Lehrstuhl für Kommunikationsnetze

Technische Universität München

A Framework for Secure and Efficient Communication in Mobile Ad Hoc Networks

Christian Schwingenschlögl

Lehrstuhl für Kommunikationsnetze

Technische Universität München

A Framework for Secure and Efficient Communication in Mobile Ad Hoc Networks

Christian Schwingenschlögl

Vollständiger Abdruck der von der Fakultät für Elektrotechnik und Informationstechnik der Technischen Universität München zur Erlangung des akademischen Grades eines
Doktors der Naturwissenschaften
genehmigten Dissertation.

Vorsitzender:
Univ.-Prof. Dr.-Ing. Klaus Diepold

Prüfer der Dissertation:

1. Univ.-Prof. Dr.-Ing. Jörg Eberspächer
2. Univ.-Prof. Dr. rer. nat, Dr. rer. nat. habil. Uwe Baumgarten

Die Dissertation wurde am 28.09.2004 bei der Technischen Universität München eingereicht und durch die Fakultät für Elektrotechnik und Informationstechnik am 08.06.2005 angenommen.

Thanks

This thesis has been done at the Institute of Communication Networks (LKN) at the Munich University of Technology (TUM) with support from many excellent colleagues and students.

I want to thank Prof. Dr.-Ing. Jörg Eberspächer for his enduring support and all the inspiring discussions. I also want to thank Prof. Dr. Uwe Baumgarten who accepted to be the second auditor of this thesis.

Special thanks go to the members of the mobile communications group, especially Stephan Eichler, Christian Bettstetter, Jin Xi and Peter Tabery.

Last but not least, I want to thank my family for their love and encouragement. Especially Emanuela for her patience and understanding.

Contents

Nomenclature

4G	Fourth Generation (Mobile Networks)
AODV	Ad Hoc On-Demand Distance Vector Routing
BER	Bit Error Rate
CA	Certification Authority
CSMA/CA	Carrier Sense Multiple Access/Collission Avoidance
CSMA/CD	Carrier Sense Multiple Access/Collission Detection
CVS	Concurrent Versions System
DAB	Digital Audio Broadcast
DoS	Denial of Service
DREAM	Distance Routing Effect Algorithm for Mobility
DSR	Dynamic Source Routing Protocol
DVB	Digital Video Broadcast
EAP	Extensible Authentication Protocol
ETSI	European Telecommunications Standards Institute
GloMoSim	Global Mobile Simulation Systems Library
GPRS	General Packet Radio Service
GSM	Global System for Mobile Communication
IETF	Internet Engineering Taskforce
iff	if and only if
IPSec	Internet Protocol Security
IPv6	Internet Protocol Version 6
IP	Internet Protocol
LAN	Local Area Network
LIP	Local Information Point
LKN-ASF	LKN Ad Hoc Security Framework
LOS	Line of Sight
LSP	Local Service Point
MAC	Medium Access Control
Manet, Edouard	Father of Impressionism (1832-1883)
MANET	Mobile Ad Hoc Network
MPR	Multipoint Relay
MSC	Message Sequence Chart
NTDR	Near-Term Digital Radio
OLSR	Optimized Link State Routing Protocol
OSI	Open System Interconnection
PAN	Personal Area Network
PARSEC	Parallel Simulation Environment for Complex Systems
PDR	Packet Delivery Ratio
PHY	Physical Layer
PKI	Public-Key Infrastructure

PKZ	Public-Key Certificate
RA	Registration Authority
RERR	Route Error
RFC	Request for Comment
RREP	Route Reply
RREQ	Route Request
RTT	Round Trip Time
SDL	Specification and Description Language
SNR	Signal to Noise Ratio
SSL	Secure Socket Layer
TBRPF	Topology Dissemination Based on Reverse-Path Forwarding Routing Protocol
TCP	Transmission Control Protocol
TND	TBRPF Neighbor Discovery Protocol
TTL	Time To Live
UDP	User Datagram Protocol
UMTS	Universal Mobile Telecommunication System
V2V	Vehicle to Vehicle (Communication)
VCP	Virtual City Portal
VND	Virtual Network Driver
VPN	Virtual Private Network
VSL	Virtual Socket Layer
WEP	Wired Equivalent Privacy
WWRF	Wireless World Research Forum
ZRP	Zone Routing Protocol

1. Introduction

The increasing availability of heterogeneous wireless technologies results in great opportunities for the telecommunications industry as well as for its customers. The focus is shifting from anytime, anywhere communication to the efficient combination of the different networks advantages. This combination provides benefits for the mobile user as he can continue his applications in a different network if his previous connection becomes unavailable. If more than one network is available, the "best" network can automatically be selected, transparently to the user. This selection can be based on application requirements, user preferences, costs, etc. Even totally self-organizing networks can be formed between clients, enabling a range of completely new applications.

1.1. MANETs: Networks for 4G

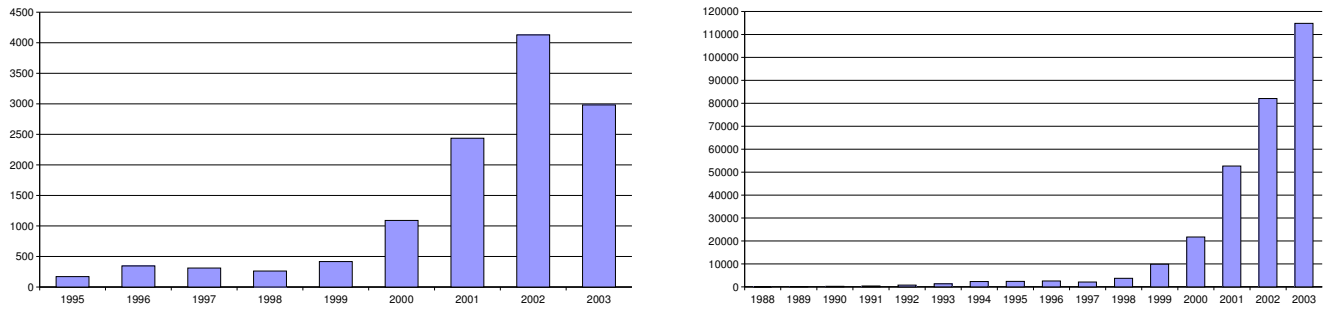
Among others, the automotive area will greatly benefit from this trend. Vehicles are not anymore self-contained systems, but connectivity and communication with the world outside the car enables new possibilities. Mobile devices connect with the vehicle environment and benefit from its technical infrastructure (display, speakers, etc). Vehicles can make use of different wireless transport systems outside, to provide services to applications inside the car. In the future, cars are expected to be more and more integrated into a global network.

The benefits of combining the advantages of various networking technologies has been recognized by the telecommunication industry as well as potential customers. After the start of 3G mobile communications, research is proceeding towards the next generation - 4G (e.g. [EC03]).

The major research questions in this area include topics in software engineering (e.g. dynamic service set-up using available modules at runtime [Rei03]), the seamless integration of heterogeneous networks with widely varying characteristics (e.g. handover issues, accounting, etc.), design of suitable user interfaces and, last but not least, the main topic of this thesis: how to meet the tough security challenges in this integrated world. This is especially the case when self-organizing networks with its performance constraints and vulnerabilities are part of the network mix.
A more extensive review of the technical issues concerning automotive applications in this environment as well as the "communication server concept" as one approach to realize such services can be found in [KBS+01] and [SGS01].

1.2. Network Security - a Big Challenge

The benefits of combining the advantages of heterogeneous wireless networks and especially the attractive properties of MANETs, namely no network costs, inherent location awareness,



(a) Reported Vulnerabilities 1995 - Q3 2003

(b) Reported Security Incidents 1988 - Q3 2003

Figure 1.1.: Reported Vulnerabilities and Security Incidents (Source: CERT)

decentralized and almost real-time information dissemination with high transmission rates, enable completely new types of applications. However, security is a major issue in this environment. Especially the highly decentralized and distributed MANETs are vulnerable to totally new classes of attacks and, at the same time, security mechanisms operating in this environment have to face strict performance constraints. Thus, it is necessary to develop a lightweight security solution with respect to its communication overhead. Especially in highly mobile scenarios it is also necessary to avoid additional delays as much as possible.

Current statistics from CERT [Uni] show the development of security problems in the Internet until today. Figure 1.1 depicts the development of reported vulnerabilities and reported security incidents. This also represents an excellent motivation for following very strict security measures for the networking of highly safety-critical systems like vehicles and all the electronic systems within.

As will be shown in section 4, the security problems are in no way limited to the wired Internet itself. Wireless devices face exactly the same risks and, in most configurations, the vulnerable wireless link that offers no physical protection at all, creates new possible holes for malicious entrance. A more detailed coverage of security issues in telecommunications is available in [ET04] and [Ebe99]. Very efficient attacks and combinations (some of them will be shown in section 4) with severe consequences even for the availability of the network itself are possible against self-organizing MANETs. The CERT statistics (figure 1.1) in addition give a very strong hint that if security holes exist, the exploits will be there.

1.3. Contribution of the Thesis

New and effective attacks are possible against MANETs. Countermeasures exist, however its highly distributed mode of operation results in the transfer of security responsibility to the network nodes. This often results in costly operations or impractical restrictions when new

nodes have to be added to the network. One of the most important problems, however, is the mostly separated view on MANET security and MANET performance. While MANETs face strict performance constraints, highly interactive security mechanisms can overload the network with its control traffic.

Within this thesis, a framework for secure and efficient communication in MANETs is presented. The framework allows the provider to keep in control of network security. It also allows the fast extension of the network with new nodes. The distributed security credentials can not only be used to secure the MANET itself but also to provide basic security mechanisms on higher layers. As an example for this point, security issues concerning the service location protocol (SLPv2) are presented. It has to be stressed that the security framework itself does not solve all security issues present in MANET environments. It provides some basic security mechanisms and efficiently distributes security credentials among the nodes. These credentials are a prerequisite for the efficient operation of more sophisticated security mechanisms on top of this framework. For simple interoperability, the framework relies on X.509v3 certificates.

To estimate the suitability of this framework even for highly mobile scenarios, measurements have been done using IEEE 802.11b WLAN cards in a vehicular testbed. The results presented in this thesis show that even at high speeds enough data can be transmitted between the nodes to keep the security protocol operational and to have reasonable room for additional user data.

The framework itself has been implemented in SDL for first functional tests. It has also been implemented in the GloMoSim Simulator for further performance studies. Static and mobile simulation scenarios with and without an activated security framework are presented to show its performance impact. In addition, different aspects of the framework are studied in more detail to be able to estimate the influence of various parameters on the protocols performance.

1.4. Overview of the Thesis

Section 2 explains the role of MANETs in 4G networks. It references mechanisms for the seamless combination of heterogeneous networks. Throughout the literature, the term MANET is used for networks with widely differing properties. Therefore section 2 shows the range of possible MANET configurations and defines the terminology used in this thesis. Based on the work of the Internet Engineering Taskforce (IETF, [iet03]), the Wireless World Research Forum (WWRF, [wwr03]) and further research, an overview about existing work, upcoming standards and open research topics is given. As useful background especially for the performance and security chapters, this overview focuses on routing protocols, ad hoc - Internet gateways and service discovery protocols.

Section 3 outlines performance issues in ad hoc networks. This includes fundamental results concerning capacity constraints of ad hoc networks as well as simulative studies and results from testbed implementations. An overview about the complexity of different routing protocol classes is given. This chapter helps to understand existing performance constraints in MANETs and the resulting necessity for efficient protocol design. It also helps to understand the impact of security algorithms on system performance as it presents results gained

in systems without additional security. The testbed results presented in this chapter focus on practical implementation issues of IEEE 802.11 based, highly mobile scenarios. Also results concerning the necessary grade of detail for modeling lower layers in MANET simulations and the suitability of the Transmission Control Protocol (TCP) for IEEE 802.11 based MANETs are presented.

Section 4 gives an overview about the current situation concerning secure communication in MANETs and 4G networks. Starting with well-known security issues in the Internet, different requirements for secure communication in wireless networks and especially in MANETs are explained. Attacks specific to MANETs and 4G networks are presented. Building blocks for securing such networks are explained and checked for its practicability. As service discovery protocols are envisioned to be widely used in self-organizing environments, related security issues and countermeasures are discussed based on the service location protocol SLPv2.

In section 5, the LKN Ad Hoc Security Framework (LKN-ASF) is presented. LKN-ASF creates a Public-Key Infrastructure (PKI) based on X.509 certificates and is applicable to MANETs that have access to a bigger network (e.g. the Internet) at least from time to time. It is not necessary to have continuous Internet access, hence it is suitable for a wide range of 4G and MANET scenarios. In addition, the combination of LKN-ASF with existing security mechanisms and security services for higher layers is discussed.

Section 6 outlines results of the LKN-ASF performance evaluation as well as details on the simulation model used. Network performance and LKN-ASF parameters have been measured in different scenarios and with a number of different LKN-ASF configurations. Promising results concerning network load and functionality of LKN-ASF even in large scenarios are presented.
Finally, section 7 concludes the thesis and discusses further research topics in this area.

2. The Role of MANETs in Fourth Generation Mobile Networks

MANETs, an essential part of upcoming 4G mobile networks, are in the focus of this chapter. As will be seen, new applications are enabled by the self-organizing behavior of these networks. As deployment scenarios for MANETS differ widely, an overview about possible scenarios and widely differing characteristics of different variations of MANETs is given. After that, basics about common MANET routing strategies are explained. This is a prerequisite for the work presented in chapter 3 and the work on security starting with chapter 4. An implementation of gateways for connections between MANET nodes and nodes in the Internet is presented. Such gateways are essential for the security mechanism presented in chapter 5. Finally, the basic operation of the service discovery protocol SLPv2 is explained. Using this protocol, chapter 4 shows the security implications of spontaneously forming networks on higher layers. As will be seen in chapter 4 and 5, the LKN-ASF security framework can also be used as foundation for enhanced security on this layer.

This chapter is organized as follows: section 2.1 outlines the idea of fourth generation networks. In section 2.2, concepts for the integration of heterogeneous networks are presented. Details about essential protocols for MANETs are given in section 2.3. Ad hoc - Internet gateways, used for the realization of the LKN-ASF security framework presented in chapter 5 are described in section 2.4. Finally, an overview about the service discovery protocol SLPv2 is given in section 2.5.

2.1. 4G - Idea and Development

The development of mobile communications is usually described in generations. Analogue cellular systems like C-Net, TACS, NMT are referred to as first generation and have been available commercially since 1979. The second generation followed with a huge success in 1991, since then digital GSM networks are available. Currently, migration to the third generation (UMTS, Universal Mobile Telecommunication System) is in progress. While the first UMTS networks are already available, second generation GSM networks and so-called 2.5 generation networks (GPRS, EDGE, HSCSD) are still around and heavily used. A coexistence of these networks is expected for quite some time, among other reasons because third generation networks will initially be available mostly in urban areas only.

The common understanding (e.g. [wwr03]) of 4^{th} generation mobile communications is not the development of one new technology to replace previous generations but the seamless integration of existing network technologies. However, also a new networking technology is commonly seen to be a part of 4G, namely Ad Hoc networks (see section 2.3 for a detailled introduction).

This step was motivated by the rise of various heterogeneous wireless networks with different characteristics. Part of this vision are not only cellular networks as outlined above but networks ranging from broadcast technologies like Digital Video Broadcast (DVB) and Digital Audio Broadcast (DAB) to short-range technologies like Bluetooth. One of the driving ideas behind this seamless integration is that, given mobile users, different types of networks are available depending on the actual location of the user. In addition, different services and applications may have different requirements on the characteristics of the underlying network. Table 2.1 gives an impression of the different characteristics of networking technologies available today. Finally, the step from 3G to 4G can not only be seen as a shift in technology but also as a shift in focus from technologies to customers and services. Among others, the success of NTT DoCoMo's iMode in Japan which has been advertised service-oriented instead of the technology-oriented marketing of its much less successfull pendant in Europe, WAP, was a strong hint in this direction.

System	Data rate	Coverage	Mobility	Frequency range	Best suited for service class
Cellular networks					
GSM	9.6 kb/s	Country-wide	High	900, 1800, 1900 MHz	Telecomm. services, transparent access to the Internet, low bit-rate data retrieval
GSM/HSCSD	$\leq$ 50 kb/s				
GSM/GPRS	$\leq$ 50 kb/s				
GSM/EDGE	$\leq$ 150 kb/s				
UMTS	about 100 kb/s (vehicular) $\leq$ 2 Mb/s (fixed)	Country-wide?	High	2 GHz	Telecomm. services, Internet, info push
Wireless LANs (WLANs)					
IEEE 802.11	$\leq$ 2 Mb/s	50-300 m	Low	2.4 GHz	Ad hoc, info station
IEEE 802.11b	$\leq$ 11 Mb/s	50-300 m	Low	2.4 GHz	Ad hoc, info station
IEEE 802.11a	$\leq$ 54 Mb/s	50-300 m	Low	5 GHz	Ad hoc, info station
IEEE 802.11g	$\leq$ 54 Mb/s	50-300 m	Low	2.4 GHz	Ad hoc, info station
Broadcast					
DAB	1,5 Mb/s	$\leq$ 100 km	High	e.g. 176-230 MHz 1452-1467.5 MHz	Info casting, info push, data retrieval
DVB-T	Mobile: 5-8 Mb/s Fixed: 16-31 Mb/s	$\leq$ 100 km	Medium to High	TV bands below 860 MHz	Info casting, info push, data retrieval
Short-range (peer-to-peer)					
Bluetooth	$\leq$ 1 Mb/s	10 or 100 m	Very low	2.4 GHz ISM band	In-car ad hoc
IrDA	Ext. SIR: $\leq$ 4 Mb/s VFIR: $\leq$ 16 Mb/s	Line of sight $\leq$ 2 m	None	IR	In-car ad hoc

Table 2.1.: Characteristics of various mobile networks

4G will allow the creation of totally new services, enabled by the transparent selection of the best suited network available. To achieve this goal, some open research questions still have to be resolved. Among them are the seamless integration of these heterogeneous networks, questions concerning the dynamic service creation in such systems (services and applications may be created dynamically from basic building blocks found in the network) as well as security and privacy implications of such systems. For the practical realization of such systems, however, also economical aspects like changing value chains play an important role [Hub01]. Currently, the Wireless World Research Forum [wwr03] with its members from industry and academia works in cooperation with standardization bodies to identify still open research issues for 4G networking and towards standardized solutions for the time after 3G.

2.2. Concepts for Network Integration

Network integration allows the usage of different networks, depending on user preferences, application requirements and network availability. Most use cases require the transparency of these vertical handovers to the user. Several techniques exist to achieve this transparent

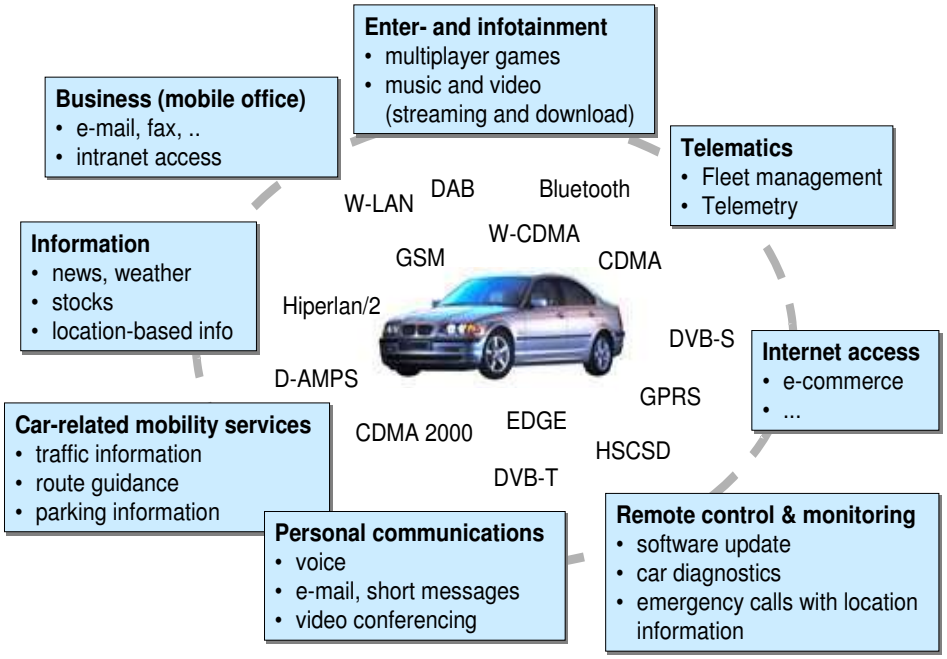


Figure 2.1.: Examples of Vehicular Applications Profiting from 4G Mobile Networks

network selection. Among them are the communication server approach, virtual sockets and HTTP proxies. These approaches are briefly explained here and further references for the interested reader are given.

The communication server is a concept for 4G mobile communication in the vehicular environment and has been presented in [KBS+01] and IPCommServ [SGS01]. The service types envisioned for the automobile environment include information casting, information push, data retrieval, Internet and intranet access, telecommunication services, access to local information points, ad hoc communication between vehicles and in-car communications. Based on this architecture, a prototype has been developed that is presented in [KSB02]. Closely related to this architecture are the security issues presented in sections 4 and 5. In addition to communication security itself, any unauthorized access to critical vehicular systems has to be prevented.

In [LÓ2], the design and implementation of a system for network selection based on virtual sockets is described. Its main components include context manager, link availability filter, service discovery manager, virtual socket layer and virtual network driver.
The context manager is responsible for the maintenance of information concerning the users environment. This includes the position as well as several other parameters e.g. the current speed, temperature, time, etc. A detailed description of all functions is available in [KSL02], [Li02] and [Per02]. Components like routing and service discovery modules make direct use of this information. The link availability filter checks whether the link from one node to a particular destination is available with a certain probability during some period of time.

This information is useful to avoid unnecessary actions by several ad hoc system components. E.g. service discovery signaling can be prevented with peers having a poor link availability estimation. These decisions are done within the service discovery manager that has been described and implemented in [Per02]. The virtual socket layer is responsible for transparent network selection based on user preferences and network availability. If a network becomes unavailable due to e.g. user mobility, the virtual socket layer automatically switches to the best remaining network. Based on the decision of the virtual socket layer, the virtual network driver forwards packets to the appropriate network interface. A detailed description of the virtual network driver and its Linux-based implementation can be found in [LÓ2].

Another concept for horizontal and vertical handovers has been developed in [Bla02]. One of its main advantages is that it works without modifications of the protocol stacks in the client devices. Instead, concerning the mobile clients, the system is designed to work on the application layer only. [Bla02] focuses on horizontal handovers in a Bluetooth environment, however, also vertical handovers to other types of networks like GSM or IEEE 802.11 Wireless LAN are possible. Also a prototypical implementation has been completed and tested during the thesis.

The work above mainly focused on technical realizations for network selection and handovers. Related to this work, a strategy for cost-optimization and prefetching in 4G environments has been developed in [Was03]. Prefetching mechanisms are especially helpful in situations where only a limited information set is available that is additionally correlated with context information. Tourist information systems are a good example for this. [Was03] describes a system to optimize costs via intelligent network selection. Costs in this context are not necessarily restricted to monetary costs but can also include parameters like waiting time or costs for energy consumption. These parameters are weighted according to the user preferences.

2.3. Ad Hoc Networks

A "mobile ad hoc network" (MANET) is an autonomous system of mobile routers (and associated hosts) connected by wireless links–the union of which form an arbitrary graph. The routers are free to move randomly and organize themselves arbitrarily; thus, the network's wireless topology may change rapidly and unpredictably. Such a network may operate in a standalone fashion, or may be connected to the larger Internet.
Source: IETF MANET charter ([IET]).

2.3.1. Different Types of Ad Hoc Networks

The idea of ad hoc networks has been around for quite some time, sometimes similar techniques have been proposed under different names. Among them, packet radio networks [ea78], [JT87], instant infrastructure [BGK+96] and mobile-mesh networking [mob95]. Also the military has been and is heavily interested in mobile ad hoc networks. In 1983, the Survivable Radio Networks (SURAN) program was initiated by DARPA. Within this program, a low-cost packet radio [FB87] and a set of network management protocols [SW87] taking a hierarchical approach to handle networks of large scale have been developed. Currently, several MANET activities are going on within the military, among others the U.S. Army's Task Force XXI

Advanced Warfighting Experiment which built a tactical internet, the U.S. Navy and Marine's Extending the Littoral Battlespace Advanced Concept Technology Demonstration and the DARPA Global Mobile Information Systems program. In March 1997, Task Force XXI showed a large-scale implementation comprising thousands of nodes of a mobile, wireless, multihop packet radio network [Per00]

So far, research has been done on different topics, probably the most active ones include routing algorithms, energy management and security. There have been many notable advances in several areas, especially a variety of different routing protocols has been proposed. However, it is difficult to find optimal solutions without defining the term "ad hoc network" in more detail. Concerning previous work, algorithms have been proposed for ad hoc networks ranging from sensor-networks e.g. Pebblenets [BHBR01] to networks for vehicular- or military applications like [KSL02] and [HBE+01]. Given the wide range of possible configurations of ad hoc networks, it becomes clear that mechanisms have to be developed or at least have to be adapted and optimized to fit the needs of the specific characteristics of network and application. Otherwise permissive overhead regarding computing time and network traffic or a complete inability to communicate are likely results. Therefore, this section provides an overview about different classes of ad hoc networks and its properties.

2.3.1.1. Applications

To illustrate the widely varying demands on ad hoc networks, the following exemplary list shows some of the applications described throughout the literature:

- **Office scenarios:** conferences, group communication, enhancement of WLAN infrastructure. Users demand connectivity even when they are gathering outside the office environment and the usual infrastructure is missing. Direct MANET-style connectivity for collaborative work may even be desirable within the office as they are often easier to establish and provide better performance than classical Internet protocols supporting mobility (e.g. Mobile IP [Per96]).

- **Personal area networks (PANs):** the short-range communication between devices that are closely associated with a single person. Mobility becomes more important if different PANs have to be interconnected.

- **Home Area Networks:** Communication with networked devices at home either remotely or, without additional configuration, locally with higher performance using MANET techniques.

- **Vehicular ad hoc networks (e.g. among cars, planes, ships, helicopters):** These types of networks have a wide field of application. Possible scenarios range from infotainment services to safety assistance of drivers.

- **Networked pedestrians:** Also low-mobile users like pedestrians or byciclists can profit from MANETS. Possible applications include the transmission of warning messages (pedestrian recognition) to vehicles in dangerous situations.

- **Sensor-networks (with and without mobility):** Some of these networks only have to organize its routing tables once after deployment as the nodes are not mobile (e.g. environmental monitoring), in other cases, continuous mobility has to be taken into account. Common among sensor network scenarios is the low performance of the network nodes with regard to computing and battery power [EGHK99], [KKP99].
- **Range extension of cellular networks (e.g. GPRS, UMTS):** These applications try to either extend coverage of cellular networks in areas of poor coverage or try to establish basic load-balancing functionality through routing some of the traffic via MANET-protocols to neighboring base-stations in overload-situations.
- **Disaster area applications:** In cases where no network infrastructure is available, MANETs can be set up relatively quickly and support the coordination of relief efforts.
- **Military networks:** For these networks, the focus is on setting up a robust and survivable communication infrastructure without the need for easily detectable central base-stations.
- **Embedded Computing Applications:** Communication among intelligent mobile machines in a smart, "ubiquitous computing" environment [Wei93].

2.3.1.2. MANET Characteristics and Nomenclature

From the variety of these applications it becomes clear that very different assumptions are made concerning the usage patterns and the environment the ad hoc network operates in. Also, different demands on the network are coming from these applications. Among them are requirements regarding Quality of Service (e.g. delay, throughput, jitter) but also scalability, security and many more. The following parameters can be used to characterize ad hoc networks and, therefore, help to compare protocols and mechanisms developed for them. As basis for the definitions below, the network is described as graph $G = (V, E)$ with its set of vertices V and edges E.

- **Node type:** the node type heavily influences the capabilities of the ad hoc network. Throughout the literature, node types ranging from static sensors to highly mobile nodes like vehicles or planes are described. Besides the mobility patterns, this parameter influences the available energy supply, the computing power within the node and the radio technology used (frequency, maximum throughput, communication range $r(v)$).
- **Network lifetime:** this parameter may vary considerably, e.g. the network can only be formed temporarily for online collaboration within a meeting or be designed for a much longer lifetime (e.g. disaster area applications).
- **Number of nodes:** $N = |V|$ in the network.
- **Number of edges:** $M = |E|$ in the network.
- **Average degree of a node (the degree is the number of direct neighbors of a node):** $\Delta = \frac{2M}{N}$.
- **Link density:** $d = \frac{2M}{N^2} = \frac{\Delta}{N}$.

- **Link breakage rate:** μ.
- **Average length of a route:** L.
- **Route creation rate per node:** λ.
- **Number of active routes per node (activity):** a.
- **HELLO rate (number of HELLO messages per node per second):** h.
- **Topology broadcast rate (number of broadcast information packets per node per second):** τ.
- **Average size of the topology broadcast packets:** δ.
- **Average number of retransmissions per broadcast:** R.
- **Broadcast optimization factor:** $r = \frac{R}{N}$.
- **Number of active nodes at time t:** number of nodes that are active (i.e. communicating) within the network at the same time. This parameter influences the demands on scalability and throughput.
- **Openness:** whether new members can join the network during operation or not.
- **Mean distance between the nodes:** $\overline{d} = \frac{\sum_{i,j} d(ij)}{|E|}$.
- **Standard deviation of distances:** $\sqrt{\frac{\sum_{i,j}(d(ij)-\overline{d})^2}{|E|}}$.
- **Environment:** this parameter defines the environment the ad hoc network is operating in with respect to radio propagation properties. I.e. is the network operating indoors, outdoors within a city or in rural areas.
- **Shares of data exchange types:** the typical communication patterns are described with this parameter. I.e. is most of the traffic within the network broadcast, multicast or point-to-point?
- **Shares of traffic types:** this parameter is very much influenced by the applications running in the network and defines traffic load (e.g. notifications sent from time to time vs. video streaming) as well as its characteristics (e.g. smooth vs. bursty).
- **Location awareness:** this parameter defines whether the nodes are aware of its positions. If this is the case, better performance can be gained by using algorithms that exploit this property. An example for the usage of location in routing algorithms is shown in section 2.3.2.5.
- **Mobility patterns:** defines whether the nodes are static, free to move arbitrarily or bound to certain movement patterns (e.g. vehicles on the road).
- **Connectivity:** this parameter gives a value for the graph connectivity, e.g. is the graph not connected, 2-connected or complete. The average number of neighbors $\frac{|E|}{|V|}$ is another parameter that can be used for this purpose.

- **Dynamicity:** the mean relative velocity between any two connected nodes $\frac{\sum_{i,j} \overrightarrow{v(ij)}}{|E|}$ can be used to estimate the dynamicity of the network.

- **Rate and distribution of static nodes and possibly access points.**

2.3.2. Routing Algorithms

Given the high demands on self-organization, mobility support and performance, it was clear from the beginning that conventional routing protocols known from the Internet would not be an optimal solution for routing in MANETs. These routing protocols have not been designed to be used in an environment in which the network topology can change rapidly and are also not prepared to handle low data rate and high bit error rate links. Experiments with existing routing protocols, e.g. OSPF (Open Shortest Path First), supported these first assumptions by showing long network settling times [SW96].

As a result of this, research started to find better routing algorithms for MANETs. Different approaches have been followed to optimize routing for different situations. The main classes of routing algorithms proposed are

- **Proactive Routing Protocols (aka hello protocols, table driven protocols):** These protocols react to every change in network topology.

- **Reactive Routing Protocols (aka flooding protocols, on-demand protocols):** Routes between two nodes are established only if data has to be transferred between those nodes.

- **Hybrid Routing Protocols:** These protocols combine proactive and reactive protocols in order to achieve better overall performance. Typically, proactive and reactive protocols are used in its original form in different parts of the network.

- **Position Based Routing Protocols:** These protocols are using the geographical coordinates of participating nodes to improve its performance.

As of 1998, nearly a dozen of candidate routing protocols [MC98] have been proposed and were discussed within the IETF. Currently, the IETF MANET working group is narrowing its focus. Ad Hoc On Demand Distance Vector Routing (AODV) [PBRD03] and the Optimized Link State Routing Protocol (OLSR) [CJ03] have been published as experimental IETF Request for Comments (RFC). Work is going on to publish two more routing protocols as RFCs, namely the Dynamic Source Routing Protocol (DSR) [JMH03] and Topology Dissemination Based on Reverse-Path Forwarding (TBRPF) [OTL03] - both protocols are available as Internet drafts currently.

The following paragraphs describe the different design approaches of MANET routing algorithms and exemplarily present one algorithm of each class in more detail. Performance considerations (e.g. the suitability of different protocols for different scenarios) are not presented here but can be found in section 3.

2.3.2.1. Proactive Routing Algorithms

Protocols within this class keep track of routes for all destinations in the ad hoc network. Thus, they have the advantage of an minimal initial delay for communications with arbitrary destinations from the point of view of the application. Anytime when two nodes need to communicate, the route information can immediately be selected from a routing table. This is the reason why such protocols are called "proactive": route information is collected and stored even before it is needed.

The disadvantage of such protocols, however, is the overhead in control traffic that is needed to keep track of the changing routes within a MANET. As this is considered to happen frequently in such environments, the repair effort (the deletion of stale routes due to link failure, detection of new routes, propagation of this information through the network) can be considered totally wasted in the case no application is using this route during its lifetime. As network capacity is scarce [LBC+01] in MANETS, any unnecessary control traffic has to be avoided. Therefore, reactive routing algorithms have been designed that find and update routing information only when needed.

2.3.2.1.1. OLSR The Optimized Link State Routing Protocol (OLSR, [CJ03]) is a proactive, table-driven protocol, i.e. it exchanges topology information with other nodes in the network regularly, also without prior requests for data communication. Its concept of forwarding and relaying are based on HIPERLAN standardized by ETSI [Com96]. To avoid excessive traffic overhead, the problem of this protocol in highly mobile scenarios, the protocol makes use of Multipoint Relays (MPRs). Further, OLSR requires only partial link state to be flooded in order to provide shortest path routes. As OLSR continuously maintains routes to all destinations in the network, the protocol is beneficial for traffic patterns where a large subset of nodes are communicating with another large subset of nodes, and where the source-destination pairs are changing over time. [CJ03] states that the larger and more dense a network, the more optimization can be achieved as compared to classic link state algorithms. OLSR defines a core functionality and auxiliary functions. Only the core functionality is described in this summary, the auxiliary functions can be found in [CJ03].

The basic steps performed by OLSR to create route tables are:

- **Link Sensing:** Each node periodically broadcasts HELLO messages via all interfaces through which connectivity is checked. A local link set, describing links between local and remote interfaces (i.e. interfaces on neighboring nodes), results from this link sensing procedure. Alternatively, if sufficient information is provided by the link-layer, it can be used instead of the HELLO broadcasts to create the local link set.
- **Neighbor Detection:** Assuming the case of only single interface nodes, any node can detect the neighbor set directly from the information exchanged as part of link sensing. [CJ03] also defines the case of nodes with multiple interfaces. Also a 2-hop neighbor set is maintained through periodic exchange of HELLO messages.
- **MPR Selection:** Each node selects a subset of its neighbors as MPRs. Hereby, the following condition has to be fulfilled: broadcast messages retransmitted by a nodes MPRs have to be received by all of the nodes neighbors 2 hops away.

- **Topology Control Message Diffusion:** Here, each node in the network has to be provided with sufficient link-state information for route calculation.
- **Route Calculation:** The routing table for each node is computed based on the link state information acquired through periodic message exchange and the interface configuration of the nodes.

2.3.2.1.2. TBRPF - Topology Dissemination Based on Reverse-Path Forwarding TBRPF [OTL03] also is a proactive, link-state routing protocol which provides hop-by-hop routing along shortest paths to each destination. Each node running TBRPF computes a source tree providing shortest paths to all reachable nodes based on partial topology information stored in its topology table. This is done using a modified Dijkstra algorithm. Only parts of each nodes source tree are reported for overhead reduction. This technique allows the support of much larger and denser networks than routing protocols based on the classical link-state algorithm (e.g. OSPF). TBRPF performs neighbor discovery using so called differential HELLO messages, i.e. only changes in the status of neighbors are reported. TBRPF basically consists of two independent[1] modules, the neighbor discovery module and the routing module that are now explained in more detail:

- **Neighbor Discovery Module:** The TBRPF Neighbor Discovery protocol (TND) allows each node to quickly detect its neighbor nodes such that a bidirectional link exists between an interface of the two nodes. TND also quickly detects when a bidirectional link breaks or becomes unidirectional. Differential HELLO messages which are much smaller as e.g. OSPF HELLO packets that include the IDs of all neighbors, are used by TND. Therefore, such packets can be sent out more frequently and topology changes can be detected faster. Compared to OLSR, the TND only detects 1-hop neighbors, the detection of 2-hop neighbors is done within the routing module. A neighbor table is maintained by each TBRPF node, containing state information for each neighbor. The status of each link can be 1-way. 2-way or lost. Also, link metrics can be used for neighbor detection. E.g. the status of a neighbor will then be changed only if the metric is above or below some threshold.
- **Routing Module:** Similar to OLSR, the TBRPF routing module tries to minimize overhead traffic in case of topology changes. Each node maintains a source tree T which provides shortest paths to all reachable nodes. This tree is calculated based on partial topology information using a modification of Dijkstra's algorithm. Only a subtree ST of T is forwarded to neighbors to minimize overhead. Periodic updates are used to report ST to neighbors, differential updates containing only changes to ST can be sent more frequently. ST is calculated as follows: ST consists of links (i, j) of T such that i is in the so called reported node set RN. A node i includes a neighbor j in RN if and only if (iff) i determines that one of its neighbors may select i to be its next hop on its shortest path to j. i can do this by calculating the shortest paths up to 2 hops from each neighbor to each other neighbor using only neighbors (or i itself) as an intermediate node. After i has determined which neighbors are in RN, each reachable node r is included in RN iff the next hop on the shortest path to r is in RN. Also i itself is included in RN. Each node is required to report ST but may report additional

[1] i.e. they can be used separately - the routing module with other routing algorithms and vice versa.

links to provide increased robustness in highly mobile networks. TBRPF also allows the usage of link metrics in topology updates, therefore paths can be calculated that are not shortest with respect to the hop-count but to the link metric used.

2.3.2.2. Reactive Routing Algorithms

Reactive routing algorithms for MANETs are only acquiring routing information when it is actually needed. Therefore, usually much less bandwidth has to be used for maintaining the route tables at each node. However, due to this fact, nodes are experiencing a longer latency before data can actually be transmitted. Therefore, protocols within this class are assumed to perform better in very dynamic MANETs. The following sections describe two protocols within this class that have been getting a lot of attention within the IETF recently. Ad hoc On-Demand Distance Vector Routing (AODV) has been published as experimental RFC [PBRD03] and the Dynamic Source Routing Protocol (DSR), which currently has the status of an Internet draft [JMH03] is expected to be published soon as experimental RFC. As can be seen in section 3, reactive routing algorithms have advantages over its proactive counterparts especially in highly mobile scenarios. Since such scenarios are in the focus of this thesis, reactive protocols will be explained in some more detail in the following sections using as example the two currently most prominent candidates - AODV and DSR.

2.3.2.2.1. AODV - Ad hoc On-Demand Distance Vector Routing AODV was designed especially for MANETs based on the experience with the Destination-Sequenced Distance-Vector (DSDV) routing algorithm [PB94]. It offers quick adaptation to changes in network topology, low processing and memory overhead and low traffic overhead. As shown below in more detail, destination sequence numbers are used to ensure loop freedom. Using this technique, problems such as counting to infinity of classical distance vector problems are avoided and quick convergence in the case of topology changes is ensured. In the case of link failure, AODV actively notifies the affected set of nodes and the route can be invalidated. The basic operation of AODV can be divided into the functions neighbor monitoring, discovery and route maintenance:

- **Neighbor Monitoring:** A node may offer basic connectivity information by broadcasting HELLO packets. These packets should only be sent out if the node is part of an active route. Every hello interval, a node checks whether it has sent a broadcast e.g. a RREQ within this interval. If not, it may broadcast a HELLO packet (which basically is a RREP with TTL=1 and the sending node's IP address as destination). Also, a node may determine connectivity by listening for packets from its set of neighbors. If a node is a forwarding node, i.e. it is participating in a currently active route, it should keep track of its continued connectivity to its active next hops. For information concerning this connectivity the node can use different link layer or network layer mechanisms. E.g. in the case of IEEE 802.11 networks, absence of a link layer ACK or failure to get a CTS after sending RTS after the maximum number of restransmission attempts indicates loss of the link to the active next hop.

- **Route Discovery:** Route discovery is done purely on demand and follows a route request (RREQ)/route reply (RREP) discovery cycle: when a source node S needs a route to a destination D, it broadcasts a RREQ. Any node N_i which has a route to that

destination (or D itself) unicasts a RREP back to the source node S. Each node stores up-to-date route information in its route table. To avoid routing loops and to eliminate stale routes, destination sequence numbers are used.
RREQ packets, created by the source node S if it does not find a current route in its table, contain the IP address IP_S of S and its current sequence number SEQ_S as well as the destinations IP address IP_D and its last known sequence number SEQ_D. Also included within the RREQ packet is a broadcast ID ID_i. Such a broadcast ID is associated with any node N_i in the network and is incremented any time N_i sends out a RREQ. Using ID_i and IP_i, RREQ packets can be identified uniquely.
After the initial RREQ is sent out by S, a timer is set by S to wait for a reply. An intermediate node N_i who receives this RREQ first checks whether it has already seen this packet. This can be done as any node N_i maintains a record of IP_S and corresponding values of ID_S from the packets it has seen during a specified time interval in the past. If the RREQ has been seen before, it is silently discarded. Otherwise, the node processes the packet as follows: first, a reverse route entry for the source node S is added to the route table of N_i. Using this entry which is of format: [IP_S, SEQ_S, number of hops to S, IP_j where N_j is the node from which the RREQ packet was received] N_i can forward a RREP packet to S in the case one is received later.
To be allowed to respond to a RREQ, a node must have an unexpired entry for D in his route table. Also, the entry for SEQ_D stored in $D's$ route table must be at least as great as that indicated in the RREQ. If N_i does not have a route to D, it rebroadcasts the RREQ packet with an incremented hop count to its neighbors until it reaches its destination D or any other intermediate node can answer the RREQ. In case the RREQ is lost, a number of retries can be configured in AODV before the destination unreachable notification is sent to the application.
As an optimization for large networks, the expanding ring search mechanism has been defined to avoid excessive flooding. The source node starts broadcasting RREQ with an initially short TTL value. If it does not receive a corresponding RREP within a certain time interval, it rebroadcasts the RREQ with an incremented TTL value. The rebroadcasts with incrementing TTL values are continued either until an RREP is received or the TTL threshold is reached. In the latter case, the source node tries to send conventional RREQ packets across the entire network.

- **Route Maintenance:** As AODV is a reactive protocol, only routes that are needed by source nodes are set up and maintained. Therefore, existing routes that are not used any more are aged out of the system. Route maintenance is also needed if active routes fail due to mobility or other reasons. In case the route fails because the source node S moves away, S can simply start a new RREQ discovery cycle. In case of route failure due to events (e.g. mobility, node failure) at intermediate nodes, Route Error Messages are propagated to make this event known to the effected nodes. Figure 2.4 shows an example of route maintenance due to node mobility.

Figure 2.2 shows an example network topology for explaining AODV. Connections in figure 2.2 represent bidirectional links, all existing connections are drawn. Based on the network topology in figure 2.2, a simple RREQ/RREP discovery cycle is shown as message sequence chart (MSC) in figure 2.3:

(1) RREQ: S does not have a route to D and broadcasts a RREQ:

$RREQ[IP_S, SEQ_S, IP_D, SEQ_D, ID_S, hopcount = 0]$

(2) RREQ: N_1 receives the RREQ and makes a reverse route entry for S:
Reverse Route Entry: $[dest = S, nexthop = S, hopcount = 1]$
N_1 has no route to D, so it rebroadcasts the RREQ with incremented hopcount:
$RREQ[IP_S, SEQ_S, IP_D, SEQ_D, ID_{N_1}, hopcount = 1]$

(3) RREQ: N_2 receives the RREQ. As it has no route to S, it makes a reverse route entry and rebroadcasts the RREQ.

(4) RREQ: As direct neighbor of N_1, also N_3 receives the RREQ.

(5) RREP: N_3 makes a reverse route entry for S:
Reverse Route Entry: $[dest = S, nexthop = N_1, hopcount = 2]$
N_3 as a direct neighbor to D knows the route to D and the destination sequence number SEQ'_D stored in the route table of N_3 is greater or equal than the destination sequence number SEQ_D received within the RREQ from N_1. N_3 therefore creates a RREP packet:
$RREP[IP_D, SEQ_D, IP_S, hopcount = 3]$
and sends it as unicast on the way back to S.

(6) RREP: N_1 receives the RREP from N_3 and forwards it via the information in its reverse route entry directly to node S.

(7) RREP: In the meantime, N_4 has processed the RREQ packet and found a valid route to D in his route table. N_4 also creates a RREP packet and unicasts it back to S.

(8) RREP: The RREP packet from N_4 on its way to S arrives at the intermediate node N_1. N_1 already received a RREP corresponding to the same previous RREQ a short wile ago. Therefore, N_1 checks whether the new packet with the alternative route to D contains a greater destination sequence number SEQ_D or has a smaller hop count than the previous one. As neither is the case, the RREP packet is discarded by N_1. This mechanism is used to keep the level of RREQ packets on the way to the requesting node S low while still having the most up-to-date and quickest routing information.

2.3.2.2.2. DSR - Dynamic Source Routing Protocol As the protocols before, the Dynamic Source Routing Protocol (DSR) has been designed especially for MANETs. It is composed of route discovery and route maintenance mechanisms that are described below. DSR is a reactive protocol, so like AODV the protocol scales it routing overhead dynamically according to routes currently in use. The probably most important difference to AODV is the utilization of source routes in DSR. This allows trivial loop freedom and avoids the need for maintaining routing information in intermediate nodes that forward packets along an active route.
DSR also works in the presence of unidirectional links and supports asymmetric routes. These features increase performance especially in cases where the wireless link may not work equally well in both directions as a result of differing propagation patterns or interference.
Some MAC protocols, however, provide bidirectional links only. In this case, various optimizations like the route reversal optimization of DSR can be used. DSR does not require any

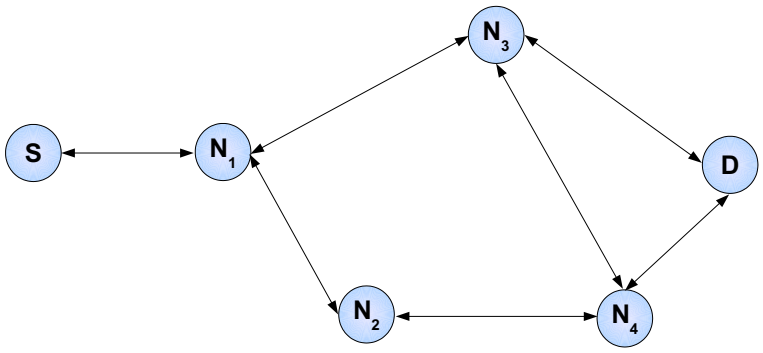


Figure 2.2.: Example Network Topology

periodic packets at any level within the network. This means it works without periodic routing advertisement, link status sensing or neighbor detection packets and does also not rely on underlying network protocols to perform these functions. Therefore, the main functionality of DSR lies in its route discovery and route maintenance mechanisms that are explained below:

- **Route Discovery:** Route discovery is only performed if a source node S wants to send one or more packets to a destination node D and does not find an appropriate entry in its route cache. The packets contain a source route giving the sequence of hops that the packet should follow. In the case of a route request (RREQ) packet being sent out by S, this field is initialized as an empty list. In addition to the source route, a DSR RREQ packet also contains the address of the source node IP_S, the address of the destination node IP_D and a unique request ID ID_S determined by the initiator S of the route discovery. This packet is locally broadcast by S to all its neighbors (i.e. all nodes within the radio range of S). When another node receives a RREQ packet, it returns a route reply (RREP) message to the source node S if it is the target of the route discovery. After reception of this RREP message, S can start to send its data on the newly discovered route to D. If the intermediate node N_i receiving the RREQ packet is not the destination of the RREQ, it checks whether it has already seen this packet or whether its address IP_{N_i} is already included within the route record of this packet. If either is the case, N_i discards the RREQ packet. Otherwise, N_i appends its own address IP_{N_i} to the route record of the RREQ packet and broadcasts it locally to all its neighbors.
  If the destination node D finally receives the RREQ packet, it first checks its route cache whether it already has an entry for a route to S. If so, it forwards the RREP packet along this route to the requesting node S. If D has no entry for S in its route cache, D may perform its own RREQ for S. In this case it is important to avoid infinite recursive RREQs, therefore, the original RREQ originated from S is piggybacked on the RREP message. For MAC protocols that support only bidirectional links, the route reversal optimization of DSR can be used, this means the RREP packet is sent back via the same route the RREQ packet has used.

