Erlend Larsen

Increasing the Performance of MANETs

Throughput and QoS Performance Enhancing Mechanisms for Unicast and Group Communication in Proactive Mobile Ad Hoc Networks

Thesis for the degree of Philosophiae Doctor

Trondheim, January 2011

Norwegian University of Science and Technology
Faculty of Information Technology, Mathematics and Electrical Engineering
Department of Telematics

NTNU – Trondheim
Norwegian University of
Science and Technology

NTNU
Norwegian University of Science and Technology

Thesis for the degree of Philosophiae Doctor

Faculty of Information Technology, Mathematics and Electrical Engineering
Department of Telematics

ISBN 978-82-471-2559-5 (printed ver.)
ISBN 978-82-471-2560-1 (electronic ver.)
ISSN 1503-8181

Doctoral theses at NTNU, 2011:21

Printed by NTNU-trykk

Abstract

Mobile Ad hoc NETworks (MANETs) have the potential for increasing the information flow in emergency and rescue operations. Rapid response in areas without existing infrastructure is currently limited to single hop technologies, primarily voice communication using "walkie-talkies".

While MANETs can offer multi-hop broadband communication for emergency and rescue operations, there are challenges that currently limit their usefulness. These challenges are linked to aspects such as capacity, density, collisions and mobility, all of which affect the Quality of Service (QoS) and Quality of Experience (QoE) offered by the MANET.

The work in this thesis investigates problems related both to unicast and group communication for MANETs. In unicast communication, it is shown that the delay between a link break and the rerouting is affected by the interface queue size and the traffic load. Several solutions to reduce the rerouting time are proposed, e.g., reducing the interface queue, or avoiding links that are likely to be broken. Along with the rerouting time, the throughput in and out of a MANET is also affected by the position of one or more gateways.

Further, it is shown that the QoS of group communication voice traffic can be maintained with other traffic in the network, using preemptive mechanisms. Finally, a typical group communication forwarding algorithm is improved by combining it with another algorithm. Also, a preemptive selection of algorithm for lower priority traffic when voice traffic is active in the network is shown to improve the overall QoS for both the voice and lower priority traffic.

Preface

This thesis is submitted to the Norwegian University of Science and Technology (NTNU) for partial fulfillment of the requirements for the degree of philosophiae doctor. The work for this dissertation started in December 2005.

The work was mainly carried out at UniK - University Graduate Center, under supervision of Professor Øivind Kure at the Norwegian University of Science and Technology (NTNU) and Professor Paal Einar Engelstad at the University of Oslo, Telenor GBD&R and Simula Research Laboratory.

The thesis work was financed through two EU projects, the ITEA Easy Wireless project with its Norwegian participating project *Quality of Service in Ad-hoc Networks (QUAD)*, and the CELTIC *Deployable High Capacity Gateway for Emergency Services (DeHiGate)* project. The Norwegian parts of these projects were funded by the Research Council of Norway and managed by Thales Norway AS. The study was supported by UniK, Applica and Baseline Communications. In addition my employer, FFI (The Norwegian Defence Research Establishment), was generous to grant me leave of absence for the main duration of the thesis work.

Both the QUAD and the DeHiGate projects worked on solutions for main problem areas within wireless broadband ad hoc technology for emergency operations. The Easy Wireless (EW) project focused on QoS in heterogeneous networks. QUAD addressed a set of sub-goals of the larger EW project, where the main sub-goal was defining and implementing architectures and mechanisms for predictable QoS in wireless ad hoc networks.

The DeHiGate project aimed at developing a broadband deployable gateway between wireless broadband ad hoc networks and the fixed infrastructure. The key idea was to integrate many available wireless technologies (e.g., WiMAX, Wi-Fi, TETRA/TEDS and GPRS) into a single system. This would allow the responding personnel to communicate within the group and back to the Headquarters (HQ) using the best combination of access networks, improving the efficiency and in-

creasing the safety during an emergency response.

Acknowledgments

First and foremost, I would like to thank my supervisors, Professors Øivind Kure and Paal E. Engelstad. Without their persistence in requesting results, and without their willingness to share of their vast knowledge through discussions and lectures, I cannot imagine that this thesis would ever have been realized.

Second, I want to thank Thales Norway AS for initiating the projects Quad and DeHiGate, and thus giving me the opportunity to carry out work within these projects. I would also like to thank the other Norwegian partners in these two projects, Applica and Baseline Communications, for our collaboration.

I want to thank my employer, the Norwegian Defence Research Establishment (FFI), for giving me the opportunity to improve my skills in wireless and ad hoc networking, and extending my leave when I needed it.

I would also like to thank my former project manager at FFI, Professor Knut Øvsthus, now at Bergen University College, who acknowledged my ambitions and arranged the circumstances so that I was able to start this work.

All my colleagues at UniK deserve my gratitude, especially Vinh Pham and Lars Landmark, for being willing discussion partners and showing much interest in my research topics. An important weekly diversion was the UniKum's floorball activity, and I want to thank all opponents and fellow players. "4 is the magic number, Jakob."

Finally, I want to thank my family, and especially my wife Kikki. Countless times, she has urged me onwards, making me forget the worries and concerns. "Du og je', Kikki!"

List of Publications

The author of this thesis has shared primary authorship of paper A, and has the primary authorship of papers B through E (appended as Part II of the thesis.) The papers A through F are co-authored with the external, technical supervisors. The author of this thesis has contributed to papers F and G as a discussion partner, and provided the initial simulation code for paper F. All papers were published through peer-reviewed conferences and workshops.

PAPER A: V. Pham, E. Larsen, K. Øvsthus, P. Engelstad, and Ø. Kure, "Rerouting Time and Queueing in Proactive Ad Hoc Networks," In proceedings of the Performance, Computing, and Communications Conference (IPCCC), New Orleans, USA, April 11–13, 2007, pp. 160–169.

PAPER B: E. Larsen, V. Pham, P. Engelstad, and Ø. Kure, "Gateways and Capacity in Ad Hoc Networks," In proceedings of the International Conference on Advances in Human-oriented and Personalized Mechanisms, Technologies, and Services, (I-CENTRIC), Sliema, Malta, October 26–31, 2008, pp. 390–399, ISBN: 978-0-7695-3371-1

PAPER C: E. Larsen, L. Landmark, V. Pham, Ø. Kure and P. E. Engelstad, "Routing with Transmission Buffer Zones in MANETs," In proceedings of the IEEE International Symposium on a World of Wireless Mobile and Multimedia Networks (WoWMoM), Kos, Greece, June 15–18, 2009, ISBN: 978-1-4244-4440-3.

PAPER D: E. Larsen, L. Landmark, V. Pham, P. E. Engelstad, and Ø. Kure, "Preemption Mechanisms for Push-to-Talk in Ad Hoc Networks," In proceedings of the 34th IEEE Conference on Local Computer Networks (LCN), Zürich, Switzerland, October 20–23, 2009, pp. 428–435, ISBN: 978-1-4244-4488-5.

PAPER E: E. Larsen, L. Landmark, V. Pham, Ø. Kure, and P. E. Engelstad, "Optimized Group Communication for Tactical Military Networks," In proceedings of the IEEE Military Communications Conference (MILCOM), San Jose, CA, USA, October 31–November 4, 2010, pp. 1445–1451, ISBN: 978-1-4244-8179-8.

Related papers:

PAPER F: V. Pham, E. Larsen, Ø. Kure, and P. E. Engelstad, "Routing of Internal MANET Traffic over External Networks," Mobile Information Systems Journal, iiWAS/MoMM special issue, Volume 5, Number 3, 2009, pp. 291–311, ISSN: 1574-017X.

PAPER G: V. Pham, E. Larsen, P. E. Engelstad, and Ø. Kure, "Performance Analysis of Gateway Load Balancing in Ad Hoc Networks with Random Topologies," Proceedings of the 7th ACM International Symposium on Mobility Management and Wireless Access (Mobiwac), Tenerife, Canary Islands October 26-30, 2009, pp. 66–74, ISBN: 978-1-60558-617-5.

Contents

List of Figures

List of Terms and Acronyms

ACK	Acknowledgment
AMRoute	Ad hoc Multicast Routing protocol
AODV	Ad hoc On-Demand Distance Vector Routing
AQM	Ad hoc QoS Multicasting
BCD	Bottleneck Collision Domain
BER	Bit Error Rate
CAC	Call Admission Control
CAMP	Core-Assisted Mesh Protocol
CBT	Core Based Tree
CDS	Connected Dominating Set
CEDAR	Core Extraction Distributed Ad Hoc Routing
CF	Classic Flooding
CSMA	Carrier Sense Multiple Access
CSMA/CA	Carrier Sense Multiple Access with Collision Avoidance
CTS	Clear To Send
CW	Contention Window
DCF	Distributed Coordination Function
DDM	Differential Destination Multicast
DeHiGate	Deployable High Capacity Gateway for Emergency Services
DIFS	Distributed Inter-Frame Space

DPD	Duplicate Packet Detection
DSR	Dynamic Source Routing
DYMO	Dynamic MANET On-Demand Protocol
E2M	Extended Explicit Multicast
ERS	Expanding Ring Search
ETT	Expected Transmission Time
ETX	Expected Transmission Count
EW	Easy Wireless
FEC	Forward Error Correction
fPrIM	Fixed Protocol Interferences Model
GPRS	General Packet Radio Service
GPS	Global Positioning System
GW	Gateway
HNA	Host and Network Association
HQ	Headquarters
iAWARE	Interference aware routing metric
IEEE	Institute of Electrical and Electronics Engineers
IETF	Internet Engineering Task Force
IGMP	Internet Group Management Protocol
IP	Internet Protocol
IR	Interference Ratio
LAN	Local Area Network
LLN	Link Layer Notification
LPW	Low Priority Window
MAC	Medium Access Control
MACT	Multicast Activation
MANET	Mobile Ad hoc NETwork

MANSI	Multicast for Ad Hoc Networks with Swarm Intelligence
MAODV	Multicast Ad-hoc On-demand Distance Vector
MCEDAR	Multicast Core-Extraction Distributed Ad hoc Routing
MHC	Minimum Hop Count
MID	Multiple Interface Declaration
MOLSR	Multicast Optimized Link State Routing
MPR	MultiPoint Relay
MTU	Maximum Transmission Unit
NAV	Network Allocation Vector
NHDP	NeighborHood Discovery Protocol
NS-2	Network Simulator 2
NS-MPR	Non-Source-based MultiPoint Relay
ODMRP	On-Demand Multicast Routing Protocol
OLSR	Optimized Link State Routing
OSI	Open Systems Interconnection
OSPF-MDR	Open Shortest Path First with MANET Designated Routers
PCF	Point Coordination Function
PDR	Packet Delivery Ratio
PHY	Physical
PTT	Push-to-Talk
QAMNet	QoS-Aware Mesh Construction to Enhance Multicast Routing in Mobile Ad Hoc Networks
QMOST	QoS-aware Multicast Overlay Spanning Tree
QoE	Quality of Experience
QOLSR	Quality of Service for Ad hoc Optimized Link State Routing Protocol
QoS	Quality of Service
QUAD	Quality of Service in Ad-hoc Networks

RCL	Route Change Latency
RD	Random Direction
RERR	Route Error
RFC	Request For Comments
RREP	Route Reply
RREQ	Route Request
RSSI	Received Signal Strength Indicator
RTS	Request To Send
RWR	Random Walk with Reflection
S-MPR	Source-based MultiPoint Relay
SA	Situational Awareness
SIFS	Short Inter-Frame Space
SMF	Simplified Multicast Forwarding
SSA	Signal Stability-Based Adaptive Routing Protocol
STA	Wireless Stations
SWAN	Service Differentiation in Stateless Wireless Ad Hoc Networks
TC	Topology Control
TCP	Transmission Control Protocol
TDMA	Time Division Multiple Access
TEDS	TETRA Enhanced Data Service
TETRA	Terrestrial Trunked Radio
UAV	Unmanned Aerial Vehicle
UDG	Unit Disk Graph
UDP	User Datagram Protocol
WCETT	Weighted Cumulative ETT
Wi-Fi	Wireless Fidelity
WiMAX	Worldwide Interoperability for Microwave Access

WG	Work Group
WLAN	Wireless LAN
WSN	Wireless Sensor Network
XF	Explicit Multicast Forwarder
ZRP	Zone Routing Protocol

Part I

Introduction

Chapter 1

Introduction

1.1 Motivation

Mobile Ad hoc NETworks (MANETs) have the potential for increasing the information flow in emergency and rescue operations. Rapid response in areas without existing infrastructure is currently limited to single hop technologies, primarily voice communication using "walkie-talkies".

MANETs can offer voice communication, situational awareness through position sharing and geographically mapped events, access to maps and construction drawings, and support for other types of improvised communication. The use of such services can increase the effectiveness of an operation.

While MANETs are able to offer multi-hop broadband communication for emergency and rescue operations, there are challenges that currently limit their usefulness. These challenges are linked to aspects such as capacity, density, collisions and mobility, all of which affect the Quality of Service (QoS) and Quality of Experience (QoE) offered by the MANET.

1.2 Challenges

Ad hoc networks are intended to function without any other infrastructure, dynamically forming temporary networks. This temporary and dynamic nature makes it hard to use a synchronized Time Division Multiple Access (TDMA) protocol that could be more efficient with respect to QoS. Instead, random access protocols us-

ing Carrier Sense Multiple Access (CSMA) are preferred. While these protocols are better suited for distributed networking, collisions and contention lead to lower medium utilization and thus less predictable QoS.

In a MANET, any node may appear or disappear without warning. Centralized routing, which would ensure that all nodes have the same view of the topology, is all but impossible in MANETs, due to the uncertainty of the fate of any one node. Therefore, the MANET routing needs to be decentralized, and all ad hoc networks employ distributed routing.

The data capacity – or throughput – of MANETs is a widely used performance metric. The applications' demand for capacity may surpass the available capacity in the network, rendering the service unsupported by the network. The available capacity is dependent on many parameters, such as the physical data rate, the number of hops to travel, the bit error rate, the link break probability, the routing overhead, competing traffic, etc. The work in this thesis is aimed at understanding and improving both the throughput and the packet loss rate, either for the entire network, or for selected types of traffic. The papers A and B focus on understanding and enhancing the throughput, while the packet loss rate is the main enhancement metric in the papers C, D and E.

One of the major factors affecting the data capacity of an ad hoc network is the density, i.e., the average number of nodes in radio range of any other node. This was established as early as in 1978 by Kleinrock and Silvester [1], where the node transmission radius is investigated. When nodes have a large transmission radius, this gives a high degree of connectivity, but also creates much interference and loss of channel throughput. With a reduced transmission radius, the interference is limited. This increases the network capacity, but at the same time also increases the number of hops that a packet must travel to reach its destination. If the transmission range is reduced too much, partitioning will occur, rendering parts of the network disconnected.

Reduced link quality caused by distance or obstacles is another MANET challenge that may cause routing protocols to be unable to function properly. The gray zone problem [2] is one example. Unfortunately, many simulators are unable to offer a physical layer with enough detail to simulate this behavior, and many network layer protocols have not been tested properly in such environments [3].

Another challenge in MANETs is the topology dynamics, caused by node mobility. This can vary greatly as a scenario unfolds, both with regards to velocity, direction, and coordination. The node movement can vary from highly coordinated in fixed patterns without relative mobility, to completely uncoordinated behavior, triggering rapid link changes requiring rerouting. The dynamics may also be different

at different regions in the network, with some links being very stable, and others breaking frequently. The physical topography can also affect how and how often link changes happen, both by affecting the participators and through blocking radio propagation. An example is a road or a valley which can encourage coordinated node movement, while a forest or a building may block potential links. With such a variety of topology dynamics, it is challenging to design and test mechanisms aimed at being optimal for a wide range of these. All the papers in the thesis study effects of mobility in some form. While paper A looks at a simple case of rerouting from single-hop to multi-hop, the papers B through E look at mobility in a more general way, with nodes moving randomly inside the simulation area.

The interaction between the different layers in the Open Systems Interconnection (OSI) network reference model [4] may also cause sub-optimal utilization of network resources. The strength of the OSI layered stack is that each layer can/shall be designed and implemented separately, facilitating independent development of protocols of each layer. However, independent development may lead to unexpected behavior when the protocols are employed. Such was the case in paper A, where the retransmissions at the Institute of Electrical and Electronics Engineers (IEEE) 802.11 Medium Access Control (MAC) layer, designed to remedy the high Bit Error Rate (BER) of the wireless medium, actually prolonged the rerouting time. Such problems can be addressed through cross-layer solutions, where more information can be shared between the layers, making the network more efficient. In paper D, the solution needed implementation at both the routing layer and the MAC layer, to improve the reaction to priority traffic events, and in paper E, the radio load metric was acquired by calculations at the MAC layer, and passed up to the routing layer. However, tweaking or changing the behavior of a specific layer may affect the performance of other layers. Combining several such performance enhancing changes may actually be detrimental for the performance of the network [5]. Cross-layer mechanisms may quickly become too complex, and should be used with caution.

Finally, it is a challenge to maintain QoS and fairness in MANETs. The random access channel and interference from other transmissions, together with varying path distances, make it a challenge to ensure QoS in ad hoc networks. While the possibilities to manipulate priorities and admission control are better at the MAC layer of the network, mechanisms can also be implemented at the routing layer. Both the papers D and E study how the routing protocol may contribute to enhancing the QoS. In paper D, the routing protocol, together with information and action at lower layers, initiates preemption of low priority traffic to achieve better service for the high priority traffic, and in paper E the radio load, in combination with properties of the multicast forwarding algorithms, are used to achieve improved

performance for the priority traffic.

1.3 Methods

There are three main methods for investigating MANETs: theoretical analysis, simulations and experiments. Theoretical analysis can give fundamental knowledge about investigated mechanisms and systems. Simulations, on the other hand, enable the investigation of the dynamics occurring when the distributed interaction is too complex to model using theoretical analysis, especially in combination with mobility. Even though simulations provide an easy way to investigate the distributed properties of algorithms, simulators cannot simulate the world in its entirety, but have different areas where they are strong and weak. The simulator employed in this thesis work is not very good at abstracting the physical world. In cases where attributes of the physical layer (i.e., real world properties) are defining for the system performance or other investigated features, experiments with real equipment in the desired conditions should be preferred. However, performing experiments with more than a few nodes requires a great effort from participators in the experiment, especially in order to support realistic mobility. The work load prior to the experiment is high, since the equipment must be prepared with the correct software versions, charged batteries etc. Thus, the infrastructure required to perform large experiments makes it more cost effective to perform research using simulators and theoretical analysis.

The objective of the thesis work was to examine and improve QoS provisioning in an ad hoc rescue network. This was to be achieved through:

- Building up expertise on ad hoc protocols, QoS and how typical emergency scenarios evolve in general.
- Investigating and analyzing models for providing QoS at different OSI layers in an ad hoc setting.
- Developing models for QoS provisioning within the link and the network layers, in addition to a working admission control.

The performance results presented in this thesis were obtained using simulations. In the introductory phase of the work on paper A, experiments were performed using real laptops. However, it became evident that the implementations of the drivers were difficult to investigate. The issues with rerouting time were then investigated further in the simulator, which gave much easier access to the inner workings of the MAC protocol. In the subsequent papers, larger topologies were

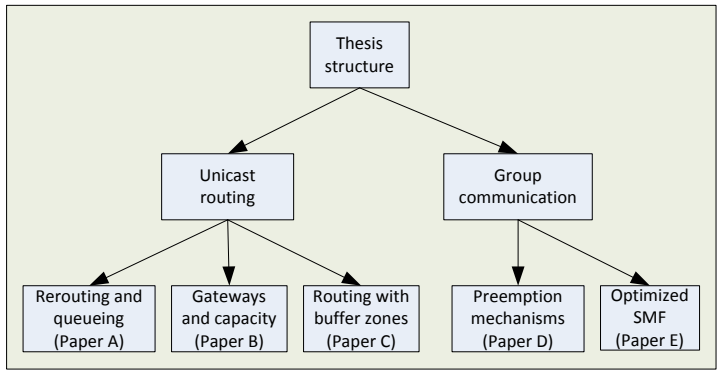


Figure 1.1: Structure of the thesis work.

investigated, and these topologies would have required considerable resources to perform investigations using experiments rather than simulations. Therefore, the work for the thesis papers was performed mainly using simulations as the preferred method of research.

The research methodology employed for the simulations is described further in Chapter 4, and also in greater detail in each of the papers in Part II.

1.4 A brief overview of the work

A brief introduction to the thesis work and the results defining the course of the research project are presented here, while the main contributions are presented in more detail in Chapter 5. The thesis addresses two main areas, *unicast routing* and *group communication*, and the overview of the work is ordered accordingly.

1.4.1 Unicast routing

Unicast routing, supporting one-to-one communication in MANETs, is the first of the two main areas of the thesis work (Figure 1.1). The work consists of three papers A, B and C, which aim at understanding some of the mechanisms that impact the network throughput. While the papers A and C mainly look at mobility, other factors such as interference, path length and channel reuse are identified in paper B as affecting the throughput. An increase in the aggregated network throughput, as achieved by the mechanisms and solutions in the papers A, B and

C, can give better service through enabling an increase in the use of the network resources.

Paper A investigated the effect of mobility on the interaction between the MAC and routing layers. MANETs differ from other networks in that they may have to handle a high degree of mobility. Proactive routing protocols for MANETs maintain link state information by declaring their presence for neighbors using HELLO messages. However, packets may be lost due to other reasons than link breaks. Thus, the routing protocol cannot conclude that a link is broken if only one HELLO message is lost. Several must be lost before action is taken. Meanwhile, the lower layer protocols have to handle the link break on their own. The IEEE 802.11 MAC protocol attempts to retransmit packets that are not acknowledged. It does so to compensate for the potentially high BER that may cause some transmissions to fail. The retransmissions may create a bottleneck as numerous retransmissions keep the interface busy for a longer time than only one transmission. A small topology of three nodes was used to analyze the impact of the queue length on the rerouting time, because of the effect of the retransmission mechanism. The paper also suggested decrementing the number of maximum allowed retransmissions to a link destination each time a packet is discarded due to lacking link layer Acknowledgment (ACK). This was through simulations shown to have the desired effect of reducing the rerouting time.

From paper A, the work shifted towards larger topologies with gateways, and in paper B, the impact of gateway positions in the ad hoc network on capacity was studied. Both scenarios with single and with multiple gateways were investigated, and the impact of handover was analyzed. In this paper, the effect of retransmissions analyzed in paper A was documented through transmission plots showing a clearly defined ring tracing the transmission area edge of the receiving gateway.

In paper C, the focus was again shifted towards studying and improving the routing protocol. Most routing protocols use shortest path routing for forwarding, where the shortest path is the path with the minimum hop count, i.e., the least number of links. Therefore, routing via shortest path implies that longer links are preferred over shorter ones. In networks with mobility, this increases the likelihood for link breaks. Based on this knowledge, a solution was proposed where the routing protocol attempts to only use neighbors closer than a certain range as forwarding neighbors. By selecting to use only "safe" links, the goodput is increased. However, forcing the routing protocol to use "safe" links increases the path length in many circumstances. There is a tradeoff between the two choices, depending on the degree of mobility in the network.

1.4.2 Group communication

Group communication, supporting one-to-many communication in MANETs, is the other main area of the thesis work (Figure 1.1). While the works on unicast routing showed that the aggregated network throughput could be increased, this does not necessarily mean that applications are better supported. Due to other limitations than bandwidth, such as packet loss or delay, applications may be unsupported despite higher capacity. The Push-to-Talk (PTT) service is a good example of an application vulnerable to loss, but without high throughput demands. The papers D and E are more focused on service quality for PTT, measured using a goodput metric defined as a percentage received of all generated packets. The solutions in both of the papers D and E are able to improve the PTT traffic, while allowing other traffic in the network at the same time.

With paper D, the work changed from studying unicast routing to group communication. In tactical military, rescue and other emergency scenarios, voice communication is an essential coordination tool. The ability of all participants to share the same understanding of the messages that are being given, and to quickly react to occurring events, has made the "walkie-talkie" a prerequisite. This service can be referred to as Push-to-Talk (PTT), and should be supported in MANETs. High priority PTT traffic, forwarded using Simplified Multicast Forwarding (SMF), was studied, and three preemption mechanisms designed to protect the PTT traffic against background traffic influence was presented. These preemption mechanisms spanned from discarding the lower priority traffic, through buffering of this traffic, to the scheduling of low priority traffic in between the high priority packets.

Paper E addressed a mobility problem with the SMF-employed Source-based MultiPoint Relay (S-MPR) algorithm that was discovered during the work with SMF in paper D. The paper further explored ways of sustaining QoS for PTT traffic in a network with simultaneous Situational Awareness (SA) traffic. The S-MPR algorithm depends on the MultiPoint Relay (MPR) selection algorithm in Optimized Link State Routing (OLSR), where a node selects a subset of its neighbors as MPRs so it can reach all 2-hop neighbors through at least one MPR. Any node selected as MPR will then forward a multicast packet received from a node that has selected it as MPR, forming a Connected Dominating Set (CDS). The S-MPR algorithm is vulnerable for mobility, and on the other hand the Non-Source-based MultiPoint Relay (NS-MPR) algorithm is vulnerable for high density and high load. First a radio metric was shown to be able to combine the two algorithms, so that NS-MPR was used at low loads and S-MPR was used at high loads. This made it possible to optimize the forwarding of SA traffic, regardless of the size of the SA traffic packets. Thereafter, the simultaneous forwarding of both traffic

types was optimized through proposing a preemptive switch to S-MPR for the SA traffic when PTT traffic is transmitted in the network. The two solutions increased the utilization of the network, while maintaining the QoS for the PTT traffic.

1.5 Thesis Outline

The thesis is organized in two parts. Part I is an introduction to the areas where the thesis contributes, whereas Part II consists of a set of published articles that present the results of our research.

The list of figures and the list of terms and acronyms given in the beginning of the thesis are restricted to Part I. Likewise, since each article includes a reference list, the reference list found at the end of Part I is exclusive to this part of the thesis.

Part I begins with a brief introduction in Chapter 1, describing the background, motivation and outline of the thesis. The thesis addresses unicast routing and group communication as main areas (Figure 1.1), and therefore background on these areas are discussed, with unicast routing in Chapter 2 and group communication in Chapter 3. The research methodology used in this work is presented in Chapter 4. Works related to each of the contributions are discussed in Chapter 5, along with the main contributions and conclusions.

Part II consists of the following five research papers, in chronological order:

PAPER A:	Rerouting Time and Queueing in Proactive Ad Hoc Networks
PAPER B:	Gateways and Capacity in Ad Hoc Networks
PAPER C:	Routing with Transmission Buffer Zones in MANETs
PAPER D:	Preemption Mechanisms for Push-to-Talk in Ad Hoc Networks
PAPER E:	Optimized Group Communication in Tactical Military Networks

Chapter 2

Unicast routing in MANETs

2.1 Introduction

This thesis investigates ways to increase the performance of MANETs based on proactive routing. All the thesis papers are based on the behavior of a proactive routing protocol. The papers A–C employ it for unicast routing, while the papers D and E also use the OLSR MPR selection mechanism as basis for multicast packet forwarding. This chapter aims to give a background on unicast routing and therefore starts with an overview of the unicast routing area. Next, the two perhaps most dominant unicast routing protocols for MANETs, Dynamic MANET On-Demand Protocol (DYMO) and OLSR, are presented. The work in this thesis is based on the OLSR routing protocol, and DYMO can be seen as the reactive counterpart to OLSR.

After the background on the routing protocols, the challenges and possibilities that metrics constitute for the routing protocol are discussed. This topic is discussed in paper C, which shows how the shortest path routing metric can yield lower performance compared to combining hop count with other metrics. Link failure detection, the next topic in this background chapter, is of relevance especially to the papers A through C, but also to paper E, where the value of detecting link breaks as quickly as possible is shown.

Even though the routing layer and routing protocol are the main tools in this thesis to achieve increased performance, the work in paper A showed how the interaction between the routing layer and the link layer can impact the performance of the routing protocol. This behavior is seen again in the papers B and C, where the IEEE 802.11 MAC retransmissions triggered by link breaks impact the network

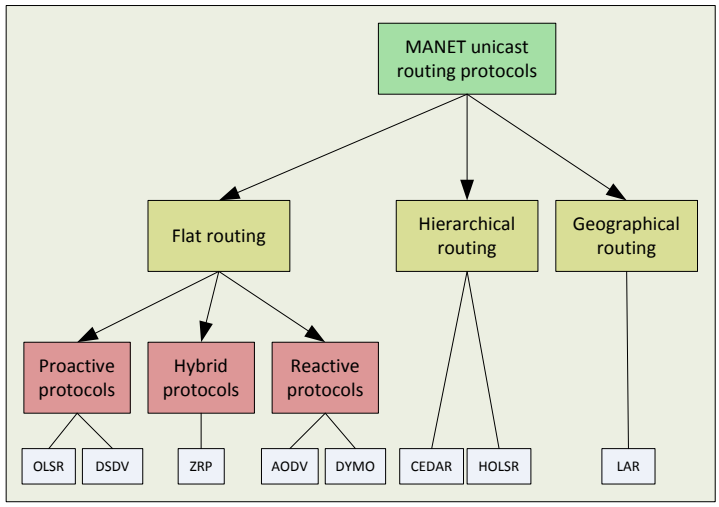


Figure 2.1: Classification of unicast routing protocols.

performance. Therefore, as the final topic in this chapter, the relevant parts of the IEEE 802.11 MAC layer are described.

2.2 Overview

The primary task of the routing protocol is to discover and maintain routes to needed network destinations. Many of the challenges in MANETs (from Section 1.2) must be dealt with by the routing protocol:

- Limited capacity, demanding low routing protocol overhead.
- Varying link quality with regards to bit error rate.
- Mobility, leading to link breaks and new links.
- Distributed routing, where it cannot be guaranteed that all network nodes have the same view of the topology.

A great number of routing protocols for MANETs have been proposed, dealing with these challenges in very different ways, and the classification of such a diverse group of protocols is equally varying. A very coarse classification of these protocols is in three sub-classes of flat routing: proactive, reactive or a hybrid of the two first types. Refining the classification, two additional main classes can be included: hierarchical and geographical routing. This classification is presented

in [6] (Figure 2.1). In routing protocols based on flat routing, all nodes that participate in routing play an equal role. The hierarchical protocol class is the complementary class of the flat routing, addressing scalability. In this class, nodes play different roles in the routing, where nodes are grouped and routing information is restricted on the basis of these groups. Protocols of the third class, geographical routing, require all nodes to know their exact position, for example using the Global Positioning System (GPS). Based on this knowledge, routing requests may for example be restricted to the geographical area where the shortest path to the destination is expected to exist.

Protocols of all the three classes employ either a proactive, reactive or a hybrid routing scheme. Proactive protocols seek to keep an updated view of the entire topology at all times. They are also referred to as "table-driven" protocols. Reactive protocols, on the other hand, only establish routes "on-demand", meaning that the routing protocol discovers the route only at the time when a packet must be routed to a destination. A hybrid protocol, employing the third routing scheme, uses both pre-discovered routes and is able to discover routes to destinations "on-demand". The Zone Routing Protocol (ZRP) [7], for example, limits the routing overhead by using proactive routing in its neighborhood, while routes to destinations farther away are only discovered when needed.

2.3 Protocols

The aim of any routing protocol, regardless of its classification, is to provide efficient routing with a minimum of overhead. The Internet Engineering Task Force (IETF) chartered a Work Group (WG) named MANET [8, 9] to focus on routing in MANETs. They are currently working on standardizing two MANET routing protocols, one for reactive (DYMO), and one for proactive routing (OLSR). These two protocols are able to operate in a very broad area considering various network parameters such as node density, mobility and link properties. Therefore, these two protocols are described in this chapter. The DYMO protocol, a very typical example of a reactive protocol, is presented briefly, while the OLSR protocol, which was used throughout the work of this thesis, is presented in more detail.

2.3.1 DYMO

The reactive routing protocol DYMO [10] is an improvement on the Ad hoc On-Demand Distance Vector Routing (AODV) [11] protocol. It has two basic operation processes: route discovery and route maintenance. Below, the two operations

are described briefly, and important properties of the routing protocol are emphasized.

Route discovery

In the route discovery process, the DYMO protocol is initiated by an application at node (A) which has created a packet to be sent to a destination node (B). The routing protocol checks if it has the destination in its routing table when the packet arrives at the routing layer. If the route is lacking, the node generates a Route Request (RREQ) which is flooded throughout the network[1]. Upon receiving the RREQ, a forwarding node can append its own Internet Protocol (IP) address to the packet, enabling routes to all upstream nodes from all downstream nodes of the RREQ packet. When the RREQ packet reaches (B), this node replies to (A) using a hop-by-hop unicast Route Reply (RREP). As the RREQ packet is forwarded through the network, routes are set up towards the upstream nodes including (A). Likewise, as the RREP is forwarded back to the RREQ originator (A), the routes to the destination (B) and the RREP upstream nodes are set up.

An intermediate DYMO router may also issue a RREP if it has routing information that can satisfy the incoming RREQ. If so, it also has to send a RREP to the RREQ target node, to ensure valid routes both ways between the source and destination.

Unidirectional links must be avoided with DYMO, and upon detecting such a link, the router may blacklist the link to ensure proper route discovery and packet forwarding.

The DYMO protocol is loop-free through its use of RREQ sequence numbers. Upon receiving a RREQ with a larger sequence number, older routing information (i.e., existing routes based on a lower sequence number) can be discarded, and stale routing information can be avoided. An advantage of a reactive protocol such as DYMO is that the route is fresh when traffic starts being forwarded on the path. However, it is a challenge that while there may exist a better route to the destination after a while, the traffic is not rerouted before the currently used route is broken.

Route maintenance

In DYMO, route maintenance is performed using two operations. First, the lifetimes for routes that are in use are extended upon successful forwarding. Second,

[1] An Expanding Ring Search (ERS) can limit the scope of the RREQ packet, if desired. With the ERS, the TTL field is first set to 1 and then increased if no route is discovered.

upon detecting a link break, a Route Error (RERR) is issued.

Link breaks are discovered either by using HELLO packets broadcasted by each node, or by notification from the link layer (LLN). Using HELLO packets, the failure to receive either a predefined number of these packets, or the timeout of the last received HELLO packet results in the conclusion that the link is broken. (Link failure detection in general is described in more detail in Section 2.4.2.)

Upon detecting a link break, the discovering node's route set is updated. Next, if a packet is attempted forwarded to a destination no longer available, a RERR is sent towards the packet source. The node that discovers the link break issues the RERR, including the unreachable destination and may also include other unreachable nodes connected through routes using the same broken link. Other nodes receiving the RERR evaluate it and remove the destinations for which they have still valid routes before forwarding the RERR towards the source. The RERR may be sent either as unicast or multicast, and is forwarded while there are destinations in the RERR packet that are unresolved, until it has reached the source or has flooded the network.

2.3.2 OLSR

The proactive protocol brought forward by the MANET WG is the Optimized Link State Routing (OLSR) [12] protocol. Currently a version 2 is being developed [13], implementing a more standardized packet format.

OLSR is a link-state protocol, where all nodes broadcast HELLO messages at regular intervals. The HELLO broadcast enables neighbors to detect the broadcasting node. Using the HELLO messages, the routing protocol is able to exchange information about its neighbors and gain information about 2-hop neighbors.

The protocol differs between packets and messages. An OLSR packet may contain more than one message, limited upwards by the Maximum Transmission Unit (MTU) of the network. The protocol defines one packet type, and several messages: HELLO, Topology Control (TC), Multiple Interface Declaration (MID) and Host and Network Association (HNA). The HELLO message is the only type not forwarded by other nodes.

The link-state information (TC messages) is distributed throughout the network using MPR-nodes [14]. All nodes select a subset of their neighbors as MPRs in such a way that (at least) all 2-hop-neighbors, i.e., nodes that are the neighbors' neighbors, are reachable by these MPRs. The way of selecting these MPRs creates a CDS that can be used to efficiently flood link-state information packets to all

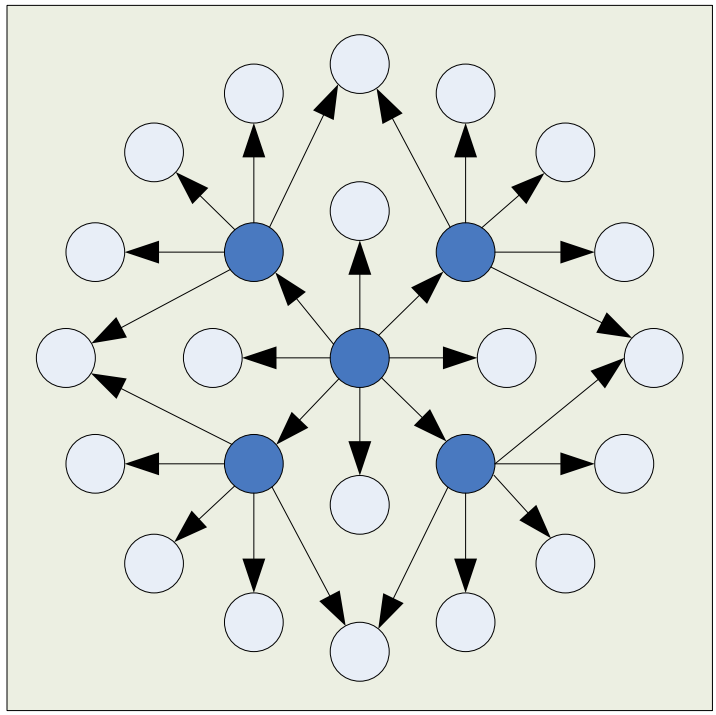


Figure 2.2: MPR-forwarding of a broadcasted packet.

nodes in the network. The MPR-based CDS can be used in various ways to forward packets meant for the entire network. This is described in more detail in Chapter 3 and in paper E. In OLSR, the packets are forwarded by an MPR node only if the node the packet was received from has selected the MPR node as MPR. This is referred to as Source-based MultiPoint Relay (S-MPR).

Adding to the efficiency, only the nodes selected as MPRs create the TC messages. The MPR forwarding is illustrated in Figure 2.2, where for example a single TC message can be forwarded in a node's 2-hop neighborhood by merely five transmissions[2].

All messages specify a validity time for a received message, i.e., the originator of the message specifies how long the information is valid. The default suggested timeouts are equal to the interval between three packets of the same type, but this can be changed as required/desired. However, the clocks at the nodes are not necessarily synchronized, and the validity time is from the moment the packet is received and processed, not from the generation of the packet. To avoid old

[2]One transmission by the source, and one by each of the four MPRs selected by the source.

information replacing newer information, all messages contain a sequence number, where information from packets with larger sequence number replaces that of packets with a lower sequence number.

2.4 Relevant routing-specific mechanisms

2.4.1 Metrics

The wireless medium has been difficult to abstract in simulators, and the most common communication model has been the Unit Disk Graph (UDG). The UDG implies that two nodes can communicate *only* if they are within a defined transmission range. This distance is equal for all nodes. In reality, received signal strengths vary, and this results in a varying transmission range.

The common design approach for MANET routing protocols has been to apply the same routing paradigms conceived for traditional wired networks. This design choice implicitly assumes that wireless links are similar to wired links, and that they can be represented as point-to-point connections.

However, mobility and link fluctuations create topology changes that the routing protocol needs to detect and handle, to maintain operational routes needed by the applications. If the routing protocol is too slow in acting on topology changes, the result is inevitably problems for the applications, due to for instance packet loss, delay or jitter.

Likewise, the routing protocols have commonly used shortest path routing, preferring longer links if this makes the path shorter. Using the UDG model, longer links are just as reliable as short ones, but in reality the reliability of a link is reduced with distance, due to the reduced signal strength. In other words, the links most likely to fluctuate are the longest links preferred by routing protocols.

