Transport-Layer Performance in Wireless Multi-Hop Networks

Jonas Karlsson
Transport-Layer Performance in Wireless Multi-Hop Networks

Jonas Karlsson
Transport-Layer Performance in Wireless Multi-Hop Networks

Jonas Karlsson

Dissertation

Karlstad University Studies  I  2013:1

ISSN 1403-8099

ISBN 978-91-7063-473-4

© The author

Distribution:
Karlstad University
Faculty of Health, Science and Technology
Department of Mathematics and Computer Science
SE-651 88 Karlstad, Sweden
+46 54 700 10 00

Print: Universitetstryckeriet, Karlstad 2013

WWW.KAU.SE
Transport-Layer Performance in Wireless Multi-Hop Networks

JONAS KARLSSON
Department of Mathematics and Computer Science, Karlstad University

Abstract

Wireless communication has seen a tremendous growth in the last decades. Continuing on this trend, wireless multi-hop networks are nowadays used or planned for use in a multitude of contexts, spanning from Internet access at home to emergency situations.

The Transmission Control Protocol (TCP) provides reliable and ordered delivery of a data and is used by major Internet applications such as web browsers, email clients and file transfer programs. TCP traffic is also the dominating traffic type on the Internet. However, TCP performs less than optimal in wireless multi-hop networks due to packet reordering, low link capacity, packet loss and variable delay.

In this thesis, we develop novel proposals for enhancing the network and transport layer to improve TCP performance in wireless multi-hop networks. As initial studies, we experimentally evaluate the performance of different TCP variants, with and without mobile nodes. We further evaluate the impact of multi-path routing on TCP performance and propose packet aggregation combined with aggregation aware multi-path forwarding as a means to better utilize the available bandwidth. The last contribution is a novel extension to multi-path TCP to enable single-homed hosts to fully utilize the network capacity.

Keywords: TCP, Multi-Path TCP, MPTCP, transport protocols, packet aggregation, multi-path routing, reordering, wireless multi-hop networks, Mobile Ad hoc NETworks, tactical MANET, wireless mesh networks, WMN
Acknowledgements

I wish to thank a number of people that have been important to me during the work on this thesis. Foremost my advisor, Prof. Andreas Kassler, for giving me the opportunity to pursue my doctoral studies and for continuously providing me with inspirational ideas and guidance throughout all of my work. The same goes for my co-advisor Prof. Anna Brunstrom. You both helped me improve this work, far beyond my own capability, thank you.

As a great source of inspiration and good friendship, I would also like to express my gratitude towards the colleagues and friends at the Computer Science department at Karlstad University and to the members of the Distributed System and Communications Research group (DISCO), thank you all.

Thanks to my friends and family and for supporting my work and encouraging me throughout all of these long years. A special thanks also goes to my sister Marie Karlsson for her endless (unpaid) proofreading, any errors that remain are my sole responsibility. I would finally like to send a special thanks to my heart’s desire, Sandra Wassberg for tolerating my absence, quirks and long hours of work.

Karlstad, December 2012                Jonas Karlsson
List of Appended Papers

This dissertation is based on the work presented in the following peer-reviewed papers, journals and technical reports. The dissertation includes minor extensions, revisions and additions to the material from these publications.

I. Jonas Karlsson and Andreas Kassler. TCP Performance in Mobile ad hoc Networks Connected to the Internet. In REVISTA CIENTÍFICA PERIÓDICA — TELECOMUNICAÇÕES, 10(1), Santa Rita do Sapucaí, Brasil, December 2007.


Comments on my Participation

In papers I, III and IV I am responsible for carrying out the experiments, and for most of the written material and ideas. In paper II, I am responsible for the TCP analysis and most of the written material. In papers V and VI, I am responsible for the TCP experiments and the implementation of the packet aggregation and the written material for the same. In paper VII, I am responsible for the written material and implementation of the flow-aware routing and jointly responsible for the analysis. In paper VIII, I am responsible for the written material and implementation of the flow-aware routing. I am also jointly responsible for the analysis and kernel implementation of the proposed scheme.
Other Publications

Apart from the papers included in the thesis, I have authored/co-authored the following papers, books, chapter in books and scientific reports:


- Jonas Karlsson and Alba Batlle Linares. TCP Performance — in Hybrid Mobile ad hoc Networks. Printed by VDM Verlag Saarbrücken Germany, 2008.

## INTRODUCTORY SUMMARY

1 Introduction 3
   1.1 Research Objective and Topics Addressed 5
   1.2 Research Methodology 6
   1.3 Main Contributions 7
   1.4 Structural Overview 10

2 Background and Related Work 10
   2.1 Wireless Multi-Hop Networks 10
   2.2 Link Layer Technologies 13
   2.3 Routing 20
   2.4 Aggregation 24
   2.5 TCP 29

3 Summary of Papers 37

4 Conclusions and Future Work 42

## PAPER I
TCP Performance in Mobile ad hoc Networks Connected to the Internet 55

1 Introduction 57

2 Internet Connected MANETs and TCP Performance 60
   2.1 Internet Connectivity for MANETs 60
   2.2 TCP Problems in Internet Connected MANETs 61
   2.3 TCP Protocols Evaluated 63

3 Performance Evaluations 65
   3.1 Chain 5 Scenario 65
   3.2 Chain 5+1 Scenario 66
   3.3 Random Mobility 70

4 Changing the Routing Protocol 71

5 Conclusions 74
PAPER II
Performance Evaluation for TCP in Tactical Mobile ad hoc Networks
1 Introduction
2 Background
3 Implementation
4 Results and Analysis
5 Conclusion

PAPER III
Impact of Packet Aggregation on TCP Performance in Wireless Mesh Networks
1 Introduction
2 Packet Aggregation in Wireless Multi-Hop Networks
  2.1 Packet Aggregation basics
  2.2 Proposed Packet Aggregation Algorithm
3 Simulation Setup
  3.1 TCP Performance in Wireless Mesh Networks
  3.2 Scenario Description
4 Simulation Results
  4.1 Goodput
  4.2 Packet Loss and Delay
5 Conclusion

PAPER IV
TCP Performance Evaluation in Wireless Mesh Networks using Packet Aggregation
1 Introduction
2 Related Work
3 Packet Aggregation in Wireless Multi-Hop Networks
  3.1 End-to-End vs. Hop-by-Hop Aggregation
  3.2 Visualization of Gain by Aggregation
  3.3 Proposed Packet Aggregation Algorithm
# Table of Contents

## Paper V

**An Aggregation Aware Multi-Path Forwarding Paradigm for Wireless Mesh Networks**

1. Introduction
   - 145

2. Architecture
   - 146
   2.1 Channel Assignment and Routing
   - 147
   2.2 Packet Aggregation
   - 147
   2.3 Forwarding Strategies
   - 148

3. Evaluation
   - 152
   3.1 Trace Analysis Single-Hop
   - 152
   3.2 Simulations in Multi-Hop Scenarios
   - 153

4. Related Work
   - 157

5. Conclusions and Future Works
   - 160

## Paper VI

**Combining Multi-Path Forwarding and Packet Aggregation for Improved Network Performance in Wireless Mesh Networks**

1. Introduction
   - 166

2. Packet Aggregation in Wireless Mesh Networks
   - 167

3. Multi-Path Routing in Wireless Mesh Networks
   - 168
   3.1 Layer-2.5
   - 168

4. Multi-Path Forwarding and Packet Aggregation
   - 169
   4.1 Aggregation Sub-Layer
   - 170
   4.2 AF-L2R
   - 171
“Not everything that can be counted counts, and not everything that counts can be counted.”

*Albert Einstein (1879-1955)*
1 Introduction

The Internet is used on a regular basis by millions of people to send emails, read news, shop, search for information or watch movies [1]. All of the above activities use the TCP/IP protocol. TCP is therefore by far the most used transport protocol on the Internet and carries 90% of the Internet traffic [2]. The success of the Internet and portable computers, such as laptops, has also lead to the rapid growth and development of wireless Internet access technologies, such as WLAN or IEEE 802.11 [3] networks.

In wireless multi-hop networks, the nodes relay packets wirelessly between the source and the destination. Wireless multi-hop networks can be divided into two subgroups, depending on the mobility of the relaying nodes and the clients. In mobile ad hoc networks (MANETs), the clients act as both clients and relays and are thus part of the core wireless network. MANET clients can be both stationary and mobile. MANETs are designed to work in challenging network conditions where the regular wired infrastructure is at best semi-functional, i.e. during disasters or military campaigns. A MANET should therefore be resilient to node failures and operate with little or no backbone infrastructure.

In military operations and due to the limited bandwidth available, the main applications for MANETs are short messages and Voice Over IP (VOIP) [4,5]. However, good TCP performance is also important in MANETs when for example rescue teams need to download maps or access other content available on the Internet. In this thesis we will distinguish between general MANETs and tactical MANETs. A tactical MANET is a subset of a general MANET where node mobility and traffic follows a special pattern.

The main focus of this thesis will be on the second subgroup of wireless multi-hop networks called wireless mesh networks or WMNs. In a WMN the relay nodes are fixed and form a wireless core network to which the mobile clients, which are not part of the core network, can connect via wireless mesh access points. WMN clients have high bandwidth and delay requirements as it can be anticipated that the applications and user behavior within a WMN will be similar to that of today’s Internet users. Therefore, the protocols and technologies in the core WMN must be highly efficient and provide good TCP performance.

Unfortunately, TCP performance degrades significantly in wireless networks and even more in wireless multi-hop networks. This is mainly due to TCP’s inability to distinguish between packet loss caused by congestion or other factors intrinsic to MANETs and WMNs, such as bit errors and (multi-hop) contention, as we will discuss further in section 2.1. Therefor, a great amount of research has also been invested in proposing new TCP variants, e.g. [6] [7] [8] [9]. Most proposals change the functionality and/or the behavior of TCP congestion control to reflect the network environment, as we will further describe in section 2.5.

WMNs and MANETs are typically built using IEEE 802.11 WLAN technology. Transmitting small packets in IEEE 802.11 WLANs reduces the network
capacity because of a fixed overhead on the medium access control (MAC) and physical (PHY) layers. However, it is possible to reduce the amount of small packets by concatenating the packets into larger units. Such packet or frame aggregation is a well-known solution to reduce the number of small packets and increase performance of e.g. VOIP traffic in IEEE 802.11 networks [10] [11] [12], as we will discuss further in section 2.4. Recent studies [2] have shown that 40% of all TCP/IP packets transported on the Internet are small, indicating that packet aggregation also can be beneficial for TCP traffic.

IEEE 802.11 uses a carrier sense multiple access (CSMA) [13] MAC layer that, common for all CSMA based networks, performs close to optimally under light to moderate load. However, under heavy load the network becomes unstable and delay increases drastically. As TCP probes for bandwidth during the slow start phase the network load increases quickly. This leads to a high probability for MAC layer contention induced packet loss, i.e. collisions. This makes CSMA less suited when time critical messages are important and the network load is uncontrolled. Instead, where bounded delays and message prioritization are important, time division multiple access (TDMA) based networks are more suitable. However, as it takes time for a TDMA based scheme to negotiate on how to divide (schedule) the medium among the nodes, node mobility can disrupt the schedule. This can lead to collisions and reduced network capacity, as we will discuss further in section 2.2.2.

To reduce the collisions and MAC layer contention, it is possible to divide a network into several collision domains by using multiple channels. When nodes use different channels, the contention is reduced as there are fewer nodes in each collision domain. In a multi-hop network, contention can also be reduced by finding paths that avoid collision domains with high utilization (bottlenecks). Typically, routing protocols are categorized as either reactive, where paths are discovered when needed, or proactive, where paths are periodically discovered. Single-path routing protocols only use one path, while multi-path routing use several paths as we will discuss in more detail in section 2.3.

Multi-path routing can also utilize the channel diversity by load-balancing the traffic on multiple paths, on disjoint collision domains [14]. However, there is a trade-off between network and transport (TCP) layer throughput when load-balancing the traffic. To achieve optimal load-balancing at the network layer, the packets of a single TCP flow should be forwarded over multiple paths. This may lead to packet reordering, which reduces the TCP throughput [15]. Load-balancing can also be performed by a multi-path capable transport layer, such as MPTCP [16] where each MPTCP connection consist of several TCP flows, i.e MPTCP subflows. Transport layer load-balancing operates on a flow-level and requires that the network layer assigns paths to flows rather than to packets. However, when using a flow-based path assignment, the network capacity is limited by the number of flows passing the disjoint paths in the bottleneck, i.e. bottleneck disjoint paths. For efficient transport layer load-balancing, there must therefore be enough flows, e.g.
MPTCP subflows, to fully utilize all bottleneck disjoint paths, as further discussed in section 2.5.3.