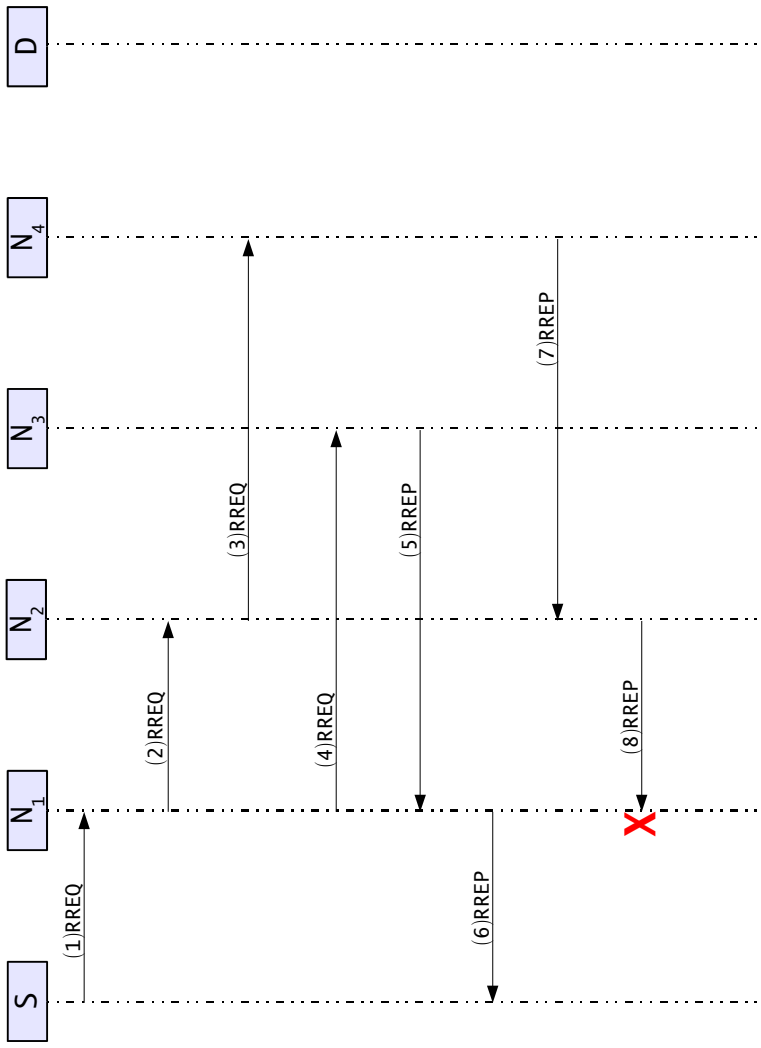
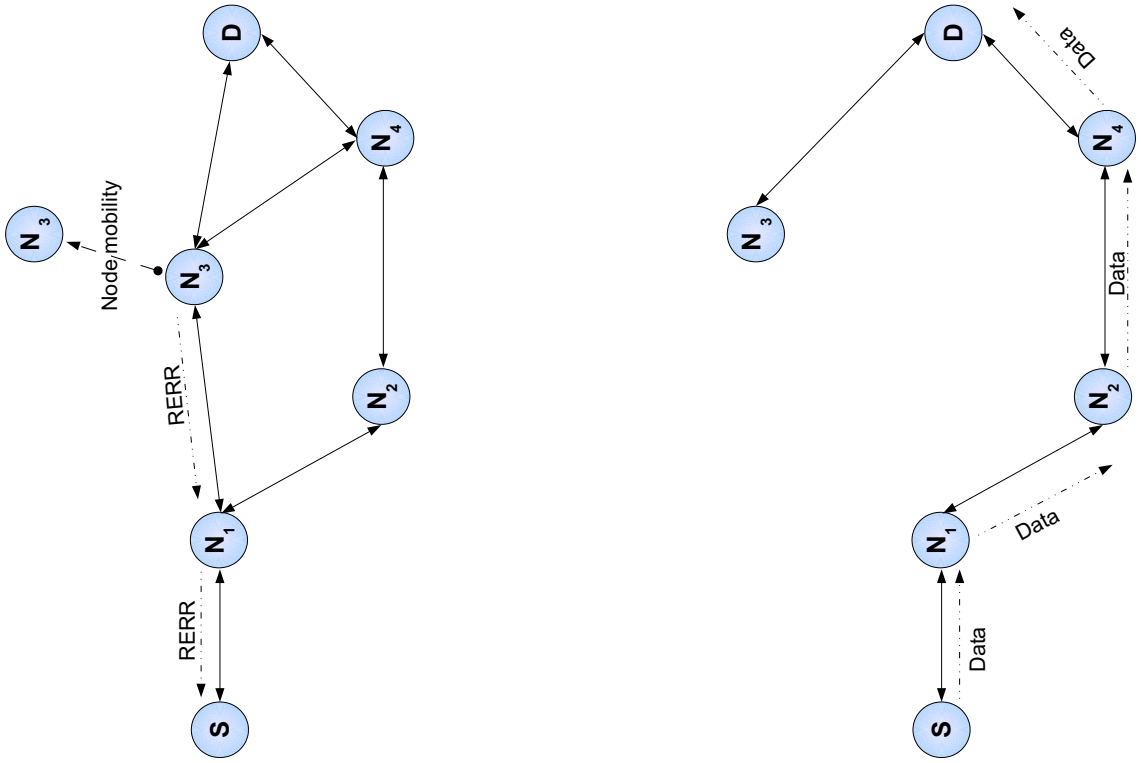


Figure 2.3.: AODV: Route Request and Route Reply (assumes that each node knows its immediate neighbors)



(a) Route Maintenance due to Node Mobility

(b) New, Re-Established Route

Figure 2.4.: The AODV Route Error (RERR) Mechanism

- **Route Maintenance:** If the route from S to D is not longer available, a DSR route error (RERR) message is sent to the source node S. The route availability is checked as follows: Each node along a route is responsible to confirm that the packet has been received by the next hop along the source route. Considering the topology from figure 2.2, if node S has sent out a packet to D using the source route via nodes N_1, N_2 and N_4 to D, node S is responsible to confirm receipt of the packet at node N_1 and node N_1 is responsible to confirm receipt of the packet at node N_2. This chain of responsibility continues to node N_4 which is responsible to confirm receipt of the packet at the destination node D. The receipt of a packet can be checked using MAC specific mechanisms. If such mechanisms are not available or accessible, passive acknowledgments can be used. This means e.g. node N_1 can listen whether node N_2 forwards the packet in question. However, also DSR specifies its own acknowledgment mechanism that can be used when necessary. A transmitting node N_1 can set a specific bit in the header of the transmitted packets, requesting an ACK to be returned by the receiving node N_2. This ACK can either be returned directly or, in the case of unidirectional links, via a different route from N_2 to N_1. After the maximum number of retransmits of a packet, the link is considered lost and a RERR, specifying the bad link, is sent back to S. S then removes the corresponding route from its route cache. If available, S can then use an alternative route from its route cache to reach D, if no such route is available, a new RREQ has to be sent out.

2.3.2.3. Cluster Based Routing

Besides the work on different routing algorithms, work is going on to enforce well known network structures in MANETs. Most of this work is going on under the keyword *cluster-based networks*. The cluster structure can, among other things, also be used to create hybrid routing algorithms that combine both, the reactive and the proactive approach for routing within and between clusters. An overview of the basic ideas of cluster-based networks is given in this section.

In MANETs, cluster based approaches are beneficial mainly because of the following reasons:

- Channel contention is reduced because of the management of wireless transmissions among multiple nodes.
- Routing traffic overhead is reduced because of the abstraction of network state information.
- Routing backbones can be created to reduce the network diameter.

Especially for the reduction of interference, the link-clustered architecture [BE81, EWB87] has been proposed. Transmissions in neighboring clusters can be isolated using different spreading codes in each such cluster. In the link-clustered architecture, the nodes organize themselves into interconnected clusters whose union of members is all network nodes. The clusters are composed of a clusterhead, one or more gateways and ordinary nodes as depicted in figure 2.5. Partially overlapping clusters are allowed in this architecture.
The following steps are needed to establish a link-clustered control structure:

1. Each node has to perform a discovery of neighbors to which it has direct bidirectional connectivity.

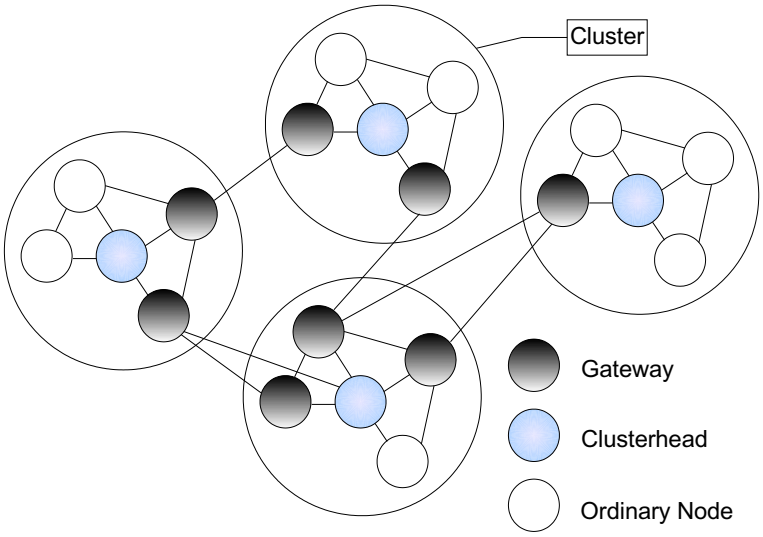


Figure 2.5.: Link-Cluster Architecture

2. Election of a clusterhead and the formation of clusters (e.g. via identified clustering as in [BE81, EWB87] or via connectivity-based clustering as in [GT95, Par94]).

3. Agreement on gateway nodes between clusters. This can be done e.g. by assigning gateway functionality to all potential candidate nodes[2] which supports connectivity when some connections are lost.

A routing backbone consisting of clusterheads is created using the link-clustered architecture, however, many protocols using this architecture (e.g. [GT95, LG97]) do not use this structure for routing. This is mainly due to the fact that in this case clusterheads tend to become congested and in addition are a single point of failure for communication across its cluster. Thus, the link-clustered architecture is primarily used to define regions for transmission management.

However, strong motivation exists to overcome these disadvantages and also use clusters for backbone formation to reduce route length and, therefore, communication delay. The problem of interference (when a node increases its transmission power to reach a more distant node) can be solved by isolating different types of communication (e.g. by frequency). Thus, one frequency can be used for intercluster communication and another frequency can be used for distant communication along the backbone. The problem of clusterheads as single point of failure has been addressed in various projects, among them the Near-Term Digital Radio (NTDR) Network [Jav97]. The basic assumption for this mobile tactical network included frequent node movement and node outages, therefore it has been optimized for robustness in these scenarios. Each node within NTDR keeps track of its neighbors using broadcst beacons. Also, all nodes are capable of quickly becoming clusterheads if needed, therefore, the routing

[2]Candidate Nodes are determined by the topology of the network.

backbone and access to it are maintained in most situations.

2.3.2.4. Hybrid Routing Algorithms

Hybrid routing algorithms use both, proactive and reactive protocols for different purposes. The motivation for this strategy is the attempt to avoid the long delay and excessive control traffic[3] of reactive protocols and at the same time avoid the creation of unnecessary routing information of proactive protocols[4]. The basic ideas of the Zone Routing Protocol (ZRP, [HP98, PH99]), which is an example for this protocol class, are explained in the following paragraph.

2.3.2.4.1. The Zone Routing Protocol (ZRP) ZRP is based on the concept of routing zones. Depending on these zones, either the Intrazone Routing Protocol (IARP) or the Interzone Routing Protocol (IERP) is used. The routing zone of a node S (the name of the local neighborhood in which IARP is used) is defined as the set of nodes with a minimum distance $d \leq r_z$ from S, where r_z is called the zone radius. Figure 2.6 shows the routing zone for node S with a zone radius r_z of 2. Nodes from N_1 to N_4 are within the routing zone, nodes from N_5 to N_9 are called peripheral nodes (the minimum distance to S is equal to the zone radius r_z) and nodes from N_{10} to N_{12} are outside the routing zone. Nodes with direct (single hop) connection to S are called neighbors of S. Every node has its own routing zone, so the zones usually overlap considerably. Each node propagates routing information to all nodes within its routing zone via a proactive scheme (IARP). Basic link-state algorithms can be used for IARP with link-state updates only sent within the routing zone radius.
If the desired destination node D is not within the routing zone of S, the reactive Interzone Routing Protocol (IERP) is used. IERP uses a strategy called bordercasting for less traffic overhead than flooding. Route discovery is done as follows: if the destination D is within the routing zone of S, the route is known and route discovery can be terminated. If this is not the case, S sends a route request to all its peripheral nodes. The peripheral nodes then execute the same algorithm until the destination D has been found and a route reply is sent back to S via source routing. If multiple route replies arrive at S, the best route can be determined according to its quality (e.g. hop count can be used as simple metric). As said before, routing zones usually overlap heavily. The simple bordercasting explained before can therefore result in worse traffic overhead than flooding algorithms. To avoid this, the early termination mechanism has been defined. It attempts to avoid redundant route requests, i.e. route requests that arrive in a previously queried routing zone. Such route requests are detected and not forwarded any more.

2.3.2.5. Position-aware Routing Algorithms

In contrast to proactive, reactive and hybrid protocols described before, position-aware routing protocols use knowledge about node positions instead or in addition to knowledge based on the network topology only. For routing decisions, a node usually has to know its own position, the position of the destination node and the position of its direct (single-hop) neighbors. Since

[3] A reactive global search requires significant control traffic.
[4] In MANETs changes of the network topology are often more frequent than RREQs.

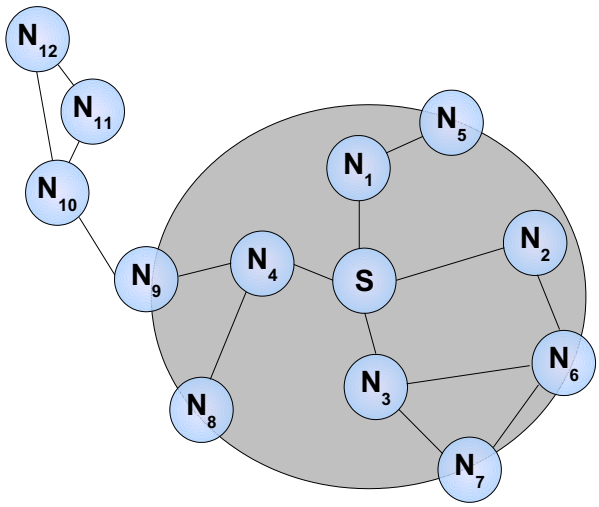


Figure 2.6.: ZRP: Routing Zone with Radius 2

position-aware routing algorithms do not have to maintain routes, neither proactively nor reactively, they have certain performance advantages in highly dynamic MANETs. Another advantage of position-aware routing algorithms is its capability to geocast certain information, i.e. to address all nodes within a specified geographical area. However, protocols from this class depend on additional hardware needed within the nodes to determine its position[5]. With the need to know also the position of the destination node, a network-wide location service becomes inevitable. Thus, position-aware routing algorithms can be seen as combination of a specific location service with a forwarding strategy. The following paragraphs give an overview about different strategies for location services and forwarding strategies. A more extensive summary of position-aware routing algorithms for MANETs can be found in [MWH01].

2.3.2.5.1. Location Services (LS) Depending on whether some specific nodes or all nodes within the network provide the location service and whether information concerning some nodes or information concerning all nodes within the network is stored, location services can be classified into the following categories: some-for-some, some-for-all, all-for-some and all-for-all. Classic cellular networks operate with dedicated position servers with well known addresses. This approach, however, is not suitable for MANETs because it would lead to a chicken-egg-problem: the positions of all nodes are stored within a dedicated server. To query this server, it has to be contacted using its current position ...
When propagating position information it has to be taken into account that the accuracy of this information can be reduced when being forwarded to remote areas of the network. The reason for this is the distance effect illustrated in figure 2.7: Node N_1 and node N_2 are moving

[5] Currently, mostly GPS is used for location information. GALILEO is expected to be an attractive alternative in the near future.

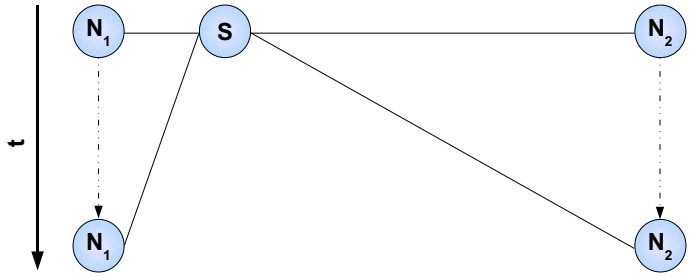


Figure 2.7.: Position-Aware Routing: Distance Effect

in the same direction at the same speed. For node S, the change in direction is bigger for node N_1 because its distance to S is less than the distance between S and N_2.
Below, the following location services are described briefly: *Distance Routing Effect Algorithm for Mobility* (DREAM, [BCSW98]), Quorum-Based Location Service [HL99] and Grid Location Service [LJC+00]. Only the basic mode of operation is shown as in-depth knowledge of location services is not a necessary prerequisite for understanding this thesis. Performance issues of position-aware routing algorithms, especially the communication complexity for location service and forwarding mechanisms are discussed in chapter 3.

DREAM: The Distance Routing Effect Algorithm for Mobility uses the all-for-all approach, this means each node maintains a position database that stores position information about each other node that is part of the network. Positions in the database are combined with timestamps to ensure better accuracy. Each node regularly floods packets to update its position information stored in other nodes. The flooding can either be controlled via the frequency for the position updates or via a value indicating how far a position information travels before it is discarded.

Quorum-Based LS: The Quorum-Based Location Service works similar to quorum systems known from database information replication. The nodes within a network are divided into quorums so that its intersection is non-empty. Information updates are sent to a quorum of available nodes, queries are sent to a potentially different subset of nodes. Quorum systems have to deal with the following tradeoff: the bigger the quorums, the higher the cost for updates and queries. However, with bigger quorums also more nodes are found in the intersections of quorums, resulting in higher resilience. Regarding the above classification, quorum systems can operate in a all-for-all, all-for-some or some-for-some configuration.

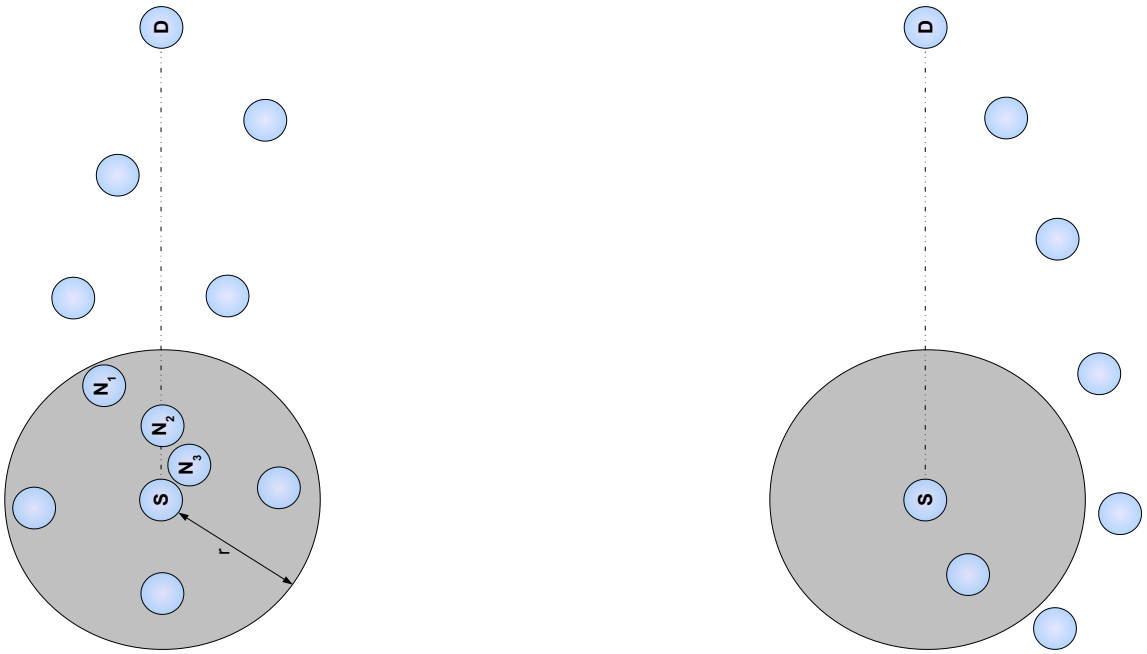
Grid LS: The Grid Location Service divides the area into a hierarchy of squares. Each n-order square contains four $(n-1)$ order squares and each node maintains a table of all other nodes within the local first-order square. Periodic position broadcasts in the area of first-order-squares are used. With the additional use of so called near-node IDs, i.e. node IDs where position information is stored or from where position information is queried respectively, the Grid Location Service implements an all-for-some approach. The density of position information is decreasing logarithmically with the distance to the node in question.

2.3.2.5.2. Forwarding Strategies After the necessary node positions have been determined via the location service, appropriate forwarding strategies have to be applied to forward data packets to its destination. Basically, arbitrary location services can be combined with arbitrary forwarding strategies as long as the location service delivers the required input for the forwarding strategy[6]. Below, some common forwarding strategies are introduced, namely greedy packet forwarding, restricted directional flooding and hierarchical approaches from the Terminodes and Grid projects.

Greedy Packet Forwarding This forwarding strategy in its simple form works as follows: the approximate position of the destination node (known from the underlying location service) is included in the packet header. When an intermediate node receives the packet, it forwards it to the next node in the direction of the recipient until the destination is reached.

Figure 2.8(a) shows an example network topology that is used for the following explanations about greedy packet forwarding. Node S denotes the source node, D the destination node. The circle with radius r depicts the transmission range of node S. One straight-forward strategy is to select the node that makes the most progress towards the destination node D as next hop. In figure 2.8(a), node N_1 is selected according to this strategy also known as MFR (most forward within r). Another greedy strategy called NFP (nearest with forward progress) selects the nearest node to S that makes any progress in the desired direction as next hop towards D. While MFR is a good strategy for nodes with fixed transmission range, NFP is better suited for nodes with variable range as it reduces interference and therefore possible collisions. Using NFP, node N_3 is selected as next hop towards D. Compass routing is another common greedy strategy, here the node that is closest to the straight line between S and D is selected as next hop. In our example network topology, node N_2 is selected according to the compass routing strategy. Random selection of an arbitrary node closer to the destination than the previous one is another greedy strategy that minimizes requirements on the accuracy of the position information available.
All those strategies, as simple and appealing they may appear, have one common disadvantage - the problem of local minima. This means there are possible node constellations with existing paths from S to D but the greedy strategies are unable to find them. One such situation is shown in figure 2.8(b): Here clearly a path from S to D exists, however, greedy algorithms can not find it because this would require them to temporarily increase, not decrease the distance to the destination.

[6]This usually is the case as most systems work with simple position information only.

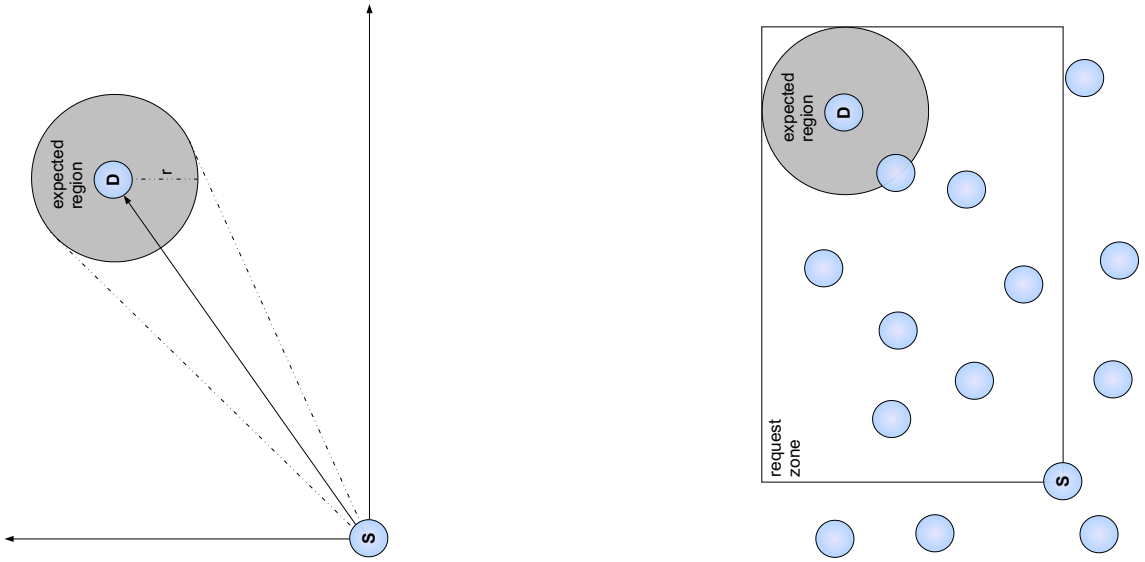


(a) Example Network Topology (b) Local Minimum

Figure 2.8.: Position-Aware Routing: Greedy Packet Forwarding

Mechanisms to overcome the problem of local minima have been published as face-2 algorithm [BMSU99] and Greedy Perimeter Stateless Routing Protocol (GPSR) [KK00]. Both algorithms are based on planar graph traversal - a packet enters recovery mode when a local minimum is reached and continues with the greedy strategy when it reaches a node closer to the destination compared to the point where it entered recovery mode.

Restricted Directional Flooding Using restricted directional flooding, an expected region for the destination node D is calculated and nodes are flooded towards this region instead of being flooded throughout the entire network. Different restricted directional flooding protocols follow a slightly different approach here - as an example the forwarding strategies of DREAM [BCSW98] and Location Aided Routing (LAR), [KV00] are explained in this section. Request zone and expected region of DREAM can be seen in figure 2.9(a), the same zones for LAR are depicted in figure 2.9(b).
In DREAM, the source node S forwards its packets to all nodes that lie in the direction of the expected region of D. Figure 2.9(a) shows the expected zone of D as seen from S. Outdated information about a nodes position can be taken into account via the maximum speed of a node in the ad hoc network, v_{max}. The expected zones radius is then set to $(t_1 - t_0)v_{max}$ where t_1 is the current time and t_0 the timestamp of the latest known position entry for D. This process is repeated until D is found. The intermediate nodes on the way from S to D may have newer information about the position of D, so each forwarding step involves a re-calculation of $D's$ expected zone based on potentially newer position information.
LAR on the opposite does not specify its own position based forwarding mechanism, it instead provides support in the RREQ phase of a reactive routing algorithm if position information is



(a) DREAM (b) LAR

Figure 2.9.: Position-Aware Routing: Expected Region

available. If no position information is available, the reactive routing algorithm is used with the available topology information as a fallback solution. The basic approach of LAR is to prevent the network-wide flooding done by reactive routing algorithms in the RREQ phase. If position information is available, it calculates expected regions and request zones similar to DREAM. RREQ flooding is then restricted to a request zone which typically also includes the expected zone. One type of request zone defined in LAR is a rectangular geographic region that is shown in figure 2.9(b). LAR defines a second type of request zone via the estimated distance to the destination node D. A RREQ packet is forwarded by an intermediate only if its distance to D is not bigger than the distance of the previous node plus a specified system parameter.

Hierarchical, Position-Aware Routing As known from the Internet, hierarchical structures are often introduced to handle large numbers of nodes and provide higher performance. Two well known approaches for hierarchical, position-aware routing in MANETs have been developed in the Terminodes project [BBC+01] and in the Grid project [gri04]. The terminodes approach uses a two level hierarchy - packets with a relatively close destination are routed with a proactive distance vector algorithm. Packets with remote destination nodes are routed using a position-based greedy strategy. When the long-distance packet arrives in the area around the destination node, the local proactive routing algorithm is used for the remaining hops to the destination.
The grid project uses location proxies for routing. The overall approach is similar to the terminodes protocol: a proactive distance vector routing protocol is used to forward local packets, long-distance packets are routed according to a position-aware routing scheme. One of the differences between the terminodes and the grid approach is that grid does not require all nodes to know its own position (i.e. also nodes without positioning hardware can partici-

pate in the network). To address a node without known position, the packets are forwarded to a position-aware proxy in the area and sent from there to the final recipient via a proactive distance vector protocol. Both protocols, terminodes and grid use similar mechanisms to avoid local minima for long-distance greedy forwarding. If no route to a destination is found (i.e. the packets may have found a local minimum), position-based source routing is used. This means certain positions that have to be traversed on the way to the destination node are specified either directly in the packet (the terminodes approach) or are selected in the case an error occurs by the sending node.

2.4. Ad Hoc - Internet Gateways

Gateways between the Ad Hoc network and the Internet are usually considered to be part of 4G networks. It is, given the growing number of wireless hotspots[7], sensible to assume Ad Hoc networks as technology growing around and extending the range of such hotspots instead of completely isolated Ad Hoc networks.[8]
Ad Hoc - Internet Gateways are used as a basis for the security framework presented in chapter 5. Despite its importance for the architecture, quite weak assumptions are made concerning its availability. This is due to performance and practicability reasons explained in more detail in chapter 3. Some, not necessarily all mobile nodes within an Ad Hoc network, have to be able to make contact to Ad Hoc - Internet Gateways at least temporarily.[9]

This section shows the required functionality of Ad Hoc - Internet Gateways and one concept for its realization based on work presented in [XB02]. Also, a prototypical implementation of this system that has been built at the LKN [dVSX03] is presented briefly. For the conceptual work, the view is restricted to IPv6 [DH98].

2.4.1. Required Functionality

Figure 2.10 shows a typical scenario: multiple mobile hosts forming an Ad Hoc network, connected to the Internet via several gateways, GW1 - GW3. A standard Ad Hoc routing protocol like AODV or DSR is assumed for the connectivity between the mobile nodes and between mobile nodes and the gateways. From the point of view of the mobile nodes, GW1 - GW3 act as access routers to the Internet.
One of the most important tasks for the gateways is the necessary protocol conversion between Ad Hoc nodes and Internet hosts. The protocol stacks of Ad Hoc nodes, gateway and Internet hosts are shown in Figure 2.11. Other important issues for gateway operation are gateway discovery, address autoconfiguration, and routing and addressing in this environment.

[7] According to [BWC03] and [Sie04], the number of WLAN hotspots in Europe has been growing from 1867 by the end of 2002 to 7243 by the end of 2003. The same reports estimate that a total of 71000 WLAN hotspots existed worldwide by the end of 2003.
[8] This can be expected at least for most commercial applications.
[9] Technically, alternative modes of operation are possible for the security framework presented in chapter 5 that do not require Ad Hoc - Internet Gateways but instead use alternative wireless interfaces. Due to space constraints, this thesis is focusing on the Gateway solution.

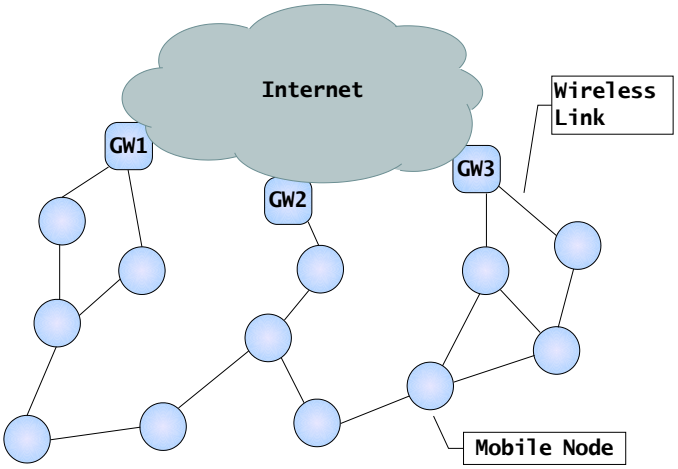


Figure 2.10.: Ad Hoc - Internet Gateways: System Overview

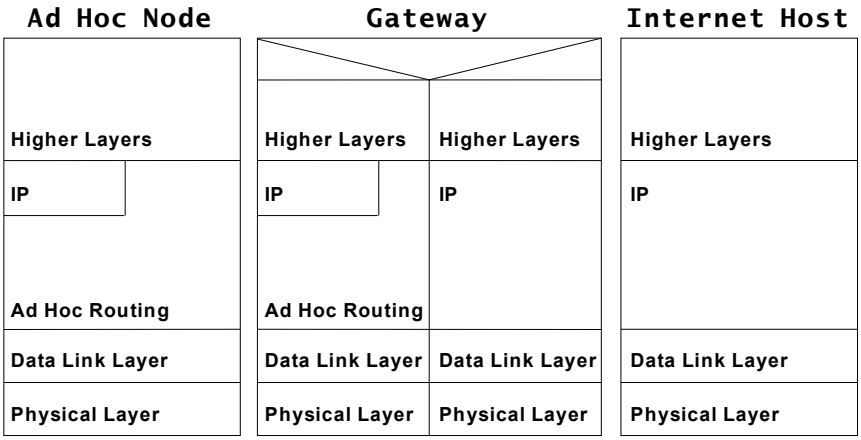


Figure 2.11.: Ad Hoc - Internet Gateways: Protocol Architecture

2.4.2. System Overview

- **Gateway Discovery:** Basically, two opposite methods can be used for gateway discovery: active discovery and passive discovery. Using active discovery, the mobile node sends out request messages that are broadcasted (usually with a hop limit to avoid excessive network load) in the Ad Hoc network. If available, also a special multicast address for Ad Hoc - Internet gateways can be used. On receipt of a request message, a gateway responds with a reply message containing its IP address that is unicast to the originator of the request message. Active discovery can be done periodically or event-triggered.
  The opposite method, passive discovery, requires gateways to send out advertisement messages periodically. These messages can be forwarded by the receiving nodes and extend the range of such gateways considerably into the Ad Hoc network. Additional mechanisms have to be used in the presence of unidirectional links, e.g. if the gateway has a higher radio range than the mobile nodes. In this case, the reception of an advertisement message does not necessarily imply an upstream connection to the gateway.
  For practical purposes, a combination of both methods to a hybrid gateway discovery mechanism seems promising. This method is presented in [XB02]. The advertisement messages sent by the gateways are stored in nearby mobile nodes for a certain time. If mobile nodes from outside this range actively send gateway request messages, any intermediate node that has the required information can reply directly, thus reducing signaling traffic. In all discovery mechanisms, several gateways may be detected by a mobile node. Standard selection mechanisms based on an appropriate metric (e.g. hop count, delay, etc.) can be used in this case.

- **Address Autoconfiguration:** Using IPv6, two different methods for address autoconfiguration can be used: stateful and stateless autoconfiguration. In [XB02], it is proposed to use a DHCP server implemented on the gateway for stateful autoconfiguration. From the gateway discovery process, the mobile nodes learn the DHCP servers IP address.
  Using stateless autoconfiguration [TN], the mobile nodes first have to select a link-local address. Having done this, IP-connectivity with neighboring nodes can be established in fixed networks. In the Ad Hoc scenario, however, the link-local prefix is problematic for multihop communication. Thus, a different reserved prefix has to be used to generate a temporary address. If necessary, a duplicate address detection mechanism can then be applied. With this temporary address, mobile nodes can start the gateway discovery procedure. After having learned the gateways prefix, a mobile node can use this prefix in combination with its EUI (equipment identifier) to generate a globally valid IP address.

- **Routing and Addressing:** Again, two different methods can be distinguished here, namely flat routing and hierarchical routing with care-of addresses.
  In the first case, the Ad Hoc network is regarded as a number of nodes without subnet partitioning. Routing between mobile nodes works similar to isolated Ad Hoc networks: a route request is sent out (if reactive routing is used) and answered by the destination node. If mobile IP is used, the gateway will also reply if it knows a path to the destination nodes home agent. If a route discovery is started for a host on the Internet, no node within the Ad Hoc network except the gateway node will answer with a route reply message.
  In the case of hierarchical routing, the network is logically separated into subnets. All

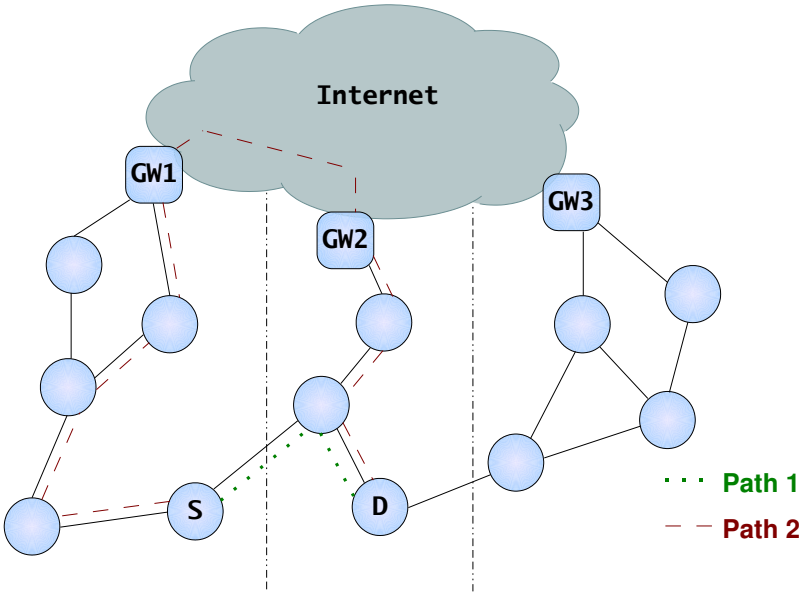


Figure 2.12.: Ad Hoc - Internet Gateways: Path Optimality

messages with a destination in a different subnet (this can be determined by comparing the address prefixes) have to be routed via the gateway. A good choice for addressing is to use hierarchical addresses also within the Ad Hoc network. Unique care-of addresses of the mobile nodes are a good choice because they already contain the gateways prefix. This leads to the main advantages and disadvantages of the two approaches: hierarchical routing usually causes slightly less signaling traffic as some of it is limited to single subnets. On the other hand, care-of addresses are necessary for all nodes to implement the hierarchical approach.

2.4.3. Path Optimality

Figure 2.12 illustrates the problem of path optimality. In densely populated Ad Hoc networks, direct links between nodes in different subnets are quite usual. If only hierarchical routing is used, possibly shorter links (Figure 2.12: Path 1) will not be considered and the packets will be routed via the gateway (Figure 2.12) in any case. [XB02] presents an approach to avoid this: a prefix cache. This means that mobile nodes store not only the prefix of its own subnet but also collect prefixes of neighboring subnets. Having this information, a mobile node S first checks whether the prefix of the destination node D can be found there. If so, S first attempts to find a path within the Ad Hoc network instead of directly asking the gateway.

2.4.4. Prototypical Implementation

A demonstrator for such Ad Hoc - Internet Gateways has been developed at the LKN and is described in [dVSX03]. It uses Linux computers with the following components:

- **IEEE 802.11b** Wireless LAN cards (Orinoco Silver).
- **HUT Mobile IP**[10], a free Mobile IP implementation.
- **iptables**[11],a packet filter.
- **Jpcap**[12] (interface library between the well known C packet capturing library libpcap and the JAVA world)
- **NIST AODV**[13] as reactive Ad Hoc routing alorithm.

2.5. SLP - The Service Location Protocol

Service discovery is an essential part of self organizing networks. Its utilization depends heavily on the envisioned application area of the network and is not only restricted to the application layer. Despite this fact, most published applications of service discovery refer to that layer: E.g. mobile users want to have access to various services when they visit a foreign network. Discovery and access to printers is an often cited example. However, also lower layers can benefit from service discovery mechanisms. Parts of the network configuration can be done using the same mechanisms, e.g. Internet gateways and other network services can be detected using service discovery. Another interesting application of service discovery mechanisms can be found in [Rei03]. Focusing on augmented reality systems, services and applications are dynamically created from modules found in the network. All these features, however, do not come without a drawback: several critical security issues arise if the protocols are used without additional precautions. The following sections describe the service discovery protocols SLP and Jini in more detail. Security issues related to these protocols will not be considered here, but will be discussed in the security-related chapter 4.

Other service discovery protocols include Universal Plug and Play (UPnP), Salutation and the Bluetooth Service Discovery Protocol (SDP). UPnP is developed by the UPnP consortium[14] with Microsoft as a major driver. Salutation is developed by the Salutation Consortium[15], including among others IBM, HP, Axis and Toshiba. It basically is a high-level solution and the transport layer is not specified. However, as Salutation Managers are contacted via well-defined APIs, this protocol could well act as the clamp that interconnects different service discovery protocols and guarantees interoperability in a heterogeneous environment. (SDP) is tailored to the needs of Bluetooth ad hoc communications and part of the Bluetooth[16] specification. Unlike the other approaches it is not intended to be a general-purpose service discovery protocol, therefore, it provides only very limited functionality. All these protocols can be used in combination up to a certain extent - e.g. a mapping of Salutation with Bluetooth SDP is described in [MP99].

[10] `http://www.cs.hut.fi/Research/Dynamics`
[11] `http://www.iptables.org`
[12] `http://www.jpcap.sourceforge.net`
[13] `http://w3.antd.nist.gov/wctg/aodv_kernel/`
[14] `http://www.upnp.org`
[15] `http://www.salutation.org`
[16] `http://www.bluetooth.org`

The Service Location Protocol is based on work done within the Srvloc working group and has been standardized by the IETF in RFC 2608. In this thesis, version 2 of SLP (SLPv2, [GPVD99]) is used. SLPv2 allows to dynamically discover the existence, location and configuration of network-connected services, devices and applications. Therefore, it assumes self-advertising services that supply its description and additional informations automatically. It is also possible to do service filtering, i.e. SLP supports mechanisms to select the service that matches the characteristics the mobile client is looking for.

2.5.1. Configuration Options

SLPv2 specifies three main components:

- **User Agent (UA):** Runs on the mobile terminal and is searching for specific services on behalf of the user.
- **Service Agent (SA):** Runs on the device offering a service. It advertises the location and characteristics of a service.
- **Directory Agent (DA):** Acts as central repository for services within a network. It collects service advertisements received from SAs and responds to queries from UAs.

Depending on network size and scalability requirements, SLPv2 can be configured in two different ways: The standard configuration (shown in figure 2.13) makes use of all three agents: UA, SA and DA. The service is registered by a SrvReg message from the SA to the DA that is acknowledged by a SrvAck message by the DA. If a UA now looks for a specific service, it sends a SrvReq message to the DA that is answered with a SrvRply message from the DA to the UA.
Several mechanisms exist to make the address of a DA known to UAs and SAs in the network. A straight-forward but not very flexible possibility is to write the address into a static configuration file. A better alternative is to use DHCP [DL99] for DA discovery. With an appropriate configuration (option 78, see [PG99]), UAs and SAs can request the DAs address from the DHCP server.
If none of these options seems appropriate, SLPv2 defines a third possibility for DA discovery: [GPVD99] defines the multicast address 239.255.255.253, port 427 for active DA discovery. If a SrvReq message with the content *service:directory-agent* is sent to this address, the DA will reply with a DAAdvert message.
UAs and DAs can also receive unsolicited DAAdvert messages. DAs send DAAdverts on startup and also periodically during operation. This interval is set to 3 hours in [GPVD99] but can be changed if necessary.

The minimal configuration of SLP (shown in figure 2.14) does not require the DA. This configuration option is especially attractive for smaller networks or a small number of agents as administrative overhead can be reduced. If no DA is available, the UAs discover services in the same way they discover a DA. The UAs will notice this situation when they receive a SAAdvert message as reply during the active DA discovery. All SAs listen to the multicast address 239.255.255.253:427 and unicast a SrvRply back to a requesting agent if they offer a service that meets the requirements in the SrvRqst message.

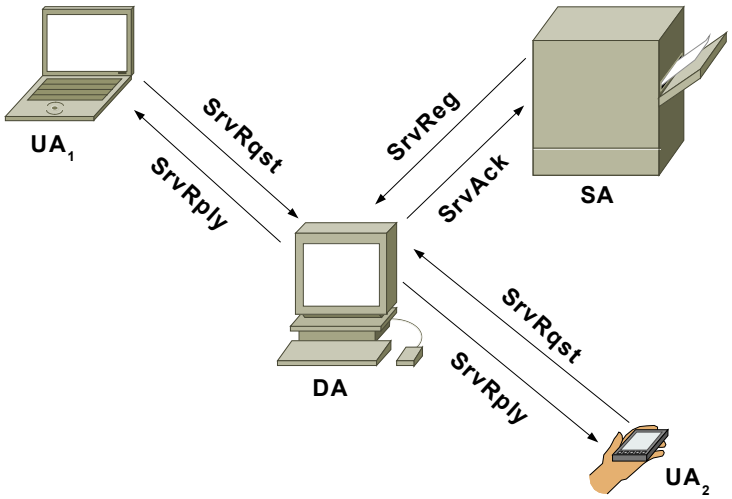


Figure 2.13.: SLPv2: Standard Configuration

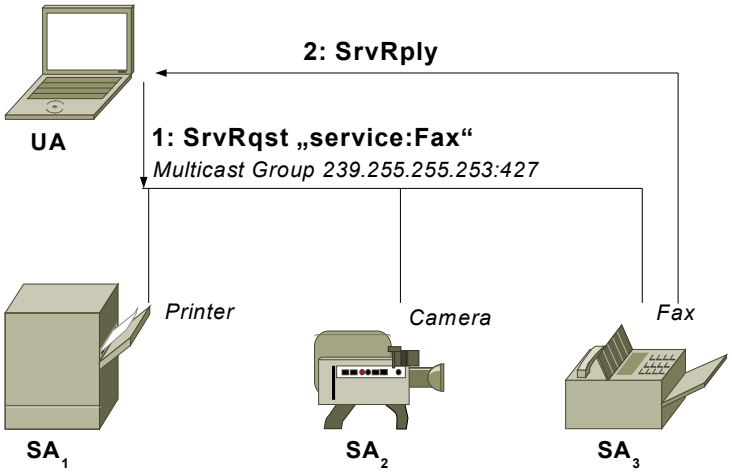


Figure 2.14.: SLPv2: Minimal Configuration

2.5.2. Service Description

Services in SLPv2 are described with its service type and its address. According to the specification in [GPVD99], this is done in the Uniform Resource Locator *service:URL*. E.g. *service:printer:lpr://129.187.222.100/lj4050_queue* defines a service with the abstract type printer and protocol type lpr. The URL is sent within Service Reply messages from DAs to the requesting UAs and contains all information the client needs to access the service. In the example above, this is the IP address of the computer hosting the printer as well as the queue name. By default, IANA[17] is considered responsible for the formulation of the service:URL and the service type template described below. However, it is easily possible to use a different naming authority if required. This is done by appending a string to the service type: e.g. *service:printer.ownauthority:lpr* indicates the responsibility of "ownauthority".

