . .                               :
:                                                               :
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
:                    Timestamp of Proof #1                      :
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
:                    Signature of Proof #1                      :
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
:                    Timestamp of Proof #2                      :
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
:                    Signature of Proof #2                      :
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
:                           . . .                               :
```

Figure 3.3: Signature message format [10].

We get the total packet size for the HELLO message from (3.1) and (3.2) without authentication as:

$$256 + (96 + M_{\text{HELLO}}) = \begin{cases} 416 + 32n & \text{bits (one enclosed link message)} \\ 448 + 32n & \text{bits (two enclosed link messages)} \end{cases} \tag{3.4}$$

With a signature message (3.3) the total packet size is:

$$256 + (96 + M_{\text{HELLO}}) + (96 + M_{\text{HELLOsig}}) = \begin{cases} 608 + 320n & \text{bits (one enclosed link message)} \\ 640 + 320n & \text{bits (two enclosed link messages).} \end{cases} \tag{3.5}$$

To calculate the HELLO message overhead for a network, we count the number of neighbors and the number of selected MPR neighbors to each node. The number of selected MPR neighbors decides whether the HELLO message contains one enclosed link message (all or none of the neighbors of a node are selected as an MPR) or two enclosed link messages.

3.3.2 TC message overhead

Each node that is selected as an MPR by any other node repeatedly broadcasts TC messages to all other nodes in the network with the help of the MPR forwarding mechanism. According to the TC message format (see Figure 3.4), the TC message consists of a 32-bit header followed by a list of 32-bit addresses to neighbors that have selected the sender node as their MPR (its MPR selector set). So if the size of the MPR selector set to a node is s, the size of the transmitted TC message is

$$M_{\text{TC}} = 32 + 32s \text{ bits} \tag{3.6}$$

For the secure case, the accompanying signature message (see Figure 3.3) contains a timestamp and a proof for each node in the MPR selector set. The size of the signature message is

$$M_{\text{TCsig}} = 192 + 160s \text{ bits.} \tag{3.7}$$

```
 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 3.4: TC Message Format [4].

We get the total packet size for the TC message without authentication from (3.1) and (3.6) as:

$$256 + (96 + M_{\text{TC}}) = 384 + 32s \text{ bits} \tag{3.8}$$

and together with a signature message (3.3) the total packet size is:

$$256 + (96 + M_{\text{TC}}) + (96 + M_{\text{TCsig}}) = 672 + 192s \text{ bits} \tag{3.9}$$

To calculate the TC traffic overhead for a network, we count the size of the MPR selector set and the required number of MPR forwarding retransmissions for each node.

3.4 Signature Overhead Results

To get an idea of how much overhead that HELLO, TC and signature messages amount to, we use simulated networks: stationary networks with random node placement in a terrain area. We say that two nodes are connected with a communication link if the estimated basic path-loss between the nodes is less than a threshold value. The basic path-loss calculations are carried out using the wave propagation library, DetVag-90® [11], with a Uniform geometrical Theory of Diffraction (UTD) model by Holm [12]. We have generated networks of size 10, 20, 40 and 60 nodes. For each network size, we generated 1000 networks with varying basic path-loss thresholds to derive networks of different connectivities.

We show the generated traffic per node from our calculations in Figures 3.5, 3.6, 3.7, and 3.8. To the left in each figure we show the overhead of normal