This thesis deals with evaluating and analyzing TCP performance in both WMNs and MANETs. The investigated scenarios stretch from tactical TDMA based MANETs used in emergency situations to CSMA based WMNs used to replace cellular and wired access networks. We present several proposals to better support TCP traffic. Among those proposals, we develop three forwarding schemes that combine packet aggregation and multi-path routing as well as a novel algorithm that enables MPTCP to fully utilize all bottleneck disjoint paths. Next, we will describe the research objective and topics addressed in this thesis.

1.1 Research Objective and Topics Addressed

The objective of this thesis is to analyze and improve TCP performance in wireless multi-hop networks.

To fulfill this objective we will focus on the following research issues.

1. How do state of the art TCP variants behave under different routing and MAC protocols, with both mobile and static nodes?

When using CSMA based MAC layers, overloading the network leads to MAC layer induced packet loss, which may falsely trigger routing updates even when nodes are static. TDMA ensures stable operation under high load but node mobility can disrupt the TDMA schedule, leading to packet loss and route errors. The routing protocol can also influence the performance depending on when and how routes are discovered and maintained. Reactive routing protocols can be sensitive to the number of source-destination pairs while proactive routing protocols can be sensitive to both the traffic load and the node mobility.

In paper I we present simulation results and identification of problems to solve for efficient operation of three TCP proposals in Internet connected MANETs using both proactive and reactive routing protocols. We simulate MANET clients, mobile and stationary, connecting both to other clients within the MANET and, via gateways, to servers located on the Internet.

In Paper II we address this research issue by assessing and analyzing the problems and weaknesses of three different TCP versions in TDMA based tactical MANETs using both proactive and reactive routing protocols. The simulations evaluate file transfers between mobile nodes in the MANET both with and without background traffic.

2. How to better support TCP traffic in WMNs?

Following from recent studies [2], around 40% of all TCP/IP packets have a size between 40-100 bytes. Therefore, reducing the overhead for small packet transfers in WMNs is important to provide good TCP performance. However, since it is sometimes needed to delay packets to
achieve good aggregation possibilities, aggregation can result in reduced TCP throughput, as further discussed in section 2.5.

In Paper III we address this issue by designing a non-intrusive and easily deployable packet aggregation scheme for WMNs. We evaluate the scheme in a small WMN topology using a limited set of traffic patterns. The evaluation is continued in Paper IV to cover larger WMNs with more diverse traffic patterns.

3. **How to combine the benefit of multi-path routing and packet aggregation to improve TCP performance in WMNs?**

One interesting technique to improve performance in WMNs is to do multi-path routing and forwarding decisions based on the knowledge of channel configuration in the network. When using such multi-path routing, the forwarding scheme actively tries to load-balance the traffic on several paths, by selecting next hops that are on different channels, to reduce the possibility for bottleneck nodes/links. However, to achieve a high packet aggregation ratio, there is a need for packets to travel to the same next hop. These two conflicting goals make uncoupled packet aggregation and multi-path forwarding less than optimal.

In Papers V-VI, we develop and evaluate three aggregation aware multi-path forwarding schemes that combine the benefit of multi-path routing and packet aggregation.

4. **How to make TCP utilize all available network capacity when using multi-path routing?**

When load-balancing the traffic over the available paths, there is a trade-off between network and transport layer throughput. This is because a per-flow based routing policy may not fully utilize the network when the number of TCP flows are limited, while a packet-based policy might cause packets to arrive reordered and reduce TCP throughput.

In paper VII, we evaluate the interaction between routing layer strategies in a WMN and TCP reordering mitigations, concluding that under the assumption of a reasonable amount of TCP flows, a flow-based routing gives the best TCP throughput.

In paper VIII, we build on this knowledge to develop a novel algorithm that, without extra probing of the network, advises MPTCP of the number of subflows that should be opened to fully saturate the network capacity.

1.2 **Research Methodology**

Two research methodologies are commonly used in computer science to test theories: analytical and experimental.

In analytical methods, theories and problems are modeled as mathematical formulas and results are obtained by solving these formulas through mathematical methods. Analytical methods often provide deep insights into how various
parameters affect the end results and hard upper and lower bounds can often be
determined. In many cases, however, extensive simplifications must be made
to the model to be able to find a solution.

Experimental methods are normally used when the problem is too complex
or requires too many simplifications when solved analytically. In experimental
methods, problems and theories are modeled by simulation, by real measure-
ments or by emulation.

Simulation uses an abstract model of the system defined in a simulation
tool called a simulator, such as [17], [18] and [19]. The simulator makes the
representation of a problem more straightforward than analytical models allow
but at the cost of high computational complexity. Simulation also enables a
higher degree of separation between different effects, due to a more controlled
environment, than when using a real system.

When using real measurements, an operational system is studied. A major
advantage of real measurements is that there is no difference between the
“model” and the real environment. The drawback of real measurements is that,
due to the unpredictability and uncertainty of the environment, it is often hard
to produce controlled and reproducible experiments.

Emulation combines simulation and real measurements, where parts of
the system are abstracted and parts run on a real system. The advantage is
the combination of repeatability and controlled environment from simulation
with the use of a real system from real measurements.

For this thesis both simulation and emulation were used as research meth-
ods. Simulation allowed us to evaluate larger topologies and more complex
traffic scenarios while emulation allowed for experiments with real network
stacks. For simulations both ns-2 [17] and ns-3 [18] were used (paper I - VII).
For emulation, ns-3 was used in emulation mode (paper VIII). Both ns-2 and
ns-3 were extended and modified as described later on.

1.3 Main Contributions

The main contributions of this thesis are summarized below.

1. We present an extensive performance evaluation of different TCP variants,
   routing protocols and MAC layer schemes in MANETs.

   Paper I evaluates the performance of TCP NewReno [8], TCP AP [25]
   and TCP Vegas [9] using both reactive and proactive routing protocols,
   i.e. AODV(UU) [21] [22] and DSDV [23], in a CSMA based Internet
   connected MANET with both mobile and stationary nodes.

   In Paper II, we present a study of the performance of TCP NewReno,
   MANET using AODV [22] and OLSR [25].

   Our key findings are that when using CSMA and a reactive routing
   protocol, such as AODV, it is beneficial to use TCP-Vegas as it keeps
   the congestion window small. This reduces the bad interactions with
   the MAC and routing layers and leads to a more stable network. When
using TDMA, the key finding is that TCP ELFN achieves the overall highest throughput regardless of the routing protocol used. However, the relative gains by using TCP ELFN are reduced in low throughput scenarios as the benefit of freezing the congestion window is less.

2. **We design, implement and evaluate a routing agnostic packet aggregation algorithm that improves TCP performance, tailored for both single-path and multi-path scenarios.**

   The aggregation algorithm uses a conservative aggregation policy to avoid packet reordering, possible flow starvation and queue overflows that would hamper TCP performance. This contribution is presented in Paper III and Paper IV.

   The conclusion is that packet aggregation can give a substantial performance increase for TCP traffic. In a small WMN, TCP throughput increased with up to 73% and the round trip time decreased with up to 40% by using our packet aggregation algorithm. In paper IV, we evaluated a scenario well known for creating unfairness and a scenario based on a packet trace from a library, with many short lived flows. We showed that here, packet aggregation can increase TCP throughput up to 60% and reduce TCP RTTs up to 30%. We further showed that slightly delaying packets before transmission, combined with aggregation, can reduce the MAC layer unfairness and thereby increase TCP flow fairness.

3. **We design, implement and evaluate three aggregation aware multi-path forwarding schemes.**

   In Papers V-VI, we develop and evaluate three aggregation aware multi-path forwarding schemes that combine the benefit of multi-path routing over a multi-channel wireless network and our packet aggregation algorithm.

   The first scheme jointly optimizes aggregation delay and aggregated packet size without considering load-balancing. The second scheme both satisfies load-balancing requirements, as mandated by the channel assignment and routing strategy [14], and gives priority to packet aggregation possibilities. The last scheme presented in Paper VI is a generalization of the two former schemes that can gracefully prioritize between aggregation and load-balancing using different parameter settings. The schemes require information of queue size, timer values (for the aggregation delay) and available buffer for aggregation. The second scheme also requires throughput and delay information for optimal parameter settings.

   Our simulation results show that aggregation always gives better TCP performance than no aggregation. In Paper V, we used five random topologies of 25 nodes with both UDP [26] and TCP traffic. The results show that with UDP traffic, both presented forwarding schemes outperform the schemes that do not take aggregation into account. However, for TCP throughput the best forwarding strategy is highly dependent
Transport-Layer Performance in Wireless Multi-Hop Networks

on the amount of packet reordering the strategy creates. A forwarding strategy that creates a higher amount of packet reordering results in lower TCP performance than one that leads to less reordering. In paper VI, we evaluate two scenarios using TCP based FTP traffic and UDP based traffic modeled after a packet trace. The results show that the forwarding scheme presented in the paper can achieve higher throughput than our proposed scheme in paper V for both TCP and UDP.

4. We evaluate and develop extensions to (MP)TCP for exploiting the benefit of multi-path routing in wireless mesh networks.

When using multi-path routing, the amount and type of reordering is severely hampering TCP performance. Two ways to cope with packet reordering are to avoid the reordering by using flow-level routing or to mitigate it at the transport layer. However, if there are not enough TCP flows, flow-level routing can not utilize the entire network capacity, while reordering mitigations at the transport layer can have problems handling the amount and types of reordering occurring in a WMN when using packet-based routing.

In paper VII, we investigate the impact of the reordering mitigation techniques and forwarding strategies on TCP performance by evaluating four different multi-path forwarding strategies, of which three were TCP flow-aware, with two transport layer reordering mitigation techniques. In paper VIII we extend the findings from Paper VII and evaluate the possibilities of using MPTCP to increase the number of flows in the network. This eliminates the situations where flow-level routing does not utilize the entire network capacity. However, MPTCP must know how many subflows to open. Therefore, we develop a novel extension to MPTCP to estimate the amount of subflows to open for utilizing the capacity available over the bottleneck disjoint paths. The algorithm, which we call PathFinder, is a sender side only modification of MPTCP. With a few basic assumptions of the congestion control and the routing, such as flow fairness and flow-level load-balancing, we show that with PathFinder MPTCP can fully utilize the capacity of the bottleneck disjoint paths.

The key finding from paper VII is that flow-level forwarding outperforms packet-based forwarding with respect to TCP throughput, as long as there are enough TCP flows to use all bottleneck disjoint paths. Furthermore, we show in paper VIII that it is possible for a single homed MPTCP sender to fully utilize a multi-path capable backbone network without requiring cross-layer information.

5. To support the experiments we have contributed several implementations.

- We implemented the packet aggregation algorithm used in papers III – VI.
- In papers V and VI we implemented three aggregation aware multi-path routing and forwarding schemes as described above.
The first two schemes were implemented in ns-2 miracle [27] while the last scheme used the ns-3 platform.

- In paper VII we implemented three TCP flow-aware multi-path routing and forwarding schemes.
  The schemes where implemented in the ns-2 miracle platform.
- In paper VIII we contributed with a Linux kernel implementation of the pathfinder algorithm.
- We have also done a multitude of experimental setups and code modifications.
  We have integrated OLSR with TCP ELFN, ported the ns-2 aggregation module to ns-3 and done various bug fixing and modifications to default settings in both ns-2 and ns-3.

1.4 Structural Overview

The rest of this introductory summary is structured as follows. We continue with providing an overview on the topics covered in this thesis in section 2. Section 3 gives a summary of the presented papers analyzing also their limitations. Finally, section 4 presents our conclusions and future work.

2 Background and Related Work

In the following section we provide a comprehensive overview of wireless networking and discuss related work and developments in the field. We will begin by giving a background on wireless multi-hop networks followed by an overview of two link layer technologies commonly used in wireless multi-hop networks. The last three subsections will present both a background to and describe related work in the areas where we have more directly contributed; beginning with routing in section 2.3 followed by packet aggregation in section 2.4 and TCP in section 2.5.