The UDG has worked to the advantage of protocols that are quick to act upon link changes, since most link breaks have been due to mobility. The methods to detect link breaks, however, can be fooled by fluctuating links which resemble some of the behavior of links broken due to mobility, but these cannot be handled the same way, i.e., through considering them lost at the very moment the symptoms start occurring.

The aim of any routing protocol is to support connectivity between nodes in a network. In MANETs, this is a challenging task, both due to mobility and due to the limited network resources available. Usually the routing protocol should

provide connectivity using the best route between the source and the destination, based on the aggregation of link cost throughout the path. As such, the best path is the shortest path, and Edsger Dijkstra's shortest path algorithm is widely used by link state routing protocols for MANETs.

Most routing protocols for MANETs use hop count as a simple link cost metric. The Minimum Hop Count (MHC) route metric is very efficient, as it minimizes the number of theoretical transmissions per packet in a flow, thus reducing the impact each packet has on the network capacity. In addition, in most cases the delay and jitter is reduced. In simulations with a free space propagation model or equivalent, without varying link conditions, the MHC metric is one of the best suited metrics.

However, the MHC metric can also be detrimental to the network performance, if used without other qualifiers. Physical links degrade with distance, and the MHC metric makes sure the number of hops are the lowest, meaning that each link on the path is stretched. In many cases the link may be stretched so that the quality is reduced so much that it would be better to add another forwarding link to the path to shorten the average link length of the path. The longer links also have a greater probability of breaking, compared to shorter links. This was emphasized in paper C, where the signal strength metric was employed to select "safer" routes.

Using heterogeneous links can also be a problem with the MHC, as the routing protocol prefers to route packets over the longest links, reducing the number of hops per path. However, the longest links are often also the links with the lowest capacity. Thus, the network utilization is not optimized.

Privileged relays can also be problematic using MHC. Consider for example an Unmanned Aerial Vehicle (UAV) used as a relay node. The link may provide limited capacity, but it is reachable from most ground nodes. Thus, the path length is in most cases the lowest if the UAV is used as a relay. However, the UAV may have been deployed to offer connectivity to parts of the network that would otherwise be without connectivity to the rest of the network, or to offer redundancy for high priority traffic. To rely only on MHC in such a case would make it difficult to reserve the use of the UAV as a relay node only for those intended.

While the MHC metric reduces the network capacity usage to a high degree, there is no change in case some parts of the network get congested. No rerouting occurs, and the traffic from all nodes continues to use the shortest path (in hop count) even if all packets are dropped along the way due to interface queue drop or collisions. However, this is not entirely true. As the network gets congested, there is a higher risk that the routing control packets are not able to either pass through or emerge from this area. Thus, when nodes no longer have routes to or through this area, the traffic must be routed around or be discarded before entering the area. I.e., the

MHC indirectly supports some load balancing, but not due to qualified choices; instead random routing packet loss steers the traffic.

Security is a different area where the MHC is not able to provide qualified paths. One could envision that some network nodes are insecure, but if needed to provide connectivity may be used. Thus, if connectivity is provided around such nodes, the path should use such a route. Using MHC, there is no control with how and through which nodes the traffic is routed.

Another issue with MHC is energy usage. While the minimum hop path may be optimal for network capacity, there is a great risk that some nodes that are more or less alone providing connectivity between two parts of the network may run out of energy much faster than the other nodes. The MHC is not able to distinguish between nodes with high energy levels and nodes without much energy left. Thus, there is a risk that the network may become partitioned a lot sooner with the MHC metric than with a metric that considers the energy levels of the nodes along different routes in the network.

Alternative metrics have been proposed, incorporating other topology properties. A very thorough classification and description of routing metrics can be found in [15]. The Expected Transmission Count (ETX) [16] path metric represents the expected number of transmissions (including retransmissions) needed to forward a packet to the destination over a given path. It is calculated by observing the number of successful and unsuccessful broadcast transmissions over each link during a time interval. The ETX information enables the routing protocol to route packets over paths that are better suited than the shortest path. In [17], ETX is discussed and compared to two other metrics, the Expected Transmission Time (ETT) and the Weighted Cumulative ETT (WCETT), which are a bandwidth-adjusted ETX and a path cumulative ETT metric, respectively. Another iteration over the ETX metric is the Interference aware routing metric (iAWARE) [18] which is ETT weighted with an Interference Ratio (IR) to capture both the link loss ratio and packet transmission rate through ETT and factor in the varying interference affecting the link using IR.

The Signal Stability-Based Adaptive Routing Protocol (SSA) [19] uses a different approach to measuring link quality, using signal strength information to weigh links. The solution is based on both the maximum signal strength and the stability of the signal strength over time. Since the signal strength information comes from a lower layer, it is considered a cross-layer solution. Other cross-layer information metrics include the average number of link-layer retransmissions per packet, Link Layer Notification (LLN), the number of errors corrected using Forward Error Correction (FEC), etc.

The OLSR RFC also includes a hysteresis mechanism to avoid using transient links. The mechanism works through both avoiding to set up a link on the basis of one or two HELLO messages only, and through more quickly tearing down a link where several consecutive HELLO messages are lost (i.e., sooner than the link would have timed out without hysteresis). This is similar to ETX in that it is based on the reception of HELLO messages, and is not a cross-layer solution. This could be comparable to continuing to route traffic using shortest path and hop count, but filtering out some links based on the quality of the link.

An IETF draft aimed at supporting various metrics for the new OLSRv2 specification is currently available [20]. Here, a motivation is given for considering other metrics beside the MHC. An overview of potential metrics for the OLSRv2 routing protocol is presented, and most importantly it presents a proposal for adapting the OLSR protocol to use other metrics besides MHC.

2.4.2 Link failure detection

Any routing protocol's challenge is to keep an updated view of the network topology, in order to support routing to destinations or gateways in the network. This can be done on-demand or proactively. The network topology consists of the nodes in the network and the links between these nodes. Regardless of whether the links that make up the path were discovered on-demand or proactively, any path that is actively used must be maintained, and upon a link break the path must be changed, so packets can continue to reach the destination.

The IETF MANET WG is bringing forth a standard for a MANET NeighborHood Discovery Protocol (NHDP) [21], aimed to be used by the routing protocols also brought forward by the MANET WG (DYMO, OLSR and SMF). The NHDP uses a local exchange of HELLO messages in order for each router to determine the presence of – and the connectivity to – its 1-hop and symmetric 2-hop neighbors. The information obtained through the HELLO packet exchanges is recorded in the form of Information Bases accessible by other protocols, including MANET routing protocols. The NHDP is based on the neighborhood discovery process of the OLSR protocol.

Link breaks in MANETs occur due to node mobility and radio channel characteristics, and the routing protocol must be able to detect such changes while not being affected by fast fading and interference from other transmitting nodes. To detect link breaks, the routing protocol has three measures at its disposal:

- *Route timeout*

- *Neighborhood discovery (polling packets)*
- *Link Layer Notification (LLN)*

Route timeout can be used to monitor the existence of the link, and can be used by reactive routing protocols. With this method, routes where packets no longer flow are timed out, meaning that either the route is no longer in use or an upstream link is broken. Both the AODV and DYMO protocols support this method.

Polling packets is an active way of monitoring the link. The routing protocol generates HELLO packets which are periodically transmitted to all neighbors. A link break is detected when no HELLO packet has been received from a particular neighbor for a predefined timeout time. Proactive protocols, such as OLSR support this method, but it is also used by on-demand protocols like AODV and DYMO in cases where LLN is not available.

LLN is a cross-layer mechanism where the link layer notifies the routing protocol in the event that a link is detected as broken by the link layer. Upon receiving this notification, the routing protocol may set the link as lost, and recalculate the routing table or start a route repair or new route discovery. The support of this mechanism depends on the link layer protocol, as well as the implementation of the routing protocol. The background for the LLN mechanism in 802.11 is described in Section 2.5.

2.5 Link-layer mechanisms – the IEEE 802.11 MAC

The IEEE 802.11 MAC is the most widespread wireless Local Area Network (LAN) technology today. It was defined as a standard in 1999 [22], and after some years with amendments to this standard, a new standard incorporating many of the amendments (a, b, d, e, g, h, i, j) was released in 2007 [23]. The basic DCF function of the 802.11 MAC, as described below, remain unchanged from the 1999 standard.

The 802.11 standard defines a Physical (PHY) layer and a MAC sub-layer, where the latter supports two modes of operation, the Distributed Coordination Function (DCF) and the Point Coordination Function (PCF). The DCF is the normal mode of operation in MANETs. With DCF, the medium is accessed in a distributed way by the Wireless Stationss (STAs), using Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA). With the basic access mechanism, each unicast DATA frame is acknowledged with an ACK frame. This is also known as the minimal frame exchange. (Multicast transmissions, however, are not followed by

an ACK frame.) With the optional 4-way frame exchange, on the contrary, each DATA frame is preceded by an exchange of a Request To Send (RTS) and a Clear To Send (CTS) frame.

When a node has some data to transmit, it has to sense the medium to verify whether it is busy or idle. If the channel is idle for a time interval equal to a Distributed Inter-Frame Space (DIFS), the source node may begin data transmission. When the receiver has received the DATA frame, it waits for a time interval equal to a Short Inter-Frame Space (SIFS), and transmits an ACK back to the source node. While data is being transmitted, all other nodes must defer their channel access for a time interval equal to the Network Allocation Vector (NAV). This is a timer indicating the amount of time that the medium has been reserved for the current transmission. When the data transmission is finished and the NAV has expired, a new contention period is entered. Here, concurrent nodes with pending data traffic must contend for the medium. In this process, each contending node must choose a random time interval called a backoff timer, selected from the Contention Window (CW). The CW ranges from 0 to a maximum value which is increased upon non-successful transmissions.

The backoff timer is decremented only after each time the medium is idle for a DIFS interval, and is frozen when the medium becomes busy. When the backoff timer of a node eventually reaches 0, the node will transmit data, if it has any to transmit. The main point of this medium access mechanism is to minimize the probability of a collision, i.e., concurrent transmissions. Since a node must go through a backoff after having transmitted a frame (also referred to as a post-backoff), the medium access mechanism also provides long term fairness to access the medium. In a wireless environment where collision detection is hard or even impossible, a positive ACK from the receiver is used to confirm a successful transmission. The absence of such an ACK message indicates a collision, link failure or other reasons for an unsuccessful transmission. When this occurs, a retransmission is scheduled, and a new backoff value is chosen. However, in order to reduce the risk for consecutive collisions, the CW is doubled after each unsuccessful transmission attempt, until a predefined CW_{max} is reached. There is a retry counter associated with the transmission of each frame, and the retry counter is incremented after each collision. After a successful retransmission, the CW is again reset to a predefined CW_{min}, and the retry counter is reset to 0.

If the number of unsuccessful retransmissions reaches the maximum allowed number, the packet is discarded, and a new packet is fetched from the interface queue. In this event, the 802.11 MAC can notify the routing protocol of the broken link. This notification is referred to as LLN, and is one of three methods of topology change detection described in Section 2.4.2. It is directly based on the positive

ACK mechanism of the IEEE 802.11 protocol. As mentioned in Section 2.4.1, various MAC or PHY information, such as metrics describing the link quality or the average number of transmissions per packet, can be sent to upper layers, constituting cross-layer information.

Chapter 3

Group communication in MANETs

3.1 Introduction

MANETs are portrayed as highly suitable for use in emergency and crisis scenarios. In such scenarios, effective support of group communication is essential for most ad hoc network applications. Group communication in terms of traffic flow constitutes information flowing between one or multiple senders, and multiple receivers.

The papers D and E focus on enhancing the service for group communication based traffic in MANETs. This chapter therefore aims to give an introduction to the group communication area of MANETs. First, an overview of the area is given, and thereafter a taxonomy including a description of several protocols is given. The MANET group communication protocols are very diverse, spanning from unicast-based to broadcast-based flooding. Some protocols depend on underlying unicast routing protocols, while others are completely independent. The thesis papers D and E, which focus on group communication, use the SMF protocol on top of OLSR for forwarding multicast packets, but considering the multitude of protocols with highly varying operational scope it was chosen to present examples of all the various classes of group communication protocols.

Finally, QoS in group communication is briefly introduced in Chapter 3.4 since both of the papers D and E propose mechanisms to control the traffic flow in the network to achieve better QoS for prioritized traffic.

3.2 Overview

Group communication can be supported using unicast traffic flows and corresponding routing (presented in Chapter 2), but unicast communication requires each receiver to have its own separate packet flow, meaning that the source (and all relay nodes) must transmit each packet once to each of the receivers. With an increased number of receivers, the higher number of transmissions will use much more network capacity than necessary. Therefore, much effort has been done to develop other protocols specialized for group communication.

The fact that identical information is to be distributed to multiple receivers has encouraged new communication types besides unicast, namely multicast and broadcast. Wireless networking is a broadcast medium, i.e., all neighbors are physically able to receive a node's transmission, and this can be an advantage for group communication. Thus, only one transmission is necessary to reach all neighbors. If all nodes in the network forward the broadcast packet *once* upon receiving it, the packet should be received by all nodes in the network, without requiring any routing overhead. However, one or more nodes may not receive the packet, due to collisions or congestion. The distribution of a packet to all network nodes by way of broadcast is referred to as flooding in this thesis. Flooding can be used even if only a subset of the network nodes are interested receivers. However, in a scenario with few interested receivers, it is not efficient to flood the entire network.

While flooding has the advantage that it requires no control traffic, the number of transmissions generated by one packet is directly correlated to the number of nodes in the network. This is also referred to as the "Broadcast Storm Problem" [24]. As such, the unicast routing protocols AODV and DYMO, which both use flooding of Route Request (RREQ) packets to discover routes, implement the Expanding Ring Search (ERS). Using ERS, the network is only flooded in a limited radius around the sender at first. However, unless it is known that all receivers are within a certain radius, such a limit cannot be enforced.

Due to the high number of transmissions generated throughout the network using flooding, multicast can be a better alternative. The concept of multicast implies that one or multiple sources generate packets for a subset of the network nodes. The packets may be forwarded by a limited number of nodes, coordinated using a multicast routing protocol, to reach all interested receivers. However, the cost of more efficient forwarding includes:

- Control traffic overhead
- Loss due to topology changes (logical partitioning)

- Less resilience to collisions
- Subscription delay

Even if flooding or multicast may be the preferred way to forward group communication traffic, such traffic faces some network challenges that differ from that of unicast:

- **Duplicate packet interference:** In one-to-one communication, the packet flow can interfere with itself, but any one packet will not interfere with itself, as it is only transmitted by one node at a time. In multicast or broadcast forwarding, on the other hand, a packet can be forwarded simultaneously by two nodes. This is not so much a problem when the nodes are neighbors (inside each other's transmission range), since random access and channel sensing reduces the probability of simultaneous transmissions. However, two nodes outside each other's transmission range will possibly interfere with a third node lying between the two, thus experiencing a collision. The combination of this interference problem with the lack of link layer acknowledgment makes multicast and broadcast traffic more exposed to loss than one-to-one traffic.

- **Network Capacity:** Multiple receivers mean that packets may have to be forwarded covering a larger area of the network than would have been the case with a one-to-one transmission. This could require several more transmissions, thus requiring more network capacity per packet.

- **Dynamic memberships:** Depending on the running applications and the type of distribution, the nodes may join and leave groups frequently, requiring multicast protocol support.

- **No MAC layer link break support:** With multicast, link breaks cannot be detected directly through the use of Link Layer Notification (LLN). LLN is based on a lack of transmission Acknowledgment (ACK), and such ACKs are only provided with unicast communication. Most random access MAC technologies, such as the IEEE 802.11 do not provide ACK for multicast and broadcast transmissions, since this would mean multiple receivers have to respond in order. Exceptions do exist, such as [25], but such adaptations increase the complexity of the MAC protocol and are not in widespread use. The consequence is that in multicast and broadcast-only networks, the rerouting time, i.e., the duration of a potential logical partitioning, can be much longer than in networks with unicast traffic.

3.3 Group communication protocols

Two clear alternative protocol types have emerged within the field of unicast routing – the reactive and the proactive – and this is reflected in the work of the IETF MANET WG, where the DYMO and the OLSR protocols are on the track to be standardized. The same convergence has not taken place within the group communication area, and therefore the various group communication protocol types are described in greater detail, compared to the chapter on unicast routing.

Network protocols for group communication span from unicast-based protocols, through standard multicast protocols (i.e., protocols that utilize link layer multicast support) to flooding protocols. The main class of group communication protocols, multicast, has been a focused area for MANET research, and many multicast protocols have been proposed [26,27]. While the classifications differ greatly [27–29] depending on the perspective and granularity, the multicast protocols can generally be divided in four classes: tree-based, mesh-based, hybrid and stateless. In addition, we consider flooding protocols to be a fifth class of multicast protocols in this thesis. Although there is no subscription process necessary with flooding protocols, the behavior with regards to network efficiency – especially in networks where most nodes are interested receivers – can be similar to that of multicast.

The group communication protocol classes may be sorted based on the ratio of interested receivers from a few to all:

- Unicast
- Stateless multicast
- Tree-based multicast
- Mesh-based multicast
- Hybrid multicast
- Flooding

Stateless multicast is a form of multicast where the source alone keeps track of which nodes that are subscribers to the information, i.e., the network holds no multicast state information. It is an adaption of unicast to the multicast paradigm, where each packet is transmitted using MAC-layer unicast, but the packet contains all receivers as destinations in the header. The source inserts the list of destinations in the packet's header and sends the packet to the next router. Each router determines the following jumps of each packet, and copies the packet as many different paths as it has to follow. Before the packet is transmitted to each of the

next hops, the destinations that are not on path via the next hop are removed from those copies of the packet. The multicast packet is propagated in the network from node to node using unicast. However, this means that the protocols must rely on a unicast routing protocol to route the packets between the receivers. This type of multicast protocol is suited for small multicast groups. Examples of stateless multicast routing protocols are the Differential Destination Multicast (DDM) [30] and the Extended Explicit Multicast (E2M) [31].

Stateless multicast is suitable as long as the number of receivers is small, but with a higher number of receivers, the destination overhead in each packet may be excessive. In such scenarios, tree-based multicast protocols may be more suitable. The tree-based protocols are best suited for static networks, mainly because the trees offer no redundant distribution paths, and in the event that a link is broken, the packet flow to the downstream nodes is interrupted. The tree-based protocols are further classified into source-based-tree and shared tree protocols. The source-based tree protocols create and maintain separate trees spanning from each multicast source, while shared tree protocols maintain one multicast tree for all sources. Two tree-based routing protocols for MANETs are presented below, the Multicast Ad-hoc On-demand Distance Vector (MAODV) [32] and the Multicast Optimized Link State Routing (MOLSR) [33]. Both protocols use a shared tree.

Considering the problems with tree-based multicast protocols, mesh-based protocols have been proposed as an alternative. Mesh-based protocols provide redundant links in the forwarding paths, increasing the probability that packets reach all destinations, even if a link break is occurring. The cost of redundancy is the increased overhead, due to both signaling the increased number of links in the forwarding paths, and also the redundant data transmissions. Mesh-based protocols include the Core-Assisted Mesh Protocol (CAMP) [34] and the On-Demand Multicast Routing Protocol (ODMRP) [35].

Whereas the tree and mesh based protocols have different ways to solve multicast forwarding using either single paths or multiple paths, the hybrid protocols seek to pair the structure and efficiency of the tree-based protocols with the robustness of the mesh protocols. Hybrid protocols include the Ad hoc Multicast Routing protocol (AMRoute) [36], Multicast Core-Extraction Distributed Ad hoc Routing (MCEDAR) [37] and Multicast for Ad Hoc Networks with Swarm Intelligence (MANSI) [38].

As the number of receivers is increased further, an efficiency limit may be reached, where it is actually more efficient to forward the packet to the entire network in a flooding operation, rather than spend much network capacity to support multicast forwarding paths, due to the overhead. While multicast protocols have been

brought forward as well suited for MANETs, research has also been focused on distributing information for the entire network in a more efficient way than Classic Flooding (CF), where all nodes retransmit a received packet once. Such efficient flooding protocols are investigated in [24, 39, 40].

In the following sections, some of the group communication protocols mentioned above are briefly presented, to give an insight into the taxonomy.

3.3.1 Stateless multicast routing protocols

DDM is motivated by the Dynamic Source Routing (DSR) for unicast routing [41], where the complete path of a packet from source to destination is specified in the packet header. Each interested receiver has to send an explicit JOIN message to the source of the multicast traffic in order to join the group. The receiver is then required to reply to packets where a POLL flag is periodically set. If it fails to do this, it is purged from the source's member set.

The key notion of DDM is forwarding computation based on destination encoded headers, and the DDM can function in two different modes, soft-state and stateless. In stateless mode, all receivers are explicitly identified in the packet header, while in the soft-state mode routers store which multicast destinations that are served by this router. Thus, in soft-state mode, packets only contain changes in the receiver set, and depend on the routers to keep track of the destinations served by the router.

E2M is aimed at overcoming the limitations of the DDM and other stateless protocols using various dynamic mechanisms. An Explicit Multicast Forwarder (XF) concept is employed, with dynamical selection of XFs. Such XF routers are hierarchical forwarders aimed at limiting the number of multicast destinations needed in the header of each forwarded multicast packet from the source. Instead, the XF keeps track of which downstream nodes it is serving.

3.3.2 Tree-based multicast routing protocols

MAODV is an extension of the unicast protocol AODV [11], and the normal operation of the AODV protocol applies to MAODV. It uses the same route discovery mechanism as AODV to discover a path to the multicast distribution tree. A shared tree is maintained for each multicast group, and the first member of a multicast group becomes the leader of that group. This leader maintains the multicast group sequence number and periodically sends group HELLO messages to maintain the group forwarding tree.

Apart from discovering and including a new node into the group, the multicast tree is maintained in two more ways: pruning the tree when a node leaves the group, and repairing a broken link. A node can actively remove itself from the group by notifying its upstream neighbor, using a special Multicast Activation (MACT) message. However, it cannot do this if it is part of the path to another member node. If so, it has to wait for the downstream node to send a MACT disconnect message, and then it can send the same notification to its upstream neighbor. A broken link is detected through the timeout of packets from the neighbor. If the neighbor has no data packets to transmit, it is obligated to transmit HELLO messages. If a link times out, the disconnected downstream node (from the leader) must initiate local repair. The local repair implies a limited flooding of the RREQ message.

MOLSR is an extension of the unicast OLSR [12] protocol. It establishes and maintains one multicast tree per source per multicast group. Any node ready to participate as forwarding nodes for multicast, broadcasts this to the entire network using the MPR forwarding of OLSR. A source that has traffic to send, likewise broadcasts a notification message to the entire network, allowing members to detect the source and attach to its multicast tree.

The tree is maintained through periodical refreshing through special multicast messages, and through the link break detection and topology changes made available by the unicast OLSR protocol. A node can detach itself from the multicast tree by sending a LEAVE message to the upstream node (parent), but if it is a parent itself, it has to wait for its downstream node to detach before it can do so itself.

3.3.3 Mesh-based multicast routing protocols

CAMP is designed to support multicast routing in highly dynamic ad-hoc networks. It builds and maintains a multicast mesh between all sources and receivers in a multicast group. A shared multicast mesh is defined for each multicast group, and it is ensured that the shortest path between receivers and sources is part of the mesh. CAMP extends the receiver-initiated approach of the Core Based Tree (CBT) [42] to create multicast meshes for MANETs. The CAMP protocol depends on a unicast protocol to provide topology information. One or several cores are defined per multicast group to assist in the join operations, and the addresses of these are distributed in group membership reports made available through Internet Group Management Protocol (IGMP) [43] or an equivalent protocol. These cores avoid the requirement that control or data packets must be flooded throughout the entire network in order to set up the routing structure. The packets from any source in a multicast group are forwarded along the shortest paths defined with the mesh

from the source to the receivers.

ODMRP is another mesh-protocol, developed at about the same time as CAMP. It uses a forwarding group concept where only a subset of nodes forwards the multicast packets via scoped flooding. It dynamically builds routes for each multicast group and maintains multicast group membership on-demand. No control packets are triggered by link breaks, and this prevents excessive signaling overhead in the event of high mobility and multiple link breaks in a short period of time. The membership maintenance is done using a soft state approach, meaning that nodes do not have to signal the intention to leave the multicast group. The Achilles heel of the ODMRP protocol is sender scalability, as it floods the network with control traffic for each multicast sender.

The paper [35] compares ODMRP with the CAMP protocol. The authors conclude that the ODMRP produces less control overhead and efficiently utilizes the control packets to deliver more data packets to multicast members. Since the primary concerns of ad hoc networks are frequent topology changes and constrained bandwidth, it is critical that the protocol supplies multiple routes and yields minimal overhead.

3.3.4 Hybrid (combined tree and mesh) multicast routing protocols

AMRoute combines user-multicast trees and dynamic logical cores for robustness and efficiency. The user-multicast trees only consist of the group senders and receivers, while some tree nodes are designated as logical cores, and these initiate and manage the signaling in AMRoute. The core node role can change between nodes, depending on group membership and network connectivity. AMRoute depends on an underlying unicast routing protocol, and creates a mesh of multicast-enabled nodes that are close together. On this mesh, user-multicast trees are maintained. Thus, the protocol does not need to track topology changes itself. Using routes provided by the unicast protocol, AMRoute creates a bidirectional, shared tree for each multicast group using only group senders and receivers as tree nodes. Unicast tunnels are used as tree links to connect neighbors on the user-multicast tree, and even in case of a dynamic network topology, the tree structure does not need to change.

MCEDAR is an extension to the Core Extraction Distributed Ad Hoc Routing (CEDAR) routing architecture [44]. It is aimed at achieving the robustness of mesh based routing protocols and at the same time the efficiency of tree-based forwarding protocols. A CDS in the network is established through each node performing neighborhood discovery and selecting the node with the most neighbors

as the dominating node, or core node. It could even select itself. A node that is selected as a core, broadcasts a packet notifying its 3-hop neighborhood of its presence, establishing virtual links to other core nodes in the area. Interestingly, core broadcast is performed using unicast, since there are relatively few core nodes in the transmission range of any one other core node. The MAC protocol is adapted to piggyback core message tag information in RTS/CTS packets. Through overhearing other core nodes transmitting RTS and CTS messages, a core node can evaluate whether the message is heard by any other core nodes in its neighborhood, and will not forward the message to this node, if the message is received through some other neighbor. This way to do core broadcasting approximates a source-based tree.

A non-core node must request to a core node to become member of a multicast group. The core node then performs a JOIN operation for the multicast group to the other core nodes as a core broadcast, to establish membership in the multicast group. An R factor is associated with each mesh multicast forwarding structure (*mgraph*), representing the robustness of this structure.

MANSI is inspired by swarm intelligence, which is applied to the multicast routing problem in mobile ad hoc networks. It is an on-demand protocol where multicast connectivity is established among members through a core node. The first multicast connection is quickly setup using flooding from a core node throughout the network. The first source takes on this core node role, but it is not a static or permanent role, avoiding a single point of failure. Each non-core member of the multicast group deploys a small packet periodically, which behaves likened to an ant. This packet explores different paths to the core. This exploration mechanism enables the protocol to discover new forwarding nodes that yield lower total forwarding costs.

3.3.5 Efficient flooding protocols

Flooding protocols can be classified as belonging to one of four categories [39]: Simple flooding, probability-based, area-based and neighbor-knowledge-based. Simple flooding is nothing more than the classic flooding, or broadcast, algorithm, where all nodes receiving a broadcasted packet rebroadcasts this packet exactly once. This algorithm is specified and studied in [40]. In [24], the problem of the simple flooding algorithm in MANETs is brought forward, encouraging more efficient broadcast algorithms.

Also in [24], two probability-based methods for more efficient flooding are proposed: the probabilistic and the counter-based schemes. The probabilistic scheme

is founded on the premise that several nodes share approximately the same transmission coverage. Thus, with a predetermined probability, some nodes forward the data, while others remain silent. In dense networks this saves radio load, while maintaining the delivery ratio, while it does not work well in sparse networks. The counter-based scheme is based on the inverse relation between the number of times a packet is received at a node and the probability that this node reaches an additional area upon rebroadcast. A random delay is selected, upon which the number of overheard transmissions of the same packet is counted, and if this number is below a threshold, the packet is rebroadcast once.

The third category, area-based methods, is also covered in [24], where two schemes are presented: distance-based and location-based. The distance-based scheme evaluates the distance a packet has traveled the last hop, i.e., the distance between a node and its upstream neighbor. A timeout is scheduled and redundant packets are cached. After this timeout, it is checked if the packet is received from a neighbor closer than a predefined threshold. If so, the packet is not rebroadcast. The location-based scheme is more refined, using a more precise estimation of the additional coverage area which would be covered in the event of a rebroadcast. The nodes communicate their position in the headers of the broadcasted packets, and are dependent on precise position knowledge, for example using GPS. The decision to forward or not is delayed using a timeout as in the distance-based scheme, and if any other nodes so close that the additional coverage area is lower than a certain threshold has broadcast the packet, the packet is not rebroadcasted.

The fourth and final category, the neighbor-knowledge-based methods include Flooding with SelfPruning [45] and the Simplified Multicast Forwarding (SMF) [46–48]. Flooding with SelfPruning is a simple protocol where the neighbors are aware of each other's neighbors. A node receiving a broadcast packet checks if it has any other neighbors apart from the sender node. If not, then no rebroadcast is performed, as it will not reach any new nodes.

Simplified Multicast Forwarding

The SMF protocol provides a set of mechanisms to support efficient flooding in a MANET. It is currently on track to become an experimental IETF RFC.

SMF consists of three modules: Duplicate Packet Detection (DPD), NeighborHood Discovery Protocol (NHDP) and a forwarding algorithm. The DPD is proposed performed as either of two mechanisms: header content identification or hash-based duplicate detection. The motive is to prevent a node from rebroadcasting a packet already broadcasted on the interface by the node. Neighborhood

discovery may be performed by the NHDP [21], by a unicast routing protocol such as OLSR or through other means. The goal is that each node has updated knowledge of its 2-hop neighborhood and is able to relate relay set selection information in this 2-hop neighborhood. SMF can employ one of multiple forwarding algorithms, and some were examined and investigated in [47].

Two of these forwarding algorithms are the S-MPR and NS-MPR. The S-MPR algorithm works as follows: Any node *a* only forwards a multicast packet if it receives the packet from a node *b* that has selected *a* as an MPR. NS-MPR is based on S-MPR, but contrary to S-MPR any node *a* selected as MPR forwards packets from *b* even if *b* did not select node *a* as MPR (i.e., non-source-specific).

3.4 Enabling QoS in group communication

Many group communication applications for MANETs become unusable if the offered service quality is too low. Voice communication (PTT) is impossible with high packet loss or long packet delays. Video communication shares the same limitations, but depends also on large bandwidth. On the other hand, periodic traffic, such as position sharing and situation updates, is more resilient to loss and delay. Messaging/mail services or file transfers have much leaner demands on delay, but depend on a transport protocol to ensure the delivery of all data.

The service requirements of applications must be fulfilled by the network, even when the available resources are less than the resources required. To share the limited network resources optimally – and to ensure important applications maintain their service longer than less important applications – there has to be mechanisms in place to control the use of these resources:

- **Admission control** - the ability to control whether a traffic flow is allowed into the network or not.
- **Scheduling** - the ability to do load balancing between traffic flows' medium access.
- **Active Queue Management** - the ability to manipulate traffic flows through dropping or marking packets based on queue usage.
- **Preemption** - the ability to stop ongoing flows, i.e., flows that were admitted, but which can no longer be allowed in the network.
- **Priority queuing** - queuing higher priority packets ahead of packets with less priority.

- **Resource monitoring** - the ability to monitor which resources that are available.
- **Flow impact on resources** - determining the impact a new flow has on the available resources if admitted.
- **Resource reservation and cancellation** - the ability to reserve and release resources for a specific traffic flow.

Group communication, whether multicast or broadcast, poses additional challenges for a QoS-supporting routing protocol. A packet flow from a single source can vary, both depending on the number of current receivers/subscribers, and depending on the efficiency of the protocol in selecting a restrictive or redundant set of forwarding nodes. Below is a brief presentation of various solutions to support QoS in group communication enabled MANETs.

In [49], an Ad hoc QoS Multicasting (AQM) protocol supporting admission control is proposed, where a receiver is only able to join the multicast session if all nodes along the path to a current member, including the source, can sustain the set QoS properties.

The QoS-Aware Mesh Construction to Enhance Multicast Routing in Mobile Ad Hoc Networks (QAMNet) [50] approach extends a reduced data overhead version of the ODMRP protocol to include traffic prioritization, distributed resource probing and admission control, and adaptive rate control based on MAC layer backoff time feedback.

Badis and Agha propose a new multicast protocol with QoS support in [51], called QoS-aware Multicast Overlay Spanning Tree (QMOST). It is a multicast protocol supporting multiple-metric routing criteria using unicast packet encapsulation to forward packets. It requires a unicast QoS link state protocol, such as Quality of Service for Ad hoc Optimized Link State Routing Protocol (QOLSR) [52].

Chapter 4

Research methodology

In this chapter, an overview of the research methodology employed during the work with the thesis is presented. A more thorough presentation of the research methodology can be found in each of the papers in Part II.

There exist three main methods for investigating MANETs: theoretical analysis, simulations and experiments. Theoretical analysis can give fundamental knowledge about investigated mechanisms and systems. Simulations, on the other hand, enable the investigation of the dynamics occurring when the distributed interaction is too complex to model using theoretical analysis, especially in combination with mobility. Even though simulations provide an easy way to investigate the distributed properties of algorithms, simulators are not very good at abstracting the physical world. In cases where physical layer attributes (i.e., real world properties) are defining for the system performance or other investigated features, experiments with real equipment in the real conditions and configurations should be preferred. However, performing experiments with more than a few nodes requires a great effort from participators in the experiment. Even before the experiment, the work load can be considerable, as all the equipment must be prepared with the correct software versions, charged batteries, etc. Thus, the substantial work effort required to perform large experiments makes it more efficient to perform research using simulators and theoretical analysis within the limits of a PhD fellowship. However, it is important to acknowledge the limitations of these methods when drawing conclusions from the work.

The performance results presented in this thesis were obtained using simulations. In the introductory phase of the work on paper A, experiments were performed using real laptops. However, it became evident that the driver implementations

were difficult to investigate. The issues with rerouting time were then investigated further in the simulator, which gave much easier access to the inner workings of the MAC protocol. In the subsequent papers, larger topologies were investigated, and with these topologies it would have required considerable resources to perform investigations using experiments rather than simulations.

The simulations were performed using the Network Simulator 2 (NS-2) [53]. It has been under continuous development since 1996, and is one of the most popular simulators in the MANET research community. In the duration of the thesis work, the version number of the simulator has changed from 2.30 to 2.34, illustrating that it is a software tool still receiving attention by the community. The changes during these version increments include bug fixes and additional functions. However, the changes have not impacted parts of the simulator code used in the simulations for this thesis work. An advantage of using the NS-2 simulator is the large community that contributes both to the code base and to the mailing list.

The NS-2 simulator produces a trace file, and this file was parsed using a java program written by the thesis author. The program was altered in several rounds to extract the data of interest, ranging from the actual node position at each transmission to the total throughput of the system. It was necessary to decide how the measurement window was to be enforced, and it became evident that each packet had to be traced individually from source through forwarding nodes, collisions and retransmissions to destination. The measurement data was extracted from the events of all packets that were created during this window. All events were traced and the interesting data was extracted for these packets, even if the packet events continued past the measurement window.

The OLSR code [54] used with the NS-2 simulator was developed by researchers at the University of Murcia, Spain. The version number was 0.8.8. One difference between the implementation and the IETF OLSR Experimental RFC was noticed, in that the implementation recalculated the MPR set and the routing table every time a new control packet was received, while the Request For Comments (RFC) states that the routing table only need to be updated when a change is detected in the local link information base or the topology set. However, this did not impact the result of the simulations, although the simulation processing time was increased to some extent.

Mobile topologies were studied in all papers, and in the papers B through E the topologies were generated using a tool developed by Santashil PalChaudhuri [55]. The node distribution was investigated through plotting each node's position every second for the duration of the simulation. It was confirmed that the tool generated topologies behaving as expected, based on the input parameters to the

mobility model. During the finalizing work with this thesis, it was discovered that due to a misunderstanding the papers B, C and D state that the Random Direction (RD) mobility model was employed, when it in fact was the Random Walk with Reflection (RWR) mobility model that was employed. The two models are described in detail in [56]. Using the RWR mobility model, the node selects a direction, speed, and a time to select a new direction and speed. If the node encounters one of the edges of the simulation area, it reflects off the wall in the reflected direction with the same speed. On the other hand, the RD only selects a new direction and speed upon reaching one of the edges of the simulation area. The RD model was designed to avoid the clustering of nodes in the center of the simulation area. However, this leads the RD model to have higher probability of nodes being near the edge of the simulation area than in the middle. This increases both the risk of partitioning and the average hop length. When comparing the RWR and RD models, there are differences in their resulting topologies. However, since the mobility model was employed to achieve topology dynamics, and the comparison of different mechanisms and solutions was performed with the same topologies, the choice of mobility model is not seen as having affected the validity of the results.

The results obtained by using the simulator are the result of a high level of abstractions at the physical layer. The propagation model employed in the works of this thesis is a perfect one, with a linear signal strength decrease with distance. Thus, the results are not directly transferable to the real world, and cannot be directly relied upon as performance metrics. However, observed performance trends should be valid also for real world experiments, unless they are based on areas in the simulator that are not good abstractions of the real world.

The evaluation of mechanisms in this work was done using throughput or goodput as success indicators. Other metrics were studied, such as delay, jitter, different type of loss, and these were used as indicators of the behavior of the mechanisms. The main reason why packet delivery in form of throughput or goodput was used as the main evaluator of the success, is the fact that other metrics like delay and jitter are very application-specific. Some applications, such as live voice communication, are very sensitive to delay and jitter, while other applications like file transfer or e-mail are insensitive to these properties.

The traffic and topology scenarios that were used in this work were selected on the basis that they were simple and helped to underline the strengths and weaknesses of the studied mechanisms. Ad hoc networks are, due to their nature, able to support highly varying types of scenarios, with different applications, topologies, density, mobility, etc. Therefore, it is impossible to study all varieties in the use of ad hoc networks. The scenarios simulated in this work explore generic ad hoc

network behavior, using both User Datagram Protocol (UDP) and Transmission Control Protocol (TCP) traffic flows, emulating voice traffic, SA traffic, file transfers and other traffic types. While ad hoc networks employed for operational use will have more complex traffic flows, the scenarios have helped achieve a better understanding of some ad hoc network challenges.

Chapter 5

Contributions and summary

5.1 A summary of the contribution as a whole

The work in this thesis addresses routing and group communication in MANETs with the aim to improve the performance of the network, both for each individual node, and for the network as a whole. The contributions can be seen as two separate works, where the papers A, B and C focus on increasing the throughput of unicast transmissions, and the papers D and E consider multicast transmissions in MANETs and enhancing all or parts of the network traffic flows. Both of these aspects are important for the overall project objective, enabling better ad hoc network services for emergency and rescue personnel.

The work in the papers A, B, C and E addresses challenges faced by the routing protocol in mobile topologies. The papers A and C address losses due to extended rerouting times and propose two alternative solutions to reduce the rerouting time. Paper B investigates the effects mobility has on a static gateway with mobile network nodes with traffic flowing into and out of the network through the gateway. Paper E investigates how mobility affects two different multicast forwarding algorithms for SMF.

The papers A, B and C aim at understanding some of the mechanisms that reduce the network throughput. The papers A and C mainly look at mobility, while paper B identifies other factors such as interference, path length and channel reuse as affecting throughput. The papers D and E explore group communication, and are more focused on service quality for PTT. The service quality is measured using goodput, or Packet Delivery Ratio (PDR). Through exploiting the network resources as much as possible, the aggregated network throughput (papers A, B

and C) can be increased. However, increasing the network throughput may not imply increased service for the application, if other service factors are reduced, such as the packet loss. The PTT service is a good example of an application vulnerable to loss, but without high throughput demands. The solutions in both of the papers D and E are able to improve the PDR of the PTT traffic, while allowing other traffic in the network at the same time.

