



Evaluating and Reducing Multipath Transport Latency

Kiran Yedugundla

Faculty of Health, Science and Technology

Computer Science

LICENTIATE THESIS | Karlstad University Studies | 2019:5

Evaluating and Reducing Multipath Transport Latency

Kiran Yedugundla

Evaluating and Reducing Multipath Transport Latency

Kiran Yedugundla

LICENTIATE THESIS

Karlstad University Studies | 2019:5

urn:nbn:se:kau:diva-71223

ISSN 1403-8099

ISBN 978-91-7867-001-7 (print)

ISBN 978-91-7867-006-2 (pdf)

© 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 2018

WWW.KAU.SE

Evaluating and Reducing Multipath Transport Latency

KIRAN YEDUGUNDLA

Department of Mathematics and Computer Science

Abstract

Access to the Internet is a very significant part of everyday life with increasing online services such as news delivery, banking, gaming, audio and high quality movies. Applications require different transport guarantees with some requiring higher bandwidth and others low latency. Upgrading access link capacity does not guarantee faster access to the Internet as it offers higher bandwidth but may not offer low latency. With increasing number of mobile devices supporting more than one access technologies (e.g., WLAN, 3G, 4G,...), there is a need to analyse the impact of using multiple such technologies at the same time. Legacy transport protocols such as TCP or SCTP are only able to connect to one access network at a time to create an end-to-end connection. When more than one access technology is used, there may be a large difference in the data rate offered by each technology. This asymmetry might impact latency sensitive applications by creating out of order delivery. In this thesis, we focus on the latency aspect of multipath transport protocol performance. We consider CMT-SCTP and Multipath TCP as available multipath protocols that were designed to exploit multiple paths for better throughput and reliability. We consider various real world traffic scenarios such as Video, Gaming and Web traffic to measure end-to-end latency. We perform simulations, emulations and experiments using heterogeneous network settings involving access networks with different bandwidth, delay and loss characteristics. MPTCP performs better in terms of latency than CMT-SCTP and TCP in certain scenarios where available paths are symmetric. However, MPTCP does not perform well in asymmetric scenarios with latency sensitive traffic. This analysis provides insights in to various areas of improvement in MPTCP such as scheduling and loss recovery to achieve low latency.

We further focus on packet loss recovery in MPTCP for specific cases of tail losses to reduce latency. Tail losses are the losses that occur at the end of a packet stream. Recovering such losses is of higher significance to latency sensitive applications. We propose a modification to the use of TLP, a mechanism in TCP for tail loss recovery. We evaluate the performance of proposed TLP modification, first using emulations and with real world network experiments. Our results show significant improvements in latency for specific loss scenarios in emulations and upto 50% improvement in experiments.

Keywords: Internet Latency, Transport Layer, Multipath TCP, Loss recovery, Tail Loss Probe, Performance Evaluation

Acknowledgements

Primarily, I would like to express my sincere gratitude to my research supervisor, Prof. Anna Brunstrom for providing me excellent research guidance and support. I have always been venerating her dedication towards research and her utmost concern for students. Her motivation to do qualitative research, inquest for perfection made my research work a wonderful learning experience. Secondly, I am thankful to my other supervisor Dr. Per Hurtig for his insightful suggestions and technical help in implementation aspects. I have always admired his exceptional programming talent and understanding of Linux kernel internals.

I would like to thank my examiner Prof. Andreas Kassler for providing valuable feedback on the thesis and constant encouragement. I would like to thank Dr. Reine Lundin, Dr. Karl-Johan Grinnemo and Dr. Stefan Alfredsson for helping me with initial steps at Karlstad University and with experimental setups.

I am also thankful to the members of DISCO group and Department of Computer Science for technical and non-technical discussions that helped me evolve and understand various topics. Thanks to the badminton group at Karlstad University that helped me be active.

Finally, a special thanks to my parents, my brother and sister for constantly encouraging me to pursue my goals and supporting me.

Karlstad, February 22, 2019

Kiran Yedugundla

Contents

List of Appended Papers	x
-------------------------	---

INTRODUCTORY SUMMARY	1
1 Introduction	3
2 Background	5
2.1 MPTCP	5
2.1.1 Path Management	5
2.1.2 Scheduling	6
2.1.3 Congestion Control	6
2.1.4 Loss Recovery	7
2.1.5 Mobility	8
2.2 MPTCP vs CMT-SCTP	8
3 Research Objective	8
4 Research Problems and Contributions	9
5 Research Methodology	10
6 Summary of Appended Papers	11
7 Conclusions and Future Work	12

PAPER I:	
Is Multipath Transport Suitable for Latency Sensitive Traffic?	19

1 Introduction	19
2 Multipath Transport	20
2.1 CMT-SCTP	20
2.2 MPTCP	21
2.3 Core Issues	22
2.3.1 Path Management	22
2.3.2 Scheduling	23
2.3.3 Congestion Control for Multipath Transport	25
2.3.4 Handling Loss and Retransmissions	26
3 Applications and their Requirements for Multipath Transport	27
3.1 Video Streaming	27
3.2 Gaming Traffic	28
3.3 Web Traffic	28