2.1 Wireless Multi-Hop Networks

Wireless multi-hop networks can extend the coverage of single-hop networks by using intermediate nodes to wirelessly forward packets on behalf of their neighbors. Multiple low power transmissions not only increase the network coverage, it also gives the possibility of spatial reuse as several simultaneous transmissions can take place in the network at the same time (at an ample distance). If the transmissions happen too closely they will compete or result in a collision. However, there is also a higher risk of packets being lost along the path as packets need to be forwarded over multiple hops to reach the destination.

In the following we will give an overview of the two categories of wireless multi-hop networks that we study in this thesis. We will begin with MANETs, used in papers I and II, followed by WMNs, used in papers III – VIII.
2.1.1 MANET

In MANETs, the clients act as both clients and relays and are therefore part of the core wireless network. The main use case for MANETs is to provide rapid Internet connectivity where there is little or no infrastructure, e.g. a disaster scenario.

A schematic Internet connected MANET, where mobile clients connect to the Internet via three gateways can be seen in figure 1. Note, that individual mobile nodes not only serve as relay stations but also can send their own data. All traffic destined towards a host located in the Internet needs to pass through at least one gateway, which can be mobile as well. A gateway is a node that has at least two interfaces, one running in infrastructure mode (wired or wireless) connected to the Internet and one running in ad hoc mode connected to the MANET. Typical for MANETs is that node mobility leads to a highly dynamic network topology, which is prone to frequent changes. The frequency of topology changes can make it difficult for routing protocols to maintain network connectivity and increases routing overhead (see section 2.3.1).

Because of the network dynamics in MANETs, a significant portion of packet losses are caused by link failures either due to high bit error rate or due to mobility of nodes resulting in route errors or network partitions. Therefore, several components, such as discovery of gateways and automatic configuration of IP addresses, must function under challenging network conditions. Solutions that address these issues and a detailed description and performance evaluation of existing schemes can be found in [28].

To limit the energy drain, most nodes will use a single radio tuned to a common channel. This increases the amount of collisions due to hidden nodes (see section 2.2.1). When using a single channel network, the transmission of earlier received packets can also interfere with the reception of later packets in the same flow, commonly called intra-flow interference. This is especially
problematic with TCP, as TCP tends to send packets in bursts, which can overflow the network and create routing errors [29]. In paper I we evaluate TCP performance in an Internet connected MANET using both DSDV and AODV as routing protocols. Further, we focused on TCP variants that do not require cross-layer information as the sender, in an Internet connected MANET, often is located outside the MANET. This means that modifications to TCP are generally not possible without introducing middle-boxes.

Similar to a general MANET, a tactical MANET should be resilient to node failures and operate with little or no backbone infrastructure. However, a tactical MANET is a relatively small network where communications are among groups which coordinate their movements [30]. Tactical MANETs are used in military and rescue operations to provide timely and accurate information to operating teams. Whereas, due to the availability of cheap IEEE 802.11 radio cards, general MANET research has mainly focused on CSMA based systems; using a TDMA scheme has several advantages in a tactical MANET as it gives better possibilities to support real-time communication than CSMA (see sections 2.2.1 and 2.2.2). While it is also important to support TCP based applications, voice and short messages are the main applications for tactical networks. Therefore, support for message priority, bounded delays and a stable network under heavy traffic loads is paramount. In paper II we investigate the TCP throughout in a multi-hop tactical MANET using both ADOV and OLSR as routing protocols.

2.1.2 WMN

A WMN is as a core network that wirelessly forwards client traffic over one or more paths to an Internet gateway or another mesh access point. A WMN is
Transport-Layer Performance in Wireless Multi-Hop Networks

built as a 2-tier architecture, consisting of mobile wireless clients as a first tier and a wireless backbone network of mesh access points, mesh gateways and mesh relay nodes forming a second tier, see figure 2. Typically, the second tier consists of nodes connecting to multiple neighbors, sometimes equipped with multiple radio interfaces using a set of orthogonal channels. Due to the widespread use and common requirement of low upfront cost, the predominating hardware platform is IEEE 802.11\footnote{IEEE 802.11} for which several commercial products exist, e.g.\footnote{Several commercial products exist} \footnote{Several commercial products exist}.

The stable core network topology and the existence of channel diversity improves the performance of single-path routing, while at the same time increasing the possibility for multi-path routing protocols to load-balance the traffic over multiple interference-free paths. Therefore, more possibilities for route and forwarding optimizations exist compared to a MANET, possibly reducing intra- and inter-flow interference and thereby increasing TCP throughput.

A common scenario in a WMN and Internet connected MANETs is that many TCP flows share the same link, e.g. towards the gateway. This results in high contention on bottleneck links around the gateway, which is escalated due to the high traffic demands in a WMN. The unfairness of the IEEE 802.11 MAC layer, where shorter packets have a lower collision risk, also poses additional problems. Due to this, TCP data-packets will have a higher collision risk than TCP ack-packets that traverse in the opposite direction, leading to reduced TCP performance. Furthermore, unfairness between TCP flows that traverse different number of hops reduces the performance for the flows that traverse more hops\footnote{Many TCP flows share the same link, e.g. towards the gateway. This results in high contention on bottleneck links around the gateway, which is escalated due to the high traffic demands in a WMN. The unfairness of the IEEE 802.11 MAC layer, where shorter packets have a lower collision risk, also poses additional problems. Due to this, TCP data-packets will have a higher collision risk than TCP ack-packets that traverse in the opposite direction, leading to reduced TCP performance. Furthermore, unfairness between TCP flows that traverse different number of hops reduces the performance for the flows that traverse more hops.}

Multi-path routing can reduce the stress on bottleneck links and paths by load-balancing the traffic among the paths in the network. This can be done by allocating a share of the traffic, either per-flow or per-packet, on each path. When the number of flows is less than the available paths, per-flow allocation may not fully utilize the network while per-packet allocation can create packet reordering see sections 2.3.2 and 2.5.2).

Although several successful multi-path routing proposals for single channel networks exist, e.g. AOMDV\footnote{AOMDV}, to fully utilize the possibilities of multi-path forwarding the network should be divided into separate collision domains by using multiple channels\footnote{There exist different strategies on how to assign channels to nodes (see section 2.3.2). However, we will focus on WMNs where the channels are statically preassigned to one channel per radio and node, although nodes may have more than one radio and many nodes can share the same channel.}. There exist different strategies on how to assign channels to nodes see section 2.3.2. However, we will focus on WMNs where the channels are statically preassigned to one channel per radio and node, although nodes may have more than one radio and many nodes can share the same channel.

### 2.2 Link Layer Technologies

When nodes share a common medium, such as links using a common frequency channel, they need to coordinate their transmissions. Much like a group of people in a crowded room must coordinate so not all talk at the same time. Similar to the group of people, the nodes can transmit on the same channel at the same time if they are far enough apart so that one transmission...
does not disturb other ongoing transmissions. The required distance for two transmissions to not disturb each other is called the interference range. All transmissions that are within the interference range are in the same collision domain. If multiple channels are used, the network will be split up in more collision domains, allowing more transmissions in parallel. This is important in multi-hop networks as it allows multiple paths to be used in parallel to increase the aggregate throughput of the network.

Two popular MAC schemes that we study in this thesis are CSMA and TDMA. In CSMA based MAC protocols, every node needs to contend for the medium before transmission. This makes the scheme easy to implement and it requires no synchronization between the nodes. Even though CSMA is based on time-domain multiplexing it is not performed in a cyclically repetitive frame structure. Therefore, CSMA naturally provides dynamic bandwidth allocation. However, when the number of nodes and traffic increases, collisions can make the network collapse, as we discuss in next section.

In TDMA based MAC protocols, the medium is divided into discrete time slots. Each node is assigned one or more time slots and is allowed to transmit only in its own time slot(s). This eliminates collisions, but requires synchronization between the nodes. In dynamic TDMA, a node can reserve a variable number of time slots. Dynamic TDMA can therefore dynamically allocate bandwidth by using a scheduling algorithm for assigning slots to nodes according to traffic demands. TDMA requires clock synchronization between the nodes, which increases the overhead and makes nodes more complex and expensive to build, as we further discuss in section 2.2.2.

2.2.1 CSMA
A pioneering work in MAC layer design was the ALOHA network that connected several Hawaiian islands [36]. One drawback of ALOHA is that there is no sensing before transmission which may result in many collisions. To reduce the number of collisions, in CSMA a node needs to sense the medium before transmission. If the medium is sensed free the node can send with the probability \( p \) between \( 0 \leq p \leq 1 \). If the medium is busy, or the station chooses not to transmit (the probability of this event is \( 1 - p \)), the sender defers the transmission for a period of time [13]. This implicit scheduling of the MAC layer reduces the amount of collisions but since two nodes might choose the same \( p \), collisions can still occur, especially in loaded networks with many nodes.

The research community has only recently developed practical full-duplex radios [37]. Therefore most proposals are still based on half-duplex radios, where the sender can not detect if there has been a collision during an ongoing transmission. Furthermore, when using a wireless medium it is not possible to know a priori the upper bound on propagation delay. Therefore, guaranteeing collision detection by all nodes as in wired Ethernet [38] is not possible. Collision Avoidance (CA) is instead used to reduce the possibility for collisions. Nodes that sense the medium busy choose a random back-off interval, which reduces the likelihood that nodes access the medium at the same time and
thereby reduces the risk of a collision. CSMA/CA is used as a base for the distributed coordination function (DCF) MAC protocol used in IEEE 802.11 [3]. Analysis has shown that the amount of collisions and CSMA throughput are strongly dependent on the number of active stations and on the total offered load [13] [39]. With more active stations, it is more likely that two stations will transmit at the same time.

When not all nodes are within transmission range of each other, sensing and MAC layer coordination is far from trivial causing several adverse effects. Two of the most known effects are the hidden and exposed node problems [40]. Due to channel fading, a sender may not hear a transmission from a neighboring node of the intended receiver side. The sender might thus schedule a transmission that collides with the transmission of the hidden node at the receiver, resulting in lower throughput and a waste of network resources [41]. An exposed node is a node in the interference range of the sender, which defers from sending even though the transmission would not have collided at the intended receiver [40]. The number of collisions and the effects of hidden and exposed nodes are directly related to the load of the network. It is therefore important to transfer the data as efficiently as possible to minimize the load. In paper III and IV we evaluate the possibility to use packet aggregation to reduce the load in a multi-hop network.

IEEE 802.11 IEEE 802.11 [3] defines two modes of MAC operation: distributed coordination function (DCF) and point coordination function (PCF). In the mandatory DCF method, IEEE 802.11 uses physical carrier sensing with virtual carrier sensing as an option. If the medium is sensed idle for a time interval that exceeds the distributed interframe space (DIFS), the node starts its transmission cycle. The length of a DIFS is equal to a short Interframe space (SIFS) + 2xslottime, where SIFS and slottime are fixed per physical layer. For IEEE 802.11g, slottime = 9 µs and SIFS = 10 µs. If the medium is busy, the transmission is deferred until the end of the ongoing transmission. A random interval is then selected, which is used to initialize the backoff timer. The backoff timer is decreased for each slot as long as the channel is sensed idle, and frozen during the time when a transmission is detected. The node starts to transmit when the backoff timer reaches zero. The backoff timer is chosen from a discrete uniform distribution in the interval (0, CW − 1) defined as the backoff or contention window. At the first transmission attempt, CW = CWmin, and is later doubled at each retransmission up to CWmax: in the current standard, CWmin = 16 and CWmax = 1024 [3]. Experimental and analytical analysis have shown that the performance of DCF is dependent on the parameterization of the contention window [42]. In a wireless network with only half-duplex radios, a node is unable to detect a collision. Therefore, DCF uses a positive acknowledgment (ack) to report a successful transmission. If no ack is received within a pre-determined period (ack timeout), which is shorter than a DIFS, the frame is retransmitted.

Physical carrier sensing can not detect hidden nodes and two nodes may therefore schedule overlapping transmissions at a common receiver. DCF
therefore includes an optional virtual carrier sensing in the form of a request to send and clear to send (RTS/CTS) mechanism. Before the transmission, a sender transmits a short request to send (RTS) packet. The receiver will then respond with a clear to send (CTS) packet if the medium is sensed free. A node that hears a RTS or a CTS defers from sending during the time of the transmission. However, the virtual carrier is detected in the interference range of both the sender and the receiver, which can make a sender defer even though the transmission would not have collided at the receiver, increasing the exposed node problem. In many situations the overhead of the RTS/CTS messages is also higher than the cost to retransmit the frames [40]. The RTS/CTS packet can also be lost or collide which further reduces the benefit of the mechanism. RTS/CTS is therefore by many implementations turned off by default.