Service type templates contain additional information about a service. These templates are registered with a naming authority and associated with a specific service type. Its contents include specific attributes for a service and information about data type and characteristics of each attribute. Single attributes can be specified as optional or mandatory by its characteristics.

2.5.3. Additional Features

A very useful feature of SLP is the possibility to define scopes. These scopes are used to group services available in a network into administrative domains, hence increasing the scalability of the protocol. The administrator groups certain services within one scope, as a result only users that are configured within this scope are allowed to access these services. Every request and registration requires at least one scope. If no other scope applies, the scope "default" has to be used.
DA discovery using scopes is only successfull if the UA finds a DA that supports its (the UAs) scopes. If so, a unicast request is sent to the DA. If no appropriate DA can be found, the UA may multicasts its request or start a SA discovery. For service registration, a SA must register its service with all DAs in range that support at least one of the scopes configured for the service. Figure 2.15 shows a service registration using scopes and its results: A SA configured for scopes A and B discovers two DAs, DA1 and DA2. DA1 supports scope B while DA2 supports the scopes A, B and C. As some of the SAs scopes are supported by both DAs, it registers its service with DA1 and DA2. As a result, DA1 offers the service for scope B while DA2 offers the same service for scopes A and B.

2.5.4. TUM-SLP Implementation

In [Vet01] and [Ren00], an implementation of SLP including its security features has been done. This implementation has been made public and was distributed as open source software under the name TUM-SLP. Its main features include:

- **Security:** Implementation of authentication mechanisms specified in [GPVD99].

[17] Internet Assigned Numbers Authority, `http://www.iana.org`

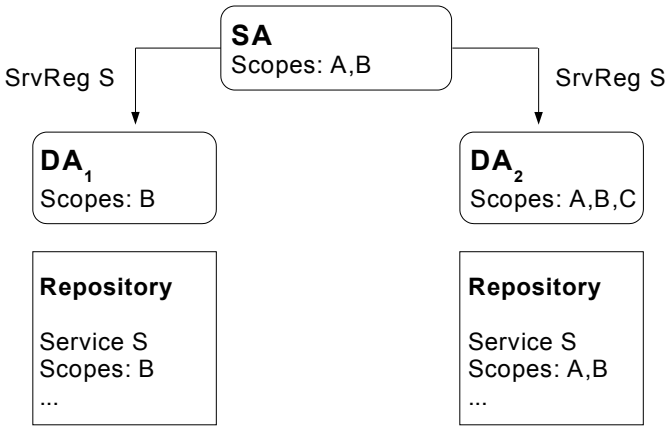


Figure 2.15.: SLPv2: Scopes

- **Service Lifetime:** Implementation of service lifetime expiration and other leasing functions. Among them are updates and removal of single attributes of registered services and periodical repetition of multicast DAAdverts.
- **Cross-Platform Interoperability:** TUM-SLP has been tested for interoperability with the MS-Windows based K&A SLP (Kempf and Associates SLP) which is another SLPv2 implementation available at `http://www.spybeam.org/kaslp/kaslp.html`

TUM-SLP provides an architecture that consists of one DA, one or more SAs and one or more UAs. All agents can be run either on the same or on different machines. Operation without a DA is not possible in TUM-SLP as the SAs do not listen do direct requests made by UAs.

2.5.5. Service Browser Implementation

In addition to TUM-SLP, also a service browser for better SLP usability has been implemented. The browser has been developed with the KDevelop IDE using C++ and the Qt-Libraries from Trolltech. It provides the following possibilities:

- Search by service requests, attribute requests and service type requests.
- Find all services within one scope.
- Browse through all services within one scope.
- Browse through all secure services registered at the DA.
- Get all attributes of a service.
- Perform periodic service discovery (period can be defined by the user).
- Show messages sent between the SLP entities after a request.

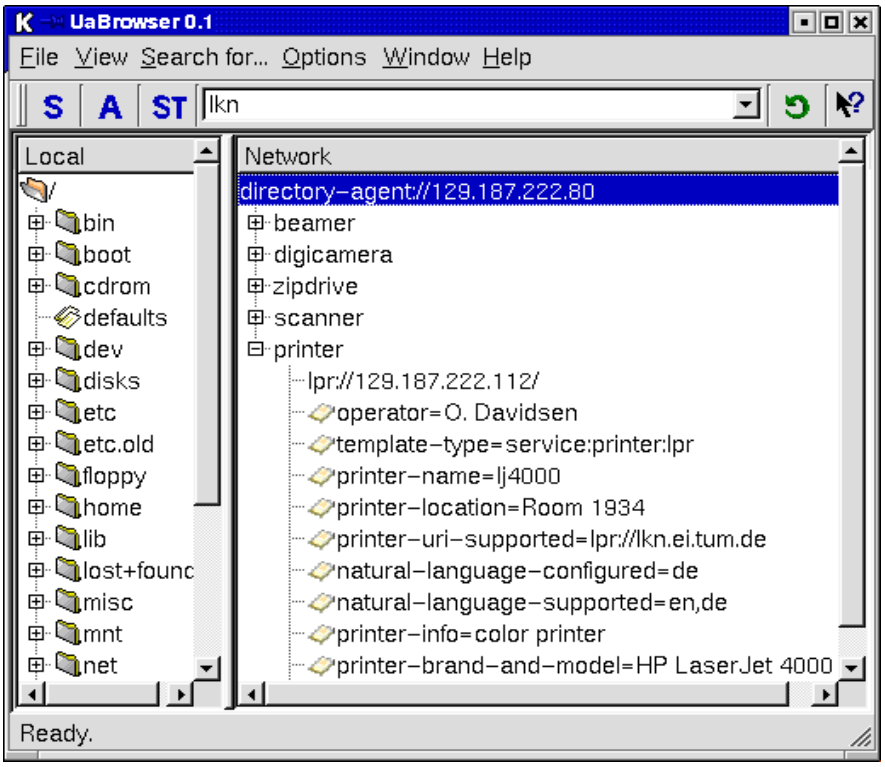


Figure 2.16.: SLPv2: Service Browser

Figure 2.16 and 2.17 show screenshots of different actions performed with the SLP service browser: Figure 2.16 show the browsing through all services offered at the LKN. Figure 2.17 shows an request for specific attributes of printers at the LKN and its result in a second window. A more detailed description of the service browser can be found in [JBS01].

2.6. Conclusion

This chapter presented possible applications of MANETs within 4G networks. As can be seen from these applications, there is no single standard MANET architecture. This leads to the conclusion that different MANET application scenarios require different protocols for optimal performance. Based on the protocol basics presented in this chapter, the performance of these protocols is discussed in chapter 3.

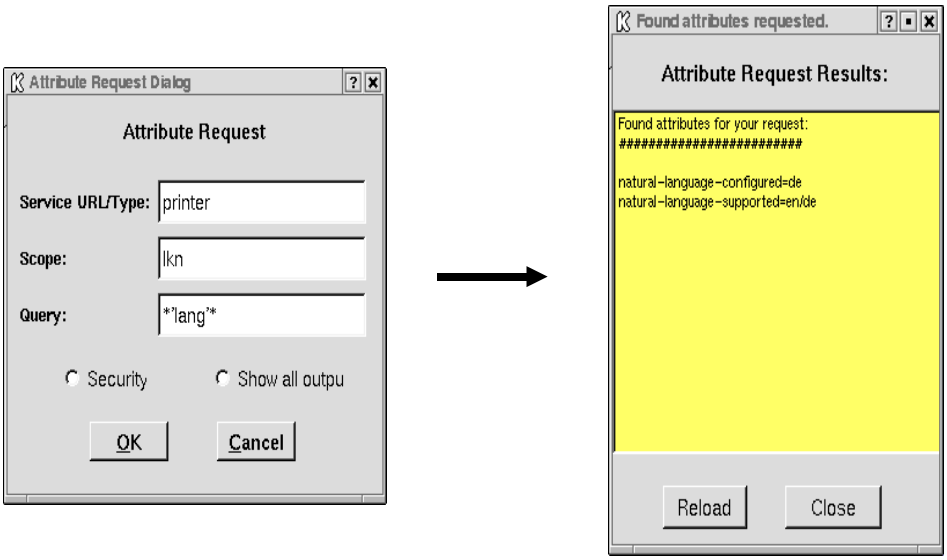


Figure 2.17.: SLPv2: Attribute Request

3. Ad Hoc Network Performance

The previous sections have shown mechanisms necessary for the realization of 4G networks with a focus on MANETs. As discussed in the previous section, protocols and concepts have been developed using different assumptions about the underlying network characteristics. As a result, the performance[1] of systems built on top of these protocols depends heavily on the environment. As will be seen in this chapter, MANETs do not scale well with respect to per node throughput. Therefore it is essential to keep the network performance in mind when designing security mechanisms.

This section shows the performance of basic MANET protocols in different environments without security mechanisms. These results are based on analytical approaches, simulations and real-world tests. The potential of manets is shown as well as the overhead caused by different classes of self-organizing routing algorithms. Finally, practical issues for the design of highly mobile MANETs with hardware and software available today (IEEE 802.11b and NIST AODV) are shown.
The organization of this chapter is done as follows: Section 3.1 presents analytical results concerning the performance of Ad-Hoc networks. This includes capacity estimations as well as the complexity of different routing classes. In section 3.2, a simulative approach is taken to estimate the performance of 802.11 in different ad hoc scenarios. This includes existing literature as well as own work. Finally, section 3.3 presents results gained in experiments with different MANET testbeds. This includes the performance of an existing AODV implementation, TCP/IP performance over IEEE 802.11b links as well as the performance of IEEE 802.11b in high-speed vehicular environments.

3.1. Analytical Results

While simulative and real-world performance are heavily depending on the underlying network technology, the analytical work presented here is independent. However, IEEE 802.11b network equipment has been used for real-world tests and analogical models have been used for the simulation studies presented in this chapter. Therefore, remarks concerning IEEE 802.11b can also be found within the analytical sections to highlight the relevance of its results and show possible network-related problems in such environments.

3.1.1. Unidirectional Links

Routing is known to be difficult in the presence of unidirectional links. However, wireless links in the real world often show this characteristic for different reasons. An interesting mathematical result, published in [Pra99], describes the effects of unidirectional links for

[1] Performance parameters considered within this thesis include route setup delay, signaling overhead, total throughput and per node throughput.

MANET routing protocols. Let n be the number of nodes in the MANET. It turns out that, for distance-vector approaches, adjacent nodes need to exchange $O(n^2)$ size messages to account for unidirectional links. This is a significant increase to the communication overhead $O(n)$ of existing routing algorithms (e.g. AODV, DSR) that assume bidirectional links only. It also turns out that, in some network configurations, two system-wide broadcasts are needed for a source node to find a path to its destination.
Possible reasons for unidirectional links can be a different radio transmission power (or receiver sensitivity) of network nodes. In addition to different radio equipment in heterogeneous networks, also homogeneous environments can experience unidirectional links as can be seen in the following paragraphs on the example of IEEE 802.11.
The scenarios regarded in this thesis assume similar transceivers in all nodes, i.e. the same or similar transmission range for all nodes. Also, power control is not a primary concern in highly mobile scenarios where usually vehicle-bound platforms are used. Thus, if an unidirectional link occurs in this scenario, the link might already be on the verge of failure and better is ignored anyway to avoid frequent re-routing operations. As a result, only bidirectional links are assumed for the rest of this thesis. Below, it is shown that significant numbers of unidirectional links can occur even in homogeneous environments and strategies are introduced to avoid this effect.

In [LNT02], the observation of an unexpected high amount of lost packets is described in an IEEE 802.11b based MANET testbed. This packet loss is caused by unidirectional links that have been frequently observed within so-called gray zones. While this specific effect is typical for IEEE 802.11b based MANETs, similar problems can occur also in different homogeneous environments. Thus, strategies presented below to avoid unidirectional links include also generic measures not depending on the utilization of 802.11 network equipment.
The main reasons why gray zones occur in 802.11b MANETs are the following:

- **Broadcast Transmission Rate:** HELLO messages used in routing protocols like e.g. AODV for neighbor discovery are sent as broadcast packets. Broadcast messages in 802.11b are always sent at a low bit rate while data transmissions are, based on the current signal-to-noise ratio, usually sent at higher bit rates up to 11 Mbit/s. As transmissions at a lower bitrate are more reliable and have a somewhat higher transmission range, HELLO packets are more likely to arrive at the receiver. Especially for links that have to overcome a high distance close to the maximum radio range, this results in a successfull neighbor discovery but lost data packets.

- **No ACK Messages:** As no acknowledgments (ACKs) are sent in response to broadcast packets, a successfull neighbor discovery using HELLO messages does not necessarily mean that data can also be sent back from the recently discovered neighbor to the inquiring node.

- **HELLO Packet Size:** The size of HELLO packets is relatively small compared to data packets. This implies a smaller probability for bit errors as well as a smaller probability for collission. Thus, HELLO packets are more likely to be received at the destination than data packets.

The characteristics described above result in the creation of gray zones, i.e. unidirectional links or links that are incorrectly marked as active while they are not capable of carrying

any data. Below, possible countermeasures are presented to avoid problematic unidirectional links in the MANET environment:

- **Control Packet SNR Threshold:** One possibility to improve neighbor sensing is the introduction of a signal-to-noise ratio (SNR) based threshold for control packets. If the value for this threshold is selected high enough, weak links that can only transport control packets with a low transmission rate are discarded.

- **Blacklists:** This countermeasure reactively eliminates unidirectional links. It is based on blacklists that have to be maintained in all nodes. If a node i detects a RREP (route reply) transmission failure, it inserts the next hop of the failed RREP message into the blacklist. When node i later receives a RREQ from a node that has an entry in its blacklist, it discards this RREQ message to avoid the possible selection of an unidirectional link for the new route. Figure 3.1 shows an example: All links except the one from node N_1 to node N_3 are bidirectional. If a RREQ from S to D is sent, N_3 receives this message from N_1. As the subsequent RREP message from N_3 to N_1 will fail, N_3 adds node N_1 to its blacklist. Thus, further RREQ messages sent to N_3 via N_1 are discarded. As a result, alternate paths can be discovered.
  This simple countermeasure works well in networks with few unidirectional links. If a high number of unidirectional links exist in the network, these links are added to the blacklists iteratively one at a time. This further increases the route set-up times that already are quite long for some routing algorithms. Another difficulty of this mechanism is the selection of an appropriate timespan when nodes are removed from the blacklists again. The selection of a static value may be close to impossible in highly mobile scenarios, however, a selection of values derived from the networks dynamicity (see section 2.3.1.2) may be possible.

- **Extended HELLO Messages:** Extended HELLO messages extend simple HELLO packets with an additional neighbor list. Each node inserts all neighbors from which it can hear HELLO messages into this list. Thus, if a node does not find itself in the hello packet from a neighboring node, it marks the corresponding link as unidirectional. The main concern regarding this mechanism is the communication overhead influenced by size and frequency of the extended HELLO messages.

- **Reverse Path Search:** During the RREQ flood, multiple loop-free reverse paths to the source node are detected. Via a distributed search procedure, multiple RREPs try to find one or more bidirectional paths in this structure. The search procedure, described in detail in [MD02], is similar to the well known depth first search algorithm. If a RREP fails at a node, the corresponding reverse path is removed and alternate reverse paths are tried, if available. If all reverse paths from one node fail, the search backtracks to upstream nodes of that node and continues there. The search procedure stops if one or more reverse paths are found at the source node or all reverse paths have been explored.

3.1.2. Capacity of Ad Hoc Networks

The question about scalability is fundamental for mobile Ad Hoc networks. This section presents a fundamental analytical result concerning the maximum capacity of such networks

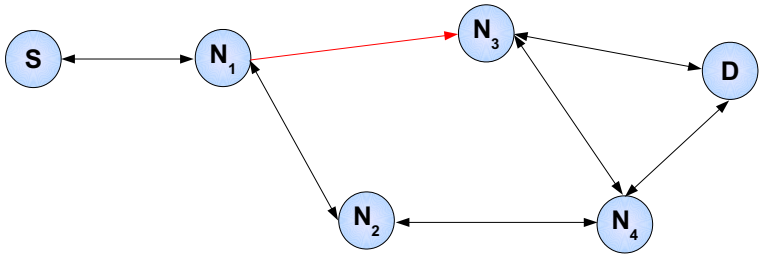


Figure 3.1.: Unidirectional Link Avoidance: Blacklists

published in [GK00]. After that, practical implications of this result are discussed.

In 2000, Gupta and Kumar published an upper bound for the capacity of ad hoc networks. Be n the number of identical randomly located nodes. If each node is capable of transmitting at W bits/sec and has a fixed range, the throughput $\lambda(n)$ obtainable by each node for a randomly chosen destination is

$$\lambda(n) = \Theta(\frac{W}{\sqrt{(n\ log\ n)}}) \tag{3.1}$$

for a non-interference protocol.

This fact is not really encouraging as it means that $\lambda(n)$ approaches zero with an increasing number number of nodes. Figure 3.2 shows the implication of equation 3.1 to IEEE 802.11b based networks with a maximum bitrate of 11 Mbit/s per node. However, this upper bound holds only for totally decentralized and self-organizing MANETs, i.e. other wireless networks, e.g. GSM networks are considered as wired network (to the base station) with only the last hop being wireless. This may also be a possibility for overcoming this limitation in MANETs: The formation of backbones and virtual subnetworks. However, this is a topic still open for research and is not considered within this thesis.

Another assumption made in [GK00] is the the traffic pattern. Each node chooses a destination node randomly from the set of all other network nodes. This leads to the effect that the average path length grows with the spatial diameter of the network, or equivalently the square root of the area. This implies, as approximation of the results in [GK00], a path length of $O(\sqrt{n})$ and a total end-to-end capacity of roughly $O(\frac{n}{\sqrt{n}})$, thus, an end-to-end throughput per node of

$$O(\frac{1}{\sqrt{n}}) \tag{3.2}$$

[LBC+01] takes the theoretical results from [GK00] and examines its implications for real-world networks. Especially the basic assumptions like random traffic patterns are questioned

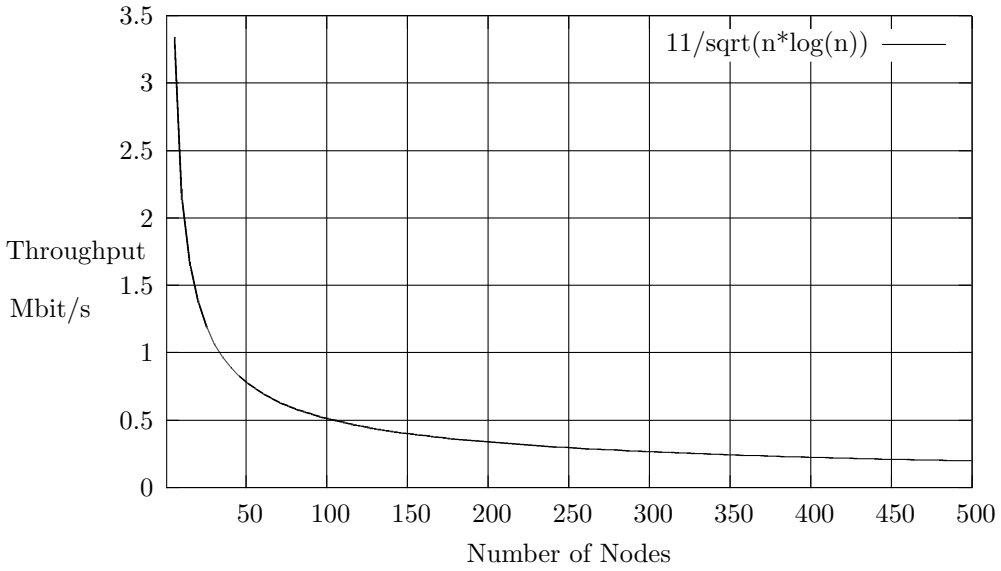


Figure 3.2.: Capacity of an 802.11b based MANET

and the effects of different traffic patterns are shown. It is argued that especially in bigger networks non-random traffic patterns will occur. Nodes are more likely to communicate with nodes nearby: Students within one lecture hall, members of the same department, inhabitants of a city. Examples of networks with predominantly local traffic patterns include LANs, the telephone network and caching systems in the Internet. Based on simulations using 802.11 PHY and MAC layer traffic patterns are presented that allow a nearly constant per node available throughput $\lambda(n)$. However, when it comes to predominantly non-local traffic patterns, the bound shown in equation 3.2 for per node capacity is approached.

3.1.3. Routing Complexity

Quite often throughout the literature, assumptions have been made concerning the performance, especially the signaling overhead of MANET routing protocols in different scenarios. Assumptions made included a tendency to better performance of reactive protocols compared to its proactive counterparts in highly mobile scenarios and vice versa in scenarios with little or low mobility. Mostly, also position-based protocols have been assumed to show better performance than conventional MANET routing algorithms.

The following section takes a closer look on these assumptions. Based on work published in [JV00] and [MWH01], the performance of conventional MANET routing protocols is shown and formulas to calculate the protocol overhead itself as well as the influence of mobility are presented. Concerning position-based routing protocols, the complexity of the different forwarding strategies is shown. Also the complexity of different location services is considered as these services are a necessary prerequisite for the different forwarding strategies.

3.1.3.1. Conventional Routing Protocols

For its analysis of routing protocol communication overhead, [JV00] classifies existing routing protocols into the two following categories: hello protocols or flooding protocols:

- **HELLO Protocols:** These protocols are based on HELLO messages emitted by all nodes to learn about the surrounding topology. Then the collected information or parts of it are broadcast to establish a basis for route computation.
- **Flooding Protocols:** This simple approach consists of flooding packets from the source node (e.g. RREQs). The path taken by packets successfully arrived at the destination node can then be used to transmit further information.

Almost all MANET routing protocols use the second approach - flooding of packets. It indeed seems very attractive as information only has to be stored about active routes. A second advantage is the quick reaction even in case of extensive topology changes. Concerning the routing algorithms presented in chapter 2, OLSR and TBRPF belong to the first class (HELLO) and AODV as well as DSR to the second (flooding).
Regarding protocol performance, the following overheads are considered: Fixed overhead and mobility overhead. The values for fixed overhead consider a MANET without any mobility. Before the results are presented in table 3.1, essential details about the underlying assumptions of this analysis are given. The complete derivation of the formulas presented in table 3.1 is available in [JV00]. An explanation of the parameters used for this protocol analysis can be found in section 2.3.1.2.

- **Radio Network Model:** It is supposed that the shape of the network is roughly stable and that it is always connected. The average number of link breakages per link, μ, is assumed to be constant. M, the number of edges in the network graph is also assumed to be constant, so average link breakage rate and average link creation rate are balanced. Hello packets are modeled as packets containing the list of neighbors of a node. Its average size therefore is Δ addresses. For HELLO protocols, the utilization of broadcast optimization is assumed, so ideally $\frac{N}{\Delta}$ packet emissions are sufficient to broadcast information to every node in the network.
- **Control Overhead in Fixed Networks:** The total control overhead for both protocol classes consists of route creation overhead plus fixed control overhead. For route creation, flooding protocols start with a RREQ flooding and if a route has been found the destination node sends a reply back to the source node. This results in a cost of $\lambda N(N + L)$ packets of constant size per second or a bandwidth cost of approximately $3\lambda N(N + L)$. Hello protocols do not cause any overhead at route creation (connection request) as they store routes and keep them up to date. However, this behavior results in a higher overhead due to mobility.
  The additional fixed control overhead is estimated as follows: Every $\frac{1}{h}$ seconds, each node emits a hello packet containing the list of nodes from which it hears hellos plus its own address (i.e. $\Delta+1$ addresses). This allows each node to learn the local topology up to two hops - information that is used to optimize broadcasting. In OLSR for example, multi-point relay sets are computed for efficient broadcasts. More on this topic can be

found in [JL00] and [CJ03]. Hellos thus cost hN packets per second which corresponds to a bandwidth of $h(\Delta+1)N$. A minimal subset of the topology has to be broadcast for route computation. This broadcasting costs rN packet emissions per broadcast which is $\tau r N^2$ packets per second and corresponds to a bandwidth of $\tau r \delta N^2$.
In case a flooding protocol uses hellos to detect link breakage, the additional overhead will amount to hN packets per second. The analysis neglects this factor and keeps only quadratic terms. Flooding protocols may also include a fixed control overhead due to active route timeouts. Regarding this additional overhead, the analysis assumes λ to be the rate of route creation or refreshing after timeout.

- **Additional Overhead due to Mobility:** Mobility for both protocol classes is visible through link breakage and creation. Link breakages are detected either when expected hello messages are missing or by link layer reporting. At link breakage, flooding protocols send a packet to the source node which proceeds with route creation. Hello protocols can cause additional overhead in the case a link of the broadcast topology breaks. This will result in an additional broadcast packet to update this information. The probability for link breakage in the broadcast topology is $\frac{\delta}{\Delta}$. The rates of floodings due to link breakage as well as the link breakage detection rate in the model are bounded by h.

As can be seen in table 3.1, both protocol classes include an $O(N^2)$ overhead. The results support and refine the common assumptions: better performance of flooding protocols in scenarios with high mobility and a low number of active routes. Hello protocols show a better performance when the number of active routes gets high. An additional conclusion of [JV00] includes the possible overhead caused by non-optimality of routes in flooding protocols. It is estimated with possibly $\geq 10\%$ of the data traffic even in sparse topologies.

Overhead	Flooding Protocols	Hello Protocols
Fixed Overhead (Packets)	λN^2	$\tau r N^2$
Fixed Overhead (Bandwidth)	$3\lambda N^2$	$(hd + \tau r \delta)N^2$
Mobility Overhead (Packets)	$min(\mu L, h)aN^2$	$min(\mu\delta, h)rN^2$
Mobility Overhead (Bandwidth)	$3min(\mu L, h)aN^2$	$min(\mu\delta, h)\delta rN^2$

Table 3.1.: Overhead of conventional MANET routing protocols

3.1.3.2. Position-Based Routing Protocols

Position-Based routing protocols are a promising approach for scenarios where location data is available. Several simulation studies about position-based routing protocol performance have been done, some of them comparing conventional algorithms with position-based ones. However, it is important to stress that in addition to the signaling overhead for packet forwarding, also the overhead created by the location service has to be considered. Some simulation studies, e.g. a comparison between GPSR and DSR published in [JM96], do not include traffic and time required to look up the position of the destination node.

Based on results published in [MWH01], the complexity of different position-based routing protocols is shown in table 3.3. The complexity of several location services is presented

in table 3.2. An introduction into all presented protocols except the GLS and Homezone forwarding strategies can be found in section 2.3.2.5. Details about GLS, developed as part of the Grid project [gri04], can be found in [LJC+00] and . Two independent proposals for using a virtual Homezone for the realization of a location service have been published in [GH99] and [Sto99].

Criterion	DREAM	Quorum System	GLS	Homezone
Type	All-for-All	Some-for-Some	All-for-Some	All-for-Some
Communication Complexity Update	$O(N)$	$O(\sqrt{N})$	$O(\sqrt{N})$	$O(\sqrt{N})$
Communication Complexity Lookup	$O(1)$	$O(\sqrt{N})$	$O(\sqrt{N})$	$O(\sqrt{N})$
Time Complexity Update	$O(\sqrt{N})$	$O(\sqrt{N})$	$O(\sqrt{N})$	$O(\sqrt{N})$
Time Complexity Lookup	$O(1)$	$O(\sqrt{N})$	$O(\sqrt{N})$	$O(\sqrt{N})$
State Volume	$O(N)$	$O(1)$	$O(log(N))$	$O(1)$
Localized Information	Yes	No	Yes	No
Robustness	High	Medium	Medium	Medium

Table 3.2.: Complexity of Position-Based Routing Protocols (Location Services)

Criterion	Greedy	DREAM	LAR	Terminodes	Grid
Type	Greedy	Restricted Directional Flooding	Restricted Directional Flooding	Hierarchical	Hierarchical
Communication Complexity	$O(\sqrt{N})$	$O(N)$	$O(N)$	$O(\sqrt{N})$	$O(\sqrt{N})$
Tolerable Position Inaccuracy	Transmission Range	Expected Region	Expected Region	Short-Distance Routing Range	Short-Distance Routing Range
All-for-All Location Service Required	No	Yes	No	No	No
Robustness	Medium	High	High	Medium	Medium

Table 3.3.: Complexity of Position-Based Routing Protocols (Forwarding Strategies)

The key aspect of the comparisons in table 3.3 and 3.2 is the scalability of the protocols with an increasing number of nodes in the MANET. A basic assumption for this analysis was a static node density in networks with different size. Below, details concerning the evaluation of the different location services are given:

The *type* field states how many nodes are required to provide location information and and for how many other nodes each of these nodes maintains location information. The value for *communication complexity* describes how many one-hop transmissions are required to find or update a nodes position. The average time it takes to perform a position update or lookup is given in the field *time complexity*. The field *state volume* indicates the amount of state required in each node that maintains the position of other nodes. If a protocol provides better quality of position information nearby the position of a node, this is indicated in the field *localized information*. The *robustness* of a location service is described as high, medium or low if it takes the failure of all nodes, a small subset of nodes or a single node to render the position of a given node inaccessible.
To summarize the results concerning location services, DREAM is the only protocol with an all-for-all approach and therefore has the highest communication complexity. While this results in a bad scalability with the number of nodes, the fast position lookups, its robustness and the availability of localized information make DREAM suitable for high-speed applications involving a limited number of nodes, e.g. local danger warning in a vehicular scenario. The results for the communication complexity and time complexity of quorum systems indicate a complexity of $O(\sqrt{N})$. However, the necessary management of the virtual backbone is not specified in [HL99] and therefore is not included in the analysis. It has to be stressed

that this approach relies on a conventional ad hoc routing protocol for backbone creation, therefore, the complexity of this protocol has to be added. GLS and Homezone are two approaches that can be seen as specialization of the quorum system but do not require an additional routing protocol. In case of predominantly local communication, GLS outperforms Homezone.

Concerning the evaluation results of forwarding strategies in table 3.3, the *type* field indicates the strategy used for packet forwarding. The values for *communication complexity* indicate the average number of one-hop transmissions necessary to send a packet from its source to its destination. This is done under the assumption that the destinations position is already known. The values for *tolerable position inaccuracy* are related to the receiving node. The *robustness* of the different forwarding strategies is defined as follows: It is low if the failure of a single intermediate node might result in packet loss and the necessity of setting up a new route. It is medium if the failure of a single intermediate node might result in packet loss but does not require the setup of a new route. Due to the characteristics of position-based routing, all approaches have at least medium robustness. A high robustness indicates that failure of a single intermediate node does not prevent the packet from reaching the destination node.

All forwarding strategies presented in table 3.3 have already been introduced in section 2.3.2.5. Concerning its performance, the following interesting observations can be made: The greedy approach shows a low communication complexity of $O\sqrt{N}$. Together with repair-strategies like face-2 [BMSU99] or perimeter routing [KK00], it can be recommended for highly mobile environments. A disadvantage of this approach is that the position of the destination nodes has to be known with high accuracy (one-hop transmission range). DREAM and LAR with its communication complexity of $O(n)$ do not scale well. While LAR works with any location service, DREAM requires an all-for-all type location service. An advantage of LAR and DREAM is its high robustness against node failures. This may qualify these approaches for networks with a high demand on reliability and low traffic. Terminodes and Grid are taking an hierarchical approach for position-based routing. As a greedy approach is used for long-distance routing, the performance of Terminodes, Grid and Greedy is similar. However, due to the usage of a non-position-based approach at the local level, Terminodes and Grid are less sensitive to position inaccuracies.

3.2. Simulative Results

While analytical methods are very beneficial to estimate theoretical limits of MANETs, they are mostly restricted to simple scenarios because of complexity reasons. Simulated models can help in this situation to gain knowledge about system performance in complex and realistic scenarios. The analytical results can then be used to validate the simulations for simple settings. To show the necessity of simulation studies, section 3.2.1 presents a simulative performance comparison of two very popular MANET routing protocols: AODV and DSR. While both protocols belong to the same class (reactive protocols, hello protocols) and therefore show the same performance in analytical performance studies, the following simulation shows fundamental differences.

3.2.1. Routing Performance

A simulative performance comparison of AODV and DSR has been published in [PRDM01]. This section provides a summary of these results to show how design decisions even within one protocol class can influence the performance in different environments.

The main differences of AODV and DSR include the different amount of routing information the protocols have access to. In DSR, every single request-reply cycle enables the source node to learn routes to all intermediate nodes on the way to the destination node. In addition, every intermediate node can learn routes to any other node on this route. Additional route information is gained by promiscuous listening of data packet transmissions. This way, routes to every node on the source route of the data packet can be learned. AODV on the contrary, does neither employ source routing nor promiscuous listening. Instead, AODV relies on RREQ floods more often what may result in a higher network overhead.
Another difference between AODV and DSR is the aggressive usage of route caching in DSR. In DSR, replies are sent to all requests reaching a destination node from a single request cycle. This way, the source learns alternative routes to the destination which can be used if the primary route fails. Therefore, necessary route discovery floods in the case of route failure are reduced accordingly. AODV, on the contrary, replies only to the first request arriving at the destination node. All further requests resulting from the same request cycle are ignored. The heavy utilization of route caching in DSR, however, does not come without problems. As stated in [ea99], stale routes may start polluting other route caches. In some circumstances it is possible that more caches are polluted by stale entries than are removed by error packets. Also, differences in the route deletion exist between AODV and DSR. While AODV uses RERR messages that are sent using a predecessor list, all nodes using a failed link on its route to any destination are informed. In DSR, RERR messages are sent back along the path the data packet took when it arrived at the failed link. Therefore, nodes that also have routes using the failed link but are not on the upstream route of this data packet are not informed instantly.

The simulations in [PRDM01] have been done using the ns-2 [FV03] simulator. The Distributed Coordination Function (DCF) of IEEE 802.11 has been used as the MAC layer protocol. The radio model was based on characteristics of Lucent's WaveLAN [Tuc93]. It was modeled as shared-media radio with a nominal bit rate of 2 Mbit/s/s and a nominal range of 250m. Traffic sources are modeled as continuous bit rate (CBR). Source-destination pairs have been spread randomly over the network. Simulation models included two different rectangular fields: a field of 1500m x 300m populated with 50 nodes as well as a field of 2200m x 600m populated with 100 nodes. The random waypoint model [ea98] has been used as mobility model. Node speeds have been uniformly distributed between 0-20m/s, the pause time has been selected from values between 0 (high mobility) to 900 (low mobility). Simulations have been run for 900 simulated seconds for the 50-node-scenario and for 500 simulated seconds for the 100-node-scenario.

The performance metrics evaluated included:

- **Packet delivery fraction:** The ratio of data packets successfully delivered to the destination to these generated by the sources.

- **Average end-to-end delay of data packets:** All possible delays of the data packets. Included are buffering due to route set-up delay, queuing at the interface queue, MAC retransmission delays as well as propagation and transfer times.
- **Normalized routing load:** The number of routing packets transmitted per data packet delivered at the destination. Each hop-wise transmission of a routing packet is counted as one transmission.
- **Normalized MAC load:** The number of routing, Address Resolution Protocol (ARP) and control packets (e.g. RTS/CTS, ACK) transmitted by the MAC layer for each delivered data packet. This value includes routing overhead as well as MAC control overhead. This metric also accounts for transmissions at every hop.

Simulation studies included varying mobility and varying number of sources as well as varying offered load with the following results:

3.2.1.0.1. Varying Mobility and Number of Sources: Experiments with 50 nodes, 10 to 40 traffic sources and a packet rate of 4 packets/s (3 packets/s for 40 traffic sources because of network congestion) showed similar packet delivery fractions for DSR and AODV with 10 and 20 sources. With 30 and 40 sources, AODV outperformed DSR by about 15 percent at lower pause times. At higher pause times (lower mobility), DSR shows better performance than AODV. Almost identical delays have been found in simulations with 10 and 20 traffic sources, simulations with 30 and 40 sources resulted in a better performance (about 25 percent) of AODV for low pause times. For higher pause times, DSR showed about 30-40 percent lower delays than AODV. Usually a 2-3 times lower routing load of DSR compared to the routing load of AODV has been found.
Experiments with 100 nodes have been done with the same parameters as with 50 nodes, except that 2 packets/s have been used as packet rate for 40 traffic sources. Packet delivery rates are similar for AODV and DSR for 10 traffic sources, however, the performance of DSR gets worse with a larger number of sources. AODV shows 22-41 percent higher packet delivery fractions compared to DSR in higher mobility scenarios. Similar effects have been observed for the delays: In scenarios with 10 nodes, AODV and DSR have similar delays. Higher delays by a factor of 2-6 for DSR have been noticed for high mobility with the factor increasing with the number of sources.
Routing load of AODV in high-mobility scenarios is about twice as much as DSR with 10 and 20 sources and about 15 percent higher than DSR for 40 sources. Concerning MAC load, DSR shows significantly higher load values than AODV for all numbers of sources except at very high pause times.
To summarize, the packet delivery fraction and delay values for DSR and AODV are similar for a low number of traffic sources. This observation holds for high and low mobility. If the number of sources is increased, DSR shows better performance under low-mobility conditions but is outperformed by AODV in high-mobility scenarios.

3.2.1.0.2. Varying Offered Load These simulations have been conducted at highest mobility (zero pause time). The 100 node model has been with either 10 or 40 traffic sources. The packet rate has been increased until throughput saturation. These simulations have shown the following results: with 10 sources, DSR's throughbut started to saturate at an offered load (combined sending rate of all data sources) of about 400kbit/s. A throughput saturation

of AODV has been detected at about 700kbit/s. The average delay of AODV has always been lower compared to the delay of DSR below 400kbit/s. Delays above this load should be regarded irrelevant as DSR then looses about half of the packets. AODV generated higher routing load than DSR. However, MAC load comparisons resulted in a reversal of trends: AODV showed a lower MAC load than DSR. In the scenario with 40 sources, saturation has been reached earlier. AODV saturated at about 300kbit/s while DSR saturated at about 200kbit/s. AODV, again, showed better delay characteristics than DSR. Routing load, as expected, was much higher for both protocols. Also, as before, while AODV showed a higher routing load, it showed a lower normalized MAC load than DSR.

3.2.2. Conclusion

The simulative comparison showed that DSR outperforms AODV in scenarios with fewer nodes and lower load as well as lower mobility. AODV outperforms DSR in more stressfull situations with a widening performance gap with increasing stress. Therefore, it is the protocol of choice in highly mobile scenarios.
Another interesting result of [PRDM01] have been effects between routing and MAC layer that can effect performance significantly. While DSR consistently showed lower routing load, the real load on the network has not been reduced. This is due to the fact that DSR generated more unicast routing packets which are expensive in the 802.11 MAC layer.

3.2.3. MAC Layer Performance

3.2.3.1. IEEE 802.11 MAC

The MAC layer of IEEE 802.11 [Com99] has to organize access to the shared medium. The basic problem here is to avoid collissions of packets in the wireless radio medium. Figure 3.3 shows an example: If medium access is not coordinated, nodes will send its data independently at arbitrary times. Thus, if node A sends a packet to node B, node C can not successfully send a packet to node D at the same time. In this case, a collission will occur at node D. As collission detection (CSMA/CD) is not possible in wireless networks[2], an alternative mechanism, collission avoidance (CSMA/CA) is used instead. However, similar problems as shown in figure 3.3 are still possible when using plain CSMA/CA. One classical problem of this approach is the so called "Hidden node problem" shown in figure 3.4. Node A and B as well as node C and B can hear each other. However, node A and C are not within radio range of each other and therefore can not directly coordinate its actions. What can happen in this situation is that node A and node C decide to send a packet to node B at the same time. This will result in a collission at node B.
In IEEE 802.11, a four-way handshake has been introduced to avoid this problem. To stay with the same example, suppose that node A wants to send a packet to node B. In this case, it first sends a RTS (request-to-send) packet to node B. Node B responds with a CTS (clear-to-send) packet that can be heard by all neighbors within the radio range of node B. Node A starts to transmit its data packet after it receives the CTS signal from node B. After reception of the data, node B answers with an acknowledgment (ACK) packet. Figure 3.5 shows the message sequence chart of this example.

[2]This problem is caused by the so-called near/far problem: a node can not send and listen for interference at his antenna at the same time.

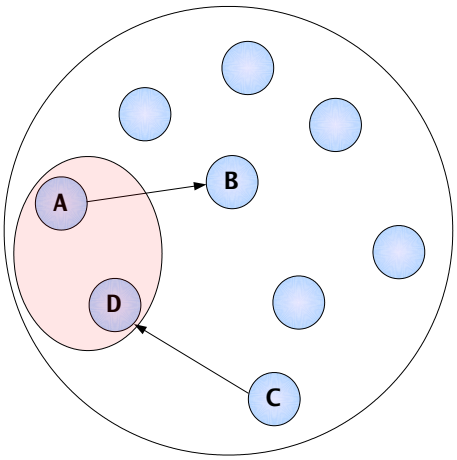


Figure 3.3.: Wireless Radio MAC

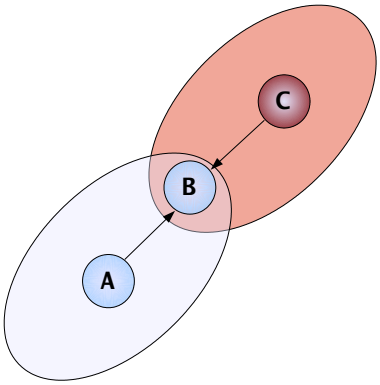


Figure 3.4.: The Hidden Node Problem

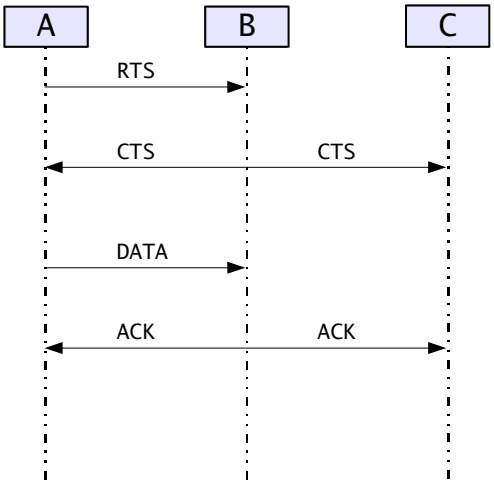


Figure 3.5.: Avoiding Hidden Nodes: Four-Way Handshake

3.2.3.2. Problems with 802.11 MAC in Ad Hoc Networking

However, while 802.11 avoids the hidden node problem using the four-way handshake, it introduces new problems concerning communication efficiency especially for the communication in MANETs. The problem introduced by the RTS/CTS mechanism and shown in figure 3.6 is called the exposed node problem: Node B transmits its CTS to node A, silencing also node C via the RTS/CTS mechanism. While it would be possible without doing any harm, node C is blocked from sending its data packets to node D at the same time.

To summarize, the problems of the IEEE 802.11 MAC layer include the silencing of all neighboring nodes while it would be sufficient to silence only the neighbors of the receiving node. This mechanism has to be used for every transmitted data packet. If collissions occur, nodes employ a backoff mechanism [Com99] as in ALOHA. This mechanism can again be wasteful. Results (e.g. [GGK01], describing a scaling experiment using a network ranging from 2 to 12 nodes), shows that the node throughput declines as $O(\frac{1}{N^{1.68}})$. This is far away from the capacity limit of $O(\frac{1}{\sqrt{n\ log\ n}})$ described in section 3.1.2 and [GK00].

3.2.3.3. Proposals for Improvement

Efficient and fair allocation of bandwidth among stations in the presence of hidden terminals has been addressed in the MACAW protocol [BDSZ94]. Collission-free transmission of one or more data packets is guaranteed by the FAMA protocol [FJLA97]. Several more mechanisms for efficient medium access are referenced in [RK01]. This publication also contains a description and simulations of a new MAC protocol called SEEDEX that is briefly explained in this section.

The idea of SEEDEX is based on the link layer protocol Adaptive Receive Node Scheduling

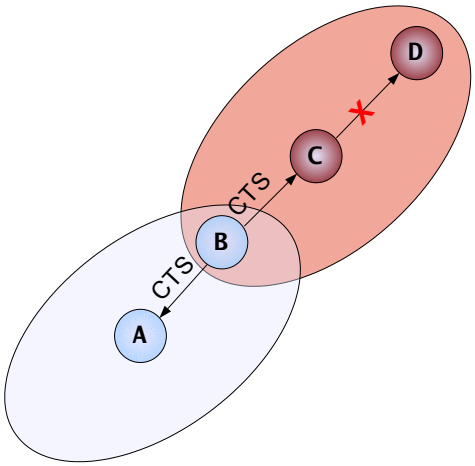


Figure 3.6.: The Exposed Node Problem

(ARNS, [RPKG88]) for a multiple satellite network. ARNS uses a pseudo-random time line to compute receiver schedules and provides each satellite with a schedule for its neighboring satellites so the intended receiver's antenna is pointed to the transmitter and is listening for a transmission. The approach of SEEDEX is similar: each node follows its schedule and either is in the state "L" which means silent, listening for packets or in the state "PT" which means possibly send a packet. It is supposed that every node T knows the schedules of all the nodes within its two hop neighborhood. T can then send a packet to its neighbor node R if either

- Node T is in state PT (i.e. it will possibly send a packet).
- Node R is in state L (i.e. it will stay silent).
- All neighbor nodes of R are in state L.

without having to fear a collision at the receiving node R.

All nodes in SEEDEX follow a random schedule. Each node chooses a probability parameter p, $0 < p < 1$. A slot is marked for "possible transmission" (state PT) with probability p and marked as "silent" (state L) with probability $1 - p$. This is done independently from slot to slot. A more complicated version of SEEDEX uses a pseudo-random number generator to drive a finite state machine.
The central idea of SEEDEX now is not to distribute all schedules of all nodes but instead to only distribute the seeds for the pseudo-random number generator. If node A knows the initial seed of the pseudo-random number generator used in node B, it knows node B's schedule. A fan-in and fan-out procedure is presented in [RK01] for distributing the seeds. While more details and configuration options concerning SEEDEX can be found in the original paper, it is interesting to have a look on a simulative performance comparison between SEEDEX and IEEE 802.11. This comparison, also presented in [RK01], has been conducted with the

network simulator ns2 in a scenario with 100 nodes. The results show a throughput that is about 10% greater than that obtainable from IEEE 802.11. Simulations concerning the transmission delay (in slots) for different normalized values of throughput in the simulated scenario (details on the normalization are given in [Kum00]) showed even better results that are presented in table 3.4.

The results from the SEEDEX performance study (similar results have been gained in numerous other studies concerning the performance of the IEEE 802.11 MAC) should be kept in mind when reading the following sections on performance. While these IEEE 802.11 based MANET performance studies already show quite acceptable results, some further performance gains can be expected using a different MAC layer.