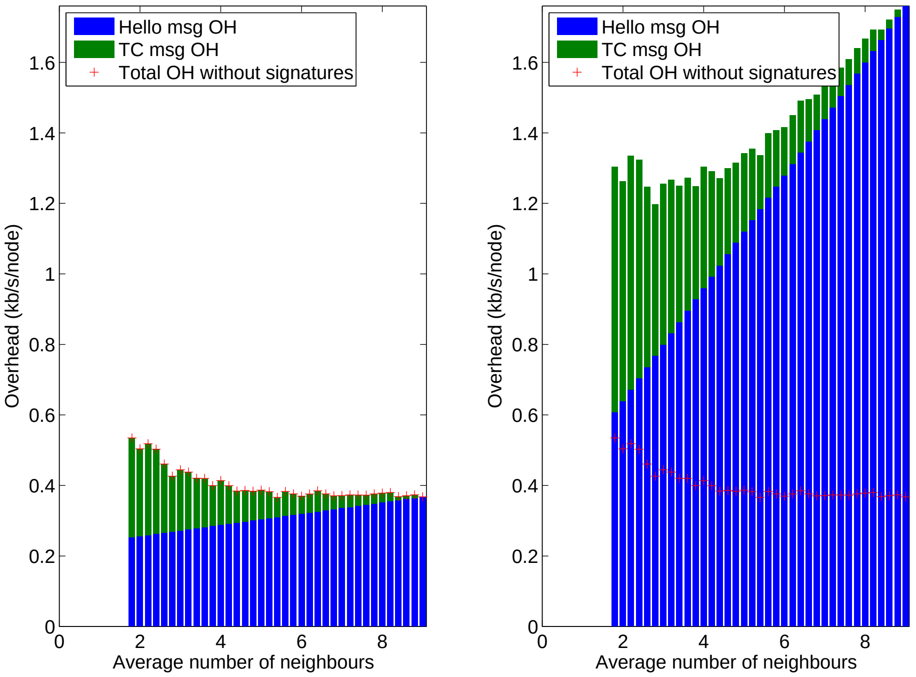


Figure 3.5: Comparison of OLSR routing overhead for different networks of size 10. Overheads without signatures are displayed to the left and overheads with signatures are displayed to the right.

OLSR and to the right, the overhead resulting from adding signatures. It is easy to see that the signature message traffic increases the overhead significantly.

As expected from (3.3) and (3.5), the HELLO message overhead increases almost linearly with the average number of neighbors in the network. For sparse networks, the TC message traffic makes up roughly half of the routing control traffic. As the average number of neighbors increases, the HELLO message overhead becomes dominant. In a fully connected network, where all nodes are neighbors, there is no need for MPR nodes and thus no TC traffic.

Before we begin to discuss the effect of signatures, we can study the overhead of basic OLSR. As can be seen, the overhead per node increases with network size. This will result in an overhead increase on network level that is faster than linear with network size, which means that the protocol does not scale that

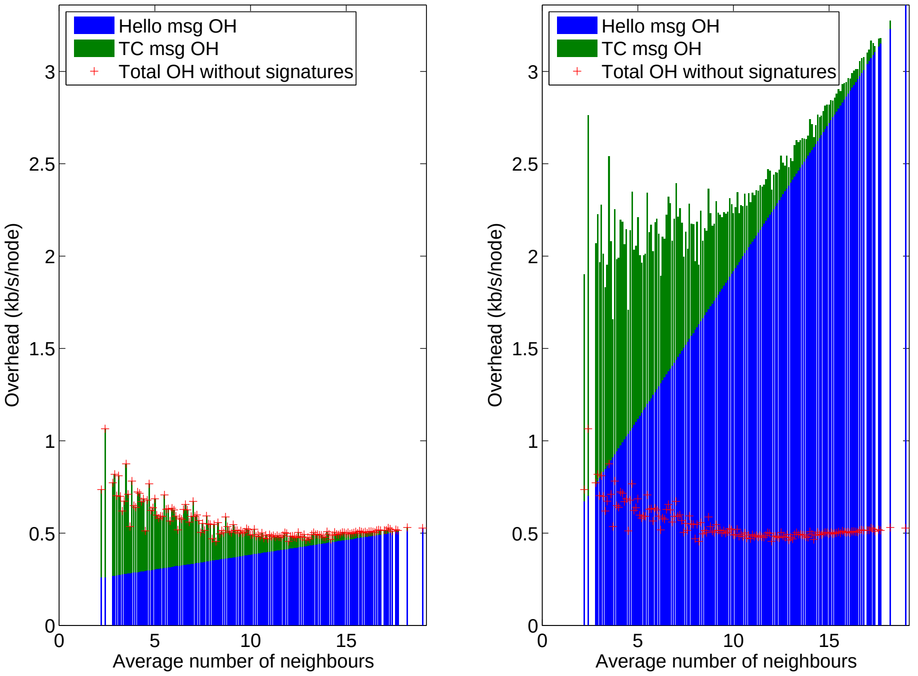


Figure 3.6: Comparison of OLSR routing overhead for different networks of size 20. Overheads without signatures are displayed to the left and overheads with signatures are displayed to the right.

well. This, however, is a property of all ad hoc protocols with non-localized traffic.

For the 60-node network, OLSR will need at least 60 kb/s for routing overhead in the entire network (1kb/node/s), even more if the network is highly or lowly connected, which will be a noticeable part of the network capacity, and even more so for larger networks. Luckily, much of the transmitted information can be compressed, e.g. full IP address length, header information and similar things that could allow for lower overhead cost. Potentially, the HELLO messages could also be generated at layer 2 instead and completely bypass the IP layer with its additional overhead. Such compression will probably not be necessary, though, unless network sizes become larger than those we study in this report.

