Internet Gateway Discovery for Mobile Ad Hoc Networks

Matthias Rosenschon

Submitted for the Degree of Doctor of Philosophy

School of Engineering and Mathematical Sciences
CITY University London

2 October 2007

Contents

List of Tables

List of Figures

Acknowledgement

I would like to thank my supervisors Veselin Rakocevic Ph.D. from CITY University and Professor Dr.-Ing. Joachim Habermann from the University of Applied Sciences in Giessen-Friedberg, Germany for their assistance and support in writing this thesis. Further my thanks go to Dipl.-Ing. (FH) Nico Bayer and Dipl.-Ing. (FH) Dmitry Sivchenko for the multiple technical discussions, constructive questions, and programming support.

I am grateful to Frank Christahl M.Phil. and Andreas Huber M.Phil. for introducing me into wireless multihop ad-hoc networks, MobileIP and NS-2.

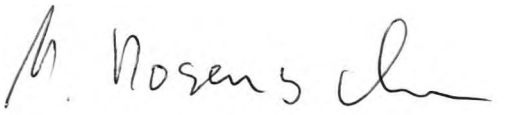
My thanks go to Dipl.-Inform. (FH) Markus Heurung for help with the NS-2 source code and implementing the first version of the gateway discovery algorithms and the Internet gateway. Further I would like to thank Dipl.-Inform. (FH) Tilmann Mänz for implementing and testing the first version of the HELLO message based Internet gateway discovery algorithm into NS-2.

This research has been partly sponsored by T-Systems. Therefore my thanks go to Dr.-Ing. Bangnan Xu and Dr. Sven Hischke. They have furthermore provided a lot of ideas and suggestions.

Further my thanks go to Dipl.-Ing. (FH) Horst Glücklich, Dipl.-Ing. (FH) Christopher Köhnen, and other people at the University of Applied Sciences in Giessen-Friedberg for support in the laboratory and for keeping the simulation computers running. Furthermore my thanks go to the staff at CITY University.

Last but not least, my thanks go to all my friends and especially to my parents and my grandfather who has passed away just before the thesis was handed in. Their unlimited and steady support was also essential for the creation of this thesis.

I grant powers of discretion to the University Librarian to allow this thesis to be copied in whole or in part without further reference to me. This permission covers only single copies made for study purposes, subject to normal conditions of acknowledgement.



Abstract

This thesis analyses Internet connectivity for *M*obile *A*d-hoc *NET*works (MANETs). A MANET consists of a number of mobile nodes, interconnected wirelessly, that together form a network without explicit routers and centralised instances. Ad-hoc routing protocols are utilised for discovering routes between nodes of the MANET whereas a route may consist of multiple relay nodes that forward data from a source to a destination node (multihop route).

In order to connect nodes of a MANET to the Internet an interface node is introduced. This interface node is called an Internet gateway that must initially be discovered by the MANET nodes to gain Internet connectivity. Therefore extended ad-hoc routing protocols are used.

In general, there exist two well-established approaches for discovering an Internet gateway. The first well-established approach is called the proactive approach, where Internet gateways flood the MANET periodically whereas in the second well-established approach nodes of the MANET solicit for Internet gateways reactively. Both approaches use MANET flooding for discovering an Internet gateway and MANET flooding is known to increase the protocol overhead in MANETS.

The main objective of the research presented in this thesis is to develop an Internet gateway discovery algorithm that avoids MANET flooding and to investigate the new algorithm in terms of the control message overhead and the provided throughput to the mobile nodes of the MANET by simulations. Additionally in the thesis a protocol efficiency index is derived from simulation results to allow a fast comparison between simulation results with different parameters.

This thesis presents the new HELLO message based algorithm for Internet gateway discovery. HELLO messages are a typical element of MANET routing protocols used for neighbourhood management and are now enhanced for Internet gateway discovery. By avoiding MANET flooding the control message overhead is minimised. The HELLO message based algorithm for Internet gateway discovery is examined and investigated by simulations and it shows a decrease of the control message overhead of up to 2.6 times compared to the well-established approaches in the simulated scenario setups.

Furthermore the thesis presents two extensions to the well-established and the HELLO message based Internet gateway discovery algorithms. The first extension utilises additional control messages in order to optimise a multihop route from the Internet gateway to an ad-hoc mobile node. This first extension applies only for proactive algorithms for Internet gateway discovery and it provides a benefit in terms of data throughput to mobile nodes by 10% in the scenario setup simulated in this thesis.

The second extension allows mobile nodes of a MANET to choose between multiple discovered Internet gateways. The selection to which Internet gateway a MANET node connects to is based upon a certain newly presented metric. This new metric is based on the length of a MANET route and the amount of traffic an Internet gateway is already forwarding. Simulations show a benefit of up to 438% in terms of throughput depending on the actual scenario setup.

Glossary

ACK	A message sent to acknowledge data or control messages. Used by e.g. TCP
ADV	Advertisements (ADVs) are flooded by Internet gateways into an ad-hoc network. Used by the advertisement based algorithm for Internet gateway discovery
AODV	Ad-hoc On-demand Distance Vector (AODV) is a reactive routing protocol for ad-hoc networks
AP	An access point (AP) is a device used in wireless LANs to provide connectivity to a wired network
AQOR	Ad hoc qos on-demand routing is an ad-hoc network routing protocol that provides quality of service to mobile nodes after a "usage" metric, i.e. traffic metric. Thus AQOR establishes least used multihop routes in ad-hoc networks
ARPANET	The ARPANET is the precursor of the modern Internet. It was developed by the U.S. in the 1960s
AS	An autonomous system (AS) consists of a number of subnetworks and is the highest network tier in the Internet hierarchy
BACK	After receiving a binding update message (BU) from a roaming node a home agent acknowledges the BU with a binding update acknowledge message called BACK
BGP	Border gateway protocol (BGP) connects autonomous systems. It is responsible for exchanging routing data between the highest tiers of the Internet hierarchy
BSS	A basic service set (BSS) consists of a number of mobile nodes and an access point (AP) to provide micro mobility to mobile nodes in a W-LAN environment
BSSID	The unique identifier of a BSS
BU	A binding update message (BU) is sent from a roaming mobile node to its home agent in the Internet to update the home agent's routing table about the mobile node's logical location in the Internet
C++	An object oriented extension to the programming language C
CBR	A special type of data traffic with constant traffic load (constant bit rate)
CIDR	The classless inter-domain routing (CIDR) notification is to reduce the amount of wasted Internet addresses by forming subnetworks of suitable sizes
CN	The correspondent node (CN) used in this thesis represents the Internet, the home network with the home agent of a mobile node, and the data source for simulations
CTS	Clear to send. Is a message of the RTS/CTS mechanism for the collision avoidance in wireless systems
CW	The contention window (CW) is a period of time in which nodes content for air time in wireless systems, e.g. W-LAN
DSDV	A proactive routing protocol for mobile ad-hoc networks (destination sequenced distance vector)
DSR	A reactive routing protocol for mobile ad-hoc networks (dynamic source routing)

DiffServ	By using preferences traffic from specific nodes can be privileged. This is achieved by using the DiffServ extension to routing protocols
DIFS	Distributed coordination Function IFS. A long waiting time (micro seconds between frames) in wireless systems
EGP	The protocol family for all routing protocols that are for interconnecting ASs. EGP is an exterior gateway protocol (EGP)
EUI	The extended unique identifier is built from a network interface's hardware MAC address to identify every network interface in a IPv6 environment
FA	A foreign agent (FA) is located in a network where mobile nodes are roaming to. The mobile nodes use FAs in MobileIPv4
FIFO	A strategy for buffering data. The one who's coming first will firstly be processed (first in first out)
FIN	A control message of TCP to stop a data connection
FTP	The file transfer protocol (FTP) is used for transferring data files
GRREP	An ad-hoc routing protocol extension message introduced by this thesis to optimise multihop routes in an Internet connected ad-hoc mobile network (gratuitous route reply)
GRREP_ACK	This message in introduced by this thesis to acknowledge a GRREP message
GSM	The global system for mobile communication (GSM) is the standard for e.g. mobile phones
GW	An Internet gateway (GW) is an interface between a mobile multihop ad-hoc network using ad-hoc routing and the Internet with its hierarchical routing approach
HA	A home agent (HA) is located in the home network of a mobile node. The mobile nodes update the HA's routing table by sending BU messages
HC	Some routing protocols use the hop count (HC) for routing decision. The hop count is the number of forwardings of packets
HELLO	A control message called HELLO to indicate a node's presence to surrounding neighbour nodes. Used in this thesis for gateway discovery in ad-hoc networks
HELLO_I	With an I-flag extended HELLO message for Internet gateway discovery used in this thesis
IEEE	Institute of Electrical and Electronics Engineers
IFS	Inter frame spacing (IFS) is a time nodes wait until they transmit a new frame (micro seconds). IFS is defined in three sizes, short IFS (SIFS), point coordination function IFS (PIFS), and distributed coordination function (DIFS)
IGP	Interior gateway protocol. The protocol family for all routing protocols that are for routing within an AS, OSPF and RIP are IGPs
IN	Intermediate node. A node that is part of a multihop ad-hoc route
IP	Internet protocol. A protocol located at layer 3 in the OSI reference model
ISO	International Organization for Standardization
IntServ	Quality of service extension to protocols. Allows reservation of resources, e.g. bandwidth
LAN	Local area network. A network of a small size. Larger than a PAN, smaller than a MAN

LL	Link layer. A protocol at layer 2 of the OSI reference model
MAC	Medium access. A protocol at layer 2 of the OSI reference model
MAN	Metropolitan area network. A network larger than a LAN, smaller than a WAN
MANET	Mobile ad-hoc network. A network consisting of a number of mobile nodes using ad-hoc routing
MN	A specific mobile node for performance evaluations by simulations in this thesis
MRP	Multi relay point. OLSR provides a special flooding strategy using MRPs to reduce overhead
NAT	Network address translation. Network nodes with private IP address are hidden behind a NAT router. This saves public IP addresses
NS-2	The network simulator 2 is a free software tool under the GPL
ODRP	On-demand delay-constrained unicast routing protocol (ODRP). A quality of service routing protocol for ad-hoc networks
OLSR	Optimized link state routing (OLSR). A routing protocol for ad-hoc networks
OPNET	A commercial network simulation software
OSI	Open Systems Interconnection Reference Model. This model defines the interaction between different network layers
OSPF	Open shortest path first. An interior gateway protocol
PAN	Personal area network. A network of a size smaller than LAN
PDA	Personal digital assistant. An electronic mobile device
PDF	Packet delivery fraction. The ratio of packets sent and packets received
PHY	Physical layer. The lowest layer of the OSI reference model
PIFS	Point coordination function IFS. A medium waiting time (micro seconds between frames) in wireless systems
RAN	Radio access network. GSM is a RAN
RAgent	Routing agent. This thesis uses NS-2 RAgents for routing decisions
RERR	Route error message. Used by AODV to indicate a broken route
RREP	Route reply message. Used by AODV as an answer to a RREQ
RREQ	Route request message. Used by AODV to find destination nodes in an ad-hoc network
RSVP	Resource reservation protocol. A protocol used for reserving quality of service resources
RTP	Real-time transport protocol. Used for real time applications like video or audio streams
RTS	Request to send. Is a message of the RTS/CTS mechanism used the collision avoidance in wireless systems
SIFS	Short IFS. A short waiting time (micro seconds between frames) in wireless systems
SOL	Solicitation message. Used in this thesis for the solicitation based Internet gateway discovery algorithm
SYN	A control message of TCP to start a data connection

SYN/ACK	The combination of a SYN and an ACK message in TCP. Used to start a data connection
TC	Topology control message. Used in the OLSR ad-hoc routing protocol
TCL	Tool command language. A common scripting language
TCP	Transmission control protocol. A very common used connection oriented protocol located at layer 4 of the OSI reference model
TDMA	Time division multiple access. A medium access method where subscribers are assigned to time slots
TTL	Time to live. Every time a packet is being forwarded its TTL is reduced by one. If the TTL equals zero the packet is discarded. Used to prevent infinite packet forwarding
UDP	User datagram protocol. A very common used connectionless protocol located at layer 4 of the OSI reference model
UMTS	Universal mobile telecommunications system. The successor of GSM
UWB	Ultra-wideband. A short range wireless system that provides high bandwidth
W-LAN	Wireless LAN. A commonly used medium range wireless system
WAN	Wide area network. A network of a big size. Larger than a MAN
WWW	World wide web. A commonly used Internet application
WiMAX	Worldwide interoperability for microwave access. A long range wireless system that provides high bandwidth

Chapter 1

Introduction

1.1 The Internet

The Internet and its penetration into the society led to the need for ubiquitous connectivity or "Internet everywhere" when consumers are en route. Wireless radio technologies allow end-users of mobile devices to stay connected to the Internet even if roaming around. Algorithms and protocols were developed to provide this kind of ubiquitous connectivity.

The Internet consists of interconnected subnetworks and therefore has a structured network topology. This structured topology of the Internet is useful as long as seamless mobility for end-users does not play an important role.

With the introduction of mobility protocols, like MobileIP [12, 13], macro mobility to the end-users of the Internet was provided. Macro mobility means that mobile devices can seamlessly connect to different subnetworks in the Internet and must not be reconfigured for every subnetwork they connect to.

Wireless radio technologies provide micro mobility to devices. Micro mobility means the possibility to be mobile but to always stay in the vicinity of a base station node or access point. Thus, the mobile device is always connected to one specific subnetwork of the Internet.

Ad-hoc networks follow a totally different approach. An ad-hoc network is formed by a number of wirelessly interconnected mobile network nodes. These mobile network nodes communicate directly to each other. Extended ad-hoc networks use multihop

forwarding to deliver data. Multihop forwarding is for accessing mobile devices beyond the radio range of a specific mobile device. With multihop forwarding other mobile devices forward data until the data is received by the destination device. Thus, each device in a mobile ad-hoc network using multihop features acts as a router for all other nodes. Routing protocols were developed to achieve this multihop routing in multihop wireless mobile ad-hoc networks with respect to the mobility of network nodes since network nodes are allowed to move randomly around.

Ad-hoc networks can be formed quickly on demand, e.g. in the case of a disaster. The drawback of pure ad-hoc networks is that they cannot be connected to the hierarchy of the Internet structure due to their different routing approach. Thus, interface nodes between the Internet and ad-hoc networks were introduced. These interface nodes are called *Internet gateways*. With the aid of Internet gateways the service range of Hot Spot Areas can be multiplied as ad-hoc nodes forward data to and from the access point to the wireless mobile ad-hoc nodes.

The goal is to provide ubiquitous Internet connectivity for members of ad-hoc networks. To achieve this Internet gateways firstly must be discovered by mobile ad-hoc network nodes using modified ad-hoc routing protocols. The discovery of internet gateways with modified ad-hoc routing protocols is one main task of this thesis. The other task of this thesis is to improve gateway discovery algorithms in order to increase the bandwidth a system provides to mobile ad-hoc nodes.

1.2 Task and Contribution of the Thesis

The first goal of this thesis is to develop and evaluate a new Internet gateway discovery algorithm. In the literature two main approaches for discovering Internet gateways are discussed. In the first approach the Internet gateway announces its presence to the mobile ad-hoc network nodes by periodically generating advertisement messages. The advertisement messages are then broadcast into the ad-hoc network and forwarded by every receiving ad-hoc network node. As a result the ad-hoc network is flooded periodically with gateway advertisements which causes much protocol overhead. This without demand approach is called a *proactive* approach.

In opposition, ad-hoc mobile nodes can request for Internet gateway services by

soliciting Internet gateways. Here every Internet gateway seeking ad-hoc network node floods the ad-hoc network with solicitation messages and available Internet gateways answer to such a request by sending a reply message. This on-demand approach is called a *reactive* approach.

The first goal of this thesis is to present a new approach for discovering Internet gateway nodes whereas the goal of the development of this new approach is firstly to reduce signalling overhead in the wireless environment caused by periodic ad-hoc network floodings and secondly to increase the provided responsiveness to the end-users' device by announcing the presence of Internet gateways proactively to mobile ad-hoc devices where proactively means that mobile ad-hoc devices are aware of Internet gateways before they need to use them.

The second goal of this thesis is to improve the performance of existing Internet gateway discovery algorithms including the newly developed algorithm. Two improvements are presented in the thesis. The first improvement is to optimise the multihop ad-hoc route from an Internet gateway to a mobile device of the ad-hoc network. The second improvement allows mobile ad-hoc nodes to select between multiple discovered Internet gateways not only after the standard metric of ad-hoc routing (the distance between ad-hoc nodes, measured in hops) but after the traffic an Internet gateway is charged with.

As a conclusion the tasks are:

- Presentation, investigation, and evaluation of the new algorithm for Internet gateway discovery and comparison with the established advertisement and solicitation based approaches
- Improving existing algorithms for Internet gateway discovery by
  - sending unrequested control messages to optimise routes between ad-hoc nodes and Internet gateways by sending newly introduced control messages and
  - select Internet gateways after new metrics based on background traffic and hop count to Internet gateway

In general the thesis provides three contributions to science. The first contribution is the presentation and introduction of a new possibility for discovering Internet gateways in mobile ad-hoc networks combining the pros of the well-known advertisement based and the solicitation based approaches. With the combined advantages of the well-known approaches it is possible to discover Internet gateways in mobile ad-hoc networks without consuming too much of the limited bandwidth resources of wireless network links which leads to more bandwidth for transferring data traffic and thus an increased throughput.

The second contribution of the thesis is the enhancement of existing Internet gateway discovery algorithms by either sending additional control messages between mobile ad-hoc nodes and the Internet gateway and by the selecting of an alternative Internet gateway if the already selected Internet gateway is burdened with network traffic.

The new algorithm for Internet gateway discovery as well as the two extensions for improving the performance of Internet gateway discovery algorithms are proven and evaluated by simulations. In order to simulate the new algorithm for Internet gateway discovery and the both improvements a complex simulation tool is needed. This simulation tool is an enhanced version of the network simulator NS-2 [4] and it is the third contribution of the thesis.

1.3 Structure of the Thesis

The thesis is structured as follows. In chapter 2 the Internet and its functionality in general is described and wireless technologies for data communication are presented. Chapter 2 further introduces ad-hoc networks and ad-hoc routing protocols. To discover Internet gateways adoptions to ad-hoc routing protocols for Internet gateway discovery are presented in chapter 3. In chapter 4 the new approach for discovering Internet gateways is introduced. This new approach is based on the HELLO messages of AODV [20] ad-hoc routing protocol. This thesis extends existing Internet gateway discovery algorithms as well as the new HELLO message based algorithm in order to improve their performance in terms of provided bandwidth. These extensions are explained in chapter 5. The new HELLO message based Internet gateway discovery algorithm's implementation as well as the performance improving extensions' implementations are

illustrated in chapter 6. In chapter 7 the Internet gateway discovery algorithms and the performance improving extensions are evaluated by simulations and are discussed. Finally, the thesis concludes in chapter 8.

Chapter 2

The Internet and Wireless Networks

2.1 Overview

This chapter introduces the general functionality of the Internet and the technologies that are responsible for modern Internet based communication. Since in the Internet data is divided into pieces that are sent separately, the Internet needs a functionality to ensure that every generated data piece is transported to the destination and delivered correctly. These data pieces are called packets and their delivery is mostly achieved using cables. Originally, mobility support is not implemented into the Internet functionality.

In the recent years wireless, i.e. radio, connections have established themselves. With wireless connections users may roam around while still being connected to the Internet. The drawback is that the radio range of wireless devices is limited. A number of wireless devices may form multihop networks. Multihop means that mobile devices are forwarding data packets for other mobile devices by creating a multihop path through the network. Such a network is called an ad-hoc network.

Next, the Internet is presented starting with a brief history of the Internet.

2.2 The Internet

2.2.1 History of the Internet

The precursor of the classical Internet, the ARPANET was developed in the U.S. in the 1960s. The anecdote tells that a distributed communication network was needed in the Cold War to resist any kind of attack. The network should have no centralised control instance that could fail and therefore disable the whole network.

Firstly in the U.S. and later in the whole world the ARPANET was used to connect universities and research establishments. The intention was to share limited research and computer capacities.

1982 the ARPANET adopted the TCP/IP protocol stack and was renamed the *Internet*. The TCP/IP protocol family is the backbone of the Internet today. One of the first applications for the Internet is the electronic mail, or e-mail. By sending an e-mail via the Internet an information can be delivered to the receiver within seconds, even around the world.

Another application is the file transfer. With the file transfer users can exchange electronic data files within seconds to a receiver everywhere in the world provided the sender and the receiver are connected to the Internet.

The popularity of the Internet was mostly driven by the invention of the World Wide Web (WWW) [73] which then was called the killer application for the Internet. The popularity of the Internet increased when the first free web browsers became available.

These days (30 June 2007) 17.8% of the world's population has Internet access [8] and one cannot figure out economic, cultural, research and other purposes without the Internet.

2.2.2 General Functionality of the Internet

In the Internet every data is divided into pieces that are called packets and the packets are delivered separately to the destination and thus the Internet uses a so called packet oriented approach.

The Internet, as a global network, consists of subnetworks. These subnetworks are interconnected by routers. A router forwards packets from one subnetwork to the

next until the subnetwork of a destination node is reached. Once the subnetwork of the destination node is reached the data packets are delivered to the destination node. Routing protocols care for the forwarding of data packets. The routing protocols are implemented in the routers. Since subnetworks are interconnected by routers a structured hierarchy results.

To connect nodes in a subnetwork of a small size (home or office use), one of the most used technologies is the Ethernet standard [37]. This standard is a wired standard which is enhanced from data transfer rates of 10 Mbit/s up to 1 Gbit/s today. Note, these are gross values and an application will only be provided about 70% of the gross rate.

Network nodes form subnetworks using network technologies (for example Ethernet). Subnetworks are switched together by routers to form bigger subnetworks. The sum of all subnetworks, routers, and nodes together form the Internet. Due to the partition into subnetworks the Internet has a hierarchical structure where some node (routers) interconnect subnetworks.

2.2.3 The Protocol Stack

The Internet Protocol (IP) [11] and the Transmission Control Protocol (TCP) [62] were combined to the TCP/IP protocol stack. Additionally, the User Datagram Protocol (UDP) [61] and IP together form the UDP/IP stack. In the Open System Interconnection (OSI) layer system [71] IP is located at layer 3 while TCP and UDP are located at layer 4. At layer 4 data connections are managed and controlled while layer 3 is responsible for the delivery of every data packet to the destination node via different subnetworks.

Figure 2.1 illustrates the OSI reference model. In the Internet case, layer 1 and layer 2 form the physical basis for transporting data (the Ethernet standard is located there) from one network node to its neighbour node. Data is generated by the layers 5 to 7 which represent the application the user is running. Note, data at layer 1 and 2 are called frames. Data at layer 3 are called packets and data at layer 4 and higher are called segments. The co-operation of the different layers of the OSI reference model is defined by the so called stack or *protocol stack*.

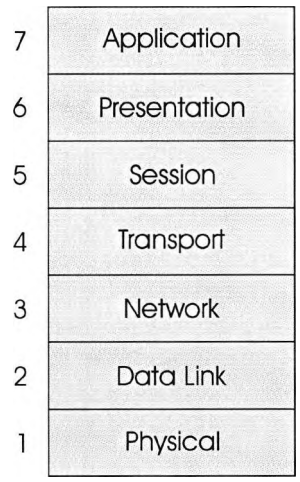


Figure 2.1: The ISO OSI reference model

As an example, the ISO OSI stack is used in this way. Assumed that an user initiates a file transfer from a source host to a destination host. The application the user is running sends the file from the higher network layers 5 to 7 to layer 4. At layer 4, the file is divided into segments where every piece is labelled with a unique number, the sequence number. This is achieved by TCP. The pieces are now called segments and are given to layer 3, which is responsible for delivering every segment to the destination node by forwarding the segments to the destination node. This is achieved by the routers. Then the layers 2 and 1 sent the data, embedded in frames, from one node to the next. At the destination node this process is reversed to put all pieces together to recompile the transferred file. TCP uses the sequence numbers to ensure that all pieces are correctly delivered and ordered correctly. As a conclusion, data at the source node is given from the top layer of the OSI stack to the bottom layer. At every layer a header of the appropriate protocol is prefixed to the data. When a frame is received at the destination node, the data is given from the bottom layer to the top layer while at every layer, the embedded data is unwrapped and the appropriate header is deleted.

This thesis concentrates on the network layer. Thus in the following the network layer and its neighbour layers, the transport layer and the data link layer, are explained in more detail.

Layer 4 - Transport Layer

Typical protocols for layer 4 are the Transmission Control Protocol (TCP) and User Datagram Protocol (UDP).

Transmission Control Protocol - TCP [62] was firstly standardized in 1981. It is a connection oriented protocol in opposition to the user datagram protocol (UDP [61]) which is not connection oriented. Connection oriented means that the source and the destination node of a data connection establish a logical connection between each other.

TCP initiates a concrete connection between two end systems by sending control messages. The main principle consists of so called SYN (connection request) messages that indicate that one end system likes to initiate a connection to another end system. The destination end system answers to a SYN message by acknowledging it with a SYN/ACK (connection granted) message. This means that first the SYN is acknowledged and second that the destination end system tests if messages destined to the initiating system can be received by the initiating system.

After receiving a SYN/ACK message the initiating system reacts by sending an ACK message to the destination system. After the destination receives that ACK message the connection between the two end systems is successfully established and that the end systems are able to receive messages and data segments from each other. The way the two end systems establish the point to point connection is called a three way handshake. The two end-systems close the connection by sending FIN messages to each other that indicate the closing of a connection session.

TCP specifies to acknowledge the correct receiving of segments by sending an acknowledge messages (ACK) to the sender system. If the sender recognises that one or more segments are not acknowledged, caused by either not correct receiving at the destination system or not correct transfer of the ACK message to the sender, the sender retransmits that segments.

The other main attribute of TCP is *flow control.* While transferring data from a sender to a destination node the segments may be routed via different networks with different connection speeds. As a result the sender needs to know how fast it can send segments towards the destination node. TCP ensures that the sender sends segments as fast as possible with a rate that the network is still able to transfer. Additionally,

TCP has a look after the congestion of a network that may cause delays or even packet loss.

To provide the most feasible sending rate TCP starts with a slow rate of sending segments and increases the rate of sending segments to a rate at which the transport to the destination system is disturbed. This disturbance is recognised by not receiving acknowledge messages (ACK) from the destination for specific segments. As the result of such a disturbance TCP reduces the sending rate.

To improve the performance of a TCP connection the protocol allows to send more than one segment simultaneously. In other words, the sender transmits two or more segments without having received an acknowledgement for segments already sent. In this way several segments are being routed within the network between the sender and the receiver simultaneously. The mechanism of sending more than one segment simultaneously is called the *sliding window* and the size of that window is adjusted according to the flow control and congestion of the transporting network.

There exist different types of TCP that differ in the way TCP reduces the sending rate in the case of a disturbance and increases the sending rate to a maximum but the main principle of all TCP flavours is the same.

User Datagram Protocol UDP [61] is not as complex as TCP. UDP was designed to transport data connectionlessly i.e. data is sent without firstly establish a transport layer connection. Typical applications that use UDP are every type of voice or video stream and in general every application where the correct delivery of data is not important but the amount of delivered data per time should be as constant as possible. Therefore, UDP needs not to be reliable and does not acknowledge received segments like TCP.

The UDP header just consists of four fields: The source and destination port the communicating applications are bound to, the length of the data and a checksum so that bad data can be discarded.

Layer 3 - Network Layer

In the Internet the protocol located at layer 3 is the Internet protocol (IP). Since layer 3 is responsible for delivering data to a destination node via multihop intermediate nodes

it is also called the routing layer. This thesis concentrates on the delivering of data packets. Thus the IP functionality and its addressing scheme is discussed in section 2.3 in detail.

Layer 2 - MAC Layer

A well known layer 2 protocol is defined in the IEEE 802.3 standard (Ethernet) [37]. IEEE 802.3 not only defines the frames at layer 2 but also defines physical parameters at layer 1. For layer 2 exist wireless protocols that are discussed below in section 2.4.

2.3 The Internet Protocol

2.3.1 Internet Protocol Addresses

In the Internet Protocol (IP), located at layer 3, every network node has its own unique network address. This section discusses version 4 of the Internet Protocol (IPv4) [11]. After that the new version of IP (IPv6) is discussed.

Internet Protocol - Version 4

An Internet network address consists of a number of digits and is 32 bits long. The 32 bits are divided into 4 Bytes which are notated in decimal format. The four decimal numbers are separated by dots in order to increase readability. The notification of a typical Internet address is 127.0.0.1 which for example is allocated to the local loopback device of a UNIX system. In classfull IP addressing, there exist three main classes of network addresses. The first one is called a class "A" network and its address range is the largest of the three classes. The two other classes are called class "B" and class "C" networks. In Table 2.1 an overview of Internet address classes is given. In the field "Address range" the first and the last possible address of a class is shown. "Netpart" and "Hostpart" indicate the number of Bytes describing the subnetwork and the host in that subnetwork. The "Network mask" gives an idea of the size of a network by indicating the number of Bytes of the IP address that stand for the netpart and the number of Bytes that stand for the hostpart. The netpart is indicated with a 255_{10} while the hostpart is indicated with a 0. The next field is about the Classless Inter-

Network	Address range	Netpart	Hostpart	Network mask	CIDR
Class A	0.0.0.0 - 127.255.255.255	1 Byte	3 Byte	255.0.0.0	/8
Class B	128.0.0.0 - 191.255.255.255	2 Byte	2 Byte	255.255.0.0	/16
Class C	192.0.0.0 - 223.255.255.255	3 Byte	1 Byte	255.255.255.0	/24

Table 2.1: IP network classes

First address	Last address	Network mask	CIDR notification
10.0.0.0	10.255.255.255	10.0.0.0	/8
172.16.0.0	172.31.255.255	172.16.0.0	/12
192.168.0.0	192.168.255.255	192.168.0.0	/16

Table 2.2: Private IP addresses

Domain Routing (CIDR) [74] notification of the Internet address. CIDR was introduced as the growing number of Internet nodes approached the maximum number of free (unallocated) Internet addresses. With CIDR smaller subnetworks can be formed as the classes listed in Table 2.1. For example a class A subnetwork is completely assigned when only one address of this class A network is allocated to an Internet subscriber thus the rest of theoretical free addresses of this subnetwork is wasted.

The number behind the "/" in the CIDR notification indicates the number of bits that describe the netpart of an Internet address and thus, intermediate subnetwork sizes can be formed. With CIDR, valuable Internet addresses can be saved.

The remaining address space in the Internet is even more efficiently used by granting Internet addresses to end-users dynamically on demand. This means that an end-user gets a different address from the Internet provider every time he connects to the Internet and thus a number of subscribers share a pool of addresses. This is achieved by the Dynamic Host Configuration Protocol (DHCP) [9].

Some address ranges are assigned for private networks [41]. Data packets originated by or destined to such a private address are not forwarded by public Internet routers and are being discarded. Three different sizes for private address ranges are given in Table 2.2. Internet addresses that are not within the address ranges of private networks are called public addresses.

Another technique for saving valuable IP addresses is the Network Address Translator (NAT) [42]. The idea is to hide a private network behind one single public IP address. A special NAT router forwards packets from the private subnetwork to the public Internet and vice versa. The advantage is that private IP addresses can be allo-

cated in private subnetworks multiple times while every complete private subnetwork consumes just one public address. The drawback is that applications in a private network behind a NAT router cannot be reached by network hosts in the Internet since the NAT router cannot decide to which host in the private network the data packets need to be forwarded.

Internet Protocol - Version 6

As a consequence of the permanent growth of the Internet the remaining address space is being rapidly exhausted and the need for new addressing schemes became necessary even after the introduction of CIDR, NAT, and dynamic IP address allocation. The Internet Protocol version 6 (IPv6) [10] reserves 16 bytes per address instead of 4 bytes for the old version 4 (IPv4) and thus, gives users the chance to obtain fixed and public IP addresses for every of their network devices (this results in $5.1 \cdot 10^{28}$ adresses per human being).

A 128 bits long IPv6 address is composed of a prefix and a suffix. The suffix is 64 bits long and an expanded version of the networks interface's fixed MAC address. A MAC address of a network interface is predefined by the manufacturer of the interface and is unique for every network interface. A padding of 16 bits is inserted into the interface's 48 bits MAC address and the resulting suffix is called an EUI-64 identifier. Thus the suffix of an IPv6 address is fixed.

In IPv6 addresses are notated in hexadecimal digits. Every four digits are separated by a colon ":". A double colon "::" indicates that all successive digits are zero. Like in the CIDR notification a slash "/" indicates how many digits of the address stand for the prefix and how many stand for the suffix of the address.

The prefix of the IPv6 address depends on the logical location within the Internet. When a network node is switched on it initialises all its network interfaces. In IPv4 the operating system assigns the network interface an allocated IPv4 address or forces the interface to query for an address by broadcasting a DHCP request. In IPv6 a node uses a standard prefix (fe80::/64) and its own MAC address (expanded to EUI-64) as an initial 128 bits *link-local* address. This link-local address is not routable in the Internet (like the private addresses in IPv4) but it is used for further discovery of the

local network structure in which the node is located at. This discovery process and self-configuration is described next.

In IPv6 networks, routers advertise their presence by sending advertisements into the network periodically. A new node attached to the network receives these advertisements and composes a *global-address* from the prefix of the advertisement sent by the router and their own fixed suffix. With this routable global-address a node gains connectivity to the rest of the Internet by using this new global address and the address of the router as a default route.

As a perspective, in ad-hoc networks there is no router that broadcasts advertisements for address autoconfiguration. Network nodes use their built-in suffix and a reserved prefix, e.g. the `MANET_PREFIX` in [51] which is defined as fec0:0:0:ffff::/64, to compose a unique ad-hoc routing address. This ad-hoc address allows the operation of an ad-hoc cluster when it needs no further connectivity to the Internet since all members of the cluster use addresses of the same subnetwork, i.e. the reserved prefix. Therefore the autoconfiguration features of IPv6 are predestined to be used in ad-hoc networks.

2.3.2 Routing in the Internet

If the destination of a data packet is located within the local subnetwork (identified by comparing source and destination addresses as well as the network mask of the CIDR notification) it will be delivered to the destination node directly. If the destination address is not part of the local subnetwork the data packet will be send to a preconfigured default route, i.e. to the router that provides connectivity to the higher hierarchy. Routers in the higher levels of the hierarchy take care for the correct delivering of the data packet. They have predefined routes to connected subnetworks and exchange routing information with other routers using routing protocols like Routing Information Protocol (RIP) [48].

The traversal of data packets in the Internet is explained by an example. An application generates data packets at layers 5 to 7 and gives them to the transport layer (layer 4). The transport layer creates a control header prefixed to the data and passes the packet down to the network layer. The network layer adds an additional

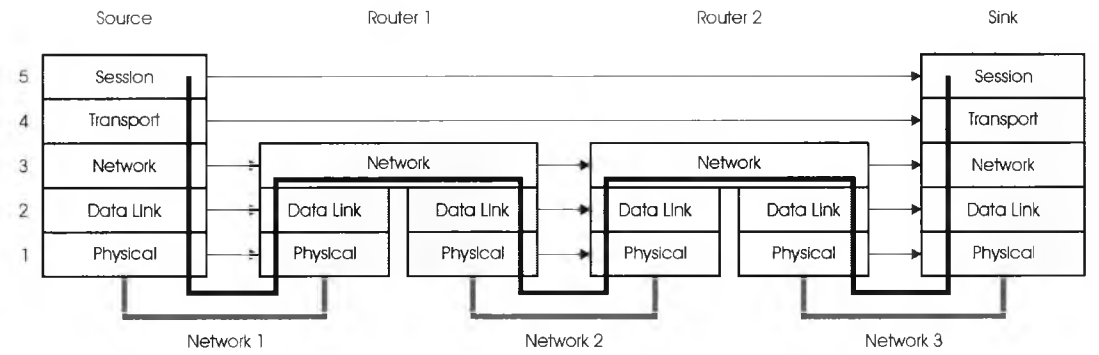


Figure 2.2: A data connection between two nodes via two routers

header for routing the packet to a destination node. Layer 1 and 2 are for delivering the packets to the next hop in the route toward the destination. For the network layer it is equal on which basis its data packets are transported. This may be a wired Ethernet [37] or a wireless W-LAN [39] solution. Other layer 2 solutions are given in section 2.4.2 on page 21.

As a conclusion, data packets travel down the OSI stack to the lowest layer, the physical layer (PHY) and at the destination node they are passed through every stack back to the application layer. When the data packet has to be routed via intermediate routers (to other subnetworks) they are only passed to the network layer because the routing algorithms in the routers do not need to know the contents of the packets but only the destination for forwarding purposes. In Figure 2.2 (enhanced from [71]) the path of a data packet through different network layers at different nodes is depicted (black line). The arrows indicate the connections of every layer to other layers of the same level at remote nodes. The physical links between networks are grey.

Global Internet Routing

The highest level of the Internet hierarchy consists of a number of so called Autonomous Systems (AS). Each AS is a distinct routing domain. ASs are usually operated by major network providers or universities. Within an AS, routers communicate with each other using intra-domain routing protocols also known as Interior Gateway Protocols (IGPs). ASs are connected via gateways routers. These gateway routers exchange routing information using Exterior Gateway Protocols (EGPs). In Figure 2.3 an example AS structure is depicted. Three ASs (AS123, AS456, and AS789) are connected via

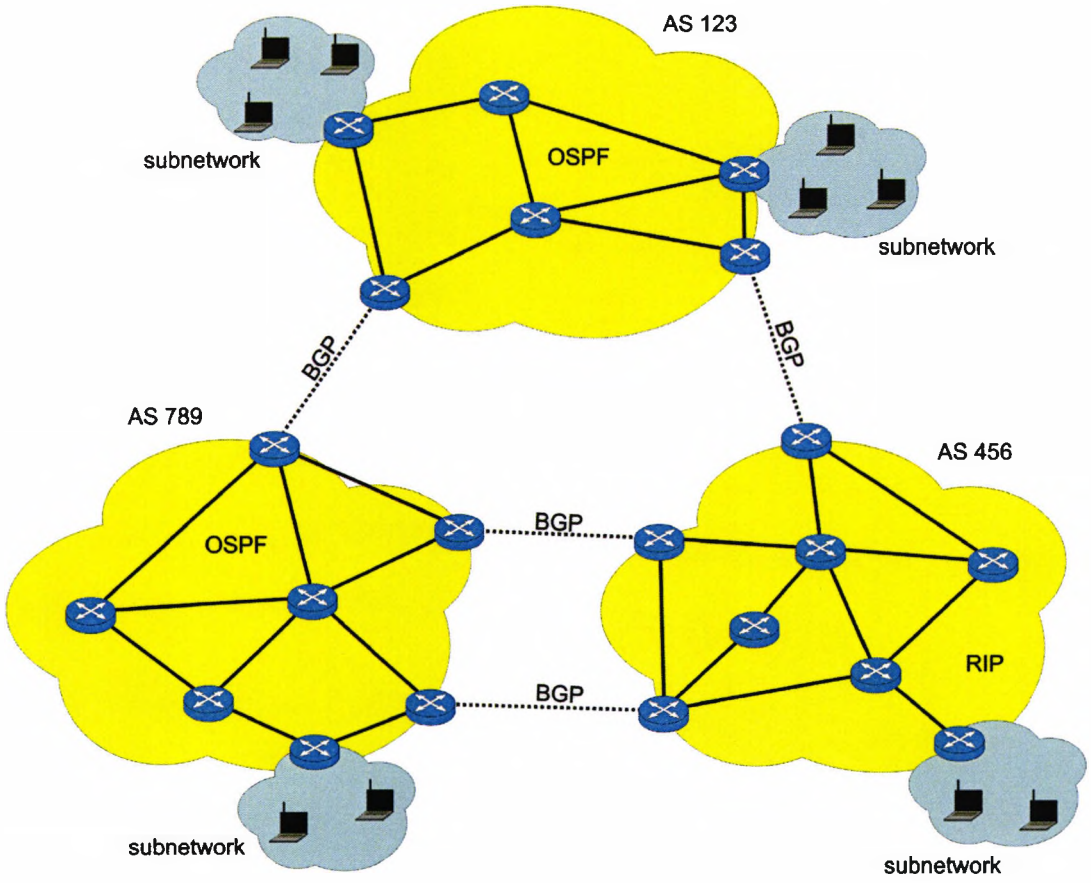


Figure 2.3: The Internet consists of autonomous systems

gateways routers. Each AS consists of a number of routers. Some of the routers of an AS may provide connectivity to subnetworks.

An older IGP is the Routing Information Protocol (RIP) [48]. RIP enables routers within an AS to exchange locally obtained information so that all routers within an AS have a coherent and up to date picture of how to reach any host within the AS. The principle functionality of RIP is that routers proactively advertise their routing tables to neighbour routers periodically. The hop count to a destination node is used as a metric for routing decisions. If a specific connection between two routers breaks the routers will use an alternative route for packet delivery.

Another IGP is called Open Shortest Path First (OSPF) [49]. OSPF is a member of the "link state" family and commonly used nowadays. Instead of exchanging hop distances to destinations, routers maintain a "map" of the whole network that will be updated quickly if a change in the network topology is detected. These maps (the link state database) is used to compute more suitable routes than RIP because OSPF uses

more metrics like bandwidth, hop count, and reliability of a link. In OSPF a router is aware of all links between all routers of an AS.

On the other hand, one AS shares routing information with other ASs using the Border Gateway Protocol (BGP) [50]. BGP provides connectivity between all ASs and therefore, BGP is essential for the Internet functionality. BGP exchanges routing tables to other ASs on-demand, i.e. when a change in the network topology was detected. For example a change occurs when a new AS is added to the Internet. Then this new AS announces itself to its neighbour ASs. The neighbour ASs give their AS routing table to the new AS. As a result, every AS knows how to reach any other AS.

2.3.3 Mobility Support in the Internet

Mobility for the end-user may be a micro or macro mobility. Micro mobility means that wired links that connect network nodes are substituted by wireless links, i.e., radio links. In such a network that provides micro mobility users are enabled to walk around while always stay connected to one subnetwork. Therefore, their devices have always the same IP address (IPv4 or IPv6). A typical example for a network with micro mobility is W-LAN.

In opposition, network nodes that move between different subnetworks use macro mobility. Macro mobility for end-users in the classical Internet can be achieved via the established Dynamic Host Configuration Protocol (DHCP). With the aid of DHCP nodes can connect to any subnetwork in the Internet, if DHCP is configured for that network. The disadvantage of this approach is that these nodes always have a different address each time they connect to a different subnetwork and therefore other nodes in the Internet cannot reach the roaming node since they are not aware of the actual address of the roaming node.

The need for a mobile addressing scheme for macro mobility is obsolete and a protocol that provides dynamic global routable addresses is discussed next.

MobileIP allows ubiquity for users even if they change their routable global addresses when roaming between different logical subnetworks. There exist MobileIP versions for IPv4 [13] and IPv6 [12]. In both versions *home agents (HA)* are located in a node's home network while additionally in IPv4 *foreign agents (FA)* are used in foreign networks i.e.,

if a node roams to a foreign network it uses the FA for MobileIP functionality. The principle of MobileIP is illustrated next.

When a node is located within its own home network there is no need to use MobileIP. The mobile node (MN) gets routable addresses either from a DHCP server or by overhearing router advertisements. In terms of MobileIP this routable address is called the *home-address* of a mobile node and the MN performs standard routing algorithms using its home-address.

If a MN roams to another (foreign) subnetwork, MobileIP is used. In IPv4 when a node is moving to a foreign subnetwork it registers at the FA to get a so called *care-off (c/o)* address. After that, it informs its HA in its home network where the node is located now by sending a *binding update (BU)*. Then every data packet from any node in the Internet destined to the MN is intercepted by the HA and forwarded to the MN to its temporary care-off address. Thus, other Internet attendants can always use the home-address of the MN even if the MN is not located within its home network.

In the MobileIPv6 standard exists no FA. MNs get valid temporary care-off addresses by IPv6 auto configuration procedures, i.e. router advertisements. This care-off address is then used to inform the HA. Again, if a correspondent node (CN) sends data packets to the MN, while the MN is located in a foreign network, the HA intercepts the data packets and forwards them to the current care-off address of the MN.

Figure 2.4 depicts a MobileIPv4 scenario. At the beginning, MN (white dot) is located in its home network. After moving to a foreign network (MN') and registering at the FA, a binding update message is sent to the MNs HA. If the CN sends data packets to the MN' (i.e. MN's home-address), these data packets are intercepted by the HA and forwarded to the MN's care-off address in the foreign network (MN').

In the depicted scenario data packets travel an unnecessary long path through the Internet. The detour route from the CN via HA to the MN is called *Triangular Routing*. MobileIP allows the MN to send an information packet (gratuitous binding update) to the CN after the first data packet from the CN has arrived at the MN. This update contains the care-off address of the MN and thus, the CN knows a more reasonable route to the MN (dotted arrow).

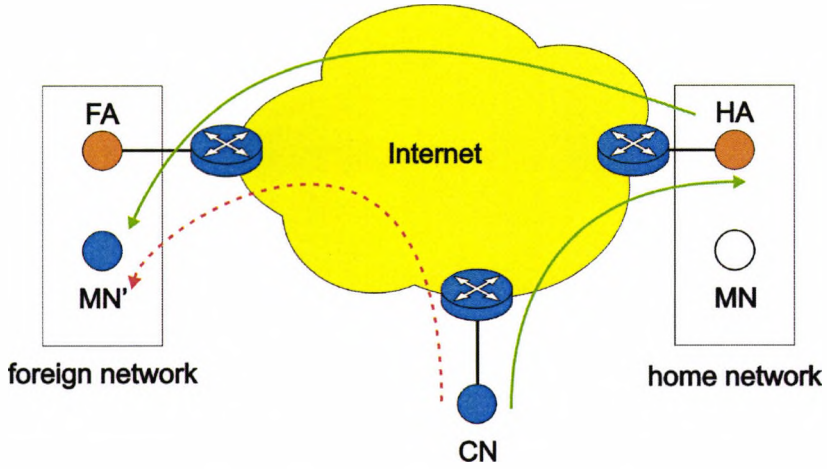


Figure 2.4: A MobileIPv4 scenario

2.4 Wireless Networks

2.4.1 The Need for Wireless Networks

This section introduces various wireless technologies of different kinds (supporting high mobility or high bandwidth, Figure 2.5). First the need for modern wireless data transmission is discussed. Next, the main features in terms of service range and bandwidth of a selection of wireless network technologies are presented. This thesis is based on Wireless Local Area Network (W-LAN) [39] technology. W-LAN is explained in more detail at the end of this section.

Radio based wireless networks enjoy great popularity by end-users due to their flexibility in connecting mobile nodes. With wireless connections, end-users may move around while always stay connected to a network if the wireless radio range provides coverage. The coverage of a wireless system depends on the radio range the system uses for connectivity and the actual geographic topology, since obstacles may either interrupt the radio connectivity or cause reflections which lead to interferences and affect the wireless system negatively.

Wireless networks provide more flexibility to the end-users in terms of expanding a network with more network nodes since no wires need to be laid to extend the network.

The mobility for users with mobile wireless devices and the simple extension with new network nodes are the main reasons for the success of wireless networks.

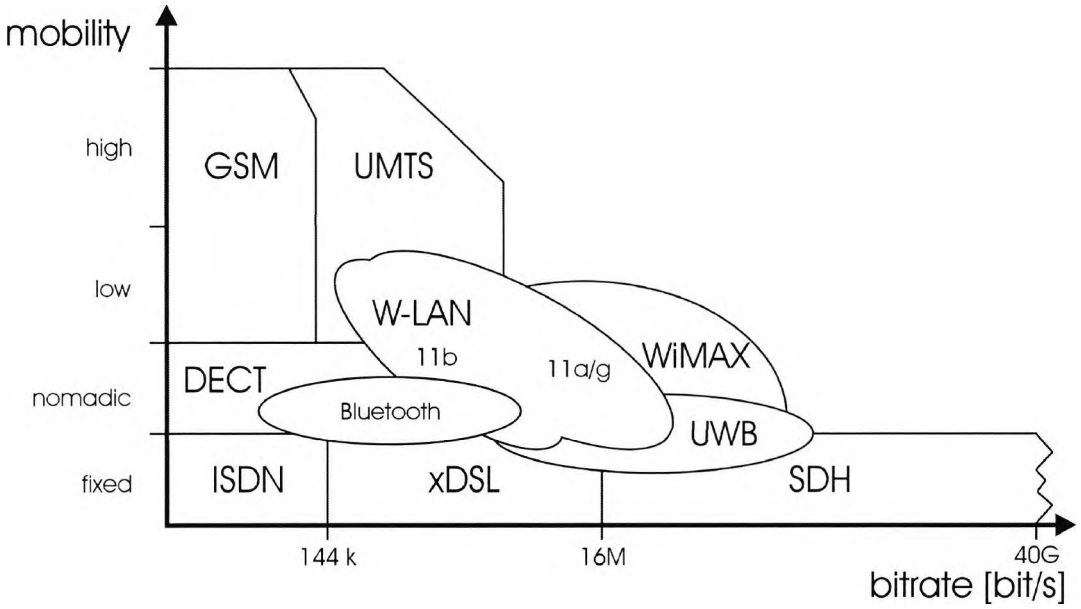


Figure 2.5: Wireless access technologies

2.4.2 Overview of Wireless Network Techniques

In accordance to their range networks are classified into different categories. These categories are called Personal Area Network (PAN), Local Area Network (LAN), Metropolitan Area Network (MAN), and Wide Area Network (WAN) while there are no sharp defined borders between the categories. For wireless networks. the radio ranges of the categories increase from about 1 meter (PAN) to several kilometres (WAN). In opposition, the bandwidth decreases from 2 Gbit/s (PAN) to 2 Mbit/s (WAN). These values are maximum values for modern wireless technologies.

In the following a number of wireless technologies is discussed. Figure 2.5 gives an overview of different wireless access technologies regarding the end-users' mobility and their provided bandwidth. In general, wireless technologies that provide high bandwidth provide less mobility and vice versa. This is since in wireless networks a handover from for example one radio cell to the next radio cell may be harmed or even physical effects may harm a wireless connection.

In Figure 2.5 wired network systems are integrated to give comparable values. Nodes of a nomadic used system are mostly static but sometimes they move to another location, e.g. a wireless computer keyboard. High mobility stands for systems that can be used to connect network nodes that move at high speeds like the speed of a train.

Typical representatives of WANs are the Global System for Mobile Communication (GSM) [76] and its successor, the Universal Mobile Telecommunication System (UMTS) [76]. GSM and UMTS are cellular based systems for mobile communication. Here, mobile end-user systems (mobile phones) connect directly to base stations in order to gain connectivity to a wired infrastructure. Mobile devices roam from one base station to the next when moving. Standard GSM provides bandwidths of 13 kbit/s while a UMTS cell scales up to 2 Mbit/s. The radio ranges of GSM and UMTS are from a few hundred meters up to several kilometres.

One of the most recent wireless technologies is the Worldwide Interoperability for Microwave Access (WiMAX) [44] standard. WiMAX is a wireless technology that provides broadband connections over long distances and it can be used for a number of applications, e.g. for "Last Mile" broadband connections. The "Last Mile" describes the connection from a switching centre of a telecommunication provider to the customers.

Furthermore, hotspots and even the backbone of cellular networks, like GSM/UMTS, are applications for this fast technology. However, the mobility support in WiMAX is not yet standardised and needs further development. WiMAX is a typical representative of MANs.

The short range radio technique Bluetooth [43] was invented to connect small gadgets wirelessly. Short range means from a few meters up to one hundred meters. A Bluetooth adapter may be implemented into mobile phones, computer mice, notebooks, headsets, and other peripheral equipment to dispose disturbing wires in a typical office environment. Bluetooth provides bandwidths of up to 1 Mbit/s.

For short range services a new technology has been developed which is called Ultra Wide Band (UWB) [40]. UWB scales from 100 Mbits/s up to 2 Gbit/s while its range is limited to a few meters. A typical application for UWB may be the connection between home entertainment devices. Bluetooth and UWB are representatives of PANs.

Between the size of PANs and MANs, LANs are settled. An established technology of this kind of networks is the Ethernet [37] standard (wired). Ethernet is often used to form a network in offices or at home. One wireless standard for LANs is Wireless LAN or W-LAN. The principle functionality of W-LAN is explained in more detail next.

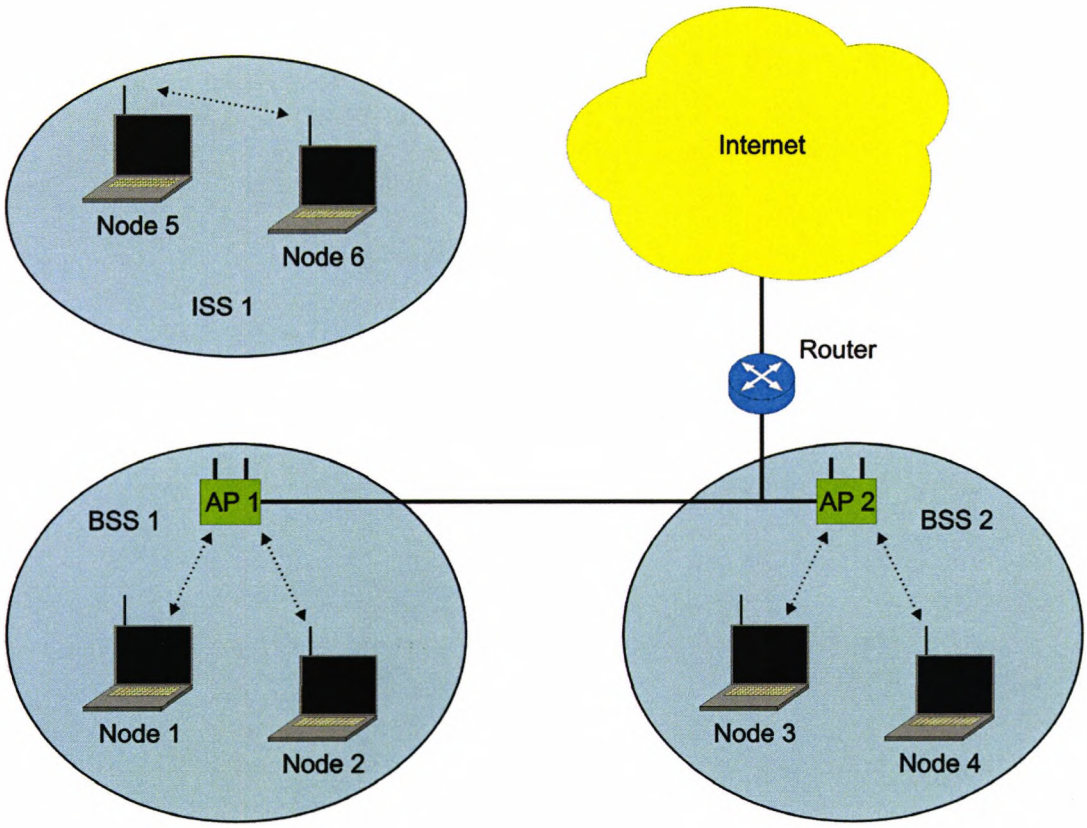


Figure 2.6: W-LAN architecture

2.4.3 Wireless LAN

For home and office use the Wireless LAN or W-LAN [39] became very popular in the recent years. It allows users to substitute LAN cables by wireless links. This leads to more flexible positioning of network devices and easy expansion of the network with new nodes. W-LAN is very popular for transportable notebook computers and Personal Digital Assistants (PDA).

There are two main operation modes for W-LAN. The first mode of operation is a structured mode where an Access Point (AP) is used. An AP allows nodes to connect to a higher level structured wired network, like the Internet, and thus the AP works as a base station. The second operation mode is the ad-hoc mode. Here, peers communicate with other peers directly.

In Figure 2.6 a number of network nodes with W-LAN interfaces are depicted. Node 1 and node 2 as well as nodes 3 and 4 form a Basic Service Set (BSS) both connected to a separate AP. Node 5 and node 6 form an Independent Service Set (ISS) using the

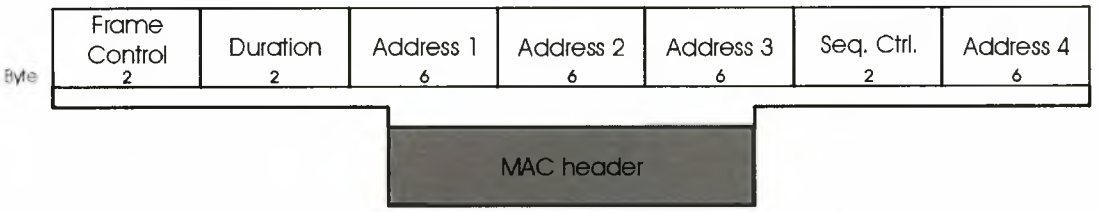


Figure 2.7: W-LAN MAC header

ad-hoc mode without an AP and therefore, without Internet connectivity.