To allow backwards compatibility, IEEE 802.11 defines two rate sets: the basic and the data rate sets. All MAC and physical layer preambles are sent at a (basic) rate that all the nodes within interference range will be capable of using. The basic rate is the max-min rate for which all nodes within transmission range can decode a transmission. This rate is normally substantially lower than the data rate. Other protocol overheads, like the contention process, interframe spacing, PHY level headers (preamble + PLCP), and acknowledgment frames, also take a fixed amount of time. Therefore, data rate improvements in later iterations of IEEE 802.11, e.g. [43] [44], do not translate to the same amount of user level throughput. To increase throughput further, the frames sent over the medium can be aggregated (concatenated) as further discussed in section 2.4.2.

2.2.2 TDMA

A TDMA channel access method allows several nodes to share the same medium by dividing the channel into small time slots. The nodes then transmit using their own time slots. The most basic form of TDMA divides the time slots statically among the nodes. Variants of TDMA are used in tactical radio systems [45], Bluetooth [46], GSM [47], HiperLAN/2 [48] and IEEE 802.16 [49], more commonly known as WiMAX.

By design TDMA has a number of advantages over contention-based approaches, such as fairness, bounded delays and asymptotic behavior under heavy traffic loads. Therefore, TDMA systems can guarantee both bandwidth and latency requirements, as required by interactive applications. The main drawback of TDMA scheduling is that it requires clock synchronization between the nodes, which increases the overhead and makes nodes more complex and expensive to build. In addition, an interference free slot assignment becomes computationally expensive in multi-hop environments.

TDMA is also beneficial for battery operated nodes that need to save energy, as the duty cycle of the radio is reduced and there is no contention-introduced overhead and collisions. Static TDMA can not adapt its frame length and time slot assignment when traffic and the number of nodes changes. Therefore, static TDMA is not so well suited for bursty TCP/IP traffic due to its inflexible bandwidth allocation. To alleviate this, new nodes in dynamic TDMA can
be accommodated by changing the frame length dynamically according to the number of nodes in the contention area. A common solution to accomplish this is to enforce new nodes to request slots in a special slot that is reserved for control traffic and slot reservation [50]. In the rest of this thesis we will use the term TDMA to refer to dynamic TDMA.

The main task of a TDMA scheduler is to allocate time slots depending on the topology and, optionally, on bandwidth requirements of a node. An interference free schedule avoids collisions by silencing all interferers in each time slot and minimizing the number of time slots, hence the delay [51]. The goal of the scheduler is therefore to determine the smallest length conflict-free assignment of slots where each link or node is activated at least once. TDMA schedulers in a multi-hop network take into account the 2-hop neighbors of the sending node to overcome the hidden and exposed node problems. However, optimally assigning slots in a mobile multi-hop network is NP hard [52]. A heuristic approach is therefore typically used to define the number of slots for each node [50] [52].

TDMA uses mainly topology information as a basis for slot scheduling. The performance of TDMA is also strongly tied to the accuracy of this information and the synchrony among the neighbor nodes. When the accuracy can be upheld, TDMA delivers very good performance especially under high contention [53] [54]. Unfortunately, in wireless multi-hop networks it is difficult to maintain this accuracy because:

1. It is difficult to precisely capture the interference relation between nodes due to interference range irregularities [55] [52].

2. Channel conditions may change interference relations among nodes over time, due to mobility.

3. Clocks can drift and tight synchronization may incur too much overhead.

In a TDMA based network using an interference free schedule, assuming no node mobility and no packets drops due to low signal quality, there should only be packet drops due to queue overflow. Routing protocols that use the reception of packets for neighbor sensing can benefit, with large enough buffers, from less packet loss, especially in high contention scenarios, by using TDMA. However, the TDMA slot assignment could (when nodes are mobile) be temporarily incorrect creating a dropping pattern whereby also in TDMA packets are lost due to collisions.

The slot assignment problem can be seen as a direct extension of the graph coloring problem in graph theory. Let the network be represented by a graph $G = (V,E)$ where $V$ is the set of nodes (or vertexes), and $E$ is the set of edges. An edge $e = (u,v)$ exists if and only if $u$ and $v$ are in $V$ and nodes $u$ and $v$ can hear each other. The problem is to find the minimum number of colors (time slots) and assign them to the vertices of the graph in such a way that no two adjacent nodes have the same color (time slot). As the graph scales bigger and more complex, the time it takes to calculate the coloring increases dramatically.
One MAC layer proposal that uses TDMA is HiperLAN/2. In terms of physical layer speed, HiperLAN/2 is very similar to IEEE 802.11a wireless local area networks. However, different from IEEE 802.11 is that through the use of TDMA the connection oriented MAC of HiperLAN/2 can support QoS [48]. HiperLAN/2 also features higher application layer throughput as compared to 802.11. While both 802.11a and HiperLAN/2 use physical layer speeds of 54Mbps, the true usable throughput of HiperLAN/2 is around 42Mbps, while for IEEE 802.11a it is around 32Mbps based on packets with a size of 1500 bytes [56]. HiperLAN/2 uses centralized schedulers for both elastic and inelastic traffic [57]. This allows high throughput in single-hop networks, but using a centralized scheduler is difficult in (tactical) MANETs, due to mobility and node failures.

**DRAND** A decentralized TDMA proposal that we evaluate in paper II is DRAND [54]. DRAND was introduced as a means for providing a better interoperability between transmitting nodes, to minimize collisions and promote bounded delay for the purpose of real-time communication (e.g., voice communication). DRAND performs distance-2 coloring, i.e., it assumes that any two interfering nodes are at most two hops away from each other. Although this scheme requires more slots than distance-1 coloring, it is motivated by the fact that the interference range is usually larger than the communication range. Hence nodes may not be able to communicate directly, but they can still influence each other’s reception. One of the benefits of DRAND is that it does not require each node to have synchronized clocks.

In the following we will give an overview of DRAND. Figure 3 shows a simplified state diagram for the DRAND implementation. There are four states that a node maintains: IDLE, REQUEST, GRANT, and RELEASE, for a detailed description of the states and all transitions see [54]. In order to be assigned a time slot, a node plays in a lottery. The chance for a node to win the lottery is inversely proportional to the number of one-hop and two-hop neighbors that have not yet decided on a time slot. If a node wins the lottery it moves to the REQUEST state, in which it broadcasts a request message to its one-hop neighbors. When a neighbor receives the request message from the initial node, it sends back a grant message and moves to the GRANT state, but only if it has been in the IDLE or RELEASE state. With the grant message, the neighboring node sends back a list of the occupied time slots by its one-hop neighbors. If the same neighboring node is already in the REQUEST or GRANT state, it sends back a reject message to the request originator. When the original node receives the reject message, it broadcasts a fail message to all of its one-hop neighbors. When a neighbor receives a fail from a node whose request turned the neighbor’s current state to GRANT, the same neighbor returns to the IDLE state if it has not decided on its slot already or to the RELEASE state if it has decided on its time slot. As a node receives a grant from its entire one-hop neighbors, it decides on its time slot to be the minimum of the time slots that have not been taken by its two-hop neighbors (this information is piggy-backed on the grant messages). Then the
Transport-Layer Performance in Wireless Multi-Hop Networks

Figure 3: DRAND State Diagram [54]

Figure 4: DRAND successful round [54]

node enters the RELEASE state and broadcasts a release message containing information about its selected time slot to its one-hop neighbors. Figure 4 illustrates a successful round.
DRAND handles packet loss via retransmissions. To handle severe packet loss or node failure, the algorithm offers nodes the possibility of “giving up”. Since only requests and grants require any response, when a node does not receive any response to these messages from a one-hop neighbor for a fixed number of retries, then it removes the neighbor from its neighbor list. This allows the node to make progress without a response from the removed neighbor. Because of its probability-based nature, DRAND requires time to adapt with topology changes and is expected to perform worse when node mobility increases.

2.3 Routing

Wireless multi-hop networks do not require any wired infrastructure for their operation. Intermediate nodes must therefore participate in the route discovery and packet forwarding to other nodes. In the routing process there are three steps required: route discovery, route maintenance and traffic forwarding. The route discovery step finds one or multiple routes between each source and destination pair. Route maintenance finds alternative routes if and when paths break due to link or node failures. Finally, traffic forwarding forwards the packets to one or multiple next hops discovered by the route discovery step.

Routing protocols in wireless multi-hop networks are commonly divided into two categories based on how and when routes are discovered and maintained: proactive and reactive routing protocols. Reactive or on-demand routing protocols establish the route to a destination only when there is a demand for it. A proactive routing protocol on the other hand, tries to maintain a consistent and updated view of the network by periodically propagating route updates in the network.

For single-path routing the routing protocol discovers one path for each destination (as described in next section). Discovering multiple paths for each source/destination pair can be used to provide both fault tolerance and higher aggregate bandwidth by load-balancing the traffic (see section 2.3.2).

2.3.1 Single-Path Routing

Single-path routing protocols discover the best possible route to each destination, according to a given routing metric. The routing metric represents the optimization goal of the routing algorithm; such as maximum bandwidth, minimum network delay, minimum hop count, path cost etc. [58].

Reactive routing protocols establish a route when needed and tend to be used in highly mobile MANET deployments. An example of a reactive single-path routing protocol is AODV [22]. AODV is a distance-vector protocol that builds routes using route request/route reply packets sent between a source and a destination. When a source node wants to establish a route to a destination for which it does not already have a route, AODV broadcasts a route request (RREQ) packet. Nodes receiving a RREQ packet update their information for the source node and set up a reverse route to the source node in their respective routing tables. A node receiving a RREQ sends a route reply (RREP) if it is
either the destination or if it has a route to the destination newer or equal to
the one identified by the sequence number contained in the RREQ. If this is
the case, it unicasts a RREP back to the source. Otherwise, it rebroadcasts the
RREQ. If a node receives a RREQ which it has already processed, it discards
the RREQ and does not forward it. As the RREP propagates back to the
source, nodes set up forward routes to the destination. Once the source node
receives the RREP, it can start to send data-packets to the destination.

A route is considered active and is maintained as long as there are data-
packets periodically going from the source to the destination. Once the source
stops sending data-packets, the route will time-out and eventually be deleted
from the routing table of intermediate nodes. If a link break occurs while the
route is active, the node upstream of the break may try to find an alternative
route (local repair) and if that fails, propagate a route error (RERR) message
to the source node to inform it of the now unreachable destination(s). This
ensures a low routing overhead when there is little traffic. Although caching
mechanisms are implemented, the routing protocol needs to perform a route
lookup on all new connections [22]. This can lead to an increase of routing
overhead in high traffic conditions. Furthermore, as we show in paper I, packet
loss due to a high offered load can lead to unnecessary route errors in scenarios
with no mobility.

Unlike reactive protocols, proactive protocols regularly exchange topology
information among their neighbors to have an up-to-date view of the net-
work. One of the pioneering proactive single-path wireless routing protocols
is destination-sequenced distance vector or DSDV [23]. As AODV, DSDV
uses sequence numbers to indicate how old a route is. When a route update
with a higher sequence number is received, the old route is replaced. In case of
different routes with the same sequence number, the route with better metric is
used. Routing information is distributed by sending full dumps of the routing
table infrequently and smaller incremental updates more frequently. Updates
are transmitted both periodically and immediately when a significant topology
change is detected. To avoid fluctuations in route updates, DSDV employs a
“settling time” (refusing route updates for a time period), to predict the time
when a route becomes stable [59].

Both DSDV and AODV use a distributed Bellman-Ford algorithm, which
does not prevent routing loops and suffers from the “counting-to-infinity”
problem. At the core of the counting-to-infinity problem is that when using
distance vector routing there is no way for node A to know if the path has
node A as a part of it when node B tells node A that it has a path somewhere.
Therefore, false routing updates can slowly propagate through the network
until the hop count to the missing nodes reaches infinity (in which case the
algorithm corrects itself). However, by adding sequence numbers on the
routing updates, which is used by both AODV and DSDV, the routing loops
and the “counting-to-infinity” problem can be avoided [23].