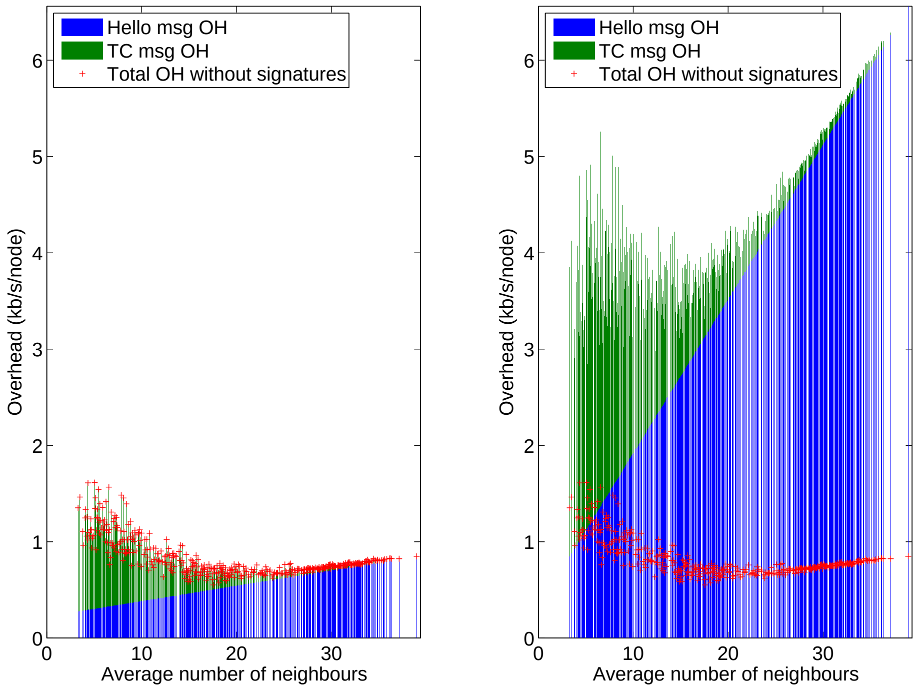


Figure 3.7: Comparison of OLSR routing overhead for different networks of size 40. Overheads without signatures are displayed to the left and overheads with signatures are displayed to the right.

If we add signatures to each message, the overhead cost will be significantly larger however. For the 60-node network we will generate a minimum of 350 kb/s, and unless we have a very high capacity network, this will be a very noticeable part of the network capacity.

To put these values into perspective, for mobile tactical ad hoc networks foreseeable values of network capacity is in the order of 1 Mb/s, see for example [13].

In addition, the cost of encrypting and decrypting so many messages will also be very expensive from a computational point of view. Furthermore, compression will not yield a significant reduction

in order to reduce as the largest part of the overhead is now the signatures:

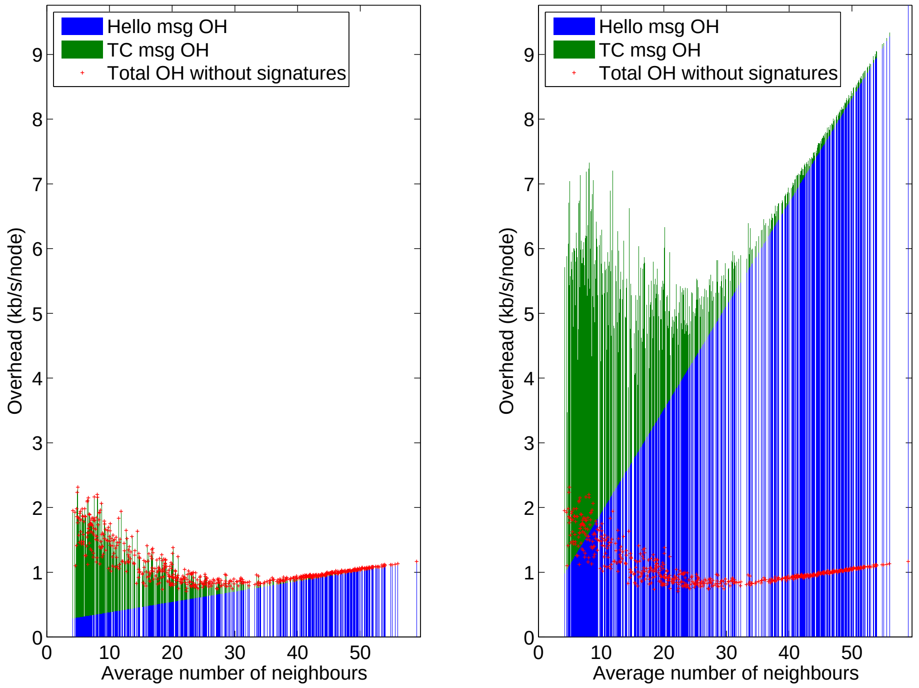


Figure 3.8: Comparison of OLSR routing overhead for different networks of size 60. Overheads without signatures are displayed to the left and overheads with signatures are displayed to the right.

header compression will only have a very small impact.

This means that in the present form the advanced signatures can probably not be used to secure OLSR in bandwidth limited tactical ad hoc networks, except for very small networks. But we can potentially use the method as part of a complete solution. If we specifically look at HELLO messages, we see that two types of attacks are especially efficient: adding non-existent links and pretending to be someone else when generating HELLO messages. The latter can be handled with with only one signature (instead of one per link), which is much cheaper, and the former can be dealt with by not allowing new links unless they are properly authenticated with advanced signatures at least once. After the first occurrence, normal HELLO messages, without all the signatures, can be used.