4	Experiment Setup	30
4.1	Evaluation Tool Sets	30
4.1.1	Simulations, CMT-SCTP using OMNeT++	31
4.1.2	Emulations, MPTCP in a Controlled Environment, using CORE	31
4.1.3	Experiments, MPTCP in a Real-Life Environment, using NorNet	31
4.2	Configuration of MPTCP and CMT-SCTP	32
4.3	Application Traffic Generation and Metrics	33
4.3.1	Video Traffic	33
4.3.2	MMO Gaming Traffic	33
4.3.3	Web Traffic	33
4.4	Background Traffic Generation	35
4.5	Network and System Characteristics	35
4.5.1	Topology	35
4.5.2	Path Characteristics	36
4.5.3	Buffer Sizes	37
5	Experiment Results	37
5.1	Video Streaming	38
5.1.1	CMT-SCTP Simulations	38
5.1.2	MPTCP Emulation	40
5.1.3	MPTCP Real Measurements	43
5.2	Gaming Traffic	44
5.2.1	CMT-SCTP	44
5.2.2	MPTCP Emulation	47
5.2.3	MPTCP Real Measurements	50
5.3	Web Traffic	51
5.3.1	CMT-SCTP Simulations	55
5.3.2	MPTCP Emulation	55
5.3.3	MPTCP Real Measurements	59
6	Discussion of Results	59
7	Related Work	62
8	Conclusions and Future Work	63

PAPER II:
Probe or Wait: Handling Tail Losses using Multipath TCP
75

1	Introduction	75
2	Background and Related Work	77

3	Experimental Setup	78
3.1	Testing Retransmission with Deterministic Loss Patterns . . .	78
4	Observations and Discussion	79
4.1	TCP	79
4.2	MPTCP	80
5	Improvements to TLP for MPTCP	84
6	Evaluation of Modified MPTCP TLP	85
7	Conclusion	85
	PAPER III:	
	Handling Packet Losses in Cloud-Based MPTCP Applica-	
	tion Traffic	91
1	Introduction	91
2	Background	92
3	Related Work	94
4	Handling Tail losses in MPTCP	94
5	Experiment Setup	95
6	Performance Evaluation of TLP variants	96
7	Conclusions	101

List of Appended Papers

1. Yedugundla V K, Ferlin S, Dreibholz T, Alay Ö, Kuhn N, Hurtig P, and Brunstrom A. Is Multipath Transport Suitable for Latency Sensitive Traffic?. *Computer Networks*, ISSN 1389-1286, E-ISSN 1872-7069, Vol. 105, p. 1-21,2016.
2. Kiran Yedugundla, Per Hurtig, and Anna Brunstrom. Probe or Wait: Handling Tail Losses using Multipath TCP. FIT Workshop, 2017 IFIP Networking Conference (IFIP Networking) and Workshops, 12-16 June 2017, Stockholm, Sweden.
3. Kiran Yedugundla, Per Hurtig, and Anna Brunstrom. Handling Packet Losses in Cloud-Based MPTCP Application Traffic. Accepted for publication in The 9th International Conference on Cloud Computing and Services Science, CLOSER 2019, Crete, Greece.

Comments on my Participation

Paper I The ideas have been developed in cooperation with my co-authors. I did the implementation, emulation and evaluation for the MPTCP part of the analysis and contributed in writing respective parts of sections: Multipath Transport, Experiment Setup, Experiment Results, Discussion of Results.

Paper II The ideas have been developed in cooperation with my co-authors. I did the writing, implementation, emulation and evaluation of proposed mechanism. Further improvements in text and analysis of results were made with suggestions and guidance from my co-authors.

Paper III The ideas have been developed in cooperation with my co-authors. I did the writing and experimentation with testbed. Further improvements in text and analysis were made with suggestions and guidance from my co-authors.

Other Publications

- **Kiran Yedugundla**, Per Hurtig, and Anna Brunstrom. Latency sensitive traffic performance using MPTCP. in Proceedings of *The 11th Swedish National Computer Networking Workshop (SNCNW)*, 2015.
- Per Hurtig, Ozgu Alay, Simone Ferlin and **Kiran Yedugundla**. Reducing Transport Latency using Multipath Protocols. in Proceedings of *EUCNC 2015 The European Conference on Communications: Special Session on Reducing Latency* 2015.