Throughput	SEEDEX	IEEE 802.11
0.2	15.52	24.34
0.3	15.74	21.56
0.4	15.50	20.34
0.5	15.54	24.04
0.55	15.64	30.13
0.6	33.63	809.09

Table 3.4.: Throughput versus Mean Delay for SEEDEX-R and IEEE 802.11

3.2.4. PHY Layer Performance

One important question concerning the simulative performance evaluation of MANET protocols is the importance of physical layer (PHY) models for the simulation. Especially, it was unclear how different levels of detail would effect the accuracy of simulation studies about routing protocols and other protocols on higher layers. While a higher grade of detail has no negative impact on the quality of the simulation results[3], it complicates the modeling as well as it results in much higher simulation times. In addition, simulation results concerning higher layer protocols would then be specific to certain PHY models. This fact would increase the number of simulation runs necessary to get generic simulation results significantly. This section presents results that have been published in [SX02] concerning the effects of PHY layer abstraction in simulations of MANETs. This includes a performance comparison of the MANET routing algorithms AODV and DSR as well as the simulation of a gateway discovery protocol with different PHY layer simulation models.

Important effects at the PHY layer include interference, fading and path loss. Effects, that are explained briefly in the following paragraphs:

3.2.4.0.1. Interference Interference and noise at each receiver are calculated as the sum of all signals on the channel other than the one being received by the radio plus the thermal (receiver) noise. This is used to calculate the signal to noise ratio (SNR) which determines the

[3]However, a higher grade of detail increases the possibility of modeling- and implementation-errors because of the higher complexity of the model.

probability of successful signal reception. Often used models to calculate the signal reception are

- **SNR threshold based:** The SNR value is used directly and compared with an SNR threshold. Only signals whose values are above the threshold at any time during the reception are accepted.
- **BER based:** In BER based models, a probabilistic decision is made concerning whether or not each frame is received successfully based on the frame length and the Bit Error Rate (BER) deduced by SNR and the modulation scheme used at the receiver.

3.2.4.0.2. Fading and Path Loss Fading denotes the variation of the signaling power at the receiver. An important factor, creating varying path conditions from transmitters, is node mobility. Commonly used models to describe fading in MANET environments are Rayleigh or Ricean distributions.
In addition to fading, also path loss occurs on the wireless link. In the simulation study conducted in [SX02], two-ray and free space models have been used to describe pathloss. The two-ray model path loss model is recommended for line of sight (LOS) microcell channels in urban environments [Rap95] but also often used for the simulation of MANET environments. This is due to similar characteristics of those scenarios like low antenna height and low transmission power.
The free space path loss model is a basic, idealized reference model for radio propagation. This results in the effect that even nodes far away from the transmitter can receive packets. Compared to simulations using the two-ray model, simulations using the free space path loss model tend to give better result in absence of other modifications.

Additional details, especially about the modeling of the physical layer preamble and a comparison of the PHY models used in Glomosim [glo], ns-2 [ns-] and OPNET [opn] can be found in [TMB01].

3.2.4.1. Previous Work

Previous work related to this area includes [HBE+00], [TMB01] and [TBLG99]. [HBE+00] gives an overview about the trade-offs of detail in wireless simulation and also gives practical examples: simulation studies include the effects of MANET routing protocols on energy consumption as well as radio-based outdoor localization and radio-based robot following. [TBLG99] describes the impact of wireless channel models on the accuracy and execution time of large-scale simulation models. An overview of the GloMoSim simulation library an also be found in [TBLG99]. [TMB01] includes performance studies of the AODV and DSR MANET routing protocols. The effects of different PHY models on the results of the performance studies are examined.

In [TMB01], simulations have been done with the GloMoSim simulator. The following simulation settings are used: the size of the simulation area is given with 1200m x 1200m. 100 nodes are placed randomly in this area. As mobility model, the random waypoint model (0 - 20 m/s, 100s pause time) has been used. To create traffic, CBR sessions have been simulated. For the simulation runs, different path loss models (free space and two ray), different fading models (none, ricean, rayleigh) and different signal reception models (SNR threshold based

and BER based) are used for a performance study of the MANET routing protocols AODV and DSR in this environment.

The analysis of these simulations showed a severe impact of the different PHY models on the results of the performance study. The results that can be found in detail in [TMB01] are summarized below:

- **Packet Delivery Ratio, Free Space Path Loss Model:** The values for packet delivery ratio (PDR) of AODV and DSR with different signal reception and fading models were all close to 1.0. However, a change in the relative ranking[4] of AODV and DSR occurred with different models.

- **Packet Delivery Ratio, Two Ray Path Loss Model:** PDR values have been observed between 0.76 and 0.25 for AODV and between 0.42 and 0.35 for DSR. This variation resulted from the usage of different PHY models. In addition, also the relative rankings of AODV And DSR did not stay constant between simulations with different models. Depending on the PHY model used, either AODV performs significantly better then DSR or DSR clearly outperforms AODV.

- **End-to-end Delays, Free Space Path Loss Model:** No significant delays have been found in this setting, both protocols (AODV And DSR) achieve values below 0.3 seconds.

- **End-to-End Delays, Two Ray Path Loss Model:** Significant differences have been experienced in this setting. While DSR showed a relative static behavior (end-to-end delays have been observed between 2.9 and 2.9 seconds), the effects of different PHY models on AODV are much higher. End-to-end delays vary between 2.5 and 10.2 seconds. Again, the relative ranking of the two protocols changes clearly depending on the PHY model.

3.2.4.2. Validation of Previous Results

As the GloMoSim configuration files for the simulations published in [TMB01] have been available on the Internet, our work for [SX02] included a first validation of these findings, partially also to check the correctness of our own installation. As expected, our own simulations completely confirmed the results from [TMB01]. However, a closer look on the simulation settings showed that the different PHY settings did have a heavy impact on the radio range. GloMoSim is delivered with a script called "radio_range" that takes standard configuration files, checks the radio settings in connection with the models used for describing the PHY layer and calculates the resulting radio range of the nodes.
This script, confirmed by a manual validation, showed a variation of the nodes radio range between approx. 376m and 2122m. Within an area of 1200m x 1200m, populated with 100 MANET nodes, these changes of the radio range can be expected to have quite an impact on the performance of MANET routing protocols. Therefore, we decided to focus on the radio range in our simulations [SX02] that are described below.

[4] In this context, a "change in relative ranking" means the following: Protocol A performs better than protocol B with a physical layer simulation model PHY 1. Simulations of exactly the same scenario using a different physical layer simulation model PHY 2 show better performance of protocol B than protocol A.

3.2.4.3. Focusing on the Radio Range

Based on the observations described above, we decided to complete another study on the influence of PHY models on MANET protocol simulation. This included, again, a performance comparison between AODV and DSR for better comparability with the results described in [TMB01]. In addition, we simulated the behavior of a gateway discovery protocol [XB02] developed at our institute and also examined its sensitivity to different PHY models.

We also used the GloMoSim simulator with a similar setting as presented in [TMB01] for comparability reasons. However, in addition to a fixed radio range of 250m for all PHY models, we decided to select a scenario of size 1000m x 1000m that has been populated with 17 uniformly distributed MANET nodes. This was done to increase the possibility of multihop routes that was zero in some scenarios described in [TMB01][5]. Other simulation settings have been[6]:

- MOBILITY RANDOM-WAYPOINT
- RADIO-TX-POWER: Adjusted for different PHY models to achieve a radio range of 250m
- RADIO-TYPE: RADIO-ACCNOISE
- RADIO-FREQUENCY: 2.4e9 (hertz)
- RADIO-BANDWIDTH: 2Mbit/s
- RADIO-ANTENNA-GAIN: 0.0
- MAC-PROTOCOL: 802.11
- MOBILITY-WP-PAUSE: 20s
- MOBILITY-WP-MIN-SPEED: 0
- MOBILITY-WP-MAX-SPEED: 10

3.2.4.4. Effects of the Radio Range

We conducted simulations using the free space as well as the two ray path loss model. They have been used in combination with different fading models: NONE[7] and the SNR threshold based signal reception model. Below, the simulation results are summarized. A complete documentation of these results can be found in [SX02].

- **Packet Delivery Ratio, Free Space Path Loss Model:** The PDR of AODV varied between 0.52 and 0.61, results for DSR varied between 0.35 and 0.39. The relative ranking was stable.

[5]The diameter of a rectangle with size 1200m x 1200m and therefore the largest possible distance between two nodes within this rectangle is approx. 1697m. In scenarios with a radio range of approx. 2122m, any node can transmit to any other node via one single-hop connection.
[6]The original names of GloMoSim configuration parameters are used whenever possible.
[7]Corresponds to the GloMoSim setting AWGN (added white gaussian noise), Ricean Rayleigh

- **Packet Delivery Ratio, Two Ray Path Loss Model:** The PDR of AODV varied between 0.53 and 0.6, PDR of DSR varied between 0.31 and 0.39. The relative ranking was stable.
- **End-to-end Delays, Free Space Path Loss Model:** The end-to-end delays of AODV varied between 0.07 and 0.1 seconds, values for DSR varied between 0.15 and 0.2 seconds for different PHY models. The relative ranking was stable.
- **End-to-End Delays, Two Ray Path Loss Model:** The end-to-end delays of AODV varied between 0.07 and 0.08 seconds, values for DSR varied between 0.17 and 0.19 seconds. Again, no changes in the relative ranking have been observed.

To conclude this section we can say that while different radio models have an effect on the results of MANET protocol simulation, they can very well be replaced by simpler and more efficient mechanisms for certain purposes. As we have seen in the previous sections, different PHY models give very similar simulation results if simple prerequisites, like a stable radio range for the different simulation scenarios, are ensured. As a result, simulations conducted for this thesis focus on one physical layer model with constant radio range.

3.2.5. LIP Performance in a Vehicular Scenario

The application ideas in section 2 as well as the idea of 4G mobile communications illustrate that connectivity of MANET nodes to other networks can significantly enhance its possibilities. The study presented in this section has its focus on connections from vehicular MANET nodes to local information points (LIPs). As described in chapter 5, at least temporary access from the MANET nodes to the Internet is also assumed in our security framework LKN-ASF. Based on a study published in [ES01], the possibilities of single-hop access to LIPs in a highly mobile scenario are shown.

3.2.5.1. Simulation Scenarios

Simulations done in [ES01] included scenarios with static and mobile nodes as well as one and two LIPs. As network simulator, GloMoSim [glo] has been used with the following settings and a radio range of 250m:

- PROPAGATION-LIMIT: -111.0
- PROPAGATION-PATHLOSS: TWO-RAY
- RADIO-FREQUENCY: 2.4e9
- RADIO-BANDWIDTH: 2000000
- MAC-PROTOCOL: 802.11
- ROUTING-PROTOCOL: STATIC[8]

[8] As only single-hop communication between the mobile nodes and the LIP is regarded, the necessary routes are specified directly via a route-file.

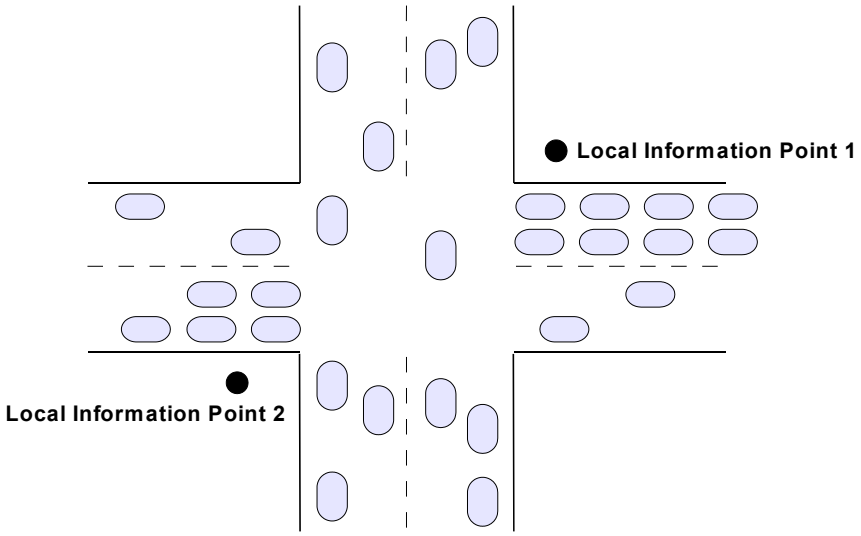


Figure 3.7.: LIP Performance: Intersection

For the static scenario, a crossing has been modeled that is depicted in figure 3.7. For simulations with only one LIP, LIP1 (upper right) has been used. For the mobile scenario, a part of a straight road has been modeled with various numbers of nodes passing by a LIP (see figure 3.8).

3.2.5.2. Simulation Results

Simulations have been done without node mobility in the crossing scenario and with node mobility in the road scenario. The crossing scenario included simulations with one LIP, simulations with two LIPs using the same frequencies and simulations with two LIPs using different frequencies. Antennas are placed in the middle of each vehicle, the height of the vehicle antennas was assumed to be 1,5m, the height of the LIP antenna was assumed to be 10m. Traffic has been modeled as FTP and HTTP. Concerning the FTP traffic, data was sent from the mobile nodes to the LIP(s). Each node tried to send 1000 packets with size 1500 Byte each, i.e. 1,5 MB per node. For the HTTP traffic model, the LIP has been used as the only server node in the simulation. The minimum waiting time between two page requests has been set to 30 seconds. Below, results of simulations in different scenarios are presented:

The expectation of a linear increase of the necessary connection time to transmit the 1,5 MB from all nodes to the LIP has been confirmed by the simulations. Given the simulation settings described above, it takes about 2 minutes for 10 nodes to transmit its data at the same time to one LIP. As shown in figure 3.9, the throughput at the LIP is almost constant. The second graph in figure 3.9 shows the average throughput per node that decreases as all nodes have to share radio capacity to connect to one single LIP.
The simulations with two LIPs using the same frequencies did not result in any improve-

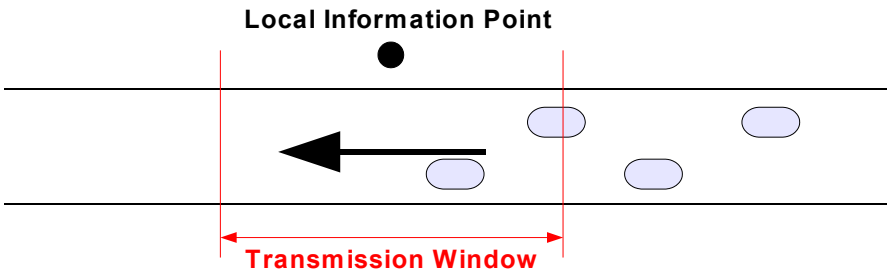


Figure 3.8.: LIP Performance: Straight Road

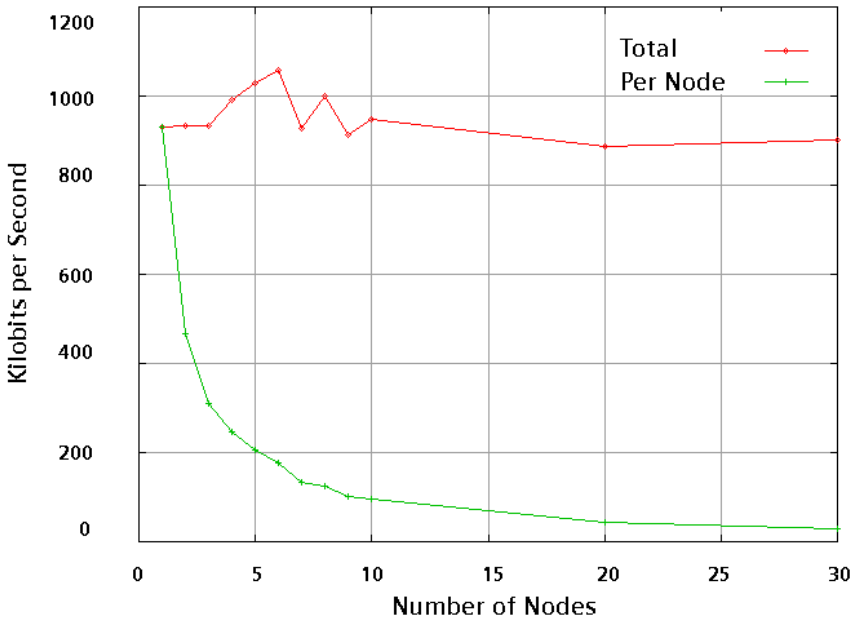


Figure 3.9.: LIP Performance: Throughput, 1 LIP, Static Scenario

ments. Instead, the throughput even decreased somewhat compared to the scenario with one LIP only. A first assumption was that this is due to the higher number of collissions in this scenario. This has been confirmed by corresponding simulation results. This indicates that there is also no decrease of the necessary durations of the connections. As before, it takes about 2 minutes for 10 nodes to transmit its data to the two LIPs.

Results of simulations with two LIPs using different frequencies have shown much better results (LIP1 has been configured to 2.40 GHz, LIP2 to 2.42 GHz). As the two LIPs do not disturb each other, about twice as good results have been obtained with respect to throughput and necessary connection time. A complete documentation of all simulation results is available in [ES01].

The mobile scenario, depicted in figure 3.8, involved a straight road with a length of 1000m, the LIP was placed in the middle at 500m. Different to the static scenario the LIP height is configured to be 20m, its distance to the road as 15m. Given all these simulation settings, table 3.5 summarizes how much time is available for a node driving at speeds from 30 km/h to 100km/h to communicate with the LIP.

Velocity	Time required for 1km	Communication Window
30 km/h	120 s	22.6 s
50 km/h	72 s	13.7 s
80 km/h	45 s	8.4 s
100 km/h	36 s	6.7 s

Table 3.5.: LIP: Communication Windows

Nodes moving with 100km/h have been the fastest nodes within the simulation. Even at such high speeds single-hop communication with the LIP is still possible with quite good throughput. In table 3.6, the absolute throughput that is possible at different node speeds is summarized.

Velocity	Communication Window	Absolute Throughput
30 km/h	22.6 s	20.47 Mbit
50 km/h	13.7 s	11.21 Mbit
80 km/h	8.4 s	7.00 Mbit
100 km/h	6.7 s	5.46 Mbit

Table 3.6.: LIP: Absolute Throughput at Different Node Speeds

3.3. Real World Tests

In the following sections, real world tests from various testbeds at the LKN and also from outdoor-testbeds are presented. Section 3.3.1 describes results of performance tests with the AODV routing protocol. Section 3.3.2 discusses the performance of TCP/IP for IEEE 802.11 based, mobile environments. Finally, section 3.3.3 presents performance results from tests using off-the-shelf IEEE 802.11b hardware in a highly mobile environment.

3.3.1. AODV Performance

In [MRS02], performance tests of AODV in a real-world testbed have been done using the following setup: 2 PC Desktop computers using IEEE 802.11b Wireless LAN cards (Orinoco Silver) via a PCMCIA adapter, two Laptop Computers and a Compaq iPAQ H3870 Pocket PCs using the same Wireless LAN equipment via a PCMCIA extension jacket. A standard Linux installation (SuSE) has been used on the Desktop computers as well as on the Laptops. The iPAQ computer (206 MHz ARM CPU, 64 MB RAM) has been set up with the *Familiar* Linux distribution [fam], version v0.6 with GPE, for handheld computers. The *skiff*[9] cross compilation toolchain has been used to compile the AODV routing protocol as it has only been available as source code. As the skiff cross compilation toolchain is a bit outdated by the time of this writing, newer alternatives for this purpose can be found at `http://handhelds.org/download/toolchain`. The Kernel-AODV [aod] implementation available from the Wireless Communication Technologies Group (WCTG) at the National Institute of Standards and Technology (NIST) has been used for routing.

For the tests, the following configuration of the wireless LAN cards has been used:

```
# Wireless LAN adapter configuration
#
case "$ADDRESS" in
*,*,*,*)
  ESSID="lknaodv"
  MODE="Ad-Hoc"
  RATE="11M"
  CHANNEL="10"
  ;;
esac
```

Normally, if no direct route from the source node to the destination node can be found, AODV looks for a multihop route as described in section 2.3.2.2. As an increase of the nodes distances would have been rather complicated, resulting in hardly reproducible effects, it was decided to leave all nodes in a static configuration within the lab. Link availability and link breakage have instead been simulated by the usage of a packet filter. In our setup, iptables [ipt] has been used for this purpose. A filter has been set in the prerouting chain of the mangle table that drops all incoming broadcasts (IP 255.255.255.255) from specified clients to the AODV socket (port 654). This results in an indirect modification of AODV's neighbor list to influence its routing decisions. The iptables-command to remove the unwanted packets as described above is:

```
iptables -t mangle -A PREROUTING -p UDP -s 192.168.1.X --sport 654 -d
 255.255.255.255 --dport 654 -j DROP
```

The results can be checked via the AODV routing table that is shown below.

```
192.168.1.102:~ # cat /proc/aodv/route_table Route Table
------------------------------------------------------------
```

[9] `http://embedded.centurysoftware.com/docs/nx/iPAQ-skiff.html`

Figure 3.10.: AODV Testbed: Example Topology

IP	Seq	Hop Count	Next Hop
192.168.1.103	43	2	192.168.1.101
192.168.1.101	1	1	192.168.1.101
192.168.1.102	5	0	192.168.1.102

Another way to check the effects of the iptables settings is the graphical tool aodv-monitor that always shows the current network topology. aodv-monitor has been available at the NIST WCTG site [aod] previously but it has now been removed.
For the simulation results described below, the iptables filtering has been used to configure chains of nodes with different lengths. Figure 3.10 shows this simple topology with 4 nodes (3 hops).

3.3.1.1. Simulation Results

A complete documentation of all simulation results is available in [MRS02]. Within this section, the most important results as well as its explanation are given.
Packet round trip times (RTT) have been one interesting point. For the measurement of RTTs, the well-known command *ping* has been used. In addition to the average RTT, especially the RTT of the first packet has been interesting. This is due to the fact that for the first packet, the AODV RREQ mechanism has to successfully find a route to the destination before this packet can be transmitted. For the average RTT, the packets following the first packet have been used. To limit the effect of outliers due to lost packets, the median has been used do determine the average RTT.

Figure 3.11 shows the observed RTTs, times are given in ms. For 1 hop, it is not necessary to start a RREQ flood because the direct neighbors are known via periodic HELLO messages. This effect can be seen clearly in the results as the RTTs for the first packets and the average RTT are almost identical. Another interesting observation was that RTTs for the first packet are growing faster than the average RTTs with a growing number of hops.

For the one hop scenario, measurements have been made with and without AODV[10]. The

[10]When the Wave LAN cards are configured in ad hoc mode, communication with direct (one-hop) neighbors is possible without the utilization of a dedicated ad hoc routing protocol.

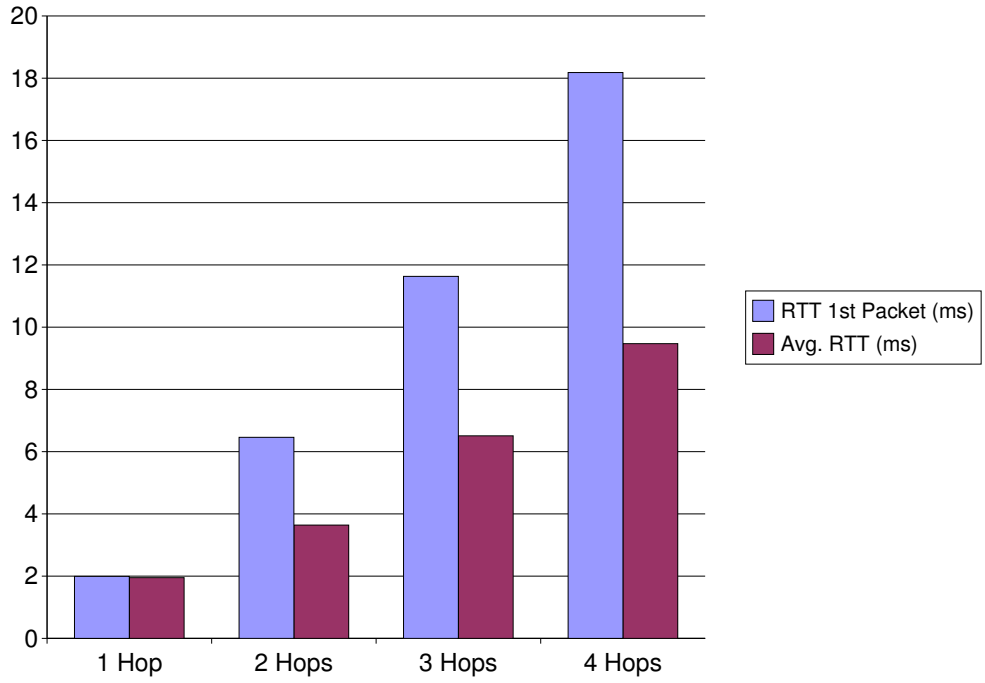


Figure 3.11.: AODV Testbed: Packet Round Trip Times

RTTs in this case did not show significant differences: without AODV, a RTT of 1.96ms has been observed for the first packet, a RTT of 1.92ms for the following packets. Compared with the one hop scenario using AODV, the RTT has been 0.03ms lower for the first packet as well as for the following packets.

Due to HELLO messages that are transmitted continuously for neighbor detection, idle traffic is generated even without any data transmissions in the network. In our testbed, an idle traffic of 0.5 kbit/s per node has been observed.

Further measurements included possible TCP and UDP throughput. These measurements have been done with the tools ntop [nto] and netperf [net]. Figure 3.12 shows the possible throughput (TCP, UDP) for 1 hop as well as for 2 hops. As expected, a higher throughput is possible with UDP in this scenario because of protocol complexity. Another fact that can clearly be seen in figure 3.12 is the effect of shared bandwidth and medium access. All stations are within radio range of each other and using the same channel. This, by the way, is not an unrealistic assumption at least for parts of a MANET. For TCP, this results in a throughput of 4.7 Mbit/s in the one-hop setup and 2.3 Mbit/s in the two-hop setup. For UDP, a throughput of 5.8 Mbit/s in the one-hop scenario and 2.9 Mbit/s in the two-hop scenario has been achieved. In experiments with 4 hops, the throughput has been below 1.5 Mbit/s.

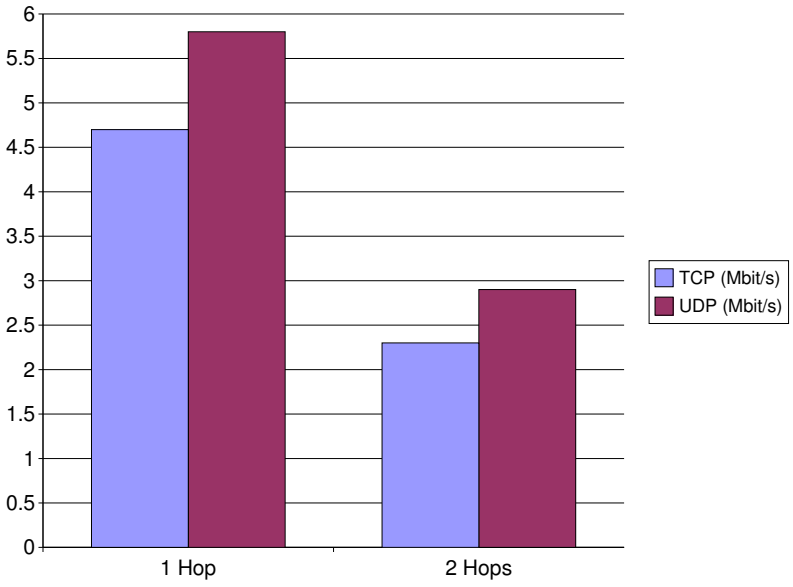


Figure 3.12.: AODV Testbed: TCP and UDP Throughput

3.3.1.2. Possible Optimizations

Possible optimizations include especially the delays for the route request procedure. The delay measurements presented in this section show only the delay introduced by a successfull RREQ procedure. The real delay from the first request to send a data packet until a route has been found can be much higher. For a chain-topology with 5 nodes, a delay of more than 5 seconds (!) has been observed to find a route from node 1 to node 5. The reason for this high delay is the way how RREQs are handled in this particular implementation: the first broadcast is only successfull if the immediate neighbors of the requesting node know a route to the destination. If this is not the case, a second broadcast is sent 2635ms later. Also this broadcast does not reach the destination, it needs a third broadcast that is sent another 2630ms later to make it to the one-hop neighbor of the destination. This node knows the route to the destination because of its periodic HELLO messages.

If protocol optimizations are done, the environment as well as the application requirements have to be taken care of. However, the following proposals might be beneficial for most scenarios:

- Enhancement of the RREQs initial lifespan (time-to-life, TTL)
- Reduction of delays between RREQ broadcasts. Based on our observations, values as low as 12ms (total) are sufficient to get feedback from nodes three hops away.
- Optimization of idle-traffic to the characteristics of the MANET, especially its dynamicity.
- Adaptation of the route lifetime (currently 5 seconds) to the MANETs characteristics.

3.3.2. TCP/IP Performance

A widely known problem of TCP is its poor performance over lossy, wireless channels. TCP has not been designed to be used over such channels but instead has been optimized for performance in the Internet. Thus, changes in the TCP stack of the communicating devices are necessary for optimal link utilization. If this way is not practicable, alternative approaches are possible. E.g. a proxy-based architecture for efficient TCP over satellite has been developed in [Stu02] with quite significant performance improvements compared to classical TCP.

As our testbeds are based on IEEE 802.11 Wireless LAN, we have been interested in the TCP performance in this environment to see whether modifications of the TCP stack might lead to significant improvements. Therefore, the test environment that is shown in figure 3.13 has been set up. This section provides a summary of the experiments and its results. The full documentation can be found in [STS03].

The test environment consisted of an Compaq iPAQ H3870 equipped with an IEEE 802.11b Wireless LAN card, a Wireless LAN access point and a desktop PC. The iPAQ was connected to the LKN intranet over a wireless link. All other devices have been interconnected with 100 Mbit/s Ethernet as shown in figure 3.13.
The experiments included measurements of the available signal strength, UDP throughput measurements as well as TCP throughput measurements. Since not only the relationship between throughput and signal strength but also between throughput and packet loss was of interest, a simple script has been implemented to get the required data. This script consisted of a server, running at the desktop PC and a client running on the iPAQ. While the server is listening for incoming UDP packets, the client initiates the transmission of 1000 UDP packets in each direction. Hereby, the data-field length can be specified by the user. The packet loss can now easily be calculated as the number of transmitted and received UDP packets is known. The TCP measurements have been done using *wget* [wge] to download a large file from the LKN intranet.
All performance measurements were done by moving the PDA to different locations in the building with the access point remaining stationary. At every location, several cycles of the following measurements have been done:

- Measurement of the access points signal strength, using iwconfig [wir].
- Measurement of the UDP packet loss rate.
- Measurement of the average TCP data rate using wget.

The results of the measurements are shown in figure 3.14 and 3.15. Figure 3.14 shows the UDP packet loss experienced at different SNR levels. The size of the data-field has been set to 1029 bytes for these measurements. Figure 3.15 shows the TCP data rate obtained during the wget measurements at different SNR levels.
As can be seen clearly, above a SNR of 15dB, no significant packet loss occurred. On the other side, for a SNR worse than 5dB, almost all packets have been lost. Concerning the coverage area of the access point, the region where a SNR between 5dB and 15dB occurs has been found to be very small. Especially indoors, it did only need a few steps away from the access point to experience this decrease of the SNR by 10dB. Therefore, one result from the experiments is that with 802.11b, packet loss occurs only at the very edge of the radio range, shortly

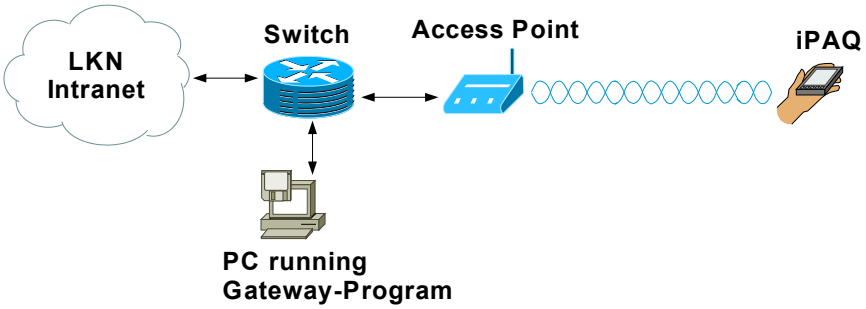


Figure 3.13.: TCP Performance over IEEE 802.11: Testbed Setup

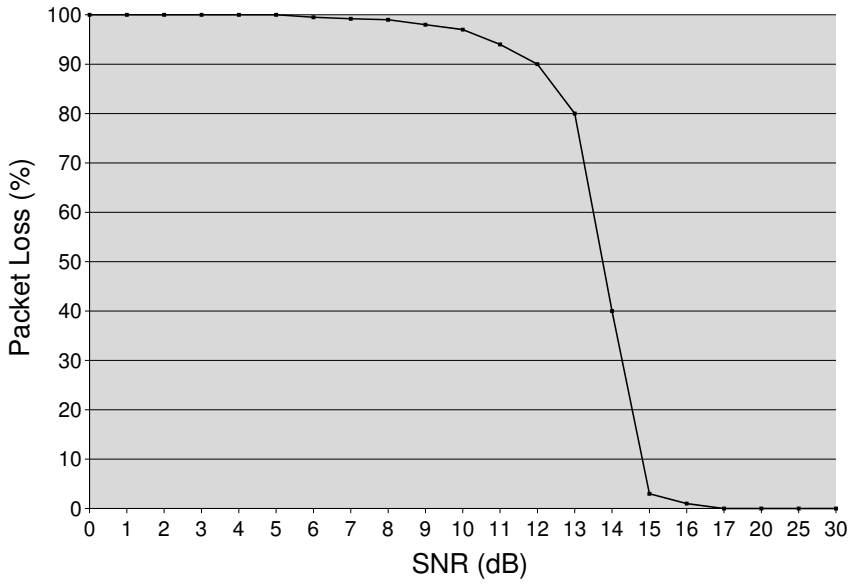


Figure 3.14.: TCP Performance over IEEE 802.11: Packet Loss

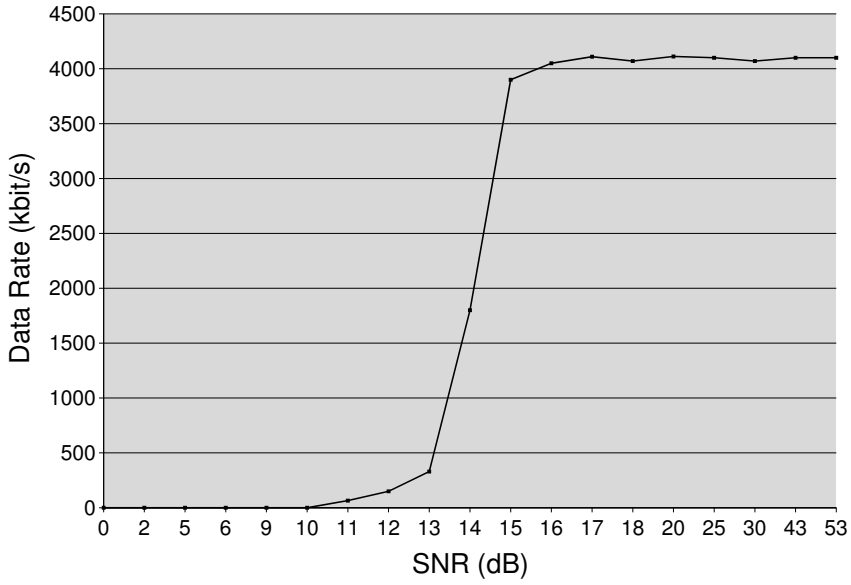


Figure 3.15.: TCP Performance over IEEE 802.11: Data Rate

before the link breaks completely. The reason for this observation can be found in the layer 2 signaling and retransmission scheme that is used by 802.11b. By default, this scheme tries to send a frame eight times before giving up. While reducing packet loss, these retransmission increase the jitter in the RTT of a packet. This may result in performance degradations of TCP due to the window size calculation by the congestion avoidance algorithms. However, as can be seen in figure 3.15, the performance of TCP remains almost constant until a SNR of about 15dB when frames start to get lost despite the layer 2 retransmission scheme.

The possible effects of layer 2 retransmissions on TCP performance have also been mentioned in several publications, e.g. [SRK03b, ea97, SRK03a]. To mitigate the problem with TCP's congestion control in wireless environments, it has been proposed to smooth the channel by link layer automatic repeat requests at a faster timescale than that of the TCP control loop. According to [SRK03b], this should result in the wireless link being perceived as a constant channel, but with lower capacity. This is exactly the effect we did observe during our experiments with IEEE 802.11b.

3.3.3. Performance of IEEE 802.11b in Vehicular Environments

An important question concerning highly mobile scenarios was how the IEEE 802.11b hardware, used in our testbeds, would perform in situations with high relative speeds. As this hardware is usually advertised for environments with low mobility, e.g. nomadic computing, we have especially been interested in the results of measurements in oncoming-traffic scenarios as shown in figure 3.16.

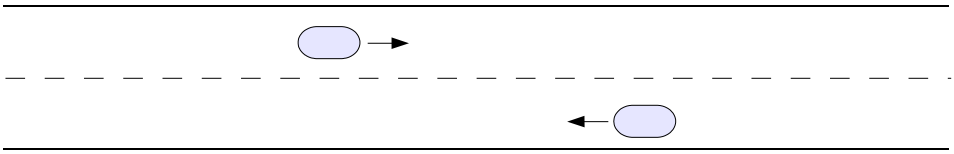


Figure 3.16.: IEEE 802.11b Performance: Oncoming Traffic Measurements

3.3.3.1. Mobile Testbed Setup

Orinoco Silver Wireless LAN cards have been used for the experiments described in this section. A detailed description of the wireless cards properties can be found in appendix A. All measurements have been done on Laptops running SuSE Linux 8.1. The mobile nodes have been equipped with external roof antennas for better connectivity. As the focus of the measurements has been the determination of TCP and UDP throughput at different relative speeds, MAC features like the RTS/CTS mechanism and fragmentation have been disabled. The data that has been logged during the measurements included the current time as well as the GPS position of the vehicles to calculate relative distance and speed. Interesting parameters have also been the SNR, the currently selected hardware bitrate and the number of lost frames. As IEEE 802.11b employs MAC level retransmissions to keep TCP transmitting at full speed (this fact has already been mentioned in 3.3.2), the number of lost packets that are given below refer to the MAC layer, not to TCP packets.

To automate all measurements as far as possible, a Perl-based measurement framework has been developed that is described in [IS04] in more detail. Also the complete documentation of all measurement results can be found in [IS04]. Information concerning the physical channel itself have been queried from the device driver. Jean Tourrilhes' wireless tools [wir] offer a convenient possibility to access this information using *iwconfig* and *iwspy*. Throughput and RTT have been measured with tools like iperf [ipe] and ping [pin]. Time and current position have been queried from a GPS device. Several Linux kernel modules [wir] exist for the IEEE 802.11b card used. *wvlan.o* has been found to be the best alternative for our purpose and therefore has been used for our measurements.

3.3.3.2. Measurements

At the beginning of each measurement, the mobile stations have been out of range. The drivers thus had to start the measurement software manually before each test run. Ping floods[11] have been used to test connectivity and to start iperf as soon as the first ICMP echo reply has been received. This mechanism was necessary to avoid a delayed start of iperf. If iperf is started before the stations are within range, it takes several seconds after connection

[11] ICMP echo requests have been sent out at intervals of 50ms.

establishment before iperf starts with its work. All testruns have been done with the IEEE 802.11b cards operated in ad hoc mode.

3.3.3.3. Results

Table 3.7 and 3.8 show a brief summary of the measurement results. All numbers presented in these tables have been measured in the oncoming traffic scenario (see figure 3.16) at the Warngau airport. Thus, no obstacles for radio propagation have been present and the two vehicles did have a direct line-of-sight at all times. All numbers in table 3.7 have been gained using iperf in TCP mode. A summary of the UDP throughput measurements is presented in table 3.8.

Rel. Speed	HW Bitrate	Duration	Throughput
15 km/h	1 Mbit/s	230s	22 MB
15 km/h	11 Mbit/s	70s	33 MB
60 km/h	auto	50s	16 MB
100 km/h	1 Mbit/s	27s	2.4 MB
100 km/h	auto	28s	10 MB
140 km/h	1 Mbit/s	20s	1.8 MB
140 km/h	auto	22s	8 MB
180 km/h	1 Mbit/s	17s	1.5 MB
180 km/h	auto	17s	6 MB
240 km/h	1 Mbit/s	15s	1.3 MB
240 km/h	auto	15s	5.5 MB

Table 3.7.: Mobile Testbed: TCP Throughput at Different Speeds

Rel. Speed	HW Bitrate	Duration	Throughput
60 km/h	auto	50s	21 MB
100 km/h	auto	29s	12 MB
180 km/h	auto	18s	7 MB
240 km/h	auto	13s	6 MB

Table 3.8.: Mobile Testbed: UDP Throughput at Different Speeds

Figure 3.17 shows the results of a TCP throughput measurement at a very slow relative speed of 15km/h. Please note that relative speed in the oncoming traffic scenario means the sum of the individual velocity of the two vehicles. For the measurement shown in figure 3.17, the hardware bitrate of the Wireless LAN cards has been set to 1Mbit/s. An interesting observation that can be made even in this low-speed scenario is the asymmetric duration of the connection. The two nodes have to be relatively close before a connection can be established. If the distance between the two nodes is increased again, the distance before the connection breaks is higher than the distance necessary to initiate the connection. The asymmetric behavior has also been observed in scenarios with even lower speeds (relative speed of 16km/h). This leads to the conclusion that not the WLAN or TCP protocol is responsible for this but the hardware itself. A reason may be an improved sensitivity as soon as some error-free

packets have been received.

Figure 3.18 shows the results of a measurement at a relative speed of 240km/h. As can be seen clearly, IEEE 802.11b seems to be insensitive to changes in relative speed of the mobile nodes up to about 240km/h. Also the automatic setting of the hardware bitrate of the WLAN cards works well and results in a higher overall throughput than a fixed setting.

So far, all results presented have been based on a somewhat idealized environment: line-of-sight between the vehicles, no obstacles (e.g. other vehicles) on the road and a relatively large, free area around the runway where the measurements have been done. This environment has been chosen to get a good impression of the impact of node velocity on IEEE 802.11b performance with as little disturbing effects as possible. However, as the good performance in idealized high speed scenarios became clear, we have been very interested in 802.11b's performance in more difficult scenarios. For this purpose, a bending road with frequent traffic through the forest has been chosen for additional measurements in an oncoming-traffic scenario. The results are presented in figure 3.19.

3.4. Conclusion

This chapter has shown the limited scalability of MANETs with respect to per node throughput and other performance constraints. Necessary additional complexity is added by self organizing routing algorithms. The complexity of different routing classes including position based algorithms has also been discussed in this chapter. However, based on hardware available today (IEEE 802.11b), it can be concluded that even highly mobile MANET scenarios can be realized if some issues are taken care of (e.g. AODV setup, 802.11 MAC, limited number of active nodes).
Given the existing performance constraints, the necessity for efficient protocols becomes clear. As will be seen in the next chapter, this problem is hard to solve when it comes to secure communication in MANETs.