W-LANs typically use a shared medium. Two nodes cannot send data (or other information) at the same time or otherwise collisions between packets (in layer 2 terminology: frames) will occur. The total bandwidth provided by the W-LAN technology is shared among the subscribers of the W-LAN and therefore divided by the number of subscribers. This is if all subscribers are within of the radio range to each other. If not, e.g. they are only within their interferance range, the provided bandwidth to subscribers will be more negatively influenced.

W-LAN MAC Header

The W-LAN technology controls the access to the radio medium by the layer 2 (MAC) protocol. The MAC protocol of W-LAN is explained in more detail next. According to the OSI reference model the MAC header and its payload is embedded into a layer 1 frame. In Figure 2.7 the header of a W-LAN frame is depicted as defined in [39]. The header is 34 Bytes long. The first two Bytes of the header define the *frame control.*

Frame Control The frame control field differs the following payload by data, management information, and control functions like the RTS/CTS/ACK mechanism. The RTS/CTS/ACK mechanism is explained in 2.4.3. Management functions include information about AP beacons or Probe Request for an AP or the Response to such a request (for an AP).

Duration The duration field of the MAC header contains the estimated time for transmitting the frame.

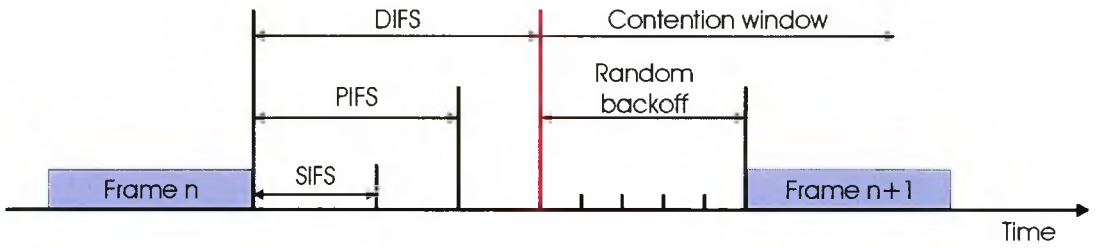


Figure 2.8: Medium access

Addresses 1 - 4 In these fields the addresses of transmitting and receiving nodes are declared. The source and destination addresses as well as the BSSID (Basic Service Set ID) of the network are included. The fourth address field remains empty. It is only used when traffic is to be forwarded from one BSS to a second BSS.

To reduce the number of simultaneous accesses to the medium a random back-off mechanism is integrated to the W-LAN standard. In Figure 2.8 this random back-off mechanism is depicted in more detail. Every node listens to the carrier signal if it is occupied or not (Carrier Sense), i.e. if the channel is free for transmitting. After a transmitted frame the node waits a period of time, the Inter Frame Spacing (IFS). There are three different IFS defined. They are called Short IFS (SIFS), Point Coordination Function IFS (PIFS) or Distributed Coordination Function IFS (DIFS). SIFS is for messages of the highest priority like ACK messages, while PIFS is for time critical services. DIFS is reserved for asynchronous services like simple data transfer. After this minimum waiting time the Contention Window (CW) begins. While the CW period, nodes wait a random time before they start to transmit their next frame. Thus, the node with the lowest random back-off time wins the contention and begins to transmit a frame.

The Hidden Terminal Problem

Assume that, as depicted in Figure 2.9, there are four wireless nodes (A, B, C, D) in a row with radio ranges symbolised by the ellipsoids. Node A wants to send a frame to node B (data). Therefore, node A will listen on its radio interface if the channel is free for transmitting. Due to the distance between the nodes node A cannot hear node C and node D. Simultaneously, node C wants to send a frame to node D. This results in a collision at node B since node B is charged with two frames simultaneously.

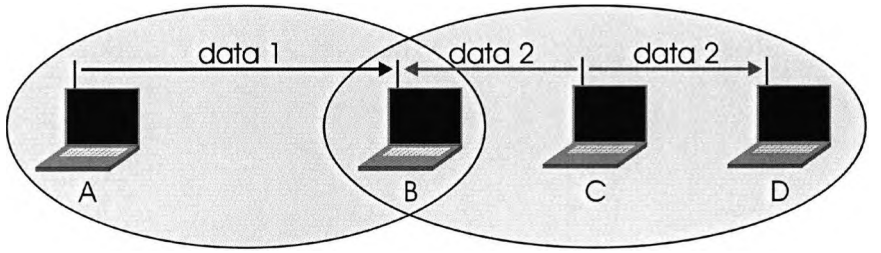


Figure 2.9: The hidden terminal problem

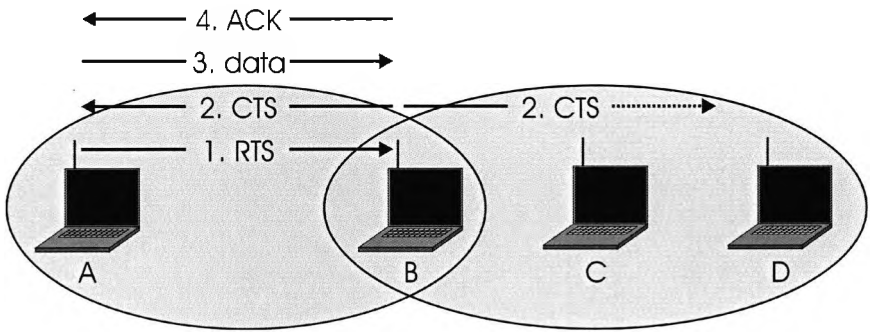


Figure 2.10: The RTS/CTS solution for the hidden terminal problem

This problem of colliding frames at an intermediate node is called the *hidden terminal* problem. The hidden terminal problem is typical for wireless systems.

There exists a solution to the hidden terminal problem, the RTS/CTS mechanism. The RTS/CTS mechanism works as follows. If node A wants to send a frame it firstly transmits a Request to Send (RTS) message. This is received by node B and answered by node B with a Clear to Send (CTS). All other surrounding nodes of node B receive this CTS, too and wait until the data transmission from A to B is finished. The end of a successful transmission is marked with an Acknowledgment message (ACK) from node B to node A. The RTS/CTS mechanism is depicted in Figure 2.10.

RTS, CTS, and ACK messages are very short messages for controlling a transmission. But it may happen that control messages may collide with data frames. In that case the transmission procedure fails and it will be re-initiated.

In this thesis W-LAN is used as a basis for transmitting data packets from one mobile node to another in a multihop ad-hoc network. The functionality of multihop ad-hoc networks is discussed next.

2.5 Ad Hoc Networks

Ad-hoc networks are spontaneous and flexible networks of mobile nodes. The mobile nodes may be a number of laptop computers or mobile phones or PDAs while the layer 2 basis of the network is in general irrelevant (W-LAN, Bluetooth, etc.). Wireless network interfaces are predestined for ad-hoc networking since wireless links allow node mobility and simplify the integration of new nodes.

In Figure 2.6 in the previous section, the ad-hoc mode on a W-LAN basis is introduced. Nodes of an ad-hoc network communicate to their peers directly and they do not have an access to the Internet since in ad-hoc mode no access points (AP) are used. In general, nodes behind the radio range of a wireless interface are not accessible. The principle of ad-hoc networks can be extended to a *multihop* ad-hoc network where network nodes forward packets for other nodes in order to access nodes that are more distant than the radio range. A hop is defined as one forwarding step in a route through an ad-hoc network. This section addresses multihop ad-hoc networks.

Opposite to structured networks with their predefined routers and subnetworks, multihop ad-hoc networks have no subnetworks and therefore, no explicit routers. In fact, every member of the ad-hoc network acts as a router for all other members. Data packets are forwarded from one node to the next until they are received by a destination node. Thus, every network node is self-responsible for discovering a valid route to a destination node. To achieve this route discovery ad-hoc network nodes use ad-hoc routing protocols. The total of all mobile network nodes that are in direct or indirect (i.e. via intermediate nodes) range are called an ad-hoc cluster or Mobile Ad-hoc NETwork (MANET). In this thesis the terms "ad-hoc network", "ad-hoc cluster", and "MANET" are used equivalently.

Next follows a list of the main characteristics of multihop mobile ad-hoc networks.

- easy to extend with new nodes
- mobility for nodes
- no centralised configuration instance needed
- may extend radio range of single nodes by multihop features
- need minimum node density for functionality
- routes may break unexpected due to mobility and must be rediscovered

- no possibility for a long-term network resource reservation
- limited bandwidth resources due to wireless communication
- easy to set up in infrastructureless environments (disaster)

If no connectivity to other networks like the Internet is needed wireless multihop ad-hoc routing features are predestined since they do not need any infrastructure like pre-given routers, APs, or further pre-configuration. The challenge of ad-hoc networks is that mobile users do move and therefore, routes may break during a data transmission and must be rebuild again. This is the task of multihop ad-hoc routing protocols.

In ad-hoc networks, when a node is switched on, it is only aware of its own suffix (EUI-64). With the defined prefix for ad-hoc networks the node composes an ad-hoc routable address which is defined as its link-local address. If no further functionality like Internet connectivity is needed this link-local address satisfies for routing in ad-hoc networks. The premise is that a traffic generating source node needs to know the EUI-64 identifier of the target node to compose an ad-hoc routable address (fe80::/64).

2.5.1 Functionality of Ad-Hoc Networks

An example ad-hoc network is depicted in Figure 2.11. Here, a number of nodes (small circles) form an ad-hoc network together with the depicted mobile node (MN) and the correspondent node (CN).

The MN establishes a connection to a destination node, the CN. Data packets of this connection are forwarded by intermediate nodes (IN1 and IN2). The dotted black lines between nodes show one feasible multihop route through the ad-hoc network. Thus, the two intermediate nodes IN1 and IN2 act as relay stations for the connection between the MN and the CN.

The connectivity of random moving ad-hoc networks depends directly on the density of nodes within the ad-hoc cluster and the node movement of specific nodes. Another parameter is the radio range of nodes' wireless network interfaces. Depending on their radio range the density of MANET nodes has major impact on ad-hoc networks. If fewer mobile nodes attend the ad-hoc network the ad-hoc network may be separated in two sets which are out of radio range of each other. Thus, only nodes of one set may

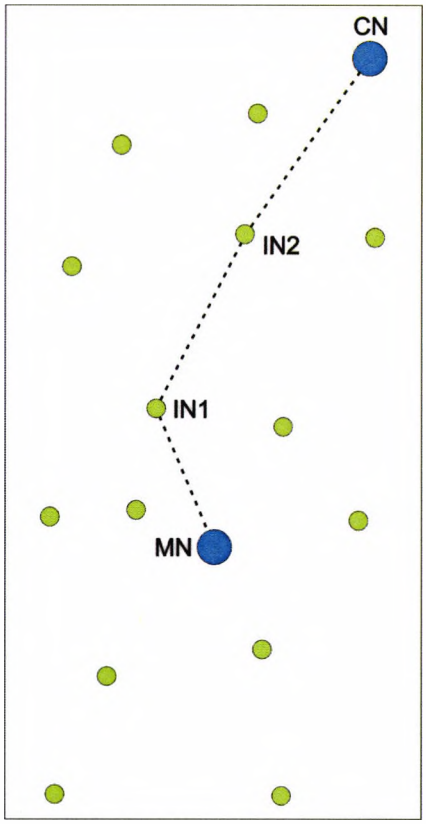


Figure 2.11: A simple ad-hoc network

connect to each other while an inter-set connection is impossible. The density of nodes of a multihop ad-hoc network is discussed in section 7.4.1.

Mathematical analysis on ad-hoc connectivity within a mobile cluster have been published in [14]. The authors derive formulas for the availability of a single link as a function of node speed and distance between nodes. Further they give formulas for the duration of multihop links (paths) in a MANET depending on the number of intermediate nodes.

Mobile nodes use ad-hoc routing protocols to find a valid multihop route through the ad-hoc network, or a single hop route, if the actual network topology is sufficient. The discovery of routes and the maintenance of routes are explained next.

2.5.2 Routing in Ad Hoc Networks

In general, there exist two main approaches for routing in MANETs. One is called the proactive approach and the other one is called the reactive approach. In Table 2.3 an overview of different ad-hoc routing protocols is given. In the proactive approach, mem-

proactive	reactive
DSDV	DSR
OLSR	AODV

Table 2.3: Ad-hoc routing protocols

bers of the MANET send routing messages to their neighbours periodically and thus, all members of the MANET permanently have valid routes to potential destination nodes. Thus, in the proactive routing approaches a specific node gets routing information to a specific destination node without request. In opposition, in reactive protocols nodes request routing information on demand and therefore, create less routing overhead since less periodic routing messages are generated.

An overview of ad-hoc routing protocols is presented in [32]. There, the general concept of route discovery and route maintenance is discussed. The routing protocols AODV [20], DSR [16], DSDV [15], and ZRP [19] are introduced and the use of ad-hoc networks is discussed. An overview of the principle functionalities of AODV, DSDV, OLSR, and DSR are given in this section.

Ad-hoc routing algorithms can be classified in *distance vector* and *link-state* algorithms. In link-state algorithms, every network node generates its own view of the whole MANET and therefore has a global view on the network. Routes are selected by adequate algorithms like the Dijkstra algorithm [72] but they are not suitable for large networks with dynamic topologies, i.e. many fast random moving nodes since link-state algorithms exchange routing information every time the network topology changes. The exchanging of routing information causes protocol overhead.

On the other hand, distance vector algorithms have a local view on the actual network topology only. MANET nodes exchange routing information with their neighbours only. The disadvantage is that routing loops may occur.

A selection of four exemplary representatives of ad-hoc routing protocols, two proactive and two reactive, are explained in more detail next.

Proactive Routing Protocols

Typical proactive ad-hoc routing protocols are the Destination Sequenced Distance Vector (DSDV) [15] and Optimized Link State Routing Protocol (OLSR) [18]. In

proactive routing protocols, the routes of data packets to their destination is known prior to the data packets' generation. Therefore, there is no latency caused by route discovery since the route to a destination node is already known. The disadvantage from this approach is that routing tables with many entries result in every network node (every node knows a valid route to all other nodes) and much routing overhead due to the periodic exchange of routing information without demand. Next, the DSDV and the OLSR protocols are explained.

Destination Sequenced Distance Vector (DSDV) is a table driven ad-hoc routing protocol that uses HELLO messages. HELLO messages are generated periodically by every network node to indicate the node's presence to its direct neighbour nodes. As a first result of this approach, every node is aware of its direct neighbour nodes (one-hop neighbour). Furthermore, every node includes its neighbours into its HELLO messages to give this information to more distant nodes. With every HELLO interval information about reachable nodes is spread deeper within the MANET. In Table 2.4 the routing tables of nodes CN, IN2, IN1, and MN are given to clarify the DSDV algorithm. The example is based on Figure 2.11. Every cell in the table has three entries. The first one is the destination for a route and the second is the next hop entry pointing to that destination. The third entry is the distance to the destination (metered in hops).

Every line of the table represents one step in the discovery process of the MANET until the MANET is totally established. It is assumed that all node are switched on simulatenously. In line 1 every network node is only aware of itself and has a routing table entry pointing to itself with a hop count of 0. In line 2, after this information was forwarded by every node with the first HELLO message, the neighbour node of each node creates a routing table entry pointing to the neighbour that sent the HELLO message. Then this new entry indicates a route to the neighbour node via the neighbour node as a next hop information and a hop count distance of 1. The intermediate nodes (IN1 and IN2) have three route entries because they have received HELLO messages from two neighbours (third entry is pointing to themselves).

Step 3 of the process goes further. Here, CN has received a HELLO message from IN2 containing information about IN2 and IN1 and thus, creates an additional entry pointing to IN1 with next hop entry IN2 and hop count of 2. In this step IN2 and IN1

	CN	IN2	IN1	MN
1.	CN CN 0	IN2 IN2 0	IN1 IN1 0	MN MN 0
2.	CN CN 0	IN2 IN2 0	IN1 IN1 0	MN MN 0
	IN2 IN2 1	CN CN 1	IN2 IN2 1	IN1 IN1 1
		IN1 IN1 1	MN MN 1	
3.	CN CN 0	IN2 IN2 0	IN1 IN1 0	MN MN 0
	IN2 IN2 1	CN CN 1	IN2 IN2 1	IN1 IN1 1
	IN1 IN2 2	IN1 IN1 1	MN MN 1	IN2 IN1 2
		MN IN1 2	CN IN2 2	
4.	CN CN 0	IN2 IN2 0	IN1 IN1 0	MN MN 0
	IN2 IN2 1	CN CN 1	IN2 IN2 1	IN1 IN1 1
	IN1 IN2 2	IN1 IN1 1	MN MN 1	IN2 IN1 2
	MN IN2 3	MN IN1 2	CN IN2 2	CN IN1 3

Table 2.4: DSDV example routing tables

has received information about all other nodes and therefore, have created entries to all other nodes.

The last step of this process leads to routing information from every node to every node. Here, every node has received the included information from all other nodes once, even the nodes at the border (CN and MN) of this example topology.

These routing table entries are being stored until a node does not receive HELLO messages from an enlisted neighbour node for a specified period of time. This may occur due to node movement or turned-off nodes and is declared as a significant change in the network topology. Now, let us assume that the CN is turned off, then the IN2 sets the route to the CN down and broadcasts this information immediately to its neighbour nodes (IN1). Again, the neighbour nodes set down all routes according to the CN and immediately forward this information. Note, information about a static network topology is distributed periodically within the network by HELLO messages but changes in the network's topology are spread on demand, i.e. immediately after they were detected.

If a data packet is to be delivered to a specific destination, nodes consult their routing tables. Since the proactive nature of DSDV every MANET node has a valid route to every probable destination node.

Optimized Link State Routing (OLSR) uses HELLO messages for direct neighbourhood discovery, too. Firstly, they are sent by every node to indicate its presence

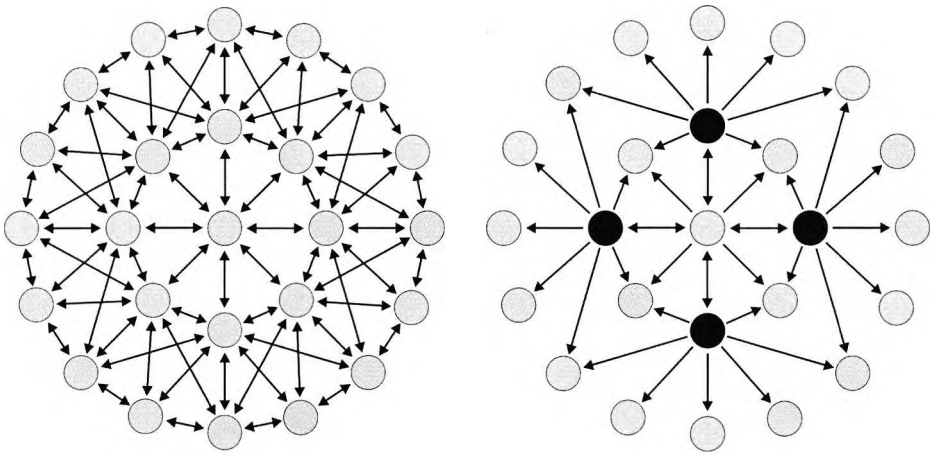


Figure 2.12: Flooding in OLSR: (a) Complete flooding; (b) Flooding with MRPs

to all surrounding neighbour nodes. Secondly, every node includes information about its neighbours into its HELLO messages. As a result, every member of the MANET knows all one-hop neighbour nodes and additionally the neighbour nodes of the neighbour nodes which are called *2-hop neighbours* in the OLSR terminology. Up to now, the algorithm works in a similar way as DSDV but the further procedure is very different.

In OLSR nodes flood information of their own view of the current network topology into the MANET. Flooding is a very simple approach for information distribution in MANETs and works as follows. Assumed that an information message is sent by a specific node, every node that receives this message retransmits the message. As a result, every network node has received the message at least once. The number of message retransmissions is: $n - 1$ where n is the number of attending MANET nodes. In OLSR this approach for information distribution is improved by the introduction of Multirelay Points (MRP) which are the only nodes that retransmit packets. Other nodes (non MRP) that have received packets do not retransmit them. In Figure 2.12 every arrow indicates one retransmission of a message. On the left the number of retransmissions is higher compared to the right with MRPs (black dots).

Next, the algorithm how the MRP nodes are selected from a number of equal MANET nodes is described in more detail. In opposition to DSDV, in OLSR nodes know only about their direct and their 2-hop neighbours and not about more distant nodes. A node which has information about a neighbour node and at least one 2-hop

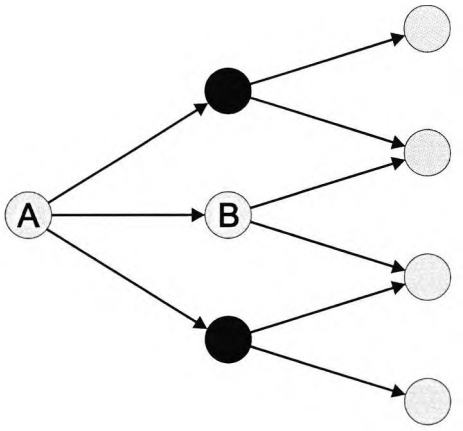


Figure 2.13: MPR selection in OLSR

neighbour node selects the direct neighbour node as its MRP. The goal of this approach is to select the least number of MRPs to cover all 2-hop neighbour nodes. This results in a minimised number of retransmissions of control messages. As an example, in Figure 2.13 node A chooses the black nodes as its MRPs because with this two MRPs all 2-hop neighbours of A are comprised.

Consequently, MRP nodes have information about the current network topology and broadcast this information by a Topology Control message (TC) to other nodes using the MPR topology. As a result, every MANET attendant has routes to every node at every time. In the example scenario 2.11 the MRP of the CN is IN2. The MRP of MN is IN1 (CN is a direct neighbour while MN is a 2-hop neighbour).

Reactive Routing Protocols

In opposition to proactive routing protocols, the reactive protocols work on a *on-demand* basis. Routes to destination nodes are not pre-cached and must be discovered first in order to deliver packets. The on-demand nature of this protocol family causes latencies in the packet delivery if a route to a destination node must firstly be discovered.

To discovery a route to a destination node, in reactive ad-hoc routing protocols (compare to Figure 2.11) the MN generates a route request (RREQ) message. In general, this RREQ message contains the originator's address and the address of the requested node (destination address). Every neighbour node of the MN receives the RREQ message and forwards it. As a result, the whole MANET cluster is flooded with

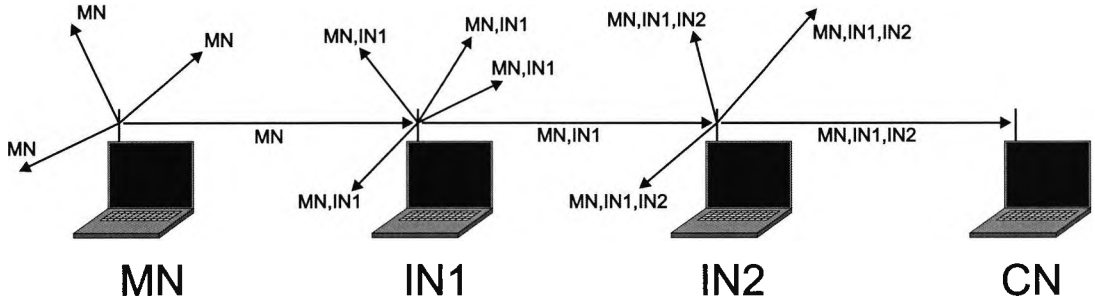


Figure 2.14: Route discovery in DSR

this single request. When a node in Figure 2.11 (here: the CN) receives the RREQ it answers by sending a route reply (RREP) message back to the MN. This RREP message follows the reverse RREQ path and is only forwarded by nodes along this reverse route request path. Therefore, RREQs flood the MANET, RREPs don't.

For every ad-hoc routing protocol there exist different header formats. In the following, different reactive ad-hoc routing protocols and their functionalities are discussed.

Dynamic Source Routing (DSR) [16] adds the addresses of every network interface into the RREQ header when nodes forward a RREQ. Thus, a chain of valid intermediate node addresses has been created when the RREQ is received by the destination node and the header of a DSR route request packet increases in size with every forwarding by an intermediate node. After receiving the RREQ, the destination node answers to the RREQ by sending back a RREP message along the reverse route. The reverse route is formed by address sequence inversing of the address chain in the RREQ header. Figure 2.14 depicts this in more detail. The MN searches a route to the CN and therefore it generates a RREQ message. If an intermediate node receives this RREQ it compares the destination address with its own address. If the addresses are different the intermediate node adds its own address to the DSR route request header and forwards the RREQ. When the addresses are equal a node knows that the RREQ is destined to itself. When the RREQ is received at the CN the resulting address sequence is: MN $\rightarrow$ IN1 $\rightarrow$ IN2. In this case the CN inverses the address sequence in order to get the reverse route from the CN back to the MN (CN$\rightarrow$ IN2 $\rightarrow$ IN1 $\rightarrow$ MN). Then the CN sends the RREP message along this reverse path.

As a result of the route request process, in DSR every node that is part of a specific route knows all members of that route and is therefore able to deliver own

data packets to every member of a known route without requesting for the destination node again. Additionally, a node with the knowledge about the reachability of other nodes may answer to another RREQ of a different node gratuitously without forwarding the RREQ.

If the RREP of a specific RREQ is not received for a specified time out the MN tries again to find a route. A RREQ time out occurs if the actual network topology is not sufficient or the CN is not available at all. RREQs have a sequence number in order to know which RREQ message is related to which RREP. The complete address list to the destination node is included into the header of every data packet to inform other nodes of valid routes by overhearing the data packet.

As mentioned before, routes may break during node movement. A network node overhears the forwarding of a data packet from its neighbour node. If a node recognises that the neighbour node does not forward the data packet it assumes that the neighbour node has disappeared and then the node generates a route error (RERR) message and drops the data packet. This RRER is then sent along the way back to the originator of the data packet. Then the originator may initiate a new route request procedure.

There are several features for DSR to improve the protocol. In the scenario above (Figure 2.11), it is assumed that the CN is moving toward the MN. Then the route from the MN to the CN (and back) may be shortened. This is done by IN1 when it receives packets (e.g. TCP-ACK packets) from the CN. Then IN1 forwards the next data packets to the MN directly and generates a RREP message to inform the CN that the route may be shortened to $CN \rightarrow IN1 \rightarrow MN$.

It is typical for DSR that network nodes know routes to other nodes even if these other nodes are not needed by an application. As an option to DSR, this fact may result in a RREP message as a reaction to a RREQ even if the answering node is not the destination of the RREQ. To ensure that a number of nodes is not answering simultaneously they wait for a random back-off time before they answer to a request gratuitously. This gratuitous answer is only sent when no other node (within radio range) has answered firstly. This feature prevents route reply storms which would lead to collisions.

Another optional feature to DSR is called package salvaging. In the case that a node

does not recognise the forwarding of a data packet to the next node it may salvage the data packet before sending a RERR. For salvaging, the node starts a route discovery for the destination node of the data packet and if this procedure is successful the data packet is delivered to the destination node via the repaired route. Additionally, a RREP message is sent back to the originator of the data packet to inform every route member of the new route.

Ad hoc On-Demand Distance Vector (AODV) is presented in [32] and the protocol specifications are listed in [20]. In AODV the route discovery process works as follows. A route seeking source node is generating a route request message (RREQ) and floods the whole MANET with this RREQ. Every RREQ forwarding node creates a reverse route entry in its own routing table as a next hop information pointing to the originator of the request. Then the RREQ is forwarded. Again, when the destination node receives the RREQ it generates a RREP. This reply is then unicast back the reverse route to the originator of the RREQ. After that process, both the source and the destination node have valid route entries to each other in their routing tables. In opposition to DSR nodes do not know the complete route to a destination node but only the next hop node in the direction to the destination node. Table 2.5 presents the routing table entries of four involved ad-hoc nodes (CN, IN2, IN1, and MN). Every cell of the table represents the routing table of a specific node at a specific step in the route discovery procedure. There are three columns per cell. The first represents the destination of a route while the second stands for the next hop where packets are to be sent to reach the destination. The third column is the distance to the destination. The first two lines (representing the forwarding of the RREQ through the MANET) show how IN1 and IN2 receive the RREQ and create reverse route entries to the MN. After the RREQ is received by CN (line three) the CN creates a reverse route entry, too and transmits the RREP. The way back of this RREP is enlisted from line four. While forwarding this RREP down the reverse route, all intermediate nodes create a route entry pointing to CN in order to be able to forward data packets from the MN to the CN. These entries are refreshed and maintained by the *neighbourhood management* as part of the route maintenance.

The neighbourhood management may be provided by layer 2 information (e.g. if

	CN	IN2	IN1	MN
1.			MN MN 1	
2.		MN IN1 2	MN MN 1	
3.	MN IN2 3	MN IN1 2	MN MN 1	
4.	MN IN2 3	MN IN1 2 CN CN 1	MN MN 1	
5.	MN IN2 3	MN IN1 2 CN CN 1	MN MN 1 CN IN2 2	
6.	MN IN2 3	MN IN1 2 CN CN 1	MN MN 1 CN IN2 2	CN IN1 3

Table 2.5: AODV example routing tables

W-LAN is used). When layer 2 detects the loss of connectivity to a neighbour node it informs layer 3 (the routing layer) and then the appropriate neighbour node is deleted from the routing tables as well as all routes according to that neighbour node.

If layer 2 does not provide information about a node's neighbours the routing protocol itself has to detect when an enlisted node is no longer reachable. In DSR nodes overhear the forwarding of data packets from their neighbours and therefore, reset the time out of the appropriate routing table entries while another approach is used in AODV. Like in DSDV and OLSR, AODV uses HELLO messages. If a mobile node does not receive HELLO messages from an enlisted neighbour node for three consecutive HELLO intervals it assumes that this neighbour node is not longer within the node's radio range and it will therefore delete the neighbour from its routing tables. This is defined in [20]. The typical HELLO interval is one second. If an intermediate node recognises the loss of a neighbour node a route error message (RERR) is sent along the path back to the source of the route.

An analysis of the differences in the performance between AODV and DSR can be found in [22]. The authors present the packet delivery fraction (PDF), average packet delay, and normalised routing load in terms of node mobility (pause time in random movement) and node density. The authors conclude that DSR outperforms AODV in less 'stressful' situations, i.e., less mobile nodes and less node mobility. With increasing node mobility and more mobile nodes AODV outperforms DSR. The reason for the poor performance of DSR in 'stressful' situations is that DSR does not delete stale routes in opposition to AODV which uses HELLO messages for neighbourhood management. Furthermore, the authors suggest that both on-demand protocols would benefit from

using congestion related metrics instead of hop counts. A congestion and hop count related metric for discovering Internet gateways is one goal of this thesis (section 5.4). The congestion factor and the hop count of a route is because the hop count plays an important role to the provided bandwidth of a route. See Appendix A for details. Note, that a leased used but longer route will not automatically provide the most bandwidth.

The AODV protocol has been enhanced to save bandwidth and to improve its general performance. These improvements are described in [28]. The authors give an overview of the functionality of AODV and propose a series of improvements like the *expanding ring search* techniques. This technique limits the time to live (TTL) of RREQs to prevent flooding the whole MANET cluster. When a RREQ was not successful the TTL of the next RREQ is incremented. The feature of the expanding ring search is implemented in the NS-2 simulation software of this thesis and is used when a MN is requesting for a CN in the local ad-hoc cluster. The actual implemented algorithm is depicted in section 6.4.1.

Another improvement to AODV of [28] is the *local repair* of broken routes. This is valuable when routes within an ad-hoc network break due to node movement. The node that detects a break does not send a RERR message back to the originator of the route but it tries to repair the route while buffering data packets. To achieve this, the node sends a RREQ to the destination of the original route. If this request is successful it forwards the buffered data packets to the destination node.

Furthermore, [28] introduces a *gratuitous route reply* (GRREP). While a route request for a specific node is running and an intermediate node knows about a route to the requested destination node it will answer by sending a gratuitous RREP to the originator of the request. Additionally, it will send a GRREP to the destination of the initial request, thus the destination node has a valid reverse route to the originator. In general, reverse routes are necessary for higher level protocols like TCP to allow the delivery of acknowledge (ACK) packets.

2.6 Quality of Service

2.6.1 Definition of Quality of Service

In general, Quality of Service, or QoS, is a term that defines the cooperation of components in a telecommunication system to provide various constraints to the end-user. For example, a communication system with quality of service features is able to provide communication parameters like a guaranteed minimum bandwidth or a maximum packet delay for data packets. More typical quality of service constraints are an allocated bandwidth for a specific logical communication connection or the maximum packet loss rate e.g. a connection via the Internet. Different applications have different demands for quality of service. A Telnet [75] or Voice over IP (VoIP) application does not need an extra high bandwidth link but the packet delay should be as short as possible to avoid delay in the communication. In opposition, a file download needs as much bandwidth as possible to download data files faster whereas the delay of a single data packet of a file transfer is not essential.

2.6.2 Quality of Service in the Internet

In a wired and static network like the Internet routers interconnect subnetworks (structured topology). Due to this structured concept Internet routers may reserve resources, e.g. minimal bandwidth, maximal delay, to provide quality of service to network hosts. Another approach for quality of service is not based upon reservation but on prioritisation.

A commonly known approach for Quality of Service in the Internet is called DiffServ [67, 68]. For signalling the priority of a packet Diffserv uses the first six bit of the Type of Service (tos) field of the IP header. The signalling of the packet's priority is initiated by the sending host and is the drawback of DiffServ since a selfish node can set wrong values. I.e. one node can set values in the tos field to prioritise its own packets. Routers along the route from the sender to the destination decide how the packet is being handled only by the tos field. With DiffServ data packets of specific applications can be privileged i.e. that two applications seeking connectivity to the Internet via a specific router can set priorisation values to reduce packet delay for example for a VoIP

connection to let the VoIP's data packets being routed prior to other data packets.

Another approach for reserving network resources like bandwidth is the IntServ approach with the Resource ReSerVation Protocol (RSVP) [55]. In opposition to DiffServ with IntServ the path via a number of routers can be reserved for a specific connection.

RSVP is not a routing protocol and therefore not responsible for the delivering of data but RSVP works with routing protocols. RSVP is receiver oriented and reserves network resources along a routing path. It operates as follows. A host needing a specific quality of service constraint sends a *path message* along the already by the routing protocol established route to the destination node. While travelling from router to router the message collects quality of service data from each router. When the path message is received by the destination router the destination router reserves resources and sends a *resv message* along the reverse path back to the originator. By forwarding the resv message every router along the reverse path reserves resources. As a result, resources for the complete route in both directions are reserved. Note that the quality of service reservation is only negotiated by network routers and not the network hosts.

2.6.3 Quality of Service in Wireless Networks

The nature of the radio interfaces of wireless networks leads to collisions when transmitting data frames if network interfaces transmit frames simultaneously. One solution to avoid collisions is the TDMA (time division multiple access) mechanism. In TDMA time slots are assigned to radio network interfaces and the interfaces are only allowed to transmit within their own time slot(s). Other time slots are reserved for other nodes' radio interfaces. Time slots can even be assigned dynamically to network interfaces. An example for a system that uses TDMA is WiMAX [44]. WiMAX uses dynamic time slot assignment and even multiple time slots can be assigned to a single radio interface to increase air time.

The IEEE 802.11 standard (W-LAN) does not use TDMA but CSMA/CA (carrier sense multiple access/collision avoidance). Here network interfaces listen at the medium if it is occupied or not. If not the interface will wait a short period of time (IFS + random back-off) before it transmits. If the medium is occupied the interface will wait until the medium is free plus the IFS time and the random time for transmission. If the

medium is then free the interface will transmit. Different IFS times (SIFS, PIFS, and DIFS) are for a priorisation of e.g. control messages like RTS/CTS messages and data packets. More details about the access method in W-LAN are given in section 2.4.3.

W-LAN uses the RTC/CTS mechanism that reserves air time for a specific node when transmitting a frame. Details on the RTC/CTS mechanism are given in section 2.4.3. All other nodes then have to wait until the reserving node has finished transmitting. If that transmitting node has a slow interface the surrounding neighbour nodes will suffer. The original IEEE 802.11 standard has no quality of service extensions to limit such problems.

With the IEEE 802.11e [63] standard quality of service is included into W-LANs by introducing priorisation parameters. The extension allows the definition of traffic types with low priority and high priority. The higher the priority of the traffic a node is transmitting the shorter the node waits for sending the frame in the competition for air time. Thus, data packets of a prioritised application are transmitted first. This is achieved by a Type of Service [64] field which is compatible to the Type of Service field in the IP header. The type of service field is 8 bit long. The first three bits are used for precedence reasons and denotes the priority of the frame. The forth bit (the `D` bit) indicates that the connection is delay critical. The `T` and the `R` bits request throughput and reliability respectively. With the seventh bit (`M`) a least monetary connection can be established (least cost). A bit combination of `0000` indicates a standard connection. The last bit is reserved and always set to zero.

The RSVP protocol and the IEEE 802.11e standard would improve the quality of service of an ad-hoc mobile network. RSVP would reserve available resources at each intermediate node of a multihop route whereas IEEE 802.11e provides different quality of service constraints for different types of traffic. Since it is not the aim of this thesis to extend ad-hoc routing protocols with quality of service features but to investigate a newly developed Internet gateway discovery protocol bandwidth improving features were integrated into the Internet gateway discovery protocol. Next, quality of service within a wireless multihop ad-hoc environment is discussed.

2.6.4 Quality of Service in Ad Hoc Networks

Due to the mobility of nodes in mobile ad-hoc networks multihop routes may break or nodes change the provided quality of service resources at any time. A multihop ad-hoc routing protocol must respect this. In proactive approaches ad-hoc nodes can discover quality of service routes after they have collected information about the ad-hoc network's status in terms of the actual topology, the amount of traffic that is send and forwarded by ad-hoc nodes, and the reliability of links if a link-layer feedback is provided. Another typical quality of service constraint is the delay a data packet will experience when travelling through the multihop ad-hoc network. An approach for discovering quality of service routes within ad-hoc networks is that a quality of service route seeking node has a complete and actual map of the whole ad-hoc cluster and can then pre-calculate a sufficient route. This requires proactive routing approaches.

When using reactive approaches a quality of service requesting node has to discover the quality of service route prior using it. While the route discovery process is running intermediate nodes can include their available quality of service resources into the route request and reply. Thus, the requesting node is able to decide if the discovered route is sufficient or not. The protocol has to ensure that the request reply mechanism does not inhibit possible quality of service routes.

The layer 2 basis of an ad-hoc network may also be used for quality of service discovery. A TDMA system is able to provide guaranteed bandwidths to nodes (as long as the nodes stay in contact to each other) by reserving time slots. Additionally, the layer 2 protocol may decide if a link to a neighbour node provides requested quality of service resources or not and then send messages (route requests or replies) using that link.

In general, in case of mobile ad-hoc networks a quality of service routing protocol should regard the following points.

- discover QoS routes
- reserve QoS resources
- maintain QoS routes

The approach of [54] uses a TDMA for quality of service routing within ad-hoc networks. The approach is based upon layer 2 and it uses cross-layer features that prevent porting to other layer 2 protocols. The authors present a routing protocol based on AODV where quality of service routes are established with the (modified) route request and reply mechanism of AODV. The protocol is able to find routes bypassing areas of congestion within the ad-hoc network by dropping route requests that would build a not satisfying quality of service route. Additionally, the protocol allows the prediction of the quality of service available bandwidth for a specific end-to-end connection by a given algorithm. A drawback of the presented approach is that two nodes cannot establish quality of service routes simultaneously as the requesting of routes is not coordinated and thus time slots cannot be correctly assigned.

The "Core Extraction Distributed Ad hoc Routing (CEDAR)" [56] is a protocol for calculating QoS routes in ad-hoc networks that provide sufficient bandwidth resources. CEDAR assigns ad-hoc nodes to "cores" proactively that are responsible for calculating quality of service routes and thus they are routing instances. RREQ messages are sent reactively and forwarded from one core to the next to reduce protocol overhead caused by network wide flooding.

A protocol called On-Demand Delay-Constrained Unicast Routing Protocol (ODRP) is presented in [53]. ODRP establishes routes in ad-hoc networks with delay constraints. To discover delay constraint routes within an ad-hoc network ODRP firstly probes existing minimum hop count routes if they fulfil the quality of service constraints. This is achieved by timers at the source and the destination node. The timers are to be exactly synchronised in time in order to give feasible results about the probe's total delay (not round trip time). The authors do not mention how the timers are synchronised. If the existing path does meet the quality of service requirements the destination node initiates a quality of service route discovery reactively by partially broadcasting the request, i.e. the request is only forwarded into the direction of the source node. Every intermediate node accumulates the delay in the request. If the request is received by the source node a delay constraint path has been discovered.

Another approach for discovering quality of service routes within an ad-hoc mobile network is presented in [52]. It is based on DSDV [15] and uses layer 2 information and

assumes synchronous clocks in all ad-hoc mobile nodes. From the layer 2 information the authors calculate a link available bandwidth to adjacent nodes and pre-compute spare routes to other nodes. For the case a route to a specific destination breaks these spare routes are used immediately.

The above presented quality of service routing protocols have in common that they use link-layer information for routing decisions. This thesis excepts cross layer approaches for routing and concentrates on easily portable algorithms. Therefore, publications on quality of service routing within ad-hoc networks without link-layer feedback are presented next.

There exist extensions to the reactive DSR [58] and AODV [59] protocols that allow mobile ad-hoc nodes to discover routes that fulfil quality of service constraints. These extensions extend RREQ messages by quality of service fields that stand for link available bandwidth and delay. Additionally, RREQ messages are only being forwarded by nodes that fulfil the required quality of service constraints and dropped by nodes that do not fulfil these constraints. As a result a specific RREQ received at the destination node has always travelled along a multihop path that is sufficient. Additionally, since nodes that do not fulfil the quality of service constraints drop the RREQ the protocol overhead is reduced.

Another quality of service routing approach is called Ad hoc QoS on-demand routing (AQOR) [57]. AQOR works firstly similar to AODV by sending RREQ and RREP messages. Also RREQ messages are only forwarded along nodes that fulfil the required quality of service constraints. The main difference between AODV and AQOR is that AQOR uses HELLO messages that contain information about a node's usage, i.e. the traffic forwarded by that HELLO messages sending node. If a route is no longer able to provide the requested quality of service the destination node of the route sends a route update message that rediscovers a new quality of service route. This lets the originator of a quality of service route know firstly that the old route is no longer valid and secondly the originator knows an alternative quality of service route to the destination node. This thesis uses HELLO messages for transporting information about the forwarded traffic by a specific gateway and not by specific ad-hoc nodes. The main drawback of AQOR is that nodes can only include information about their own traffic

that they are forwarding. If nodes are not located within the transmit range of each other but within the interference range they cannot know about the traffic surrounding neighbour nodes are forwarding without link-layer feedback. Finding the best quality of service route in terms of e.g. bandwidth within the ad-hoc cluster is not the task of this thesis and needs further development and research by the community.

One could think about investigating the bandwidth an ad-hoc network provides by probing the ad-hoc available bandwidth between specific network nodes (file transfer). This would lead to enormous complications when working with congestion related routing metrics since then, depending on the actual routing protocol, nodes would be affected by that probing and initiate new route discovery procedures themselves. Additionally, assumed that multiple nodes would probe the ad-hoc network for bandwidth the limited bandwidth resources of a wireless system would be wasted.

2.7 Conclusion

This chapter introduces the Internet and the functionality of the Internet. It discusses the approaches the Internet utilises for delivering data packets to a destination node using routing protocols and routers and presents the protocol stack in general. It introduces the Internet Protocol and Internet protocol addresses. Furthermore, the chapter presents different network topologies.

Mobility support was not implemented into the Internet originally. With the introduction of MobileIP users are now allowed to move to other logical subnetworks of the Internet while still being logically connected to their home network. Such a mobility is called macro mobility.

Wireless radio communication links are used to grant mobility to Internet users. The drawback of wireless links is the limited radio range of devices using such wireless links. Short range mobility support is called micro mobility.

Ad-hoc networks can route data packets via multihop paths through a number of mobile devices to destination device. Therefore, ad-hoc networks can be used to extend the physical radio range of a single mobile device. The task with ad-hoc networks is to find reasonable multihop routes through the ad-hoc network while every device the ad-hoc network consists of is allowed to move randomly around and to be switched off.

Further, new devices may appear every time if a mobile device is switched on by its user.

Ad-hoc networks in general are not connected to the Internet because of their different routing approach. To connect the devices of ad-hoc networks with the Internet a new node is introduced in the next chapter.

Chapter 3

Interconnecting MANETs into the Internet

3.1 Overview

In opposition to the Internet with predefined routers that connect subnetworks, ad-hoc networks use a flat or unstructured routing approach. A new node, the gateway or Internet gateway, is introduced to interconnect ad-hoc mobile nodes to the Internet. The Internet gateway is a part of both, the ad-hoc network and the Internet simultaneously. To get connected to the Internet, the Internet gateway must firstly be discovered by the mobile devices of an ad-hoc network. There exist different approaches, based on known ad-hoc routing protocols, to achieve this discovery.

This chapter presents solutions for the Internet gateway discovery in ad-hoc networks and discusses the routing between the structured Internet and the unstructured ad-hoc networks using Internet gateways.

Next, the Internet gateway nodes are introduced.

3.2 Internet Gateways

The hierarchical routing algorithms in the Internet and flat multihop routing algorithms of ad-hoc networks, or MANETs, are to be combined in order to give members of an unstructured ad-hoc network access to the structured Internet topology [24]. In general,

an interface between networks that use different routing protocols is called a gateway. In this thesis the gateway is called *Internet gateway* since it connects ad-hoc networks that use ad-hoc routing protocols with the Internet which uses hierarchical Internet routing protocols.

Standard ad-hoc routing protocols for mobile nodes in an ad-hoc network do not provide mechanisms for detecting such an Internet gateway. Therefore, the standard protocols have to be improved in order to allow mobile nodes of an ad-hoc network to discover a gateway and to use it for Internet connectivity. The Internet gateway either may be a static node or a mobile node.

A mobile Internet gateway is equipped with two wireless interfaces, one as part of the ad-hoc network and the second as an uplink to a, e.g., W-LAN access point (AP) and the structured network. The gateway is part of both networks simultaneously and in opposition to W-LAN APs it may be mobile.

3.2.1 Example Scenario for Internet Gateways

An example scenario for Internet gateways is depicted in Figure 3.1. On the left side an ad-hoc network is depicted. Within the ad-hoc network a specific mobile node (MN') is located. Other intermediate mobile nodes are labelled with IN1, IN2, and IN3. These four nodes (and the other nodes, represented by small circles) form an ad-hoc network. Dotted lines indicate wireless links. Solid lines stand for wired links used to connect the access point (AP) to the Internet via a router. Above the ad-hoc network an Internet gateway (GW) is depicted. This Internet gateway is connected wirelessly to the AP and it may be installed static or mobile. The MN may move from its original position in its home network on the right into the ad-hoc network (MN→MN').

In general, in order to get an Internet connection mobile nodes need to discover the Internet gateway using enhanced ad-hoc routing protocols. Section 3.3 discusses how Internet gateways are discovered by mobile nodes and the way mobile nodes route data packets through the Internet gateway to a correspondent node in the Internet.

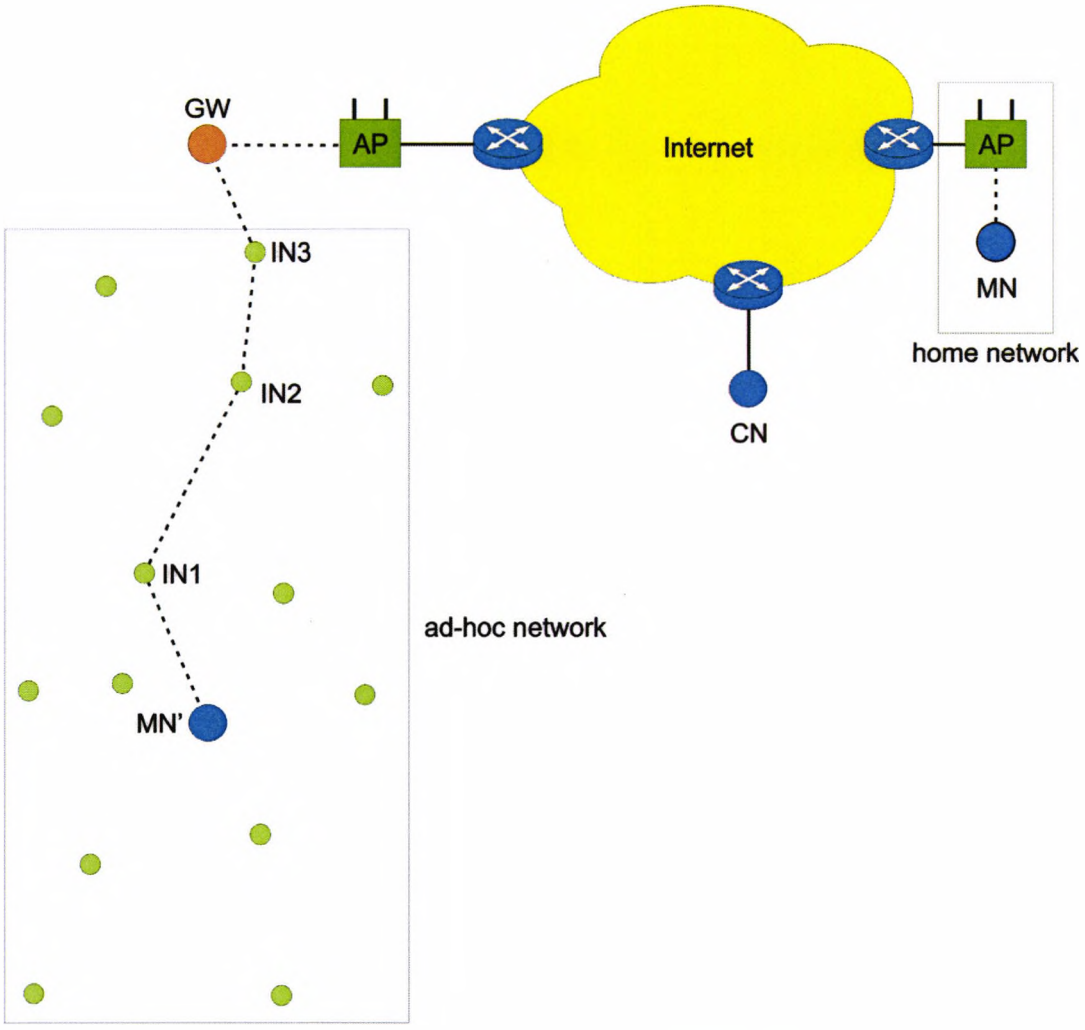


Figure 3.1: The integration of ad-hoc networks with the Internet

3.2.2 Related Work on Internet Gateways

The connectivity of ad-hoc network nodes to the Internet and their ability to roam between different Internet gateways to allow mobile end-users ubiquitous access to the Internet is the objective of the IPonAir project [38].

Further, the objectives of the IPonAir project are to define and develop wireless architectures for mobile Internet access. This Internet access is provided by several radio access technologies like long range GSM or UMTS and short range access technologies like W-LAN. The task is to achieve an ubiquitous Internet access depending on the actual available radio access technology (coverage) and to roam between different radio access technologies seamlessly. The IPonAir project defines the Internet gateway in different ways. One way is a static Internet gateway. A static Internet gateway may be

used for extending the service range of a W-LAN installation with an ad-hoc network. If the Internet gateway is static it may be integrated into the access point device to create a single device (layer 2 and layer 3 device). This saves the wireless link between the access point and the Internet gateway. Nevertheless, the Internet gateway must be discovered by the mobile nodes of the ad-hoc network using Internet gateway discovery protocols.

Further, in the IPonAir project a mobile Internet gateway is discussed. The advantage of a mobile Internet gateway is that it can be installed on, e.g. a vehicle. Passengers of this vehicle can now connect via an ad-hoc network to the Internet gateway and the Internet gateway connects the whole ad-hoc network via a static access point to the Internet.

Even if the Internet gateway is static it may be connected wirelessly to the structured Internet (e.g. using WiMAX). This is useful to connect mobile nodes with the Internet by extending the service range of W-LAN access points using multihop ad-hoc networks. This application for Internet gateways can be used in the case of a disaster or in any other infrastructureless environments. In general, the task is to discover an Internet gateway whether it is installed static or mobile.

In [23] a description of addressing schemes for the interworking of ad-hoc networks and the Internet is given. MobileIP for IPv4 is used for this purpose, and thus foreign agents (FA) are included into the concept of [23]. Furthermore, in [23] MMCS (MIPMANET Cell Switching) is introduced where nodes only switch to a new FA if the new FA is at least two hops closer than the old FA, and if the nodes receive two consecutive advertisements from the new FA. AODV is used for performance simulations with link layer feedback and therefore no HELLO messages for neighbourhood management are facilitated. For portability reasons, this thesis assumes that the link layer does not provide feedback of the quality of links and therefore HELLO messages are used for neighbourhood management. The authors of [23] present simulation results for packet delivery fractions and MobileIP registrations. Of course, results for gateway discovery time or handover time are not presented.

In [24] AODV is used for range extension to access points. Again MobileIP is used as a macro mobility protocol. One difference to [23] is that FAs at which mobile nodes

are registered with can answer to RREQs from the ad-hoc network since the FA is then aware of the destination node of the RREQ. The authors claim a randomisation of rebroadcasting ADVs to prevent synchronisation and subsequent collisions. In [24] the authors do not examine the influence on traffic within the ad-hoc cluster and they do not investigate handover procedures and the time mobile nodes in the ad-hoc cluster need to discover a gateway node or FA, respectively.

An overview of addressing schemes for IPv6 in combined networks and gateway discovery algorithms is given in [25]. The authors give a good overview of the principles of addressing in ad-hoc and structured networks as well as the combination of both. Furthermore, they describe approaches for the so called *Path Selection Problem*. This problem occurs in overlapping ad-hoc clusters, when a specific node recognises two gateways with different prefixes. The question is which route to a correspondent node is the best route for the originating node. First, packets can be routed through the first MANET to the first gateway and through the fixed Internet to the second gateway and cluster. Second, the data packets may be routed directly through the two overlapping MANETS without using any gateway services. The authors' solution for the Path Selection Problem is a prefix cache for mobile nodes where nodes record information about neighbour clusters and neighbour gateways.

Additional, in [25] the principle of gateway discovery and address auto configuration for MANETs is presented and different address types of IPv6 are introduced for the usage in ad-hoc networks. This is discussed in section 2.5 on page 28 in this thesis.

The authors of [23] examine the overhead caused by MobileIP, AODV, and MobileIP re-registrations. Additional, the delay of data packets and the fraction of received packets related to the number of nodes that register at a FA is examined. They do not investigate the influence of parameters like advertisement/beacon intervals or solicitation time outs and they do not stress the ad-hoc network with heavy traffic load near the saturation border. The saturation border of one wireless link is typically a little bit more than a half of the gross data rate and it is additionally less if logical communication links are using multihop routes. This thesis addresses different node densities and node movement to examine their influence on gateway discovery and handover times.

In [34] the proactive, reactive, and hybrid [30] discovery algorithms are compared by means of packet delivery ratio, packet delay and control overhead. The author does not examine how long a node needs until it discovers a gateway when it enters a MANET or how long a handover from one gateway to another lasts. In fact, it is not investigated how long a specific node needs to perform a handover from one gateway to the next, although nodes perform handovers. The influence of varying traffic load within the cluster and the mobility of nodes (pause time, max. speed) is not investigated and therefore, in opposition to this thesis, the performance of the discovery algorithms with these parameters is not evaluated.

In [27] parameters like the number of gateways within a MANET cluster providing Internet connectivity and the mobility of the MNs were investigated. The authors use the Average Link Duration as a scale for the mobility of nodes to examine the PDF (Packet Delivery Fraction), signalling overhead and packet delay in established MANET clusters. In this thesis the emphasis is on the gateway discovery time after a node has entered a group of established mobile nodes within a MANET. Thus, parameters like the interval time of gateway advertisements and mobile node solicitations as well as the density of nodes within an ad-hoc cluster and additional traffic created by other nodes are examined.

3.3 Solutions for Internet Gateway Discovery

As mentioned before, the standard ad-hoc routing protocols do not provide the functionality of detecting Internet gateways, thus the protocols have to be extended. The extensions to the standard ad-hoc routing protocols are based upon special ad-hoc routing messages.

In section 2.5.2 on page 29 different routing protocols for ad-hoc networks of proactive and reactive kinds were discussed. In general, proactive protocols announce information without demand. On the other hand, reactive protocols work on a on-demand basis. According to that, there exist proactive and reactive gateway discovery approaches. Proactive gateway discovery approaches provide information about an Internet gateway (and the multihop route to it) without demand while reactive gateway discovery approaches work on a on-demand basis.