Introductory Summary



1 Introduction

Internet data traffic has seen significant growth during the last few years. Innovative technologies both in the data transmission and storage led to the rapid archiving and retrieving of huge volumes of data on the Internet. The traffic characteristics have changed over time along with the evolution of the Internet technologies with increasing content categories (e.g., documents, audio, video). Access to data is possible from a wide variety of devices ranging from fixed networked desktops to mobile devices. Multimedia content is a large part of the data accessed from mobile devices [7]. Often mobile devices support multiple technologies such as WLAN, and 3G/4G increasing the need to exploit the use of multiple networks simultaneously. This concept of using multiple network interfaces for a single connection is known as multi-homing. User demands can vary based on the applications with some requiring higher throughput and others lower latency. Interactive multimedia content became significant part of the Internet traffic that brought up more stringent latency requirements. Increasing capacity by using high bandwidth links do not necessarily reduce the latency at data flow level. Latency intensive interactive applications are not able to utilize the advantages of multiple technologies due to lack of support from underlying transport layer. To improve the data rate, applications can use multiple flows of Transmission Control Protocol (TCP) [22] or User Datagram Protocol (UDP) [21] on a single network technology. TCP allows binding to only one network address as an end-point in a connection. A major bottleneck to research in this direction is that any new transport protocol proposed should be supported by many devices in the Internet. The penetration of TCP in the Internet has been so high that applications inherently assume TCP or UDP as the transport layer. Any new transport protocol trying to utilize multiple networks must be within the TCP semantics or bypass the TCP with its own reliable transport layer to be successful. Concurrent multipath transfer (CMT) [12] using Stream Control Transmission Protocol (SCTP) [25] is an initial attempt to use multiple paths at the transport layer mainly for concurrent use of multiple paths.

In 2011, Multipath TCP (MPTCP) [10] was developed as an experimental standard in IETF as an extension of TCP. MPTCP was developed with specific goals to improve the robustness of transport layer by increasing the redundancy and resource utilization. Initial use-cases of MPTCP using multihomed devices showed significant improvement in throughput making it a widely acceptable approach to exploit multiple access networks simultaneously. MPTCP does not change the way applications perceive the transport layer (i.e., as a TCP flow) through application transparent design, which made MPTCP easily deployable. Application performance does not just come with high throughput, but also with low latency for many applications. Several delays in different layers of end to end networking add to the latency [4]. Even though, minimizing latency is not a design goal of MPTCP, in order to be acceptable to applications, latency performance of a flow using MPTCP makes a desirable study. It is an obvious curiosity to compare the latency performance of MPTCP to that of CMT and

TCP.

In this thesis, we focus on the latency perceived by data flows using multipath transport protocols that facilitate multi-homing. Data flows on the Internet are often from different applications and any analysis that assumes a single type of data may not be complete. Video traffic constitutes the largest percentage of traffic on the internet with live video growth predicted to increase 15-fold between 2016 and 2021 [7], and is therefore naturally part of our investigations. Live video is latency sensitive in nature with increasing use of live broadcast in online social media platforms and live video conference events. We consider interactive live gaming traffic as another traffic type to study the performance and compare multipath protocols along with video traffic. Web traffic is another latency sensitive traffic type used in this thesis considering the growth in online banking and authentication transactions that are often short flows in nature. In the initial part of the thesis, we compare the performance of three transport layer protocols with respect to latency for different traffic types, namely MPTCP, CMT-SCTP and TCP. With two protocols being multipath in nature, another dimension of path asymmetry comes in to focus in the analysis. Our results also indicate the effect of path asymmetry in flow latency for multipath transport protocols.

MPTCP latency depends on the flow characteristics as well as path asymmetry. Various parts of the MPTCP implementation use several methods to schedule packets on multiple paths, congestion control and handling losses. In the second part of the thesis, we focus on loss recovery in MPTCP especially for the special cases of tail losses. Tail losses are the losses at the end of packet flow and cause more delays for short flows. TCP has a mechanism to handle tail losses namely tail loss probe (TLP) [8]. In Linux MPTCP implementation, TLP is implemented per flow as a design decision. We provide a MPTCP level implementation that exploits multiple paths in handling tail losses. Emulations show an improvement in latency without adding much traffic as overhead. Further research presented in this thesis focus on this modified TLP for MPTCP which is less conservative in nature. Experiments with real traffic scenarios, using cloud based applications on a real WLAN and 4G network, show up to 50 percent improvements in latency for certain tail loss scenarios.

The rest of this introductory summary is structured as follows: Section 2 provides background to Multipath TCP along with challenges associated with improving protocol performance and comparison to CMT-SCTP. Section 3 outlines research objectives to provide scope of the thesis. Section 4 provides research problems and challenges inline with research objectives. Section 5 discusses the research methodology used in solving the research problems and achieving results. Section 6 provides short summaries of the appended papers. Section 7 concludes the introductory summary and outlines future research work.

2 Background

The idea of using multiple paths between endpoints was widely researched in the context of the network layer [17, 27] mainly for improving fault tolerance. The traffic carried between endpoints at the network layer may not always be from a single data stream. These data streams correspond to TCP flows at the transport layer. MPTCP allows a single data stream to be split across multiple paths. The basic principle behind MPTCP comes from pooling available network resources as described in [28], which was later imparted to the transport layer. Initial theoretical models [13] prove the feasibility of combined routing and rate control algorithms without compromising stability. These claims of flow stability was further strengthened in [11] and a congestion controller was proposed for multiple flows in [15], with a salient feature that it cannot be tuned to a single RTT. But these models did not find a noticeable practical use case until devices started rolling out with multiple simultaneously Internet connected interfaces. This use case was often referred to as a case of multihomed scenario with multiple interfaces.