This of course, is not sufficient. If a new node enters an area, it will not have received that first HELLO message. Furthermore, a malicious node would have the potential to retain a link forever if it was not periodically forced to reprove that it still was valid. Nevertheless, if signatures are not needed on every link update but only on new ones and regular updates are done more seldom than normal HELLO and TC messages, we would be able to significantly reduce the overhead of advanced signatures. A drawback with this, of course, is that a malicious node could claim the existence of links longer than otherwise possible, but this is less of a problem.

How much overhead such a scheme would result in will be examinated in further work, as it will be dependent on mobility, due to the need to send signatures every time a new link is created.

TC messages are more difficult to secure as they are sent more seldom to start with. Consequently, signing only a fraction of them implies that it may be a long time before a link is authenticated. Furthermore, it is not possible for a node to know when a new node has entered the network so we cannot have a reactive approach as for the HELLO messages. In many cases this might not be a significant problem, because unicast routing packets are normally routed hop-by-hop, which means that only a general “direction” is necessary. For multicast we ideally want spanning trees though, which need to be predetermined at the source to be efficient (fixes later on would need information about which nodes each split packet should reach. If links have failed since the last updates, such packets might need to be sent back on links that have already sent the packet earlier). The risk of long updating times, especially as malicious nodes can retain links, means that such trees will be more difficult to secure without excessive overhead.

Chapter 4

Broadcast Evaluation

In this chapter we will study how efficiently different broadcast methods reach all nodes in the network. In general, the more efficient a scheme is the less retransmissions (fewer channel resources) it uses. As was seen in the last chapter, it is easier to secure the information generated close by (HELLO) than information generated further away (TC). It is therefore interesting to study how much we might gain by using information from further away compared with only local information to give us a good tradeoff between security and communication efficiency. Furthermore, use of information from further away is also more sensitive to mobility, but that will not be the primary concern in this chapter. The methods we will study are flooding, MPR flooding, TC tree generation, and full graph trees.

- *Flooding* is the simplest form of broadcast routing as it requires no information at all about the network. Every time a packet is received it will be retransmitted (unless this has already been done). This means that all nodes in the network will retransmit each packet. Although very inefficient (especially for dense networks), it is nevertheless very robust against all forms of manipulation and mobility because packets will be received by all nodes that have at least one path to the originator without a malicious node in it.
- *MPR flooding* is essentially what is used to disseminate the TC messages. In this case the messages are retransmitted by the MPRs only. This is less robust than flooding, but if there are several paths through MPRs, a

message will still reach the destination even if one MPR fails to retransmit a message. Moreover, it is only based on HELLO messages, which are more secure than TC messages.

- *TC trees* are network spanning trees based on those links that are advertised in the TC messages. Note that only links between nodes and their chosen MPRs are advertised, which can be considerably less than the full set of nodes. This solution is less robust than those above and is based on correct TC messages, which are less secure than HELLO messages, especially as malicious nodes can retain links even after they have failed. However, if we already use OLSR for unicast traffic, it does have the advantage of being efficient while not adding any extra overhead.
- *Full graph trees* are network spanning trees based on all existing links in the network. Properly chosen, broadcast based on all link information in the network can produce the highest capacity of all solutions described. However, in order to use such trees we need full information about every link in the network, which is both expensive from an overhead point of view and difficult to secure.

4.1 Tree Generation

For radio broadcast traffic, we want to minimize the number of retransmissions needed for a packet to reach all nodes in the network. This problem can also be described as finding a minimum connected dominating set in the network [14], which is known to be NP-complete [15]. As this is very time-consuming for large networks, we use the following heuristic algorithm both for TC trees and full graph trees. Initiate by choosing the source node as root. Among the included nodes, find the node v with the highest number of neighboring nodes that is not yet included. Include all these neighboring nodes and the edges from v to these nodes. This is repeated until all nodes are included in the tree.

4.2 Simulation Results

The networks used in this chapter have been generated in the same way as described in the previous chapter.

In Figures 4.1 and 4.2 we show the average number of retransmissions needed for the four schemes for network sizes 10, 20, 40 and 60.

MPR flooding

As can be seen, flooding is very inefficient. Even for very lowly connected networks, it costs about twice the number of retransmissions the other schemes need. Adding HELLO-message information and using only the MPRs for retransmission, we see a large improvement in terms of retransmissions. For highly connected networks we are very close (or even equal) to what is possible with a full graph tree, although this was expected as most or all of the network nodes will be within two hops of each other in such cases.

For lower connectivity, the difference is greater (at least for larger network sizes) but MPR flooding still achieves very good results with only two-hop information.