3.3.1 Proactive Approaches

Proactive approaches for gateway discovery work on a pre-given routing information without demand. The gateway sends routing information without request into the MANET. It can accomplish this by flooding the MANET with so called *gateway advertisement* messages (ADV) periodically [7]. Gateway advertisements contain information about the gateway's address. As a result, advertisement receiving nodes know the gateway's address and a route to the gateway from the reverse path of the advertisement. Then nodes can use the gateway for their Internet traffic and do not have to discover the gateway by themselves.

In [26] an enhancement of the proactive discovery algorithm is described. Here, the gateway sends ADVs periodically but MNs do forward this ADV only if the received ADV describes a shorter route to the gateway than any other ADV from other gateways. This results in the effect that every gateway in overlapping ad-hoc clusters is flooding only that part of the cluster where it is located in and thus, the routing overhead in the proactive approach for gateway discovery is reduced. In a second step, the gateway additionally sends advertisements on demand beside the standard regular advertisements if it detects changes in the cluster topology where it is attached at. The authors call this *adaptive advertisement* in opposition to [29] where the authors suggest a dynamic TTL of gateway advertisements. In [26] advertisements are sent on demand when they are needed but with a fixed TTL.

3.3.2 Reactive Approaches

In reactive gateway discovery approaches, nodes of a MANET request for gateway information on-demand. In general, this is accomplished by network flooding using modified RREQ messages. Then, either the gateway itself or any other intermediate node, that is aware of the gateway, answers to the request by sending a modified RREP message back to the originator of the modified RREQ. This modified RREQ is called a *solicitation* message (SOL) [7]. The reactive approach for gateway discovery is closer to the reactive principles of the DSR and AODV routing protocols.

The authors of [23] describe a method for reducing routing overhead by combining a RREQ for a specific node with a request for an Internet gateway, or FA respectively.

Any intermediate node with a valid gateway route may answer to this request. The disadvantage of this enhancement is that if a node has an invalid gateway address and is not aware of this invalidity, wrong information is sent into the MANET and thus, in the implementation of this thesis only gateway nodes answer to solicitation messages to ensure that gateway information is always valid. As a result, a solicitation answer receiving node is always sure that the information included into the solicitation answer is true.

An alternative approach on gateway discovery can be found in [31]. There, the authors place one gateway node with several attached access points to the ad-hoc cluster. They call their proposal "Common Gateway Architecture" and use just one gateway, and therefore there will be no additional MobileIP overhead caused by multiple gateways with different prefixes since all nodes in this ad-hoc cluster use the same IPv6 prefix and thus, they do not need to perform MobileIP re-registrations. In future wireless systems that provide IP-based mobility users will roam around between different subnetworks and thus, this approach must be interpreted as an isolated solution. The advantage of the authors' approach is in the reactive gateway discovery method. The access points will answer to RREQs directed to the gateway, since they always have a valid route to it (fixed wired link). Thus, the MNs within the cluster find short cut routes to a gateway via the access points. The authors do not take a look at the proactive discovery method. In the proactive method all access points would flood the network and therefore this approach will perform very badly since the flooding is multiplied by the number of used access points. Furthermore, the authors give no results how long a specific node needs to find a valid route to the gateway and since they place only one gateway into their scenario no handover procedures between gateways are considered.

Additionally, a mathematical view on gateway discovery algorithms is described in [29]. The authors calculate routing overhead for the proactive and reactive methods and they propose an "Adaptive Gateway Advertisement" with a dynamic adjustable TTL with an optimum value of 2. The authors did not simulate varying interval times and additional traffic within the MANET cluster and do not give results of gateway discovery and handover times.

3.3.3 Other Approaches

Beside the two main proactive and reactive gateway discovery approaches other approaches exist. First, a mixture of the advertisement based and solicitation based algorithms is discussed. Furthermore, a new kind of proactive gateway information spreading in MANETS is presented.

Hybrid Approach

The hybrid gateway discovery algorithm is a mixture of the advertisement and solicitation based algorithms. Like every IP packet, advertisements have a time to live (TTL) field in their header. The TTL value of a packet is set by the originator of the packet. Every time a packet is forwarded by a node this TTL value is decremented by 1. After a number of forwardings the TTL is reduced to zero and then the packet will be discarded. Thus, packets have a "range". This mechanism prevents packets to stuck in a routing loop.

A mixture of the two main gateway discovery approaches (advertisement based and solicitation based, cp. section 3.3.1 on 54) is described and evaluated in [30]. There, the gateway broadcasts advertisements periodically but with a limited TTL, and thus they do not flood the complete MANET but only the surrounding area around the gateway. Distant mobile nodes from the gateway do not receive advertisements and will ask reactively for a gateway by sending solicitations. This hybrid algorithm combines benefits from the proactive and the reactive gateway discovery approaches. The proactive part contributes high mobility for nodes while the reactive part is responsible for relatively low routing overhead per node. The authors present simulation results and claim the best advertisement rate to be 10 - 15 seconds. Additionally, the authors give an optimal TTL value of 2. This results in high connectivity while keeping overhead costs low.

The hybrid gateway discovery algorithm uses this TTL field to control advertisement messages. The originating gateway sets the TTL value of its advertisements to a number smaller than the allowed maximum of the fundamental ad-hoc routing protocol. As a result, nodes that are located behind the TTL range of the advertisements do receive gateway advertisements and therefore, these nodes have to request for a gateway by broadcasting solicitations as described in the reactive gateway discovery approach.

In [29] the hybrid approach is investigated and the authors suggest an optimum TTL value for advertisements of 2 to 3 hops.

The benefit of this mixed approach is that the number of periodic advertisement forwardings is limited dramatically but (assumed that most users are located in the vicinity of a gateway) nearly every member of the ad-hoc network is aware of the gateway.

HELLO Approach

The HELLO message based algorithm provides gateway information to mobile nodes without demand. This new approach uses AODV HELLO messages for spreading gateway information in the ad-hoc network [1]. The gateway information is spread by forwarding the information about a gateway with every transmitted HELLO packet from one ad-hoc node to the next node. The functionality of this approach is described in chapter 4 on page 65 in details. Since the HELLO algorithm provides gateway information without demand it is related to the proactive type of gateway discovery approaches but uses no explicit gateway messages like advertisements. This thesis compares the HELLO approach with the advertisement based and the solicitation based approaches. The HELLO approach is also evaluated in [2, 3].

3.3.4 Adaptations to Selected Protocols

Adaptations to DSDV

As mentioned before, in DSDV MANET nodes include their own address and information about their neighbour nodes into HELLO messages that are broadcast to all neighbour nodes periodically. After a number of HELLO cycles, every node in the MANET is aware of every other node and has a valid route to any other node. In [46] this is expanded by Internet gateway discovery features. The Internet gateway (like any other node in DSDV) includes its own address into its HELLO messages. With every new HELLO message the Internet gateway address is distributed deeper into the MANET cluster. MANET nodes compose a routable global IPv6 address with the Internet gateway's prefix and their own fixed suffix. Additionally, as a result of this process all MANET nodes have a route to the Internet gateway and may choose

between multiple detected Internet gateways. In [46] this decision is based on the hop count to the Internet gateways.

Adaptations to OLSR

In [45] an approach to the OLSR ad-hoc routing protocol for gateway discovery features is proposed. The authors introduce a new message type to OLSR, called Prefix Advertisement (PA) messages. These PAs include the gateway's IPv6 prefix and MANET nodes compose globally routable IPv6 addresses with their own fixed suffix. PAs are flooded into the MANET using the MPR structure of OLSR.

Adaptations to DSR

The adoption for gateway discovery for DSR is suggested in [17]. The authors propose a gateway solicitation request and reply mechanism. The reply is a proxy reply from the Internet gateway to indicate a MANET node that the requested node is located within the MANET although it is located in the Internet. Therefore, the reply is a bluff. The problem with this approach is that if the requested node is located within the MANET, both the requested node and the gateway answer to the request. This leads to confusing routing information since the requesting node may choose the gateway for connectivity to the CN, although the CN is located within the MANET. A solution for this problem is that the requesting node waits a small period of time to give both, the gateway and the CN, a chance to answer.

Adaptations for AODV

This thesis uses AODV as the basic ad-hoc routing protocol for investigations on ad-hoc networks with Internet connectivity and therefore, the Internet gateway discovery expansions for AODV are explained in more detail.

The two main principles for Internet gateway detection (proactive and reactive) can be used for AODV while AODV still remains reactive even if it uses a proactive gateway discovery approach. This means that only the expansion for gateway discovery uses proactive algorithms. This thesis examines proactive and reactive extensions for gateway discovery that are based on the AODV route request/reply mechanism [7].

Proactive Gateway Discovery for AODV uses modified route requests (RREQ) that are flooded into the MANET. The modification consists a special flag which is called the I-flag [34]. This flag indicates that the route request is not an ordinary route request and that it is originated by an Internet gateway. The modified RREQ is called a gateway advertisement (GWADV) or RREQ_I. The source address in the message header is the address of the gateway. The destination address of such a RREQ_I is set to the broadcast address. Thus, every node in the MANET receives the RREQ_I and forwards it. A sequence number indicates the freshness of the information. Before forwarding the gateway advertisement every node sets up the route to the gateway with the next hop entry pointing to that node where the advertisement was received from. As a result of this procedure, every node in the TTL range of the advertisements has a valid gateway route. In general, the advantage of this approach is that nodes have gateway routes without requesting for a gateway. On the other hand, the periodic flooding of the MANET causes a high consumption of limited bandwidth resources. Note, there is no reverse route entry created in the Internet gateway using this algorithm.

Reactive Gateway Discovery for AODV uses modified route request and route reply (RREP) messages. These modified route requests are called solicitation messages or (RREQ_I) according to [34]. They are sent every time a specific node needs a gateway. Using the standard forwarding procedures of AODV (with reverse route setup) the RREQ_I message is received by the gateway. Then the gateway answers to the request by sending a RREP_I message, again with the I-flag set. Nodes that do overhear such a gateway reply may create or update their own routing tables. This approach is much closer to the reactive nature of AODV and like every reactive approach it leads to delays in packet delivery.

3.4 Routing with Internet Gateways

Mobile nodes of an ad-hoc network need two pieces of information to perform routing via an Internet gateway to connect to their home networks in the Internet. These two pieces are compiled next:

Entry #	Destination	Next hop	Hop count
1	IN1	IN1	1
2	IN2	IN1	2
3	IN3	IN1	3
4	GW	IN1	4
5	-10	GW	4
6	CN	-10	4

Table 3.1: AODV routing table for Internet connectivity

- The address of the Internet gateway. To route data packets and control messages to the Internet gateway ad-hoc nodes must be aware of the gateway's ad-hoc routable address
- A default route pointing to the Internet gateway. All packets that are destined to the Internet are routed via the default route

For gaining the gateway information, i.e. the gateway's address and route to the gateway, either the advertisement based (proactive) or the solicitation based (reactive) approach may be utilised. However, assumed that the modified ad-hoc routing protocol has successfully provided this information to a mobile node in the ad-hoc network. The ad-hoc routing protocol AODV is used for explanation since AODV is used in this thesis. The functionality of AODV and how AODV routing tables are organised is explained in section 2.5.2 on page 37. The scenario setup of Figure 3.1 is used for explanation. In Table 3.1 the resulting routing table of the MN is given.

Each line in the AODV routing table for Internet connectivity via an Internet gateway stands for one entry pointing to a specific destination node. Abbreviations for nodes stand for the appropriate node addresses. Line 1 to 3 are standard routes for MANET routing. They define routes to the intermediate nodes IN1, IN2, and IN3.

In principle, line 4 is a *gateway route* that declares the presence of a gateway as well as the next hop pointing to that gateway and the distance to the gateway metered in hops. If more than one gateway is recognised by the node, more gateway routes appear in the routing table. Line 5 is the *default route* indicated by the destination address of -10 [34]. This default route is set after the node has decided to use a specific gateway for internet communication. If the node is aware of more than one gateway it has to decide which one to use. A metric for the decision can be the distance to the

gateway, packet delay, throughput, or link-durability. Note, a node may have several gateway routes but only one default route. The next hop entry of the default route is the gateway the node has decided to use and the distance to it is copied from the according gateway route (line 4). Line 6 is the resulting route to the CN. The next hop entry for the CN is the default route. This indicates that the CN is reachable via an Internet gateway.

Assumed that the MN wants to deliver a data packet to the CN using the routing table above. First the MN looks for the entry in the routing table with the address of the CN as the destination (line 6). The next hop entry of line 6 is the default route. Therefore, the packet will be destined to the default route which is the destination entry of line 5. The default route is pointing to the gateway's address (GW). The MN will find the gateway by the next hop entry of line 4 which is the IN1 where the packet is finally forwarded to.

3.5 Ubiquitous Internet Connectivity using Gateways

A combination of several wireless access technologies may be used to satisfy the end-users demand of ubiquitous Internet connectivity. Mobile network nodes travel from various access media (like wide range GSM/UMTS) to other wireless (W-LAN) structures while moving. In Figure 3.2 the Internet is depicted as a cloud. Beneath the Internet is a series of radio access networks (RAN) for mobile nodes. On the left side there is a GSM/UMTS network that provides large-area Internet connectivity for mobile users. The second and the third access media are hot spot areas. These hot spots are extended by ad-hoc networks (ad-hoc 1 and ad-hoc 2) to extend the service area of the hot spots using multihop ad-hoc networks and Internet gateways (GW1 and GW2). The Internet gateways are the interface between the different routing domains of the ad-hoc networks and the Internet and thus they are connected to the Internet and the ad-hoc networks simultaneously. The last access media represents any other future radio access technology.

If a mobile node (MN) moves from its home network to a position within a wide range service only it can connect via this wide range network (UMTS network in the left side of Figure 3.2) to a correspondent node (CN) in the Internet. Since the user is

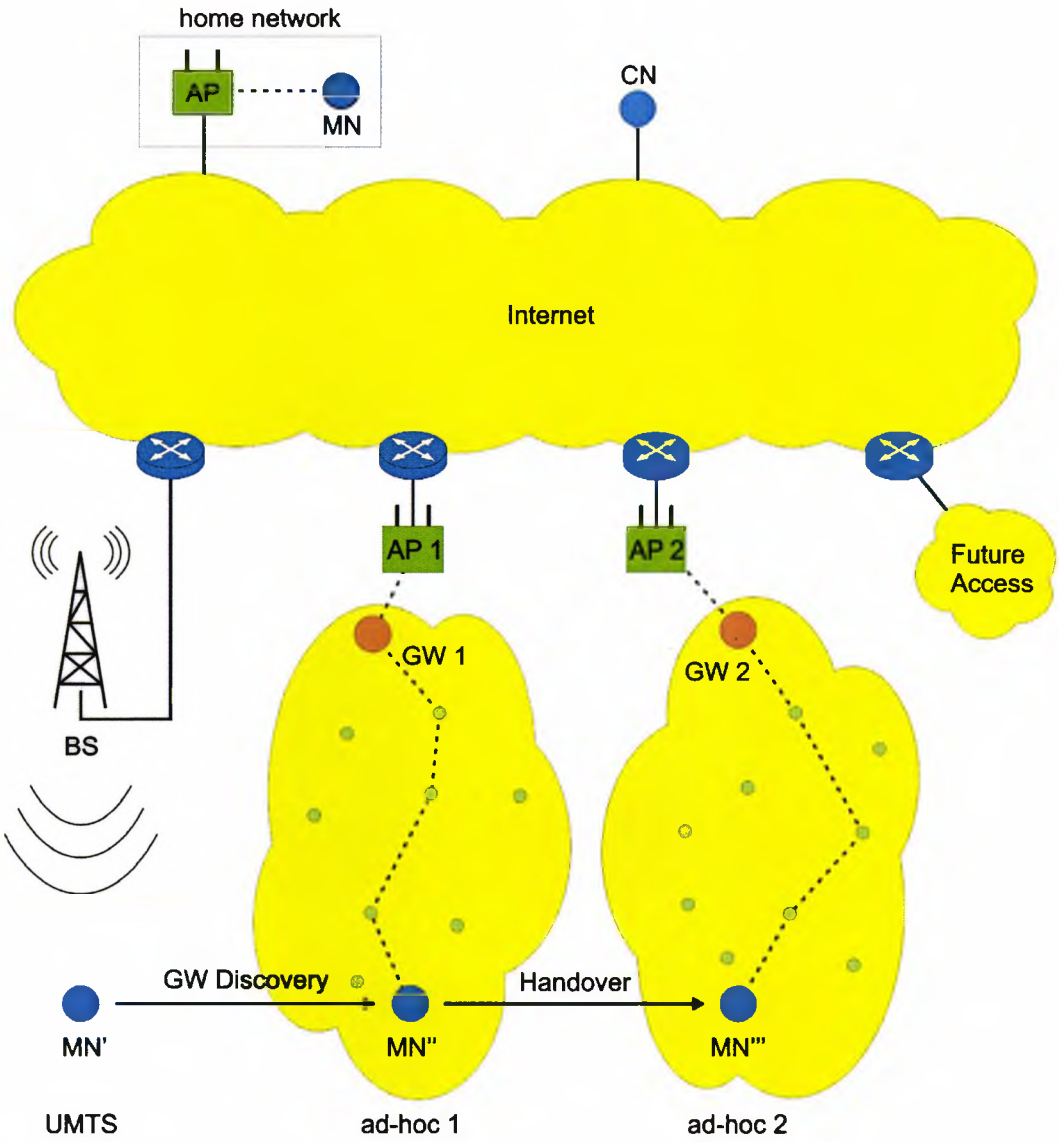


Figure 3.2: Scenario for ubiquitous Internet connectivity using Internet gateways

mobile he can travel to an area with additional accesses to the Internet like hot spot areas. Attached to this hot spot area is a multihop ad-hoc network to extend the hot spot's service area using an ad-hoc gateway (GW1 and GW2). After deciding to change the access media the end-users device needs to connect to the Internet gateway of the multihop ad-hoc network, i.e. the Internet gateway must firstly be discovered. The discovery of the MN to the Internet gateway is achieved by enhanced ad-hoc routing protocols. The time the MN needs to connect to the Internet gateway and to register with its home agent in its home network is called *MN register time.*

The thesis investigates the following attributes of protocols for gateway discovery:

- Register Time

  This is the time a mobile node needs to firstly discover an Internet gateway and secondly to register with its home agent in its home network

- Throughput

  Here, the thesis investigates the throughput a specific Internet gateway discovery algorithm provides to the mobile nodes. This is metered using a test file download of 1MB in size. This size is large enough to generate a reasonable transfer time and small enough to be transferred within the simulation time. Thus it is chosen after empirical issues.

- Protocol Overhead

  To provide Internet connectivity to mobile nodes and connectivity within the ad-hoc network itself the algorithms sent control messages. This is called protocol overhead

- Protocol Efficiency

  By introducing a protocol efficiency index one can easily compare different algorithms. The efficiency index is calculated from the throughput provided by an algorithm and the amount of protocol overhead generated to achieve this throughput

3.6 Conclusion

This chapter introduces Internet gateways to connect the mobile devices of an ad-hoc mobile network with the Internet. These Internet gateways have to be discovered using (modified) ad-hoc routing protocols. There exists proactive and reactive approaches to achieve this goal. Proactive approaches provide the mobile nodes of an ad-hoc network with information about Internet gateways and how they can be reached using multihop routing without demand. Reactive approaches are based on a on-demand basis. Using reactive approaches mobile nodes have to discover Internet gateway themselves.

Further, the chapter discusses the routing with Internet gateways and presents the terms default route and gateway route that are necessary for the upcoming chapters.

Finally, the chapter depicts the scenario one can imagine for the usage if Internet connected ad-hoc networks and how end-users are served with Internet connectivity while still being able to roam around using macro and micro mobility simultaneously.

Chapter 4

Hello Message Based Internet Gateway Discovery

4.1 Overview

This chapter introduces a newly developed Internet gateway discovery algorithm for mobile ad-hoc networks. The new algorithm is based on HELLO messages of the AODV ad-hoc routing protocol and therefore it is called the HELLO algorithm. One task of the thesis is to develop an Internet gateway discovery algorithm that combines the benefit of the advertisement based algorithm (fast discovery and re-discovery of Internet gateways) and the benefit of the solicitation based algorithm (less overhead) and the HELLO algorithm is the solution to this task.

The HELLO algorithm uses HELLO messages for distributing Internet gateway routing information, i.e. information about available Internet gateways within an ad-hoc network. The HELLO algorithm is of the proactive family, since ad-hoc network nodes receive gateway information without demand.

In the AODV ad-hoc routing protocol, every node must always be aware of its neighbour nodes to manage routing information correctly. This is achieved by the neighbourhood management functionality of AODV. The neighbourhood management of the AODV protocol can either be supported by link-layer feedback or by HELLO messages. In order to port algorithms to MAC protocols not supporting link-layer feedback the decision for HELLO message based neighbourhood management was made.

With HELLO messages, every node sends HELLO messages periodically with a time to live (TTL) of 1. Since these HELLO messages are derived from RREP messages a receiving node will handle the included information like a unrequested route reply from its neighbour nodes and therefore will create or update routing table entries to its neighbour nodes. The basic idea of the HELLO algorithm is that Internet gateways and ad-hoc mobile nodes embed information about Internet gateways into HELLO messages to distribute information within the ad-hoc cluster.

First, the basic functionality of the HELLO algorithm is presented in detail. Second, the advanced functionality of the correct distribution of information to more distant ad-hoc nodes is described. Third, the handover of a mobile node between two ad-hoc clusters is illustrated. This is to describe the development of the HELLO algorithm to a full featured Internet gateway discovery protocol for mobile ad-hoc networks. Finally, the chapter ends with a conclusion.

4.2 Basic Functionality of the HELLO Algorithm

Every ad-hoc mobile node must be aware of all surrounding neighbour nodes since neighbour nodes are used for forwarding data packets and protocol messages in ad-hoc networks. This is part of the neighbourhood management of ad-hoc routing protocols. The ad-hoc routing protocol AODV uses HELLO messages for neighbourhood management if no link-layer feedback is available from the underlying layer 2 protocol. In general, if a neighbour node moves out of range of a specific node the specific node then must detect this loss of connectivity. In AODV the neighbourhood management is provided by the periodic transmitting of HELLO messages that indicate all receiving neighbours that the sending node is still available for ad-hoc networking and can be used for routing. The loss of connectivity to a neighbour node is detected by not receiving HELLO messages from this neighbour for a specified time. Originally, HELLO messages are only to inform direct neighbour nodes and therefore they have a limited hop range of one ($TTL_{HELLO} = 1$).

The task of this thesis is to develop and examine a new Internet gateway discovery algorithm that uses HELLO messages of the AODV protocol for gateway discovery. Since HELLO messages are sent anyway by every ad-hoc mobile node including every

gateway node, the information for routing via an Internet gateway is embedded into HELLO messages.

In [30] a hybrid Internet gateway discovery protocol is presented. This hybrid protocol uses periodic advertisements to announce the presence of an Internet gateway to mobile ad-hoc nodes. These advertisements have a limited TTL value to reduce overhead caused by flooding the ad-hoc cluster. Ad-hoc nodes multiple hops away from the Internet gateway will not receive advertisements due to this limited flooding range (TTL set to far less than the network diameter). For the hybrid gateway discovery algorithm if a distant node gets no gateway information by advertisements it will solicit reactively for a gateway by broadcasting solicitations. In opposition to the HELLO algorithm [30] uses solicitations for distant nodes that cause protocol overhead. Solicitations are not needed with the HELLO algorithm. Additionally, the HELLO algorithm does not use any kind of flooding advertisement messages.

Standard HELLO messages of the AODV protocol are derived from route reply messages (RREP). Thus, a HELLO message receiving node learns about the HELLO sending node and creates an entry in its routing table (list of neighbours). Beside any standard mobile ad-hoc node the gateway nodes send HELLO messages, too. The difference between HELLO messages sent by standard nodes and gateway nodes is that in the HELLO algorithm the gateway nodes set a special flag in their HELLO messages to 1 (true). To keep apart standard HELLO messages and HELLO messages sent by Internet gateways this new flag was introduced into the header of HELLO message. If this flag is set every receiving node will know that the modified HELLO message was sent by an Internet gateway otherwise, it will know that the HELLO message was sent by a standard mobile ad-hoc node. According to [34] this flag was called the I-flag and thus, the modified HELLO messages are called HELLO_I messages. Since the HELLO messages were derived from RREPs there are unused fields in the header that can be used to carry information needed for discovering Internet gateways. To form a standard HELLO message from a RREP the `Destination IP address` field in the header is set to the HELLO message sending node's address. Additionally, to create a HELLO_I message the I-flag is set in the `Reserved` area of the header and the gateway address is included into the `Originator IP address` area of the header. This last

field is originally used by the IP-address of the initiator of a RREQ where a RREP is related to. In standard HELLO messages this field is set to the broadcast address. It is important for all nodes to have an information if the received HELLO_I message is the one with the latest information about an Internet gateway and thus the sequence number of the Internet gateway is written into the `Destination Sequence Number` field. The HELLO_I message format is depicted in section 4.5.

In the following, the sum of data embedded in HELLO_I messages about an Internet gateway is called *Internet gateway information*. The Internet gateway information consists of the gateway's IP address, the hop count to the gateway, the sequence number of the gateway, and the next hop node of the route pointing to the gateway which is the HELLO_I sending node.

If a normal node in the vicinity of an Internet gateway receives such a HELLO_I message it learns two things from it. Firstly, it creates an entry in its routing table pointing to the originating node (the gateway) and secondly, it creates a default route pointing to that gateway. An AODV routing table that contains entries for routing with Internet gateways is depicted in section 3.4 in Table 3.1 on page 60. A mobile node that does not receive HELLO_I messages from an Internet gateway due to the one hop limit of HELLO(_I) messages is not provided with gateway information, i.e. is not able to set default and gateway routes for Internet connectivity via an Internet gateway.

In Figure 4.1 an Internet gateway (GW) among a number of mobile ad-hoc network nodes (small solid circles) is depicted. The circle around the GW has a radius of r that stands for the transmission range of the underlying MAC protocol. Nodes that are located in the vicinity of the GW (within the circle) have information about the GW by receiving HELLO_I messages directly from the GW and are printed in blue. Other nodes, outside the circle, have to receive information about Internet connectivity provided by the GW from nodes within the circle and are printed in green.

4.3 Advanced Functionality of the HELLO Algorithm

In the described first approach to the algorithm, distant nodes (more than one hop away from the gateway) do not receive HELLO_I messages directly from the gateway. Since

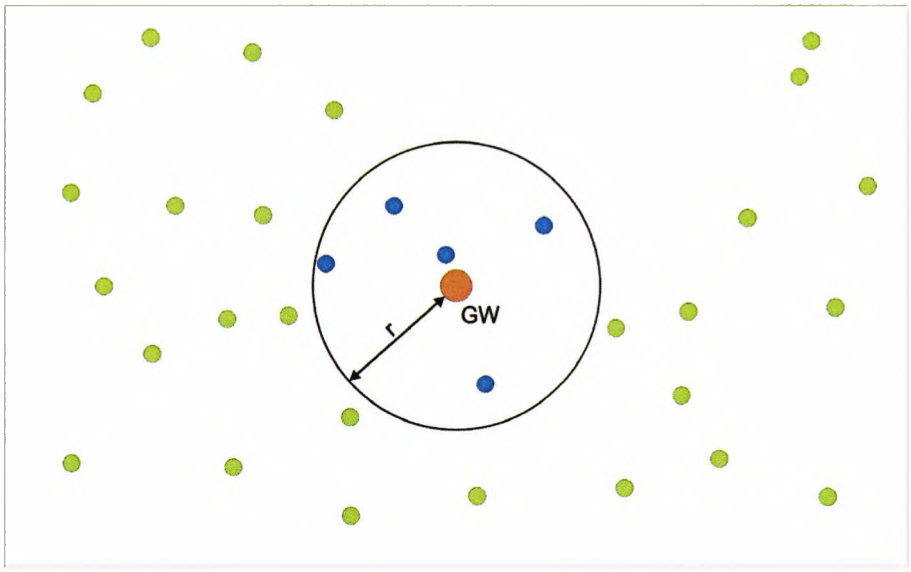


Figure 4.1: An Internet gateway among ad-hoc nodes

all nodes send periodic HELLO messages to their neighbours one node can include the information about a known, valid and reachable gateway into its own HELLO messages and broadcast them again with a limited TTL of one to its neighbours. If a distant node from the gateway receives this kind of HELLO_I message it learns its neighbour node (the origin of the HELLO_I packet), the address of the Internet gateway and the route to the Internet gateway (via the neighbour node where the HELLO_I was received from). The distance to the gateway is indicated by the `Hop Count` field in the HELLO header and its metric is the hop count. The HELLO or HELLO_I messages are not forwarded since they have a TTL equal to one but the information of the presence (i.e. address, sequence number, hop count, ...) of an Internet gateway is included into the next scheduled HELLO_I message of a specific node. Note, HELLO and HELLO_I messages are send periodically without demand at a rate of *HELLO_INTERVALL* (standard: one second). Thus, when a node receives gateway information, it will not immediately send a HELLO_I message but the node will wait until the time out of its HELLO interval and include the received gateway information into its next scheduled HELLO message that then is evolved to a HELLO_I message.

This approach for discovering gateways within an ad-hoc network is to be classified as a proactive algorithm because, mobile nodes get Internet gateway information

without demand. In opposition to the classical proactive algorithm (gateway advertisement based) the HELLO algorithm utilises no periodical flooding of the MANET cluster with advertisements and no solicitation broadcasts like the reactive (solicitation based) algorithm. Thus, no additional routing overhead for gateway discovery burdens the limited bandwidth of wireless network resources.

If a gateway is switched on it takes some time until every node in the whole ad-hoc cluster will have received gateway information at least once since a node does not forward the HELLO_I information immediately after receiving it but includes the information into its next scheduled HELLO message and this may take almost the complete interval time of one second.

If a node is X hops away from the gateway, the node has to wait a time $t_{discovery}$ according to equation 4.1 until it receives information about the gateway since the ad-hoc nodes are not synchronised in sending HELLO messages (cp. to Figure 4.1). Assumed that before the information is being forwarded to a next node in the ad-hoc network the mean time the information is delayed by one node equals half the interval time.

$$t_{discovery} = \frac{X \cdot \text{HELLO\_INTERVAL}}{2} \tag{4.1}$$

The gateway's address and the route to the gateway are permanently refreshed by the gateway as the gateway is permanently sending HELLO_I messages to its direct neighbours with increasing sequence numbers. As a result, if a gateway is switched on just one time this initial disadvantage of slow information dispersion becomes insignificant because distant nodes are aware of the Internet gateway but with lower sequence number.

4.4 Multiple Internet Gateways and Handover

In general, in ad-hoc networks with Internet connectivity provided by gateways a specific mobile node (MN) may select between multiple gateways if multiple gateways are attached to the ad-hoc cluster. The advertisement based discovery algorithm is able to provide information about multiple gateways because every gateway floods the com-

plete ad-hoc cluster periodically. A receiving MN then can set multiple routes i.e. one route for every gateway, and one default route.

The solicitation based algorithm is able to discover more than one gateway, too. If a MN demands Internet connectivity and broadcasts solicitation messages throughout the whole ad-hoc cluster multiple gateway may answer with a RREP_I message. As a result, the MN has one gateway entry in its routing table for every received solicitation answer by each gateway and then can decide to which gateway to connect to and set a default route to that chosen gateway.

HELLO messages are derived from RREP messages and the RREP message header format is only designed for one single replying node. If the HELLO algorithm should support the correct discovery of multiple Internet gateways in an ad-hoc network the header format of HELLO_I messages must contain more than one set of Internet gateway information.

In Figure 4.2 two Internet gateways GW1 and GW2 are located among a number of mobile ad-hoc nodes. Nodes that receive gateway information directly from the gateways are printed in blue, other nodes in green. One intermediate node IN1 is located in the vicinity of GW1 and GW2 and therefore receives information from both gateways. IN1 is then aware of both gateways, i.e. IN1 has got valid routing table entries with a hop count of 1 to both gateways and has to include this information about both gateways into its next scheduled HELLO_I message. The intermediate node IN2 is located within the radio range of GW1 and therefore has information about GW1 at a distance of 1 hop. In the next step, after all blue nodes have sent their scheduled HELLO_I message, IN2 receives information about GW2 from IN1 with a hop count of 2 and is then aware of both gateways, too. Below IN1 another intermediate node IN3 is depicted. IN3 receives no direct gateway information since it is out of range of both gateways but it receives information of both gateways indirectly from IN1 and IN2 with a hop count of 2. The intermediate node IN4 is informed by IN2 about GW1 and knows a route to GW1 with a hop count of 2. Later again after the next scheduled HELLO_I message of IN2 the node IN4 is aware of both gateways. This is since the spreading of Internet gateway information is based on the HELLO sending time-out (HELLO cycle) of ad-hoc nodes. The dispersion of gateway information can be observed in Figure 4.3.

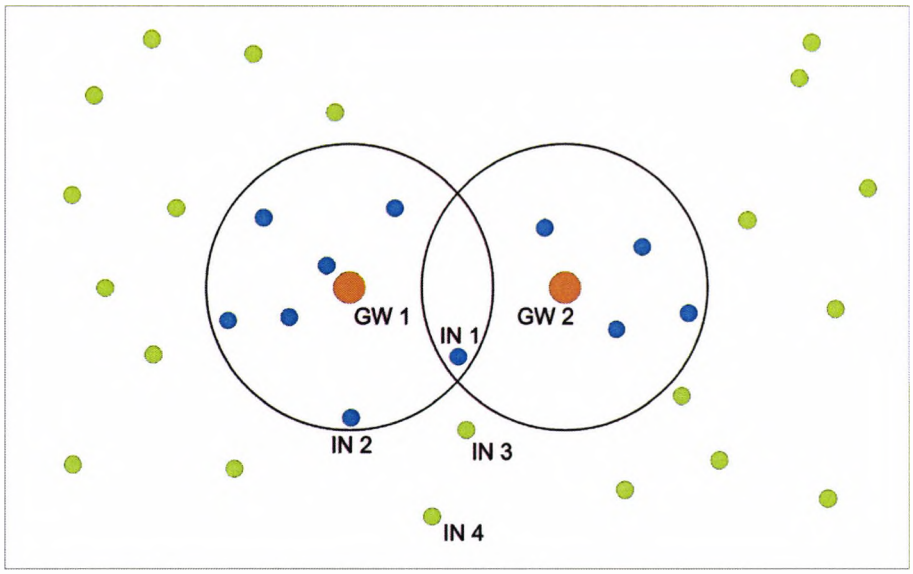


Figure 4.2: Two Internet gateways in one ad-hoc cluster

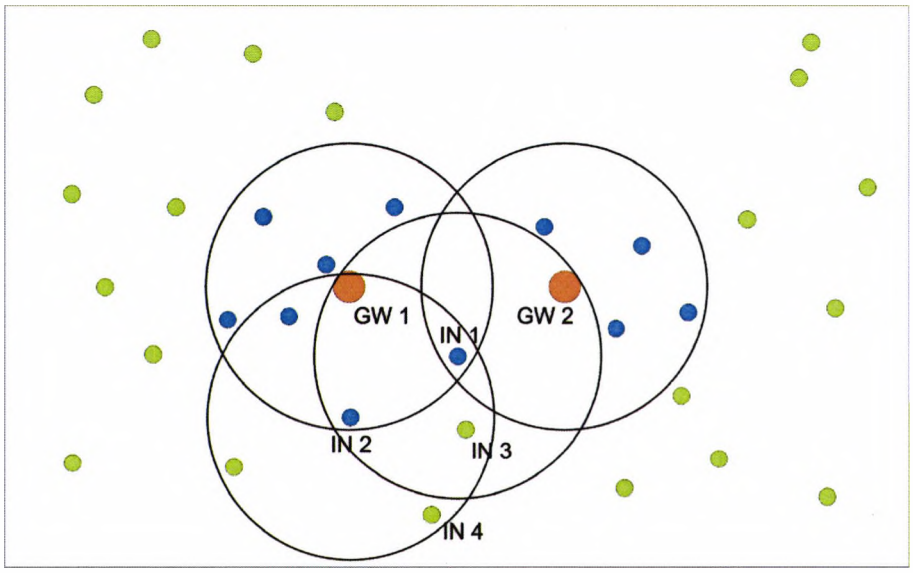


Figure 4.3: Gateway information for 2-hop neighbours

However, the setting of the default route to an Internet gateway is possible after the MN has decided to which of the discovered Internet gateways it wants to connect to. The metric for this decision in the standard HELLO algorithm implementation is the hop count.

It is very necessary that a node knows if it has to include gateway information in its HELLO messages or not because it will be very counterproductive if it would advertise wrong or old information. If a mobile node performs a handover between two ad-hoc networks and both ad-hoc networks do not overlap the handover performing node will advertise out-dated information for a short time before it discards the gateway information. The time until the gateway information is discarded is after three consecutive HELLO_I messages not received from neighbour nodes containing information about the old gateway. The time is called Δt_{loss} and it can be calculated using equation 4.2. Thus, the algorithm provides handover functionality since expired gateway information will never be advertised for a long time. In opposition, in overlapping networks with two or more Internet gateways the handover performing mobile node will receive gateway information from two or more Internet gateways and therefore it has to decide to which of them it wants to connect to. In such a case the MN will include information about all discovered Internet gateways into its next scheduled HELLO_I message.

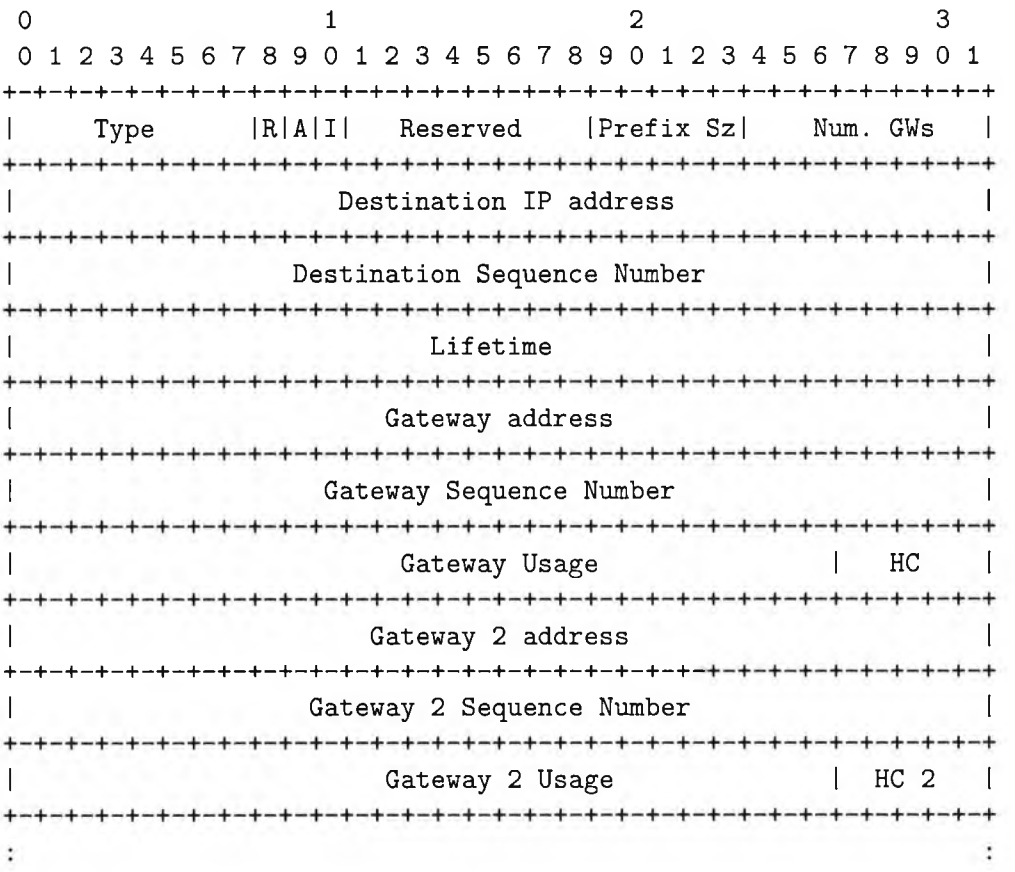
$$\Delta t_{loss} = 3 \cdot \text{interval time} \tag{4.2}$$

For a 30 second interval time Δt_{loss} computes to 90 seconds. This would lead to very long handover times with the HELLO message based gateway discovery algorithm since a node would need a long time to detect that is has lost connectivity to an Internet gateway before it decides to connect to another gateway. The interval time as an algorithm parameter is investigated in [2, 3].

In the standard implementation of the HELLO algorithm (and the advertisement based and solicitation based algorithms) a node decides for the closest Internet gateway by using the hop count information in the received HELLO_I message (advertisement or solicitation, respectively).

4.5 Header Formats of the HELLO Algorithm

The HELLO algorithm uses modified HELLO messages for Internet gateway information distribution within a MANET. The header of such a modified HELLO message is depicted below whereas the non modified message format can be found in [20]. The `Destination IP address` field and the `Destination Sequence Number` field contain the address and the sequence number of the node that generated this HELLO_I message. The Internet gateway(s) information consists of the address of the Internet gateways and the hop count to the gateway as well as the gateway's sequence number and the amount of forwarded traffic for the Load Switching feature. They are given in the `Gateway address`, `Gateway Sequence Number`, `Gateway Usage`, and the `HC` field. Multiple sets of gateway information can be included into the HELLO_I message. The number of included Internet gateways' information is given in the `Num. GWs` field. The I-flag is set in the originally `Reserved` field of the HELLO header.



4.6 Conclusion

The thesis presents a new algorithm for Internet gateway discovery developed by the author. The new discovery algorithm that uses HELLO messages is different from the classic advertisement based and solicitation based algorithms. The main difference is that the HELLO algorithm utilises no ad-hoc network flooding. Since in AODV (and other ad-hoc routing protocols) HELLO messages are used for neighbourhood management anyway, no additional overhead is caused when using modified HELLO messages for Internet gateway discovery. According to the "I"-flag introduced for advertisements and solicitations (RREQ_I) the modified HELLO message for gateway discovery is called HELLO_I message. Thus, the format of the HELLO_I messages is derived from standard HELLO messages and unused fields are utilised for the Internet gateway discovery functionality of HELLO messages. Additionally, the HELLO(_I) message format is to be expanded in order to allow the discovery of multiple Internet gateway simultaneously.

The sum of data needed for connecting to an Internet gateway is called Internet gateway information. Internet gateway information includes knowledge of the gateway's address, the hop count to the gateway, its sequence number, and the next hop entry of the route pointing to that Internet gateway. In opposition to the advertisement based Internet gateway discovery algorithm the HELLO message based algorithm provides information about multiple gateways simultaneously, like the solicitation based algorithm almost does. But the HELLO algorithm has the advantages of proactive algorithms in general, which allow a fast rediscovery of lost gateways.

Another pro for the HELLO algorithm is that it does not depend on link-layer feedback and therefore it is portable to MAC-protocols that do not support information about a link's characteristic to the routing layer.

The AODV ad-hoc routing algorithm's decisions are based on the hop count to a destination node (newer routes indicated by higher sequence numbers are always preferred). Since the advertisement based, the solicitation based, and the HELLO message based algorithms are derived from AODV they decide for gateway routes after the hop count, too. This hop count based approach does not decide for Internet gateways with respect to the gateway's utilisation, i.e. the Internet gateway's network traffic load. An

additional extension to all three algorithms is presented in this thesis. The extension is to make decisions not only based on the hop count distance to a gateway node. This extension therefore improves the quality of service characteristics of Internet gateway discovery algorithms and is presented and discussed in the next chapter.

Chapter 5

Extensions to Internet Gateway Discovery Algorithms

5.1 Overview

This thesis has two main tasks. The first task is to introduce a new algorithm for Internet gateway discovery. The second task is to extend existing algorithms for Internet gateway discovery (including the new one) in order to improve their performance in terms of provided bandwidth. This chapter is about the extension of the existing algorithms.

This chapter presents two new extensions made to common Internet gateway discovery algorithms (section 3.3) as well as the newly presented algorithm based on HELLO messages (chapter 4). The first extension consists of an unrequested sending and acknowledging of control messages of a mobile node and its selected Internet gateway if that mobile node detects a change in the route to that gateway. This happens if the route to the gateway is shortened or lengthened or if the next hop entry in the routing table of the mobile node pointing to the selected Internet gateway has changed. The extension allows the protocol to update the route from a specific mobile node to the Internet gateway and, additionally, the reverse route, i.e. the route from the Internet gateway to the mobile node.

This first extension applies for proactive gateway discovery algorithms (like the advertisement and HELLO message based) since there the Internet gateway provides routing information unrequested and therefore, only the ad-hoc mobile nodes update

their routes to the Internet gateway frequently. Mobile nodes using the standard discovery algorithms do not inform the gateway about route changes. Thus, the provided bandwidth to ad-hoc mobile nodes will be decreased by unnecessary long routes from the Internet gateway to the ad-hoc mobile nodes since the routing table entries in the gateway are not updated. This first extension does not apply for the solicitation based Internet gateway discovery protocol since there the mobile nodes solicit for the replying gateway and therefore both the mobile nodes and the Internet gateway, know about the most actual route to each other. The first extension only applies for proactive gateway discovery algorithms and if a mobile node decides not to change to a new gateway but only the route to an old (already used) gateway. Thus, it does not apply for handovers.

The second extension allows traffic switching between Internet gateways by giving mobile nodes the ability to select between multiple already discovered Internet gateways. The selection is based upon a function of the hop count to the gateway and the traffic a gateway already transports for mobile nodes. This leads to the selection of the least used gateway even if it is more distant within the ad-hoc network (metered in hops). Thus the selection is based upon a trade-off between the hop distance and the Internet gateway traffic.

The second extension allows mobile nodes to increase the resulting bandwidth to the Internet via an Internet gateway and utilises the total transfer capacity of the ad-hoc cluster better. In opposition to the first extension the second extension applies for proactive and reactive gateway discovery algorithms.

This thesis introduces both extensions to increase the provided bandwidth for ad-hoc nodes and the quality of the multihop routes to Internet gateways in an ad-hoc network in terms of the provided bandwidth which is one of the objectives of the thesis (besides the HELLO message based discovery algorithm). In this thesis the increase of the provided bandwidth is handled as a quality of service (QoS) extension to the discussed Internet gateway discovery algorithms.

The chapter firstly presents an overview of resources in ad-hoc networks. Then established quality of service enhancements to known protocols are discussed. This is followed by the presentation of the first extension that consists of the sending of gratuitous route reply messages (GRREP) from a mobile node (MN) to an Internet

gateway (GW) and how they are being acknowledged by GRREP-ACK messages is presented. Then, the chapter illustrates the second extension that allows mobile nodes to switch to alternative Internet gateways. Finally, the chapter ends with a conclusion.

5.2 Resource Metrics of Ad-Hoc Networks

This section discusses network resource metrics of a MANET. Network resource metrics are the bandwidth provided to a specific mobile node, the packet loss of a (multihop) connection between network nodes of the MANET, or the reliability of a link between two network nodes. As mentioned above these are called quality of service (QoS) constraints. The term quality of service is defined and the discussion about the cooperation of wireless networks and quality of service is given in section 2.6 on page 40.

In ad-hoc mobile networks the provisioning of network resources is much more complicated compared to a wired network. Since all attending mobile nodes may move around randomly or may be switched on or off there are no static routes possible but routes may change frequently. Therefore, a mobile ad-hoc network can never guarantee connectivity from one specific ad-hoc node to another (except users would be prevented to walk around and to switch their devices off). Additionally, there is no possibility for a long-term prediction of bandwidth resources or packet delays because intermediate nodes along a multihop ad-hoc route may start generating data traffic at any time. Therefore, it is not possible to achieve absolute quality of service guarantees like minimum bandwidth in mobile ad-hoc networks whereas relative guarantees, e.g. priority between different traffic types, are still possible.

In this thesis the term quality of service refers to the increase of provided bandwidth. The two presented extensions do not reserve bandwidth but increase provided bandwidths. The extensions concentrate on the gateway route shortening and the selection of an Internet gateway as a function of traffic load metric and hop count. Finding the "best" route within an ad-hoc cluster is not the task of the thesis. There exist other concepts and approaches for ad-hoc routing protocols to find the "best" route from one node to another or to an Internet gateway [53, 56, 57, 58, 59]. The citations are discussed in section 2.6.4.

As mentioned above there are several approaches to find routes within ad-hoc net-

works using different metrics. The commonly used metric is the hop count of a multihop route. Other metrics are the reliability of such a route, the maximum bandwidth, or the minimum packet delay an ad-hoc multihop route can provide. To achieve the goal of increasing the provided bandwidth to mobile nodes the presented discovery algorithms were extended. These extensions are presented and discussed next.

5.3 Gratuitous Route Reply

If in an ad-hoc network a proactive Internet gateway discovery protocol is used the mobile nodes are supplied with routing information about available Internet gateways permanently. The information is sent by the Internet gateway(s) itself. This leads to the fact that newer (higher sequence number) or better (shorter) routes from a specific mobile node to a gateway are only known by the mobile node and not by the gateway since there is no reply mechanism integrated into the standard proactive discovery protocols. Thus unsymmetric routes are being established between the mobile node and the Internet gateway and back. In contrast to the proactive gateway discovery algorithms the solicitation based (reactive) Internet gateway discovery algorithm consists of a request-reply mechanism that cares for valid routes in both directions, i.e. from the mobile node to the gateway and from the gateway to the mobile node. Such a reply mechanism is now integrated into the proactive Internet gateway discovery protocols.

To achieve the goal of informing the Internet gateway about a route change the mobile node generates a reply message destined to the Internet gateway. In accordance with the standard request-reply mechanism of reactive ad-hoc routing protocols and the fact that this reply is sent without request it is called a gratuitous route reply message (GRREP).

The gratuitous route reply (GRREP) message represents the reply of the reactive Internet gateway discovery algorithm and implements this into proactive Internet gateway discovery algorithms. It is sent to the selected Internet gateway if an ad-hoc mobile node is supplied with newer (sequence number) information about the multihop ad-hoc route to the selected Internet gateway and the newer route is shorter, longer, or has a different next hop entry pointing to the Internet gateway. Thus, the GRREP is sent if

the route to the Internet gateway has changed.

The main difference between the standard reactive and the GRREP extended proactive gateway discovery algorithms is that the sending of the GRREP applies after the detection of a newer or better (shorter) route to a gateway whereas in the reactive discovery algorithm the gateway route is established while the Internet gateway is being detected.

If another gateway is evaluated as "better" the MN will perform a standard handover and indicate this handover by sending MobileIP binding updates (BU). The BUs ensure a valid route from the MN to the new gateway and therefore no additional route reply message is necessary since the forwarding of a BU message and the forwarding of the acknowledgment according to the BU (BACK) updates the routing table entries in all involved nodes along the route from the Internet gateway to a mobile node.

Note that in AODV [20] a gratuitous RREP is sent if an intermediate node of an ad-hoc mobile network receives a RREQ message and answers with a RREP message instead of the RREQ destination node. In such a case the intermediate node must sent a gratuitous RREP to the destination node of the RREQ to inform the destination node about the RREP sent to the originating node. This is to establish routes in the originating as well as in the destination node for e.g. reverse channel traffic types like TCP [62]. The GRREP message in this thesis is to establish symmetric routes between a mobile ad-hoc network node and an Internet gateway. Additionally, in this thesis the GRREP messages are being acknowledged by the Internet gateway. In [20] the gratuitous RREP messages are not acknowledged.

In Figure 5.1 an example to the gratuitous route reply extension is depicted. Figure 5.1(a) shows an initial situation. A mobile node (MN) is connected via a multihop route to an Internet gateway (GW). If the MN is moving toward the gateway (Figure 5.1(b)) the MN receives information about the route to the Internet gateway proactively and knows a shorter route to the gateway (dotted arrow). The gateway is not aware of that shortened route and still uses the old (longer) route indicated by solid arrows. In that situation the MN will send a GRREP message along its new discovered path to the Internet gateway. That forces the gateway and the intermediate nodes along the multihop path to update their routes pointing to the MN. As the result, the gateway and

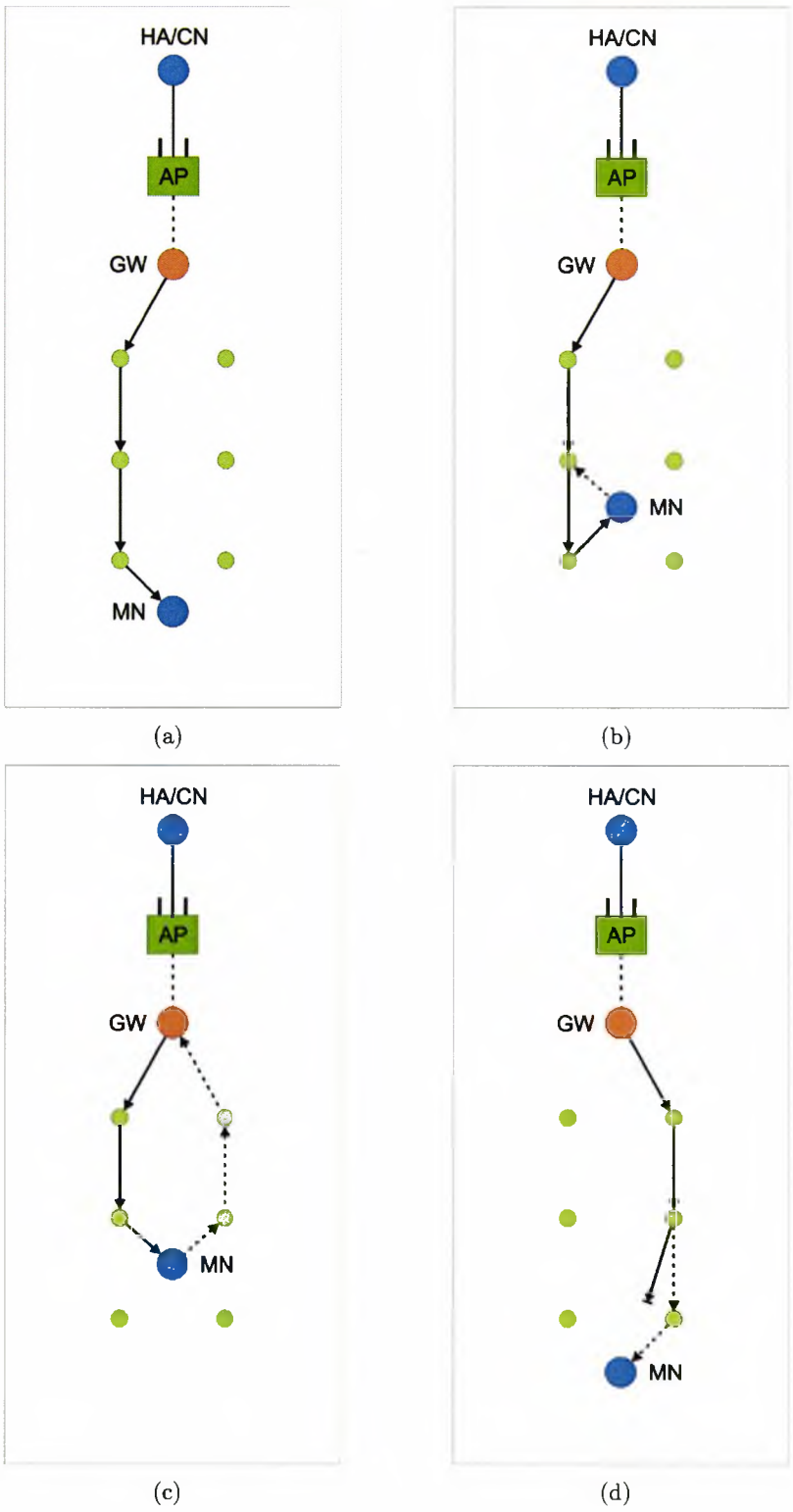


Figure 5.1: Gratuitous route reply example

the intermediate nodes use the path as shown in Figure 5.1(c) (solid arrows). Then, the MN receives an advertisement or a HELLO_I message from the right column of ad-hoc nodes with a higher sequence number and therefore updates its route to the gateway via the right column of intermediate nodes. In order to have symmetric routes from the GW to the MN and back the MN sends a GRREP to inform all involved nodes. Thus, the MN does not select another gateway but only the route to the same gateway that the MN already uses. In Figure 5.1(d) the MN is moving to a more distant position from the gateway and the intermediate node will lose the connectivity to the MN (indicated by an interrupted solid arrow). But the MN will permanently receive information about the Internet gateway by advertisements or HELLO_I messages respectively and will know a valid route to the gateway. To inform the gateway and all intermediate nodes along the route the MN sends a GRREP message to the gateway. Thus, the gateway will have a valid route to the MN.