2.1 MPTCP

The first practical implementation of MPTCP was published in [3] and the corresponding implementation on Linux was released during the same time. MPTCP was simultaneously published as an experimental standard published in IETF RFC 6182 [10]. MPTCP was developed with an objective of improving the robustness of the transport layer by increasing the redundancy and resource utilization. MPTCP implementations must meet two kinds of goals namely functional goals which steer services and features that MPTCP must provide and compatibility goals, which determine how MPTCP should appear to entities that interact with it. Many middleboxes can easily detect any deviations in TCP flow sequence due to wide usage of TCP in the Internet. To make MPTCP flows compatible to the existing middleboxes, MPTCP does not alter the sequence number mechanism of TCP. Instead MPTCP has a separate sequence numbering scheme associating a MPTCP packet to a TCP packet. TCP packet options field has MPTCP stream information so that a middlebox can assume the flow as a TCP flow even if the middlebox does not support MPTCP. But as the acceptance of MPTCP increases, middlebox vendors seem to have started supporting MPTCP by checking the options field of the TCP header for validation.

The throughput performance advantage of MPTCP comes along with certain challenges as described in the following subsections. The original specification of MPTCP does discuss and provide guidelines to solve the below mentioned challenges in accordance with MPTCP design principles.

2.1.1 Path Management

The functional goal of path management is to detect and use multiple paths between two hosts. MPTCP uses the presence of multiple IP addresses at

one or both of the hosts as an indicator of availability of multiple network interfaces. The path management features of the MPTCP protocol are the mechanisms to signal alternative addresses to hosts and mechanisms to set up new subflows joined to an existing MPTCP connection.

Path management decides the nature of interface announcement and creation of new subflow on the corresponding path. The proposed standard [10] suggests the use of stack internal algorithms that may implicitly try to self-optimize the behavior according to the application needs. The current Linux MPTCP implementation provides three different options for path management selection.

- *default*: A host will not announce different IP-addresses or initiate the creation of new subflows. It will accept subflow creation in passive mode.
- *fullmesh*: A host announces all available interfaces and supports subflow creation across all the paths in connected mesh.
- *ndiffports*: A host creates x subflows across same pair of endpoint IP addresses by modifying the source port. The number x can be set at runtime.

2.1.2 Scheduling

Schedulers play an important role in MPTCP on both ends due to the inherent complexity in choosing the right flow for the packet to be transmitted [19]. Packet scheduler breaks the byte stream received from the application into segments to be transmitted on one of the available subflows. The MPTCP design uses data sequence mapping to associate segments sent on different subflows to a connection-level sequence numbering thus allowing segments sent on different subflows to be correctly re-ordered at the receiver. The packet scheduler is dependent upon information about the availability of paths exposed by the path management component and then makes use of the subflows to transmit queued segments. In order to make scheduling decisions, a scheduler makes use of various information such as state of each TCP subflow, congestion window and RTT estimation.

2.1.3 Congestion Control

Congestion control is one aspect of transport layer that are well researched [1, 11] and yet provides opportunity for improvement due to variations in traffic affecting the performance of existing techniques. TCP uses the window-based congestion-control mechanism to adjust the transmission rate at the endpoints. Congestion control algorithms of TCP work at the subflow level of MPTCP if not coupled. Congestion control is an important part of the modifications that were proposed in MPTCP as the single path TCP congestion control algorithms have a set of issues in the multipath context. One of the prominent problems is that running existing algorithms such as standard TCP

independently on each path would give the multipath flow more than its fair share at a bottleneck link traversed by more than one of its subflows. In MPTCP, the congestion control is performed at the subflow level. Each subflow has its own congestion window. But the congestion windows of different subflows may be coupled to improve the performance.

All the algorithms developed [5, 14, 18, 29] for congestion control in MPTCP have common goals with little deviations. The design goals can be described as:

- Goal 1. Improve Throughput : Multipath flow must ensure at least the performance of a single path flow on the best available path.
- Goal 2. Fairness: Multipath flow should not take anymore capacity than a single path flow would take on a certain route.
- Goal 3. Balance Congestion : Multipath flow must send more traffic on its least congested routes while achieving the first two goals.

Current Linux version of MPTCP has three different congestion controls available: Coupled (LIA) [23], OLIA [16], and BALIA [20]. LIA is multipath variant of Reno congestion control algorithm used in single path TCP. OLIA and BALIA are improvements of LIA identifying certain conditions in which LIA performance is not optimal.