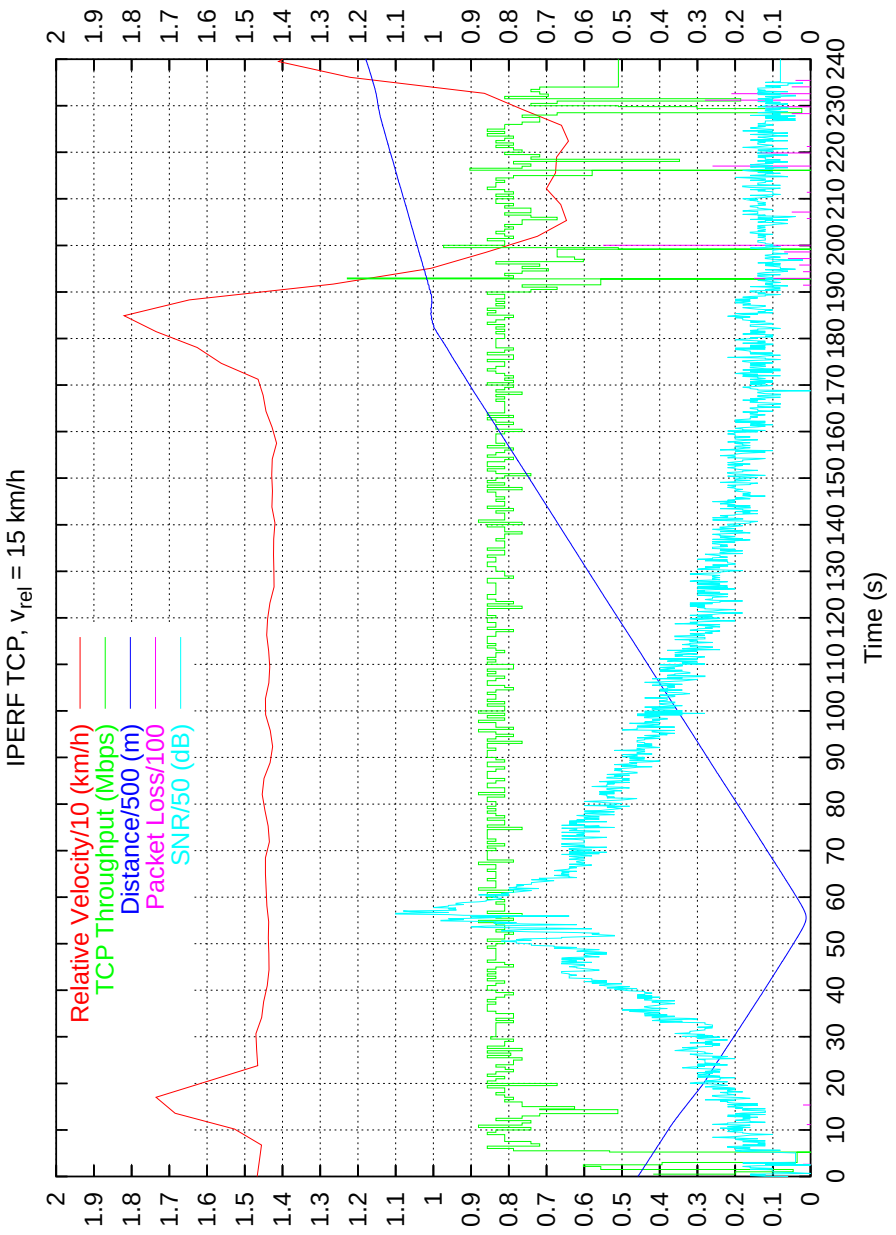


Figure 3.17.: TCP Throughput, v_{rel}=15km/h, Bitrate=1Mbit/s, Idealized Environment

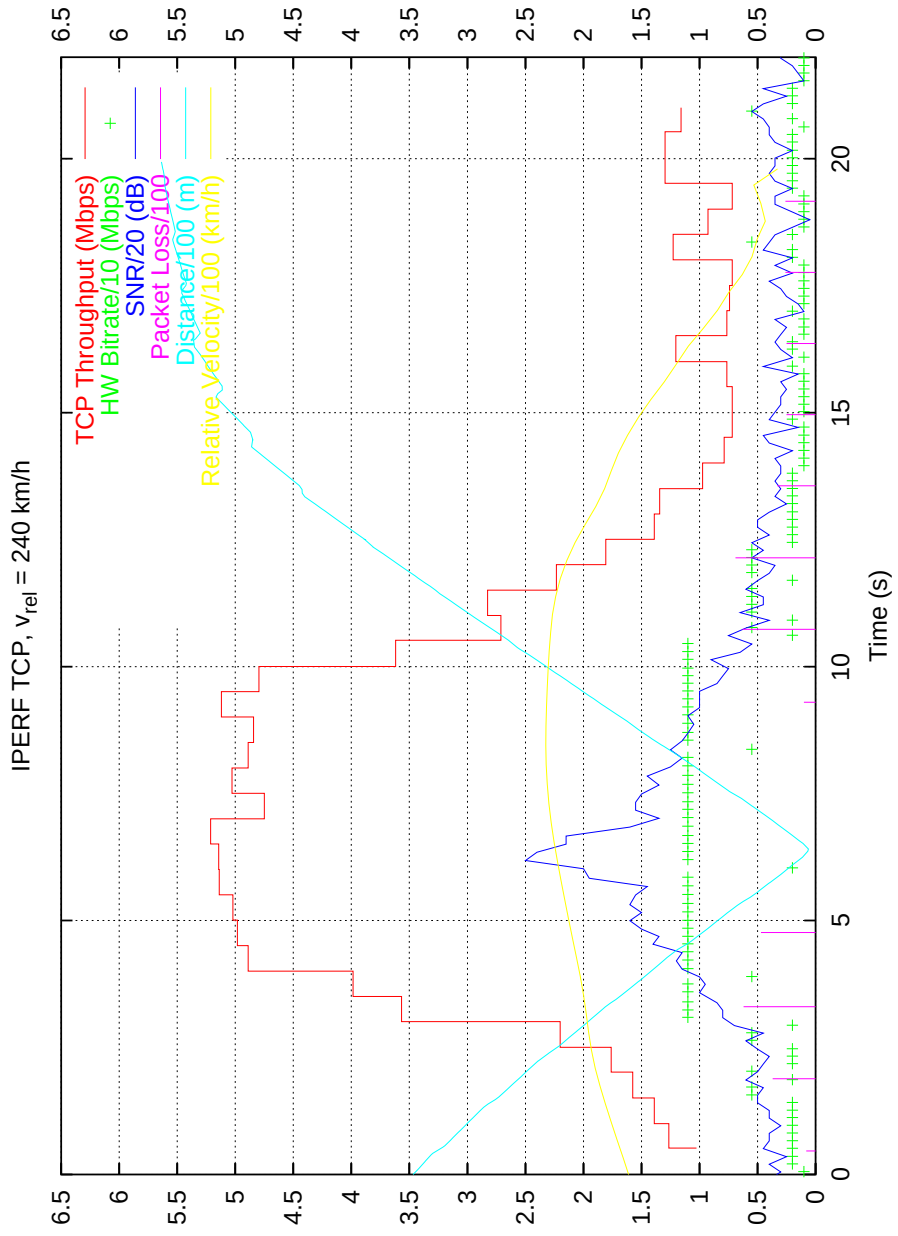


Figure 3.18.: TCP Throughput, v_{rel}=240km/h, Bitrate=Auto, Idealized Environment

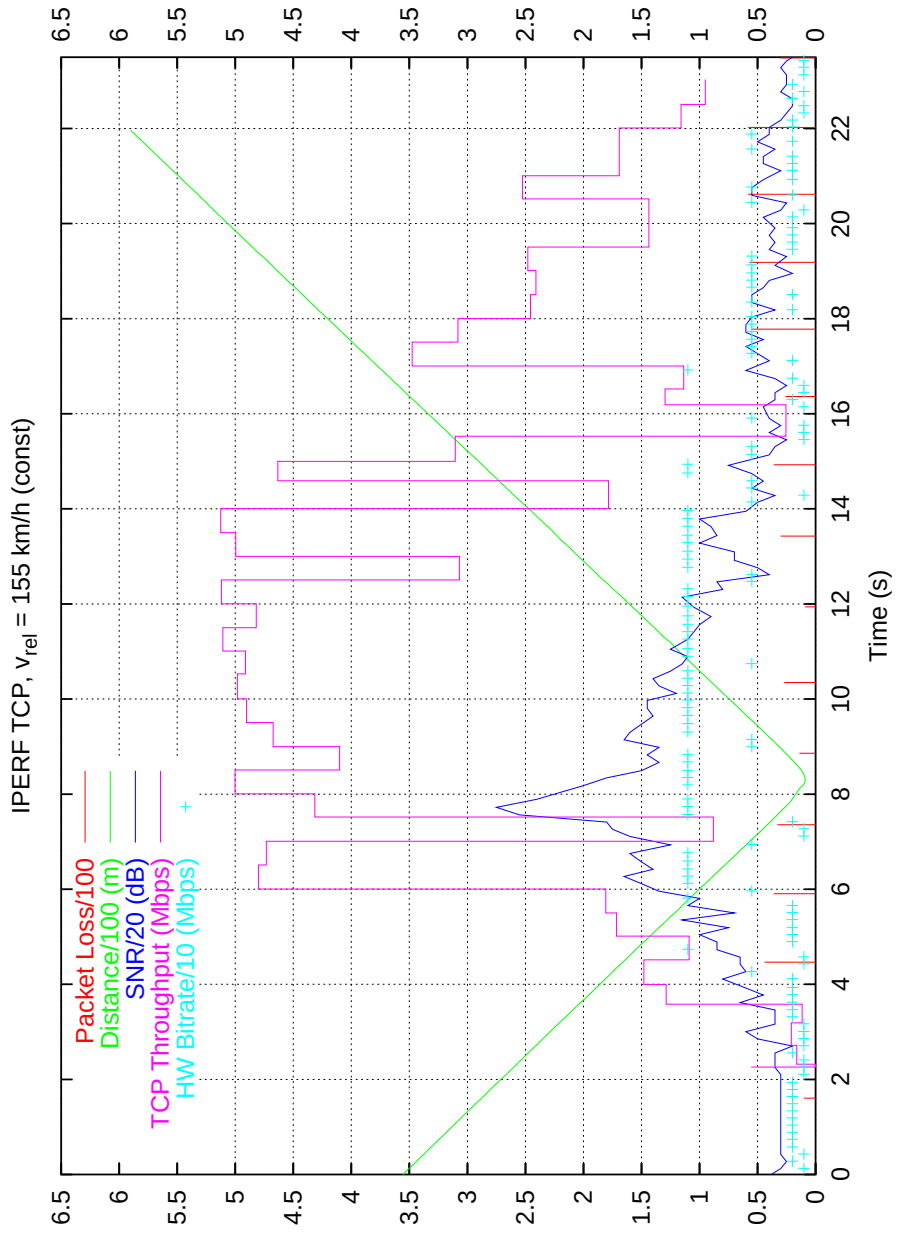


Figure 3.19.: TCP Throughput, v_{rel}=155km/h, Bitrate=Auto, Highway-Environment

4. New Requirements for Secure Communication in Mobile Ad Hoc Networks

In the introduction of this thesis, the development of security vulnerabilities and security incidents based on data from CERT have been shown. The dramatic increase of these numbers emphasizes the necessity for robust security mechanisms. Some basic security mechanisms should even be deployed in networks without security-critical applications and sensitive data. If not, very simple denial-of-service (DoS) attacks can render single services and even the complete network unusable.

MANETs are especially vulnerable to very effective DoS and other attacks. However, among other problems, its performance characteristics that have been discussed in chapter 3 make it hard to find suitable security mechanisms for such networks. This chapter explains the differences between security mechanisms for conventional wireless networks like Wireless LAN and security mechanisms for MANETs. Starting with security on the routing layer, also higher layer topics are addressed in this chapter. In particular, requirements and mechanisms for secure service discovery are presented. Finally, based on a summary of existing work on MANET security, necessary prerequisites for scalable and secure operation of MANETs are discussed.

This chapter starts with an introduction into secure communication in section 4.1. Beginning with a brief overview about secure Internet communication in section 4.2, the challenge of secure communication in wireless networks is explained in section 4.3. As an example, Wireless LAN security is discussed in this section. Specific threats faced by MANETs are presented in section 4.4. This section also presents countermeasures - building blocks for secure communication in MANETs. Also higher layer MANET security issues are discussed based on the service discovery protocol SLPv2.

4.1. Security Services

"Security" of a system can be seen as composition of one or more security services that are deployed corresponding to the systems security requirements:

- **Availability:** The availability of the offered services for authorized parties has to be ensured.
- **Confidentiality:** Data handled by the system has to be protected against attacks. Usually passive attacks (e.g. eavesdropping) are used to compromise confidentiality.

- **Integrity:** Data handled by the system has to be protected against alteration by unauthorized parties.
- **Authentication:** It has to be ensured that each entity is the entity it claims to be.
- **Non-repudiation:** Any party has to be prevented from denying a transmitted message. This means the receiver can prove that the alleged sender sent the message.
- **Privacy:** This requirement is often contradictory to the goals of other security services. Privacy basically means the protection of various personal data against disclosure. This can include the protection of single data items or the ensurance of privacy protection in presence of third parties that combine available data items to get additional information via its correlation. Also the nondisclosure of the users current location or even the usage of the service itself may be desirable.

The following attacks are not specific to MANETs but are applicable in almost any network. However, these different active and passive attacks give a good first impression about what can be expected in the MANET scenario with its numerous vulnerabilities and constraints.

1. Active attacks against the network:
   - Insertion of malicious routing information – certain nodes can be excluded from the network and traffic can be forced to pass one specific node to make eavesdropping attacks more effective.
   - Denial of service attacks – brute force attacks like extensive flooding or physical jamming as well as more subtle attacks to exclude single nodes or groups of nodes from the network.
2. Active attacks against users and applications:
   - Insertion of fabricated or modified packets – the whole range from false data to false application signaling.
   - Replay of packets – e.g. replay of authentication protocol messages or man-in-the-middle attacks.
   - Unfairness – dropping packets from specific users.
3. Passive attacks against the network:
   - Eavesdropping of signaling data – gathering of information that can be used in active attacks.
4. Passive attacks against users and applications:
   - Eavesdropping – gathering user data as well as signaling (e.g. authentication) information.
   - Traffic analysis – identifying important nodes on the basis of traffic quantity (who is talking to whom and how often).

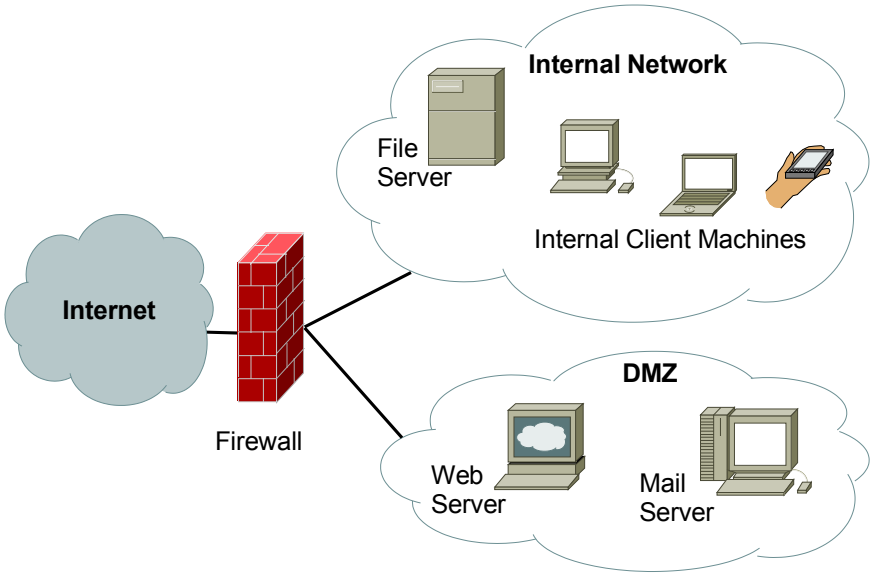


Figure 4.1.: Schematic Network Architecture with Firewall

4.2. Internet Security

Despite the relatively high number of security incidents on the Internet, basic security concepts for this environment are clearly defined. Most security incidents in this environment are happening because these concepts are not deployed or deployed wrong, by implementation errors (e.v. buffer overflows) that can be exploited or by wrong user behavior (e.g. download of malicious software). However, as a first line of defense, the network can be separated into different subnetworks, each part having a different trust level. Also, clearly defined policies (e.g. access restrictions) can be enforced in the subnetworks and between them. Nodes have clearly defined functions that make a differentiation of security policies possible. Thus, the main advantage of this environment is the possibility to separate ones network, even if it consists of subnetworks with different trustlevels, from the insecure Internet. Figure 4.1 shows a schematic picture of this concept.

4.3. Wireless Security

The advent of WLANs did not only make networking more convenient. It did also result in a number of serious security issues. As pointed out before (see figure 4.1), network security often relies on a firewall as a first line of defense. The installation of WLANs did seriously undermine this concept. While the firewall still offered protection against attacks from the Internet, direct access was now possible from outside the previously protective building. Figure 4.2 shows the additional, unprotected access possibility. Various reports (e.g. [CP01]) show a widespread unawareness of that problem. Test measurements found and continue to find a large part of WLAN installations even without the most basic security precautions.

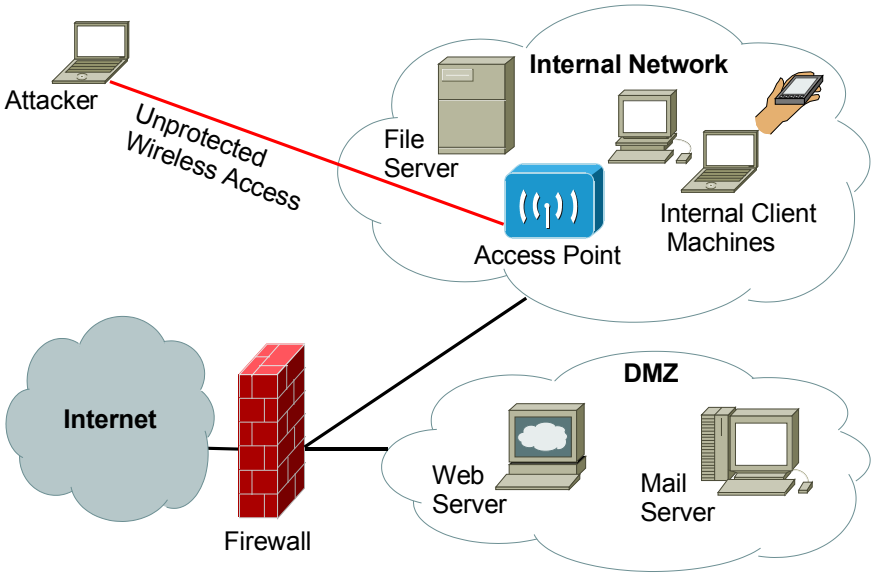


Figure 4.2.: WLAN: Implications on Network Security

For a secure integration of WLANs into existing networks, the Wired Equivalent Privacy (WEP) security concept has been specified within the 802.11 standard [Com99]. Its goal has been to provide the same level of security as in wired LANs, especially confidentiality, integrity and authentication. While the utilization of WEP is much better than running the WLAN without security precautions, it only provides very limited protection. Unfortunately, WEP did not become famous for achieving its goals but for achieving none of them.

To provide confidentiality and integrity, WEP uses the encryption algorithm shown in figure 4.3: First, an integrity check value *(ICV)* of the message M is calculated using the CRC algorithm. This checksum is concatenated to the message M to obtain the plaintext P. ICV and therefore also P do not depend on the key K. In the second step, the plaintext is encrypted using the RC4 algorithm [Sch96]. An initialization vector IV is chosen and after that a pseudo random key sequence PRN is generated by RC4, depending on the initialization vector IV and the key K. Then, via XOR (exclusive-or, $\oplus$), the plaintext P is mixed with the key sequence PRN to optain the ciphertext C. The third step is to transmit the ciphertext C, together with the unencrypted initialization vector IV.

Summarizing the three steps, the following data is transmitted via the wireless link:

$$A \rightarrow B : IV, (P \oplus RC4(IV, K)), where$$

$$P = \langle M, ICV(M) \rangle.$$

As depicted in figure 4.4, the decryption procedure is a simple reversion of the encryption

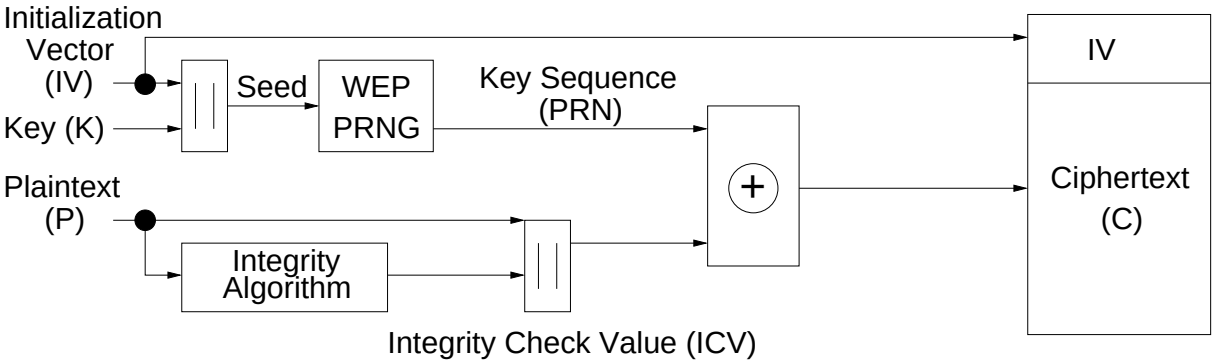


Figure 4.3.: IEEE 802.11: WEP Encryption

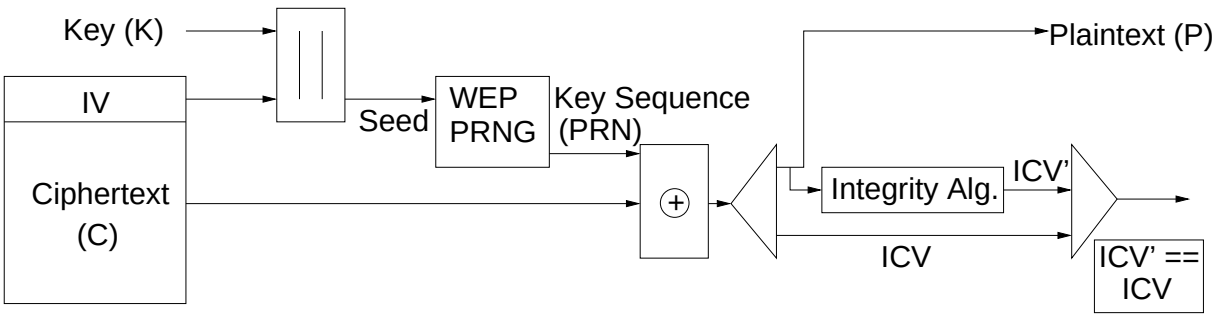


Figure 4.4.: IEEE 802.11: WEP Decryption

process. The recipient of the message knows the key[1] K and the initialization vector IV and has therefore all he needs for decryption:

$$P' = C \oplus RC4(IV, K)$$

$$P' = (P \oplus RC4(IV, K)) \oplus RC4(IV, K)$$

$$P' = P$$

The security of WEP is based on the problem of discovering the shared key via a brute force attack. WLAN cards capable of using WEP are available starting with keylengths of 40 bits. Even if a brute force attack is considered, this encryption can be broken without major problems ([Bec95], [Dol95]). A keylength of 104 bits that is available in the so called "128-bit" version of the WLAN cards, renders the brute force attack impossible. However, shortcut attacks on WEP exist [BGW01, Sch01] that profit from design flaws and are independent of the keylength. Despite this fact, it has been a widespread myth that longer keys help WEP security. The main error in the design of WEP was the handling and the length of the initialization vector IV. Given the fact that IV is included in the unencrypted part of the transmission, it can easily be read by anyone in range with 802.11b equipment. As WEP defines no key management, the shared secret key K that has to be in place on any machine accessing the WLAN via WEP changes very rarely. In most cases, K can be assumed to be static. Hence, if the attacker is able to intercept two messages with the same IV and he knows the plaintext of one message, the plaintext of the other message follows immediately. The standardized length of the IV of 24 bits does nearly guarantee the reuse of the same IV for different messages. To obtain known plaintext, it is possible to send an IP packet via the Internet to a mobile host within the WLAN installation. It is even simpler when the WLAN network is a mixture of clients with and without WEP. The access point has to send broadcast packets in encrypted and in unencrypted form. As the attacker also learns the key sequence PRN when he knows the plaintext of an intercepted message, he can use it do decrypt any other packet with the same IV. In [BGW01], the space requirements for a dictionary that includes all possible values of the IV are calculated. The result: 24GB are enough for the translation of the whole encrypted WLAN-traffic.
In addition to this attack, also message modification is possible. The WEP checksum is a linear function of the message. This implies that

$$ICV(x \oplus y) = ICV(x) \oplus ICV(y)$$

for all values of x and y. Thus it is possible to make arbitrary modifications to a ciphertext without possibility of detection. It is even possible for an attacker to modify a packet with only partial knowledge of its contents.
As the WEP checksum is an unkeyed function of the message and it is possible to reuse old IV values without triggering any alarms, WEP does also not provide secure access control.
Summarizing these facts, we can say that WEP does not manage to meet any of its goals: there is no confidentiality, no secure access control and no data integrity.

Ready-to-use tools are available to exploit these vulnerabilities, e.g. WEPCrack [wep] and AirSnort [air]. For this reason, other mechanisms than WEP are recommended for secure

[1] WEP does not specify how to distribute the keys

WLAN utilization. Better alternatives are the usage of common mechanisms like Virtual Private Networks (VPN) to ensure integrity and confidentiality and the usage of protocols like RADIUS or Kerberos [ARPK] for authentication and access control. Also the IEEE offers a new standard to overcome the WEP weaknesses. It is called IEEE 802.1x and relies on the Extensible Authentication Protocol (EAP) to provide both user authentication and a stronger method of implementing shared keys for WEP. It does not rely on the shared key to prove a users identity but provides user authentication instead. This can be done in various forms, e.g. passwords, digital certificates and tokens.

4.4. MANET Security

As outlined in the previous sections, the challenges to create secure environments are increasing from fixed networks to wireless networks. As network nodes are usually protected by the perimeter of a companies building in the fixed network scenario, security mechanisms can be built around this assumption. A basic separation between the unsecured Internet and sensitive, internal machines can easily be enforced. Different trust levels for different users can be enforced physically (i.e. different sections for employees, external partners, visitors). Even without multihop communication, the introduction of WLAN into this scenario requires greater efforts to create a secure system. The access points can potentially be accessed from anywhere within range, even from outside the building. Insecure implementations leave the internal network open to attacks via the access point, circumventing the firewall as primary defense shield. Also passive attackers can gain a lot of information by listening to plaintext transmission of sensitive data or by decrypting WEP protected traffic (this can be done with little effort as can be seen in section 4.3).

In this section, the security threats faced by MANETs are described. Requirements for MANET security mechanisms are highlighted and first approaches for secure communication in these environment are presented. If we regard the architecture of a typical MANET, security solutions face new difficulties:

- **Dynamic Nature:** The topology of ad hoc networks can change very fast, depending on the movement of the nodes. This results in regular changes of the availabe routes, changes in the reachability of different nodes and changing participants.
- **No central entities:** Due to the dynamic nature of the ad hoc network, the functionality of all nodes should be equal. Central servers are not recommended for most application scenarios because of the complete break-down of the service when the server becomes unreachable for any reason. Regarding security services, also the high physical vulnerability[2] of a single node should be taken into account.
- **Unidirectional Links:** Due to different power and transmission ranges of the nodes, unidirectional links can occur.
- **Constraints regarding:**
  - Processing power: Most of the scenarios for ad hoc networks assume mobile devices with small processors.

[2] i.e. it can be easy to gain physical access to a node and therefore to remove, compromise or destroy it

  - Battery power: If battery powered devices are used, the transmission power and the processor utilization will directly effect the battery lifetime.
  - Bandwidth: Is a scarce ressource in the wireless world. This is especially true for mobile ad hoc networks that have to cope with a lot of additional signaling information and also have to act as relay-stations for neighboring network nodes.

- **Scalability:** Due to the high demands on decentralization and selforganisation also scalability is an issue. Straightforward security mechanisms can have a high negative impact on system scalability.

Very often, user-convenience or system-performance are used as arguments for limited use of security mechanisms in networks. Given the constraints above, these arguments are especially important for MANETs.

4.4.1. Specific Threats against MANETs

Given the very different application scenarios of ad hoc networks, a wide range of security requirements exists. Application scenarios start with very basic things like communication between mobile telephones and headsets[3]. More demanding scenarios involve multiparty communication e.g. participants at a conference, sharing presentation slides and other information with their Laptops and PDAs. Other scenarios involve more security-critical issues like wireless payment. Common scenarios include also larger MANETs, e.g. voice communication or also data-communication between cars. For application scenarios in the business-world wireless payment and the transmission and access of confidential data are the most demanding ones. In addition, also emergency and military application scenarios of ad hoc networks are very common. In these scenarios, also physical attacks against the network nodes have to be considered. This can result in serious security incidents as the attacker can get access to hard- and software known to the network and can possibly perform successfull authentication, eavesdrop messages or inject arbitrary or malicious data into the network.

Clearly, another variable factor in the selection of the right security architecture for ad hoc networks is the transmission technique used. If the network is based on very short-range transmission like IrDA [ird], the restrictions of the network interface itself already provide some basic protection. This does change given techniques like Bluetooth [blu](radio range either 10m or 100m) and becomes even more important in systems based on IEEE802.11 [Com99] with an approximate range of 200m and above, depending on the environment. Radios used for emergency and military scenarios do have even wider transmission ranges. The danger, however, is not necessarily limited to the range of the ad hoc network. Ad hoc gateways to the Internet are a common scenario.

In addition to the protection of payload data, another big security challenge arises for MANETs: the protection of the network itself. This is mainly due to the highly decentralized and self-organizing nature of such networks. Usually, any node acts as router and therefore is trusted by other nodes to provide correct information, necessary for the proper functionality of the network. This blind, mutual trust can have severe consequences in the

[3] some definitions of ad hoc networking include simple peer-to-peer communication, other definitions demand at least basic ad hoc routing functionality

presence of malicious nodes: very few attackers or even a single attacking node can heavily damage the networks capabilities.

In addition to attacks known from conventional wired and wireless networks, several new attacks are possible in MANETs. Throughout the literature, numerous different attacks have been presented. E.g. [KW03] gives an overview and a classification of different attacks with a focus on sensor networks. Most of these attacks can also be applied to MANETs, either directly or with only slight modifications. Below, a brief summary of different attack-classes is given. It is not intended to give a complete listing. Instead the principal threats faced by a MANET are illustrated, especially, the damage that can be done to the network by only a few nodes.

- **Spoofed, altered, or replayed routing information:** Usually every node in an ad hoc network acts as router, any participating node (insider) can mount such an attack. Effects include routing loops, attraction or distraction of network traffic, generation of false error messages, partitioning of the network, etc.
- **Selective Forwarding:** Malicious nodes drop messages instead of forwarding them further along its path. More subtle forms of this attack that make detection very hard if not impossible include the selective dropping of certain packets only. This attack works best if the attacker is directly included in the forwarding path. Malicious nodes can achieve this e.g. using the sinkhole attack or, especially also with routing protocols that utilize multiple routes to the destination using the sybil attack.
- **Sinkhole Attack:** This type of attack typically works by making a malicious node looking especially attractive to surrounding nodes with respect to the routing algorithm. The goal is to lure as much traffic as possible from a particular area through a compromised node, creating a sinkhole with the adversary at the center. Despite this is a general attack applicable to a wide range of networks, it is especially effective in networks with a special communication pattern (e.g. sensor networks or ad hoc networks near an Internet gateway where lots of packets share the same destination). Figure 4.5 illustrates this attack.
- **Sybil Attack:** This class of attacks can seriously undermine the effectiveness of diverse fault-tolerant and distributed schemes created to improve performance, reliability and security in ad hoc and peer-to-peer networks. In a Sybil attack, a single node presents multiple identities to other nodes in the network. [Dou02] describes these attacks that can also be applied to location aware systems in more detail, an illustration of a sybil attack is shown in figure 4.6
- **Wormholes:** Wormhole attacks [HPJ02b] typically involve at least two cooperating adversaries. An adversary tunnels messages from one part of the network usually via an low-latency out-of-bound channel to another part of the network where they are replayed. These attacks can, among others, be used to distort routing, create sinkholes and to exploit routing race conditions and work even in the presence of authenticated and encrypted routing information.
- **HELLO flood attack:** This type of attack has been introduced in [KW03]. It is applicable against all protocols that use some sort of HELLO messages to form neighborship-

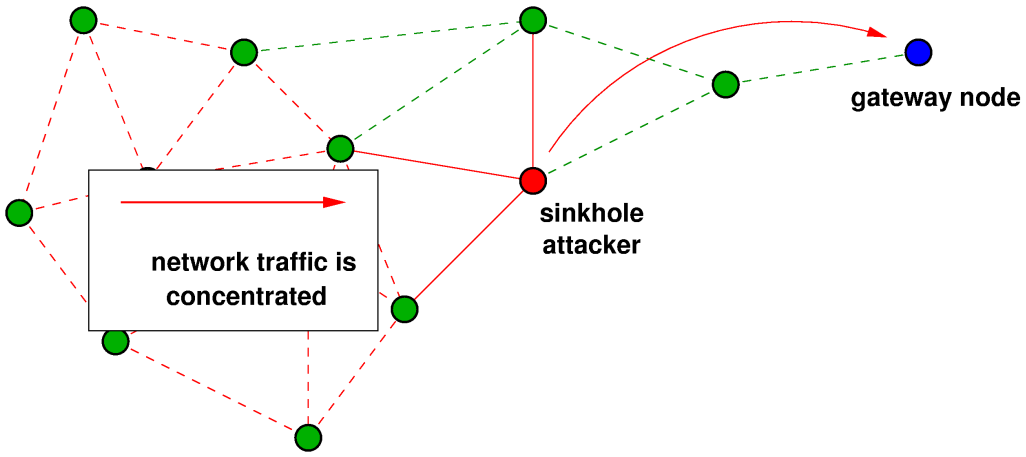


Figure 4.5.: The Sinkhole Attack

relations among the nodes. A malicious node can send (or record and replay) HELLO-messages into the network with high transmission power and therefore convince every node in the network that he is its neighbor. This attack leaves the network in confusion as most nodes simply send their packets into oblivion.

- **Acknowledgment spoofing:** This attack is applicable against protocols that use link layer acknowledgments. Spoofed link layer acknowledgments can convince nodes that a weak link is strong or that disabled nodes are alive. Among others, this technique can be used to mount selective forwarding attacks.

4.4.2. Building Blocks for Secure Communication

This section provides an overview about building blocks for secure communication in MANETs. Especially techniques to tackle the vulnerability of single network nodes are regarded:

- If security relevant functions are provided by only one server node, a simple denial of service attack on this node[4] will effect the security of the whole network.

- Due to their low physical protection, ad hoc nodes have a higher risk to become compromised. Techniques that require a benign operation of *all* nodes are therefore inappropriate.

As pointed out in [ZH99], one of the first articles about MANET security, secret- and function sharing protocols provide characteristics that can help to solve these security challenges. In the following paragraphs, a brief overview of these building blocks is given whereas a more detailled discussion of this topic can be found in e.g. [SH02] and [Hor01].

[4]that can be a physical attack/destruction of the node itself

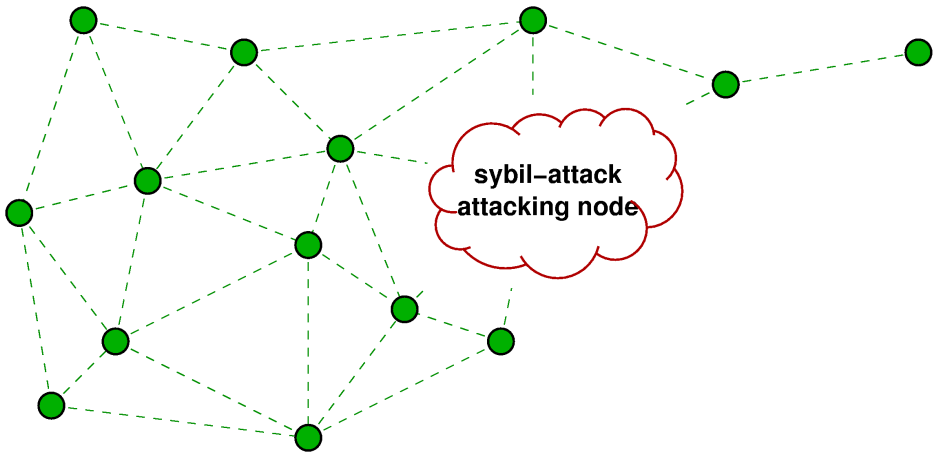


Figure 4.6.: The Sybil Attack

4.4.2.1. Secret Sharing

Secret sharing is defined as distribution of a secret s over n parties. To reconstruct the secret s, $t \leq n$ parties are necessary. $t-1$ malicious parties can not gain additional information about the secret s when they combine their shares. Secret sharing techniques are especially suited for ad hoc networking scenarios because

- the secret is protected, even if an adversary has access to max. $t-1$ shares.
- the secret remains available, even if shares get lost or are temporarily unavailable for some other reason.

Let n be the number of available shares. t the minimum number of shares that are necessary to reconstruct the secret. S is defined as the "secret space", i.e. the set of all possible secrets. A secret sharing protocol needs the following participants:

- A dealer who knows the secret and performs the splitting to get the secret-shares,
- shareholders who receive the secret-shares from the dealer and a
- combiner, i.e. a party that reconstructs the secret from the secret-shares.

Figure 4.7 (a) shows how a secret is shared and figure 4.7 (b) shows how it is reconstructed again. $s \in S$ is the secret, s_i a secret-share and s^* denotes the reconstructed secret.
The point made before that the secret is protected even if an adversary has access to max. $t-1$ shares can be expressed more strictly with the following formula:

$$P_D(s|(s_{i_1}, \ldots, s_{i_{t-1}})) = P_D(s) \tag{4.1}$$

P_D denotes any probability distribution D over the secret set S, s_i denotes a shared secret and $i_1, \ldots, i_{t-1} \in \{1, \ldots, n\}$.

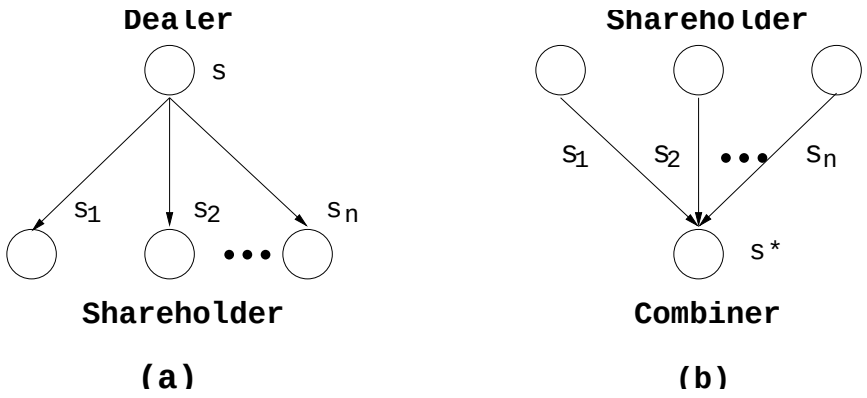


Figure 4.7.: Secret Sharing Protocol - Sharing and Reconstruction

Problems arise when a shareholder delivers a faulty share for the reconstruction. Mechanisms for the verification of shares would be an advantage. An even more difficult problem is to verify the correctness of a share delivered by the dealer. Two more problems for practical use are the separation of dealer and combiner and the possibility to securely reconstruct the secret more than once.
During the last years, solutions for these problems have been found. "Verifiable Secret Sharing" deals with the verification of shares - shares that come from the dealer as well as shares that are delivered by the shareholders to the combiner. The separation of dealer and combiner and the multiple secure reconstruction of the secret is addressed in "Function Sharing Protocols". These types of protocols are introduced in the next two subsections.
Some approaches propose to completely remove the dealer. One possibility is the distributed generation of the shares by the shareholders themselves without making the secret available to any party. More on this topic can be found in e.g. [BBWBG98] and [BF97].

4.4.2.2. Function Sharing

The term function sharing was introduced in 1994 by De Santis, Desmedt, Frankel and Yung in [SDFY94]. It basically is an analogon to secret sharing: A backdoor is opened and a function can be calculated if partial results from at least t out of n participants are available. However, there is one important difference to secret sharing: the cryptographic function can be computed multiple times without reducing system security. The cryptographic function is not made public at any time, only partial results and the result for a given input. In secret sharing algorithms the secret is made available at least to the combiner and should not be used any more after that. Figure 4.8 shows the distributed signing of a document using function sharing:
Every participant can access the plain data and computes a partial result with his share of the function. If a combiner has t out of n possible partial results, he can calculate the final result. No secure channels are necessary between the shareholders and the combiner. Also, the combiner has not to be completely trustworthy. This results from the fact that the final result can easily be verified. A fact that makes the design of a secure function sharing protocol much easier.

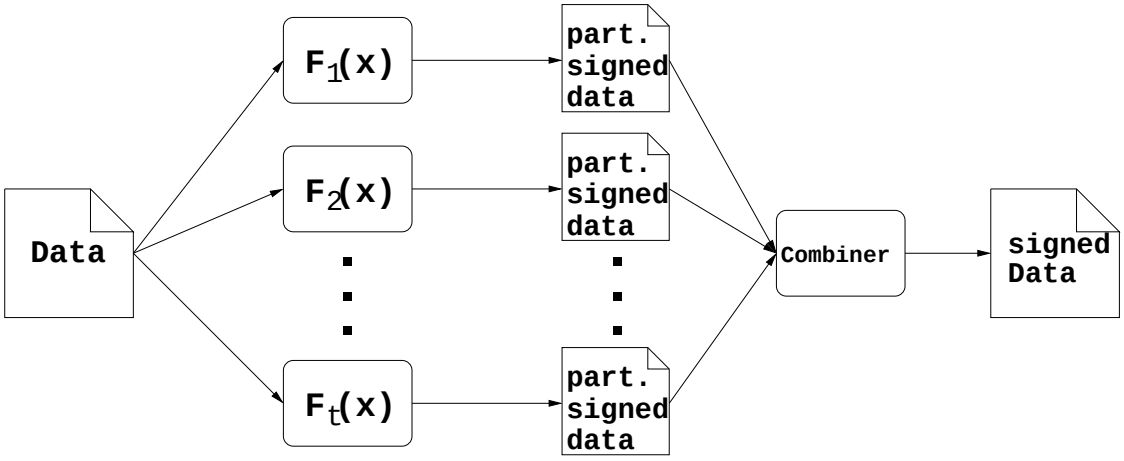


Figure 4.8.: Distributed Signature with Function Sharing

4.4.2.3. Verifiable Secret Sharing

Verifiable secret sharing is an enhancement of ordinary secret and function sharing protocols. Usually, a so called "witness" is created by the dealer for every shareholder. If necessary, the combiner can verify each share or each result from a partial function using the witness. Faulty shares can be detected with this technique and the faulty or malicious shareholders can be isolated. Verifiable Secret Sharing protocols are divided into interactive and non-interactive protocols. Interactive protocols have a separate communication protocol to verify the partial results, in non-interactive protocols the combiner knows the witnesses directly.
If we imagine a (t, n) Secret Sharing protocol with e.g. $t = 4$ and $n = 10$ we can see the importance of the possibility to verify the shares: 210 possible combinations of parties that can create a valid signature (compute a valid result from their shares) are possible. If only 3 parties are compromised and deliver faulty shares, only 90 possible combinations are left. If the single shares can not be verified, the combiner has to try out different combinations until he finds a correct one. As the combination of shares is computationally expensive, malicious shareholders have an easy possibility to launch denial-of-service attacks if they can not be identified.

4.4.2.4. Proactive Secret Sharing

In proactive secret sharing protocols it is possible for the shareholders to refresh their shares within given time-intervals. This results in a substantial increase in security, especially for systems with long operating times. An adversary has to compromise at least t shareholders within two resharing cycles to be successfull. When a refresh is done, the "network secret[5]" stays the same, only the shares are changed. With proactive secret sharing, the adversary can compromise all nodes of the network during the system lifetime without being successfull. To get control over the network, the adversary has to compromise more than $t - 1$ nodes during the time interval between two refresh cycles[6]

[5] i.e. the final secret that we get by combining at least t shares
[6] this time interval is often called "Window of Vulnerability".

4.4.2.5. Protocol Properties

The most important features of such building blocks include:

- **Security:** A threshold cryptosystem is said to be secure if $t-1$ compromised or faulty nodes are not able to fake a signature with a possibility of $1-\epsilon$,($\epsilon < 1$).
- **Robustness:** A threshold cryptosystem is said to be robust if $t-1$ compromised or faulty nodes are not able to forestall the generation of a signature.
- **Optimality:** A threshold cryptosystem is said to be optimal if it is robust for the maximum possible number $f = \lfloor \frac{n-1}{2} \rfloor$ of compromised or faulty nodes.
- **Efficiency:** A threshold cryptosystem is often said to be efficient, however, no common definition of this feature was found by the author. Efficiency is used when the cryptosystem behaves reasonable with regard to memory usage and computation time.
- **Interactivity:** A threshold cryptosystem is said to be interactive if the parties have to communicate to generate a signature. E.g. some threshold cryptosystems include interactive sub-protocols for verifiable secret sharing. Also proactivation-protocols are highly interactive. An example for a non-interactive protocol is the function sharing protocol in Figure 4.8.
- **Synchroneity:** A threshold cryptosystem is said to be synchron if it defines hard limits on the maximum delay for transmitting or processing of messages.

Proactivation Protocols are highly interactive. Each node has to communicate with each other node to assign a new sub-share and the necessary witnesses to it. Given the dynamic nature of ad hoc networks, one can not assume the availability of all n nodes for performing the share refresh. This leads to the demand for a mechanism that does a refresh step-by-step. This results in the situation that there are no strictly defined time periods during that new shares can be assumed. Even worse, malicious nodes can increase the "window of vulnerability" if they hold back messages or simulate connection loss. It is very hard, if not impossible, to find the guilty nodes. How can the status of a node be detected if it doesn't participate in the proactivation protocol? Is the node out of range? Is the node damaged or should it be considered malicious? To complete the protocol, the nodes have to expire timers and exclude the nodes that do not answer from the system or mark them as faulty. The system does damage itself. This is the reason why all proactivation protocols have synchroneity demands on the network. Another problem in this context is that there is no global clock, but it is necessary for the nodes to come to a decision about the right time for a share-refreshing. Byzantine agreement protocols can be very useful to solve this problem.

Byzantine Agreement Protocols can be used to make correct distributed decisions in the presence of faulty or malicious parties. The adversaries can behave arbitrary without being able to hinder the correct parties from making their decision. Up to now, byzantine agreement protocols are widely used in fault-tolerant, distributed systems. It is possible to give an upper bound for the number of faulty nodes that the system can still tolerate and make the correct decisions. For a system with n participants, this upper bound is $f = \lfloor \frac{n-1}{3} \rfloor$.
A lot of publications about byzantine agreement protocols are available, but not all of them meet the demands for the utilization in ad hoc networks. Some of them are not efficient

enough, others assume synchroneity, have very long initialization-phases or are not optimal[7]. One publication of such a protocol that is also suitable for ad hoc networks is [CL99b]. An even more efficient version that does not work with digital signatures but does use message authentication codes instead is described in [Cas99]. The proof of correctness of this protocol was published separately in [CL99a].
Byzantine Agreement Protocols are highly interactive. They transmit a high number of messages to reach the same state and make the right decision. In ad hoc networks, such protocols can be used to enable "democratic" decisions. This is especially useful if nodes show a benign behavior, but do not always have the same opinion about its decisions. Byzantine agreement protocols ensure that a large number of nodes was behind the decision. To work smoothly, the parameters f and t have to be set reasonably (e.g. $t = n - f$). It is important for the usage in ad hoc networks to set the parameter t not too high. The real degree of availability of the nodes has to be kept in mind. One security problem of these protocols in ad hoc networks is obvious: the high interactivity makes the system vulnerable to various kinds of denial-of-service attacks.

4.4.3. Protocols for Secure Communication

Besides cryptography itself, careful protocol design is necessary because of the limited applicability of end-to-end security mechanisms. Almost all work concerning security protocols in MANETs has been focusing on the routing layer. This section gives an overview about this research. It summarizes the main developments and shows still unresolved issues for practical utilization of these protocols.

A good overview about different threat models and attacks can be found in [KW03]. Although it is focused on sensor networks, most of the techniques described are applicable directly or with slight modifications in different classes of ad hoc networks. Some traffic patterns described here for sensor networks might also be profitable in some scenarios of highly mobile vehicle-to-vehicle networks[8], introducing the same vulnerabilities also to this class of networks. Some ideas to secure parts of the ad hoc communication are presented, however, the paper concludes that no secure routing protocol exists so far and leaves this as an open problem. Besides cryptography itself, the necessity of careful protocol design is highlighted because the limited applicability of end-to-end security mechanisms.

A *Lightweight Hop-by-Hop Authentication Protocol* for ad hoc networks is proposed in [ZXSJ03]. It uses hop-by-hop authentication for verifying the authenticity of all the packets transmitted in the network and a one-way key chain (TESLA [PCTS00]) for packet authentication and for reducing the overhead for establishing trust among nodes. The authentication of packets helps especially to defend against traffic inserted by external (i.e. not part of the network) malicious nodes as these packets are not routed any more. However, also assumptions are made that are probably hard to meet in some ad hoc networks. For bootstrapping, a working PKI is assumed as the last value in the key chain, $K(0)$, has to be signed with the senders private key so that anybody who knows its public key can verify the signature and hence the authenticity of $K(0)$. Furthermore, the receivers are required to be loosely synchronized with the sender - the upper bound of the synchronization error between any two nodes has to be

[7] i.e. they can not tolerate the maximum number f of faulty participants
[8] especially in-network processing, aggregation, duplicate elimination, etc.

known. TESLA itself includes that receivers cannot authenticate packets immediately on its arrival. One of its variants [PCST01] allow an immediate authentication, however, requires packet buffering at the sender side.

A security-aware ad hoc routing algorithm is presented and its applicability to AODV is shown in [YNK01]. The goal is to characterize and explicitly represent the trust values and trust relationships associated with ad hoc nodes and to use this values to make routing decisions. Only nodes that provide the required level of security can generate or propagate route requests, updates, or replies. If one or more routes exist that satisfy the required security attributes, the shortest of these routes is found. However, a nontrivial prerequisite is required for this protocol, namely an appropriate key management scheme for ad hoc networks.

In [SDL+02], a protocol called "authenticated routing for ad hoc networks (ARAN)" is presented. Also some specific attacks against DSR (Dynamic Source Routing) and AODV (Ad Hoc On Demand Distance Vector) are described. ARAN introduces authentication, message integrity and non-repudiation into the routing protocol - protecting against malicious actions by third parties and peers in certain ad hoc environments. The security of this protocol is based on certificates, a preliminary certification process that guarantees end-to-end authentication has to be performed before ARAN can start working.

A route discovery protocol which guarantees that fabricated, compromised, or replayed route replies would either be rejected or never reach back the querying node is presented in [PH02]. The protocol is capable of operating without the existence of an on-line certification authority, however, a security association (SA) between the source node and the destination node is assumed. Such trust relationships typically are instantiated via public key cryptography.