5.2 Contribution of paper A: Rerouting Time and Queueing in Proactive Ad Hoc Networks

A routing protocol can determine topology changes using route timeout, neighborhood discovery or LLN, as described in Section 2.4.2. The default method for OLSR and other proactive routing protocols is neighborhood discovery using polling packets referred to as HELLO messages. In case of a link break, the routing protocol has to time out before the rerouting is completed. The delay may cause packets to be pushed down to the interface queue with an invalid next hop address, and the MAC layer wastes time and resources while attempting to transmit these packets.

5.2.1 Related Works

Voorhaen and Blondia investigate the difference between OLSR HELLO neighbor sensing, Fast-OLSR and IEEE 802.11 LLN in [57]. While the authors show that the delay in link break detection creates a buildup of packets in the interface queue, no analysis of this behavior is offered.

An effective solution to the rerouting problem is proposed in [58], where routing is delayed until the medium is available. Named "ingress queuing", the process of queuing the outgoing packets is altered so that packets are not assigned a next hop until the MAC layer is ready to transmit a packet. This way, if a packet is being retransmitted due to a link break, the next packet is not assigned a next hop until the current packet is discarded, avoiding a new futile transmission attempt towards a lost next hop.

In [59], the OLSR protocol is studied and sought improved by introducing a metric named Route Change Latency (RCL). The RCL is a metric defined as the time in seconds from a link break physically occurs until the routing protocol detects the break. This metric is studied in a real network environment with a non-static topology. Based on the difference between the expected RCL and the RCL achieved on

the real system, the authors propose to tune OLSR parameters to reduce the RCL, thereby optimizing the network behavior.

5.2.2 Contributions

In neither of the related works is there an explanation of why and how the rerouting time is affected by a link break and the following packet buildup due to MAC layer retransmissions, although [57] notes that the packet buildup is a problem, and [58] proposes a solution for it. Landmark presents one well-performing solution in [58], but it depends on a link between the routing and MAC layer to perform routing immediately before transmission. Gomez et al. [59] focus on adapting routing protocol parameters to achieve a better network behavior. Our contribution describes the relation between link breaks, offered load, MAC layer parameters and rerouting time. Our proposed solution relies on the MAC layer to, by itself, dynamically adapt to the situation, without a need to communicate with the routing layer.

Paper A shows that rerouting time depends on the size of the interface queue. In a MANET where nodes move frequently, the probability of connectivity loss between nodes can be high, and communication sessions may easily lose connectivity during transmission. The routing protocol is designed to reroute, i.e., find alternative paths, in these situations. This rerouting takes time, and the latency is referred to as the rerouting time. The rerouting time is defined as the time interval starting from the last HELLO message sent by the downstream node is received by the upstream node, lasting to the moment when the downstream node receives the first packet from the upstream node after the link break.

Proactive routing protocols, such as OLSR and Open Shortest Path First with MANET Designated Routers (OSPF-MDR), maintain links through the exchange of control packets. A link break is normally not detected until either a certain number of consecutive HELLO packets have been lost, i.e., the lack of periodic updates results in a link timeout. (LLN is not considered in this paper.) Using the default parameter settings of OLSR and OSPF-MDR, a link break should normally be detected after approximately 4–6 seconds.

The IEEE 802.11 [22] is the most widely used Wireless LAN (WLAN) technology for ad hoc networks today. The MAC layer in this standard implements link layer retransmissions, a very useful mechanism. In noisy environments, MAC-layer retransmissions provide fast recovery from packet loss. However, this mechanism can be detrimental in mobile environments, since packets are attempted retransmitted to destinations out of reach.

At the link layer, the outbound packets are received by the device driver queue (referred to as *the interface queue*). This queue should be of a size large enough to allow traffic to be sent without loss from applications at a rate that can be higher than the instantaneous network capacity, as the network bandwidth can vary over time.

$$t_{rerouting\_max} = t_d + \frac{B}{R_{out}} \tag{5.1}$$

Simulations show that the rerouting time depends on the number of MAC-layer retries in combination with the interface queue size, and a model is developed to predict the rerouting time ($t_{rerouting}$). Equation 5.1 calculates the maximum rerouting time, i.e., where the queue at some point during the rerouting is filled to its maximum. In this equation t_d is the routing protocol's HELLO timeout value, B is the queue size, and R_{out} is the rate at which the packets are sent from the interface. This rate depends on several factors, including the maximum number of retransmissions for each packet, the transmission rate, use of RTS and CTS, the backoff calculation, etc.

$$t_{rerouting} = \frac{R_{in}}{R_{out}} \cdot (t_d - t_b) + t_b \tag{5.2}$$

For rates into the queue (R_{in}) lower than those filling the queue during the rerouting, Equation 5.2 gives a good estimate of the rerouting time. In addition to the variables from the former equation, t_b is the time interval between the last received HELLO packet from the destination, to the actual link break.

In addition to predicting the rerouting time, the paper proposes a solution to the problem of the rerouting time problem, named "Adaptive retry limit". It is a MAC-layer solution proposing that the retransmission limit is decremented by one for each discarded packet. Through the reduction of the number of retransmissions per packet, the rate R_{out} is prevented from decreasing when a link break occurs. Simulations and analysis show that the solution eliminates the entire problem of increased rerouting time in many situations.

5.3 Contribution of paper B: Gateways and Capacity in Ad Hoc Networks

Ad hoc networks used in emergency scenarios may be expected to support connectivity with other networks. This must be provided using one or several gateways,

and these gateways will be the focus of traffic in and out of the ad hoc network. Understanding how the relative position of gateways and other gateway-related behavior affect the network capacity is useful for improving the QoS in ad hoc networks.

5.3.1 Related Works

Ahmed et al. [60] propose an architecture where deployed mobile gateways can provide range extension. The authors propose an algorithm calculating the optimal trajectory for one gateway in an ad hoc network is presented. The trajectory is based on a combination of a calculated optimal position for the gateway coupled with the gateway's movement limitations. The optimal position is defined as a weighted geographic centroid with input parameters being the ad hoc network nodes' position, offered load and node priority.

A way to determine the nominal capacity of a network is presented in [61]. The Bottleneck Collision Domain (BCD), representing the nominal network capacity, is the single link in the network yielding the lowest capacity due to interference with other links. The paper also presents the estimation of the available throughput per node, assuming absolute fairness among the nodes.

In [62], Li et al. address the challenge of placing a number of gateways in a multi-hop wireless mesh network to optimize the throughput. The mesh nodes are static, and the optimization is based on calculating the optimal throughput considering the interference on the links based on the Fixed Protocol Interferences Model (fPrIM). Aoun et al. [63] also address gateway placement optimization in wireless mesh networks, where optimization is based on various QoS constraints. The solution is compared to other alternative schemes by the number of required gateways for various scenarios.

Finally, in [64], the optimal gateway position in a Wireless Sensor Network (WSN) is investigated briefly. A WSN is very limited in terms of energy usage, and energy efficiency is important. To preserve energy, the gateway position optimization is based both on load balancing among the nodes and on limiting the number of relays for each traffic flow. While not proposing a solution, the author concludes that the least optimal placement for a gateway is at the edge of the network, and that the gateway ought to be placed in the center of the topology.

5.3.2 Contributions

While several of the related works focus on determining an optimal position for gateways, our contribution instead investigates how the position of a gateway affects the throughput. The work also considers the effect of mobility, which is not addressed by the other related works. Thus, our contribution sought to create a better understanding of how varying gateway numbers and positions affect the network performance.

Paper B shows that the capacity of an ad hoc network can depend on the relative position of the gateway, or multiple gateways, in the ad hoc network. While many papers propose to connect ad hoc networks with external networks, the position of the required Gateway (GW) between the networks has received little attention. An intuitive position for the GW is along the edge of the ad hoc network. Here, the GW provides connectivity between the networks while reducing the necessary stretch of the other network as much as possible. This utilizes the multi-hop property of the ad hoc network. However, the simulation results show that the average path length is the parameter that affects capacity the most. This implies that the optimal position for one GW is in the center of the network.

Other contributions of the paper include:

- Identifying different characteristics between uplink and downlink traffic through the GW.
- Routing protocol control traffic is vulnerable for collisions. As a consequence a high number of data packets are discarded due to *no route*.
- Interference between two GWs can impact the throughput negatively, especially in networks dominated by downlink traffic.
- An increasing number of GWs affects the throughput positively, until all nodes reside in one hop coverage of a GW.
- The effect of not implementing handover was shown to reduce throughput with time in a randomly moving topology, in effect cancelling the gain achieved from an extra GW in the two GWs scenario.

The paper was able to identify the MAC-layer retransmissions which increase the rerouting time, as described in paper A. This can be seen in Figure 5.1, where a ring of transmissions forms around the receiving gateway just outside the transmission area edge.

An important difference between uplink and downlink traffic was identified. At high loads the longer rerouting time reduces the obtainable throughput in the

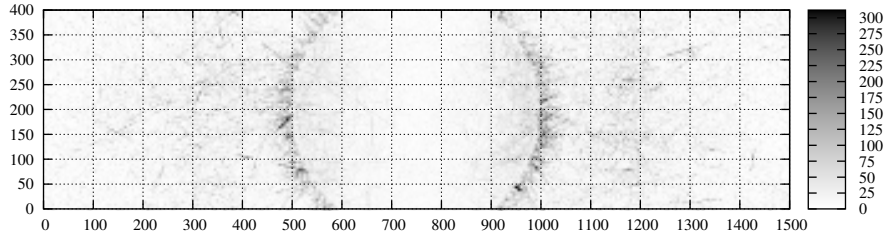


Figure 5.1: Simulation area with transmission plot for the reference configuration.

downlink scenario compared to the uplink scenario. This is because the GW is the only originating sender in the network. In topologies with mobility, the stale routes force the GW to go through exponential backoffs for each packet to an unreachable destination. In the uplink scenario, other senders have packets to send in these large backoff periods, while in the downlink scenario, the effect of this reduced rate out of the interface queue of the GW is lower total achievable throughput.

With one GW, the average path length between the gateway and the other nodes defines the capacity of the network. In addition, the loss due to *no route* is very high at high system load, especially with the GW placed near the edge of the simulation area. This indicates that the TC messages of the routing protocol fail to reach all network nodes.

The scenarios with two GWs emphasize the relation between the average path length and the capacity of the ad hoc network, and another effect also manifests itself. If the two GWs are inside each other's interference range, they are not able to transmit independently. This does not give a large impact in the uplink scenario, where all surrounding nodes transmit traffic towards the GWs. In the downlink scenario, however, the GWs are unable to transmit independently of each other. Thus, when they move out of each other's sensing range, the throughput has a sudden increase, while the loss due to drops from the interface queue is reduced. However, the reduction in interface queue drop does not transfer directly into increased throughput, since the loss due to maximum MAC layer retries also makes a sudden jump here.

The number of GWs has impact on the capacity of the network. The more GWs there are in the network, the more capacity is made available. The capacity increases with a reduction in the average path length, on average requiring fewer transmissions per packet in the network. However, when all nodes are in one hop range of a GW, adding more GWs yields no further gain. (Load balancing was out of scope of the paper.)

The effect of handover, and lack of handover, between the GWs in scenarios with two GWs on the throughput was also investigated. The conclusion was that without handover, the system over time gives the same performance as that of one GW. This is founded on the expectation that mobility leads to random distribution of the nodes, i.e., 50% are connected to its closer GW and the other half is connected to the farther GW. While more than 50% of the nodes are connected to the closer GW, the performance lies between that of two GWs with handover, and the performance of one GW.

5.4 Contribution of paper C: Routing with Transmission Buffer Zones in MANETs

Routing in ad hoc networks is commonly performed using shortest path routing, where the hop count determines the chosen route. Fewer hops may imply the use of longer links, and in mobile scenarios these longer links are most likely to be broken, compared to shorter links. Evaluating whether a link is reliable is challenging. As described in Section 2.4.1, many metrics can give information about the link quality, including ETX and Received Signal Strength Indicator (RSSI). However, if the topology changes rapidly, the link information soon gets old, and disseminating the information throughout the network may cause many routes to be calculated based on the wrong information.

The work in paper C focuses on the problem of routing over links that are about to be broken. Using a different metric besides shortest path locally to select paths can reduce the rerouting time. However, evaluating which links that are likely to break requires more information about the link besides hop information.

5.4.1 Related Works

One way to determine the length of a link is by using geographical information. Each neighbor announces its position, and the link distance can be calculated accordingly. This is done in [65], where GPS information is used with a reactive routing protocol to continuously measure the length of the link. If the link is too long, a preemptive local route repair is initiated.

GPS information is also employed in [66], for both reactive and proactive routing protocols to predict at which time the link or the route will be broken. Route reconstruction can be initiated proactively, based on this information. This eliminates transmissions of control packets otherwise needed to reconstruct the route

and thus reduce overhead. The problem with using geographical information for estimating whether a route must be repaired, is that link quality is not necessarily directly related to distance.

The authors of [66] propose to use signal strength as metric to determine the link expiration time if GPS information is unavailable. The signal strength metric is also the employed metric in [67–69], but with a focus on reactive routing protocols. Intriguingly, the solution by Goff et al. [67], while for a reactive routing protocol, bears resemblance to the safe and unsafe zones in our contribution. Signal strength using OLSR is addressed in [70], where it is used in combination with the hysteresis mechanism to evaluate whether a link is improving or deteriorating.

5.4.2 Contributions

The solutions of the related works are based on different ways to determine if a link is untrustworthy and cannot be used for routing. Likewise, the rerouting solution itself is dependent on whether it is based on a reactive or proactive protocol. Our contribution is different from the other solutions in that the proactive routing protocol makes it possible to evaluate whether another route exists to the destination, and not discard a link if it means a destination is made unavailable. This is a big problem for the solutions employing a reactive protocol, but is also done for proactive protocols. The overhead is also limited with our contribution, while other works propose to increase the number of transmitted control packets when a link is considered at risk of failing.

Paper C proposes to divide the transmission area of a node into a safe zone and an unsafe zone, and distribute this information to all neighbors. A node can select routes preferring neighbors in the safe zone, based on the knowledge of whether its neighbors are in the safe or unsafe zone. This information is only distributed locally using HELLO packets, thus avoiding the risk of nodes far away making decisions based on old link quality information. Signal strength is used to determine whether a neighbor is in the safe or unsafe zone. This metric is directly dependent on the distance, with the radio propagation model used in the simulator, but the evaluation of which zone a neighbor resides in could also be based on other link quality metrics, such as BER, link layer retransmissions etc.

It is proposed to use the signal strength of received routing control packets at the source node (A) to determine the relative distance between the node and its neighbors. The transmission range of the nodes is divided into a safe zone and a buffer zone (Figure 5.2). The nodes in the buffer zone are considered to be at a higher risk of disappearing during the next HELLO timeout period. These buffer zone

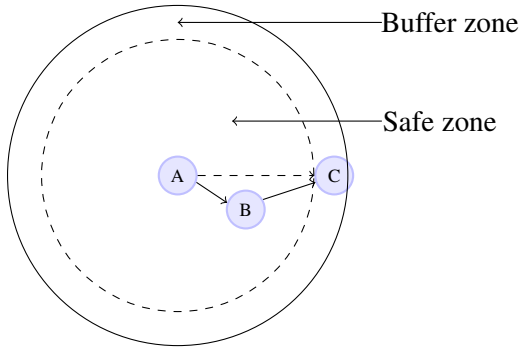


Figure 5.2: Transmission area zones of node A with safe node (B) and unsafe node (C).

nodes are first sought to be routed around. If another node (B) exists that is both in the source node's safe zone and also has the node C in its own safe zone, it would be safer to route packets via node B to node C.

The routing solution requires a change in the HELLO message structure. While the HELLO message lists all known neighbors, one bit has to be included for each symmetrical neighbor to indicate which zone the neighbor is considered as existing in.

The routing algorithm of OLSR is altered so that routes to all known destinations in the network are first attempted found using only neighbors within the safe zone. Unsafe nodes are only used for routing if there are destinations that cannot be reached using safe nodes as next hop. In the case where an unsafe node has to be used for routing, but neither of the safe neighbors has it in its safe zone, the unsafe neighbor should be used directly, instead of going via a safe neighbor. This is also the reason why the HELLO messages must include the zone parameter.

Some issues stand out as unresolved with this solution:

- In the real world, signal strength is known to be highly variable [71]. First, it means that it is difficult to determine the exact distance between two nodes, and whether a neighbor is in the buffer zone or the safe zone. Second, it cannot be expected that the transmission range is a perfect circle. This could affect the probability that a neighbor in the buffer zone will disappear or not in the next HELLO timeout interval.
- This method causes some increase of loops in the network, since the routing is more complex, but also because more packets are rescued before being discarded because of reaching the maximum MAC retransmissions.

- In a static network, this solution will only cause longer paths without providing any gain, since none of the nodes are apt to move out of range of any other nodes. However, other metrics like packet loss probability or ETX may make this solution provide gain also in static networks.
- The concept of "safe" links require unsafe links to be links where the distance between the two nodes are such that the neighbor is likely to move out of the transmission range during the next HELLO timeout period. Another dilemma is measuring the signal strength of routing control packets to determine the link distance. In [2], it was demonstrated how routing control traffic using different modulation than the data packets can create "gray zones" where the link is viewed as existing, but no data traffic can be received successfully.

5.5 Contribution of paper D: Preemption Mechanisms for Push-to-Talk in Ad Hoc Networks

Ad hoc networks have limited resources. Supporting group communication services with a high resource demand, such as a PTT voice application, in a network with other traffic is very challenging. This was also explained in Section 3.4.

5.5.1 Related Works

Most network-layer QoS-improving solutions for ad hoc networks focus on QoS routing, while scheduling and priority solutions reside mostly at the link/MAC-layer [72]. QoS for multicast was studied in [73], but here too most of the work was focused on routing and the enhancement of multicast distribution.

Some QoS solutions implement Call Admission Control (CAC), but only a very few, such as [74, 75] consider preemption. Canales et al. [74] propose cross-layer routing using a TDMA-based MAC-protocol and a reactive routing protocol. Works on the preemption of traffic flows primarily focus on the preemption of the real time flows, such as Service Differentiation in Stateless Wireless Ad Hoc Networks (SWAN) [75].

In [76], Elmasry et al. propose a model to manage QoS for Secure Tactical Wireless Ad Hoc Networks, directed towards the future US Army tactical backbone network. Based on traffic characteristics measurements, a congestion severity level

can be calculated. This level then defines which admission and preemption policies that should be adhered to.

5.5.2 Contributions

From the related works, it was clear that the preemption of lower priority flows has barely been studied from a routing layer perspective for group communication in MANETs. The combination of preemption and scheduling, mainly based on routing layer decisions was interesting to pursuit, especially for a type of network where most or all nodes are interested receivers of the priority multicast traffic, such as is the case for PTT traffic in ad hoc networks.

Paper D proposes three ways to do preemption in ad hoc networks with PTT traffic. Ad hoc networks to be used for emergency services should support *Push-to-Talk*, as PTT is a very important tool for organizing emergency operations. PTT traffic is better forwarded using multicast, as this is a more bandwidth efficient distribution method for point to multi-point traffic than that of unicast. In such operations, different roles may carry different priorities, which should be reflected in the traffic QoS. For instance, the leader should be able to reach through with important orders, regardless of other ongoing traffic, whether it is voice or other applications.

If no mechanisms for prioritizing traffic are employed, the packet loss for multicast traffic is reduced similarly to the background traffic as the background traffic load is increased. This reduces the voice quality and makes communication very difficult. Simulations with priority queuing show that it is possible to halt the increased packet loss for priority traffic as the background traffic is increased. However, this break only occurs after the goodput is already reduced by 20% due to collisions.

Three different preemption methods where proposed:

- *Discard*
- *Buffer*
- *Low Priority Window (LPW)*

The *Discard* method is the basic preemption method. Here, all lower priority traffic is discarded and not forwarded as long as higher priority traffic is observed. This lasts until a predefined period after the last higher priority packet. The method is very effective for discarding unicast UDP background traffic, but it is negative that the background traffic flows are unable to utilize the remaining capacity while the high priority traffic roams the network. This is the case even if the high priority traffic had not been affected by the background traffic.

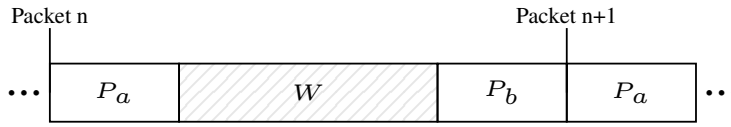


Figure 5.3: Preemption with a window to transmit the low priority packets.

Therefore, two improved preemption mechanisms, the *Buffer* and *LPW* mechanisms are proposed. The *Buffer* method is based on the premise that the background traffic still has value when the high priority session is over. If so, traffic that was on the way to the destination when the high priority traffic interfered, can continue on its way afterwards. I.e., the resources already spent to forward it are not in vain. A challenge with the *Buffer* method is that the queue where the traffic is buffered is of a limited size, and if the interface queue gets full, it must be decided what data to discard. Another problem is the fact that mobility may cause next hop destinations to move out of reach during the buffering time, but this can be addressed through the use of *Ingress queuing* [58]. Ingress queuing is a method where the packets are routed when they are about to exit the interface queue. This way, they are routed using the latest available information, instead of the information available at the time when they were queued.

The *LPW* method employs buffering in the same manner as the Buffer method, but in addition it exploits the time in between each high priority packet (W) to transmit low priority packets (Figure 5.3). If the high priority traffic is multicast voice, the produced traffic will move in packet "waves" through the network. The packet is distributed outwards from the originator and between each packet of this flow there is high priority traffic silence where background traffic can propagate the network.

Finally, the preemption mechanisms are tested with TCP background traffic. TCP is known to be unsuitable for ad hoc multi-hop networks due to its greedy rate control [77]. The results show that without any prioritization, the goodput of the priority traffic is below 20%. The results are better with the Discard method, at 57.5%, and each of the two other improved preemption mechanisms yield better than 50% performance. Thus, the results show that while the mechanisms can yield better performance than the normal behavior, the preemption mechanisms are affected by extensive background traffic bandwidth use.

5.6 Contribution of paper E: Optimized Group Communication in Tactical Military Networks

Traffic in tactical military networks will include voice (PTT) and some kind of Situational Awareness (SA) traffic. These two group communication service types have widely different QoS demands, with voice tolerating neither large packet loss nor large delay. The SA service, on the other hand, allows more packet loss as it is periodical, and the delay may be seconds without being a problem for the service. Nevertheless it is important that these two service types can exist in the network simultaneously, requiring QoS control in the network.

As described in Chapter 3, when most of the nodes in the network are interested receivers, an efficient flooding protocol, such as SMF, can be the best alternative. However, through the work in paper D, it became evident that the preferred forwarding algorithm, the S-MPR, is vulnerable for high mobile topologies. Therefore, this algorithm might not suit all types of traffic.

5.6.1 Related Works

Several works have evaluated and proposed enhancements to the MPR selection mechanism in OLSR, the basis of the S-MPR and NS-MPR algorithms. Jacquet et al. investigates the MPR selection in two scenarios in [78], and Busson et al. studies the MPR selection in [79].

Mobility as a challenge to OLSR is addressed in [80], where nodes experiencing a high degree of mobility reduces the HELLO interval, aka. Fast-OLSR.

Qin and Kunz discuss mobility metrics to enable adaptive MANET routing in [81]. The same authors discuss adaptive routing in MANETs in [82], performing a case study using the number of monitored link breaks as key mobility metric.

In [83], Cho and Adjih propose to use MPRs to optimize multicast forwarding, much in the same way as SMF. A directional MPR algorithm is proposed, where the MPR forwarding algorithm is optimized through only forwarding packets if on the shortest path between the source and destination. Also, MPR flooding and a combination of MPR flooding and mesh is proposed. Another efficient broadcasting method based on hop-limited shortest-path trees is proposed in [84].

The main basis for the work in paper E is found in [47], where the S-MPR, NS-MPR and two other forwarding algorithms for SMF are studied. It was shown that the S-MPR and NS-MPR had very different properties when stressed with

mobility and offered load. NS-MPR performed similar to CF, with high resilience to mobility, at the cost of a high number of redundant transmissions. S-MPR provided the best performance at high loads. However, little attention was paid to the algorithm performance at low traffic loads, (e.g., PTT or SA traffic).

5.6.2 Contributions

Of the related works, our contribution is most closely related to the work in [47], where several SMF forwarding algorithms are investigated. However, this work does not shed light on the possibility to choose between different algorithms, depending on what properties are desired of the forwarding process. The focus was only on limiting overhead while achieving high packet delivery ratio. Our contribution sought to investigate the advantages of such a selection between the algorithms. In addition, the selection mechanism was tested in a tactical military network setting, where high priority PTT traffic has to reside together with lower priority SA traffic.

The contribution of paper E was threefold:

- Showing that S-MPR and NS-MPR behave differently when challenged with mobility and varying traffic load.
- Demonstrating how a *radio load* metric can be used to select the best suited forwarding protocol of S-MPR and NS-MPR.
- Showing how the accumulated goodput of simultaneous PTT and SA traffic can be increased through employing a preemptive switch to S-MPR-based forwarding for the SA traffic.

The efficient flooding framework protocol SMF shares its vulnerability to mobility with that of the routing protocol that elects the MPRs and uses them to forward control packets. This vulnerability is especially pronounced with the S-MPR forwarding algorithm. It is also shown that S-MPR is more vulnerable to collisions than NS-MPR.

To increase the performance of SMF forwarded traffic in varying network conditions, the S-MPR and NS-MPR algorithms are sought combined using radio load. This metric measures how much of the time the interface is busy either transmitting or receiving packets. It proves to be very well suited for balancing between the two algorithms and in a distributed way selecting the optimal traffic forwarding algorithm.

It is best to employ the radio load solution with traffic of varying load, since traffic

of low load is better forwarded using only NS-MPR. This is especially true for traffic that is vulnerable for packet loss, such as PTT. When PTT and SA traffic is run simultaneously, the PTT traffic achieves the highest goodput when the PTT traffic is forwarded with NS-MPR, and the SA traffic with S-MPR. The radio load metric is able to make the SA traffic forwarded using S-MPR when the traffic load is high enough. However, it is not necessarily able to choose S-MPR when a PTT traffic session is initiated in the network. It is only better for the SA traffic to change to S-MPR when the network is very close to congestion. The performance of the PTT traffic is reduced due to collisions at much lower loads than such loads causing congestion. The results show that using a preemptive switch to force the SA traffic to be forwarded using S-MPR while a PTT session exists in the network enhances the performance of the PTT traffic. At the same time, the SA traffic is allowed to be forwarded, although with a greater packet loss as result.

5.7 Contribution of additional work

The papers F and G constitute the additional work performed within this thesis. Paper F has a focus on load reduction in the MANET, as well as increased performance for the intranet communication. Paper G has a focus on load balancing. Both papers aim at the goal of increasing the performance of MANETs.

Paper F proposes a metric that enables the routing of internal MANET traffic, i.e., traffic between two MANET nodes, to be routed via a non-interfering backbone network (called transit routing). This metric can be added to existing routing protocols. The cost metric algorithm is implemented in the OLSR protocol and many simulations on three random topologies are executed in NS-2. Results show that the cost metric algorithm is able to perform the desired behavior, triggering transit routing in situations which in most cases resulted in improved performance in terms of throughput. Although the work is focused on the optimization of the throughput on one single flow of traffic, simulations with background traffic also shows a considerable average enhancement in the throughput (50.4%).

Paper G addresses load balancing in random MANETs topologies with gateways. The paper explores the potential benefits of load balancing when the network topologies are random. Two different load balancing mechanisms are considered and investigated using simulations in NS-2. With the aggressive mechanism, the traffic load is equally distributed on the two different gateways, at the cost of additional overhead in terms of tunneling overhead and network resources. The moderate mechanism, on the other hand, reduces the additional implementation complexity. The results show that load balancing may increase the overall throughput.

The enhancement is low for uniformly distributed topologies, but with artificially asymmetric topologies the average enhancement is clear. The results indicate that with random (mostly uniform) topologies, load balancing has a limited potential.

5.8 Concluding remarks

The contribution of the papers A to E is the increased performance of MANETs. However, some of the suggested solutions have a potential for further exploration and optimization.

In paper A, the potential unfairness related to the nodes that experience a reduced number of retransmissions, and thus less time in backoff, should be investigated. Other ways to exploit the knowledge of the correspondence between the queue length and the rerouting time should also be investigated, for instance the dynamical configuration of the queue size.

Paper B investigated gateway positioning and handover, and one aspect to investigate further is the setting where continuous handover is too costly to maintain, either due to signaling cost or due to interruptions in the end-to-end flow. Based on the relative mobility, handovers could be triggered at specific times or when the performance of the network sinks below a defined level. Another issue to pursue is to develop algorithms indicating to network managers where the optimal or local optimal GW positions are, based on the current network topology and the current traffic pattern. Third and last, heterogeneous access networks, i.e., situations where GWs are connected to dissimilar access networks in terms of capacity or latency, should be investigated in the context of optimal GW positioning.

The transmission buffer zones mechanism presented in paper C is left with some unresolved issues. First, the size of the buffer zone is dependent on the expected mobility in the network, as well as the local propagation properties affecting the link reliability. Thus, a dynamic variation of this size is desirable for optimization purposes, but could prove hard to define. Second, through arguing for increased path length to avoid unreliable links, the mechanism addresses a well known problem with the MHC being a problematic metric in MANETs, which is currently a much discussed topic in the standardization of OLSRv2.

The work in paper D shows great promise regarding the possibility to use Push-to-Talk (PTT) in MANETs. However, a problem with the buffer and LPW methods is the delay that these cause for the background traffic. Real experiments can show whether the mechanisms can really help rescue some of the background traffic, or if the transport protocols end up forcing retransmissions that cause more harm

than good to the network service quality. In addition, real experiments will tell more about the QoE, and whether the service can be used with the proposed mechanisms.

Finally, paper E showed interesting results for forwarding OLSR control packets using S-MPR. The OLSRv2, as specified by the draft [13], currently continues to use S-MPR as flooding algorithm without offering alternatives. A study should be carried out investigating the effects of dynamically using alternative forwarding algorithms with the OLSRv2 protocol. The radio load metric has showed promising results, but should be tested in experiments. The radio load metric should be further optimized to anticipate the optimal threshold in varying topologies.

The work presented in this thesis has focused on improving the QoS in MANETs. However, there are still many unresolved issues in the research of better QoS in MANETs, both for group communication and unicast communication. It is the hope of this author that the contributions in this thesis can be of use in the ongoing effort to enable MANETs for use in emergency and rescue scenarios.

Bibliography

[1] L. Kleinrock and J. Silvester, "Optimum transmission radii for packet radio networks or why six is a magic number," in *IEEE National Telecommunications Conference, (NTC '78)*, vol. 1. Institute of Electrical and Electronics Engineers, December 3–6 1978, pp. 4.3.1–4.3.5.

[2] H. Lundgren, E. Nordström, and C. Tschudin, "Coping with communication gray zones in IEEE 802.11b based ad hoc networks," in *Proceedings of the 5th ACM international workshop on Wireless mobile multimedia (WOWMOM)*. New York, NY, USA: ACM, 2002, pp. 49–55.

[3] I. Stojmenovic, A. Nayak, and J. Kuruvila, "Design guidelines for routing protocols in ad hoc and sensor networks with a realistic physical layer," *Communications Magazine, IEEE*, vol. 43, no. 3, pp. 101–106, March 2005.

[4] H. Zimmermann, *OSI reference model—The ISO model of architecture for open systems interconnection*. Norwood, MA, USA: Artech House, Inc., 1988, ch. 1, pp. 2–9.

[5] V. Kawadia and P. R. Kumar, "A cautionary perspective on cross-layer design," *Wireless Communications, IEEE*, vol. 12, no. 1, pp. 3–11, 2005. [Online]. Available: http://dx.doi.org/10.1109/MWC.2005.1404568

[6] X. Hong, K. Xu, and M. Gerla, "Scalable routing protocols for mobile ad hoc networks," *Network, IEEE*, vol. 16, no. 4, pp. 11–21, Jul/Aug 2002.

[7] Z. Haas, "A new routing protocol for the reconfigurable wireless networks," in *Universal Personal Communications Record, 1997. Conference Record., 1997 IEEE 6th International Conference on*, vol. 2, 12-16 1997, pp. 562 –566 vol.2.

[8] Mobile Ad-hoc Networks (manet) working group. (2009, October) IETF. [Online]. Available: http://www.ietf.org/dyn/wg/charter/manet-charter.html

[9] S. Corson and J. Macker, "Mobile Ad hoc Networking (MANET): Routing Protocol Performance Issues and Evaluation Considerations," RFC 2501, pages 1–12, pp. 1–75, January 1999, network Working Group. [Online]. Available: http://ietf.org/rfc/rfc2501.txt

[10] I. D. Chakeres and C. E. Perkins, "Dynamic MANET On-demand (DYMO) routing protocol," IETF Internet-Draft, July 2010, work in progress. Intended status: Standards Track, Expires: January 27, 2011. [Online]. Available: http://www.ietf.org/id/draft-ietf-manet-dymo-21.txt

[11] C. Perkins and E. Belding-Royer, "Ad-hoc On-Demand Distance Vector Routing," in *Proceedings of the 2nd IEEE Workshop on Mobile Computing Systems and Applications*, February 1999, pp. 90–100, new Orleans, LA.

[12] T. Clausen, P. Jacquet (editors), C. Adjih, A. Laouiti, P. Minet, P. Muhlethaler, A. Qayyum, and L.Viennot, "Optimized Link State Routing Protocol (OLSR)," RFC 3626, pages 1–75, pp. 1–75, October 2003, network Working Group.

[13] T. Clausen, C. Dearlove, P. Jacquet, and the OLSRv2 Design Team, "The Optimized Link State Routing Protocol version 2," IETF Internet-Draft, MANET WG, draft-ietf-manet-smf-09, September 2009, work in progress. Intended status: Standards Track, Expires: March 29, 2010. [Online]. Available: http://tools.ietf.org/html/draft-ietf-manet-olsrv2-10

[14] A. Qayyum, L. Viennot, and A. Laouiti, "Multipoint Relaying: An Efficient Technique for flooding in Mobile Wireless Networks," INRIA, Tech. Rep. RR-3898, February 2000. [Online]. Available: http://en.scientificcommons.org/304504

[15] R. Baumann, S. Heimlicher, M. Strasser, and A. Weibel, "A Suvey on Routing Metrics," Computer Engineering and Networks Laboratory, ETH-Zentrum, Switzerland, TIK Report 262, February 2007.

[16] D. S. J. De Couto, D. Aguayo, J. Bicket, and R. Morris, "A High-Throughput Path Metric for Multi-Hop Wireless Routing," in *Proceedings of the 9th ACM International Conference on Mobile Computing and Networking (MobiCom '03)*, San Diego, California, September 2003.

[17] R. Draves, J. Padhye, and B. Zill, "Routing in multi-radio, multi-hop wireless mesh networks," in *MobiCom '04: Proceedings of the 10th annual international conference on Mobile computing and networking*. New York, NY, USA: ACM, 2004, pp. 114–128.

[18] A. P. Subramanian, M. M. Buddhikot, and S. Miller, "Interference aware routing in multi-radio wireless mesh networks," in *Wireless Mesh Networks, 2006. WiMesh 2006. 2nd IEEE Workshop on*, 2006, pp. 55–63. [Online]. Available: http://dx.doi.org/10.1109/WIMESH.2006.288620

[19] R. Dube, C. Rais, K.-Y. Wang, and S. Tripathi, "Signal stability-based adaptive routing (SSA) for ad hoc mobile networks," *Personal Communications, IEEE*, vol. 4, no. 1, pp. 36–45, Feb 1997.

[20] C. Dearlove, T. Clausen, and P. Jacquet, "Link Metrics for OLSRv2," IETF Internet-Draft, MANET WG, draft-dearlove-olsrv2-metrics-04, June 2010, work in progress. Intended status: Informational, Expires: December 17, 2010. [Online]. Available: http://tools.ietf.org/html/draft-dearlove-olsrv2-metrics-05

[21] T. Clausen, C. Dearlove, J. Dean, and the OLSRv2 Design Team, "MANET Neighborhood Discovery Protocol (NHDP)," IETF Internet-Draft, MANET WG, draft-ietf-manet-nhdp-10, July 2009, work in progress. Intended status: Standards Track, Expires: January 14, 2010. [Online]. Available: http://www.ietf.org/internet-drafts/draft-ietf-manet-nhdp-10.txt

[22] IEEE, "Wireless LAN medium access control (MAC) and physical layer (PHY) specification," IEEE standard 802.11, June 1999.

[23] ——, "Wireless LAN medium access control (MAC) and physical layer (PHY) specification," IEEE standard 802.11-2007, June 2007.

[24] S.-Y. Ni, Y.-C. Tseng, Y.-S. Chen, and J.-P. Sheu, "The broadcast storm problem in a mobile ad hoc network," in *MobiCom '99: Proceedings of the 5th annual ACM/IEEE international conference on Mobile computing and networking*. New York, NY, USA: ACM, 1999, pp. 151–162.

[25] M. te Sun, L. Huang, S. Wang, A. Arora, and T.-H. Lai, "Reliable MAC layer multicast in IEEE 802.11 wireless networks," *Wireless Communications and Mobile Computing*, vol. 3, no. 4, pp. 439–453, 2003.

[26] O. S. Badarneh and M. Kadoch, "Multicast Routing Protocols in Mobile Ad Hoc Networks: A Comparative Survey and Taxonomy," *EURASIP Journal on Wireless Communications and Networking*, vol. 2009, p. 42, 2009.

[27] L. Junhai, Y. Danxia, X. Liu, and F. Mingyu, "A survey of multicast routing protocols for mobile Ad-Hoc networks," *Communications Surveys & Tutorials, IEEE*, vol. 11, no. 1, pp. 78–91, Quarter 2009.

[28] P. Mohapatra, C. Gui, and J. Li, "Group Communications in Mobile Ad Hoc Networks," *Computer*, vol. 37, no. 2, pp. 52–59, 2004.

[29] S. Yang and J. Wu, *New technologies of multicasting in MANET*. Nova Science Publishers, 2005, ch. 10, p. 17.

[30] L. Ji and M. S. Corson, "Explicit multicasting for mobile ad hoc networks," *Mob. Netw. Appl.*, vol. 8, no. 5, pp. 535–549, 2003.

[31] H. Gossain, C. Cordeiro, K. Anand, and D. Agrawal, "E2M: a scalable explicit multicast protocol for MANETs," in *Communications, 2004 IEEE International Conference on*, vol. 6, June 2004, pp. 3628–3632 Vol.6.

[32] E. M. Royer and C. E. Perkins, "Multicast Operation of the Ad-Hoc On-Demand Distance Vector Routing Protocol," in *Mobile Computing and Networking*, Seattle, WA, USA, 1999, pp. 207–218. [Online]. Available: citeseer.ist.psu.edu/article/royer99multicast.html

[33] A. Laouiti, P. Jacquet, P. Minet, L. Viennot, T. Clausen, and C. Adjih, "Multicast Optimized Link State Routing," INRIA, Research Report RR-4721, February 2003. [Online]. Available: ftp://ftp.inria.fr/INRIA/publication/publi-pdf/RR/RR-4721.pdf

[34] J. J. Garcia-Luna-Aceves and E. L. Madruga, "The Core-Assisted Mesh Protocol," *IEEE Journal on Selected Areas in Communications*, vol. 17, pp. 1380–1394, 1999.

[35] S.-J. Lee, W. Su, and M. Gerla, "On-Demand Multicast Routing Protocol in Multihop Wireless Mobile Networks," *Mobile Networks and Applications*, vol. 7, no. 6, pp. 441–453, December 2002.