The fact that the information used for MPR flooding is limited to only two hops leads in some cases to significantly reduced efficiency. This is especially the case if the network contains 5-hop loops (or more) as shown in Figure 4.3. Assuming that node 1 is the source, a possible sending tree is nodes 1, 2, and 5, as this will reach all nodes. However, using MPRs there is no knowledge about 3-hop information, e.g. node 4 does not know about the link between nodes 1 and 2. This means that when node 4 (being MPR for node 5) receives the message, it will retransmit it towards node 3, not knowing that node 3 will receive it from node 2 (unless node 3 relays the message first of course). But even in such a case, it might be difficult for node 4 to avoid sending the message for practical reasons; for example the packet may already have been queued for transmission and sent to the MAC layer when the packet from node 3 arrives. Furthermore, in more complex scenarios (all neighbors of the node might not already have sent the message), it is not as easy to determine that it is not necessary to retransmit the packet. After all, node 4 is MPR for both nodes 3 and 5, so in this example all 5 nodes will retransmit the packet, resulting in an increase of 67% in transmissions compared with the three transmissions of the tree-based approach. This might be one explanation for the difference between MPR flooding and the broadcast based on full graph trees.

In general, however, MPR flooding will give very good results in most cases and only requires two-hop information, which is much simpler to secure than

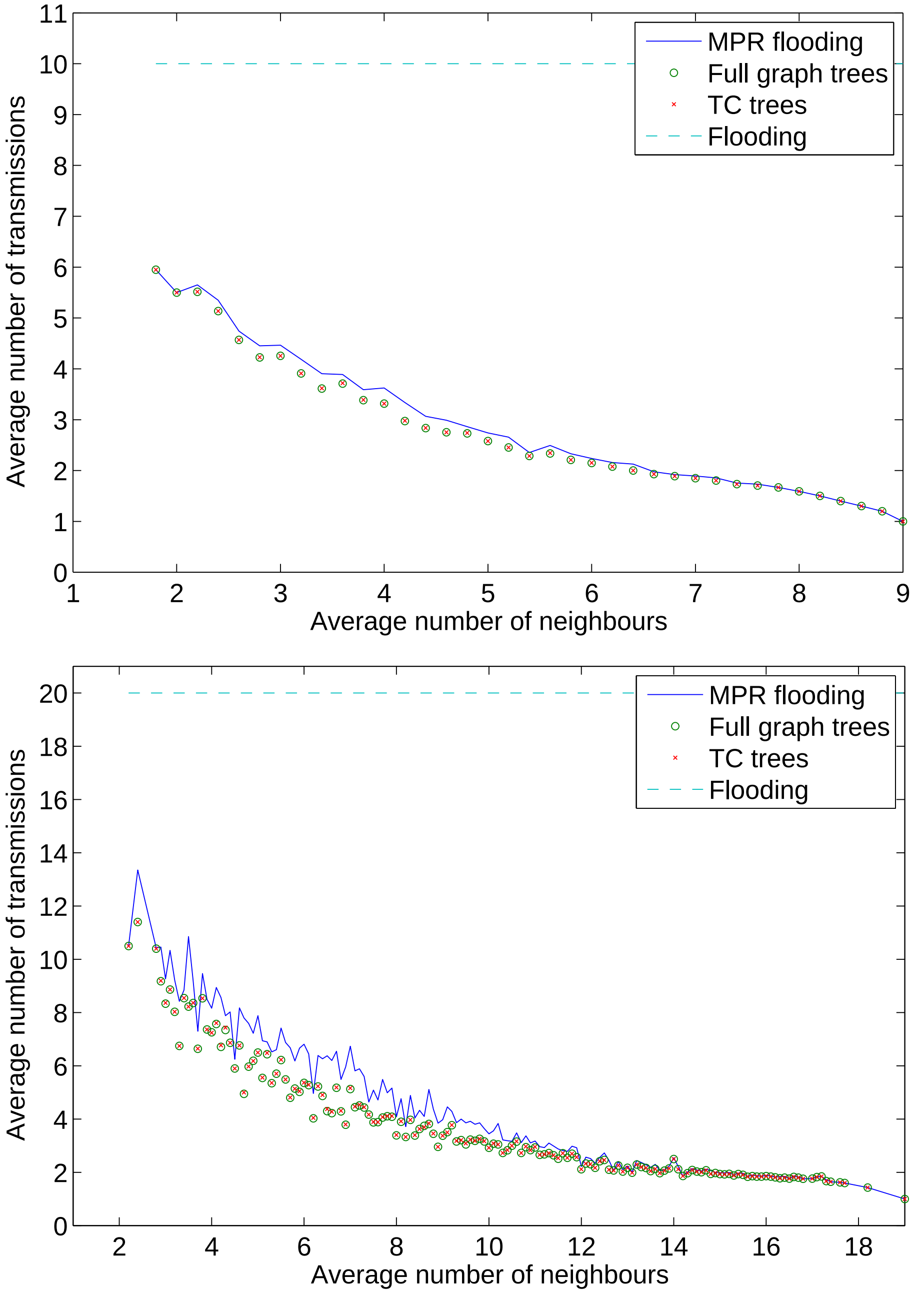


Figure 4.1: Comparison of broadcast techniques for different networks of size 10 (top) and 20 (bottom). Note that the results for TC trees and full graph trees almost coincide.