2.1.4 Loss Recovery

TCP has two main mechanisms for detecting and recovering from packet losses namely Fast Recovery (FR) and Retransmission Timeout (RTO). FR is quicker than RTO when there is data sent after a lost packet as it depends on duplicate ACKs. For long flows where there are sufficiently large number of packets to be transferred, FR is useful in detecting and recovering packet losses efficiently. Short flows are a majority in the web traffic which contain a few packets. A single packet loss in a short flow can be recovered by FR if the loss is in the middle of the flow. Otherwise, the loss recovery may take several RTTs as there are insufficient number of packets to send, to trigger the required number of duplicate ACKs. All the losses at the end of a packet flow whether it is long or short must depend on RTO to recover. For long flows, recovery with RTO might not affect the flow completion time due to the length of the flow, but for short flows the effect will be large.

TCP Loss Probe (TLP) [8] is a mechanism that allows flows to detect and recover from tail losses much faster than an RTO, thereby speeding up short transfers. With TLP a packet loss in the middle of a packet train as well as at the tail end will now trigger the same fast recovery mechanisms. It assumes other algorithms such as early retransmit [2] and FACK threshold based recovery are present.

In Multipath TCP [9], a connection can have multiple TCP subflows using different interfaces on different routes. Packet losses in each subflow are assumed to be detected and recovered in a similar fashion as that of TCP.

However, it is not completely clear how the loss recovery happens in the implementation and which subflow retransmits the lost packets. If the recovery is handled at the meta level, the lost packet may be rescheduled and retransmitted at the available subflow with lowest RTT. If the recovery is handled at the flow level, the packet may be retransmitted in the same subflow.

In the Linux implementation of MPTCP, a heuristic was used for loss recovery. The heuristic says that if the retransmission is a fast retransmit, then the same subflow is used for retransmission. If the retransmission is for a timeout, then the scheduler would re-evaluate the packet transmit options. In addition, the lost packet is always retransmitted on the original subflow as required by TCP.

2.1.5 Mobility

MPTCP RFC argues that host mobility can be addressed at the transport layer instead of at network layer with Mobile IP [24]. The best use case of MPTCP is the multihomed scenario with a mobile device connected to multiple wireless networks. Mobility is an important requirement for such scenario. Estimated values such as RTT within the working of the protocol often change with mobility. These estimates used in various algorithms of MPTCP such as scheduler might disturb the stability of the protocol performance [26].

2.2 MPTCP vs CMT-SCTP

Both MPTCP and CMT-SCTP provide multipath transport with resource pooling principle as the basis. CMT-SCTP was proposed earlier than MPTCP. Some of the drawbacks of CMT-SCTP towards acceptance by middleboxes like firewalls or Network Address Translation (NAT) with Port Address Translation (PAT) to work in the Internet are addressed in the design of MPTCP. Both protocols have different signalling mechanisms for establishing multihomed connections. MPTCP uses coupled congestion control where as CMT uses uncoupled congestion control based on SCTP implementation. Both protocols support various scheduling schemes based on RTT.

3 Research Objective

In this thesis, research mainly focuses on improving the Internet transport layer performance in terms of latency. Towards this objective, we study the multipath transport protocols to answer the following questions and contribute to identifying challenges for achieving low latency.

- (O1) How do multipath transport protocols perform in different traffic scenarios (Heterogeneous and Asymmetric)?
- (O2) What characteristics of multipath transport protocols affect the performance in these scenarios?

- (O3) How can the retransmission behavior of MPTCP be modified to improve latency for different traffic and loss scenarios?

4 Research Problems and Contributions

1. **Latency performance analysis of multipath protocols with various traffic scenarios:** For an increasing number of applications, latency plays an important role as it directly impacts their performance. Still, most work considering multi-path communication is solely focused on resilience and throughput maximization. The work presented in this thesis tries to bridge this gap by evaluating whether multi-path communication can help latency-sensitive applications satisfy their users' requirements. The first paper of the thesis addresses this issue examining three latency-sensitive applications: video, gaming and web traffic.

The results indicate that multi-path communication can reduce latency significantly, but only when paths are symmetric in terms of delay and loss rate. The potential gain comes mainly from two factors: the possibility to distribute short bursts of data over multiple interfaces and the ability to select the best of the available paths for data transmission. In asymmetric scenarios where the latency reduction is not as significant (or non-existent), applications may still benefit from other properties of multi-path communication, without increasing latency. This is, however, highly dependent on the scheduling mechanism used. This contribution addresses research objectives O1 and O2.