[36] J. Xie, R. R. Talpade, A. Mcauley, and M. Liu, "AMRoute: ad hoc multicast routing protocol," *Mob. Netw. Appl.*, vol. 7, no. 6, pp. 429–439, 2002.

[37] P. Sinha, R. Sivakumar, and V. Bharghavan, "MCEDAR: multicast core-extraction distributed ad hoc routing," in *Wireless Communications and Networking Conference, 1999. WCNC. 1999 IEEE*, 1999, pp. 1313–1317 vol.3.

[38] C.-C. Shen and C. Jaikaeo, "Ad hoc multicast routing algorithm with swarm intelligence," *Mob. Netw. Appl.*, vol. 10, no. 1-2, pp. 47–59, 2005.

[39] B. Williams and T. Camp, "Comparison of broadcasting techniques for mobile ad hoc networks," in *MobiHoc '02: Proceedings of the 3rd ACM international symposium on Mobile ad hoc networking & computing*. New York, NY, USA: ACM, 2002, pp. 194–205.

[40] C. Ho, K. Obraczka, G. Tsudik, and K. Viswanath, "Flooding for reliable multicast in multi-hop ad hoc networks," in *DIALM '99: Proceedings of the 3rd international workshop on Discrete algorithms and methods for mobile computing and communications*. New York, NY, USA: ACM, 1999, pp. 64–71.

[41] D. B. Johnson and D. A. Maltz, *Dynamic Source Routing in Ad Hoc Wireless Networks*, ser. The International Series in Engineering and Computer Science. Kluwer Academic Publishers, 1996, vol. 353, ch. 5, pp. 153–181.

[42] T. Ballardie, P. Francis, and J. Crowcroft, "Core based trees (CBT)," *SIGCOMM Comput. Commun. Rev.*, vol. 23, no. 4, pp. 85–95, 1993.

[43] S. Deering, "Host extensions for IP multicasting," RFC 1112, August 1989.

[44] R. Sivakumar, P. Sinha, and V. Bharghavan, "CEDAR: a core-extraction distributed ad hoc routing algorithm," *Selected Areas in Communications, IEEE Journal on*, vol. 17, no. 8, pp. 1454–1465, Aug 1999.

[45] H. Lim and C. Kim, "Multicast tree construction and flooding in wireless ad hoc networks," in *MSWIM '00: Proceedings of the 3rd ACM international workshop on Modeling, analysis and simulation of wireless and mobile systems*. New York, NY, USA: ACM, 2000, pp. 61–68.

[46] J. P. Macker, J. Dean, and W. Chao, "Simplified multicast forwarding in mobile ad hoc networks," *Military Communications Conference, 2004. MILCOM 2004. IEEE*, vol. 2, pp. 744–750 Vol. 2, 2004. [Online]. Available: http://dx.doi.org/10.1109/MILCOM.2004.1494892

[47] J. Macker, I. Downard, J. Dean, and B. Adamson, "Evaluation of distributed cover set algorithms in mobile ad hoc network for simplified multicast forwarding," *SIGMOBILE Mob. Comput. Commun. Rev.*, vol. 11, no. 3, pp. 1–11, July 2007. [Online]. Available: http://portal.acm.org/citation.cfm?id=1317425.1317426

[48] J. Macker and the SMF Design Team, "Simplified Multicast Forwarding," MANET WG, draft-ietf-manet-smf-09, July 2009, work in progress. Intended status: Experimental, Expires: January 14, 2010. [Online]. Available: http://www.ietf.org/internet-drafts/draft-ietf-manet-smf-09.txt

[49] K. Bur and C. Ersoy, "Multicast routing for ad hoc networks with a quality of service scheme for session efficiency," *Personal, Indoor and Mobile Radio Communications, 2004. PIMRC 2004. 15th IEEE International Symposium on*, vol. 2, pp. 1000–1004 Vol.2, Sept. 2004.

[50] H. Tebbe, A. J. Kassler, and P. M. Ruiz, "QoS-aware mesh construction to enhance multicast routing in mobile ad hoc networks," in *InterSense '06: Proceedings of the first international conference on Integrated internet ad hoc and sensor networks*. New York, NY, USA: ACM, 2006, p. 17.

[51] H. Badis, "A QoS-aware multicast routing protocol for multimedia applications in mobile ad hoc networks," in *MSWiM '08: Proceedings of the 11th international symposium on Modeling, analysis and simulation of wireless and mobile systems*. New York, NY, USA: ACM, 2008, pp. 244–251.

[52] H. Badis and K. A. Agha, "QOLSR, QoS routing for ad hoc wireless networks using OLSR," *European Transactions on Telecommunication*, vol. 16 Issue 5, pp. 427–442, 2005, special Issue: Self-Organisation in Mobile Networking.

[53] J. Heidemann and T. Henderson (Editors). (2009, October) Network Simulator 2. [Online]. Available: http://nsnam.isi.edu/nsnam/

[54] F. J. Ros and P. M. Ruiz. (2009, October) MANET Simulation and Implementation at the University of Murcia (MASIMUM). [Online]. Available: http://masimum.dif.um.es/

[55] S. PalChaudhuri. (2009, October) Ns-2 code for random trip mobility model. [Online]. Available: http://monarch.cs.rice.edu/~santa/research/mobility

[56] T. Camp, J. Boleng, and V. Davies, "A Survey of Mobility Models for Ad Hoc Network Research," *Wireless Communications & Mobile Computing (WCMC): Special issue on Mobile Ad Hoc Networking: Research, Trends and Applications*, vol. 2, no. 5, pp. 483–502, 2002. [Online]. Available: http://citeseer.ist.psu.edu/camp02survey.html

[57] M. Voorhaen and C. Blondia, "Analyzing the Impact of Neighbor Sensing on the Performance of the OLSR protocol," in *Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks, 2006 4th International Symposium on*, April 2006, pp. 1–6.

[58] L. Landmark, K. Øvsthus, and Ø. Kure, "Alternative Packet Forwarding for Otherwise Discarded Packets," in *Future Generation Communication and Networking (FGCN 2007)*, vol. 1, Dec. 2007, pp. 8–15.

[59] C. Gomez, D. Garcia, and J. Paradells, "Improving performance of a real ad-hoc network by tuning OLSR parameters," in *Computers and Communications, 2005. ISCC 2005. Proceedings. 10th IEEE Symposium on*, June 2005, pp. 16–21.

[60] M. Ahmed, S. Krishnamurthy, R. Katz, and S. Dao, "An architecture for providing range extension by deploying mobile gateways in ad hoc networks," in *Personal, Indoor and Mobile Radio Communications, 2002. The 13th IEEE International Symposium on*, vol. 4, 2002, pp. 1660–1664 vol.4. [Online]. Available: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1045461

[61] J. Jun and M. L. Sichitiu, "The nominal capacity of wireless mesh networks," *Wireless Communications, IEEE [see also IEEE Personal Communications]*, vol. 10, no. 5, pp. 8–14, 2003. [Online]. Available: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1241089

[62] F. Li, Y. Wang, and X.-Y. Li, "Gateway Placement for Throughput Optimization in Wireless Mesh Networks," *Communications, 2007. ICC '07. IEEE International Conference on*, pp. 4955–4960, 2007. [Online]. Available: http://dx.doi.org/10.1109/ICC.2007.818

[63] B. Aoun, R. Boutaba, Y. Iraqi, and G. Kenward, "Gateway Placement Optimization in Wireless Mesh Networks With QoS Constraints," *Selected Areas in Communications, IEEE Journal on*, vol. 24, no. 11, pp. 2127–2136, Nov. 2006.

[64] E. L. Kuan, "A study into the practical issues related to a deployed ad hoc wireless sensor network," in *Wireless Communications and Networking Conference, 2005 IEEE*, vol. 4, 2005, pp. 1952–1957 Vol. 4. [Online]. Available: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1424818

[65] S. Crisostomo, S. Sargento, P. Brandao, and R. Prior, "Improving AODV with preemptive local route repair," *International Workshop on Wireless Ad-Hoc Networks*, pp. 223–227, 2004.

[66] W. Su, S.-J. Lee, and M. Gerla, "Mobility prediction and routing in ad hoc wireless networks," *International Journal of Network Management*, vol. 11, no. 1, pp. 3–30, 2001.

[67] T. Goff, N. B. Abu-ghazaleh, D. S. Phatak, and R. Kahvecioglu, "Preemptive Routing in Ad Hoc Networks," *Proceedings ACM/IEEE MobiCom*, pp. 43–52, 2001.

[68] L. Meng, J. Zang, W. Fu, and Z. Xu, "A novel ad hoc routing protocol research based on mobility prediction algorithm," *Proceedings of International Conference on Wireless Communications, Networking and Mobile Computing*, vol. 2, pp. 791–794, Sept. 2005.

[69] G. Chauhan and S. Nandi, "QoS Aware Stable path Routing (QASR) Protocol for MANETs," *First International Conference on Emerging Trends in Engineering and Technology (ICETET)*, pp. 202–207, July 2008.

[70] H. M. Ali, A. M. Naimi, A. Busson, and V. Vèque, "An efficient link management algorithm for high mobility mesh networks," *Proceedings of the 5th ACM international workshop on Mobility management and wireless access (MobiWac)*, pp. 42–49, 2007.

[71] K.-W. Chin, "The Behavior of MANET Routing Protocols in Realistic Environments," *Asia-Pacific Conference on Communications*, pp. 906–910, Oct. 2005.

[72] P. Mohapatra, J. Li, and C. Gui, "Qos in mobile a hoc networks," *Wireless Communications, IEEE*, vol. 10, no. 3, pp. 44–52, June 2003.

[73] A.-H. A. Hashim, M. M. Qabajeh, O. Khalifa, and L. Qabajeh, "Review of Multicast QoS Routing Protocols for Mobile Ad Hoc Networks," *International Journal of Computer Science and Network Security*, vol. 8, no. 12, pp. 108–117, December 2008.

[74] M. Canales, J. Gallego, A. Hernandez-Solana, and A. Valdovinos, "Cross-Layer Routing for QoS Provision in Multiservice Mobile Ad Hoc Networks," *Personal, Indoor and Mobile Radio Communications, 2006 IEEE 17th International Symposium on*, pp. 1–5, Sept. 2006.

[75] G.-S. Ahn, A. Campbell, A. Veres, and L.-H. Sun, "SWAN: service differentiation in stateless wireless ad hoc networks," *INFOCOM 2002. Twenty-First Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, vol. 2, pp. 457–466 vol.2, 2002.

[76] G. Elmasry, C. McCann, and R. Welsh, "Partitioning QoS management for secure tactical wireless ad hoc networks," *Communications Magazine, IEEE*, vol. 43, no. 11, pp. 116–123, Nov. 2005.

[77] S. Xu and T. Saadawi, "Does the IEEE 802.11 MAC protocol work well in multihop wireless ad hoc networks?" *Communications Magazine, IEEE*, vol. 39, no. 6, pp. 130–137, Jun 2001.

[78] P. Jacquet, A. Laouiti, P. Minet, and L. Viennot, "Performance analysis of OLSR Multipoint Relay flooding in two ad-hoc wireless network models," INRIA, Research report 4260, September 2001. [Online]. Available: http://www.inria.fr/rrrt/rr-4260.html

[79] A. Busson, N. Mitton, and E. Fleury, "An analysis of the Multi-Point Relays selection in OLSR," INRIA, Research report 5468, January 2005. [Online]. Available: http://www.lri.fr/~fragile/IMG/pdf/RR-5468_analyseMPR.pdf

[80] M. Benzaid, P. Minet, and K. Al Agha, "Analysis and simulation of Fast-OLSR," *The 57th IEEE Semiannual Vehicular Technology Conference (VTC)*, vol. 3, pp. 1788–1792, April 2003.

[81] L. Qin and T. Kunz, "Mobility Metrics to Enable Adaptive Routing in MANET," *IEEE International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)*, pp. 1–8, June 2006.

[82] ——, "Adaptive MANET Routing: A Case Study," in *ADHOC-NOW '08: Proceedings of the 7th international conference on Ad-hoc, Mobile and Wireless Networks*. Berlin, Heidelberg: Springer-Verlag, 2008, pp. 43–57.

[83] S. Cho and C. Adjih, "Optimized multicast based on multipoint relaying," in *Wireless Internet, 2005. Proceedings. First International Conference on*, July 2005, pp. 42–46.

[84] S. Shioda, K. Ohtsuka, and T. Sato, "An efficient network-wide broadcasting based on hop-limited shortest-path trees," *Computer Networks*, vol. 52, no. 17, pp. 3284 – 3295, 2008. [Online]. Available: http://www.sciencedirect.com/science/article/B6VRG-4T9VPCY-1/2/6d0aecb50001527eb2f9caa74f9dbd85

Part II

Research papers

Paper A :

Rerouting Time and Queueing in Proactive Ad Hoc Networks

V. Pham, E. Larsen, K. Øvsthus, P. Engelstad, and Ø. Kure

In proceedings of the Performance, Computing, and Communications Conference (IPCCC), New Orleans, USA, April 11–13, 2007, pp. 160–169.

Rerouting Time and Queueing in Proactive Ad Hoc Networks

Vinh Pham[1], Erlend Larsen[1], Knut Øvsthus[2], Paal Engelstad[1] and Øivind Kure[3]
[1]UniK, Norway [2]Bergen University College, Norway [3]Q2S, NTNU, Norway
E-mail: [1,3]{vph, erl, paalee, okure}@unik.no [2]knut.ovsthus@hib.no

Abstract

In a MANET network where nodes move frequently, the probability of connectivity loss between nodes might be high, and communication sessions may easily loose connectivity during transmission. The routing protocol is designed to find alternative paths in these situations. This rerouting takes time, and the latency is referred to as the rerouting time. This paper investigates the rerouting time of proactive routing protocols and shows that the rerouting time is considerably affected by queueing. Simulations and analysis are conducted to explore the problem. Finally, we propose a MAC-layer solution that reduces the rerouting time problems due to queueing. Simulations and analysis show that the solution is so effective that it eliminates the entire problem in many situations.

1. Introduction

The research efforts in the field of ad hoc networking have been going on for many decades. Ad hoc networking enables communication directly between nodes, without the need for extra infrastructure. This makes it very suitable for military and rescue operations. The standardization of routing protocols has been undertaken by the Mobile Ad Hoc Networking (MANET) working group in IETF [1]. They are set to bring forward two protocols, one reactive and one proactive.

A common characteristic of ad hoc networks is that links may break due to changes in radio conditions, node mobility and other types of network dynamics. The routing protocol is designed to find alternative paths in these situations. The time period before new paths are found is referred to as the rerouting interval, and the duration of the rerouting interval is referred to as the rerouting time.

During the rerouting interval, stale routes exist over the link that has been broken. Rerouting can only take place after the routing protocol has detected that the link is broken. In fact, a significant part of the rerouting time is associated with the detection of the link break.

With proactive routing protocols, such as Optimized Link State Routing (OLSR) and Open Shortest Path First with MANET Designated Routers (OSPF-MDR), a link is maintained by the exchange of control packets. A link break is normally not detected until either a certain number of HELLO packets have been lost, or the lack of periodic updates results in a link timeout [2-4]. (Some implementations might let the link layer detect link breaks and signal this information to the routing protocol. Such cross-layer optimizations are outside scope of this paper. Here, we explore the common layered approach where HELLO packets are necessary for the detection of link breaks.)

With the default parameter settings of OLSR and OSPF-MDR, a link break should normally be detected after approximately 6 seconds. However, we conducted a series of lab experiments of OLSR [3] and OSPF-MDR [4] and observed rerouting times typically in the order of 20 - 40 seconds. Since the rerouting time depended on transmission rates of data traffic and on size of the transmission queues, we realized that the increased rerouting time in our experiment was mainly caused by the queueing of the data packets.

During the rerouting interval, the network layer at the node upstream to the broken link might try to forward data packets over the broken link. Instead, these packets are accumulated in the output queue. Due to the layered design, the link layer (L2) will keep trying to transmit the queued data traffic already designated to the broken link, even after the network layer (L3) has timed out the link. This does not only consume scarce radio resources. It also blocks the MAC layer. Thus the network layer is not able to announce that the link is broken, and the rerouting time increases correspondingly.

Finally, when all the stale data packets designated to the output queue have been dropped, the MAC layer is ready to transmit the link state announcement to establish new routes throughout the network and to

serve packets waiting in the output queue designated to reachable receivers.

In summary, the rerouting time due to link breaks depends on the time to carry out the following processes:

- Detection of a link break
- The emptying of all stale packets from the output queue
- Network-wide link-state announcement to establish new paths

While both link break detection and routing convergence have received considerable attention in the research community, surprisingly little focus has been directed to the effects of queueing. Indeed, the main contribution of this paper is to explore how queueing increases the rerouting time.

The rest of the paper is organized as follows. Section 2 gives background information on relevant technologies. In Section 3 we present the simulation setup, define the rerouting time, and show simulation results. Section 4 gives an analysis of the factors contributing to the rerouting time. Section 5 presents a proposed solution to the rerouting problem and in Section 6 we present some related work. Finally, in Section 7 the conclusion is presented and further work is sketched out.

2. Background

2.1. The MAC layer of IEEE 802.11

Today, IEEE 802.11 [5] is the most widely used wireless local area networking technology. The standard defines a Physical (PHY) layer and a Medium Access Control (MAC) sub-layer, where the latter supports two modes of operation, namely the Distributed Coordination Function (DCF) and the Point Coordination Function (PCF). Since DCF is the most common mode of operation, we focus only on DCF in this paper.

With DCF, the wireless stations (STAs) access the medium in a distributed way, using *carrier sense multiple access with collision avoidance* (CSMA/CA). With the basic access mechanism, each unicast DATA frame is acknowledged with an ACK frame. This is also known as the minimal frame exchange. (Multicast transmissions, however, are not followed by an ACK frame.) With the optional 4-way frame exchange, on the contrary, each DATA frame is preceded by an exchange of a *request to send (RTS) and a clear to send* (CTS) frame. The use of RTS/CTS is particularly useful to avoid collisions due to hidden terminals [6].

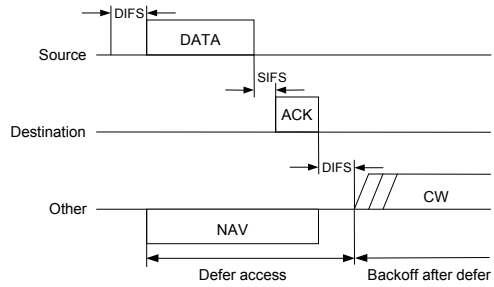


Fig. 1. Basic CSMA/CA.

The basic access mechanism with the minimal frame exchange is illustrated in Fig. 1. When a node has some data to transmit, it has to sense the medium to verify whether it is busy or idle. If the channel is idle for a time interval equal to a Distributed Inter-Frame Space (DIFS), the source node may begin data transmission. When the receiver has received the DATA frame, it waits for a time interval equal to a Short Inter-Frame Space (SIFS), and transmits an ACK back to the source node. While data is being transmitted, all other nodes must defer their channel access for a time interval equal to the Network Allocation Vector (NAV). This is a timer indicating the amount of time that the medium has been reserved for the current transmission. When the data transmission is finished and the NAV has expired, a new contention period is entered. Here, concurrent nodes with pending data traffic must contend for the medium. In this process, each contending node must choose a random time interval called *Backoff_timer*, selected from the contention window (CW) in the following way:

$$Backoff\_timer = rand[0, CW] \cdot slottime$$

where

$$CW = [CW_{\min}, CW_{\max}]$$

The value for the *slottime* is dependent on the PHY layer type. The backoff timer is decremented only after each time the medium is idle for a DIFS interval, and is frozen when the medium becomes busy. Eventually when the backoff timer of a node expires, it might transmit data. The main point of this medium access mechanism is to minimize the probability of a collision, i.e. of concurrent transmissions. Since a node must go through a backoff after having transmitted a frame (also referred to as a *post-backoff*), the medium access

mechanism also provides long term fairness to access the medium.

In a wireless environment where collision detection is hard or even impossible, a positive ACK from the receiver is used to confirm a successful transmission. The absence of such an ACK message indicates a collision, link failure or other reasons for an unsuccessful transmission. When this occurs, a retransmission is scheduled, and a new backoff value is chosen. However, in order to reduce the risk for consecutive collisions, after each unsuccessful transmission attempt, the CW is doubled until a predefined CW_{max} is reached.

There is a retry counter associated with the transmission of each frame, and the retry counter is incremented after each collision. After a successful retransmission, the CW is again reset to a predefined CW_{min}, and the retry counter is reset to null.

The maximum number of retransmissions for a frame is defined in the *dot11ShortRetryLimit* and *dot11LongRetryLimit* variables. The first variable is applicable for MAC frames transmitted with the minimal frame exchange (i.e. with length less than or equal to the *dot11RTSThreshold* parameter), while the latter is applicable to frames transmitted with RTS/CTS. For instance, each time a MAC frame of length less than or equal to the *dot11RTSThreshold* is transmitted, and it fails, the *short retry counter* is incremented. This will continue until there is a successful transmission or the counter has reached the dot11ShortRetryLimit and the packet is discarded. When this happens the short retry counter is reset to zero.

For simplicity, throughout the rest of this paper, we will use the term dot11ShortRetryLimit and "retry limit" interchangeably.

2.2. Queueing in the protocol stack

The unicast packets (multicast is considered out of scope) created by applications are passed down the protocol stack to TCP or UDP using the socket interface (Fig. 2). If the packet is a TCP packet, it may be queued to accommodate flow control. For UDP, and TCP eventually, the packet is passed down to L3 (i.e. the IP layer) for routing and designation of a next hop link layer address before passed down to the L2 (i.e. the link layer). There it is queued in the queue of the device driver until the buffer of the network interface is empty, and is then pulled onto the network interface. When the transmission medium is available, the packet is transmitted. If no ACK is received, the packet is assumed lost due to a collision, and the packet will be scheduled for retransmission.

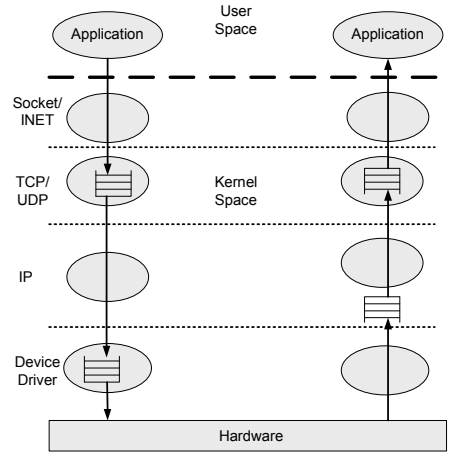


Fig. 2. Linux protocol stack [7].

When a packet is received at an interface, it is put in a backlog queue. Then L3 processes it, and either forwards it out on an interface or pushes it up the stack to UDP or TCP. TCP has a receive queue to serve flow control.

The L2 queue should be of a minimum size to allow traffic to be sent without loss from applications at a rate higher than the network capacity, as the network bandwidth can be variable due to fading, mobility, interference, contention etc.

Both Linux and the network simulator *ns-2* [8] implement a L2 queue for outgoing packets. In ns-2, using the CMU Monarchs wireless extensions, packets are queued in the interface priority queue (IFq). The network stack for a mobile node consists of a link layer (*LL*), an ARP module connected to LL, an interface priority queue, a MAC layer, a network interface, all connected to the channel. When a packet is created by the source application, the packet is queued in the IFq until all previous packets have been either sent or discarded.

2.3. Optimized Link State Routing Protocol (OLSR)

OLSR is a proactive routing protocol for ad hoc networks. The protocol is built around the notion of Multi Point Relay nodes (MPRs). The main purpose of MPRs is to create and forward link state messages. The MPRs are selected individually by each node in the

network in such a way that all nodes can reach their 2-hop neighbor nodes through an MPR.

The two most important message types in OLSR are the HELLO and the TC (Topology Control) messages:

1) HELLO Messages: Every node broadcasts HELLO messages periodically, to support link sensing, detection of neighbors and signaling of MPR selection. The recommended emission interval for HELLO messages is 2 seconds, and the holding time for neighbor information is 6 seconds. Thus a neighbor is considered lost 6 seconds after the last HELLO message received from the neighbor.

2) TC Messages: Based on the information collected through HELLO messages, link state (TC) messages are created and broadcasted throughout the network by each MPR. The recommended emission interval for TC messages is 5 seconds, and the holding time is 15 seconds.

3. Simulations

3.1. Description of the scenario

The scenario explored can be described as follows: Three nodes A, B and C form an ad hoc network where A sends traffic to C at a Constant Bit Rate (CBR). At the beginning, A and B stretch out the network. Then C moves past B, and loses connectivity with A until traffic from A is rerouted via B. In this scenario, C has always direct connectivity with B.

Although the scenario seems simple, it is realistic and sufficient to explore important aspects of the rerouting time. Note also that all nodes are within a two-hop distance of each other. This means that the dissemination of TC messages will not affect the rerouting time, and we are able to explore the rerouting time associated only with the detection of the link break and with the queueing effects.

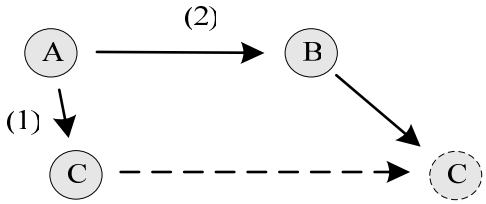


Fig. 4. Conceptual model for the simulations.

3.2. Simulation setup

All simulations were carried out using the network simulator ns-2. The setup for our test scenario (Fig. 4) is equivalent to the scenario in Fig. 3. In the beginning, all three nodes A, B and C are in the immediate neighborhood of each other. Node A, which is the sender, sends UDP data packets of rate R_{in} packets per second directly to the receiver node C. (This flow is marked with "(1)" in Fig. 4.). While the data transmission is ongoing, node C moves away from node A. At a certain point where node A and C are no longer in the immediate neighborhood of each other, the connection between these two nodes is broken. In order to re-establish connectivity between node A and node C, node A has to reroute the traffic through node B. (This flow is marked with "(2)" in Fig. 4.). B forwards this traffic further on to node C.

In all simulations, the packet size was fixed at 1000 bytes. IEEE 802.11b [9] was used with the basic DCF mechanism (i.e. without RTS/CTS) and a nominal transmission rate of 11 Mbps. The RTS/CTS handshake mechanism is not necessary since there is no hidden node problem in our scenario. All nodes are inside each others sensing range.

Table 1. Simulation parameter settings.

Simulator	ns-2 version 2.30
Radio-propagation model	TwoRayGround
MAC type	802.11b
Interface queue type	FIFO with DropTail
Antenna model	OmniAntenna
Data rate	11 Mbps
Basic rate	1 Mbps
Packet Size IP	1000 Bytes
Movement speed of node C	3.3 m/s
OLSR HELLO_INTERVAL	2 seconds
OLSR REFRESH_INTERVAL	2 seconds
OLSR TC_INTERVAL	5 seconds
OLSR NEIGHB_HOLD_TIME	6 seconds
OLSR TOP_HOLD_TIME	15 seconds
OLSR DUP_HOLD_TIME	30 seconds

The implementation of OLSR by the University of Murcia was used as the proactive routing protocol for ns-2 [10]. In the OLSR configuration, the time interval between HELLO packets was set to 2 seconds, and the HELLO packets were given priority over data packets to avoid route instability. Furthermore, a link is considered down after the loss of 3 consecutive HELLO packets, leading to a detection time of link breaks of approximately 6 seconds:

$$HELLO\_INTERVAL = 2\text{seconds}$$
$$NEIGHB\_HOLD\_TIME = 3 \cdot HELLO\_INTERVAL$$

Essential parameters used in the simulations setup are summarized in Table 1.

3.3. Definition of the rerouting time

In the simulations that were conducted, we mainly focused on measuring the rerouting time, i.e. the time duration from when the link between A and C is broken to the time when connectivity is re-established via the intermediate node B. However, our experience through many experiments - both in a real test-bed and in simulations - is that the rerouting time measured in this way will have a high degree of variance caused by random effects during rerouting. In order to minimize variance in the measurements, we have chosen to define the rerouting time $t_{reroute}$ as the time interval from the last HELLO message from node C received by node A before link break, to the moment where the connectivity is re-established, i.e. until the instant of time where the first UDP packet is received at C after the link break.

3.4. Simulation results

The results from the simulations for various retry limits (Fig. 5) show that a higher retry value gives a longer rerouting time. This is as expected, because each packet in the L2 queue is transmitted a number of times defined by this retry value. We also notice that the rerouting time is linearly proportional with the L2 queue size.

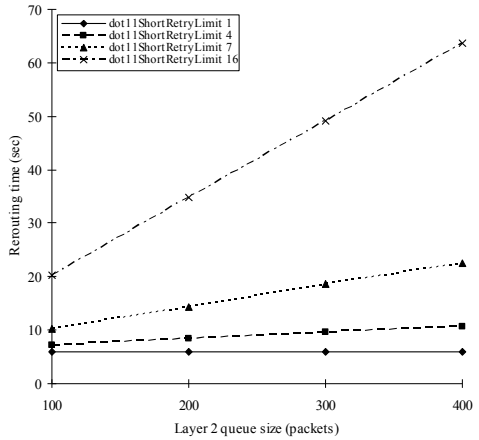


Fig. 5. Simulation results of rerouting time over layer 2 queue size.

Fig. 6 shows simulation results for the rerouting time as a function of the transmitted packet rate (marked as crosses, squares and triangles). Here, the retry limit is set to 7, and the queue size is set to 100, 400 and 1000 packets. The figure shows that for small packet rates, the rerouting time is at the minimum value of 6 seconds, which equals to the NEIGHB_HOLD_TIME. As the packet rate increases, the rerouting time also increases linearly up to a certain point where it suddenly stops to increase, and the rerouting time stabilizes at its maximum value. The maximum rerouting time depends on the queue size. For a queue size of 100, the maximum rerouting time is slightly more than 10 seconds. For a queue size of 400 it is nearly 23 seconds, while for a queue size of 1000 the maximum rerouting time lies around 47 seconds. With a queue size of 400, we see that at packet rates of 100 pkts/sec and over, the queue is filled at the time when rerouting takes place, and this results in a rerouting time converging on approximately 23 seconds.

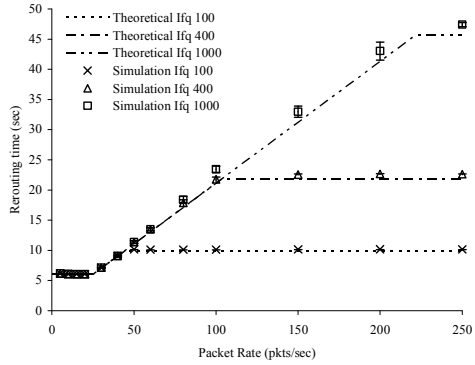


Fig. 6. Simulation results for rerouting time over packet rate for layer 2 queue size 100, 400 and 1000 packets, 7 MAC retries. (95% conf. int.)

It is also observed (Fig. 6) that at low rates (i.e. well below 20 pkts/sec) the rerouting time is flat at 6 seconds.

4. Analysis

4.1. Analysis of the problem

From the log file produced by ns-2 we can observe various incidents affecting the rerouting time. These incidents, which occur at node A, are illustrated in Fig. 7 and are explained below:

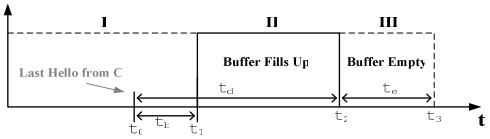


Fig. 7. Illustrating different incidents at node A's queue as function of time.

1) In region I, data packets are continuously inserted into the transmit queue at node A. At time t_0, the last HELLO message from C is received at A (short line in the figure). Then, after t_b seconds, the direct link between A and C is broken at t_1.

2) Although the link between A and C is broken, the routing protocol is still not aware of this, and therefore has not updated the routing table. As a result, "garbage" data packets with stale routing information continue to be put into the queue at node A.

3) In region II, i.e. $t_1 < t < t_2$, the queue at node A is being filled up. This happens since each garbage data packet in the queue at node A is retransmitted L times, where L is the retry limit. Because of all the retransmissions for each packet, the packet rate out of queue R_{out} will be reduced considerably. As long as the packet rate R_{in} into the queue is higher than R_{out}, the queue will be filled up. This will last for $t_d - t_b$ seconds, where t_d is the timeout value for the routing protocol's HELLO packets (which is equivalent with *NEIGHB_HOLD_TIME* in OLSR).

4) At t_2, garbage packets are no longer put into the queue. The routing protocol has now updated the routing tables. New data packets are instead correctly rerouted to B.

5) In region III, i.e. $t_2 < t < t_3$, the queue is being emptied for garbage packets. This will last for t_e seconds, depending on parameter values like R_{in}, L, packet size, queue size etc.

Note that packets are attempted transmitted and removed from the queue both in region II and region III. Thus, the queue will fill up in region II only if $R_{in} > R_{out}$. However, for the lowest packet rates, we will have $R_{in} < R_{out}$, and the queue will not be filled. In the latter case, both region II and region III will be non-existent, and the routing interval consists of only region I. This explains why the rerouting time is flat at 6 seconds for the lowest packet rates in Fig. 6.

In summary, the incidents in the time interval $t_0 < t < t_3$ are the main contributions to the rerouting time as defined above. It is also worth noting that in our test scenario, the delay from a TC message is irrelevant for the rerouting time. This is due to the fact that prior to the link break, node A will have node C in both its 1-hop and 2-hop neighbor sets. When A discovers a link break between A and C, A still has a route to C in its 2-hop neighbor table, i.e. through node B. Therefore, node A does not need to wait for any TC message from B in order to figure out how to reach C.

4.2. A model for the rerouting time

Based on the observations above we have derived a simple analytical model that can be used to predict the rerouting time. The derivation of the model is given below:

t_{difs}	DCF interframe space
t_{bo}	backoff time
t_{data}	delay for transmitting the data packet
t_{sifs}	short interframe space
t_{ack}	delay of acknowledge
T_e	slot time in IEEE 802.11
t_{RTS}	delay of a RTS packet
t_{CTS}	delay of a CTS packet
L	number of retries
B	queue size

1) According to the 802.11 standard, the delay of attempting to transmit a single packet over a broken link is

$$t_{packet} = T_c + t_{bo} \tag{1}$$

where T_c is the delay associated with the transmission attempt, and t_{bo} is the delay associated with the backoff. In our scenario, no ACK is received when the link is broken, and the transmission attempt is therefore perceived as a collision. However, according to the standard, a node must wait an ACKTimeout amount of time without receiving an ACK frame before concluding that the transmission failed. In our case, this ACKTimeout corresponds to the transmission of an ACK for a successfully transmitted frame. Thus, the delay associated with the transmission attempt, T_c, is equal to the delay associated with a successful transmission, T_s:

$$T_c = T_s = t_{difs} + t_{data} + t_{sifs} + t_{ack}. \tag{2a}$$

With the RTS/CTS mechanism, on the contrary:

$$T_c = t_{difs} + t_{RTS} + t_{CTS\_timout}. \tag{2b}$$

Prior to each packet transmission, a backoff time is uniformly chosen in the range (0, W_j -1). Here we

define W_j, where $j \in (0, m)$, as the contention window at "backoff stage" j, and m is the number of the maximum backoff stage. Let us also define L as the number of retries, and we can thus write the definition of the contention window as:

$$W_j = \begin{cases} 2^j W_0 & L \le m \\ 2^m W_0 & L > m \end{cases}. \quad (3)$$

Eq. (3) states that for the first transmission attempt, the contention window is W_0 which is equal to CW_{min}. (Note that this definition of the contention window is slightly different from the definition in Section 2. In fact, the IEEE 802.11 standard refers to W_j-1 as the contention window [5]. For convenience, we have defined the contention window differently in this paper.) After each unsuccessful transmission, the contention window is doubled, and the packet is attempt retransmitted. This will continue until we reach the maximum contention window $W_m = 2^m W_0 = CW_{max}$, where it remains for consecutive retransmission attempts. If a retransmission is successful after a number of retries, or the number of retransmission has reached the retry limit, the contention window is again reset to its initial backoff stage W_0.

In our scenario, when the link between A and C is broken, each "garbage" packet in the queue is retransmitted L times, and eventually is discarded because the maximum number of retries has reached.

The mean total delay for one single packet with L retries is then approximately:

$$t_{packet\_L} = (L+1) \cdot T_c + \sum t_{bo} \quad (4)$$

where

$$\sum t_{bo} = T_e \cdot \begin{cases} (\frac{W_0}{2}) \cdot \left[2^{L+1} - 1\right] - \frac{1}{2}(L+1) & L \le m \\ (\frac{W_0}{2}) \cdot \left[2^m \cdot (L - m + 2) - 1\right] - \frac{1}{2}(L+1) & L > m \end{cases} \quad (5)$$

which is the sum of the approximate mean backoff time. Here, T_e is the slot time. Note that T_e, W_0 and m are parameters that depend on the PHY-layer used. For 802.11b, $T_e = 20$ µs, $W_0 = 32$ and $m = 5$.

We have intentionally tried to keep the scenario as simple as possible, to derive a simplified model that is intuitive and easy to analyze. One of the simplifications made is the assumption that A is the only node trying to access the medium when the link is broken. Thus, during backoff the medium is always idle, and the duration of each backoff state is therefore T_e. It is not difficult to extend our analysis for the case when multiple nodes contend for the same medium. In [11], for example, Engelstad and Østerbø calculated the queueing delay by applying a Bianchi model that is extended to non-saturation conditions. Thus, extending our analysis is not hard to do, but draws attention away from the main objective of this paper. It is also considered out of scope due to space limitations, but might be addressed in a follow-on publication.

2) The packet rate R_{out} out of queue when each packet has to be retransmitted L times, is therefore:

$$R_{out} = \min\left[\frac{1}{t_{packet\_L}}, R_{in}\right] = \min\left[\frac{1}{(L+1) \cdot T_c + \sum t_{bo}}, R_{in}\right]. \quad (6)$$

3) The total rerouting time is:

$$t_{rerouting} = t_d + t_e, \quad (7)$$

where (depicted in Fig. 7):

$$t_e = \frac{1}{R_{out}} \cdot \min\left[(t_d - t_b) \cdot (R_{in} - R_{out}), B\right]. \quad (8)$$

4.3. Discussion

Eq. (7) equals the rerouting time as defined above, where only the most significant mechanisms contributing to the total delay of the rerouting time is considered. This delay is equal to $t_3 - t_0$ in Fig. 7. Here, we assume that the delay of transmitting one single packet through the alternative path, from A to B and then to C, is very small compared to t_d and t_e. This delay is therefore omitted in the equation.

The first term of the equation is a constant defined by the proactive ad hoc routing protocol configuration (this is equivalent to the NEIGHB_HOLD_TIME in OLSR). This value is also the absolute minimum rerouting time. The second term is variable, depending on parameters like R_{in}, the retry limit L, the queue size B, etc.

From Eq. (7) and Eq. (8) it is clear that there is a lower and an upper limit on the rerouting time. The lower limit occurs when $R_{in} = R_{out}$, in Eq. (8). Thus, for the lowest packet rates the rerouting time is equal to t_d, as we also observed in the simulations.

The upper limit occurs when the queue is filled and is constrained by the queue size B. Hence, for the highest packet rates (i.e. when $R_{in} > B/(t_d - t_b) + R_{out}$) the maximum rerouting time is:

$$t_{rerouting\_max} = t_d + \frac{B}{R_{out}}. \quad (9)$$

Furthermore, in the case when the rerouting time is larger than t_d and smaller than $t_{rerouting\_max}$, Eq. (7) yields:

$$t_{rerouting} = \frac{R_{in}}{R_{out}} \cdot (t_d - t_b) + t_b. \quad (10)$$

This reveals that the rerouting time is linear and proportional to R_{in} in this region.

Table 2. Comparison of the delay components of R_{out} and the resulting value for R_{out}. Values are given in milliseconds for the delay terms. R_{out} is given as packets per second.