The table driven OLSR [25] is a single-path proactive link-state routing
protocol that inherits the concept of forwarding and relaying from Hiperlan
[62]. The basic concept of link-state routing is that every node constructs
a connectivity map of the network [61] from topology messages sent by its neighbors. This allows each node to independently calculate paths to every destination in the network. The collection of best paths will form the node’s routing table. OLSR regularly exchanges topology information with its neighbors using “topology control” and “Hello” messages. To reduce the amount of broadcasts, only nodes chosen as multipoint relays (MPRs) announce the topology information periodically. In route calculation, the MPRs are also used to form the route from a given node to any destination in the network. Even with the MPRs, the topology control message updates give a high overhead when there is limited amount of traffic. As the routing overhead of a proactive routing protocol is not directly influenced by the traffic volume, the relative overhead is low when the traffic volume is large. However, with high mobility, the hello message frequency in OLSR might not be sufficient to track the neighborhood changes of a node. Therefore, routes to this node may become invalid and messages sent to the node may be lost. Increasing the frequency of hello messages would overcome this problem at the cost of higher protocol overhead [62].

2.3.2 Multi-Path Routing

In dense WMN topologies with multiple possible disjoint links and paths, multi-path routing algorithms can be used for load-balancing the traffic and increasing the total throughput. As nodes maintain multiple paths for each destination, the robustness of the network is also increased [63].

One key component in multi-path routing is traffic forwarding, which refers to the strategy of the node to select one or more out of the multiple next hops to send the packets to. When load-balancing the traffic, the forwarding component aims to fully utilize the capacity of the links to avoid bottleneck nodes/links. This is done by splitting traffic along two or more link-disjoint paths leading towards the destination. When using multi-path routing for increased robustness, the strategy is often to either forward all packets to one neighbor and keep the other paths in case the first one fails or send duplicate packets to all its neighbors, e.g. [64]. To know which neighbors it is possible to forward to, the route discovery finds for each source/destination pair a set of paths using a certain metric, avoiding paths that create loops. Only neighbors that belong to paths which fulfill the metric constraint, e.g. shortest hop, are included in the forwarding set. A packet scheduler in the nodes then forwards traffic over the multiple paths according to the chosen strategy, such as load-balancing or for increased robustness.

As an example, a flow-based forwarding strategy would pin all the packets of one flow to a single path. A packet-based forwarding strategy would distribute different packets from all flows amongst the paths found by the route discovery, without respecting which flow a packet belongs to. For TCP there is a trade-off between network and transport layer throughput when distributing traffic on each path. A per-flow forwarding might be unable to fully utilize the network, as with fewer (TCP) flows than paths it can be difficult to evenly load-balance the traffic within the network. Packet-based forwarding allows the routing
layer to use a more fine-grained forwarding and control over the resources compared to per-flow forwarding. However, load-balancing packets from the same flow over several paths with different delay properties might cause packets to arrive reordered. For TCP, packet reordering is a well-known problem that can seriously hamper application performance [65]. In paper VII we show that the clients must use as many TCP flows as there are bottleneck disjoint paths in order to fully utilize the available capacity. However, to exactly determine the number of bottleneck disjoint paths (and thereby the number of TCP flows necessary) without cross-layer support is difficult.

Horizon combines back-pressure scheduling with multi-path routing, where the lower layers are tailored to specifically support TCP traffic [66]. Horizon uses a heuristic back-pressure approach to obtain an 802.11-compatible packet forwarding scheme and light-weight path estimator. The authors also propose to use a delayed reordering algorithm that keeps TCP packet reordering to a minimum while avoiding TCP timeouts. As a system approach, Horizon achieves good results; however, as the approach focuses on a scenario where all nodes are located within a single WMN, it is not clear what effects the proposal would have when the sender (or receiver) is located in a wired network. Furthermore, the performance of Horizon depends on the quality of the routes, but the protocol is only evaluated with hand picked routes using two paths per flow. The authors do not consider how to find the optimal routes and how many paths to use.

In order to improve both capacity and reliability, each node in the network can be equipped with multiple radio interfaces using a set of orthogonal channels. This increases both network capacity and reliability, because it reduces contention on highly occupied links. IEEE 802.11b/g and IEEE 802.11a have three and thirteen (as defined by FCC) orthogonal (non-overlapping) channels [67]. There exist different strategies on how to assign channels to nodes. In dynamic channel assignment schemes, the channels are changed based on topology changes, traffic load or the current state of the medium. These schemes require that the channel changes are infrequent enough so that the channel switching delay becomes negligible [68]. This often also requires multiple radios, as when a radio is tuned to a particular channel, it cannot hear communication taking place on a different channel increasing the risk for collisions when switching to use a new channel [69] [70]. However, since the channel assignment influences the link capacity and therefore the routing strategy, routing and channel assignment should be jointly optimized [71] [72]. As jointly optimizing routing and channel assignment is a NP-complete problem, approximative methods must be used, e.g. [71].

A interesting heuristic approach to jointly optimize channel assignment and routing is proposed in [14] as Flow-aware Channel and Rate Assignment (FCRA) and Layer-2.5 routing algorithm. FCRA takes as input the expected load on links, and tries to assign channels to guarantee the required bandwidth to them. The outcome of the FCRA algorithm is a schedulable flow-rate for each link, where a flow-rate is defined in the form of a relative value, e.g. send 50% of the packets on link A. The Layer-2.5 routing algorithm distributes
traffic on outgoing links, which are on the discovered path, in proportion to these flow-rates. When a node has to take a routing decision for a certain packet, it calculates a set of potential next-hops, by comparing the distance of each neighbor from the destination with a max hop field contained in each packet. The next-hop is then selected from the set of potential next-hops, depending on the flow-rates of the links that connect to them. In order to avoid that packets are sent over too long paths, the flow-rates are weighted depending on the distance of the neighbors. Thus, flow-rates of links to neighbors that are closer to the destination are increased, and therefore those neighbors are more likely to be selected as next-hop. The original FCRA and Layer-2.5 was designed without taking into account packet aggregation. When considering packet aggregation, the decisions made by Layer-2.5 is not optimal performance wise since packet aggregation only works efficiently when packets are destined for the same next hop. Therefore, in paper V we extend the layer-2.5 routing algorithm by limiting the set of possible next hops to increase the aggregation possibilities. In paper VI, we modify the weights of the flow-rates to increase packet aggregation possibilities.

2.4 Aggregation

Transmitting small packets in IEEE 802.11 based networks leads to low performance because of a fixed MAC/PHY layer overhead. This favors sending fewer but larger MAC frames [73]. A well known technique to create larger MAC frames and thus to improve network throughput is to reduce the amount of small packets by using packet aggregation [74]. Aggregation can reduce the overhead and the number of attempts to access the medium by combining several smaller data units into a larger one. Several studies have considered the usefulness of packet aggregation on wireless networks in a great variety of operational conditions [75] [73] [74] [76].

To have enough packets to aggregate, it is common to introduce an artificial delay and queue packets [74]. The right choice of such aggregation delay is an important design parameter. When the traffic load is low a higher aggregation delay yields higher aggregation ratio, but also a higher end-to-end delay. Furthermore, with larger packets there is less contention but the negative impact of each dropped aggregated packet can be more severe as it might contain more than one packet.

For multi-hop networks there are more benefits than the sole reduction of MAC layer overhead. This is because aggregating more packets leads to a reduction of the overall number of packets in the network and thus reduces multi-hop contention. However, with multiple hops the additional delay will also be multiplied with each link, reducing the robustness with regards to choosing an aggregation delay.

An important design decision in multi-hop networks is the placement of (de)aggregation capabilities [76]. In end-to-end aggregation the sender/ingress node aggregates the packets while the receiver/egress node de-aggregates them. Intermediate nodes just forward the aggregated packets. In hop-by-hop aggreg-
End-to-end aggregation is transparent to the core network. This simplifies deployment as the aggregation functionalities only need to be implemented at the end-nodes, i.e. access points and gateways. The drawback is that valuable aggregation possibilities might be lost as no aggregation is attempted inside the network.

End-to-end aggregation has been shown to almost double the throughput for constant bit-rate traffic and improve throughput by 20% for HTTP traffic compared to an unaggregated case [77]. The approach in [77] uses an IP-based packet aggregation scheme that adaptively selects an optimal packet size as the goal for the aggregation, based on the route quality. Therefore, [77] is different from the one we present in papers II – VI in both that it depends on support from the routing protocol and that it operates end-to-end. In contrast, our approach is independent of the routing protocol and operates hop-by-hop, see section 2.4.2. Furthermore, [77] focuses on CBR traffic and does not analyze the impact on TCP traffic in detail.

A variant of end-to-end aggregation is the accretion aggregation algorithm, where forced delay is only added at the ingress node [76]. Different from end-to-end aggregation, the accretion algorithm also aggregates at intermediate nodes when the media access delay allows the queues to build up naturally. However, as we show in paper III, TCP creates enough traffic so that a fixed small forced delay performs adequate. When there is no congestion in the network, there is less benefit of aggregation with TCP traffic, regardless of the forced delay. Therefore, in paper III we argue that with TCP there is no additional benefit to distinguish between ingress and intermediate nodes over using hop-by-hop aggregation.

In hop-by-hop aggregation, every (back-haul) node de-aggregates and re-aggregates. This enables the network to combine packets to different (final) destinations into one packet as long as the next hop is the same. Hop-by-hop aggregation therefore allows for efficient inter-flow aggregation as it is possible to aggregate at each hop in the network. This can be seen in figure 5 where node 3 can aggregate together packets or frames destined to node 0, from both nodes 4 and 5 when forwarding them to the gateway.

A large forced delay can yield a higher aggregation rate in low traffic scenarios. However, it can also yield a higher end-to-end delay since the end-to-end delay increases on each hop. If the traffic volume is high enough, the delay by the natural medium access mechanism is sufficient to queue enough packets to fill an MTU and schedule the aggregated packet for immediate transmission. With enough traffic, the impact of the aggregation delay is thus reduced since the aggregated packet will be sent as soon as the medium becomes idle.
Hop-by-hop aggregation can be performed either as frame aggregation or as packet aggregation. The difference between frame aggregation and packet aggregation is mainly that the former operates at the MAC level and the latter operates at the IP-level, as discussed in the next two paragraphs.

**Frame Aggregation** Frame aggregation has shown to be beneficial for single-hop infrastructure WLAN as it considerably reduces the overhead of sending small packets [78]. Frame aggregation has also been incorporated into the
Transport-Layer Performance in Wireless Multi-Hop Networks

Figure 7: A-MPDU frame format in IEEE 802.11n

latest IEEE 802.11n standard [44]. Which defines two different but combinable frame aggregation modes depending on where the aggregation is done [44].

In the aggregated MAC service data unit or A-MSDU mode, the packets are aggregated when they enter the MAC layer. For an overview of the A-MSDU format see figure 6. Each A-MSDU combines multiple MSDUs into a single MAC protocol unit (MPDU) by adding a MAC header and a frame check sequences (FCS) field. The maximum A-MSDU size allowed by 802.11n is 7935 bytes [44]. When using A-MSDU, a receiver can acknowledge an entire A-MSDU frame by a single acknowledgement (ack) frame, thus reducing the overhead. The disadvantage of A-MSDU is that on a transmission error, the entire A-MSDU needs to be retransmitted [79].

In the aggregated MAC protocol data unit or A-MPDU mode, packets are aggregated when they leave the MAC layer (figure 7). When using A-MPDU, each frame can contain up to 65535 bytes of payload (at the PHY layer). Each A-MPDU consists of one or more A-MPDU subframes padded to a multiple of 4 bytes, except the last subframe. Each A-MPDU subframe consists of an MPDU delimiter followed by an MPDU. The purpose of the MPDU delimiter is to be able to locate the MPDUs within the A-MPDU when one or more MPDU delimiters are received with errors. IEEE 802.11n mandates the possibility to acknowledge multiple MPDUs with a single ack frame called block acknowledgments (BA), as first presented in IEEE 802.11e [80]. When using BA combined with A-MPDU, selective retransmission of subframes can be done. This can be very useful in environments which have a high number of collision or transmission errors [79].

Besides 802.11n there are also other proprietary techniques for frame ag-
aggregation, such as [81] and [78]. The drawback with all frame aggregation techniques is that they require a change of the underlying MAC/PHY layer. In contrast to packet aggregation, frame aggregation can not be deployed without changing current 802.11a/b/g hardware [82].

**Packet Aggregation**  Packet aggregation operates at the IP layer. Hence, it can be used with any IEEE 802.11 hardware and can therefore be more easily deployed and integrated with existing wireless networks. The main drawback of packet aggregation is that MAC specific performance enhancements like block acknowledgements are not easily replicated at the IP layer.