Without the GRREP extension to the discovery algorithms the updating of the routing table in the Internet gateway node and the intermediate nodes would be initiated by the forwarding of binding update (BU) messages of the MobileIP protocol. But the ad-hoc nodes then would have to wait for the scheduled sending of the BU messages and this would lead to delayed information about out-dated routes. Simulations show the benefit of the extension in section 7.5.

The motion of mobile ad-hoc network nodes and network traffic may cause a loss of GGREP messages from the MN to the selected gateway. Thus, the extension uses gratuitous route reply acknowledgment (GGREP-ACK) messages to indicate that the selected gateway has received the GGREP and updated its routing table entries pointing to the MN successfully. If the MN does not receive this GGREP-ACK message within a time-out (one second) it sends another GGREP message to the selected gateway until it receives a GGREP-ACK message. This is to ensure that all, the MN, the gateway node, and the involved intermediate nodes have valid and symmetric routes to each other.

5.4 Load Switching between Internet Gateways

Internet gateways forward traffic from an ad-hoc cluster to the Internet and from the Internet to nodes within the ad-hoc cluster. Thus, the Internet gateway represents a zone of network congestion if ad-hoc nodes use Internet connectivity simultaneously. This is especially if ad-hoc nodes need much bandwidth e.g. downloading files from the Internet. To avoid congested Internet gateways the second extension allows mobile nodes to choose between multiple detected Internet gateways not only after the hop count to the gateways but additionally after the traffic through the gateways. Thus the second extension is called the Load Switching extension. In this terms the "best" Internet gateway is the one that provides the most bandwidth for e.g. a file download.

The Load Switching extension applies for all three investigated algorithms, i.e. the advertisement, the solicitation, and the HELLO message based.

The Load Switching extension consists of three main steps. Firstly, to find the least used Internet gateway, all Internet gateways have to calculate a value that stands for the utilization of that gateway and therefore every Internet gateway has an utilization value that varies as the traffic through the gateway increases or decreases. This utilization value, or usage, is an averaged value over a period of one second and it is the first part of a metric after the mobile ad-hoc nodes decide which gateway to connect to. The second part of the metric for the decision is the hop count to the gateway which is different for every mobile node. The hop count as a part of the metric is because more distant gateways provide less bandwidth in general. See Appendix A for the relation of hop count and bandwidth.

Secondly, the usage value is to be sent to the mobile nodes of an ad-hoc network to allow them to decide which gateway to connect to. The ability of spreading the gateway usage information is to be included into the gateway discovery protocols. Therefore, one task is to enhance the Internet gateway discovery algorithms and to extend the message headers of the protocols by a 27 bit usage field. This usage field is just added to the end of the message header and it contains the usage information of a specific Internet gateway.

The advertisement based discovery algorithm transports the usage information of a specific gateway by embedding the usage information into the periodically broadcast

advertisement messages. A mobile ad-hoc node therefore receives the usage values of all Internet gateways available. This applies only if no range limitations of advertisements are integrated into the discovery protocol like in [26].

In the solicitation based gateway discovery algorithm the mobile nodes solicit for gateway presence. If a gateway receives a solicitation it answers by sending a reply (RREP_I) message back to the originator of the solicitation and includes the gateway usage value into the reply. Thus, mobile nodes get information about the presence of Internet gateways and additionally the usage value of each discovered gateway. Since the replies (RREP_I) are not received simultaneously by the originating mobile node the mobile node initially selects the Internet gateway of whom the first reply was received from.

If multiple replies will be received by the mobile node the mobile node may change its Internet gateway very frequently with every receiving of a RREP_I with a better metric. This will cause handovers and additional protocol overhead. To avoid the problem of re-selecting less used Internet gateways very frequently (RREP_I messages are delayed if multihop routes are longer, i.e. consist of more intermediate hops) the Internet gateway delays the reply message by a time equal to a factor of its usage value. As a result, the solicitation answer of the least used gateway is expected to be received first by the requesting mobile node. The time a gateway schedules the solicitation answer is computed as described in equation 5.1. It is a function of the usage value of the gateway U_{GW}. The denominator is set to $4 \cdot 10^6$ in order to avoid frequent re-selection of the mobile nodes by delaying control messages. This factor is an empirical value found through simulation tests.

$$delay(U_{GW})[ms] = \frac{U_{\mathrm{GW}}}{4 \cdot 10^6} \tag{5.1}$$

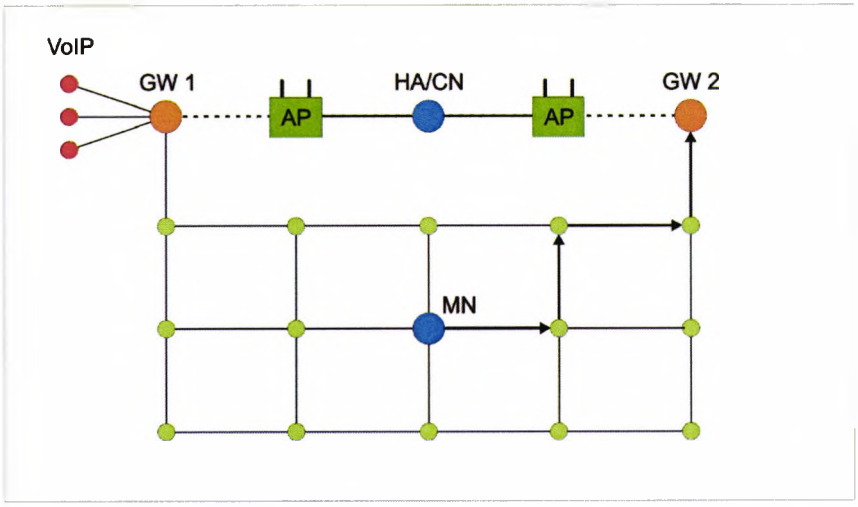
The newly developed HELLO message based Internet gateway discovery protocol was designed to reduce overhead to the ad-hoc network and to allow mobile nodes a fast Internet access without pre-connection delay which is the advantage of the proactive gateway discovery algorithms. Since the new algorithm is based on AODV HELLO messages the HELLO and HELLO_I message format has changed, respectively. First, HELLO_I message must contain the ad-hoc routable address of the Internet gateway as

well as the hop count to that gateway and the sequence number of the gateway. This is now extended by the usage value of the Internet gateway. The discussed information are being handed from node to node in the ad-hoc cluster with every cycle of the sending HELLO_I messages. If there are multiple gateways available in an ad-hoc network HELLO_I messages must contain information about all available gateways. Thus, the size of a HELLO_I message increases with every additional gateway. If a node receives more than one gateway information by a single HELLO_I message it includes all collected information of all known gateways into it next scheduled HELLO_I message. Thus, every ad-hoc node in the network will know about every available gateway.

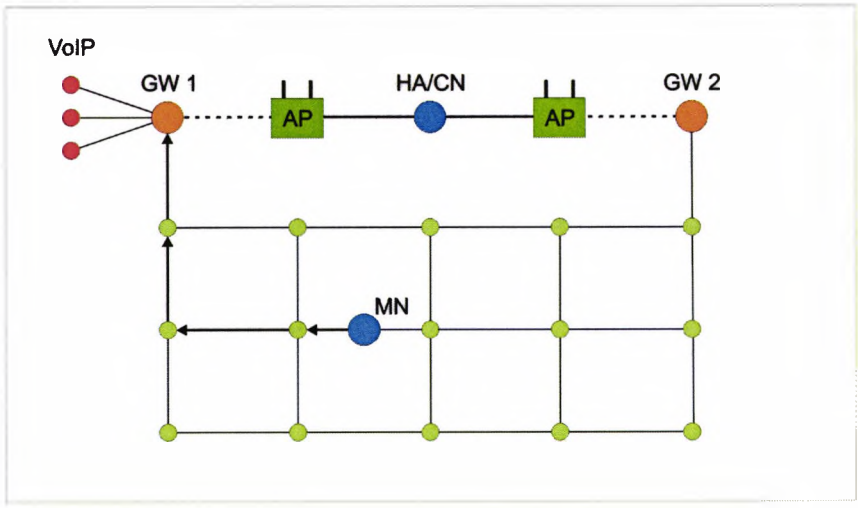
Thirdly, after the mobile ad-hoc nodes have received the gateway usage information using the advertisement, the solicitation, or the HELLO message based gateway discovery algorithm they have to make a decision for which Internet gateway they connect to. As mentioned above the decision is based on the received usage values of the Internet gateways and the hop count to the Internet gateways.

The Load Switching extension is now being explained by an example. Figure 5.2(a) gives an example of the Load Switching extension in a symmetric environment, i.e. the routes to both Internet gateways, GW1 and GW2 have the same hop count. One possible path (route) from the MN to the selected Internet gateway is depicted by solid arrows. In the case there are no active VoIP connections through GW1 the mobile node (MN) connects randomly to one gateway. If GW1 is forwarding Internet traffic (depicted as VoIP connections) the MN will select GW2 for Internet connectivity since GW2 is not burdened with any traffic. In Figure 5.2(b) the MN is one hop closer to GW1 compared to GW2 and then has to decide which gateway is expected to provide the most bandwidth. In Figure 5.2(b) the MN decides for GW1 because, the traffic load of GW1 does stress the Internet connection of the MN less than a gateway that is one hop more distant until the traffic load in GW1 exceeds a certain threshold.

A mobile node MN may select an already known gateway even if the route to this already known gateway is longer compared to a closer gateway because, the MN assumes more provided bandwidth. This assumption is based upon the hop count to the selected gateway and the usage value the gateway announces into the ad-hoc cluster. Thus, the MN may switch the Internet gateway while downloading a file from the Internet and



(a)



(b)

Figure 5.2: Load switching example

therefore this protocol extension is called Load Switching between Internet gateways. The function after a MN decides for a specific Internet gateway is given is equation 5.2. The factor F stands for a usage value at which the mobile node decides for a more distant but unused gateway before selecting a closer gateway with traffic. The usage value of an Internet gateway is U_{GW} and the hop count to the Internet gateway is called HC.

$$metric(HC_{GW}, U_{GW}) = F \cdot (HC_{GW} - 1) + U_{GW} \tag{5.2}$$

A lower metric indicates an Internet gateway where the selecting mobile ad-hoc node is expecting a higher provided bandwidth for the file download. The factor F is adjusted that a mobile node will select the closer gateway until the number of simultaneous VoIP connections is less than $320\frac{\text{kbit}}{\text{s}}$ which equals 2 simultaneous full-duplex VoIP connections. If the gateway traffic is increased the mobile node will select the more distant gateway with $U_{GW2} = 0$. If the feature of Load Switching extension is not enabled ad-hoc network nodes will decide after the route's hop count to the Internet gateway.

5.5 Conclusion

This chapter discusses network resources in MANETS and emphasises extensions made to the three investigated Internet gateway discovery algorithms that give the algorithms quality of service features. Due to the mobility of ad-hoc mobile nodes and the layer 2 independent design of the discovery algorithms it is not possible to reserve quality of service constraints in ad-hoc networks. For this thesis quality of service is defined as the improvement of ad-hoc mobile available bandwidth via Internet gateways to mobile nodes. This is achieved by introducing two extensions. It is not the aim of the extensions to find quality of service routes to other nodes and Internet gateways that fit restricted quality of service constraints. Quality of service routing within ad-hoc nodes is e.g. possible with the AQOR [57] algorithm.

There are two extensions presented. The first extension gives nodes the ability to update routes to the Internet gateway if a shorter or longer route to an already used Internet gateway has been detected or the next hop entry pointing to the Internet gateway in the routing table of a node has changed. All nodes along the route to the Internet gateway are informed of that changing as the detecting mobile node is sending a gratuitous route reply (GRREP) message. To ensure that all nodes have received that GRREP message (including the gateway node) the gateway replies by sending an acknowledge message (GRREP-ACK). As a result, all nodes of a multihop route to a gateway and the gateway itself have updated their routes for Internet connectivity of the mobile nodes. This extension applies only for the advertisement and the HELLO message based gateway discovery algorithms. Note that mobile nodes only

send GRREP messages if they decide to keep connected to a specific gateway and not to perform a handover.

The second extension gives mobile nodes the ability to switch between multiple Internet gateways after another metric and not the hop count to that gateway as it is in the standard implementation of the Internet gateway discovery algorithms. Network traffic through an Internet gateway burdens the gateway and thus traffic from a specific node through the Internet gateway may suffer. With this Load Switching extension a mobile node now can select an alternative gateway that is less burdened with traffic from other nodes. The main functionality principle of the Load Switching extension is that Internet gateways monitor the traffic they forward within the ad-hoc cluster and from the ad-hoc cluster to the Internet. This leads to a parameter that describes the traffic burden of a gateway and it is called the *usage value* of a specific gateway (U_{GW}). The usage value is given to the mobile nodes of the ad-hoc cluster by embedding it into advertisements, RREP_Is, and HELLO_I messages of the gateway discovery algorithms. After receiving usage information about multiple Internet gateways each mobile node decides which gateway to connect to. Thereby, the mobile nodes have to select between gateways that are more distant (hop count HC_{GW}) and gateways that are less stressed with additional traffic (usage value). Less burdened gateways may provide smaller bandwidths if they are more distant metered in hops. Thus the Load Switching algorithm has to find a trade-off between a less burdened but more distant Internet gateway and a closer but traffic burdened Internet gateway. This trade-off is the *switching point* that is represented by the factor F.

The next chapter describes and illustrates the implementation of the gateway discovery algorithms and depicts the implementation of the extensions to the Internet gateway discovery algorithms.

Chapter 6

Implementations in NS-2

6.1 Overview

The Network Simulator 2 (NS-2) is a discrete event driven simulation software for network research. With the aid of NS-2, protocols of different network layers (section 2.2.3) over wired and wireless networks can be simulated. This chapter describes the new implementations into the Network Simulator 2 [4] that allow simulations with Internet gateways and the modified AODV protocol for Internet gateway discovery and handovers. First, NS-2 is discussed in more detail. Second, the new Internet gateway node is described and its implementation is depicted. Third, the implemented Internet gateway discovery algorithms and the extensions for bandwidth improvement (chapter 5) are presented. Therefore a number of operating plans was created to clarify the algorithm's functionality. Furthermore, this chapter discusses the reliability of simulation results before it is concluded.

6.2 The Network Simulator NS-2

6.2.1 Functionality Principle

The network simulator NS-2 is free software under the GNU Public Licence (GPL) [47] written in two programming languages. First in C++ and second in an object oriented version of the Tool Command Language (TCL). Both the extended TCL and the source code of NS-2 can be downloaded from [4]. With the aid of TCL scripts, simulation scenarios can be created and node configurations are made. On the command line of a

Linux system the simulator is started with `./ns <scenario-file.tcl>`. NS-2 logs simulation events in so called *trace-files* which contain simulation results i.e. information about send and received data and control messages at the MAC layer, the routing layer, and the application layer.

The fast computation speed of C++ and the way to combine C++ objects to new types of network nodes using the object oriented TCL is the primary feature of NS-2. The general functionality of NS-2 is given in [4] and the principle of programming the simulation tool is shown in [5]. Furthermore, in [6] a tutorial for using NS-2 can be found. In this thesis version 2.26 of NS-2 is used for simulations that can be downloaded from [4].

6.2.2 TCL Extensions

The Tool Command Language (TCL) is a common script language. oTCL as an extension to Tcl/Tk allows an object oriented programming with TCL. Additionally, TCL with classes (TclCL) is the interface to C++. TclCL is responsible for combining precompiled C++ objects to form network nodes.

6.2.3 Split-Level Programming

NS-2 uses an uncommon programming technique called split-level programming [33]. In NS-2, C++ is used to provide fast computable parts of network nodes that are glued together by TCL. This style of programming combines both fast computation of compiled programs and the flexibility of scripting languages for modifying node structures. If new protocols have to be implemented, both C++ objects and TCL scripts have to be modified.

In NS-2, wireless routing is provided by so called routing agents that are included in all wireless nodes. Routing functionality in Internet gateway nodes requires a new agent, the *AGWAgent*, which had to be developed and inserted into the structure of Internet gateway nodes. The modifications of the node structure are discussed in section 6.3.3. In opposition to the existing basestation node in NS-2 this new gateway node is mobile. Thus the Internet gateway needs two wireless interfaces for mobility. More details about the implementation of Internet gateway nodes and their functionality are

depicted in [1] and [35]. The Internet gateway node is then equipped with an adopted routing agent that allows Internet gateway discovery as described in the chapters above. The implementations of the algorithms are described in section 6.4.1 and in [36].

6.2.4 Customisation of NS-2

The NS-2 network simulator has some drawbacks that need to be corrected when working with multiple ad-hoc network nodes. The first drawback is that in the standard implementation of the NS-2's AODV routing agent all network nodes send HELLO message simultaneously. This is because all ad-hoc nodes (including the Internet gateway node) use the same AODV routing agent and therefore start sending HELLO messages almost simultaneously (with a jitter between 0 and 0.01 seconds). This is not realistic for simulating ad-hoc network nodes. In the implementation for this thesis the ad-hoc network nodes send HELLO messages with a static interval of 1 second but every node with a different offset and thus the simulation is more realistic. This is important especially for the HELLO message based Internet gateway discovery algorithm. Additionally, the sending of advertisements in the advertisement based algorithm is affected by the synchronisation of Agents. Unfortunately, it could not be ascertained if other papers and theses do pay attention to that fact.

Another drawback of the NS-2 simulation suite is the `setdest` program. The `setdest` program allows the creation of movement patterns for random moving ad-hoc mobile nodes by generating the according Tcl commands. In the standard implementation of `setdest` every network node starts with the beginning of its pause time. After this initial pause time is over all nodes start to move simultaneously to their random destination. This thesis uses a slightly modified `setdest` that generates more realistic outputs for moving mobile nodes by letting nodes move from the beginning of a simulation run even if the pause time is set equal to the total simulation run time.

Next, the implementations necessary for the thesis are presented.

6.3 Implementation of Nodes

The NS-2 simulator provides different node types. First, static nodes that are interconnected with wires. Second, there are wireless nodes with a radio interface that are

mobile and third, NS-2 provides a combination of both, a static node with a wired and a wireless interface for simulating e.g. access points.

6.3.1 Implementation of Wired Nodes

The thesis uses wired nodes for representing the Internet and the home network of moving mobile ad-hoc network nodes. Since the wired part of a network with ubiquitous connectivity for wireless mobile ad-hoc nodes is not the focus of this thesis wired network nodes are only presented briefly. Further information about wired network nodes in NS-2 can be found in [4].

A wired network node consists of a number of cascaded classifiers that decide where to route a packet to. Each classifier decides if the destination of a received packet belongs to the classifier's routing information. Thus, in a four tier hierarchy network each wired network node has four classifiers. One classifier for each tier. One example for this approach is an IPv4 network address with its 4 octets. After deciding if the packet is destined to the packet receiving node the packet is passed to a last classifier which is called the port classifier, or port demux. However, if the classifiers decide that the packet is to be sent to another wired network node the packet is directly passed to a link object. The link object is then responsible for delivering the packet to the destination node.

In this thesis network nodes with both, a wired and a wireless interface are used for connecting Internet gateways with the Internet via access points (AP) (cp. to Figure 3.1). This kind of network nodes is already implemented in NS-2. The difference to standard wired nodes is that access point nodes have one additional wireless network interface stack implemented. Since access point nodes are already implemented in NS-2 they are not subject for discussion here but used in the thesis for connecting Internet gateway nodes wirelessly to the wired infrastructure network.

6.3.2 Implementation of Wireless Nodes

Pure wireless nodes that can form an mobile ad-hoc network (MANET) are very important for the thesis. Thus pure wireless network nodes are presented and discussed next. In Figure 6.1 the structure of a gateway node is shown. Some objects of the

gateway node are illustrated in grey. These grey objects are only for gateway nodes and thus standard wireless nodes do not have these grey objects implemented. It can be observed that therefore standard wireless network nodes have only one wireless network interface stack and the `entry_`-point of the nodes is only served by one network interface. Additionally, the `defaulttarget_` of the address demux classifier and the routing port (`255`) of the port demux classifier are directly connected to the routing agent (RAgent). More information about the functionality principle and structure of wireless nodes is discussed when the Internet gateway node type is introduced next.

6.3.3 Implementation of Gateway Nodes

Principle of Gateway Nodes

In opposition to static nodes with one wired and one wireless interface a gateway node is derived from a standard mobile node but it has two wireless interfaces. Standard wireless nodes are discussed in section 6.3.2. Thus Internet gateway nodes are still mobile but they need an additional component to manage the two wireless interfaces. This component is called the `AGW-Agent` and it decides where to route packets to, either to the outer network (the Internet) or to the inner network (the ad-hoc network).

The gateway nodes of this thesis were derived from the standard mobile node class to provide mobility for gateway nodes. Additionally, NS-2 provides a base station node with one wired interface but the basestation node is not mobile because it was designed to simulate base stations or access points (AP).

The deciding object, the `AGW-Agent`, has two targets that are called `inTarget_` for packets to the inner (i.e. ad-hoc) network and `outTarget_` for packets to the structured network (i.e. the Internet). A target is an exit for data packets and usually, targets point to other objects that process the packets further. Both targets of the AGWAgent point to a routing agent that needs further modifications to allow gateway discovery. The functionality of the routing agents is to discover the route to a destination node whether the destination node is a normal node or a gateway node. Details of the implementation and modifications on a standard mobile node can be found in [1] while a short explanation is presented here.

Structure of Gateway Nodes

In Figure 6.1 the structure of the new introduced gateway node is depicted. Every circle, square or trapezoid represents an object that is written in C++. The arrows are TCL script commands that connect all C++ objects to form a complete NS-2 node. The original structure of a mobile node can be found in [4]. Additional parts of a standard mobile node's structure to form an Internet gateway node are illustrated in grey.

The data source or sink in a network node is attached to the `port demux` object. Data packets are generated by the source object and the data packets are received by the `entry_`-point. Additionally, packets from the `uptarget_`s of both network stacks are received by the `entry_`-point, too. The `entry_`-point sends all packets to the `addr demux` object(s). In Figure 6.1 only one address demux classifier is depicted representing a number of classifiers according to the set-up address hierarchy depth. From the `addr demux` object, packets destined to the local node are forwarded to the `port demux` where a decision is made if the packets are destined for the local node or if they are destined to the broadcast address (255). However, the packets are forwarded to the `AGWAgent` to decide if the packet has to be sent into the MANET or to the structured network. Both exits of the `AGWAgent` result in the `RAgents` (routing agents) where the route resolving procedures (and gateway discovery algorithms) are implemented. Unlike a gateway node, a standard wireless mobile node does not have an `AGWAgent` since it has only one wireless interface.

The network interface stacks consist of 5 objects. These objects are the `LL` object (link layer), the `IFq` object (interface queue), the `MAC` object, the `NetIF` object (network interface, and finally the `Channel` object. Data packets are given from one object of the stack to the next in order to simulate a complete wireless transmission system. The `Channel` objects interconnect wireless nodes. More information about wireless mobile ad-hoc gateways and testing simulations can be found in [35].

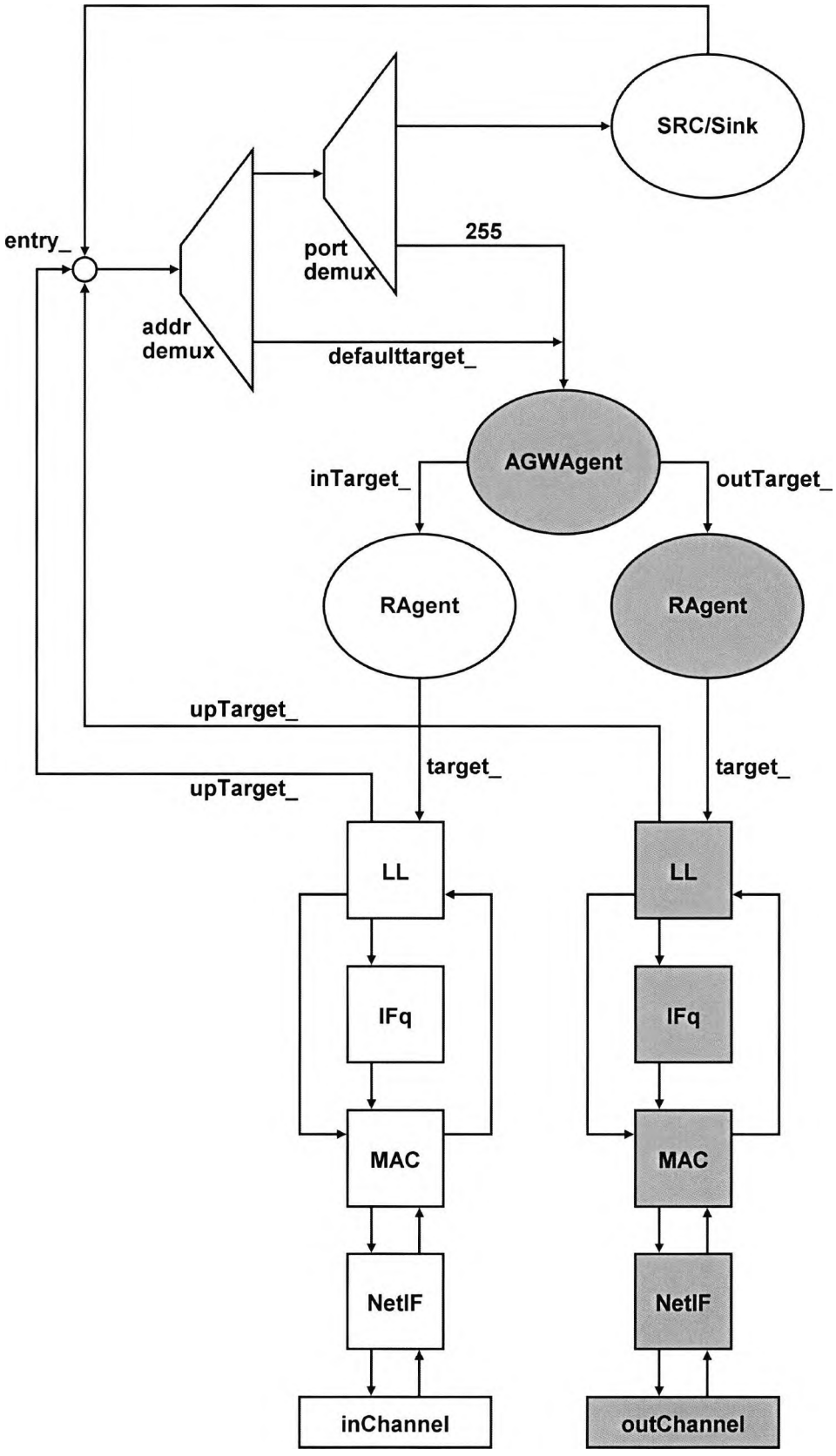


Figure 6.1: Node structure with one or two (grey) wireless interfaces

6.4 Implementations of Algorithms

6.4.1 Implementation of Gateway Discovery Algorithms

The AODV routing protocol needs to be extended to allow attending mobile ad-hoc nodes to discover gateways within mobile ad-hoc networks. This section gives an overview how these algorithms work in detail. The routines described in the following are executed for every received packet (i.e. data packets and routing packets). Therefore, every routine may be called twice or even more if one routine waits for time outs and another packet is received in the meantime. Routines of the standard AODV functionality are not illustrated. The standard AODV functionality is described in detail in [20].

Note, that every **send** instruction increases a node's sequence number to the next even number regardless of the type of routing message the node is sending. Mobile network nodes of an ad-hoc cluster receive information about available Internet gateways and decide to which gateway they connect to after a certain metric. The standard metric is the hop count. Another metric is described and discussed in section 5.4.

General Gateway Discovery Strategy

In [7] the authors propose a general search strategy for destination nodes where mobile nodes first search the destination node within the local ad-hoc cluster. Therefore, they broadcast standard RREQ messages three times with increasing TTL (5, 7, and `NETWORK_DIAMETER` ($\hat{=}$ 30) hops) and time outs. The time outs are computed as described in equation 6.1. TTL is the Time-To-Live of the route request, CNT stands for the number of route discovery (re-)tries, and the RoundTripTime is a fixed value of 0.03 seconds. This results in time outs of 0.3, 0.84, and 5 seconds since, the upper bound of route request time-outs is 5 seconds. The idea of this approach is that [7] assumes that the destination node is located within the local ad-hoc cluster. The strategy of increasing TTL and time-outs is called the *expanding ring* search strategy. In this thesis nodes always have to connect to an Internet gateway and therefore they discover Internet gateways first.

$$timeout = 2 \cdot TTL \cdot CNT \cdot RoundTripTime \tag{6.1}$$

Especially for the solicitation based gateway discovery algorithm the initial local search of [7] results in long gateway discovery times because a mobile node waits the total of all standard RREQ time outs before it broadcasts the first solicitation message. Proactive gateway discovery algorithms provide gateway information without necessity and thus a mobile node uses the default route to a correspondent node (CN) in the Internet earlier compared to the solicitation based algorithm.

In this thesis it is assumed that mobile ad-hoc network nodes always have the ambition to discover an Internet gateway for connecting to their home agent in the home network. This applies also for the solicitation based algorithm. Thus in this thesis, and in opposition to [7], ad-hoc mobile nodes using the solicitation based algorithm send out solicitation messages after they have been switched on. Note that [7] does not discuss the HELLO algorithm.

6.4.2 Initialisation

After all Tcl objects were created by Tcl simulation and initialisation scripts some of theses objects that belong to the Internet gateway discovery algorithms need to be initialised. These objects are the extended AODV routing agents (`RAgent`). They are depicted in Figure 6.1. Note that the `AGWAgent` objects are only part of gateway nodes.

The first instruction in Figure 6.2 ("calculate random offset per node") sets a random offset for periodic timers in each network node's AODV routing agent. This is very important since in the standard implementation of AODV agents all agents send out HELLO messages almost simultaneously because all use the same agent object. In the case the advertisement based discovery algorithm is selected and the nodes would not have a per node offset for timers every gateway node would flood the ad-hoc network simultaneously. Now, in this thesis in every simulation every AODV agent has its own timer offset achieved by different pseudo random generator seeds for every simulation run. This is to ensure more realistic simulations.

Every wireless network node sends HELLO messages periodically. Therefore every node will be initialised with the `sendHELLO()` (Figure 6.7) function. If the initialising

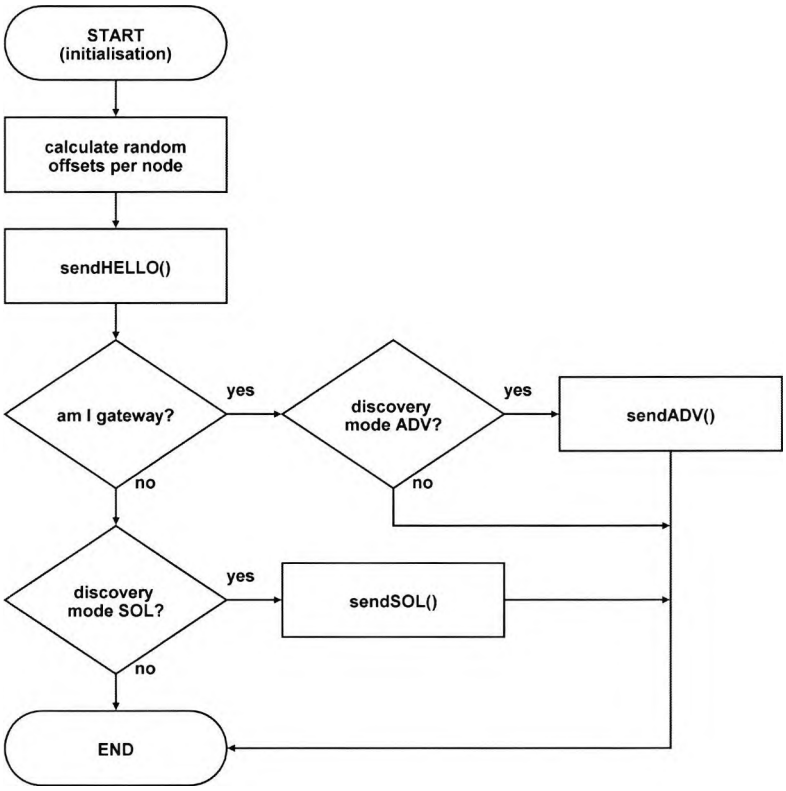


Figure 6.2: Initialisation of nodes

node is a gateway and the gateway discovery mode is set to the advertisement based algorithm the node additionally initialises the `sendADV()` (Figure 6.9) function. If the node is a standard mobile node and the gateway discovery algorithm is set to the solicitation based algorithm the node alternatively initialises the `sendSOL()` (Figure 6.11) function. In every case, including the HELLO message based gateway discovery algorithm, nodes send HELLO messages. The function `START()` ends. If the discovery algorithm is set to advertisement based standard mobile ad-hoc nodes wait for receiving advertisements. Similar, if the discovery algorithm is set to solicitation based gateway nodes wait for solicitations to answer. Every node gets information about its role (standard node or gateway node) by the scenario describing TCL script.

6.4.3 Functions of Algorithms

The `receive packet()` function (Figure 6.3) is for every network node and it is called after the initialisation phase if a node receives any type of message or packet. If the

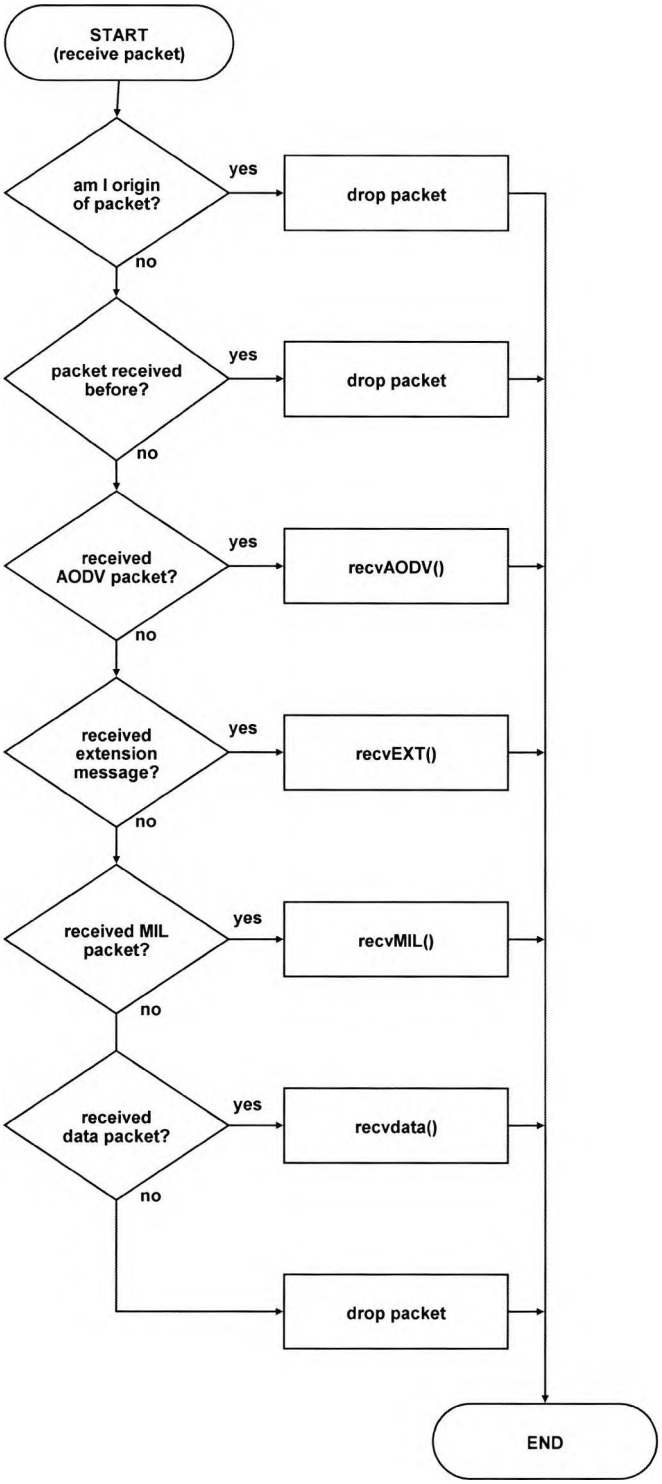


Figure 6.3: Receiving packets (1)

receiving network node is the originator of the received packet, determined by the source address, the node immediately drops the packet. This may occur if the node receives the forwarding of its own send packets by neighbour nodes. Additionally, if the packet was received before, like a flooding advertisement or solicitation, the node will drop the received packet, too. This is determined by the packet's source address and its sequence number.

If the node receives an AODV message it calls the `recvAODV()` (Figure 6.4) function. According to that the receiving of an extension message, i.e. message of the investigated GRREP feature of the proactive gateway discovery algorithms, will call the `recvEXT()` (Figure 6.5) function. If the node receives a message of the MobileIP protocol or a data packet it calls the `recvMIL()` (Figure 6.6) or the `recvdata()` (Figure 6.20) functions, respectively. In any other case the node will drop the received packet. Any other function that is called from here will return. Finally, the function ends.

The next three functions (`recvAODV()`, `recvEXT()`, and `recvMIL()`) are for deciding more precisely which type of message has been received. They are depicted in Figures 6.4, 6.5, and 6.6 and call the appropriate functions for processing the received messages further.

In Figure 6.4, if the node receives a message of the standard AODV routines (route request, route reply, route error) are not explicitly discussed in the thesis. More information about the standard AODV routines can be found in [20].

sendHELLO() The `sendHELLO()` (Figure 6.7) function is called for every node in the ad-hoc network including the gateway nodes right after the initialisation phase. Firstly, the function sets and waits for the initial HELLO timer with the initial offset to prevent synchronised sending of HELLO messages. If the gateway discovery algorithm is set to the advertisement or solicitation based algorithm the node just creates and sends HELLO messages every time the HELLO timer expires. If the gateway discovery algorithm is set to the HELLO message based algorithm and the executing AODV agent is part of an Internet gateway it includes its own gateway information (address, hop count, sequence number, gateway usage) into the HELLO message and sets the I-flag. If the executing agent is part of an ad-hoc mobile node it includes information

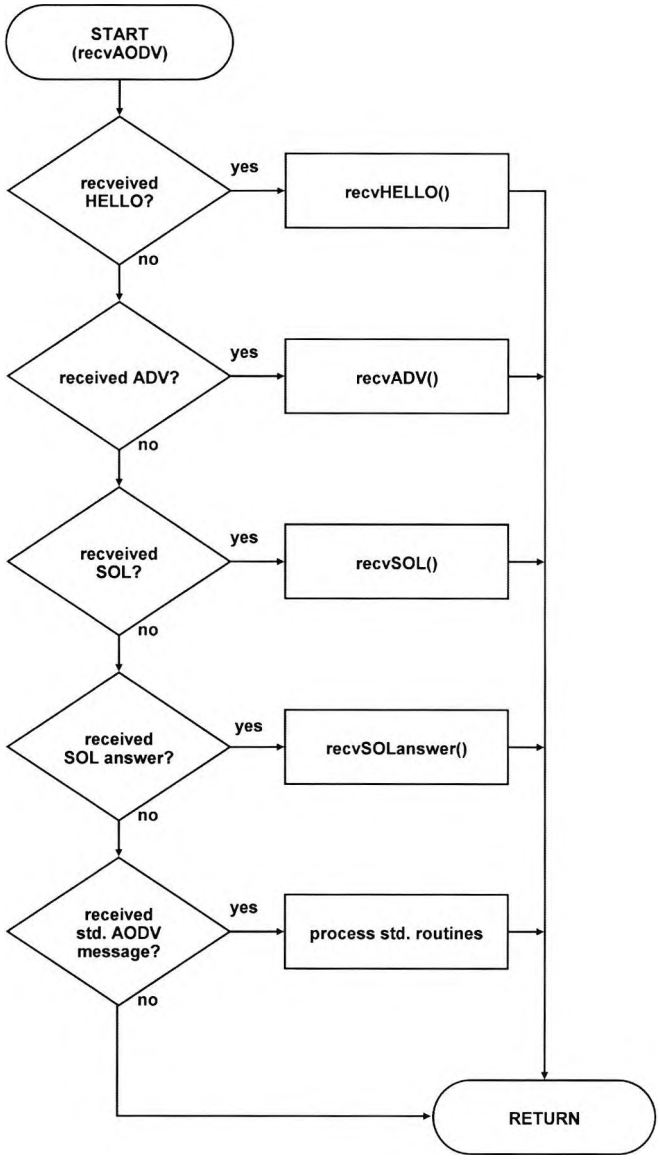


Figure 6.4: Receiving packets (2)

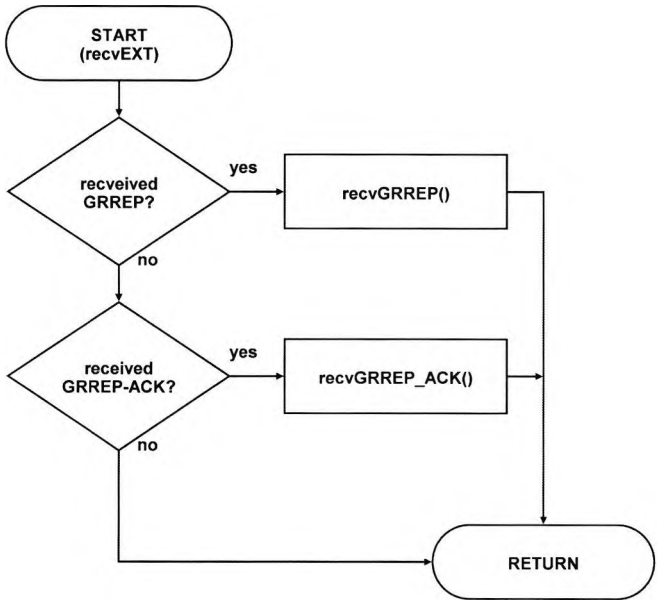


Figure 6.5: Receiving packets (3)

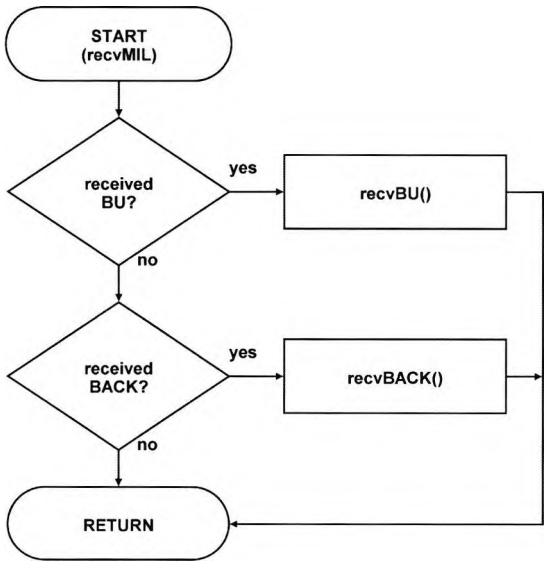


Figure 6.6: Receiving packets (4)

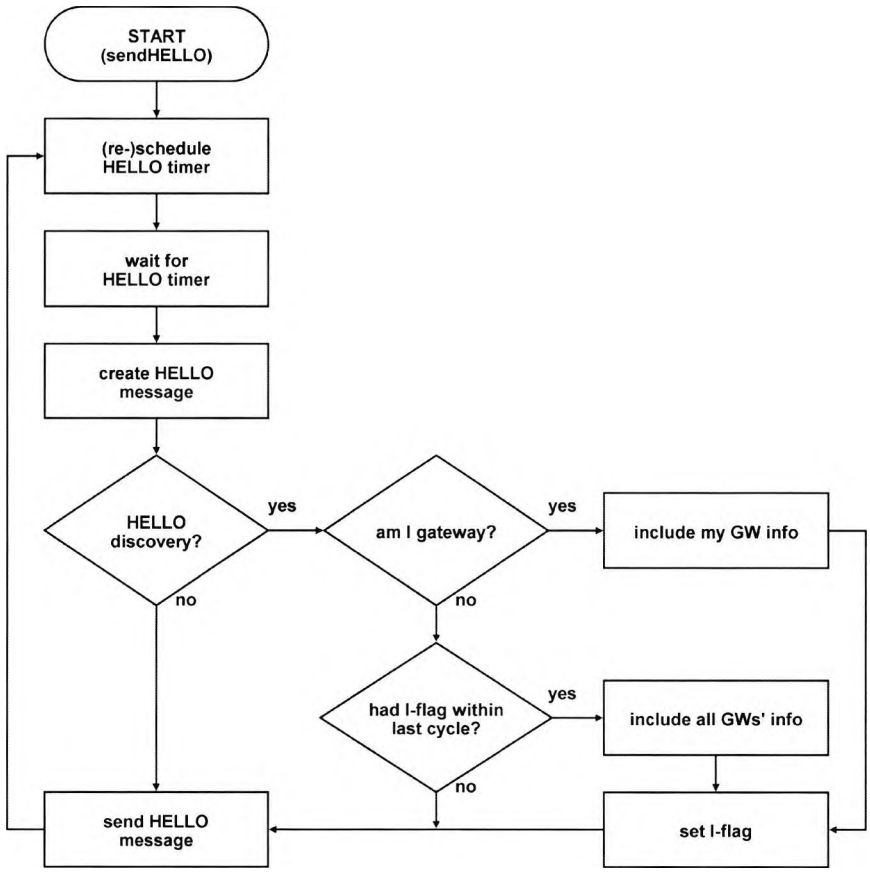


Figure 6.7: Sending a HELLO message

about Internet gateways if it self has received Internet gateway information within its last HELLO timer period. Note, if a mobile node knows multiple Internet gateways it will include information about all known Internet gateways into one HELLO message.

In any case, after the node has sent a HELLO message it sets its HELLO timer to the interval time which is fixed to one second and waits for the time out of the HELLO timer. Then the node will do the process of sending HELLO messages again. This function runs until the end of the simulation run.

recvHELLO() The function `recvHELLO()` is depicted in Figure 6.8. In any case, if a node receives a HELLO message it creates or updates the route to the originator of the HELLO messages respectively. I.e. that the receiving node now is aware of the neighbour node. If the I-flag is set (HELLO message based gateway discovery algorithm is used) the receiving node additionally, creates or updates the routes to all

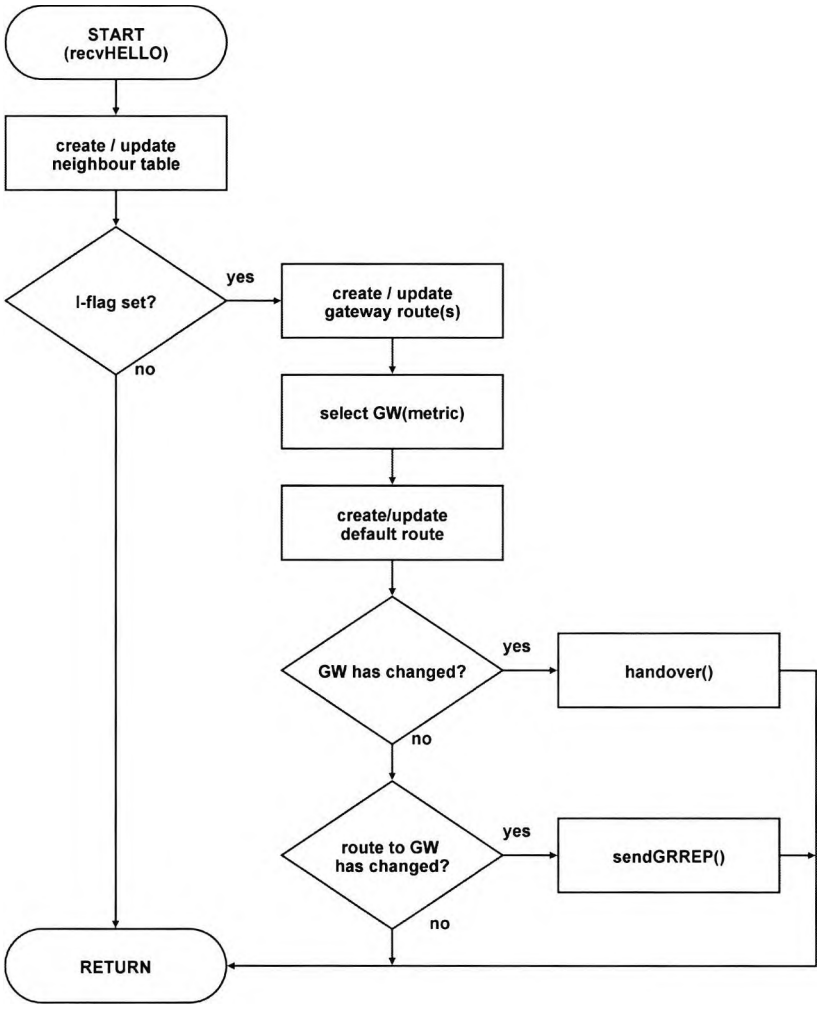


Figure 6.8: Receiving a HELLO message

in the message included Internet gateways. The node selects the "best" gateway by a certain metric (the metric is described in section 5.4). If the node finds that now a "better" gateway is available it calls the `handover()` function (Figure 6.14). Or, if the node does not find a "better" gateway but a "better" route to its already selected gateway and the first extension feature (GRREP) is enabled it calls the `sendGRREP()` function (Figure 6.17). If the I-flag is not set in the HELLO message the node only add the discovered neighbour node into its list of known neighbours (neighbour table).

sendADV() If the gateway discovery algorithm is set to advertisement based, gateway nodes call the `sendADV()` function from `initialisation()` (Figure 6.2). The function (Figure 6.9) firstly schedules the ADV timer with the initial offset to prevent

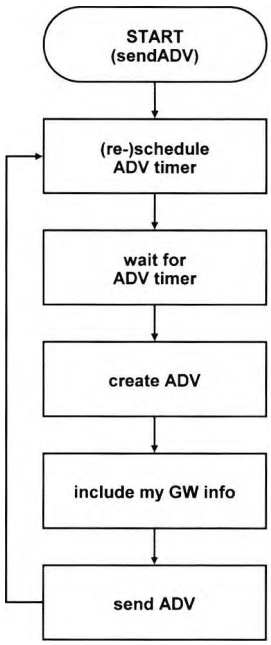


Figure 6.9: Sending an advertisement message (ADV)

synchronised sending of advertisement by all gateways. After the timer has timed out the function creates an advertisement, includes its gateway information, and sends it to the broadcast address of the ad-hoc network. Finally, the function reschedules the ADV timer. This function never returns and it will be stopped at the ending of the simulation run by the NS-2 `simulator` instance.

recvADV() If a mobile ad-hoc network node receives an advertisement from an Internet gateway it firstly creates or updates the route to the gateway where the advertisement was sent from. Thus, a network node is aware of all Internet gateways. If the advertisement is not from a gateway with a better metric ("better" gateway) the receiving node just forwards the advertisement, thus the ad-hoc network is flooded by the advertisement. If the advertisement is sent by an Internet gateway with a "better" metric the receiving mobile node creates or updates its default route. If the new default route points to a new gateway the receiving node calls the `handover()` function (Figure 6.14). Otherwise, if the new default route points to the old selected gateway, i.e. the next hop entry of the default route has not changed, but the route to the old Internet gateway is now shorter or longer and the gratuitous route reply feature

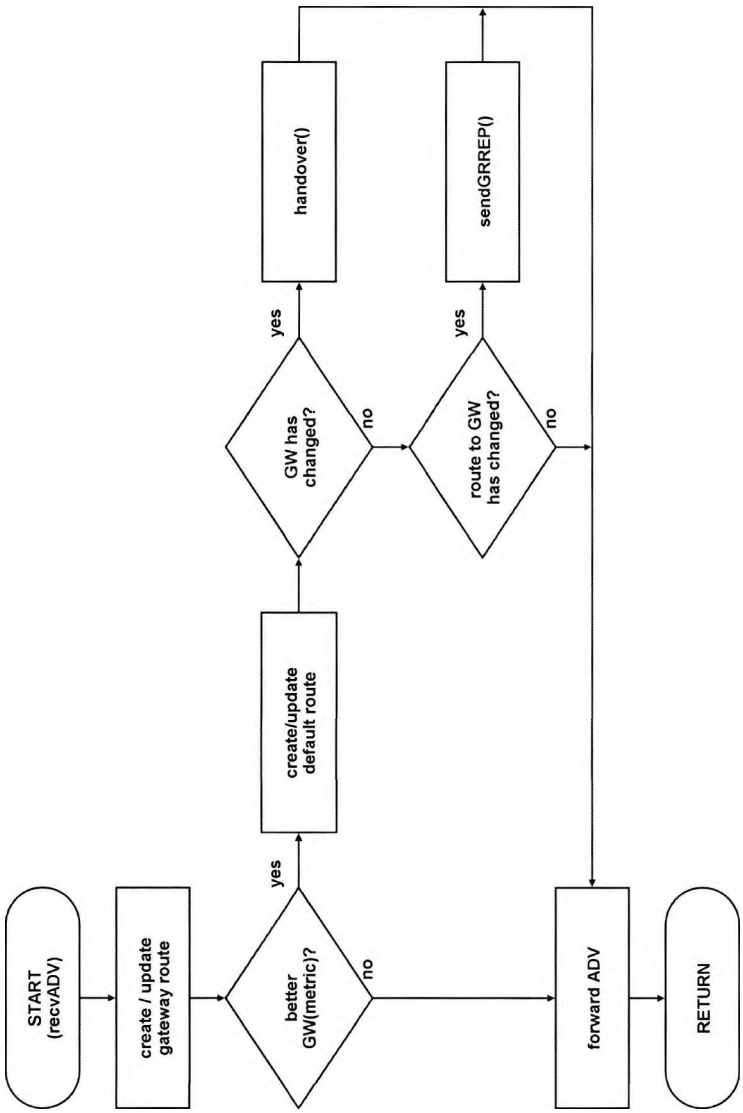


Figure 6.10: Receiving an advertisement message (ADV)

(GRREP) is enabled the node calls the `sendGRREP()` function (Figure 6.17). In any case the ad-hoc network is flooded with this advertisement (refer to [26] for partial flooding the ad-hoc network with advertisements).

sendSOL() If the gateway discovery algorithm is set to the solicitation based algorithm ad-hoc mobile network nodes have to search for Internet gateway connectivity reactively. This is achieved by calling the `sendSOL()` function. The `sendSOL()` function firstly schedules the SOL timer to the initial offset to prevent synchronised flooding of the ad-hoc network and after the SOL timer has timed out the network node will cre-

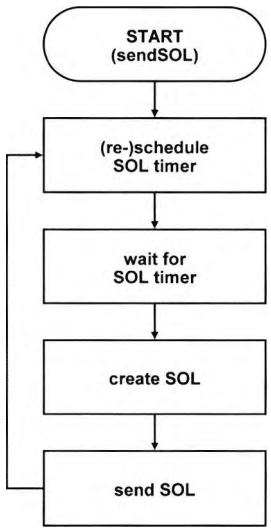


Figure 6.11: Sending a solicitation message (SOL)

ate and send the SOL (solicitation) message. After that the node reschedules the SOL timer to the standard interval time of one second and waits for the time-out again. This process is repeated until a solicitation answer from an Internet gateway is received (refer to function `recvSOLanswer()` and Figure 6.13). The `sendSOL()` function is depicted in Figure 6.11.

recvSOL() The `recvSOL()` function (Figure 6.12) may be called by any solicitation request forwarding node or by gateway nodes. In any case the receiving node creates or updates a reverse route entry to the originating node. If the receiving node is a mobile ad-hoc node it then forwards the solicitation request and thus, the ad-hoc network is flooded with that request. If the receiving node is an Internet gateway it generates an answer to that solicitation request and send the solicitation answer to the requesting ad-hoc node.

Note, only gateway nodes answer to solicitation requests. Never will an intermediate node that is aware of an Internet gateway answer to a request. This is to ensure that solicitation answers are always correct.

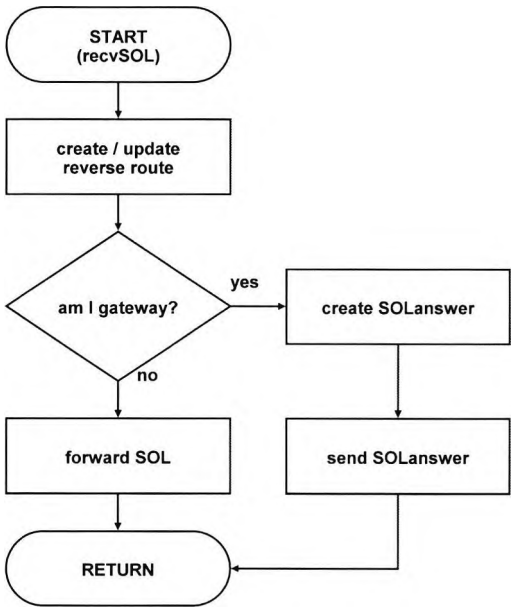


Figure 6.12: Receiving a solicitation message (SOL)

recvSOLanswer() After receiving a solicitation request a gateway node sends an answer to the requesting ad-hoc mobile node. The answer is then received and processed by mobile ad-hoc nodes. This is depicted in Figure 6.13. Firstly, when receiving the solicitation answer the receiving mobile node creates a reverse route entry to the originating gateway node. If the solicitation answer is from a "better" gateway the receiving mobile node updates its default route for Internet connectivity. Like receiving advertisements and HELLO_I messages the solicitation answer receiving mobile node will then either call the `handover()` or the `sendGRREP()` function (Figures 6.14 and 6.17) if the gateway has changed or the route to an already selected Internet gateway has been shortened or lengthened. However, if the receiving mobile node is the destination of the solicitation answer it cancels its pending solicitation timer or if the receiving mobile ad-hoc node is just an intermediate node it forwards the solicitation answer to the originally gateway requesting ad-hoc node.