L	0	1	2	3	4	5	6	7
$\sum t_{bo}$	0.31	0.94	2.21	4.76	9.87	20.1	30.3	40.56
$(L+1)T_s$	1.32	2.65	3.97	5.3	6.62	7.94	9.27	10.59
R_{out}	100	100	100	99.44	60.64	35.66	25.25	19.55

The packet rate out of the transmit queue R_{out} is also an important parameter for the rerouting time. A decreasing R_{out} means an increasing rerouting time. By inspecting Eq. (6), we see that the first term in the denominator is linearly proportional with L, while the second term is increasing exponentially with L [Eq. (5)]. This means that the second term will grow much faster than the first term, and therefore will be the dominating term when L is large. This is illustrated in Table 2 where only the results for the eight first retry values were calculated. Here, a packet size of 1000 bytes with a transmission rate of 11 Mbps was used to calculate the delay of t_{data} (in T_c) in Eq. (6). The rate in R_{in} was set to 100 packets per second.

The results from Table 2 show that for the given setting, R_{out} is rapidly decreasing for retry values above 4.

A plot of the estimated rerouting times based on Eq. (7) is shown for three different queue sizes (100, 400 and 1000 packets) as dashed curves in Fig. 6. The curves were calculated using a value of t_b = 0.9 seconds, which corresponds to the average t_b value observed in the simulation results shown in the figure. As the result shows, the estimated rerouting times are almost equal to the simulated results obtained from ns-2. This verifies that the derived formula is a good approximation for the expected rerouting time in the given scenario. We observe, however, that the simulations give a slightly higher rerouting time. This can be explained by the ARP request burst triggered by all packets sent to the Layer 2 in the time lapse from the route through B is chosen, until node B's MAC address is obtained. This behavior of ns-2 is a violation of the recommendations given in [12]. The ARP storm problem is bigger for higher packet rates and larger queue sizes, which can be observed in the figure.

5. Proposed solution

5.1. Adaptive retry limit

At the time the routing protocol becomes aware that the direct connection to the destination has been broken, the packets in the L2 queue no longer have a reachable link layer destination. These packets will be discarded only after being transmitted onto the medium for a number of times defined by the IEEE 802.11 dot11ShortRetryLimit. We argue that a solution to this problem should be implemented as a layered solution, to keep it as small and simple as possible. The link layer protocol will be able to detect the link break earlier than the routing protocol, so it is natural to implement a solution at the link layer. Our analysis shows that at the link layer it is the queue size and the retry limit that are the main contributors to the extended rerouting time. Reducing the queue size could be an option, but to have any effect, this reduction would have to be initiated as soon as the queue usage starts to grow. In this case it would be more efficient to keep the queue small at all times, instead of varying it, but this would restrain the flexibility of having a large queue.

Instead, we propose a solution to the accumulated queue time problem by introducing an adaptive retry limit into the IEEE 802.11 DCF MAC. For each successive packet with the same destination MAC address that is discarded due to reaching the retry limit, the retry limit is reduced by 1, until each packet is only attempted transmitted 1 time. If the original retry limit is 7, the retry limit is reduced to 0 after 7 consecutive packets are dropped due to reaching the retry limit. As soon as a packet is transmitted successfully, the retry limit is reset to its original value equal to that of the legacy IEEE 802.11 standard.

5.2. Discussion

It is very rare that many retry counter expirations occur directly following each other, unless something is wrong. To lower the retry limit gradually will probably not affect the functionality of the 802.11 MAC under normal network conditions. However, a problem with the adaptive retry limit solution is that it might lead to

an unfair resource distribution in terms of collision avoidance. The node sending packets that go unacknowledged will be able to contend for the medium with a high probability of a smaller backoff-counter than its peers. On the other hand, the emptying of stale packets from the queue takes place in a small period of time, and it is much more efficient to send a garbage packet only one time, than sending it multiple times. Another drawback is that the transmission attempts of garbage packets consume network resources. More complex solutions where the MAC layer discards packets without attempting to transmit them is certainly also possible. In summary, there are a number of variations of the proposed adaptive retry limit solution. The performance of a number of these variations in various networking scenarios will be detailed and discussed in a follow-on publication.

5.3. Implementation

To do the actual implementation in ns-2 we needed to introduce two new variables. The first of these new variables, *IDt,* keeps track of the destination of the last transmission attempt, and the second variable, called *PCnt*, counts the number of packets discarded because the retry counter has reached the retry limit.

Each time a packet is discarded because the retry counter has reached the retry limit, the *PCnt* is increased, until it reaches the value of the retry limit.

The *PCnt* is subtracted from the original retry limit, so that the effective retry limit gets lower and lower as the *PCnt* increases, until new packets are only transmitted once, and then discarded if not acknowledged by the receiving node.

If a packet is transmitted to a new destination, *PCnt* is set to 0 and *IDt* is updated. If the transmission was successful (indicated by a received ACK), the *PCnt* is set to 0.

5.4. Simulation results

In the simulation results of the adaptive retry limit solution (Fig. 8, with L2 queue size 400 packets and 7 MAC retries) we observe that with the proposed solution the rerouting time is kept at 6 seconds (which equals to NEIGHB_HOLD_TIME) until the packet rate exceeds 600 pkts/sec. At this packet rate the bit rate approaches the theoretical maximum throughput (TMT) of 5.03 Mbps (for 1000 bytes sized packets, and for a network with one sender, where backoff time has to be taken into account). When the packet rate is higher than TMT, the L2 queue gets filled also when the link between A and C is not broken. This is because R_{out} is smaller than R_{in} at all times.

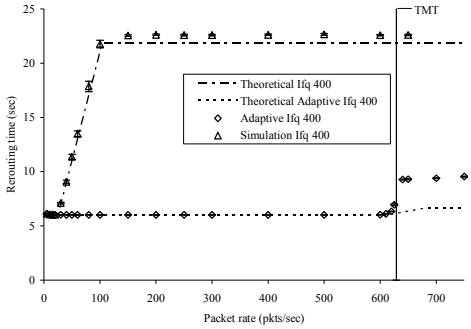


Fig. 8. Simulated results for solution with adaptive retry limit. (95% conf. int.)

Our solution prevents R_{out} from decreasing when the link between A and C is broken, by limiting the accumulated number of retransmissions. The results from the simulations have proven that the adaptive retry limit solution effectively eliminates the delay related to the queueing problem.

The proposed solution is a layered approach based at the link layer, but in some cases it would be more convenient to solve the problem at the IP-layer. This is left for further work.

6. Related work

An analysis of several neighbor sensing approaches is presented in [13]. The objective is to better be able to optimize performance in an OLSR network.

In [14], the OLSR routing protocol is evaluated through both simulations and experiments. Both route-flapping and control packet collisions are described, and solutions for these problems are proposed.

The tuning of routing protocol parameters in order to improve the end-to-end connectivity is studied in [15]. A performance metric called Routing Change Latency (RCL) is defined and analyzed. This metric is defined as “the time needed to determine a new route after a link failure”, but it also comprises a time lapse after the new route is discovered, until it is actually used. This time lapse is denoted as T_{new_route}. It is not explained, but observed to vary between 4.62 s and 8.86 s

7. Conclusions and further work

The rerouting time is an important performance measure in MANETs where node mobility is usually high, and connectivity between nodes may be disrupted frequently. For ongoing data traffic that suffers from link failures, it is highly desirable to reestablish connectivity through alternative paths as fast as possible. In this paper we have looked closer on a simple scenario where we have identified that queueing is among the main factors having considerable impact on the rerouting time.

The latency related to queueing is mainly affected by two parameters, namely the transmit queue size and the retry limit. A large transmit queue size may result in a too high amount of garbage packets with stale routing information being inserted into it. In addition, a high retry value may result in too many wasted retransmission attempts for these garbage packets. The combination of these factors might extend the rerouting time considerably.

We have derived a simple model that can be used to estimate the rerouting time. Comparisons of the estimated and simulated rerouting times have shown that the model is a good approximation. The analysis is used to explain how queueing might increase the rerouting time. In order to solve this problem, we have proposed a simple but very effective solution based on adaptive retry limit in the 802.11 DCF MAC. The queueing problem is resolved by decrementing the maximum retry value when successive packets for the same MAC destination are discarded due to expiration of the retry limit. The proposed solution was implemented and tested in simulations, and the results have shown how effective it can be. In fact, as long as the data rate into the queue is safely below the capacity of the MAC, the solution eliminates the queueing problem associated with the rerouting time.

Although the proposed solution seems to be very effective, there might be some problems associated with it. For example, the solution might lead to an unfair resource distribution in terms of collision avoidance. This needs to be explored in detail, and will be addressed by a follow-on publication.

It might also be possible to implement more complex solutions where the MAC layer discards packets without attempting to transmit them. Various variations of our solution will also be studied.

The proposed solution is a simple way to resolve queueing related delays. We believe there are other possibilities in solving the problem or improving the existing solution. A solution based on cross-layering, where L2 can send a notification up to L3, helping the routing protocol to detect link breaks much earlier is an exciting area. All this is also left to future works.

8. Acknowledgment

This work was supported by the ITEA Easy Wireless and CELTIC DeHiGate projects. We also thank the FFI project Tipper for their support.

References

[1] IETF working group Mobile Ad-hoc Networks, http://www.ietf.org/html.charters/manet-charter.html
[2] T. Clausen, P. Jacquet, “Optimized Link State Routing Protocol (OLSR)”, RFC 3626, October 2003.
[3] http://www.olsr.org/
[4] IETF MANET OSPF design team repository, http://hipserver.mct.phantomworks.org/ietf/ospf/
[5] ANSI/IEEE Std 802.11, 1999 Edition (R2003).
[6] F. Tobagi and L. Kleinrock, “Packet Switching in Radio Channels: Part 2-The Hidden Node Problem in Carrier Sense Multiple Access Modes and the Busy Tone Solution”, IEEE Trans. Comm., vol. 23, no. 12, pp. 1417-1433, 1975.
[7] G. Chuanxiong, Z. Shaoren, “Analysis and evaluation of the TCP/IP protocol stack of LINUX”, International Conference on Communication Technology Proceedings, 2000. WCC - ICCT 2000, Vol. 1 (2000), pp. 444-453 vol.1.
[8] Network Simulator ns-2, http://www.isi.edu/nsnam/ns/
[9] ANSI/IEEE Std 802.11b, 1999 Edition (R2003).
[10] MANET Simulation and Implementation at the University of Murcia (MASIMUM), http://masimum.dif.um.es/
[11] Engelstad, P.E., Østerbø, O.N., "Analysis of the Total Delay of IEEE 802.11e EDCA and 802.11 DCF", Proceedings of IEEE International Conference on Communication (ICC'2006), Istanbul, June 11-15, 2006. (See also: http://folk.uio.no/paalee)
[12] R. Braden (Editor), “Requirements for Internet Hosts – Communication Layers”, RFC 1122, October 1989.
[13] M. Voorhaen, C. Blondia, “Analyzing the Impact of Neighbor Sensing on the Performance of the OLSR protocol”, Proceedings of the 4th International Symposium on Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks, 03-06 April 2006.
[14] T. Clausen, G. Hansen, L. Christensen, G. Behrmann, ”The Optimized Link State Routing Protocol Evaluation through Experiments and Simulation”, IEEE Symposium on “Wireless Personal Mobile Communications, September 2001.
[15] C. Gomez, D. Garcia, J. Paradells, “Improving Performance of a Real Ad-hoc Network by Tuning OLSR Parameters”, Proceedings of the 10th IEEE Symposium on Computers and Communications (ISCC), 2005.

Paper B :

Gateways and Capacity in Ad Hoc Networks

E. Larsen, V. Pham, P. Engelstad, and Ø. Kure

In proceedings of the International Conference on Advances in Human-oriented and Personalized Mechanisms, Technologies, and Services (I-CENTRIC), Sliema, Malta, October 26–31, 2008, pp. 390–399, ISBN: 978-0-7695-3371-1

Gateways and Capacity in Ad Hoc Networks

Erlend Larsen
erl@unik.no
Q2S NTNU
Norway

Vinh Pham
vph@unik.no
Q2S NTNU
Norway

Paal Engelstad
paal.engelstad@telenor.com
Telenor
Norway

Øivind Kure
okure@unik.no
Q2S NTNU
Norway

Abstract

Connectivity with external networks is an essential feature of many ad hoc networks, and such connectivity is enabled by gateway nodes. This paper investigates how the gateways affect the throughput in the ad hoc network. The throughput's dependency on gateway positions, number of gateways and handover properties is sought uncovered by simulations. The results show that the relative positions of gateways in ad hoc networks may have defining impact on the performance of the network. The average path length and the gateways' shared interference coverage are parameters that affect the performance.

1 Introduction

In recent years, ad hoc networking research is mainly focusing on ad hoc networks that are connected to external networks, enabling communication with other networks or even with the Internet. Connectivity with external networks is a relevant feature e.g. in emergency operations. In such a scenario, a connection to the headquarters (HQ) would enhance the usefulness of the ad hoc communication infrastructure. Through the connection to the HQ, updated information regarding the ongoing operation could be communicated both ways.

The access to external networks is enabled by introducing gateways (GWs) between the ad hoc network and other networks. However, there are few works studying the impact that the gateways have on the ad hoc network. The gateways are in the transmission range of other nodes in the ad hoc network, and they affect the performance of the ad hoc network, depending on their position, not only as forwarders of traffic into the ad hoc network, but also as receivers of traffic from the ad hoc nodes. Positioning the gateways strategically could improve the gateway coverage and reduce the path length for traffic traveling the ad hoc network

There are several different ways the gateways could be positioned in the ad hoc network:

- Static, placed independently of the ongoing scenario
- Semi-mobile, moved only when necessary
- Mobile, but collocated with a regular ad hoc node
- Mobile and independent of the ad hoc nodes

Several issues may limit the possibilities for moving a gateway: the access network availability (wired or wireless), available power sources, the physical gateway size and type of terrain.

In addition to the gateways' positions, their number also affects the performance of the ad hoc network. Operating with only one gateway, for example, the connectivity with external networks could be vulnerable, both in terms of reliability and coverage. Introducing more gateways provides several important benefits. First, the probability of having a working connection to the access network increases. If one gateway fails, nodes can send traffic via another gateway. Second, with several gateways the average path length for traffic traveling between the ad hoc network and external networks is reduced. Third, the risk of partitioning and disconnection from the access network in case of drifting is reduced. Finally, load balancing may be employed by routing traffic in such a manner that the capacity of both or all gateways is better utilized than the capacity provided by only one gateway. To control the number of gateways, it is assumed that the gateways can be activated or deactivated as necessary.

In most cases the gateways will not be optimally positioned relative to the ad hoc nodes. Thus, we need to understand how this affects the capacity and, in the event that the gateways' positions can be changed or their number controlled (e.g. by turning gateways on or off), be able to advice the best action.

The intention of this paper is to investigate the potential for different strategies and algorithms concerning the positioning, activation and mobility of gateways. The main reason is not to identify optimal gateway positions or the

optimal number of gateways in the scenario, but rather to investigate the throughput behavior with suboptimal gateway positions and without the optimal number of gateways. Through such understanding, we can develop algorithms that take advantage of these features, rather than chasing an optimum that is rarely achievable.

The rest of the paper is structured as follows. Section 2 provides an overview of related works. The simulation setup is presented in Section 3 and the simulation results in Section 4. Finally, conclusions are drawn in Section 5.

2 Related Works

In [5] the issue of finding the optimal position for one gateway in an ad hoc network is addressed. An algorithm calculating the optimal point is presented, where the optimal point is defined as the weighted geographic centroid of the node positions in the domain. The weights can be a multitude of metrics, but in the simulations the authors have used the load of each node, and the priority of the node.

Although little work has been published looking at the effect of gateways positions on capacity in ad hoc networks, several papers with a foundation in mesh networks deal with this issue. The nominal capacity of wireless mesh networks is studied in [10]. The paper shows how to determine the nominal capacity of the network through isolating the *bottleneck collision domain*, and how to calculate the throughput available to each node, with absolute fairness among the nodes being assumed. The bottleneck collision domain is the single link in the network that yields the lowest capacity due to interference with other links.

Two other mesh networks papers are [6] and [12], where the authors investigate how to determine the optimal position for gateways to minimize the interference in the network and optimize the capacity of the network.

In [11] the optimal position of a gateway on a set topology is researched briefly. Although the paper is aimed at studying wireless sensor networks, the PHY/MAC technology employed is the IEEE 802.11b. The authors draw the conclusion that a position in the center of the network is the best position for a gateway. However, the research only focuses on one single gateway, and also only one static topology is studied.

3 Simulations

All scenarios explored comprise an ad hoc network with one or more gateways (GWs) connected to an access network, through which they can reach the HQ (Fig. 1). The HQ can be a rescue center from where emergency operations are coordinated. Traffic between a node in the ad hoc network and the HQ has to pass through one of the gateways.

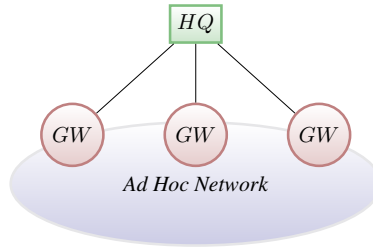


Figure 1. In all scenarios explored there are gateways (GW) positioned in the ad hoc network providing connectivity to the HQ.

To simplify the analysis, all scenarios explored are limited by the following: No traffic is transmitted between the ad hoc nodes, only to and from the HQ. Furthermore, one ad hoc routing protocol runs on all nodes including the gateways and the HQ, forming all together one routing domain. Moreover, the links between the HQ and the gateways have enough capacity to avoid being the bottleneck for the traffic flowing between the networks. In this way it is easier to isolate and investigate the gateways' impact on the ad hoc network.

Simulations were carried out with the ns-2 simulator [2] version 2.31 with the Carnegie Mellon University (CMU) Monarch project's wireless extension [15]. The IEEE 802.11 [9] MAC protocol was used with a sensing range of 550.0 m and a transmission range of 250.0 m. The nominal transmission rate was 2.0 Mbps and the control rate was 1.0 Mbps. The packet size, including the IP header, was 1024 bytes. The number of runs per data series varied between 10 and 30, and the random number generator was seeded heuristically.

A multiple wireless interfaces patch to the ns-2 simulator was developed by following the comprehensive guide by Agüero Calvo and Pérez Campo [4]. The patch creates one channel per network interface, so that a mobile node has multiple network stacks from the link layer and downwards for every channel it is connected to.

Furthermore, the proactive routing protocol Optimized Link State Routing (OLSR), which was implemented for ns-2 by the University of Murcia [1], was used for the routing. Though it includes code to enable Multiple Interfaces Declaration (MID) messages, it had to be extended to allow OLSR to route over multiple interfaces. The default settings for HELLO, TC and MID packets were used. The routing packets were given priority using ns-2's PriQueue, meaning that a routing packet is always inserted at the head of the interface queue. This ensures that routing packets are

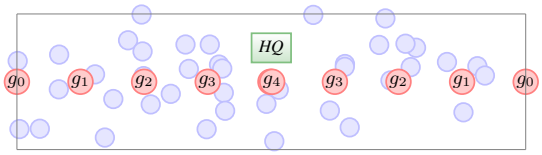


Figure 2. Simulation area with 40 nodes plus all gateway positions, both for the one and two gateway scenario.

Table 1. Gateway positions

Name	GW 1 x-pos	GW 2 x-pos	GW separation
g_0	0.0	1500.0	1500 m
g_1	187.5	1312.5	1125 m
g_2	375.0	1125.0	750 m
g_3	562.5	937.5	375 m
g_4	747.5	752.5	5 m
g_4	(1 gw) 750.0	-	-

not discarded and not delayed unnecessary if the interface queue holds many packets.

In the simulations, traffic was sent from all normal ad hoc nodes to a HQ node through both one and multiple gateways (uplink), and also the other way round, i.e. from one HQ node to all normal ad hoc nodes through one and multiple gateways (downlink). To explore scenarios without gateway handover, simulations with traffic flow from multiple HQ nodes to a subset of ad hoc network nodes (downlink) and vice versa (uplink) were also carried out. Every node generated or received the same traffic rate throughout each simulation, and jitter was introduced to avoid synchronization of packet flows.

The HQ node(s) were configured to transmit on other channels than the rest of the ad hoc nodes, and the gateway(s) were configured with two interfaces, to be able to relay traffic from the ad hoc network to the HQ node. Each gateway was only connected to one HQ node, and in simulations with two gateways each of the gateways communicated with the HQ node over a different channel than the other gateway to save bandwidth. The channels used to communicate with the HQ node were modified to support longer range transmissions, so that the HQ node could be placed in the center, while the gateway nodes were placed on the edges of the 1500 m long simulation area. The nodes' receive threshold was lowered and the 802.11 MAC timers were extended to allow for a longer propagation time. Although this reduces the capacity a little for these channels, it has not affected the simulation results, as congestion has been verified only to occur in the ad hoc network channel, and not in the other channels.

For the scenario in Fig. 1 a simulation area of 1500x400 m^2 was chosen, both to limit the number of nodes due to the simulation processing time and to have a node degree so large that network partitioning would not dominate the results.

Random topologies were generated for each simulation run, consisting of 40 nodes plus gateway and HQ nodes. An example showing the various gateway positions for both one and two gateways can be seen in Fig. 2, and the x-axis positions are noted in Table 1. All gateways were centered on the y-axis, in position 200.0. When simulating with only one gateway, the position of the left gateway in the two gateway scenario was used.

All simulations were run with mobility, with all nodes moving at a constant speed of 5.0 m/s using the random waypoint with reflection mobility model. The mean travel length before direction change was 100.0 s and the travel length variation was 50.0 s. The node positions and mobility were generated using an application developed by S. PalChaudhuri [3].

In the simulations, partition occurred with some probability. Partitioning was impossible to avoid, and the varying degree of partitioning does to some extent affect the simulation results, causing a lower throughput average and a broadening of the confidence intervals.

The simulations were run for 500.0 s. Traffic transmission began at 50.0 s, and the throughput/loss measurement period started at 60.0 s and lasted until the end of the simulation, leaving a 10.0 s settling period of traffic without measurement.

Table 2 lists various simulation parameters used in the simulations.

4 Results

This section presents the simulation results. First, a reference configuration is established as a benchmark that other results can be compared to. The simulation results from this configuration are thoroughly investigated. Then, the effect of moving the gateway of the reference configuration towards the edge of the simulation area is explored. We also investigate how the throughput varies when we add another gateway to the reference configuration and vary the position of the two gateways. Next, the effect of adding several more gateways to the reference configuration is studied. Finally, simulations are run without dynamic gateway rerouting to see how this affects the system throughput.

Table 2. Simulation parameter settings

Simulator	ns-2 version 2.31
Radio-propagation model	TwoRayGround
MAC type	802.11b
Interface queue type	FIFO with DropTail and PriQueue
Antenna Model	OmniAntenna
Nominal transmission rate	2 Mbps
Basic rate	1 Mbps
Packet Size with IP header	1024 bytes
Movement speed	5.0 m/s
Mobility model	Random Direction with Reflection
Number of nodes excl. GWs	40
Size of simulation area	1500 x 400 m^2
Simulation time	500 s
Traffic type	CBR
Pause time	0 s
OLSR HELLO interval	2 s
OLSR HELLO timeout	6 s
OLSR TC and MID interval	5 s
OLSR TC and MID timeout	15 s

4.1 Reference configuration

4.1.1 Uplink traffic

To make the comparison between the various simulation results easier to study, a reference configuration with one single gateway in the center of the simulation area is defined. The topology mobility has a set speed of 5 m/s for all ad hoc nodes. Traffic is sent uplink from the ad hoc nodes to the receiver on the other side of the gateway.

Fig. 3 shows the throughput and the average number of hops for our reference configuration (uplink traffic) with a center positioned gateway. (It also shows the same results for downlink traffic, which will be discussed later.)

For the uplink traffic in Fig. 3 it is observed that already at a 100 kbps load there is packet loss, which remains at around 30% until 600 kbps load. At a load of 600 kbps the loss starts to rise sharply beyond 30%, indicating that for the reference configuration congestion occurs for loads over 600 kbps. The average ad hoc network path length is just over two hops, meaning that an average packet has to be transmitted twice. Since the first hop interferes with the second hop, the bandwidth with 1 kB size packets is effectively reduced from around 1500 kbps to around 700 kbps, as shown by Li et al. [13]. However, this leaves a loss of 100 kbps unexplained by capacity limitations.

The reasons for loss shown in Fig. 4 reveals that at a load of 100 kbps the loss caused by maximum retransmission limit (RET) is slightly below 20%. As the input rate is increases, packets discarded because of a full interface queue increases. However, at lower rates the retransmission limit is the main reason for packet loss. The underlying reason for the RET loss is the latency in the routing protocol's link break detection. This effect is described in [14]. The loss abbreviations are explained below:

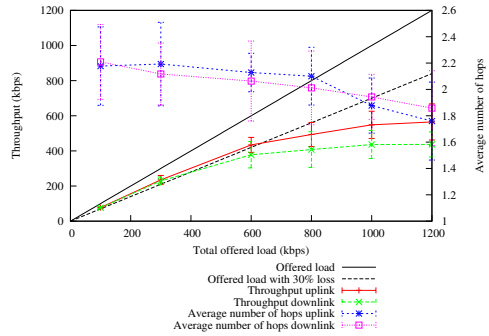


Figure 3. Throughput and average number of hops for one gateway at the center of the topology. Uplink and downlink traffic. (95% confidence interval)

RET The packet was retransmitted at the MAC layer the maximum number of times allowable, and was discarded.

ARP The packet's MAC-layer destination was unknown, and an ARP request was sent out. However, a new packet to the same destination had to be stored while waiting for ARP reply, and the packet was discarded.

IFQ The packet was discarded as the interface queue was full, leaving no room for this packet.

NRTE The packet was discarded as the routing protocol could find no valid route to the destination (No Route To Host).

LOOP The packet was discarded because the received packet's sender IP address was the same as the current node's IP address, meaning that the packet has gone in a loop.

TTL The packet was discarded due to its Time To Live IP-header field reaching 0.

Fig. 5 shows the positions for all CBR transmissions over 30 runs with a total system load of 100 kbps, meaning that each node creates and sends a packet to one of the gateways

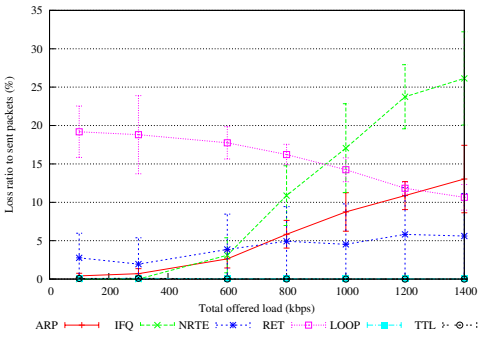


Figure 4. Reasons for loss for uplink traffic with one gateway in the center of the area. (95% confidence interval.)

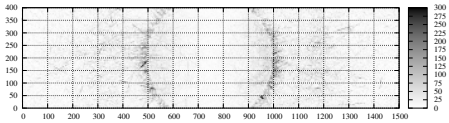


Figure 5. Transmission plot for the reference configuration with 100 kbps offered system load and 30 runs per plot. 5x5 m^2 resolution.

every 3.2 second. Traffic direction is uplink. By keeping the offered load this low, the contention effect observed at higher loads is avoided. A ring of a very high number of transmissions is seen just outside the transmission range of the gateway. The reason for this ring can be explained as follows: Assume that a node has packets in the interface queue destined to the gateway as next hop. When this node moves away from the gateway, each of these packets will be attempted to be transmitted on the radio medium, until the retransmission limit is reached and the packet is removed from the queue. This happens just outside the transmission range of the gateway. The same effect is also present for all other nodes when packets in a node's interface queue are destined to an unreachable next hop. However, as the only static object in our simulation is the gateway, the effect at the gateway's transmission range stands out, and the ring is thus observed.

This retransmission effect is in fact encouraged by all shortest hop routing protocols as they will tend to select forwarding nodes along the edge of the transmission range in order to reduce the path length as much as possible. In a topology with mobility this has negative consequences for the throughput.

4.1.2 Downlink traffic

In the reference configuration there is only uplink traffic. However, downlink traffic will be just as relevant in the scenarios explored. Therefore it is necessary to investigate if there is any difference in throughput when traffic is sent from the gateway outwards into the ad hoc network, compared with the other way around. In the downlink traffic scenario, where one gateway transmits traffic into the ad hoc network, an increase in exposed node events is expected, since the traffic is forwarded away from the sender in all directions. In the uplink scenario, on the other hand, the hidden node effect is dominant, since the traffic is directed towards one point. While the exposed node problem would result in a higher contention than necessary, again leading to congestion at lower loads, the hidden node problem would show itself by an increased number of collisions. The hidden and exposed node problems are explained well in [7].

In addition to showing the results for the reference configuration, Fig. 3 also shows the throughput and average number of hops with downlink traffic with the same topology. A lower throughput for the downlink traffic than for the uplink traffic is observed at higher loads. At lower loads (100 and 300 kbps), however, uplink and downlink have the same performance. This means that the hidden/exposed node effects of the downlink and uplink traffic simulations have about the same impact on throughput. However, the fact that the throughput is lower for the downlink simulations with mobility still remains to be explained.

Fig. 6 shows the IFQ loss for the uplink and downlink traffic. In addition, the collision ratios for the two scenarios are shown. The collision ratio difference corresponds well with the expected effect that the hidden node problem would create in the uplink scenario, as the collision rate is very much lower for the downlink scenario than for the uplink scenario. However, for the downlink traffic simulation congestion occurs already at 600 kbps, which cannot be explained by the hidden/exposed node effects. The reason for this behavior must be linked to the topology mobility. Since only one gateway transmits all the downlink traffic, the transmission rate is limited by the capacity of this node, while in the uplink scenario all nodes compete for the channel. In the downlink scenario the sending gateway may encounter the same amount of link destinations moving out of range as in the uplink simulations. However, as the rate is fully dependent on the gateway's transmissions, the defer time caused by retransmissions affects the throughput

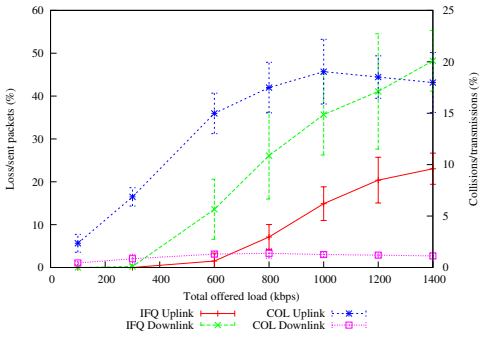


Figure 6. Losses due to drops from the link layer queue (IFQ) along with collision ratio (COL) for uplink and downlink traffic for one gateway in position g_4.

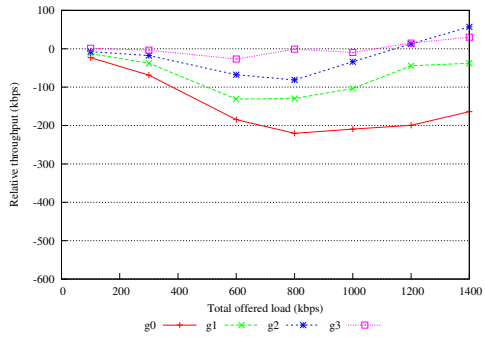


Figure 7. Throughput results for one gateway, relative to the center position. Uplink traffic.

in the downlink simulations much harder than in the uplink simulations.

4.2 The positioning of one gateway

Fig. 7 shows the throughput results for the gateway positions g_0-g_3 relative to g_4, where g_4 represents the results of the reference configuration. It is observed that the worst position is on the edge of the simulation area, in position g_0, while g_3 perform about equal to the reference configuration, g_4. As the gateway comes nearer to the edge of the simulation area, the average packet has to travel longer to reach the gateway. By moving the gateway from the center position to g_0 a reduction in the throughput, relative to the reference configuration, is observed. With a load of 600 kbps the throughput is reduced from 415 kbps to 235 kbps, i.e. only 57% of the maximal throughput in the center position. Likewise, in g_1 the throughput is only 73% of that in the throughput at the optimal position. This behavior is expected, as moving the gateway towards the edge of the area increases the average path length. As long as the transmission radius is inside the area on both sides, the movement will bring some nodes closer. However, when the node is only 250 m from the edge, movement towards the edge will no longer shorten any path, since all nodes close to the edge already is covered by the gateway. Therefore, the average path length will increase more rapidly in the last 250 meters than farther from the edge.

Fig. 8 shows the throughput and loss for each 10th meter when moving the gateway from the edge to the center of the simulation area. This figure confirms that the throughput is higher closer to the center than nearer the edge. In addition it reveals that the lower throughput closer to the edge is due to the unavailability of routes, causing 30% of the sent packets to be lost. When the gateway is closer to the center of the simulation area this loss is reduced to below 10%. The reason for this lack of routing information can be traced to the average path length. When the average path length is high, routing information (OLSR Topology Control (TC) messages transmitted by broadcast) runs a higher risk of getting lost on the way.

4.3 Adding a second gateway

At this point in the analysis, a second gateway is introduced into the simulation area. From the one gateway results, it is expected that the average path length indicates what positions are optimal and yield the best performance. This would point to positions g_2 as the optimal ones, where each gateway covers exactly half of the simulation area. However, the two gateways could also be affected by interference when being too close to each other, thus losing the gain of having two gateways.

Fig. 9 shows the results relative to the results of our reference configuration. It is observed that all positions g_0-g_3 yield better throughput than the one gateway simulations. The positions g_1 and g_2 are the best positions for the gateways, while the center position comes out worse of all positions. As expected g_2, which has the lower average path length, shows the best performance. According to the average path length the pair positions g_1 and g_3, and g_0 and g_4 also have equal path length, due to the symmetry around each half's center. However, they do not yield equal throughput. The reason is that in the positions g_3 and g_4 the gateways are so close that they begin to share the channel, as nodes transmitting to one of the gateways at the same time interfere with the other gateway. This is most outspo-

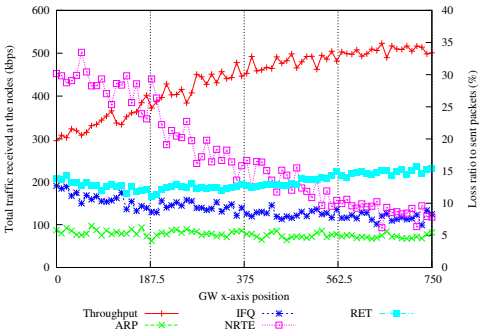


Figure 8. Throughput and loss results for one gateway with horizontal position from the edge to the center on the x-axis. Uplink traffic, 800 kbps.

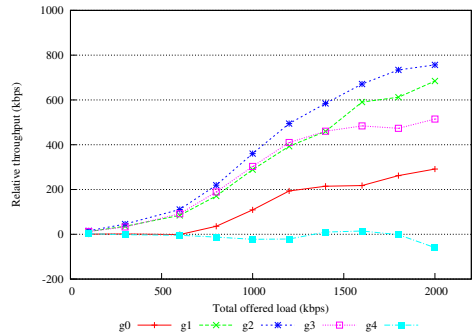


Figure 9. Throughput results for two gateways, relative to the reference configuration results. Uplink traffic.

ken in position g_4, where the two gateways are placed side by side. Here there is no gain in having two gateways instead of one. In the downlink results (Fig. 10), however, g_4 shows a very slight increase in throughput. This can be explained, as the defer effect is reduced due to two gateways competing for the medium, instead of one. The other results for the downlink simulations show the same pattern as the uplink results.

For the two-gateway configurations the positions at the edge yield worse performance than those nearer the center in the same way as for the one-gateway configurations discussed above. However, different from the one-gateway configurations, the center is not the optimal position to put both gateways. Instead, to reduce the average path length as much as possible, halfway between the center and the edge is the optimal position.

Unlike for the one-gateway configurations, there is a considerable difference between results from gateway positions with the same average path length. The explanation is the shared interference coverage of the two-gateway configurations, ultimately making the two gateways perform as one gateway in position g_4. Thus, as long as the gateways are positioned in range of the ad hoc network and with a distance between the gateways being more than the interference range, the performance will always be better than that of one gateway.

Fig. 11 shows the throughput and loss of uplink (a) and downlink (b) traffic as the distance between the gateways varies. The behavior is clearly the same for uplink and downlink. However, the downlink results have a distinct increase in throughput at a 550 meter distance, corresponding to the interference range. The reason is that as the gateways transmit traffic into the ad hoc network, the effective transmission rate depends on the channel capacity. When they are inside each other's interference range, they share the channel, and only one can transmit at a given time. When they are outside each other's interference range, on the other hand, both gateways can transmit at the same time. Nevertheless, this increases the number of packets lost due to maximum MAC layer retransmissions. The reason is that when both gateways transmit simultaneously, the nodes that are located between the gateways are still covered by the interference range of both gateways and will experience collisions. Finally, the packet loss due to the lack of a route to the destination is greatly reduced compared to the one gateway simulations. This is due to a shorter average path length.

4.4 More than two gateways

The analysis shows that that going from one to two gateways normally increases the throughput performance. However, in the worst case, where the two gateways are placed side by side, the performance is about the same as in the one-gateway simulations. This raises the question of what number of gateways yields the maximum performance. To investigate this, simulations with up to seven gateways were carried out. The gateways were evenly distributed horizontally from 0 to 1500 on the x-axis. The results (Fig. 12) show that there is an increase in the throughput for each gateway added, up to five gateways. The topology shown in Fig. 13 reveals that with five gateways, the complete simulation area is within the transmission range of one or more gateways, allowing all nodes to reach a gateway in one hop. With this particular simulation setup, increasing the number of gateways above five yields no higher throughput.

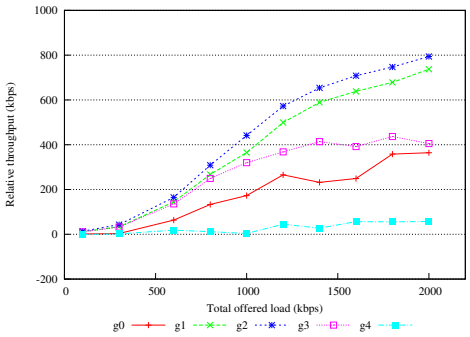
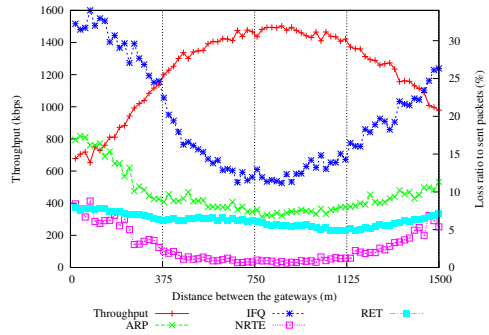
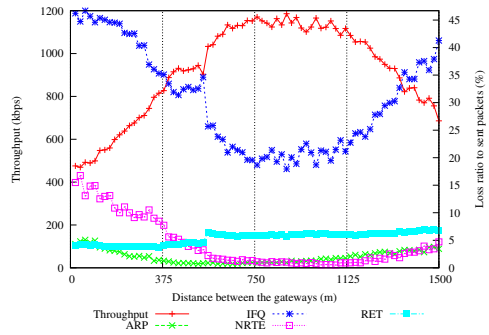


Figure 10. Throughput results for two gateways and downlink traffic, relative to the reference configuration with downlink traffic results.



(a) Uplink traffic, 2000 kbps offered load



(b) Downlink traffic, 1600 kbps offered load

Figure 11. Throughput and loss for two gateways with the separation between the gateways on the x-axis.

4.5 Handovers between the gateways

All simulations presented so far have used dynamic rerouting, and traffic has been routed between a mobile node and its nearest gateway at any one time. With node mobility, this feature requires handover between gateways. However, in real systems handing over a communication session from one gateway to the other normally means a change of some IP address to ensure correct routability of packets in and out of the ad hoc network. As a change of IP address will normally break a communication session (e.g. such as a TCP session), some mechanism for gateway handovers (e.g. based on a modified version of mobile IP or network address translation) is normally required. As described in [8], such handovers may be complex and generate disruptions and unwanted overhead.