The packet aggregation algorithm that we are using in papers III – VI is based on earlier work on improving VOIP capacity in WMNs [10]. In contrast to the more aggressive approach in [10] of serving the queue with the most packets first, the aggregation scheme first proposed in paper III improves flow fairness by adjusting the queuing strategy so that the queue with the oldest packets is always served first. The aggregation algorithm is controlled by $\text{AggregationMaxDelay}$, which denotes the maximum forced delay for a single packet. This parameter induces artificial delay and increases the number of packets in the queue when the network traffic is low and thereby increases the aggregation ratio. For a detailed description of the algorithm see paper IV.

The frame aggregation strategy most similar to our packet aggregation scheme is A-MSDU. Due to MTU limitations in IEEE 802.11a/b/g, we have limited the size of the aggregated packets to 2304 bytes. However, due to the separate responsibilities and modularity between the concatenation of packets into larger packets and the virtual queue structure, it would be possible to delegate the concatenation and de-concatenation step to the MAC layer while keeping the rest of the functionality as a thin layer between IP and the MAC layer. This could reap the benefits of both IEEE 802.11n (e.g. block acks and higher MAC payload) and our packet aggregation scheme (e.g. the high aggregation ratio achieved by delaying packets) in spirit of the approach in [83].

While the benefit of a forced delay is evaluated in [74], the authors have focused on UDP and concluded that aggregation does not harm TCP throughput. In paper III and IV we have extended the results by an in-depth evaluation of TCP performance, building on the knowledge about forced delay obtained in [74].

Aggregation with QoS support has also been shown to be beneficial in multi-hop scenarios with mixed UDP and TCP traffic, where VOIP capacity increased greatly [84]. The algorithm proposed in [84] uses four queues to divide packets according to their priority class, ranging from best effort to high priority packets. The authors of [84] restricted their aggregation scheme to only aggregate packets within the same traffic class, focusing on QoS and VOIP performance. In contrast, in papers III and IV we concentrate on a detailed study of TCP behavior in multi-hop networks.

The benefits of robust header compression combined with packet aggregation are evaluated in [85]. The experiments showed that header compression alone can improve the throughput by 20 %. However, up to 10 times improve-
Transport-Layer Performance in Wireless Multi-Hop Networks

Efficient transport can be achieved by combining header compression and packet aggregation in a small WMN [85]. In paper III, we focus on a similar topology but instead of UDP packets of 20 bytes, which are easily aggregated, we focus on much larger (up to 1460 bytes) TCP packets.

In paper [86] the authors propose a novel aggregation and back pressure scheduling approach that combines the fairness and good network utilization of back-pressure scheduling with packet aggregation. The main idea of back-pressure scheduling is that contention among links should be resolved by scheduling the link that has the largest product of queue differential backlog between its neighbors and transmission rate of the link. The combination of back-pressure scheduling and aggregation can improve UDP throughput up to 2.5 times compared to using only back-pressure scheduling. Although the proposal in [86] gives a significant improvement, it requires changes to the MAC drivers to support the MAC layer prioritization required for back-pressure scheduling. Our work in papers V and VI has a similar intention of integrating aggregation with packet forwarding. However, in paper V and VI we combine aggregation with multi-path routing, which does not require changes to the MAC driver. We further evaluate performance with both TCP and UDP traffic.

2.5 TCP

TCP [87] is a connection-oriented end-to-end transport protocol that provides end-to-end flow and congestion control. It guarantees reliable, ordered delivery of data. TCP utilizes sequence numbers for detecting packet loss and reordering. A receiver acknowledges data-packets as they arrive, and if the sender receives no acknowledgement (TCP ack) for a TCP data-packet within a predefined time frame, the packet is assumed lost and retransmitted. The congestion window reflects the available network capacity, while the advertised receiver window reflects the rate at which the receiver can process the received data.

TCP’s congestion control mechanism was designed to react in a robust way to changing network conditions that can occur in a wired network. However, the characteristics of wireless networks and especially wireless multi-hop networks are significantly different compared to wired networks. Topology changes, route breaks and low link quality may result in high and fluctuating packet loss rate and delay. Those effects significantly deteriorate TCP’s performance.

2.5.1 Protocol Basics

TCP’s congestion control [88] probes the network by sending more and more data until the network starts to drop packets. The congestion window (cwnd) starts with maximum four segments and increases as more data is acknowledged [89]. However, recent proposals have suggested that the initial cwnd should be raised to ten segments to improve the performance when browsing the web, such as using web search engines [90]. For every acknowledgment received the cwnd is increased by one segment, resulting in an exponential growth of
the cwnd. The amount of data the sender can transmit is the minimum of the cwnd and the receiver’s advertised window. During this “slow start” phase the sending rate can quickly approach and surpass the network capacity.

When a loss is indicated, either by the expiration of the retransmission timer (RTO) or by the reception of three duplicate acknowledgments (DUPACKs), half the value of the current cwnd is recorded in the slow start threshold (ssthresh). If the loss was triggered by a RTO timeout, cwnd is reset to one and the transmission is resumed in the slow start phase. If the loss is indicated by receiving three DUPACKs, cwnd is set to the value of sstresh. When the value of cwnd reaches the value of ssthresh without packet loss events, TCP enters the congestion avoidance (CA) phase. In CA, the cwnd grows linearly by one segment per round-trip time (RTT) until loss occurs. When loss is detected, the congestion window is cut in half. The result is the typical saw-tooth behavior that oscillates around the network capacity [91].

The main factors that influence congestion control and thereby determine TCP throughput are round-trip time (RTT) and packet loss [92]. When RTT decreases, packet loss ratio increases, under the assumption that the throughput is the same or higher. For TCP performance with a given loss rate, packet loss occurring in bursts is less severe than random packet losses [93]. Since packet aggregation will make the packets arrive to a node in bursts, the losses will intuitively also be more bursty. This is because if the queue is full and an aggregated packet arrives, more than one of the received TCP packets will be dropped.

**Congestion Control Algorithms** The goal of TCP’s congestion control mechanism is to provide efficient and fair sharing of the resources while avoiding congesting the network. There exist several variants of TCP’s congestion control. One of the most well spread variants is TCP NewReno [8] that modifies TCP’s behavior after receiving three duplicate acknowledgments (DUPACKs). The purpose of TCP NewReno’s modification is to keep the transmit window full during loss. A DUPACK indicates that a packet has been received at the destination and that it is safe to send a new packet. Furthermore, when an TCP ack arrives, confirming only parts of the packets in the cwnd, TCP NewReno assumes that this TCP ack points to a loss hole in the sequence space and a new packet beyond the confirmed sequence number can be sent. This allows NewReno to maintain a high throughput while retransmitting lost packets in the sequence number space.

TCP Vegas [9] has a different approach to probe for available bandwidth as it pro-actively adapts the cwnd to avoid packet loss. The idea is to measure and control the amount of extra data that a connection has in transit, which would not have been sent if the bandwidth used by the connection exactly matched the available bandwidth of the link. If too much extra data is sent, congestion will arise. If too little extra data is sent, the reaction to a transient increase in available bandwidth will be delayed. TCP Vegas computes the expected flow rate by using the current window size and the minimum round trip time as an estimation of the propagation delay of the path. The current flow rate is
Transport-Layer Performance in Wireless Multi-Hop Networks

calculated using the actual measured round trip time. The difference between the expected and the current flow rate is used to update the cwnd. Analytical models and simulations with wired networks have shown that TCP Vegas measures congestion by end-to-end queuing delay \[94\]. If the queue sizes at intermediate nodes are large, TCP Vegas can improve goodput and packet loss ratio. In the case of inadequate queue sizes, TCP Vegas cannot utilize its improved congestion detection mechanism and reverts to the same behavior as TCP \[95\].

TCP Westwood+ \[6\] is a sender side modification of TCP NewReno that is intended to better handle large bandwidth delay product (BDP) paths with potential packet loss due to transmission or other errors. This property makes TCP Westwood+ very attractive to use in wireless systems. TCP Westwood+ relies on monitoring the TCP ack stream for information to help set the congestion control parameters, i.e. slow start threshold (ssthresh) and congestion window (cwnd). Due to mobility and rerouting, the TCP ack stream will have larger fluctuations in a MANET than in a static network. These (none) congestion related fluctuations can reduce the ability of TCP Westwood+ to correctly determine the available throughput.

TCP with Explicit Link Failure Notification (ELFN) \[7\] is a cross-layer approach to inform the TCP layer about route failures. TCP ELFN was designed for DSR \[96\] and uses an ELFN message, which is transported by or piggybacked on routing messages to the sender upon a route break. On receiving the ELFN message, the source responds by disabling its retransmission timers and entering a “frozen” state. During the “frozen” period, the TCP sender probes the network to check if the route is restored. When the route is restored (i.e. the sender starts to see acknowledgments of the probe packets); the TCP sender leaves the “frozen” state and resumes its state as before the freeze event. Upon route restoration, the proposals uses the values of RTO and cwnd from prior to the route failure. In paper II, we extended TCP ELFN with support for AODV and OLSR.

TCP with adaptive pacing (AP) \[20, 97\] is designed for wireless multi-hop networks with both mobile and static nodes. It implements rate control by adaptively pacing the send rate at the sender side while retaining TCP end-to-end semantics. The main benefit of TCP-AP is a substantially reduced MAC layer contention for packets within the same flow traversing multiple nodes belonging to the same collision domain. In the earlier work \[20\], TCP AP assumes a string of at least three hops in between the sender at receiver. It calculates the maximum send pace as the time elapsed between transmitting a TCP packet by node \(i\) and receiving the packet at node \(i+4\). In \[97\], the authors extend the send pace calculation to also take into account propagation delay. Instead of using a static delay variable (4 hops), the new algorithm calculates an approximate of the out-of-interference delay according to an estimate of the carrier sensing range of the TCP source node. As shown in both \[20\] and \[97\], TCP AP outperforms TCP NewReno. However, the approach requires the senders to be located in the multi-hop network. This is normally not the case as most servers are located in the Internet. Due to this, the authors of TCP-AP...
Introductory Summary

proposed TCP-GAP, which uses a pacing buffer at the gateway to pace the TCP flows that originate in the wired Internet [98]. This removes the need for changes to existing Internet hosts at the cost of a more complex gateway design. However, TCP-GAP still requires changes to the wireless clients as they are required to run TCP-AP.

Packet Reordering  Packet reordering is a rather common event that poses negative effects on TCP. Reordered data can be misinterpreted as packet loss by TCP [88] causing unnecessary retransmissions and invocation of congestion control procedures. Retransmissions and reductions of the congestion window only occur if the reordering is severe enough to generate three duplicate acknowledgments at the receiver. However, shorter reordering events also affect TCP. For example, as TCP guarantees in-order data delivery, all packets arriving out-of-order must be buffered, potentially requiring large receive buffers.

In order to mitigate the performance issues related to reordering, a number of approaches can be applied at the transport-layer. Typically, it is possible to classify a reordering mitigation approach as reactive, pro-active or mixed. Reactive schemes, such as [99], are often designed to revert TCP state changes that were made under the false assumption that packets were lost, when in fact reordering occurred. This type of mitigation technique can help TCP to maintain the sending rate by restoring the congestion window when reordering is detected. Pro-active schemes, such as [100] [101] are instead designed to inhibit reordered packets from triggering the TCP loss recovery and thus the congestion control. To accomplish this, such proposals often extend the loss detection phase in TCP to allow more time to distinguish between loss and reordering. Practically, such loss detection extensions are often achieved by increasing the number of duplicate acknowledgments needed to trigger a fast retransmit, i.e. the duplicate acknowledgment threshold. The increase can be based on previously observed reordering events, on the amount of outstanding data or other relevant metrics. As the reactive and pro-active approaches are fairly orthogonal, many mitigation schemes are designed to mix them to offer better overall robustness to reordering, e.g. [102], [103].

2.5.2 TCP in Wireless Multi-Hop Networks

The standard TCP congestion control algorithm cannot distinguish between packet loss caused by congestion, interference or routing layer issues in wireless multi-hop networks [104]. In MANETs, a significant portion of packet losses are caused by link failures due to mobility of nodes resulting in route errors. Route errors can also happen in MANETs and WMNs because of high contention and the complex cross-layer interaction between the MAC, routing and transport layer when using TCP [29].