In [KHG03], an interesting, new proposal for anonymous routing is presented. The approach, that guarantees route anonymity and location privacy, combines broadcast with trapdoor information. This idea is used on top of the well-known MIX design, introduced by Chaum to achieve an anonymous email solution [Cha81]. Using the proposed protocol (ANODR, ANonymous On Demand Routing), an enemy can neither link network members' identities with their locations, nor follow a packet flow to its source and destination. Ad hoc routing is dissociated from the network member's identity. ANODR also ensures there is no single point of compromise. Location privacy of participating nodes is not compromised by a single node intrusion - an on-demand route is traceable only if all forwarding nodes on that route are intruded. A performance analysis that shows room for improvement especially in highly mobile scenarios is presented. Security features besides anonymity are outside the scope of [KHG03]. However, end-to-end security protocols (e.g. SSL/TLS, host-to-host IPSec) are proposed for this purpose.

Wormhole attacks and its implications on some routing algorithms are introduced in [HPJ03]. Basically, the defense is based on so-called packet leashes. Geographic and temporal leashes are also considered in [HPJ03]. Geographical leashes are used to ensure that the recipient of a packet is within a certain distance from a sender, temporal leashes allow to ensure an upper bound on packet lifetime. The TIK protocol presented in this paper implements temporal leashes. One necessary prerequisite for TIK is that a node can obtain an authenticated key for any other node.

A routing protocol that functions in the presence of byzantine failures is presented in [AHNRR02]. This includes nodes that drop, modify, or mis-route packets in an attempt to disrupt the routing service. Links are assigned different weights according to its stability (i.e. links that fail often will be assigned higher weights) and an on-demand route discovery protocol is used to find a least weight path to the destination. The protocol consists of the following three phases:

- **Route discovery with fault avoidance:** A least weight path from the source to the destination is found using flooding and a faulty link weight list.
- **Byzantine fault detection:** Faulty links on the path from source to destination are detected. Be n the length of the path, the technique proposed in [AHNRR02] finds a faulty link after $\log n$ faults have occurred.
- **Link weight management:** Links are penalized with high weights if faults occur and are rehabilitated over time. This information is used in the route discovery phase to avoid faulty paths.

Pairwise shared keys that are established on-demand are required by the protocol. A PKI, either distributed or Certificate Authority (CA) based is assumed for this purpose.

Ariadne, a secure on-demand routing protocol for ad hoc networks is presented in [HPJ02a]. It prevents attackers or compromised nodes[9] from tampering with uncompromised routes consisting of uncompromised nodes and also prevents some types of Denial-of-Service attacks. The protocol only uses highly efficient symmetric cryptographic primitives, a simulation-based performance evaluation is given that compares Ariadne with two versions of DSR: the standard DSR protocol and, for fairness reasons, a version of DSR with all optimizations not present in Ariadne disabled. Different types of authentication mechanisms supported by Ariadne require different setup preparations:

- **Pairwise shared secret keys:** A mechanism has to be in place to set up the necessary $n(n+1)/2$ keys in a network with n nodes.
- **TESLA [PCTS00]:** Shared secret keys between communicating nodes and one authentic public TESLA key for each node have to be set up.
- **Digital Signatures:** Authentic public keys, one for each node have to be distributed.

4.4.4. Summary Contermeasures

The countermeasures proposed above help to defend against specific threats against ad hoc networks. In practical implementations, usually some sort of trust distribution has to be done before the network can be used securely. The main problems of these protocols operating stand-alone include either the scalability (no new nodes can be included in the network during operation) or the vulnerability against certain attacks (e.g. sybil attacks against threshold schemes). To the best of my knowledge there seems to be no way yet to get both,

[9] the protocol provides security against one compromised node and arbitrary active attackers

scalability and security in a fully distributed stand-alone ad hoc network.

With the LKN Ad Hoc Security Framework (LKN-ASF) that will be presented in chapter 5, a scalable, certificate-based security framework has been realized that can be used in any MANET with at least temporary access to one or more gateways. The possibility to combine LKN-ASF with the AODV routing protocol has been shown. A related performance evaluation can be found in chapter 6. In addition to that, the LKN-ASF framework can be used to provide an infrastructure for the implementation of most of the existing countermeasures in a scalable way.

4.4.5. Higher Layer Security - Service Discovery

While securing the routing layer of a MANET is of utmost importance, security issues also arise at higher layers. This is due to the fact that communication peers, services and even system components are usually selected dynamically based on its availability. Summaries of the security requirements of such environments and first proposals for appropriate security mechanisms can be found in [Sta02] and [HMSU03].
As service discovery is a fundamental mechanism for most selforganizing systems, this section focuses on security issues of SLP. After an introduction into the optional security mechanisms of SLPv2 [GPVD99], attacks are presented that are still possible even in the presence of these security mechanisms. Finally, proposals to avoid these attacks are given.

4.4.5.1. Security Mechanisms in SLPv2

As described in section 2.5, SLP can be used with and without a directory agent (DA). Usually, installations with a DA are preferable in fixed networks with heavy SLP utilization whereas systems without a DA are better suited for MANETs. However, even in MANETs, especially if they are operated within a 4G system, parts of the network may experience predominantly localized traffic. This may be the case around Internet gateways or within vehicles when various in-vehicle peripheral devices are networked with the drivers mobile devices and connected to outside networks via a communication server [KBS+01]. In such cases, SLP configurations with a DA may prove beneficial.

The optional security mechanisms of SLP consist of an authentication mechanism with timestamp protection against replay attacks. The authentication is based on asymmetric keys and enables SLP to guarantee that the received service information has been transmitted by trustworthy service agents (SAs) and DAs. To achieve this, the sender of an SLP message includes a digital signature which is calculated over selected parts of this message. [GPVD99] specifies the generation of signatures for URLs, attribute lists, DAAdvert and SAAdvert messages. As signatures are also included in service reply messages (SrvRply) to user agents (UAs), the UAs can be sure that replies to service requests come from trustworthy agents.

4.4.5.2. Possible Attacks

Despite the offered authentication mechanisms it has been shown that it is possible to launch attacks against SLP in certain situations [VBS01]. In addition to the different replay scenarios presented below, the authentication procedure itself may be problematic in certain situations.

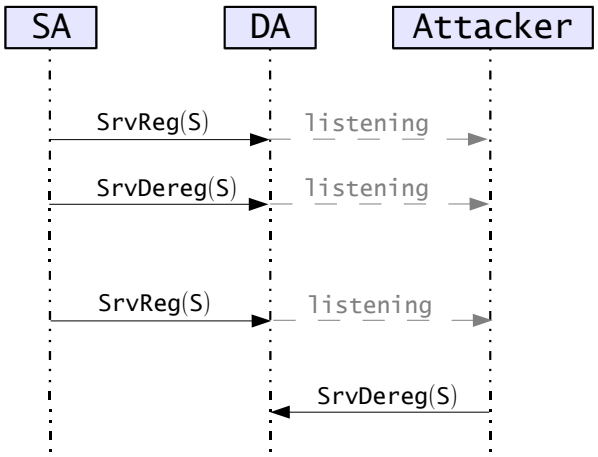


Figure 4.9.: SLP Attack: DoS against SA

Especially in the situation when UAs have to work in an unknown environment, the confirmation that a SrvRply message comes from a specific SA does not really help as the UA has no direct and simple way to check whether the SA really provides the advertised service. The UA still has to rely on the data provided by the SA whose authenticity is checked by the DA, however, also a foreign DA is not really trustworthy for a mobile client in an unknown environment.

Figure 4.9 shows a possible denial of service attack against a SA. It is applicable when a SA deregisters its service and wants to re-register it again later. If an attacker eavesdrops the service deregistration (SrvDereg) message, it can fake the identity of the SA and replay this SrvDereg message.

Another attack that is possible by simple eavesdropping of the SLP traffic is the following: At time t_1, a SA registers the service S whose lifetime expires at time t_4. An attacker eavesdrops on the SrvReg message. When the service administrator of the SA decides to change some attributes associated with S and therefore re-registers the updated service, eventually with a different lifetime at time t_2, the attacker can reset the service to its old status. This includes the change of attributes as well as a possible change of the service lifetime.

An attack with potentially more severe consequences is presented in figure 4.10. A SA registers its service with an expiry time at t_10. An attacker is listening and records the SAs service registration. If the SA deregisters its service at a time before t_10, the attacker can impersonate the SA and offer its own service instead. This is especially easy for an attacker in a mobile environment, where the SA might deregister its service and move out of range. In this case the attacker can simply reuse the IP address of the SA and wait for incoming connections. An example for this attack is a printing service - if a user trusts this service and submits a confidential document, it will be received by the attacker. This attack is especially hard to detect if everything works as expected, i.e. the attacker can simply forward the document to a real printer after making a copy.

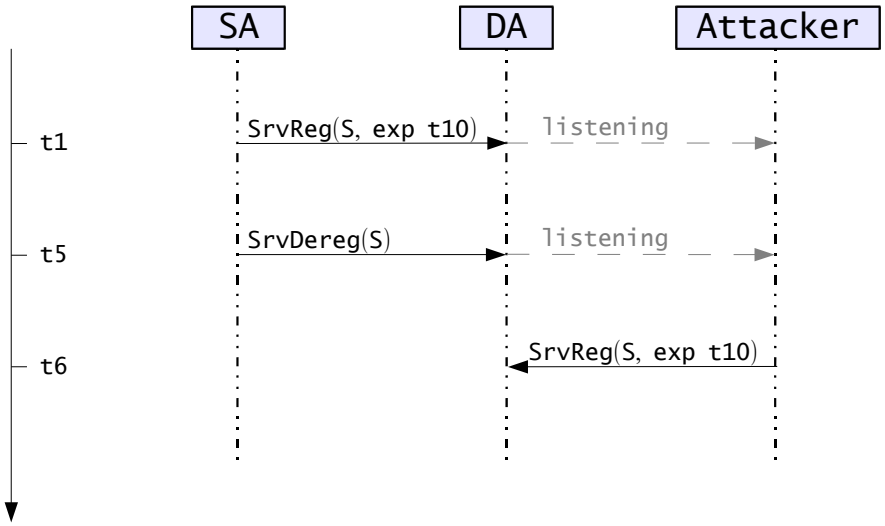


Figure 4.10.: SLP Attack: Impersonation

4.4.5.3. Countermeasures

To mitigate effects of the attacks described above, some simple modification of SLP can be made. While below a summary of these modifications and its implications is given, the original proposals have been published in [VBS01].

One first improvement is the usage of incremental lifetimes for SA Service advertisements. This means, the timestamp included in an authentication block should always be greater than the timestamp of the last calculated block. The DA should then cache the last timestamp of an authentication block and accept only incremental timestamps. In particular, the most recent timestamps should be cached separately for SrvReg and SrvDereg messages. Due to the additional rules for timestamps, the attacks described above can be avoided because the DA would reject the replayed packets. Another proposal in [VBS01] is to always deregister a service if changes have do be done instead of a simple re-registration. A key change during the deregistration process would make further attacks (e.g. the impersonation attack described in figure 4.10) impossible because the replayed SrvReg message will be detected at the DA.

4.5. Conclusion

While most work on protocols for MANET security focuses primarily on the important aspect of the security of its own design, the network performance aspect remains largely unaddressed. This resulted in various protocols with sophisticated security properties, however, it often remained unclear whether the assumptions taken to design the protocol can be met in real MANETs. As has been pointed out in chapter 3, the performance of network protocols is a critical issue in this environment with its limited scalability and scarce bandwidth re-

sources. Protocols with completely distributed operation and decision-making that have been presented in section 4.4.3 also require a relatively high communication overhead to achieve its goals. As the achievement of fault-tolerant consensus is known to be impossible in asynchronous systems (see e.g. [AW98]), it may even be not feasible at all to apply these algorithms to certain MANET scenarios.

Another important aspect of MANET security is the initialization of trust relationships among the nodes as well as the trust relationship between nodes in the network and new nodes that want to join the network. Often, manually preconfigured trust relationships are assumed. This mostly results in the impossibility to efficiently remove trust assignments for single nodes (revocation) and to add new, previously unknown nodes without heavily affecting the configuration of the remaining nodes.

Concerning higher layer security, some improvements for the service discovery protocol SLPv2 have been presented in section 4.4.5. Despite these changes, fundamental security issues remain: pre-configured trust relationships may be feasible for wired networks with only nomadic mobility and low user fluctuation. However, they are impractical in most MANET scenarios with very dynamic changes of communication peers and available resources.
Furthermore, a problem exists that is familiar from the Internet: The user can make a secure connection to a communication peer, e.g. `http://www.lkn.ei.tum.de`. He even is assured that he is really communicating with exactly this site, using strong cryptographic mechanisms. Most users are quite happy with this fact and send all kinds of confidential data to such sites, knowing they are really talking to `http://www.lkn.ei.tum.de` and no attacker can intercept and understand the encrypted data. This completely ignores the fact that `http://www.lkn.ei.tum.de` might be the attacker himself as nothing is known about this site except its address.
Especially in highly dynamic MANET environments it might therefore be advantageous to be able to trust certain attributes of a node instead of its name only. E.g. it can be certified that `http://www.lkn.ei.tum.de` provides a trusted Internet Gateway service and the service discovery user agent can then be configured to display only certified services to the user.

In chapter 5, a certificate-based security framework is presented that addresses the above issues for MANET scenarios that have at least infrequent access to Internet gateways.

5. LKN Ad Hoc Security Framework (LKN-ASF)

As has been pointed out in the previous section, completely distributed security mechanisms for MANETs have their limits. This is especially the case concerning the following points:

- The distribution of key material and initialization of trust relationships at system startup.
- Efficient introduction of new nodes and exclusion of misbehaving nodes from the network.
- At least limited control over the network by a service provider (who is allowed to take part, who is excluded from the network, from service access). In the distributed case, trust management is handed over completely to the network nodes.
- Network performance. While MANETs are especially sensitive to high traffic loads, distributed mechanisms need a lot of control traffic to achieve consensus.

Despite Ad Hoc networking is an important part of upcoming telematics systems and allows the efficient realization of new, location aware applications like local danger warning, MANETs are not seen as the only communication possibility in this scenario. Typical systems include several different network interfaces as transparent access to other networks combines the advantages of these technologies and extends the usability of the system [KBS+01].
This situation does not in any way mitigate the security risks faced by the network nodes when it comes to MANET communication. However, it is possible to use the additional network interfaces for the design of a MANET security mechanism. This can even be beneficial in scenarios with heterogeneous nodes, i.e. some nodes are equipped with several different network interfaces while other nodes do only support MANET communication.
LKN-ASF makes use of these advantages to create a framework for secure communication in MANETs. Figure 5.1 shows the network environment that is assumed as basis for LKN-ASF: One or more self-organizing MANETs with at least temporary access to a trusted party, typically located in a non-mobile network e.g. somewhere in the Internet. For compatibility reasons, LKN-ASF is based on the X.509 standard. This means that all protocols that support this open standard can make use of it. In addition, attribute certificates are available that can be used to arrange user groups and grant access rights to certain MANET nodes. Another use of these certificates is the secure confirmation that a node actually provides the services it pretends to provide. E.g. the functionality as Internet gateway can be certified as well as higher layer services like a printing service. While the former usage is essential for secure network operation, the latter usage of attribute certificates proves beneficial for higher layer security. E.g. problems with service location security that have been described in section 4.4.5 can be avoided. The secure operation of the system can be made much more user-friendly and easier to understand by the usage of attribute certificates. After an initial

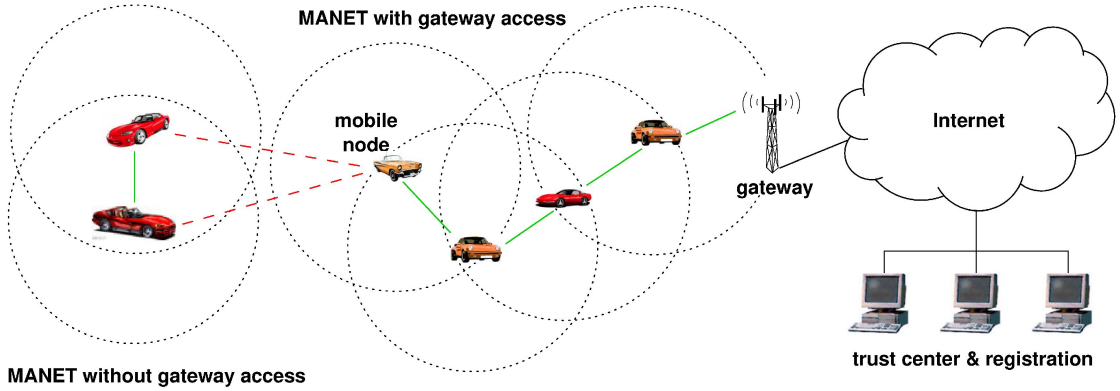


Figure 5.1.: Telematic Applications: 4G Network Environment

configuration of the required security level, the system will only present services that meet the required security level to the user.
LKN-ASF minimizes the usage of additional network interfaces as far as possible so it can be used even in scenarios with only sporadic access to the trusted authority. Its utilization allows secure communication in MANETs while important decisions concerning network security are not completely given to the participating nodes. This results in a lower communication overhead as the need for expensive distributed agreements is lowered. Also, service providers have the possibility to restrict access to certain services. Even the participation in the MANET itself can be restricted to subscribing nodes. This is especially useful not only to telematics systems but to all scenarios where secure user groups have to be supported. To guarantee a constantly high security level and the possibility to react to changes in the nodes trust levels, LKN-ASF supports certificate revocation as well as a periodic and triggered renewal of certificates. For performance reasons, i.e. to minimize signaling traffic among the MANET nodes as well as between the MANET nodes and the trusted third party, certificate caching is used.
LKN-ASF acts as a security framework only, i.e. it provides the necessary prerequisite for secure and efficient MANET communication. As it relies on the open X.509 standard, it can be coupled with almost arbitrary routing protocols and other mechanisms for secure communication. Parts of this ongoing work are described in chapter 7.

This chapter is organized as follows: section 5.1 presents the underlying assumptions concerning the network environment of LKN-ASF. Section 5.2 explains the functionality of LKN-ASF, especially certification and revocation in more detail. Details on the protocol implementation that has been done in SDL and the simulation model available for GloMoSim are presented in section 5.3. Finally, section 5.4 describes the establishment of secure connections using LKN-ASF.

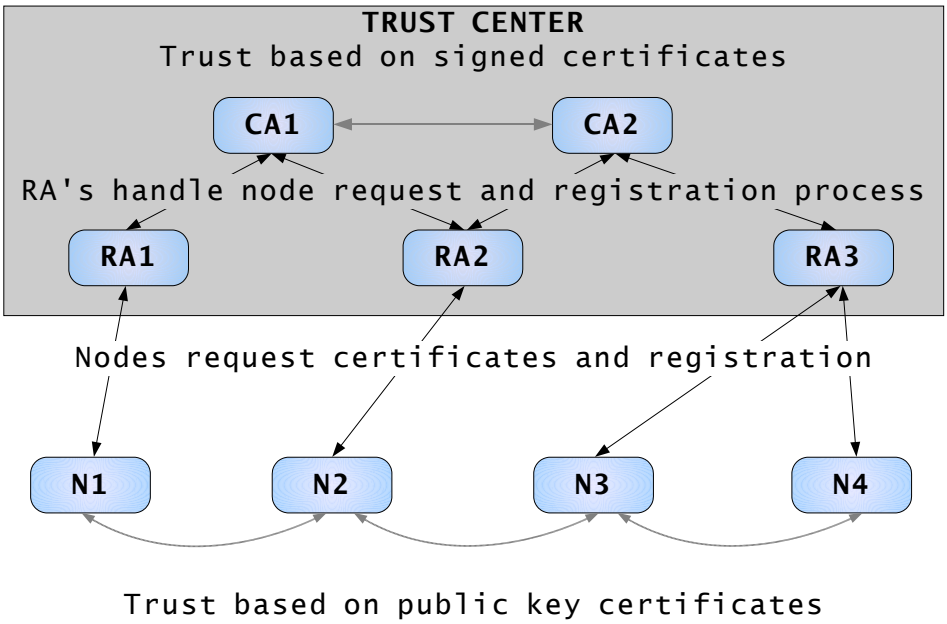


Figure 5.2.: LKN-ASF: Public Key Infrastructure

5.1. Network Environment

As outlined in the introduction of this chapter and shown in figure 5.1, LKN-ASF requires at least temporary access to a trusted third party, usually placed within a fixed network. While LKN-ASF can be implemented using various network interfaces, the descriptions within this thesis and also the performance evaluations presented in chapter 6 are considering local information points (LIPs) for any communication with the trust center. On the one hand, this scenario permits also nodes with only one network interface to fully use the security infrastructure. On the other hand, this is the most demanding scenario concerning network performance as all signaling messages are transmitted within the MANET. A similar mechanism for secure MANET communication using DVB-T as additional unidirectional link to the network nodes has been published in **[Chr02]** with a related performance analysis presented in [Nad02]. These assumptions on the existing network environment are necessary to implement a public-key infrastructure (PKI) that acts as foundation for LKN-ASF.

One fundamental requirement for secure communication is the bootstrapping or the establishment of trust between the communication partners. LKN-ASF uses a standard public key infrastructure for this purpose. Figure 5.2 shows the concept of the PKI using three different entities: the certification authorities (CA) and the registration authorities (RA) that make up the trust center. The different CA entities establish a network of trust by cross-signing their public key certificates. The RA's handle all node requests and the general node registration to the network, thus removing some load from the CAs. This PKI as well as a review of other PKI trust models can be found in [Per99].

Network nodes receive one public key certificate each, signed by one of the CAs. Two nodes have to exchange its certificates if they want to communicate securely. Secure communication can only be established if both nodes own a certificate with a valid signature by one of the trustworthy CAs. The following section provides details concerning the PKI implementation in LKN-ASF.

Finally, it can be stated that the approach using PKI certificates to secure a 4G-embedded MANET is the most practical one. A lot of currently used protocols in the Internet use PKI technology and can also be used within MANET environments - IPSec is one example. These protocols have been widely used and tested, therefore their security mechanisms had to withstand many challenges. They are still considered secure today. PKI scenarios work no matter the size of the network and are relatively easy to administrate which is another reason why a PKI approach is a good choice for this environment.

5.2. LKN-ASF Properties

As pointed out before, secure and authentic communication can be realized by means of public key cryptography [RSA78]. Each entity has its own pair of private and public key and any two entities wanting to communicate exchange their public keys. Using these keys, they can establish a secure and authentic communication channel using a protocol like IP-Sec or Transport Layer Security (TLS). To achieve the authenticity the two entities have to be sure the association between public key and entity is valid and trustworthy. Therefore, this association has to be verifiable. A trusted third party is needed – the certification authority (CA). The CA verifies that the association between entity and key is valid and issues a certificate signed with its own private key. The public key of the CA is publicly known and trusted. Hence all entities can validate public key associations using the dedicated certificate. If a key has been compromised the associated certificate has to be revoked to ensure the security of the infrastructure. Therefore, a public key infrastructure (PKI) has to have a database for valid and revoked keys.

For a large PKI it is useful to distribute tasks. Several registration authorities (RA) can be introduced to perform the steps of the registration process for new entities. The RA forwards all valid requests to the CA which generates the certificate [Per99, Nyk00].

A norm for certificates is X.509 issued by the IETF. The X.509 standard allows public-key certificates (PKC) as well as attribute certificates. An attribute certificate can be used to bind certain attributes to an entity. Therefore, access control for special services can be realized [Nyk00]. One example for the use of attribute certificates would be access to the Internet via a gateway node. The gateway has to be trustworthy and can prove that with a valid attribute certificate.

If a PKI is used to introduce security in MANETS several restrictions have to be made. Every node has to have a valid PKC to be part of the network. Every data packet sent has to be signed in order to enable the receiving nodes to check the validity of the data sender. An attacker can not insert e.g. false routing information without having a certificate. To enable communication with Internet nodes not being part of the PKI a gateway with a proxy

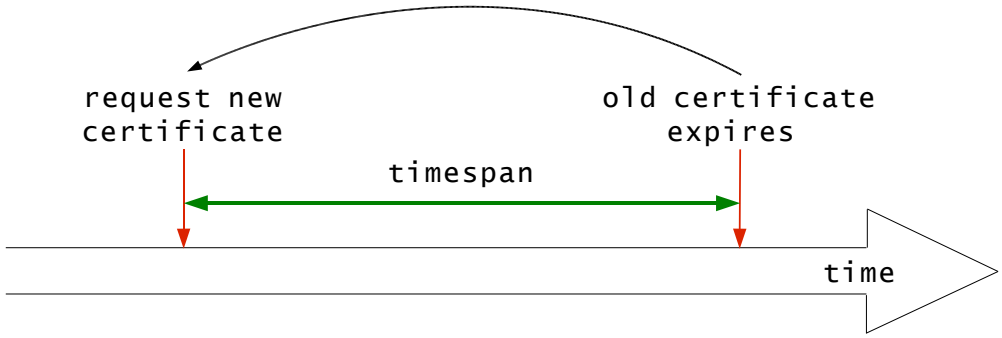


Figure 5.3.: LKN-ASF: Certificate Renewal

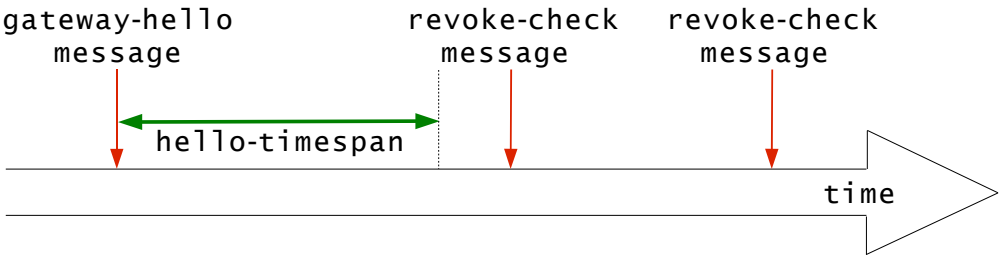


Figure 5.4.: LKN-ASF: Certificate Revocation

server has to exist to convert the unsigned packets from the Internet to valid packets for the MANET and vice versa.

Within a plain MANET without access to a gateway it is rather difficult to establish a database of revocated certificates. Also, it is not possible to obtain a new certificate if the old one expired. Therefore, securing a MANET with a PKI is possible but only useful for networks with some kind of gateway access. Otherwise a distributed CA with function sharing has to be used to make the services of the CA available within the network.

5.2.1. Trust Center

A public key infrastructure (PKI) with a trust center – the certification authority CA – is used to introduce trust within the network. To be able to communicate a node has to be registered at the trust center. The node obtains a certificate for its public key during the

registration process. The certificate is signed with the key of the trust center. The CA has to check if node and key belong together and if the node is trustworthy before issuing the signed certificate. Every subscriber within the network knows the public key of the trust center and can check the validity of any public key certificate issued by the trust center. Therefore, any two nodes can exchange and validate their public keys without having access to any other node or gateway. If the certificates are valid the nodes can trust each other and establish a secure connection.

In some PKI configurations the trust center can consist of one CA only. In large networks a single CA would be overloaded with the requests from all nodes. Therefore, registration authorities (RA) are introduced. The RAs handle requests from the nodes for the CA. The identity verification of the subscriber and the check whether this node really created the key is done by the RA. The CA only has to issue the certificates, thus it is not overloaded anymore. Therefore, the trust center consists of the CA and one or several RAs depending on the size of the network.

In a typical vehicular MANET scenario, it is possible to store the users credentials (key material, certificates) on a chipcard. This is especially important e.g. for rental cars. With this mechanism, the personalization of the in-vehicle communication servers is possible with little effort. Similar mechanisms are discussed e.g. in [EC03].

5.2.2. Certificates

LKN-ASF is used to exchange certificates. These certificates are used to verify the trustability of communication partners within the network. Every node needs at least one public key certificate to be able to communicate within the network. During certificate creation, it is not always sensible to include all properties of a node into this certificate. Thus, the identity and additional properties or permissions can be managed separately. The identity is proven via the public key certificate while additional properties are stored in one or more attribute certificates.

5.2.2.0.1. Public Key Certificates The public key certificate is used to certify the identity of a given node. Given this certificate and knowledge of the CA's public key, each node in the network can verify the authenticity of its communication peer. Hence, every participating node needs a private and public keypair. The trust center issues a public key certificate to every node after the association of node and key has been validated. Within the Internet the X.509 standard for certificates exists. This standard is used for the certificates in LKN-ASF to support common protocols also relying on the X.509 standard. The minimum necessary contents of a X.509 certificate are shown in table 5.1, a complete X.509 certificate is shown in chapter B.

5.2.2.0.2. Attribute Certificates Besides the public key certificates also attribute certificates can be used with LKN-ASF. These special certificates can only be used in combination with a valid public key certificate and provide additional information about the subscriber. This feature is especially important in ad hoc networks as the sole identity of a node is not always significant. Attribute certificates can e.g. be used to certify that a node is a valid gateway. Hence, it is harder for untrustworthy or corrupt nodes to act as gateway and run

X.509v3 Certificate Contents
Certificate Version Number
Certificate Serial Number
Issuer (Entity that Signed and Issued the Certificate)
Algorithm used for the Creation of this Certificate
Certificate Lifetime
Name of the Certificate Owner (Subject)
Subject Public Key Info
CAs Key Identifier
Key Usage (e.g., encipherment, signature, certificate signing)
Extensions
The CAs Certificate

Table 5.1.: Minimum Required X.509v3 Certificate Contents

related attacks. In addition, these certificates can be used to define user groups and therefore enforce access rights or related services within the network. Thus it is possible to charge fees for certain services in the MANET. E.g. payment for Internet gateway access can be realized this way.
Another very useful property of attribute certificates is the possibility to change attributes, to add or remove rights of certain nodes without the necessity to change the public key certificate. Possibilities to combine attribute certificates with public key certificates include a combination via the public key certificates serial number or the name that is stored in the public key certificate. If the combination is done via the name, the attribute certificates remain valid even if the corresponding public key certificate is changed.

5.2.3. Certificate Caching

In order to work efficiently in highly mobile networks (e.g. vehicle-to-vehicle communication), LKN-ASF uses caching of exchanged certificates. Thus it is possible to reuse already exchanged certificates. This reduces the amount of certificate exchange activities within the network. Therefore, it reduces network load which is critical in MANET environments as has been pointed out in chapter 3, especially in section 3.1. Since mobile nodes have a limited memory capacity only a certain number of certificates can be cached. Hence, a previously exchanged certificate can be reused as long as it is still saved in the certificate cache. The size of the cache can influence the performance of the protocol, especially if a number of connections have to be handled simultaneously. Our implementation currently uses simple LRU (least recently used) caching of certificates, however, we are planning to conduct more simulations to examine the effects of different caching strategies.

5.2.4. Certificate Revocation

The security of a PKI can only be maintained if compromised nodes can be excluded. In addition to that, any certificate issued by the CA should only have a limited lifetime. This is due to the fact that computing power is steadily increasing. A public key encryption with a given keylength that is considered to be safe today can eventually be broken after some time.

Even the publication of a new attack against a given public key system is possible resulting in the need for a new algorithm and new certificates independently from the current computing power. Hence, it must be possible to revoke certificates owned by formerly trustworthy nodes. This certificate revocation is done by the CA which issues revocation messages. A MANET typically is not connected at all times. Therefore, revocation messages can't be reliably sent to all nodes at the time of creation. LKN-ASF has to ensure the reliable and timely distribution of the revocation messages. Therefore, revocation messages have an individual packet ID and contain the number of previously issued revocation messages. Using this information nodes receiving a revocation message can determine if messages are missing and demand these messages from either neighboring nodes or a gateway.
As long as a node has access to a gateway no checks on missed revocation messages are necessary. In figure 5.4 the demand process for missed revocation messages is shown. Only if no gateway-hello messages are received anymore the revoke-check messages are broadcast to neighboring nodes to demand missed messages. If a node reestablishes a connection to a gateway it demands all missed revocation messages at once.

5.2.5. Certificate Renewal

To increase or sustain trust and security all certificates are issued with an expiration date. The validity of a public key is limited to a certain time period to decrease the possibility that a key is compromised by an attacker. Hence, the certificates issued within the network have to be renewed from time to time. Depending on the size of the encryption keys the timespan should be chosen between several months and several years. A new certificate can only be issued by the trust center. Hence, the nodes can not request a new certificate at any time, a connection to a gateway is required. At the time of expiration a connection to a gateway might not exist, leaving the node without a valid certificate. Therefore, the renewal process has to be started within a reasonable timeframe before the expiration of the current certificate. Otherwise the node has no valid certificate, thus it can no longer communicate within the network.
In figure 5.3 the certificate renewal process is shown. The timespan depends on the frequency of gateway contacts as well as on the timespan of validity. Therefore, a node having almost constant gateway access can choose a short renewal timespan. Remote nodes with very infrequent gateway contact have to trigger certificate renewal within an appropriate time before its certificate expires. As a last resort, before the node is excluded from the MANET because its certificate expires, also a connection via a different available network interface to the trust center is possible.

5.2.6. LKN-ASF and IPSec

As has been pointed out in section 4.3, available WLAN security mechanisms like WEP are not sufficient to realize secure communication between the wireless nodes. In [Zen02], a secure WLAN environment using IPv6 and IPSec (IPSecurity) has been developed to circumvent these shortcoming. While various other alternative security mechanisms are available from different vendors and are also developed within IEEE WLAN standardization, only IPSec is considered here as it provides the most flexibility and interoperability. As IPSec does also rely on a PKI, LKN-ASF can be used to enable its deployment in a MANET environment. However, while it is possible to secure data transmissions using LKN-ASF and IPSec in com-

bination, the underlying MANET security problems described in 4.4.1 still remain. Ongoing work to combine different MANET specific security protocols with LKN-ASF is presented in chapter 7 while first results on the performance of the combined approach will be published in [ESDE04].

5.3. Protocol Implementation

This section describes the different node types in LKN-ASF, its functions and the basic protocol mechanisms. Also its integration with the AODV routing algorithm (introduced in chapter 2.3.2.2) is explained.

5.3.1. Different Node Types

Nodes in LKN-ASF are separated into client nodes, gateway nodes and server nodes with the following functionality:

- **Client Nodes:** These nodes are directly participating in the MANET. They do not provide any special functionality and profit from the security protocol. The client nodes require a valid user identification and can then start operating with the discovery of neighboring nodes. Hereby, the security policies defined by the infrastructure are applied. Service providers as well as other entities can define further security policies. E.g. while the infrastructure does only require certain messages like passwords or payment information to be encrypted, the user can also specify that all his data should be encrypted. For the exchange of data within one security class, the appropriate measures are taken to ensure that the security requirements are met. Therefore, all participating nodes have to exchange its public keys using the PKI certificates before. The nodes can check the validity of any foreign public key using the certificates signature. Data exchange does only take place among successfully verified nodes.
  Attribute certificates can be used to verify certain properties of client nodes. E.g. if an Internet gateway wants to provide its service only to registered nodes it can request the corresponding attribute certificate to check whether a particular node owns a valid subscription.

- **Gateway Nodes:** The gateway nodes are important entities for LKN-ASF. The trust center that can be reached via the gateways stores information concerning the PKI and attribute certificates in its database. It manages the users keys and its attributes. Also decisions concerning certificate revocation are made in the trust center and then distributed via gateways into the MANET. In order to avoid successfully forged gateway announcements by arbitrary nodes, the gateways identities and functionality have to be verified and certified by the trust center. Thus, only gateways with a valid attribute certificate are accepted by the MANET nodes.

- **Server Nodes:** In LKN-ASF, all server nodes respectively the trust center are located outside the MANET. The trust center manages certificates and issues new certificates for keys and attributes. To maintain MANET security, the trust center also issues revocation messages that are distributed using the gateway nodes.

5.3.2. Integration with the AODV Routing Protocol

As mentioned in section 5.2.6, the credentials that are made available by LKN-ASF throughout the MANET can be directly used for secure data transmission, e.g. using standardized protocols like IPSec. However, as pointed out in section 4.4, routing security is essential in MANETs because of its vulnerability against the attacks of single or few malicious nodes.
As shown in section 3.1.3, reactive protocols show better performance in highly mobile scenarios. Within the IETF, the two most promising MANET routing protocols that have been published as experimental RFC are AODV and DSR. As simulative performance studies like [PRDM01]show better results for AODV in highly mobile scenarios, AODV has been the protocol of choice for the LKN-ASF performance studies. An overview of AODV can be found in section 2.3.2.2 within this thesis.

When a node S needs a route to another node D within the network, S first sends a route request message (RREQ). The RREQ message is forwarded through the network until a valid route to D is found or the TTL of the message is exceeded. If a valid route is found, a route reply message (RREP) is sent back to node S. The RREP message is sent back along the same route that has been used by the successful RREQ message.
It is somewhat problematic to secure this procedure as a key exchange between S and D can only be done after a valid route has been established. In order to verify the validity of RREQ and RREP messages, however, these certified keys are necessary and the situation results in a vicious circle. To break this circle, the routing protocol has to be integrated into the security protocol or it has to use its own, additional security mechanism. The work presented in [Nad02] makes use of DxB broadcast messages to distribute the certificates among the nodes. While the additional usage of broadcast messages is helpful to establish secure communication in a MANET [Chr02], a separate network interface is inevitable. Therefore, Internet gateways are used here to distribute certificates among the nodes. Within LKN-ASF, the routing protocol is modified so that a key exchange between two neighboring nodes takes place before RREQ messages are processed. Thus, a connection is established only if both, S and D as well as all nodes on the route from S to D are trustworthy. Due to the characteristics of the wireless medium, any node in range can hear the transmitted messages, however, the credentials distributed with LKN-ASF can be used to encrypt these messages and make them unreadable for potential eavesdroppers.

To avoid any misunderstandings it has to be mentioned that LKN-ASF distributes and manages necessary credentials to secure MANETs. Without additional security mechanisms, it does not prevent attacks against the MANET and its protocols. As LKN-ASF relies on the X.509 standard, it is possible for compatible protocols to make direct use of this framework. Among other topics, the development of a secure routing algorithm for MANETs based on LKN-ASF is presented in chapter 7.

5.4. Secure Connection Establishment

LKN-ASF has been specified using SDL [SS00, sdl]. Functional tests have been done using the Telelogic TAU SDL suite. For performance evaluation, LKN-ASF has been completely implemented in GloMoSim [glo]. While this section provides an overview of LKN-ASF based on the SDL specification, a more detailed SDL specification is available in the appendix of [Eic03].

Details concerning the GloMoSim implementation are presented in chapter 6.

Figure 5.5 shows how a connection is set up at the sender (source). A source node S which wants to send data to a destination node D first checks if a secure connection to D already exists. The next step is the transmission of a SYN message from S to D. This message contains requests for required certificates. After that, node S receives requested certificates as well as ACK messages from D. Using the ACK messages, D can request required certificates from S. When all necessary certificates are exchanged, the connection is considered to be secure and node S can start to transmit its data.
The subprocess "check connection" searches the connection database whether a secure connection from S to D has already been established. If this is the case, the SYN message contains only requests for possible additional certificates. If no existing secure connection is found, a new entry is made in the connection database and a *SYN* message containing a request for all required certificates is sent.

Figure 5.6 shows the connection setup at the receiver (destination). The subprocess "check connection" is also used here. The receiver can request required certificates and notify the sender when it (the receiver) is ready to receive data transmissions using ACK messages. The data transmission starts as soon as the sender receives an ACK message confirming the setup of a secure connection and no further certificates have to be exchanged.

If a node can not identify itself with the required certificates, LKN-ASF drops packets from this node. This results in the behavior depicted in figure 5.7: the shortest possible route is not established if nodes on its way fail to authenticate. Instead, a secure route is chosen.

As has been pointed out in previous sections, timely certificate revocation and certificate renewal are essential for the systems security.
The receipt of a revocation message, declaring previously valid certificate as invalid, is shown in figure 5.8. On receipt of a new revocation message, it is stored in the nodes database and forwarded to neighboring nodes. This is done within the subprocess "add revocation". Additional functionality of this subprocess includes the check of the revocation database. This is done to detect whether a previous revocation message has been missed. In this case, a revoke-miss message is sent to request this message from a gateway or from neighboring nodes.

As certificates have a limited lifetime only, it has to be made sure that nodes can participate in the MANET even after the expiration of its certificates. This is done via the certificate renewal process shown in figure 5.9. For the renewal process, a connection to a gateway is necessary as the trust center has to be contacted. As it can not be guaranteed that a node is connected to a gateway when its certificate expires, certificate renewal requests have to be made within a reasonable timeframe before the expiry. This timeframe can be calculated depending on the frequency of previous gateway contacts. Figure 5.9 also shows what happens if the certificate renewal process is triggered without a gateway in range[1]: the renewal process is put on hold until a gateway advertisement is received. In this case, the renewal is continued immediately.

[1]The gateway can be contacted securely via a single-hop link but also via a multi-hop link.

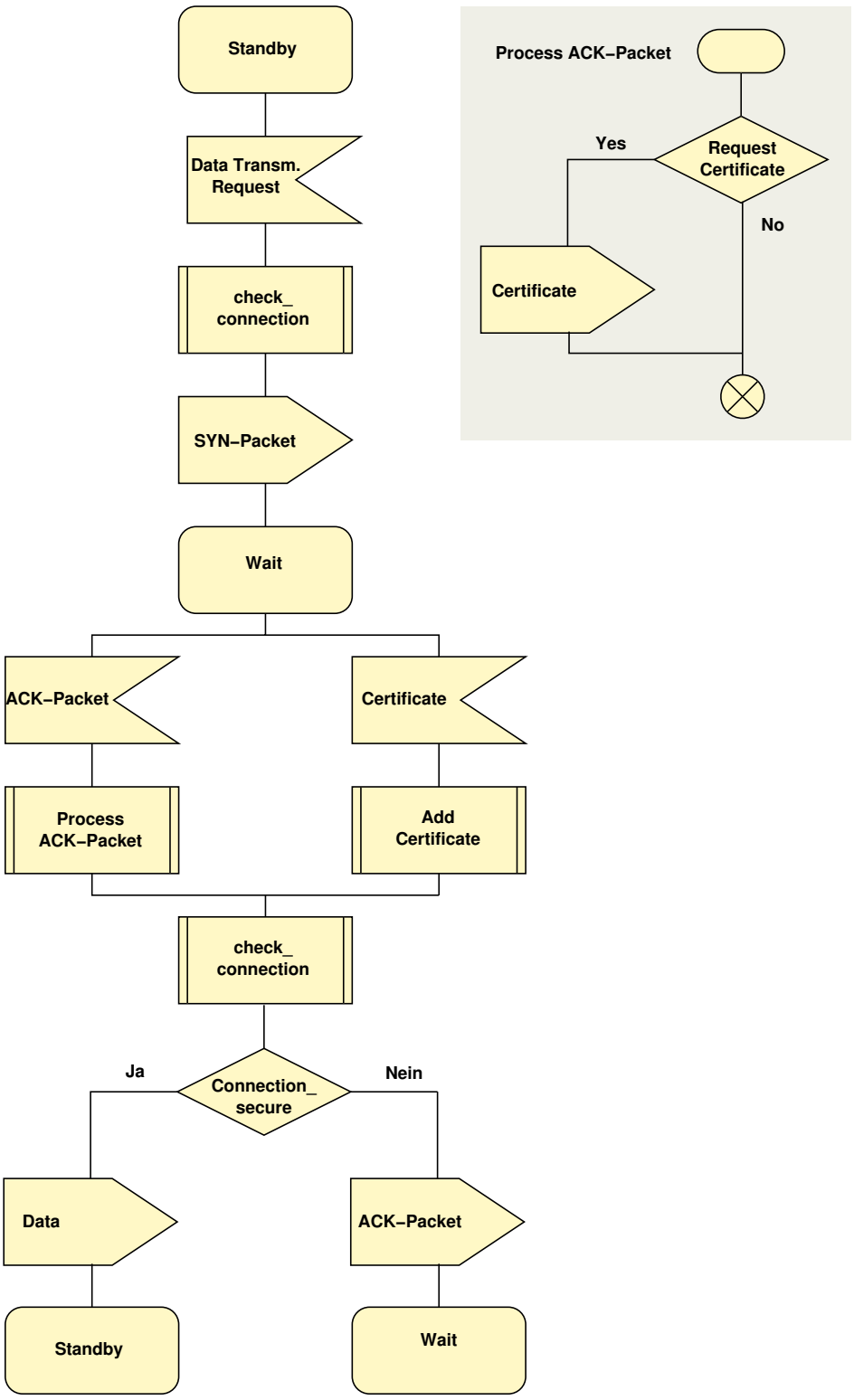


Figure 5.5.: LKN-ASF: Connection Setup (Sender)

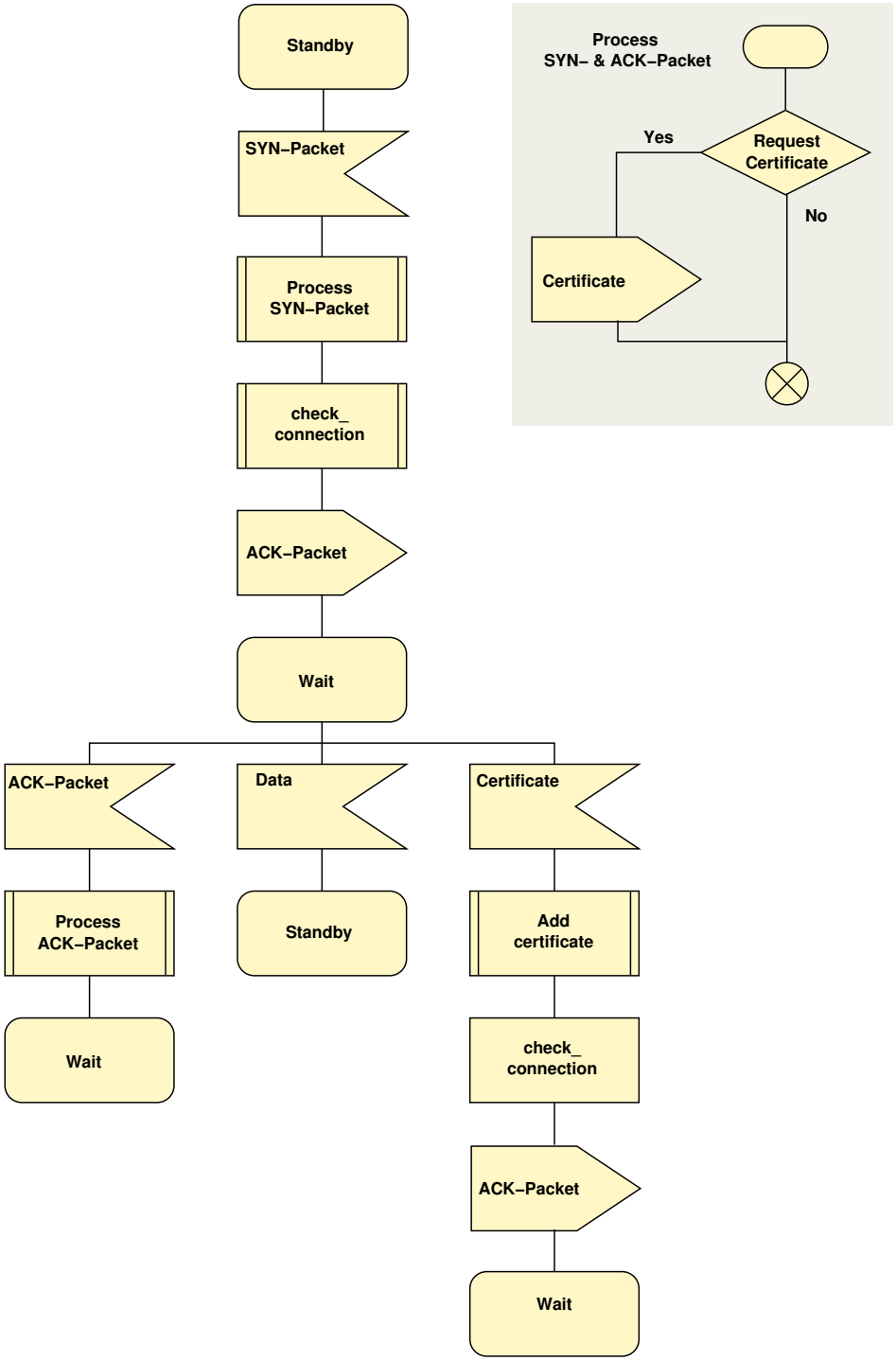


Figure 5.6.: LKN-ASF: Connection Setup (Receiver)

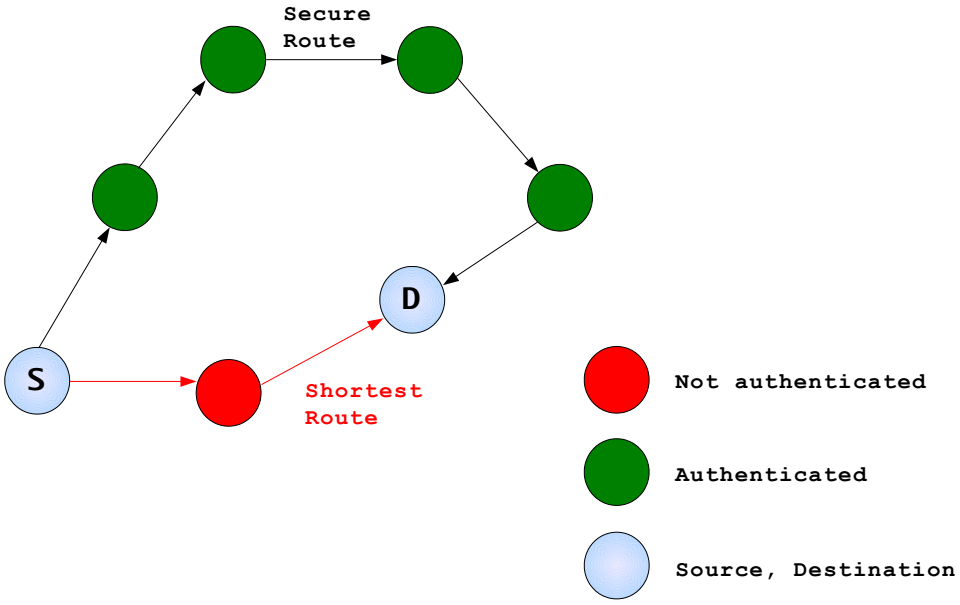


Figure 5.7.: LKN-ASF: Route Selection

5.5. Conclusion

LKN-ASF, a framework for secure communication has been presented. If sporadic access to a trust center is possible, all required security services can be realized. While LKN-ASF is only a framework, it can be combined with routing algorithms and also higher layer protocols like service discovery protocols to achieve security in a highly distributed environment. The next chapter examines the impact of LKN-ASF on network performance as well as the performance of ASF itself.