With mobility and without handover, the probability that a node is connected to the nearest of two gateways decreases and approaches 50% as times goes by. In other words, it approaches the results of using static gateway selection. Therefore it is interesting to study the performance and behavior of static gateway selection, as it can be regarded as the worst case of performance of a system with mobility, but without handovers between the gateways.

The simulations of static gateway selection were carried out with a topology where two HQ nodes are connected to a gateway each. Randomly each ad hoc node selects one of the two HQ nodes to be the receiver, in such a way that 20 nodes transmit to each of the HQ nodes.

Fig. 14 shows the throughput for static gateway selection with two gateways at the positions g_0, g_2 and g_4. The figure also shows the corresponding results with one gateway. There seems to be no gain in having two gateways over one gateway here, as the one and two gateway results are almost identical, i.e., the same positions yield the same throughput. This is expected, because as the gateway positions move towards the edge of the simulation area, the path length increases. The reason is that nodes on the opposite side of the area that are connected to the gateway do not change gateway but continues to transmit to the same more distant gateway, causing longer paths in the simulation in the same way as simulations with one gateway. Thus, the way the throughput relates to the gateway positions vary in the same manner as for the one gateway scenario, where the edge positions are less favorable positions than centered positions. This means that in a mobility scenario the results for two gateways with static gateway selection will perform worse than the results obtained with two gateways and dynamic gateway selection, but better than the results obtained with static gateway selection.

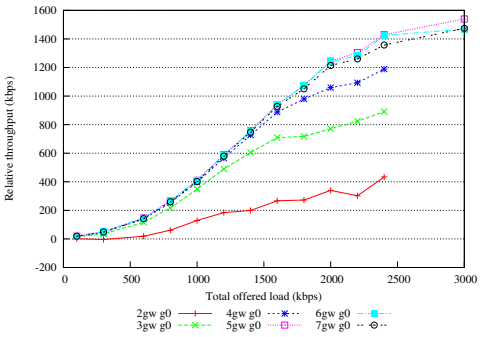


Figure 12. Throughput comparison for 2-7 gateways relative to the reference configuration.

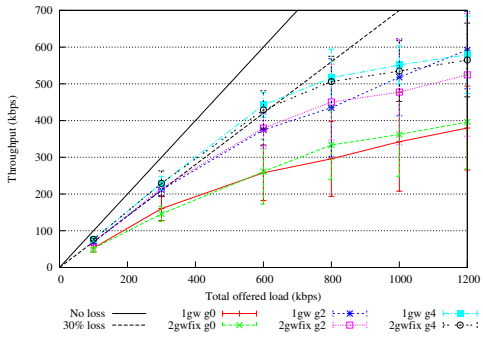


Figure 14. Throughput results for one and two gateways with static gateway selection.

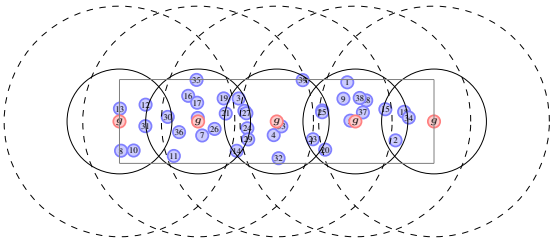


Figure 13. Gateway positions for the five gateways scenario.

Ultimately, if all nodes transmit via a random gateway there is no gain in having two gateways over one gateway. However we have only portrayed what eventually will be the case in a scenario without handover with randomly moving nodes. If each node at the start of the simulation instead was sending and receiving traffic through the nearest gateway, then, considering the mobility velocity and direction of the nodes, one could have handover at set times, balancing between the need to optimize the network paths and the disruption and extra network load a handover may incur.

Considering a suboptimal positioning of the gateways in this static gateway selection scenario, the higher degree of random selection of gateway, the more preferable is the center position. However, in the simulations with dynamic gateway selection positioning both gateways in the center gave the worst performance. This means that in a configuration where nodes are less than 50% but more than 0% likely to be transmitting traffic through the nearer gateway, the gateways should still be outside each other's interference range. They should on the other hand be placed closer to each other than would be the case in a dynamic gateway selection scenario. At the same time the question arises as to what network with so much mobility could be expected to remain at the same location for a longer period of time, or if it would be more inclined to drift?

5 Conclusions

The presented analysis has sought to clarify what impact gateways have on the capacity of an ad hoc network, with regard to their positions and number, traffic patterns and whether there is a mechanism that allows traffic of a session to be handed over between two gateways.

The analysis showed that retransmissions due to mobility will have a negative effect on throughput by wasting channel time. Most shortest path routing protocols will unintentionally support this behavior, as the shortest path will select forwarding nodes close to the node's transmission area edge. It would be better to seek relay nodes closer to the sender node, at least when the relative mobility is high. Link layer notification or limiting the retransmissions might also reduce this problem [14]. Downlink traffic has other features than uplink traffic, as the gateway becomes the bottleneck. Due to the retransmissions caused by mobility, the gateway spends much time in defer state, and this reduces the throughput compared to uplink traffic.

Positioning the gateway at the edge of the area yields high packet loss due to lack of route. The farther from the center the gateway was placed, the lower was the throughput, due to the increasing average path length.

Results with two gateways showed that additional gateways will greatly increase the capacity. The gateway position yielding the best throughput was the position where the

average path length was the lowest. Also, as the gateways were put closer together, they interfered with each other, reducing the combined capacity of the gateways. Considering the risk of suboptimal gateway positions, and also the risk of a gateway becoming inoperable, the connectivity to external networks is much more robust with two or more gateways than with only one gateway.

The analysis showed that as long as there are not enough gateways to cover the entire ad hoc network with a gateway's direct transmission coverage, both the positions of the gateway relative to the ad hoc nodes and the position relative to other gateways need to be considered. It was observed that until the complete network is covered by a gateway, adding a new gateway will increase the overall throughput capacity. Thus, although increasing the number of gateways may cause interference between the gateways, it is better to reduce the average path length than to avoid interference, since a new gateway causes no reduction in throughput. The only risk is a smaller increase in throughput than the potential increase.

It was shown that with random node mobility and without a mechanism to hand over traffic sessions between gateways, the performance advantages of having two gateways will decrease over time and approach the a lower performance boundary that is almost equal to the performance of a network with only one gateway. The higher the mobility, the faster is it expected that the performance advantage of multiple gateways will decrease and reach this boundary. Although the analysis demonstrated the benefits of having a handover mechanism, such a mechanism might be complex and generate disruptions and unwanted overhead. Thus, depending on where the gateways are positioned and the mobility of the ad hoc nodes, an optimal time might exist for triggering a handover for all or a subset of the nodes. This issue is left for further study.

Another issue for further work is to develop algorithms to support optimal positioning of gateways based on the possibilities offered at the scene of an emergency. One should also study how gateways connected to access networks with differing capacity can affect the optimal positions of the gateways in the network.

6 Acknowledgement

This work was supported by the ITEA Easy Wireless and CELTIC DeHiGate projects.

References

[1] MANET Simulation and Implementation at the University of Murcia (MASIMUM). http://masimum.dif.um.es/.

[2] Network simulator 2 - ns2. http://nsnam.isi.edu/nsnam/.

[3] ns-2 code for random trip mobility model. http://www.cs.rice.edu/~santa/research/mobility/.

[4] R. Agüero Calvo and J. Pérez Campo. Adding multiple interfaces support in ns-2. Technical report, University of Cantabria, 2007.

[5] M. Ahmed, S. Krishnamurthy, R. Katz, and S. Dao. An architecture for providing range extension by deploying mobile gateways in ad hoc networks. In *Personal, Indoor and Mobile Radio Communications, 2002. The 13th IEEE International Symposium on*, volume 4, pages 1660–1664 vol.4, 2002.

[6] B. Aoun, R. Boutaba, Y. Iraqi, and G. Kenward. Gateway placement optimization in wireless mesh networks with QoS constraints. *Selected Areas in Communications, IEEE Journal on*, 24(11):2127–2136, Nov. 2006.

[7] V. Bharghavan, A. Demers, S. Shenker, and L. Zhang. MACAW: A media access protocol for wireless LANs. In *SIGCOMM94 Conf. on Communications Architectures, Protocols and Applications*, pages 212–225, Aug. 1994.

[8] P. E. Engelstad, A. Tonnesen, A. Hafslund, and G. Egeland. Internet connectivity for multi-homed proactive ad hoc networks. *Communications, 2004 IEEE International Conference on*, 7:4050–4056 Vol.7, 2004.

[9] IEEE. Wireless LAN medium access control (MAC) and physical layer (PHY) specification. IEEE standard 802.11, June 1999.

[10] J. Jun and M. L. Sichitiu. The nominal capacity of wireless mesh networks. *Wireless Communications, IEEE [see also IEEE Personal Communications]*, 10(5):8–14, 2003.

[11] E. L. Kuan. A study into the practical issues related to a deployed ad hoc wireless sensor network. In *Wireless Communications and Networking Conference, 2005 IEEE*, volume 4, pages 1952–1957 Vol. 4, 2005.

[12] F. Li, Y. Wang, and X.-Y. Li. Gateway placement for throughput optimization in wireless mesh networks. *Communications, 2007. ICC '07. IEEE International Conference on*, pages 4955–4960, 2007.

[13] J. Li, C. Blake, D. S. J. De Couto, H. I. Lee, and R. Morris. Capacity of ad hoc wireless networks. In *MobiCom '01: Proceedings of the 7th annual international conference on Mobile computing and networking*, pages 61–69, New York, NY, USA, 2001. ACM Press.

[14] V. Pham, E. Larsen, K. Ovsthus, P. Engelstad, and O. Kure. Rerouting time and queueing in proactive ad hoc networks. In *Performance, Computing, and Communications Conference, 2007. IPCCC 2007. IEEE International*, pages 160–169, 2007.

[15] The CMU Monarch project. Wireless and mobility extensions to ns-2. http://monarch.cs.cmu.edu/cmu-ns.html.

Paper C :

Routing with Transmission Buffer Zones in MANETs

E. Larsen, L. Landmark, V. Pham, Ø. Kure, and P. E. Engelstad

In proceedings of the IEEE International Symposium on a World of Wireless Mobile and Multimedia Networks (WoWMoM), Kos, Greece, June 15–18, 2009, ISBN: 978-1-4244-4440-3.

Routing with Transmission Buffer Zones in MANETs

Erlend Larsen, Lars Landmark, Vinh Pham, Øivind Kure
Q2S NTNU
{erl, larslan, vph, okure}@unik.no

Paal E. Engelstad
UniK/UiO
SimTel (Telenor/Simula)
paalee@unik.no

Abstract

Dealing with link breaks in MANETs is a challenge for the routing protocol. This paper proposes a mechanism to reduce the negative impact of link breaks on the routing. The transmission area of a node is divided into a safe zone close to the node and an unsafe zone (i.e. buffer zone) near the end of the transmission range. The probability is high that link breaks occur with neighboring nodes located in the buffer zone, while links to neighboring nodes in the safe zone are expected to be more stable. Thus, neighbors in the safe zone are preferred as relay nodes, while neighbors in the buffer zone are only used if necessary to avoid network partitioning. The main cost of this mechanism is that the mean number of hops between two nodes is higher than without the mechanism, but simulations show that the solution offers increased throughput.

1 Introduction

Mobile ad hoc networks (MANETs) are designed to function without any prior infrastructure in place, making them attractive for use in emergency and military scenarios. However, ad hoc networks are limited in performance due to their nature of distributed wireless communication and often random and rapid topology changes. One way to increase the network performance in MANETs has been by sharing and utilizing information between the network layers (cross-layering), since traditional wired networks – for which the network stack was invented – have not had to deal with the conditions that MANETs have to handle. Due to the faster timings of the lower layers, the routing protocol can for example act much faster in detecting changes in connectivity by using lower layer information, and determine the distance to neighboring nodes using the signal strength of incoming transmissions.

The routing protocol selects a route at lowest cost, and the most widely used cost metric in ad hoc networks is *shortest path*. With this metric, the routing protocol selects a route with a minimal number of hops between a source and a destination. The advantages are effective use of network resources, little delay and little overhead. However, a disadvantage of shortest path routing is that the routing protocol tends to select nodes on the edge of the transmission area as relay nodes, since this normally reduces the number of hops between a source and a destination. The problem is that with node mobility, it is the nodes close to the transmission area edge that have the highest probability to move out of the transmission area. Furthermore, links to nodes near the edge of the transmission range also have a higher probability of bit errors, as the signal strength decreases with distance.

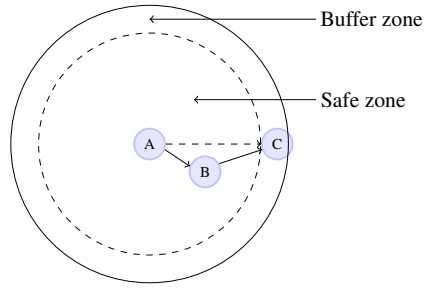


Figure 1. Transmission area zones of node A with safe node (B) and unsafe node (C).

The main contribution of this paper is the proposal of a mechanism to reduce the disadvantages of shortest path routing in MANETs. The key idea is to divide the transmission area of a node into a *safe zone* close to the node and an *unsafe zone* (i.e. *buffer zone*) near the end of the transmission range (Fig. 1). The routing protocol prefers neighbors that are located in the safe zone as relay nodes, while neighbors in the buffer zone are only used if necessary to avoid network partitioning.

The routing protocols for ad hoc networks can be divided into two groups, proactive and reactive. A proactive routing protocol aims to have an updated view of the

978-1-4244-4439-7/09/$25.00 ©2009 IEEE

network with routes to all nodes at any time, while a reactive protocol only establishes routes to the nodes where an application needs to send traffic. This paper focuses on MANETs using a proactive routing protocol.

Although the simulations and evaluations presented in this paper are based on using the Optimized Link State Routing protocol (OLSR) [9] as the proactive routing protocol, the results and analysis presented here should be applicable to the use of other proactive routing protocols as well. Without loss of generality, it is also assumed that IEEE 802.11 [12] is used as the technology at the physical link and link layer. IEEE 802.11 uses Carrier Sense Multiple Access (CSMA) with exponential backoff. Retransmissions occur until the DATA frame is successfully transmitted (i.e. the ACK frame successfully received) or until the retry counter reaches the retry limit upon which the transmitting node discards the DATA frame.

The rest of the paper is structured in the following way. First, Section 2 present some background on how routing protocols deal with link breaks. Then, in Section 3 the devastating effect link breaks have on Medium Access Control (MAC) retransmissions and on the overall throughput is documented through simulation. The simulation setup is presented in this context. Section 4 presents the proposed solution in full. The solution is then evaluated in Section 5 by simulations. Finally, related work is presented in Section 6, before the paper is concluded in Section 7.

2 Background

The normal way of detecting link breaks for a routing protocol is through lost polling packets (i.e. lost Hello packets). The Hello packets of OLSR are transmitted between one-hop neighbors at a specified time frequency (e.g. every 2 seconds, which is the recommended transmission frequency of OLSR) and provide neighborhood connectivity information and a means for link break detection. If no Hello packet from a neighbor is received within a specified time interval (e.g. within 6 seconds, the recommended interval of OLSR), the neighbor is considered unavailable and a link to this neighbor is considered as broken and invalid.

Another way for the routing protocol to detect link breaks is to leave it up to a mechanism implemented at the underlying link layer. The routing protocol must then be notified explicitly about a link break by the link layer. The disadvantage of this *Link Layer Notification* (LLN) approach might be the cost of additional implementation complexity. However, the advantage is that the link layer is normally able to detect link breaks sooner. As a link layer, IEEE 802.11 is normally capable of detecting a link break considerably faster than a second. In contrast, without LLN and with the recommended values of

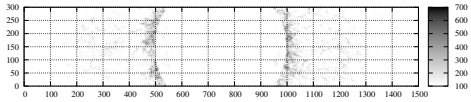


Figure 2. Accumulated transmissions diagram.

OLSR, a link break will not be detected before 4 seconds at best and 6 seconds at worst. This paper focuses first on link break detection through lost Hello packets, while the use of LLN will be discussed and evaluated by the end of the paper.

It is important for the overall performance to detect the link break in a timely fashion, since two negative effects occur in the period between the physical link break and the detection by the routing protocol. First, the packets queued in the interface queue are marked with an unreachable next hop address. This means that these packets will never reach their destination, and are at this point lost. Second, these packets will be attempted transmitted several times by the MAC layer before they are discarded. This will steal valuable medium time from packets transmitted from other nodes with a valid next hop address.

The retransmission effect is illustrated through a simulation where a node was placed in the center of the simulation area and set up to receive data from 40 nodes moving randomly inside the simulation area at 10 m/s (Fig. 2). In this simulation, a node inside the transmission area of the receiving node successfully sends traffic to the receiving centered node until it moves out of the receiving node's transmission area. At that point a link break occurs, but it is not detected by the transmitting node's routing protocol for another 4-6 seconds. At the time when the link break is detected by the routing protocol, the node may have travelled 40 to 60 m past the edge of the transmission area of the receiving node. During this time the MAC layer will transmit each packet with the receiving node as MAC destination several times. Fig. 2 shows the simulation area with the positions for all occurring transmissions plotted in. A ring of an increased number of transmissions is observed outside the transmission area of the receiving node, a direct effect of link breaks and subsequent retransmissions.

3 Analysis of the effects of link breaks

In this section the normal behavior of OLSR as a proactive routing protocol is investigated. First the simulation setup is presented, and then the results documenting the link detection problem are shown. To simplify the analysis, it is assumed that all link changes occur as a consequence of node mobility, while the radio conditions

Table 1. Simulation parameter settings

Radio-propagation model	TwoRayGround
Interface queue type	FIFO with DropTail and PriQueue
Interface queue size	300 packets
Maximum MAC retries	7
Antenna Model	OmniAntenna
Nominal transmission rate	2 Mbps
Basic rate	1 Mbps
Simulation time	500 s
Random seed	Heuristic
Traffic TTL	32
OLSR Hello interval	2 s
OLSR Hello timeout	6 s
OLSR TC interval	5 s
OLSR TC timeout	15 s

are considered as stable, assuming full radio connectivity within the radio transmission area and no connectivity outside this range.

The simulations were performed using the ns-2 network simulator [2] version 2.31. The Optimized Link State Routing Protocol (OLSR) [9], [1] was used for multi-hop routing and the IEEE 802.11 protocol [12] was used as MAC layer. The nodes were divided into two equally large groups, and each node in each group transmitted packets to all other nodes in that group. The traffic type was UDP at constant bit rate, and the packet size was 64 bytes.

The simulation area was 1500 x 300 m^2 with 40 nodes in all simulations. The dimensions were selected to get a topology with many hops and little partitioning without the need of a very large number of nodes. The mobility model is Random Direction with Reflection with a 10 s travel time before direction change and a 5 s travel time delta. All nodes had the same velocity, and there was no pause time. All simulations with the same velocity were run on the same set of 10 different topologies, to make the comparison between the different algorithms as fair as possible. The simulation results were sampled over 10 simulation runs, and all results are presented with a confidence interval of 95%. Other simulation parameter settings are presented in Table 1.

Running simulations with the standard OLSR implementation (i.e. without a buffer zone) reveal that the goodput (Fig. 3) is significantly lower with higher mobility, and the loss is at over 40% even at a total data load of 50 kbps for 10 m/s node velocity. For 10 m/s node velocity the loss (Fig. 4) up to 150 kbps load is mainly caused by discards due to maximum MAC retries (RET). (When the number of retransmissions of frames reaches the retry limit, the transmitting node discards the DATA

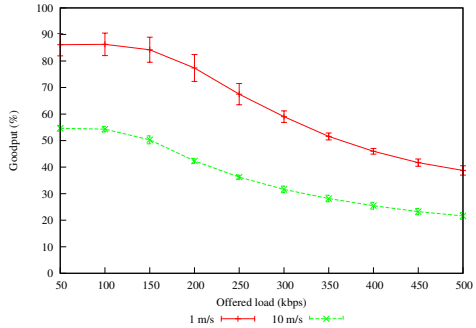


Figure 3. Goodput results, standard OLSR.

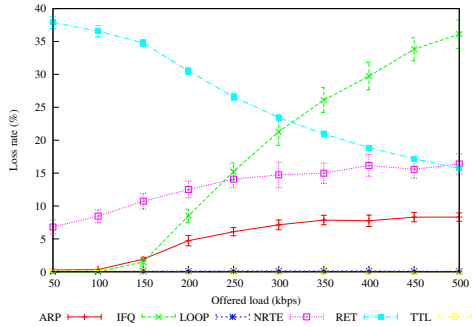


Figure 4. Loss reasons, 10 m/s, standard OLSR.

frame, and a RET loss occurs.) This confirms the suspicion that mobility and the delayed link break reaction of the routing protocol causes considerable loss.

Above 150 kbps the loss caused by tail drops from the interface queue (IFQ loss) increases, together with the loss from the Address Resolution Protocol buffer (ARP loss), while the RET loss ratio declines. Thus, above 150 kbps, the network gets more and more congested. In addition, some loss is caused by the lack of route to the destination (NRTE loss), increasing slowly as more routing control packets are lost due to collisions. This means that topology information is lost, leading to more packets being dropped because of a lack of route to the destination (NRTE).

4 Proposed solution

This section presents the buffer zone solution, which is an algorithm for improved handling of mobility and link breaks in the ad hoc network.

4.1 Analysis

The closer a neighbor node comes to the edge of the transmission area, the more likely is a link break to occur. The detection of this link break will delay until no Hello packet has been received within a given time interval. Thus, a node in the area where a link break is likely to occur could be considered an unsafe node that should not be relied on to forward traffic, if an alternative is possible.

The part of the transmission area where it is no longer guaranteed that a link to a neighbor will remain stable is referred to as the unsafe buffer zone, and the other part is thus referred to as the safe zone, as can be seen in Fig. 1. Likewise, a node in the safe zone is referred to as a safe neighbor, while a node in the unsafe buffer zone is referred to as an unsafe neighbor.

The minimum time for a node at the edge of the safe zone to disappear can be expressed as the distance to the transmission edge divided by the maximum relative velocity between the two neighboring nodes of the link. With the recommended settings of OLSR, the routing protocol will detect a link break between 4 and 6 seconds after the link break has occurred. In the worst case, a node with direction directly opposite of the transmitting node with a velocity of 10 m/s would be able to travel up to $2s \cdot 20m/s = 40m$ before the first Hello packet is lost, and $6s \cdot 20m/s = 120m$ from the link break has happened until it is detected and acted upon. This means that to be sure that a neighbor does not move out of the transmission area before it is marked as unsafe, any node closer to the edge than 120 m should be marked as unsafe.

However, 120 m would be the maximum distance from the edge where a link break could occur due to the movement of a node. Defining such a large part of the transmission area as a buffer zone has two drawbacks. First, the mean number of hops between pairs of nodes in the MANET would almost be doubled, leading to an increased number of transmissions in the network and a lower end-to-end traffic capacity. Second, nodes close to the transmission area edge are treated the same way as nodes 120 m away from the edge, despite that the latter nodes have a very low probability of a link break compared to the nodes located close to the transmission area edge. The optimal buffer zone is to be found as a tradeoff between these effects, and thus somewhere in the range between 0 and 120 m in this particular example.

4.2 Zone routing algorithm

The buffer zone solution is based on defining nodes as safe or unsafe, and either using them as relay nodes, in case they are safe, or avoiding them as relay nodes in case they are unsafe. Also, traffic to unsafe nodes inside the sending node's transmission area should be attempted relayed through safe nodes, if possible.

The signal strength of the Hello packets can be used as parameter to be able to determine which nodes are in what is considered the safe zone and the unsafe zone with varying mobility speeds. However, other means of determining this are also possible, including the use of GPS.

The zone status of each neighbor must be added to each link entry in the Hello packets and announced to the other neighbors, in order to support neighboring nodes in routing traffic to its unsafe neighbors. It is necessary to avoid routing a packet to a relay node which has the destination as an unsafe neighbor, if the source node also has the destination node as an unsafe neighbor.

Fig. 1 compares standard OLSR routing to OSLR routing with the proposed buffer zone algorithm. The dashed arrow shows the normal routing, where all packets from A to C are transmitted directly to node C. This makes the transmission vulnerable to mobility in case node C moves away from node A. The continuous arrows show the packet path using the zone algorithm, where traffic from node A to node C is routed via node B, because node C is in the unsafe buffer zone of node A. This means that the traffic path is not vulnerable to node C moving out of the transmission area of node A.

The routing table of each node is first calculated based only on nodes in the safe zone, and if this leads to partitioning, routes via nodes in the unsafe buffer zone are included in the routing table. The principle of buffer zone routing is to only use nodes in the safe zone to forward traffic. The nodes in the unsafe buffer zone should only be used for forwarding if it is impossible to obtain full connectivity without them.

As the neighbor set, two-hop neighbor set and topology set are traversed, no route updates to the already defined routes are allowed. This means that if a node already is represented in the routing table as a destination, the newly found route to the same destination is discarded, even if it is of fewer hops than the first route. The steps of the buffer zone routing algorithm are shown in Table 2.

5 Evaluation of the buffer zone solution

5.1 Outline of the evaluation

This section presents the key behavior of the buffer zone routing algorithm, and compares it to the behavior of standard OLSR. For all graphs where the x-axis represents the threshold between the safe and unsafe zone, the results at 250 m threshold correspond to the complete transmission area being the safe zone. Thus, the buffer zone results at a threshold of 250 m are equal to the performance of standard OLSR without the buffer zone solution. (In fact, the latter is not entirely true, as the buffer

Table 2. The zone routing algorithm

1.	Clear routing table.
2.	Add route to all neighbors (1st time: only neighbors in the safe zone).
3.	*Add 2hop neighbors*
3.1.	Add route to all 2hop neighbors that both are in the safe zone of the relaying neighbor and where the neighbor is in the routing table.
3.2.	Add route to all 2hop neighbors that are this node's neighbors with direct route.
3.3.	Add route to all 2hop neighbors in the unsafe zone of their neighbor while the neighbor is in this node's safe zone.
3.4.	On 2nd iteration: Add 2hop neighbors in the unsafe zone of their neighbor while the neighbor is in this node's unsafe zone
4.	Add route to all topology tuplets with increasing hop count.
5.	If first time, return to step 2, else exit.

zone algorithm has a marginally higher routing overhead, as will be shown in Section 5.3.)

To give a fair comparison between the buffer zone routing and standard OLSR, not only is the buffer zone routing compared to standard OLSR at a transmission range of 250 m. It is also compared to standard OLSR with a reduced range of next hop neighbors (see the curves marked '*Hello_discard*'). This way it can be ensured that the advantages of the buffer zone routing do not stem from some effects of reducing the range of next hop neighbors, but rather from dividing the transmission area into a safe zone and an unsafe zone. Thus, the graphs marked as '*Hello_discard*' imply a simple discard mechanism for the Hello packets at thresholds below 250 m, preventing the Hello packets received from nodes in the unsafe zone from being processed, but allowing reception and acknowledgement of data packets. In addition, the buffer zone algorithm is compared to an implementation of standard OLSR configured with an overall reduced transmission range (see the curves marked '*Reduced_tx_range*'), where the effect of reducing the reception radius for both data traffic and control traffic is shown.

In the following, first results with low traffic load (i.e. 50 kbps of traffic in total) and high node mobility (i.e. 10 m/s velocity for each node) are presented. Both the goodput, loss, average path length and routing load are investigated. After having gained insight about the performance at high mobility, the goodput results are compared with results for low mobility. Then, these low traffic load results are compared with similar results generated at a high traffic load (i.e. of 500 kbps of traffic in total), for both

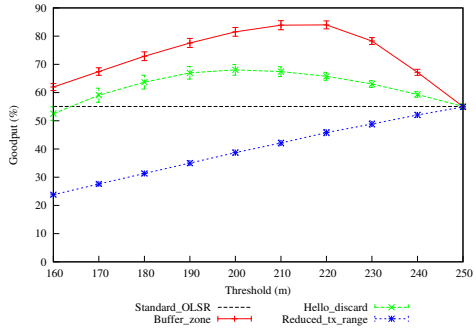


Figure 5. Goodput with node speed 10 m/s and 50 kbps system load.

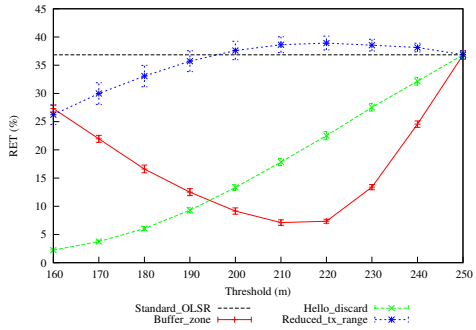


Figure 6. RET loss. (Loss caused by MAC maximum retransmissions discards.)

low and high mobility. Finally, the goodput when LLN is implemented is explored.

5.2 Exploring the benefits of the buffer zone solution

Fig. 5 shows the goodput results with a relatively high node velocity of 10 m/s and a light total traffic load of 50 kbps. Comparing the results of the buffer zone algorithm at lower thresholds than 250 m to the result of standard OLSR (which is equal to the result of the buffer zone algorithm with a threshold of 250 m), the gain of using the buffer zone algorithm over standard OLSR is $84\% - 55\% = 29\%$ at 220 m. Fig. 6 indicates that the main advantage of the buffer zone algorithm stems from the fact that the retransmission (RET) loss is considerably reduced compared to the RET loss of standard OLSR.

One could on the other hand suspect that the advan-

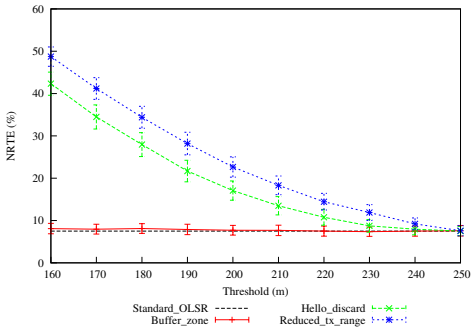


Figure 7. NRTE loss. (Loss caused by lack of route.)

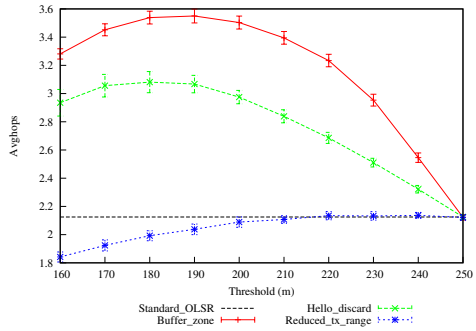


Figure 8. Average number of hops.

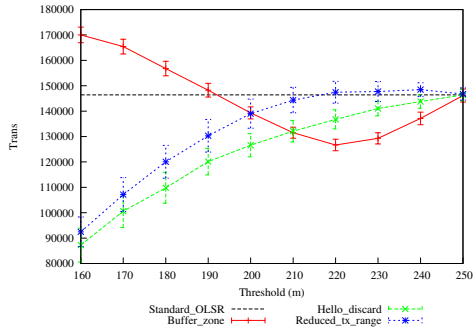


Figure 9. Total number of MAC transmissions including retransmissions.

tages of the buffer zone routing stems from some effects of reducing the range of next hop neighbors. However, Fig. 5 shows that by letting standard OLSR discard Hello packets at a threshold lower than 250 m (*Hello_discard*), it still performs worse than the buffer zone solution. This is mainly due to the increased probability of partitioning caused by the discard of Hello packets. This again results in more packets being lost due to lack of route (NRTE loss), as observed in Fig. 7. The figure shows that the probability of partitioning for the *Hello_discard* method increases with a decreasing threshold distance.

Nevertheless, Fig. 5 shows that even the simple *Hello_discard* method has a higher goodput than standard OLSR. The reason is that the discard of Hello packets forces the routing protocol to use shorter links. A bulk of link breaks is then avoided, because nodes outside the discard zone still can receive the packet and reply with an acknowledgment before the neighbor moves beyond the transmission radius. The RET loss is therefore reduced also for the *Hello_discard* method (Fig. 6). This benefit outweighs the disadvantage of a higher probability of partitioning, leading to a totally higher goodput than the goodput of standard OLSR (Fig. 5).

Reducing the transmission range itself (i.e. the *Reduced_tx_range* method) offers no buffer zone outside the threshold where nodes that have moved outside the threshold can continue to communicate. Therefore, a clear advantage of a reduced RET loss compared to that of standard OLSR is not observed in Fig. 6. Instead, the *Reduced_tx_range* method performs worse than OLSR (Fig. 5). The main reason is that the reduced transmission range causes more network partitioning as the number of neighbors is reduced, leading to the high NRTE loss observed in Fig. 7. The loss is even higher than the loss of the *Hello_discard* method, because TC messages from nodes outside the threshold are also discarded.

5.3 Exploring the costs of the buffer zone solution

Having identified a reduction in the RET loss as a main benefit of the buffer zone algorithm, it is interesting to explore the cost of this solution.

There was no difference between the buffer zone solution and standard OLSR in term of packets being lost due to lack of route (Fig. 7). Thus, the buffer zone algorithm does not increase the chances of network partitioning compared to standard OLSR. This is expected, since the buffer zone algorithm is forming links to neighbors in the buffer zone whenever necessary.

However, there is a difference between the buffer zone solution and standard OLSR in terms of the mean number of hops between a source and a destination. The number of hops per path (Fig. 8) is increased with the buffer zone solution, as it favors nodes in the safe zone as relay nodes. The increased hop length is a main disadvantage of the buffer zone solution.

First, the increased hop length leads to an increased number of needed transmissions for the same end-to-end

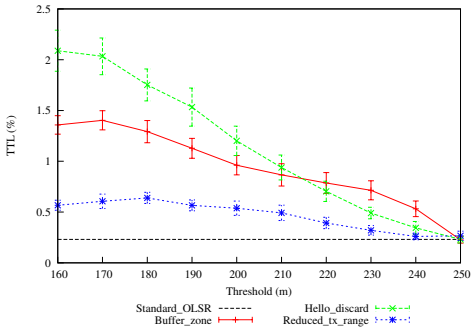


Figure 10. TTL loss. (Loss caused by exhausted time to live.)

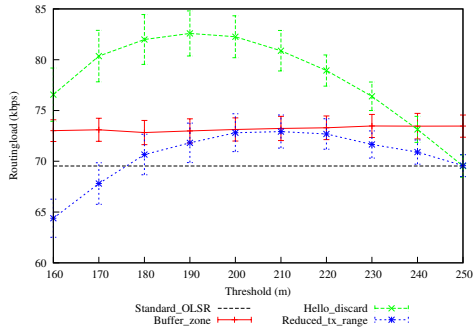


Figure 11. Routing load.

traffic streams, thus reducing the total available capacity per traffic stream. However, since the buffer zone solution reduces the number of packets lost due to RET, the total number of transmissions (Fig. 9) is actually lower for the thresholds above 190 m.

Second, as the paths get longer, there is an increased risk that the topology information held by the forwarding nodes is wrong. There is a higher probability that the topology (both the real topology and the topologies given by the routing tables) changes while the packet is in transit between the source and destination nodes. Thus, the risk of a packet loop, or a considerable detour, is increased. Both an increased average path length, an increased risk of a packet detour and an increased risk of a packet loop add to the probability of a Time-To-Live (TTL) exhaustion. Indeed, the ratio of packets being discarded because of exhausted Time To Live (i.e. TTL loss) is higher for the buffer zone algorithm than for standard OLSR (Fig. 10).

One might argue that the increased TTL loss of the buffer zone algorithm is quite small (e.g. only 1% at a threshold of 200 m). However, the main problem with packets discarded by TTL is that they are transmitted extensively, and at least a number of times equivalent to the original TTL value set by the source node. As all source nodes in the simulations set the TTL value of the packets they are originating to 32, it means that at a threshold of 200 m, 1% of all sent packets have been transmitted at least 32 times. All these transmissions ultimately proved to be worthless. Thus, even a low percentage of TTL loss might represent a large and unnecessary consumption of the network resources.

In addition to a higher mean path length, the cost of the buffer zone solution also includes a higher routing load, in terms of a higher payload overhead of the Hello messages. The reason is that the buffer zone solution depends on publishing the zone status of the neighbor nodes in the Hello messages. At a 250 m threshold the increase in the routing load for 40 nodes at 10 m/s is around 3 kbps (Fig. 11), which is quite low compared to the overall network capacity. In fact, for simplicity the zone status field in the Hello message was implemented in the simulator as one extra byte for each address in the message, although the information – safe zone or buffer zone – could have been represented by only one bit per address. Furthermore, only symmetric links need this information, so the zone information could have been skipped altogether for the asymmetrical links. Thus, the additional routing overhead of the buffer zone algorithm of 3 kbps might be made much smaller in an optimized implementation. Nevertheless, when it was stated above that the buffer zone algorithm has equal performance and behavior as standard OLSR when the threshold is set to 250 m, this is not exactly true, due to the marginal extra routing overhead of the buffer zone algorithm.

Interestingly, the routing overhead of the *Hello_discard* and *Reduced_tx_range* methods is increasing when the threshold decreases from 250 m. The reason is that these methods require an increased number of multi-point relay (MPRs) nodes – nodes that generate and forward TC messages in the network – as the threshold decreases. This leads to higher routing overhead. However, as the threshold gets even lower, the routing load starts to decrease again. This is due to the increased probability of network partitioning observed at low thresholds (Fig. 7).

5.4 The impact of node mobility

Reducing the node velocity reduces the number of occurring link breaks. Since the buffer zone solution is aimed at reducing the number of packets lost due to link breaks, it is expected that the advantage of the solution is decreasing with decreasing mobility. However, the buffer

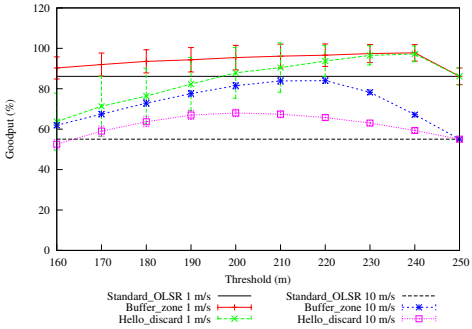


Figure 12. Goodput for 1 and 10 m/s at 50 kbps load.

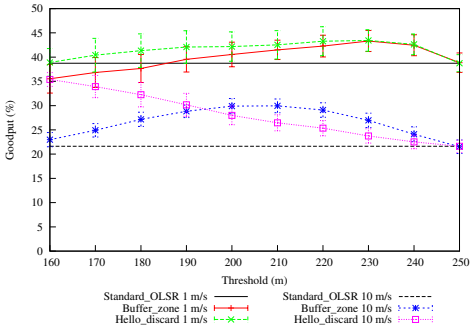


Figure 13. Goodput for 500 kbps load.

zone solution still provides a gain in throughput also at a low node velocity of 1 m/s (Fig. 12), obviously because the performance cost of the buffer zone solution is quite low. For the lowest level of mobility (i.e. 1 m/s) the optimum goodput of the buffer zone solution is over 95% at 240 m threshold, providing a gain of $95\% - 85\% = 10\%$. The *Hello_discard* mechanism also achieves this gain, but performs worse at lower thresholds due to partitioning.

With reduced velocity, the probability of a link break due to a neighbor moving out of the transmission area is lower, simply due to a lower node speed. Therefore, the threshold range can easily be set higher to achieve the same advantage, while the disadvantage of increased path lengths is reduced. It is expected that the optimal threshold range in terms of maximized goodput is increasing with decreasing node mobility, and that the optimal threshold range is 250 m when the node mobility is zero. This expectation is supported by the simulation results. The optimal threshold range is 240 m at a node velocity of 1 m/s and only 220 m for a node velocity of 10 m/s.