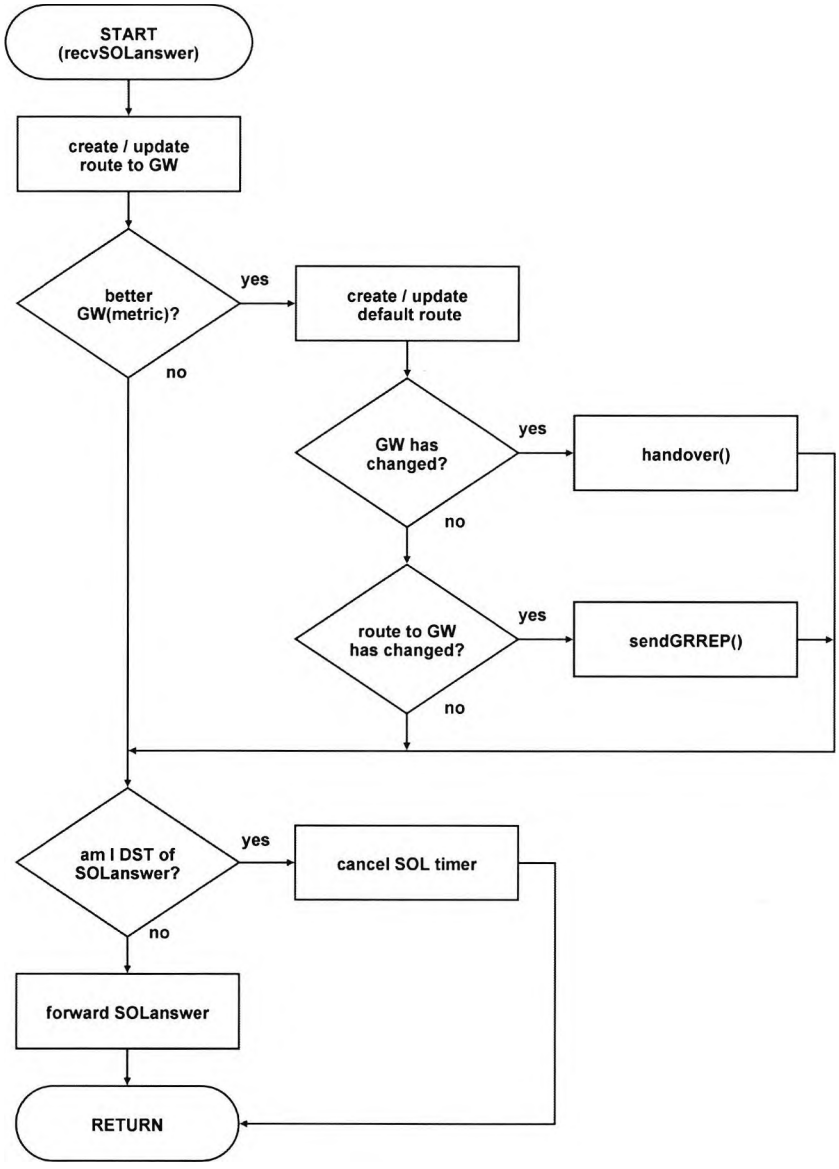


Figure 6.13: Receiving a solicitation answer message (SOLanswer)

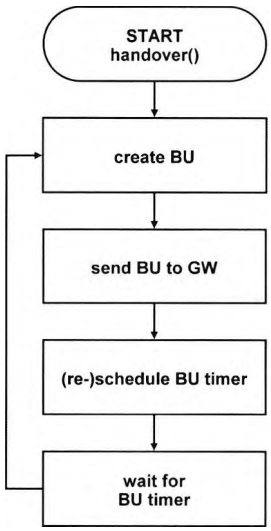


Figure 6.14: Performing a handover

Implementation of MobileIP

handover() The functions of the MobileIP protocol are implemented into MobileIP agents that are integrated into ad-hoc mobile network nodes and into the correspondent node (CN) representing the Internet and the home network of ad-hoc mobile nodes. See Figure 7.1 on page 132 for scenario setup.

An ad-hoc mobile node that firstly detects an Internet gateway or discovers a "better" Internet gateway has to inform its home agent. This is called a handover and the handover is managed by the `handover()` function. The function is depicted in Figure 6.14. A node performing a handover firstly creates a binding update (BU) message and sends it to the new Internet gateway using the just discovered Internet gateway route. Then the sending node schedules its BU timer. If the BU timer expires the node generates and sends a BU message again until the node receives an acknowledgement of the BU message (BACK) from its home agent. The time-out of the BU timer starts with one second and is then doubled for each re-transmission of a BU to a maximum of 32 seconds.

recvBU() In any case, if an ad-hoc mobile node receives a binding update (BU) message from another ad-hoc mobile node it creates or updates a reverse route entry pointing to the sending node. This is depicted in Figure 6.15. The `recvBU()` function

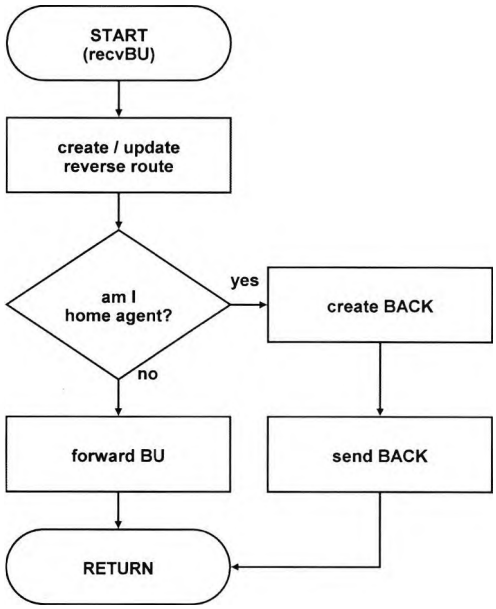


Figure 6.15: Receiving a binding update message (BU)

then decides whether the receiving node is a home agent or not. This is configured by the TCL scenario definitions. If not, thus it is an ad-hoc mobile node, it forwards the BU to its destination, i.e. the home agent. Otherwise, if a home agent receives a BU message it creates and sends an acknowledgement to the BU to the BU sending mobile ad-hoc node. This acknowledgement is called a binding update acknowledgement (BACK) message. The `recvBU()` function is given in Figure 6.15.

recvBACK() In Figure 6.16 the receiving of a binding update acknowledgement (BACK) message is illustrated. After setting the reverse route entry to the forwarding Internet gateway the receiving mobile ad-hoc node decides if it is the destination of the BACK message. If not the receiving node forwards the BACK to the destination node. If yes it sets its pending BU timer to the maximum time-out of 32 seconds.

Implementation of the GRREP extension

sendgrrep() If a mobile ad-hoc network node gets gateway information by either advertisements, solicitations, or HELLO_I messages it may discover a shorter or longer route to its already selected Internet gateway. In such a case the node sends a gratuitous reply (GRREP) message to its Internet gateway by calling the `sendGRREP()` function.

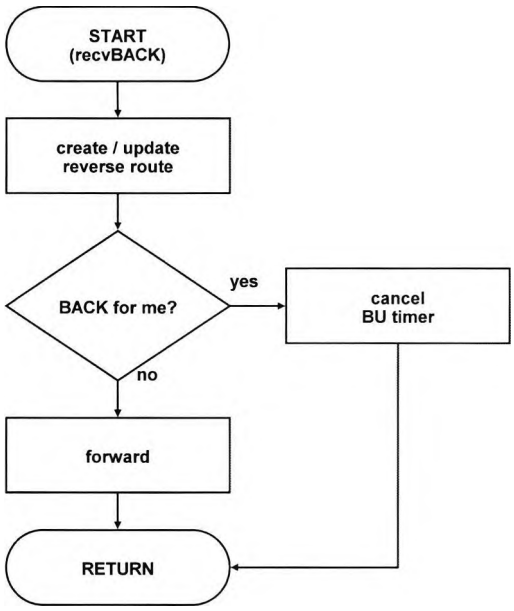


Figure 6.16: Receiving a binding update acknowledge message (BACK)

The function is depicted in Figure 6.17. A node sending a GRREP message firstly creates the GRREP message and send it to the selected Internet gateway. The sending node then sets a time-out for receiving an acknowledgement of the correct receiving by the Internet gateway. Until the acknowledgement has not arrived the sending mobile node will re-schedule its GRREP timer until an acknowledgement has been successfully received by the `recvGRREP-ACK()` function (Figure 6.19). Note that this only applies if the GRREP extension to the both proactive algorithms is enabled. This function never returns and thus it is endless. It will be interrupted either at the end of the simulation run by the `simulator` instance or the by the `recvGRREP-ACK()` function.

recvGRREP() A node receiving a GRREP message creates or updates a reverse route entry pointing to the originator of the GRREP message. Next, the function decides if the receiving node is an Internet gateway or not. If it is not an Internet gateway it forwards the GRREP to the Internet gateway's address in the destination field of the GRREP message. If it is an Internet gateway the function creates a gratuitous reply acknowledgement (GRREP-ACK) message and sends it to the GRREP originating mobile node. The `recvGRREP()` function is depicted in Figure 6.18.

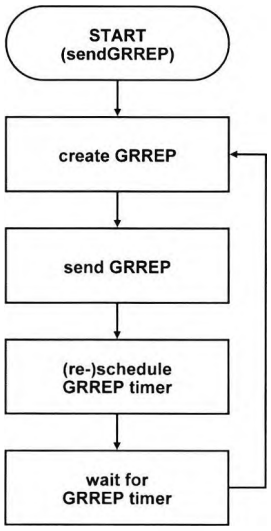


Figure 6.17: Sending a gratuitous route reply message (GRREP)

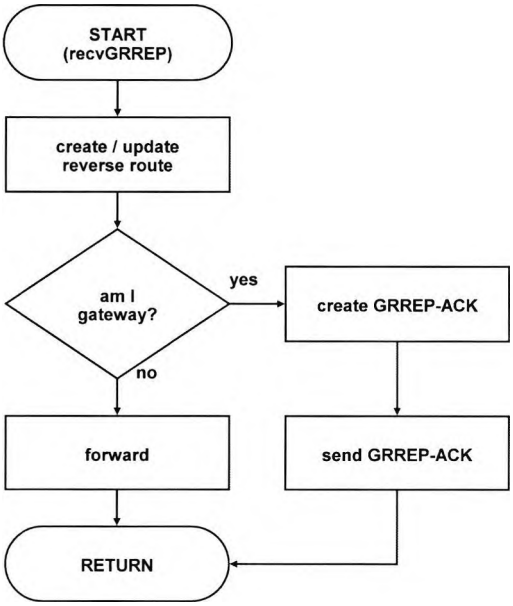


Figure 6.18: Receiving a gratuitous route reply message (GRREP)

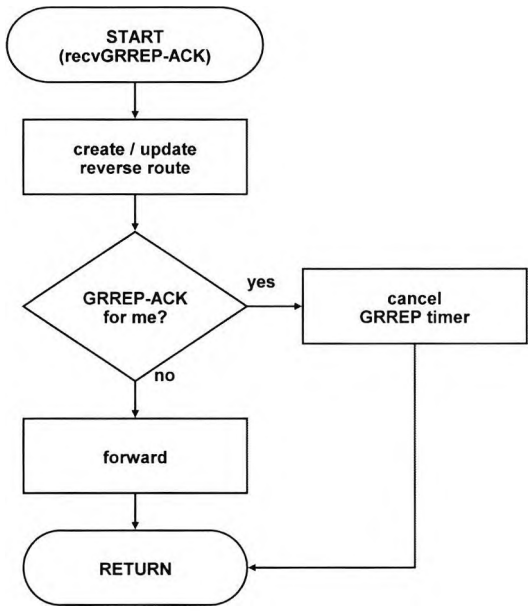


Figure 6.19: Receiving a gratuitous route reply acknowledge message (GRREP-ACK)

recvGRREP-ACK() A node that receives a gratuitous route reply acknowledgement (GRREP-ACK) message creates or updates a reverse route entry pointing to the sending Internet gateway. If the GRREP-ACK message is not destined to the receiving node the receiving node forwards the GRREP-ACK message to the destination node. If the destination node of a GRREP-ACK message receives the GRREP-ACK message it cancels its pending GRREP timer and thus, the GRREP routine ends. The `recvGRREP-ACK()` function is given in Figure 6.19.

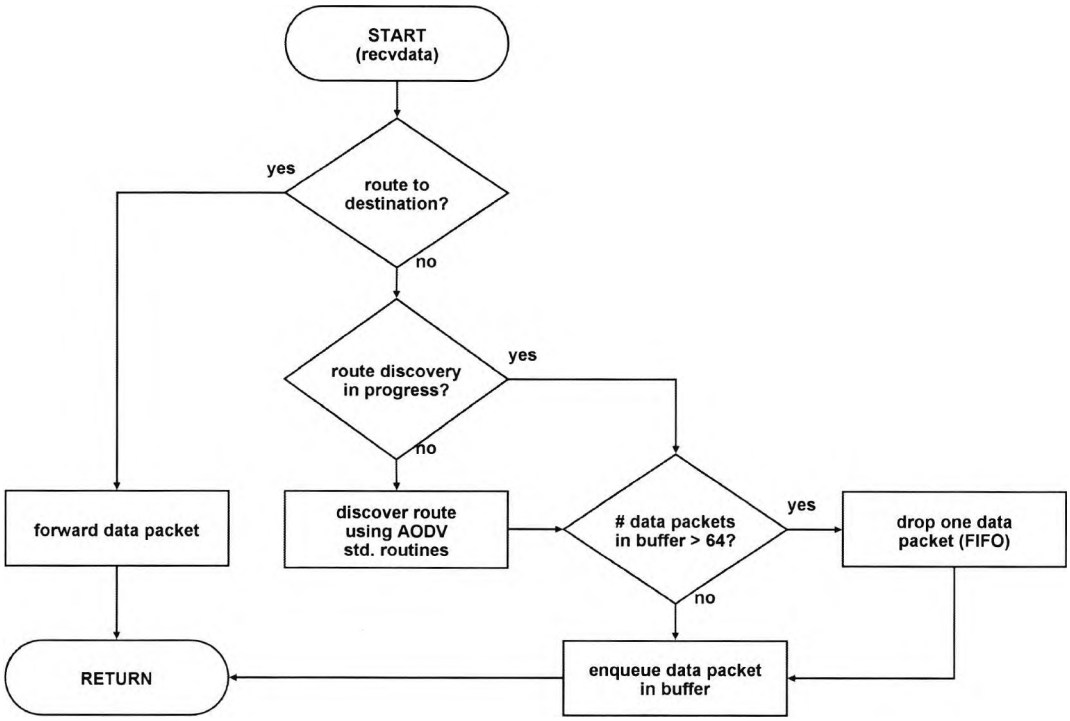


Figure 6.20: Receiving data packets

Implementation of data packet routines

recvdata() If an ad-hoc mobile node receives a data packet which is destined to that receiving node the algorithm delivers it to the appropriate data agent (Sink). This initial process is included into the standard routines of mobile nodes and decided by the classifier hierarchy and therefore not depicted here. If the data packet is not destined to the receiving node and, thus processed by the routing agent, the receiving node firstly looks if a valid route to the destination node exists. If such a route exists the node forwards the data packet to the destination node. If such a route is not known by the data packet receiving node the node performs an AODV route discovery if an AODV route discovery is not yet in progress and buffers the data packet to a maximum of 64 packets. If there are more than 64 packets in the buffer the oldest packet is discarded after the first in first out strategy (FIFO). The `recvdata()` function is depicted in Figure 6.20.

6.5 Reliability of Simulation Results

This section is to analyse the reliability of the NS-2 simulator by a literature research. Due to scalability reasons a software simulator was chosen to obtain performance results of the implemented algorithms and network nodes.

In the simulation model used for this research the following assumptions were made:

- The world is flat without any obstacles
- The radio range of nodes has the shape of a perfect circle
- We have symmetric links, i.e. if the destination node can receive the sending node the sending node can also receive the destination node
- If in radio range nodes can read their neighbours perfectly

These assumptions were made to simplify the simulation model and are very common in the literature. Nevertheless, such assumptions are expected to model the reality not very accurately although this depends on the situation a system is to be used in.

Publication [70] provides a good overview of assumptions in wireless simulations. The authors compile a number of axioms and test them in a real testbed. The paper concludes with a number of hints and suggestions to developers and researchers about the introduction of unsymmetric links, a 3-D terrain, and different radio propagation models. The authors do not conclude about the reliability of simulation results with a complex topology and scenario setup but only for simulation setups of specific axioms that are investigated in the paper.

The authors of [77] use the NS-2 simulation software and compare simulation results with results from a testbed consisting of 5 mobile nodes (cars) using a modified DSR routing protocol. First the authors study the IEEE 802.11 MAC model of NS-2 in a variety of scenarios. They conclude that the IEEE 802.11 MAC model of NS-2 appeared to work correctly. This result gives substantial confidence to further simulation results. In general, the paper is suggesting validation approaches rather than giving a statement on the validity of simulation results except the correct functionality of the IEEE 802.11 MAC implementation in NS-2.

The authors of [78] compare simulation results of the OPNET simulator [79] and the NS-2 simulator with network testbed experiment results. UDP and TCP data connections are investigated. The authors conclude that for a simple UDP traffic NS-2 can accurately model the testbed behaviour. In opposition, when modelling TCP traffic the NS-2 simulator was found to not model the dynamic behaviour of TCP adequately. Thus, the IEEE 802.11 MAC layer and the CBR traffic generation is well implemented into NS-2.

In [69] the authors investigate the packet delivery ratio, the connectivity ratio, and the packet latency in reality and in simulation. Their scenario setup consists of 16 static nodes forming an ad-hoc network with a maximum hop distance of six hops. The nodes run an OLSR like ad-hoc routing protocol called AWDS (developed by the authors). The simulations are carried out using NS-2 while the real experiment is located at the University of Magdeburg (office environment).

The authors suggest to use the shadowing radio propagation model which is firstly to be calibrated properly. In opposition, this thesis uses the two ray ground model of NS-2 in a flat world topology. A flat world topology is a topology with no obstacles or hills. This decision was made because the thesis investigates routing algorithms located at layer 3. A shadowing radio propagation model in a complex topology with obstacles and reflection increases the level of complexity and is suitable to investigate radio propagation.

In [69] the authors claim that the packet delivery ratio as well as the connectivity graphs can be modelled with a high accuracy whereas the quality of simulation results of packet delays is lower because of the bad implementation that does not consider real network properties such as hardware issues. Furthermore, the authors claim an error of the simulation against the real experiment of less than 1% for up to four hops in terms of packet delivery ratio. Packet delivery ratio is defined as the ratio of packets sent and packets received successfully via a multihop path. One must be careful when interpreting these results since in [69] the ad-hoc network is static in order to ensure that no mobility inaccuracies cause any additional errors. Such an accurate simulation model is expected to provide accurate simulation results in terms of throughput as investigated in this thesis (for static networks) since the throughput is significantly

influenced by the successful receiving of data and control packets. The connectivity graphs of the simulation in [69] shows an almost equal network topology compared to the real experiment. The authors conclude an error of 10% in terms of connectivity graphs.

For the research in this thesis a number of implementations were made to the NS-2 simulator. These new implementations are the advertisement, the solicitation, and the HELLO message based algorithm for Internet gateway discovery, the Internet gateway node, and MobileIP. All new implementations were tested by the author to work properly using simple scenario setups. Trace files and output of additional information about nodes' and algorithms' statuses were used to analyse the test simulation results. A trace file is generated and written to the hard disk while NS-2 is simulating. The trace file contains information about the sending and receiving of data and control messages at different layers of the OSI stack. By comparing the test simulation results with expected results the new implementations can be declared as working correctly.

By researching the literature it can be concluded that the wireless MAC layer protocol (IEEE 802.11) used in this thesis is accurately implemented and therefore NS-2 is expected to generate reasonable results, at least with UDP traffic. In the literature, TCP as the transport layer protocol is not found to be implemented accurately [78]. Additionally, the assumptions to the simulated world (flat world, no obstacles, radio range has shape of perfect circle) are known to decrease the level of accuracy of the simulation results in a complex topology like an office environment.

The interaction of different traffic types together with the influence of (flooded) control messages in a wireless environment consisting of a number of network nodes is very challenging to investigate. This is further discussed in chapter 8.

All implementations made for the research work in this thesis were tested by the author in detail.

6.6 Conclusion

In this chapter the implementations on the NS-2 simulation suite are presented. The chapter introduces the NS-2 software simulator in general and presents the implementation of different types of network nodes. Internet gateway nodes that are standard

mobile nodes extended by a second wireless network interface stack and an AGWAgent are introduced and their functionality is explained. Furthermore, the implementation of the advertisement based, the solicitation based, and the HELLO message based Internet gateway discovery algorithms is explained in detail and the routines that were implemented into NS-2 are depicted and discussed.

The next chapter analyses the gateway discovery algorithms' performance and investigates the parameters that have impact on the algorithms.

Chapter 7

Algorithm Evaluations

7.1 Overview

This task of this chapter is to examine and to analyse the operation of the *HELLO message based gateway discovery algorithm* and to compare its performance with the advertisement and the solicitation based gateway discovery algorithms. Therefore scenario setups were developed and simulated using the NS-2 [4] software simulator with appropriate implementations to the software. Details on the HELLO message based Internet gateway discovery algorithm can be found in chapter 4.

Further the thesis introduces two newly developed extensions to Internet gateway discovery algorithms as presented in chapter 5 that are investigated and examined here, too.

To test the HELLO message based and the well known advertisement and solicitation based algorithms and investigate their performance a number of mobile ad-hoc nodes are deployed to form a multihop mobile ad-hoc network. Internet gateways are part of the multihop mobile ad-hoc network providing Internet connectivity for the mobile ad-hoc nodes. The algorithm's behaviour is then tested and compared with the nodes' mobility as a parameter. The nodes of an ad-hoc network move around randomly according to the random waypoint model as used in [2, 3].

Another parameter for investigation is the number of nodes the ad-hoc network consists of. An ad-hoc network needs a minimum number of attending network nodes to work, i.e. a minimum node density. If there are not enough nodes the probability of finding routes will decrease. Thus, the number of simulated nodes is set to a value that

provides a reasonable probability of ad-hoc multihop route discovery. To investigate the algorithms with a higher node density the number of nodes is then doubled. Refer to section 7.4.1 for the probability of finding routes within an ad-hoc cluster successfully depending on the number of nodes with a fixed radio range. A third parameter for investigating mobile ad-hoc networks is the traffic within an ad-hoc network. Such traffic is called background traffic and the thesis investigates its influence on the operation of Internet gateway discovery protocols. Therefore the ad-hoc network is charged with CBR/UDP and FTP/TCP as background traffic.

The second task of the thesis is to examine the extensions made to the Internet gateway discovery algorithms. The thesis introduces two extensions. The first extension is called *Gratuitous Route Reply* extension and it allows mobile nodes of ad-hoc networks to deal with routes to discovered Internet gateways more effectively. Details on the first extension are given in chapter 5.3.

The second extension is called *Load Switching*. It allows mobile nodes to decide between discovered Internet gateways not only by the length of the route (hop count, like in standard AODV) to the gateways but by a function of the hop count and the traffic the gateway is already forwarding for other nodes. Details on the extension are given in chapter 5.4.

In the present chapter, for all Figures the simulated results of the advertisement based algorithm are printed in green and marked with a "$\times$", the results of the solicitation based algorithm are printed in blue and marked with a "$*$" while the results of the HELLO message based algorithm are printed in purple and marked with a "$\square$". Simulation results are subject to statistical variance. This is because of random factors in the simulations like random movement of nodes where some nodes are traffic sources and some are traffic destinations. Depending on the actual random topology a specific simulation result can be very low or very high. Thus, always a number of simulation runs is performed to increase the level of expressiveness of mean values. Further the 95% confidence interval is given to improve the expressiveness of the simulation results. The mobility and traffic models are discussed in section 7.2.1.

In the scenario setups a testing mobile node (MN) is switched on while the simulation is in progress. This gives time to other mobile nodes to establish an ad-hoc

network, i.e. to discover all needed routes and to register with the home agent in the Internet. Depicted results only refer to the time after the testing MN is switched on, i.e. that protocol messages generated to establish the initial ad-hoc network do not count for the analysis but only messages sent after the 50^{th} second do count. This is since nodes that are switched on are expected to generate e.g. protocol overhead above average due to registering with their home agents. Experience was made by simulations that a network similar to the networks used in this thesis and the fact that distant nodes receive gateway information with delay (as described in equation 4.1) will be established completely after a maximum of 3 seconds. Thus a time span of 50 seconds is expected to be adequate for establishing the complete ad-hoc network.

The thesis introduces an algorithm performance index. This performance index simplifies and advances the comparison of algorithms. The index is calculated from the bandwidth an Internet gateway discovery algorithm provides to ad-hoc network nodes (measured with a test file download) and the routing protocol overhead needed.

In this chapter a performance analysis of the defined Internet gateway discovery algorithms is provided with the node movement, the background traffic, and the node density as topology parameters. Firstly, in order to provide the analysis more simulation parameters are defined next. This is followed by a detailed definition of the investigated characteristics of the gateway discovery algorithms and the new performance metrics used in this thesis. The chapter then provides the performance analysis of the new HELLO message based Internet gateway discovery protocol. After that both extensions made to the gateway discovery algorithms are investigated using the same performance metrics.

7.2 Simulation Parameters

The thesis's evaluations use the parameters given in Table 7.2. Next the simulation models are described in detail. This is followed by the presentation of the algorithm parameters.

7.2.1 Simulation Models

Traffic Model

To stress the ad-hoc network and to examine the protocol reactions to background traffic different types of background traffic are set up. Background traffic is set up between a number of source and destination nodes. Source and destination node pairs are the same for every simulation run but, their locations on the simulation plane are randomly chosen with a uniform distribution. The background traffic connections start with the simulation run (at $t_{SIM} = 0$ seconds) and end with the simulation run. Thus they are infinite.

One type of background traffic is infinite CBR/UDP traffic at rates from 0 $\frac{\text{kbit}}{\text{s}}$ to 640 $\frac{\text{kbit}}{\text{s}}$ modelling a number of VoIP connections with a full-duplex rate of 160 $\frac{\text{kbit}}{\text{s}}$ each. A traffic rate of 640 $\frac{\text{kbit}}{\text{s}}$ is high enough to congest the network and therefore it is set as the maximum. Note, that in a multihop connection a CBR/UDP data stream is consuming more bandwidth (air time) compared to a single hop connection since a data packet is forwarded and while a data packet is being forwarded surrounding nodes suffer. A mean hop length between a randomly located source and a randomly located destination node of 2.7 hops is found using the investigated simulation plane and can be interpreted as a multiplier for the background traffic. Thus, a data rate of e.g. 160 $\frac{\text{kbit}}{\text{s}}$ that is forwarded 2.7 times consumes $160 \frac{\text{kbit}}{\text{s}} \cdot 2.7 = 432 \frac{\text{kbit}}{\text{s}}$ bandwidth.

The other type of background traffic is up to two infinite FTP/TCP connections. The source and destination nodes are the same for every simulation run but, as described above, randomly located.

All background traffic is set up within the ad-hoc network except in section 7.6 (Analysis of Load Switching Extension) where the background traffic is set up from ad-hoc nodes to the Internet gateway. The impact of network background traffic is investigated in section 7.4.3.

Mobility Model

The ad-hoc mobile nodes located in the simulation plane are moving according to the random waypoint model. The random waypoint model is a mobility model and it is very popular in the literature and also used in [2, 3]. Nodes using the random waypoint

model choose a random destination (waypoint) and a random movement speed. Both are uniformly distributed whereas in this thesis the maximum movement speed is set up to 10 $\frac{m}{s}$ as in [2, 3]. After reaching the destination point a specific mobile node waits a certain time until it chooses another random waypoint within the simulation plane. This waiting time is called the *pause time* and in this thesis it is the parameter for the nodes' mobility.

The pause time is varying from 0 seconds (high node mobility, node moves immediately to a next random destination) to 900 seconds (low node mobility, node retains position). Since the total simulation time is set to 900 seconds the pause time parameter is from 0 seconds to 900 seconds in steps of 225 seconds in order to simulate a number of intermediate values. This means that a pause time of 0 seconds stands for permanent node movement whereas a pause time of 900 seconds stands for an almost static network. Almost static because nodes may move when the simulation run is being initiated and then stop at a specific position but such a node will wait for 900 seconds (= pause time) at this specific position and will never move again in this simulation run. To achieve this more realistic node movement the `setdest` program of the NS-2 simulation suite has been slightly modified. This modification is discussed in section 6.2.4.

Topology Model

The number of nodes an ad-hoc mobile network consists of with respect to the area the nodes are located within plays an important role. The node density is defined as the number of nodes per area and it is investigated in this thesis in section 7.4.2. The thesis investigates 30 and 60 nodes.

Using a simulation plane of 1000 metres $\times$ 800 metres with increasing number of nodes the probability to find a multihop route to a destination node increases. To choose a suitable node density simulations were carried out with different node densities and the connectivity ratio, i.e. the number of successful established multihop connections, is investigated. The results are compiled in Table 7.1. By following the results in Table 7.1 the minimum number of nodes for the simulations in this thesis is set up to 30 nodes in the simulation plane since then the probability for a specific node (the MN) to find

Number of nodes	Connectivity ratio [%]	based on X runs
5	1.25	400
10	18.3	300
15	49.5	200
20	77.0	200
25	88.0	100
30	98.0	100

Table 7.1: Connectivity ratio for different node densities

an Internet gateway is 98%. To investigate the node density's influence the number of nodes is then doubled to 60 nodes.

7.2.2 Algorithm Parameters

gateway discovery algorithm The algorithm mobile nodes use to discover Internet gateways within a mobile ad-hoc network can be set to the advertisement based, the solicitation based, and the newly developed HELLO message based algorithm. More details on the advertisement and the solicitation based Internet gateway discovery algorithms can be found in section 3.3.4. Details on the HELLO message based Internet gateway discovery algorithm can be found in chapter 4.

interval time The interval time of an Internet gateway discovery algorithm has different meanings for each algorithm. For the advertisement based algorithm the interval time is the time between two consecutive advertisements of each Internet gateway. In the case of the solicitation based algorithm the interval time is the time-out of a solicitation request by a mobile node. In both, the advertisement and the solicitation based algorithms, the interval time of sending HELLO messages in fixed at one second. For the HELLO message based Internet gateway discovery algorithm the interval time is the time between two consecutive HELLO(_I) messages and therefore the sending rate of HELLO(_I) messages is variable and not fixed. Refer to section 7.3.5 for details on the interval time. In this thesis the interval time of all examined gateway discovery algorithms is fixed to one second.

gratuitous RREPs Is the first extension made to the discovery algorithms. It can be switched on and off and is investigated in section 7.5.

parameter	fixed/variable	value
gateway discovery algorithm	variable	ADV, SOL, and HELLO based
interval time	fixed	1s
gratuitous RREPs	variable	on or off
Load Switching	variable	on or off
node density	variable	30 and 60 nodes
pause time	variable	0-900s in steps of 225 s
node movement speed	variable	0-10 $\frac{m}{s}$ (uniformly distributed)
background traffic (infinite duration)	variable	0-640 $\frac{kbit}{s}$ CBR/UDP or 2 FTP/TCP connections

Table 7.2: Simulation parameters

Load Switching Is the second extension made to the Internet gateway discovery algorithms. The Load Switching extension can be switched on and off and it is investigated in section 7.6.

7.3 Investigated Algorithm Characteristics

The thesis provides a performance analysis of Internet gateway discovery algorithms. Therefore the metrics of the performance of Internet gateway discovery algorithms is defined next.

7.3.1 Register Time

The register time (t_{register}) is the time that elapses from the moment an ad-hoc network node is switched on within an already established ad-hoc cluster until the node has discovered an Internet gateway and registered with its home agent successfully. The register time is composed of two parts. The first part is the time a mobile node needs to detect an Internet gateway within an ad-hoc cluster ($t_{\text{discovery}}$) using an Internet gateway discovery protocol. The second part is the time the mobile node needs to register successfully with its home agent in the Internet using MobileIP (section 2.3.3) after it has discovered an Internet gateway (t_{MobileIP}). This is illustrated below in equation 7.1.

$$t_{\text{register}} = t_{\text{discovery}} + t_{\text{MobileIP}} \tag{7.1}$$

The register time can be interpreted as an important characteristic of an Internet gateway discovery algorithm since end-users want to get an Internet connection as quickly as possible when switching their mobile devices on and do not want to wait for connectivity. Thus the register time should be as short as possible. It is necessary to point out that the delay of the MobileIP registering procedure is assumed to be equal for all discovery algorithms and it is typically 30 ms. Thus, the depicted results on the register time in this chapter may be subtracted by 30 ms to get the gateway discovery time of an ad-hoc mobile node. As a conclusion the register time is mainly driven by the Internet gateway discovery time and not by the time the MobileIP protocol needs for registering.

The re-register time of MobileIP in the case of a handover is irrelevant, respectively. If an ad-hoc node decides to change the Internet gateway by analysing the ad-hoc network and information provided by the gateway(s) the node suffers a break in the Internet connectivity. This break is a result from disconnecting from one gateway and connecting to the next gateway. Obviously the handover time should be as short as possible in order to allow a seamless service of connectivity to the end-user. Thus, an established VoIP connection should not even recognise that the connection to an Internet gateway has changed. In the thesis the handover time is defined as the time from the decision of the mobile node to change the Internet gateway until it has re-registered with its home agent in the Internet successfully. If the MN decides to change to another Internet gateway data packets via the old gateway will still be delivered to the MN if the old gateway still has a valid route to the MN. Only packets sent by the correspondent node (CN, representing the Internet, the home network, and the home agent) after the CN has received a Binding update message BU from the MN will be sent via the new selected gateway (the MN sends binding update messages to the home agent and the corresponding node to prevent triangular routing, cp. to section 2.3.3). Thus, the MN will not recognise a significant break in the Internet connectivity.

7.3.2 Provided Throughput

The bandwidth of an Internet connection is a major sales argument for providers. Thus, this thesis evaluates the bandwidth of an Internet connection of a specific MN using an

Internet gateway via a multihop mobile ad-hoc network. Therefore, a testing mobile node in an ad-hoc network downloads a file from the Internet of a specific size (1 MB) and the provided bandwidth is given as a percentage value of the maximum throughput of a single hop connection to the Internet gateway. See Appendix A for more details on the maximum throughput. Thus, the given bandwidths in this chapter are always referenced to the moment the file download is in progress and are net values for the application the end-user is running.

The size of the test file is small enough to finish the download within the simulation time and large enough to be finished not too fast. Thus, the test file size is an empirical value.

7.3.3 Protocol Overhead

Every routing protocol needs to send messages to other nodes in the network to provide connectivity. The more messages a protocol needs for establishing and maintaining a connection the more bandwidth is wasted by the routing protocol. The total number of routing protocol messages is called the routing protocol overhead. As in [22, 26, 27, 60, 66] the overhead is the sum of packets generated by the gateway discovery algorithm, the MobileIP protocol, and the extensions made to the algorithms. In this thesis the protocol overhead is given as a multiple of the protocol overhead of a standardised scenario. The standardised scenario and the protocol overhead generated there is discussed in Appendix B.

7.3.4 Protocol Efficiency Index

A routing protocol performs better if it causes less overhead when it provides the same connectivity between network nodes compared to another routing protocol or if it provides better connectivity with less overhead. There is always a trade-off in sending routing protocol messages and providing bandwidth when looking at mobile ad-hoc networks. The protocol efficiency index is introduced to allow a fair and quick comparison of the algorithms for Internet gateway discovery.

A protocol that provides short multihop routes (shorter multihop routes are more resistant to node movement and provide faster downloads) and consumes less bandwidth

for sending protocol messages is considered as a protocol of high efficiency. The thesis introduces the protocol efficiency index as defined in equation 7.2.

$$\text{protocol efficiency index} = \frac{\text{throughput}}{\text{overhead}} \left[\frac{\text{kbit}}{\text{s} \cdot \text{packets}} \right] \tag{7.2}$$

7.3.5 Influence of Interval Time

When investigating Internet gateway discovery algorithms one could think about the interval time as a main parameter for investigations. The interval time has a different meaning for the three gateway discovery algorithms. For the advertisement based algorithm the interval time is the time between two consecutive advertisements from a specific Internet gateway. For the solicitation based algorithm the interval time is the time between two consecutive Internet gateway solicitation requests of a specific node (RREQ_I time-out). The advertisement and the solicitation based algorithms use only standard HELLO messages for their neighbourhood management and send HELLO messages every second and thus, regardless of the interval time, the sending rate of HELLO messages is constant.

The HELLO message based algorithm uses HELLO messages for the Internet gateway discovery and therefore the interval time is the time between two consecutive HELLO messages sent by a single ad-hoc node including the gateway node and thus, the interval time for the HELLO algorithm is a main simulation parameter compared to the advertisement and solicitation based algorithm where the HELLO interval is static.

Additionally, since HELLO messages are used for neighbourhood management, the HELLO message based gateway discovery algorithm will suffer from long interval times in sending HELLO messages. Then a mobile node needs more time to detect the loss of connectivity to a neighbour node. This time is called Δt_{loss} and it is described in equation 4.2.

In general, a long HELLO interval time will lead to bad neighbourhood management for all algorithms using HELLO messages for neighbourhood management. Therefore, in this thesis the interval time is not chosen as a simulation parameter. The interval time as a simulation parameter is investigated in [2, 3]. In this thesis the interval time

is fixed to 1 second. Next the performance of the HELLO message based gateway discovery algorithm is evaluated.

7.4 Performance Evaluation of the HELLO Algorithm

7.4.1 Impact of Node Density

The new Internet gateway discovery algorithm based on HELLO messages is to be compared to the established advertisement and solicitation based algorithms for Internet gateway discovery. Firstly the three algorithms are investigated to find the impact of the density of the ad-hoc network nodes. Therefore, a number of nodes is moving around randomly within the simulation plane of 1000 metres $\times$ 800 metres.

The number of nodes is set to 30 and 60 nodes, respectively. Each node's radio interface is adjusted to 250 metres range while the interference range between the nodes is set to 550 metres. I.e. that two ad-hoc nodes at a distance of 250 meters can establish a link between each other perfectly. If the distance between the two ad-hoc nodes is set to 251 metres the nodes cannot connect to each other but they will disturb radio transmissions of the other node. Only for distances of more than 550 metres an undisturbed communication link can be established between ad-hoc node pairs each.

In the moment an Internet gateway is announcing its presence into the ad-hoc network by flooding advertisements or a mobile ad-hoc network node requests Internet gateway connectivity by broadcasting solicitations all nodes of an ad-hoc network can be interpreted as static nodes, i.e. that for this short period of time neither node is moving. With the above mentioned density of ad-hoc nodes the following connectivity ratios result between a test mobile node MN and two Internet gateways. The scenario setup is similar to the depicted setup in Figure 7.1 with a MN located below the CN, i.e. between the two Internet gateways at MN(500|100). The results are compiled in Table 7.1.

Following the results in Table 7.1 the minimum node density in the simulation setups in this thesis is set to 30 nodes (except determined scenarios) to generate reasonable results with low node densities. Using less than 30 nodes for this specific topology setup the worse connectivity ratio's influence to the algorithms' performance will increase.

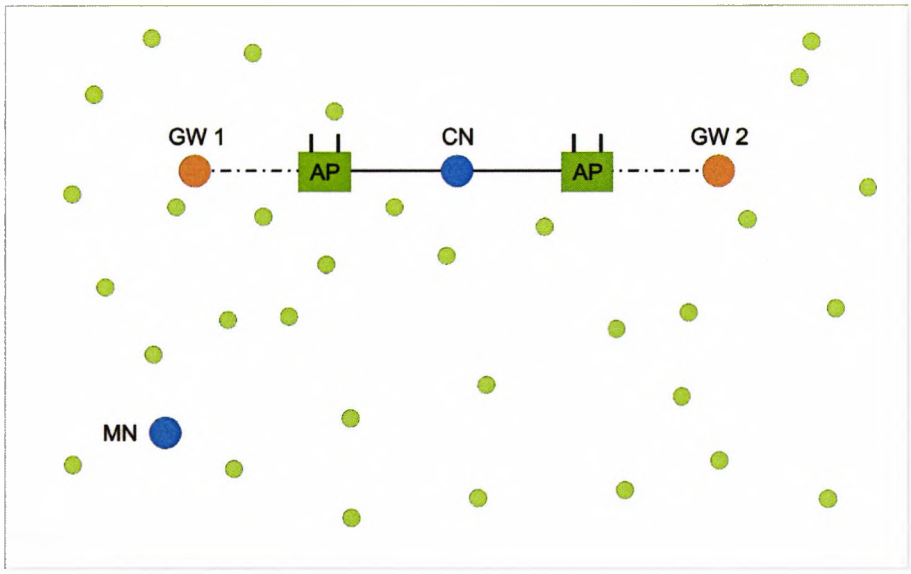


Figure 7.1: Simulation setup for the HELLO algorithm

With a doubled node density of 60 nodes the protocol overhead caused by network nodes increases. This protocol overhead increase is subject of investigations in this thesis.

For the following algorithm evaluations the scenario setup depicted in Figure 7.1 applies. While the mobile nodes are moving around randomly one additional test mobile node (MN) is moving at $y_{\mathrm{MN}} = 400\mathrm{m}$ from the left to the right and back again while the MN is downloading a test file of 1 MB in size ($x_{\mathrm{MN}_{\mathrm{left}}} = 200\mathrm{m}$, $x_{\mathrm{MN}_{\mathrm{right}}} = 700\mathrm{m}$, $v_{\mathrm{MN}} = 10\frac{\mathrm{m}}{\mathrm{s}}$). The MN is switched on at $t_{\mathrm{SIM}} = 50$ seconds and the test file download starts at $t_{\mathrm{SIM}} = 100$ seconds. The number of simulation runs is 200.

With an increased node density an ad-hoc network provides more possible routes between mobile ad-hoc nodes and gateways respectively. Therefore, a higher node density will lead to shorter register time delays for all three investigated algorithms since the testing MN's possibility is higher to discover a route to the gateway faster.

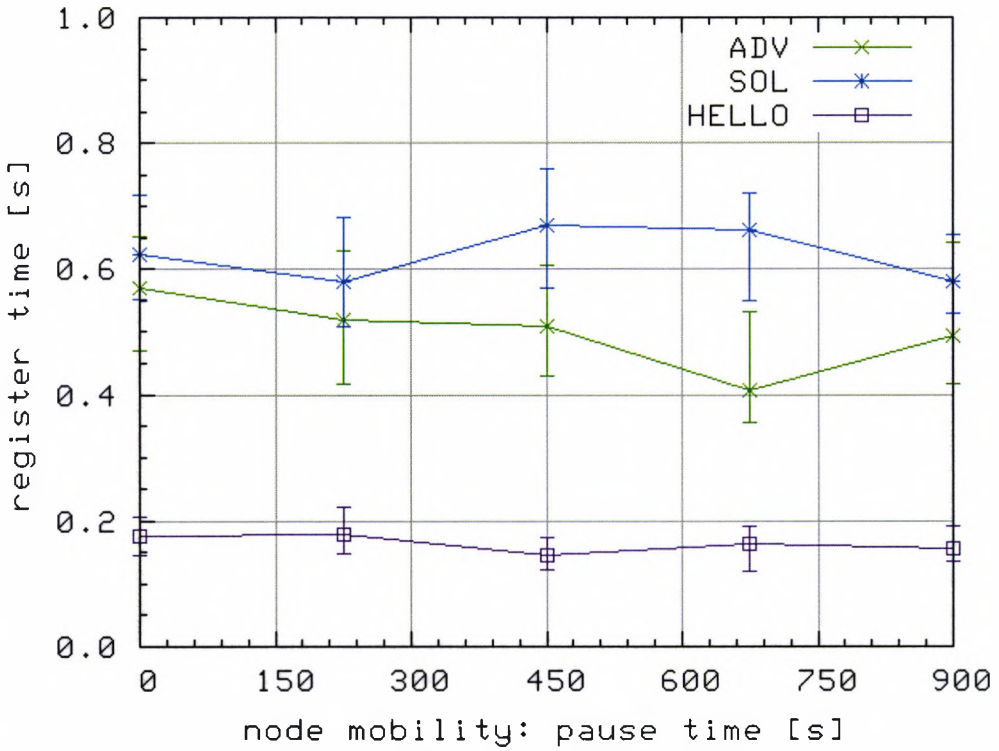
To compare the three algorithms the simulation results of the register time of the MN, the bandwidth while the test file transfer is in progress, the protocol overhead, and the protocol efficiency index are given in the following figures. To investigate the influence of the nodes' density the results of the low node density as well as the results of the high node density are given on top of each other for comparison. In this

first instance the simulations are without any background traffic. See section 7.4.3 for simulations with background traffic.

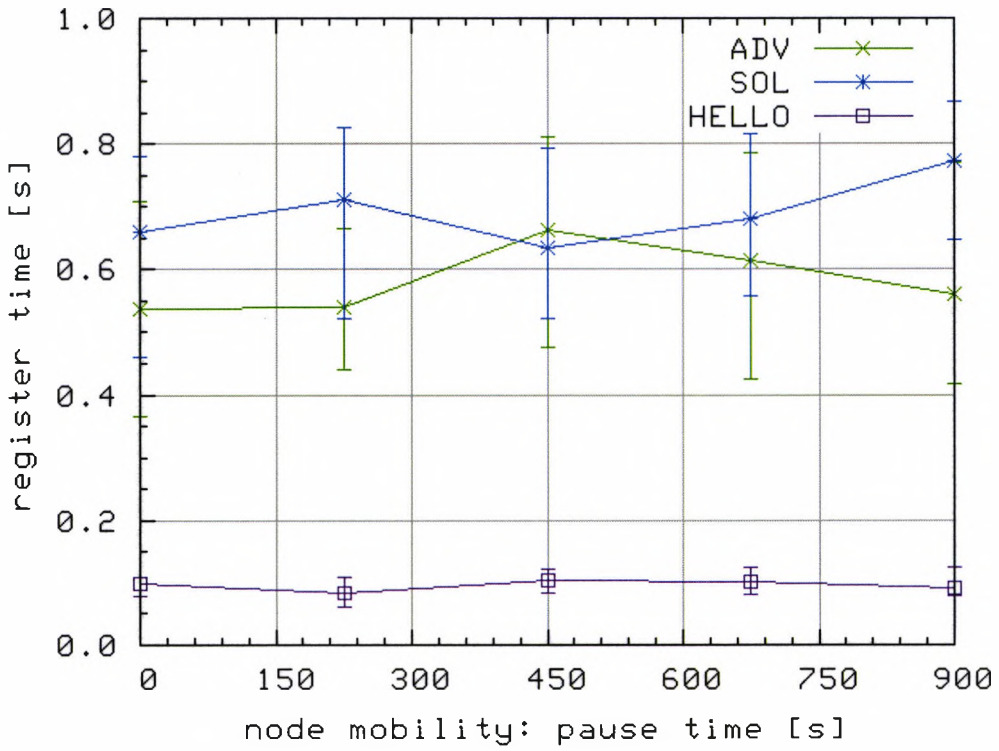
The time the MN needs to firstly discover an Internet gateway and secondly to register with the home agent is called the MN register time. The results are given in Figures 7.2(a) and 7.2(b). The graphs show the median value as well as the 5% and the 95% percentile of all simulation runs. Depending on the actual random topology of a specific simulation run a mobile node can either detect the Internet gateway very fast (< 0.2 seconds) or has to wait for Internet gateway connectivity provided by random moving mobile nodes. Thus the random topology is either able to provide a multihop route to a specific destination node very fast or not. Observations show maximum register times of up to 1 minute and more in 1.5% of all simulation runs. 14.5 % of all simulation runs lead to a register time between 1 second and 60 seconds. This extrem variance is one fundamental result of the thesis.

For the advertisement based algorithm the median register time is about 0.4 to 0.6 seconds whereas the register time for the solicitation based algorithm is about 0.6 to 0.7 seconds. The HELLO message based algorithm shows the shortest register times of about 0.1 to 0.2 seconds. The short register time results of the HELLO message based algorithm are explained with the unsynchronised sending of HELLO messages, i.e. the MN needs not to wait too long until the next HELLO_I message is received from a neighbour node. Using the advertisement based algorithm the MN needs to wait for one of the next advertisements from the Internet gateways. With the solicitation based algorithm the MN broadcasts for Internet gateway connectivity and if such a broadcast fails due to network congestion or collisions, the MN will wait for one second (interval time) until the solicitation is repeated. Thus the MN needs more time to discover an Internet gateway compared to the advertisement and HELLO message based algorithms.

With an increased number of ad-hoc mobile nodes (60 nodes) the results change. Using the advertisement and the solicitation based algorithms the register time increases about 0.1 second. This fact is explained with the unsuccessful discovery of the Internet gateway due to collisions of broadcast messages. With the HELLO message based algorithm the register time decreases about 0.1 second. This is due to the in-



(a) Register time with low node density



(b) Register time with high node density

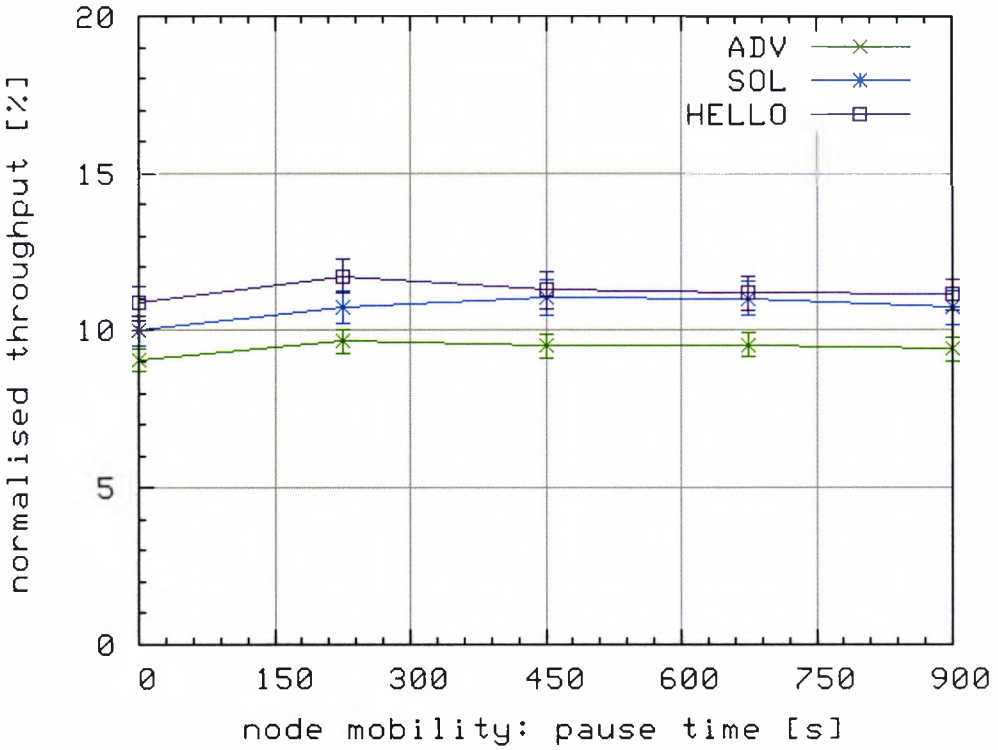
Figure 7.2: Register time with low and high node density and the nodes' mobility as parameter

creased number of direct neighbour nodes to the MN. With an increased number of direct neighbour nodes the MN does not need to wait too long for the next Internet gateway information containing HELLO_I message. Obviously the HELLO message based algorithm benefits from the increased node density whereas the advertisement and the solicitation based algorithms suffer from their broadcast strategy.

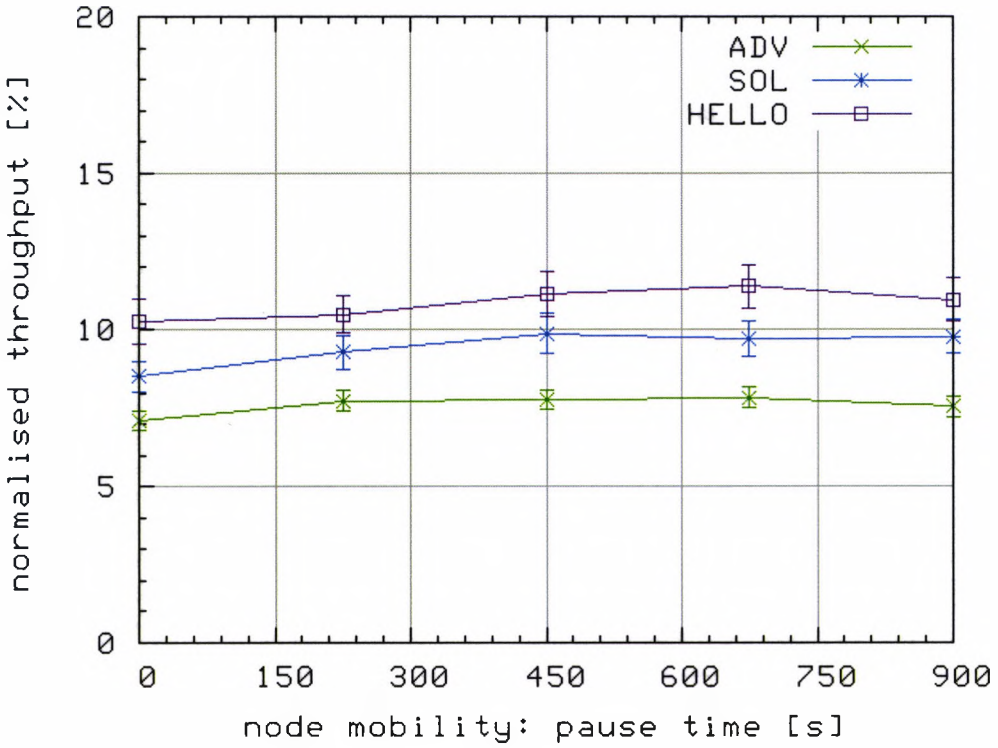
The throughput the Internet gateway and the attached multihop ad-hoc network provides to the MN while the test file download is in progress is depicted in Figure 7.3. It can be observed that the mean throughput is higher than 11.5% of the throughput of the standardised connection (see Appendix A). This maximum value is achieved by the HELLO algorithm. The advertisement based algorithm provides 9.5%. The solicitation based algorithm's provided throughput is between the HELLO and the advertisement based algorithms. The less throughput provided by the advertisement based algorithm is explained with the bandwidth consuming periodic flooding of the ad-hoc network with advertisements. Looking at the solicitation based algorithm that reduces the flooding of the ad-hoc network, the provided bandwidth is increased to 11% compared to the advertisement based algorithm.

With the increased number of ad-hoc mobile nodes to 60 nodes the provided throughput by the HELLO message based algorithm is close to 11% and therefore almost equal to the results with 30 nodes. The two other algorithms show a slightly decreased throughput to 7.5% (advertisement based) and 9% (solicitation based) which is explained with the increased control overhead for a higher node density caused by network-wide flooding.

The mean provided throughput for the advertisement based algorithm is less compared to the other discovery algorithms. This is because of the increased protocol overhead for the advertisement based algorithm (Figure 7.4). Reciprocally, the mean throughput achieved using the solicitation and HELLO message based algorithms is higher. It can be concluded that the advertisement based algorithm consumes much more of the limited bandwidth resources of an ad-hoc network compared to the solicitation and HELLO message based algorithms. The circumstance of not decreased throughput caused by increased protocol overhead is one main benefit of the HELLO message based algorithm and it is proven with the results of the protocol overhead.