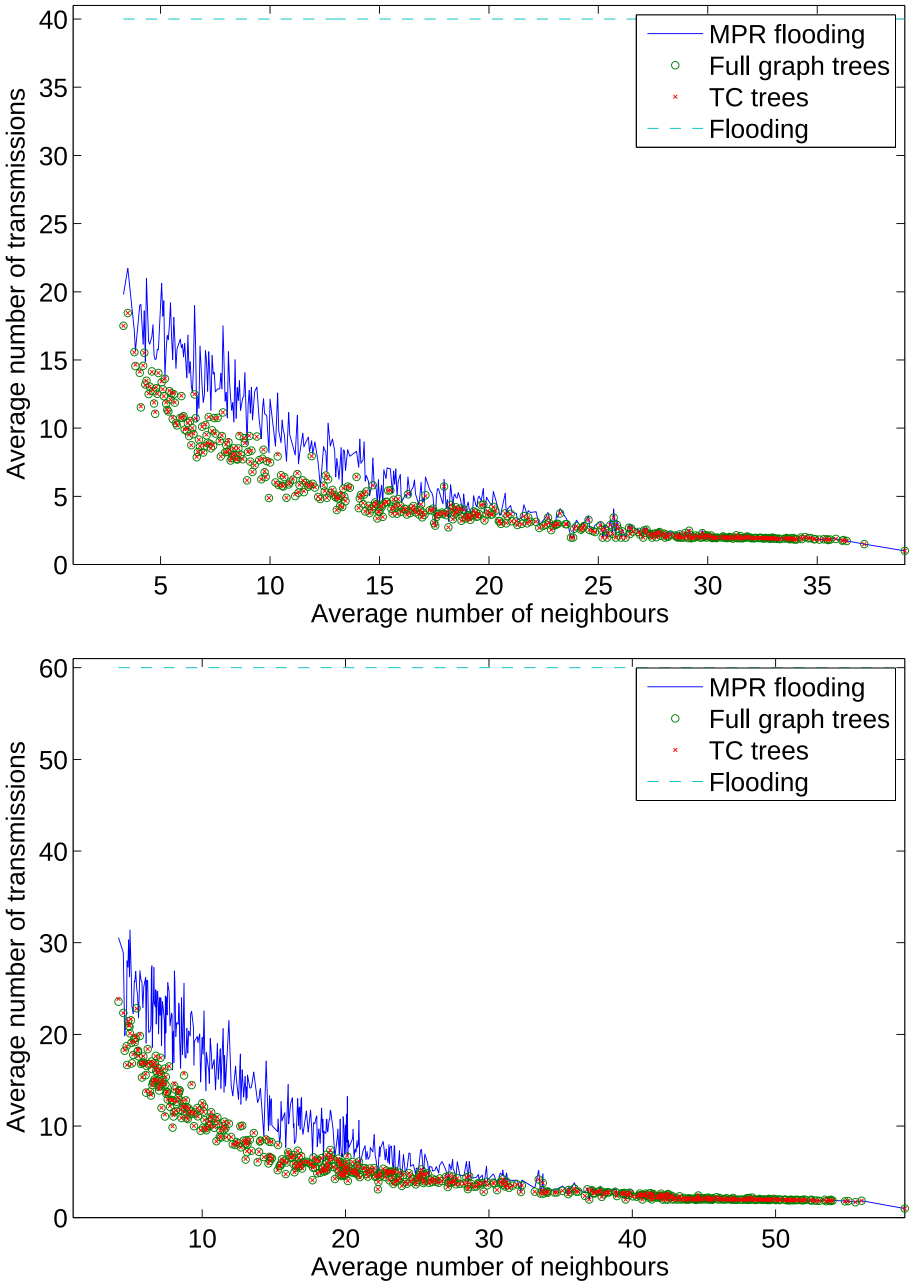


Figure 4.2: Comparison of broadcast techniques for different networks of size 40 (top) and 60 (bottom). Note that the results for TC trees and full graph trees almost coincide.