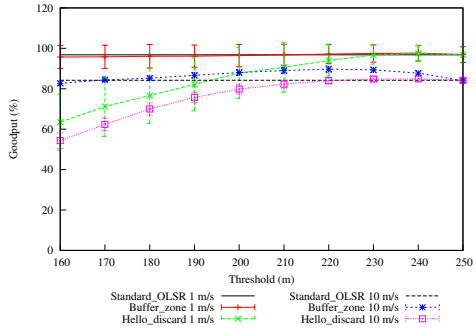


Figure 14. Goodput with LLN.

5.5 The impact of traffic load

The buffer zone algorithm increases the throughput compared to the Standard_OLSR results even for high traffic loads (Fig. 13). The relative gains of the buffer zone solution as a percentage of the goodput of the standard OLSR in the high load results are comparable with the relative gains of the low load results in Fig. 12. However, the total gain of the buffer zone solution is naturally lower for such high traffic loads. The reason is that the entire network is stressed with a large number of unnecessary transmissions and increased packet loss, leaving a lower share of the total network capacity to the successfully transmitted traffic, be it traffic routed by standard OLSR or traffic where the buffer zone algorithm is used.

For thresholds lower than 190 m, the *Hello_discard* algorithm yields better throughput than the buffer zone solution (Fig. 13), because the *Hello_discard* algorithm increases the partitioning of the network, and this reduces the load. At the same time it mends the link-break-induced retransmission problem, and the combination of partitioning and reduced retransmissions in the event of link breaks makes the throughput higher for the lower thresholds. However, due to the partitioning, the packets are highly probable to be traveling only a few hops, thus increasing the unfairness between the short path and the long path traffic.

5.6 The buffer zone solution with LLN

Even with LLN, the buffer zone algorithm gives better goodput than both the *Hello_discard* method and standard OLSR (Fig. 14). However, at very low node mobility (i.e. 1 m/s) the gain is marginal. Furthermore, the gain is considerably less than for the results without LLN (Fig. 12). The reason for the gain over standard OLSR is the fact that the buffer zone solution prevents the effects of the link break, as less traffic is routed over the link

when it breaks. This also helps the other neighboring nodes to be aware that the neighbor is in danger of being lost. When LLN is employed without the buffer zone solution, any neighboring nodes must wait until the next Hello packet to get knowledge of the lost link. This could be up to 2 seconds after the link break has occurred. With the buffer zone solution, on the other hand, the neighbors are aware in advance that the node is unsafe.

5.7 Discussion

The presented results show that the buffer zone solution offers a throughput gain over both the standard OLSR routing protocol and the simpler *Hello_discard* algorithm, and the buffer zone solution is robust to variations, with both varying node velocity and traffic load. The throughput gain comes from the reduced packet loss caused by maximum MAC retransmissions (RET). The RET loss is reduced because the buffer zone solution prefers neighbors in the safe zone as relay nodes, and since these links are less apt to break, less links that break are in active use.

The optimal threshold range is affected by mobility. At 1 m/s node speed the buffer zone solution delivers high goodput in a broad threshold range, while for 10 m/s the threshold range where the throughput gain is greatest is much smaller, and focused around 200-220 m. Thus, if a static zone threshold is preferred, 210 m could be a good compromise. However, the threshold could also be set dynamically, based on one or several various parameters such as mobility, safe-to-unsafe nodes ratio, link break probability based on learning, etc.

Signal strength was used as the parameter to classify the neighbor nodes as either safe or unsafe. Because the TwoRayGround propagation model was used, the distance to each neighbor was available through the signal strength parameter, and this made the prediction and classification of which neighbors that were safe or unsafe straightforward. Signal strength is however a parameter that may only be of high value in a simulator environment. Other research, such as [8] and [7], has shown that in real experiments the signal strength is highly variable and must be filtered over many samples to provide a trustworthy value. Likewise, the transmission range threshold – where the packet loss by only moving a centimeter at the edge of the transmission area instantly goes from full to nothing – is not equal to the real world experience. Another problem with using the signal strength is illustrated through the problem of gray zones [13]. Since broadcast and unicast packets are sent with different transmission rates, the signal strength of a Hello packet, which is broadcasted, would not be directly transferable to the signal strength of a unicast packet from the same node at the same distance.

Instead of signal strength, other parameters could have been used, such as geographical position, packet loss, the number of errors corrected with forward error correction (FEC), or MAC retransmissions. These would not give the same distance precision as has been achieved through the signal strength of Hello packets. However, the buffer zone algorithm could also be seen as independent from the distance and mobility perspective that has been used in this paper. Through classifying nodes as either safe or unsafe, based on for example the number of MAC retries for the last number of transmitted packets, the classification could instead determine what links are more reliable. Using the same algorithm for constructing the routing table, the result would perhaps be enhanced throughput. This should be researched further.

6 Related work

This paper is a follow up of [15], where the retransmission problem is investigated thoroughly in relation with the interface queue size. However, there exist several other works dealing more directly with the problem of link breaks due to mobility and proposing solutions to mend this problem. Qin and Kunz [16], for example, present a solution to detect and mend the effect of mobility through evaluating the rate of link breaks.

Many solutions focus only on reactive routing protocols, where the rerouting overhead reduction potential is high. These include [10], which is based on GPS location information, along with [11], [14] and [5], all based on signal strength evaluation. In fact, the solution by Goff et al. [11] bears some resemblance to the *Hello_discard* algorithm presented in this paper, with a *preemptive region* comparable to the unsafe buffer zone.

On the other hand, Su, Lee and Gerla [17] present a mobility prediction solution implemented for both proactive and reactive protocols, where both GPS and signal strength are proposed used to establish the positions and relative distances of network nodes. A link break prediction table is presented in [6], to be utilized by both reactive and proactive routing protocols.

Ali et al. [3] propose to use signal strength with hysteresis in OLSR to both anticipate link breakages and avoid establishing links that are transient. Fast-OLSR [4] is another modification to OLSR where the Hello interval is varied with the degree of mobility.

7 Conclusions and further work

The introduction of a transmission buffer zone in OLSR gives improved throughput compared to standard OLSR (or compared to no buffer zone, which is approximately equal to standard OLSR). The advantage of using a buffer zone is observed both for low and high traffic loads.

A too large buffer zone, however, leads to an unnecessary higher mean number of hops between pairs of nodes in the MANET and a higher probability of network partitioning. Thus, the size of the buffer zone should be optimized.

The optimal size of the buffer zone (which is given directly by the optimal threshold range of the buffer zone algorithm) is increasing with increasing node mobility. At no node mobility, the optimal size of the buffer zone is zero, assuming that all link breaks are caused by mobility. However, in a realistic network scenario where link breaks are also caused by changing radio conditions, it is reason to believe that the buffer zone algorithm is useful also at no mobility.

Finding a means to estimate the optimal size of the buffer zone, depending on parameters such as node mobility and network load, is an important issue for future work. Furthermore, it is also a need for investigating the buffer zone algorithm with a more realistic radio model than used in this paper, i.e. both using a better radio channel model and investigating scenarios where link breaks are caused by changing radio conditions. Finally, the presented buffer zone algorithm can be improved and extended, using other criteria apart from distance to classify neighbor nodes as safe or unsafe.

8 Acknowledgment

This work was supported by the ITEA Easy Wireless and CELTIC DeHiGate projects.

References

[1] Manet simulation and implementation at the university of murcia (masimum). http://masimum.dif.um.es/.

[2] Network simulator 2 - ns2. http://nsnam.isi.edu/nsnam/.

[3] H. M. Ali, A. M. Naimi, A. Busson, and V. Vèque. An efficient link management algorithm for high mobility mesh networks. *Proceedings of the 5th ACM international workshop on Mobility management and wireless access (MobiWac)*, pages 42–49, 2007.

[4] M. Benzaid, P. Minet, and K. Al Agha. Analysis and simulation of fast-olsr. *The 57th IEEE Semiannual Vehicular Technology Conference (VTC)*, 3:1788–1792, April 2003.

[5] G. Chauhan and S. Nandi. Qos aware stable path routing (qasr) protocol for manets. *First International Conference on Emerging Trends in Engineering and Technology (ICETET)*, pages 202–207, July 2008.

[6] M. Chegin and M. Fathy. Optimized routing based on mobility prediction in wireless mobile adhoc networks for urban area. *Fifth International Conference on Information Technology: New Generations (ITNG)*, pages 390–395, April 2008.

[7] K.-W. Chin. The behavior of manet routing protocols in realistic environments. *Asia-Pacific Conference on Communications*, pages 906–910, Oct. 2005.

[8] K.-W. Chin, J. Judge, A. Williams, and R. Kermode. Implementation experience with manet routing protocols. *SIGCOMM Computer Communication Review*, 32(5):49–59, 2002.

[9] T. Clausen, P. J. (editors), C. Adjih, A. Laouiti, P. Minet, P. Muhlethaler, A. Qayyum, and L.Viennot. Optimized link state routing protocol (OLSR). RFC 3626, pages 1–75, October 2003. Network Working Group.

[10] S. Crisostomo, S. Sargento, P. Brandao, and R. Prior. Improving aodv with preemptive local route repair. *International Workshop on Wireless Ad-Hoc Networks*, pages 223–227, 2004.

[11] T. Goff, N. B. Abu-ghazaleh, D. S. Phatak, and R. Kahvecioglu. Preemptive routing in ad hoc networks. *Proceedings ACM/IEEE MobiCom*, pages 43–52, 2001.

[12] IEEE. Wireless LAN medium access control (MAC) and physical layer (PHY) specification. IEEE standard 802.11, June 1999.

[13] H. Lundgren, E. Nordströ, and C. Tschudin. Coping with communication gray zones in ieee 802.11b based ad hoc networks. In *Proceedings of the 5th ACM international workshop on Wireless mobile multimedia (WOWMOM)*, pages 49–55, New York, NY, USA, 2002. ACM.

[14] L. Meng, J. Zang, W. Fu, and Z. Xu. A novel ad hoc routing protocol research based on mobility prediction algorithm. *Proceedings of International Conference on Wireless Communications, Networking and Mobile Computing*, 2:791–794, Sept. 2005.

[15] V. Pham, E. Larsen, K. Ovsthus, P. Engelstad, and O. Kure. Rerouting time and queueing in proactive ad hoc networks. In *IEEE International Performance, Computing, and Communications Conference (IPCCC)*, pages 160–169, 2007.

[16] L. Qin and T. Kunz. Mobility metrics to enable adaptive routing in manet. *IEEE International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)*, pages 1–8, June 2006.

[17] W. Su, S.-J. Lee, and M. Gerla. Mobility prediction and routing in ad hoc wireless networks. *International Journal of Network Management*, 11(1):3–30, 2001.

Paper D :

Preemption Mechanisms for Push-to-Talk in Ad Hoc Networks

E. Larsen, L. Landmark, V. Pham, P. E. Engelstad, and Ø. Kure

In proceedings of the 34th IEEE Conference on Local Computer Networks (LCN), Zürich, Switzerland, October 20–23, 2009, pp. 428–435, ISBN: 978-1-4244-4488-5.

Preemption Mechanisms for Push-to-Talk in Ad Hoc Networks

Erlend Larsen*, Lars Landmark*, Vinh Pham*, Paal E. Engelstad† and Øivind Kure*
*Q2S NTNU
†SimTel (Telenor/Simula)
Email: erl@unik.no, larsla@q2s.ntnu.no, {vph, paalee, okure}@unik.no

***Abstract*—Using push-to-talk applications in ad hoc networks is not straightforward. There are no inherent mechanisms to support priority of the voice traffic, to avoid great jitter and packet loss in face of large background traffic loads. This paper presents three preemption mechanisms that can be applied to support push-to-talk traffic in multi-hop ad hoc networks. The mechanisms differ in the way the background traffic is treated: discard, buffering and inter-scheduling. It is shown that there is a trade-off between the impact on the background traffic and the service for the push-to-talk traffic. Discarding or buffering the background traffic leaves the push-to-talk traffic with very little impact by the background traffic, while inserting the low priority packets in the interval between the high priority packets incurs some cost to the push-to-talk traffic.**

I. Introduction

The coordination of emergency and military operations has traditionally been done by Push-to-Talk (PTT) using two-way radio transceivers (walkie-talkies). With a single push on a button, the user switches from voice reception mode to transmit mode. PTT is a half-duplex method, meaning that when a sender transmits, it is unable to hear other radios transmitting at the same time. This inability to interrupt has made PTT best suited for quick communication exchanges between users.

Contrary to the low capacity analog communication supported by walkie-talkie systems, Mobile Ad Hoc Networks (MANETs) are able to support high capacity digital communication in environments where no network infrastructure is available. Because of the important coordination function of PTT, it is vital to support this service also in MANETs. In some circumstances, the PTT service can mean the difference between life and death, e.g. when calling for support or alerting of immediate danger. The combination of inability to interrupt and different urgency of the PTT calls should be reflected by the network in terms of service priority.

Push-to-talk is a one-to-many service, i.e. there is one sender and several receivers. For such applications, multicast can be a more efficient distribution method than unicast. With unicast, each packet is forwarded to one single destination. For each receiver a unique packet must be created and forwarded. An advantage of multicast is that one packet transmission can be received by multiple nodes, and then forwarded by these nodes, distributing the information to the whole or larger parts of the network. However, this efficiency makes for less reliable link layer transmissions, due to more receivers per transmission, where unicast with one receiver can rely on acknowledgment of each packet.

Voice is a traffic type with high network service demands, especially in terms of delay/jitter and loss rate. In a MANET, the end-to-end voice communication is faced with several challenges, including interference, packet loss and congestion. In addition, multicast traffic is troubled by self-interference when a received packet is transmitted almost simultaneously by several forwarders.

This paper proposes three mechanisms based on preempting the lower priority traffic and compares these to the well known priority queuing mechanism. Through simulations it is shown how preempting the lower priority traffic can increase the network performance for the push-to-talk voice traffic.

The rest of the paper is structured in the following way. First, in Section II, the problem of using PTT in ad hoc networks without priority is explained. Second, in Section III, the well known priority queuing mechanism is presented. A first effective preemption mechanism is introduced in Section IV, and then two more gentle preemption mechanisms are investigated in Section V. In Section VI, the behavior of the preemption mechanisms is scrutinized when TCP is used as background traffic transport protocol. Related work is presented in Section VII, and finally, in Section VIII, the conclusions of this paper is presented.

II. Received network service for push-to-talk

The normal network behavior is investigated in this section, and then the results documenting the service problem for PTT in ad hoc networks are shown. First, the simulation setup is presented, and then the results documenting the impact of the background traffic on the network service are shown.

Ns-2 [1] version 2.33 was used to run simulations evaluating the mechanisms and solutions presented in this paper. Unless otherwise specified, the following settings were used for the simulations: The IEEE 802.11 MAC was used for medium access with 2 Mbps data rate and 1 Mbps basic rate. The interface queue size was 100 packets. The interference radius was 550 m, and the transmission radius was 250 m. OLSR was used as routing protocol, and link layer notification was enabled. The simulation topology was 30 nodes moving in a 1500 x 300 m^2 area using random direction with reflection mobility model at a constant velocity of 5 m/s generated by the tools in [2]. The nodes changed direction every 10 s $\pm$ 5 s,

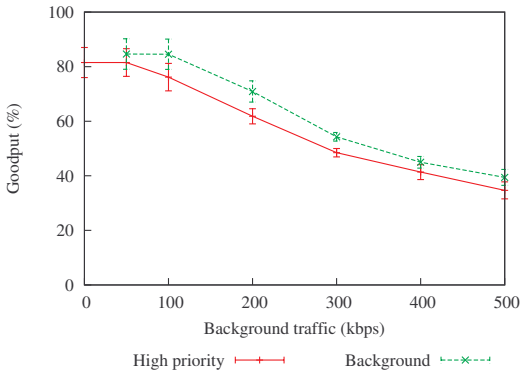


Fig. 1. Normal received network service for push-to-talk traffic with increasing background traffic load.

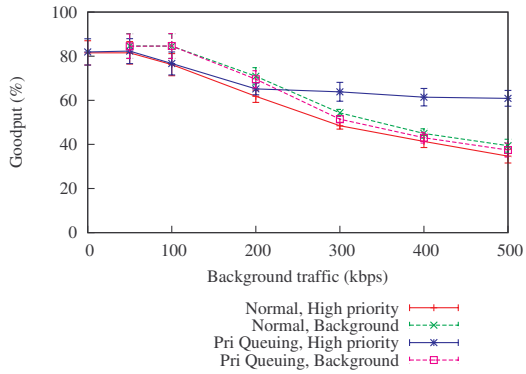


Fig. 2. Goodput with and without the queue priority mechanism.

and the simulations lasted 600 s. Each simulation configuration was run 10 times with different random seed (heuristic) and topology, and the same 10 random topologies were used for all simulations for comparison fairness. The confidence intervals are given with a confidence coefficient of 95%.

The traffic pattern was as follows: At 10 s, the background traffic starts. High priority traffic at constant bit rate 5.3 kbps in 20 byte multicast packets, with an interval of 30 ms, was transmitted in sessions of 5 seconds. This traffic rate corresponds to the G.723.1 voice encoding standard. Simplified Multicast Forwarding (SMF) [3], a mesh-based efficient flooding protocol, was used to forward the multicast traffic. The Source-specific Multi Point Relay (S-MPR) forwarding algorithm was used [4]. For each session, a new random node was designated as the sender, and a 5 s pause separated the sessions. The background traffic was unicast, where all the nodes were divided into two groups. Each node sent traffic to all other nodes in the same group. The packet size for the background traffic was 64 bytes, and UDP was used as transport protocol for both the multicast and the unicast traffic. Measurements were started at 60 s and continued until 590 s.

In the "Normal" case, where there are no mechanisms in place to enhance the service for the priority traffic, the goodput results with increasing background traffic (Fig. 1) show that the priority traffic without any competition from the background traffic manages 80% goodput. The 20% loss is caused by collisions and mobility. As the background traffic is introduced and increased, the priority traffic is impacted very negatively, because the collision rate increases and the interface queues begin filling up, causing tail drops. Preferably, the performance for the priority traffic should be kept at the same level as without any background traffic.

In the following sections, mechanisms that treat packets differently by the priority they are assigned, are analyzed. First, the well known queue priority mechanism is presented. Then, various preemption mechanisms are proposed and analyzed.

III. Priority Queuing

When packets have been processed by the routing layer, they are passed through the link layer and queued in the interface queue on the MAC layer. The interface fetches the packet at the head of the queue and transmits it. Normally, no difference is made between packets with different priority classification. This can result in high priority packets being inserted in the interface queue behind several low priority packets, and delayed considerably. It also risks being discarded by tail drop if the queue gets full.

A solution to this problem is to place the new packets in the queue according to priority, so that all the high priority packets are placed in front of any low priority packets. This mechanism is part of the interface queue behavior. Placing the high priority packets in front of the lower priority packets ensures that no high priority packets will be dropped from the queue before all the low priority packets have been discarded. Also, the high priority packets are not delayed by the low priority packets.

The cost of this mechanism is taken by the low priority traffic. The queue tail drops impact the low priority traffic first, since no high priority packets are discarded unless the queue only contains high priority packets. In fact, the low priority traffic risks starvation if there is capacity only for the high priority traffic. Also, the background traffic always has to wait until all higher priority packets have been transmitted, leading to higher delay and jitter.

Simulations with and without the priority queuing mechanism have been run. With the priority queuing, the goodput (Fig. 2) for all traffic initially drops the same way as without priority queuing. This is due to an increase in collisions for the high priority traffic. However, with the priority queuing, the high priority traffic stabilizes at a background traffic load of 200 kbps and higher. This is because above 200 kbps all new background traffic is lost due to queue tail drops. This is confirmed through observations showing that as the background load increases past 200 kbps, the background

traffic losses increase linearly. Without the priority queuing, the priority traffic starts experiencing tail drops as the load increases above 100 kbps, while using the priority queuing avoids all the high priority packet losses due to tail drop.

IV. Preemption by Discard

While the queue priority mechanism is able to limit the background traffic in the network as the network approaches congestion, background traffic will still be transmitted in the network. The nodes that are not originating or forwarding the high priority traffic are free to transmit background traffic. So are also the other nodes, after transmitting their high priority packets. Thus, the high priority traffic has to compete with the low priority traffic on the medium, and this reduces the available bandwidth for the high priority traffic.

Preempting the background traffic from the network when the high priority traffic is being sent, is a way to further prevent interference from the background traffic. A crude form of preemption is to discard all the lower priority traffic. Each node discards the background traffic instead of forwarding it during the high priority session. If the lower priority traffic is voice, it is sensible to discard the traffic, as it will not be heard at the receiving nodes, and only expend network resources.

The initialization and end phases are challenging with the discard mechanism. The initialization phase starts with the source of the high priority traffic beginning to send packets down through the routing layer into the interface queue. At the insertion into the interface queue, the preemption mechanism is activated on this first node, and all the lower priority packets are discarded until the preemption times out. It is important that priority queuing is activated, not delaying the high priority packets out of the source node. The high priority packet is transmitted on the medium, and the neighbor nodes hear the packet. Some of the neighbors also forward the packet further out in the network. As the nodes hear the high priority packet, the preemption is activated, and the node stops transmitting low priority traffic. The preemption mechanism should be implemented between the interface queue and the MAC layer, so that the low priority packets already in the interface queue can be easily dropped, instead of being transmitted. Finally, after the first few packets of the high priority traffic flow have been multicasted, all nodes have activated the preemption, and no lower priority traffic is transmitted any longer.

The end phase, when the high priority session is over, represents another challenge. It is difficult for any given node to determine the end of the session, unless the application sends out a disconnect notification. Using a timeout after the last received high priority packet is a simple way to detect the end of the high priority session, but this timeout must be larger than the space between two consecutive packets, since the packets may arrive with different delays, or may even be lost. Although the timeout value could be optimized through measuring the delay between packets, a one second timeout is suggested in this paper, for simplicity.

A benefit of employing the discard mechanism is that any competition between the high priority traffic and the background traffic is effectively avoided. Thus, the resulting goodput for the high priority traffic should be equal to (or close to) the goodput without any background traffic in the network. The cost, on the other hand, are all the discarded background packets that would otherwise have reached their destinations, including traffic already en route to the destination. In addition, since the network is not being fully utilized during, and immediately after, the high priority session, a lot of network resources are wasted.

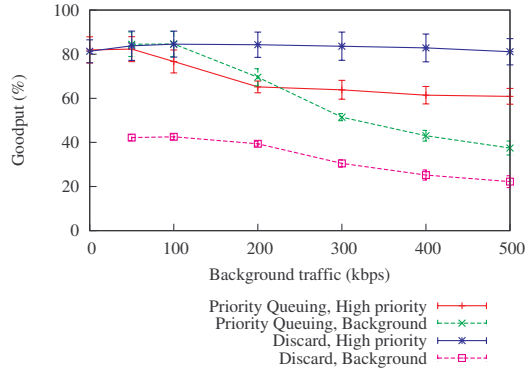


Fig. 3. Goodput for the discard mechanism.

The preemption mechanisms are dependent on priority queuing to work as proposed. Therefore, these mechanisms have been simulated using priority queuing and are compared to the priority queuing results. Looking at the results for increasing background traffic (Fig. 3), the discard mechanism is very efficient in keeping the high priority goodput at the same level as without the background traffic, but it does so at the expense of the background traffic. Using the discard preemption mechanism, the background traffic sees a reduction from over 80% goodput with the priority queuing, to well below 40%.

Due to the pauses between the high priority traffic sessions, the goodput for the background traffic is as high as 35%. The background traffic goodput is completely relient on pauses in the high priority traffic, and the gain in the priority traffic goodput comes at the expense of the background traffic. Thus, there is no definite answer to what the better mechanism is, when comparing the discard and priority queuing mechanisms, since it is dependent on how much one is willing to penalize the background traffic, to achieve lower loss and delay for the priority traffic.

V. Improved Preemption Mechanisms

It may be difficult to choose either preemption or only priority queuing, since it depends on how the importance of the lower priority traffic is weighed. In this section, attempts are made to find a “both ways” solution, which gives good priority for PTT and, at the same time, low consequences for the

background traffic (i.e. mechanisms that help the background traffic).

A. Buffering of the background traffic

Instead of discarding the lower priority packets set for transmission during a high priority session, these packets could be held back in the interface queue until the end of the session, and then transmitted. This mechanism is called preemption with buffering. The issues with initializing and ending the preemption are the same as for the discard mechanism. The priority queuing is also required to keep any higher priority packets from being held back in the interface queue along with the lower priority packets, when the preemption is in effect.

An advantage of the buffering mechanism is that the low priority packets created during the high priority session can be transmitted after the session ends, instead of this information being lost. Thus, the goodput would increase. Another advantage is that any packets caught by a high priority session while en route to the destination would not be discarded. These packets have already spent network resources to come somewhere along the path, and so to discard the packets would be to waste these resources.

The mechanism will work best with a low load for the low priority traffic, with a short high priority session and with a large queue, since the queue may fill up fast if the load is high, or if the priority session lasts for a long time. Packets arriving after the queue is filled up will have to be discarded, and the advantage of the buffering approach is reduced.

When packets buffered in the queue have to be discarded, a choice must be made to either discard the oldest or the newest low priority packets. Discarding the newer packets would spare the packets already on the way to the destination, as the forwarding nodes may produce new packets, filling the queue. Discarding the old packets would mean losing the gain of storing already forwarded packets. However, protocols relying on packet acknowledgments, such as TCP, would benefit from discarding the old low priority packets, since these probably would time out before the high priority session is finished.

The buffer mechanism may hold the low priority packets for extensive periods of time. The mobility can cause the next hop of a low priority packet to be gone by the time the packet is due to be transmitted. A packet held in the buffer for a longer period of time will be more susceptible to have lost its next hop. This calls for a mechanism to enable route lookup and next hop insertion right before transmission, instead of the regular way of doing this before inserting the packet into the interface queue. This mechanism is explained below.

B. Ingress queuing

All packets to be transmitted by an ad hoc node have to be assigned a next hop by the routing protocol. The normal function is that the routing protocol assigns the packet a next hop, and then puts it in the interface queue. Here, the packet may stay for quite some time, before it reaches the head of the queue. If there is mobility in the network, there is a chance that the next hop assigned by the routing protocol is no longer reachable. The problem increases with higher mobility and with more packets in the queue. This forms a vicious circle where more packets in the queue leads to even more packets in the queue, only limited by the queue length.

Ingress queuing [5] remedies this problem through queuing the packets to be forwarded before assigning them a next hop. The next hop decision is taken as the packet is about to be transmitted, instead of doing so before the queue insertion. Thus, the packet is routed using the current topology information, instead of at the time of queuing.

The ingress queuing mechanism works only on unicast packets, since multicast and broadcast packets are assigned a special broadcast next hop. However, indirectly it works to the advantage for the multicast traffic, since the reduction of the number of unicast packets suffering retransmissions reduces the risk of collisions for the multicast traffic.

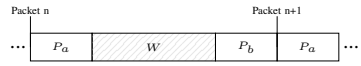


Fig. 4. Preemption with a window to transmit the low priority packets.

C. Low priority window

The Low Priority Window (LPW) preemption is the third proposed mechanism for preempting the background traffic, enhancing the buffering mechanism above. Considering that the high priority packets move in waves throughout the network, the interval between any two consecutive high priority packets could be used to transmit the low priority traffic.

The preemption is initialized in the same way as with the previous preemption mechanisms. Upon hearing a high priority packet, the nodes start buffering the low priority packets. The window for transmitting the low priority traffic is determined based on the time of the last received high priority packet. Fig. 4 shows the receival times of the packets n and $n+1$, and the low and high priority time spans surrounding them. Before the packets is a time span P_b, wherein the high priority packet is being received by the upstream node. No low priority traffic should be sent in this period. After the high priority packet is received by the current node, there is a time span P_a, wherein the high priority packet is forwarded by the downstream node(s). Here too, no low priority traffic should be sent. The P_a and P_b time spans are hereafter referred to as "guard windows".

After the guard window P_a is the low priority window W. During this time span, until the period P_b starts, the low priority traffic can be transmitted in the network. For each new received high priority packet, the expected time for the next high priority packet is estimated, and the low priority window is set accordingly. This continues until no new high priority packet has been received for an extended period (one second).

A premise for this solution to work is that the high priority packets are transmitted at a relatively constant interval, and that the interval between each high priority packet is so large that jitter does not cause packets to be received out-of-order

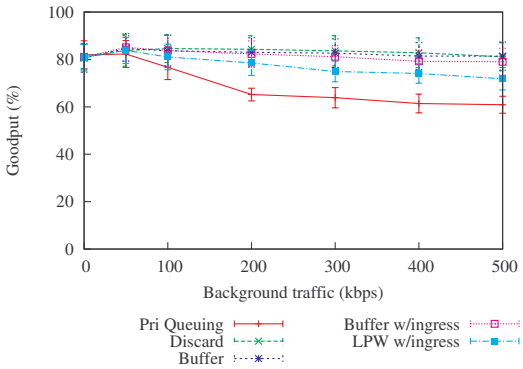


Fig. 5. Priority traffic goodput for the improved preemption mechanisms.

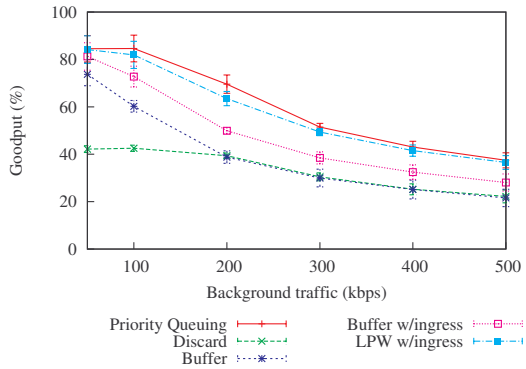


Fig. 6. Background traffic goodput for the improved preemption mechanisms.

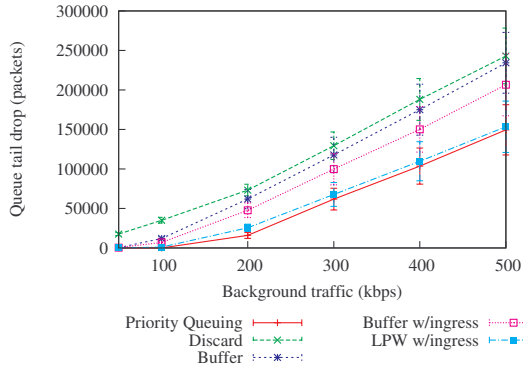


Fig. 7. Interface queue loss for the improved preemption mechanisms.

at any destination. Packets received out-of-order would mean that the space between any two packets cannot be depended on for use by the low priority traffic, as a high priority packet may be transmitted in the neighborhood at any given time.

The LPW preemption mechanism faces several challenges in calculating the receival of the next high priority packet, thus identifying the start of the P_b time span: First and foremost, it is necessary to know the rate at which the packets are sent from the source. This can either be calculated from an average of the incoming packets, or it can be known through a predetermined codec selection and hard-coded before the network is started up.

Second, it is necessary to detect lost packets. If all the packets of the same flow have an incremental sequence number, it is easy to detect a missing packet. Another way of detecting lost packets is to compare the time of the incoming packets with the known interval between the packets, and see if there is a gap considerably larger than the expected gap between two consecutive packets.

A third challenge is to cope with the jitter between the packets. Jitter can be observed in terms of the variation in delay between any two consecutively received packets. The determination of when the LPW is in effect is done locally, based on the time of the received high priority packet. Thus, the dissemination delay is not a problem, although it leads to nodes operating with local LPWs different from each other. Both the delay and the jitter grow larger for each hop the packet is sent outwards from the sender. Therefore the mechanism is best suited for networks of limited size.

D. Evaluation of the improved preemption mechanisms

The priority traffic achieves a goodput without any background traffic at around 80% with the buffer mechanism (Fig. 5), and manages to maintain this goodput as the background traffic increases. 80% is the same performance as the discard mechanism. The difference is that the background traffic goodput is increased, compared to the discard mechanism (Fig. 6). As long as the background load is low, any low priority packets are buffered until the high priority session is over, and then transmitted. However, for the buffer mechanism more and more packets are lost due to tail drop as the background traffic load increases (Fig. 7). Thus, the low priority goodput is reduced, compared to the priority queuing results.

The results for the buffer mechanism with the ingress queuing (Fig. 6) show that using the ingress queuing increases the background traffic results by some 10%. With the buffer mechanism and ingress queuing, the number of tail drop losses (Fig. 7) is reduced compared to without the ingress queuing, since the packets already in the interface queue are assigned a more correct next hop.

The LPW preemption mechanism has been simulated with the ingress queuing enabled, since the ingress queuing clearly has a positive impact on the background traffic performance. The P_a and P_b guard windows enclosing the high priority packet transmission (Fig. 4) have for simplicity been assigned the same size in the simulations at 10 ms each, leaving 10 ms for the LPW transmissions (30 ms high priority packet interval). In reality, the P_a window, which protects the medium

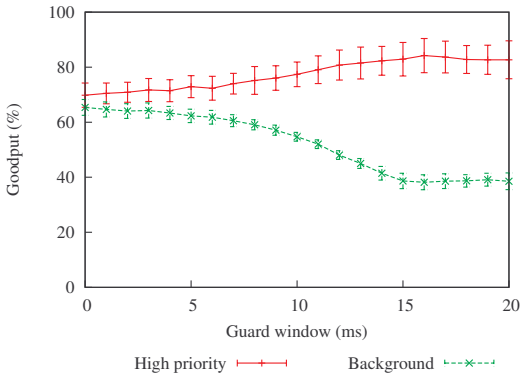


Fig. 8. Goodput with an increasing guard window.

after an actually received packet, is less, since the beginning of the window is determined directly by the receival of the packet. Thus, jitter can only affect the end of this guard window, while P_b, ensuring that the upstream high priority sender and forwarders can use the medium without any interfering low priority traffic, is more exposed to jitter, and should as such be larger than P_a. However, the difference is considered negligible in our study.

The LPW mechanism yields high priority traffic results (Fig. 5) better than using only the priority queuing, but worse than using the buffer or discard preemption mechanisms. It is clearly affected by the increasing background traffic, dropping around 8% when the background traffic is increased from zero to 500 kbps. At the same time, the background traffic goodput is increased compared to both the buffer and discard mechanisms.

However, it is observed that the jitter for the high priority traffic is kept very low (0.01 s), compared with using only the priority queuing (>0.1 s), making the buffer mechanism better suited to support PTT than by using only the priority queuing. Thus, the LPW preemption mechanism could be suited for networks where the PTT traffic can tolerate some loss to accommodate the lower priority traffic.

To better understand the behavior of the LPW mechanism, simulations have been run where the guard window on each side of the packet is increased by one ms at a time with 200 kbps background traffic load. These results (Fig. 8) show that there is no clear optimum where the guard window is perfectly matched to achieve the maximum goodput for both the high priority and the background traffic. The high priority traffic goodput reaches maximum with a 15 ms guard window. This is expected, as the two guard windows P_a and P_b at 15 ms leave no window W where low priority traffic can access the network (the interval between the high priority packets was 30 ms).

The results show that preemption with buffering and the ingress queuing can be used to maintain the same performance for the high priority PTT traffic, but avoid the negative impact on the background traffic caused by preemption with discard. If a somewhat lower goodput for the high priority traffic is accepted, however, LPW with the ingress queuing could be preferred instead. Then the background traffic will not suffer the losses that it does with the discard mechanism.

TABLE I
GOODPUT FOR PRIORITY TRAFFIC USING TCP FOR BACKGROUND TRAFFIC.

	Normal	Discard	Buffer[1]	LPW[1]
Goodput (%)	18.3	57.5	53.9	51.9

[1] with ingress queuing enabled.

VI. PREEMPTION AND TCP

The previous simulation results are based on UDP as the transport protocol for the background traffic. However, TCP is the protocol mostly used for packet transportation in the Internet. It is preferred due to its rate control, but in ad hoc networks this feature can be detrimental for the performance [6].

The difference between TCP and UDP is mainly rate control and packet acknowledgments. TCP automatically attempts to take as much as possible of the medium, while not causing congestion. UDP, on the other hand, has no rate control and blindly sends what is received from upper layers. With the setup used in the simulations (i.e. where all nodes send traffic to half of the other nodes in the network), the one hop flows (i.e. the flows going directly between two neighbors) will support the highest throughput. Thus, TCP will end up transmitting most packets via these flows.

Using the TCP protocol for the background traffic yields merely 18% goodput without any extra applied mechanisms (Table I). TCP keeps on pushing packets for the one hop flows at a high rate, corresponding to a very high constant bit rate system load, while reducing the load over multi-hop flows. This limits the propagation of the high priority packets initiating the preemption, resulting in a lower goodput for the discard mechanism. The initialization phase of the preemption is harder to accomplish when TCP is used for background traffic. It is observed that even as long as one second after the first high priority packet is propagated in the network, parts of the network are still transmitting TCP traffic. Due to a combination of hidden node and TCP's high rate transmissions over one hop, nodes that should forward the high priority traffic experience collisions, either when receiving the high priority traffic, or when forwarding it. Thus, although a particular node may have stopped transmitting the lower priority packets, other nodes in the neighborhood are unaware of this.

In fact, with TCP, the initialization problem gets worse when some nodes have received a high priority packet and subsequently stop transmitting the TCP background traffic. This event will make more of the medium available to other TCP flows, and these will quickly increase the load to use the additional capacity. As the high priority traffic spreads

outwards from the source, the TCP flows that increase in load act as hidden nodes for the high priority traffic, increasing the probability of collisions and hence the delay in initializing preemption. This can be seen as a race condition, where the high priority traffic, initializing the preemption, competes with the rate control of TCP.

The even lower goodput for the buffer and LPW preemption mechanisms, compared to the discard mechanism, is a result of the always full queues at the start of the high priority sessions. These packets are to be pushed out, either after the high priority session is over, or during the low priority window between the high priority packets.

The results testing the preemption mechanisms with the background traffic carried by TCP were lower than with UDP, as the high priority flow at best, using the discard mechanism, achieved 20% less goodput, going from around 80% to 58% goodput. The two other mechanisms buffer the background packets, and this makes them more vulnerable to TCP's rate control. This was due to the vulnerable preemption initialization phase. However, this problem is closely related to the challenge of using TCP in ad hoc networks, and is not a problem only faced by these preemption mechanisms. Investigating the preemption mechanisms with the use of TCP, and TCP modified for ad hoc networks, is an interesting topic for further work.

VII. Related work

There is little work to be found on preemption in PTT ad hoc networks. Most solutions for Quality of Service (QoS) in ad hoc networks focus on QoS routing [7]. QoS for multicast has been studied, but here too most of the work has been focused on routing and the enhancement of multicast distribution [8].

Some QoS solutions implement call admission control, but only a very few, such as [9] consider preemption. Works on the preemption of traffic flows primarily focus on the preemption of the real time flows, such as [10].

In [11], Elmasry et al. propose a model managing QoS for Secure Tactical Wireless Ad Hoc Networks, directed towards the future US Army tactical backbone network. It is based on traffic characteristics measurements, calculating congestion severity levels and, based on this, generating admission and preemption policies.

For wireless sensor networks, several works on real-time scheduling have been published. A more recent is JiTS [12], a Just-in-Time scheduling protocol which works by delaying packets so they are received "just in time" at the destination node. The result is greater resilience against traffic bursts causing congestion.

VIII. Conclusions and future work

In this paper we have showed how employing priority mechanisms can improve multicast priority traffic conditions in ad hoc networks. The individual effects of the mechanisms were investigated. It was shown that priority queuing can stop the negative impact of background traffic on the priority traffic, but only through interface queue tail drops. To be able to maintain the same goodput as without any background traffic, it was necessary to use discard or buffering preemption of the background traffic, thus effectively removing the low priority traffic from the medium as the high priority traffic flow was active. The low priority window preemption was a compromise between the buffer preemption and only priority queuing, increasing the guard windows to enhance the high priority traffic performance at the expense of the background traffic.