(a) Throughput with low node density

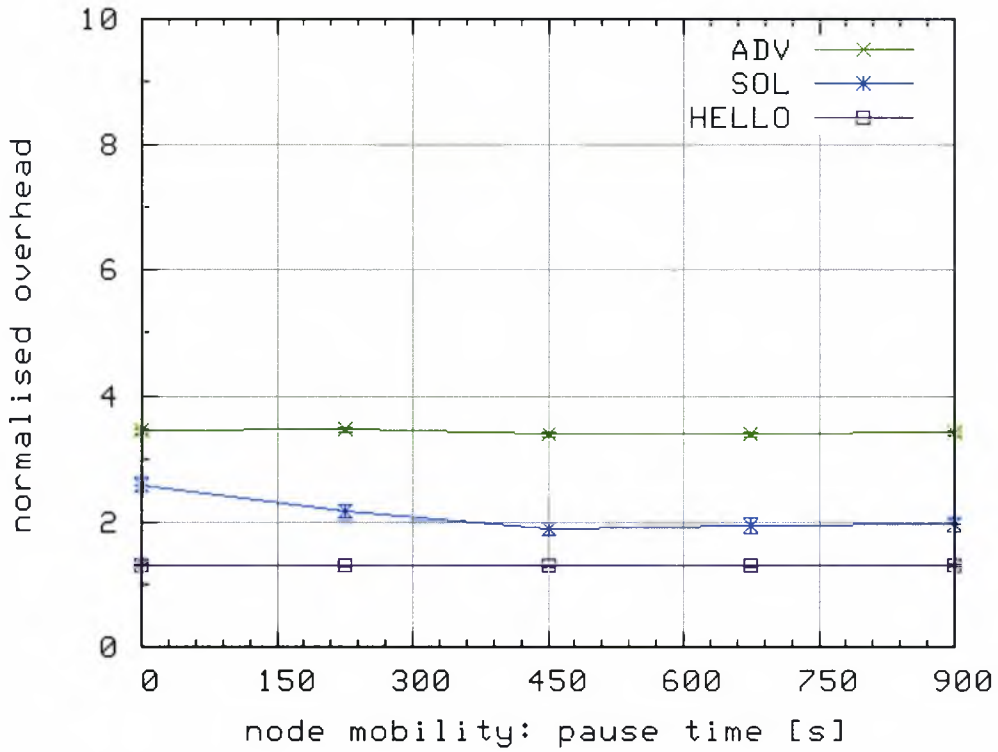


(b) Throughput with high node density

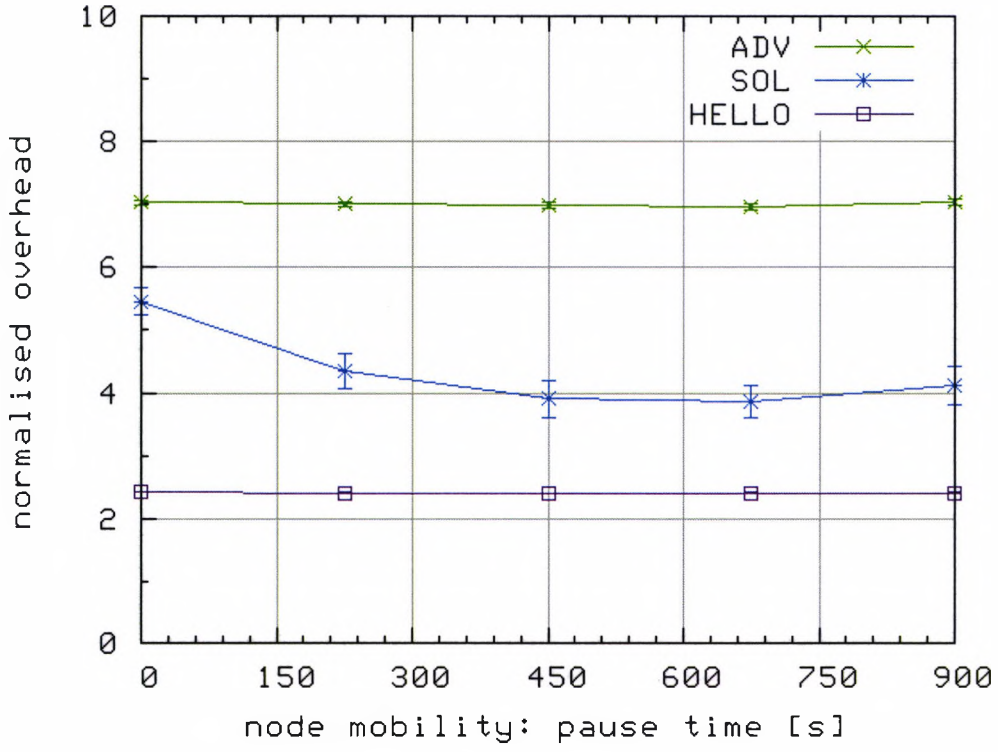
Figure 7.3: Throughput with low and high node density and the nodes' mobility as parameter

The reason for the higher throughput provided by the HELLO algorithm in general is explained with its omitting of network-wide floodings. To prove this, in Figure 7.4(a) the protocol overhead for all three algorithms related to a network size of 30 nodes is given. It can be observed that the advertisement based algorithm causes the most overhead since the algorithm permanently floods the whole ad-hoc network (in this case two gateways flood the ad-hoc network) periodically. Thus the normalised overhead of sent control messages is about 3.4. The solicitation based algorithm at a pause time of 0 seconds generates a total normalised protocol overhead of 2.6. This decreases to 2.0 for a pause time of 900 seconds. This decrease is since the higher the nodes' mobility (pause time = 0 seconds) the more the nodes lose connectivity to the Internet gateway and have to re-discover the Internet gateway by flooding the ad-hoc network with solicitations (RREQ_I). The HELLO message based algorithm causes the least overhead and since it utilises only HELLO messages the control overhead of the HELLO algorithm can be interpreted as an offset for the two other algorithms. The normalised protocol overhead for the HELLO message based algorithm is 1.3. Note, a normalised overhead of 1.0 is generated with 30 nodes that send one HELLO message every second for 850 seconds ($\hat{=}$ 25500 messages).

The results in Figure 7.4(b) are from simulations with the increased node density of 60 nodes. The solicitation based algorithm causes more overhead compared to the HELLO message based algorithm. This is caused by the broadcasting and re-broadcasting of solicitations for Internet connectivity. Additionally, broadcast messages in the advertisement based algorithm where both Internet gateways flood the ad-hoc network periodically causes very much overhead. Therefore the results show a normalised overhead of 7.0 for the advertisement based algorithm while the HELLO algorithm generates only 2.4 times the normal overhead for the whole simulation time of 850 seconds. The difference of 4.6 is then caused by both gateways and the forwarding of the advertisement messages by all attending mobile nodes. With a look at the solicitation based algorithm it can be observed that the normalised protocol overhead decreases from 5.5 to 4.1 when the nodes become more static, i.e. with increasing pause time because, if nodes are more static they do not loose connectivity to other nodes and therefore to the gateway not as often as a network of high node mobility.



(a) Overhead with low node density



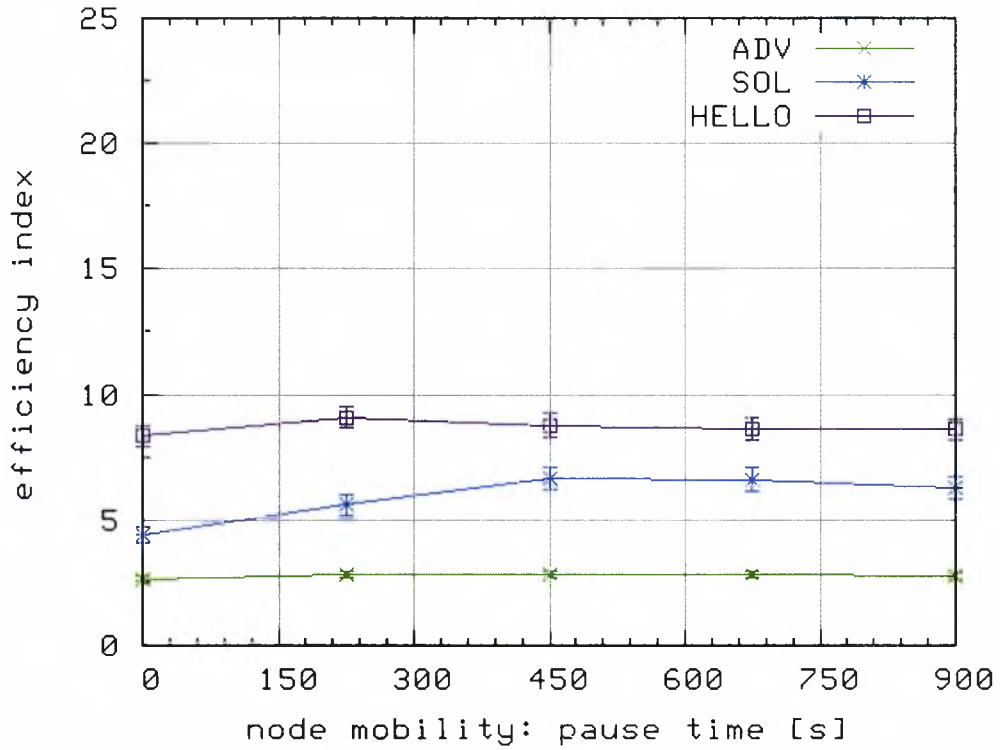
(b) Overhead with high node density

Figure 7.4: Overhead with low and high node density and the nodes' mobility as parameter

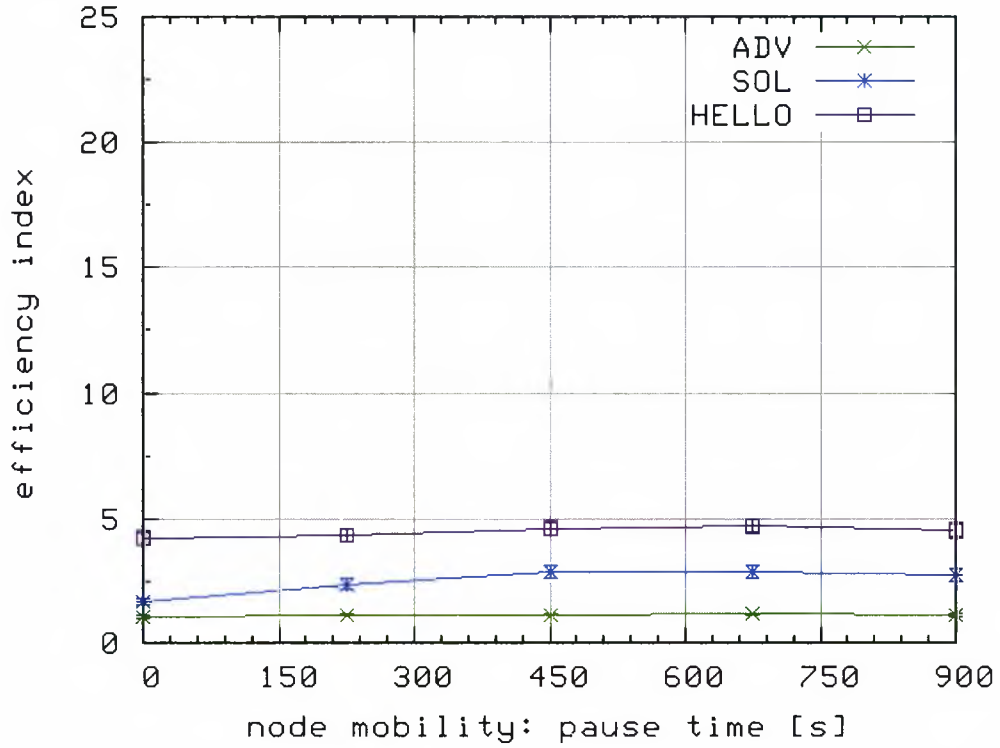
In general, with the increased number of network nodes the normalised protocol overhead increases for all three investigated Internet gateway discovery algorithms. This is firstly driven by the increased (doubled) number of sent HELLO messages for all three investigated algorithms and secondly, by the number of forwardings of broadcast messages. Additionally, with the increased number of nodes the registering of the MN with the home agent in the Internet may fail due to network congestion caused by flooding. This leads to the re-sending of MobileIP messages that additionally increase the protocol overhead. The solicitation based algorithm shows no flat graph for the protocol overhead like the both proactive algorithms do. This is the influence of the nodes' mobility. With increased node mobility (pause time = 0 seconds) routes fail more often and route failures cause re-broadcasts of route request messages and gateway solicitation requests.

The protocol efficiency index (equation 7.2) with 30 mobile nodes is depicted in Figure 7.5(a). The HELLO algorithm shows the best protocol efficiency index since it provides the most bandwidth to the testing MN while causing the least overhead as shown in Figures 7.3(a) and 7.4(a) and discussed above. The advertisement based algorithm performs the worst since it causes much more overhead compared to the two other algorithms. The solicitation based algorithm shows a moderate index. Additionally, it can be observed that no algorithm shows significant dependency on the nodes' mobility or the pause time except the solicitation based algorithm at a very high node mobility of 0 seconds pause time. This is explained with the frequent loss of connectivity due to the high node mobility and therefore frequent re-connections of the solicitation based algorithm which causes network wide floodings.

With the increased number of ad-hoc mobile nodes the protocol efficiency index decreases for all investigated gateway discovery algorithms compared to the results with low node density. The results are depicted in Figure 7.5(b). For the advertisement based algorithm it decreases from 2.7 to 1.0. The efficiency index for the solicitation based algorithm is almost the half of the results with the lower node density. Again, the HELLO algorithm shows the best index but like the other algorithms its index decreases from 9 to about 4.5. The efficiency index decrease of all three investigated discovery algorithms with the increased number of nodes is explained with the clear



(a) Efficiency index with low node density



(b) Efficiency index with high node density

Figure 7.5: Efficiency index with low and high node density and the nodes' mobility as parameter

increase of control packet overhead and a slightly decreased provided bandwidth to the nodes. Thus, the impact of the control overhead is greater than the impact of the provided throughput.

7.4.2 Impact of Node Mobility

The mobility of the ad-hoc mobile network nodes, parameterised in the pause time, does not play a significant role. This can be observed in the results above as the graphs are almost flat. The reason for this is that in the moment the MN is switched on the ad-hoc mobile network can be interpreted as a static network (snap shot). Thus the register time is constant with varying pause time. The throughput, the protocol overhead, and therefore the protocol efficiency index are not influenced by the nodes' mobility, too. The throughput is taken by the simulation of a test file transfer of 1 MB in size. This test file size was chosen since 1 MB is enough not to let the download be finished too fast and small enough to be finished within the simulation time of 900 seconds. The protocol overhead is taken by counting the control message overhead for the total simulation time of $t_{\mathrm{SIM}} = 850$ seconds. Since the resulting graphs of the throughput and the protocol overhead are the basis for the protocol efficiency index the graph for the protocol efficiency index is almost flat, too.

The nodes' mobility has a significant influence only for the solicitation based algorithm. This influence is due to the approach the solicitation based algorithm works with. Using proactive algorithms (advertisement or HELLO message based) the MN can update the route to the Internet gateway frequently but when using a reactive algorithm (solicitation based) the MN utilises an already discovered route as long as this route is up. Just in the moment the route to the Internet gateway breaks (nodes' mobility) the MN broadcasts new solicitation messages to re-discover an Internet gateway. This broadcast to re-discover the Internet gateway is not in the proactive algorithms thus the solicitation based algorithm suffers in terms of protocol overhead and throughput (and therefore for the efficiency index too) from the nodes' mobility and the pause time, respectively.

Protocol	Size [Bytes]
G.711 A-Law	160
RTP	12
UDP	8
IP	20

Table 7.3: Composition of a VoIP packet

7.4.3 Impact of Background Traffic

In general, the background traffic is expected to have a negative impact on the performance of all investigated gateway discovery protocols. This thesis uses Voice over IP (VoIP) connections as traffic background and these VoIP connections are set-up within the ad-hoc network, i.e. there are sending mobile ad-hoc nodes connected via ad-hoc multihop routes to receiving mobile ad-hoc nodes whereas the sending and receiving nodes are moving around randomly. In [65] the authors suggest a VoIP packet rate of 50 packets per second. With the G.711 codec the data rate of one half-duplex VoIP connection is 64 $\frac{\text{kbit}}{\text{s}}$ (uncompressed). With the IP, the UDP, and the RTP [65] headers (total 40 Bytes $\widehat{=}$ 320 bits) the total bandwidth requirement increases to 80 $\frac{\text{kbit}}{\text{s}}$ since 320 bits are sent 50 times a second ($\widehat{=}$ 16 $\frac{\text{kbit}}{\text{s}}$).

Every full-duplex VoIP connection in this thesis generates 50 CBR data packets per second with a size of 200 Bytes each and thus equals a data rate of 160 $\frac{\text{kbit}}{\text{s}}$. Table 7.3 gives an overview how VoIP packets are composed. The simulation parameter for background traffic is the number of simultaneously established full-duplex VoIP connections, i.e. the multiple of 160 $\frac{\text{kbit}}{\text{s}}$. Note, the VoIP application of NS-2 used in this thesis supports no silence suppression. Simulating silence suppression is not the aim of the thesis but, the investigation of the background traffic's influence to Internet gateway discovery is the aim of the thesis.

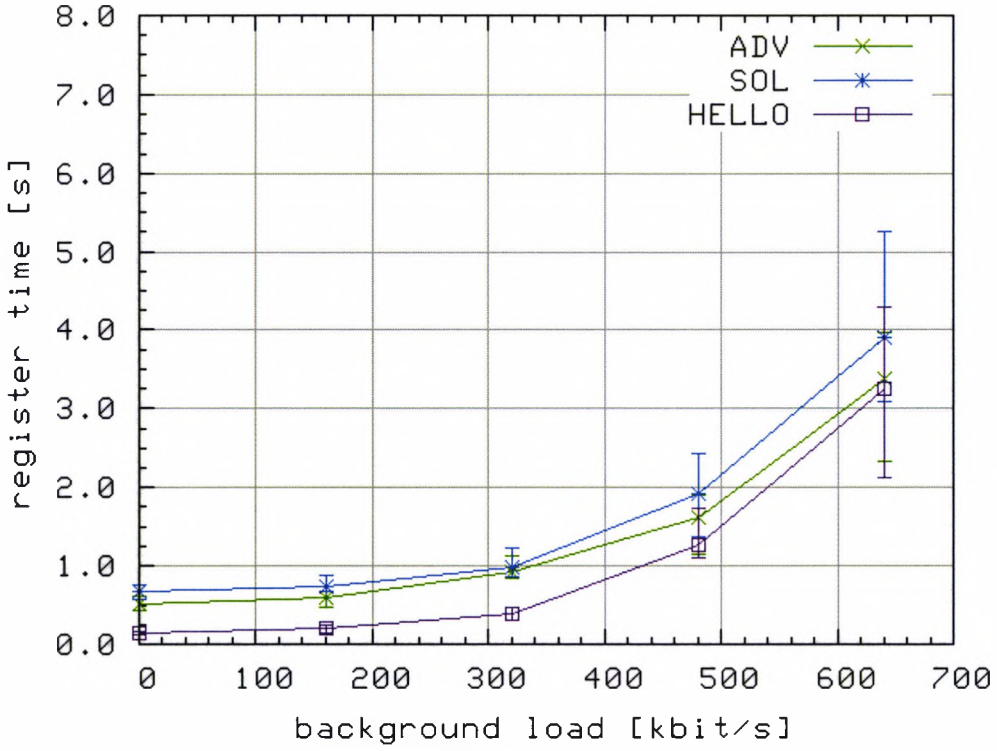
The reason for setting up background traffic as VoIP traffic is that a CBR/UDP connection can be adjusted to a specific traffic rate. This is in opposition to a FTP/TCP connection. TCP always tries to maximise the throughput of a connection while with CBR the throughput of a connection is constant and adjustable. Nevertheless, FTP/TCP connections are simulated to investigate the algorithm's behaviour with respect to the provided bandwidth to the test mobile node MN. The simulation results with the FTP/TCP background traffic are presented and discussed on page 146. In

Figure 7.6 the simulation results for 30 nodes (lower node density) with a fixed pause time of 450 seconds (node mobility) and varying background traffic is depicted. A pause time of 450 seconds is chosen since in section 7.4.1 the pause time was rated insignificant and a pause time of 450 seconds is the mean value between a high and a low node mobility.

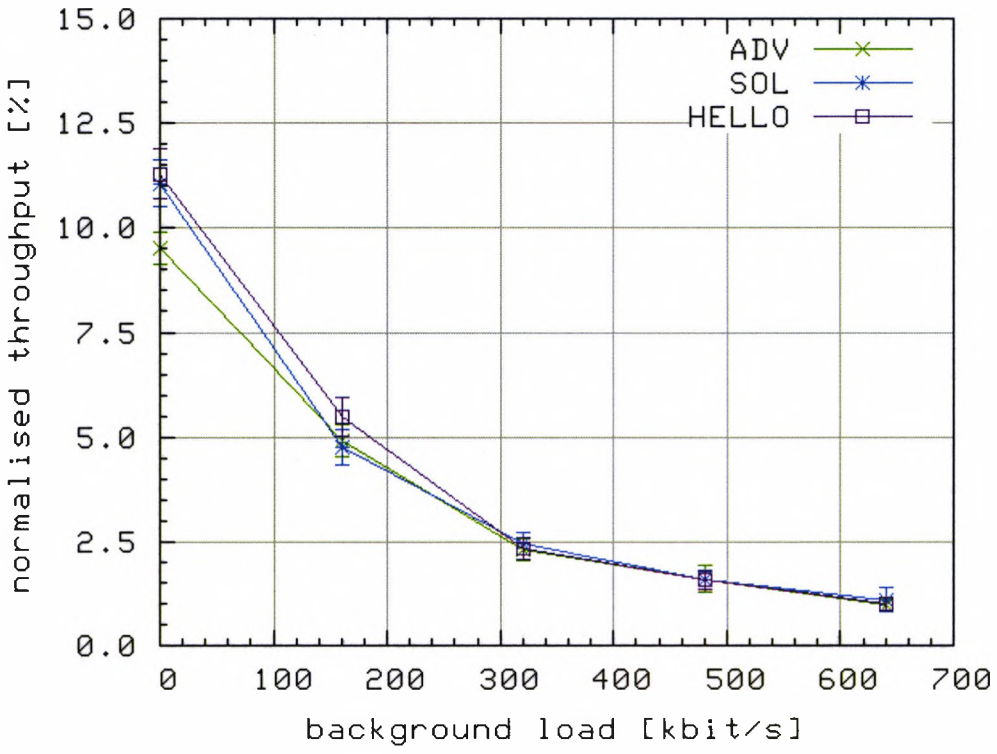
In general, the background traffic in the ad-hoc network has a negative influence on all evaluated algorithm characteristics. The register time of a mobile node in the ad-hoc network increases dramatically with increasing network traffic from less than 1 second to a maximum of 3-4 seconds. This is since the background traffic load firstly prevents the mobile nodes from discovering an Internet gateway by congesting the ad-hoc network and secondly the background traffic load delays the MobileIP message exchange and even prevents the mobile node to register with the home agent in the Internet successfully. The HELLO message based gateway discovery algorithm is least influenced by the background traffic. This is explained with the frequent re-freshing of the default and the gateway routes whereas the other algorithms cause network wide floodings. For a background traffic rate of 640 $\frac{\text{kbit}}{\text{s}}$ the simulated results are very close to each other so that there is no statistical statement possible. This can be observed when looking at the simulation results as the confidence intervals are overlapping.

The provided throughput for all three evaluated algorithms is comparable whereas the advertisement based algorithm provides less bandwidth when no background traffic is set up. The graph of the bandwidth starts at 9.5% and decreases non-linearly to 1%. The disadvantage of the advertisement based algorithm is explained with its periodic flooding strategy. The HELLO message based algorithm's throughput for no background traffic is 11% and decreases to 1% like the other algorithms.

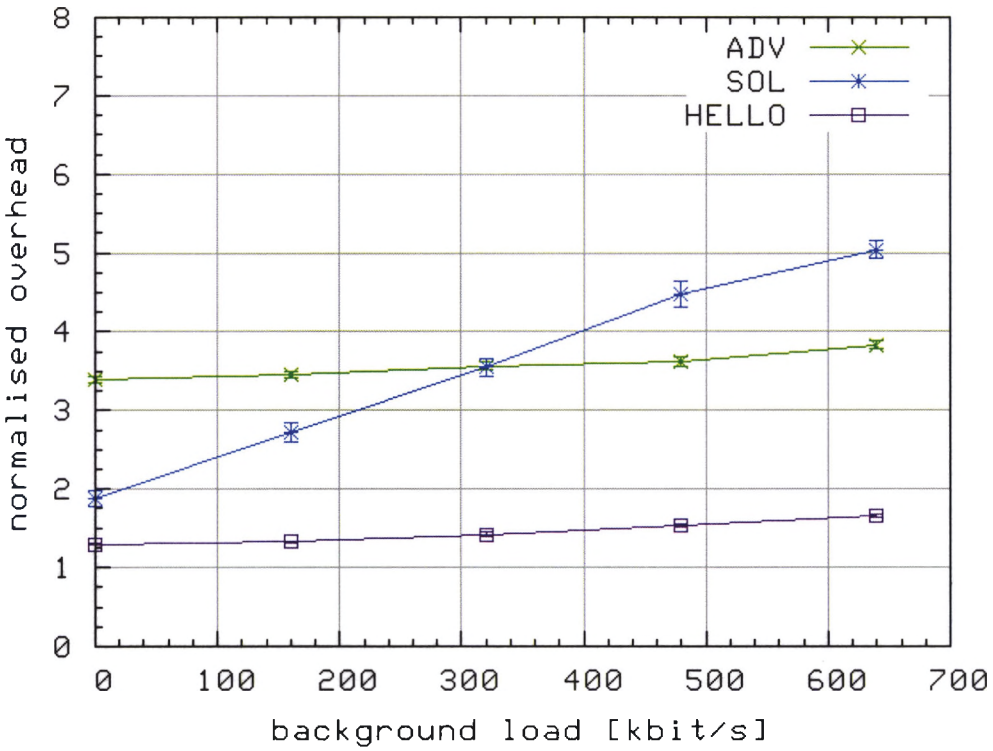
Looking at the protocol overhead it can be observed that for all three investigated discovery algorithms the overhead increases with increasing network traffic load. This increase is much more dramatic in the solicitation based algorithm compared to the both proactive algorithms (increase from 1.9 to 5 times the standardised overhead). If a mobile node cannot connect to an Internet gateway the node floods the ad-hoc network with a gateway solicitation request. With increasing background traffic more solicitation request/reply procedures fail and thus unconnected nodes solicit again.



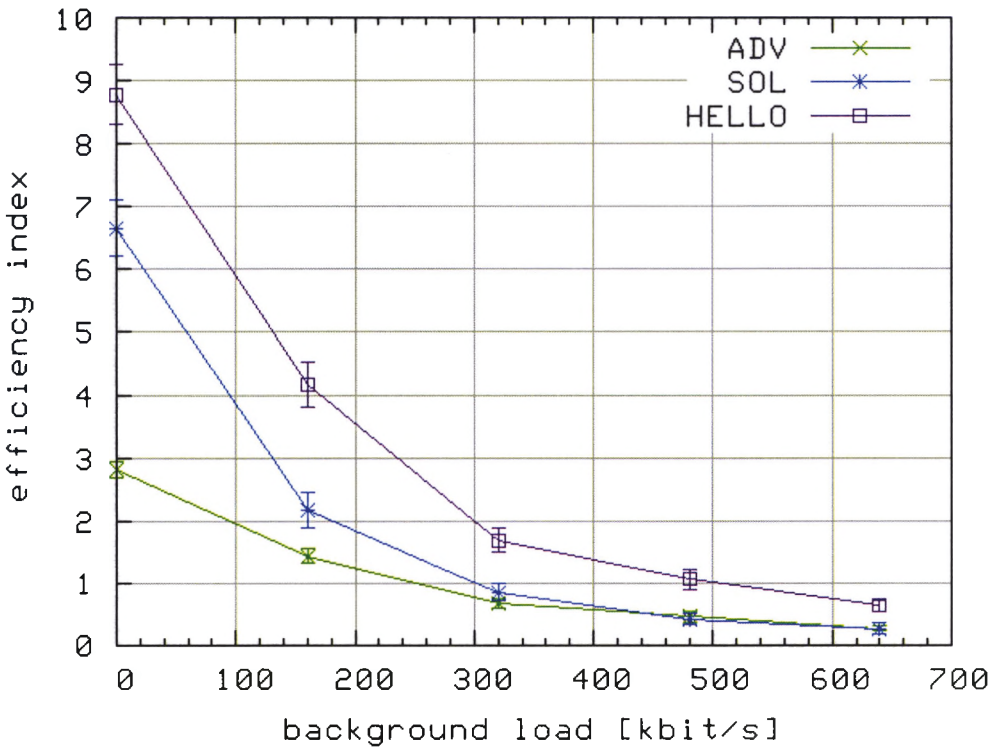
(a) Register time



(b) Throughput



(c) Overhead



(d) Efficiency index

Figure 7.6: The HELLO algorithm with background traffic as parameter

This applies for all nodes within the ad-hoc network and thus the protocol overhead for the solicitation based algorithm does not scale well with network traffic load which is indicated by a steep resulting graph of the protocol overhead for the solicitation based algorithm.

Ad-hoc nodes using one of the both proactive algorithms are mostly unimpressed by the increased background traffic in terms of their protocol overhead. In a proactive algorithm an ad-hoc network node will receive information about an existing Internet gateway from its surrounding neighbour nodes (either by advertisements or by HELLO_I messages) and if one advertisement or HELLO_I message fails the mobile node will get information from another neighbour node very soon, i.e. within one interval period of 1 second (advertisement based algorithm) or the quotient of the interval time and the direct one-hop neighbours of a node (HELLO message based algorithm). Thus proactive discovery algorithms are more resistant to background traffic than reactive algorithms.

The protocol efficiency index of the simulations with additional background traffic is depicted in Figure 7.6(d). Again, due to the minimised control overhead and the amount of provided bandwidth the HELLO message based algorithm shows a good performance index. The advertisement based algorithm and solicitation based algorithm show less performance index due to their flooding approaches when discovering Internet gateways.

TCP Background Traffic When evaluating the provided bandwidth to a specific ad-hoc mobile node by simulating a test file download the background traffic load of the ad-hoc network cannot be adjusted precisely when using FTP/TCP as background traffic since FTP/TCP background traffic can only be switched on and off. With a CBR/UDP connection it is possible to adjust the network background load in steps and therefore CBR/UDP is qualified for evaluating the algorithms performance in terms of the background traffic load. Additionally, to stress the ad-hoc network and to investigate the ad-hoc network's behaviour FTP/TCP background traffic is set up.

FTP/TCP connections are set up within the ad-hoc network to investigate the influence of a background TCP file transfer to the provided bandwidth to the MN. The background TCP connection is started at $t_{\mathrm{SIM}} = 50$ seconds, i.e. 50 seconds before the

algorithm	throughput [%] (1 FTP)	throughput [%] (2 FTP)
ADV	5.1 ± 0.4	3.7 ± 0.3
SOL	5.4 ± 0.4	3.7 ± 0.3
HELLO	5.6 ± 0.5	3.7 ± 0.4

Table 7.4: Bandwidth with FTP/TCP background traffic

test file download from the CN to the MN starts and ends when the simulation run ends at $t_{\mathrm{SIM}} = 900$ seconds.

The results show a significant difference in the provided bandwidth to the MN when using the advertisement, solicitation, or HELLO message based gateway discovery algorithm if there is one or two FTP/TCP background transfers. Table 7.4 gives the results for the low node density of 30 nodes in $1000 \times 800\mathrm{m}^2$. The results are based on 250 simulation runs. This number of simulation runs is necessary because of the randomised topology and as a basis for the statistical evaluation.

A provided bandwidth of 5% of the standardised bandwidth equals the same provided bandwidth with one full-duplex VoIP background connection and therefore has the same bad influence on the provided bandwidth to the MN. Note, the size of the data packets are different for the CBR/UDP and the FTP/TCP connection.

7.4.4 Conclusions on the HELLO Algorithm

The results above firstly prove the functionality of the HELLO message based Internet gateway discovery algorithm. Secondly, it is proven that the HELLO message based algorithm causes least control overhead compared to the advertisement and solicitation based algorithms as expected. This decreased control message overhead leads to more provided bandwidth to mobile nodes in the ad-hoc network. Thirdly, when downloading a test file of 1 MB in size the random mobility of ad-hoc network nodes does not play an important role for the throughput provided to the MN except for the solicitation based algorithm with high node mobility.

In general observations show that with the increased node density the resulting graphs are more steady than the graphs of the results of the low node density. This is since the low node density is close to the lower border of a reasonable ad-hoc network.

It can be concluded that the number of attending network nodes in general cause a linear increase of control message overhead. The increase of control message overhead

leads to a significant decrease of the provided bandwidth for the advertisement based algorithm. The HELLO message based algorithm is least impressed with the increased number of ad-hoc network nodes and therefore, shows the highest efficiency index. The solicitation based algorithm's results are located between the advertisement and the HELLO message based results.

In general, the background traffic decreases the provided bandwidth by Internet gateways to mobile nodes due to network congestion. Since the HELLO message based algorithm resigns of network wide flooding less bandwidth is consumed for control overhead and more bandwidth is provided to the mobile nodes. Additionally, proactive Internet gateway discovery algorithms are more resistant to background traffic within the ad-hoc network. This can be observed when looking at the dramatically increased protocol overhead observed for the solicitation based algorithm.

A provided bandwidth of 5% to the MN equals a background traffic load of one full-duplex VoIP connection. With the given simulation results above one FTP/TCP connection has the same bad impact to the provided bandwidth to the MN as one full-duplex VoIP connection. The CBR/UDP connection has a packet size of 200 Bytes whereas the FTP/TCP file transfer utilises a packet size of 1000 Bytes per packet and is therefore much more effective in terms of throughput. Additionally, the route lengths of the FTP/TCP connection and the CBR/UDP connection are not equal. Refer to Appendix A for a discussion about a route's length and the packet size of a transmission to the estimated throughput. One must be careful when comparing bandwidths of multihop routes with different traffic types and different packet sizes.

Next, the benefit the two proactive Internet gateway discovery algorithms which are either based on advertisements or HELLO messages, gain from the gratuitous route reply extension are evaluated. This extension detects the change of ad-hoc connectivity to an Internet gateway selected by mobile nodes using a proactive gateway discovery algorithm and generates additional control messages if such a change has been detected.

7.5 Analysis of Gratuitous Route Reply Extension

If the route from a mobile node to the gateway changes in terms of the number of hops or the next hop entry pointing to the gateway the extension sends an information

message called gratuitous route reply (GRREP) message to the selected gateway. This GRREP is acknowledged by the gateway with a gratuitous route reply acknowledge message (GRREP-ACK). The GRREP and the GRREP-ACK messages ensure that for a specific route from a mobile node to a gateway all involved nodes including the gateway node can update their routing tables to all communication participants of the route. The extension only applies to the proactive gateway discovery algorithms and not to the reactive, i.e. the solicitation based algorithm. More information about the functionality principle of the GRREP extension can be found in chapter 5.

Additionally, the extension applies only if a mobile node in the ad-hoc network detects a change in the route to an already selected gateway and not to another gateway. The change to another gateway is called handover and is managed by the MobileIP protocol.

Firstly, in this section the extension is simulated with a static and determined scenario to prove the correct functionality and the benefit to the (proactive) gateway discovery algorithms. Later the scenario changes to a more realistic one with random movement of nodes and additional background traffic.

7.5.1 Determinated Setup

The determined scenario is set up as follows. One Internet gateway GW(200|700) (it is not necessary to evaluate two gateways since the extension does not apply for handovers) is located at the top of the simulation plane. Beneath the GW a number of intermediate static nodes IN_{1-6} are positioned. They are located at IN_1(150|500), IN_2(150|300), IN_3(150|100), IN_4(250|500), IN_5(250|300), IN_6(250|100). This is to test the correct sending of GRREP messages when the MN is moving toward and away from the Internet gateway and if the next hop entry in the routing table of the MN has changed. Therefore there are two columns of nodes positioned. The setup is depicted in Figure 7.7.

A testing mobile node (MN) is located at MN_{start}(200|460). This position ensures that the MN has firstly a direct (one hop) connection to the GW. The MN is switched on at a total simulation time t_{SIM} of 50 seconds to let the network enough time to establish itself and to find all needed routes for the other nodes. At $t_{SIM} = 100$ seconds

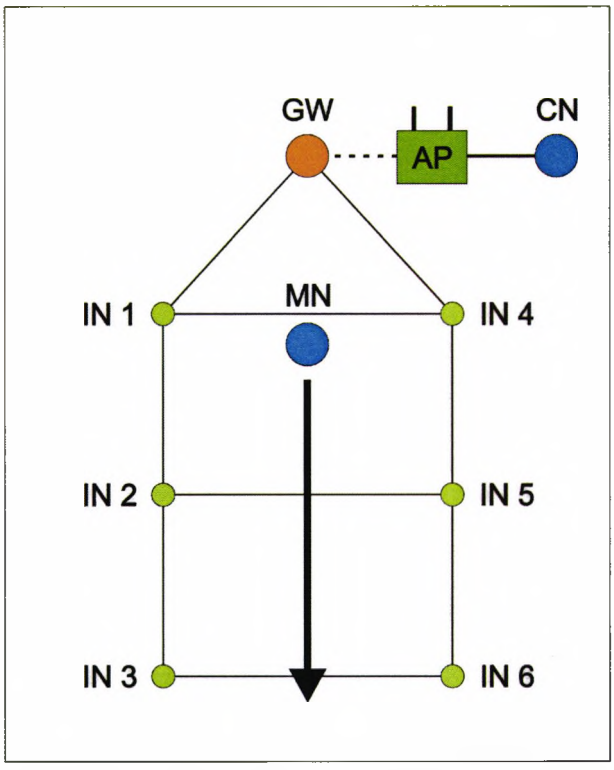


Figure 7.7: Simulation setup for the GRREP extension (determined setup)

the MN starts moving towards $MN_{stop}(200|1)$ and back. While moving the MN downloads a test file from the Internet represented by the correspondent node (CN). The performance results of the simulated determinated setup are given in Figure 7.8. Note, the normalised throughput is now related to the provided throughput of the solicitation based algorithm that is defined as 100%. This is to compare the algorithms in general. Further, Figure 7.8 depicts two histograms for the advertisement and the Hello message based algorithms each, one without enabled GRREP extension and one with enabled GRREP extension for comparison. 50 simulation runs were performed.

The provided bandwidth of the solicitation based gateway discovery algorithm is less compared to the advertisement and HELLO message based even without the enabled GRREP extension. This is since nodes using the solicitation based algorithm always need time until they detect the loss of connectivity to other nodes (equation 4.2). Additionally, if a specific MN is moving toward the Internet gateway a possible route shortening with the Internet gateway will not be performed until the node loses connectivity to its next hop neighbour of its gateway route. The same applies if a node is moving to a more distant destination away from the Internet gateway. Then the node

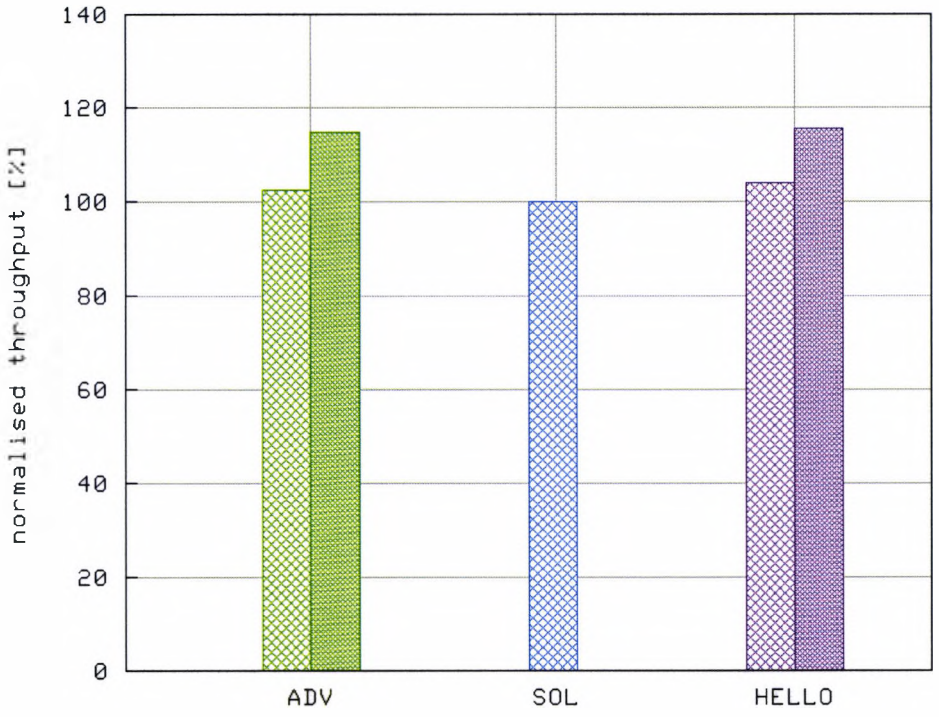


Figure 7.8: Simulation results of the GRREP extension (determined setup)

loses connectivity to its next hop neighbour, too, and has to re-discover the default and gateway routes. This is in opposition to the proactive gateway discovery algorithms. Using proactive gateway discovery algorithms a node can always find the shortest route to a gateway since routes updates are permanently possible. Since the GRREP extension does not apply for the solicitation based algorithm the results from the solicitation based algorithm are depicted for comparison and give the 100% mark and thus, the results of the proactive algorithms for the determinated scenario refer to this reference.

The bandwidth provided by the advertisement based algorithm increases from 102.3% to 114.8% with the GRREP protocol extension. For the HELLO message based algorithm an increase in the provided bandwidth was also found. Here the bandwidth increases from 104.0% to 115.6%. Note, since the results were generated using a determined scenario without movement random factors they are expected to be very accurate.

The reason for the higher provided bandwidth of the HELLO message based algorithm is discussed in section 7.4. It can be observed that both proactive algorithms benefit from the GRREP extension. This is achieved by the frequent refreshing of the

	w/o GRREP	w/ GRREP
ADV	2.68	2.54
SOL	2.95	
HELLO	2.75	2.54

Table 7.5: Mean route lengths (determined scenario)

	result (w/o GRREP)	result (w/ GRREP)	benefit
ADV	807.7 ± 38	906.2 ± 28	+12.2%
HELLO	820.8 ± 39.3	914 ± 34.6	+11.4%

Table 7.6: Benefit of GRREP extension (determined scenario)

route in the GW to the MN and the fact that shorter routes in general provide more bandwidth to an end-to-end connection.

To prove this, the mean route lengths of the discovered routes between the MN and the Internet gateway are given in Table 7.5. It can be observed that both proactive algorithms discover shorter mean routes to the gateway while the solicitation based algorithm discovers longer mean routes. Further the mean route lengths with enabled GRREP extension provide even shorter routes. Thus, the route shortening functionality of the GRREP extension is confirmed.

In Table 7.6 the absolute simulated throughput as well as the benefit of the GRREP extension to the proactive Internet gateway discovery algorithms in a determinated scenario are compiled.

7.5.2 Random Setup

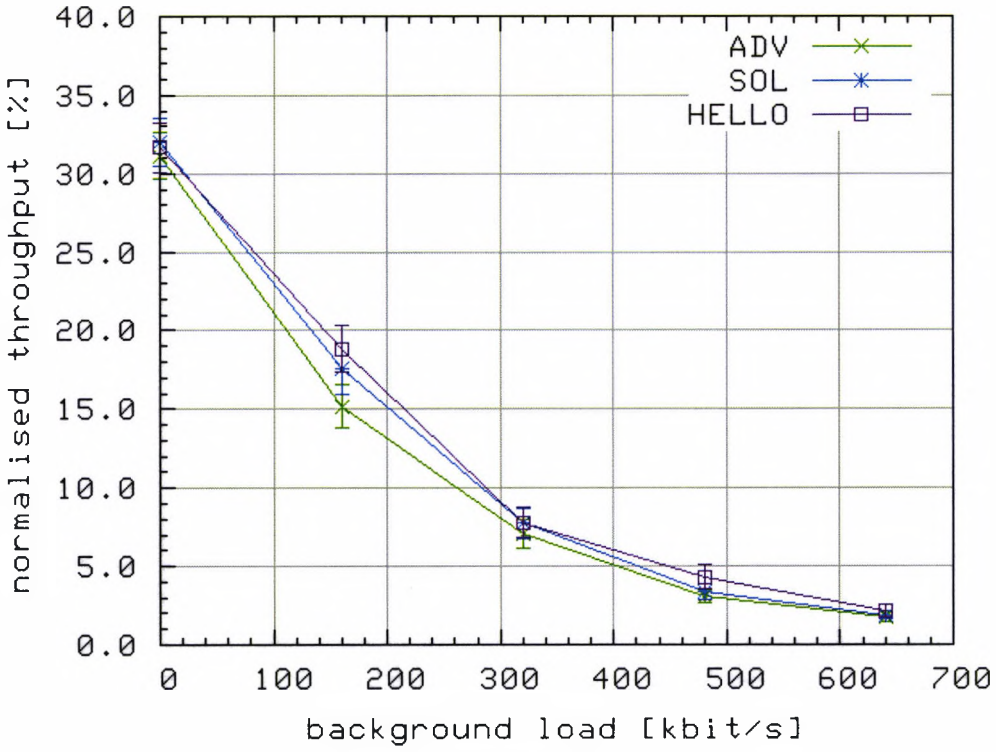
The scenario so far is for the analysis of the functionality of the GRREP extension and its benefit to both investigated proactive discovery algorithms within a determinated scenario setup. To analyse the benefit of the GRREP extension in a more realistic scenario the setup changes as follows. The ad-hoc cluster consists of 15 nodes that are moving around randomly according to the random waypoint model with a pause time of 450 seconds (medium mobility). The number of nodes equals the node density in section 7.4.1. Simulations with a varying pause time from 0 seconds (high mobility) in steps to 900 seconds (almost static) show that the mobility of ad-hoc network nodes has no significant impact on the provided bandwidth to the MN (section 7.4.2).

To analyse the benefit of the GRREP extension, the following algorithm character-

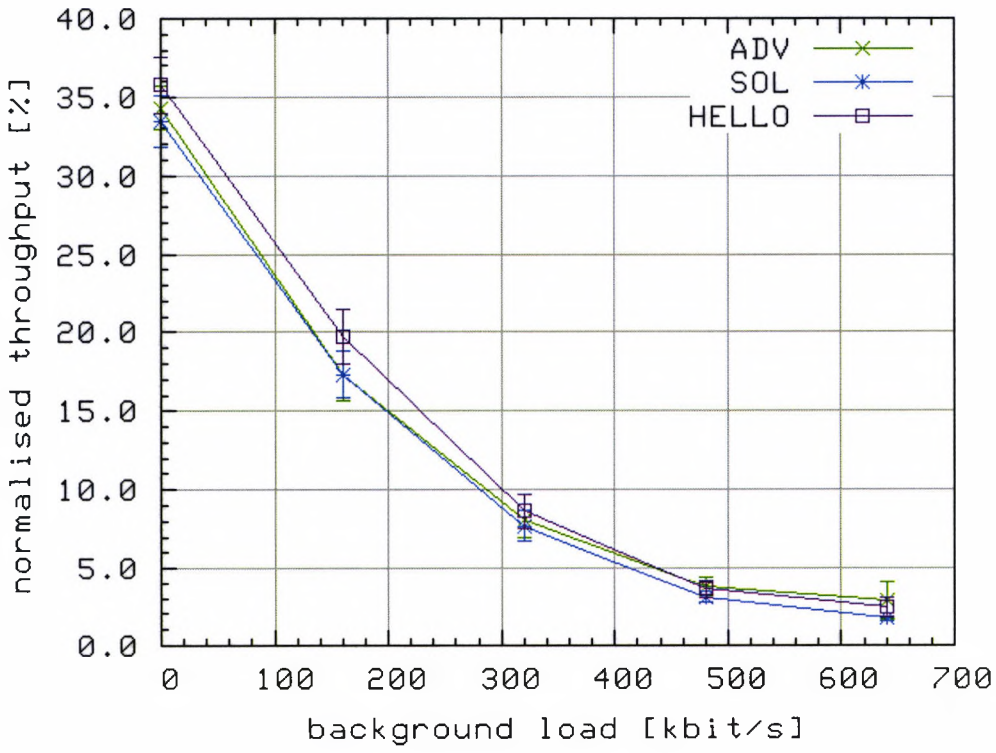
istics have been evaluated, i.e. the provided bandwidth as well as the protocol overhead and the protocol efficiency. The protocol efficiency is defined in equation 7.2. Figure 7.11 gives the mean results of 200 simulation runs. The background traffic sources are set-up parallel CBR/UDP data streams with VoIP parameters, i.e. 50 $\frac{\text{packets}}{\text{second}}$ with a size of 200 Bytes per packet per stream like in section 7.4.3. For comparison each algorithm characteristic is presented in two figures. The upper figure depicts the simulation results without enabled GRREP extension while the lower figure depicts the simulation results with enabled GRREP extension.

In Figures 7.9(a) and 7.9(b) the provided bandwidth to the testing MN is given. It can be observed that with enabled GRREP extension the provided bandwidth increases from 31.1 to 34.3 for the advertisement based algorithm. For the HELLO message based algorithm an increase from 31.7 to 35.8 was found. The increase is 9.3% and 11.4% for no background traffic. This decreases for high background traffic rates of 640 $\frac{\text{kbit}}{\text{s}}$. Obviously, the GRREP extension works well in networks with little or none background traffic. This is explained with the loss of GRREP messages and the corresponding GRREP-ACK messages due to heavy background traffic and therefore unsuccessful route updates in intermediate nodes along the route to the Internet gateway and the Internet gateway itself. Thus, high background traffic has a bad influence on the benefit of the GRREP extension to the gateway discovery algorithms.

Looking at the protocol overhead, the advertisement based discovery algorithm scales bad with increasing background traffic. Without the GRREP extension the normalised protocol overhead (incl. AODV and MobileIP messages) increases from 1.2 to 1.3. This is interpreted with the loss of connectivity between the ad-hoc nodes and the Internet gateway due to increasing traffic. A loss of connectivity means more overhead caused by MobileIP messages (binding updates and acknowledgments) and generated route error messages of AODV. With the GRREP extension the advertisement based algorithm's protocol overhead increases from 2.4 up to 2.7. Again, the additional background traffic has a very bad influence on the advertisement based discovery algorithm. The strong increase in the protocol overhead can be explained with the frequent loss of connectivity and, additionally, with the frequent unsuccessful route updates by GRREP messages. Note that all attending ad-hoc nodes perform the GRREP extension.

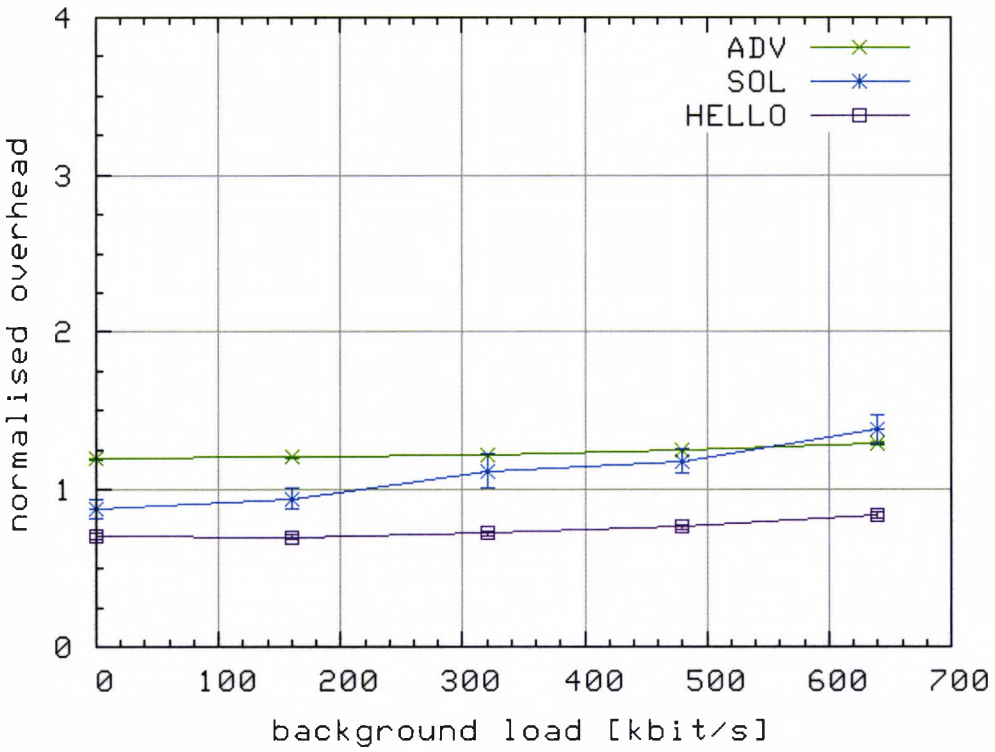


(a) w/o grrep extension

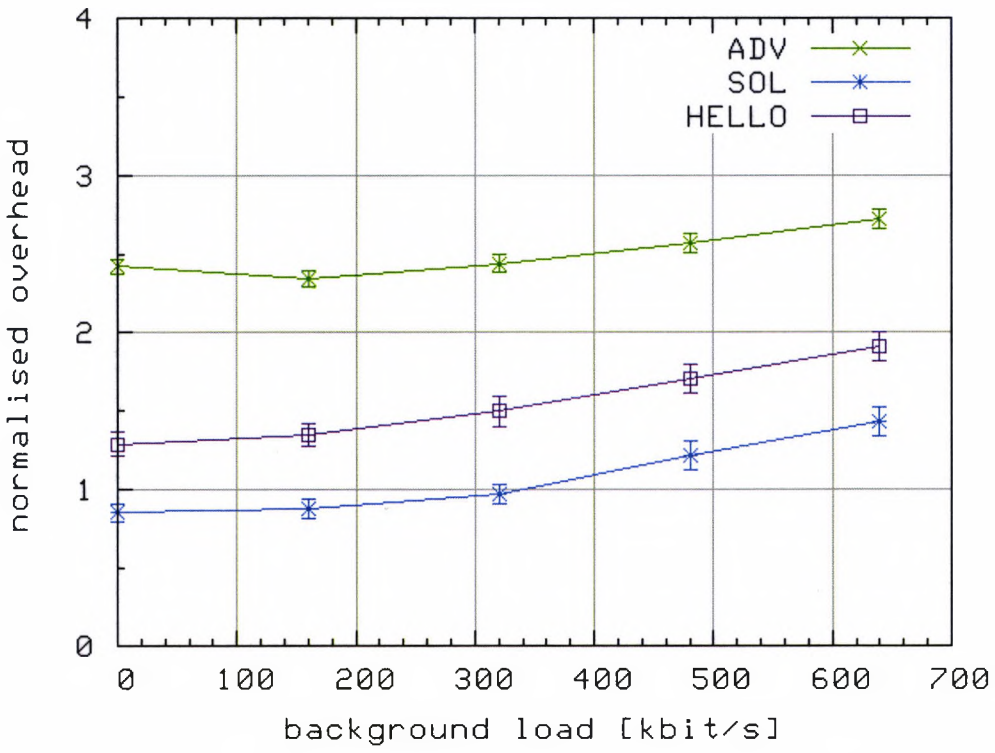


(b) w/ grrep extension

Figure 7.9: Simulation results of the GRREP extension (random setup)



(a) w/o grrep extension



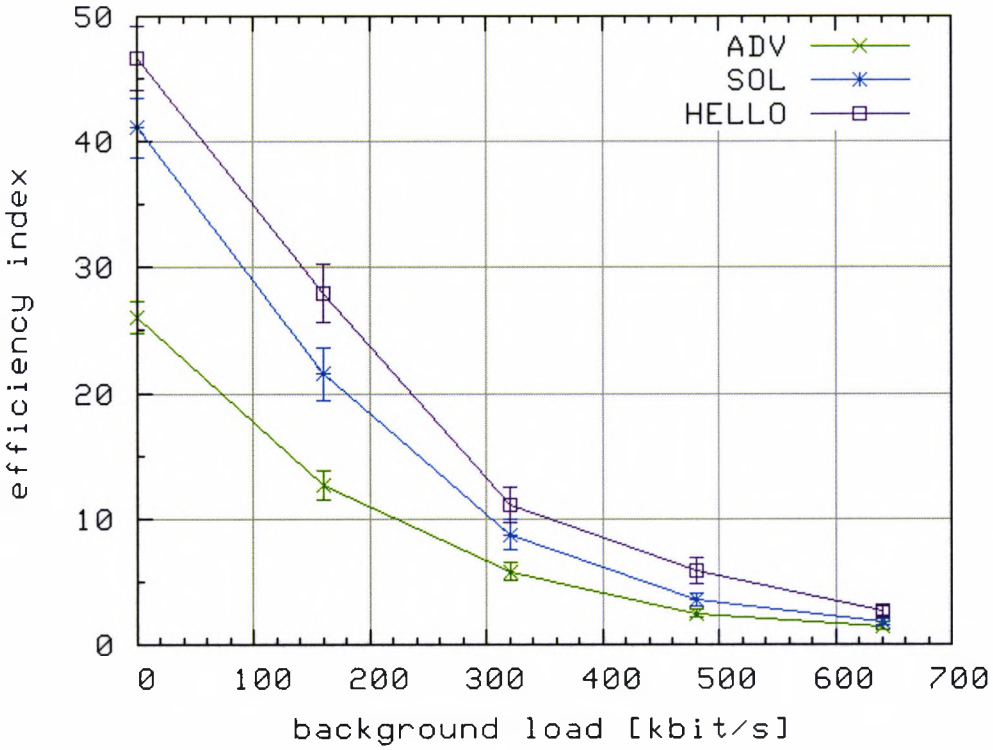
(b) w/ grrep extension

Figure 7.10: Simulation results of the GRREP extension (random setup)

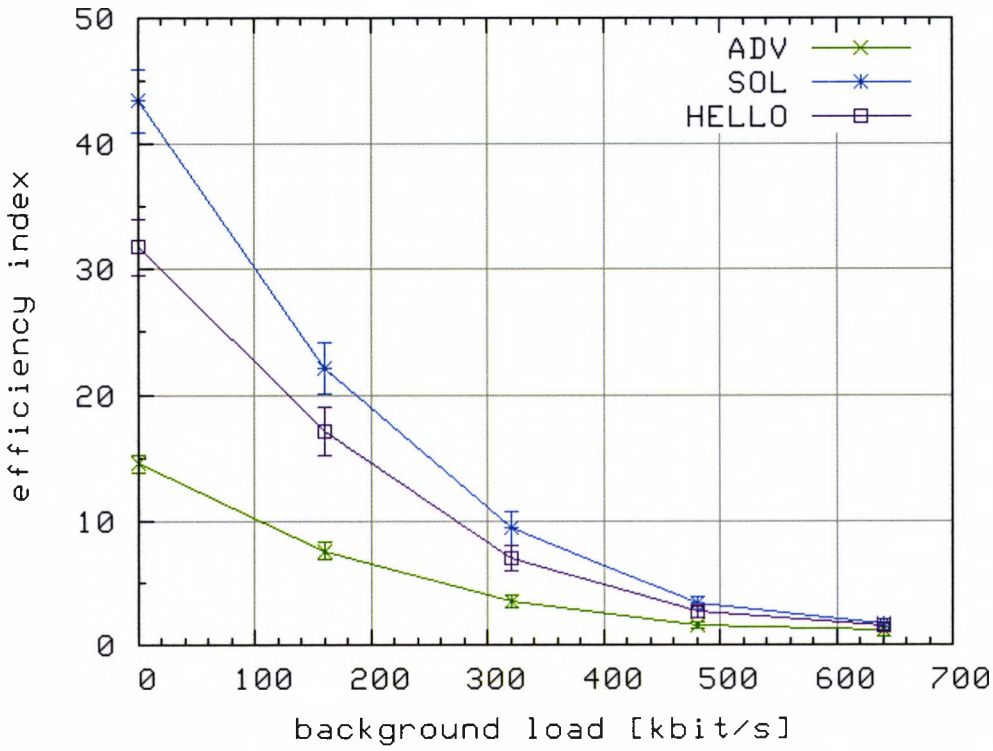
The HELLO message based gateway discovery algorithm shows the least protocol overhead as expected when the GRREP feature is not enabled. This can be explained with the attributes and characteristics of the HELLO messages based algorithm as discussed in section 7.4. When the GRREP extension is enabled the protocol overhead increases. Additionally, with increasing background traffic all resulting graphs grow. From the simulation results it can be observed that the protocol overhead increase is mostly driven by the enabling of the GRREP extension and not by the background traffic load. So the protocol overhead increases from 0.7 - 0.9 to 1.3 - 1.9 (without and with enabled GRREP extension to the HELLO message based algorithm). This fact is explained with the frequent re-sending of protocol messages of the GRREP extension. But the HELLO message based algorithm causes less protocol overhead compared to the advertisement based algorithm since it does not flood the ad-hoc network. The advertisement based algorithm's simulation results show a significant increase from 1.2 - 1.3 to 2.4 - 2.7 (without and with enabled GRREP extension to the advertisement based algorithm). Note, the solicitation based algorithm is only given for comparison reasons.