2. **Tail loss recovery in MPTCP: Improvements and Analysis:** Tail losses cause significant increase in flow latency. Effective loss recovery can reduce the latency to some extent. Through emulations and experimentation, we analyze the performance of existing techniques. We propose a less conservative approach to loss recovery in MPTCP. TLP is a loss recovery mechanism in TCP that allows tail losses to be recovered faster by using a probe timeout (PTO), which is smaller than an RTO. Current implementation of MPTCP uses TLP from TCP, but only triggers retransmission on an alternate path in the event of an RTO. Our approach improves upon the existing approach by triggering retransmission on an alternate path also in the event of probe timeout. Our emulation experiments using a modified Linux implementation, show that the proposed approach in fact improves the burst completion time in most cases and equals the existing implementation in other cases. The proposed approach improves the performance when the alternate path has lower delay than the original loss path and the advantage increases with degree of asymmetry between the paths. Temporary path failures can cause probe loss along with packet loss. The approach is very effective in case of probe loss which otherwise incur RTO timeout to trigger retransmission on either path. However, the latency improvement comes at a small cost in the form of additional retransmitted

packets if the packet is recovered on the primary path. In the case of primary path failure, there is no additional cost of using this mechanism. This contribution addresses research objective O3.

5 Research Methodology

Research methodology varies with the field of study. Computer science is a diverse area of research with its roots in mathematics and broadly classified as science with lots of engineering and technology involvement. The diversity is evident even in the naming of degrees in various educational institutes: some naming degree in science and engineering and some naming degree of technology in computer science. Influences of mathematics is visible in the theoretical models developed in computer science. Often the theoretical models are engineered to various requirements by adding additional constraints accordingly.

Modern computers are often seen as connected computing machines compared to the initial computers which merely does the work of modern day calculators. Connectivity allows the computers to share and process information at different locations. The chunks of information is called as packets and a packet switched network led to the modern Internet. The importance of Internet in everyday life and in the area of computer science motivated significant amount of research in computer science on the Internet and related technologies.

The research presented in this thesis is based on study of functional performance of computer networks. In networking, there are three ways of studying the functional performance, namely Simulation, Emulation and Experiments. This thesis uses all three methods of study. We use OMNET++ Simulator to perform simulations using CMT-SMTP protocol. Multipath TCP implementation is available in Linux operating system along with several proposed algorithms. We use CORE emulator to emulate various network conditions and evaluate latency performance.

Emulation and Experiments offer their own benefits and come with their own set of problems. Emulation offers near realworld results with less infrastructure and provides a controlled environment to assess the desired metrics. Measurements in testbed experiments provide results with several real world uncontrolled events factored in to the desired metrics. Emulation accuracy depends on the way it is implemented. There might be deviations from actual technology behavior as the noise in operational metrics such as delay is eliminated to most extent. Measurements in a testbed is expensive compared to emulations. Testbeds capture the real world technology behavior in the measurements. Several repetitions might be needed to reduce the noise in measurements. In this thesis, we needed to evaluate TLP for MPTCP and the proposed modification to improve latency. We started with controlled emulation experiments that provided a trend and then moved to validation in more realistic testbed experiments.

6 Summary of Appended Papers

Paper I – Is Multipath Transport Suitable for Latency Sensitive Traffic?

This paper assesses whether multipath communication can help latency-sensitive applications to satisfy the requirements of their users. This paper is a work towards the first two research objectives (O1 and O2). We consider CMT-SCTP and MPTCP and evaluate their proficiency in transporting video, gaming, and web traffic over combinations of WLAN and 3G interfaces. To ensure the validity of our evaluation, several experimental approaches were used including simulation, emulation and live experiments. When paths are symmetric in terms of capacity, delay and loss rate, we find that the experienced latency is significantly reduced, compared to using a single path. Using multiple asymmetric paths can increase latency depending on the degree of asymmetry, but might benefit from other advantages of multi-path communication. In the light of our conclusions, multi-path transport is suitable for latency-sensitive traffic and mature enough to be widely deployed.

Paper II – Probe or Wait: Handling Tail Losses using Multipath TCP

This paper is a work towards the third research objective (O3). Packet losses are known to affect the performance of latency sensitive applications in the Internet such as media streaming and gaming. Transport protocols recover from packet loss in order to provide reliable end to end communication and improving the quality of user experience. The efficiency of loss recovery greatly influences the completion time of flows. In this paper we focus on the state of the art loss recovery mechanisms for TCP and MPTCP. We use controlled tail loss scenarios to evaluate the performance of loss recovery mechanisms. Based on the observations, we propose an enhancement to the tail loss recovery in Multipath TCP to improve the loss recovery time. Our experiment results show consistent end to end latency performance improvement in considered scenarios.

Paper III – Handling Packet Losses in Cloud-Based MPTCP Application Traffic

This paper extends the work towards the third research objective (O3). With this research, we validate the research results of Paper II with realworld traffic from cloud applications on an experimental testbed. Internet traffic is comprised of data flows from various applications with unique traffic characteristics. For many cloud applications, end-to-end latency is a primary factor affecting the perceived user experience. As packet losses cause delays in the communication they impact user experience, making efficient handling of packet losses an important function of transport layer protocols. Efficient packet loss recovery is thus important to achieve desirable flow completion times for interactive cloud-based applications. In this paper we evaluate the performance of MPTCP in handling tail losses using traffic traces from various

cloud-based applications. We investigate the performance of TLP in MPTCP, comparing the standard implementation to the less conservative approach proposed in Paper II of this thesis. Our experimental results show that a less conservative implementation of TLP performs significantly better than the standard implementation in handling tail losses, reducing the average burst completion time of cloud based applications when tail loss occurs by upto 50 percent in certain cases.