The simulations using TCP for the background traffic showed that although the preemption with discard or buffering mechanisms were very effective with UDP, TCP poses different challenges. The analysis on the UDP background traffic presented in this paper is a good starting point for continued work on the preemption mechanisms and TCP background traffic. One solution to mend the initialization problem could be to transmit special initialization packets at very short intervals beginning the initialization phase. This way, the TCP algorithm would not be able to increase the rate as much as with only ordinary voice traffic with much larger intervals between the packets.

Initial studies on the QoS support of IEEE 802.11 (802.11e) MAC protocol [13] were performed. However, these showed that the performance of the high priority traffic was reduced compared to not using the QoS mechanism, due to the increased number of collisions stemming from the reduced contention window settings for the high priority traffic. Therefore, the 802.11e was not investigated further in this paper, but could be considered as future work.

References

[1] "Network simulator 2 - ns2." [Online]. Available: http://nsnam.isi.edu/nsnam/index.php/Main_Page

[2] "ns-2 code for random trip mobility model." [Online]. Available: http://monarch.cs.rice.edu/~santa/research/mobility/

[3] J. P. Macker, J. Dean, and W. Chao, "Simplified multicast forwarding in mobile ad hoc networks," *Military Communications Conference, 2004. MILCOM 2004. IEEE*, vol. 2, pp. 744–750 Vol. 2, 2004. [Online]. Available: http://dx.doi.org/10.1109/MILCOM.2004.1494892

[4] A. Hafslund, T. T. Hoang, and O. Kure, "Push-to-talk applications in mobile ad hoc networks," *Vehicular Technology Conference, 2005. VTC 2005-Spring. 2005 IEEE 61st*, vol. 4, pp. 2410–2414 Vol. 4, May-1 June 2005.

[5] L. Landmark, K. Øvsthus, and O. Kure, "Alternative packet forwarding for otherwise discarded packets," in *Future generation communication and networking (fgcn 2007)*, vol. 1, Dec. 2007, pp. 8–15.

[6] S. Xu and T. Saadawi, "Does the ieee 802.11 mac protocol work well in multihop wireless ad hoc networks?" *Communications Magazine, IEEE*, vol. 39, no. 6, pp. 130–137, Jun 2001.

[7] P. Mohapatra, J. Li, and C. Gui, "Qos in mobile a hoc networks," *Wireless Communications, IEEE*, vol. 10, no. 3, pp. 44–52, June 2003.

[8] A.-H. A. Hashim, M. M. Qabajeh, O. Khalifa, and L. Qabajeh, "Review of multicast qos routing protocols for mobile ad hoc networks," *International Journal of Computer Science and Network Security*, vol. 8, no. 12, pp. 108–117, December 2008.

[9] M. Canales, J. Gallego, A. Hernandez-Solana, and A. Valdovinos, "Cross-layer routing for qos provision in multiservice mobile ad hoc networks," *Personal, Indoor and Mobile Radio Communications, 2006 IEEE 17th International Symposium on*, pp. 1–5, Sept. 2006.

[10] G.-S. Ahn, A. Campbell, A. Veres, and L.-H. Sun, "Swan: service differentiation in stateless wireless ad hoc networks," *INFOCOM 2002. Twenty-First Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, vol. 2, pp. 457–466 vol.2, 2002.

[11] G. Elmasry, C. McCann, and R. Welsh, "Partitioning qos management for secure tactical wireless ad hoc networks," *Communications Magazine, IEEE*, vol. 43, no. 11, pp. 116–123, Nov. 2005.

[12] K. Liu, N. Abu-Ghazaleh, and K.-D. Kang, "Jits: just-in-time scheduling for real-time sensor data dissemination," in *Pervasive Computing and Communications, 2006. PerCom 2006. Fourth Annual IEEE International Conference on*, March 2006, pp. 41–46.

[13] IEEE, "Wireless LAN medium access control (MAC) and physical layer (PHY) specification," IEEE standard 802.11-2007, June 2007.

Paper E :

Optimized Group Communication for Tactical Military Networks

E. Larsen, L. Landmark, V. Pham, Ø. Kure, and P. E. Engelstad

In proceedings of the IEEE Military Communications Conference (MILCOM), San Jose, CA, USA, October 31–November 3, 2010, pp. 1445–1451, ISBN: 978-1-4244-8179-8.

Optimized Group Communication for Tactical Military Networks

Erlend Larsen*, Lars Landmark†, Vinh Pham*, Øivind Kure* and Paal E. Engelstad‡
*†Q2S NTNU, ‡SimTel (Telenor/Simula)
Email: *‡{erl, vph, paalee, okure}@unik.no, †larsla@q2s.ntnu.no

***Abstract*—In tactical networks there is a need for group communication applications, such as position and information sharing (Situational Awareness data), and Push-to-Talk (PTT) voice communication. This paper focuses on group communication in tactical military ad hoc networks, where most of the nodes are interested receivers. In this case, an efficient flooding protocol will be the best solution for the group communication. Efficient flooding can be achieved with the Simplified Multicast Forwarding (SMF) framework. The performance of SMF depends on the chosen forwarding algorithm. Two plausible alternatives are S-MPR and NS-MPR. The former is the more bandwidth efficient, while the latter is more robust to mobility.**

This paper investigates the limitations of the forwarding algorithms and investigates measures to mend S-MPR's mobility problem. Further, the paper suggests combining S-MPR and NS-MPR using the *radio load* as metric. Finally, the PTT and Situational Awareness (SA) traffic types are evaluated when run simultaneously, and a preemptive switch to S-MPR is proposed for the SA traffic. Through employing the methods suggested in this paper, the performance for PTT and SA traffic forwarded using SMF in tactical military networks can be increased.

I. Introduction

Mobile ad hoc networks are highly suitable for use as tactical military networks, due to the autonomous behavior and the independence from any infrastructure. In tactical networks there is a need for group communication applications, such as position and information sharing, i.e. Situational Awareness (SA) data, and Push-to-Talk (PTT) voice communication. This paper focuses on group communication in tactical military ad hoc networks, where most of the nodes are interested receivers. In this case, an efficient flooding protocol [1] will be the best solution for the group communication.

Simplified Multicast Forwarding (SMF) [2] is a framework protocol supporting efficient broadcast. It can employ one of several algorithms to provide a Connected Dominating Set (CDS) to distribute group communication traffic to all receivers. The Source-based MultiPoint Relay (S-MPR) is one of the recommended forwarding algorithms for SMF and is also used for flooding routing information in OLSR [3]. S-MPR is a highly efficient algorithm for SMF, but vulnerable to mobility and collisions. Non-Source-based MPR (NS-MPR) is another SMF algorithm, more robust in mobile scenarios than S-MPR, but at the expence of more network resources than that of S-MPR.

The main contributions in this paper are:

- Examining the limitations of the S-MPR and NS-MPR algorithms for use with PTT and SA traffic.
- Proposing an effective combination of S-MPR and NS-MPR using a *radio load* metric.
- Proposing a preemptive switch to S-MPR for SA traffic when PTT traffic is in the network.

The rest of the paper is structured in the following way: Related work is described in Section II. The limitations and strength of the S-MPR and NS-MPR forwarding algorithms, based on the traffic types PTT and SA, are determined in Section III. In Section IV, ways to combine the S-MPR and NS-MPR algorithms to provide better performance over a broader set of scenarios are discussed, while Section V explores the challenges encountered when running both the PTT and SA services in a network. The conclusion and further work concludes the paper in Section VI.

II. Related work

Several works have evaluated and proposed enhancements to the MPR selection mechanism in OLSR, the basis of the S-MPR and NS-MPR algorithms. Jacquet et al. investigates the MPR selection in two scenarios in [4], and Busson et al. studies the MPR selection in [5]. Mobility as a challenge to OLSR is addressed in [6], where nodes experiencing a high degree of mobility reduces the HELLO interval, aka. Fast-OLSR.

In [7], Cho and Adjih propose to use MPRs to optimize multicast forwarding, much in the same way as SMF. A directional MPR algorithm, where the MPR forwarding algorithm is optimized through only forwarding packets if on the shortest path between the source and destination is proposed, along with MPR flooding and a combination of MPR flooding and mesh. Another efficient broadcasting method based on hop-limited shortest-path trees is proposed in [8].

Qin and Kunz discuss mobility metrics to enable adaptive MANET routing in [9]. The same authors discuss adaptive routing in MANETs in [10], performing a case study using the number of monitored link breaks as key mobility metric.

The S-MPR, NS-MPR and two other forwarding algorithms for SMF were studied in [11]. It was shown that the S-MPR and NS-MPR had very different properties when stressed with mobility and offered load. NS-MPR performed similar to Classic Flooding (CF), with high resilience to mobility, at the cost of a high number of redundant transmissions. S-MPR provided the best performance at high loads. However, little attention was paid to the algorithm performance at low traffic loads, (e.g. PTT or SA traffic).

III. Current limitations

In [12], it was observed that PTT traffic forwarding using S-MPR in mobile topologies suffered loss ($\sim$15%), even without interfering traffic. This spurred an interest in investigating what performance that can be expected from a PTT service forwarded by means of SMF in a MANET, and further to optimize the performance. This section first presents the three investigated algorithms, S-MPR, NS-MPR and CF, and then introduces the traffic types and the simulation setup. Next, simulation results with varying density show how mobility is a problem for S-MPR. After this, the impact of increasing traffic load on the performance of the algorithms shows how choosing S-MPR over NS-MPR and CF can give increased performance, even in mobile scenarios. Finally, the section ends with a discussion on improving the S-MPR for mobile topologies.

The S-MPR algorithm works as follows: Any node *a* will only forward a multicast packet if it receives the packet from a node *b* that has selected *a* as an MPR. These conditions limit the number of transmissions that each multicast packet generates as it is spread throughout the network. Thus, the number of redundant packet transmissions is kept low. This is an advantage with high traffic loads and in dense networks, where the medium is near congestion and a reduction in the number of transmitted packets saves precious capacity. With low traffic loads or in sparse networks, however, the low packet transmission redundancy makes the S-MPR algorithm vulnerable for collisions.

In addition to collisions, S-MPR is also vulnerable to mobility, because it only forwards the packets received from an MPR selector. One or more of its MPR selectors may have transmitted the packet, but mobility can have caused the selector to be out of range. This way, many packets that could have been forwarded and reached more receivers, are instead lost. The minimization of the number of neighbors needed to cover all 2-hop neighbors also contributes to mobility vulnerability. To maximize the 2-hop coverage, the node will select neighbors as close to its own transmission area edge as possible. Thus, selected MPRs are more likely to travel beyond the transmission range than other neighbors.

NS-MPR is based on S-MPR, but contrary to S-MPR any node *a* selected as MPR will forward packets from *b* even if *b* did not select node *a* as MPR (i.e. non-source-specific). Therefore, the redundancy is higher with NS-MPR than S-MPR, since all nodes selected as MPR forward the packet. Also, since NS-MPR is not source-specific, it does not share S-MPR's broken relations mobility problem.

CF is the traditional broadcast algorithm, where all nodes hearing a packet forwards it once. This requires no neighborhood information, so the algorithm is very robust. However, each packet generates n number of transmissions in a n size network, causing the broadcast storm problem [13]. However, for very low traffic loads it may still be a viable algorithm. The NS-MPR algorithm has been likened to the CF algorithm, and the results with the CF algorithm are included to emphasize the difference between the two algorithms, and to investigate partitioning. It is also interesting to see whether it is at all necessary to maintain neighborhood information to support group communication in tactical military networks.

Efficient broadcast is a well suited transmission method for both PTT and SA traffic, but the two traffic types differ in QoS requirements. While PTT is dependent on a low packet loss rate to be usable, the SA can sustain much higher packet loss. PTT traffic is generally limited to one source in the network at a time, handled semantically by the users. The SA traffic is transmitted periodically by all network users, and the packet size will vary, based on the implementation and the amount of SA information. In this paper, both very small packet sizes, potentially containing little more than a position and an ID, at 40 bytes packet size, up to 800 bytes of varying SA information, are considered. The PTT and the SA services are both considered essential in a tactical military network, and they are therefore investigated both separately and in combination in this paper.

The average percentage of goodput was used to evaluate the performance of the algorithms. The goodput per packet was calculated so that a multicast packet received at a subset (s) of all nodes (n) in the network was calculated as $\frac{s}{n}$ received. The aggregated percentage of goodput was the average of this fraction for all packets sent during the measurement interval. Also, the average delay and jitter values were observed, to avoid excessive values that would render a service such as PTT useless. The simulation results were sampled over 10 simulation runs, and the results are presented with a confidence interval of 95%.

The simulations were performed using the ns-2 network simulator [14] version 2.34. The Optimized Link State Routing Protocol (OLSR) [3], [15] was used to provide neighborhood information and MPR selection, and the IEEE 802.11 protocol [16] was used as MAC layer. The traffic type was UDP multicast, and depending on the traffic type, the minimum packet size was 40 bytes (SA) and 21 bytes (PTT). With SA, all nodes sent one packet every 2 s. Thus, with the SA traffic, the traffic load increased with the number of nodes in the network. With PTT traffic, one random source transmitted traffic at a time, and the source changed every five seconds. The interval between packets was set to 67.5 ms, corresponding to 1.2 kbps load using the MELPe [17] voice encoding standard with 1:2 Forward Error Correction (FEC). All multicast packets were forwarded with a 1 ms jitter to prevent collisions among synchronically forwarded multicast packets.

The TwoRayGround radio propagation model was employed, where neither the Doppler spread nor the gradual link quality degradation by distance is taken into account. Also, the IEEE 802.11 random access protocol was selected in favor of scheduled access mechanisms. The main reason for this choice of lower layer model and protocol was due to most of the related works using this simulation configuration. Thus, this enables comparable results.

The simulation area was 1500x300 m^2. The dimensions

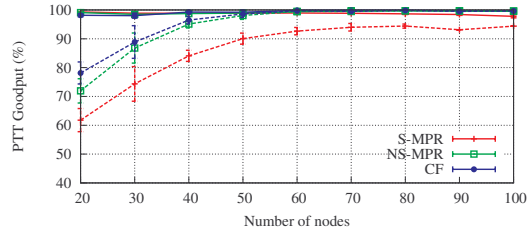
TABLE I
SIMULATION PARAMETER SETTINGS

Interface queue type	FIFO with DropTail and PriQueue
Interface queue size	30 packets
Antenna Model	OmniAntenna
Nominal transmission rate	2 Mbps
Basic rate	1 Mbps
Transmission radius	250 m
Sensing radius	550 m
Simulation time	600 s
Measurement time	60-590 s
Random seed	Heuristic
Traffic TTL	32
OLSR Hello interval	2 s
OLSR Hello timeout	6 s

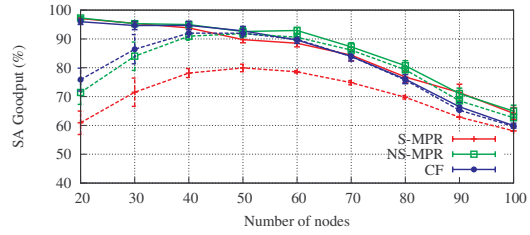
were selected to get a topology with many hops and little partitioning without the need of a very large set of nodes. The mobility model was Random Walk with Reflection with a 10 s $\pm$5 s travel time before direction change, created using code from [18]. All nodes had the same velocity and there was no pause time. It was made sure that none of the 10 topologies were partitioned at 0 m/s mobility (static topology). It was impossible to avoid partitioning of the low density topologies at some time during the simulations with 10 m/s mobility. However, all simulations with the same velocity were run on the same set of 10 different initial topologies, to make the comparison between the different algorithms as fair as possible. Other simulation parameter settings are presented in Table I. All results are presented with solid lines representing static topology results, and dotted lines representing results with 10 m/s topology mobility.

The algorithms were based on the OLSR neighborhood information and MPR selection, and the implementations of the algorithms were validated through close inspection of packet propagation in simulations with different topology sizes and distributions. The results were also compared to the low data rate results in [11], and the comparison showed similar results.

In Fig. 1(a), the goodput results for PTT traffic with varying density can be observed. Since the load is so small, there is negligible loss for all three algorithms at zero mobility. Even so, the S-MPR algorithm shows some problems when the number of nodes approach 100, due to collisions. The loss is not caused by congestion, as no packets are discarded from the interface queue. At 10 m/s mobility (dotted lines) the S-MPR algorithm is incapable of achieving the near 100% goodput achieved by the two other algorithms. At very low densities, all three algorithms have problems with the goodput, due to partitioning. At 50 nodes density, however, the goodput is about the same as without mobility for the CF and NS-MRP algorithms, while S-MPR suffers a 10% loss. In addition to the collision vulnerability, which also increases with mobility, S-MPR is more vulnerable to a changing topology than NS-MPR, due to the relation between MPR selector and MPR selectee which reduces the number of links in the CDS, causing a higher probability of logical partitioning.



(a) PTT goodput



(b) SA goodput

Fig. 1. Goodput results at varying density.

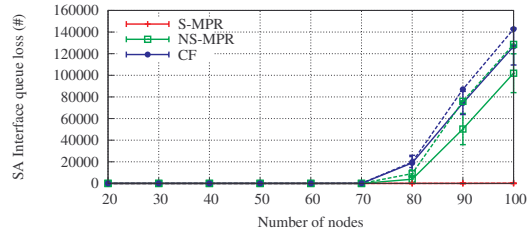


Fig. 2. Interface queue loss for SA traffic at varying density.

With the PTT traffic, there is only one source sending at a given time. Thus, the propagation of packets is always outwards from the source, posing a reduced risk of hidden node collisions for a node downstream from the source. I.e. an occurring collision is more likely to be experienced at a node that has already heard the packet. The SA traffic, on the other hand, may be sent simultaneously from several sources in the network, increasing the risk of a hidden node collision for nodes where the packet has not been heard before, if they are positioned between two sources. Such a collision is problematic, because it may potentially ruin the chance of the packet being heard at that node and all potential CDS downstream nodes. This problem makes it interesting to also investigate the performance of the algorithms with regards to the SA traffic.

The goodput results for the SA traffic (Fig. 1(b)) are compared at 0 and 10 m/s node mobility for increasing density. The SA traffic with 40 bytes packets constitutes a low data load (4.8 kbps at 30 nodes density). Without mobility (solid lines)

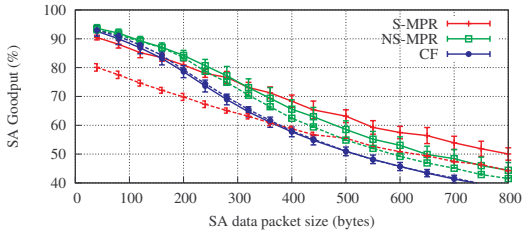


Fig. 3. SA goodput results, varying packet payload size.

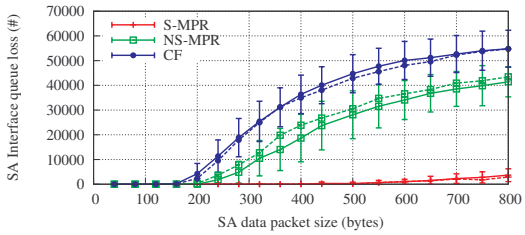


Fig. 4. Queue loss results for SA traffic at varying packet payload size.

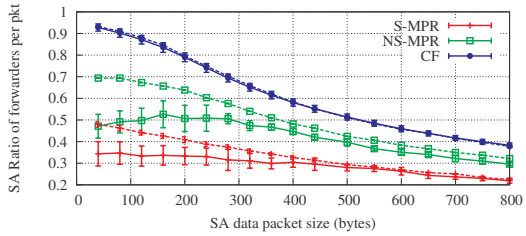


Fig. 5. Ratio of forwarders in the topology, SA traffic with increasing packet payload size.

there is little difference between the algorithms at low density. As the density increases past 40 nodes, the NS-MPR algorithm is somewhat better (∼5%) than S-MPR, until congestion starts occurring for NS-MPR past 80 nodes density, as visualized through the queue loss results (Fig. 2).

The difference in goodput between the algorithms is more evident at 10 m/s mobility (Fig. 1(b), dotted lines), with more than 10% separation at all densities. While both algorithms experience losses due to partitioning at low densities, the NS-MPR is able to reach 90% goodput at 50 nodes density, before succumbing to the increasing number of collisions. It also experiences tail drop queue loss (Fig. 2) when congestion happens, and this is due to an increased number of MPRs, which will be discussed later in the paper. The S-MPR algorithm, on the other hand, is unable to achieve 80% goodput, before the increased collision rate reduces the goodput. Even at 100 nodes density, congestion does not occur with S-MPR.

The problems that are observed for S-MPR are related to two issues. First, the number of forwarding nodes is lower for S-MPR than for NS-MPR, due to S-MPRs restriction on which nodes to forward packets from. Thus, a collision has a higher potential of causing a packet loss for parts of the network. Second, the mobility can cause the relation between the MPR selector and the MPR selected to be broken, risking a partitioning of the CDS. With fewer links in the CDS, the S-MPR is a more vulnerable algorithm than NS-MPR, where any MPR selected node will forward a packet once.

Even though the S-MPR algorithm has problems with mobility and collisions, its restrictive forwarding behavior is a highly valued attribute. To explore this behavior, the performance for the SA application is analyzed as a function of increased traffic load. Fig. 3 displays the performance of the three algorithms at 50 nodes density with an increasing SA packet size, with the same 2 seconds packet interval from each source. Without mobility it is not possible to distinguish between S-MPR and NS-MPR at low packet sizes, but the S-MPR clearly performs better than NS-MPR as the packet size increases. This is due to the much lower strain that S-MPR puts on the network, whereas the NS-MPR and CF algorithms cause congestion (Fig. 4) at larger packet sizes. With the additional challenge of 10 m/s mobility (dotted lines), the S-MPR trails the results of NS-MPR for larger packet sizes, until the load is so high that the S-MPR's restrictive forwarding again makes it better suited for the high load environment. The difference between CF and NS-MPR also becomes evident in Fig. 3. The CF performs much worse than NS-MPR at high loads. CF reaches congestion at a lower packet size than NS-MPR, and CF has a much higher ratio of forwarders in the topology. On the other hand, mobility does not impact the CF results.

The reason why the S-MPR is better suited at higher loads can be observed directly in Fig. 5, where the ratio of forwarders in the topology is shown. Using the NS-MPR algorithm, ∼40% of the nodes are forwarding packets in the simulations without mobility. As the packet payload size is increased past 200 bytes, congestion starts occurring for the NS-MPR algorithm, and this causes a decline in the forwarding nodes ratio. For S-MPR, the forwarding ratio on average lies at around half of the NS-MPR ratio. With 10 m/s mobility (dotted lines), the forwarding ratio is higher for both algorithms, but here too, the difference in ratio is maintained with increasing packet size, leading to S-MPR performing better than NS-MPR at high data load.

There is major difference between the ratio of forwarders with and without mobility, especially for NS-MPR, but also for S-MPR. With mobility, the ratio increases, and this is an effect of high mobility causing nodes to have more logic neighbors. As nodes move around, it takes less time to establish new neighborships than to time out lost neighbors. Since it is faster to select a node as MPR than to time it out when it is out of reach, the number of nodes selected as MPR increases with mobility.

As the S-MPR is vulnerable for mobility, it was attempted to improve performance for S-MPR in mobile scenarios. The

first attempt sought to increase the number of MPRs selected by each node through setting the OLSRs MPR_COVERAGE parameter higher than 1. This forces all nodes to select redundant MPRs per 2-hop neighbor. While this gave the desired effect of increasing the goodput in low load mobile topologies, it increased the number of transmissions and thus lowered the load threshold for congestion in high load scenarios.

The other attempted improvement sought to use implicit acknowledgment (IAck) to detect link breaks, and thus topology changes, earlier than OLSR's own link break detection. This is made possible through the MPR selection process, where a node a knows which neighbors that should forward a packet – namely all neighbors that have been selected as MPR by node a. However, this solution also faced problems in high load scenarios, where a missing IAck is likely to be caused by collisions or queue drop. In such a situation, selecting a new neighbor as MPR would only increase the network load, creating more congestion.

Thus, the conclusion of this section is that S-MPR has problems with mobility, but is very efficient in high load networks. It is difficult to improve S-MPR for mobility without reducing its high load efficiency. Thus, if one wants the best performance both in mobile, low load environments and in static high load environments, a combination of S-MPR and NS-MPR could be a good option.

IV. Combining S-MPR and NS-MPR

Instead of deciding in advance what forwarding algorithm the network should use, anticipating what scenarios the network must operate in, it is an option to combine S-MPR and NS-MPR dynamically. NS-MPR can be employed when this is optimal, i.e. in high mobile and low loads situations, while S-MPR is employed in low mobile and high loads situations. The dynamic combination of S-MPR and NS-MPR is possible because the two algorithms are based on the same MPR forwarding CDS, with S-MPR being more restrictive. Thus it is possible that some nodes in the network forward based on S-MPR while others forward based on NS-MPR. While packet redundancy, queue load or mobility are parameters that could, under given circumstances, be calibrated to switch between S-MPR and NS-MPR optimally, preliminary simulations showed that it was the *radio load* metric that gave the most encouraging results, and this was explored further.

Instead of using indicators such as queue load or redundant packets to anticipate whether the medium is heavily loaded, one could use information at the MAC layer to generate a radio load parameter based on the medium time that all packets either received (regardless of link destination) or transmitted occupies, relative to the total time of measurement. This metric is referred to as *Radio load*. When the radio load is low, the cost of using NS-MPR is low. As the radio load increases, it is better to switch over to S-MPR to save radio capacity. The obtained *radio load* metric can be used to decide which forwarding algorithm to employ. Consider a scenario where NS-MPR is used and the offered load is high. As the network becomes congested, the *radio load* metric would approach 1,

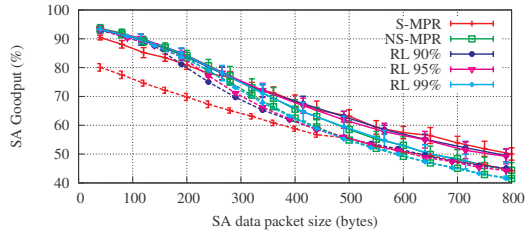


Fig. 6. SA goodput results, comparing the *radio load (RL)* combination to S-MPR and NS-MPR.

and it would then be preferable to switch to the more effective, but less robust, S-MPR forwarding algorithm.

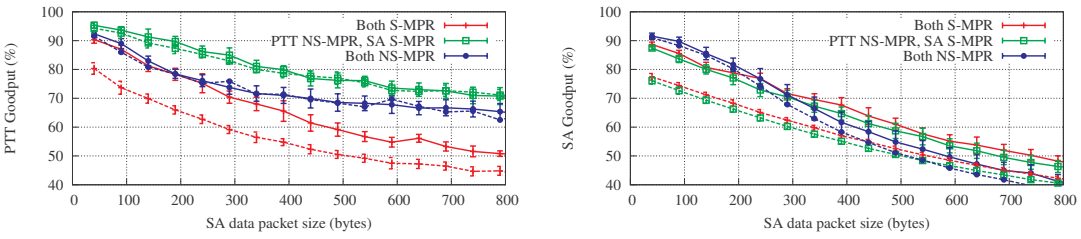
The *radio load* metric was evaluated using 50 nodes topologies at 0 and 10 m/s. The rest of the simulation settings were the same as in the previous simulations in the paper. The radio load threshold was investigated at 90%, 95% and 99% where a radio load higher than the threshold set the packet forwarding algorithm to S-MPR, while a radio load lower than the threshold set it to NS-MPR. As the results show, the *radio load* metric has potential to be further optimized. The radio load was sampled once each second, measured at the MAC-layer through adding all the time spent either receiving or transmitting packets during the measurement time (one second). The resulting usage $R_{measured}$ is weighted with $\alpha = 0.3$ against the last calculated radio load R_{last} so the radio load $R = \alpha \cdot R_{measured} + (1 - \alpha) \cdot R_{last}$ for a more stable value, but at the cost of delayed reaction to a sudden shift in the value.

Since the *radio load* metric only switches between S-MPR and NS-MPR, it can be expected that its results will lie between those of S-MPR and NS-MPR. The simulations results (Fig. 6) show that the optimal radio load threshold is different depending on the mobility. While the 90% threshold results show very good performance at 0 mobility, following the best algorithm both at high and low loads, it is not able to follow the NS-MPR results at low load with 10 m/s mobility (dotted lines); it switches to S-MPR too soon. With the 99% threshold, the *radio load* metric is able to select the better algorithm at low loads, both with and without mobility, but switches to S-MPR too late, choosing NS-MPR even when the load is too high. Although there is some loss compared to NS-MPR at low loads with mobility, the results with a 95% threshold show that this threshold represents a good compromise, and is able to select the NS-MPR at low loads and S-MPR at high loads.

The *radio load* metric can be used with success to distributedly and automatically switch between the two forwarding algorithms S-MPR and NS-MPR to obtain the best throughput at varying load and mobility.

V. Optimizing Group Communication

Group communication in tactical military networks will include both PTT and SA traffic. However, introducing the two



(a) PTT goodput (b) SA goodput

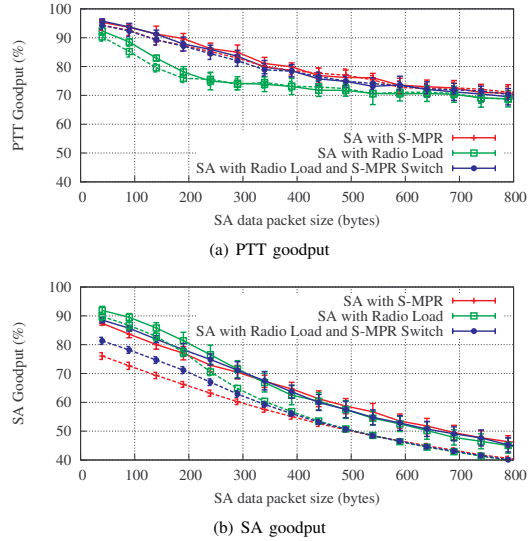
Fig. 7. Goodput results with simultaneous transmission of PTT and SA traffic. PTT and SA traffic forwarded with varying algorithms.

services into a network will require priority handling to avoid loss of PTT traffic, since PTT is much more vulnerable to loss than SA. As shown in [12], priority queuing will reduce the risk of packet loss for the PTT traffic in congested networks, since the SA packets will be discarded before any PTT packets. Also, with priority queuing the delay and jitter for the PTT packets are reduced, since the PTT traffic packets are inserted ahead of any SA packets. Based on the experiences in [12], priority queues are used in the simulations, prioritizing PTT traffic in front of SA traffic.

Packet loss is not only caused by packet drops from the interface queue. At loads lower than the congestion limit, the PTT traffic suffers from loss due to collisions. As the results show, the PTT traffic is impacted with higher loss even at the minimum SA packet size, when the SA traffic is forwarded using NS-MPR. The PTT traffic is impacted the least (Fig. 7(a)) when the SA traffic is forwarded using S-MPR, while the PTT traffic at the same time is forwarded using NS-MPR. Even when the PTT traffic must share the network resources with the SA traffic, it is better to forward the PTT traffic using NS-MPR, also at 0 m/s mobility (solid lines). This is due to S-MPR's vulnerability to collisions. If both the PTT and the SA traffic is forwarded using S-MPR, the PTT goodput results are lower compared to if PTT is forwarded with NS-MPR and SA is forwarded using S-MPR. For the SA traffic (Fig. 7(b)), it is interesting to see that the trend for the goodput results is very similar to the earlier results without PTT traffic, in that as the packet size increases, the optimal algorithm still changes from NS-MPR to S-MPR.

Although the *radio load* based combination of S-MPR and NS-MPR should detect the increased traffic due to an initiated PTT session, this could be delayed due to the radio load stabilization weight. It could also be that the increased load is not enough to force nodes in the network over to using S-MPR. While a preemptive stop to the SA data would be best to secure the best goodput for the PTT traffic, it is not recommendable, since a long duration PTT session would be detrimental for the SA service. Therefore, considering that PTT achieves the best goodput when the SA traffic is forwarded using S-MPR, we propose a preemptive switch to S-MPR for the SA traffic for the duration of the PTT session.

To evaluate this S-MPR switch, the PTT traffic was modified



(a) PTT goodput

(b) SA goodput

Fig. 8. Goodput results for simultaneous PTT and SA traffic. SA forwarded with pure S-MPR and with our proposed dynamic solutions.

so that a pause of 5 seconds was introduced after each 5 s session. This way it is possible to investigate the performance in a network where there are periodic PTT sessions, but also periods with only SA traffic. The preemptive switch to S-MPR was implemented so that a node checks if it has heard a PTT packet the last 0.5 seconds each time it is about to forward a packet. If so, the SA traffic is forwarded using S-MPR. The radio load threshold for the SA traffic was set at 95%.

The PTT goodput results (Fig. 8(a)) show that using the *radio load* metric alone is not enough to force the SA traffic to be forwarded using S-MPR when a PTT traffic session is in the network. With only this mechanism employed, the PTT traffic suffers, compared to SA traffic only forwarded with S-MPR. On the other hand, the results show that the preemptive S-MPR switch is very effective, achieving the same results as when the SA traffic is forwarded using only S-MPR.

The SA traffic goodput results (Fig. 8(b)) also show inter-

esting behavior. Here, using only the *radio load* metric yields the best results, but this is at the expence of the PTT traffic, as was seen above. The preemptive switch results show worse results than using only the *radio load*, but still much better than the results forwarding the SA traffic only by S-MPR.

The results in Fig. 8 show that using our proposed solutions of combining S-MPR and NS-MPR using *radio load* and employing a preemptive switch to S-MPR can improve the overall utilization of the network, increasing the goodput of the SA traffic while avoiding a reduction of the goodput for the PTT traffic. Even with the improved service for the PTT traffic, the goodput is reduced below 80% when the SA data packet size is larger than around 350 bytes. At higher packet loss than 20%, supporting PTT will be very difficult, and access control or scheduling mechanisms must be employed. However, we remain confident that the proposed solution can play a part in securing the coexistence of PTT and SA traffic in tactical military networks.

VI. Conclusions and future work

This paper has analyzed the behavior of two forwarding algorithms for SMF, the S-MPR and NS-MPR algorithms, for PTT and SA, two essential traffic types in a tactical military network. The algorithms have different limitations, making them suitable in different network conditions. The NS-MPR is robust, but quickly creates congestion with increasing load, while the S-MPR is vulnerable to mobility.

Some mechanisms were explored to mend the mobility problem for S-MPR. Although some improvement was achieved, the mechanisms must be switched off at high loads, since they are based on increasing the number of forwarders in the network. Instead of improving the S-MPR, a combination of S-MPR and NS-MPR could be used. The *radio load* metric showed itself as the most effective method of combining S-MPR and NS-MPR, being able to select the best of the two algorithms at varying loads.

The SA traffic was shown to affect the goodput of the PTT, and a preemptive forwarding algorithm switch to S-MPR was proposed for the SA traffic. This optimized the performance of the PTT traffic, while allowing the SA service to operate during a PTT session. Despite a suboptimal choice of radio load thresholds, the results showed that the *radio load* combination of S-MPR and NS-MPR, combined with the preemptive switch to S-MPR for the lower priority SA traffic, can optimize the group communication in tactical military networks.

Further work should focus on testing the *radio load* combination metric under more varying scenarios, and explore how the threshold may be optimized, either statically or dynamically. Also, the combination metric should be investigated using experiments. Finally, the behavior of the two proposed solutions in networks with other traffic types, such as unicast traffic, and additional mechanisms to support QoS should also be investigated.

References

[1] F. Dai and J. Wu, "Performance analysis of broadcast protocols in ad hoc networks based on self-pruning," *IEEE Trans. Parallel Distrib. Syst.*, vol. 15, no. 11, pp. 1027–1040, 2004.

[2] J. Macker and the SMF Design Team, *Simplified Multicast Forwarding*, MANET WG Internet-Draft, Rev. 09, March 2010, work in progress. Intended status: Experimental, Expires: September 7, 2010. [Online]. Available: http://www.ietf.org/internet-drafts/draft-ietf-manet-smf-10.txt

[3] T. Clausen, P. Jacquet (editors), C. Adjih, A. Laouiti, P. Minet, P. Muhlethaler, A. Qayyum, and L.Viennot, "Optimized link state routing protocol (OLSR)," RFC 3626, pages 1–75, pp. 1–75, October 2003, network Working Group. [Online]. Available: http://ietf.org/rfc/rfc3626.txt

[4] P. Jacquet, A. Laouiti, P. Minet, and L. Viennot, "Performance analysis of olsr multipoint relay flooding in two ad-hoc wireless network models," INRIA, Research report 4260, September 2001. [Online]. Available: http://www.inria.fr/rrrt/rr-4260.html

[5] A. Busson, N. Mitton, and E. Fleury, "An analysis of the multipoint relays selection in olsr," INRIA, Research report 5468, January 2005. [Online]. Available: http://www.lri.fr/~fragile/IMG/pdf/RR-5468_analyseMPR.pdf

[6] M. Benzaid, P. Minet, and K. Al Agha, "Analysis and simulation of fast-olsr," *The 57th IEEE Semiannual Vehicular Technology Conference (VTC)*, vol. 3, pp. 1788–1792, April 2003.

[7] S. Cho and C. Adjih, "Optimized multicast based on multipoint relaying," in *Wireless Internet, 2005. Proceedings. First International Conference on*, July 2005, pp. 42–46.

[8] S. Shioda, K. Ohtsuka, and T. Sato, "An efficient network-wide broadcasting based on hop-limited shortest-path trees," *Computer Networks*, vol. 52, no. 17, pp. 3284 – 3295, 2008. [Online]. Available: http://www.sciencedirect.com/science/article/B6VRG-4T9VPCY-1/2/6d0aecb50001527eb2f9caa74f9dbd85

[9] L. Qin and T. Kunz, "Mobility metrics to enable adaptive routing in manet," *IEEE International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)*, pp. 1–8, June 2006.

[10] ——, "Adaptive manet routing: A case study," in *ADHOC-NOW '08: Proceedings of the 7th international conference on Ad-hoc, Mobile and Wireless Networks*. Berlin, Heidelberg: Springer-Verlag, 2008, pp. 43–57.

[11] J. Macker, I. Downard, J. Dean, and B. Adamson, "Evaluation of distributed cover set algorithms in mobile ad hoc network for simplified multicast forwarding," *SIGMOBILE Mob. Comput. Commun. Rev.*, vol. 11, no. 3, pp. 1–11, July 2007. [Online]. Available: http://portal.acm.org/citation.cfm?id=1317425.1317426

[12] E. Larsen, L. Landmark, V. Pham, P. Engelstad, and O. Kure, "Preemption mechanisms for push-to-talk in ad hoc networks," in *Local Computer Networks, 2009. LCN 2009. IEEE 34th Conference on*, oct. 2009, pp. 428 –435.

[13] Y.-C. Tseng, S.-Y. Ni, Y.-S. Chen, and J.-P. Sheu, "The broadcast storm problem in a mobile ad hoc network," *Wirel. Netw.*, vol. 8, no. 2/3, pp. 153–167, 2002.

[14] J. Heidemann and T. Henderson (Editors). (2009, October) Network Simulator 2. [Online]. Available: http://nsnam.isi.edu/nsnam/index.php/Main_Page

[15] F. J. Ros and P. M. Ruiz. (2009, October) MANET Simulation and Implementation at the University of Murcia (MASIMUM). [Online]. Available: http://masimum.dif.um.es/

[16] IEEE, "Wireless LAN medium access control (MAC) and physical layer (PHY) specification," IEEE standard 802.11, June 1999.

[17] J. Collura and D. Rahikka, "Interoperable secure voice communications in tactical systems," in *Speech Coding for Algorithms for Radio Channels (Ref. No. 2000/012), IEE Seminar*, 2000, pp. 7/1 –713.

[18] S. PalChaudhuri. (2009, October) Ns-2 code for random trip mobility model. [Online]. Available: http://monarch.cs.rice.edu/~santa/research/mobility