The protocol efficiency index is directly calculated from the throughput and the protocol overhead the algorithms cause. Since the amount of control messages for the advertisement based algorithm is higher compared to the HELLO message based algorithm the protocol efficiency index is low and decreases with higher background traffic. The slight increase in the provided bandwidth is not able to have a positive influence on the advertisement based discovery algorithm's protocol efficiency index. The HELLO message based algorithm's efficiency index decreases, too. This is explained with the higher amount of control messages needed to provide connectivity as depicted in Figure 7.10(b). Again the increase in the provided bandwidth is not able to increase the efficiency index value due to the much higher protocol overhead with enabled GRREP feature. For comparison the solicitation based algorithm shows the best efficiency index when the both proactive algorithms use the GRREP extensions.

If a system provides less bandwidth at layer 2 the protocol overhead plays an important role. Thus, when the gateway discovery algorithms are used with a narrowband system the GRREP extension is not feasible. When using a broadband layer 2 basis



(a) w/o grrep extension



(b) w/ grrep extension

Figure 7.11: Simulation results of the GRREP extension (random setup)

the GRREP feature is valuable. Additionally, it could be observed that the mobility of nodes does not play an important role to the discovery algorithms for downloading a test file and thus the results from simulations with node movement with the pause time as parameter are not given. This fact is discussed in section 7.4.2.

7.6 Analysis of Load Switching Extension

In this section the Load Switching extension to the Internet gateway discovery algorithms is evaluated. The extension allows mobile ad-hoc nodes to select between multiple discovered Internet gateways on the basis of quality of service constraints. Quality of service and its relation to this thesis is defined and discussed in section 2.6 on page 40. Firstly the extension's functionality is analysed using determined scenarios. The benefit of the extension is then evaluated using random scenarios. In both cases the performance benefit of the algorithm extension is taken by comparing the provided bandwidth to mobile nodes.

7.6.1 Symmetric Setup

The symmetric setup is used to prove the correct functionality of the Load Switching extension made to the gateway discovery algorithms using two gateways in the same hop distance. In Figure 7.12 a testing mobile node MN is located in a static ad-hoc multihop network. Two Internet gateways GW1 and GW2 are available for Internet connectivity. GW1 is serving up to six ad-hoc nodes with VoIP traffic (red dots). The scenario setup was chosen to stress the algorithms with background traffic. The main simulation parameter is the number of simultaneous VoIP connections i.e. the used bandwidth by the VoIP connections (one full-duplex VoIP connection equals a bandwidth of $160\frac{\text{kbit}}{\text{s}}$). Besides the number of simultaneous VoIP connections the other simulation parameter is the Internet gateway discovery algorithm with enabled or disabled Load Switching extension.

The VoIP background traffic is switched on at $t_{\text{SIM}} = 105$ seconds after the MN is switched on at $t_{\text{SIM}} = 50$. Thus the MN has the chance to discover both Internet gateways. The MN starts downloading the test file at $t_{\text{SIM}} = 100$ seconds, i.e. right before the background traffic starts. Simulation results are depicted in Figures 7.13,

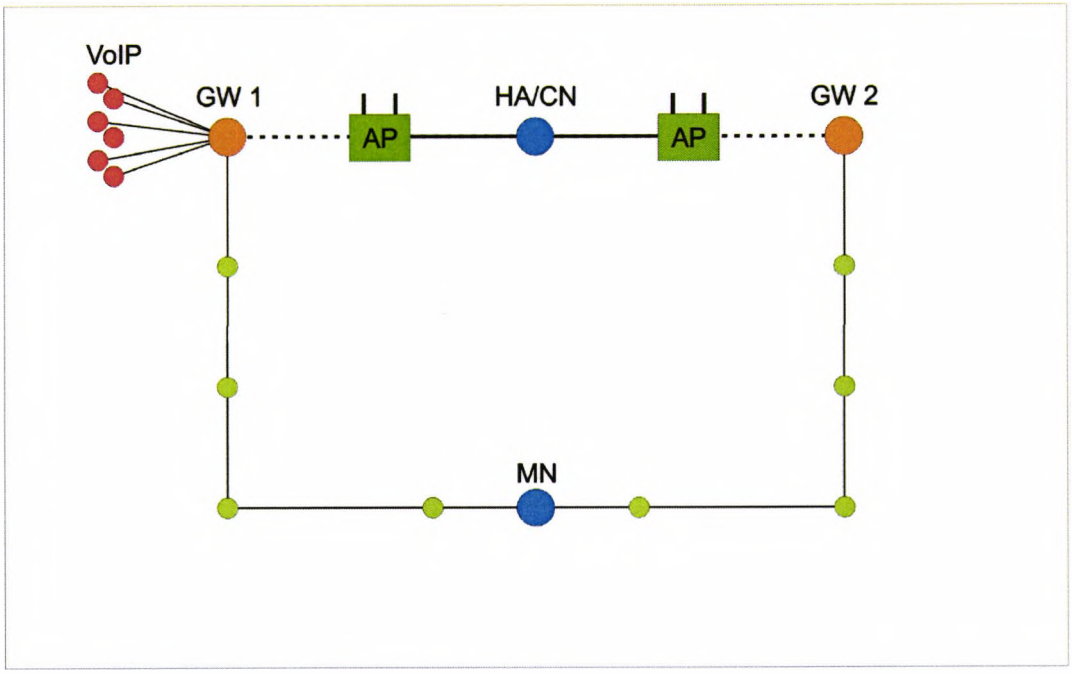
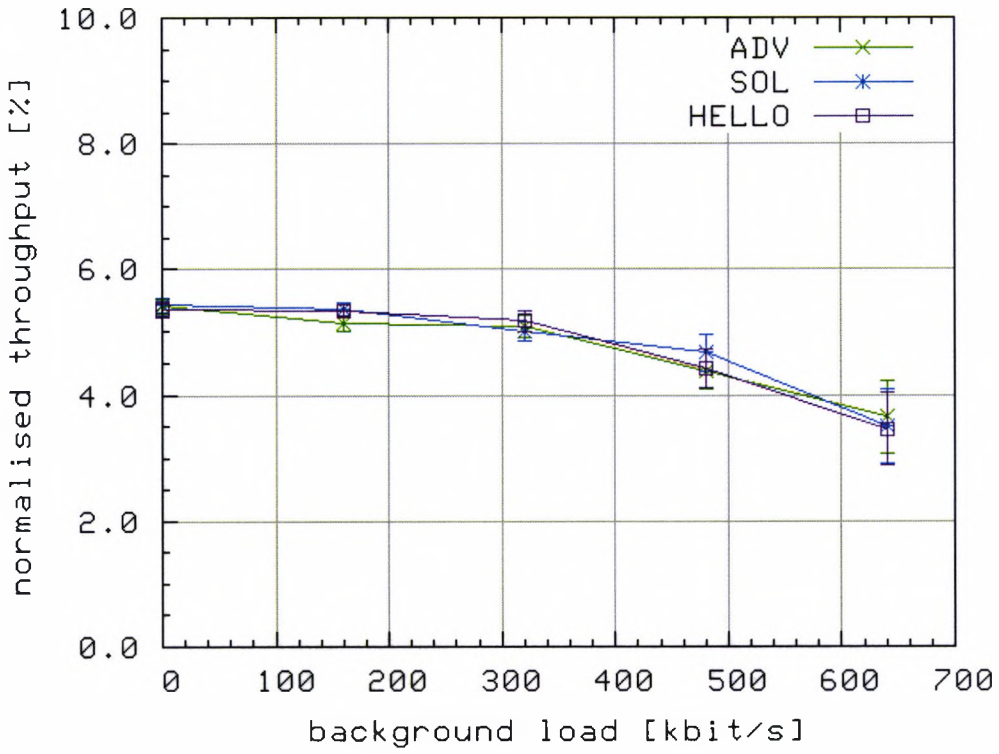


Figure 7.12: Simulation setup for the Load Switching extension (symmetric setup)

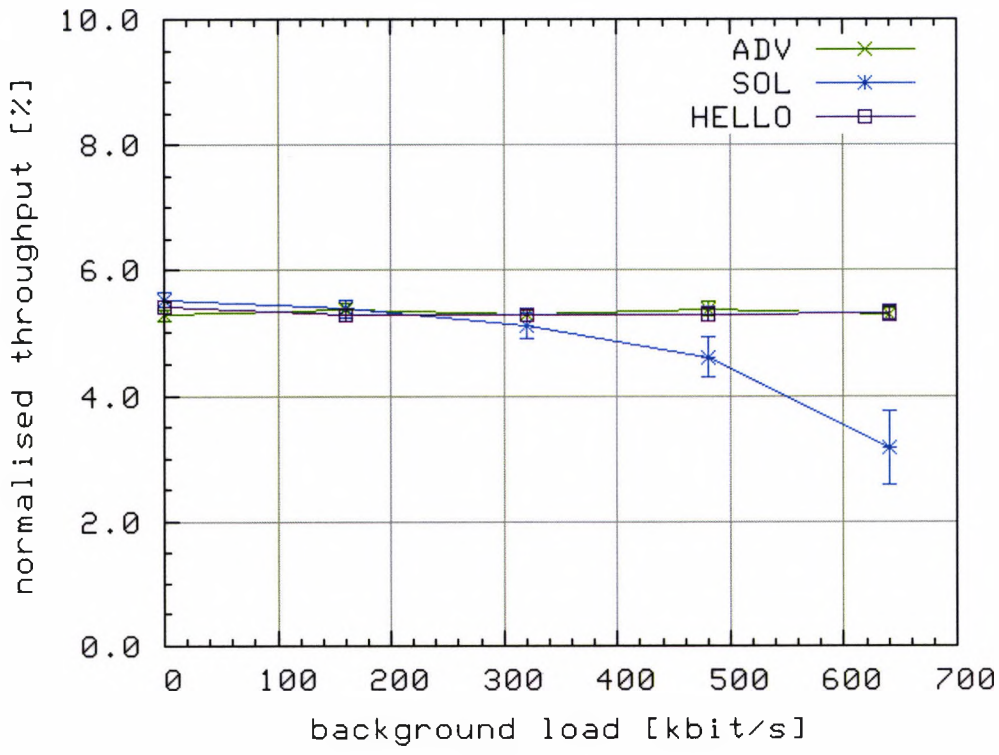
7.14, and 7.15. They only refer from the time after the MN is switched on, i.e. they are for a time span of 850 seconds.

The simulated mean throughput for the test file transfer in the symmetric setup is 5.4% of the standardised throughput (ref. to Appendix A) for no background traffic for all three gateway discovery algorithms with and without enabled Load Switching extension. With increasing background traffic the provided bandwidth decreases to 3.5% for every gateway discovery algorithm. This is due to the traffic load in GW1. In Figure 7.13(b), with enabled Load Switching extension, the provided bandwidth to the MN remains almost constant at 5.3% because the MN switches to GW2 when it detects the additional background traffic load in GW1. This is not for the solicitation based algorithm because nodes using the solicitation based algorithm are not provided with newer (sequence number) information about the traffic load within GW1 and therefore the solicitation based algorithm does not recognise the load change in GW1.

Looking at the control message overhead it can be observed that the both proactive algorithms, i.e. the advertisement and the HELLO message based algorithm, show unchanged control overhead with and without enabled Load Switching extension whereas

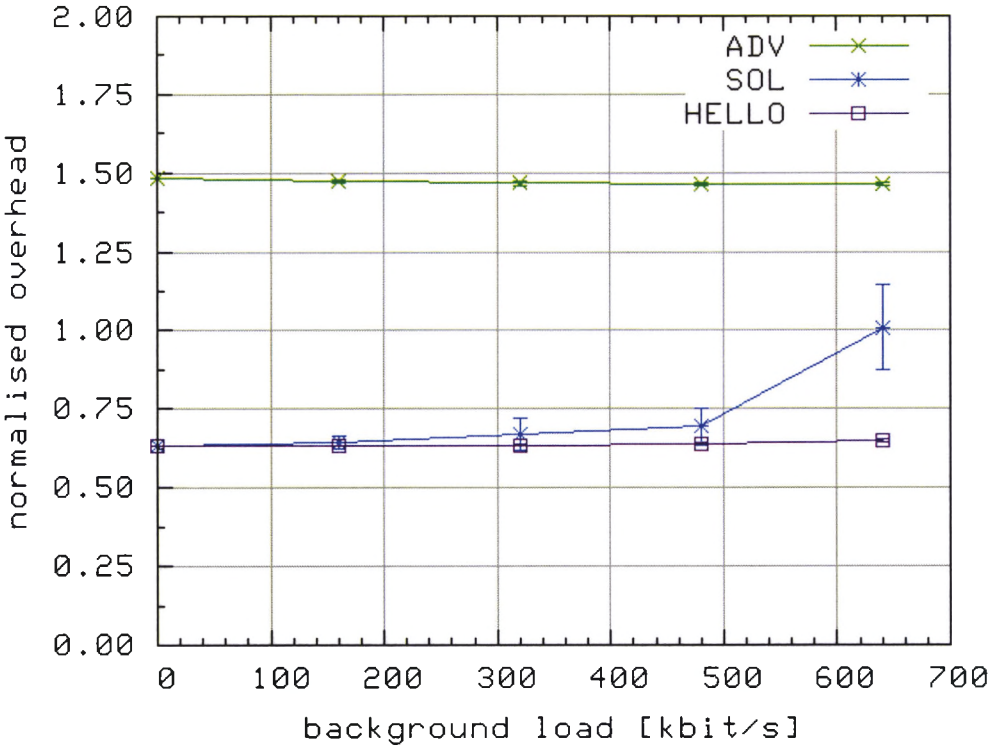


(a) w/o extension

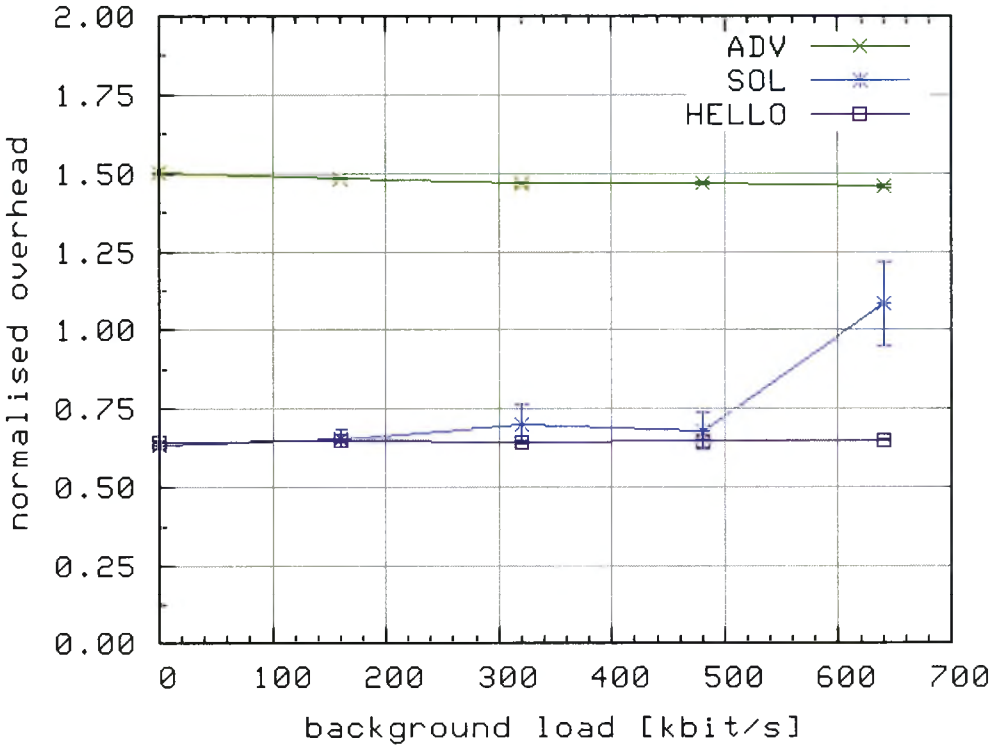


(b) w/ ls extension

Figure 7.13: Simulation results of the Load Switching extension (symmetric setup)



(a) w/o extension



(b) w/ ls extension

Figure 7.14: Simulation results of the Load Switching extension (symmetric setup)

algorithm	w/o Load Switching	w/ Load Switching
ADV	52% / 48% ± 14	7% / 93% ± 1
SOL	52% / 48% ± 14	50% / 50% ± 14
HELLO	44% / 56% ± 14	17% / 83% ± 1

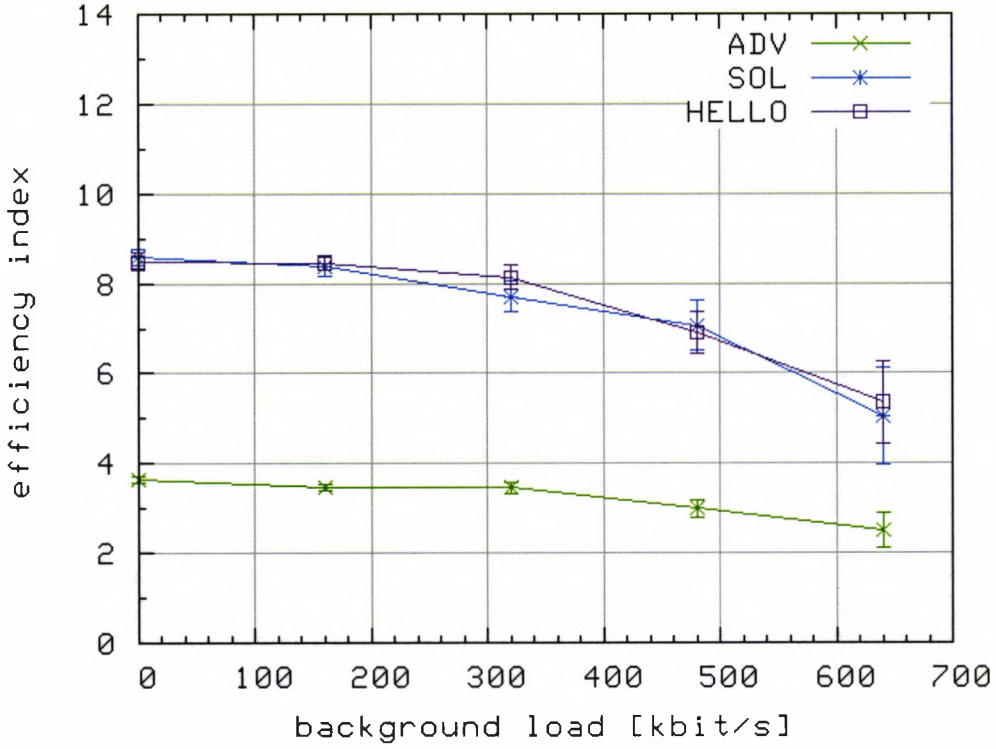
Table 7.7: Traffic distribution in symmetric setup at 320 $\frac{\text{kbit}}{\text{s}}$ background traffic

the advertisement based algorithm causes much more control overhead compared to the HELLO message based algorithm. This was also found in section 7.4 where the HELLO algorithm is investigated and it is confirmed here. Only the solicitation based algorithm causes more overhead with increasing traffic load of 640 $\frac{\text{kbit}}{\text{s}}$ in GW1. With high traffic load in GW1 the possibility of a connection loss between GW1 and the MN increases and therefore the MN broadcasts for alternative gateways which causes overhead by network flooding. Nevertheless, there is no significant change in the control message overhead found with enabled or disabled Load Switching extension.

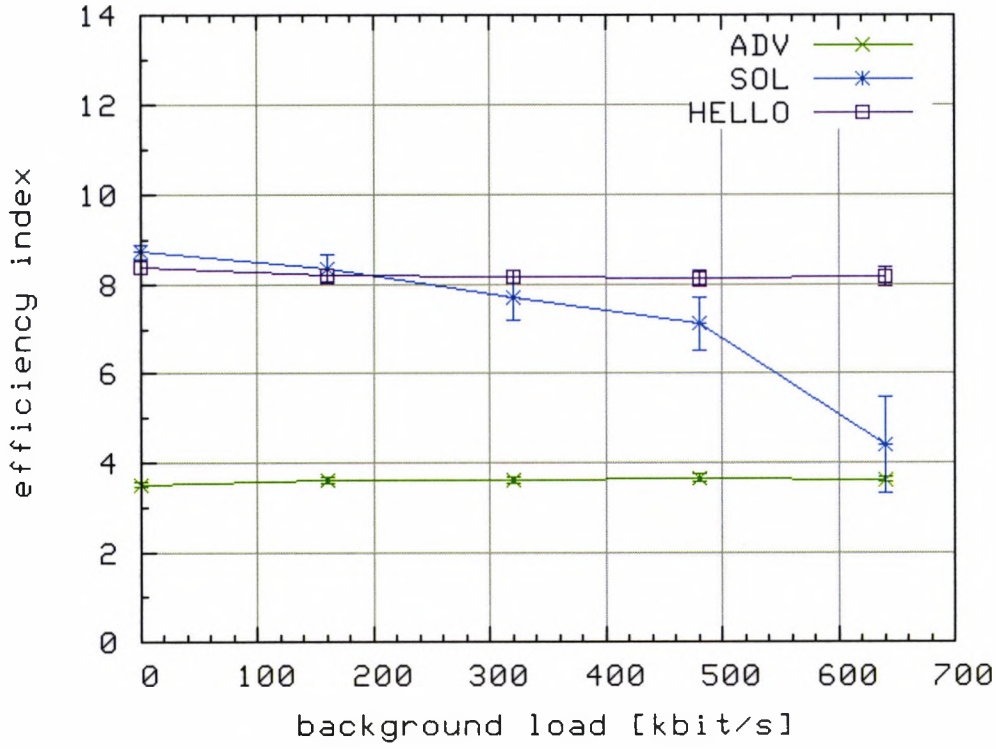
Investigating the protocol efficiency index, all investigated algorithms suffer when the Load Switching feature is not enabled whereas the advertisement based algorithm shows the worst efficiency index. With enabled Load Switching feature the efficiency index remains almost constant at 8.2 for the HELLO message based algorithms and about 3.6 for the advertisement based gateway discovery algorithm. For the solicitation based algorithm the efficiency index decreases from 8.7 to 4.4 due to the less provided bandwidth and the increased protocol overhead and the increase of background traffic.

The switching to an alternative gateway can also be observed when looking at the distribution where data packets destined to the MN are being routed via, GW1 or via GW2. In Table 7.7 this distribution is compiled. Each result stands for the amount of data packets in percent routed via one of the two gateways. Thus, a value of 7% / 93% means that 7% of all data packets have been routed via GW1 and 93% have been routed via GW2.

It can be observed that with the enabled Load Switching extension more data packets have travelled via GW2 compared to the results when the Load Switching extension is not enabled. Not all data packets travel through GW2 with the Load Switching extension enabled since the background traffic at GW1 is switched on after the test file download has started. Of course, with disabled Load Switching extension



(a) w/o extension



(b) w/ ls extension

Figure 7.15: Simulation results of the Load Switching extension (symmetric setup)

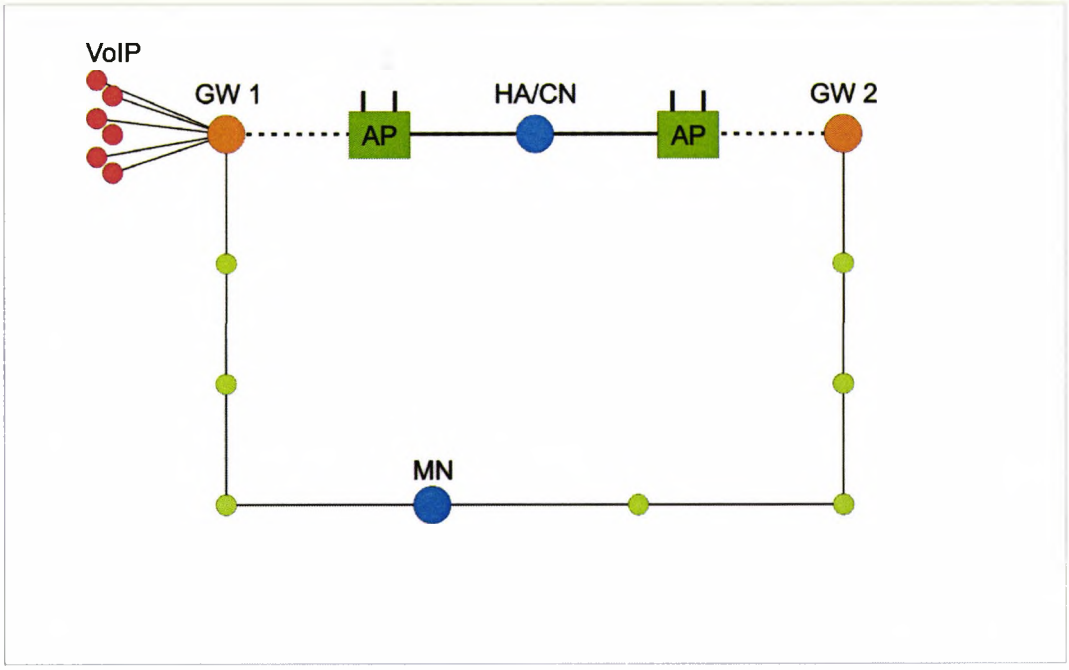


Figure 7.16: Simulation setup for the Load Switching extension (unsymmetric setup)

the results for all investigated algorithms are almost 50%, i.e. that both GWs are equally used by the MN.

As a first conclusion, the Load Switching feature works in a determined scenario with both gateways in equal hop distance. The Load Switching extension is now investigated in an unsymmetric scenario setup.

7.6.2 Unsymmetric Setup

In opposition to the symmetric setup the mobile node now is located at a position where GW1 is one hop closer to the MN compared to GW2. In Figure 7.16 the route from the MN to GW1 consists of four hops while the route from the MN to GW2 is five hops long. Thus, the MN will connect to GW1 as long as the VoIP traffic load in GW1 is not too high. The switching point of the MN is set to 320 $\frac{\text{kbit}}{\text{s}}$, i.e. two simultaneous VoIP connections to GW1. If the MN has switched to GW2 the download of the test file is not affected by the VoIP traffic to GW1. The switching point is adjusted to not reduce the throughput by switching to GW2 too early.

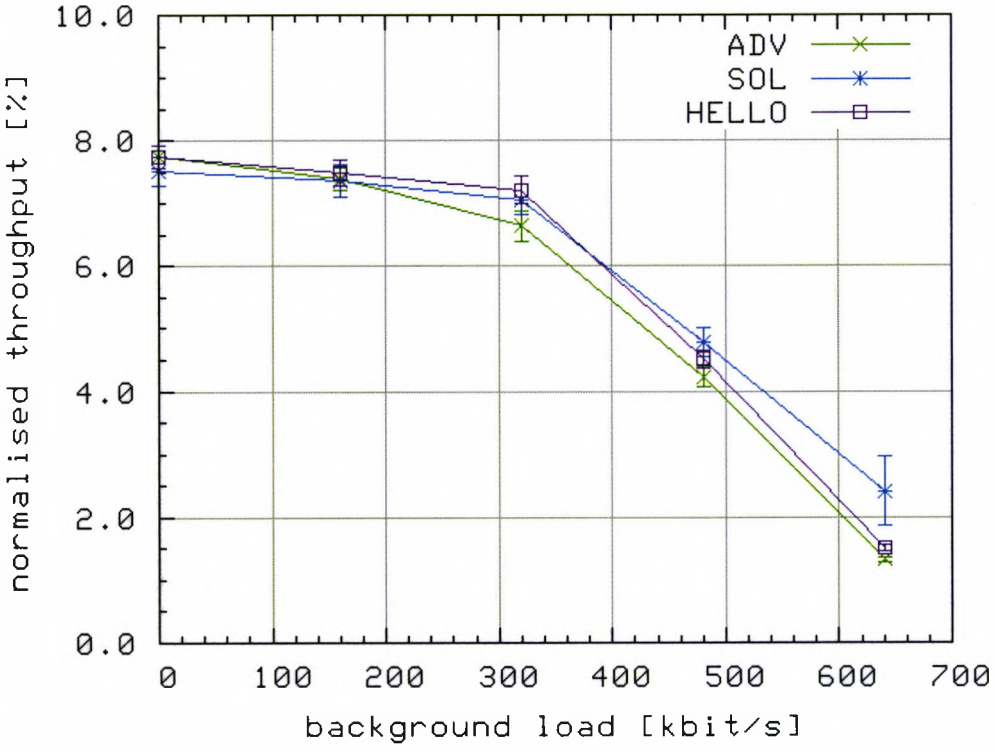
The simulation results of the unsymmetric setup are depicted in Figures 7.17, 7.18,

algorithm	w/o Load Switching	w/ Load Switching
ADV	52% / 48% ± 0.4	10% / 90% ± 1
SOL	80% / 20% ± 11	90% / 10% ± 7
HELLO	100% / 0% ± 0	10% / 90% ± 1

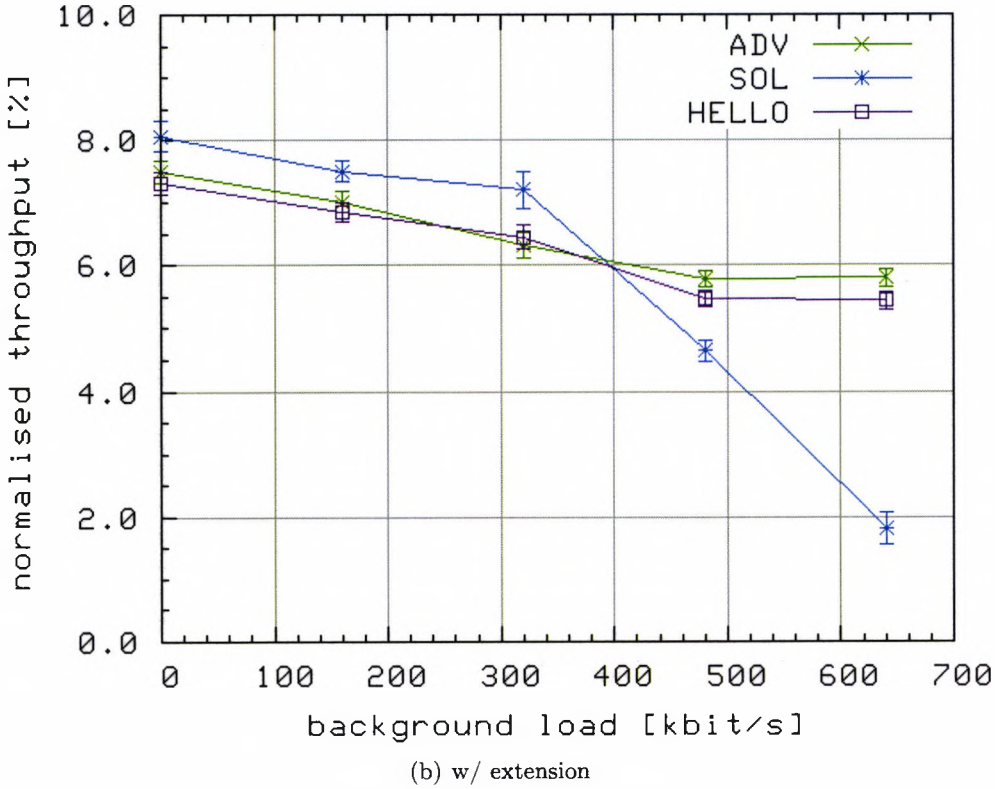
Table 7.8: Traffic distribution in unsymmetric setup at 640 $\frac{\text{kbit}}{\text{s}}$ background traffic

and 7.19. It can be observed that the throughput is about 7.8 for all three investigated gateway discovery algorithms when no background traffic is charged to GW1 and with the disabled Load Switching extension. Compared to Figure 7.13(a) (throughput in symmetric setup) the throughput is increased due to the shorter route from the MN to GW1. The relationship between a route's length and the bandwidth of a multihop connection is discussed in Appendix A. Furthermore, with increasing background traffic charged to GW1 the three investigated gateway discovery algorithm suffer likewise whereas the solicitation based algorithm shows slightly better results. This is explained with the loss of connectivity between the MN and GW1 for all algorithms and the reactive re-discovery of GW2. Both proactive algorithms will always re-connect to GW1 since the route to GW1 is shorter compared to the route to GW2. Using the solicitation based algorithm with increased background traffic GW1 may not be re-discovered successfully in all simulation runs due to the traffic in GW1 and thus the MN will connect to GW2 which is not charged with traffic. This applies before the background traffic reaches the switching point. With enabled Load Switching extension the solicitation based algorithm shows comparable results as with disabled Load Switching extension. In fact, the solicitation based algorithm does not switch clearly to the alternative GW2. This can be observed looking at Table 7.8. Like above the values given in Table 7.8 represent the amount of data packets in percent that have been routed via GW1 or via GW2.

Using the both proactive algorithms the MN switches successfully to GW2 with increasing traffic rate charged to GW1. Before the MN decides for GW2 the throughput decreases slightly from 7.8% to 7.5% and then increases from 1.3% to 5.8% for a background traffic rate of 640 $\frac{\text{kbit}}{\text{s}}$ after switching to GW2 and remains at that level whereas the solicitation based algorithm scales bad with the increasing traffic in GW1 and drops to almost 2% with and without enabled Load Switching extension.

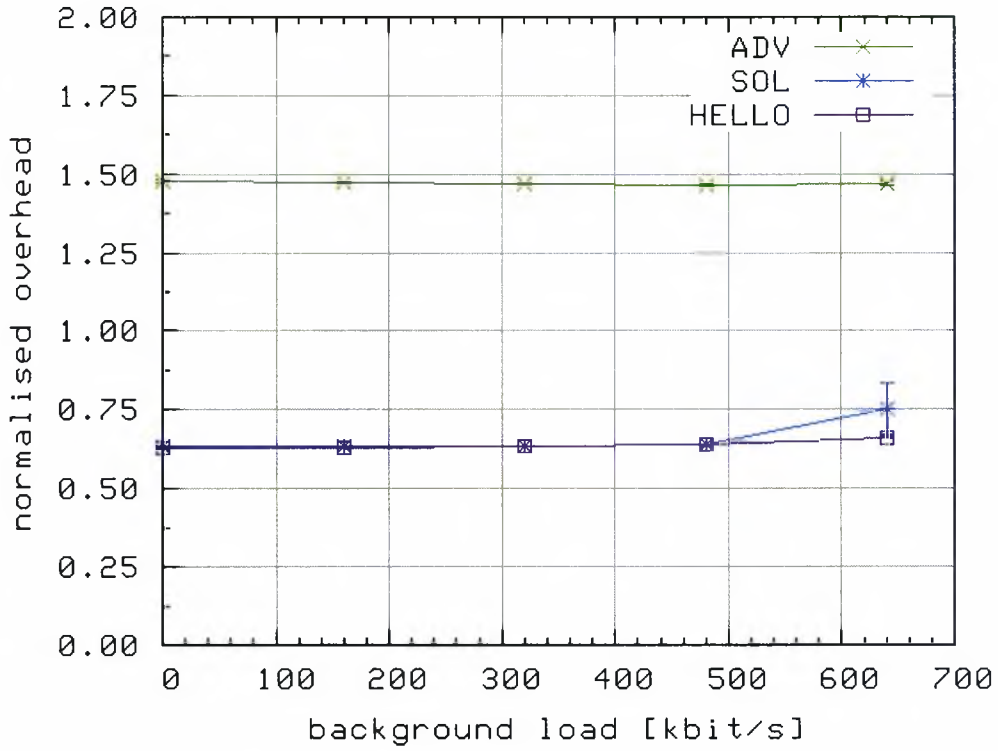


(a) w/o extension

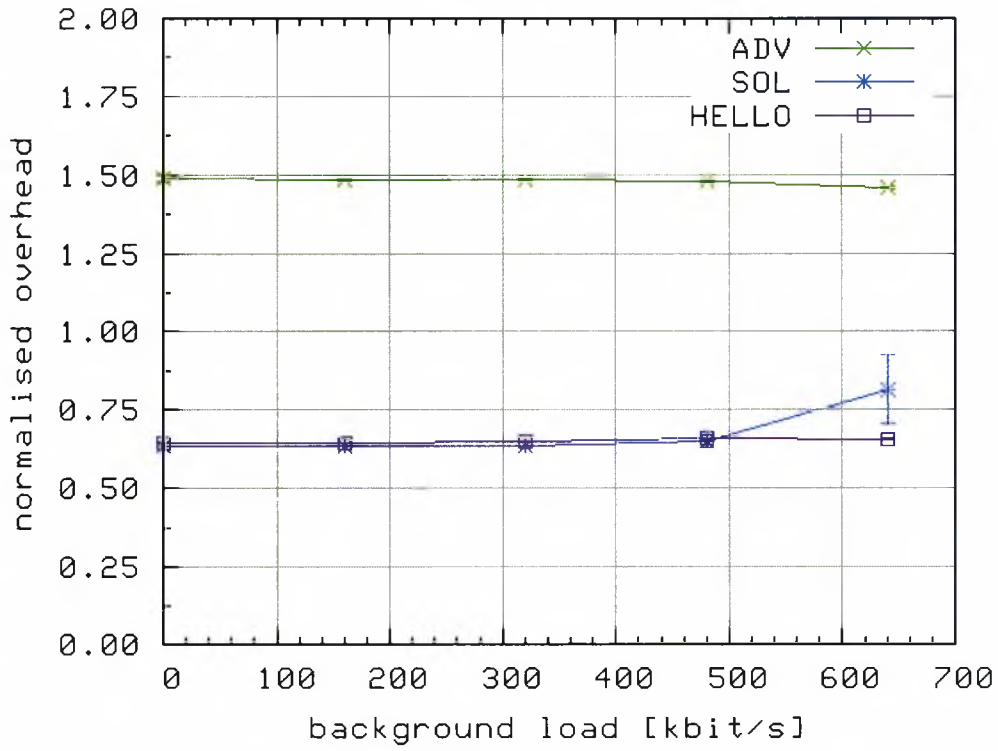


(b) w/ extension

Figure 7.17: Simulation results of the Load Switching extension (unsymmetric setup)

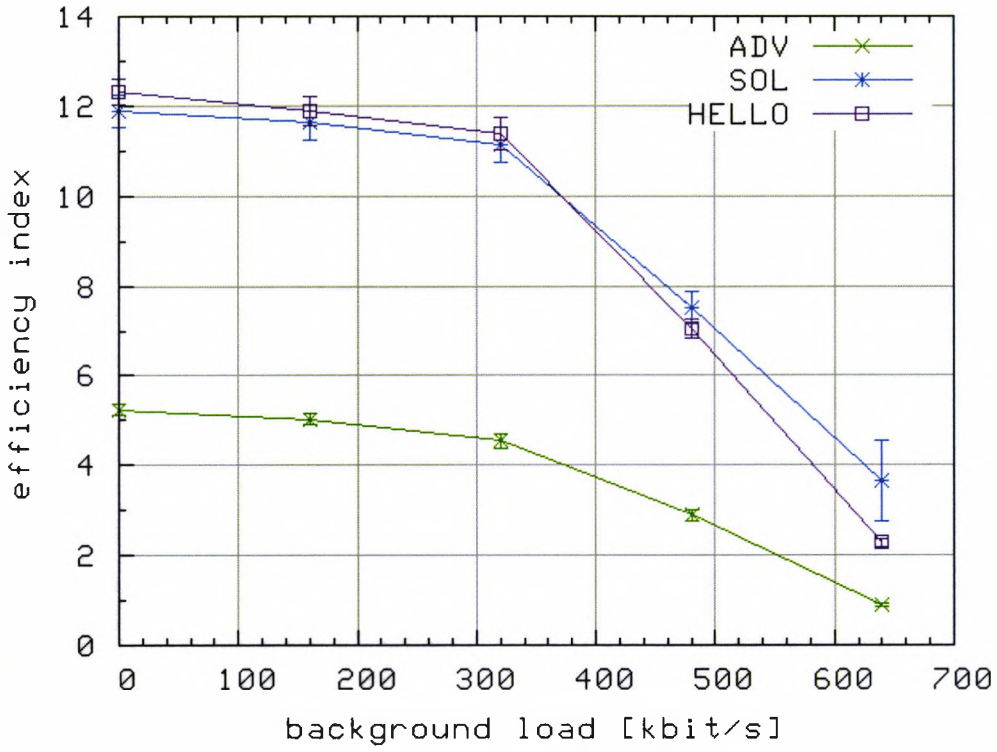


(a) w/o extension

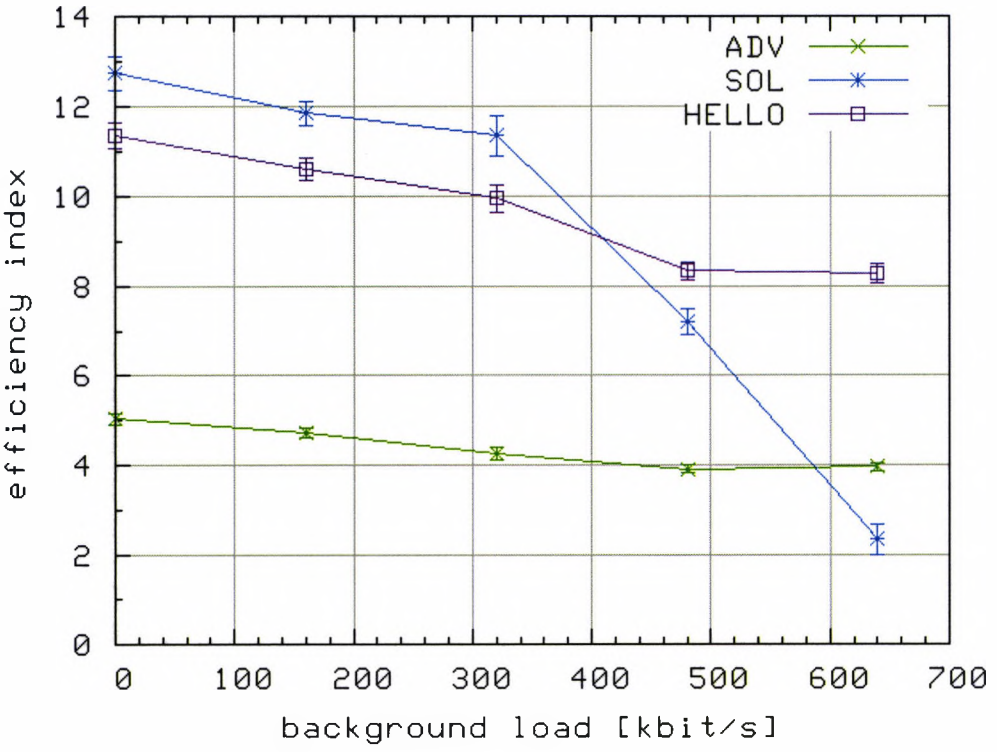


(b) w/ extension

Figure 7.18: Simulation results of the Load Switching extension (unsymmetric setup)



(a) w/o extension



(b) w/ extension

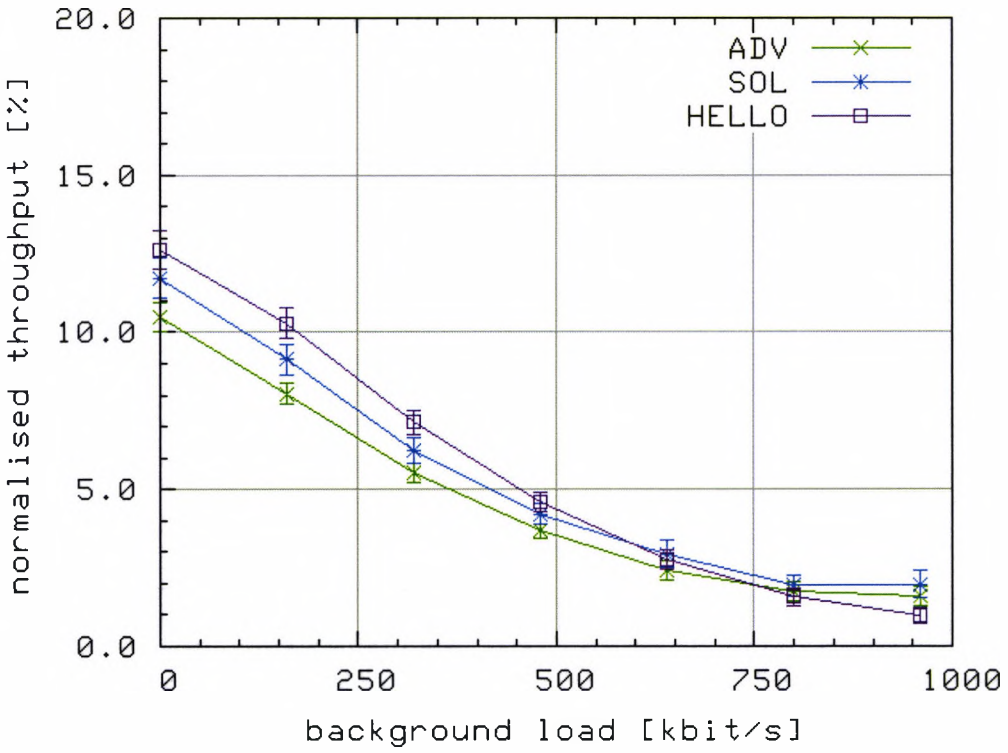
Figure 7.19: Simulation results of the Load Switching extension (unsymmetric setup)

The protocol overhead for all three investigated algorithms is almost constant with and without the enabled Load Switching extension and increasing background traffic, since the Load Switching extension does not send additional control messages. Only the solicitation based gateway discovery algorithm's protocol overhead increases with high traffic load of 640 $\frac{\text{kbit}}{\text{s}}$. This is because the MN losses connectivity to the selected Internet gateway and broadcasts solicitations into the ad-hoc network for re-gaining connectivity. This broadcasting of solicitations applies for enabled and disabled Load Switching extension.

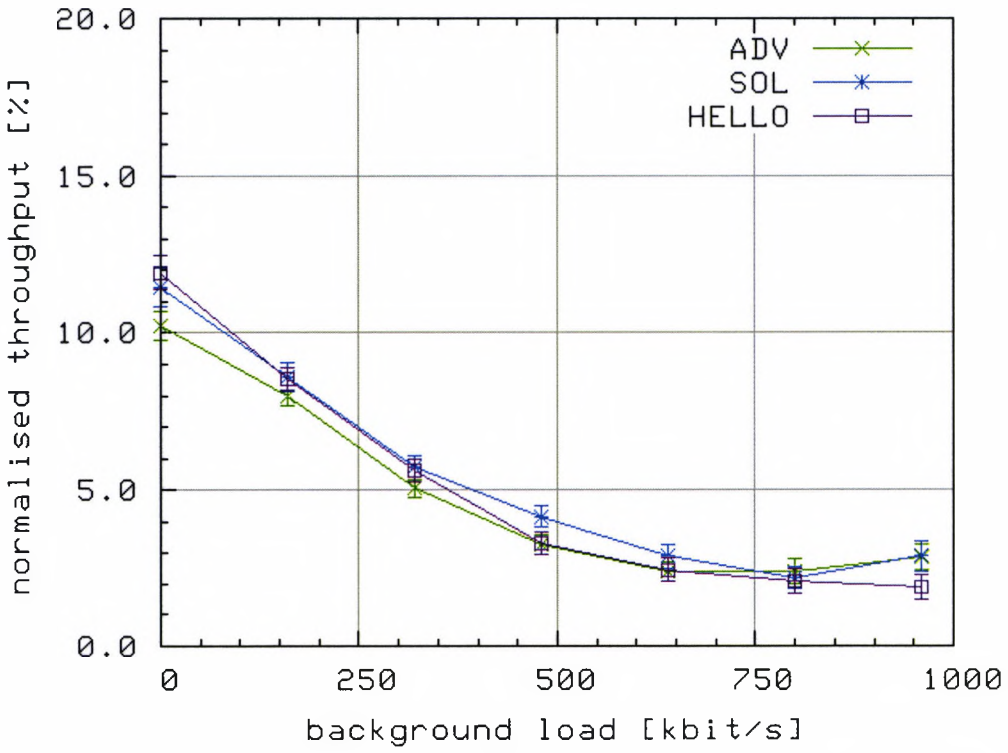
Looking at the protocol efficiency index it can be observed that the proactive algorithms lead to higher efficiency index since the MN is able to select an alternative Internet gateway and therefore the provided bandwidth to the MN is not as dramatically decreased as compared to the reactive (solicitation) based algorithm. The HELLO message based Internet gateway discovery algorithm shows the best efficiency index at high background traffic rates if the Load Switching extension is enabled.

7.6.3 Random Setup

In the next scenario setup, the mobile nodes the ad-hoc network consists of are moving around randomly in accordance with the random waypoint model as described in section 7.2.1. The number of ad-hoc mobile nodes is set to 30 nodes that equals the same node density as in section 7.4.1. One test mobile node MN downloads a test file of 1 MB in size from the corresponding node CN. The MN is located statically in the middle of the simulation plane at MN(400|100) and the background traffic at GW1 is switched on after the test file download has been initiated at $t_{\text{SIM}} = 105$ seconds. Figures 7.20, 7.21, and 7.22 depict the simulation results. The following algorithm characteristics have been investigated. These are the provided throughput to the MN, the control message overhead, and the efficiency index. The results are depicted on the top of each other for better comparability whereas the upper figure gives the simulation results with disabled Load Switching extension and the lower figure gives the simulation results with enabled Load Switching extension. Totally, 200 simulation runs were performed. Like above, this number of simulation runs is necessary because of the randomised topology and as a basis for the statistical evaluation.