Several solutions for TCP in wireless multi-hop networks as well as new transport layer protocols have been proposed and a good overview can be found at e.g. [105]. In Internet connected wireless multi-hop networks, it is
also possible to run a specialized transport protocol in the wireless network and split the connection at the gateway. Although several such approaches exist, e.g. MTCP [106], splitting the connection at the gateway violates TCP end-to-end semantics and can incur high overhead and difficulties with handling handover between gateways [107]. It is therefore important in Internet connected multi-hop networks to provide good application performance with standard TCP.

TCP in MANETs In a MANET, when a node moves closer to the destination and/or a new shorter route is found, the available bandwidth increases as the traffic traverse less number of hops. Therefore, it is crucial to quickly determine the available bandwidth when the routes are changed [7]. In such circumstances, a too slow increase of the congestion window and probing of available bandwidth is not beneficial as precious resources are wasted [108]. A too fast increase is also not beneficial as the probability for MAC layer contention induced packet loss increases. This might lead to lost data and routing packets, which may cause the routing protocol to trigger route error messages even though the routes are still valid, thus increasing the problem even more.

However, different routing protocols react differently on packet loss. This follows from Paper I, where TCP Vegas [9] less aggressive approach performs slightly better than TCP NewReno’s more aggressive approach when using a reactive routing protocol (AODV) and vice versa with a proactive (DSDV) routing protocol. In our simulations, an aggressive probing can quickly utilize the full bandwidth of a new path, while a less aggressive probing reduces the amount of packet loss. Therefore, a more conservative routing protocol, which tolerates more packet loss and thereby switches later to a new path, benefits from a more aggressive probing. A more agile routing protocol, which quickly switches to a new path might more easily mistake packet loss as a route error and thereby performs better with a less aggressive probing.

In [109], the authors compare TCP performance using DSR and DSDV in a MANET. The simulation results show that both routing protocols could provide a similar TCP throughput. However, when using DSR the source node had for a large number of packet transmissions a stale route. This is caused by channel errors that corrupts the route reply packets causing caches to expire without being refreshed. When the source finds a stale route in the cache it invokes route discovery to find new routes, which may lead to route fluctuations. In case of DSDV, when the routing table becomes stale due to lost routing messages, they are not refreshed until the next periodic routing table update. This can lead to long routing outage and high packet losses. Therefore, when using DSR, TCP performance was limited by out-of-order packets while when using DSDV the TCP performance was limited by packet loss.

TCP in Tactical MANETs A tactical MANET typically uses TDMA, which gives a more predictable delay than using a probabilistic MAC layer such as CSMA. This is favorable for TCP performance as TCP is indirectly influenced by the retransmission timer (RTO). The RTO keeps track of how long TCP
waits for an acknowledgement and when TCP should retransmit. The RTO value is doubled each time the RTO expires. Less delay jitter reduces this timer leading to quicker detection of packet loss and improved performance.

In [110] TCP performance is evaluated in a TDMA based satellite access network. The authors propose a new TDMA protocol called CA-GRAP, which is compared to both slotted ALOHA and [111], on which CA-GRAP is based. CA-GRAP employs cumulative acknowledgments and a contention index that controls how many packets that can be transmitted in each slot. Compared to slotted ALOHA, CA-GRAP increase the throughput by using a combination of random and reservation access. Due to the contention index and cumulative acknowledgment, CA-GRAP can also increase the efficiency and reliability over [111]. The results from these experiments are not directly comparable to our setting in paper II as a satellite system has different mobility, delay and bandwidth characteristics from a multi-hop MANET.

In [112], the authors propose a new TDMA based MAC, routing and time synchronizing protocol. The main novelty is the synchronization mechanism that allows network wide clock synchronization of all pairs of clocks at microsecond granularity. The authors evaluate TCP throughput over a static two-hop star topology. The proposed TDMA protocol achieves close to optimum TCP throughput when the bit error rate is low. However, as the protocol does not retransmit lost frames, TCP throughput degrades significantly when bit error rates go up. The main difference in these two evaluations from our TCP evaluation in paper II is that we use a different TDMA protocol, use mobile clients and evaluate the TCP performance using two different routing protocols.

The authors of [113] propose a novel reactive QoS routing protocol for TDMA based MANETs. With the proposed protocol, the authors show that it is possible to search for and establish routes in a MANET supporting a given bandwidth constraint. The bandwidth requirement is realized by letting the route reply reserve time slots as it traverses links on the reverse path. Our work in paper II differs in that we evaluate TCP and different routing protocols that are independent of the MAC layer.

**TCP in WMNs** The user and traffic pattern in a general purpose WMN is expected to be similar to that of today’s Internet. Therefore, a WMN should provide similar service at a lower cost than current WLAN deployments. This places high requirements not only on throughput but also on delay and fairness. In a WMN, the core nodes are stationary and route breaks caused by e.g. mesh node failures should rarely happen. We furthermore assume that clients connected to a WMN use the same mesh access point during the duration of their TCP connection(s). Even if there is no mobility induced loss in a WMN, the complex cross-layer interaction between the MAC, routing and transport layer can provoke route breaks and trigger route error/repair messages [29]. When updating the route, route repair messages contribute to the contention and during this time all traffic for the affected flow(s) is stopped, which might lead to a TCP timeout [29]. This effect is more pronounced in a
Transport-Layer Performance in Wireless Multi-Hop Networks

WMN since the bandwidth requirements are higher. Furthermore, since there are in general less route breaks (due to mobility) in a WMN, TCP will operate with a larger cwnd than in a MANET. This increases the number of packets in flight and thereby the contention. When TCP starts to react, the routing protocol might already use a different route. This causes high fluctuations of the cwnd and unnecessary route changes [29] [114].

Without rate control, a TCP flow from a node closer to the destination traverses less hops and therefore will experience a lower loss rate than flows from nodes further away. This can lead to a situation were the flow closer to the destination starves the other flows. To avoid this, the authors of [115] propose to use a Cooperative Neighborhood Airtime-limiting (CNA) mechanism that explicitly allocates airtime transparent to TCP and IEEE 802.11. Simulation results show that CNA on average performs within 12 % of the optimal on the topologies evaluated. In paper IV, we could observe that if nodes applied an aggregation delay when aggregating packets, the nodes delayed their transmissions. As a result, the available bandwidth was shared more equally between the flows. An interesting alternative that is left for future work is to combine packet aggregation with the airtime limits calculated in [115] to dynamically set the aggregation delay.

2.5.3 Multi-Path Transport Protocols

Multi-path transport protocols allow a single connection to be striped across multiple paths over the network. Since such load-balancing is implemented within the transport layer, it is normally transparent to applications. Multi-path transport protocols have several use cases stretching from offloading traffic from 3G to WiFi, managing traffic within a data center and reducing the congestion in access networks. Such protocols achieve this by resource pooling and allowing multiple links to behave as a single pooled resource [116]. The aim is to increase reliability and efficiency by moving traffic away from congested or broken paths. To accomplish this, each path needs to be monitored and the traffic divided on the paths accordingly. The most obvious benefit of using multiple paths simultaneously is the throughput increase resulting from the aggregated capacities of the paths.

The idea of using multiple paths is not new and there are several proposals to provide multi-path capabilities at the transport layer. For example, SCTP [117] maintains a backup path for increased reliability in case the primary path goes down. Although this improves reliability it does not improve efficiency. A proposal that aims for improving both efficiency and reliability is MPTCP [16] [118].

MPTCP is a major TCP extension that stripes a single connection (seen from the application layer), over separate subflows. One of MPTCP’s goals is to provide a drop-in replacement for TCP that requires no modifications of current applications or networks. MPTCP transparently divides a TCP connection into subflows and distributes them over a host’s network interfaces. MPTCP’s capabilities and parameters are negotiated whenever two TCP peers perform their initial handshake. After the establishment of a connection, any
host can establish new subflows. The subflows can either use the same pair of IP addresses as the first subflows, with different ports, or make use of any additional IP address that is available on either peer.

When multiple subflows have been created, the sending peer is responsible for load-balancing the traffic across all subflows. Although each subflow is a TCP connection and maintains and uses its own congestion control parameters, MPTCP tries to link all subflows and steer traffic away from congested paths by using the algorithm specified in equation 1 [119]. Here, $w_r$ is the congestion window on subflow $r$ and $w_{\text{total}}$ is the sum of the windows on all subflows. The parameter $a$ determines the aggressiveness of the algorithm and is calculated as described in [119]. There are two key parts to this algorithm. First, by making the window increase depending on the total window size, subflows that have large windows increase faster than subflows with small windows. This actively moves traffic from more congested paths to less congested ones, i.e. load-balancing the network. Second, by adapting $a$, MPTCP can compensate for different RTTs and can ensure that if all the subflows of a connection traverse the same bottleneck, they will compete fairly with a regular TCP flow. However, if the subflows encounter multiple unloaded paths, one connection should be able to fill them all (if enough subflows are used). The design of the coupled congestion control algorithm has been detailed in [120].

$$w_r = \begin{cases} w_r + \min \left( \frac{w_r}{w_{\text{total}}}, \frac{1}{a} \right) & \text{all TCP ack on subflow } r, \\ w_r - \frac{w_r}{T} & \exists \text{loss on subflow } r. \end{cases}$$

(1)

The failure of a path looks like severe congestion to the transport-layer, and MPTCP’s reaction is to shift traffic to other paths using its congestion control algorithm (see above). MPTCP also improves robustness in mobility scenarios by enabling handovers, using separate simultaneous subflows for the old and new paths. However, the main benefit is that MPTCP’s congestion control will automatically balance traffic over the available paths to avoid congestion. Compared to TCP’s congestion control, which only lowers the sending rate over congested paths, MPTCP actually moves traffic from such paths. This enables efficient operation of flow-based multi-path routing algorithms.

There is a slight increase of memory and processing power required when using MPTCP [121]. When the subflows of an MPTCP connection traverse paths with different characteristics, data can be reordered. To reconstruct the original send order an extra level of sequence numbering and stream reconstruction is introduced. The degree of buffering and stream reconstruction increases with the number of subflows [121]. This may lead to problems for memory constrained devices, such as mobile phones.

Furthermore, single-homed hosts cannot easily benefit from MPTCP’s ability to load-balance traffic over several subflows, as subflows typically are created on a per-interface basis. Some effort has been made to construct mechanisms that solve this problem for single-homed hosts. In [122] the amount of disjoint paths is conveyed to MPTCP using an optional DHCP field,
informing MPTCP to open a corresponding number of subflows. In [123] [124], middleboxes with multiple network interfaces are used to transform a regular TCP connection to a MPTCP connection with multiple subflows going over the network. Both approaches (slightly) invalidate MPTCP’s goal of compatibility with legacy networks and rely on dependencies between end-nodes and the network infrastructure to function. Research on specific environments like data-centers have given a “rule-of-thumb” value of eight for the number of subflows required to utilize all bottleneck disjoint paths [125]. Although it is in certain environments possible to a priori estimate the optimal number of subflows, it is much more difficult in the general case. In paper VIII, we propose a general transport-level algorithm that is able to estimate the number of subflows to fully utilize all available bottleneck disjoint paths without requiring a priori or explicit cross-layer information. The proposals can operate with any multi-path routing protocol that does flow-based load-balancing.

### 3 Summary of Papers

This section contains summaries of the papers included in this thesis.

**Paper I – TCP Performance in Mobile ad hoc Networks Connected to the Internet**

In this journal article we investigate the performance of three different TCP variants in a wireless multi-hop network connected to the Internet using static gateways. We consider small to medium sized topologies with both mobile and static single-radio nodes. In the evaluation we use: two single-path routing algorithms, AODV by Uppsala University (AODV-UU) [21] and DSDV; four traffic patterns and three TCP variants, TCP Vegas, TCP NewReno and TCP-AP. In total, we perform 96 different simulations over three different topologies.

We show that TCP Vegas reduces the amount of packets in flight and the number of false route errors while providing the same or better performance as compared to TCP NewReno and TCP-AP. TCP Vegas’ more accurate bandwidth estimation gives it a slight advantage over TCP NewReno’s more aggressive but less accurate estimation. We also highlight that TCP NewReno can create excessive route flapping, which is due to its aggressive probing for bandwidth. We also show that although the overall performance was slightly lower with DSDV than with AODV-UU (with mobile nodes), the advantage gained by using TCP Vegas is reduced with DSDV as routing protocol. This is because TCP NewReno could more quickly utilize the temporary shorter routes when the node moved closer to the gateway.