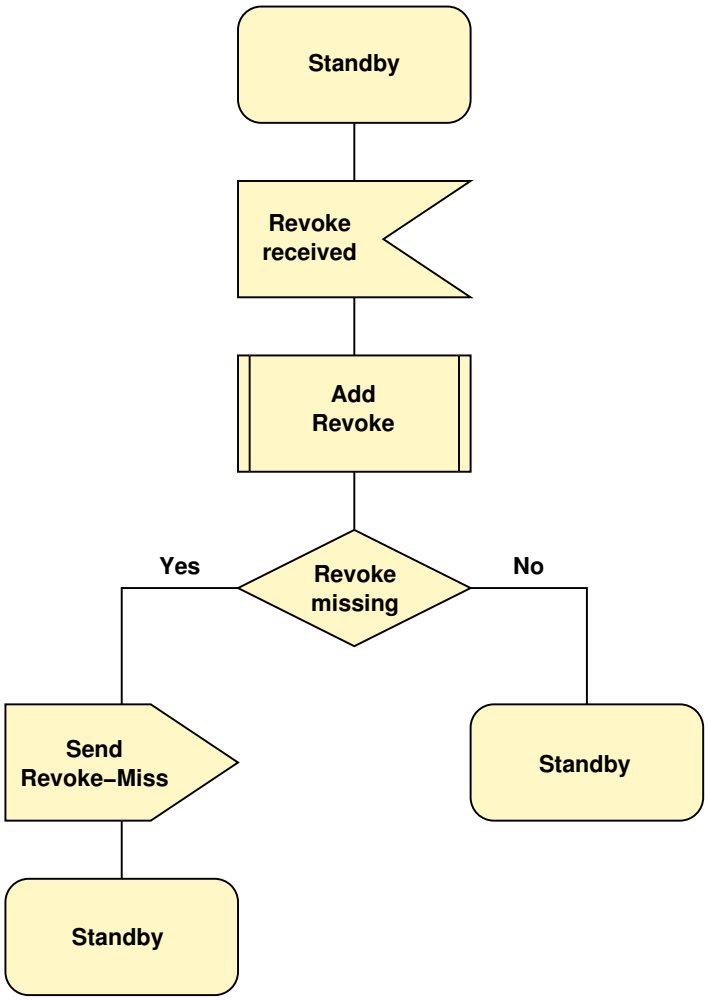


Figure 5.8.: LKN-ASF: Certificate Revocation

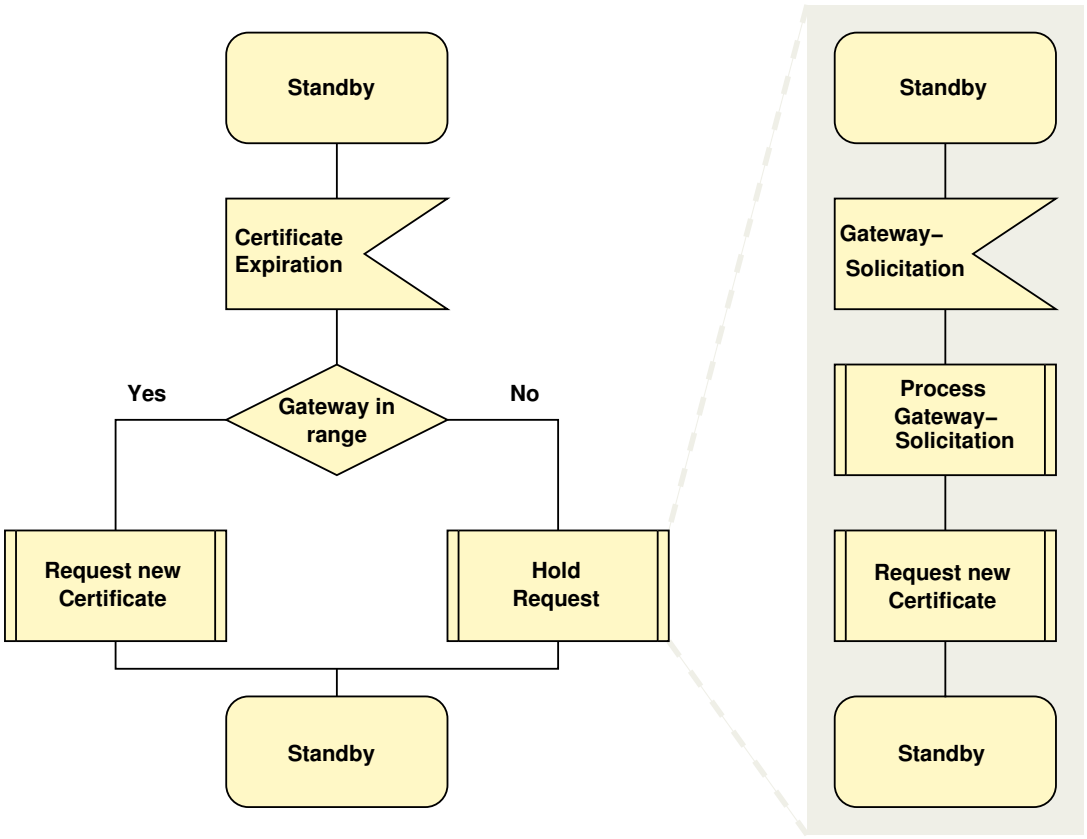


Figure 5.9.: LKN-ASF: Certificate Renewal

6. LKN-ASF Protocol Simulation and Results

As has been shown in the preceding chapters, especially in chapter 3, MANETs face severe performance constraints. This leads to a bad suitability of distributed, highly interactive protocols for MANET security.

In this chapter, the performance of LKN-ASF is discussed. The results presented in this chapter have been gained using the Global Mobile Simulation Systems Library (GloMoSim, [glo]) with different simulation scenarios.

The chapter is organized as follows: section 6.1 provides a brief summary of GloMoSim. Section 6.2 describes the implementation of LKN-ASF in GloMoSim and the necessary changes that had to be made within the simulator. Simulation results for different simulation scenarios are presented in section 6.3.

6.1. Global Mobile Simulation Systems Library (GloMoSim)

GloMoSim has been developed at the UCLA (University of California) and is available without cost for academic purposes. Its focus is on the simulation of large, wireless networks. For this purpose, a lot of models and protocols on different OSI-layers are already pre-defined in GloMoSim. Table 6.1 shows some of the models and protocols available in GloMoSim.
The simulator has been programmed in a C-like language called PARSEC (Parallel Simulation Environment for Complex Systems, [BMT+98]). PARSEC supports sequential and parallel execution of time-discrete procedures. Thus, simulation events can be simulated in parallel, resulting in a high efficiency of GloMoSim that allows the simulation of large-scale networks [BTAB99].

Area	**Protocols**
Applications	CBR, FTP, HTTP, TELNET
Transport	TCP, UDP
Network	IP, some routing algorithms
MANET Routing	AODV, FISHEYE, DSR, etc.
MAC	IEEE 802.11, CSMA, etc.
Fading, Path Loss	Two ray, free space
Interference	SNR based, BER based
Mobility	Various random models, manual specification

Table 6.1.: Summary of Models and Protocols available in GloMoSim

All events within GloMoSim are linked to a certain node within the simulation. The data structure *GlomoNode* is very central in this respect: all properties, statistics and protocols of

a node are contained in this structure.
The data structure *Message* is another important part of GloMoSim. All messages used in the simulator use this structure, e.g. timer events or arriving data packets. The timer events are used to trigger node-internal events like the aging of route-table entries. Concerning packet headers, GloMoSim offers the possibility to add optional fields to any packet header to transmit additional information. This feature has also been used for the implementation of LKN-ASF.

6.2. LKN-ASF Implementation in GloMoSim

While good templates and entry points for the creation of new protocols are prepared in GloMoSim, it has not been practical to use them for the implementation of LKN-ASF. The reason for this decision was that LKN-ASF has not been intended to exist as alternative protocol besides other options. LKN-ASF was designed to be used together with most other protocols. An additional requirement therefore was the necessary unrestricted access to all data packets.

Therefore, it has been decided to use the source file *nwip.pc* as entry point. All IP packets pass through the functions defined in this file. Function *ProcessPacketForMeFromMac* is called when a node receives a packet. The packet header contains the sub-protocoll used. Depending on this sub-protocol, appropriate further functions are used to process the packet. The function *ProcessPacketForAnotherFromMac* is used for packets that are only forwarded by the receiving node. The main functionality of function *NetworkIpLayer* is the handling of timer events. These functions within *nwip.pc* have been used to start with the implementation of LKN-ASF in GloMoSim.
AODV [PBRD03] has been used as routing protocol for all simulations. Reasons for this decision have been that AODV is a very popular reactive routing protocol and well suited for highly mobile scenarios like vehicle-to-vehicle communication. In addition to that, simulation studies suggest a better performance of AODV to a second candidate routing protocol, DSR [JMH03]. Especially in scenarios with high mobility, the performance of AODV has been found to be better considerably. Details on performance comparisons can be found in chapter 3, especially in section 3.2.1. The basic functionality of AODV and DSR is explained in section 2.3.2.2. The routing process is used by all data packets. Existing routes are used if available, new routes are discovered if necessary. Packets that can not be transmitted because no route is available are buffered within the node and a timer is started. If no valid route is found before the timer expires, the packets are discarded.

The data structure *GlomoNode*, mentioned above, has been extended to be able to store and manage key certificates, attribute certificates as well as statistical values for the simulation results. This has been done using an additional pointer to the new data structure *GlomoSecurity* that contains all informations a node needs to be able to use LKN-ASF:

```
typedef struct glomo_security_str
{
  BOOL                  securityActive;
  BOOL                  securityStats;
  BOOL                  freePackets;
```

```
  BOOL                  CAGateway;
  BOOL                  GatewayPresent;
  BOOL                  gatewayCheck;
  BOOL                  gatewayHelloRecvd;
  BOOL                  needNewKey;
  clocktype             gatewayStamp;
  NODE_ADDR             gatewayAddr;
  clocktype             gatewayHello;
  clocktype             revokeCheck;
  clocktype             lastRevoke;
  clocktype             expireTime;
  unsigned int          numRevokedKeys;
  KeyCertificate        *nodeKey;
  AttributeCertificate  *nodeAttribute;
  long                  CAKey;
  KeyCacheType          *nodeKeyCache;
  int                   nodeCacheSize;
  AttributeCacheType    *nodeAttribCache;
  RevokeCertificate     *nodeRevokeCache;
  AODV_SEC_Conntrack    *secConntrack;
  GlomoSecurityStats    *nodeSecStat;
} GlomoSecurity;
```

LKN-ASF also requires entries in the packet headers. In GloMoSim, an entry containing the packets signature has been added. For this purpose, the function *AddIpOptionField* is used. LKN-ASF does this in its function *SetSecurityOptionHeader* for adding a header field. On receipt of a packet, the security header is read and interpreted by the function *ReadSecurity-OptionHeader*.

In GloMoSim, different IP protocoll types are defined. Depending on these types, appropriate functions are called to process incoming packets. For the implementation of LKN-ASF, new protocol types have been added to the file *nwcommon.h* to be able to directly access LKN-ASF messages. A summary and short explanation of the new protocol types is shown in table 6.2.

For the configuration of simulations using LKN-ASF, the main configuration file of GloMoSim, *config.in* can be used. Table 6.3 shows the new options that can now be specified in GloMoSim. The parameters KEY-VALID and CA-GATEWAY have to be used with the preceding parameter NODE_ADDRESS[from:to] if different values should be assigned to different nodes. For debugging reasons, the parameter DELETE-INVALID-PACKETS has been introduced. This parameter allows for the processing of invalid packets within the simulation. Thus it is possible to check whether invalid[1] packets have been the reason for failed communication attempts.
The parameters REVOKE-CONFIG-FILE and EXPIRE-CONFIG-FILE are used to specify additional configuration files for revoke- and expiration-events. Thus it is possible to revoke or expire certificates of arbitrary nodes at a predefined time.

[1] invalid from a security perspective

Protocol Type	Usage
IPPROTO_SEC	Requests for and submissions of key certificates or attribute certificates, packets containing new certificates and packets containing revocation messages
IPPROTO_SEC_KEY	Requests for new key certificates
IPPROTO_SECGW_HELLO	Timer messages to trigger periodic gateway solicitations
IPPROTO_SECGW_REVOKE	Timer messages to trigger revoke-check messages
IPPROTO_SEC_EXPIRE	Timer messages for key expiration
IPPROTO_SECFAILED	Mark for invalid packets
IPPROTO_SECGW_CHECK	Timer for gateway connectivity checks
IPPROTO_SEC_REVOKE	Requests for missed revocation messages
IPPROTO_SEC_HELLO	Gateway solicitation messages

Table 6.2.: New GloMoSim Protocol Types for LKN-ASF

6.3. Simulation Results

In this section, simulation results concerning the performance of LKN-ASF are presented. Simulations have been done with and without mobility. Figure 6.1(a) shows the scenario that has been used for some simulations results described below. The radio range has been set to 220 meters in all simulations, the simulations using figure 6.1 as topology include 20 regular nodes (6-25) and five gateway nodes. This initial setup resulted in the topology shown in figure 6.1(b).

6.3.1. Direct Connection between two Nodes

First tests of the simulation models functionality included direct communication between two neighboring nodes with and without LKN-ASF. The results gained from these simulations are presented in table 6.4 and 6.5 respectively for comparison reasons. The simulations have been done without node mobility. A FTP connection from node 1 to node 2, transmitting 1 MB of data (2000 packets, 500 byte each) has been used for traffic generation[2].

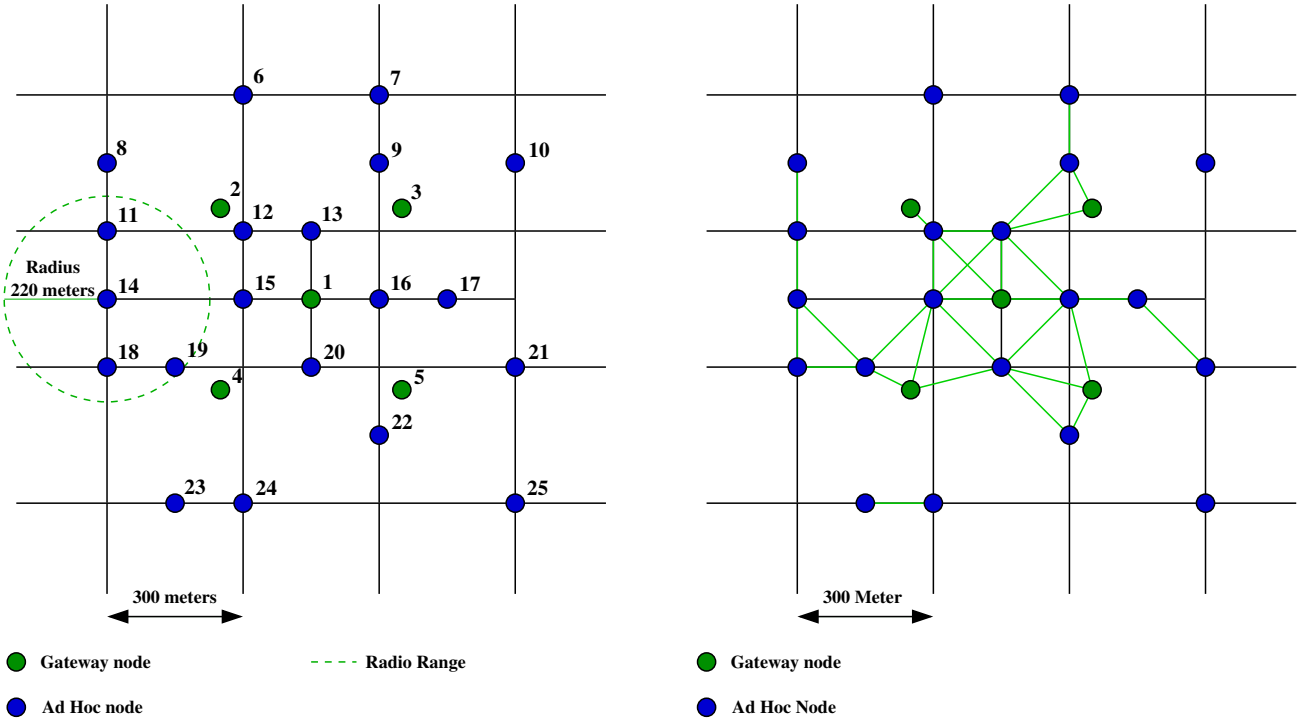
6.3.2. Multihop-Connections with LKN-ASF

Further simulations involved multihop connections between nodes in the MANET. Table 6.6 shows the results for 1 intermediate node (2 hops). This simulation, again without node mobility, consisted of an FTP connection transmitting 1 MB of data from node 1 to node 2 via node 3[3].

The values in table 6.6 show that compared to the results presented in section 6.3.1, the throughput is only about half as high and thus the connections duration is doubled. This result can be explained with the fact that the intermediate node (node 3) in this simulation

[2]This simulation consisted of 2 nodes only. Node 1 and node 2 do not correspond to the nodes shown in figure 6.1.

[3]This simulation consisted of 3 nodes only. Node 1, node 2 and node 3 do not correspond to the nodes shown in figure 6.1.



(a) Road-based Simulation Scenario

(b) Related Connectivity

Figure 6.1.: Road-based Simulation Scenario Setup and Connectivity

Parameter	Usage
SECURITY	Activation and deactivation of LKN-ASF.
KEY-VALID	Determines validity of keys.
CA-GATEWAY	Defines one or more nodes as gateway.
GATEWAY-HELLO	Interval for periodic gateway solicitations.
GATEWAY-CHECK-FACTOR	At least one gateway solicitation has to be received within this time window. Else gateway connectivity is assumed to be lost.
CACHE-SIZE	Size of the certificate cache within a node.
DELETE-INVALID-PACKETS	Determines whether invalid packets are discarded.
REVOKE-CHECK-TIME	Interval for periodic revoke-check messages.
RENEW-KEY-ADVANCE	Time between key renewal request and key expiration.
REVOKE-CONFIG-FILE	Configuration file for revocation events.
EXPIRE-CONFIG-FILE	Configuration file for expiration times.
SECURITY-STATISTICS	Activation and deactivation of simulation statistics.

Table 6.3.: GloMoSim/LKN-ASF Configuration Parameters (valid within config.in)

Parameter	Node 1	Node 2
Transmission start	200.005s	200.006s
Transmission end	208.49s	208.615s
Duration	8.485s	8.609s
Throughput	942780bps	929273bps

Table 6.4.: Results for Singlehop Communication without LKN-ASF

setup can not send and receive at the same time. The same behavior has also been measured in real world tests with an IEEE 802.11b based AODV testbed. Results of these measurements as well as details about the setup of this testbed are presented in section 3.3.1.

This multihop scenario with three nodes has also been simulated with LKN-ASF activated. The results presented in table 6.7 show a slight increase of the connections duration and with it a slight decrease of the achievable throughput. However, these small differences do not have an significant impact on the connections quality.
Additional simulations using longer multihop routes in the scenario depicted in figure 6.1 have been done and are presented in detail in [Eic03]. As expected and analyzed in section 3, the per node throughput in these scenarios decreases with the number of active nodes in the network. However, the additional activation of LKN-ASF does not have an additional significant impact on throughput and connection duration. However, with the growing size of the network, significantly higher numbers of revoke-check messages have been observed with potentially negative effects on the networks performance. This effect has been studied in an additional simulation setup presented in section 6.3.5.

In [Ste04], an additional simulation setup has been designed to study the impact of different route lengths. Again, LKN-ASF showed very good performance as can be seen below. The

Parameter	Node 1	Node 2
Transmission start	200.015s	200.017s
Transmission end	208.587s	208.720s
Duration	8.572s	8.703s
Throughput	933281bps	919137bps

Table 6.5.: Results for Singlehop Communication with LKN-ASF

Parameter	Node 1	Node 2	Node 3
Transmission start	200.096s	200.099s	-
Transmission end	217.045s	217.198s	-
Duration	17.051s	17.099s	-
Throughput	472002bps	467874bps	

Table 6.6.: Results for 2-Hop Connections without LKN-ASF

simulation setup assumed static nodes, the network topology used for these simulations can be seen in figure 6.2. This results in the situation that a node can only communicate directly with its immediate neighbors, multihop links have to be used for all other destinations. The nodes at the endpoints set up a FTP connection and transmit 100 packets of 1024 byte data each.

Figure 6.3 shows the throughput that has been achieved in this scenario. Without security protocol, the throughput decreased with a growing number of hops as predicted in chapter 3. However, it can be seen that the additional usage of LKN-ASF does only slightly effect the achievable throughput.
Related to these results, the average connection duration for a FTP connection as described above is presented in figure 6.4. A noticeable difference between simulations with and without security protocol can only be observed for multihop links above three hops. However, also the increase for links above three hops is reasonable so LKN-ASF can be used in these scenarios with only a slight performance decrease.

The usage of LKN-ASF in a gateway supported MANET increases the network load slightly. The main influencing factors in this environment are gateway detection and revocation messages. For the simulations presented here, periodic gateway advertisements are broadcast by the gateway. Other possible mechanisms for gateway discovery that can also be used together with LKN-ASF are presented in section 2.4. In figure 6.5, the throughput of a normal connection using LKN-ASF is compared with the throughput of a LKN-ASF gateway connection. Also here, almost the same values have been observed for LKN-ASF links between normal ad hoc nodes and LKN-ASF links between normal nodes and a gateway.

6.3.3. Cache Size Variations

When network nodes exchange its certificates at connection setup, these certificates are cached within the nodes. The size of the nodes caches can be set in the GloMoSim configuration file *config.in*. If all positions within the cache are allocated and a new certificate arrives, the

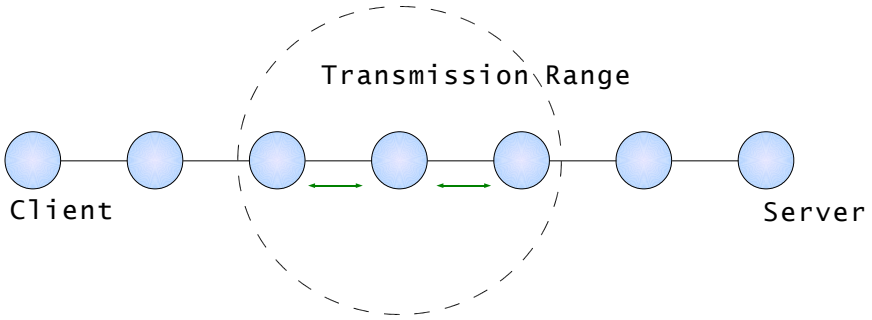


Figure 6.2.: Network Topology (Chain)

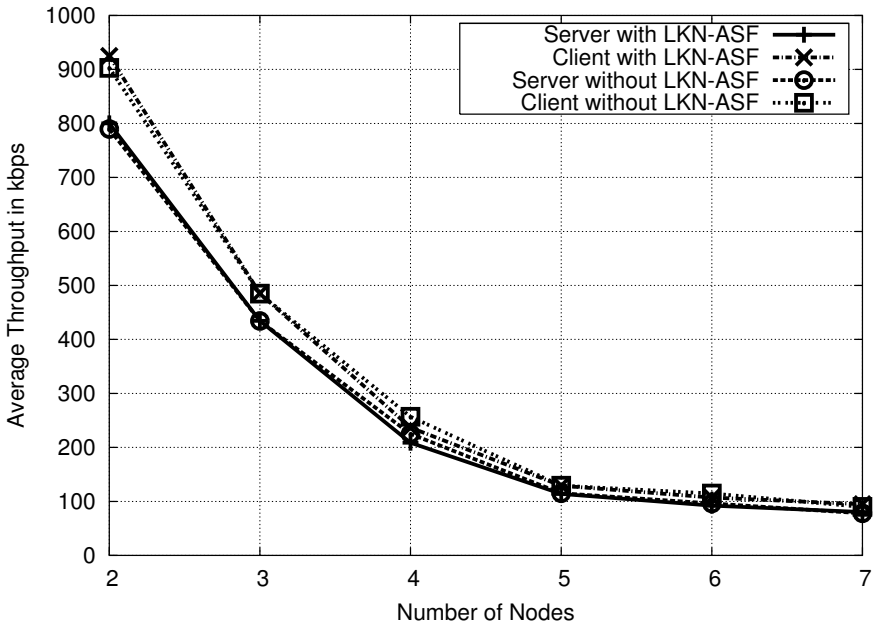


Figure 6.3.: Data Throughput

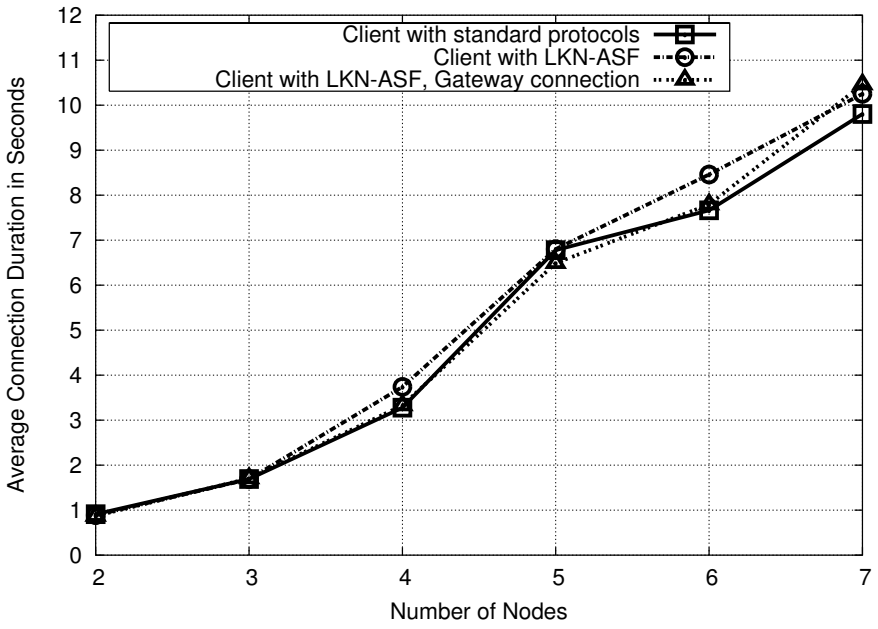


Figure 6.4.: Duration of Connections

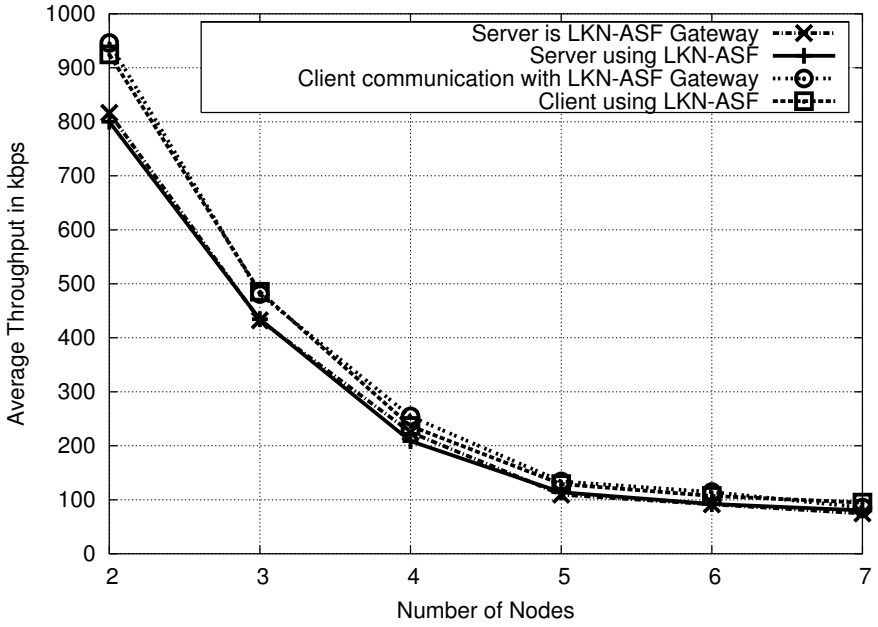


Figure 6.5.: Data Throughput with Gateway

Parameter	Node 1	Node 2	Node 3
Transmission start	200.117s	200.124s	-
Transmission end	217.267s	217.409s	-
Duration	17.150s	17.285s	-
Throughput	466486bps	462844bps	-

Table 6.7.: Results for 2-Hop Connections with LKN-ASF

oldest certificate is overwritten. The simulations within this section show how a small cache size (lower or equal as the number of active connections) effects LKN-ASF.
For the simulations, node 12, 13, 15, 16 and 17 as well as gateway node 1 from figure 6.1 have been used. A static topology without node mobility has been selected in order to focus completely on cache size effects. All nodes start a FTP connection to the gateway node at the same time with the intention to transmit 1 MB of data. The connection times summarized in table 6.8 have been measured at the gateway node.
The values in table 6.8 represent results from one simulation run per cache size. The following interpretation of the simulation results explains why no average has been calculated.

The results for cache size ≥ 5 show that all connections are completed successfully, the connection duration is quite similar for all nodes.
Simulation results with cache size 4 show that, despite the number of available entries within the cache is smaller than the number of connections, all connections are completed successfully. However, the duration of one connection is increased by about 120s. Other connections benefit from this situation and show smaller durations.
First transmission failures have been noticed in simulations with cache size 3. The connection from node 12 is not started. The remaining results for this simulation show a very high connection duration of one node and again, the other nodes drawing advantage from this situation.
Simulations for cache size 2 and cache size 1 show similar results. Some connections are not completed (node 17, cache size 2), the duration of other connections is increased noticeably.
To summarize, it is possible to have more concurrent connections than available cache entries. However, it is possible that some connections are not set up or not completed because of time outs. This occurred even in situations without node mobility. Another observation that has been made is the considerable increase of connection durations when the cache size is reduced. This results in a reduced probability of successfully completed connections which is especially problematic in mobile scenarios. No average has been calculated for the simulation results in table 6.8 as in different simulation runs different nodes have been able to draw advantage from problems at other nodes. As the trend has been the same in all simulation runs, the exemplary results from one simulation represent this situation properly.

6.3.4. Dissemination of Revocation Messages

To maintain a constantly high level of security in gateway supported MANETs, the dissemination of revocation messages is fundamental. Simulations have been done with and without node mobility in the scenario shown in figure 6.1. The results for the static scenario have been as expected: all nodes connected to the gateway via single-hop or multi-hop links received

	Node 12	Node 13	Node 15	Node 16	Node 17
Cache size $\geq$ 5					
Duration	30.893s	37.297s	36.444s	32.062s	31.535s
Cache size = 4					
Duration	25.66s	167.834s	30.294s	25.932s	25.913s
Cache size = 3					
Duration	145.53s	-	24.590s	21.935s	19.518s
Cache size = 2					
Duration	154.036s	29.718s	144.951s	16.385s	∞
Cache size = 1					
Duration	151.476s	151.232s	10.005s	41.978s	9.977s

Table 6.8.: Effects of LKN-ASF Cache Size Variations

the revocation messages without significant delay (0.001s to 0.005s). As the nodes are not moving in this scenario, nodes without gateway connection are not informed about revoked certificates.

In mobile scenarios, the dissemination of revocation messages is also influenced by the connectivity situation to the gateway when an revocation messages is distributed by the gateway. The results for nodes connected to the gateway either single-hop or multi-hop have been the same in static and mobile scenarios. The interesting aspect here is the dissemination of revocation messages throughout the complete network, i.e. the time it takes until also remote nodes are informed about revoked certificates. This is mainly influenced by two factors: the mobility of the nodes and the revoke-check interval. The effects of mobility have been studied in [CHB03]. This paper shows that mobility is advantageous for security, i.e. security credentials are distributed faster among the nodes if its mobility increases. The influence has been shown for several settings of the random walk mobility model and a restricted random waypoint mobility model.

The intervals of revoke-check messages are another important parameter for the dissemination of revocation messages among the nodes. Table 6.9 and 6.10 show how fast revocation messages are distributed depending on different revoke-check intervals. These intervals, however, also influence the protocol overhead of LKN-ASF as will be shown in section 6.3.5. Both simulations used deterministic node movements (urban vehicular speeds of 0 - 50 km/h) along the roads shown in figure 6.1. These movements had to be configured on a per-node basis within the GloMoSim configuration file.
Simulations done in a mobile scenario with 200 nodes, high load and a high node density (described in section 6.3.6) also showed very promising results. Revocation messages sent at different simulation times have been received by all nodes within 1 second after the message has been sent by the gateway.

6.3.5. Network Load due to Revoke-Check Packets

As explained in section 5.2.4, LKN-ASF makes use of revoke-check packets to distribute new revocation messages among nodes without direct gateway contact. The results presented in

Delay	Node
0.000s	Gateway
0.001s	8, 9, 15
0.002s	10, 24
0.004s	14, 17, 20
0.005s	13, 18
10.479s	21
39.063s	16, 19
67.084s	6
67.092s	23
69.063s	12
97.032s	25
97.271s	7
97.406s	22
98.461s	11

Table 6.9.: Dissemination of Certificate Revocations, Revoke-Check Interval 5 minutes

section 6.3.4 showed that revocation messages are distributed faster if smaller revoke-check intervals are used. However, the usage of shorter revoke-check intervals also leads to higher protocol overhead. Figure 6.6 and figure 6.7 show the traffic caused by different revoke-check intervals. The simulations are based on the network topology shown in 6.1 without node mobility.
Acceptable delays for the dissemination of revocation messages may differ for various scenarios. However, a revoke-check interval of two minutes should be sufficient for most applications. Thus, the required functionality can be provided with moderate traffic overhead.

6.3.6. Larger Simulation Scenarios

Further simulations included simulation scenarios with 50 nodes (static) and simulation scenarios with 200 nodes (with node mobility).
The static setup with 50 nodes is shown in figure 6.8. The simulation time has been set to 30 minutes, 30 connections (FTP, 1MB data traffic each) have been simulated between random network nodes and random gateway nodes. These settings resulted in some few connection errors even in simulations without LKN-ASF. All connections started, 5 connections failed to transmit all its data. This setup has been selected intentionally to have a demanding setting for LKN-ASF. The same traffic pattern has been used in simulations with and without LKN-ASF.
The additional activation of LKN-ASF resulted in comparable results: one connection did not start at all due to a longer delay for route setup caused by LKN-ASF. All other connections have been set up successfully. Four connections failed to transmit all its data.

The simulations with 200 nodes have been done within a rectangular simulation area of 2700 x 1500 meters. 10 nodes out of the 200 nodes have been randomly selected as gateway nodes. All other nodes have been configured as mobile network nodes and have been able to move freely within the simulation area. 35 FTP connections have been configured within a

Delay	Node
0.000s	Gateway
0.001s	8, 9, 15
0.002s	10, 24
0.004s	14, 17, 20
0.005s	13, 18
10.479s	21
37.624s	23
37.868s	19
38.524s	11
39.076s	16
39.422s	12
39.637s	25
40.835s	22
67.064s	6
97.186s	7

Table 6.10.: Dissemination of Certificate Revocations, Revoke-Check Interval 2 minutes

simulation time of 30 minutes.
Simulations without LKN-ASF resulted in the successfull completion of 20 connections. All connections have been set up successfully while 15 connections have not been finished due to node mobility.
In simulations using LKN-ASF, all connections have been set up successfully. 12 connections have been completed successfully while 18 connections failed to transmit all data packets. Further investigation showed that this behavior was mainly due to the longer delays in route setup times due to LKN-ASF. From the 18 failed connections, 5 connections have been able to transmit 80% or more of its data before they failed.
The main protocol overhead in this scenario has been caused by revoke-check messages. Based on simulation results presented in the previous section, the revoke-check interval has been set to two minutes. This lead to 1131 revoke-check messages in the simulation. The number of forwarded revoke-check packets has been 202188 which results in an overhead of 203319 packets in the LKN-ASF simulation compared to the simulation without security protocol.

6.4. Conclusion

This section presented simulation results of LKN-ASF with a focus on signaling overhead, throughput and delay. Results have been gained using the Global Mobile Simulation Systems Library (GloMoSim). Simulations started with very simple scenarios to check the functionality of LKN-ASF and to validate the simulation model. It was possible to reproduce results that have been measured in testbed experiments with the simulation model. To summarize, the performance of LKN-ASF looks very promising, i.e. it does not create permissive signaling overhead or delays if configured properly[4]. LKN-ASF has been found to also work in relatively

[4]Proper configuration means sensible settings of revoke-check intervals and gateway advertisements that have been found to create permissive traffic overhead if set to very small values.

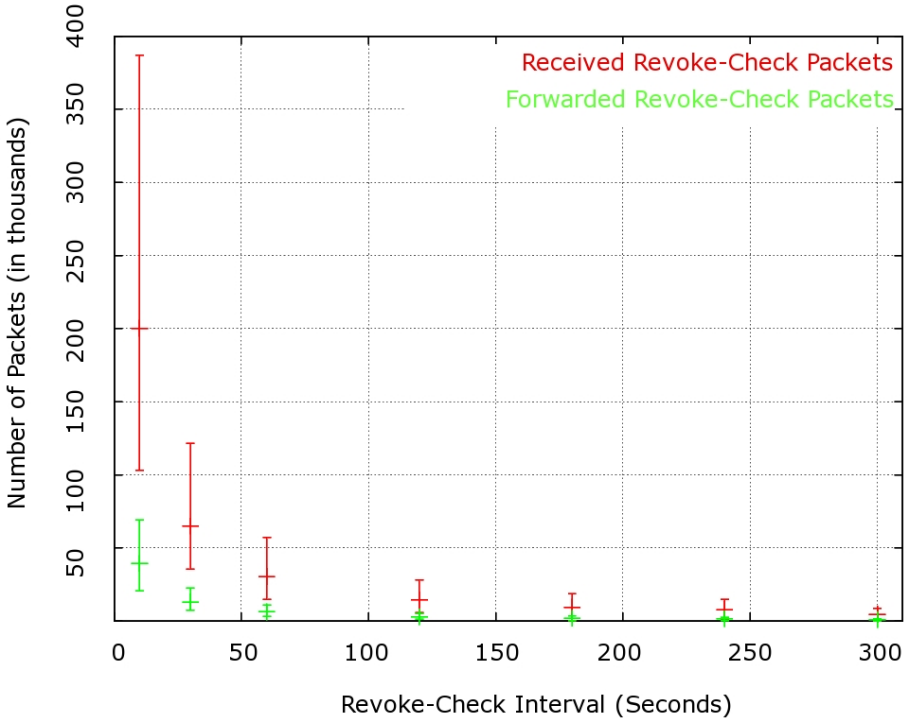


Figure 6.6.: Received and Forwarded Revoke-Check Messages

large scenarios with 200 mobile nodes. The continuation of this work is outlined in chapter 7, especially the close combination of LKN-ASF with the AODV routing protocol and efficient strategies for certificate revocation for large user groups.

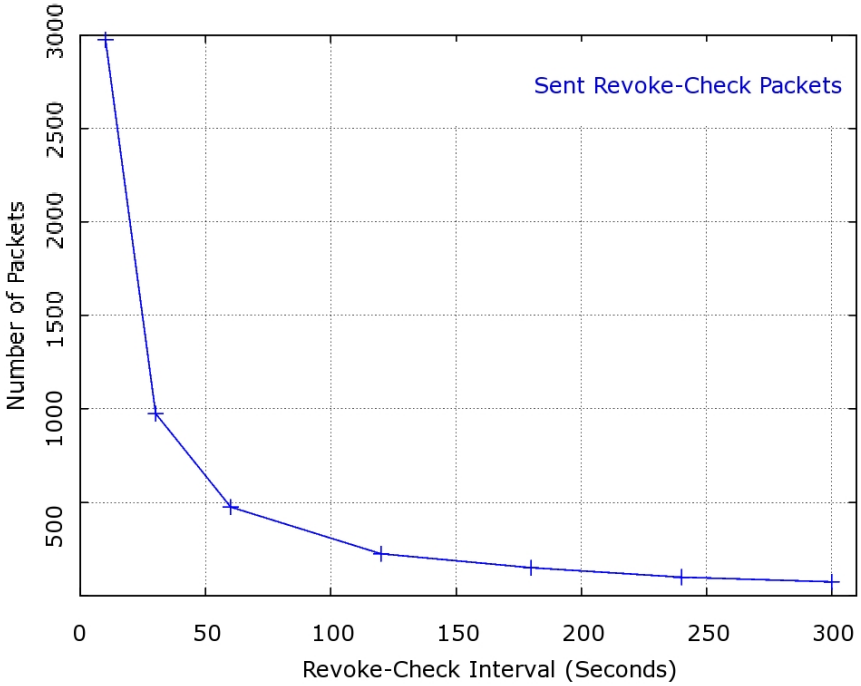


Figure 6.7.: Sent Revoke-Check Messages

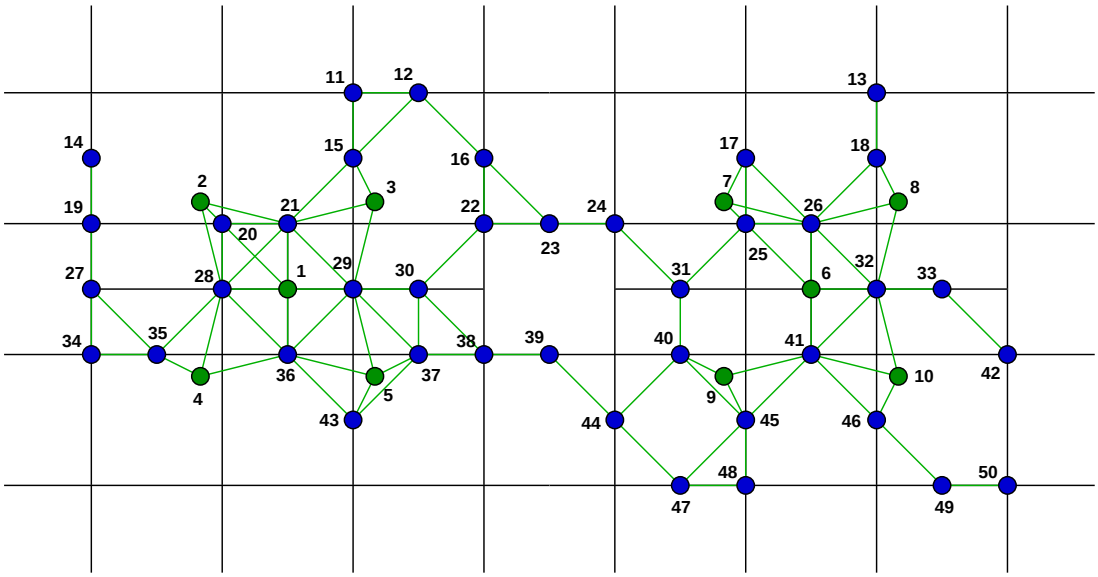


Figure 6.8.: Simulation with 50 Nodes, no Mobility

7. Summary and Outlook

This thesis presented a framework for secure and efficient communication in mobile ad hoc networks (MANETs). It started with an introduction into MANETs and 4G networks and outlined the performance constraints of such systems. Hints for a practical realization of such systems using IEEE 802.11 have been given and the necessity of security mechanisms in such systems has been motivated. An overview about existing security mechanisms and its suitability for MANETs has been provided. As completely distributed systems result in a high signaling overhead or lack scalability properties, LKN-ASF has been presented as framework for secure and efficient communication in MANETs within 4G networks. The framework, its description in SDL and simulation results for different scenarios have been described.

While the framework provides the basis and distributes credentials for efficient and scalable security mechanisms, its application alone does not result in a secure system. Possible enhancements include the integration of LKN-ASF in MANET routing protocols, privacy mechanisms based on LKN-ASF [Eic03] and efficient revocation mechanisms for large user groups.

Some of these topics are regarded in ongoing work. [Car04] described the integration of LKN-ASF into a MANET routing protocol for highly mobile vehicular environments. The AODV routing protocol has been chosen for this task and resulted in encouraging simulation results of the system. Relatively high delays during the route setup phase have been found to occur due to certificate validation at intermediate hops along a route. This delays, however, are expected to be minimized in the near future due to the increase in available processing power. Other ongoing activities in this area include the combination of LKN-ASF with mechanisms for efficient certificate revocation. This is essential if LKN-ASF is used in scenarios with large groups of potential users, e.g. the system envisioned in [EC03]. The work currently done in [Bru04] is based on quasimodo trees [EGR04] to achieve this task.