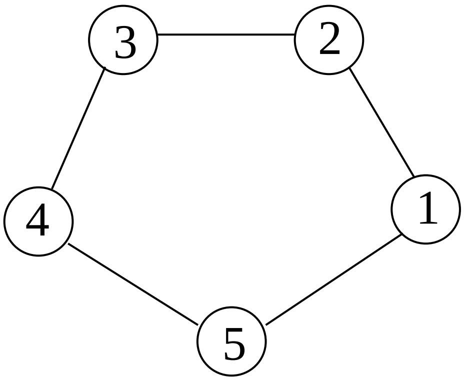


Figure 4.3: Example of a 5-node network where the 2-hop information of HELLO messages makes MPR flooding perform badly.

the TC messages, as was shown in the previous chapter.

TC trees

If we add information from the TC messages and use this for tree generation, the result is very good, as is shown in Figures 4.1 and 4.2 It gives more or less the same result as trees based on the full graph would. However, if we show the percentage of the links in the full graph that is unknown by the nodes through TC and HELLO messages, shown in Figure 4.4, we can see that almost all of the links in the network will be known. Given this the results are not that surprising. This means that if OLSR is run in the network for unicast traffic, its information can also be used for broadcast information and will yield very efficient broadcast trees.

However, there are limitations to using TC information for broadcasting. TC messages are harder to secure, as was discussed in previous chapter. Mobility also creates additional problems. In [16], MPR flooding, tree, and mesh-based broadcast routing are compared for mobile networks, examining, among other things, delivery rates for different mobility. They show that with mobility, tree-based approaches have a much lower delivery ratio than that of MPR flooding when mobility gets high.

For a tree-based approach to be as efficient as possible, a node should be

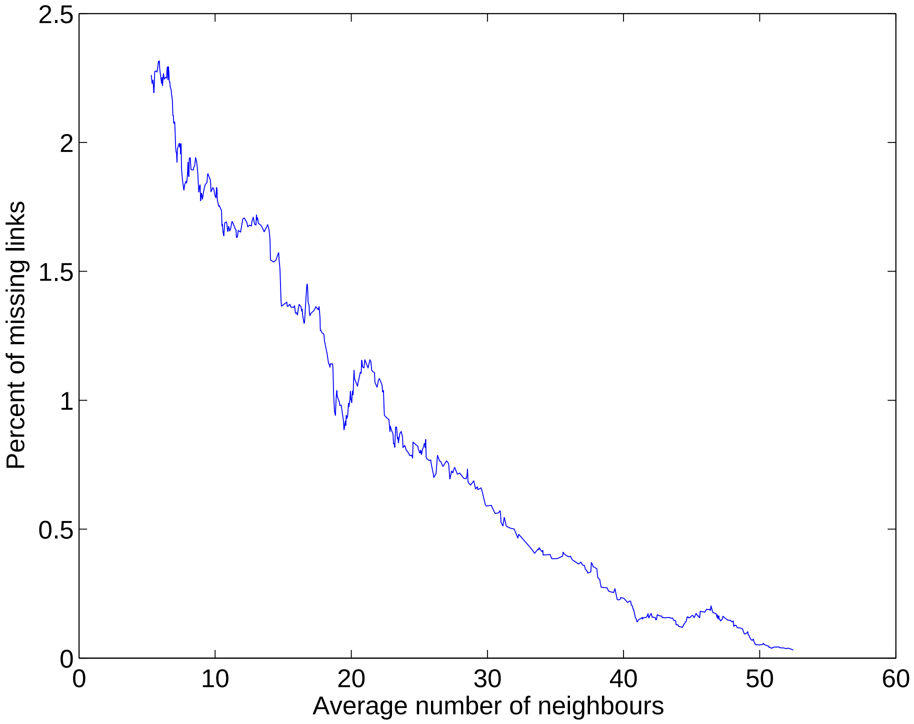


Figure 4.4: The average percentage of missing links in the topology information taken from received HELLO and TC messages in a node. The network size for the generated networks in this example is 60 nodes.

able to determine based solely on the packet source whether it should retransmit a packet. To do this, all nodes must have the same information, something that is very difficult to achieve in a mobile network. If the nodes do not have the same information, they must at least be given information about which nodes are further ahead in the tree. Otherwise a failed link between updates may lead to nodes not receiving the message (or possibly unneeded transmissions if links have been added). However, information added to the packets about the broadcast tree will lead to increasing overhead.

For OLSR TC messages are sent by default every 5 seconds. This means that it can take a long time for nodes far away to determine that a link has failed, necessitating tree updates. HELLO messages, which generates the information for MPR flooding, on the other hand, are sent every two seconds and are only local. Furthermore, there are usually several paths to each node; if one fails, another will function automatically because the packet will simply be received by other MPRs.

4.3 Concluding remarks

For relatively static networks where updates seldom happen, TC information can probably be used for efficient broadcasting, especially if capacity is high (which is more likely in a mostly static network compared with a mobile one).

However, if capacity is lower and mobility is an issue, both securing TC messages and updating them sufficiently fast will be very difficult without creating networks that are only capable of carrying overhead traffic. In such cases, MPR flooding is probably the preferred solution.