The main limitations of this article are the synthetic traffic models and topologies used. We used an FTP type traffic pattern that highlights the problem areas addressed, which is only to a limited extent comparable to the more diverse traffic patterns found in a real system.
Paper II – Performance Evaluation for TCP in Tactical Mobile ad hoc Networks

Emerging tactical networks need to support high-data rate TCP such as applications for providing digital images and maps. However, this must be implemented without endangering the services and stability offered by traditional TDMA based tactical networks. Therefore, in paper II we focus on the performance of TCP in a TDMA based tactical MANET. We implement support for AODV and OLSR in TCP ELFN and compare the performance of TCP NewReno, TCP Westwood+ and TCP ELFN using OLSR and AODV. In the experiments we used a TDMA based MAC layer with DRAND as slot scheduler.

TCP ELFN freezes its cwnd upon information from the routing layer until the route is reestablished. It achieves the overall highest throughput irrespective of the routing protocol. However, the gains are reduced in low throughput scenarios as the cwnd is already low and the benefit of freezing the cwnd thereby becomes less. We further show that the two other TCP variants react differently depending on which routing protocol is used. With a low node density there is almost no impact of the routing protocol. However, in high node density scenarios TCP NewReno performs best with AODV, while TCP Westwood+ performs best using OLSR.

The main limitation of this paper is that the TDMA scheduler that we used was not specifically targeted towards TCP in tactical MANETs. When the mobility pattern and traffic is known, a modified TDMA schedule could obtain better performance, by e.g. taking TCP ack-packets into account. Therefore, the paper does not display the characteristics of a fully optimized tactical network. The paper instead shows the performance of TCP using a general TDMA scheduler that works both for stationary and mobile networks.

Paper III – Impact of Packet Aggregation on TCP Performance in Wireless Mesh Networks

In this paper we examine the possibilities of using packet aggregation to better support standard TCP traffic in small Internet connected single radio WMNs with no hidden nodes. We also introduce our novel packet aggregation algorithm. It captures the key parameters for aggregating TCP traffic and favors packet order and fairness over aggregation possibilities. We evaluate the algorithm in an arrow topology and use standard TCP NewReno (with selective [127] and delayed acks [128]), both with and without deploying packet aggregation. We simulate large file transfer(s) with an FTP like application using two different maximum segment sizes (MSS), 1460 and 536 bytes.

The simulation results show that adding a small artificial delay of 10 ms when aggregating increases TCP goodput up to 70 % and reduces round trip time (RTT) up to 40 %. This is because of the improved MAC layer performance with fewer but larger packets.

The main limitations of this paper are the size of the WMN topology, the consideration of only good links, where all wireless bit errors could be
recovered by MAC layer retransmissions, and the synthetic traffic models used. We further increased the number of allowed “hello”-messages lost in order to achieve stable routing, as our simulations should not be influenced by instability of the routing layer.

Paper IV – TCP Performance Evaluation in Wireless Mesh Networks using Packet Aggregation

In this article we give a detailed description of and evaluate the proposed packet aggregation scheme in single radio WMN topologies with hidden nodes. We examine the packet aggregation algorithm with a diverse set of topologies and traffic patterns, including patterns based on wireless traces and patterns well-known for causing TCP performance problems. The user scenarios are both an FTP like file transfer, using TCP maximum segment sizes (MSS) of 1460 and 536 bytes, and a trace-based approach using a TCP MSS of 1460 bytes.

We show that with FTP like traffic, packet aggregation increases TCP goodput up to 60% and reduces the impact of the TCP packet size on goodput. We further show that aggregation can decrease TCP round trip time in loaded networks by up to 30%. In “parking lot” scenarios we could also see a slight improvement of fairness by using aggregation with forced delaying of packets. In the trace based simulation, in which many TCP flows do not enter steady state, the TCP performance improvement is less than in the FTP like scenarios. However, the simulation does highlight the importance of the traffic pattern where the highest gains by aggregation can be seen when there are many simultaneous TCP flows in the network. We further show that the total amount of packet loss was similar with and without packet aggregation. The burstier packet losses experienced with aggregation gives an increase of TCP goodput as compared to not using packet aggregation. This is because random losses are worse for TCP performance than more correlated losses [93], for a given packet loss rate. We also show that aggregation does not change the ratio between TCP packets resent due to fast retransmit and timeout.

The main limitations of this article are that we only include good links and neglect the influence of routing. We used topologies and techniques (similar to those in Paper II) in order to achieve good links and stable and fixed routes. This isolates the effects we study, but in a real system instabilities and efficiency of the routing protocol and links must be considered. In this paper we also did not consider the use of multiple radios/channels.

Paper V – An Aggregation Aware Multi-Path Forwarding Paradigm for Wireless Mesh Networks

In this paper we extend the previous work to cover multi-path routing over multi-channel/radio nodes. Deploying packet aggregation in multi-channel multi-path environments may result in suboptimal performance. This is because multi-path routing algorithms spread packets among different next-hop neighbors with the aim of achieving efficient load-balancing, whereas only
packets that are forwarded to the same next hop can be aggregated. We propose two new aggregation aware forwarding paradigms for multi-path routing. The proposed paradigms try to either maximize throughput or minimize aggregation delay. We evaluate the approaches using both TCP and UDP traffic in two random topologies. UDP throughput is increased by around 90% and TCP throughput by around 30%. Although the throughput increase with UDP is promising, TCP performance cannot fully benefit from the increased network capacity. This is due to a massive amount of reordering of up to 30% of the packets. This shows that the forwarding paradigms significantly increase network saturation at the cost of a higher packet reordering.

The main limitation of this paper is the lack of detail in the evaluation of the difference between UDP and TCP throughput. The proposals are also limited in that they give an unbalanced weight towards packet aggregation possibilities, which can reduce the effectiveness of the load-balancer. We also did not address the packet reordering created by the forwarding paradigms. These issues are however addressed in paper VI and paper VII respectively.

**Paper VI – Combining Multi-Path Forwarding and Packet Aggregation for Improved Network Performance in Wireless Mesh Networks**

In this paper we propose a new routing and forwarding scheme that exploits the FCRA channel assignment algorithm and extends Layer-2.5 to be aggregation aware. The proposed scheme, Aggregation and Flow-rate aware Layer-2.5 Routing (AF-L2R), analyses the state of the sending queues and modifies the flow-rates assigned to outgoing links in order to increase packet aggregation ratios. Compared to our previous work in paper V, wherein the algorithm limited the available set of possible next hops so as to favor aggregation possibilities, AF-L2R instead operates directly on the flowrates. This allows a more fine grained control of the balance between packet aggregation ratio and link utilization. It also reduces the likelihood of using too long or too utilized paths at the expense of a slightly lower packet aggregation ratio.

We evaluated two main scenarios: a TCP based FTP scenario and a UDP scenario based on a packet trace. The FTP traffic scenario represents a worst case scenario for our proposed scheme, as aggregation only has limited impact on the performance and any deviation from the assigned flow-rates can reduce the throughput. With properly tuned parameters our proposed scheme performs slightly better than the other evaluated schemes. It provides a good balance between the aggregation achieved, the length of the path used and the packet loss ratio. In the UDP scenario our proposed scheme outperforms the other evaluated schemes, including the scheme we proposed in paper V.

The main limitation of the paper is the lack of an in-depth TCP evaluation and analysis of the difference between UDP and TCP performance. The proposal does not evaluate the possibility of using flow-aware routing or multi-path capable transport layer protocols.
Paper VII – The Interaction between TCP Reordering Mechanisms and Multi-Path Forwarding in Wireless Mesh Networks

In this paper we focus on the use of multiple paths to achieve load-balancing and thus increased application performance. We develop and evaluate both flow-based and packet-based path allocation using a general multi-path routing protocol that requires only flow information about the upper layers. We use two topologies in the paper where one of the topologies is based on a subset of a commercial WMN deployed in the town of Chaska, Minnesota \[129]\.

We further evaluate three different TCP implementations, a TCP variant without mechanisms to mitigate reordering effects, the standard Linux TCP implementation with built-in reordering mitigation techniques and TCP with Non-Congestion Robustness (NCR) \[100]\ extensions.

Our findings show that TCP reordering mitigations can not handle the amount and type of reordering that occur in a typical WMN using packet based allocation. Therefore, when the amount of TCP-flows gets close to the amount of bottleneck disjoint paths, flow-based path allocation becomes superior to using packet-based allocation, regardless if TCP is equipped with mechanisms to mitigate the effects of reordering or not.

Due to the different possibilities for multi-path routing, different path lengths etc., the TCP performance is highly related to the network topology. The main limitation of this paper is therefore the limited number of topologies evaluated. Furthermore, the evaluation is also limited to single-path TCP.

Paper VIII – MPTCP PathFinder — Finding Your Way(s) to Aggregated Bandwidth

In this paper we present the PathFinder algorithm. PathFinder is a general algorithm suitable for all types of multi-path networks where traffic might be routed over disjoint paths. The algorithm estimates the number of subflows required to fully utilize the network capacity by opportunistically trying to open new MPTCP subflows as long as this contributes to the total throughput. Thereby, PathFinder enables single-homed hosts to reap the benefits of MPTCP without modifying current network infrastructure.

To evaluate PathFinder we implemented it as a kernel module in Linux 3.2 and conducted extensive experiments using real traffic sent over a simulated WMN topology. The evaluation shows that PathFinder is able to open enough subflows to fully utilize all bottleneck disjoint paths, and thereby increases the throughput significantly when compared to using standard MPTCP.

The main limitation of this paper is the topology and traffic pattern used to evaluate PathFinder, which does not take into account the impact of MAC layer contention and routing layer issues specific to WMNs. A further limitation is the lack of CPU and memory consumption measurements compared to the amount of opened subflows on related hardware.
4 Conclusions and Future Work

Wireless multi-hop networks allow for rapid and flexible deployment, making them attractive for numerous applications ranging from broadband Internet access to surveillance systems. Like WLANs, wireless multi-hop networks offer the major advantage that they can be deployed in places where wired communication is too costly or otherwise impossible, e.g. inside historical buildings. However, good TCP performance is required to enable wireless multi-hop networks to be as successful as Ethernet and WLANs are today.

In this thesis, we investigate how we can improve TCP performance in wireless multi-hop networks. In the first part of the thesis we investigate the performance of different TCP versions using both fixed and mobile nodes. We also evaluate the interaction between routing and MAC layer using both CSMA and TDMA based protocols and how this influences TCP performance. We show in paper I, that TCP can create extensive route flapping while probing for available bandwidth. We further show that the benefit of freezing the congestion window on route breaks is dependent on the throughput and that the benefit is less in low throughput scenarios.

We propose packet aggregation and aggregation aware multi-path forwarding as two complementary methods for improving network layer services for good TCP performance. Our packet aggregation algorithm can be deployed with current hardware as an IP layer solution, or with slight modifications be used to complement existing MAC layer solutions, such as IEEE 802.11n. The novel aggregation aware forwarding is independent of the aggregation technology and requires only a minimal exchange of cross-layer information. We show that the aggregation aware forwarding combined with packet aggregation can improve the performance of TCP in multi-radio multi-path environments.

We further evaluate how the amount and type of packet reordering in a WMN impacts TCP performance. We show that using a traffic forwarding strategy that operates on TCP flows provides the highest TCP throughput under the assumption that there is at least the same number of TCP flows as the amount of bottleneck disjoint paths. We also develop a novel extension to MPTCP, which allows to estimate the number of subflows to open in order to fully utilize the network capacity.

For future work we have left to address some major limitations of this thesis, such as that the current approaches only consider locally available information. Although this is advantageous as it reduces overhead and complexity, it can create globally suboptimal solutions. More specifically, the aggregation algorithm in papers III and IV could consider the airtime limits [115]. To benefit from the increased aggregation capabilities in 802.11n, a short term goal is also to re-factorize the aggregation algorithm to provide higher aggregation possibilities on top of IEEE 802.11n.

The PathFinder algorithm in paper VIII currently does not remove subflows. If it overestimates the number of required subflows, these extra subflows persist until the end of the MPTCP connection. This increases the stability and avoids in the short time scale adverse interaction with MPTCP’s coupled
congestion control. However, for sufficiently long connections, these extra subflows should be terminated to save memory and computational resources in the end-nodes.

References


Transport-Layer Performance in Wireless Multi-Hop Networks


Introductory Summary


[38] IEEE. IEEE 802.3 - carrier sense multiple access with collision detection (CSMA/CD) access method and physical layer specifications. *IEEE Standards (IEEE-SA)*, June 2005.


Transport-Layer Performance in Wireless Multi-Hop Networks


49