7 Conclusions and Future Work

Research presented in this thesis represents an initial step towards understanding the latency performance of MPTCP for various traffic types. For the first time, the effect of path asymmetry on multipath protocol latency performance is well established with this research. Most of the prior work on multipath transport layer focused on throughput maximization. The work presented in this thesis evaluates whether multipath communication help latency sensitive applications satisfy their user requirements. By considering a variety of applications such as video, gaming and web, our analysis is based on realworld traffic.

Another important MPTCP challenge towards low latency is packet loss recovery. This thesis provides a method to handle tail losses in MPTCP flows by exploiting availability of multiple paths. Emulations with synthetic traffic and experiments with real world traffic provide insights in to the use of less conservative approaches in handling tail losses. The latency reduction achieved with the proposed method is significant for short flows.

Solutions to packet loss recovery and other challenges towards low latency are evolving continuously for TCP. We plan to evaluate the new TCP approaches such as Recent acknowledgement (RACK) [6] using MPTCP and suggest possible improvements in our future research work. Furthermore, there are technological changes in the access networks such as 5G. Path asymmetry might change to a different dimension with WLAN and 5G compared to WLAN and 3G/4G. Buffers play a significant role in future as the data rates of different access technologies converge. In this context, we plan to explore the effect of asymmetric buffer sizes and loss recovery in MPTCP for various loss scenarios.

References

- [1] A. Afanasyev, N. Tilley, P. Reiher, and L. Kleinrock. Host-to-Host Congestion Control for TCP. *Communications Surveys Tutorials, IEEE*, 12(3):304–342, Third 2010.
- [2] M. Allman, K. Avrachenkov, U. Ayesta, J. Blanton, and P. Hurtig. Early Retransmit for TCP and Stream Control Transmission Protocol (SCTP). RFC 5827 (Experimental), May 2010.

- [3] Sébastien Barré, Christoph Paasch, and Olivier Bonaventure. Multi-Path TCP: From Theory to Practice. In Jordi Domingo-Pascual, Pietro Manzoni, Sergio Palazzo, Ana Pont, and Caterina Scoglio, editors, *NETWORKING 2011*, pages 444–457, Berlin, Heidelberg, 2011. Springer Berlin Heidelberg.
- [4] B. Briscoe, A. Brunstrom, A. Petlund, D. Hayes, D. Ros, I. Tsang, S. Gjessing, G. Fairhurst, C. Griwodz, and M. Welzl. Reducing internet latency: A survey of techniques and their merits. *IEEE Communications Surveys Tutorials*, 18(3):2149–2196, thirdquarter 2016.
- [5] M. Handley C. Raiciu, D. Wischik. Practical congestion control for multipath transport protocols. Technical report, UCL Technical Report, 2010.
- [6] Yuchung Cheng and Neal Cardwell. RACK: a time-based fast loss detection algorithm for TCP. Internet-Draft draft-cheng-tcpm-rack-04, Internet Engineering Task Force, July 2016. Work in Progress.
- [7] Cisco and its affiliates. Cisco visual networking index: Global mobile data traffic forecast update, 2016–2021, 2017.
- [8] N. Dukkipati, N. Cardwell, Y. Cheng, and M. Mathis. Tail loss probe (tlp): An algorithm for fast recovery of tail losses. (Work In Progress) draft-dukkipati-tcpm-tcp-loss-probe-01.txt, February 2013.
- [9] A. Ford, C. Raiciu, M. Handley, and O. Bonaventure. TCP Extensions for Multipath Operation with Multiple Addresses. RFC 6824 (Experimental), January 2013.
- [10] Alan Ford, Costin Raiciu, Mark Handley, Sébastien Barré, and Janardhan R. Iyengar. Architectural Guidelines for Multipath TCP Development. Informational RFC 6182, IETF, March 2011. ISSN 2070-1721.
- [11] Huaizhong Han, Srinivas Shakkottai, C. V. Hollot, R. Srikant, and Don Towsley. Multi-path TCP: a joint congestion control and routing scheme to exploit path diversity in the Internet. *IEEE/ACM Transactions on Networking*, 14(6):1260–1271, December 2006.
- [12] Janardhan R. Iyengar, Paul D. Amer, and Randall R. Stewart. Concurrent Multipath Transfer using SCTP Multihoming over Independent End-to-End Paths. *IEEE/ACM Transactions on Networking*, 14(5):951–964, October 2006. ISSN 1063-6692.
- [13] Frank P. Kelly and Thomas Voice. Stability of end-to-end algorithms for joint routing and rate control. *Computer Communication Review*, 35(2):5–12, 2005.
- [14] Peter Key, Laurent Massoulié, and Don Towsley. Combined multipath routing and congestion control: a robust internet architecture. Technical Report MSR-TR-2005-111, Microsoft Research (MSR), August 2005.