A. Mobile Testbed - Hardware Setup

All measurements have been done using three up-to-date laptops running SuSE Linux 8.1.

IEEE 802.11 PCMCIA cards from ORiNOCO/agere have been used for the measurements. The chip set specification described in the data sheet[1] is summarized below:

Nominal Output Power	15 dBm
Receiver Sensitivity	-82 dBm at 11Mbps -87 dBm at 5.5 Mbps -91 dBm at 2 Mbps -92 dBm at 1 Mbps
Delay Spread (at FER of <1%)	65ns at 11 Mbps 225ns at 5.5 Mbps 400ns at 2 Mbps 500ns at 1 Mbps

Table A.1.: IEEE 802.11b PCMCIA Cards: Chipset Specification

To measure the cars' position, GPS devices from the Garmin[2] eMap series have been selected. For the measurements, the firmware has been upgraded to version 2.90

[1] http://www.orinocowireless.com
[2] http://www.garmin.com/products/emap

B. X.509 v3 Certificate

```
Certificate:
   Data:
       Version: 3 (0x2)
       Serial Number: 1 (0x1)
       Signature Algorithm: md5WithRSAEncryption
       Issuer: C=DE, ST=Bavaria, L=Munich, O=Institute of Communication Networks,\\
       OU=Mobile Communications Group,\\
       CN=Christian Schwingenschloegl/emailAddress=schwinge@lkn.ei.tum.de
       Validity
           Not Before: Jun 14 16:41:54 2004 GMT
           Not After : Apr 10 16:41:54 2005 GMT
       Subject: C=DE, ST=Bavaria, L=Munich, O=Institute of Communication Networks,
       OU=Mobile Communications Group,
       CN=Peter Tabery/emailAddress=peter.tabery@nomail.de
       Subject Public Key Info:
           Public Key Algorithm: rsaEncryption
           RSA Public Key: (2048 bit)
               Modulus (2048 bit):
                   00:e8:56:cf:02:25:89:59:d7:13:99:82:78:68:03:
                   e3:56:64:d0:0c:aa:5b:46:0c:c3:a6:7f:c9:1e:a8:
                   f6:2c:58:a6:4b:cc:a0:f8:ef:93:66:ad:c4:10:af:
                   de:c2:54:e4:66:65:06:e6:72:0e:63:f8:a4:e8:79:
                   80:e7:3a:71:ac:77:d2:e2:8f:2d:50:6e:d0:a9:c0:
                   06:a5:53:7b:da:33:05:26:da:c9:09:ff:a1:bb:79:
                   96:45:7f:54:c7:53:4a:0f:8d:a4:25:86:73:86:8b:
                   86:72:8b:62:1d:fe:39:e7:1c:9b:ad:99:97:7f:77:
                   17:e0:06:39:5f:5c:62:8a:85:12:af:52:9e:e3:51:
                   01:7d:48:26:7c:83:61:e1:11:40:76:16:8e:02:6a:
                   2a:72:29:f1:6b:91:3a:10:0a:34:98:11:f1:d7:b4:
                   45:03:71:28:fc:28:aa:64:38:ab:d9:03:1d:a6:a1:
                   78:d6:23:f4:ed:f2:4e:26:3a:3f:c6:7c:03:a7:d5:
                   2d:64:d8:55:11:04:c7:be:a2:94:be:b1:96:90:09:
                   b7:9b:16:bb:b8:8b:1b:61:d2:1d:8a:b0:0e:a7:85:
                   7f:ed:b0:d2:a9:40:35:56:af:2c:b9:b5:75:ef:0d:
                   0b:fc:79:a6:a1:1d:d3:af:71:db:bd:fc:94:b0:32:
                   74:df
               Exponent: 65537 (0x10001)
       X509v3 extensions:
           X509v3 Basic Constraints:
               CA:FALSE
           Netscape Comment:
               OpenSSL Generated Certificate
           X509v3 Subject Key Identifier:
               D9:9B:0A:FB:60:16:B3:FD:44:F5:9D:BC:7F:CF:32:A9:D7:96:83:D8
           X509v3 Authority Key Identifier:
```

```
                keyid:7E:DD:61:09:44:5A:62:AB:C3:41:86:52:F7:26:E7:BF:7A:F7:24:90
                DirName:/C=DE/ST=Bavaria/L=Munich/O=Institute of Communication
                Networks/OU=Mobile Communications Group/CN=Christian
                Schwingenschloegl/emailAddress=schwinge@lkn.ei.tum.de
                serial:00
    Signature Algorithm: md5WithRSAEncryption
        b6:24:c9:9c:dd:3a:da:4e:24:5b:63:fa:d3:5e:cd:9a:b6:f9:
        ed:32:1a:e3:c5:57:73:0f:01:eb:9a:d1:40:f2:45:06:5f:3d:
        cd:1b:cc:1b:95:a3:a7:83:c7:71:de:8e:10:de:98:51:12:7a:
        93:a8:e5:d8:52:f5:94:e2:50:cc:90:90:dd:a2:c0:27:23:4c:
        fa:80:ab:58:80:63:df:05:a6:12:1c:5e:94:f7:3d:7f:fa:c3:
        5f:b3:b9:3f:f0:34:fa:de:be:4f:db:42:44:b9:57:d5:41:be:
        27:50:6b:d1:66:a8:bc:8d:ab:f5:4a:3d:ad:78:0a:10:c1:d5:
        7b:c5:36:f9:c6:2d:47:ff:22:5b:34:79:ed:b7:68:00:c1:fb:
        de:2e:ed:25:79:d8:ca:e5:26:29:c4:03:0d:9e:7b:82:48:0f:
        aa:a6:be:93:2c:f6:35:0e:ba:0e:98:15:4f:65:0f:3b:50:93:
        21:83:f8:06:cb:1e:59:4f:2f:76:ee:77:84:a6:23:f1:a1:79:
        33:2d:d6:a1:65:83:a1:f7:94:6f:7a:0c:2f:11:a6:12:53:9b:
        1a:99:31:9c:f2:53:a1:00:b2:09:17:da:76:3a:07:a6:55:c9:
        91:3c:20:67:8a:4b:c0:11:bc:8d:d1:e0:40:19:a9:aa:dc:e3:
        71:2f:e1:66
-----BEGIN CERTIFICATE-----
MIIFbzCCBFegAwIBAgIBATANBgkqhkiG9w0BAQQFADCB0TELMAkGA1UEBhMCREUx
EDAOBgNVBAgTB0JhdmFyaWExDzANBgNVBAcTBk11bmljaDEsMCoGA1UEChMjSW5z
dGl0dXRlIG9mIENvbW11bmljYXRpb24gTmV0d29ya3MxJDAiBgNVBAsTG01vYmls
ZSBDb21tdW5pY2F0aW9ucyBHcm91cDEkMCIGA1UEAxMbQ2hyaXN0aWFuIFNjaHdp
bmdlbnNjaGxvZWdsMSUwIwYJKoZIhvcNAQkBFhZzY2h3aW5nZUBsa24uZWkudHVt
LmRlMB4XDTA0MDYxNDE2NDE1NFoXDTA1MDQxMDE2NDE1NFowgcIxCzAJBgNVBAYT
AkRFMRAwDgYDVQQIEwdCYXZhcmlhMQ8wDQYDVQQHEwZNdW5pY2gxLDAqBgNVBAoT
I0luc3RpdHV0ZSBvZiBDb21tdW5pY2F0aW9uIE5ldHdvcmtzMSQwIgYDVQQLExtN
b2JpbGUgQ29tbXVuaWNhdGlvbnMgR3JvdXAxFTATBgNVBAMTDFBldGVyIFRhYmVy
eTElMCMGCSqGSIb3DQEJARYWcGV0ZXIudGFiZXJ5QG5vbWFpbC5kZTCCASIwDQYJ
KoZIhvcNAQEBBQADggEPADCCAQoCggEBAOhWzwIliVnXE5mCeGgD41Zk0AyqW0YM
w6Z/yR6o9ixYpkvMoPjvk2atxBCv3sJU5GZlBuZyDmP4pOh5gOc6cax30uKPLVBu
0KnABqVTe9ozBSbayQn/obt5lkV/VMdTSg+NpCWGc4aLhnKLYh3+Oeccm62Zl393
F+AGOV9cYoqFEq9SnuNRAX1IJnyDYeERQHYWjgJqKnIp8WuROhAKNJgR8de0RQNx
KPwoqmQ4q9kDHaaheNYj9O3yTiY6P8Z8A6fVLWTYVREEx76ilL6xlpAJt5sWu7iL
G2HSHYqwDqeFf+2w0qlANVavLLm1de8NC/x5pqEd069x2738lLAydN8CAwEAAaOC
AV0wggFZMAkGA1UdEwQCMAAwLAYJYIZIAYb4QgENBB8WHU9wZW5TU0wgR2VuZXJh
dGVkIENlcnRpZmljYXRlMB0GA1UdDgQWBBTZmwr7YBaz/UT1nbx/zzKp15aD2DCB
/gYDVR0jBIH2MIHzgBR+3WEJRFpiq8NBhlL3Jue/evckkKGB16SB1DCB0TELMAkG
A1UEBhMCREUxEDAOBgNVBAgTB0JhdmFyaWExDzANBgNVBAcTBk11bmljaDEsMCoG
A1UEChMjSW5zdGl0dXRlIG9mIENvbW11bmljYXRpb24gTmV0d29ya3MxJDAiBgNV
BAsTG01vYmlsZSBDb21tdW5pY2F0aW9ucyBHcm91cDEkMCIGA1UEAxMbQ2hyaXN0
aWFuIFNjaHdpbmdlbnNjaGxvZWdsMSUwIwYJKoZIhvcNAQkBFhZzY2h3aW5nZUBs
a24uZWkudHVtLmRlggEAMA0GCSqGSIb3DQEBBAUAA4IBAQC2JMmc3TraTiRbY/rT
Xs2atvntMhrjxVdzDwHrmtFA8kUGXz3NG8wblaOng8dx3o4Q3phREnqTqOXYUvWU
4lDMkJDdosAnI0z6gKtYgGPfBaYSHF6U9z1/+sNfs7k/8DT63r5P20JEuVfVQb4n
UGvRZqi8jav1Sj2teAoQwdV7xTb5xi1H/yJbNHntt2gAwfveLu0ledjK5SYpxAMN
nnuCSA+qpr6TLPY1DroOmBVPZQ87UJMhg/gGyx5ZTy927neEpiPxoXkzLdahZYOh
95RvegwvEaYSU5samTGc8l0hALIJF9p2OgemVcmRPCBnikvAEbyN0eBAGamq3ONx
L+Fm
-----END CERTIFICATE-----
```

List of Figures

List of Tables

C. Bibliography

C.1. Own Publications

[Chr02] Christian Schwingenschlögl , *Ad-hoc Security Support Using DVB-T*, Proceedings of the 1st International Conference on Ad-Hoc Networks and Wireless (Michel Barbeau and Evangelos Kranakis, eds.), Proceedings in Informatics, no. 16, Carleton Scientific, 2002. 5.1, 5.3.2

[dVSX03] Vincent de Veyrac, Christian Schwingenschlögl, and Jin Xi, *Ad Hoc - Internet Gateway Implementation*, Tech. report, Technische Universität München, 2003. 2.4, 2.4.4

[ES01] Stephan Eichler and Christian Schwingenschlögl, *Simulation of IEEE 802.11b Performance in Vehicular Local Information Point Scenarios* , Tech. report, Technische Universität München, 2001. 3.2.5, 3.2.5.1, 3.2.5.2

[ESDE04] Stephan Eichler, Christian Schwingenschlögl, Florian Dötzer, and Jörg Eberspächer, *Secure Routing in a Vehicular Ad Hoc Network*, Accepted for IEEE Vehicular Technology Conference 2004-Fall (VTC2004-Fall): Wireless Technologies for Global Security, 2004. 5.2.6

[IS04] Matthias Ihmig and Christian Schwingenschlögl, *Throughput-Measurements in IEEE802.11*, Tech. report, Technische Universität München, 2004. 3.3.3.1

[JBS01] Eivind Jåsund, Christian Bettstetter, and Christian Schwingenschlögl, *A Service Browser for the Service Location Protocol Version 2 (SLPv2)*, Proceedings of the 7th EUNICE Open European Summer School (EUNICE'01) and the IFIP Workshop on IP and ATM Traffic Management (WATM'01), September 2001. 2.5.5

[KBS+01] Wolfgang Kellerer, Christian Bettstetter, Christian Schwingenschlögl, Peter Sties, Karl-Ernst-Steinberg, and Hans-Jörg Vögel. *(Auto)Mobile Communication in a Heterogeneous and Converged World*, IEEE Personal Communications Magazine 8 (2001), no. 6, 41–47. 1.1, 2.2, 4.4.5.1, 5

[KSB02] Timo Kosch, Christian Schwingenschlögl, and Christian Bettstetter, *Situative IP-basierte Fahrerinformationssysteme: Szenarien, Routing und prototypische Realisierung*, VDE Kongress 2002 Networlds, ITG-Fachtagung: Technologien und Anwendungen fr die mobile Informationsgesellschaft, October 2002. 2.2

[KSL02] Timo Kosch, Christian Schwingenschlögl, and Ai Li, *Information Dissemination in Multihop Inter-Vehicle Networks - Adapting the Ad-hoc On-demand Distance Vector Routing Protocol (AODV)*, The IEEE 5th International Conference on Intelligent Transportation Systems, September 2002. 2.2, 2.3.1

[MRS02] Bernd Müller-Rathgeber and Christian Schwingenschlögl, *Performance Tests of an Ad-hoc Testbed*, Tech. report, Technische Universität München, 2002. 3.3.1, 3.3.1.1

[Sch01] Christian Schwingenschlögl, *Secure Integration of IEEE 802.11 in University- and Enterprise Networks*, Proceedings of IST2001, Tehran, 2001. 4.3

[SGS01] Christian Schwingenschlögl, Ingo Gruber, and Karl-Ernst Steinberg, *IPCommServ, an architecture for an IP-based communication server for vehicles*, Tech. report, Institute of Communication Networks, TU München and BMW AG, EW14, 2001. 1.1, 2.2

[SH02] Christian Schwingenschlögl and Marc-Philipp Horn, *Building Blocks for Secure Communication in Ad-hoc Networks*, European Wireless 2002. Next Generation Wireless Networks: Technologies, Protocols, Services and Applications, February 2002. 4.4.2

[SS00] Christian Schwingenschlögl and Stefan Schönauer, *Application and Service Development using UML and SDL*, Eunice2000, Innovative Internet Applications, University of Twente, September 2000, pp. 31–38. 5.4

[STS03] David Schmidt, Peter Tabery, and Christian Schwingenschlögl, *Experimental Comparison of a UDP-Based Transport Protocol and TCP Over IEEE 802.11b*, Tech. report, Technische Universität München, 2003. 3.3.2

[SX02] Christian Schwingenschlögl and Jin Xi, *Effects of Physical Layer Abstraction in Ad-Hoc Network Simulation*, Proceedings SCI 2002. The 6th World Multi-Conference on Systemics, Cybernetics and Informatics, July 2002. 3.2.4, 3.2.4.0.2, 3.2.4.2, 3.2.4.4

[VBS01] Marco Vettorello, Christian Bettstetter, and Christian Schwingenschlögl, *Some Notes on Security in the Service Location Protocol Version 2 (SLPv2)*, Proceedings of the workshop on ad hoc communications, held in conjunction with ECSCW2001, 2001. 4.4.5.2, 4.4.5.3

C.2. Other Publications

[AHNRR02] B. Awerbuch, D. Holmer, C. Nita-Rotaru, and H. Rubens, *An On-Demand Secure Routing Protocol Resilient to Byzantine Failures*, ACM Workshop on Wireless Security (WiSe), 2002. 4.4.3

[air] *AirSnort Homepage*. URL `http://airsnort.shmoo.com/`. 4.3

[aod] *NIST Wireless Communication Technologies Group: Kernel AODV*. URL `http://www.antd.nist.gov/wctg/aodv_kernel/`. 3.3.1

[ARPK] Henri Moelard Anand R. Prasad and Jan Kruys, *Security Architecture for Wireless LANs: Corporate and Public Environment*, Proceedings VTC2000, pp. 283–287. 4.3

[AW98] Hagit Attiya and Jennifer Welch, *Distributed Computing - Fundamentals, Simulations and Advanced Topics*, McGraw-Hill Publishing Company, 1998. 4.5

[BBC+01] L. Blazevic, L. Buttyan, S. Capkun, S. Giordano, J. Hubaux, and J. Le Boudec. *Self-organization in mobile ad-hoc networks: the approach of terminodes*, IEEE Communications Magazine (2001). 2.3.2.5.2

[BBWBG98] S. Blackburn, S. Blake-Wilson, M. Burmester, and S. Galbraith, *Shared generation of shared RSA keys (CORR98-19)*, Tech. report, Department of Combinatorics and Optimization, University of Waterloo, 1998. 4.4.2.1

[BCSW98] S. Basagni, I. Chlamatac, V. Syrotiuk, and B. Woodward, *A Distance Routing Effect Algorithm for Mobility (DREAM)*, Proceedings of 4th Annual ACM/IEEE International Conference on Mobile Computing and Networking (MOBICOM), 1998, pp. 76–84. 2.3.2.5.1, 2.3.2.5.2

[BDSZ94] V. Bharghavan, A. Demers, S. Shenker, and L. Zhang, *A medium access control for multihop networks*, Proceedings of ACM SIGCOMM '94, August 1994. 3.2.3.3

[BE81] D. J. Baker and A. Ephremides. *The Architectural Organization of a Mobile Radio Network via a Distributed Algorithm*, IEEE Transactions on Communications (1981), 1694–1701. 2.3.2.3, 2

[Bec95] Alan Beck. *Netscape's export SSL broken by 120 workstations and one student*, HPCwire (1995). 4.3

[BF97] D. Boneh and M. Franklin, *Efficient generation of shared RSA keys*, Advances in Cryptology - Crypto '97, Lecture Notes in Computer Science 1294, Springer Verlag, 1997, pp. 425–439. 4.4.2.1

[BGK+96] R. Bagrodia, M. Gerla, L. Kleinrock, J. Short, and T.-C. Tsai, *A Hierarchical Simulation Environment for Mobile Wireless Networks*, Tech. report, University of California at Los Angeles, Computer Science Department, 1996. 2.3.1

[BGW01] Nikita Borisov, Ian Goldberg, and David Wagner, *Intercepting Mobile Communications: The Insecurity of 802.11*, Proceedings of 7th Annual International Conference on Mobile Computing and Networking, ACM Sigmobile, July 2001. 4.3

[BHBR01] Stefano Basagni, Kris Herrin, Danilo Bruschi, and Emilia Rosti, *Secure Pebblenets*, ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc), 2001. 2.3.1

[Bla02] Tim Blachnitzky, *HTTP-Proxy-Konzept fr Handover in einer heterogenen Mobilfunk- und Bluetoothumgebung*, Master's thesis, Technische Universität München, 2002. 2.2

[blu] *Bluetooth SIG*. URL `www.bluetooth.org`. 4.4.1

[BMSU99] P. Bose, P. Morin, I. Stojmenovic, and J. Urrutia, *Routing with guaranteed delivery in ad hoc wireless networks*, Proceedings of 3rd ACM International Workshop on Discrete Algorithms and Methods for Mobile Computing and Communications (DIAL M99), 1999, pp. 48–55. 2.3.2.5.2, 3.1.3.2

[BMT+98] R. Bagrodia, R. Meyer, M. Takai, Y. Chen, X. Zeng, J. Martin, B. Park, and H. Song. *Parsec: A Parallel Simulation Environment for Complex Systems*, IEEE Computer (1998). 6.1

[Bru04] Bernhard Bruhn, *Certificate Revocation in Ad Hoc Networks*, Master's thesis, Technische Universität München, 2004, Ongoing Thesis in cooperation with DoCoMo Communications Laboratories USA. Advisors: Stephan Eichler and Christian Schwingenschlögl. 7

[BTAB99] Lokesh Bajaj, Mineo Takai, Rajat Ahuja, and Rajive Bagrodia, *Simulation of Large-Scale Heterogeneous Communication Systems*, Proceedings of MILCOM'99, 1999. 6.1

[BWC03] BWCS, *W-LAN Continuum*, Tech. report, BWCS Telecommunications Consulting and Communications, 2003. URL `www.bcws.com`. 7

[Car04] Francisco Javier Fabra Caro, *Secure Routing in an 802.11 based Vehicular Ad hoc Network*, Master's thesis, Technische Universität München, June 2004. 7

[Cas99] Castro and Liskov, *Authenticated Byzantine Fault Tolerance Without Public-Key Cryptography*, Tech. Report MIT/LCS/TM-589, MIT, Laboratory for Computer Science, 1999. 4.4.2.5

[Cha81] D. L. Chaum. *Untraceable Electronic Mail, Return Addresses, and Digital Pseudonyms*, Communications of the ACM (1981), no. 24, 84–88. 4.4.3

[CHB03] Srdjan Capkun, Jean-Pierre Hubaux, and Levente Buttyán, *Mobility Helps Security in Ad Hoc Networks*, Proceedings of the ACM Symposium on Mobile Ad Hoc Networking and Computing (MobiHOC), 2003. 6.3.4

[CJ03] T. Clausen and P. Jacquet, *Optimized Link State Routing Protocol (OLSR)*, IETF Experimental RFC, October 2003. URL `http://www.ietf.org/rfc/rfc3626.txt`. 2.3.2, 2.3.2.1.1, 3.1.3.1

[CL99a] Castro and Liskov, *A Correctness Proof for a Practical Byzantine-Fault-Tolerant Replication Algorithm*, Tech. Report MIT/LCS/TM-590, MIT, Laboratory for Computer Science, 1999. 4.4.2.5

[CL99b] Castro and Liskow, *Practical Byzantine Fault Tolerance*, Proceedings of the Third Symposium on Operating Systems Design and Implementation, 1999. 4.4.2.5

[Com96] ETSI STC-RES10 Committee, *Radio equipment and systems: HIPERLAN type 1*, Functional Specifications ETS 300-652, June 1996. 2.3.2.1.1

[Com99] IEEE LAN/MAN Standards Committee, *IEEE Std. 802.11-1999, Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications*, August 1999. 3.2.3.1, 3.2.3.2, 4.3, 4.4.1

[CP01] Phil Cracknell and Tim Pickard, *Out of Thin Air ... A Wireless Network Security Survey of London*, Tech. report, Orthus Information Security Solutions and RSA Security, 2001. 4.3

[DH98] C. Deering and R. Hinden, *IPv6 specification*, IETF RFC 2460, December 1998. 2.4

[DL99] Ralph Droms and Ted Lemon, *The DHCP Handbook*, Macmillan Technical Publishing, 1999. 2.5.1

[Dol95] Damien Doligez, *SSL Challenge Virtual Press Conference*, 1995. URL `http://pauillac.inria.fr/~doligez/ssl/press-conf.html`. 4.3

[Dou02] John R. Douceur, *The Sybil Attack*, 1st International Workshop on Peer-to-Peer Systems (IPTPS), March 2002. 4.4.1

[ea78] R. Kahn et al. *Advances in Packet Radio Technology*, Proceedings of the IEEE (1978), 1468–1496. 2.3.1

[ea97] H. Balakrishnan et al. *A Comparison of Mechanisms for Improving TCP Performance over Wireless Links*, IEEE/ACM Transactions on Networking (1997). 3.3.2

[ea98] J. Broch et al., *A Performance Comparison of Multihop Wireless Ad Hoc Network Routing Protocols*, Proceedings of IEEE/ACM MOBICOM'98, October 1998, pp. 85–97. 3.2.1

[ea99] D. Maltz et al. *The Effects of On-demand Behavior in Routing Protocols for Multihop Wireless Ad Hoc Networks*, IEEE JSAC 17 (1999), no. 8. 3.2.1

[Ebe99] Jörg Eberspächer (ed.), *Vertrauenswürdige Telekommunikation*, Hüthig Verlag, Heidelberg, 1999. 1.2

[EC03] ERTICO and GST Consortium, *GST - Global System for Telematics enabling On-line Safety Services*, Accepted Integrated EC FP6 Project Proposal, 2003. 1.1, 5.2.1, 7

[EGHK99] D. Estrin, R. Govindan, J. Heidemann, and S. Kumar, *Scalable Coordination in Sensor Networks*, Fifth Annual ACM/IEEE International Conference on Mobile Computing and Networks (MOBICOM), August 1999, pp. 263–270. 2.3.1.1

[EGR04] Farid F. Elwailly, Craig Gentry, and Zulfikar Ramzan, *QuasiModo: Efficient Certificate Validation and Revocation*, Public Key Cryptography - PKC 2004: 7th International Workshop on Theory and Practice in Public Key Cryptography (Feng Bao, Robert Deng, and Jianying Zhou, eds.), Lecture Notes in Computer Science, Springer-Verlag Heidelberg, February 2004, pp. 375–388. 7

[Eic03] Stephan Eichler, *Entwurf und Leistungsanalyse eines Protokolls zur sicheren Kommunikation in gatewaygestützten Ad-hoc Netzen*, Master's thesis, Technische Universität München, 2003. 5.4, 6.3.2, 7

[ET04] Jörg Eberspächer and Heinz Thielmann (eds.), *Sicherheit und Schutz in der Informationsgesellschaft*, Hüthig Telekommunikation, Bonn, 2004. 1.2

[EWB87] A. Ephremides, J. E. Wieselthier, and D. J. Baker. *A Design Concept for Reliable Mobile Radio Networks with Frequency Hopping Signaling*, Proceedings of the IEEE (1987), 56–73. 2.3.2.3, 2

[fam] *The Familiar Project*. URL `http://familiar.handhelds.org`. 3.3.1

[FB87] W. Fifer and F. Bruno. *The Low-Cost Packet Radio*, Proceedings of the IEEE (1987), 33–42. 2.3.1

[FJLA97] Chane L. Fullmer and J.J.Garcia-Luna-Aceves, *Solution to hidden terminal problems in wireless networks*, Proceedings of ACM SIGCOMM'97, September 1997. 3.2.3.3

[FV03] K. Fall and K. Varadhan, *The ns Manual (formerly ns Notes and Documentation)*, Tech. report, UC Berkeley, LBL, USC/ISI, and Xerox PARC, 2003. URL `http://www.isi.edu/nsnam/ns/doc/index.html`. 3.2.1

[GGK01] P. Gupta, R. Gray, and P.R. Kumar, *An experimental scaling law for ad hoc networks*, Tech. report, University of Illinois at Urbana-Champaign, May 2001. 3.2.3.2

[GH99] S. Giordano and M. Hamdi, *Mobility management: The virtual home region*, Tech. report, October 1999. 3.1.3.2

[GK00] P. Gupta and P. R. Kumar, *The Capacity of Wireless Networks*, IEEE Transactions on Information Theory, March 2000, pp. 388–404. 3.1.2, 3.1.2, 3.1.2, 3.2.3.2

[glo] *GloMoSim: Global Mobile Simulation Systems Library*. URL `http://pcl.cs.ucla.edu/projects/glomosim`. 3.2.4.0.2, 3.2.5.1, 5.4, 6

[GPVD99] Erik Guttman, Charles Perkins, John Veizades, and Michael Day, *Service Location Protocol, Version 2*, IETF RFC 2608, June 1999. 2.5, 2.5.1, 2.5.2, 2.5.4, 4.4.5, 4.4.5.1

[gri04] *The Grid Project Homepage*, 2004. URL `http://www.pdos.lcs.mit.edu/grid/`. 2.3.2.5.2, 3.1.3.2

[GT95] M. Gerla and J. T.-C. Tsai. *Multicluster, Mobile, Multimedia Radio Network*, Wireless Networks (1995), 255–265. 2, 2.3.2.3

[HBE+00] John Heidemann, Nirupama Bulusu, Jeremy Elson, Chalermek Intanagonwiwat, Kun chan Lan, Ya Xu, Wei Ye, Deborah Estrin, and Ramesh Govindan, *Effects of detail in wireless network simulation*, SCS Communication Networks and Distributed Systems Modeling and Simulation Conference, September 2000. 3.2.4.1

[HBE+01] Hannes Hartenstein, Bernd Bochow, Andre Ebner, Matthias Lott, Markus Radimirsch, and Dieter Vollmer, *Position-aware Ad Hoc Wireless Networks for Inter-Vehicle Communications: The Fleetnet Project*, ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc), 2001. 2.3.1

[HL99] Z. J. Haas and B. Liang. *Ad hoc Mobility Management with Uniform Quorum Systems*, IEEE Transactions on Communications (1999), 228–240. 2.3.2.5.1, 3.1.3.2

[HMSU03] Dieter Hutter, Günter Müller, Werner Stephan, and Markus Ullmann (eds.), *Security in Pervasive Computing - First International Conference, Boppard, Germany*, Lecture Notes in Computer Science, LNCS 2802, Springer, 2003. 4.4.5

[Hor01] Marc-Philipp Horn, *Secret und Function Sharing in mobilen Ad Hoc Netzwerken*, Master's thesis, Technische Universität München, 2001. 4.4.2

[HP98] Z. J. Haas and M. R. Pearlman, *The Performance of Query Control Schemes for the Zone Routing Protocol*, Proceedings of SIGCOMM, September 1998, pp. 167–177. 2.3.2.4

[HPJ02a] Y. Hu, A. Perrig, and D. Johnson, *Ariadne: A Secure On-Demand Routing Protocol for Ad Hoc Networks*, The 8th ACM International Conference on Mobile Computing and Networking (MobiCom), September 2002. 4.4.3

[HPJ02b] Y.-C. Hu, A. Perrig, and D.B. Johnson, *Wormhole Detection in Wireless Ad Hoc Networks*, Tech. report, Rice University, Dept. of Computer Science, June 2002. 4.4.1

[HPJ03] Yih-Chun Hu, Adrian Perrig, and David B. Johnson, *Packet Leashes: A Defense against Wormhole Attacks in Wireless Ad Hoc Networks*, Proceedings of the Twenty-Second Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM), April 2003. 4.4.3

[Hub01] Martin Huber, *Veränderung der Wertschöpfung im Mobilfunk durch Mobile Ad Hoc Netze: Eine ökonomische und technische Analyse*, Master's thesis, Technische Universitt München, Lehrstuhl für Kommunikationsnetze, 2001. 2.1

[IET] IETF, *MANET Charter*. URL `http://www.ietf.org/proceedings/01dec/183.htm`. 2.3

[iet03] *Internet Engineering Taskforce (IETF)*, 2003. URL `http://www.ietf.org`. 1.4

[ipe] *Iperf Version 1.7.0*. URL `http://dast.nlanr.net/Projects/Iperf/`. 3.3.3.1

[ipt] *netfilter: firewalling, NAT and packet mangling for Linux 2.4*. URL `http://www.netfilter.org/`. 3.3.1

[ird] *Infrared Data Association*. URL `http://www.irda.org`. 4.4.1

[Jav97] J. Javgren, *NTDR Mobility Management Protocols and Procedures*, Proceedings of the IEEE Military Communications Conference (MILCOM), November 1997. 2.3.2.3

[JL00] P. Jacquet and A. Laouiti, *Analysis of mobile ad-hoc networking routing protocols in random graph models*, Tech. Report RR-3835, INRIA, 2000. 3.1.3.1

[JM96] D. Johnson and D. Maltz, *Mobile Computing*, ch. 5 - Dynamic Source Routing, Kluwer Academic Publishers, 1996. 3.1.3.2

[JMH03] David B. Johnson, David A. Maltz, and Yih-Chun Hu, *The Dynamic Source Routing Protocol for Mobile Ad Hoc Networks (DSR)*, IETF Internet Draft, April 2003. URL `http://www.ietf.org/internet-drafts/draft-ietf-manet-dsr-09.txt`. 2.3.2, 2.3.2.2, 6.2

[JT87] J. Jubin and J. D. Tornow. *The DARPA Packet Radio Network Protocols*, Proceedings of the IEEE (Special Issue, Packet Radio Networks) 75 (1987), no. 1, 21–32. 2.3.1

[JV00] Philippe Jacquet and Laurent Viennot, *Overhead in Mobile Ad-hoc Network Protocols*, Tech. report, Institut National de Recherche en Informatique et en Automatique (INRIA), June 2000. 3.1.3, 3.1.3.1

[KHG03] Jiejun Kong, Xiaoyan Hong, and Mario Gerla, *An ANonymous On Demand Routing Protocol with Untraceable Routes for Mobile Ad-hoc Networks*, Tech. Report TR-030020, UCLA Computer Science Department, April 2003. 4.4.3

[KK00] B. Karp and H. T. Kung, *Greedy perimeter stateless routing for wireless networks*, Proceedings of the 6th Annual ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom), August 2000, pp. 243–154. 2.3.2.5.2, 3.1.3.2

[KKP99] J. M. Kahn, R. H. Katz, and K. S. J. Pister, *Mobile Networking for "Smart Dust"*, Fifth ACM/IEEE International Conference on Mobile Computing and Networking (MOBICOM), August 1999, pp. 271–278. 2.3.1.1

[Kum00] P. R. Kumar. *New technological vistas for systems and control: The example of wireless networks*, IEEE Control Systems Magazine (2000), 24–37. 3.2.3.3

[KV00] Y.-B. Ko and N. H. Vaidya. *Location Aided Routing (LAR) in mobile ad hoc networks*, ACM/Baltzer Wireless Networks (WINET) journal (2000), 307–321. 2.3.2.5.2

[KW03] Chris Karlof and David Wagner, *Secure Routing in Wireless Sensor Networks: Attacks and Countermeasures*, Proceedings of First IEEE International Workshop on Sensor Network Protocols and Applications, May 2003. 4.4.1, 4.4.3

[LÓ2] Juan Fco Escuderos López, *Integration of Heterogeneous Wireless Networks into an Automobile Ad-hoc Environment*, Master's thesis, Technische Universität München, 2002. 2.2

[LBC+01] Jinyang Li, Charles Blake, Douglas S. J. De Couto, Hu Imm Lee, and Robert Morris, *Capacity of Ad Hoc wireless network*, Mobile Computing and Networking (MobiCom), 2001, pp. 61–69. URL `http://citeseer.nj.nec.com/li01capacity.html"`. 2.3.2.1, 3.1.2

[LG97] C. R. Lin and M. Gerla. *Adaptive Clustering for Mobile Wireless Networks*, IEEE Journal on Selected Areas of Communications (1997), 1265–1275. 2.3.2.3

[Li02] Ai Li, *Design and Implementation of a Geocast Enhancement for the AODV Routing Protocol*, Master's thesis, Technische Universität München, 2002. 2.2

[LJC+00] J. Li, J. Jannotti, D. S. J. De Couto, D. R. Karger, and R. Morris, *A scalable location service for geographic ad hoc routing*, Proceedings of the 6th Annual ACM/IEEE International Conference on Mobile Computing and Networking (MOBICOM), 2000, pp. 120–130. 2.3.2.5.1, 3.1.3.2

[LNT02] Henrik Lundgren, Erik Nordstrm, and Christian Tschudin. *The Gray Zone Problem in IEEE 802.11b based Ad hoc Networks*, ACM Mobile Computing and Communications Review 6 (2002), no. 3. 3.1.1

[MC98] J. Macker and M. S. Corson. *Mobile Ad Hoc Networking and the IETF*, ACM Mobile Computing and Communication Review (1998), 9–12. 2.3.2

[MD02] Mahesh K. Marina and Samir R. Das, *Routing Performance in the Presence of Unidirectional Links in Multihop Wireless Networks*, Mobihoc02, June 2002. 3.1.1

[mob95] *A Survey of Defence Technology: The Software Revolution - To Dissolve, to Disappear*, The Economist (1995). 2.3.1

[MP99] Brent Miller and Robert Pascoe, *Mapping Salutation Architecture APIs to Bluetooth Service Discovery Layer, Version 1.0*, July 1999. URL `http://www.bluetooth.com`. 2.5

[MWH01] M. Mauve, J. Widmer, and H. Hartenstein. *A survey on position-based routing in mobile ad hoc networks*, IEEE Network Magazine (2001), 30–39. 2.3.2.5, 3.1.3, 3.1.3.2

[Nad02] Antar Nader, *Development and Performance Analysis of a Secure Routing Algorithm for Ad-Hoc Networks*, Master's thesis, Technische Universität München, 2002. 5.1, 5.3.2

[net] *The Public Netperf Homepage*. URL `http://www.netperf.org/`. 3.3.1.1

[ns-] *ns-2: The Network Simulator*. URL `http://www.isi.edu/nsnam/ns/`. 3.2.4.0.2

[nto] *ntop - network top*. URL `http://www.ntop.org/`. 3.3.1.1

[Nyk00] Toni Nykänen, *Attribute Certificates in X.509*, Tech. report, Helsinki University of Technology, 2000. 5.2

[opn] *OPNET Modeler*. URL `http://www.opnet.com/products/modeler/`. 3.2.4.0.2

[OTL03] R. Ogier, F. Templin, and M. Lewis, *Topology Dissemination Based on Reverse-Path Forwarding (TBRPF)*, IETF Internet Draft, October 2003. URL `http://www.ietf.org/internet-drafts/draft-ietf-manet-tbrpf-11.txt`. 2.3.2, 2.3.2.1.2

[Par94] A. K. Parekh. *Selecting Routers in Ad-Hoc Wireless Networks*, Proceedings of the SBT/IEEE International Telecommunications Symposium (1994). 2

[PB94] C. E. Perkins and P. Bhagwat. *Highly Dynamic Destination-Sequenced Distance-Vector Routing (DSDV) for Mobile Computers*, ACM SIGCOMM '94 Computer Communications Review (1994), 234–244. 2.3.2.2.1

[PBRD03] C. Perkins, E. Belding-Royer, and S. Das, *Ad hoc On-Demand Distance Vector (AODV) Routing*, IETF Experimental RFC, July 2003. URL `http://www.ietf.org/rfc/rfc3561.txt`. 2.3.2, 2.3.2.2, 6.2

[PCST01] Adrian Perrig, Ran Canetti, Dawn Song, and J.D. Tygar, *Efficient and Secure Source Authentication for Multicast*, Network and Distributed System Security Symposium, NDSS, February 2001, pp. 35–46. 4.4.3

[PCTS00] Adrian Perrig, Ran Canetti, J.D. Tygar, and Dawn Song, *Efficient Authentication and Signing of Multicast Streams over Lossy Channels*, IEEE Symposium on Security and Privacy, May 2000, pp. 56–73. 4.4.3

[Per96] C. Perkins, *IP Mobility Support*, IETF RFC 2002, October 1996. 2.3.1.1

[Per99] Radia Perlman. *An Overview of PKI Trust Models*, IEEE Network (1999), 38–43. 5.1, 5.2

[Per00] Charles E. Perkins, *Ad Hoc Networking*, Addison-Wesley, 2000. 2.3.1

[Per02] Antonio Delgado Peris, *Discovery, Filtering and Management of Services in Automobile Ad-hoc networks*, Master's thesis, Technische Universität München, 2002. 2.2

[PG99] Charles Perkins and Erik Guttmann, *The String Representation of LDAP Search Filters*, IETF RFC 2254, June 1999. 2.5.1

[PH99] M. R. Pearlman and Z. J. Haas. *Determining the Optimal Configuration of the Zone Routing Protocol*, IEEE Journal on Selected Areas of Communications (Special Issue Ad-Hoc Networks) (1999). 2.3.2.4

[PH02] Panagiotis Papadimitratos and Zygmunt J. Haas, *Secure Routing for Mobile Ad hoc Networks*, Proceedings of the SCS Communication Networks and Distributed Systems Modeling and Simulation Conference (CNDS), January 2002. 4.4.3

[pin] *The Story of the PING Program*. URL `http://ftp.arl.mil/~mike/ping.html`. 3.3.3.1

[Pra99] R. Prakash, *Unidirectional Links Prove Costly in Wireless Ad Hoc Networks*, Third International Workshop on Discrete Algorithms and Methods for Mobile Computing and Communications (DIALM), August 1999, pp. 15–22. 3.1.1

[PRDM01] Charles E. Perkins, Elizabeth M. Royer, Samir R. Das, and Mahesh K. Marina. *Performance Comparison of Two On-Demand Routing Protocols for Ad Hoc Networks*, IEEE Personal Communications (2001), 16–28. 3.2.1, 3.2.2, 5.3.2

[Rap95] T.S. Rappaport, *Wireless Communications: Principles and Practice*, Prentice Hall, 1995. 3.2.4.0.2

[Rei03] Thomas Reicher, *A Framework for Dynamically Adaptable Augmented Reality Systems*, Ph.D. thesis, Technische Universität München, 2003. 1.1, 2.5

[Ren00] Christoph Renner, *Service Discovery and Profile Mobility in a Car Environment*, Master's thesis, Technische Universität München, 2000. 2.5.4

[RK01] R. Rozovsky and P.R. Kumar, *SEEDEX: A MAC protocol for ad hoc networks*, Proceedings of The 2001 ACM International Symposium on Mobile Ad Hoc Networking & Computing (MobiHoc 2001), October 2001, pp. 67–75. 3.2.3.3

[RPKG88] I. M. Jacobs R. P. Kosowsky and K. S. Gilhousen, *ARNS: A new link layer protocol*, Proceedings IEEE MILCOM 88, September 1988, pp. 515–519. 3.2.3.3

[RSA78] R. L. Rivest, A. Shamir, and L. A. Adleman. *A Method for Obtaining Digital Signatures and Public-Key Cryptosystems*, Communications of the ACM 21 (1978), no. 2, 120–126. URL `http://citeseer.nj.nec.com/rivest78method.html`. 5.2

[Sch96] Bruce Schneier, *Applied Cryptography*, John Wiley & Sons, Inc., 1996. 4.3

[SDFY94] A. De Santis, Y. Desmedt, Y. Frankel, and M. Yung, *How to Share a Function Securely*, 26th Annual ACM Symposium on Theory of Computing, 1994, pp. 522–533. 4.4.2.2

[sdl] *Sdl forum society*. URL `http://www.sdl-forum.org/`. 5.4

[SDL+02] Kimaya Sanzgiri, Bridget Dahill, Brian Neil Levine, Clay Shields, and Elizabeth M. Belding-Royer, *A Secure Routing Protocol for Ad Hoc Networks*, Proceedings of 2002 IEEE International Conference on Network Protocols (ICNP), November 2002. 4.4.3

[Sie04] Richard Sietmann, *Hot Spot für Hotspots*, Heise Newsticker, January 2004. URL `http://www.heise.de/mobil/newsticker/meldung/43426`. 7

[SRK03a] S. Shakkottai, T. S. Rappaport, and P. C. Karlsson, *Cross-Layer Design for Wireless Networks*, Tech. Report TR-2003-04-00001, University of Texas at Austin, 2003. 3.3.2

[SRK03b] Sanjay Shakkottai, Theodore S. Rappaport, and Peter C. Karlsson. *Cross-Layer Design for Wireless Networks*, IEEE Communications Magazine (2003). 3.3.2

[Sta02] Frank Stajano, *Security for Ubiquitous Computing*, Wiley Series in Communications Networking & Distributed Systems, John Wiley & Sons, Ltd, 2002. 4.4.5

[Ste04] Stephan Eichler, *Security Challenges in MANET-based Telematics Environments*, Proceedings of 10th Open European Summer School and IFIP WG 6.3 Workshop, June 2004. 6.3.2

[Sto99] I. Stojmenovic, *Home agent based location update and destination search schemes in ad hoc wireless networks*, Tech. report, Computer Science, SITE, University of Ottawa, September 1999. 3.1.3.2

[Stu02] Bernd Sturm, *TCP/IP over Satellite*, Master's thesis, Technische Universität München, 2002. 3.3.2

[SW87] N. Shacham and J. Westcott. *Future Directions in Packet Radio Architectures and Protocols*, Proceedings of the IEEE (1987), 83–99. 2.3.1

[SW96] J. Strater and B. Wollman, *OSPF Modeling and Test Results and Recommendations*, Tech. Report 96W0000017, Mitre, Xerox Office Products Division, March 1996. 2.3.2

[TBLG99] Mineo Takai, Rajive Bagrodia, Addison Lee, and Mario Gerla, *Impact of channel models on simulation of large scale wireless networks*, Proceedings of the 2nd ACM International Workshop on Modeling, Analysis and Simulation of Wireless and Mobile Systems, August 1999, pp. 7–14. 3.2.4.1

[TMB01] Mineo Takai, Jay Martin, and Rajive Bagrodia, *Effects of wireless physical layer modeling in mobile ad hoc networks*, Proceedings of the 2001 ACM International Symposium on Mobile ad hoc networking and computing (MobiHoc), 2001, pp. 87–94. 3.2.4.0.2, 3.2.4.1, 3.2.4.2, 3.2.4.3

[TN] S. Thomson and T. Narten, *IPv6 Stateless Address Autoconfiguration*, IETF RFC 2462. 2.4.2

[Tuc93] B. Tuch. *Development of WaveLAN, an ISM Band Wireless LAN*, AT&T Tech. J. 72 (1993), no. 4, 27–33. 3.2.1

[Uni] Carnegie Mellon University, *CERT, CERT/CC*. URL `http://www.cert.org`. 1.2

[Vet01] Marco Vettorello, *Security and Leasing Concepts in SLPv2*, Master's thesis, Technische Universität München, 2001. 2.5.4

[Was03] Christian Wasel, *Statistische Kostenoptimierung mit Handover und Prefetching in heterogenen Mobilfunknetzen*, Master's thesis, Technische Universität München, 2003. 2.2

[Wei93] Mark Weiser. *Some Computer Science Issues in Ubiquitous Computing*, Communications of the ACM (1993). 2.3.1.1

[wep] *WEPCrack - An 802.11 key breaker*. URL `http://wepcrack.sourceforge.net/`. 4.3

[wge] *Wget - Retrieves files from the Web*. URL `http://www.gnu.org/directory/wget.html/`. 3.3.2

[wir] *Wireless Tools for Linux*. URL `http://www.hpl.hp.com/personal/Jean_Tourrilhes/Linux/Tools.html`. 3.3.2, 3.3.3.1

[wwr03] *Wireless World Research Forum (WWRF)*, 2003. URL `http://www.wireless-world-research.org`. 1.4, 2.1, 2.1

[XB02] Jin Xi and Christian Bettstetter, *Wireless Multi-Hop Internet Access: Gateway Discovery, Routing, and Addressing*, Proceedings of International Conference on Third Generation Wireless and Beyond (3Gwireless), May 2002. 2.4, 2.4.2, 2.4.3, 3.2.4.3

[YNK01] Seung Yi, Prasad Naldurg, and Robin Kravets, *Security-Aware Ad-Hoc Routing for Wireless Networks*, Tech. Report UIUCDCS-R-2001-2241, UILU-ENG-2001-1748, Department of Computer Science, University of Illinois at Urbana-Champaign, August 2001. 4.4.3

[Zen02] Michael Zenz, *Security Solutions for WLANs Based on IPv6 and IPSec*, Master's thesis, Technische Universität München, 2002. 5.2.6

[ZH99] Lidong Zhou and Zygmunt J. Haas. *Securing Ad Hoc Networks*, IEEE Networks, Special Issue on Network Security (1999), 24–30. 4.4.2

[ZXSJ03] Sencun Zhu, Shouhuai Xu, Sanjeev Setia, and Sushil Jajodia, *LHAP: A Lightweight Hop-by-Hop Authentication Protocol For Ad-Hoc Networks*, Tech. report, Center for Secure Information Systems, George Manson University, Fairfax, VA and Department of Information and Computer Science, Univ. of California at Irvine, 2003. 4.4.3