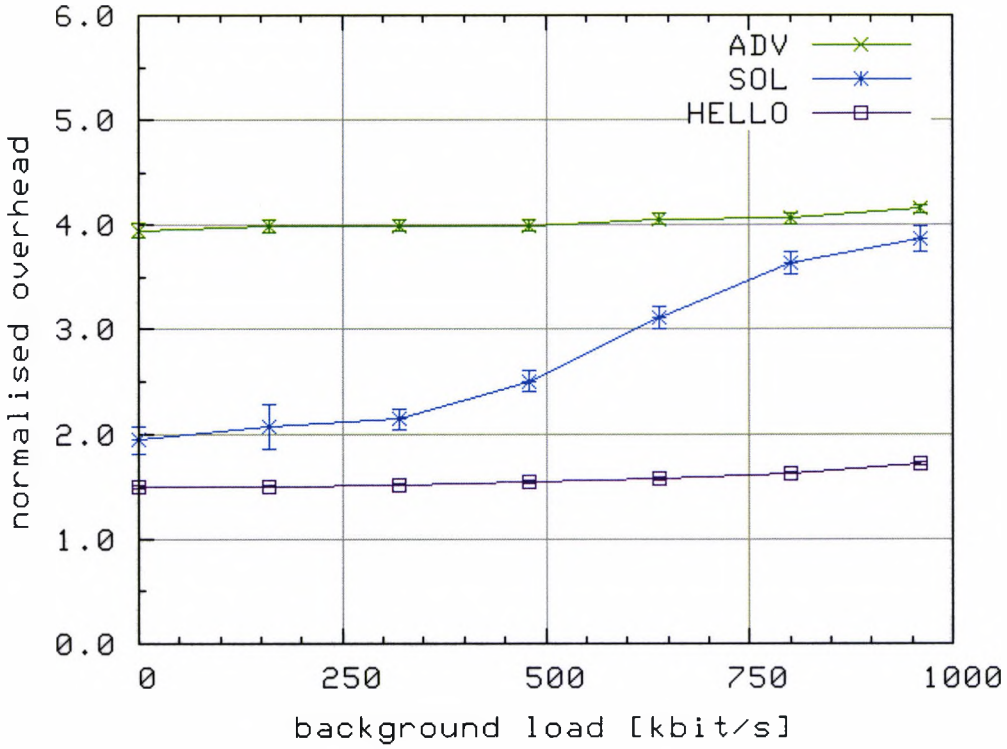


(a) w/o extension

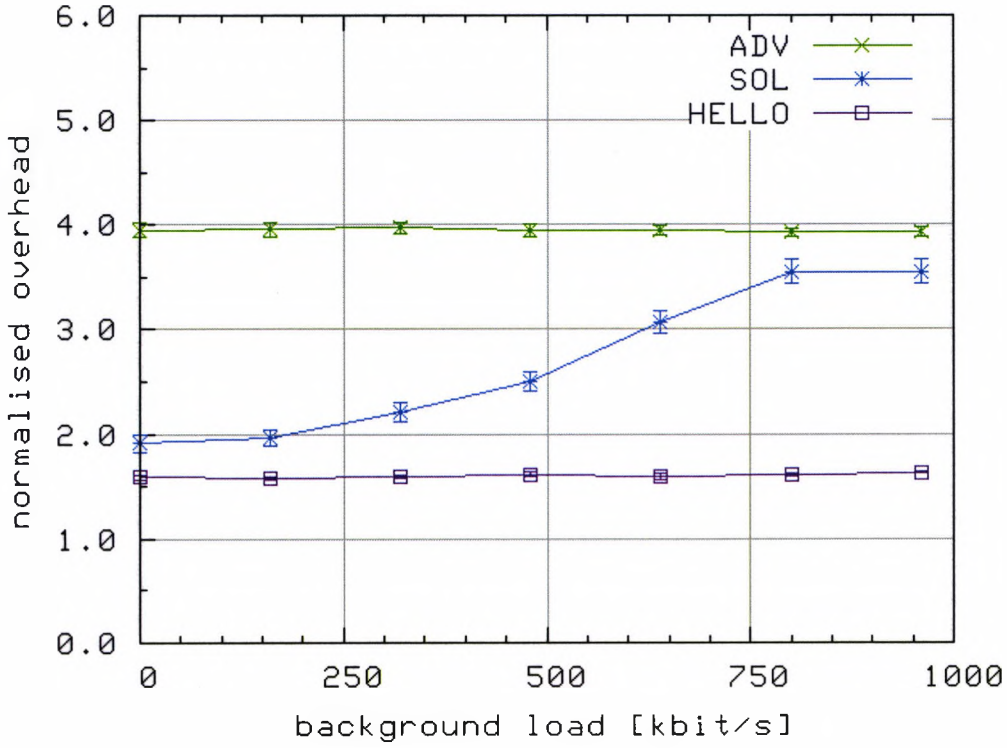


(b) w/ ls extension

Figure 7.20: Simulation results of the Load Switching extension (random setup)

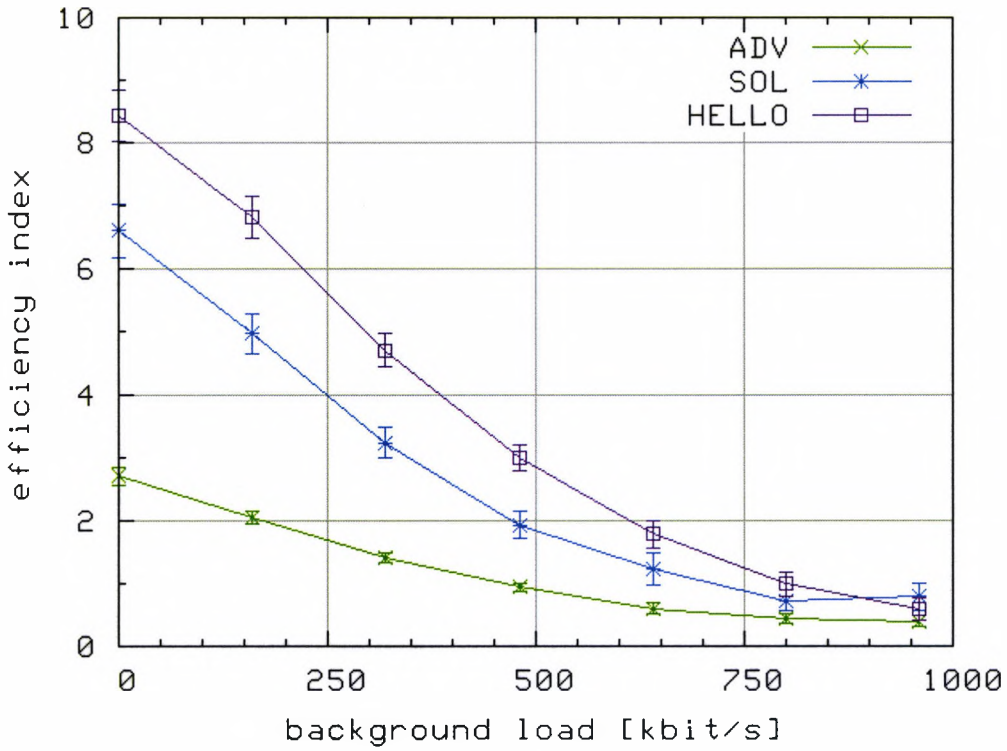


(a) w/o extension

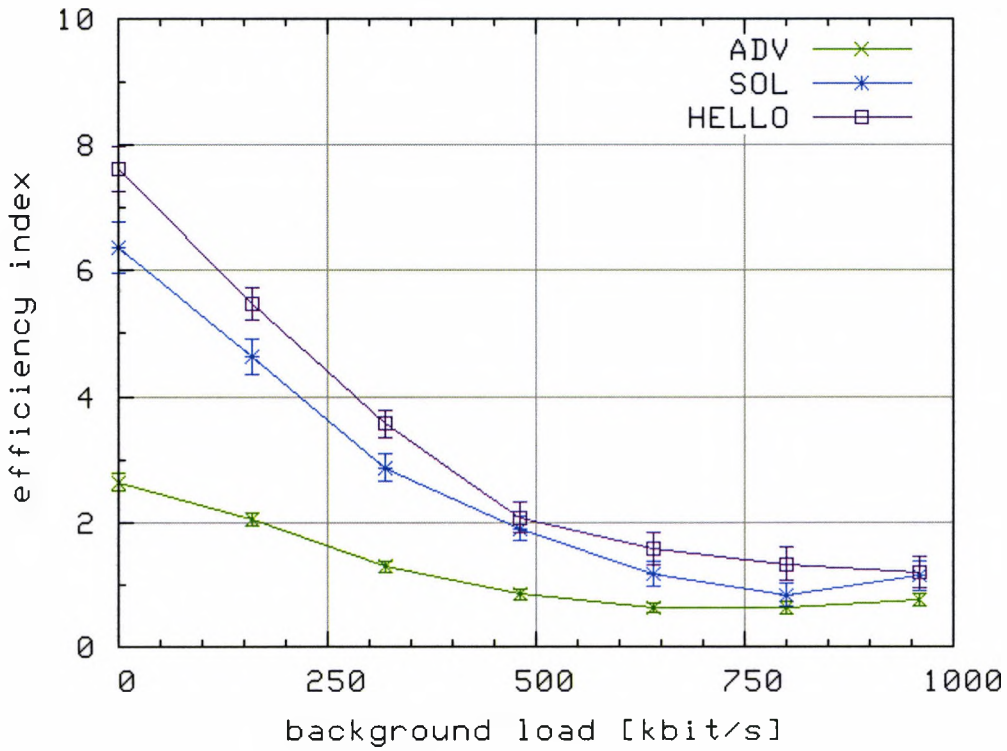


(b) w/ ls extension

Figure 7.21: Simulation results of the Load Switching extension (random setup)



(a) w/o extension



(b) w/ ls extension

Figure 7.22: Simulation results of the Load Switching extension (random setup)

In general, with increasing background traffic charged to GW1 every investigated algorithm suffers in terms of the throughput. Both proactive gateway discovery algorithms benefit from the extension. The provided bandwidth to the MN increases from 1.6% to 2.9% for the advertisement based algorithm and from 1.0% to 1.9% for the HELLO message based algorithm when the background traffic load in GW1 is set to 960 $\frac{\text{kbit}}{\text{s}}$ which equals 6 simultaneous parallel full-duplex VoIP transmissions.

The protocol overhead is comparable for all investigated algorithms with and without enabled Load Switching extension. This is since the Load Switching extension does not send any additional messages into the ad-hoc cluster like, e.g. the GRREP extension does (see section 7.5). The protocol message overhead is constantly high at 4 times the standardised protocol overhead for the advertisement based algorithm and 1.6 for the HELLO message based algorithm. With increasing background traffic the overhead of the solicitation based algorithm increases from 1.9 to 3.7. This dramatic increase is explained with the loss of connectivity of the mobile nodes to the Internet gateway and the re-discovery of Internet gateways by network wide flooding. The overhead generated by the solicitation based algorithm even approaches the overhead of the advertisement based algorithm when the background traffic rate is set to 960 $\frac{\text{kbit}}{\text{s}}$. This is explained with the network wide flooding of multiple ad-hoc mobile nodes. Thus, the solicitation based gateway discovery algorithm scales bad with high traffic load charged to a specific Internet gateway.

Looking at the protocol efficiency index the HELLO message based gateway discovery algorithm shows the best results compared to the other gateway discovery algorithms. This was also found when the HELLO algorithm is initially investigated in section 7.4 and is confirmed here.

Now, the chapter ends with a conclusion of the simulation results.

7.7 Conclusion

This chapter investigates Internet gateway discovery algorithms and extensions made to the advertisement, the solicitation, and the HELLO message based gateway discovery algorithms. The chapter firstly investigates the HELLO message based algorithm with the density of network nodes in a mobile multihop ad-hoc network. Further the

algorithms' behaviour on background traffic load is investigated as well as the chapter discusses the impact of node mobility. This is to give a statement of the contribution of this thesis to the science. The three investigated algorithms have been compared in terms of the time a testing mobile node needs to discover an Internet gateway and to register with its home agent in the Internet. Secondly the provided bandwidth to a testing mobile node is investigated by downloading a test file of a specific size (1 MB). The test file transfer time is then transformed into a bandwidth statement the system provides to the testing mobile node. The third algorithm characteristic that is evaluated is the protocol overhead. The protocol overhead is the sum of all control messages that are sent by the AODV and MobileIP protocol. To give a statement about an algorithm's performance an efficiency index is calculated from the simulation results. The efficiency index is based upon the provided bandwidth and the routing protocol overhead generated. The formula for the protocol efficiency index is given in equation 7.2.

Next the simulation results of the HELLO algorithm and the two extension made to the advertisement, the solicitation, and the HELLO message based Internet gateway discovery algorithms are discussed in detail.

7.7.1 Benefit of the HELLO Algorithm

It can be concluded that the goal of designing a new Internet gateway discovery algorithm has been achieved successfully. The task was to design the new algorithm in order to combine the benefits of proactive Internet gateway discovery algorithms (fast response due to pre created route table entries) with the benefits of reactive Internet gateway discovery algorithms (reduced control overhead) whereas the new algorithm even outperforms the classical reactive solicitation based gateway discovery algorithm in terms of control overhead. The achievement of the goals is proven by simulations.

It is obviously that the HELLO message based gateway discovery algorithm shows very good results in terms of protocol overhead. This is since the HELLO algorithm uses messages that are sent by the underlying ad-hoc routing protocol anyway. Due to this, the HELLO algorithm shows the best protocol efficiency index too, since the protocol efficiency index is directly taken from the protocol overhead and the bandwidth

the algorithm provides to mobile nodes in an ad-hoc network. The periodic flooding with advertisements of the advertisement based discovery algorithm consumes much bandwidth. Additionally, ad-hoc nodes using the solicitation based algorithm flood the ad-hoc network when discovering routes and when multiple nodes want to connect to Internet gateways the number of network wide floodings increases. It can be concluded that flooding generally decreases the amount of available bandwidth to nodes dramatically. This is evaluated by simulations. Since the flooding of ad-hoc networks is a forwarding of control messages by all network nodes every algorithm that is based on ad-hoc network flooding scales bad with an increasing number of ad-hoc nodes or Internet gateways, respectively. The HELLO message based algorithm does not have this disadvantage since it abjures network floodings.

Section 7.4.2 investigates the influence of the mobility of mobile ad-hoc network nodes to the HELLO message based algorithm and the classical advertisement and solicitation based algorithms. It is observed that the mobility of network nodes' influence is almost negligible with the exception of the solicitation based algorithm. The solicitation based algorithm's provided bandwidth is mostly effected by high node mobility. The higher the nodes' mobility (less pause time) the more often routes break and the solicitation based algorithm needs to re-discover destination nodes or the Internet gateway. Indeed, this applies for all algorithms but the solicitation based algorithm firstly needs time to detect the loss of connectivity and then has to re-discover the destination node and secondly the both proactive algorithms (advertisement and HELLO message based) provide ad-hoc network nodes with Internet gateway information without demand and thus, nodes using a proactive algorithm can update their gateway and default routes frequently.

The number of attending ad-hoc network nodes plays an important role. The increased number of nodes dramatically increases the amount of control overhead for the advertisement based algorithm. Thus, the advertisement based algorithm scales bad with an increasing number of network nodes and Internet gateways due to the plain forwarding of advertisement messages (flooding). In [26] the authors suggest to limit the area advertisements are forwarded within but this approach will prevent the discovery of multiple Internet gateways.

The bad scalability with increasing node density applies also for the solicitation based algorithm but not in such a dramatic way. Indeed, all ad-hoc network nodes flood the ad-hoc network with solicitation requests but this is not as often as multiple Internet gateways do in the advertisement based algorithm. Using the HELLO message based algorithm only the number of sent HELLO messages increases linearly with the increased number of network nodes and therefore scales best with the increasing number of ad-hoc network nodes compared to the two other algorithms.

Looking at the time a mobile node needs to discover an Internet gateway and to register with its home agent in the Internet using the MobileIP protocol the number of surrounding neighbours plays an important role whereas the number of neighbour nodes at a specific time is random. With an increased number of neighbour nodes the delay for a specific mobile node to find a route to the Internet gateway decreases. This is because of the unsynchronised sending of HELLO messages. The unsynchronised sending of HELLO messages results in shorter register delays when the number of mobile ad-hoc nodes increases.

The influence of background traffic on the HELLO message based discovery algorithm in terms of the provided bandwidth is dramatic but comparable to the classic discovery algorithms and thus all investigated algorithms suffer similar from background traffic. The background traffic load is configured to CBR/UDP (VoIP) to adjust the load precisely. Additionally, a FTP/TCP background traffic load was evaluated. One could think that the bandwidth of the FTP/TCP background traffic load should be equal to the provided bandwidth to the testing MN but the mean route length of the background FTP/TCP traffic and the test file download via the Internet gateways to the MN are different. This is discussed in section 7.4.3.

It can be concluded that the HELLO message based algorithm is very effective when the underlying layer-2 protocol does not provide much bandwidth for spreading Internet gateway routing information with a flooding strategy and when available bandwidth plays an important role for an ad-hoc network provider. Thus the new developed and investigated HELLO message based Internet gateway discovery algorithm is suitable for systems that do not provide much layer 2 bandwidth.

7.7.2 Benefit of GRREP Extension

Secondly, this chapter evaluates the GRREP extension made to the proactive Internet gateway discovery algorithms. The extension allows frequent route updates for mobile nodes if the discovered route to the Internet gateway changes in terms of the route's length. Thus, this extension provides shorter routes for mobile nodes in the ad-hoc network to Internet gateways and therefore increases the available bandwidth for ad-hoc nodes. Appendix A discusses the relationship between a route's length and the bandwidth the route provides.

The ad-hoc mobile nodes send gratuitous route reply (GRREP) messages when being provided with more actual routes to their selected Internet gateway by proactive (advertisement based or HELLO message based) gateway discovery algorithms. GRREP messages are then being acknowledged by the Internet gateway by sending a GRREP-ACK message back to the originator of the GRREP message.

The GRREP Internet gateway discovery protocol extension does only apply for the both proactive algorithms i.e. the advertisement and the HELLO message based. Therefore, the results for the solicitation based algorithm are only given for comparison reasons. Using a proactive algorithm a mobile node is able to find shorter, i.e. newer, routes to an Internet gateway without demand and therefore can always shorten the discovered gateway and default routes. The GRREP extension allows mobile nodes to update the routing table entry in the Internet gateways to shorten the routes there, too. Since shorter routes provide more bandwidth the GRREP extension improves the performance of the proactive algorithms in terms of the provided bandwidth. Additionally, proactive gateway discovery algorithms lengthen gateway and default routes if moving nodes discover newer (higher sequence number) routes to Internet gateways. Therefore, proactive discovery algorithms always provide the newest information about Internet connectivity within a mobile ad-hoc network and do not need to wait for time outs of a link failure to a neighbour node in the route to the selected Internet gateway.

In Table 7.6 the improvement to the proactive algorithms in a determinated scenario setup is given as percentages. Additionally, for comparison the solicitation based algorithm performs bad compared to the proactive algorithms. This is since the solicitation based algorithm is not able to shorten routes if the ad-hoc network could

	benefit [%]
ADV	9.3
HELLO	11.4

Table 7.9: Benefit of GRREP extension in random scenario

provide shorter routes and needs time according to equation 4.2 for detecting the loss of connectivity to neighbour nodes and gateway nodes, respectively.

It can be concluded that the GRREP extension works and proactive Internet gateway discovery algorithms benefit from the extension. Table 7.9 compiles the benefit of the GRREP extension for the random scenario setup, whereas the results with high background traffic are not given since the simulation results are too imprecisely, i.e. the confidence intervals are too wide so that no statistical statement is possible. Next, the second extension to the advertisement, the solicitation, and the HELLO message based Internet gateway discovery algorithms is examined.

7.7.3 Benefit of Load Switching Extension

Thirdly, the chapter examines the Load Switching extension for all three investigated algorithms. The Load Switching extension allows mobile ad-hoc nodes to select an alternative Internet gateway if the initial selected Internet gateway is charged with network traffic by other mobile ad-hoc nodes. To achieve this switching the Internet gateway uses a traffic counter and provides the amount of forwarded network traffic (measured in $\frac{\text{Byte}}{\text{s}}$) to the ad-hoc mobile nodes. The ad-hoc mobile nodes then decide after a certain metric which Internet gateway is the "best" gateway. This metric includes the amount of traffic forwarded by an discovered Internet gateway and the distance in hops to that Internet gateway. To provide the mobile ad-hoc network nodes with that information the gateway discovery algorithms have been enhanced for selecting the "best" Internet gateway and the protocol messages have been extended by a `Gateway Usage` field that contains the amount of traffic within a specific Internet gateway.

The thesis proves the functionality of the Load Switching extension using two determined scenarios. In the first determined scenario two Internet gateways are located at an equal distance to a testing mobile node MN. It can be observed that the Load

	benefit [%]					
scenario	symmetric		unsymmetric		random	
background traffic	no	high	no	high	no	high
ADV	-2.3	54.0	-3.3	438.7	-2.6	80.8
SOL	1.8	-9.6	7.3	75.2	-2.4	48.1
HELLO	0.7	53.8	-5.8	362.4	-5.5	95.3

Table 7.10: Benefit of the Load Switching extension

Switching extension allows the MN to select an alternative Internet gateway if the already selected Internet gateway is charged with traffic. Ad-hoc nodes utilising the solicitation based discovery algorithm do not recognise a change in the traffic load of a specific Internet gateway since the solicitation based algorithm works reactively. The Load Switching extension provides the most benefit to the both proactive gateway discovery algorithms when the traffic load in the first gateway is set to a high rate (640 $\frac{\text{kbit}}{\text{s}}$). Nevertheless, the benefit is 0% when neither gateway is loaded with traffic. Table 7.10 compiles the simulation results of the Load Switching extension.

In the unsymmetric determined scenario setup for the Load Switching extension the route to one gateway is one hop shorter than a route to another gateway. It can be observed that the MN firstly selects the closest Internet gateway and when the firstly selected Internet gateway is charged with traffic the MN switches to the alternative gateway if the traffic load in the firstly selected gateway exceeds a certain threshold value. This threshold is set to equal a background traffic rate of 320 $\frac{\text{kbit}}{\text{s}}$ (two simultaneous full-duplex VoIP connections). Again, nodes using the solicitation based algorithm do not recognise a traffic load change in the selected Internet gateway and thus nodes using the solicitation based algorithm do not change to an alternative Internet gateway.

If the nodes of an ad-hoc network are mobile and perform the Load Switching extension the results are very different compared to the determined scenario setups. It is found that the solicitation based Internet gateway discovery algorithm is resistant to the Load Switching extension in the simulated scenarios with random moving mobile nodes. This is due the non-recognising of a load change in the firstly selected Internet gateway caused by the reactive discovery approach of the solicitation based algorithm.

There is no benefit found to the provided bandwidth when the background traffic

load in the firstly selected Internet gateway is set to 0 $\frac{\text{kbit}}{\text{s}}$. Thus the Load Switching extension generates a benefit when the Internet gateways are unbalancedly loaded only. This is since in a totally symmetric setup with both Internet gateways equally loaded with background traffic the mobile nodes' routing decision for Internet connectivity is only based upon the hop count to the Internet gateway.

The investigations of the HELLO message based algorithm and the algorithms' extensions clarify the complexity and challenges when using TCP in mobile ad-hoc networks. Since TCP was not originally designed for wireless networks it is very sensitive for data packet and control message losses. This fact and the attempt of TCP to maximise the throughput have a bad impact on wireless mobile multihop ad-hoc networks. The unsteady throughput of a TCP connection (window size, congestion actions, etc., section 2.2.3) lead to frequent losses of connectivity for mobile ad-hoc nodes due to congestion. When network congestion happens routes must be re-discovered which causes protocol overhead. Additionally, due to route changes and disruptions the TCP flow control mechanism reacts with congestion actions of the TCP protocol. Congestion actions decrease the bandwidth a mobile node in an ad-hoc mobile network is provided with.

Next, the thesis ends with a conclusion.

Chapter 8

Conclusion

The first goal of this thesis is to develop and to evaluate algorithms and protocols to achieve Internet connectivity for mobile ad-hoc networks (MANET). The second goal is to extend existing and a new algorithm for Internet connectivity for mobile ad-hoc nodes to find better routes from an Internet gateway to a specific mobile ad-hoc node and to allow to select between multiple Internet gateways if more than one has been detected by mobile ad-hoc nodes using the investigated algorithms.

In the literature two main approaches for the discovery of Internet gateways are discussed. The first discussed main approach is to flood the ad-hoc network periodically with advertisements. The other discussed main approach is that ad-hoc mobile nodes solicit for Internet gateways when a connection to the Internet is required. The advertisement based approach is providing mobile ad-hoc nodes with information about the presence of an Internet gateway without demand and is therefore a proactive approach while the solicitation based approach is a reactive, i.e. on-demand approach.

Information about an Internet gateway includes the Internet gateway's address, the hop count to the Internet gateway, the sequence number of the Internet gateway as well as the neighbour node used as a next hop toward the Internet gateway as well as an information about the amount of traffic an Internet gateway is already forwarding to other mobile ad-hoc nodes. This last information applies for the enhanced Internet gateway discovery algorithms that are presented in this thesis.

Both, the proactive (advertisement based) and the reactive (solicitation based) approaches have pros and cons. The pro of the advertisement based approach is that

ad-hoc mobile nodes are always provided with up to date information about Internet connectivity. This enables mobile ad-hoc nodes to use an Internet gateway immediately when it is needed. Additionally, due to the sequence number ad-hoc mobile nodes are able to select a newer route to the Internet gateway when the newer route is shorter or even longer. The drawback of this advertisement based approach is that the periodic flooding of the ad-hoc network causes protocol overhead. This protocol overhead increases if the number of ad-hoc mobile nodes increases since when flooding the ad-hoc network with advertisements all ad-hoc mobile nodes are forwarding the advertisement once. Additionally if an ad-hoc network is attached to more than one Internet gateway all Internet gateways will flood the ad-hoc network periodically. Thus, the advertisement based approach scales badly with increasing number of network nodes and Internet gateways.

The solicitation based approach is reducing network wide flooding since ad-hoc nodes only solicit for an Internet gateway if one is needed. The drawback of the solicitation based approach is that ad-hoc multihop routes to an Internet gateway are not updated periodically. Depending on the actual ad-hoc network topology this not updating can cause nodes to use routes to an Internet gateway even if shorter routes were possible. A shorter route in general is known to provide more bandwidth and less packet delays.

The thesis introduces a new approach for discovering Internet gateways within mobile ad-hoc networks. Based on the Ad-hoc On-demand Distance Vector (AODV) routing protocol for mobile ad-hoc networks it is able to distribute Information about present Internet gateways among the mobile ad-hoc nodes. This is achieved by HELLO messages that the AODV protocol uses for neighbourhood management, i.e. to observe which nodes are within the direct vicinity of a specific node. Neighbourhood management is also performed by the Internet gateway since the Internet gateway is a node of the ad-hoc network, too. The HELLO message based algorithm works as follows. The mobile ad-hoc nodes in the vicinity of the Internet gateway are aware of the Internet gateway as they are receiving HELLO messages directly from the Internet gateway. To indicate that the HELLO messages are sent by an Internet gateway node and that the originating address can be used for Internet connectivity the Internet gateway sets a

flag, the I-flag, in the HELLO messages that therefore are called HELLO_I messages.

To distribute the information about a detected Internet gateway deeper into the mobile ad-hoc network Internet gateway aware nodes include this information about the presence of an Internet gateway into their own HELLO messages that therefore are being transformed into HELLO_I message, too.

Standard HELLO messages need to be extended to carry Internet gateway information. The extension is necessary to carry information about more than one Internet gateway whereas the HELLO messages are still used for their original purpose. To include information about more than one Internet gateway the size of HELLO message increases like the route error message (RERR) of the AODV protocol does when informing mobile ad-hoc nodes about the loss of connectivity to multiple nodes. As a result, the HELLO message based Internet gateway discovery algorithm is able to provide mobile ad-hoc nodes in an ad-hoc network with information about multiple Internet gateways simultaneously without sending additional protocol messages. The HELLO message based Internet gateway discovery algorithm is investigated and evaluated in this thesis.

The advertisement based, the solicitation based as well as the new HELLO message based Internet gateway discovery algorithms are compared in this thesis. The comparison of the algorithms is to evaluate under which conditions and circumstances which algorithm performs best in terms of the time a mobile ad-hoc network node needs to firstly discover an Internet gateway and to register with its home agent in the Internet, the throughput a mobile ad-hoc network node is provided with, and the control message overhead generated to provide this initial discovery time and throughput. In order to compare the Internet gateway discovery algorithms at a glance the thesis introduces an algorithm performance index based on the provided throughput and the control message overhead needed to provide this throughput.

The benefit of the HELLO message based approach for detecting Internet gateways within a mobile ad-hoc network avoids network flooding and since the advertisement based and the solicitation based approaches are also using HELLO messages for neighbourhood management no additional protocol overhead is generated. This is the main advantage of the new HELLO message based approach. Another advantage is that ad-

hoc mobile nodes are provided with information about Internet connectivity without demand and therefore, like the advertisement based algorithm, the HELLO message based algorithm is a proactive one and can update and shorten the multihop route to a discovered Internet gateway periodically.

In a second step, the thesis enhances the advertisement, the solicitation, and the HELLO message based algorithms for detecting Internet gateways. The first enhancement provides the Internet gateway node with up to date information about the ad-hoc multihop route to a specific ad-hoc node when using a proactive Internet gateway discovery algorithm. Since in proactive Internet gateway discovery algorithms the Internet gateway is proactively announcing information about available Internet connectivity mobile nodes are aware of the newest route to an Internet gateway but the Internet gateway itself has no up-to-date reverse route entries to nodes that are connected to the Internet using this Internet gateway. The extension allows a route update in the Internet gateway by sending unrequested route reply messages (GRREP) from mobile nodes to the Internet gateway if the mobile nodes detect a newer route, even if it is a longer or a shorter route. These unrequested route reply messages are then being acknowledged by the Internet gateway (GRREP-ACK). Using this reply acknowledge procedure all ad-hoc nodes along a route from a specific mobile node to an Internet gateway and the Internet gateway as well are informed about route changes. The GRREP extension only applies for proactive Internet gateway discovery approaches since nodes using a reactive approach will always stay connected to an already discovered Internet gateway until the multihop route to the Internet gateway breaks.

The second enhancement to Internet gateway discovery algorithms allows the selection between multiple detected Internet gateways not only by the hop count, like in standard implementations but, additionally by the amount of network traffic the detected Internet gateways are already forwarding. This is achieved by including information about forwarded traffic into the appropriate Internet gateway discovery messages, i.e. into advertisement, solicitations, and HELLO_I messages. Here, nodes discovering multiple Internet gateways have to decide after a certain metric if a gateway that is one hop closer but charged with more traffic compared to an alternative Internet gateway can be assumed as better in terms of the provided throughput to the mobile nodes.

The above mentioned algorithms and protocols were implemented into the Network Simulator NS-2 in order to evaluate and compare the algorithms by simulations.

When working with multihop mobile ad-hoc networks the performance of a mobile ad-hoc network is very difficult to predict. Due to the random movement of nodes routes may be lengthened and shortened by the ad-hoc routing protocol and bottleneck-nodes may appear and disappear any time (refer to Appendix A about the relationship between a route's length and the bandwidth it provides). A bottleneck-node is a node that forwards traffic for multiple connections within the ad-hoc network. Therefore, if two multihop connections within the ad-hoc network are sharing a bottle-neck node the bottle-neck node will decrease the provided bandwidth to the multihop connections dramatically.

In opposition, if two parallel connections are being established within an ad-hoc network and the multihop routes of the connections avoid each others radio range and even each others interference range with different source and destination nodes the maximum transfer capacity of the ad-hoc network is doubled. Due to the random movement of ad-hoc nodes and the random selection of source and destination nodes both multihop route possibilities (routes with bottle-neck node and strictly separated routes) are impossible to predict. Additionally, in mobile ad-hoc networks the random movement of nodes may cause a network to split into two or more sets. If a node tries to get connected to an Internet gateway and the node is randomly located within such a split network it has no chance to discover the Internet gateway at all and therefore it has to wait until the random movement of nodes eventually provides possible routes to the Internet gateway in the future.

Another unpredictable factor to the performance of ad-hoc mobile networks is traffic within the ad-hoc network and to the Internet. Network traffic can cause protocol messages to get lost by collisions and congestion which leads to re-discovery actions of nodes and therefore additional control message overhead by the ad-hoc routing protocol. Thus, when comparing gateway discovery algorithms the provided throughput to a specific mobile node is decreased with background traffic.

It is obvious that the performance of a mobile ad-hoc network consisting of random moving nodes and its algorithms and protocols, incl. gateway discovery algorithms, is

challenging to investigate and therefore a lot of simulation runs (up to 250) are needed to get reasonable results in terms of mean values [2, 3]. This thesis extents simulation results by providing 95% confidence intervals in order to enhance the expressiveness of the simulation results.

As mentioned above, firstly the thesis investigates the time a just switched-on node needs to discover an Internet gateway and to register with its home agent in the Internet (register time) by simulations. Secondly the thesis investigates the throughput a gateway discovery algorithm is providing to the mobile ad-hoc network nodes and the amount of control messages needed for that provisioning. Last, the thesis introduces a performance index to compare the gateway discovery algorithm easily. The investigated Internet gateway discovery algorithms are the advertisement based, the solicitation based, and the newly introduced HELLO message based algorithms and they are investigated in this thesis using simulations.

From simulations, it can be concluded that the protocol overhead of the solicitation based algorithm is twice the protocol overhead of the HELLO message based algorithm if the mobile nodes of an ad-hoc network move with a pause time 0 seconds (high mobility). The protocol overhead of the advertisement based algorithm is 2.6 times the protocol overhead generated by the HELLO message based algorithm.

The simulated throughput provided in networks using one of the three investigated algorithms is independent of the nodes' mobility but the density of network nodes plays an important role. So the provided throughput of the HELLO message based algorithm is almost constant with increasing node density while the advertisement based algorithm suffers from the increased node density. With an increase from 30 to 60 nodes the provided throughput of the advertisement based algorithm decreases to two-thirds of the HELLO message based algorithm.

The influence of the nodes' mobility is in general low except to the solicitation based algorithm with high node mobility. A high node mobility leads to frequent route breaks and therefore to frequent re-discovery procedures by the mobile nodes and the Internet gateways which causes protocol overhead. So the protocol overhead of the solicitation based algorithm decreases from 5.5 to 4 times of the normalised overhead with an increasing node pause time from 0 seconds to 900 seconds.

Combining the provided throughput and the generated overhead, the efficiency index shows the best results for the HELLO message based algorithm. See section 7.4 on page 131 for all results.

The second goal of this thesis is to enhance existing Internet gateway discovery algorithms (including the new HELLO message based). There are two enhancements presented and investigated in this thesis. The first enhancement is about the sending of gratuitous route reply (GRREP) messages from mobile ad-hoc nodes to Internet gateways.

Simulations firstly show that the GRREP extension works properly. In a determinated scenario setup with static nodes the enhancement to the algorithms provides about 11% to 12% more throughput to the ad-hoc nodes with enabled GRREP feature.

Secondly, using a random topology scenario the simulated throughput is about 10% higher with the GRREP extension enabled. The enabled GRREP extension causes additional protocol overhead. The protocol overhead with enabled GRREP feature is twice as much as without the GRREP extension for the advertisement and the HELLO message based algorithm. Note, the GRREP extension does not apply to the solicitation based algorithm.

In spite of the doubled protocol overhead the GRREP extension is able to provide 10% more throughput to mobile ad-hoc nodes whereas the efficiency of the extended protocol decreases.

The second extension to Internet gateway discovery algorithms is called the Load Switching extension. The Load Switching extension allows mobile nodes of an ad-hoc network to select between multiple discovered Internet gateways not only by the hop count to the discovered gateways but additionally by the amount of Internet traffic the already discovered Internet gateways are forwarding.

To allow the Load Switching to the mobile nodes an Internet gateway includes a usage information about the traffic it is forwarding into its advertisements, the solicitation replies, and the new HELLO_I messages. Using the standard Internet gateway information distribution approach of the discovery algorithms the usage information about Internet traffic within a gateway is provided to the mobile ad-hoc nodes whereas the message formats of the protocols need to be slightly modified in order to contain

the gateway usage information. The usage information is metered in the number of Bytes an Internet gateway is forwarding per second.

The HELLO messages of the HELLO message based algorithm need to be modified, too. Since mobile nodes using the HELLO algorithm may be aware of more than one Internet gateway, each node including Internet gateways, have to include the usage information of all discovered Internet gateways into their own HELLO_I messages. Therefore, with every discovered Internet gateway the size of a HELLO_I message increases. See section 4.5 for details on the HELLO_I header format.

The Load Switching extension is presented and investigated in this thesis. Firstly, simulations were carried out to prove the functionality of the extension in a symmetric scenario setup where the multihop paths of a specific mobile node to two Internet gateways are equal in terms of the hop count. The second functionality test is an unsymmetric setup where one path to an Internet gateway is one hop shorter compared to an alternative path to an alternative Internet gateway whereas the alternative Internet gateway is charged with Internet traffic. Then a specific mobile node has to decide if the closer but traffic loaded Internet gateway or the more distant but less used Internet gateway is better for Internet connectivity in terms of the provided throughput. This decision is located within the mobile nodes of an ad-hoc mobile network and the nodes calculate a metric based upon the hop distance and the usage value of each discovered Internet gateway.

Simulations with a static determined symmetric and unsymmetric scenario setup show that the Load Switching extension is working properly. Internet gateways are loaded specifically with CBR/UDP traffic with VoIP parameters. This allows a precise adjustment of the background load within the Internet gateways.

The main issue with the Load Switching extension is the trade-off before selecting an alternative but more distant Internet gateway or stay connected to a closer but traffic charged Internet gateway (switching point). The simulations show that only for high background traffic rates the switching to an alternative Internet gateway is worthwhile. The switching point is this thesis is set to two parallel full-duplex VoIP connections through the Internet gateway which equals $320\frac{\text{kbit}}{\text{s}}$. This is an empirical value found by simulations.

In order to show the lack of Internet gateway information spreading of the solicitation based algorithm the background traffic through the Internet gateway is switched on after a specific mobile node has discovered one of two Internet gateways. Using this setup it can be observed that the both investigated proactive algorithms for Internet gateway discovery (advertisement and HELLO message based) provide the information about a change in the background traffic load in the Internet gateway. The reactive solicitation based algorithm is not able to distribute the information about a change in the background traffic load since it works on a on-demand basis.

In the simulated scenario setups with the enabled Load Switching extension the throughput increases by >50% for the both proactive algorithms with a high background traffic load within the Internet gateway whereas the throughput with solicitation based algorithm still suffers. This suffering is a result from the late switching on of the background traffic within the Internet gateways. With no or less background traffic the benefit of the Load Switching extension is negligible for any of the three investigated algorithms.

With a random walk based topology of the mobile ad-hoc network the algorithms benefit from the extension too but the benefit is less compared to the determined scenarios. It is obvious that this is a direct effect from the random walk of the mobile ad-hoc nodes. With the random movement of mobile nodes the different lengths of routes from the mobile ad-hoc nodes to the Internet gateways come into play. The Load Switching extension is adjusted to a specific switching point and this adjustment prevents mobile nodes to select a less used but more distant Internet gateway if this alternative Internet gateway is two hops farer away. In such a case the mobile node will always stay connected to the initially selected Internet gateway.

The Load Switching extension is able to provide up to 430% more throughput to mobile nodes of an ad-hoc mobile network if an alternative Internet gateway has been detected. Simulations show this significant benefit in static scenarios. Therefore the Load Switching extension is recommended for applications in static scenarios.

The thesis' extensions are to improve the provided bandwidth to mobile nodes. One could think about the improvement of the packet delay quality of service constraint for Internet connected mobile ad-hoc network nodes. This could be achieved by taking the

round trip time from a specific mobile node to all discovered Internet gateways. To get more suitable results from the round trip time in a background traffic charged ad-hoc network the mobile node should send more than one probing packet to all Internet gateways in order to avoid problems with outliers since, as a result of this thesis, background traffic is found to have a strong impact on ad-hoc networks.

The introduction, presentation and investigation of the new HELLO message based algorithm and its comparison with the well known advertisement based and solicitation based algorithms for Internet gateway discovery is the first goal of this thesis. This goal is achieved.

Furthermore, the thesis presents two extensions to Internet gateway discovery algorithms that increase the throughput to mobile nodes of an ad-hoc network. The first extension (GRREP) is portable to all proactive algorithms for Internet gateway discovery that are based on reactive algorithms for multihop ad-hoc routing. In spite of the increasing amount of control overhead this first extension provides more throughput to the mobile nodes. The second extension (Load Switching between Internet gateways) is portable to all ad-hoc routing protocols that are to be extended with Internet gateway discovery features and it provides a significant benefit in static scenarios. The goal of the two extensions to increase the amount of throughput to mobile nodes of an ad-hoc mobile network is achieved.

Appendix A

Throughput in Wireless Multihop Environments

This chapter discusses the maximum throughput of multihop wireless connections in ad-hoc networks. The throughput of such a connection depends mainly on the length of the route between a source and a destination node.

Simulations were carried out to get an idea of the maximum throughput of a multihop ad-hoc connection when downloading a test file of 1 MB in size with FTP/TCP and a packet size of 1000 bytes. The simulation scenario consists of one Internet gateway (GW) connecting the correspondent node in the Internet (CN) via an access point and the testing mobile (MN) in the ad-hoc network via an ad-hoc multihop route. The simulated throughput of a one hop (direct) connection between the Internet gateway and the testing mobile node is defined as 100% throughput and used to scale the results in chapter 7 (unless otherwise stated). By creating the same test scenario using different protocols, packet sizes, and algorithms one can easily compare results of theses. The scaling to 100% allows a comparison with simulation results of other layer 2 protocols that provide different bandwidths. Thus the simulation results in chapter 7 can be easily transformed to other layer 2 protocols by only determining the 100% value.

To demonstrate the influence of the route's length on the throughput the route from the Internet gateway to the mobile node is lengthened by up to six intermediate nodes. The simulation results are compiled in Table A.1. In Figure A.1 the results are depicted. On the x-axis the hop distance between the Internet gateway and the testing

mobile node is given while the y-axis represents the absolute throughput in $\frac{\text{kbit}}{\text{s}}$ of the file transfer from the CN via the Internet gateway to the mobile node. Concluded from the simulation results, in this thesis a throughput value of 100% equals an absolute throughput of 1882.2 $\frac{\text{kbit}}{\text{s}}$ (unless otherwise stated).

# of interm. nodes	hop distance	throughput
0	1	1882 ± 30
1	2	1009 ± 16
2	3	410 ± 10
3	4	168 ± 3
4	5	133 ± 2
5	6	120 ± 2
6	7	114 ± 2

Table A.1: Throughput of a TCP file download via a multihop route

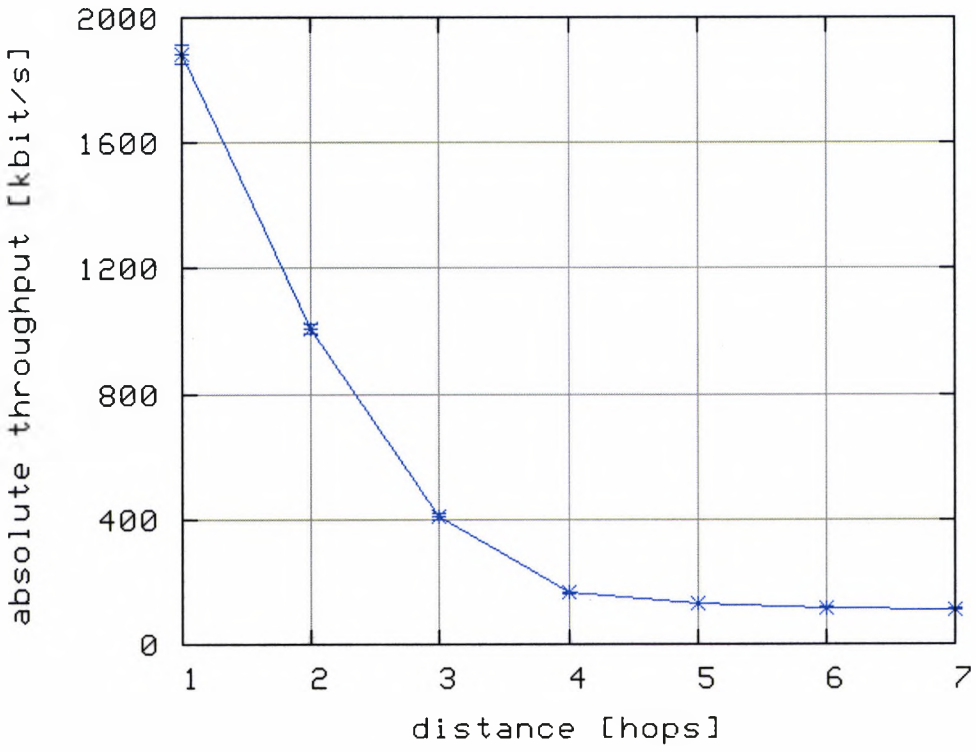


Figure A.1: Throughput of a TCP file download via a multihop route

Appendix B

Normalised Protocol Overhead

To allow a comparison of the protocol overhead a specific Internet gateway discovery algorithm generates for Internet connectivity the thesis uses a standardised protocol overhead. This standardised protocol overhead is related to a network consisting of 30 nodes where every node generates one control message per second. This control message is the HELLO message every node using AODV without cross-link features is sending. Hello messages were chosen as the basis for the normalised control overhead since they are the only messages sent independently of a MANET's topology, mobility, or background traffic.

Since the thesis lets the mobile nodes time to establish an ad-hoc network (the initial establishment of an ad-hoc mobile network is assumed to cause overhead above average) the measuring starts at $t_{\mathrm{SIM}} = 50$ seconds. Thus, the normalised protocol overhead is always related to the time from $t_{\mathrm{SIM}} = 50$ seconds to the end of the simulation run at $t_{\mathrm{SIM}} = 900$ seconds.

A normalised protocol overhead of 1 equals a protocol overhead of 25500 control messages for a total simulation time of 850 seconds ((900 seconds – 50 seconds) · 30 nodes).

Bibliography

Author's Publications

[1] M. Rosenschon, et al.; *New Implementations into Simulation Software NS-2 for Routing in Wireless Ad-Hoc Networks*; Eurosim 2004, Paris, France, September 2004

[2] M. Rosenschon, et al.; *Gateway Discovery Algorithm for Ad-Hoc Networks Using HELLO Messages*; IWWAN 2005, London, United Kingdom, May 2005

[3] M. Rosenschon, et al.; *Performance Comparison of Gateway Discovery Algorithms in Ad Hoc Networks with Mobile Nodes*; European Wireless 2006, Athens, April 2006

Other Publications

[4] The VINT-Project; *The Network Simulator - ns-2*; http://www.isi.edu/nsnam/ns/

[5] The VINT Project; *The ns Manual (formerly ns Notes and Documentation)*; http://www.isi.edu/nsnam/ns/doc/

[6] The VINT Project; *Tutorial for the Network Simulator "ns"*; http://www.isi.edu/nsnam/ns/tutorial

[7] R. Wakikawa, et al.; *Global Connectivity for IPv6 Mobile Ad Hoc Networks*; http://www.wakikawa.net/Research/paper/draft/manet/↩ draft-wakikawa-manet-globalv6-03.txt

[8] Internet World Stats, Usage and Population Statistics; *INTERNET USAGE STATISTICS - The Big Picture*; http://www.internetworldstats.com/stats.htm

[9] IETF; Network Working Group; *Dynamic Host Configuration Protocol*; http://www.ietf.org/rfc/rfc2131.txt

[10] IPv6; http://www.ipv6.org/

[11] IPv4; *INTERNET PROTOCOL, DARPA INTERNET PROGRAM, PROTOCOL SPECIFICATION*; http://rfc.net/rfc791.html; September 1981

[12] D. Johnson, et al.; *Mobility Support in IPv6*; Network Working Group; RFC 3775; http://www.ietf.org/rfc/rfc3775.txt

[13] C. Perkins; *Mobile IP: Design Principles and Practises*; Addison-Wesley, ISBN: 0-201-63469-4; 1998

[14] D. Yu, et al.; *Path Availability in Ad Hoc Network*; ICT 2003; Feb 2003

[15] C. E. Perkins, P. Bhagwat; *Highly Dynamic Destination-Sequenced Distance-Vector Routing (DSDV) for Mobile Computers*; Computer Communications Review, pp. 234-244, Oct. 1994

[16] D. B. Johnson, et al.; *The Dynamic Source Routing Protocol for Mobile Ad Hoc Networks (DSR)*; http://www.ietf.org/internet-drafts/draft-ietf-manet-dsr-10.txt

[17] J. Broch, et al.; *Supporting Hierarchy and Heterogeneous Interfaces in Multi-Hop Wireless Ad Hoc Networks*; International Symposium on Parallel Architectures, Algorithms and Networks (ISPAN99), Workshop on Mobile Computing, Perth, Western Australia, June 1999

[18] T. Clausen, et al.; *Optimized Link State Routing Protocol (OLSR)*; http://ietf.org/rfc/rfc3626.txt

[19] Z. Haas, et al.; *The Zone Routing Protocol (ZRP) for Ad Hoc Networks*; http://www.ietf.org/proceedings/02nov/I-D/draft-ietf-manet-zone-zrp-04.txt

[20] Network Working Group; *Ad hoc On-Demand Distance Vector (AODV) Routing*; http://rfc.net/rfc3561.html

[21] C. Perkins, et al.; *AODV for IPv6*; http://www.join.uni-muenster.de/Dokumente/drafts/draft-perkins-aodv6-01.txt http://tools.ietf.org/wg/manet/draft-ietf-manet-longlived-adhoc-routing/↩ draft-ietf-manet-longlived-adhoc-routing-00.txt

[22] S. Das, et al.; *Performance Comparison of Two On-demand Routing Protocols for Ad Hoc Networks*; IEEE Personal Communications, February 2001

[23] U. Jönsson, et al.; *MIPMANET - Mobile IP for Mobile Ad Hoc Networks*; Proceedings ACM MobiHoc, 2000

[24] Y. Sun, et al.; *Internet Connectivity for Ad hoc Mobile Networks*; International Journal of Wireless Information Networks, special issue on "Mobile Ad Hoc Networks (MANETS): Standards, Research, Applications"; April 2002

[25] J. Xi, C. Bettstetter; *Wireless Multihop Internet Access: Gateway Discovery, Routing, and Addressing*; Proceedings International Conference on Third Generation Wireless and Beyond; 3GWireless, San Francisco, USA, May 2002

[26] J. Lee, et al.; *Hybrid gateway advertisement scheme for connecting mobile ad hoc networks to the Internet*; VTC Spring 2003

[27] M. Ghassemian, et al.; *Performance Analysis of Internet Gateway Discovery Protocols in Ad Hoc Networks*; WCNC 2004; March 2004

[28] E. Belding-Royer, C. Perkins; *Evolution and Future Directions of the Ad hoc On-demand Distance Vector Routing Protocol*; Networks Journal, Pages 125-150, July 2003

[29] P. Ruiz, A. Gomez-Skarmeta; *Enhanced Internet Connectivity for Hybrid Ad hoc Networks Through Adaptive Gateway Discovery*; International Conference on Local Computer Networks, LCN 04, Tampa, Florida, November 2004

[30] P. Ratanchandani, R. Kravets; *A Hybrid Approach to Internet Connectivity for Mobile Ad Hoc Networks*; Proceedings of WCNC 2003, Volume 3, Pages 1522-1527, March 2003

[31] M. Michalak, T. Braun; *Common Gateway Architecture for Mobile Ad-hoc Networks*; Wireless On demand Network Systems and Services, WONS 2005, St. Moritz, Switzerland, January 2005

[32] C. Perkins; *AD HOC NETWORKING*; Addison Wesley, ISBN 0-201-30976-9

[33] J. Osterhoud; *Scripting: Higher Level Programming for the 21st Century*; http://home.pacbell.net/ouster/scripting.html; Computer Magazine, March 1998

[34] A. Hamidian; *A Study of Internet Connectivity for Mobile Ad Hoc Networks in NS 2*; Department of Communication Systems Lund Institute of Technology, Lund University, Sweden, January 2003

[35] M. Heurung; *Softwareanalyse und Erweiterung des Netzwerksimulators NS im Rahmen des deutschen Forschungsprojektes "'IP on Air"'* (in german); Fachhochschule Giessen-Friedberg, University of Applied Sciences, Friedberg, Germany, April 2003

[36] T. Mänz; *Entwicklung und Implementation eines Mobilittsprotokolls für den Netzwerksimulator NS-2 im Rahmen des deutschen Forschungsprojektes "IP on Air"* (in german); Fachhochschule Giessen-Friedberg, University of Applied Sciences, Friedberg, Germany, November 2004

[37] IEEE Computer Society; *Carrier sense multiple access with collision detection (CSMA/CD) access method and physical layer specifications*; http://standards.ieee.org/getieee802/download/802.3-2002.pdf

[38] *IPonAir NEXT GENERATION WIRELESS INTERNET*; http://www.iponair.de

[39] IEEE Computer Society; *Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications*; http://standards.ieee.org/getieee802/download/802.11-1999.pdf

[40] Multiband OFDM Alliance; *Ultrawideband: High-speed, short-range technology with far-reaching effects*; http://www.multibandofdm.org/papers/MBOA_UWB_White_Paper.pdf

[41] Network Working Group; *Address Allocation for Private Internets*; RFC 1918; http://http://www.ietf.org/rfc/rfc1918.txt?number=1918

[42] Network Working Group; *The IP Network Address Translator (NAT)*; RFC 1631; http://www.ietf.org/rfc/rfc1631.txt?number=1631

[43] Bluetooth; *IEEE 802.15 Working Group for WPAN*; http://grouper.ieee.org/groups/802/15/

[44] The IEEE 802.16 Working Group on Broadband Wireless Access Standards; *IEEE 802.16 LAN/MAN Broadband Wireless LANS*; http://standards.ieee.org/getieee802/802.16.html

[45] S. Ruffino; *Automatic configuration of IPv6 addresses for nodes in a MANET with multiple gateways*; http://www.ietf.org/internet-drafts/draft-ruffino-manet-autoconf-multigw-00.txt

[46] B. Xie, A. Kumar; *Integrated Connectivity Framework for Internet and Ad hoc Networks*; 1st IEEE International Conference on Mobile Ad-hoc and Sensor Systems, October 2004, Fort Lauderdale, Florida, USA

[47] Free Software Foundation; *GNU GENERAL PUBLIC LICENSE*; http://www.gnu.org/copyleft/gpl.html

[48] Network Working Group; *Routing Information Protocol*; http://www.ietf.org/rfc/rfc1058.txt?number=1058

[49] Network Working Group; *OSPF Version 2*; http://www.ietf.org/rfc/rfc1247.txt

[50] Network Working Group; *A Border Gateway Protocol (BGP)*; http://www.ietf.org/rfc/rfc1105.txt?number=1105

[51] C. Perkins, et al.; *IP Address Autoconfiguration for Ad Hoc Networks*; http://people.nokia.net/charliep/txt/aodvid/autoconf.txt

[52] C. Lin, J.-S. Liu; *QoS Routing in Ad Hoc Wireless Networks*; IEEE Journal of Selected Areas in Communications, vol. 19, no. 8, August 1999

[53] B. Zhang, H. Mouftah; *QoS Routing for Wireless Ad Hoc Networks: Problems, Algorithms, and Protocols*;IEEE Communications Magazine, October 2005

[54] C. Zhu, M. Corson; *QoS routing for mobile ad hoc networks*; In Proceedings of IEEE Infocom, Anchorage, Alaska, June 2001

[55] Network Working Group, R. Braden, et al.; *Resource ReSerVation Protocol (RSVP)*; http://tools.ietf.org/html/rfc2205

[56] R. Sivakumar, et al.; *Core Extraction Distributed Ad hoc Routing (CEDAR)*; http://tools.ietf.org/id/draft-ietf-manet-cedar-spec-00.txt

[57] Q. Xue, A. Ganz; *Ad hoc QoS on-demand routing (AQOR) in mobile ad hoc networks*; Journal of Parallel and Distributed Computing archive Volume 63, February 2003, Special issue on Routing in mobile and wireless ad hoc networks, Pages: 154 - 165, ISSN: 0743-7315

[58] R. Leung, et al.; *MP-DSR: A QoS-aware Multi-path Dynamic Source Routing Protocol for Wireless Ad-Hoc Networks*; In Proc. of 26th Annual IEEE Conference on Local Computer Networks (LCN 2001), Tampa, Florida, USA, 2001

[59] C. Perkins, et al.; *Quality of Service for Ad hoc On-Demand Distance Vector Routing*; http://people.nokia.net/charliep/txt/aodvid/qos.txt

[60] H. Li, et al.; *Multihop Communications in Future Mobile Radio Networks*; 13th IEEE International Symposium on Personal, Indoor and Mobile Radio Communications, 2002. ISBN: 0-7803-7589-0

[61] J. Postel; *User Datagram Protocol*; RFC 768; http://rfc.net/rfc768.html

[62] Information Sciences Institute, University of Southern California; *TRANSMISSION CONTROL PROTOCOL*; RFC793; http://rfc.net/rfc793.html

[63] IEEE Standards Association; *IEEE 802.16 LAN/MAN Broadband Wireless LANS*; http://standards.ieee.org/getieee802/802.16.html

[64] P. Almquist; *Type of Service in the Internet Protocol Suite*; RFC 1349; http://rfc.net/rfc1349.html

[65] H. Schulzrinne, et al.; *RTP: A Transport Protocol for Real-Time Applications*; http://www.ietf.org/rfc/rfc1889.txt

[66] Y. Sun, E. Belding-Royer; *Application-Oriented Routing in Hybrid Wireless Networks*; Next Generation Internet Symposium, Anchorage, Alaska, USA, May 2003

[67] K. Nichols; *Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers*; http://tools.ietf.org/html/rfc2474

[68] S. Blake; *An Architecture for Differentiated Services*; http://tools.ietf.org/html/rfc2475

[69] S. Ivanoc, et al.; *Experimental Validation of the ns-2 Wireless Model using Simulation, Emulation, and Real Network*; Proceedings of the 4th Workshop on Mobile Ad-Hoc Networks (WMAN'07); Bern, Switzerland, February/March 2007, ISBN 978-3-8007-2980-7

[70] D. Kotz, et al.; *Experimental evaluation of wireless simulation assumptions*; Proceedings of the 4th Workshop on Mobile Ad-Hoc Networks (WMAN'07); Proceedings of the 7th ACM international symposium on Modeling, analysis and simulation of wireless and mobile systems; Venice, Italy; 2004; ISBN 1-58113-953-5

[71] ISO/IEC; *Information technology - Open Systems Interconnection - Basic Reference Model: The Basic Model*; http://standards.iso.org/ittf/PubliclyAvailableStandards/s020269_ISO_IEC_7498-1_1994(E).zip

[72] E. W. Dijkstra; *note on two problems in connexion with graphs*; Numerische Mathematik. 1 (1959), pages 269271

[73] W3C, World Wide Web consortium; *W3C, World Wide Web consortium*; http://www.w3.org

[74] V.Fuller, et al.; *Classless Inter-Domain Routing (CIDR): an Address Assignment and Aggregation Strategy*; http://www.ietf.org/rfc/rfc1519.txt

[75] J. Postel, J. Reynolds; *TELNET PROTOCOL SPECIFICATION*; http://www.ietf.org/rfc/rfc0854.txt

[76] 3GPP; *3GPP, a Global Initiative*; http://www.3gpp.org

[77] D. B. Johnson; *Validation of Wireless and Mobile Network Models and Simulation*; Proceedings of the DARPA/NIST Network Simulation Validation Workshop, Fairfax, Virginia, USA, May 1999

[78] G. Lucio; *OPNET Modeler and Ns-2: Comparing the Accuracy of Network Simulators for Packet-Level Analysis using a Network Testbed*; Proceedings of 3rd WEAS International Conference on Simulation, Modeling and Optimization (ICOSMO 2003), October 2003

[79] Technologies, Inc.; *OPNET*; http://www.opnet.com/