- [15] Peter Key, Laurent Massoulié, and Don Towsley. Combining multipath routing and congestion control for robustness. In *CISS 2006, 40th Conference on Information Sciences and Systems*. IEEE Information Society, Institute of Electrical and Electronics Engineers, Inc., March 2006. <http://www.princeton.edu/optnet/>.
- [16] R. Khalili, N. Gast, M. Popovic, and J. Le Boudec. Mptcp is not pareto-optimal: Performance issues and a possible solution. *IEEE/ACM Transactions on Networking*, 21(5):1651–1665, Oct 2013.
- [17] Wenjing Lou and Yuguang Fang. A multipath routing approach for secure data delivery. In *2001 MILCOM Proceedings Communications for Network-Centric Operations: Creating the Information Force (Cat. No.01CH37277)*, volume 2, pages 1467–1473 vol.2, Oct 2001.
- [18] A. Ongena. Multipath congestion control. Technical Report Masters Thesis, Universite catholique de Louvain, 2009.
- [19] Christoph Paasch, Simone Ferlin, Özgü Alay, and Olivier Bonaventure. Experimental Evaluation of Multipath TCP Schedulers. In *Proceedings of the ACM SIGCOMM Workshop on Capacity Sharing Workshop (CSWS)*, pages 27–32, New York/U.S.A., 2014. ACM. ISBN 978-1-4503-2991-0.
- [20] Q. Peng, A. Walid, J. Hwang, and S. H. Low. Multipath tcp: Analysis, design, and implementation. *IEEE/ACM Transactions on Networking*, 24(1):596–609, Feb 2016.
- [21] Jonathan Bruce Postel. User Datagram Protocol. Standards Track RFC 768, IETF, August 1980. ISSN 2070-1721.
- [22] Jonathan Bruce Postel. Transmission Control Protocol. Standards Track RFC 793, IETF, September 1981. ISSN 2070-1721.
- [23] Costin Raiciu, Mark Handley, and Damon Wischik. Coupled Congestion Control for Multipath Transport Protocols. RFC 6356, IETF, October 2011. ISSN 2070-1721.
- [24] Costin Raiciu, Dragos Niculescu, Marcelo Bagnulo Braun, and Mark James Handley. Opportunistic Mobility with Multipath TCP. In *Proceedings of the Sixth International Workshop on MobiArch*, MobiArch '11, pages 7–12, Bethesda, Maryland/U.S.A., 2011. ACM. ISBN 978-1-4503-0740-6.
- [25] Randall R. Stewart. Stream Control Transmission Protocol. RFC 4960, IETF, September 2007. ISSN 2070-1721.
- [26] Yonghao Sun, Yong Cui, Wendong Wang, Tianze Ma, Yuri Ismailov, and Xin Zheng. Mobility support in multi-path tcp. In *Communication Software and Networks (ICCSN), 2011 IEEE 3rd International Conference on*, pages 195–199, May 2011.

- [27] A. Tsirigos and Z. J. Haas. Multipath routing in the presence of frequent topological changes. *IEEE Communications Magazine*, 39(11):132–138, Nov 2001.
- [28] Damon Wischik, Mark Handley, and Marcelo Bagnulo Braun. The resource pooling principle. *Computer Communication Review*, 38(5):47–52, 2008.
- [29] Damon Wischik, Costin Raiciu, Adam Greenhalgh, and Mark Handley. Design, Implementation and Evaluation of Congestion Control for Multipath TCP. In *Proceedings of the 8th USENIX Conference on Networked Systems Design and Implementation*, NSDI'11, pages 8–8, Berkeley, CA, USA, 2011. USENIX Association.



Evaluating and Reducing Multipath Transport Latency

With an increasing number of mobile devices supporting more than one access technologies (e.g., WLAN, 3G, 4G), there is a need to analyse the impact of using multiple such technologies at the same time. The inherent asymmetry among these technologies might affect latency sensitive applications by creating out of order delivery. In this thesis, we consider CMT-SCTP and Multipath TCP as available multipath protocols designed to exploit multiple paths for better throughput and reliability. We perform simulations, emulations and experiments using various real-world traffic scenarios such as Video, Gaming and Web traffic to measure end-to-end latency. MPTCP performs better in terms of latency than CMT-SCTP and TCP in certain scenarios where available paths are symmetric. This analysis provides insights into various areas of improvement in MPTCP such as scheduling and loss recovery to achieve low latency. We further focus on packet loss recovery in MPTCP for specific cases of tail losses (losses that occur at the end of a packet stream) to reduce latency. This thesis presents a modification to the use of Tail Loss Probe (TLP) in MPTCP that provides improvements in latency for specific loss scenarios in emulations and upto 50% improvement in experiments.

ISBN 978-91-7867-001-7 (print)

ISBN 978-91-7867-006-2 (pdf)

ISSN 1403-8099

LICENTIATE THESIS | Karlstad University Studies | 2019:5