From the broadcast evaluation on random networks of size 10, 20, 40 and 60 nodes, we conclude that MPR flodding performs fairly well, but the number of transmissions can still be reduced by approximately 50% by using TC information in the 60-nodes networks.

Chapter 5

Conclusions

In this report we have studied the efficiency of some broadcast routing algorithms and Advanced Signatures as proposed in [10].

Broadcast and multicast are very important for military networks, especially so for mobile ad hoc networks. At the same time, these networks are both difficult to secure and have very limited capacity, making this problem highly relevant.

Adding advanced signatures is one way of securing HELLO messages and TC messages in OLSR, but if such methods are used on every packet the overhead cost for using OLSR will simply be too high in the kind of networks envisioned in the foreseeable future. However, it should be possible to tune OLSR with advanced signatures, so that not every message contains signatures. This might decrease overhead to a manageable level, although this needs to be balanced against the consequence of a less secure network. TC messages, in particular, are sensitive to this.

We have also studied the efficiency of broadcasting with different amounts of information. For relatively static networks where updates seldom happen, TC information can probably be used for efficient broadcasting, especially if capacity is high, which is more likely in a mostly static network compared with a mobile network.

However, for the mobile military networks, where capacity is highly limited, securing TC messages and updating them sufficiently swiftly will be very difficult while at the samt time being able to carry a reasonable amount of payload data. In these networks, MPR flooding is probably the preferred solution.

5.1 Future Work

OLSR is a good candidate as a routing protocol for military ad hoc networks. In this report we have shown that it can also be efficiently used for broadcast traffic. However, several things remain unclear, especially concerning mobile networks.

An extension of advanced signatures needs to be developed so that signatures are not needed on each transmitted message, probably in conjunction with an IDS that can detect whether nodes try to avoid sending signatures (or send too many in order to drown the network in overhead traffic).

It would also be important to conduct further simulations with mobility to determine delivery rates of messages versus overhead for the different methods

Bibliography

[1] M. Guerrero, "SAODV," *draft-guerrero-manet-saodv, internet draft*, (work in progress).

[2] Y. Hu and A. Perrig et al., "Ariadne: A secure on-demand routing protocol for ad hoc networks," in *Proceedings of MOBICOM 2000*, 2002, pp. 275–283.

[3] D. Raffo, "Security schemes for the OLSR protocol for ad hoc networks," Doctoral thesis, INRIA Rocquencourt, sep 2005.

[4] T. Clausen and P. Jacquet, "Optimised link state routing protocol (OLSR)," *RFC 3626*, 2003.

[5] Y. Zhang, W. Lee, and Y. Huang, "Intrusion detection techniques for mobile wireless networks," *Mobile Computing and Communications Review*, vol. 7, no. 1, pp. 74–94, jan 2003.

[6] C. Perkins et al, "On-demand distance vector (AODV) routing," *RFC 3561*, 2003.

[7] E. Hansson, J. Grönkvist, and J. Nilsson, "Intrångsdetektering i mobila ad hoc-nät," Technical Report FOI-R--1375--SE, Swedish Defence Research Agency., Div. of Command and Control. Linköping, Sweden, 2005, (In Swedish).

[8] Adjih et al, "Securing the OLSR protocol," in *Proc. of MedHoc 2003*, jun 2003, vol. 2.

[9] Marti et al., "Mitigating routing misbehavior in mobile ad hoc networks," in *Proc. 6th Annual int Conf. Mobile Comp. and Net.*, 2001, pp. 255–65.

[10] D. Raffo, C. Adjih, T. Clausen, and P. Mühlethaler, “An advanced signature system for olsr,” in *Proc. of the 2nd ACM Workshop on Security of ad hoc and Sensor Networks, SASN 2004*, oct 2004.

[11] B. Asp, G. Eriksson, and P. Holm, “Detvag-90® — Final Report,” Vetenskaplig Rapport FOA-R–97-00566-504–SE, Försvarets Forskningsanstalt, Avdelningen för ledningssystemteknik, Linköping, Sept. 1997.

[12] P. D. Holm, “UTD-diffraction coefficients for higher order wedge diffracted fields,” *IEEE Trans. Antennas Propagat.*, vol. AP-44, no. 6, pp. 879–888, jun 1996.

[13] J. Stevens et al., “Scenario based analysis of dynamic tdma ad-hoc tactical battlefield networking,” in *Proc. of MILCOM 2003*, 2003.

[14] Bevan Das and Vaduvur Bharghavan, “Routing in ad-hoc networks using minimum connected dominating sets,” in *ICC (1)*, 1997, pp. 376–380.

[15] M. R. Garey and D. S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-completeness*, Freeman, 1979.

[16] S.-Y. Cho, “Optimized multicast based on multipoint relaying,” in *Proceedings of the First International Conference on Wireless Internet*, 2005, pp. 42 – 46.